Simr Software Partners

Engineering Case Studies with 42 ISV codes in the Cloud

ADSCFD Ansys AUTODESK

DASSAULT SYSTEMES DYNA MORE CFDsupport

esi get it right® OpenFOAM FlowVision NUMECA INTERNATIONAL ROCKY

GROMACS FAST. FLEXIBLE. FREE. KUAVA WUFI®

OpenFOAM SIEMENS SIMSCALE . . . AND MORE

May 2024

https://www.simr.com

UberCloud is now SIMR

Welcome!

This Simr (formerly UberCloud) Compendium presents case studies about 42 projects based more than 42 different software packages from Independent Software Vendors (ISVs) describing on-boarding of technical computing applications to the High-Performance Computing (HPC) cloud. Like its predecessors between 2013 – 2020, this edition has been generously sponsored by **Hewlett Packard Enterprise and Intel**, and by our Media Sponsors **Digital Engineering DE247 and HPCwire**.

Simr is supporting an online community where engineers and scientists discover, try, and buy engineering Software as a Service, on demand. Engineers and scientists can explore and discuss with us how to operate and use HPC cloud to solve their demanding problems and identify roadblocks and solutions.

The goal of the former UberCloud Experiments was to perform engineering simulation experiments in the HPC cloud with real engineering applications in order to understand the roadblocks to success and how to overcome them. This Compendium is a way of sharing these results with our broader HPC, IT, and Engineering communities. Our efforts are paying off. Based on the experience gained over the past 12 years, we were able to increase the success rate of the individual experiments to 100%, as compared to 40% in 2013 and 60% in 2014, when we started these hands-on experiments.

As a major milestone in our company's history, in 2015, based on our experience gained from the previous cloud experiments, we introduced **UberCloud HPC software container technology**. Use of these containers by the teams shortened UberCloud Experiment times dramatically from an average of three months to just a few days. It also simplifies access, use, and control of HPC resources, on premise and remotely in the cloud. Essentially, users are working with a powerful remote desktop in the cloud that is easy and familiar to use. Users don't have to learn anything about HPC, nor about system architecture, nor about cloud for their projects. This approach inevitably led to the increased use of HPC for daily design and development, even for novice HPC users.

We also recognized **new applications** now benefiting from the power of the cloud, like Living Heart simulations or Artificial Intelligence based projects (Deep Learning, Machine Learning, Predictive Maintenance, or Natural Language Processing).

Last but not least, we just changed our company name from **UberCloud to Simr**. Our developments resulted in a software technology that provides engineers and scientists, at their fingertips, with a fully automated, secure, self-service platform that is **SIM**ulation-**R**eady.

At the occasion of our latest milestone, May 21 2024, we publish this collection of UberCloud/Simr case studies, as a THANK YOU to all our Independent Software Partners who supported us over the last 12 years by providing free test licenses, support, and last but not least, their friendship. Without them we, and our HPC and Engineering Community, wouldn't be where we are now!

Wolfgang Gentzsch and Burak Yenier
The UberCloud, Los Altos, CA, May 2024

PS: The following case studies have been the result of real hands-on projects with UberCloud customers in the past 10 years. Therefore, they still (correctly) refer to UberCloud (instead of Simr).

The UberCloud Experiment Sponsors

We are very grateful to our Compendium sponsors **Hewlett Packard Enterprise and Intel**, our Primary Media Sponsor **Digital Engineering**, and to our sponsors Advania Data Centers, ANSYS, Microsoft Azure, Dassault Systemes SIMULIA, and Opin Kerfi. Their sponsorship allows for building a sustainable and reliable Simr (formerly UberCloud) community platform:

Big Thanks also to our media sponsors HPCwire, Digital Engineering, Bio-IT World, scientific computing world, insideHPC, and Primeur Magazine for the widest distribution of this UberCloud Compendium of case studies in the cloud:

Thank You!

Table of Contents

AERODYNAMICS SOLUTIONS CFD

Fast and Cost-Effective Compressor Map Generation Using Cloud-Based CFD

> *"A compressor map of 39 operating points was obtained in approximately 2.5 hours at a total cost of $580 in the cloud."*

MEET THE TEAM

End-user: Anonymous independent consultant that does not have the interest or the financial capability to invest in large clusters and CFD licenses.
Software Provider: Michael Ni – Product Manager - AeroDynamic Solutions Inc. - a provider of CFD software and consulting services.
Resource Provider: AWS Amazon Web Services.

USE CASE

When analyzing or designing centrifugal compressors, understanding the performance characteristics of the compressor is a significant challenge for any individual or small business without the hardware and software computing resources of their larger industry counterparts. Many complex simulations must be performed, and often the individual must tradeoff between turnaround time (and a reduced number of design iterations) and software and hardware expense.

The end-user in this case study was interested in understanding the effectiveness of cloud-based CFD for analyzing the NASA CC3 compressor. The NASA CC3 consists of an impeller with a splitter and a vaned diffuser and is representative of a typical centrifugal compressor configuration.

END TO END PROCESS

Cloud Leo is a cloud-based turbomachinery CFD simulation service based on the RANS solver Code Leo and designed to take advantage of Amazon Web Services. Designed for compressor and turbine aerodynamic design, Cloud Leo allows users to provision CFD analysis capacity on demand and use it on a pay-as-you-go basis. Cloud Leo runs on cloud computing infrastructure provided by Amazon Web Services (AWS). AWS has many years of experience in designing large data centers and has comprehensive policies in place to ensure physical security, secure services and data privacy. AWS is SAS 70 Type II certified, ISO 27001 certified and PCI compliant. Cloud Leo simplifies the process by automatically provisioning instances, loading the appropriate AMI, and configuring the firewall.

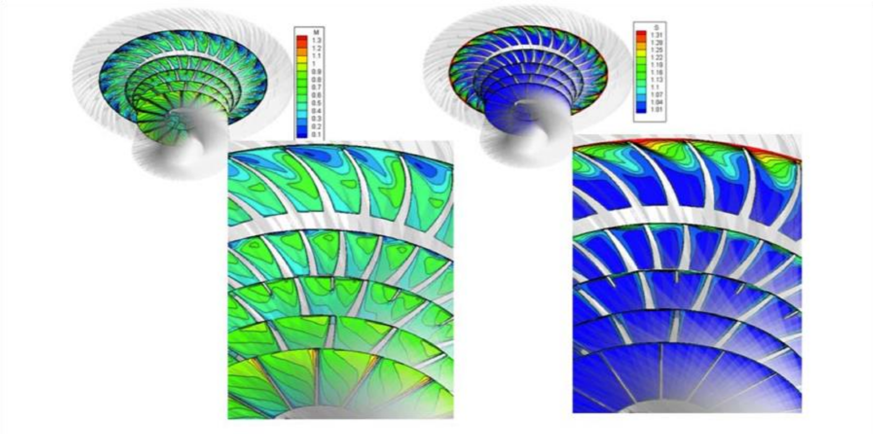


Figure 1: Mach Number and Entropy Contours for the NASA CC3 simulation.

The service is accessed through a graphical workbench running on a local computer. The workbench enables cases to be configured locally—including mesh generation with Code Wand—and transferred securely to specially developed AWS AMIs for cloud-based execution. The workbench contains a queuing and batching system to insure that cloud resources are not over-subscribed and are utilized efficiently. As execution occurs, the status of the simulation can be monitored within the workbench. When cases complete, results are downloaded securely back to the workbench for post- processing and permanently deleted from the AWS instance as an added measure of security.

A user begins by simulating an operating point at the design point condition for each of the 5 operating conditions. The initial points are then examined for convergence prior to generating the rest of the map. Once validated, the workbench generates 39 operating points configured to run in parallel on 15 cc2.8xlarge AWS EC2 instances. These instances are equivalent to 2 Intel Xeon ES-2670 processors with 16 cores, 60GB RAM, and interconnected via 10Gb Ethernet. The simulations completed in 2.5 hours including transfer of data to and from the instances.

CHALLENGES

There were many challenges in making the service seamless to the end-user. To begin, an intensive study of the effects of various AWS instance types was done. It was found that outside of the compute cluster instances, the network delays of the less expensive virtual instances were so large that it made using them impossible for any significant computations.

Another significant challenge was in developing the user experience so that it was intuitive and simple. The target end-users were not expected to understand IT infrastructure, networking, securing data over the Internet, or resource load balancing. This challenge was a significant source of development effort as it was built to be resource provider agnostic.

Lastly, architecting and implementing the licensing structure such that it was truly charged per hour and secure against fraud was a challenge. Our licensing technology had to be re-implemented to handle metered simulations as nothing existed that could be integrated with our software.

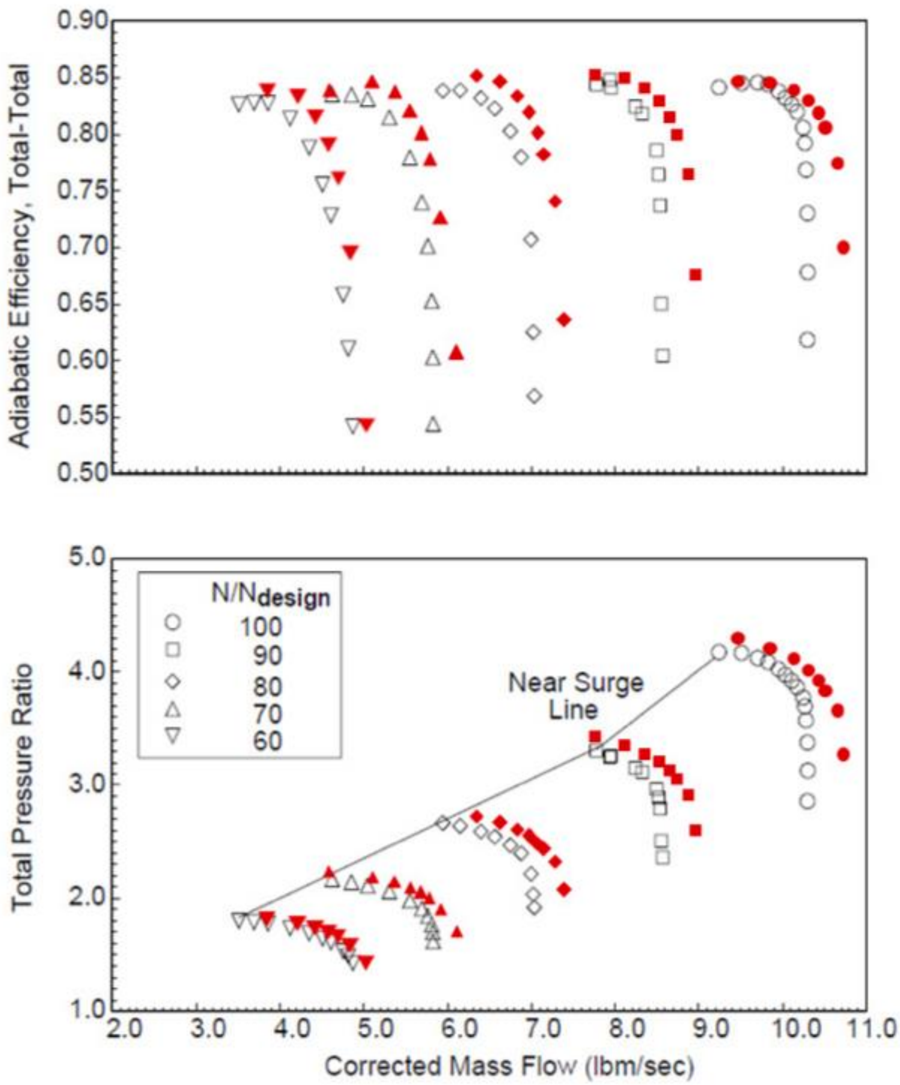


Figure 2: CFD Results compared against experimental data.

BENEFITS

The end-user benefitted from the team collaboration and knowledge gained from this project. These include:

- The costs of demanding CFD simulations will not exceed their budget.
- The user can now scale CFD related expenditures as demand rises, rather than before demand exists.
- The user can apply CFD technologies that were not previously available to them.
- The user can turn around CFD design and analysis in the same design windows as their larger competitors.
- The user can now size their CFD IT expenditures to their typical simulation load rather than sizing to their peak load.

Case Study Author – Michael Ni

ANSYS CFX

Wind Turbine Aerodynamics with UberCloud ANSYS Container in the Cloud

"The HPC cloud provided a service to solve very fine mesh models and thus reduced the simulation time drastically."

MEET THE TEAM
End-User/CFD Expert – Praveen Bhat, Technology Consultant, INDIA
Software Provider – ANSYS, Inc. and UberCloud Container
Resource Provider – ProfitBricks

USE CASE
With an ever-increasing energy crisis occurring in the world, it is important to investigate alternative methods of generating power other than fossil fuels. Wind energy is an abundant resource in comparison with other renewable resources. Moreover, unlike solar energy, its use is not affected by climate and weather. A wind turbine is a device that extracts energy from the wind and converts it into electric power.

This case study describes the evaluation of the wind turbine performance using a Computational Fluid Dynamics (CFD) approach. Standard wind turbine designs were considered for this UberCloud experiment. The CFD models were generated with ANSYS CFX. The simulation platform was built on a 62-core 240 GB HPC cloud server at ProfitBricks, the largest instance at ProfitBricks. The cloud environment was accessed using a VNC viewer through a web browser. The CPU and the RAM were dedicated to the single user. The ANSYS software ran in UberCloud's new application containers.

Process Overview
The following defines the step-by-step approach in setting up the CFD model in the ANSYS Workbench 15.0 environment.

1. Import the standard wind turbine designs, which are in the 3D CAD geometry format. These were imported into the ANSYS Design modeler. The model was modified by creating the atmospheric air volume around the wind turbine design.
2. Develop the CFD model with an atmospheric air volume surrounding the wind turbine in ANSYS Mesh Modeler.
3. Import the CFD model in the ANSYS CFX Computational Environment.
4. Define the model parameters, fluid properties, and boundary conditions.

5 Define the solver setup and solution algorithm. This portion of setup was mainly related to defining the type of solver, convergence criteria, and equations to be considered for solving the aerodynamic simulation.
6 Perform the CFD analysis and review the results.

The simulation model needed to be precisely defined with a large amount of fine mesh elements around the turbine blade geometry. The following snapshot highlights the wind turbine geometry considered and ANSYS CFX mesh models.

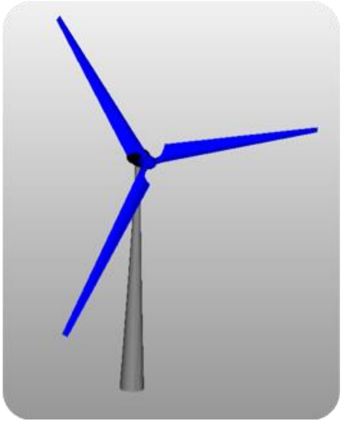


Figure 1: Wind turbine Geometry

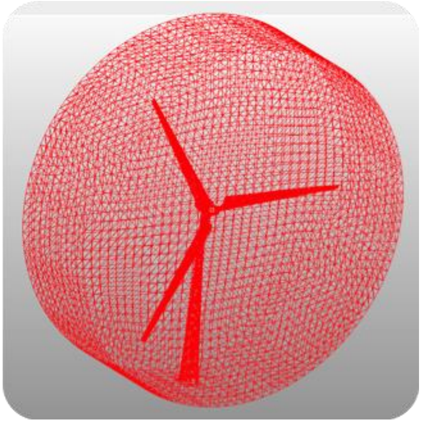


Figure 2: CFD model of wind turbine

The CFD simulation evaluated the pressure distribution and velocity profiles around the wind turbine blades. The wind turbine blades are subjected to average wind speed of 7 to 8 m/min. The following plots highlight the pressure and velocity distribution around the wind turbine blades.

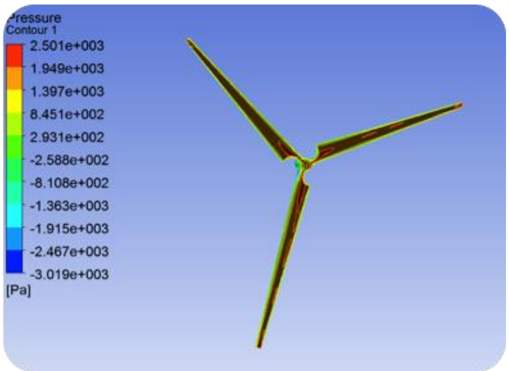


Figure *3*: Plot of pressure distribution on the wind turbine blades

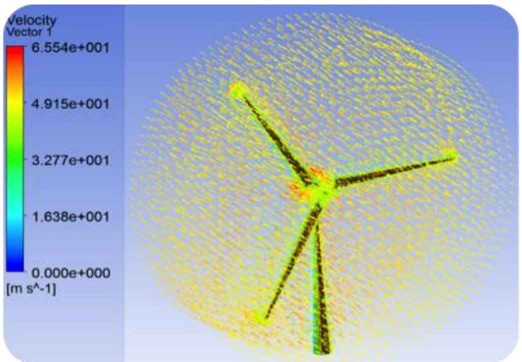


Figure *4*: Vector plot of velocity profiles around the wind turbine blades

HPC Performance Benchmarking

The aerodynamic study of wind turbine blades was carried out in the HPC environment that is built on a 62-core server with a CentOS Operating System and an ANSYS Workbench 15.0 simulation package. Server performance is evaluated by submitting simulation runs for different parallel computing environments and mesh densities. The simulation runs were performed using ANSYS CFX by varying the mesh densities and submitting the jobs for different numbers of CPU cores. Three different parallel computing environments were evaluated: Platform MPI, Intel MPI and PVM Parallel. Each of the parallel computing platforms was evaluated for their performance on total compute time and successful completion of the submitted jobs. Further the best parallel computing environment is proposed based on the experiments conducted and results achieved.

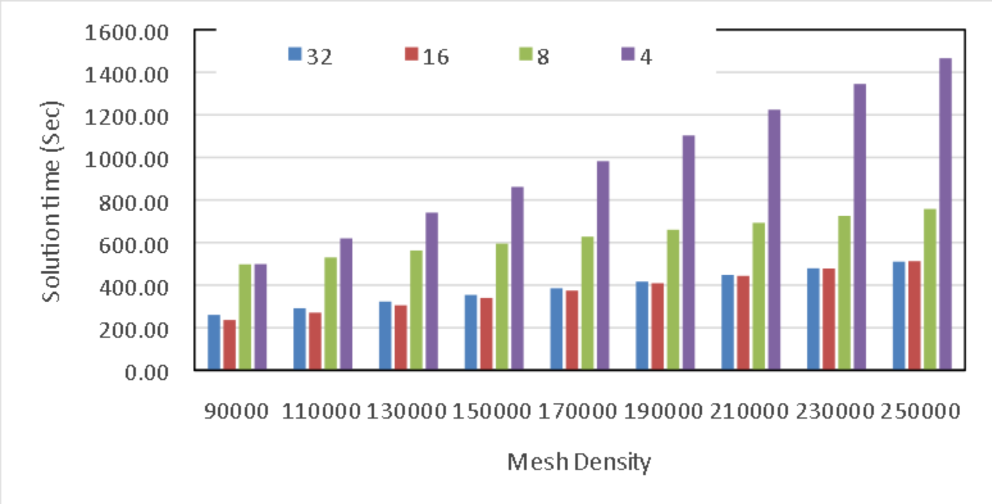


Figure 5: Solution time for different element density using Intel MPI

Figures 5 shows the solution time required for different mesh density where the simulation models are solved using Intel MPI. The Intel MPI parallel computing platform shows a stable performance with the solution time decreasing with increases in the number of CPU cores (Ref. Figure 5).

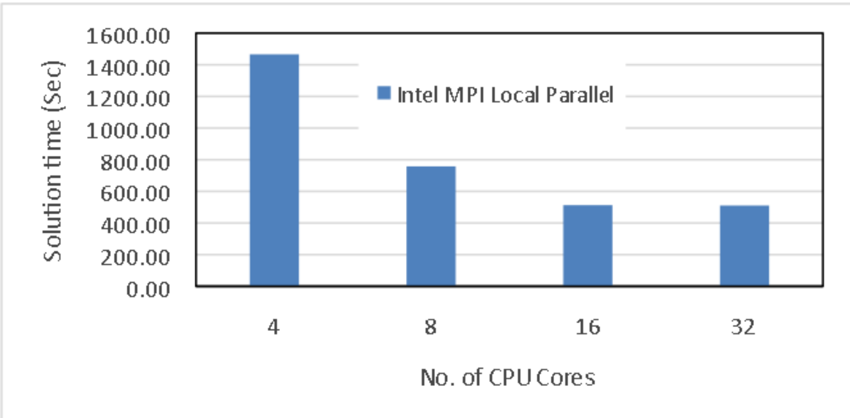


Figure 6: Performance comparison for a mesh density of 250K

Effort Invested

End user/Team expert: 75 hours for simulation setup, technical support, reporting and overall management of the project.

UberCloud support: 16 hours for monitoring & administration of host servers and guest containers, managing container images (building and installing container images during any modifications/bug fixes) and improvements (such as tuning memory parameters, configuring Linux libraries, usability enhancements). Most of this effort is one time occurrence and will benefit future users.
Resources: ~600 core hours were used for performing various iterations in the simulation experiments**.**

CHALLENGES

The project started with setting up the ANSYS 15.0 workbench environment with ANSYS CFX modeling software on the 62-core server. Initial working of the application was evaluated, and the challenges faced during the execution were highlighted. Once the server performance was enhanced, the next set of challenges faced was related to technical complexity. This involved accurate prediction of wind turbine blade behavior under aerodynamic loads, which is achieved through defining appropriate element size for the mesh model. The finer the mesh the higher the simulation time required; therefore the challenge was to perform the simulation within the stipulated timeline.

BENEFITS

1 The HPC cloud environment with ANSYS 15.0 Workbench made the process of model generation easier with process time reduced drastically because of the use of the HPC resource.
2 The mesh models were generated for different cell numbers where the experiments were performed using coarse-to-fine to highly fine mesh models. The HPC computing resource helped in achieving smoother completion of the simulation runs without re-trials or resubmission of the same simulation runs.
3 The computation requirement for a very fine mesh (2.5 million cells) is high, which is next to impossible to achieve on a normal workstation. The HPC cloud provided the ability to solve very fine mesh models and drastically reduce simulation time drastically. This allowed us to obtain simulation results within an acceptable run time (~1.5 hours).
4 The use of ANSYS Workbench helped in performing different iterations in the experiments by varying the simulation models within the workbench environment. This further helped to increase the productivity of the simulation setup effort and provided a single platform to perform the end-to-end simulation setup.
5 The experiments performed in the HPC Cloud environment showed the possibility and provided the extra confidence required to setup and run the simulations remotely in the cloud. The different simulation setup tools required were installed in the HPC environment and this enabled the user to access the tool without any prior installations.
6 With the use of VNC Controls in the web browser, the HPC Cloud access was very easy with minimal or no installation of any pre-requisite software. The whole user experience was similar to accessing a website through the browser.
7 The UberCloud containers helped with smoother execution of the project with easy access to the server resources, and the regular UberCloud auto-update module through email provided huge advantage to the user by enabling continuous monitoring of the job in progress without any requirement to login to the server and check the status.

CONCLUSION AND RECOMMENDATIONS

1. The selected HPC Cloud environment with UberCloud containerized ANSYS on ProfitBricks cloud resources was a very good fit for performing advanced computational experiments that involve high technical challenges and require higher hardware resources to perform the simulation experiments.

2. There are different high-end software applications that can be used to perform wind turbine aerodynamics study. ANSYS 15.0 Workbench environment helped us to solve this problem with minimal effort in setting up the model and performing the simulation trials.
3. The combination of HPC Cloud, UberCloud Containers, and ANSYS 15.0 Workbench helped in speeding up the simulation trials and also completed the project within the stipulated time frame.

Case Study Author – Praveen Bhat

ANSYS Discovery Live

Ninth Graders Designed Flying Boats in in the Azure Cloud

"The students learned the ANSYS software extremely fast. It was really inspiring to see the many concepts they came up with."
"To see such a spirit among the students was the most inspiring moment in the entire project."

MEET THE TEAM
End-Users: 25 9th-grade students from Torsdad Middle School in Sandvika, Norway
Software Provider: Scott Gilmore, ANSYS Director of Business Development, with ANSYS Discovery Live
Cloud Provider: Microsoft Azure and the UberCloud
HPC Expert and Service Provider: Ender Guler and Ronald Zilkovski, UberCloud.

USE CASE
This UberCloud project #208 has been collaboratively performed by a class of 25 9th-grade students from Torsdad Middle School in Sandvika, Norway. Help was also given by their physics teacher Ole Nordhaug, engineer Håkon Bull Hove from ANSYS channel partner EDRMedeso, and HPC Cloud service provider UberCloud. They implemented ANSYS Discovery Live on Microsoft Azure NV6 compute instances, each equipped with an NVIDIA Tesla M60 GPU.

The students used ANSYS Discovery Live, released in the first quarter of 2018, which provides instantaneous 3D simulation, tightly coupled with direct geometry modeling, to enable interactive design exploration and rapid product innovation. It is an interactive, multiple physics simulation environment in which users can manipulate geometry, material types, or physics inputs, and instantaneously see changes in performance. It allows users to test more design iterations in a much shorter amount of time, perform feasibility studies on new concepts and bring products to market faster. It is a great tool for CAD engineers who can check the physics of their different designs on demand.

Hired by fictional company "FlyBoat"
Through a course called "Research in Practice" the students got a taste of the work of an engineer. They were told that a fictional company called "FlyBoat" was planning to develop a combination of a boat and a seaplane. FlyBoat had hired them to do a concept study. Within some given parameters, like a predefined height, width and length, the students were free to innovate. Eventually, FlyBoat would pick one of the concepts for their design. To win the competition, the students had to come up with the best overall design, which required attention to many different aspects of boat and

plane design. At the end of the semester, all the boats were 3D-printed, and the students held sales presentations to convince a panel that their concept was indeed the best.

Figure 2: The winners from Torsdad Middle School in Sandvika, Norway, and their flying boat.

3D MODELLING IN ANSYS DISCOVERY LIVE

The students designed a 3D model of the boat in ANSYS Discovery Live. None of the students had prior experience with CAD modelling nor simulations, but they took the challenge head-on. *"The students learned the software impressively quickly. It was really inspiring to see what concepts they came up with",* says Håkon Bull Hove, engineer at EDRMedeso. Together with teacher Ole Nordhaug, he demonstrated Discovery Live to the students and helped them with their simulations.

Perhaps the most challenging task of the project was to prove that the flying boats could indeed fly. To do this, the students performed aerodynamic analyses of the wing in ANSYS Discovery Live, examining both lift and drag. Their simulations did not only prove that the wings had enough lift, but provided excellent visualization of the physics as well.

"It is much easier to understand foil theory when you see it live on your screen," says enthusiastic teacher Ole Nordhaug. *"To see such a spirit among the students was the most inspiring moment in the entire project."*

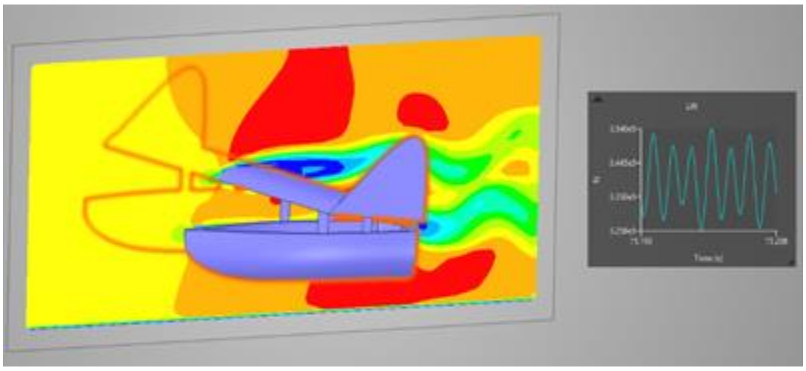


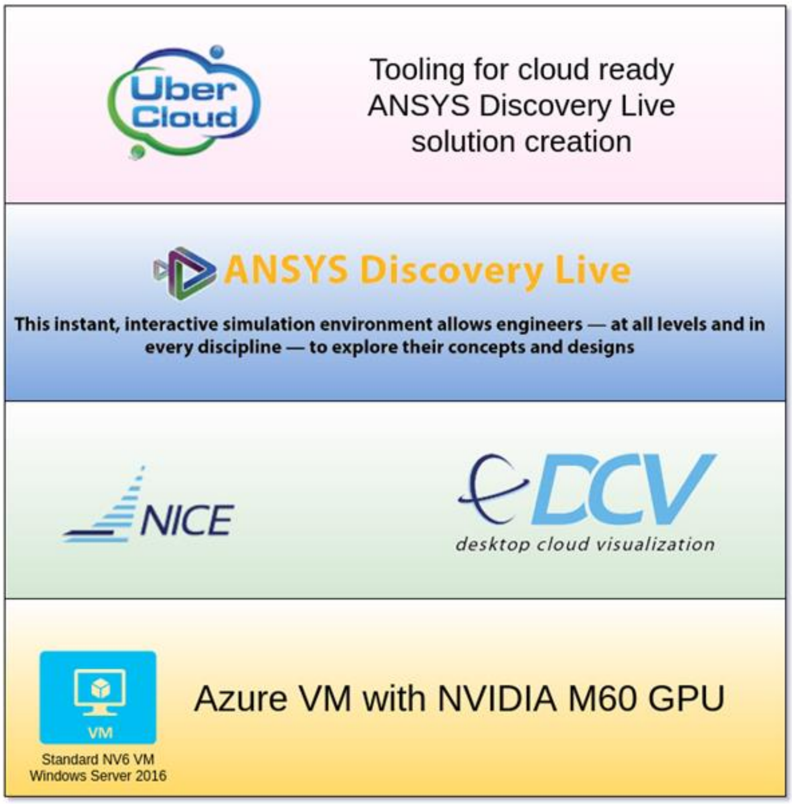
Figure 3: Students model their designs and use simulations to prove their boats can fly.

USING UBERCLOUD ON MICROSOFT AZURE

During the whole course of the design and simulation project – from March to June this year – the students have been supported by UberCloud which provided – every Wednesday – ten ANSYS Discovery Live environments sitting on ten Azure NV6 Windows compute nodes, each equipped with 6 Intel Xeon E5 compute cores, 56 GB, and an NVIDIA Tesla M60 GPU for accelerating compute and real-time remote visualization. Cloud resources were located in Microsoft's Azure datacenter in Amsterdam which the students accessed instantly, with login and password, through their web browser.

At the beginning of the project, data traffic between the cloud and the students' low-end Chromebooks was quite slow. To further speed up remote visualization during students' interactive work, NICE DCV software has been used for accelerating the rendering of visual elements and sending the rendered elements to the students' web browser as compressed data. With this technology the network bandwidth and the overall user experience was very satisfactory, even with the low screen resolution of the students' Chromebooks.

UberCloud's Cloud Ready ANSYS Discovery Live Solution



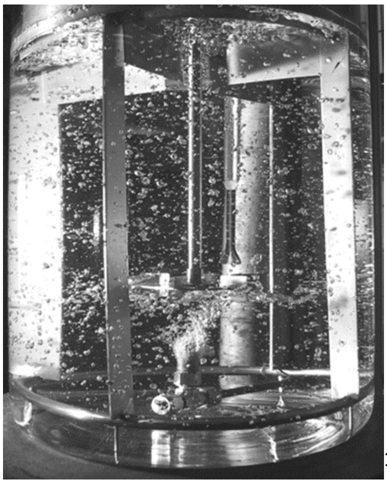
CONCLUSION

This middle school project with 14-year old students and their physics teacher in Norway is just another impressive demonstration of the current trend towards more user-friendly application software, combined with extremely fast HPC Cloud infrastructure (equipped with GPUs) available for everyone at their fingertips; a big step forward towards "democratizing" high performance computing and engineering simulation.

Case Study Author – Håkon Bull Hove and Wolfgang Gentzsch

ANSYS Fluent

Establishing the Design Space of a Sparged Bioreactor on Microsoft Azure


[1])

"The combination of Microsoft Azure with UberCloud ANSYS FLUENT Container provided a strong platform to develop an accurate virtual simulation model that involved complex multi-phase flow and tank geometries."

MEET THE TEAM

End-User/CFD Expert: Sravan Kumar Nallamothu, Sr. Application Engineer, and Marc Horner, PhD, Technical Lead, Healthcare, ANSYS, Inc.
Software Provider: ANSYS, Inc. and UberCloud Fluent Container
Resource Provider: Microsoft Azure
HPC Expert: Shitalkumar Joshi, ANSYS, and Wolfgang Gentzsch, UberCloud.

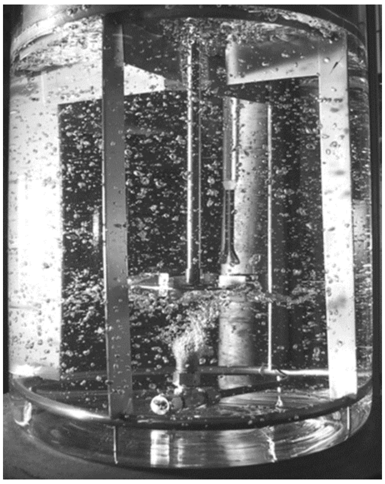
USE CASE

The scale-up of pharmaceutical laboratory mixers to a production tank is not a trivial task as it requires a thorough understanding of complex turbulent and multiphase processes impacting oxygen mass transfer. The interplay between the geometric design of the tank and tank operating parameters are critical to achieving good mixing, esp. at (larger) production scales. In an effort to improve process understanding, international regulators suggest a Quality by Design (QbD) approach to process development and process control. In the Quality by Design (QbD) framework, significant emphasis is placed on the robust characterization of the manufacturing processes by identifying the engineering design space that ensures product quality. There are various geometry and operating parameters influencing oxygen mass transfer scale-up from lab scale to production scale. Understanding the effect of these parameters can lead to robust design and optimization of bioreactor processes.

The main objective of this study is to understand the impact of agitation speed and gas flow rate on the gas holdup and mass transfer coefficient, which are two critical parameters that help process engineers understand mass transfer performance. The general-purpose CFD tool ANSYS Fluent is used for the simulations and the simulation framework is developed and executed on Azure Cloud resources running the ANSYS Fluent UberCloud container. This solution provided a scalable platform for achieving sufficient accuracy while optimizing the solution time and resource utilization.

[1] Picture from Marko Laakkonen (reference see next page)

PROCESS OVERVIEW



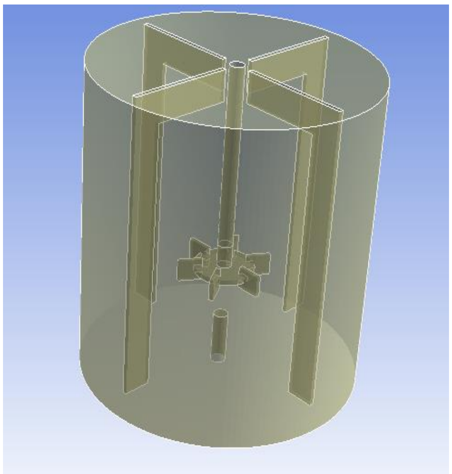


Figure 1: 194L Tank used for experiments[1] and representative CFD Model.

The stirred tank is agitated by a 6-bladed Rushton turbine blade for dispersing the air bubbles generated by the sparger. Four custom baffles are included to prevent vortex formation. Experimental conditions and results are taken from the extensive study performed by Laakkonen[2].

A full 3D model of the 194 L tank is considered for this CFD study, which is meshed using polyhedral elements. The Eulerian multiphase model is used for simulating the two phases: water and air. The population balance model with quadrature method of moments (QMOM) is used to simulate bubble coalescence and breakup processes. The Ishii-Zuber drag model is used to account for momentum exchange between water and air bubbles. For bubble coalescence, a model based on the Coulaloglou-Tavlarides model is used and the breakup model is based on the work of Laakkonen. It was observed that non-drag forces did not significantly impact gas holdup and mass transfer. A zero-shear boundary condition was applied for the water phase at the upper free surface, and a degassing boundary condition is used to remove the air bubbles.

The steady-state solver is used for running the simulations. Each simulation is solved until gas holdup and mass transfer coefficient reach steady values. The mass transfer coefficient is calculated using a custom field function, formulated based on a correlation derived from penetration theory[3]. A volume-averaged mass transfer coefficient is defined as an output parameter of the simulations to facilitate comparison of the various process conditions. Specifically, a design of experiments (DOE) study is performed with agitation speed and gas flow rate as input parameters and volume-averaged mass transfer coefficient as the output parameter. ANSYS Workbench with DesignXplorer is used to run the DOE and study the bioreactor design space.

RESULTS

As shown in Figure 2, air bubbles undergo breakup near the impeller blades and coalesce in the circulation regions with low turbulent dissipation rates. This leads to bubble size varying throughout

[1,2] Marko Laakkonen, Development and validation of mass transfer models for the design of agitated gas-liquid reactors, https://pdfs.semanticscholar.org/c6bd/d98a364a73fecb84468da9352659e475344d.pdf

[3] J.C. Lamont, D. S. Scott, An eddy cell model of mass transfer into the surface of a turbulent liquid, AIChE J. 16 (1970) 513-519

the tank. Since interfacial area depends on bubble size, bubble size distribution plays a critical role in oxygen mass transfer.

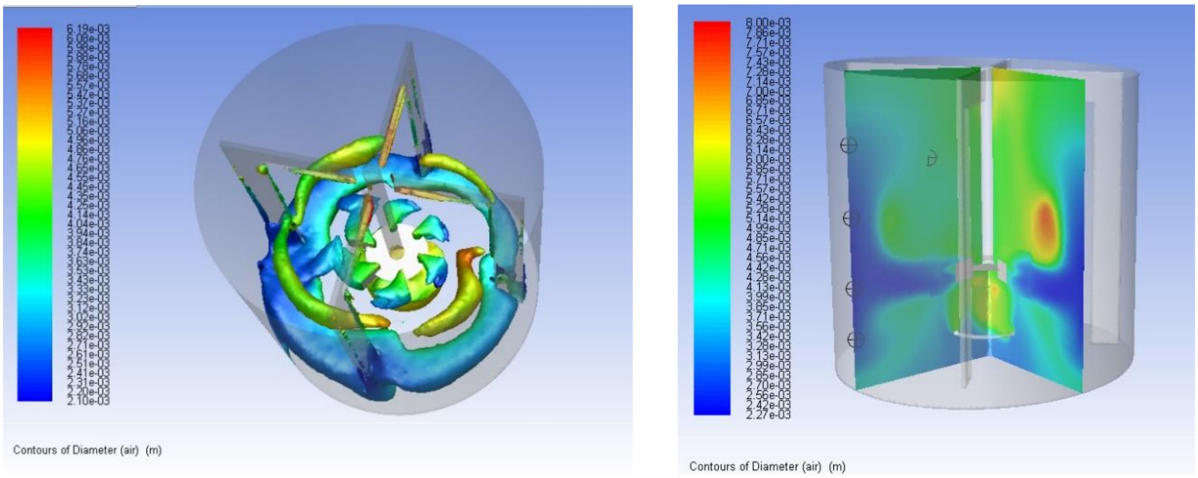


Figure 2: a) Iso-surface of gas volume fraction colored with bubble diameter b) Contour plot of bubble size distribution.

To study the design space of the bioreactor, a DOE study has been performed to generate the response surface for the average mass transfer coefficient. From the response surface shown in Figure 3, we can see that agitation speed has a greater impact on the mass transfer coefficient versus gas flow rate. Even though we can increase the agitation speed to increase the mass transfer coefficient, there is a limit on maximum speed since some processes involve mammalian cells that are sensitive to hydrodynamic shear.

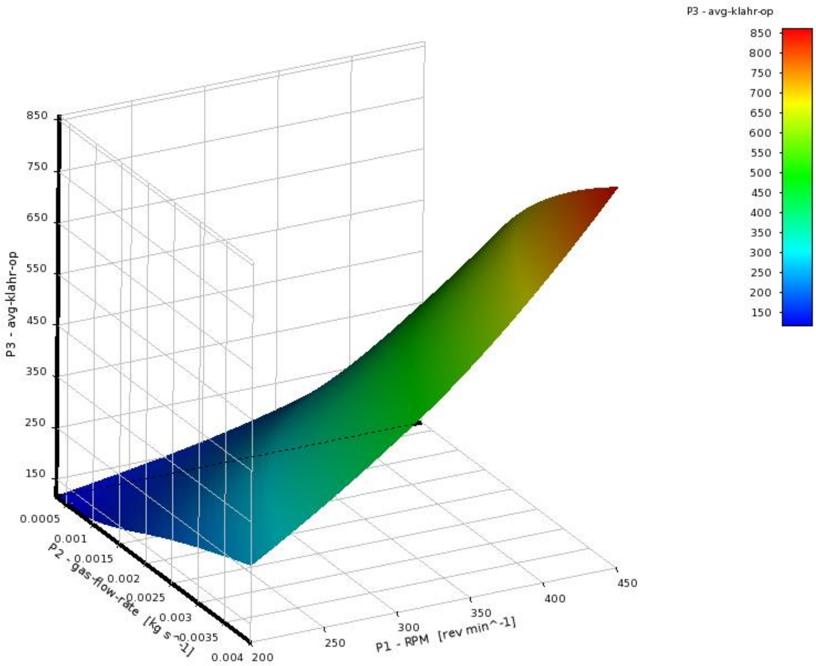


Figure 3: Response surface of average mass transfer coefficient versus gas flow rate and agitation speed.

Therefore, studying the design space with several input parameters provides an opportunity to optimize the operating conditions to identify a safe operational range for the bioreactor.

HPC PERFORMANCE BENCHMARKING

We used cloud resources in Microsoft's Singapore data center because this is relatively close to the ANSYS office in Pune, India. The experiment start date was: 2017-12-27, and experiment finish date was: 2018-01-30. Simulations started on 1 node (16 cores) and the last run was on 16 nodes (256 cores). Instance node type: Standard_H16r; FDR InfiniBand (56 Gbps bandwidth); Azure compute instances: 16 CPU cores (Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz), 112 GB of memory.

The software used to simulate the gas sparging process is ANSYS Workbench with FLUENT in an UberCloud HPC container integrated with the Microsoft Azure cloud platform. The solution methodology is tested with fine and coarse tetrahedral and polyhedral meshes. The time required for solving the model with different mesh densities is captured to benchmark the HPC performance in solving high density mesh models. Boundary conditions, solution algorithm, solver setup and convergence criteria were identical for all models.

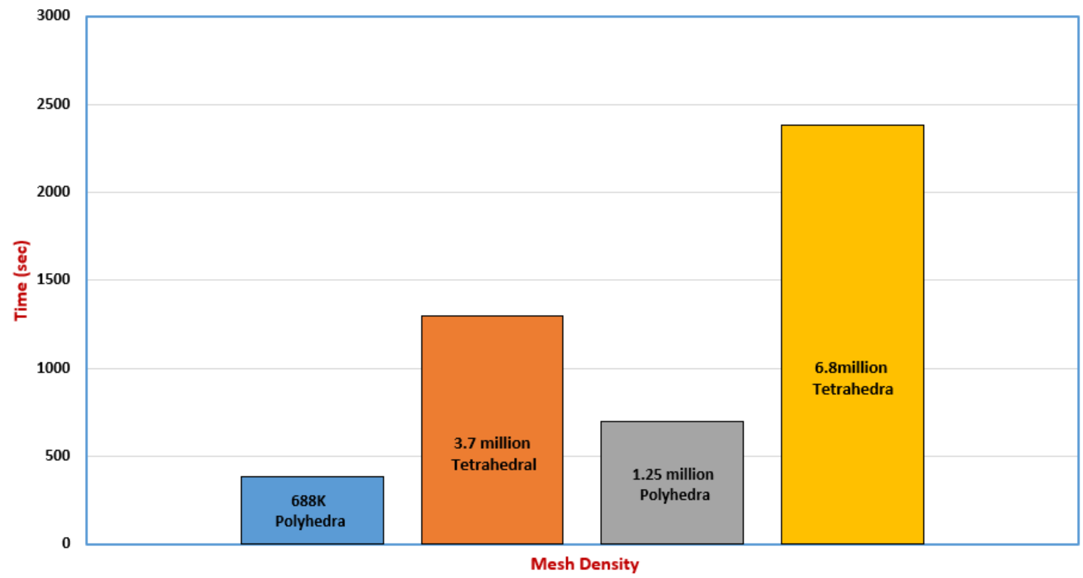


Figure 4: Run time comparison for different mesh densities using 24 CPU cores.

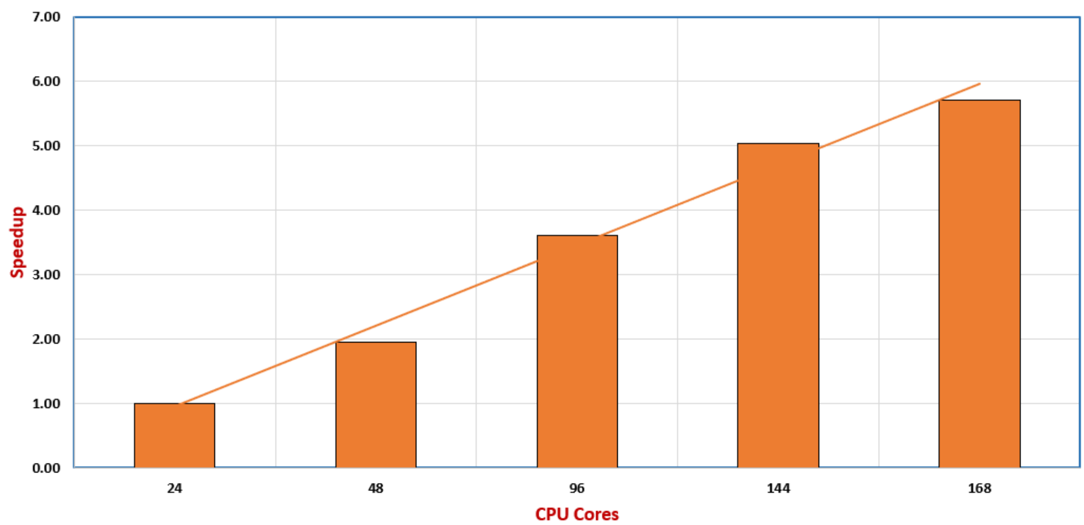


Figure 5: Speedup of 688K polyhedral mesh at different CPU cores.

Figures 4 compares the time required to run 200 iterations for different mesh densities with 24 CPU cores. The comparison of the solution time shows a significant reduction in the solution time when converting the meshes from tetrahedral to polyhedral. This is primarily due to the lower number of mesh elements with minimal impact on solution accuracy. Figures 5 summarizes the scalability study, which was based on the 688K polyhedral mesh. As can be seen from the figure, the solution speed scales close to linear up to 168 CPU cores. Figure 6 shows the decrease of simulation run time as the number of cores is increased. When using 168 cores, each simulation takes less than an hour, making it possible to run the entire design space of the bioreactor in less than 24 hours.

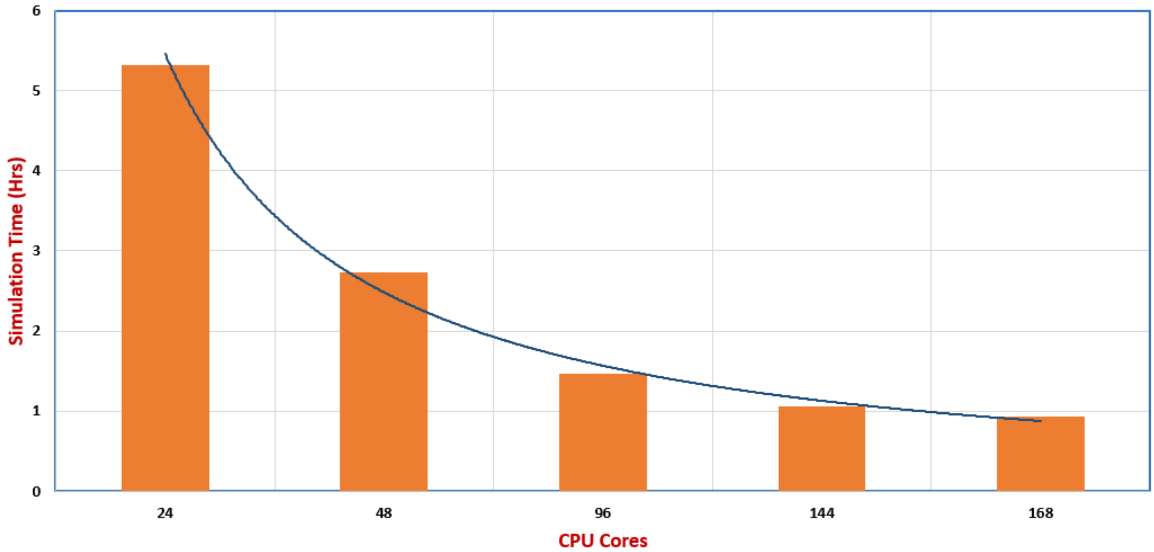


Figure 6: Simulation Run time comparison for 688K polyhedral mesh on different number of CPU cores.

A similar speedup study has been performed for the different types of meshes generated for this study. The solution speed scale-up results are plotted and compared with linear scale-up speed to compare the scale-up at different mesh densities. As shown in Figure 7, the solution speed scale-up is observed to move closer to the linear increase in solution speed as the mesh density increases.

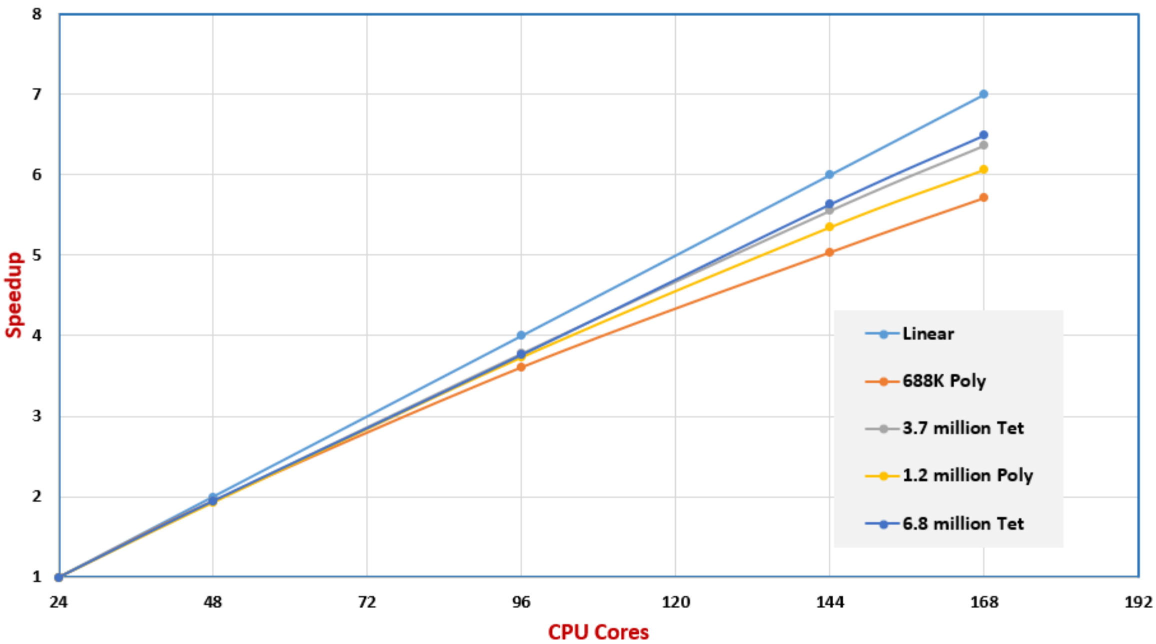


Figure 7: Comparison of solution speed scale-up with different mesh densities.

BENEFITS

1. The HPC cloud computing environment with ANSYS Workbench with FLUENT and DesignXplorer streamlined the process of running a DOE with drastically reduced process time.
2. Running the 10 design point simulations and generating the response surface took only 24 hours of run time with 144 CPU cores. This means design engineers can quickly execute DOE analyses to study the scale-up behavior of their bioreactors.
3. With the use of VNC Controls in the web browser, HPC Cloud access was very easy with minimal installation of any pre-requisite software. The entire user experience was similar to accessing a website through the browser.
4. The UberCloud containers helped smooth execution and provide easy access to the server resources. The UberCloud environment integrated with the Microsoft Azure platform proved to be powerful as it facilitates running parallel UberCloud containers, with a dashboard in the Azure environment which helped in viewing the system performance and usage.

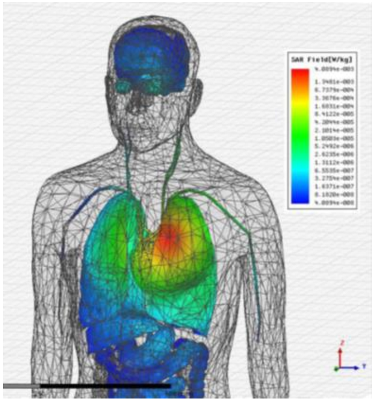
CONCLUSION & RECOMMENDATIONS

1. Microsoft Azure with UberCloud HPC resources provided a very good fit for performing advanced computational experiments that involve high technical challenges with complex geometries and multi-phase fluid flow interactions that would not typically be solved on a normal workstation, reducing the time required to establish a two-parameter design space for a bioreactor to a single day.
2. The combination of Microsoft Azure, HPC Cloud resources, UberCloud Containers, and ANSYS Workbench with FLUENT helped to accelerate the simulation trials and also completed the project within the stipulated time frame.

Case Study Author – Sravan Kumar and Marc Horner, ANSYS Inc.

ANSYS HFSS

Implantable Planar Antenna Simulation



"ANSYS HFSS in UberCloud's application software container provided an extremely user-friendly on-demand computing environment very similar to my own desktop workstation."

MEET THE TEAM

End user – Mehrnoosh Khabiri, Ozen Engineering, Inc. Sunnyvale, California
Team Expert – Metin Ozen, Ozen Engineering, Inc. and Burak Yenier, UberCloud, Inc.
Software Provider – Ozen Engineering, Inc. and UberCloud, Inc.
Resource Provider – Nephoscale Cloud, California.

Use Case

In recent years, with rapid development of wireless communication technology, Wireless Body Area Networks (WBANs) have drawn a great attention. WBAN technology links electronic devices on and in the human body with exterior monitoring or controlling equipment. The common applications for WBAN technology are biomedical devices, sport and fitness monitoring, body sensors, mobile devices, and so on. All of these applications have been categorized in two main areas, namely medical and non-medical, by IEEE 802.15.6 standard. For medical applications, the wireless telemetric links are needed to transmit the diagnostic, therapy, and vital information to the outside of human body. The wide and fast growing application of wireless devices yields to a lot of concerns about their safety standards related to electromagnetic radiation effects on human body. Interaction between human body tissues and Radio Frequency (RF) fields are important. Much research has been done to investigate the effects of electromagnetic radiation on the human body. The Specific Absorption Rate (SAR), which measures the electromagnetic power density absorbed by the human body tissue, is considered as an index by standards to regulate the amount of exposure of the human body to electromagnetic radiation.

In this case study implantable antennas are used for communication purposes in medical devices. Designing antennas for implanted devices is an extremely challenging task. The antennas require to be small, low profile, and multiband. Additionally, antennas need to operate in complex environments. Factors such as small size, low power requirement, and impedance matching play significant role in the design procedure. Although several antennas have been proposed for implantable medical devices, the accurate full human body model has been rarely included in the simulations. An implantable Planar Inverted F Antenna (PIFA) is proposed for communication between implanted medical devices in human body and outside medical equipment. The main aim of this work is to optimize the proposed implanted antenna inside the skin tissue of human body model and characterize the electromagnetic radiation effects on human body tissues as well as the SAR distribution. Simulations have been performed using ANSYS HFSS (High-Frequency Structural Simulator) which is based on the Finite Element Method (FEM), along with ANSYS Optimetrics and High-Performance Computing (HPC) features.

ANSYS HUMAN BODY MODEL AND ANTENNA DESIGN

ANSYS offers the adult-male and adult-female body models in several geometrical accuracy in millimeter scale [17]. Fig. 1 shows a general view of the models. ANSYS human body model contains over 300 muscles, organs, tissues, and bones. The objects of the model have geometrical accuracy of 1-2 mm. The model can be modified by users for the specific applications and parts, and model objects can simply be removed if not needed. For high frequencies, the body model can be electrically large, resulting in huge number of meshes which makes the simulation very time-consuming and computationally complex. The ANSYS HPC technology enables parallel processing, such that one has the ability to model and simulate very large size and detailed geometries with complex physics.

The implantable antenna is placed inside the skin tissue of the left upper chest where most pacemakers and implanted cardiac defibrillators are located, see Figure 1. Incorporating ANSYS Optimetrics and HPC features, optimization iterations can be performed in an efficient manner to simulate the implantable antenna inside the human body model.

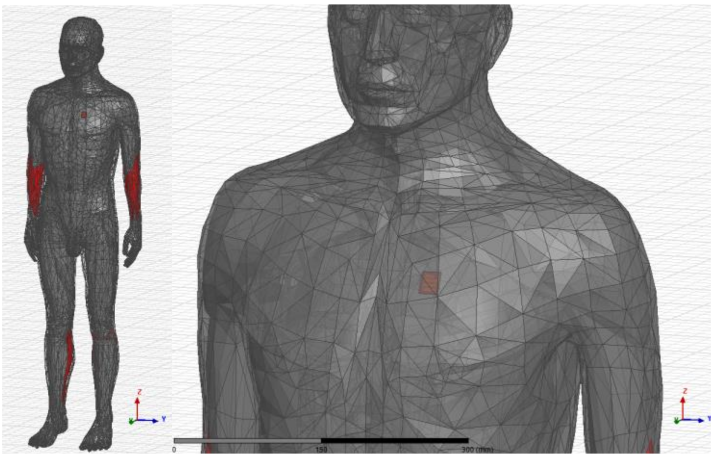


Figure 1: Implanted antenna in ANSYS male human body model.

The antenna is simulated in ANSYS HFSS which is a FEM electromagnetic solver. Top and side view of proposed PIFA is illustrated in Figure 2 (left), the 3D view of the implantable PIFA is demonstrated in Figure 2 (right). The thickness of dielectric layer of both substrate and superstrate is 1.28 mm. The length and width of the substrate and superstrate are L_{sub}=20mm and W_{sub}=24mm, respectively. The width of each radiating strip is W_{strip}=3.8mm. The other parameters of antenna are considered to be changed within the solution space in order to improve the PIFA performance. HFSS Optimetrics, an integrated tool in HFSS for parametric sweeps and optimizations, is used for tuning and improving the antenna characteristics inside the ANSYS human body model.

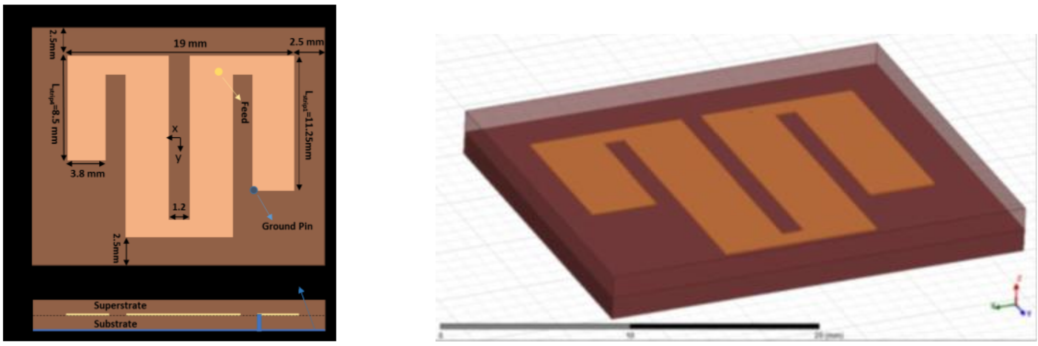


Figure 2: Top and side view of PIFA (left) and 3D view of PIFA geometry in HFSS (right).

RESULTS AND ANALYSIS

Figure 3 illustrates the far-field radiation pattern of the proposed PIFA at 402 MHz. Since the antenna is electrically small and the human body provides a lossy environment, the antenna gain is very small (~-44 dBi) and the EM fields are reactively stored in the human body parts in vicinity.

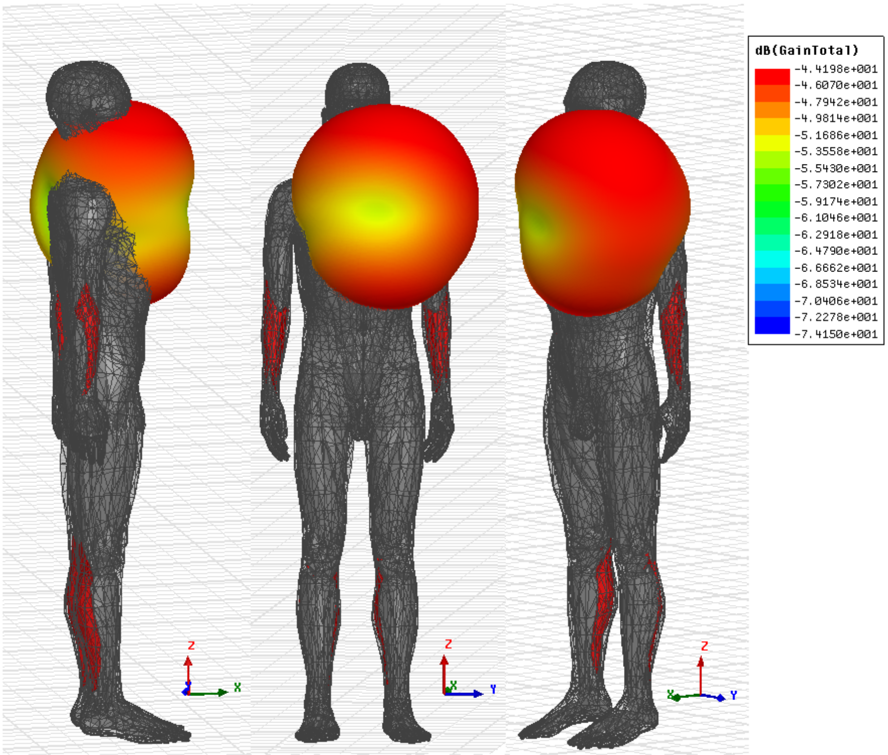


Figure 3: 3D Radiation pattern of implanted PIFA inside the human body model.

Figure 4 shows the simulated electric field distributions around the male human body model at 402 MHz center frequency. The electric field magnitude is large at upper side of the body, and it becomes weaker as going far away from the male body chest.

The electromagnetic power absorbed by tissues surrounding the antenna inside the human body model is a critical parameter. Hence, SAR analysis is required to evaluate the antenna performance. SAR measures the electromagnetic power density absorbed by the human body tissue. SAR measurement makes it possible to evaluate if a wireless medical device satisfies the safety limits.

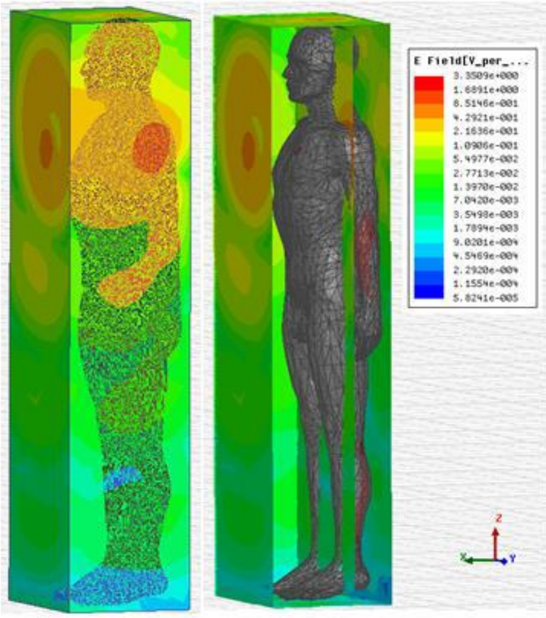
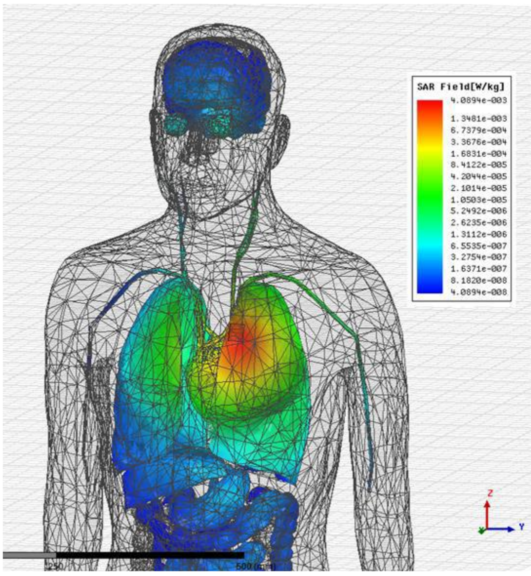


Fig. 4 Electric Field distribution around male body model at 402 MHz.

SAR is averaged either over the whole body or a small volume (typically 1 g or 10 g of tissue). ANSYS HFSS offers SAR calculations according to standards. The 3D plots of the local SAR distribution are shown in Figure 5 and Figure 6. In Figure 5, the detailed male body model with heart, lungs, liver, stomach, intestines, and brain are included. It can be observed that the left upper chest region where SAR is significant is relatively small. The peak SAR of the PIFA is smaller than the regulated SAR limitation. Figure 5 shows the SAR distribution on the skin tissue of the full human body model.



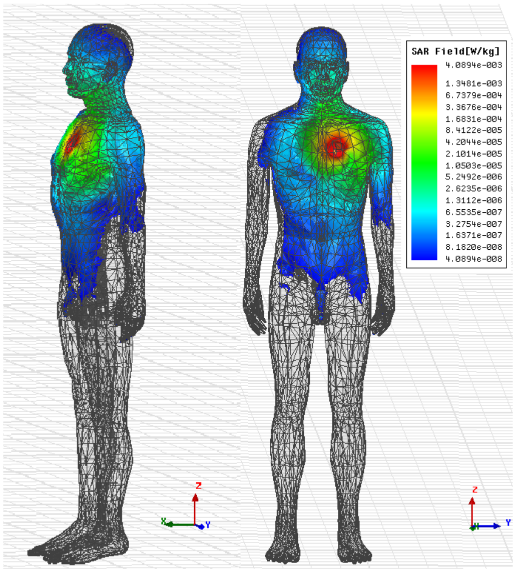


Figure 5 & 6: Local SAR distribution on upper side of male body model at 402 MHz. & Local SAR distribution on the skin tissue of male body model at 402 MHz.

A more detailed discussion of this use case by Mehrnoosh Khabiri can be found in the Ozen Engineering white paper "Design and Simulation of Implantable PIFA in Presence of ANSYS Human Body Model for Biomedical Telemetry Using ANSYS HFSS".

CONCLUSIONS

Design modification and tuning of antenna performance were studied with the implantable antenna placed inside the skin tissue of ANSYS human body model. The resonance, radiation, and Specific Absorption Rate (SAR) of implantable PIFA were evaluated. Simulations were performed with ANSYS HFSS (High-Frequency Structural Simulator) which is based on Finite Element Method (FEM). All simulations have been performed on a 40-core Nephoscale cloud server with 256 GB RAM. These simulations were about 4 times faster than on the local 16-core desktop workstation.

ANSYS HFSS has been packaged in an UberCloud HPC software container which is a ready-to-execute package of software designed to deliver the tools that an engineer needs to complete his task in hand. In this experiment, ANSYS HFSS has been pre-installed, configured, and tested, and running on bare metal, without loss of performance. The software was ready to execute literally in an instant with no need to install software, deal with complex OS commands, or configure.

This technology also provides hardware abstraction, where the container is not tightly coupled with the server (the container and the software inside isn't installed on the server in the traditional sense). Abstraction between the hardware and software stacks provides the ease of use and agility that bare metal environments lack.

Case Study Author: Mehrnoosh Khabiri, Ozen Engineering, and Wolfgang Gentzsch, The UberCloud

ANSYS LS-DYNA

Airbag simulation in the Microsoft Azure Cloud

"Microsoft Azure resources with UberCloud Containers and ANSYS LS-DYNA provide an excellent platform to develop and run accurate simulation models that involve complex impact physics."

MEET THE TEAM

End-User/FEA Expert: Praveen Bhat, Technology Consultant, INDIA
Software Provider: ANSYS INC. and UberCloud LS-DYNA Container
Resource Provider: Microsoft Azure with UberCloud Containers
HPC Expert: Burak Yenier, Co-Founder, CEO, UberCloud

USE CASE

Automobile airbags are the results of some incredible engineering. In a high speed crash, the driver of the car can be hurled into the steering wheel, but in an airbag equipped car small electronics will inflate the airbag providing enough cushion to protect the driver from impact. Fatality and serious injury rates have been reduced since the widespread installation of airbags.

The main objective of this project is to understand the air bag inflation behaviour under dynamic conditions. The simulation framework is developed and executed with ANSYS LS-DYNA in an UberCloud container on Microsoft Azure computing resources to achieve good accuracy in result prediction and also with respect to the solution time and resource utilization.

PROCESS OVERVIEW

Figure 1: Geometry & Mesh model for a steering with closed airbag model.

1. The steering wheel with folded air bag is meshed using the 2D quad mesh elements. The contacts and interactions between different components in the steering wheel assembly and air bag is defined.
2. The material properties for the steering wheel assembly with air bag are defined. The section properties are defined which involved thickness definition for different components in the assembly.
3. The next step of the model setup is defining the model boundary conditions and assigning load curves. The steering wheel geometry is fixed, and the load curve provides the air bag opening forces which are defined on the air bag component.
4. Solution algorithm and convergence criteria are defined along with output parameters and results to be used for post processing.
5. The Model is solved in ANSYS LS-DYNA with parallel computing on 1 to 16 cores. The final result is used to view the output of the simulation result, and the respective result components are captured using the post-processing software tool in ANSYS.

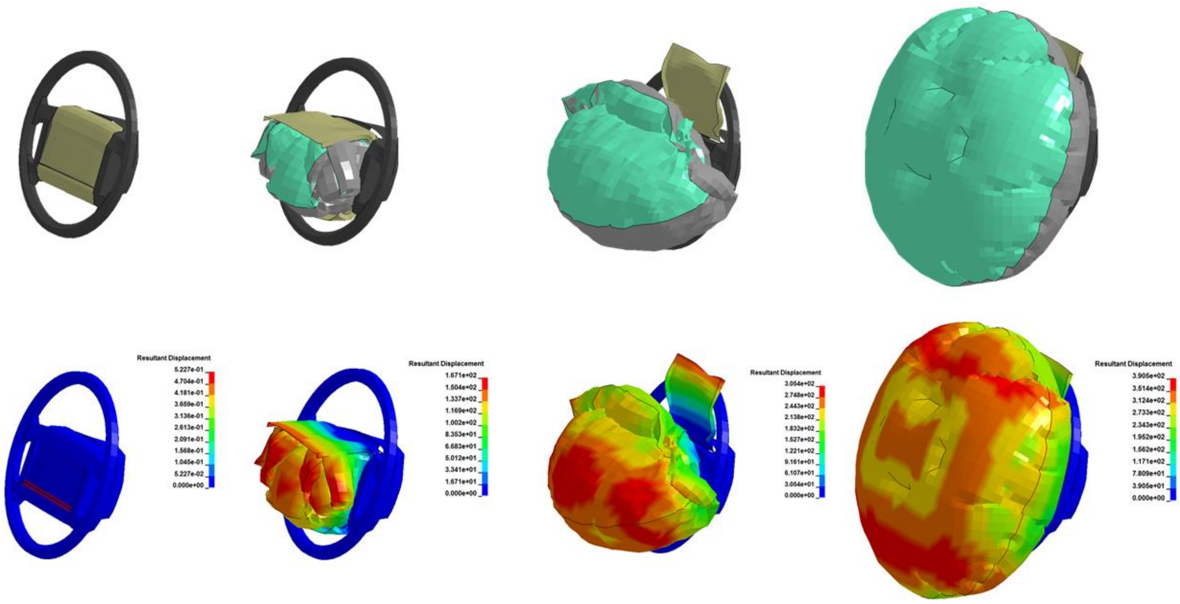


Figure 1: Deformation plot of air bag (a) Opening sequence of air bag (b) Contour plot on the steering and air bag assembly.

HPC PERFORMANCE BENCHMARKING

The HPC system is a Microsoft Azure GS5 Instance: 32 cores, 448 GB RAM, Max Disk size OS = 1023 GB and local SSD = 896 GB, Cache size 4224, and Linux operating system. The air bag model is simulated using ANSYS LS-DYNA in an UberCloud Container on the Microsoft Azure cloud platform. The model is evaluated for the air bag behaviour, and it also determines the rate of air bag opening and the stresses developed on the air bag material.

Different finite element models are developed for fine and coarse mesh. The model data are submitted to the ANSYS LS-DYNA container. The time for solving the model with different mesh intensity is captured to benchmark the performance in solving high density mesh models. Boundary conditions, solution algorithm, solver setup and convergence criteria remain same for all models.

Figures 3 & 4 provide a comparison of solution times required for different mesh density model with and without parallel processing. The comparison of the solution time with single core processor and 32 core processors shows that the time required to solve using parallel computing is significantly less when compared with running the same simulations with single core.

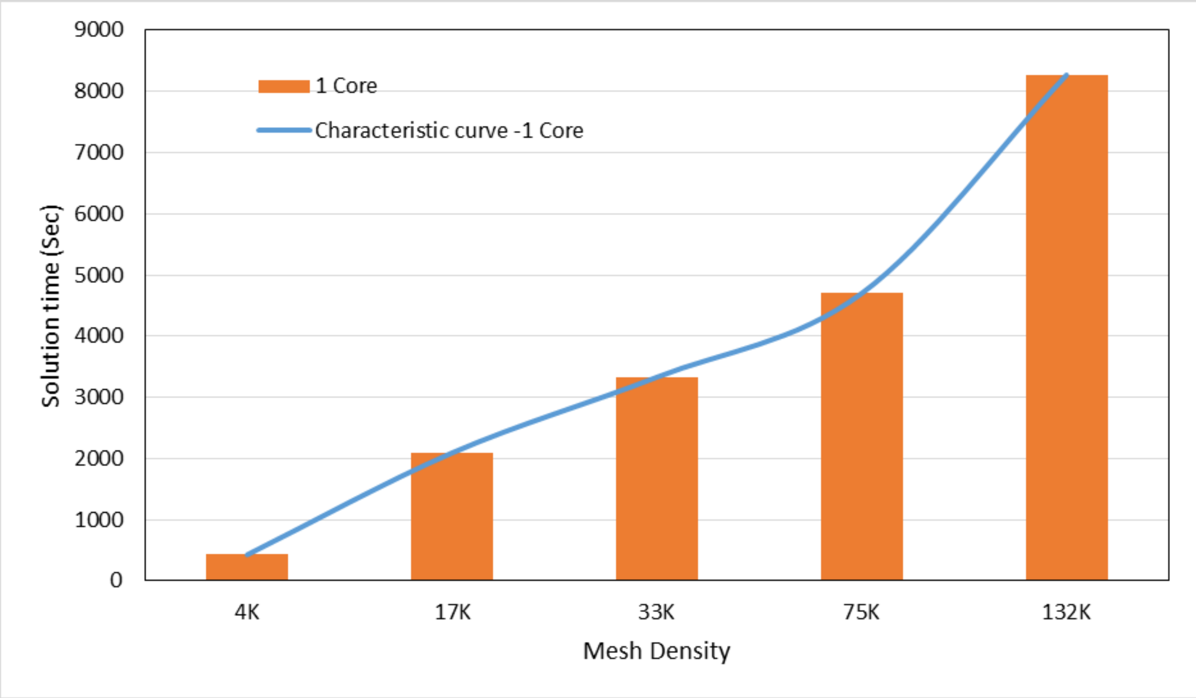


Figure 3: Solution time required for different mesh density with single CPU Core.

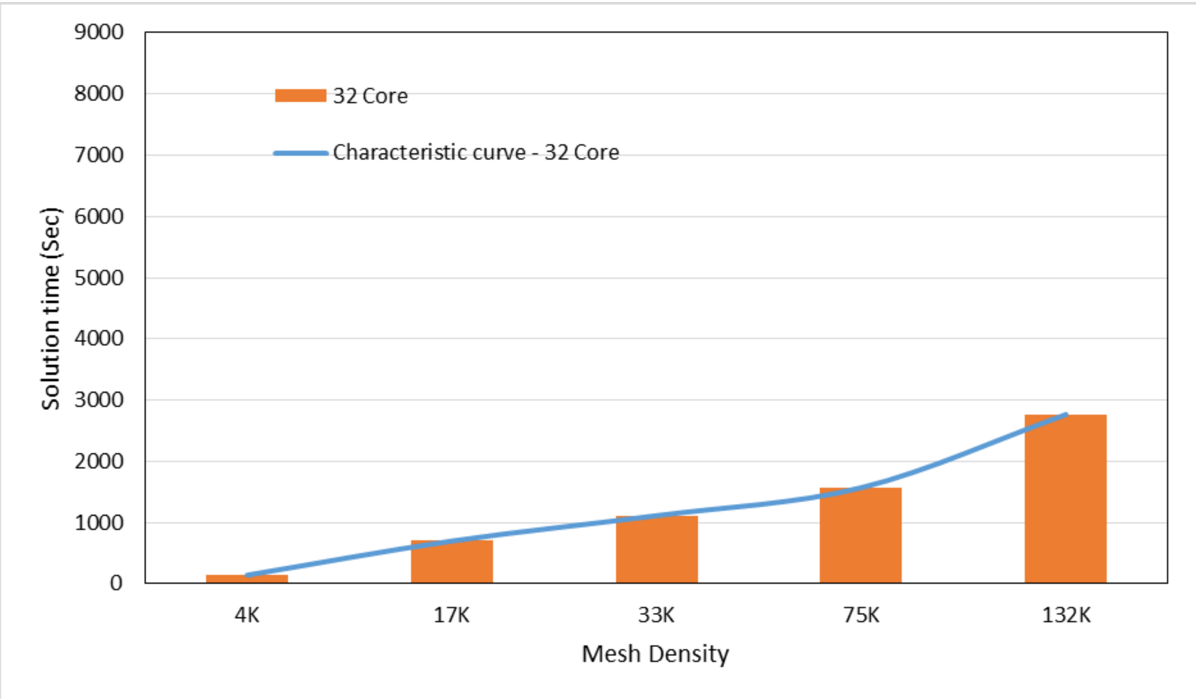


Figure 2: Solution time required for different mesh density using 8 CPU Core.

Figure 5 shows the comparison of the solution time required for a model with 132K elements which is submitted with different CPU cores. Figure 6 provides a comparison of the solution for different mesh models using different CPU cores. The comparison of the solution time with single core processor and 32 core processor again demonstrates that the time with parallel computing is significantly less when compared with running the same simulations with single core.

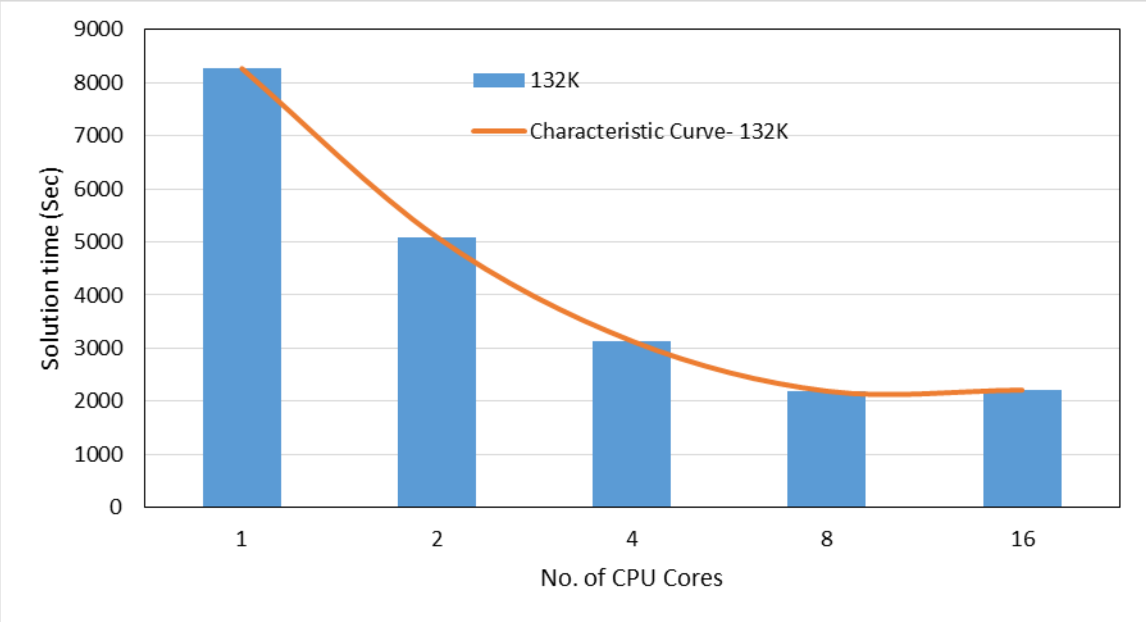


Figure 3: Solution time for a model with 132K elements solved using different HPC Core configuration.

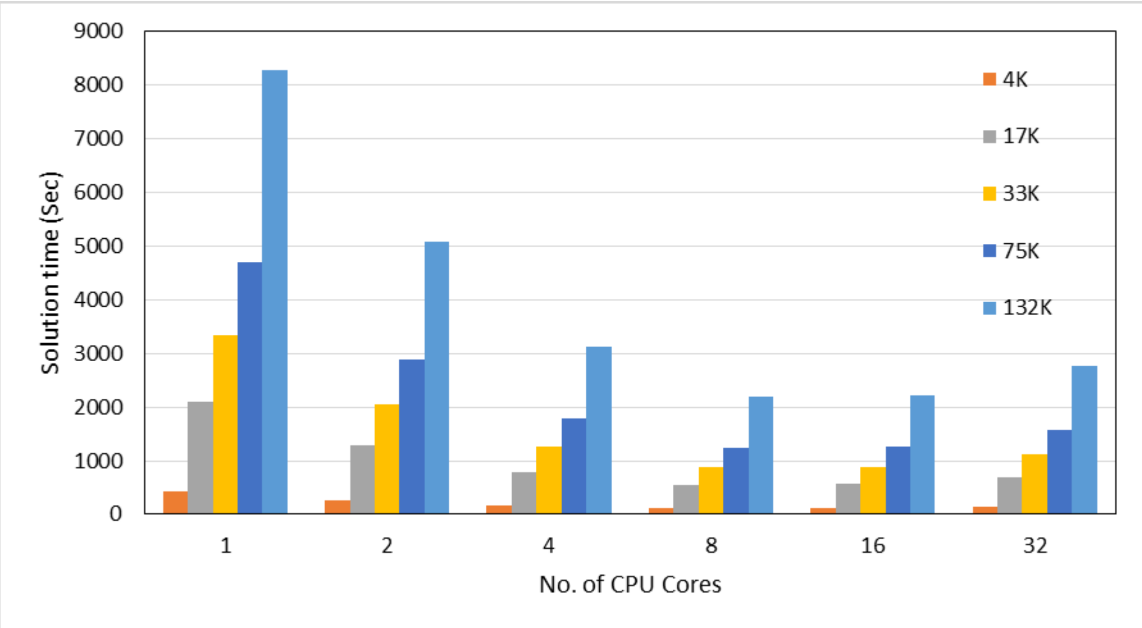


Figure 4: Solution time for models with different mesh densities using different HPC core configurations.

EFFORT INVESTED

End User/Team Expert: 10 hours for the simulation setup, technical support, reporting and overall management of the project.

UberCloud support: 1 hour for monitoring & administration of the performance in the host server.

Resources: ~1000 core hours used for performing various iterations in the simulation experiments**.**

CHALLENGES

The project challenges faced were related to technical complexity of the application and the ability to run the dynamic simulation within a very short period of execution time. Hence it was necessary to perform trials with different mesh density models to accurately capture the air bag behaviour. The finer the mesh the better is the simulation result accuracy, but the higher is the simulation runtime, and hence the challenge was to perform the simulation within the stipulated timeline. Getting exposure to the Azure Cloud platform and using its features consumed some time at first as this required going through learning and following written instructions provided by Azure.

BENEFITS

5. The HPC cloud computing environment with ANSYS workbench & LS-DYNA made the process of model generation easier with process time reduced drastically along with result viewing & post-processing because of the ANSYS / Azure / UberCloud HPC set-up.
6. The mesh models were generated for different cell numbers where the experiments were performed using coarse - to - fine to very fine mesh models. The HPC computing resource helped in achieving smoother completion of the simulation runs without re-trails or resubmission of the same simulation runs thereby helping the user to achieve highly accurate simulation results.
7. The computation time requirement for a fine mesh (~132K cells) is quite high, which is nearly impossible to achieve on a normal workstation. The HPC Cloud provided this feasibility to solve highly fine mesh models and the simulation time drastically reduced providing an advantage of getting the simulation results within acceptable time (~30 min).
8. The experiments in the HPC Cloud showed the possibility and gave extra confidence to setup and run the simulations remotely in the cloud. The different simulation setup tools were pre-installed in the HPC container and this enabled the user to access the tools without any prior installations.
9. With the use of VNC Controls in the Web browser, the HPC Cloud access was very easy with no installation of any pre-requisite software. The whole user experience was like accessing a website through the browser.
10. The UberCloud containers helped with smooth execution of the project with easy access to the server resources. The UberCloud ANSYS container integrated with the Microsoft Azure platform proved to be powerful as it facilitates running parallel UberCloud containers. A dashboard in the Azure helped in viewing the system performance and usage.

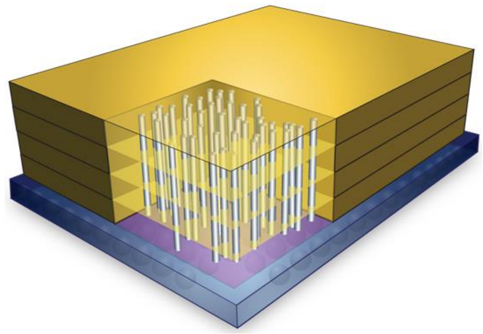
CONCLUSION & RECOMMENDATIONS

3. The selected HPC Cloud environment with UberCloud containerized ANSYS Workbench with LS-DYNA on Microsoft Azure was an excellent fit for performing complex simulation that involved huge hardware resource utilization with a high number of simulation experiments.
4. Microsoft Azure with UberCloud Containers enabled performing advanced computational experiments that involve high technical challenges with complex geometries and which cannot be solved on a normal workstation.

Case Study Author – Praveen Bhat, Technology Consultant, INDIA

ANSYS Mechanical

Finite Element Analysis for 3D Microelectronic Packaging in the Cloud



"Our experiment showed that the performance of the HPC cloud is sufficient for solving large FE analysis problems."

MEET THE TEAM

End User – Dazhong Wu, Iowa State University, Xi Liu, Georgia Tech
Resource Provider – Steve Hebert, Nimbix
Other Cloud Resources – NephoScale and Microsoft Azure
Software Provider – Wim Slagter, ANSYS Inc.

USE CASE

Although both academia and industry have shown increasing interest in exploring CAE in the Cloud, little work has been reported on systematically evaluating the performance of running CAE applications in public clouds against that of workstations and traditional in-house supercomputer. In particular, an important question to answer is: "Is the performance of cloud computing services sufficient for large-scale and complex engineering applications?" As an initial step towards finding an answer, the experiment evaluated the performance of HPC clouds via quantitative and comparative case studies for running an FE analysis simulation model using HPC in three public clouds. Specifically, the primary aspect of the experiment was evaluating the performance of the Nimbix Cloud using multiple nodes, NephoScale Cloud using a single node, and Azure Cloud using a single node.

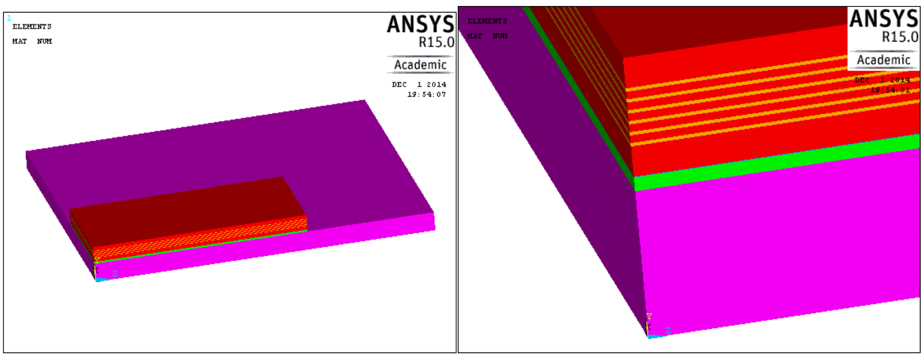


Figure 1: Schematic view of 3D microelectronic package

The application used in the experiment was the thermo-mechanical warpage analysis of a 3D stacked die microelectronic package integrated with through silicon vias (TSVs), as shown in Figure 1. Over the last decade, digital information processing devices for HPC systems require an increasing level of computing power while using less power and space. 3D integrated logic devices with stacked memory using through-silicon vias have the potential to meet this demand. The shorter and highly parallel connection between logic and high-capacity memory can avoid the von Neumann bottleneck, reduce power consumption, and realize the highest device density. However, the challenges pertaining to 3D packaging with TSVs include yield, assembly, test, and reliability issues. In particular, the 3D stacked die package warpage problem is one of the key challenges for 3D package assembly. Understanding the package warpage behavior is crucial to achieve high package stack yield because the different warpage directions of top and bottom package will impact the yield of package stacking.

To address these issues, we created a FE model with detailed package features, shown in Figure 1, to investigate the warpage behaviors of stacked die 3D packages, not well understood for various reasons. One is that 3D stacked dies interconnected with TSVs are still in development stage. Thus, a very limited amount of prototype samples are available for investigating the warpage problem. The other reason is numerical simulation of 3D packages is computationally intensive. For example, in 3D packages, the in-plane dimensions are at millimeter scale. However, the out-of-plane dimensions, TSVs, and microbumps are at the micrometer scale, which results in a significantly increased finite-element mesh density to meet the element aspect ratio requirements. In addition, there are generally hundreds or thousands of TSVs/microbumps between each stacked die.

PROCESS OVERVIEW

1. Define the end-user project
2. Contact the assigned resources and set up the project environment
3. Initiate the end-user project execution
4. Monitor the project
5. Review results
6. Document findings

RESULT OF THE ANALYSIS

Table 1: Hardware specifications for the workstation

Processor Model	Two Intel® Xeon® CPU E5530@ 2.40GHz
Number of Cores	8
Memory	24 GB
Parallel Processing Method	Shared Memory Parallel (SMP)

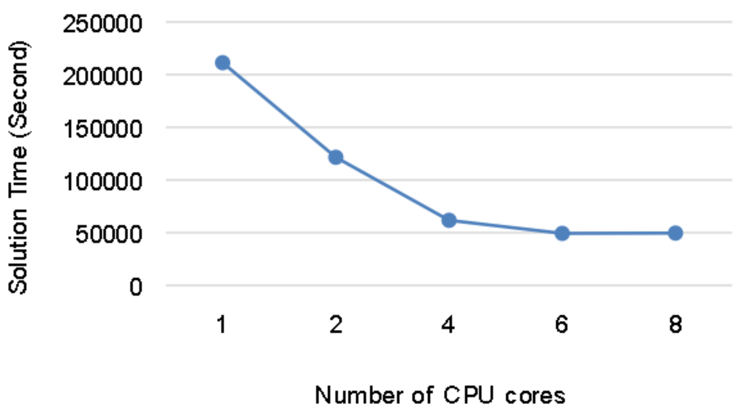


Figure 2: Solution scalability on a workstation

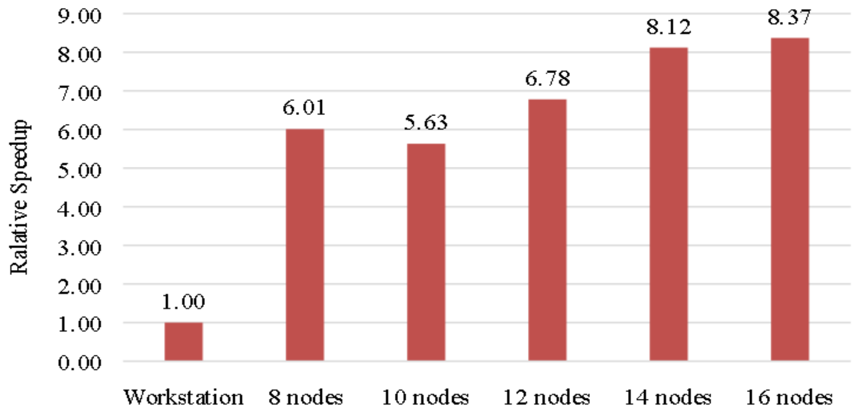


Figure 3: Relative speedup versus number of nodes

Table 1 lists the workstation hardware specifications. Figure 2 shows the solution scalability on the workstation. Figure 3 shows the comparison of the run time using the 6-core configuration on a standard workstation against the mean run times using 8 to 16 nodes in the cloud. As expected, we found that distributed ANSYS performance significantly exceeds that of SMP ANSYS. Specifically, the maximum speed-up over the workstation is 8.37 times, with using 16 nodes in the cloud.

Table 2: Hardware specifications for the NephoScale Cloud

Processor Model	Intel Xeon CPU E5-2690 v2 @ 3.00 GHz
Number of Cores	20
Memory	256 GB
Parallel Processing Method	Shared Memory Parallel (SMP)

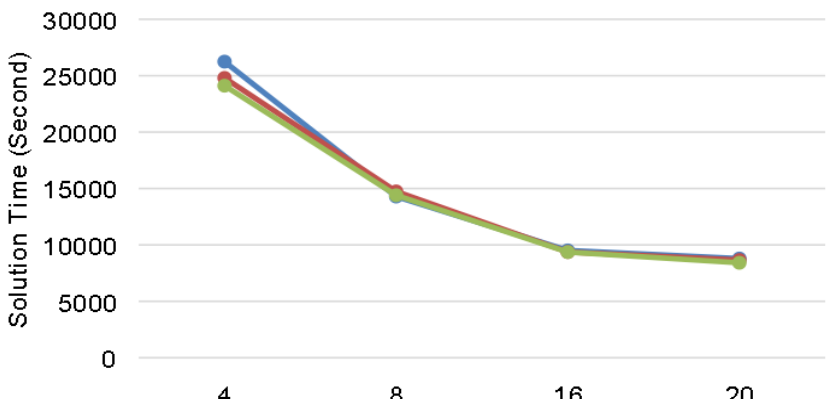


Figure 4: Solution scalability on the NephoScale Cloud

Table 2 lists the hardware specifications of the NephoScale Cloud. The single node on the NephoScale Cloud has 20 CPU cores and 256GB memory. Figure 4 shows the solution scalability on the NephoScale Cloud using 4, 8, 16, and 20 CPU cores on a single node.

Table 3: Hardware specifications for the Azure Cloud

Processor Model	Intel Xeon CPU E5-2698B v3@ 2.00GHz
Core	32
Memory	448 GB
Parallel Processing Method	Shared Memory Parallel (SMP)

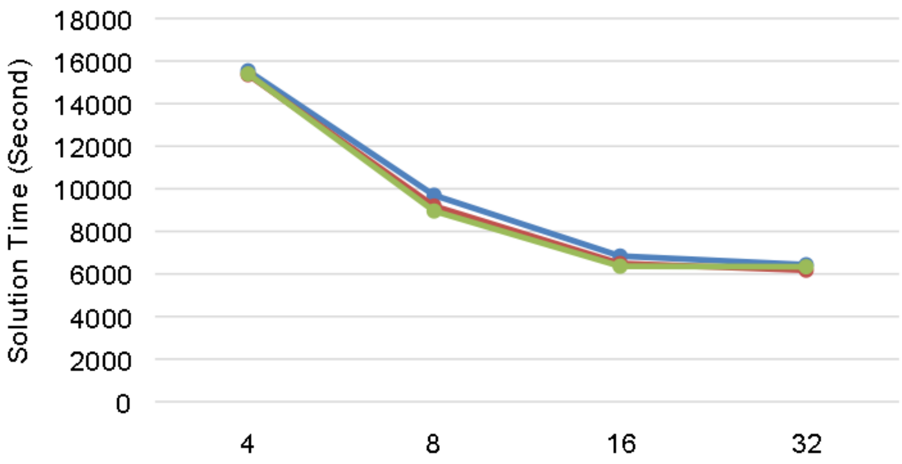


Figure 5: Solution scalability on the Azure Cloud

Table 3 lists the hardware specifications of the Azure Cloud. The single node on the Azure Cloud has 32 CPU cores and 448 GB memory. Figure 5 shows the solution scalability on the Azure Cloud using 4, 8, 16, and 32 CPU cores on a single node.

Table 4 Hardware specifications for the dedicated on premise supercomputer

Processor Model	Two 8-Core Intel Xeon CPU E5-2650 v2 @ 2.6 GHz
Cores per Node	16
Memory per Node	128 GB
Interconnect	40 Gbps QDR Infiniband
File system	Lustre parallel file system
Parallel Processing Method	Distributed Memory Parallel (DMP)

Table 4 lists the hardware specifications of the dedicated on-premise supercomputer. Figure 6 shows the solution scalability on this supercomputer using 8, 10, 12, 14, and 16 nodes.

This study has shown that performance bottlenecks may exist in HPC clouds using multiple nodes. The two single machines on the two public clouds significantly outperformed the workstation as shown in Figures 2, 4, and 5. In addition, the run times using SMP on a single node was very stable. Moreover, when using multiple nodes, low efficiency in cloud-based HPC is due to a severe system imbalance from factors such as CPU and memory size limitations, slow I/O rate, slow interconnects between processors on distributed memory systems, and slow solver computational rate.

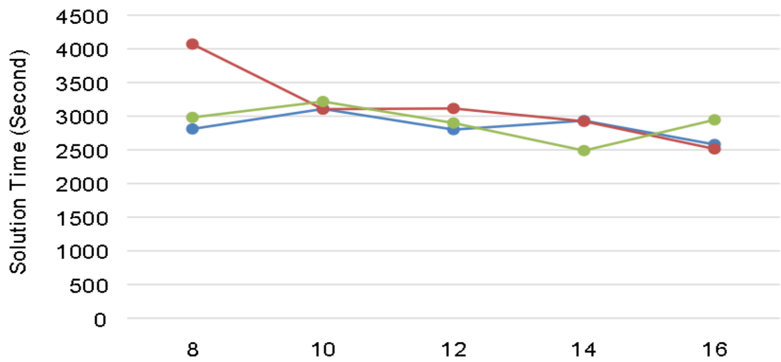


Figure 6: Solution scalability on the dedicated on premise supercomputer

Based on the experimental results, some of the general observations for cloud-based HPC simulations for finite element analysis are as follows:

- **Node count** – Adding more nodes does not always accelerate simulations.
- **Memory –** Having powerful nodes with sufficient memory is more desirable than having nodes with limited memory.
- **Interconnects** – Poor interconnects between CPU cores result in run times going up as CPU cores are added.
- **Scalability** – Cloud-based HPC typically cannot achieve linear scalability for FEA applications using the ANSYS Mechanical software package.

BENEFITS

The benefits of applying cloud-based HPC in FEA applications include:

- **Anytime, anywhere access –** Cloud-based HPC enables users to access state-of-the-art FEA software package from ANSYS and HPC computing hardware from Nimbix via a web portal and/or application program interfaces (APIs) anytime, anywhere.
- **Cost efficiency –** Cloud-based HPC allows users to solve complex problems using FEA simulations that typically require high bandwidth, low latency networking, many CPU cores, and large memory size. In particular, cloud-based HPC enables users to not only improve computing performance as dedicated on premise HPC clusters, but also reduce costs by using on-demand computing resources and the pay-per-use pricing model without large capital investments.
- **High flexibility –** Cloud-based HPC has the potential to transform dedicated HPC clusters into flexible HPC clouds that can be shared and adapted for rapidly changing customer requirements through private, hybrid, and public clouds.
- **High throughput –** Cloud-based HPC can significantly increase simulation throughput as opposed to standard workstations by allowing globally dispersed engineering teams to perform complex engineering analysis and simulations concurrently and collaboratively.

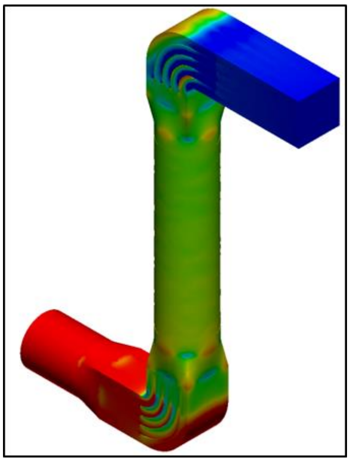
CONCLUSIONS AND RECOMMENDATIONS

In this experiment, we evaluated the performance of an HPC cloud for a FE analysis application. As opposed to traditional computing paradigms, such as standard workstations, HPC in the cloud enables scalable accelerated computationally expensive FE simulations. In response to the initial research question, our experimental results show that the performance of the HPC cloud is sufficient for solving the large example FE analysis problem. In the future, it will be worthwhile to compare the performance of an HPC cloud with that of a supercomputer. Such a comparison can help users make decisions when facing a tradeoff between performance and cost associated with performing a large and complex computer-aided engineering analysis. Finally, evaluating the performance of a HPC cloud for different problem sizes and solvers would also be an important direction for future research. By identifying the largest scale of an engineering problem that cloud-based HPC can address, we can identify the boundary between cloud-based HPC and supercomputers.

Case Study Author – Dazhong Wu

Autodesk CFD Flex

High Efficiency Duct Elbow Optimization in the Cloud



"Having access to cloud resources is an excellent option for running parallel CFD simulations! The software is streamlined and robust in terms of setting-up, accessing and processing simulation data."

MEET THE TEAM

End User – The end user (remaining anonymous) is a Canadian company specialized in designing, developing, and manufacturing equipment for plastic blown film production. The engineers use data from CFD simulations to design and optimize their cooling products.
Software and Resource Provider – Autodesk® supplied the end-user with the fluid-flow simulation software Autodesk® CFD Flex 2016 and the supporting cloud infrastructure.
HPC Expert – Jon den Hartog, P.E., Autodesk® Sr. Product Line Manager and Heath Houghton, Autodesk® Simulation CFD Product Manager.

USE CASE

The throughput of blown film is greatly influenced by the efficiency and stability of the central air-cooling system. As a result, it is desirable to have a cooling system which has a relatively low pressure drop, high flow rate, and uniform air distribution at the exit of the cooling device. It is also beneficial to have this system as compact as possible. The end user was seeking to develop and optimize a high efficiency duct elbow that will satisfy the performance metrics listed above.

The goal of this experiment was to explore cloud simulation using Autodesk® CFD Flex 2016. With the High Performance Computing (HPC) cloud, the end user was interested in reducing simulation time while also improving result accuracy through the ability to refine the mesh size within their model.

END-USER-PROJECT

The first step was generating multiple CAD models that represented the duct system (see geometry of the duct in Figure 1). The configurations ranged from smooth circular elbows to mitered elbows. To assist with uniform air distribution and reducing pressure drop, turning vanes were also added to these designs. Figure 2 displays an ideal velocity profile at the exit of the duct system. Sixteen design variations were proposed for this experiment, including a performance comparison study with and without turning vanes. Additionally, a mesh and configuration sensitivity analysis were conducted on the best performing duct elbow.

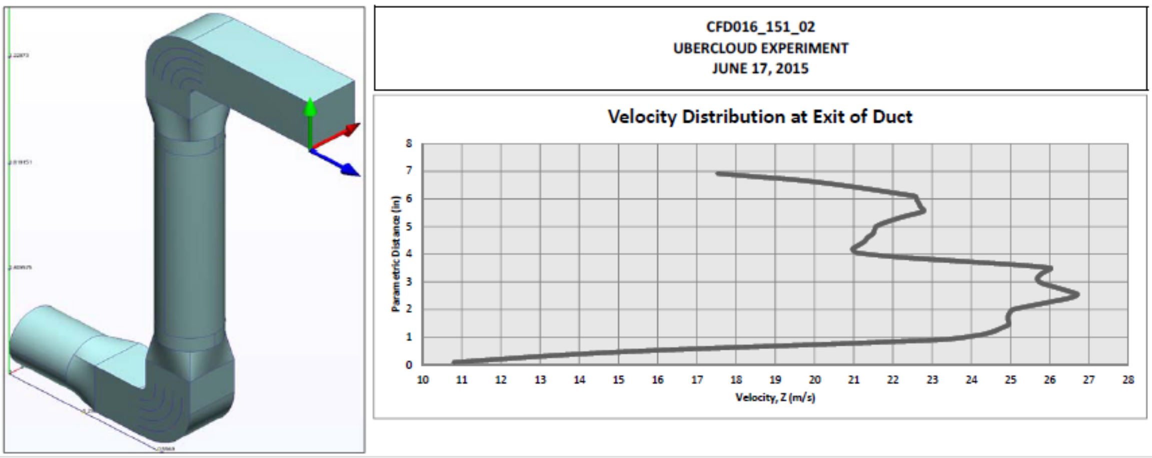


Figure 1: Geometry and the velocity distribution at the exit of the duct

In CFD analysis, resolution and accuracy of the model is dependent on the mesh refinement. The models were meshed with solid elements which generated an element count between 1,500,000 to 3,200,000 for each model. Individual components were meshed to ensure that an adequate mesh density was established. Boundary mesh enhancement techniques were also used to help the solver generate accurate results. To better understand the natural phenomenon of swirling and jet effects, the HPC experts recommended using the SST k-Omega solver.

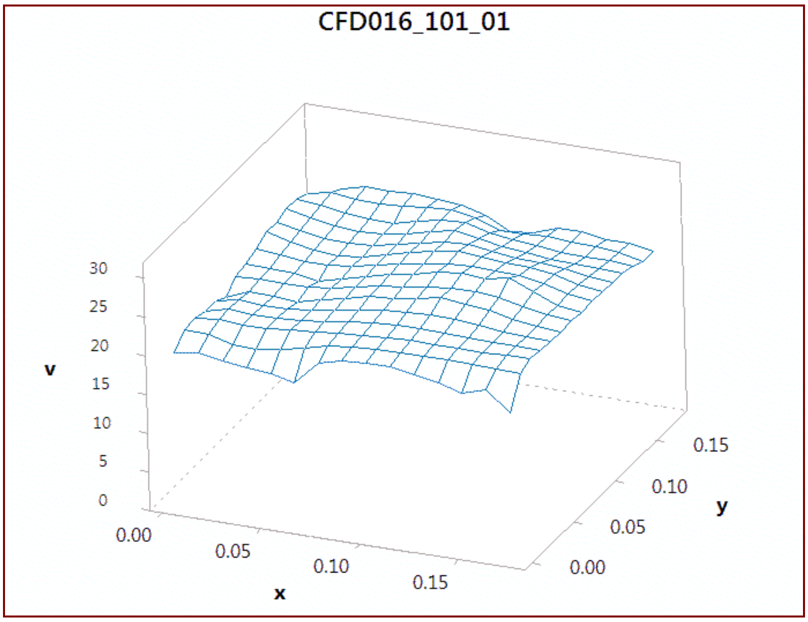


Figure 2: Velocity Profile at the Exit of the Duct System

CHALLENGES

The end user was very familiar with using the desktop version of Autodesk® Simulation CFD 2016 and experienced no significant obstacles while conducting this experiment. The primary benefit of Autodesk® CFD Flex 2016 cloud computing is the ability to run several simulations in parallel. However, for projects that require simulations to be solved sequentially, the benefits over solving the simulations locally on a relatively powerful workstation are limited. For example, in some cases the results from one simulation are necessary in order to generate the geometry for the next simulation.

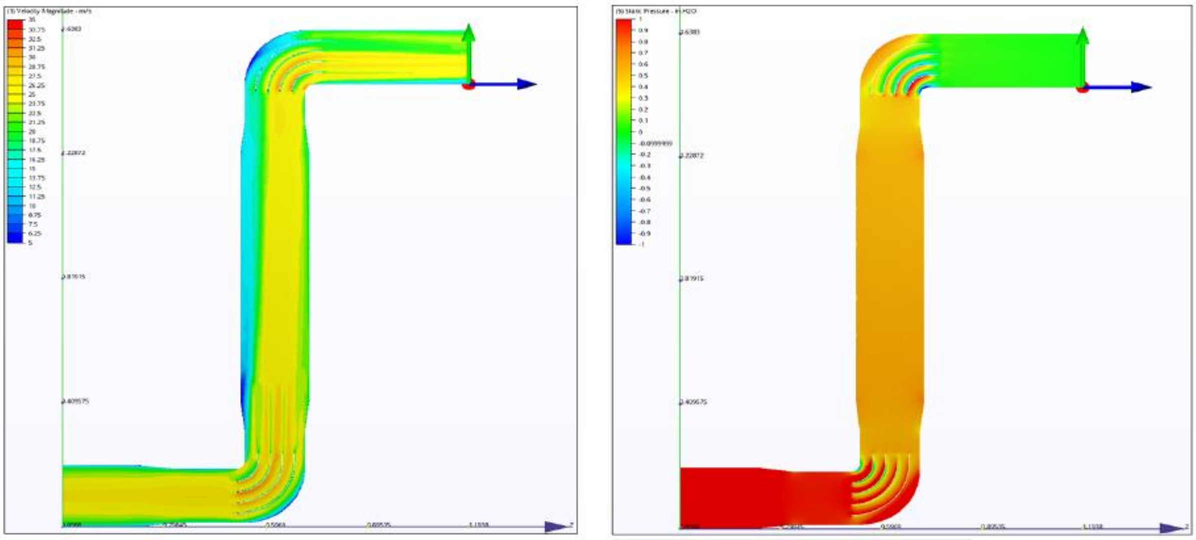


Figure 3: Velocity (Left) and Pressure (Right) results from scenario CFD015_151_01

BENEFITS

The end user experienced many benefits through use of Autodesk® CFD Flex 2016 cloud simulations as opposed to using a local desktop computer.

- The cloud enabled parallel simulations, which could not be done using a local desktop computer. The overall project simulation time was dramatically reduced by solving in parallel. The simulations for this project solved within the cloud for a total computing time of 85 hours. If each simulation were to be run in series, the simulation time would be approximately 290 hours.
- Additionally, a 30% reduction in simulation turnaround time was observed when solving in the cloud, as compared to solving on our local workstation.
- The option to solve locally or in the cloud provides true flexibility for the end user. Simulations can be solved locally to build confidence in the model and when the user is ready, they can send up a batch of scenarios to the cloud. Thus, users are no longer required to invest in high performance computers. A key advantage is that all local hardware resources are preserved, allowing the user to continue working on product development without experiencing a reduction in computer performance.

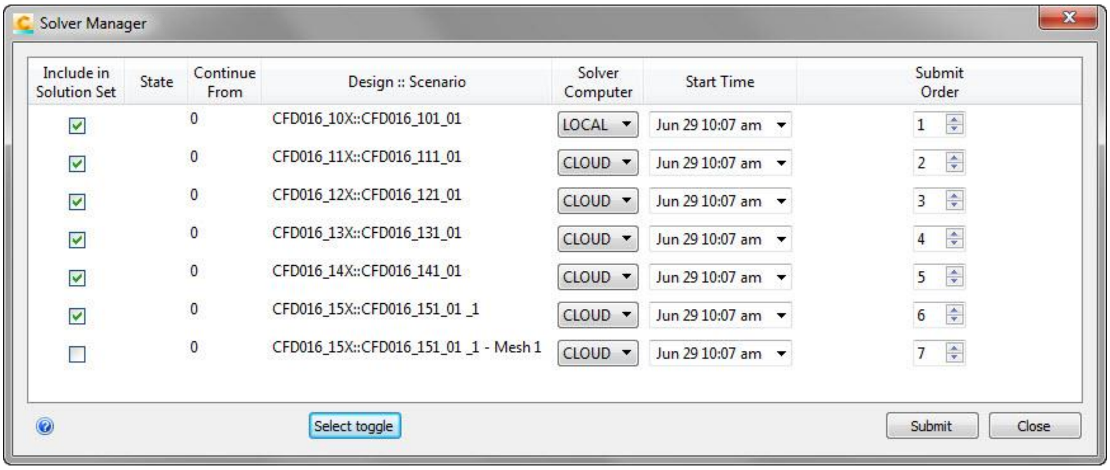


Figure 4: Autodesk CFD Flex Solver Manager

- It was relatively easy to monitor live simulations. Within the Simulation Job Manager interface, the user was able to track the progress of each simulation. Detailed information including the start time, finish time, status, job type, and priority level can be easily accessed within that application.

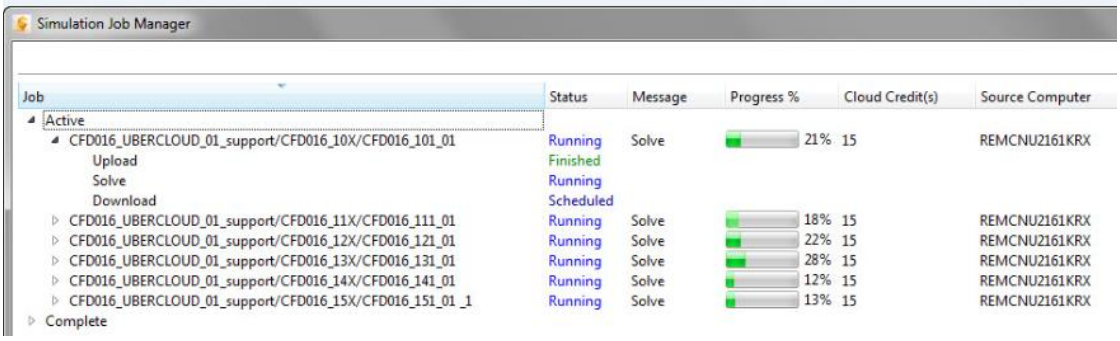


Figure 5: Screen shot of the Autodesk CFD Flex Simulation Job Manager showing the status of the six user jobs in list mode running in parallel in the Autodesk Cloud

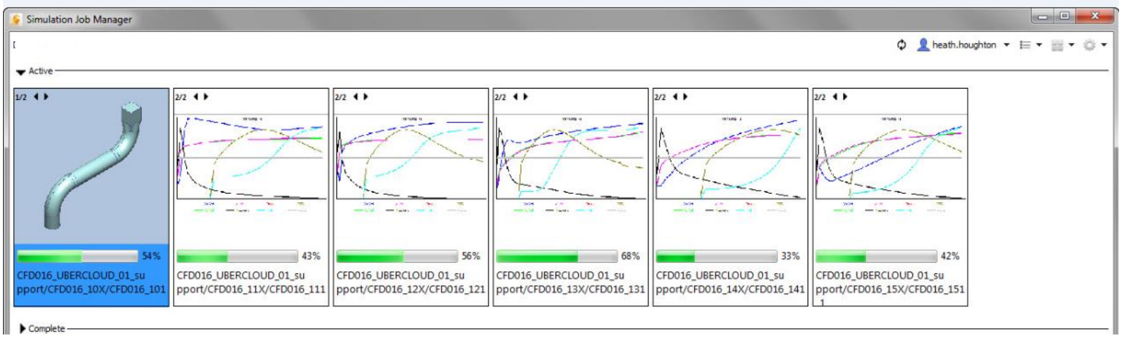


Figure 6: Screen shot of the Autodesk CFD Flex Simulation Job Manager showing the status of the six user jobs in icon mode running in parallel in the Autodesk Cloud

- Autodesk® CFD Flex 2016 also allowed the end user to setup various configurations while the program was simulating in the cloud. Once setup, the scenarios were released to the cloud. This process was completed without stopping the simulations. For instance, if the end user's local computer restarts or shuts downs after the simulations have been uploaded to the cloud, the simulations are unaffected and will continue to solve without any interruptions. Once the simulation is complete, the end user can download the results to their local desktop computer for further analysis.

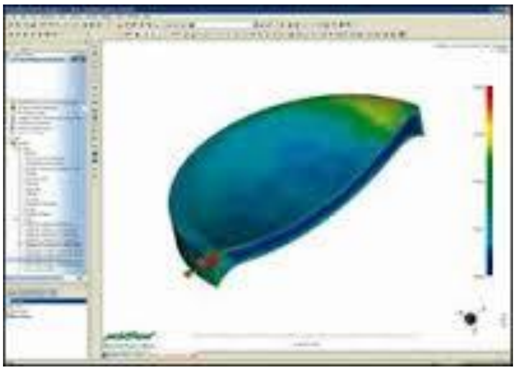
CONCLUSION AND RECOMMENDATIONS

This experiment offered the end user the opportunity to use and evaluate the cloud computing capacity available from Autodesk®. Multiple scenarios of simulations were generated in one batch to observe the true performance of parallel simulations in the cloud. Overall, having access to cloud resources is an excellent option for running parallel CFD simulations. The software is streamlined and robust in terms of setting-up, accessing and analyzing simulation data.

Case Study Author – The Team 174 End-User

Autodesk Moldflow

Examining Heat Transfer on Thermoplastic Olefins in Injection Molded Parts



"Without the cloud usage, the analysis for this project would probably still be running."

MEET THE TEAM
End User – David Okonski, Research Engineer and Project Lead, General Motors (GM)
Software and Resource Provider – Jay Shoemaker and Syed Rehmathullah, Autodesk
Team Experts – Jay Shoemaker and Syed Rehmathullah, Autodesk
Team Engineers – Adam Arb, Scott Smith, Tim Slaughter, Tyree Grasty, Western Michigan University

USE CASE
Plastics play a crucial role in manufacturing consumer products. After extensive research and analysis, studies on the Heat Transfer Coefficients (HTC) determined if the coefficients are sufficient for all plastic injection molded parts. These studies were created with Autodesk Moldflow. The research helps General Motors and other plastic manufacturers reduce the number of defective parts, costs, and energy associated with injection molded part.

GM and other manufacturers are always looking for ways to improve their profit margins. One way they can do this is by having a better understanding of the warpage and shrinkage within their simulation software. Understanding the influence of heat transfer on warpage and shrinkage predictions can help manufacturers cut their tools to account for these predictions. This means they do not have to send the tool back to be altered, a process that can cost thousands of dollars and weeks of time before they can begin to make parts.

PROJECT DESCRIPTION
General Motors is exploring the influence of heat transfer coefficients (HTC) on a TPO (Thermoplastic Olefin) during the simulation of the injection molding process for better prediction of warpage and shrinkage associated with injection molded parts. Current default values used by GM and other manufacturers may be causing inaccuracies in warpage and shrinkage predictions. These inaccuracies cause the manufacturers to repeatedly send their tools out to be altered until they create parts to correct specifications. The group analyzed HTC values within Autodesk's Moldflow Simulation Software to determine the influence of warpage and shrinkage.

BACKGROUND

Plastic injection molding is a common plastics process where plastic material is injected into a mold and then solidified to create a plastic part. A typical cycle time, approximately thirty seconds, is repeated resulting in a rapid rate of production.

Heat Transfer Coefficient (HTC): is a numerical characteristic that represents a resistance of heat transfer between two bodies. For the purpose of this research, it is the resistance of heat transfer between the mold wall and the polymer as it flows into the mold.

Thermoplastic Olefin (TPO): is a multiphase plastic composition containing a base resin of polypropylene, elastomer/rubber filled of polyethylene, and an EDPM impact medication giving it more impact resistance. STOp physical properties include low weight, easy moldability, and low temperature impact resistance.

Autodesk Moldflow: is a plastic injection molding simulation software that gives users the ability to mold parts virtually to collect result and analyze them. Benefits of this software is that it saves manufactures time and money creating parts – they are able to create efficient processes within this software and then transfer the process setting to the injection molding machine.

CREATING A DESIGN OF EXPERIMENTS (DOE)

For this project, there were lots of different studies, variables, and analyses that needed to be collected. A proper organizing tool needed to be created to help keep tabs on everything that is being adjusted and studied. To do this, a proper design of experiments (DOE) was created using Excel. This specific DOE helped identify what was to be tested, what data would be collected, and compared the different process settings that were studied.

To begin the DOE, a proper naming convention was established. The naming convention contains the trial number, the analysis sequence, and various other traits that were important to each study. The naming convention is important to set up because it helps save the group time when referring to a specific trial. Group members knew the identity of each trial what was being analyzed without having to view each file. After creating the naming convention, the specific mesh type was added to the DOE.

Three mesh types are used in the DOE; midplane, dual domain, and 3D. The midplane mesh consists of a plane that runs through the center of the part and is made up of various triangles and nodes (at the corners of each triangle). The nodes and triangles represent areas where the simulated material flows. Each of these nodes and triangles can be selected and used as analysis points to view temperature, pressure, and warpage at those specific locations. Each midplane mesh provides over 70,000 different triangles and nodes to choose from.

The second mesh type, dual domain, is very similar to the midplane mesh. It too is made up of triangles and nodes. However, instead of a plane being taken through the center of the part, the triangles and nodes make up the outer edges, sides, top, and bottom of the part.

The final mesh type that was selected in Moldflow was the 3D mesh, the most complex of the meshes selected. It was made up of nodes and tetrahedrals. These tetrahedrals are like pyramids in shape, but have a triangle for the bottom. The 3D mesh's tetrahedrals make up the faces of the parts, as well as the points throughout the rest of the body of the part throughout the center. After the mesh types were selected, Moldflow allowed different analysis sequences to be set. These analysis sequences made up the various stages of the injection molding process: fill, pack, cool, warp, and their combinations.

Finally, the only thing that needed to be added to the DOE is what was analyzed and studied in this project. HTC results were also added to the DOE. There were three HTC results that were adjusted in Moldflow in this study: HTC filling, HTC packing, and HTC detached values. The HTC values were changed and set throughout the DOE and analysis to see if there was any variation from trial to trial.

In Table 1, a portion of the created DOE is shown including Trials 1-15 along with the dual domain mesh type, the material, etc. Trials 1, 2 and 3 all look the same, until the HTC Value columns. Trial 1 shows control/default values for the HTC settings. In trial 2, the HTC value fill was cut in half from the default value. This drastic cut was made to determine if there was any variation caused by changing this value. In trial 3, the HTC value was doubled from the default value. Again, this was done to easily show if there was any variation by changing this value from study to study.

Trial #	Mesh Type	Material	Analysis Sequence	HTC Value (Fill)	HTC Value (Pack)	HTC Value (Detach)	Study name/Code
1	DD	TPO	F	HF5000	HP2500	HD1250	T1_DD_TPO_F_HF5000_HP2500_HD1250
2	DD	TPO	F	HF2500	HP2500	HD1250	T2_DD_TPO_F_HF2500_HP2500_HD1250
3	DD	TPO	F	HF10000	HP2500	HD1250	T3_DD_TPO_F_HF10000_HP2500_HD1250
4	DD	TPO	P	HF5000	HP2500	HD1250	T4_DD_TPO_P_HF5000_HP2500_HD1250
5	DD	TPO	P	HF2500	HP2500	HD1250	T5_DD_TPO_P_HF2500_HP2500_HD1250
6	DD	TPO	P	HF10000	HP2500	HD1250	T6_DD_TPO_P_HF10000_HP2500_HD1250
7	DD	TPO	BC	HF5000	HP2500	HD1250	T7_DD_TPO_BC_HF5000_HP2500_HD1250
8	DD	TPO	BC	HF2500	HP2500	HD1250	T8_DD_TPO_BC_HF2500_HP2500_HD1250
9	DD	TPO	BC	HF10000	HP2500	HD1250	T9_DD_TPO_BC_HF10000_HP2500_HD1250
10	DD	TPO	FC	HF5000	HP2500	HD1250	T10_DD_TPO_FC_HF5000_HP2500_HD1250
11	DD	TPO	FC	HF2500	HP2500	HD1250	T11_DD_TPO_FC_HF2500_HP2500_HD1250
12	DD	TPO	FC	HF10000	HP2500	HD1250	T12_DD_TPO_FC_HF10000_HP2500_HD1250
13	DD	TPO	BCP	HF5000	HP2500	HD1250	T13_DD_TPO_BCP_HF5000_HP2500_HD1250
14	DD	TPO	BCP	HF2500	HP2500	HD1250	T14_DD_TPO_BCP_HF2500_HP2500_HD1250
15	DD	TPO	BCP	HF10000	HP2500	HD1250	T15_DD_TPO_BCP_HF10000_HP2500_HD1250

The final column shows the naming convention that was selected. It shows the trial number, mesh, and all the settings and values that were used for that study. Overall, a total of 187 different trials were created and ran between changing HTC values and process settings.

3D CAD MODELS

CAD models created in Autodesk Inventor 2015 were used for Moldflow simulations and analysis. The main part of the body along with runner, mold cavity, mold core, and water cooling lines are imported into to Moldflow. Once the parts are imported, Moldflow requires properties to be assigned to those models. The runner system requires hot and cold runner, cold sprue, and gate. The water lines require hose properties and inlet/outlet water locations assign. Injection location on the gate must be placed on the runner system, which determines where the material enters the mold. Then we compile all parts together to create a model for simulation.

To set up and prepare the simulations, Moldflow needs to analyze a part. We imported the 3D CAD model with assigned properties into Moldflow and created the mesh. DOE determines all the mesh types, analysis sequence, and HTC values. Once all the sequences were selected and value changed, it was time to run the study.

USING THE CLOUD

For this project, there was an option of running the studies locally on individual's computers, or in the cloud. This project used the cloud for several reasons. For example, the cloud was selected because it made it easier for the group members to retrieve and store information to a single location. Multiple people could add or download different studies at the same time.

Another reason why the cloud was used was because of the amount of data needed to run each trial: nearly 95GBs of information from just running the analysis alone, which could slow down a computer. By using the cloud, the team was able to run the trials while being able to do other tasks needed for this project. Once the trials finished running, the cloud then sent the finished results right back to the group's computers. The cloud saved space on the group's computers and eliminated the need for extra storage. The group didn't have the time nor the money to purchase additional CPUs or RAM storage.

Using the cloud, nearly 3.6 million CPU seconds were used, which is equivalent to walking from Michigan to Los Angeles, California, and halfway back. Without the cloud usage, the analysis for this project would probably still be running and no data would have been accessed yet. The cloud was very easy to use and was easy to access from any location that had Internet access.

Checking to make sure the analysis was completed in the cloud; the first analysis was a Fill time simulation which allows the user to check and make sure all models and flow directions are working properly, and to verify the process settings that were entered are being matched in the simulation. When the cloud studies are complete and the model is working properly inside the Moldflow, the analysis results can be obtained.

NODE LOCATIONS

Before the construction of single graphs in Moldflow, the locations of the nodes had to be discovered. Without the nodes track the data on the plaque would not be possible. Verifying the node locations is crucial because each node provides different information. To obtain a sufficient variation of data in the nodes, a node zone system was developed. Within each zone it had its own individual node to make sure that data is being tracked throughout the whole plaque. Below is the schematic of the node zones that were used for the project.

Figure 1

CREATING MOLDFLOW GRAPHS

Once the nodes are properly laid out, the creation of the graphs in Moldflow starts. Each trial had an average of four to nine graphs to be created. The purpose of the different types of graphs is to figure out what heat transfer coefficients actually have an impact. From the numerous trials that were run, there were an estimated total of a thousand graphs that had to be developed. Listed below are some of the graphs that had to be created:

- Time to reach injection temperature
- Temperature of part and mold(cavity/core side)
- Volumetric shrinkage
- Deflection (warpage)

GRAPH PLOT LAYOUT

Before the graphs can be fully created, a plot system had to be selected. The two plot systems that were used for this experiment were XY plot and Path plot. The path plot layout shows the Moldflow graph with one trending line from node to node. Using the XY plot gives you a graph that shows separate trending lines from each node. If the wrong plot system is used in the creation of the graphs, it will be hard to read and understand the graphs. Below are examples of what the graphs look like using XY and Path plot layout.

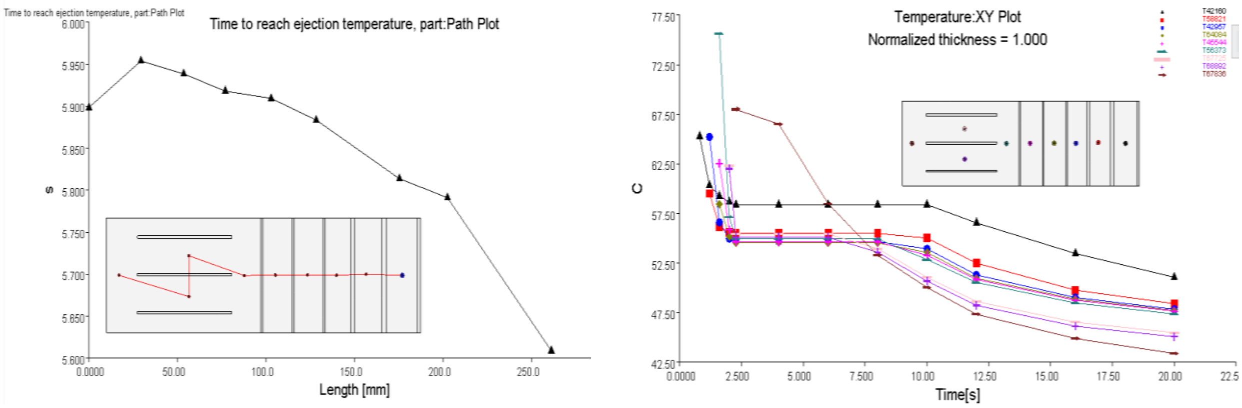


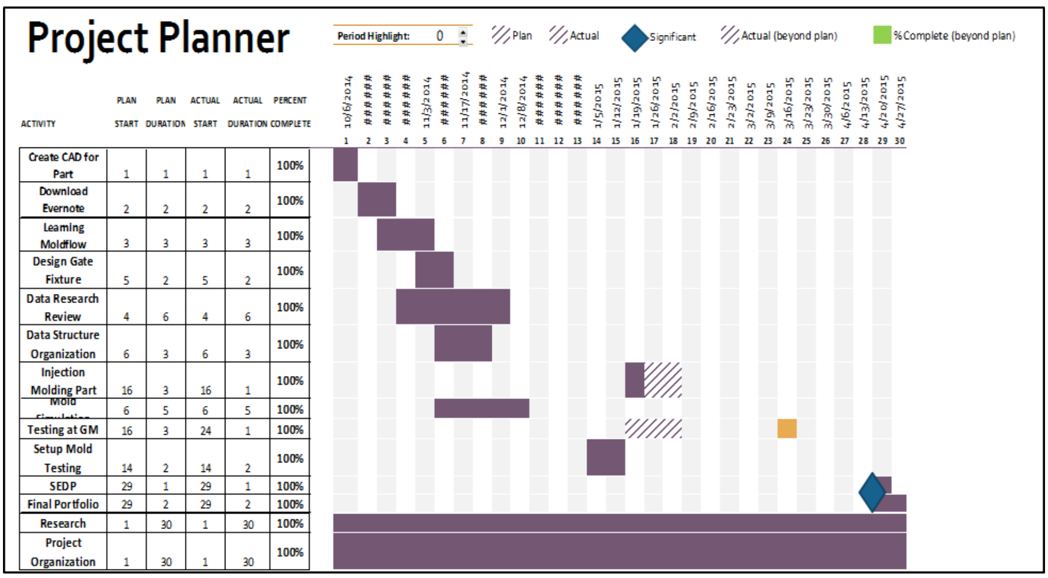
Figure 2

Using Excel Macro

After finishing all the graphs in Moldflow, there needed to be a way to compare multiple graphs at once. The data from the Moldflow graphs were extracted and converted to text files so they could be uploaded into a macro on Excel to help with this issue. In the macro it only takes a couple clicks to upload and combine multiple trials to one graph. The problem with just looking at graphs in Moldflow is that there can be small increments between the graphs that are hard to see with the naked eye. By combining all the graphs using the macro it is easier to see all the variations between the graphs. This macro played an important part in our observation of the data – without the macro the time it took to look through all the graphs would have probably tripled.

TIMELINE

This has been an ongoing project over the past eight months. Many components went into obtaining our final results to present to General Motors. Below shows a Gantt chart of this project, illustrating all of the different components and the length of time each required. For this project the SEDP and final portfolio were marked as significant milestones. The milestone indicates the completion of the first stage in research for this project. All information to this date will be presented and displayed for public knowledge.



WARPAGE AND SHRINKAGE RESULTS

General Motors is particularly interested in understanding the influence HTC values have on the warpage and shrinkage predictions. The team investigated all the volumetric shrinkage and deflection (warpage) results. To determine if there is an influence on warpage and shrinkage, the study will need to show variation between the trials. No variation between trials indicates the HTC values do not influence the warpage and shrinkage predictions; variation between trials determines that the HTC values are influencing the predictions.

Looking in-depth at figure 3 volumetric shrinkage graphs, it can be determined which coefficient values influence the prediction. The fill coefficient values show variations between the trials and determine the values that influence the shrinkage prediction. The pack values also show variation between trials and influence the prediction as well. Looking at the detached values indicates no variation between the trials and includes no influence on shrinkage.

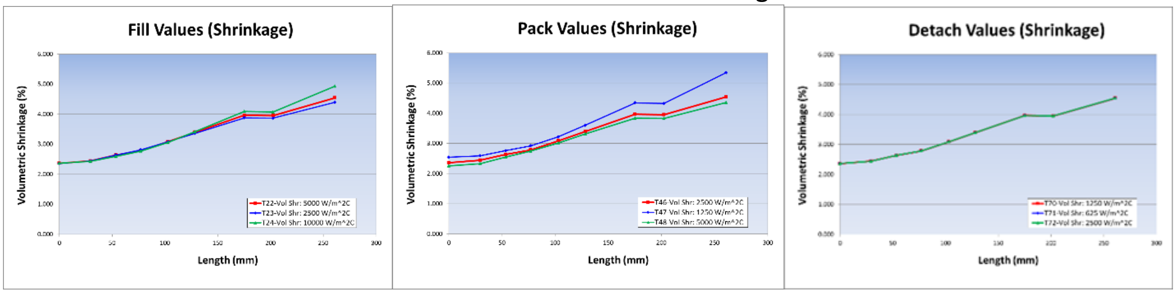


Figure 3

Looking at the warpage graphs in figure 4, you can determine if the HTC influences the prediction using the same methods as shrinkage. The fill coefficients values show variation between the individual trials overlaid on a single graph. This variation indicates the fill values influence the warpage prediction. The pack values also show variation between the trials, which indication influence on warpage. The detach values do not show variation between the trials and are not an influence on the warpage prediction.

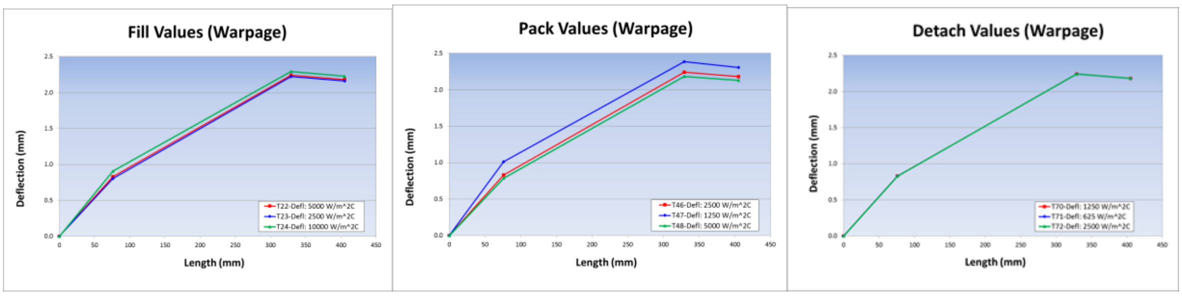


Figure 4

In conclusion, when the information from the graphs is compared there is sufficient data to determine that HTC values for fill and pack cause variation between trials. Changing the HTC values for pack had the most influence on variation, while the fill values had moderate influence. The HTC detached values had no variation and no influence on the warpage and shrinkage prediction.

CHALLENGES AND SETBACKS

During this project, there were challenges that the group encountered which need to be addressed. Although these can be seen as negative, the group saw these more as learning points for this and future projects. Some of these challenges were learning the Moldflow software, understanding HTC, and finding the correct cloud storage to hold the analysis data.

Moldflow is a very powerful simulation software. Group members in this project had never used Moldflow, so it took time to learn its capabilities. From start to finish, group members learned not only how to setup the analyses and how to run them, but also how to gather the data once they were run. This took a lot of time, but it was time that the group needed to learn this simulation program.

The basis of the project was to learn about HTC and what can influence it. Group members again had only a basic knowledge of what HTC was. Without having taken classes on HTC and what affects it in the molding process, the group had to dedicate time to learn and fluently understand background information on HTC and former relevant studies.

Dealing with so much data and information, the group needed a place to store all of the information gathered from the studies, along with the simulations. With the project being so data heavy, it was important – and difficult – to find a massive storage location to hold everything. Having enough cloud credits to run the analysis was an issue, and most storage devices that were used filled up very quickly. The group needed to find a storage location that could be accessed by everyone and have enough space to hold all the data. This took up a lot of time and caused some frustration.

RECOMMENDATIONS

It is important to note that this project is a starting point for a much bigger project. Although there was a major mount of work completed within this project, there are still steps needed to fully prove how HTC values affect the simulations. They include comparing simulation results to reality and analyzing other variables that affect warpage and shrinkage.

Comparing simulation results to actual molding is necessary for future work. By doing so, Autodesk and General Motors will be able to better predict how the HTC values relate to actual molding parameters. Future projects will need to match up HTC values and their outcomes with physical molding outcomes for warpage and shrinkage. By matching the most realistic HTC value settings, injection molders that wish to use Moldflow can better predict warpage and shrinkage in the mold without having to make molding modifications. This would mean that future analysis would have to match the node locations with the thermocouple locations in General Motors molds.

Although it has been confirmed that HTC fill and pack values show variations between changing their values, there may be other factors that affect warpage and shrinkage predictions that were not looked at. This project focused on a single TPO material. Other materials will be affected differently by these changing HTC values. The material that the mold is made out of will also influence these HTC values and warpage/shrinkage predictions.

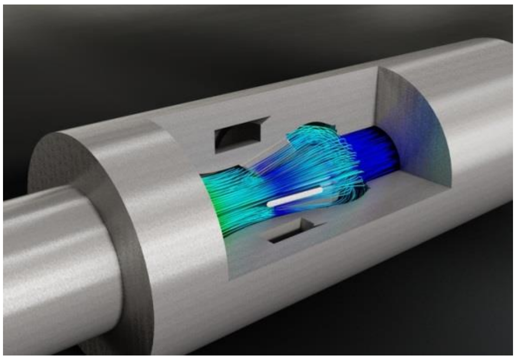
ACKNOWLEDGEMENTS

This research would not have been possible without the guidance and support from our key advisors: David Okonski, Syed Rehmathullah, Jay Shoemaker, and Betsy Aller.

Case Study Authors – Jay Shoemaker, Adam Arb, Scott Smith, Tim Slaughter, and Tyree Grasty

Autodesk Simulation CFD 360 (SimCFD)

Virtual Testing of Severe Service Control Valve



> *". . . sending solutions to the cloud could reduce the time to collect a complete data set by an order of magnitude."*

MEET THE TEAM

End User - Mark A. Lobo, P.E. - Principal, Lobo Engineering PLC. Mark Lobo has experience in industrial product engineering and analysis spanning 30 years of modeling and analysis software evolution and dramatic increase in computer workstation power, with special focus on control valve design and application.

Software and Resource Provider – Autodesk provides the fluid flow simulation software Autodesk Simulation CFD 360 (SimCFD) and supporting cloud infrastructure.

HPC/CAE Application Expert – Software, resources, and support provided jointly by Jon den Hartog, P.E., Autodesk Sr. Product Line Manager and Heath Houghton, Autodesk Simulation CFD Product Manager.

USE CASE

Flow control valve specifications include performance ratings in order for a valve to be properly applied in fluid management systems. Control systems sort out input parameters, disturbances and specifications of each piping system component to react and produce a desired output. System response is chiefly a function of the accuracy of control valves that respond to signals from the control system. Valve performance ratings provide information to the system designer that can be used to optimize control system response.

Good valve engineering practice requires a thorough understanding of the relationship between internal physical configuration and performance rating. The engineer can then optimize a control valve design for the intended service. Empirical relationships established through testing provide guidance to the designer to create suitable flow path components. Physical testing of prototype models is required to verify performance goals are met, and in-service testing under worst case conditions is required to validate service life expectations.

As shown in the Figure above, a control valve model with idealized flow path was used in this UberCloud experiment to minimize effects of a complex body cavity and trim design. The control valve restriction components or "trim" reduces the annular area as the cavity profile on the right moves to the left. The location of highest velocity is indicated in red. The application domain is shown in Figure 2.

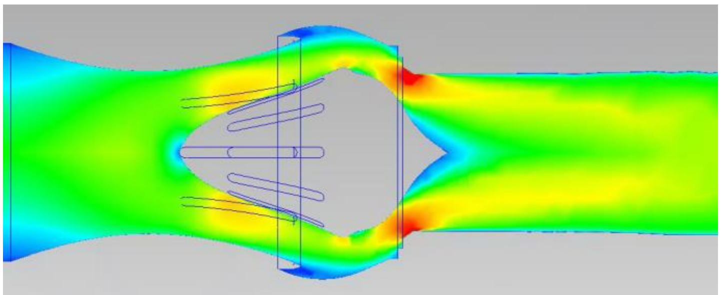


Figure 2: Application Domain

Design and testing of control valves has benefited from the development of 3D modeling software and computer-based test equipment. CFD analysis tools now enable close, quantitative observation of the flow path and guide the designer's decision process. CFD has become more accessible to engineers, however cost and resource barriers have traditionally limited broad adoption.

SimCFD software includes a vast array of tools that the valve engineer can apply actively to optimize product design. Intuitive controls, robust help features, and deep online support allow rapid adoption as a tool in the designer's toolbox.

CHALLENGES

Control valves by nature are required to perform in a variety of conditions through the range of system demands. Predictable restriction verses input signal range, the inherent characteristic, requires hundreds of simulations to verify. And design changes driven by the results require more simulations to verify the sought after performance improvement. The designer/engineer/analyst with a single workstation is challenged to make best use of this computing resource yet is constrained by usability issues of software sharing local resources.

SimCFD used in this project employs solution methods based on well understood fluid dynamics principles. The iterative approach to solving the embedded fluid flow equations is presented graphically to provide a qualitative appraisal of the program's progress converging toward a solution. This can be used to judge the appropriateness of the software setting scheme, including mesh controls, solution methods, advection model and more.

SimCFD gives the user the option of solving problems locally or in the cloud. After acceptable results are realized locally, with confidence in the program settings, multiple scenarios can be sent to the cloud, freeing up local resources while reducing analysis time.

The engineer / analyst can view the progress of the iterations by viewing the data produced as SimCFD seeks convergence. The flow rate is the average velocity normal to the inlet port times the area, therefore that velocity is the parameter of importance. The appearance of the convergence curve reflects the flow rate of a physical test wherein the pressure drop increases from a minimum to that prescribed by the boundary conditions. As the iteration steps accumulate, convergence is reflected by the flattening of the curves of all simulated parameters including velocities in three directions along with temperature, density and a scalar factor.

The term "severe service" is applied to conditions of pressure drop across the control valve high enough to vaporize the fluid. The change from liquid to gas is due to the static pressure falling below the fluid vapor pressure. The low static pressure downstream of the restriction is due to increase in velocity and dynamic pressure. The phase change transition is sudden, with rapid de-compression and pressure recovery of the fluid making for a bi-stable fluid environment. This phenomenon is known as cavitation.

SimCFD simulates the onset of cavitation and predicts regions where it will begin to occur by reporting induced vapor fraction. When the cavitation zone becomes large enough local velocities will dramatically increase in the vapor region and signal that the maximum flow rate has been reached. This allows an approximation of choked flow performance.

THE VIRTUAL VALVE TEST

The valve test simulation boundary conditions are set in the same manner as would be a physical test setup – the upstream and downstream pressures are applied and SimCFD solves for the flow rate. Accuracy requirements challenge physical instrumentation and data collection methods, which in a way justify "virtual valve testing" with CFD that inherently must idealize real-world conditions. So, in the balance, a multiple-scenario CFD analysis of a control valve should be able to replace physical testing to generate performance ratings. The investment to equip a valve manufacturer's test lab would be one to two orders of magnitude greater than the resource investment required to have the capability to run a "test in the cloud" of a control valve.

During the course of this project, over 200 simulation scenarios were solved in the cloud. Flow data was collected for six control positions of a representative control valve and ten combinations of pressure drop for each position, or travel step, from 20% open to 100% open. The data was analyzed in the same manner as if it was generated through physical testing to produce the inherent characteristic for fully developed turbulent flow and the effect of choked flow on performance. The resulting performance characteristics were plotted as shown in Figure 3.

The premise of this project was not only to explore virtual valve testing, but to evaluate the practical and efficient use of CFD by the non-specialist design engineer. As a benchmark, the end user had no prior experience with the software when the project initiated, no formal training in the software, and depended on the included tutorials, help utility, thorough documentation to produce good results and good data.

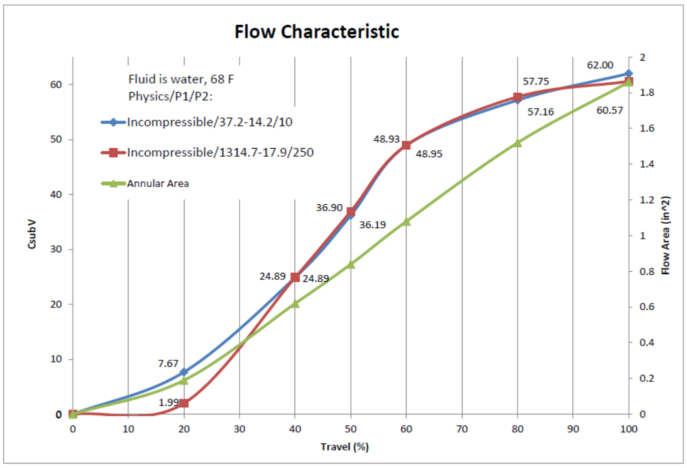


Figure 3: Valve Flow Coefficient (C_v) Versus Travel

The CAE Application Expert suggested running the solution with the cavitation setting off, a thicker wall mesh and different solution model than default. The inherent characteristic produced was acceptable and corresponded closely to predictions.

The first barrier was the performance specification for the valve design, which included a requirement that the valve be optimized for severe service choked flow control.

CHOKED FLOW PERFORMANCE INFERENCE

A severe service control valve must have the ability to maintain a predictable flow characteristic when pressure downstream is very low. Choked flow is defined as a rate that is not affected by downstream pressure; as downstream pressure varies in that region the Valve Flow Coefficient, C_V, is no longer a constant.

A condition-dependent effective C_V is a required parameter for a control system to maintain a set point in the fluid management system during choked flow conditions as other parameters change. The Liquid Fluid Pressure Recovery Factor, F_L, and Pressure Drop Ratio Factor, F_F, are employed to calculate the effective C_V and in turn permit the control system to accurately adjust the control valve as appropriate for the fluid process. F_L, and F_F are derived from valve test data generated during choked flow conditions.

The virtual valve test is challenged to simulate choked flow conditions in order to produce results that can be used to accurately derive these factors. Flow rates are tabulated throughout a full range of downstream pressure conditions while upstream pressure remains constant. During this experiment, the mesh and solution settings in the software were adjusted to find a balance that would produce acceptable convergence and reasonable results for the full range of valve restriction settings. Figure 4 reports flow rate versus the square root of differential pressure given a constant upstream pressure.

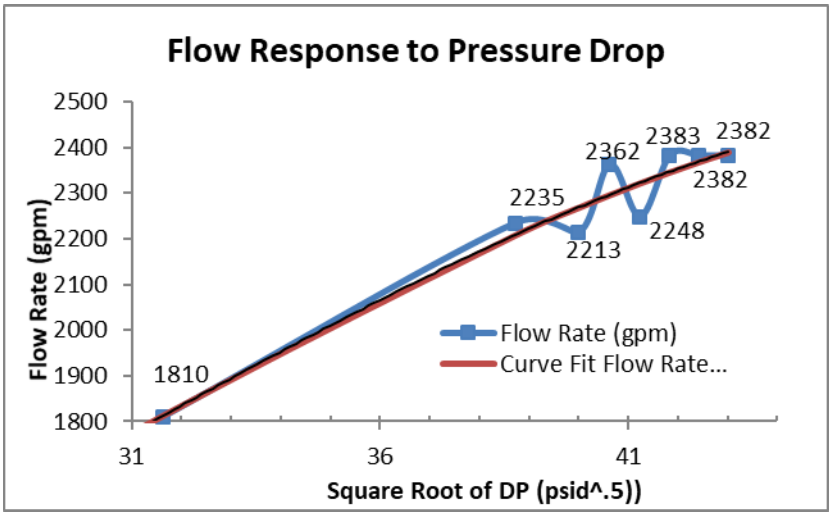


Figure 4: Flow Response Versus Change in Pressure Drop

BENEFITS

In many cases the most important output from a simulation is comparative data that allows engineers to review trends. These trends may concern the operating conditions or changes in the design geometry that affect overall performance. Either way, several simulations must be run in order to perform these comparisons. However, due to the computational requirements for each simulation, engineers often cannot set up and run the spectrum of simulations they would like within the time available. Limited hardware and software licenses often act as the bottleneck in a simulation project leaving engineers to virtually test only a limited number of cases.

It is possible to get around this problem by making a significant investment in computing hardware and software licenses in the form of HPC clusters and simulation solvers. This approach is appropriate for specialized simulation analysts at large organizations that can maintain high enough utilization on the hardware and software to get a good return. However, the vast majority of engineers have fluctuating demands for simulation that depend upon their project work. Most engineers involved in design work have intermittent simulation needs, but when they do require it they demand hardware and licenses as would any full-time analyst. Historically it hasn’t been easy for these engineers to get access to the simulation power they need at a reasonable cost.

Cloud computing enables the design engineer to access a large amount of computing power in a cost effective way. Rather than owning the hardware and software licenses, engineers can pay for what they need when they need it rather than making a substantial upfront investment.

In this project, over 200 simulations were run in the cloud. Given the runtimes involved and allowing for data download upon completion of the runs, it is possible for all of these simulations to be solved within a day. It would be possible to match this performance with a large scale computer cluster but would require a substantial upfront investment. For an engineer with 1 simulation license on a single workstation, this would have required 800 hours (approximately 30 days) to complete if the simulations were running nonstop one after another. Table 1 compares the approximate time and investment that would be required for various solving approaches.

Table 1: Comparison of Desktop, Cloud, and HPC Solving Options

Simulation Solving Approach	Approx Time to Complete	Investment Required
Single Desktop Machine + Local Computing	800 hours (1 month)	Engineering Workstation + Simulation SW License
Single Desktop Machine + Cloud Computing	24 hours (1 day)	Engineering Workstation + Simulation SW License + $1200 Cloud Compute Fee
Single Desktop Machine + Private HPC Cluster + Multiple Solver Licenses	24 hours (1 day)	Engineering Workstation + Simulation SW License + 30 Node Compute Cluster + 30 Simulation Solver Licenses

SimCFD can be licensed to solve locally on the engineer's workstation, remotely on one or more individual machines inside the engineer's network, remotely on a private computing cluster inside the engineer's network, or remotely in the Autodesk® cloud. This project required one license that supported both local solving on the end user's workstation and solving on the cloud. When simulations are solved in the cloud, the engineer is not limited by licensing and may solve as many concurrent simulations as desired.

We observed that running solutions in the cloud permitted solution time reduction from local solution time by at least a factor of three, with the local workstation dedicated to the CFD analysis. During the project, we also observed local solution times increasing by another factor of three when the local workstation was in use for intensive modeling work while a simulation was running. If a workstation is required to be actively used for other work during simulations, sending solutions to the cloud could reduce the time to collect a complete data set by an order of magnitude.

CONCLUSIONS AND RECOMMENDATIONS

1. The user must have technical knowledge and experience background in fluid dynamics to judge the quality of the virtual testing results.
2. Training is required, with examples focused on the required application of the software. Tutorials and robust help utility are crucial components of training.
3. Mentoring and support must be available to help the design engineer navigate the vast array of solution settings in the program.
4. Characterization studies will require many simulations to complete. Given the need to evaluate multiple configurations of the flow path to identify the optimum design, approximately 25 to 50 scenarios are required. What may take days to solve on a single workstation could be solved in hours on the cloud.

REFERENCES

The cloud solves were performed on a pool of Amazon EC2 M2.4x large instances, each with 8 physical cores and 64.8 GB RAM.

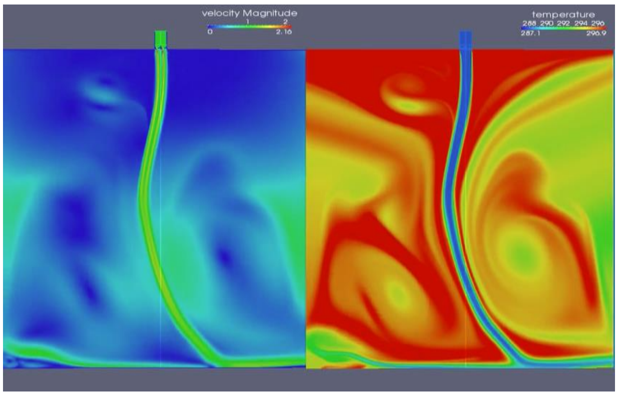
The local solves were performed on a HP Z800 Dual 2.4GHz Quad Core E5620 Workstation, 24GB RAM, w/ATI FirePro V5800 1GB Graphics Adapter.

Applicable valve testing procedures include ISA 75.01.01 Flow Equations for Sizing Control Valves, ISA 75.02.01 Control Valve Capacity Test Procedures, and ISA 75.11.01 Inherent Flow Characteristic and Rangeability of Control Valves.

Case Study Authors – Mark A. Lobo, Jon den Hartog

CAELinux

CFD Coupling With Heat Transfer



"Concerning the ease of using cloud computing resources, we concluded that this working methodology is very friendly and easy to use through the CloudBroker Platform."

MEET THE TEAM

- Lluís M. Biscarri, Director, and Pierre Lafortune, CAE Expert, Biscarri Consultoria SL
- Wibke Sudholt and Nicola Fantini, CloudBroker GmbH
- Joël Cugnoni, researcher and developer of CAELinux
- Peter Råback, CSC — IT Center for Science, Development Manager

USE CASE

There are many engineering problems where fluid dynamics is coupled with heat transfer as well as many other multiphysics scenarios. The simulation of such problems in real engineering cases tends to produce very big numerical models to be solved, so that big computational power is required in order for simulation cycles to be affordable. For SME industrial companies in particular it is hard to implement this kind of technology in-house, because of its investment cost and the IT specialization needed.

There is great interest in making these technologies available to SME companies, in terms of easy-to-use HPC platforms that can be used on demand. Biscarri Consultoria SL, as the end user of this project, is committed to disseminate parallel open source simulation tools and HPC resources in the Cloud.

CloudBroker is offering its platform for various multiphysics, fluid dynamics, and other engineering applications, as well as life science applications for small, medium and large corporations along with related services. The CloudBroker Platform is also offered as a licensed in-house solution.

Current State

Biscarri Consultoria SL is exploring the capabilities of cloud computing resources for performing highly coupled computational mechanics simulations, as an alternative to the acquisition of new computing servers to increase the computing power available.
For a small company such as BCSL, the strategy of using cloud computing hardware resources to cover HPC needs has the benefit of not needing an IT expert to maintain in-house parallel servers. The idea is to concentrate our efforts in our main field of competence as much as possible by externalizing non-core business. To solve the needs of the end user, the following hardware and software resources already existing on the provider side were employed by the team:

- Elmer (http://www.csc.fi/english/pages/elmer), an open source multi-physical simulation software mainly developed by the CSC — IT Center for Science
- CAELinux (http://www.caelinux.com), a computer aided engineering Linux distribution including the Elmer software as well as a CAELinux virtual machine image at the AWS Cloud
- CloudBroker Platform (public version under https://platform.cloudbroker.com), CloudBroker's web-based application store offering scientific and technical Software as a Service (SaaS) on top of Infrastructure as a Service (IaaS) cloud resources, already interfaced to AWS and other Clouds
- Amazon Web Services (AWS, http://aws.amazon.com), in particular Amazon's IaaS Cloud offerings EC2 (Elastic Compute Cloud) for compute and S3 (Simple Storage Service) for storage resources

Experiment Procedure

Technical Setup

The technical setup for the HPC Experiment was performed in several steps. These followed the principle to start with the simplest possible solution and then to grow it to fulfil more complex requirements in an agile fashion. If possible, each step was first tested and iteratively improved before the next step was taken. The main steps were:

1. All team members were given access to the public CloudBroker Platform via their own account under a shared organization created specifically for the HPC Experiment. A new AWS account was opened by CloudBroker, the AWS credit loaded onto it, and the account registered in the CloudBroker Platform exclusively for the experiment team.
2. The Elmer software on the already existing CAELinux AWS machine image was made available in the CloudBroker Platform for serial runs and tested with minimal test cases by CloudBroker and Joël Cugnoni. The setup was then extended to allow parallel runs using NFS and MPI.
3. Via Skype calls, screen sharing, chatting, email and contributions on Basecamp, the team members exchanged knowledge on how to work with Elmer on the CloudBroker Platform. The CloudBroker Team gave further support for its platform throughout HPC Experiment Round 2. CloudBroker and BCSL performed corresponding validation case runs to test the functionality.
4. The original CAELinux image was only available for normal, non-HPC AWS virtual machine instance types. Therefore, Joël Cugnoni provided Elmer 6.2 separately as optimized and non-optimized binaries for Cluster Compute instances. Also, the CloudBroker Team deployed these on the CloudBroker Platform for the AWS HPC instance types with 10GBit Ethernet network backbone, called Cluster Compute instances.
5. BCSL created a medium benchmark case, and performed scalability and performance runs with different numbers of cores and nodes of the Amazon Cluster Compute Quadruple and Eight Extra Large instance types and different I/O settings. The results were logged, analyzed and discussed within the team.
6. The CloudBroker Platform setup was improved as needed. This included, for example, a better display of the number of cores in the web UI, and the addition of artificial AWS instance types with fewer cores, as well as the ability to change the shared disk space.
7. BCSL tried to run a bigger benchmark case on the AWS instance type configuration that turned out to be preferable from the scalability runs – that is, single AWS Cluster Compute Eight Extra Large instances.

Validation Case

Before performing the benchmark cases, a validation case was defined to test the whole simulation procedure. The validation case was intentionally simple, but had the same general characteristics as the more complex problems that were used for the rest of the experiment. It was an idealized 2D room with a cold air inlet on the roof (T = 23ºC, V = 1m/s), a warm section on the floor (T = 30ºC, V = 0.01m/s) and an outlet on a lateral wall near the floor (P = 0.0Pa). The initial air temperature was 25ºC.

The mesh was created with Salome V6. It consists of 32,000 nodes and 62,000 linear triangular elements. The solution is transient. Navier-Stokes and Heat equations were solved in a strong coupled way. No turbulence model was used. Free convection effects were included. The mesh of the benchmark analysis was a much finer one of the same geometry domain, consisting of about 500,000 linear triangular elements. The warm section on the floor was removed and lateral boundaries hade open condition (P = 0.0Pa).

Job Execution

The submission of jobs to be run at AWS was done through the web interface of the CloudBroker Platform. The procedure was as follows:

- A job was created on the CloudBroker Platform, specifying Job Name, Software, Instance Type and AWS Region
- Case and mesh partition files were compressed and uploaded to the CloudBroker Platform attached to the created job
- The job was submitted to the selected AWS resource
- Result files were downloaded from the CloudBroker Platform and postprocessed in a local workstation
- Scalability parameters were calculated from job output log file data

Team Roles and Communication

Concerning the setup of the project team, the organizers of the HPC Experiment, Wolfgang Gentzsch and Burak Yenier, were very successful in putting Team members together to work on this project:

- Software providers: Peter Råback / CSC-Elmer and Joël Cugnoni / CAElinux
- Cloud computing resource provider: AWS
- Middleware software provider: CloudBroker
- Team experts and coordinators: Wibke Sudholt and Nicola Fantini / CloudBroker
- End User: Pierre Lafortune and Lluís Biscarri / Biscarri Consultoria SL

The BCSL team was essential for formulating the project goals, creating the test cases, and performing the scalability and performance runs.
The advice of Peter Råback was a great help to better understand the Elmer operation and to find the best numerical treatment for the case provided by the end user.
The work of Joël Cugnoni in the development of CAElinux facilitated a lot of the installation of the software Elmer 6.2 in AWS via the CloudBroker Platform.
With their excellent work, the rest of the team made significant contributions to the success of the project.

The CloudBroker Platform has proved to be a very useful tool to facilitate project planning, job definition, submission and monitoring. Project coordination, information interchange and project discussions have been developed in a very easy and friendly way using the project management platform Basecamp. In addition, weekly calls via Skype were performed, partly including screen sharing and chatting. Email and Google Docs were used as further communication channels.

CHALLENGES

The first challenge for BCSL in this project was to learn if the procedure to run Elmer jobs in a cloud computing resource such as AWS is easy enough to be a practical alternative to in-house calculation servers. The second challenge was to determine the level of scalability of the Elmer solver running at AWS. Here we encountered good scalability when the instance employed has only one computational node. When running a job on an instance with two or more computational nodes the scalability is reduced dramatically, showing that communications between cores of different computational nodes slows down the process. AWS uses 10Gbit Ethernet as backbone network, which seems to be a limitation for this kind of simulations.

After the scalability study with the mesh of 500 Kelems was performed, a second scalability test was tried with a new mesh of about 2000 Kelems. However, jobs submitted for this study to Cluster Compute Quadruple Extra Large and Cluster Compute Eight Extra Large instances have not been successfully run yet. Further investigations are in progress to better characterize the network bottleneck issue as a function of problem size (number of elements per core) and to establish if it is related to MPI communication latency or NFS throughput of the results.

Resource Provider and Team Expert
On the technical side, most challenges were mastered by already existing features of the CloudBroker Platform or by small improvements. For this it was essential to follow the stepwise agile procedure as outlined above, partly ignoring the stiffer framework suggested by the default HPC Experiment tasks on Basecamp.

Unfortunately, AWS HPC Cloud resources are limited to a 10 GBit Ethernet network. 10 Gbit Ethernet was not sufficient in terms of latency and throughput to run the experiment efficiently on more than one node in parallel.

The following options are possible:

1. Run the experiment on one large node only, that is the AWS Cluster Compute Eight Extra Large instances with 16 cores
2. Run several experiment jobs independently in parallel with different parameters on the AWS Cluster Compute Eight Extra Large instances
3. Run the experiment on another Cloud infrastructure which provides low latency and high throughput using technology such as Infiniband

The CloudBroker Platform allows for all the variants as described above. Variants 2 and 3 were not part of this experiment, but would be the next reasonable step to explore in a further experiment round. In the given time, it was also not possible to try out all the different I/O optimization possibilities, which could provide another route to improve scalability.

A further challenge of the HPC Experiment was to bring together the expertise from all the different involved partners. Each of them has experience on a separate set of the technical layers that were needed to be combined here (actual engineering use case, Elmer CAE algorithms, Elmer software package, CloudBroker Platform, AWS Cloud).

For example, often it is difficult to say from the onset which layer causes a certain issue, or if the issue results from the combination of layers. Here it was essential for the success of the project to stimulate and coordinate the contributions of the team members. For the future, we envision making this procedure more efficient through decoupling – for example, by the software provider directly offering an already optimized Elmer setup in the CloudBroker Platform to the end users.

Finally, a general challenge of the HPC Experiment concept is that it is a non-funded effort (apart from the AWS credit). This means that the involved partners can only provide manpower on a “best effort” basis, and paid projects during the same time usually have precedence. It is thus important that future HPC Experiment rounds take realistic business and commercialization aspects into account.

BENEFITS
Concerning the ease of using cloud computing resources, we concluded that this working methodology is very friendly and easy to use through the CloudBroker Platform.

The main benefits for BCSL regarding the use of cloud computing resources were:
- To have external HPC capabilities available to run medium sized CAE simulations
- To have the ability to perform parametric studies, in which a big number of small/medium size simulations have to be submitted
- To externalize all IT stuff necessary to have in-house calculation servers

For CloudBroker, it was a pleasure to extend its platform and services to a new set of users and to Elmer as a new software. Through the responses and results we were able to further improve our platform and to gain additional experience on the performance and scalability of AWS Cloud resources, particularly for the Elmer software.

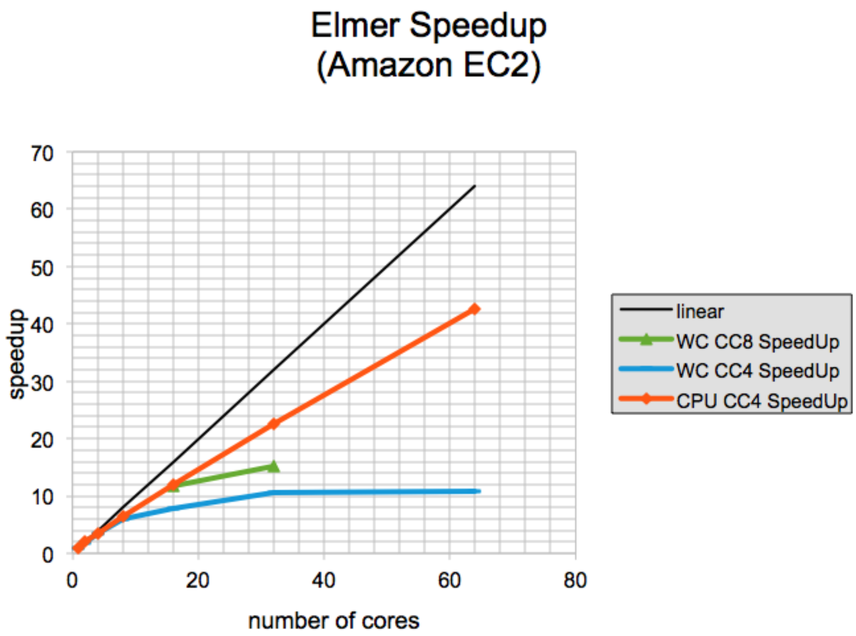
CONCLUSIONS AND RECOMMENDATIONS
The main lesson learned at Biscarri Consultoria SL arising from our participation in HPC Experiment Round 2 is that collaborative work through the Internet, using on-line resources like cloud computing hardware, Open Source software such as Elmer and CAElinux, and middleware platforms like CloudBroker, is a very interesting alternative to in-house calculation servers.

A backbone network such as 10Gbit Ethernet connecting computational nodes of a cloud computing platform seems not to be suitable for computational mechanics calculations that need to be run on more than one large AWS Cluster Compute node in parallel. The need for network bandwidth for the solution of strongly coupled equations involved in such simulations makes the use of faster network protocols such as Infiniband necessary to achieve time savings when running it in parallel on more than a single AWS Cluster Compute instance with 16 cores.

For CloudBroker, HPC Experiment Round 2 has provided another proof of its methodology, which combines its automated web application platform with remote consulting and support in an agile fashion. The CloudBroker Platform could easily work with CAELinux and the Elmer software at AWS. User requirements and test outcomes even resulted in additional improvements, which are now available to all platform users.

On the other hand, this round has shown again that there are still needs – for example, a reduction of latency and improvement of throughput (i.e., by using Infiniband instead of 10 GBit Ethernet) to be fulfilled by dynamic Cloud providers such as AWS regarding highly scalable parallel HPC resources. Their Cloud infrastructure is currently best suited for loosely or embarrassingly parallel jobs such as parameter sweeps, or highly coupled parallel jobs limited to single big machines. Finally, despite online tools, the effort necessary for a project involving several partners like this one should not be underestimated. CloudBroker expects though that in the future more software like Elmer can be directly offered through its platform in an already optimized way, making usage more efficient.

APPENDIX
Scalability Benchmark: The results of the scalability test performed up on to 64 cores at AWS cloud computing resources are shown below:



Elmer log data give two values of running time: the time strictly consumed by CPUs, and the wall clock time that the simulation has lasted, which is the time AWS charges for. The wall clock time is equal to CPU time plus the time that is spent in communication and writing to disk.

The curves in the plot correspond to:

- Black line is the ideal scalability, where speedup is proportional to the number of cores used
- Orange curve is the CPU time
- Blue curve is the wall clock time corresponding to the jobs submitted to CC4 instances (Cluster Compute Quadruple Extra Large)
- Green curve is the wall clock time corresponding to the jobs submitted to CC8 instances (Cluster Compute Eight Extra Large)

From the curves in the plot the following may be concluded:

- CC4 jobs show good scalability up to 8 cores, which is the maximum number of cores in one CC4 node
- CC8 jobs show good scalability up to 16 cores, which is the maximum number of cores in one CC8 node
- Scalability is acceptable when the backbone network has not to be used; if more than one computational node is used the speedup drops dramatically as it can be observed in the green and blue curves
-

Model Used in the Simulations

This figure shows the model employed in the scalability benchmark. The image on the right shows the temperature field, while the left image shows the velocity field, at a certain time of the transient simulation.

Case Study Authors – Lluís M. Biscarri, Pierre Lafortune, Wibke Sudholt, and Nicola Fantini

CAPVIDIA FlowVision

Water Flow Around a Self-propelled Ship

"The Introduction of reliable CFD methods in the practice of ship design promises significant decreases in costs and time."

MEET THE TEAM

End User – Andrew Pechenyuk is with Digital Marine Technology (DMT). DMT was established in 2002 by a group of specialists in the field of shipbuilding, ship repair and computer technologies. Today the main activities are: ship hydrodynamics such as hull form design; ship propulsion calculations, cargo stowage and sea-fastening projects such as heavy lift transportation projects, and strength calculations.

Software Provider – Andrey Aksenov, TESIS. Capvidia/ТЕСИС is an international company offering advanced engineering products and services. FlowVision CFD software has been under development since 1991 by a team from the Russian Academy of Sciences, Institute for Computer-Aided Design, Institute for Mathematical Modeling, and Computing Centre. In 1999, the team joined Capvidia/ТЕСИС and formed the CFD department. At Capvidia/ТЕСИС, FlowVision is being developed further and commercialized. The first commercial version of FlowVision was released in March 2000.

Resource Provider – Jesus Lorenzana, Foundation of Supercomputing Center of Castile and León. FCSCL is a public entity created by the Regional Government of Castile and León and the University of León, to improving research at the university, the researching centers, and the companies of Castile and León.

HPC Expert – Adrian Jackson, EPCC, University of Edinburgh. EPCC is a leading European centre of excellence in advanced research, technology transfer and the provision of high-performance computing services to academia and industry. Based at The University of Edinburgh, it is one of Europe's leading supercomputing centers.

USE CASE

The goal of the project was to run a CAE simulation of a self-propelled ship with a true geometrical model of the propeller. The type of simulation remains a serious challenge for the use of CFD.

The simulation of a self-propelled ship is a well-known problem of ship hydrodynamics. The simulation results provide the characteristics of interaction between hull and propeller, and are used in ship design. Traditionally the characteristics of interaction have been determined using model tests in towing tanks. Now CFD researchers are attempting to replicate and potentially replace model experiments by numerical simulation. The introduction of reliable CFD methods in the practice of ship design promises significant decreases in costs and time.

However, in this case, the CFD approach required relatively large computational resources. There are two main ways to simulate the propeller – either as a simplified body force model; or using true propeller geometry. The simulation best meets the conditions of the experiment in the latter case. But working with the complex geometry of a marine propeller requires very fine grids. Therefore, only HPC clusters are able to solve the simulation given the required RAM volume and a reasonable period of computation.

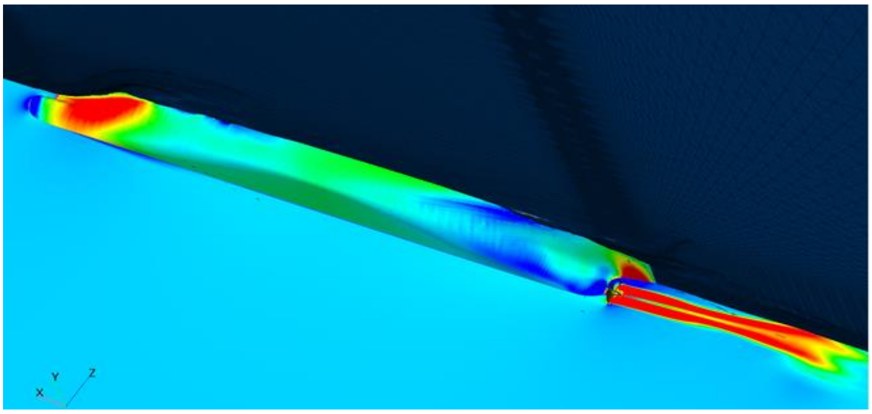


Figure 1: Pressure distribution on the ship surface, velocity distribution in the central plane and the free surface. The ship flow (pressure distribution and free surface) is not yet fully developed, because the period of simulation was relatively small. The propeller flow takes less time for formation – therefore a long jet behind the ship (velocity distribution) can be seen.

CHALLENGES

This Round 3 project was based on the successful Rounds 1 and 2 projects (CAE simulation of water flow around a ship hull). The initial data was mostly the same, and we used the same software and resources. Despite this, the team met some challenges and barriers that resulted in time losses and the necessity to continue the project in Round 4.

There were two main challenges during Round 3. The first was unexpected changes in the cluster configuration. CAE FlowVision provides connection between the client side and solver side through a local network protocol. In our case, the solver on the cluster is available only through the SSH protocol. Special SSH tunnels are used for matching the software and resources requirements. If some changes in the cluster configuration (head node's IP or port) take place, the SSH tunnels become unworkable. This problem was not critical and was quickly resolved by reassignment of the SSH tunnels. But unaided diagnostics and resolving of these kinds of problem require personnel with adequate skills in cluster interfaces.

The second problem concerned the CAE FlowVision technologies that we planned to use for the project. Initially we planned on using a sliding mesh for propeller simulation. Theoretically a sliding mesh is most effective in this case, because the computational grid may be static in relation to the propeller. Unfortunately, in the current version of CAE FlowVision this new capability is not stable enough, especially when used with free surface (VOF method) in a complex project. As a result, the use of a sliding mesh for propeller simulation was unworkable.

To resolve this problem, the software provider suggested an alternative approach to propeller simulation. This involved moving body technology where the propeller geometry revolved in relation to the grid (computational grid updates on the propeller boundaries at every iteration) and the ship.

BENEFITS

From the end-user's point of view, the main benefit of the solution arrived at in Round 3 is a more stable solution process. An additional benefit is some simplification in the geometry of the computational domain, grid formation and simulation model.

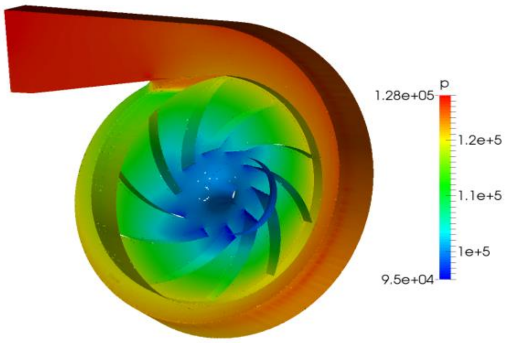
CONCLUSIONS AND RECOMMENDATIONS

The solution was developed just before Round 3 ended, and the updated project is being continued in Round 4 of the *UberCloud HPC Experiment*, to be able to answer a few more open questions concerning simulations in the cloud.

Case Study Author – Andrew Pechenyuk

CFDSupport TurboCFD

Radial Fan CFD Simulation in the Azure Cloud



"The Azure cloud together with UberCloud Containers provides excellent performance for Turbomachinery CFD users who want to obtain higher throughput or analyze more complex models."

MEET THE TEAM

End User & Team Expert – Luboš Pirkl, Co-founder & Technical Director, CFD Support ltd.
Software Provider – Turbomachinery CFD, CFD Support ltd.
Resource Provider – Microsoft Azure Cloud
Technology Experts – Hilal Zitouni Korkut and Fethican Coskuner, UberCloud Inc.

USE CASE

CFD Support supports manufacturers around the world with numerical simulations. One of the main CFD Support's businesses is providing full support for virtual prototyping of rotating machines: compressors, turbines, fans and many other turbomachinery. All the rotating machines need to be simulated to test, confirm or improve its aerodynamic efficiency, which has major effect on its energy consumption. Each machine design is tested many times and is optimized to find the best efficiency point. In practice these CFD simulations are very demanding, because of complexity and number of simulations to run.

CFD Support is aiming at demonstrating the use of UberCloud's OpenFOAM offer on Microsoft Azure to be able to scale the number of parallel simulations to minimize the "time to market" of its complex consultancy projects. On the business side this will result in performing simulations in greater detail (resulting in higher quality turbomachineries and increased competitiveness), and much faster because of using many more and faster cloud resources (resulting in faster time to market and increased competitiveness).

In this project, we calculated the radial mechanical fan characteristics in the Azure Cloud, on different number of cores. The purpose of this project is to test the current fan design and to get the best fan efficiency possible to save energy costs. The project presents a smooth workflow of performing complex CFD analysis of radial fan using Turbomachinery CFD based on the OpenFOAM® software. Detailed information for this test case is also available here:

[*http://www.cfdsupport.com/radial-fan-cfd-study.html*](http://www.cfdsupport.com/radial-fan-cfd-study.html)

The fan model is designed in CFturbo. CFturbo is a modern powerful software for interactive design of turbomachinery. It's easy to use and enables the designer to either start a model from scratch or redesign existing geometries.

The designed model data are then exported from CFturbo. The surface model data together with the initial physical flow data are loaded into Turbomachinery CFD. This CFD methodology employs a multi-region approach, which means the model is split into a certain number of regions. Each region is being computed separately and communicates with other (neighboring) regions via interfaces.

The CFD workflow is fully automated, or can be automated for every other type of machine. Finally, in effective work, the user just runs one single script; all is done automatically. The mesh is created, the set-up is done, the CFD simulation is run and finally the results are evaluated. The mesh is created automatically within snappyHexMesh. Every region has its own mesh. In this fan case there are two independent regions (meshes): rotor and stator.

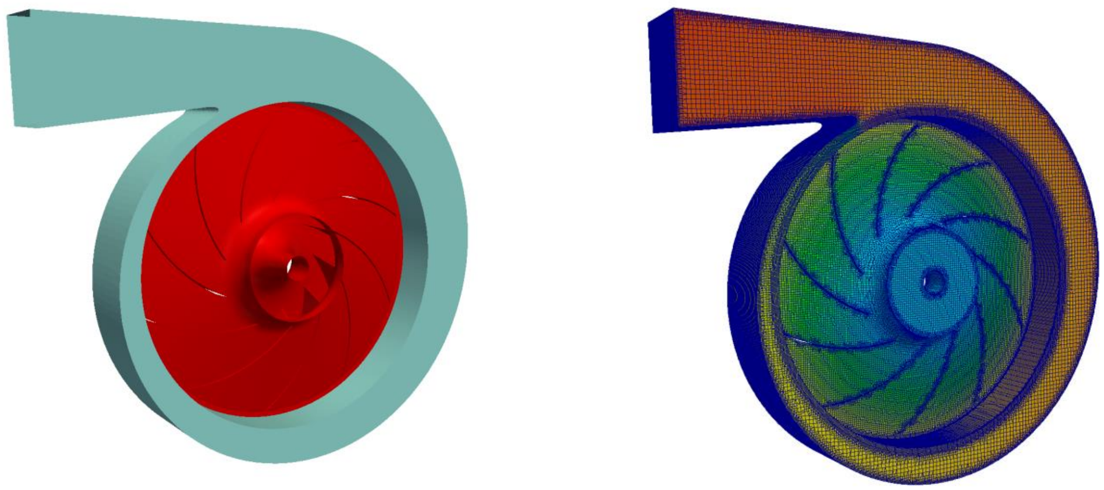


Figure 1: Turbomachinery CFD fan nq28 compressible noHousing: physical model and full computational mesh.

PROCESS AND BENCHMARK RESULTS

The initial CFD simulation set-up consists of the following steps and parameters:

- Compressible flow model
- Steady-state flow model
- Medium: air
- BEP Pressure ratio: $\Delta pTot = 1.265$ [-]
- Temperature at inlet: T = 40 [ºC]
- Viscosity: $\mu = 1.831e-5$ [Pa.s]
- Rotation speed: 2980 [RPM]
- BEP Flow Rate: 38000 [m3/h]
- Interface: mixingInterface (radial averaging)
- Turbulence Model: k-ω SST
- Mesh: snappyHexMesh, hexadominant
- Mesh Cells: 996112
- Mesh Average y+ (full/segment): 90 [-]

For more details about this CFD Simulation Set-up please see the Turbomachinery CFD Manual.

Platform	Processor	TCFD	#cores	Time [hrs]
local	Intel Core i7-3970X CPU @ 3.50GHz	v. 15.09	6	8.80
Azure Cloud	Intel Xeon E5-2698B v3 @ 2.00GHz	v. 15.09	2	16.30
Azure Cloud	Intel Xeon E5-2698B v3 @ 2.00GHz	v. 15.09	4	10.05
Azure Cloud	Intel Xeon E5-2698B v3 @ 2.00GHz	v. 15.09	8	7.45
Azure Cloud	Intel Xeon E5-2698B v3 @ 2.00GHz	v. 15.09	16	4.97
Azure Cloud	Intel Xeon E5-2698B v3 @ 2.00GHz	v. 15.09	32	2.96

Figure 2: Performance comparison for local workstation versus Azure Cloud G5 instances and different number of cores.

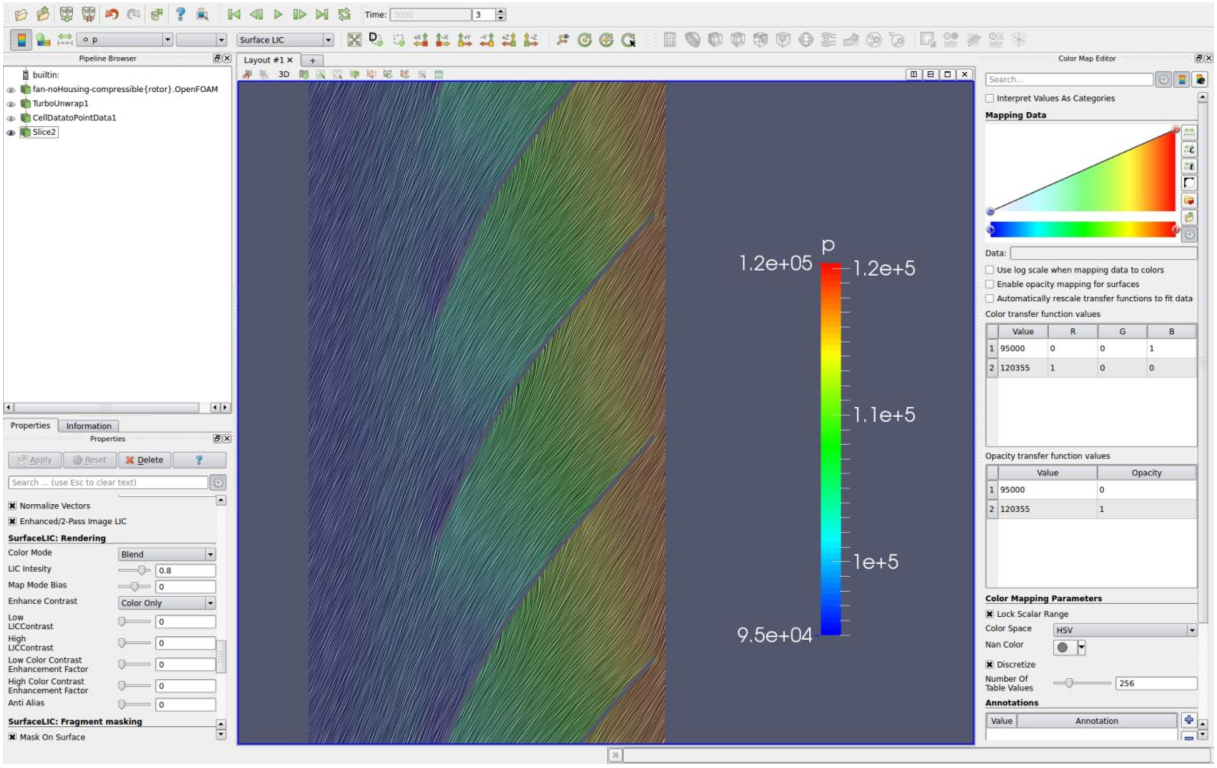


Figure 3: Turbomachinery CFD fan nq28, compressible turbo blade post streamtraces.

BENEFITS

This use case helped us understand the performance benefits offered by the Azure Cloud together with UberCloud Software Containers. CFD Support considers the UberCloud service to be a viable alternative to a powerful local workstation. Additional benefits include the on-demand access and use of the software and hardware resources, a reduction in overhead required when managing virtual instances and to maintain software updates.

No challenges were experienced in downloading project files into the Turbomachinery CFD container, running the simulation, or retrieving data. The remote desktop user interface was responsive without any significant delays. Logging into the system is simple and the Turbomachinery CFD software pre-installed in an UberCloud container runs without any user setup.

CONCLUSIONS

- We showed that the Azure cloud solution together with the UberCloud Software Containers provides excellent performance advantages for Turbomachinery CFD users who want to obtain higher throughput or analyze larger, more complex models.
- Azure Cloud and UberCloud effectively eliminate the need to maintain appropriate in-house HPC expertise.
- The container approach provides immediate access to suitable high performance computing resources and application software without software or hardware setup delays.
- The browser-based user interface is simple, robust, and responsive.

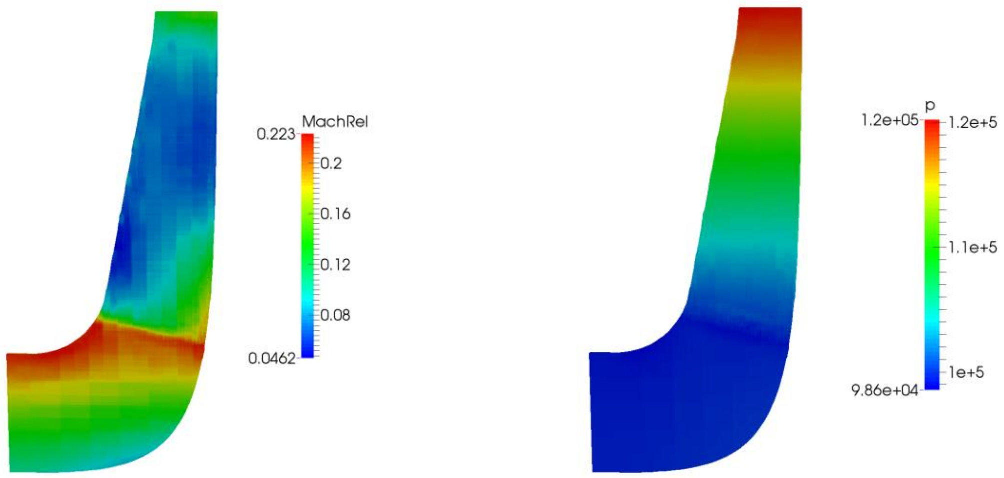


Figure 4: Turbomachinery CFD fan nq28 compressible meridional average relative Mach versus average Pressure.

APPENDIX: UberCloud Application Containers for Turbomachinery CFD

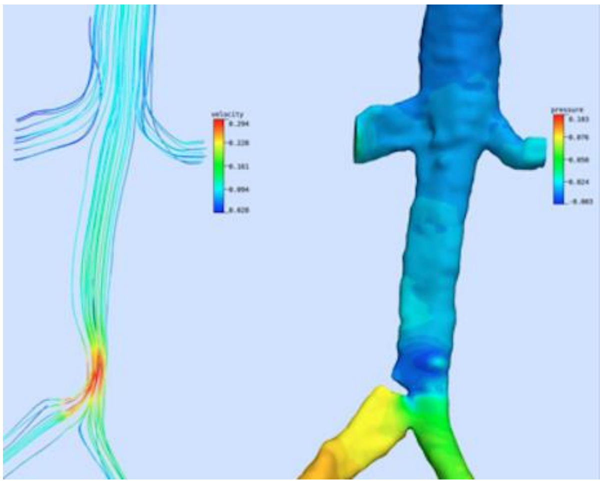
UberCloud Containers are ready-to-execute packages of fully portable software. These packages are designed to deliver the tools that an engineer needs to complete his task in hand. In this experiment, the Turbomachinery CFD software has been pre-installed in a container, configured, and tested. The software was ready to execute literally in an instant with no need to install software, deal with complex OS commands, or configure.

The UberCloud Container technology allows wide variety and selection for the engineers because the containers are portable from server to server, Cloud to Cloud. The Cloud operators or IT departments no longer need to limit the variety, since they no longer have to install, tune and maintain the underlying software. They can rely on the UberCloud Containers to cut through this complexity.

The container software technology also provides hardware abstraction, where the container is not tightly coupled with the server (the container and the software inside isn't installed on the server in the traditional sense). Abstraction between the hardware and software stacks provides the ease of use and agility that bare metal environments usually lack.

Case Study Author – Luboš Pirkl, CFD Support ltd.

CIESPACE CFD
Cardiovascular Medical Device Simulations



"Cloud-based computing resources enable a wide variety of simulation studies that otherwise would not be feasible."

MEET THE TEAM

End User – Mike Singer is the Founder and President of Landrew Enterprises, Inc. He has over 15 years of HPC experience, which includes algorithm development, simulations, and analysis. Mike provided the application used for this experiment, and performed the CFD simulations and analyses.

Software/Resource Provider – Sanjay Choudhry is the CTO at Ciespace Corporation [3]. He has over 25 years of scientific and technical computing experience and has published and presented over 40 papers in international journals and conferences on computational mechanics and numerical methods.

HPC Expert - Oleh Khoma, Head of ELEKS' HPC unit. With a background in applied mathematics and more than12 years of experience in software engineering, Oleh leads the company's HPC efforts. Viktor Chopey, Project Manager, ELEKS was also involved in the UberCloud Experiment.

USE CASE

Our project investigated flow through a patient-specific blood vessel and represents a typical use case of CFD for cardiovascular flow. The patient-specific geometry is extracted from CT image data obtained during a normal medical imaging exam. In our case, the extracted geometry, which is a triangulated surface mesh, contains the inferior vena cava (IVC), the right and left iliac veins, and the right and left renal veins (Figure 1).

The iliac and renal veins provide inflow into the IVC, and our primary interest is flow through the IVC. Boundary conditions, which are computed from clinical flow measurements, are prescribed on the faces of the iliac and renal veins and represent clinically relevant flow conditions. All vessel walls are taken to be rigid, and the flow is presumed to be Newtonian.

Cardiovascular and medical device simulations frequently require HPC resources due to large computational meshes and/or complex physics. In particular, the broad range of spatial scales

[3] This and the following hyperlinks indicate service providers which are exhibiting at http://exhibit.hpcexperiment.com/

arising from relatively large blood vessels that contain thin boundary layers or small medical devices frequently necessitate the use of adaptive meshing strategies. In addition, for many cardiovascular simulations, the wall shear stress is an important flow quantity that has clinical implications for device efficacy.

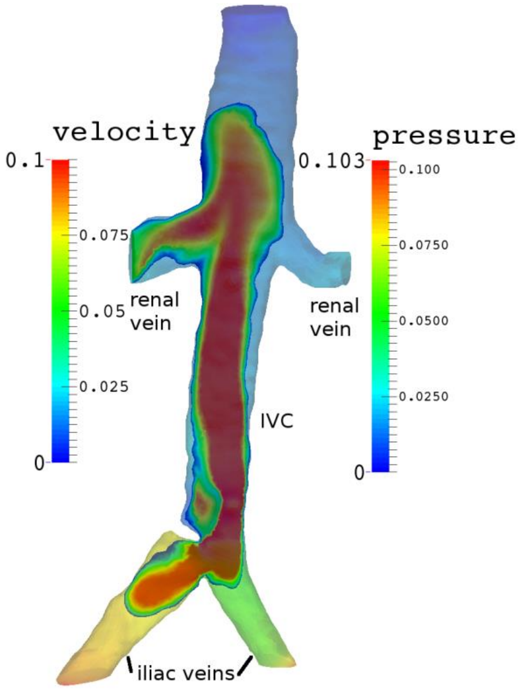


Figure 1 - Computational domain including the iliac veins (inflow), renal veins (inflow), and the inferior vena cava (IVC). The pressure on the surface of the vessels is shown; a slice, colored by velocity, down the center of the IVC is also shown.

Because the wall shear stress involves velocity gradients, proper resolution of near-wall flow is important; adequate flow resolution throughout the boundary layer and the entire computational domain usually requires several million mesh elements. Further, some cardiovascular flows are “transitional” and require relatively expensive turbulence modeling to capture the physical phenomena of interest.

Cloud resources provide a mechanism to address HPC requirements of cardiovascular simulations. Specifically, the use of cloud-based CFD alleviates the need for large, in-house clusters. In addition, cloud resources may enable the timely execution of parameter and sensitivity studies, which are important for biofluids simulations that often contain uncertain or variable model parameters. Hence, the purpose of this experiment was to explore the use of cloud-based simulation solutions for enabling cardiovascular simulations.

CHALLENGES

Our project was met with several challenges, most of which were surmountable. The following items obstructed rapid progress:

- Coordination, organization, and prioritization among team members, which includes delays due to vacation and travel schedules
- The Ciespace software platform currently accepts only CAD models as input (STL, OBJ, and triangulated surface mesh support are planned for release later in 2013) – therefore, a file type/conversion was necessary and required additional time and effort.

Despite the obstacles mentioned above, the team accomplished its goal of running a patient-specific cardiovascular flow simulation in the cloud (Figure 2). Also, the user experience and results of our experiment demonstrate the potential success of further cloud-based cardiovascular flow simulations.

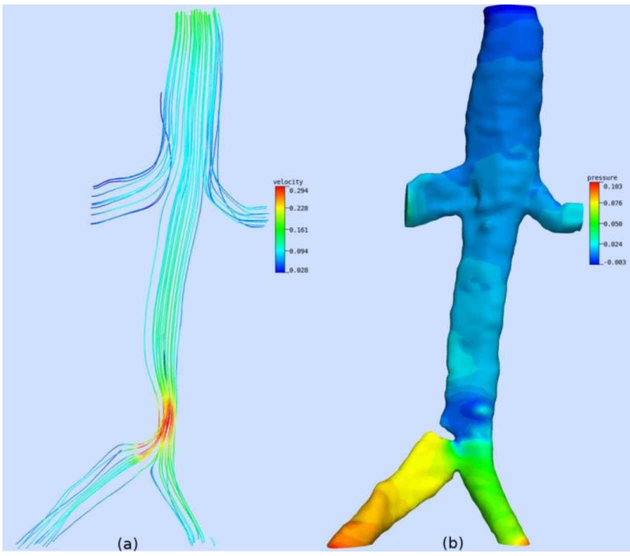


Figure 2 - Sample results: (a) streamlines, colored by velocity and (b) pressure on the surface of the vessels.

BENEFITS

The Ciespace software platform provides straightforward access to cloud computing resources. All Ciespace resources were accessed through a web browser (i.e., Firefox, Chrome, Safari, and Internet Explorer 11.0), and no special downloads or plugins were required. As a result, the Ciespace platform is easily accessible and no special hardware is required. These attributes are especially noteworthy for medical device companies that may use CFD simulations as part of their design process, but may not house staff who specialize in complex HPC hardware/software issues. In addition, the Ciespace platform supports easy case and file sharing, which facilitates collaboration among engineers, managers, and other technical staff who desired access to results.

Network speed was not prohibitive in performing any calculations or visualizing results. The client-side network speed was approximately 13.5Mbps download and 5.5Mbps upload, as measured by www.speedtest.net. Though the network speeds above are not particularly fast by US standards, the client-server approach that Ciespace takes while running within a web browser enables a responsive user experience where network lags have minimal impact on productivity.

Costly CFD software licenses were not required to perform this experiment. Instead, the Ciespace platform uses a subscription-based pricing model, which alleviates the up-front cost of commercial software licenses and is reminiscent of a pay-as-you-go SaaS (software as a service). For this experiment, Ciespace provided the end user with a trial software license. The actual cost of Ciespace resources required to perform this experiment is unknown.

Access to cloud-based computing resources, such as those provided through the Ciespace platform, enable numerous parameter and optimization studies that are unprecedented for the medical device community. Although these studies were not within the scope of this initial experiment, the end user gained valuable insight into the power of cloud-based computing resources.

CONCLUSIONS AND RECOMMENDATIONS

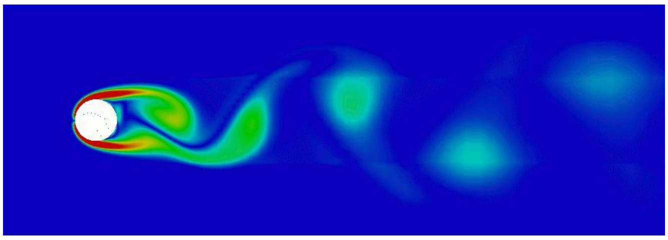
Based on the results of our HPC experiment, we conclude and recommend the following items to help ensure a successful and productive cloud-based computing experience:

- The user interface to cloud-based resources should enable a positive user experience
- Ideally, learning cloud-based software should require minimal overhead so end users can focus on using their specialties rather than learning new software applications
- Software and hardware costs (up-front and recurring) must be considered when determining the cost-benefits of cloud-based solutions
- Rendering performance and network speed must be sufficiently fast as to enable technical progress: large latency is unlikely to be well-tolerated amongst end users
- Cloud-based computing resources enable a wide variety of simulation studies that otherwise would not be feasible

---

Case Study Authors – Mike Singer and Sanjay Choudhry

Code_Saturne
Overhead Power Lines Lifespan Assessment in Preventive Maintenance



"The overall project -the set-up, the simulations, team support, cloud enabling technologies used, quality of final results, and project duration - and the overall user experience exceeded RTE's expectations."

1 MEET THE TEAM

End-User: Fikri Hafid, Project leader OLLA (Overhead Lines Lifespan Assessment), Réseau de Transport d'Électricité (RTE), and Lei Zhang, Postdoc Research Fellow at Kannon MSD, a spin-off of the Center for Mathematics and its Applications ENS Paris-Saclay, France.
Software Provider: Code_Saturne open source CFD software developed at Électricité de France (EDF), and UberCloud HPC containers managed by Atrio's Composable Cloud (ACC) management platform.
Resource Provider: Jón Þór Kristinsson, technical director and Guy D'Hauwers, Regional Director Europe at Advania Data Centers HPCFLOW Cloud in Iceland, with its bare-metal capabilities powered by HPE/Intel HPC technologies.
HPC Cloud Experts – Serkan Koyuncu and Ronald Zilkovski at the UberCloud Inc., providing novel HPC application container technology for ease of cloud access and use for Code_Saturne.
Project Managers – Mohamed Bekhti, HPE, project manager, Norbert Bianchin HPE, IT resource manager and EAP program manager, and Reha Senturk, UberCloud Inc., IT lead.
Sponsors – Hewlett Packard Enterprise, Intel, represented by Bill Mannel and Jean-Luc Assor, HPE.

2 HYBRID HPC EARLY ACCESS PROGRAM (EAP)

Hybrid HPC enables new levels of agility. It combines HPE GreenLake Flex Capacity, Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) capabilities that can be delivered both on-premise and off-premise.

All combinations can be difficult to evaluate without trying. For this reason, HPE has launched the Hybrid HPC Early Access Program in 2018 to support HPC cloud environments for customers and service providers.

Objective is to provide to customers a starter kit or a cloud environment in one of HPE's Centers of Excellence (CoE) and a set of services (consulting, demos and trials) where they can experiment new technologies and their solution before acquiring. Try and test first, buy if you are satisfied with the results.

For this project, the EAP program was used to deliver a compute environment in the cloud with Advania Data Centers CoE in two phases (4 nodes and 32 nodes). The configuration and implementation of the compute environment and consulting for containerization of the application software has been provided by UberCloud.

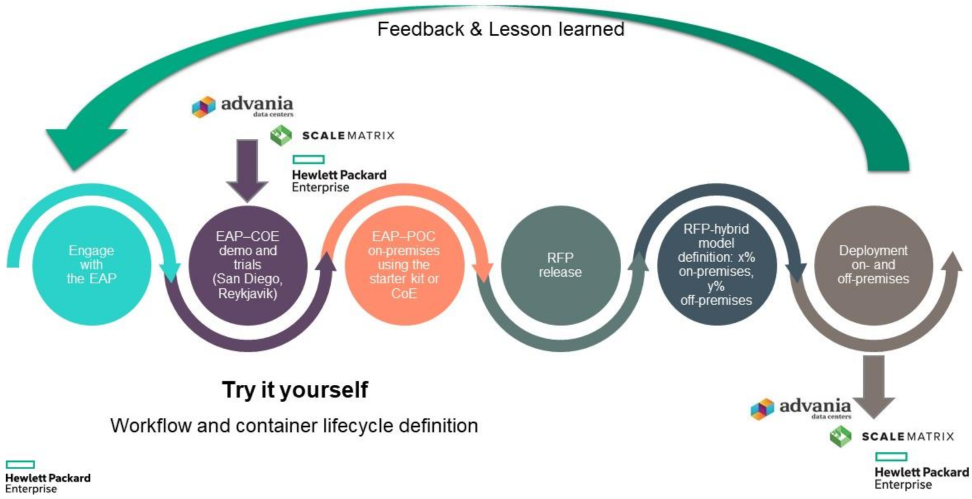


Figure 5: Hybrid HPC EAP life cycle.

3 USE CASE

Réseau de Transport d'Électricité (RTE) is a subsidiary of the French electricity utility company Électricité de France (EDF), and RTE is the operator of the high voltage electricity transmission network in France. With more than 100,000 kilometers of high and very high voltage overhead lines, this network is the largest in Europe. Given the associated investment amounts, the management of these physical assets (overhead conductors, pylons, etc.) is a top priority. The challenges are particularly high in the field of overhead conductors, as the costs of replacing them will reach several hundred million Euros annually in the coming years. Preventive Maintenance plays a major role in providing uninterrupted services and reducing outages and costs due to equipment fatigue and failures.

The goal of this OLLA (Overhead Lines Lifespan Assessment) project is to help maintenance teams prioritize the replacement of overhead lines. To do that we have studied the phenomena that can have an impact on the aging of overhead conductors (cables of an overhead line). One of these phenomena is fretting fatigue (a type of mechanical fatigue). This phenomenon is induced by Aeolian vibrations. The vibrations are due to the interaction between the conductor and the wind, that’s why we ran 1000-cores simulations. The first purpose of the 1000-core simulation is to enhance our understanding of the interaction between the overhead line conductor and the wind (vibrations frequency and amplitude in particular). In particular, we need to have a good evaluation of the vibration frequency and associated amplitude. These two values are needed to assess the damage caused by those vibrations. The second purpose is to establish a reference case from which simplified models can be derived or against which new simplified models can be compared. Indeed, considering the wide variety of conductors and the wide wind loadings on them, it is not conceivable to run simulations on all those cases. We hope that the results at hand will allow us to derive “simplified” models that will be used in our software for conductor lifespan inter-ranking.

For that, RTE gathers massive amount of data from sensors in field operations (power transportation, green energy production, weather information), generating 500,000 samples every 10 seconds. Processing all of this information and generating lifespan assessment requires high performance computing (HPC) resources.

RTE and Kannon MSD, a spin-off of the Center for Mathematics and its Applications ENS Paris-Saclay, France, have been working together to study the phenomenon of vortex-induced vibrations of overhead power lines used in electric power transmission and distribution to transmit electrical energy across large distances. It consists of one or more conductors (commonly multiples of three) suspended by towers or poles. It is essential to predict the line vibrations for a better estimation of the lifespan of overhead lines. In this study, we analyzed lift and drag forces created by low wind speeds, and line vibrations due to cyclic stresses which cause fatigue failure of the wires.

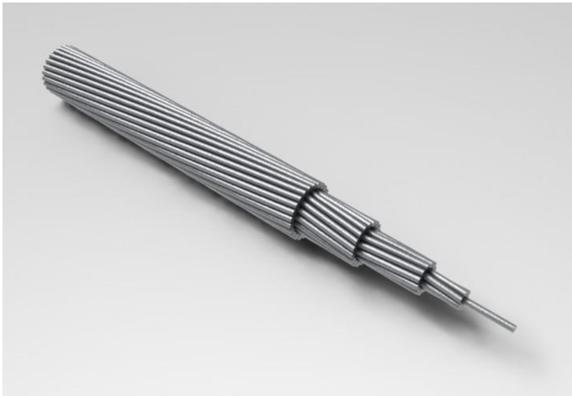



Figure 2: Conductors and pylons from the French grid (right), Overhead line conductor (All Aluminum Alloy Conductor AAAC, left).

Researchers developed a fluid-structure interaction process using Code_Saturne for Computational Fluid Dynamics (CFD) simulating the airflow around the overhead lines, and the overall line dynamics captured by an in-house solid solver (FEM). The code coupling was used to determine interaction between the wind and the power line. The solution used a Quasi-3D method to simplify the coupling problem, with CFD performed separately for a series of sections of the overhead line (fluid solver), the overall line dynamics constructed from the FEM solver, and message-passing MPI communication between fluid and solid solvers.

The simulation process evaluates each section of the cable and also the interaction between them to simulate the impact. Each cable section is assigned to an individual CPU core to solve the CFD problem, and the FEM model manages the interaction. This requires a large number of computational sections, thus requiring a large number of cores to properly evaluate the impact on the whole cable. The HPC cluster available in RTE's research center provided a 240-core cluster for the initial phase of the project. However, RTE needed much more resources to have a comprehensive result which finally required 1000 cores and a high number of iterations (i.e. timesteps).

UberCloud and Advania Data Centers HPCFLOW provided the cloud HPC resources and a flexible environment for ease of access and use. Advania provided scalable resources built on HPE's Apollo hardware with Intel CPUs and Interconnect. UberCloud provided the cloud infrastructure, a CentOS Docker container with Code_Saturne CFD, Salome-Meca, and Paraview installation, and additional technical support during the project.

4 CAE SOFTWARE

Docker based UberCloud HPC application containers managed by Atrio's Composable Cloud (ACC) management platform were used to create a flexible and scalable application environment for accessing and running CAE applications in HPC clusters. The containerized simulation environment was deployed on an HPE based 32-node compute cluster at Advania Data Centers. The following CAE software has been containerized by UberCloud.

Code_Saturne is free, open-source, general-purpose computational fluid dynamics software solving the Navier-Stokes equations for 2D, 2D-axisymmetric and 3D flows, steady or unsteady, laminar or turbulent, incompressible or weakly dilatable, isothermal or not, with scalars transport if required. Developed since 1997 at Électricité de France, Code_Saturne is distributed under the GNU GPL license.

SALOME is an open-source software that provides a generic platform for Pre- and Post-Processing for numerical simulation. SALOME platform itself does not include any calculation codes.

Salome-Meca is a standalone application, it is an external project based on the SALOME platform. Salome-Meca represents the integration of the Code_Aster solver in the SALOME platform (see http://www.code-aster.org). Code_Aster is an open source software package for Civil and Structural Engineering finite element analysis and numeric simulation in structural mechanics originally developed as an in-house application by EDF.

ParaView is an open-source multiple-platform application for interactive, scientific visualization. It has a client–server architecture to facilitate remote visualization of datasets, and generates level of detail models to maintain interactive frame rates for large datasets.

Gnuplot is a command-line program that can generate two- and three-dimensional plots of functions, data, and data fits. The program runs on all major computers and operating systems. It is a program with a fairly long history, dating back to 1986. Despite its name, this software is not part of the GNU project.

5 PERFORMING CFD / FEM SIMULATIONS IN THE CLOUD

CFD simulations have been set up such that each cable (treated as a cylinder) of the multicore cable is distributed to one processor core. Then, 10,000 timesteps have been simulated. After each CFD step over all the cylinders, one solid FEM step is performed. The test configuration consisted of 60 cores, and then has been increased to 240, 480, 720, and finally 1000 cable cores (on 1000 CPU cores). In detail, the CPU time for one timestep for 60 cores was 1.1s; for 240 cores 4.5s; for 480 cores 10s; for 720 cores 14s; and for 1000-cores 17.6s. Thus, we can observe a linear correlation between the number of cores and the execution time for one timestep. After the 10,000 timesteps, which correspond to the build-up phase of the electric cable vibration, the cable vibrations were still very weak and the flow field was almost the same as the flow past a fixed cable, with a week vortex shedding phenomenon of flow around each cylinder.

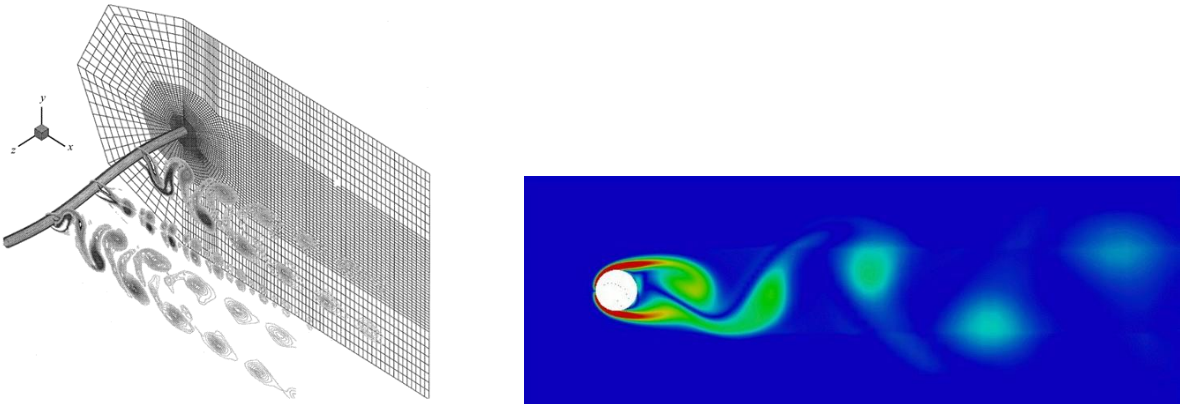


Figure 3: Vortex-shedding around a 2D section of overhead line.

Therefore, physically those results were not yet that interesting, and we decided to further increase the number of timesteps for the 1000-core case, using 32 compute nodes (1000 compute cores) on the Advania Data Centers cloud. On this 32-node cluster, the 10,000 iterations took 78.4h to complete, and for reaching near-steady vibrations after about 55,000 timesteps the simulation took 431 hours or 18 days. The 55,000 computational timesteps correspond to a physical time of 55 seconds. At that time, the vibrations become stable, i.e. the vibration magnitude becomes constant. The present 1000-core run simulates a 325 m overhead line represented by the 1000 cylinders, subjected to tension H = 20 kN, with linear density mL = 1.575 kg/m. The wind velocity is u = 5.47 m/s with a Reynolds number Re = 10,000; vortex shedding frequency is fv = 39.6 Hz; the overhead line fundamental frequency is f1 = 0.173 Hz; and therefore, the activated model number is N = fv/f1 = 228.

The numerical results in Figure 3 show that the alternating lift forces around the cable result in so-called aeolian vibrations of the cable. This physical phenomenon can be observed correctly from numerical results. This CFD simulation result corresponds to a simulation of a cable at full scale. So far, no numerical simulation exists at such a scale, thus this result can be used as a reference to show the physical difference between a full scale cable and a reduced scale cable, and also to show that traditional industrial approach, like energy balance principle (EBP) has limits and is not able to reproduce well the cable dynamics. Thus, from these numerical simulation results, we can already observe the vibration characteristics of a long multicore power cable. Among others, the information obtained from present work can be used to determine the fatigue of the cable.

By the way, these wind-induced vibrations of long power cables are caused by so-called von Karman vortex shedding which is also responsible for the aeolian vibration of the cable observed in this project. Influencing parameters are e.g. mechanical tension, linear mass, span of the cable, and the Reynolds number of fluid flow. A famous von Karman phenomenon was the collapse of the Tacoma bridge in Washington State in 1940, watch the bridge video here https://aerospaceengineeringblog.com/the-von-karman-vortex-street-and-tacoma-narrows-disaster/.

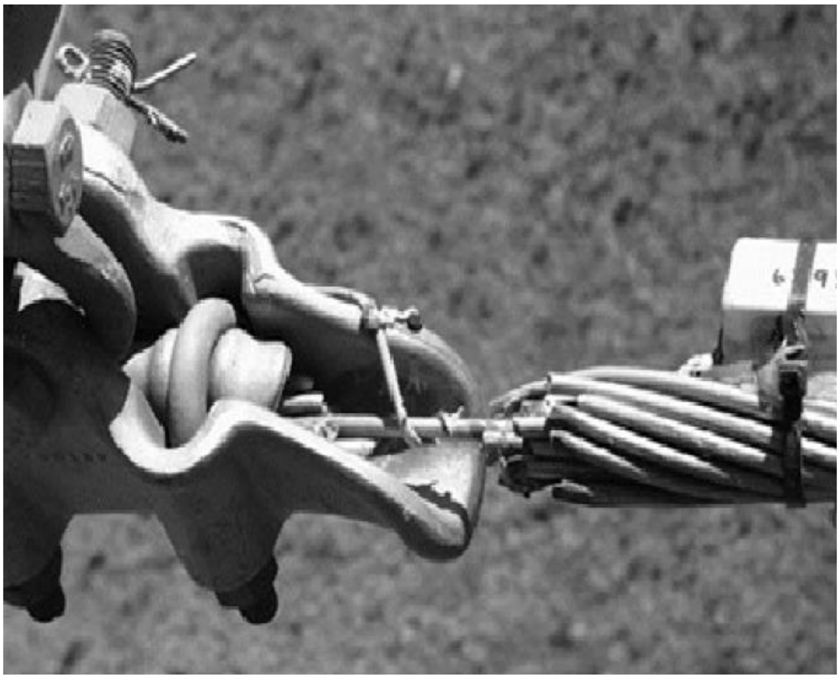


Figure 4: Failure due to mechanical fatigue at clamp/conductor system.

Cloud Environment

The cloud hardware specifications of the Advania Data Centers compute cluster hosting the UberCloud containers are as follows:

- 32 compute nodes each with 2 Intel E5-2683v4 CPU with 32 cores @ 2.1 GHz (Broadwell)
- GPU: none
- Memory: 256 GB RAM per node
- Intel Omni-Path cluster interconnect network

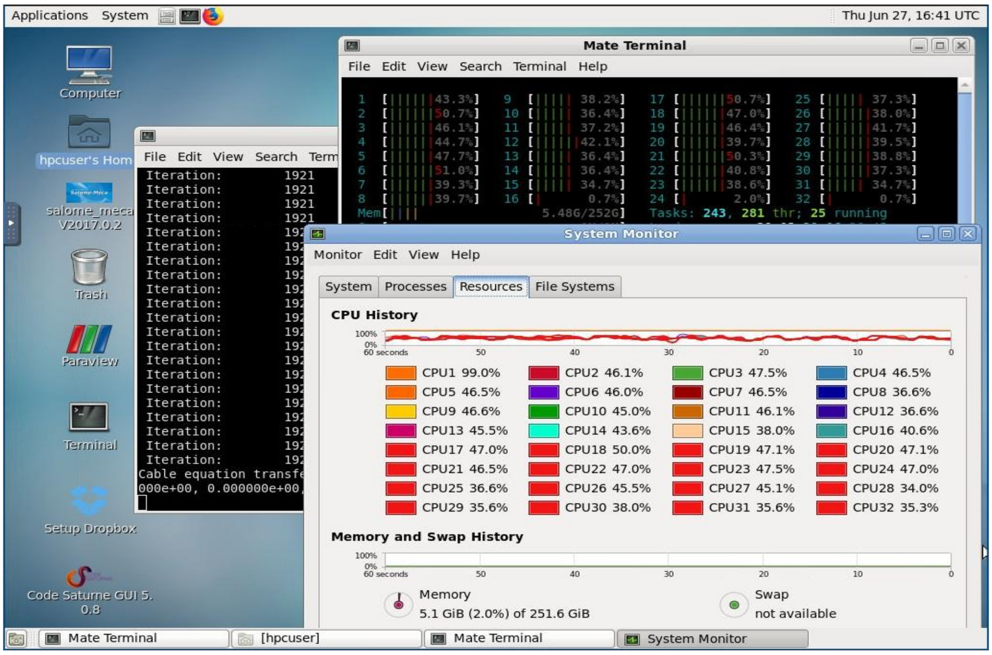


Figure 5: Cloud Monitoring - On 32 compute nodes, execution history in the user's Terminal window.

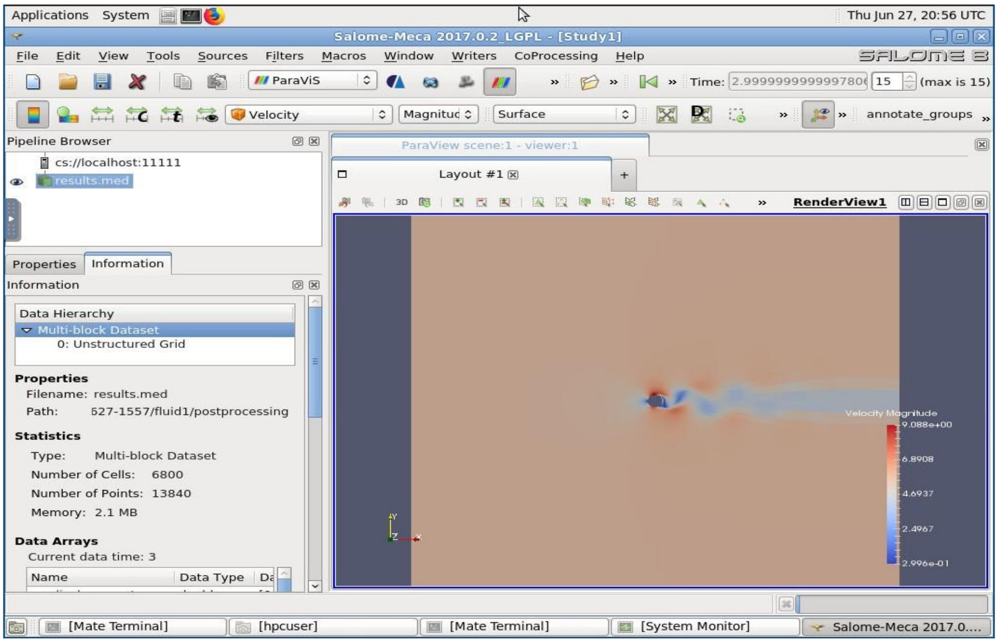


Figure 6: Remote Visualization – Airflow around the powerline in post-processing tool ParaView.

Cloud Tests

The code was recompiled in the Advania cloud environment to run a test with standard 61 cores and 3000 iterations. The test completed in 57 minutes.

1000 Core Run

Fluids initialization: As a first step, the code was recompiled based on 1000 cores and 3,000 iterations and ran for 23.5 hours. After this initialization, the code was recompiled based on 1000 cores and ran for 55,000 timesteps. This simulation completed in 18 days.

6 SUMMARY AND CONCLUSIONS

This project (and case study) demonstrates the importance of high-performance computing (HPC) and big data analytics for applications related to Predictive and Preventive Maintenance, in this case for overhead power lines lifespan assessment. Large amounts of data are generated from sensors and field operations, and 500,000 samples are processed every 10 seconds. French company Réseau de Transport d'Électricité (RTE) had already started working on an in-house 4-node 80-core HPC cluster, but it turned out to be impossible to solve the task at hand in reasonable time, and therefore, they turned to the Advania Data Centers cloud and used a 32-node HPC cluster to run an 18-day simulation on its 1,000 cores, reducing the project time from an estimated 6 months to 18 days.

The objective of this project was to analyze vortex-induced vibrations of high-voltage overhead powerlines. Predicting the line vibrations is essential for a better estimation of the lifespan of these overhead lines. The low wind speeds (approximately 1 to 10m/s) induce vortex shedding downstream of the overhead line, creating alternating lift and drag force which then cause line vibrations, generating cyclic stresses that can lead to fatigue failure of the individual wire strands.

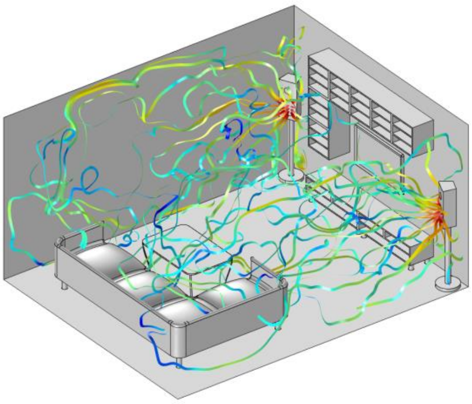
RTE’s application software Code_Saturne and Salomé is hosted in UberCloud’s Docker-based HPC containers and managed by Atrio ACC cloud management platform, creating the end-user cloud experience while using Advania’s HPC bare metal capabilities powered by HPE/Intel HPC technologies.

The approach used in this project was based on CFD simulations of the air flow around the overhead line. The CFD and FEM analysis codes have been coupled to determine the interaction between the wind and the overhead line. A quasi-3D method simplified the coupling problem, with CFD performed separately for a series of sections (cylinders) of the overhead line, overall line dynamics simulated by a custom solid solver (FEM), and fluid solver and solid solver communicated via Intel MPI.

The final 1000-cylinder simulations on the 1,000-core cluster still took 18 days, an estimated speed-up of about 10 compared with RTE’s in-house cluster. **The overall project, from the set-up phase to the final simulations, the team support, the cloud enabling technologies, the quality of the final results, the project duration, and the overall user experience exceeded RTE’s expectations.**

Case Study Authors – Fikri Hafid, Lei Zhang, Mohamed Bekhti, Reha Senturk, and Wolfgang Gentzsch

COMSOL Multiphysics
Acoustic Modelling and Parametric Radio Frequency Heating



> *"Utilizing cloud services make it easy to extend your on premise hardware for demanding applications. The license models and technology of today provides HPC with just a few mouse clicks."*

MEET THE TEAM

End user and Team Expert – Pär Persson Mattsson, Technical Support Engineer and HPC Consultant at COMSOL Multiphysics GmbH
Software Provider – Winfried Geis, Branch Manager at COMSOL Multiphysics GmbH
Resource Provider – Thomas Gropp, Engineer IT-Systems, and Christian Unger, Project Manager Resource Area, both at CPU 24/7 GmbH. CPU 24/7 is a leading provider of CAE as a Service solutions for all application areas of industrial and academically/university research and development. Headquartered in Potsdam/Germany, CPU 24/7 develops and operates unique on demand services for High Performance Computing that are based on the latest globally accepted industry standards for hardware, software, and applications.

USE CASE

In this test project, we wanted to find out how the cloud can help in speeding up and enabling high performance FEM simulations with COMSOL Multiphysics®. The aim of the project was to find out how HPC cloud providers can augment the internal on premise hardware to allow for more detailed and faster simulations.

COMSOL Multiphysics is a general-purpose software platform, based on advanced numerical methods, for modeling and simulating physics based problems. COMSOL Multiphysics, with its multiphysics capabilities, is especially suited for coupled and multiphysics simulations, making the description of real-world multiphysics phenomena and systems possible.

For this project two COMSOL applications were used – one acoustic model and one parametric radio frequency (RF) heating model from the COMSOL Multiphysics application library. The acoustic model simulates the acoustic pressure at certain frequencies in a typical living room equipped with stereo speakers and other typical living room furniture. The RF model is a model from the COMSOL Multiphysics application library. It shows dielectric heating of an insulated block, caused by microwaves travelling in an H-bend waveguide.

These two models were chosen since they represent two different forms of parallelization of multiphysics simulations. In the acoustic model where higher frequencies require a finer mesh, each frequency step is parallelized and the matrix is distributed among the different cores and nodes, computing one frequency after another.

On the other side of the spectrum there is the RF model, which is a small model that could easily be computed on a modern laptop, but where a large amount of parameters need to be computed. It can be parallelized so that several frequencies and geometric parameters are computed at the same time. This model yields what is called an embarrassingly parallel computation.

CHALLENGES

Both of the models used in this project bring with them their own challenges:

- In the case of the RF model, the model is small and possible to compute on almost any modern computer, but the amount of frequencies that needs to be computed, together with the different geometric parameters, causes a high number of simulations that need to be performed. Even if the computation of one parameter only takes 30 minutes, the total computation times can become unacceptably large when the number of parametric values increases.
- For a frequency model with a large geometry where higher frequencies are of interest, a large amount of memory is needed to be able to compute the model at all. Many users in small or medium companies do not have the hardware needed on premise, but might still need to handle these types of models sporadically.

PROCESS AND BENCHMARK RESULTS

The computations were performed on a 10 node class “medium” cluster, where each node was equipped with dual socket Intel® Xeon® X5670 and 24 GB of RAM, giving a total count of 120 cores and 240 GB of RAM. The nodes were connected with Mellanox Infiniband QDR. This hardware setup was chosen since it is suitable for a first test of the capabilities of HPC computing in the cloud. The hardware (bare-metal, non-shared infrastructure) was supplied by CPU 24/7, and COMSOL Multiphysics was pre-installed by CPU 24/7’s technicians.

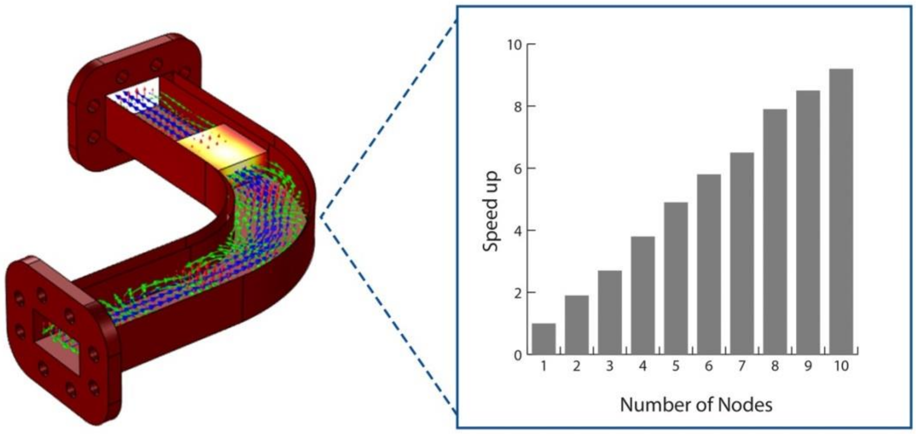


Figure 1: Simulated H-Bend Waveguide. Left image shows the magnetic field (Arrows) of the waveguide and the temperature in the dielectric block (yellow). The image to the right shows the almost perfect speed up gained when adding more compute nodes.

Since the COMSOL Floating Network License (FNL) and the COMSOL Server License (CSL) allow for remote and cloud computing out of the box, the only thing we needed to do was to use the built in COMSOL Multiphysics functionality for starting jobs on remote clouds and clusters, and forwarding access to the on premise license manager. Once the functionality is configured, any model can be sent to the cloud cluster in a matter of seconds.

The benchmarks were performed for one through ten nodes (each equipped with dual Intel® Xeon® X5670 processors and 24 GB of RAM) for the RF model, and for the acoustics model a local workstation (equipped with a single Intel® Xeon® E5-2643 processor and 32 GB) was used as a reference system, while one run where all ten nodes were performed on the cluster. After the simulations finished the computed results were collected and evaluated.

For the RF model the interesting property was the decreased solution time for the parametric computation of the model, which is demonstrated in Figure 1. Since the RF model is a model with a large parametric sweep, i.e. an embarrassingly parallel computation, the almost perfect speed up is expected.

For the acoustics model, we were less interested to see the change in computation speed as we added more nodes, but rather what frequency we could compute in comparison with a standard high end workstation equipped with 32 GB of RAM. As can be seen from Figure 2, a finer mesh could be used in the cluster simulation, compared to the simulation made on the workstation. In this particular model, it meant going from being able to compute frequencies up to 500 Hz the local workstation, to frequencies up to 1200 Hz on the cluster.

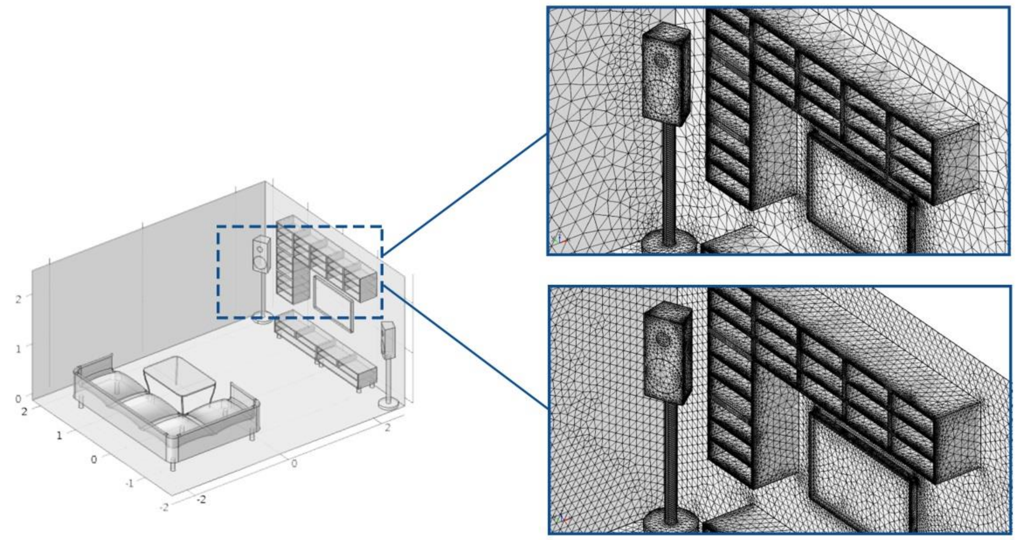


Figure 2: The geometry simulated in the acoustics model. The images to the right show the resolution of the mesh that was possible on the workstation (top) and on the ten node cluster (bottom). The finer resolution made it possible to simulate up to 1200 Hz on the cluster, in comparison to 500 Hz on the workstation.

BENEFITS

For the user, the use of COMSOL together with the HPC cloud services from CPU 24/7 brings with it many advantages. A few of them are:

- Increased number of cores and memory channels yield higher throughput for time critical projects.
- The increased amount of memory and the faster memory access obtained from an increased number of processors allows us to compute more realistic and detailed models, which might be impossible to compute on local hardware.
- One can have fast access to a bare-metal cluster in the cloud, thanks to the fast and competent support from CPU 24/7. Since COMSOL Multiphysics was preinstalled by the CPU 24/7 technicians,

and access was enabled both via remote desktop solutions and SSH, there was no reduction in usability compared to an on premise cluster.

- The easy access to the cloud through the COMSOL Multiphysics GUI, and the possibility to forward your license makes it easy to extend your on premise hardware when needed.

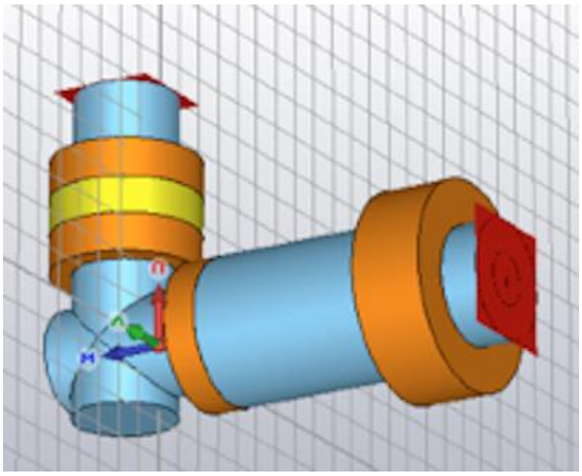
CONCLUSIONS

- We showed that the CPU 24/7 HPC bare-metal cloud solution is a beneficial solution for COMSOL users who want to obtain higher throughput and more realistic results in their simulations.
- For parametric studies, the speed up on the cloud based clusters delivered by CPU 24/7 has been shown to be almost perfect, meaning that a ten node cluster can increase productivity almost tenfold.
- To use cluster computing, there is no investment in in-house HPC expertise needed, since CPU 24/7 offers permanently available and tailored HPC bare-bone cloud cluster solutions.
- The time invested from when CPU 24/7 had provided access to their cloud based cluster, until the first job was sent was approximately two hours, including the initial configuration of the remote computing settings in the COMSOL GUI.

Case Study Author – Pär Persson Mattsson, Comsol Multiphysics GmbH

CST Microwave Studio

Simulation of a Multi-resonant Antenna System



MEET THE TEAM

End User – Dr. Nicolas Freytag, PhD physicist and engineer working for Bruker Biospin Corporation as innovation manager responsible for new sensor applications, new markets and fundamental research. Bruker is one of the world's leading analytical instrumentation companies with 6,000 employees at more than 70 locations around the globe. The local site has 500 employees.
Software Provider – Dr. Felix Wolfheimer is a Senior Application Engineer with CST AG
Resource Provider – Amazon Web Services
HPC Expert – Chris Dagdigian is a Principal Consultant employed by the BioTeam.

USE CASE

The end user uses CAE for virtual prototyping and design optimization on sensors and antenna systems used in NMR spectrometers. Advances in hardware and software have enabled the end-user to simulate the complete RF-portion of the involved antenna system. Simulation of the full system is still computationally intensive although there are parallelization and scale-out techniques that can be applied depending on the particular “solver” method being used in the simulation.

The end-user has a highly-tuned and over-clocked local HPC cluster. Benchmarks suggest that for certain “solvers” the local HPC cluster nodes are roughly 2x faster than the largest of the cloud-based Amazon Web Services resources used for this experiment. However, the local HPC cluster averages 70% utilization at all times and the larger research-oriented simulations the end-user was interested in could not be run during normal business hours without impacting production engineering efforts.

Remote cloud-based HPC resources offered the end-user the ability to “burst” out of the local HPC system and onto the cloud. This was facilitated both by the architecture of the commercial CAE software as well as the parallelizable nature of many of the “solver” methods.

The CST software offers multiple methods to accelerate simulation runs. On the node level (single machine) multithreading and GPGPU computing (for a subset of all available solvers) can be used to accelerate simulations still small enough to be handled by a single machine. If a simulation project needs multiple independent simulation runs (e.g. in a parameter sweep or for the calculation of different frequency points) that are independent of each other, these simulations can be sent to different machines to execute in parallel. This is done by the CST Distributed Computing System, which takes care of all data transfer operations necessary to perform this parallel execution. In addition, very large models can be handled by MPI parallelization using a domain decomposition approach.

End-user effort: >25h for setup, problems and benchmarking. >100h for software related issues due to large simulation projects, bugs, and post-processing issues that would also have occurred for purely local work.

ISV effort: ~2-3 working days for creating license files, assembling documentation, following discussions, debugging problems with models in the setup, debugging problems with hardware resources.

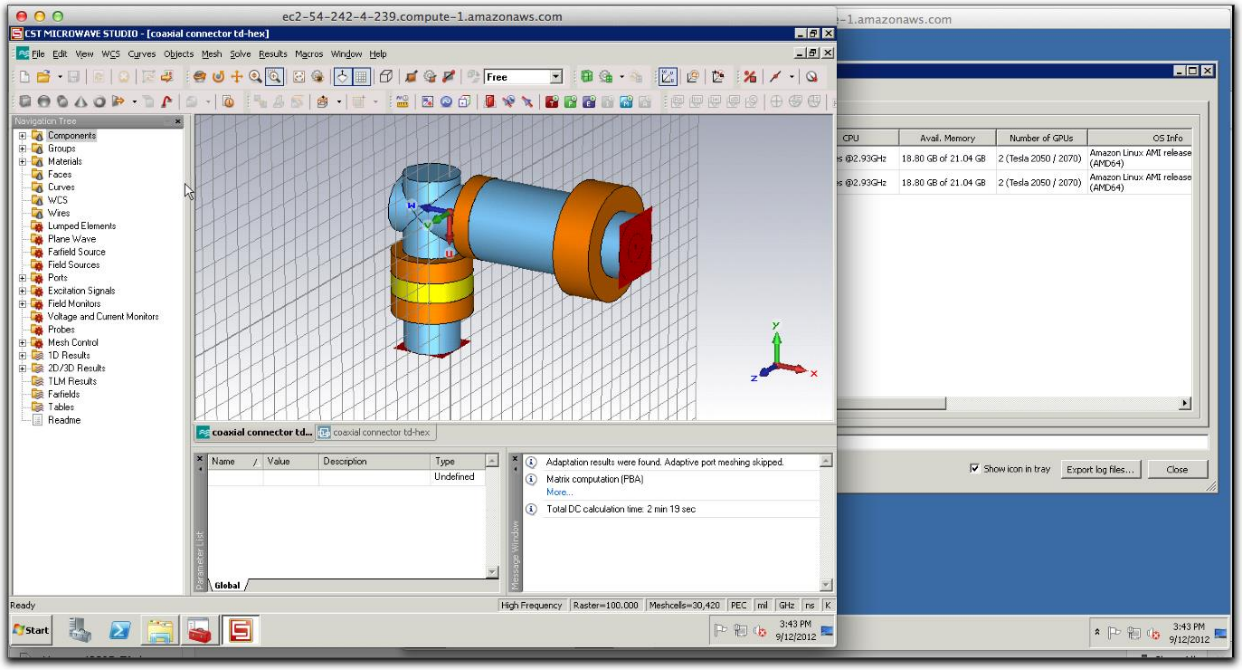
PROCESS

1. Define the ideal end-user experiment
2. Initial contacts with software provider (CST) and resource provider (AWS)
3. Solicit feedback from software provider on recommended “cloud bursting” methods; secure licenses
4. Propose Hybrid Windows/Linux Cloud Architecture #1 (EU based)
5. Abandon Cloud Architecture #1; User prefers to keep simulation input data within EU-protected regions. However, AWS has resources we require that did not yet exist in EU AWS regions. End-user modifies experiment to use synthetic simulation data, which enables the use of US, based cloud systems.
6. Propose Hybrid Windows/Linux Cloud Architecture #2 (US based) & implement at small scale for testing
7. Abandon Cloud Architecture #2. Heavily secured virtual private cloud (VPC) resource segregation front-ended by an internet-accessible VPN gateway looked good on paper however AWS did not have GPU nodes (or the large cc2.* instance types) within VPC at the time and the commercial CAE software had functionality issues when forced to deal with NAT translation via a VPN gateway server.
8. Propose Hybrid Windows/Linux Cloud Architecture #3 & implement at small scale for testing.
9. The third design pattern works well; user begins to scale up simulation size
10. Amazon announces support for GPU nodes in EU region and GPU nodes within VPC environments; end-user is also becoming more familiar with AWS and begins experimenting with Amazon Spot Market to reduce hourly operating costs by very significant amount.
11. Hybrid Windows/Linux Cloud Architecture #3 is slightly modified. The License Server remains in the U.S. because moving the server would have required a new license file from the software provider. However, all solver and simulation systems are relocated to Amazon EU region in Ireland for performance reasons. End-user switches all simulation work to inexpensively sourced nodes from the Amazon Spot Market.
12. The “Modified Design #3” in which solver/simulation systems are running on AWS Spot Market Instances in Ireland, while a small license server remaining in the U.S. reflects the final “design.” As far as we understood, the VPN-Solution that did not work in the beginning of the project would actually have worked at the end of the project period because of changes within the AWS. In addition the preferred “heavily secured” solution would have provided fixed MAC addresses, thus avoiding having to run a license instance all the time.

CHALLENGES

Geographic constraints on data – End-user had real simulation and design data that should not leave the EU.

Unequal availability of AWS resources between Regions – At the start of the experiment, some of the preferred EC2 instance types (including GPU nodes) were not yet available in the EU region (Ireland). This disparity was fixed by Amazon during the course of the experiment. At the end of the experiment we had migrated the majority of our simulation systems back to Ireland.
Performance of Remote Desktop Protocol – The CAE software used in this experiment makes use of Microsoft Windows for experiment design, submission and visualization. Using RDP to access remote Windows systems was very difficult for the end-user, especially when the Windows systems were operating in the U.S.



Front-end and two GPU solvers in action

CAE Software and Network Address Translation (NAT) – The simulation software assumes direct connections between participating client, solver and front-end systems. The cloud architecture was redesigned so that essential systems were no longer isolated within secured VPC network zones.
Bandwidth between Linux solvers & Windows Front-End – The technical requirements of the CAE software allow for the Windows components to be run on relatively small AWS instance types. However, when large simulations are underway a tremendous volume of data flows between the Windows system and the Linux solver nodes. This was a significant performance bottleneck throughout the experiment. The project team ended up running Windows on much larger AWS instance types to gain access to 10GbE network connectivity options.
Node-locked software licenses – The CAE software license breaks if the license server node changes its network hardware (MAC address). The project team ended up leveraging multiple AWS services (VPC, ENI, ElasticIP) in order to operate a persistent, reliable license serving framework. We had to leave the license server in the US and let it run 24/7 because it would have lost the MAC-address upon reboot. Only in the first setup did it have a fixed MAC and IP.
Spanning Amazon Regions – It is easy in theory to talk about cloud architectures that span multiple geographic regions. It is much harder to implement this "for real." Our HPC resources switched between US and EU-based Amazon facilities several times during the lifespan of the project. Our project required the creation, management and maintenance of multiple EU and US specific SSH keys, server images (AMIs) and EBS disk volumes. Managing and maintaining capability to operate in the EU or US (or both) required significant effort and investment.

BENEFITS

End-User

- Confirmation that a full system simulation is indeed possible even though there are heavy constraints, mostly due to the CAE software. Model setup, meshing and post-processing are not optimal and require huge efforts in terms of manpower and CPU-time.
- Confirmation that a full system simulation can reproduce certain problems occurring in real devices and can help to solve those issues.
- Realize the reasonable financial investment for additional computation resources needed for cloud bursting approaches.
- Realize that the internet connection speed was the major bottleneck for a cloud bursting approach but also very limiting for RDP work.

Software Provider

- Confirmation that the software is able to be setup and run within a cloud environment and also, in principle, using a cloud bursting approach (see comments regarding the network speed). Some very valuable knowledge was gained on how to setup an "elastic cluster" in the cloud using best practices regarding security, stability and price in the Amazon EC 2 environment.
- Experience the limitations and pitfalls specific to the Amazon EC2 configuration (e.g. availability of resources in different areas, VPC needed to preserve MAC addresses for licensing setup, network speed, etc.).
- Experiencing the restrictions of the IT department of a company when it comes to the integration of cloud resources (specific to the cloud bursting approach).

HPC Expert

- Chance to use Windows-based HPC systems on the cloud in a significant way was very helpful
- New appreciation for the difficulties in spanning US/EU regions within Amazon Web Services

CONCLUSIONS AND RECOMMENDATIONS

End-User

- Internet transfer speed is the major bottleneck for serious integration of cloud computing resources to the end users design flow and local HPC systems.
  - Internet transfer speed is also a limiting factor to allow for remote visualization.
- Security and data protection issues as well as fears of the end users IT department create a huge administrative limitation for the integration of cloud based resources.
- Confirmation that a 10 GbE network can considerably speed up certain simulation tasks compared to the local clusters GbE network. The local cluster has been upgraded in the meantime to an IB network.

HPC Expert

- Rapid evolvement of our provider’s capability constantly forced the project team to re-architect the HPC system design. The cloud is normally advertised as “enabling agility” and “enabling elasticity” but in several cases it was our own project team that was required to be agile/nimble simply to react to the rapid rate of change within the AWS environment.
- The AWS Spot Market has huge potential for HPC on the cloud. The price difference is extremely compelling and the relative stability of spot prices over time makes HPC usage worth pursuing.
- Our design pattern for the commercial license server is potentially a useful best-practice. By leveraging custom/persistent MAC addresses via the use of Elastic Network Interfaces (ENI) within Amazon VPC we were able to build a license server that would not “break” should the underlying hardware characteristics change (common on the cloud).

- In a “real world” effort we would not have made as much use of the hourly on-demand server instance types. Outside of this experiment it is clear that a mixture of AWS Reserved Instances (license server, Windows front-end, etc.) and AWS Spot Market instances (solvers and compute nodes) would deliver the most power at the lowest cost.
- In a “real world” effort we would not have done all of our software installation, configuration management and patching by hand. These tasks would have been automated and orchestrated by a proper cloud-aware configuration management system such as Opscode Chef.

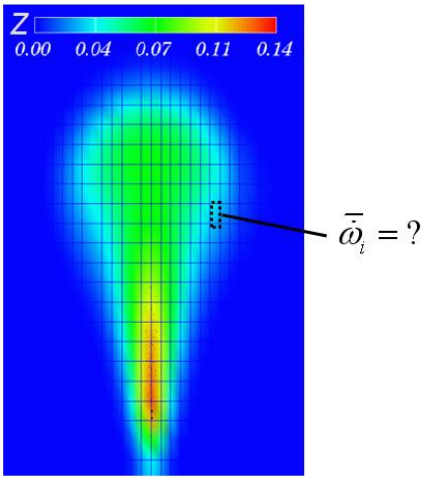
Software Provider:

The setup of a working setup in the cloud is quite complex and needs quite some IT/Amazon EC2 expertise. Supporting such a setup can be quite challenging for an ISV as well as for an end user. Tools to provide simplified access to EC2 would be helpful.

Case Study Authors – Nicolas Freytag, Felix Wolfheimer, and Chris Dagdigian

DACOLT Combustion

DACOLT Combustion Training in the Cloud



> *"UberCloud container technology harnessing Ansys Fluent CFD software on CPU24/7 HPC resources, accessed from a browser on a laptop computer, provided a light and seamless user experience."*

MEET THE TEAM

End-user: A. de Jong Group, Energy and Environmental Technologies, The Netherlands
Combustion Software Provider & Expert: Ferry Tap, Dacolt, The Netherlands
Software Provider: Wim Slagter, ANSYS Inc. and UberCloud Containers
Ansys Container Provider: Fethican Coskuner, UberCloud
Resource Provider and HPC Experts: Thomas Gropp, Alexander Heine, Christian Unger, CPU 24/7, Potsdam, Germany.

USE CASE

Dacolt provides highly appreciated Combustion CFD trainings for ANSYS Fluent since 2012. When delivering such trainings on-site, a number of challenges are faced:

- **Does the end-user have sufficient CFD licenses available?**
- **Does the end-user have sufficient HPC resources available?**
- **How are the HPC resources accessed?**

Not so long ago, some training sessions involved running on laptop computers which had to be physically moved around as well as being up-to-date from the Operating System and the CFD software perspective, involving substantial logistics and potential IT headaches.

In this UberCloud Experiment, the ANSYS software is provided in a Linux (Docker-based) Container from UberCloud, which runs on CPU 24/7 HPC Cloud resources. The trainer accesses the HPC system via a web browser, using the end-user company's guest WIFI network. The four end-user trainees each access the HPC system from their local workstations, also directly in the web browser.

USER EXPERIENCE

The end-users and trainer used Fluent on their own workstations. The login process is simple, getting files in and out of the HPC Cloud system works without any problem using a web-based file exchange system, in this case Dacolt's Basecamp account. The whole experience was so natural that it seemed as if this way of working was daily routine. For the trainer, the UberCloud container technology provides a very simple and scalable solution to provide training in the field of HPC, having to bring only a laptop computer.

BENEFITS

1. The UC Ansys container is very intuitive to use, it is a remote desktop running within the web browser. Also non-Linux users did not have any trouble to run their tutorials.
2. For the end-user, the company did not have to prepare any logistics to host the training.
3. For the trainer, the logistics only consisted in being on time, knowing the required resources were up and running in the Cloud.

CHALLENGES

1. The only real challenge encountered was on the back-end, to let the UC Ansys containers with Fluent check-out a license from the CPU24/7 license server. Through very effective team work and excellent support from Ansys, UberCloud and CPU24/7 resolved this issue swiftly.

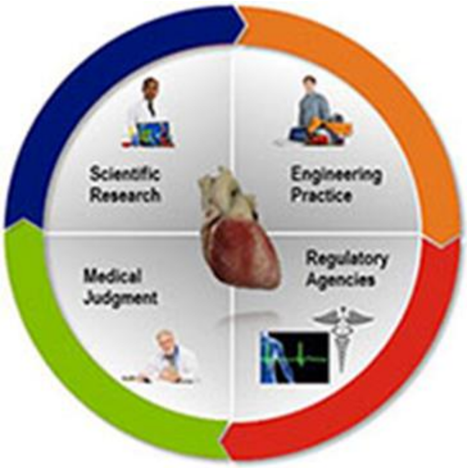
CONCLUSION & RECOMMENDATIONS

1. The selected HPC Cloud environment with the UberCloud Ansys container was a very good combination to provide the training to multiple trainees on a customer site.
2. The HPC resources from CPU24/7 were more than sufficient to allow the users to run their tutorials.
3. The light-weight web-access to the training environment is very comfortable for both trainer and trainees.

Case Study Author – Dr. Ferry Tap, Dacolt

DASSAULT SYSTEMES Abaqus for Healthcare

Studying Drug-induced Arrhythmias of a Human Heart



"We were able to easily access sufficient HPC resources to study drug-induced arrhythmias in a reasonable amount of time. With our local machines, with just 32 CPU cores, these simulations would have been impossible."

MEET THE TEAM

End User – Francisco Sahli Costabal, PhD Candidate, and Prof. Ellen Kuhl, Living Matter Laboratory at Stanford University.
Software Provider – Dassault/SIMULIA (Tom Battisti, Matt Dunbar) providing Abaqus 2017 software and support.
Resource Provider – Advania Cloud in Iceland (represented by Aegir Magnusson and Jon Tor Kristinsson), with access and support for the HPC server from HPE.
HPC Cloud Experts – Fethican Coskuner and Wolfgang Gentzsch, UberCloud, with providing novel HPC container technology for ease of Abaqus cloud access and use.
Sponsor – Hewlett Packard Enterprise, represented by Stephen Wheat, Bill Mannel, Jean-Luc Assor.

"Our successful partnership with UberCloud has allowed us to perform virtual drug testing using realistic human heart models. For us, UberCloud's high-performance cloud computing environment and the close collaboration with HPE, Dassault, and Advania, were critical to speed-up our simulations, which help us to identify the arrhythmic risk of existing and new drugs in the benefit of human health."

Prof. Ellen Kuhl, Head of Living Matter Laboratory at Stanford University

USE CASE

This cloud experiment for the Living Heart Project (LHP) is a follow-on work of Team 196 first dealing with the implementation, testing, and Proof of Concept in the Cloud. It has been collaboratively performed by Stanford University, SIMULIA, Advania, UberCloud, and sponsored by Hewlett Packard Enterprise. It is based on the development of a Living Heart Model that encompasses advanced electro-physiological modelling. The goal is to create a biventricular finite element model to study drug-induced arrhythmias of a human heart.

The **Living Heart Project** is uniting leading cardiovascular researchers, educators, medical device developers, regulatory agencies, and practicing cardiologists around the world on a shared mission to develop and validate highly accurate personalized digital human heart models. These models will establish a unified foundation for cardiovascular in silico medicine and serve as a common technology base for education and training, medical device design, testing, clinical diagnosis and regulatory science —creating an effective path for rapidly translating current and future cutting-edge innovations directly into improved patient care.

Cardiac arrhythmias can be an undesirable and potentially lethal side effect of drugs. During this condition, the electrical activity of the heart turns chaotic, decimating its pumping function, thus diminishing the circulation of blood through the body. Some kind of arrhythmias, if not treated with a defibrillator, will cause death within minutes.

Before a new drug reaches the market, pharmaceutical companies need to check for the risk of inducing arrhythmias. Currently, this process takes years and involves costly animal and human studies. With this new software tool, drug developers would be able to quickly assess the viability of a new compound. This means better and safer drugs reaching the market to improve patients' lives.

The Stanford team in conjunction with SIMULIA have developed a multi-scale 3-dimensional model of the heart that can predict the risk of this lethal arrhythmias caused by drugs. The project team added several capabilities to the Living Heart Model such as highly detailed cellular models, the ability to differentiate cell types within the tissue and to compute electrocardiograms (ECGs). A key addition to the model is the so-called Purkinje network. It presents a tree-like structure and is responsible of distributing the electrical signal quickly through the ventricular wall. It plays a major role in the development of arrhythmias, as it is composed of pacemaker cells that can self-excite. The inclusion of the Purkinje network was fundamental to simulate arrhythmias. This model is now able to bridge the gap between the effect of drugs at the cellular level to the chaotic electrical propagation that a patient would experience at the organ level.

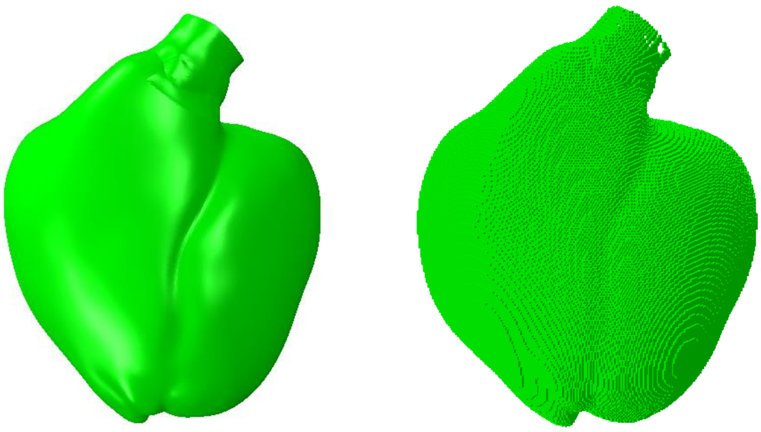


Figure 1: Tetrahedral mesh (left) and cube mesh (right)

A computational model that is able to assess the response of new drug compounds rapidly and inexpensively is of great interest for pharmaceutical companies, doctors, and patients. Such a tool will increase the number of successful drugs that reach the market, while decreasing cost and time to develop them, and thus help hundreds of thousands of patients in the future. However, the creation of a suitable model requires taking a multiscale approach that is computationally expensive: the electrical activity of cells is modelled in high detail and resolved simultaneously in the entire heart. Due to the fast dynamics that occur in this problem, the spatial and temporal resolutions are highly demanding.

During the preparation and Proof of Concept phase (UberCloud Experiment 196) of this LHP project, we set out to build and calibrate the healthy baseline case, which we then used to perturb with different drugs. After creating the UberCloud software container for SIMULIA's Abaqus 2017 and deploying it on HPE's server in the Advania cloud, we started refining the computational mesh which consisted of roughly 5 million tetrahedral elements and 1 million nodes. Due to the intricate geometry of the heart, the mesh quality limited the time step, which in this case was 0.0012 ms for a total simulation time of 5000 ms. After realizing that it would be very difficult to calibrate our model with such a big runtime, we decided to work on our mesh, which was the current bottleneck to speed up our model. We created a mesh that was made out of cube elements (Figure 1). With this approach, we lost the smoothness of the outer surface, but reduced the number of elements by a factor of ten and increased the time step by a factor of four, for the same element size (0.7 mm).

Figure 2: The final production model with an element size of 0.3 mm. The Purkinje network is shown in white. Endocardial, mid layer and epicardial cells are shown in red, white and blue respectively.

After adapting all features of the model to this new mesh with now 7.5 million nodes and **250,000,000 internal variables that are updated and stored within each step of the simulation** (Figure 2), we were able to calibrate the healthy, baseline case, which was assessed by electro-cardiogram (ECG) tracing (Figure 3) that recapitulates the essential features.

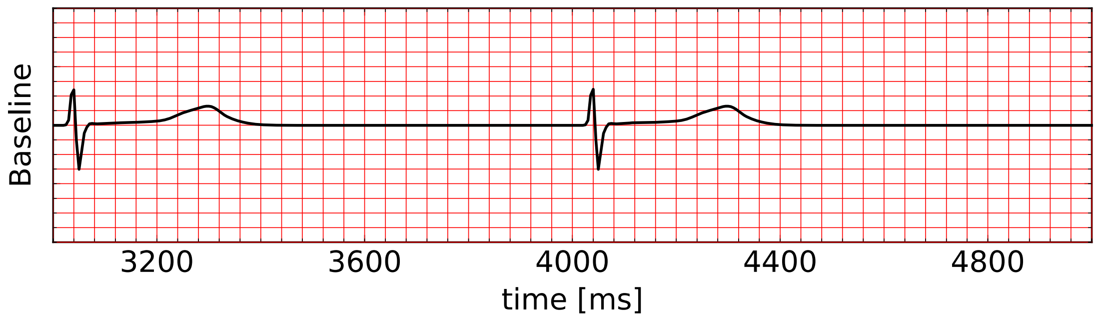


Figure 3: ECG tracing for the healthy, baseline case.

During the final production phase, we have run 42 simulations to study whether a drug causes arrhythmias or not. With all these changes we were able to **speed up one simulation by a factor of 27** which then (still) took 40 hours using 160 CPU cores on Advania's HPC as a Service (HPCaaS) hardware configuration built upon HPE ProLiant servers XL230 Gen9 with 2x Intel Broadwell E5-2683 v4 with Intel OmniPath interconnect. We observed that the model scaled without a significant loss of performance up to 240 compute cores, making the 5-node sub-cluster of the Advania system an ideal candidate to run these compute jobs. In these simulations, we applied the drugs by blocking different ionic currents in our cellular model, exactly replicating what has been observed before in cellular experiments. For each case, we let the heart beat naturally and see if the arrhythmia is developing.

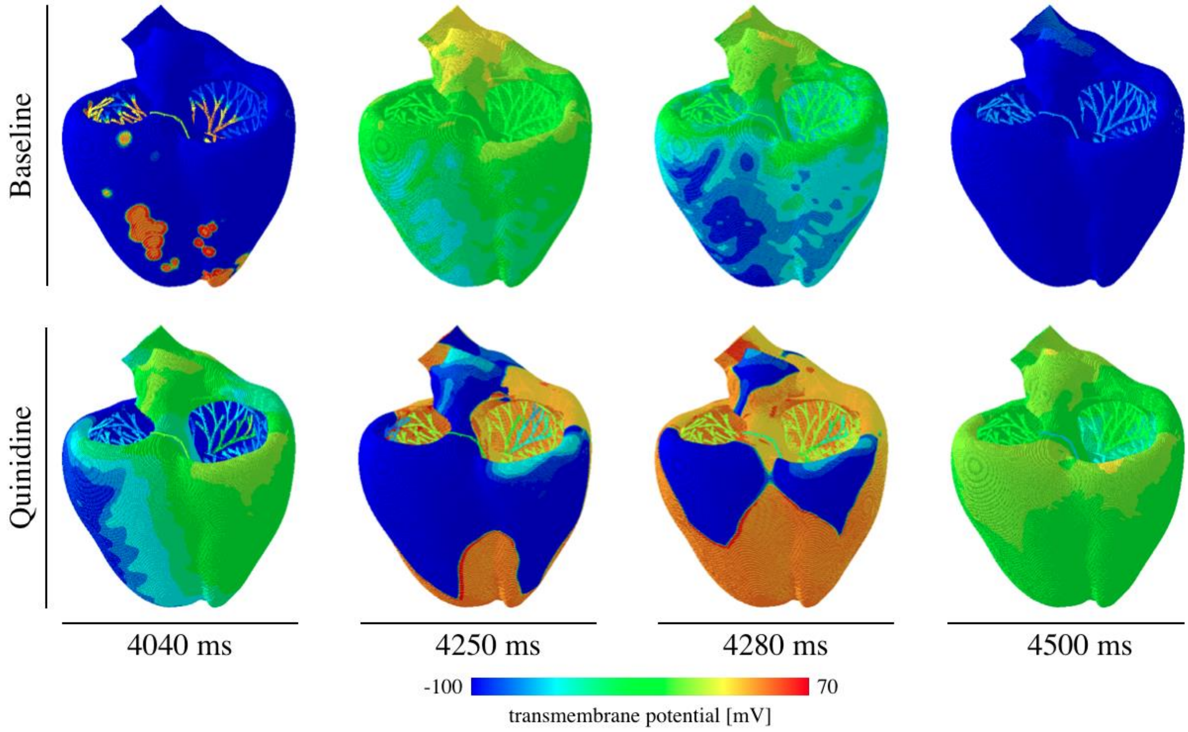


Figure 4: Evolution of the electrical activity for the baseline case (no drug) and after the application of the drug Quinidine. The electrical propagation turns chaotic after the drug is applied, showing the high risk of Quinidine to produce arrhythmias.

Figure 4 shows the application of the drug Quinidine, which is an anti-arrhythmic agent, but it has a high risk of producing Torsades de Pointes, which is a particular type of arrhythmia. It shows the electrical transmembrane potentials of a healthy versus a pathological heart that has been widely used in studies of normal and pathological heart rhythms and defibrillation. The propagation of the electrical potential turns chaotic (Figure 4, bottom) when compared to the baseline case (Figure 4, top), showing that our model is able to correctly and reliably predict the anti-arrhythmic risk of commonly used drugs. We envision that our model will help researchers, regulatory agencies, and pharmaceutical companies rationalize safe drug development and reduce the time-to-market of new drugs.

Some of the challenges that we faced during the project were:

- Although the remote desktop setup enabled us to visualize the results of our model, it was not possible to do more advanced operations. The bandwidth between the end user and the servers was acceptable for file transfer, but not enough to have a fluid remote desktop. We suggested to speed-up remote visualization which has now been implemented including NICE Software's DCV into the UberCloud software container, making used of GPU accelerated data transfers.
- Running the final complex simulations first on the previous-generation HPC system at Advania took far too long and we would have not been able to finish the project in time. Therefore, we moved our Abaqus 2017 container seamlessly to the new HPC system (which was set up in July 2017) and got an immediate speedup of 2.5 between the two HPE systems.

Some of the benefits that we experienced:

- Gaining easy and intuitive access to sufficient HPC resources enabled us to study drug-induced arrhythmias of a human heart in a reasonable amount of time. With our local machines, with just 32 CPU cores, these simulations would have been impossible.

- As we had a dedicated 5-node HPC cluster in the cloud, it was easy to run post-processing scripts, without the need of submitting a second job in the queue, which would be the typical procedure of a shared HPC resource.
- Since all project partners had access to the same Abaqus 2017 container on the HPC server, it was easy to jointly debug and solve problems as a team. Also, sharing models and results between among the end user and the software provider was straight-forward.
- The partnership with UberCloud has allowed us to perform virtual drug testing using realistic human heart models. For us, UberCloud's high-performance cloud computing environment and the close collaboration with HPE, Dassault, and Advania, were critical to speed-up our simulations, which help us to identify the arrhythmic risk of existing and new drugs in the benefit of human health.

Case Study Author – Francisco Sahli Costabal together with Team 197.

Appendix

This research has been presented at the Cardiac Physiome Society Conference in Toronto November 6 – 9, 2017, https://www.physiome.org/cardiac2017/index.html.

Title: Predicting drug-induced arrhythmias by multiscale modeling

Presented by: Francisco Sahli Costabal, Jiang Yao, Ellen Kuhl

Abstract: Drugs often have undesired side effects. In the heart, they can induce lethal arrhythmias such as Torsades de Points. The risk evaluation of a new compound is costly and can take a long time, which often hinders the development of new drugs. Here we establish an ultra high resolution, multiscale computational model to quickly and reliably assess the cardiac toxicity of new and existing drugs. The input of the model is the drug-specific current block from single cell electrophysiology; the output is the spatio-temporal activation profile and the associated electrocardiogram. We demonstrate the potential of our model for a low risk drug, Ranolazine, and a high risk drug, Quinidine: For Ranolazine, our model predicts a prolonged QT interval of 19.4% compared to baseline and a regular sinus rhythm at 60.15 beats per minute. For Quinidine, our model predicts a prolonged QT interval of 78.4% and a spontaneous development of Torsades de Points both in the activation profile and in the electrocardiogram. We also study the dose-response relation of a class III antiarrhythmic drug, Dofetilide: At low concentrations, our model predicts a prolonged QT interval and a regular sinus rhythm; at high concentrations, our model predicts the spontaneous development of arrhythmias. Our multiscale computational model reveals the mechanisms by which electrophysiological abnormalities propagate across the spatio-temporal scales, from specific channel blockage, via altered single cell action potentials and prolonged QT intervals, to the spontaneous emergence of ventricular tachycardia in the form of Torsades de Points. We envision that our model will help researchers, regulatory agencies, and pharmaceutical companies to rationalize safe drug development and reduce the time-to-market of new drugs.

DASSAULT SYSTEMES Abaqus for Structures

Heavy Duty ABAQUS Structural Analysis

"The major challenge – and now widely accepted to be the most critical – was the end user perception and acceptance of the cloud as a smooth part of the workflow."

MEET THE TEAM
End User – Frank Ding, Engineering Analysis and Computing Manager at Simpson Strong-Tie
Software Provider – Matt Dunbar, Chief Architect and CAE technical specialist at Simulia Dassault Systems/SIMULIA
Resource Provider – Steve Hebert, founders and CEO of Nimbix, and Rob Sherrard, co-founder of Nimbix and VP of Service Delivery.
HPC Expert and Team Manager – Sharan Kalwani, independent HPC Segment Architect
Antonio Arena, Solutions Architect, NICE software, network and "middleware" team expert
Cynthia Underwood, IT consultant and subject matter expert at NICE Software
Dennis Nagy, Mentor, Principal at Beyond CAE.

USE CASE
In the first project, previously, the team established that indeed computational use cases could be submitted successfully using the cloud API and infrastructure. The objective of this second part of the project was to explore the following:

- How can the end user experience be improved? For example, how could the post processing of HPC CAE results kept in the cloud be viewed at the remote desktop?
- Was there any impact of the security layer on the end user experience?

The end to end process remains – widely dispersed end user demand was tested in two different geographic areas: Continental USA and Europe. The network bandwidth and latency were expected to play a major role since it impacts the workflow and user perception of the ability to deliver cloud HPC capability – not in the compute, but in the pixel manipulation domain.

Here is an example of the workflow:
1. Once the job finishes, the end user receives a notification email, the results files remain at the cloud facility – i.e. they are NOT transferred back to the end user's workstation for post-processing
2. The post-processing is done using remote desktop tool – in this case NICE-Software DCV infrastructure layer on the HPC provider's visualization node(s).

Typical network transfer sizes (upstream and downstream) were expected to be modest, and it is this impact that we hoped to measure – thus making them “tunables”. This represented the major component of the end user experience.

The team also expanded by almost 100%, to help bring in more expertise and support to tackle the last stage of the whole process and make the end user experience “adjustable” depending on several network layer related factors.

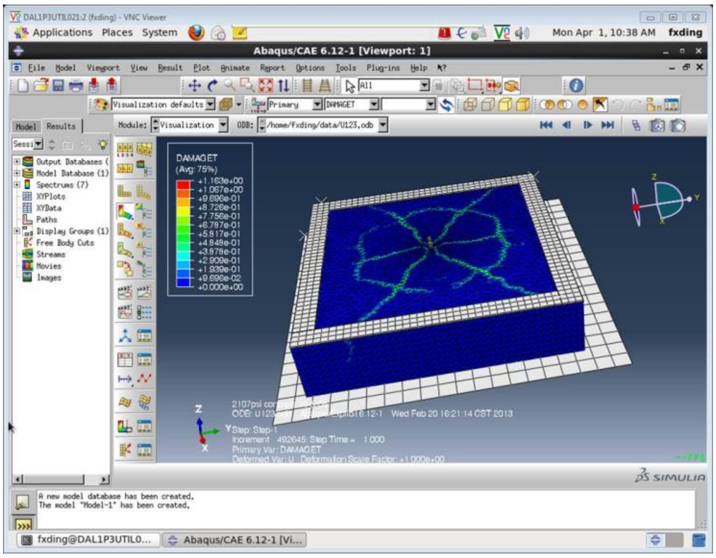


Fig 1. Typical end user screen manipulation(s)

CHALLENGE

The major challenge – and now widely accepted to be the most critical – was the end user perception and acceptance of the cloud as a smooth part of the workflow. Here remote visualization was necessary to see if the simulation results (left remotely in the cloud) could be viewed and manipulated as if it were local on the end user desktop. To contrast with Round 1, and to get real network expertise to bear on this aspect, NICE’s DCV was chosen to help deliver this, as it is:

- Application neutral
- Has a clean and separate client (free) and server component
- Provided some tuning parameters which can help overcome the bandwidth issues

Several tests were conducted and carefully iterated, such as image update rate, bandwidth selection, codecs, etc. A screen shot is shown below for the final successful user acceptance of remote visualization settings:

TABLE 1. CAST-IN-PLACE MECHANICAL ANCHOR CONCRETE ANCHORAGE PULLOUT CAPACITY ANALYSIS (FEA STATS)

Materials	Steel & Concrete
Procedure	3D Nonlinear Contact, Fracture & Damage Analysis
Number of Elements	1,626,338
Number of DOF	1,937,301
Solver	ABAQUS/Explicit in Parallel
Solving Time	11.5 hours on a 32-core Linux Cluster
ODB Result File Size	2.9 GB

The post processing underlying Infrastructure (cloud end):

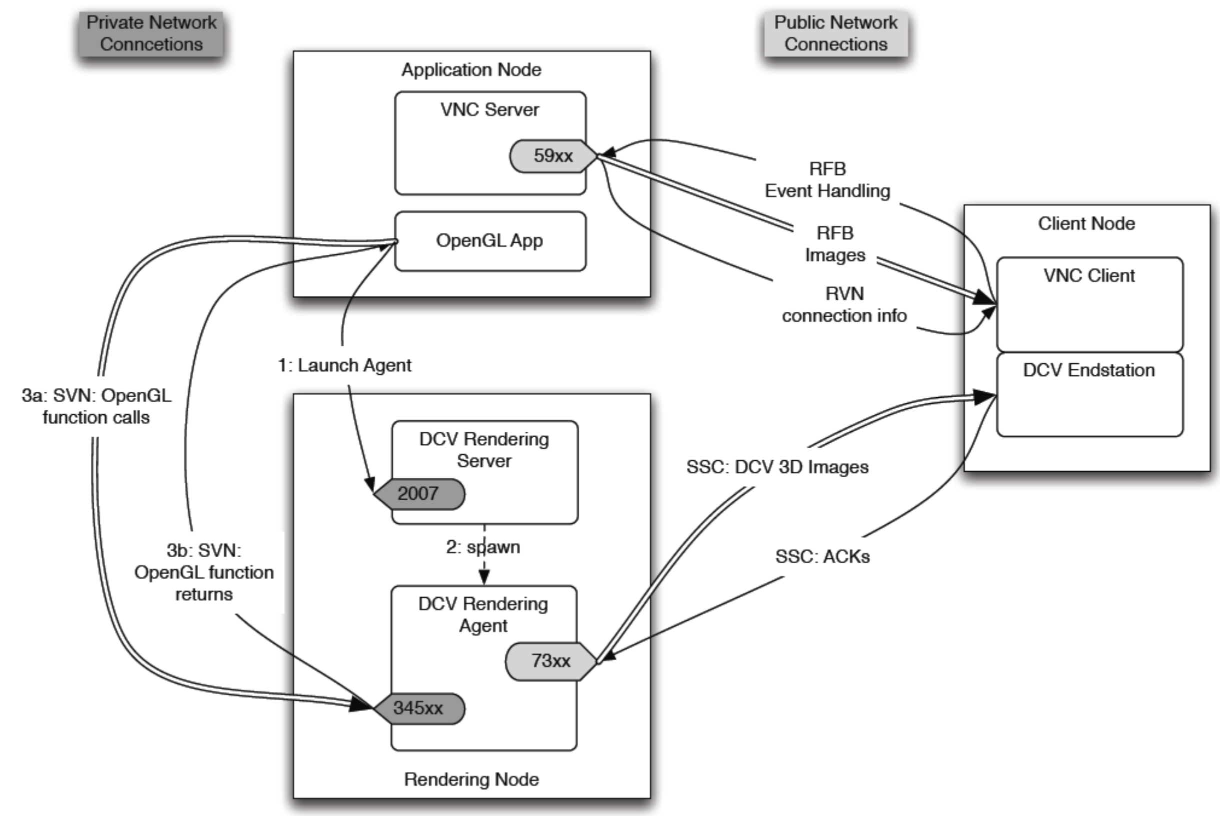


Fig 2. DCV layer setup

The post processing underlying Infrastructure (end user space):

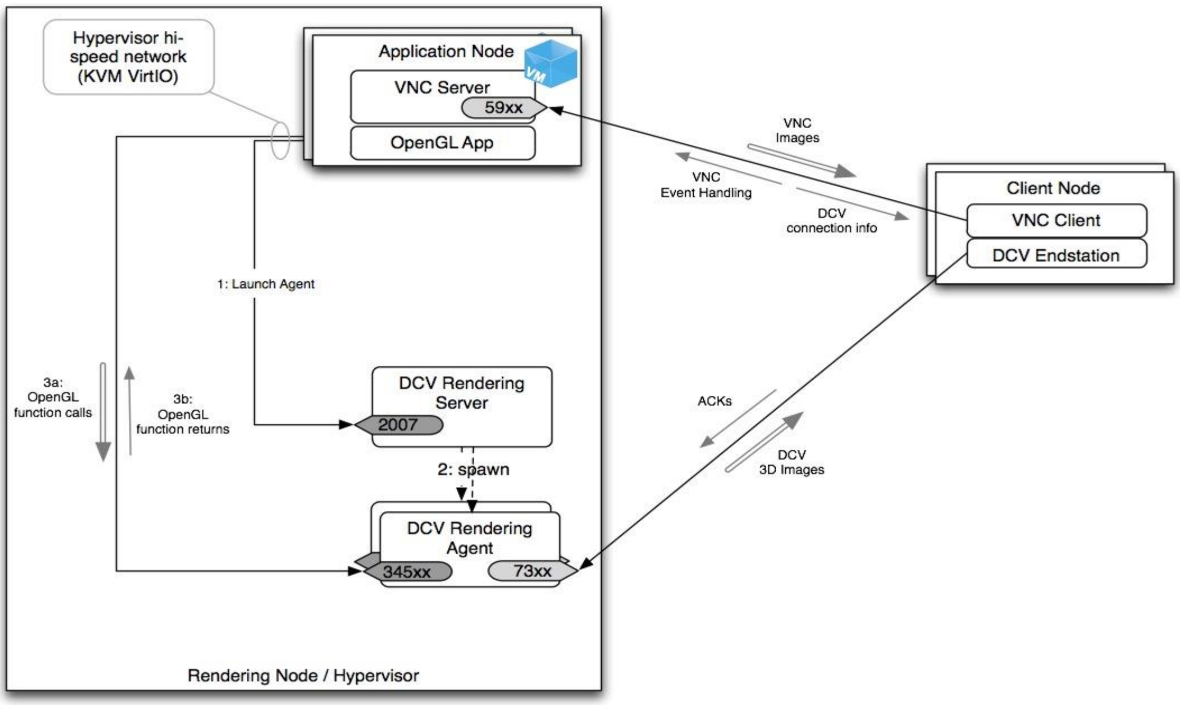


Fig 3. DCV enabled post processing (end user view)

Setup

We made a number of end user trails. First the DCV was installed with both a Windows and Linux client. Next a portal window was opened, usually at the same time as the end user trial to observe the demand on the serving infrastructure (see diagram). This ensured that there was sufficient bandwidth and capacity at the cloud end .

The end node was hosting an NVIDIA graphics accelerator card. An initial concern was if the version was supported or had an impact. DCV has the ability to do a "sliding scale" of the pixel compression and this involves skipping certain frames in order to keep the flow smooth.

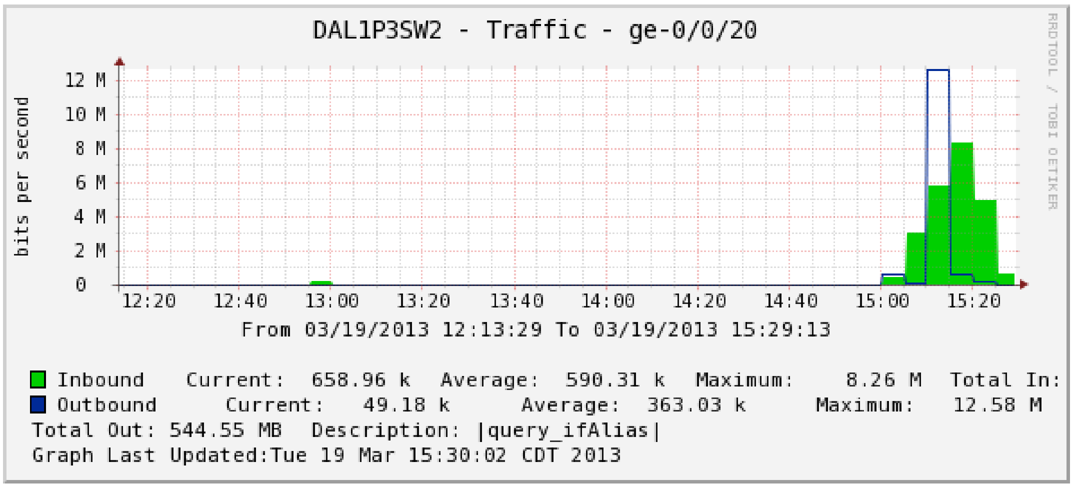


Fig 4. Ingress/Egress test results/profile

Figure 4 basically shows us that the cloud Internet measurements peaked out at 12 Mbits/sec, but generally hover at or below 8 Mbits for this particular session. This profile graph is a good representation of what has been seen in the past on DCV sessions. The red line (2 Mbits/sec) is where consistent end user experience for this particular graphic size was "observed."

CONCLUSIONS AND RECOMMENDATIONS

Here is a summary of key results:

- End point Internet bandwidth variability: Depending on when it is conducted, a vendor neutral test applet result ranges from 1Mbps ~ 10Mbps. The pipe Bandwidth was expected to be 20 Mbits/sec, but when it was shared by the office site using normal enterprise applications such as server Exchange, Citrix, etc., such variation was not conducive to a qualitative end user experience.
- Switching to another pipe (with burst mode of 50 to 100 Mbits/sec): More testing showed that the connection was not stable, and ABAQUS/Viewer graphics windows freeze was experienced after being idle for a while. This required local IT to troubleshooting the issue.
- There were no significant differences between Windows or Linux hosted platforms. The NICE DCV/EnginFrame is a good platform for remote visualization if a stable Internet BW is available. Some of the parameters for the connection performance:
  - VNC connection line-speed estimate: 1 ~ 6 Mbps, RTT ~ 62 ms
  - DCV bandwidth usage: AVG 100 KiB ~ 1 MiB
  - DCV Frame Rate: 0~10 FPS, >5 FPS acceptable, >10 FPS smooth
- We tried both Linux and Windows desktop. Because of the BW randomness & variability, it was not possible create a good baseline to compare the performance of the two desktops.

- The graphics cards did not have any impact on the end user experience. However the model size and graphic image pixel size perhaps play a major role, and the current experiment did not have enough time to study and characterize this issue. The ABAQUS model used in this test case does not put much demand on the graphics card. We've seen only 2% usage on the card.
- There was usually sufficient network capacity and bandwidth at the cloud serving end.
- The “last mile” delivery or capability at the end user site was the most important and perhaps only determining factor influencing the end user experience and perception.
- Beyond the Cloud service provider, a local or end user IT support person with network savvy is perhaps a necessary part of the infrastructure team in order to deliver robust and repeatable post processing visual delivery. This incurs a cost.
- The security aspect could not be tested, as the time and effort required were not sufficient in the time allotted.
- Part of the end user experience learned from Round 1 was to better document the setup which can be found in the Appendix and clearly shows a smooth and easy to follow up flow.

Major single conclusion and recommendations

Any site that wishes to benefit from this experience needs to prioritize the “last mile” issue.

End User Experience Observations & Data Tables:

Use Case	Measure	Remote thru DCV	Local Desktop
Loading ODB file	response time (seconds)	2	1.5
Copy & paste contour plots	functionality	does not work	works
Copy & paste XY data	functionality	works	works
Creating animation	response time (seconds)	12	8
Model dynamic manipulation	response time	Acceptable when Bandwidth > 2 Mbits/sec	no delay

Bandwidth Usage

	Average (KiB)	Peak (MiB)
Image Quality = 10	450	1.1
Image Quality = 80	750	2.1

Note:

Image Quality: Specify the quality level of dynamic images when using TCP connections. Higher values correspond to higher image quality and more data transfer. Lower values reduce quality and reduce bandwidth usage.

Network Latency for round trip from the DCV remote visualization server

Ping statistics for 70.36.18.101: Packets: Sent = 4, Received = 3, Lost = 1 (25% loss).

Approximate round trip times in milli-seconds: Minimum = 56ms, Maximum = 58ms, Average = 56ms.

Case Study Author – Frank Ding

ESI Group OpenFOAM

Aerodynamic Simulation of an Airfoil using UberCloud Containers on Microsoft Azure

"The combination of Microsoft Azure with UberCloud's OpenFOAM showcased the possibilities and easiness in performing highly complex simulations in the cloud."

MEET THE TEAM

End-User/CFD Expert – Praveen Bhat, Technology Consultant, INDIA.
Software Provider – ESI Group with OpenFOAM.
Resource Provider – Microsoft Azure with UberCloud OpenFOAM Container

USE CASE

The aerodynamic study was performed with an incompressible air flow around a 2D airfoil. The model setup included the geometry preparation for a surrounding air volume with the airfoil profile at the center. The airfoil profile needed to be accurately modelled to capture the variation in the airflow pattern around the airfoil. The model setup was done using the open source software OpenFOAM. UberCloud's OpenFOAM containerized software is available on the Microsoft Azure Cloud and in the Azure Store on the UberCloud Marketplace. The CFD simulations were performed on Azure. The main objective of this project was to evaluate the HPC performance of Azure.

PROCESS OVERVIEW

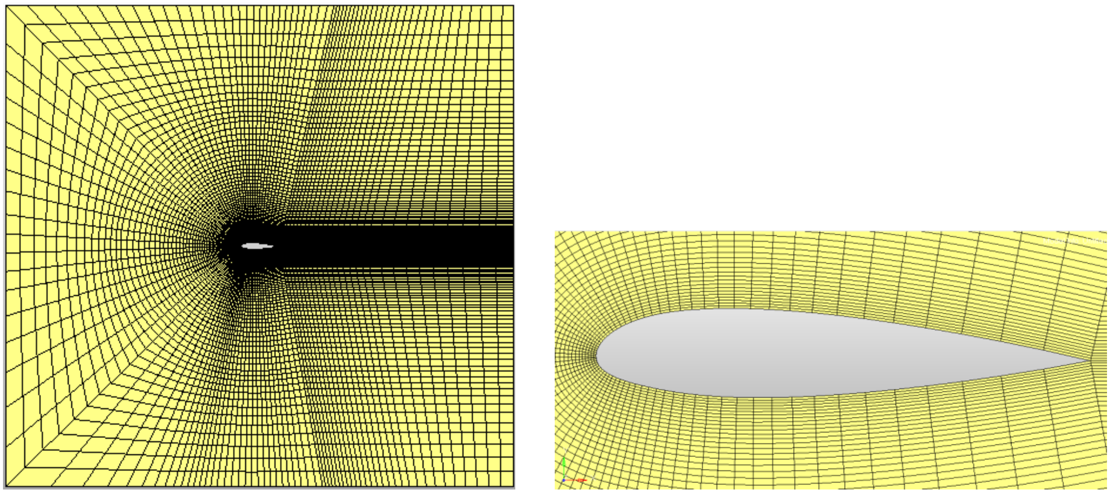


Figure 1: Mesh model for Aerofoil.

The meshing density was very fine around the airfoil and also along the path of the trailing edge. The grid was modelled more coarsely as it moved away from the airfoil region. The coarseness of the grid increased near the air volume boundary (air inlet and outlet). The following steps were performed in the simulation setup using OpenFOAM:

1. The Finite Element volume mesh model was generated, which was followed by the fluid properties definition. The volume surrounding the airfoil was incompressible air.
2. The fluid properties were defined as Newtonian, which posits a linear relationship between the shear stress (due to internal friction forces) and the rate of strain of the fluid.
3. The atmospheric air was turbulent in nature and there was a phase of transition from turbulent to laminar in the region near the airfoil. Because of this transition behavior the mesh model needed to be refined accurately near the airfoil region along with defining the turbulence behavior of the air. The behavior was captured through defining turbulence models which in this specific case was considered as a Spalart – Allmaras turbulence model.
4. Defining model boundary conditions and assigning the pressure and velocity initial values. Boundary conditions were assigned where the airfoil edges were considered as walls. Three sides were inlets; the edge following the trailing edge of airfoil was considered as air outlet.
5. Setting up of the solution algorithm. The problem was solved as steady state and the OpenFOAM solver used was SimpleFOAM. The solution parameters were: Start time: 0 sec; end time=500 sec; time step= 1sec. SimpleFOAM used the Gauss-Seidel method. The pressure field was provided with a relaxation factor of 0.3 and the velocity field was assigned a relaxation factor of 0.7. Along with the relaxation factor, the residual parameter was set at $1x10^{-5}$. The above parameters defined the convergence criteria of the model.
6. The OpenFOAM model was then modified for parallel processing with the existing model decomposed according to the number of the available compute nodes. Independent processes were created for the decomposed model with each process on one processor.
7. The model was solved in parallel and once the solution converged, the decomposed solved model was reconstructed to get the final results. Next, the output of the airfoil simulation was viewed using the post-processing software tool ParaView. The different plots in the sections below show the flow of air and laminar behaviour observed in the airfoil region.

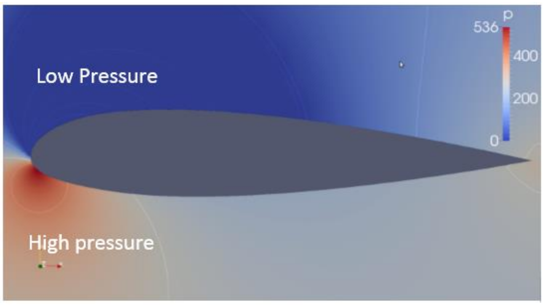


Figure 2: Pressure distribution around airfoil with high & low pressure zone.

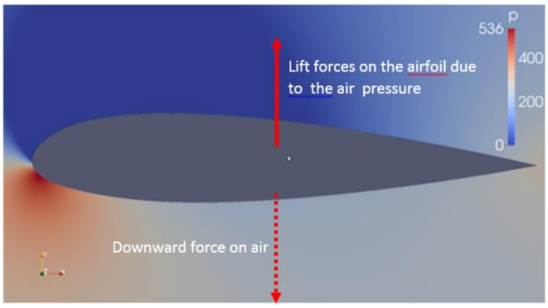


Figure 3: Lift forces represented in the airfoil.

The pressure plot shows the air pressure distribution in the airfoil sections. The first diagram represents the pressure variation around the airfoil where we observe the low pressure region at the upper section of the leading edge of the airfoil and a higher pressure region in the lower section of the leading edge. The low pressure and high pressure variation section in the air volume is shown in the second diagram and the high pressure section near the airfoil creates a lift forces on the airfoil. The lift on the airplane wing can be considered to be Newton's third law with a reaction force in the form of downward force on the air. The lift on the airplane wing should be consistent since it is the conservation of energy in the fluid. Angle of attack is the orientation of the airfoil cord with respect to the travel direction. The state of stall can be analysed by determining the pressure co-efficient distribution over the airfoil for various angles of attack and evaluate how the pressure co-efficient varies with the increase or decrease in the angle of attack.

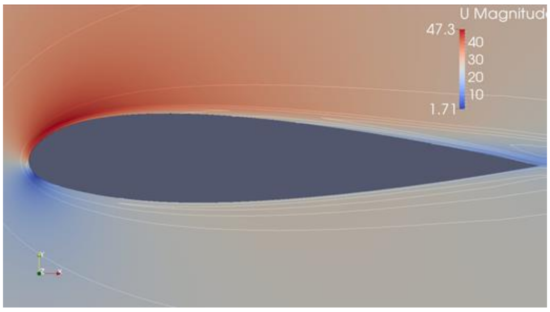


Figure 4: Velocity contour of streamlines.

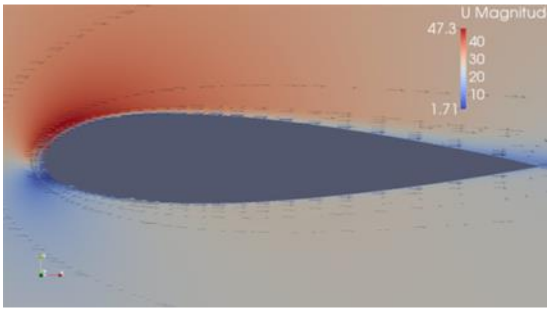


Figure 5: Velocity contour with air flow vectors.

The behavior of air flow will be turbulent in the air volume and the transition of the air behavior from turbulent to laminar is observed in the air volume nearing the airfoil section, and the flow behavior of air will be laminar around the wall of the airfoil. The airflow path it follows near the wall boundary of the airfoil is laminar which is evident from the Figures 4 and 5. The vector flow path of the air in the airfoil region is also represented where the flow path represents the individual air particle flow near the wall boundary of the airfoil.

HPC PERFORMANCE BENCHMARKING

The HPC system is a 32 core system with 32 GB RAM CentOS v6 operating system. The software installed is OpenFOAM version 2.3 with pre-installed MPI and ParaView software in the UberCloud container which is integrated with the Microsoft Azure cloud platform. The model was evaluated for the accuracy of prediction of air flow behavior around the airfoil. Different finite volume models were developed for fine and coarse meshes. The time required for solving the model with different mesh densities was then captured. The boundary conditions, solution algorithm, solver setup and convergence criteria remained the same for all the models. The simulations were performed on Azure G5 virtual machine instances featuring latest Intel Xeon processor E5 v3 family.

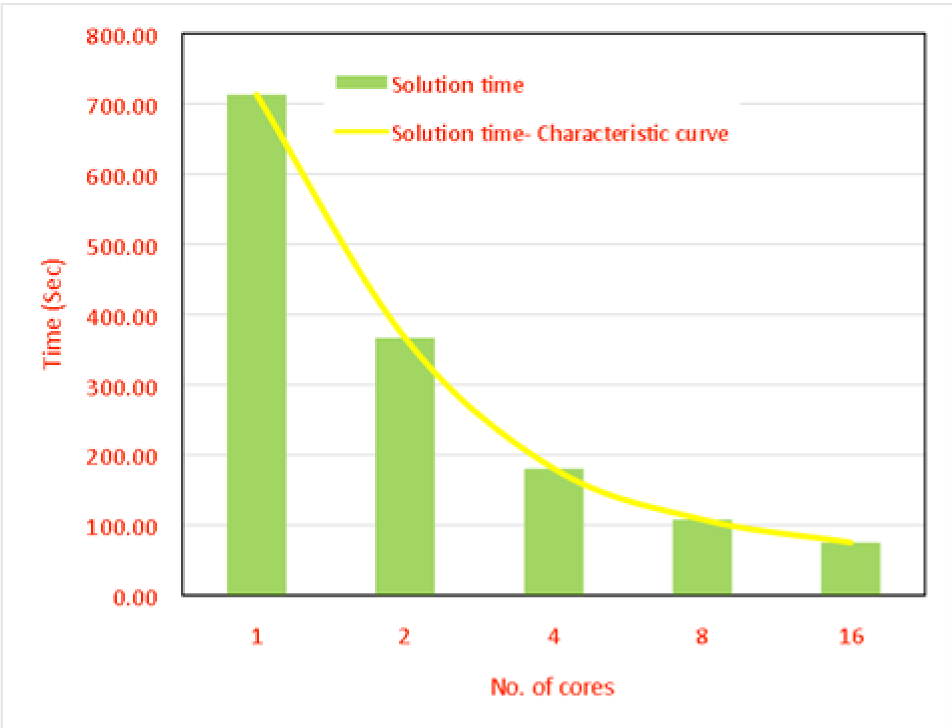


Figure 6: Solution time for a model with 375K elements using different core configuration.

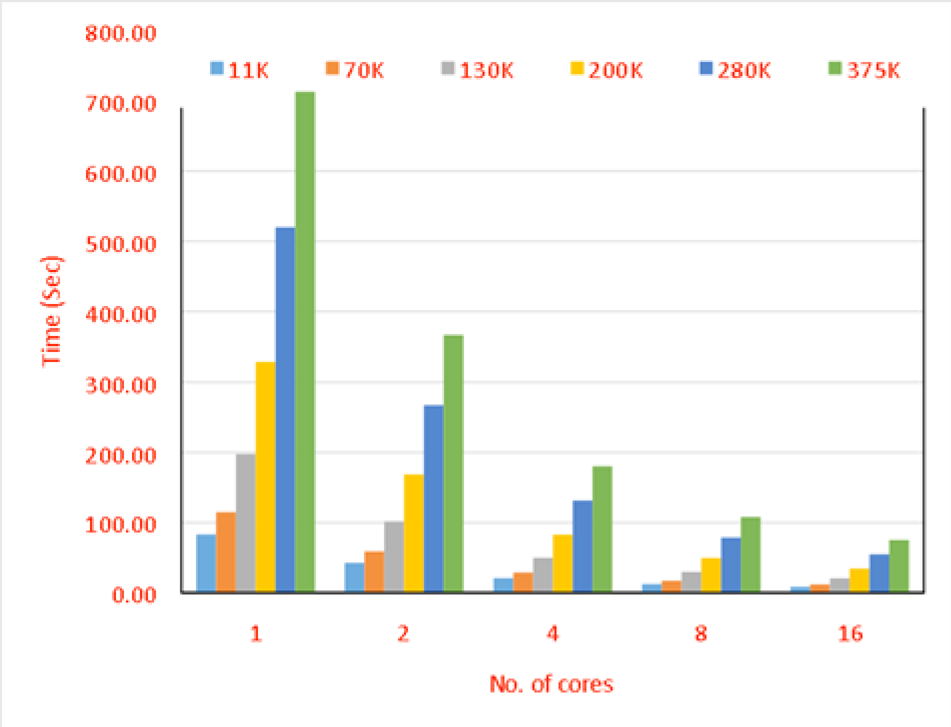


Figure 7: Solution time for different mesh density models solved using different core configurations.

The simulations were performed using the UberCloud OpenFOAM container on the Microsoft Azure. The OpenFOAM instance has been controlled through VNC. The performance of the solver in the UberCloud OpenFOAM container was recorded while the simulation runs were in progress.

EFFORT INVESTED

End user/Team Expert: 10 hours for simulation setup, technical support, reporting and overall management of the project.
UberCloud support: 1 hour for monitoring and administration of the performance in the host server.
Resources: ~50 core hours were used for performing various iterations of the experiments**.**

CHALLENGES

The project challenges faced were related to technical complexity. This involved accurate prediction of flow behaviour around airfoil which is achieved through defining appropriate element size to the mesh model. The finer the mesh, the higher is the simulation runtime required and so the challenge was to perform the simulation within the stipulated timeline. Getting registered at and exposure to the Azure cloud platform and using its features consumed some time as this required going through, learning and following written instructions provided by Azure.

BENEFITS

1. The UberCloud container environment with OpenFOAM & ParaView made the process of model generation extremely easy with process times reduced drastically along with result viewing and post-processing.
2. The mesh models were generated for different cell numbers where the experiments were performed using coarse-to-fine to highly fine mesh models. The OpenFOAM container helped in achieving smooth completion of the simulation runs without re-trials or resubmission of simulation runs.
3. The computation requirement for a highly fine mesh (2 million cells) is high, which is almost impossible to achieve on a normal workstation. The HPC cloud provided made it feasible to solve

highly fine mesh models with the simulation time drastically reduced, providing the advantage of getting the simulation results within acceptable run time (~30 min).

4. The experiments performed in the HPC Cloud environment provided the confidence to setup and run the simulations remotely in the cloud. The different simulation setup tools required were installed in the UberCloud OpenFOAM container which enables the user to access and use the tools without any prior installations.
5. With UberCloud's VNC Controls in the Web browser, The HPC Cloud access was very easy with minimal or no installation of any pre-requisite software. The whole user experience was similar to accessing a website through the browser.
6. The UberCloud containers provided smooth execution of the project with easy access and use of the server resources. The regular UberCloud email auto-update module provided a huge advantage to the user in that it enabled continuous monitoring of the job in progress without any requirement to log-in to the server and check the status.
7. UberCloud environment integrated in Microsoft Azure platform proved to be powerful as it facilitated running parallel UberCloud containers and also has a dashboard in the Azure environment which helped in viewing the system performance and usage.

CONCLUSION AND RECOMMENDATIONS

1. UberCloud containerized OpenFOAM on Microsoft Azure was a great fit for performing complex simulation that involved huge hardware resource utilization with high numbers of simulation experiments.
2. Microsoft Azure with [UberCloud application container] was a very good fit for performing advanced computational experiments that involve high technical challenges and cannot be solved in a normal workstation.
3. There are different high-end commercial software applications which can be used to perform virtual simulation. OpenFOAM -- an open source tool with HPC environment --helped us to solve this problem with minimal effort in setting up the model and performing the simulation trials.
4. The combination of Microsoft Azure, UberCloud Containers, and OpenFOAM helped in speeding up the simulation trials and also completed the project within the stipulated time frame.

Case Study Author – Praveen Bhat

DYNAmore LS-DYNA

Simulating Car Frontal Impact in the Cloud

> *"The combination of HPC Cloud, CPU24/7 support, and scalability of the LS-DYNA code resulted in huge speed-up of the simulation."*

MEET THE TEAM

End-User/FEA Expert – Dr. Stefan Castravete, General Manager, Caelynx Europe, ROMANIA
Software Provider – Prof. Dr. U. Göhner, DYNAmore GmbH
Resource Provider and HPC Experts – Thomas Gropp, Alexander Heine, Christian Unger, CPU 24/7

USE CASE

A full Toyota Yaris Sedan car model was analyzed using the Finite Element Method provided by the commercial software LS-DYNA. The full car FE Dyna deck (without dummy and airbags) is available over the web at no cost from NCAC. The team added a frontal airbag to this model. The analysis was of a frontal impact with a rigid wall at 56 km/h.

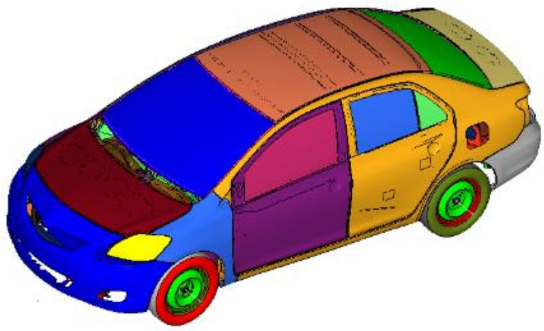


Figure 1: Toyota Yaris Full Vehicle Model

Process Overview

The FEA car model consists of 1479087 nodes and 1517933 elements. The barrier was modelled as a rigid wall. Time of the analysis was 0.2s. Figure 2 shows the model set-up. The model was correlated with physical tests.

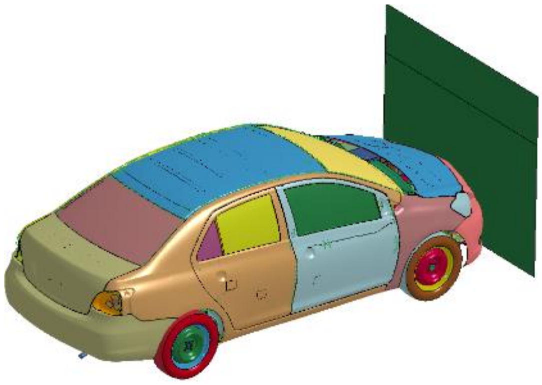


Figure 2: Model Set-up

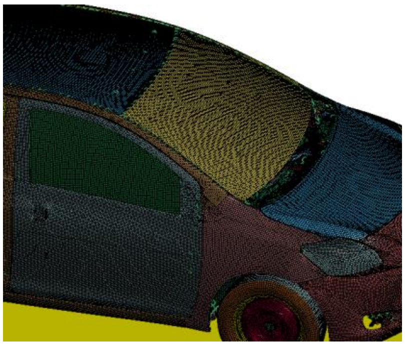


Figure 3: Mesh Sample

An accelerometer was placed on the cross-member of the driver seat as shown in figure 4. Velocity and acceleration was monitored at the location of the accelerometer.

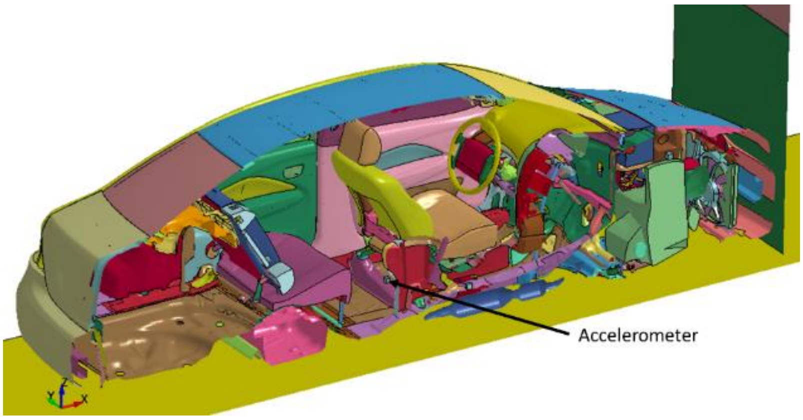


Figure 4: Accelerometer position

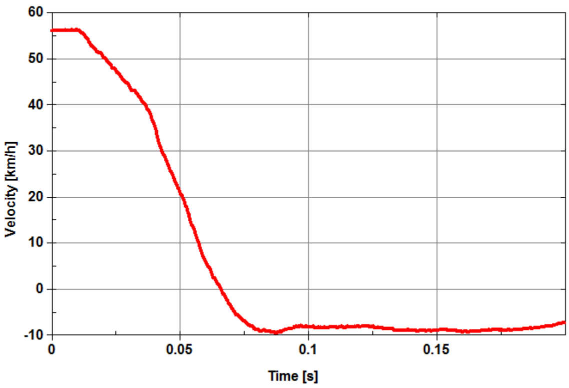


Figure 5: Velocity

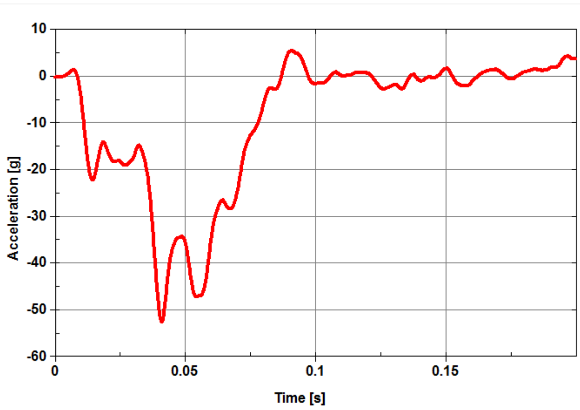


Figure 6: Acceleration

Full impact appears at about 66 ms after that the vehicle rebounded. Maximum deceleration was about 53g. For the acceleration plot, a SAE60 filter was used. Figure 7 shows a sequence of car behaviour at impact.

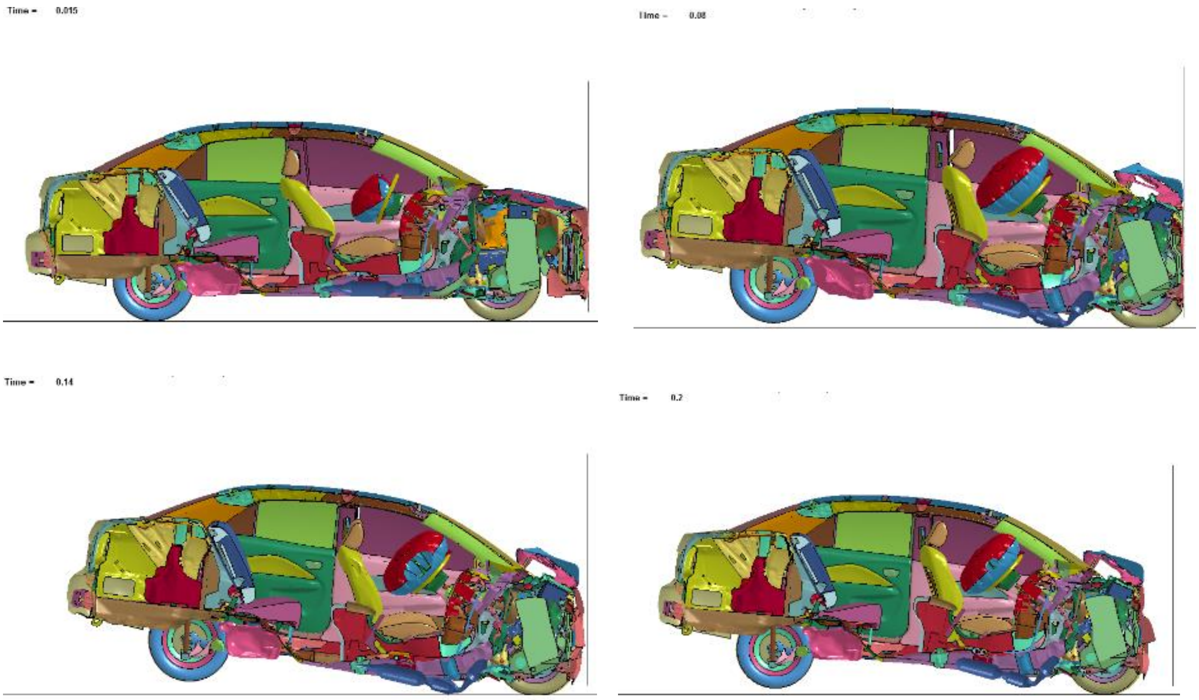


Figure 7: Impact at different times

HPC Performance Benchmarking

The analyses were performed in the CPU24/7 computing cloud where it was scaled with different numbers of CPU cores. The HPC Cloud system used consists of two 12 core machines.

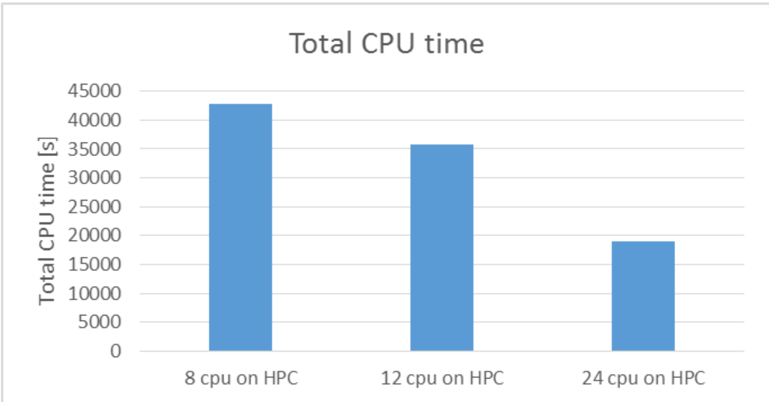


Figure 8: Solution time for a model with 1.5M elements using different HPC core configurations on CPU 24/7 cloud resources

EFFORT INVESTED

End user/team expert: 10 hours for simulation setup, technical support, reporting and overall management of the project.
UberCloud support: 3 hours for monitoring and administration of host servers
Resources: ~120 core hours for performing various iterations in the simulation experiments.

CHALLENGES

Initial challenges were the file transfer to the cloud, running the model, and transferring results back. These challenges were solved in a WebEx training with CPU24/7. The next challenge was the debugging of the model. We started with the model as we downloaded it from the NCAC website. At first the model didn't run completely. There were MPI issues, which were solved by the CPU24/7 team. On the initial model we added the airbag and again that model needed debugging since it didn't run the first time. The model then ran on 8, 12, and 24 cores.

When we tried to run the model on 6 cores, it didn't finish completely – it died after 6 hours. The output of LS-DYNA, unfortunately, did not provide reliable information about the reason for the failure. Therefore, to get an estimate of the run time for the 6-core run, we applied the speed-up factor of the 12:24 core runs (= 1.88) to the 12-core run shown in Figure 9.

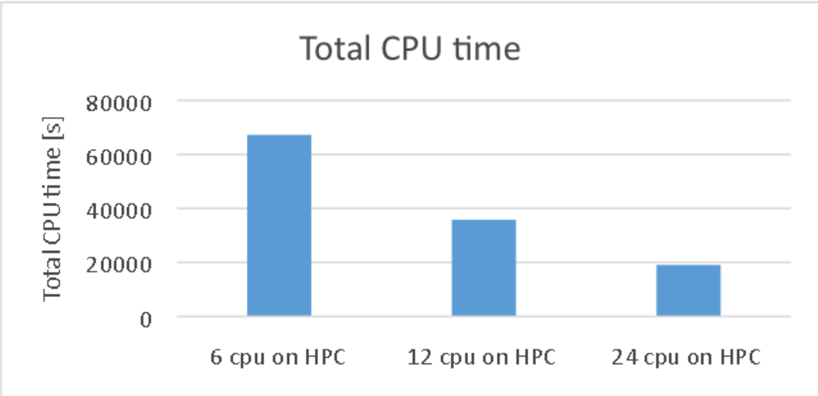


Figure 9: Solution time for a model with 1.5M elements using different HPC core configurations on CPU 24/7 cloud resources (with extrapolated run time for 6 cores)

BENEFITS

1. The HPC cloud computing environment is similar to the one that Caelynx team members are familiar with so the CPU24/7 training on transferring the files and running the jobs was done in a very short time.
2. The file transfer speed was, in general, at acceptable levels. There were periods when the transfer was low and other periods when the transfer was very good.
3. The team of the CPU24/7 computing cloud was very helpful and prompt with answering questions and solving problems.

CONCLUSION AND RECOMMENDATIONS

1. The selected HPC Cloud environment with UberCloud was a very good fit for performing advanced computational experiments that involve high technical challenges and require high-performance hardware resources to perform the simulation experiments.

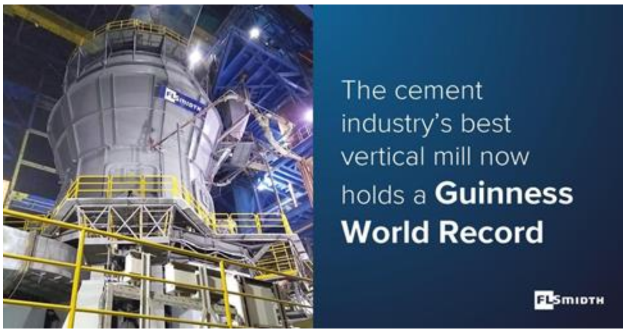
2. Cloud resources were a very good fit for performing advanced computational experiments that involved high technical challenges and required high-performance hardware resources to perform the simulation experiments.

3. The combination of HPC Cloud, CPU24/7 support, and the scalability of DYNAmore's LS-DYNA code helped in speeding up the simulation trials and completing the project within the stipulated time frame.

Case Study Author – Dr. Stefan Castravete, Caelynx Europe

ESSS Rocky

Migrating Engineering Workloads to the Azure Cloud – An FLSmidth Case Study



> *"From all the cloud service providers we have worked with so far, your team at UberCloud is the most professional and the most pleasant one to work with."*

MEET THE TEAM
Customer – Sam Zakrzewski, Fluid Dynamic Specialist and Janaina Hyldvang, IT Infrastructure Architect at FLSmidth A/S, Copenhagen, Denmark.
Engineering Software Provider – Wim Slagter, Director, HPC & Cloud Alliances at ANSYS, Inc.
Cloud Resource Provider – Microsoft Azure.
HPC Cloud Experts –Ozden Akinci, Ronald Zilkovski, and Ender Guler at the UberCloud Inc.
Project Manager – Reha Senturk and Alexander Gatzemeier at the UberCloud Inc.

FLSMIDTH
FLSmidth is the leading supplier of engineering, equipment and service solutions to customers in mining and cement industries. With more than 135 years of experience and with activities in more than 100 countries worldwide, FLSmidth is productivity provider no. 1 to its customers. Sharing customers' ambitions, FLSmidth brings better solutions to light by improving their safety standards and enhancing their performance.

FLSmidth's 11,000+ employees use their unique process knowledge about projects, products and services to meet their customers' needs for technical innovations, digitalization and sustainable life-cycle management. Together with their customers, they challenge conventions, explore opportunities and drive success through sustainable productivity enhancement.

USE CASE
In September 2018, Sam Zakrzewski from FLSmidth approached UberCloud to perform an extensive Proof of Concept to evaluate whether the timing was right to consider moving their engineering simulation workload to the Cloud. During this half-year project, FLSmidth A/S and UberCloud implemented some of the engineering simulations on Microsoft Azure. The UberCloud project team worked with FLSmidth's IT Group and subject matter experts to design and configure FLSmidth's Azure cloud environment for running ANSYS software in an HPC configuration, to benchmark FLSmidth applications, and to understand engineering simulation usage of Cloud HPC resources and related workflows.

The FLSmidth engineering team, which is distributed between locations in Copenhagen, India, South Africa, and Brazil, drafted a list of their actual requirements with a two-year roadmap of moving different simulation scenarios using ANSYS CFX and ESSS Rocky to Azure. FLSmidth currently has its own on-premises Haswell-based HPE cluster with 512 cores and Infiniband FDR for its 20 habitual users. In the next step, it wishes to increase its user base and upgrade the on-premises environment with cloud bursting for mission-critical applications in CFD (multi-phase, combustion) using ANSYS CFX and Fluent, STAR CCM+, and ANSYS Mechanical (static, thermal, modal, fatigue). In addition, the company is applying the discrete element method (DEM) to simulate granular and discontinuous materials with ESSS Rocky.

FLSmidth HPC Cluster in Azure

FLSmidth's IT Team created the pilot cluster in FLSmidth's Azure subscription, and established access for the UberCloud team. The pilot cluster has IP based firewall security to allow proper access and testing. The HPC cluster will be moved to FLSmidth's subnet when it is rolled out to production.

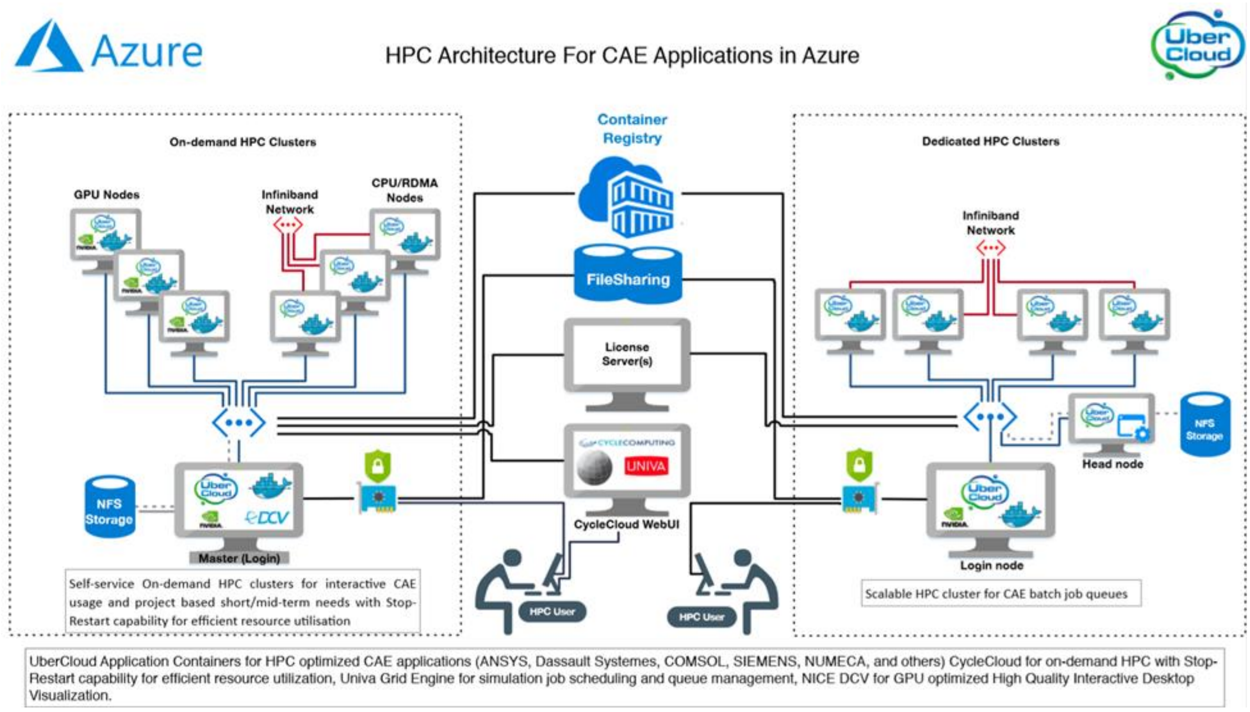


Figure 1: HPC architecture for CAE applications in Azure, with UberCloud CAE containers, CycleCloud, Univa Grid Engine, and NICE DCV.

Based on FLSmidth project team's decision, West Europe and Central India datacenters were identified as POC sites. The Azure West Europe Data Center is the main site for FLSmidth's datacenter migration. India was chosen as the second site to provide regional HPC resources to eliminate their latency issues.

Based on FLSmidth IT requirements and engineering usage needs, UberCloud created a flexible global HPC Cluster Architecture (Figure 1). FLSmidth IT naming standards were used to properly identify the resources. UberCloud created a Cloud HPC Architecture which is able to handle different workloads and easily expands to other sites providing maximum flexibility and cost efficiency. The main site in West Europe was created with two cluster types; one dedicated 256 core cluster for batch jobs managed by Univa Grid Engine based on queues with different priorities, and on-demand HPC clusters are used for all extra workloads and additional simulations that require powerful compute resources. Azure's CycleCloud provides self-service cluster provisioning and management capability directly to engineers without adding any overhead to IT. UberCloud created custom web interfaces in CycleCloud to provide these on-demand resources to engineers based on FLSmidth's consumption rules built into the system. Resource groups were created, 1 TB centralized storage account was created for storing test cases and shared files, nodes were created as head node, compute nodes, visualization nodes, container registry was created as resource group, and a license server was created. Finally, resources in the India cluster were created.

Application: FLSmidth JETFLEX® Kiln Burners

One of FLSmidth's major applications is the simulation of the JETFLEX® kiln burner which offers maximal flexibility for solid pulverized and alternative fuel firing, with a unique design utilizing fixed or rotatable rectangular jet air nozzles, see Figure 2. Fast and easy shipping keeps initial costs low, while a common solid fuel channel reduces usage of cold fuel-conveying air.

The JETFLEX® burner is a highly flexible kiln burner, designed to produce the best flame shape at lowest NOx emissions for various fuel types and operating conditions. It fires rotary kilns ('ovens') with pulverised coal or coke, oil, natural gas, or any mixture of these fuels. Alternative fuel firing of plastic chips, wood chips and sewage sludge can also occur through the same common fuel channel. The JETFLEX® burner is available for any fuel combination and maximum capacity ranging from 10 to 250 MW, catering for even the largest of rotary kilns. The primary air supply to the kiln burner enables a flame momentum of 7 N/MW up to 11 N/MW.

FLSmidth offers two kiln burner models: The standard JETFLEX® burner and the JETFLEX® PLUS burner. The **standard JETFLEX® burner** has no moving parts, offering easy operation and reliability as fewer parts are exposed to wear. The kiln burner flame shape or momentum is easily controlled by simple regulation of the primary air pressure and flow. If you find a series of optimum kiln burner settings work for you, these can be easily repeated. This improves plant production by enabling smooth transition between production qualities or fuels. For optimum combustion flexibility, **JETFLEX® PLUS burner** offers superior combustion of cost-effective grade fuels, complete flame-forming control and increased fuel retention time. The two design features that characterise the JETFLEX® PLUS burner model are rotatable jet air nozzles and a retractable centre pipe for alternative fuel firing. The concentric placement of the rotatable jet air nozzles enables high suspension of the fuel inside the flame. The swirler is the main mechanism for shaping the flame during start-up and daily operation.

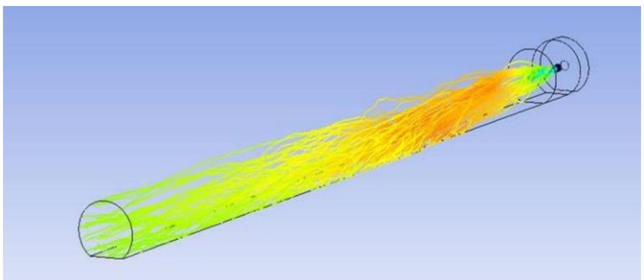



Figure 2: FLSmidth JETFLEX® kiln burner. The simulation shows a multi-phase Lagrangian CFD calculation, including solid fuel combustion.

APPLICATION AND TECHNOLOGY TESTING

A container registry was created in FLSmidth's Azure HPC environment, and UberCloud ANSYS Container images (ANSYS19, ANSYS19.2, and Rocky DEM 4.1 with CPU and GPU versions) were stored. Containers were deployed on the FLSmidth pilot cluster in Microsoft Azure West Europe datacenter. As a part of this project, enabling tools like Nice DCV for high-resolution remote visualization, CycleCloud for resource management, Univa Grid Engine for scheduling, GPU and Infiniband/RDMA were configured and successfully tested.

The test clusters were created for FLSmidth engineers to perform the benchmark tests using ANSYS Mechanical, Fluent, CFX, and Rocky DEM. UberCloud assisted FLSmidth engineers during the tests by setting up license servers, providing CLI and UI support, setting up RSM, troubleshooting computation errors, and resolving Azure resource issues. Benchmark tests were successfully run and results were reported.

Performance

ANSYS Fluent, CFX, and Mechanical benchmark tests were performed using H16 nodes in Azure which have the same CPUs as FLSmidth's on-premise HPC cluster. Single node runs brought very close results.

Multi-node runs showed good scalability when using many compute nodes for large simulations. Performance on Azure was 10-30% slower due to the difference of InfiniBand speed for interconnect (FLSmidth cluster has 56 Gb FDR vs H16's with 20 Gb FDR). The ability to scale up to large number of resources enabled FLSmidth to run multiple jobs simultaneously reducing multi-job duration and significantly increasing HPC throughput for engineers. Some of the benchmark results are shown in Figures 3 and 4, for FLSmidth's in-house Quasar cluster versus Azure (H16r and HC44 compute instances).

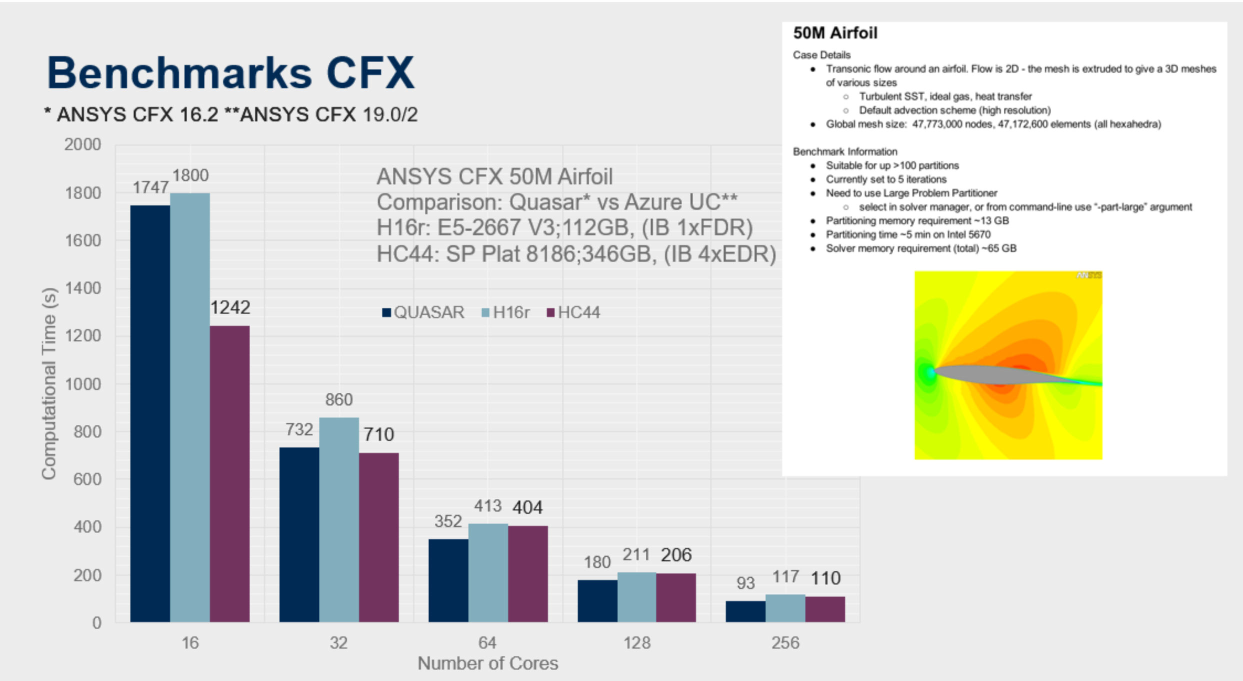


Figure 3: ANSYS CFX Benchmarks - External flow over an airfoil. approximately 48 million nodes (47 million hexahedral elements), solving compressible fluid flow with heat transfer using the SST turbulence model.

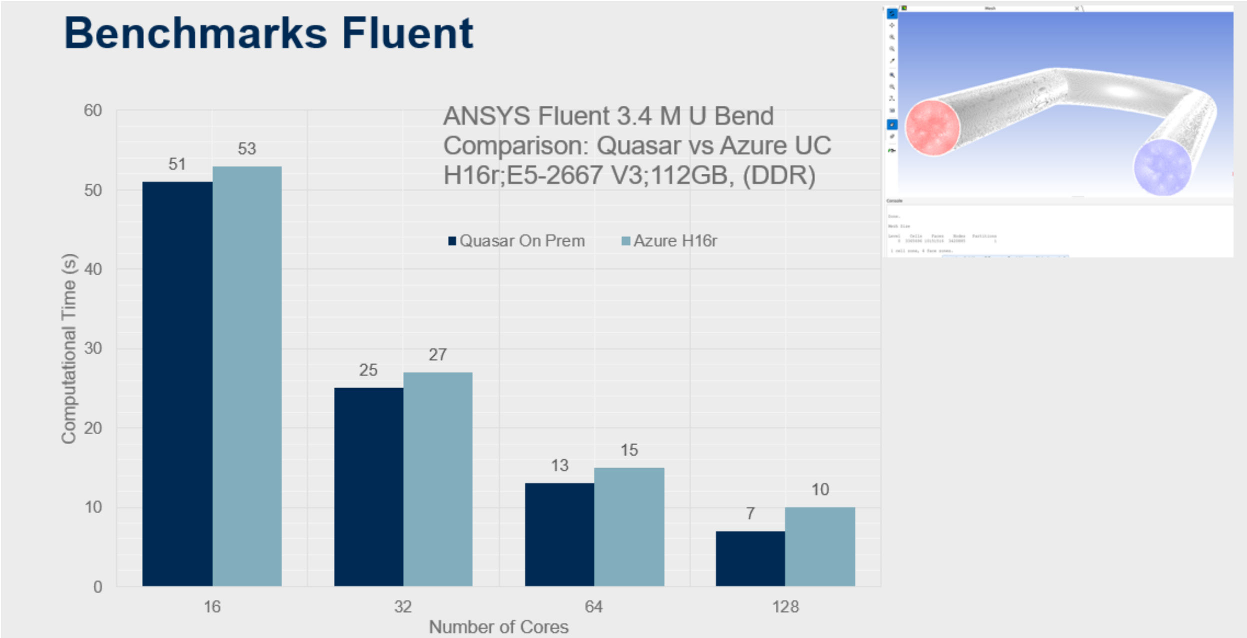


Figure 4: ANSYS Fluent Benchmarks – CFD analysis of flow through a U-Bend pipe.

CycleCloud and Univa Grid Engine (UGE)

Microsoft CycleCloud offers an easy to use, web-based cluster provisioning and management interface for engineers to create their own cloud computing resources on-demand. CycleCloud's policy-based templates allow administrators to set up the rules and usage limitations to control HPC utilization of different groups.

Univa Grid Engine provides a powerful multi-cloud workload management platform. Combined with CycleCloud, on-demand clusters can be created with built-in workload management capabilities. This combination delivers self-service HPC capability to engineers to create and control their own HPC resources which should reduce IT dependency of HPC resources.

CONCLUSION AND NEXT STEPS

During this project, FLSmidth's engineering applications (ANSYS Mechanical, Fluent, CFX, and Rocky DEM) were implemented in Azure HPC and tested in West Europe and Central India datacenters. FLSmidth's engineering and IT workflows were analyzed to provide the best cloud environment for the engineering processes which also reduces the IT management overheads. Cloud HPC will allow FLSmidth engineers to use simulation applications and vast resources on demand allowing faster results and turnaround time while providing complete control and management to IT. Moving the complete engineering simulations to Azure will provide scalable infrastructure and by keeping data in Azure's secure data centers will eliminate any delays of data transfer to local systems. Some of the outcomes are summarized below:

- Azure provides a variety of HPC compute resources for different application needs of FLSmidth engineers. New compute resources are regularly added, providing faster and cheaper resources for FLSmidth.
- Azure HPC resources are available in many datacenters around the world, providing low-latency regional access for FLSmidth sites.
- UberCloud provides complete engineering workflow capability in the cloud including full graphical user interface for pre- and post-processing, and batch.
- UberCloud HPC containers provide portability of FLSmidth's engineering applications, and allow multi-location and multi-cloud usability to application workloads.

- UberCloud HPC containers allow engineers to collaborate and increase productivity by working on the same simulation from any location.
- UberCloud HPC containers provide a standard HPC Application Management for all HPC applications including the ability to containerize other simulation applications including in-house codes.
- Cloud access allows worldwide usage of FLSmidth's applications from all sites.
- Cloud HPC providing almost unlimited number of resources enables FLSmidth engineers the freedom to innovate. They can use the resources to speed-up their simulations, run multiple iterations, models, physics, perform parametric sweeps and Design of Experiments (DoE).
- Cloud HPC allows global access to software licenses, reducing the number of local licenses or using them more efficiently (e.g. by more engineers).
- Cloud HPC eliminates the hardware bottlenecks during peak usage and fluctuating project loads.

Microsoft Azure with UberCloud CAE containers provides the modern Cloud HPC environment for FLSmidth. During this project, FLSmidth's IT and engineering workflows were evaluated and the following operational tasks were identified as critical processes to be developed before enterprise roll-out:

- **Engineering Workflow and Usage Optimizations:** Cloud HPC usage requires better planning of the simulations, results and data storage, job-based vs interactive usage and processing. UberCloud and FLSmidth teams started creating the list of "Engineering Best Practices in Cloud for FLSmidth Engineers" and documented in "FLSmidth Azure HPC Operations and Enterprise Roll-Out Map".
- **Consumption Monitoring and Billing:** User/Group based consumption should be monitored to ensure proper use of cloud resources and minimize idle resources.
- **Simulations generate large amount of data**. FLSmidth HPC Architecture is designed with regional storage and also replication for data that needs global access. Storage, archival and cleanup policy must be developed in alignment with FLSmidth Data Center Migration project.

As the result of this project, the HPC Cloud architecture was built and tested successfully. An Enterprise Roll-Out map was created identifying the steps necessary to establish steady-state operation. The next phase will focus on the Enterprise Roll-Out tasks and help FLSmidth complete its cloud migration of the HPC workloads.

Case Study Authors – Sam Zakrzewski and Reha Senturk

GERRIS Flow Solver

Unsteady Flow Around Aircraft Landing Gear

"…among the lessons learned was that it is essential that the resource provider have significant expertise both in CAE and *HPC."*

MEET THE TEAM

End User - Patrick Bourdin, Bombardier

Software Provider - Gerris Flow Solver (open source free software)

Resource Provider - Joris Poort and Sunny Manivannan, Rescale

Experts - Mulyanto Poort, Rescale, and Amish Thaker, Thaker Sim Tech LLC.

USE CASE

Team 60 focused on setting up and successfully running a simulation of unsteady airflow around aircraft landing gear. Rescale's web browser-based simulation platform was used to set up and submit the job, and Rescale also provisioned the required computing capacity directly from its hardware partners. The software of choice was Gerris Flow Solver, an open-source program for solving partial differential equations describing fluid flow.

Process Overview

1. The end user described the problem, including expected computing requirements and software tools to be used. Rescale agreed to provide up to 1,000 core-hours free of charge for this particular job. Rescale also agreed to perform the development work necessary to display real-time images showing Gerris results (such as mesh, velocity, and vorticity plots) directly on its web-based platform.
2. Rescale procured Gerris, integrated it into our web-based platform, and worked with the end user to securely transfer the input files for a test run on a smaller problem with similar physics (i.e., aircraft landing gear flow analysis with a coarser mesh).
3. Upon successful completion of the test run, including real-time image visualization directly on Rescale's platform, Rescale worked with the end user to securely transfer input files for the final run and set up the actual run.
4. On the actual run, the initial configuration led to an analysis that resulted in Rescale exceeding the 1,000 core-hour limit by many orders of magnitude. The end user and Rescale then iterated several times until we reached a configuration that would result in the final analysis converging within the 1,000-core-hour limit.
5. Rescale securely transferred files back to end user, transferring output files up to 60 GB in size.

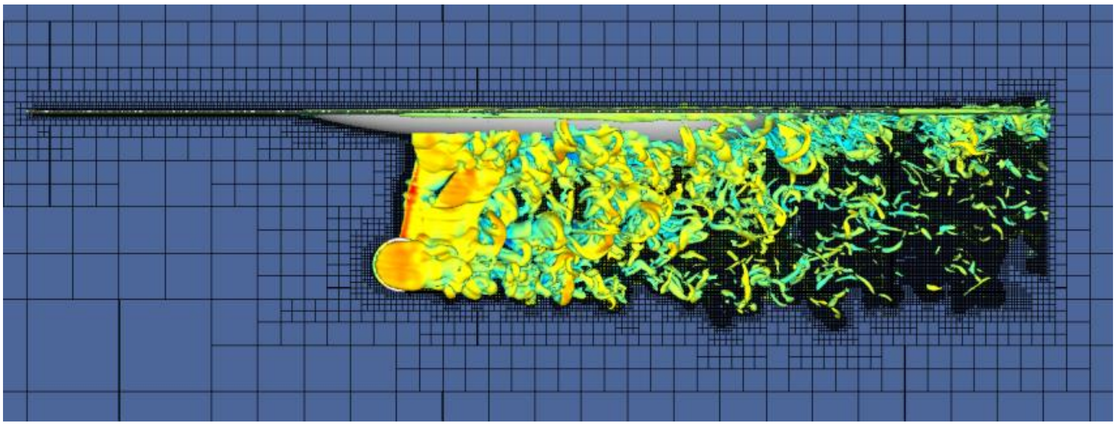


Figure 1: Vorticity plot taken at one of the intermediate time steps of the analysis.

CHALLENGES

Rescale's perspective – The main challenge was that the actual analysis took much longer than expected to converge, even on the best-in-class HPC used by Rescale. The analysis was new to everyone, and it took a few tries for both the end user and Rescale to understand how adjusting different variables affected the expected time to convergence. The other challenge was transfer of large files. For files as large as 60 GB, this took a few hours, even with Rescale's proprietary file transfer capabilities. This is why we give users the option to display selected results directly on our web platform. If the analysis had converged within the 1,000 core-hour limit, we would have had some time to explore this option, which would have saved the user a tremendous amount of time.

End User perspective - We wanted to capture too much physics at once with the original model, which led to excessive memory requirements and forced us to downgrade the physical modeling in order to get something running in the scope of the experiment. That was a challenge, but from a project point of view only. From a resource provider/end-user interaction point of view, the challenges and barriers were quite minor. Due to the end-user's company IT policy, there are serious limitations when it comes to downloading and uploading data from and to external servers, but this limitations could be removed in the frame of a formal business relationship. In my opinion, the main challenge (not addressed during the experiment) will come from legal aspects, like exchanging proprietary data or in-house codes.

BENEFITS

From Rescale's perspective, the main benefit to the end user was that the disparate aspects of running a successful industrial-grade simulation, namely

- access to simulation software codes;
- use of high-performance computing (HPC) hardware;
- and access to engineering and CAE expertise,

were all provided directly by Rescale and accessible through any web browser. In addition, Rescale's team leveraged its domain expertise in both engineering and high-performance computing in order to resolve customer demands promptly.

From the end user perspective, Rescale was very responsive when it came to installing and getting Gerris running on their system. The web interface is intuitive and offers enough commodities to monitor complex CFD jobs on the fly (via log file tailing and displaying flow visualization snapshots dumped by the software). The ability to download check-pointed solution files while the job keeps running would be a bonus to an already good interface.

CONCLUSIONS AND RECOMMENDATIONS

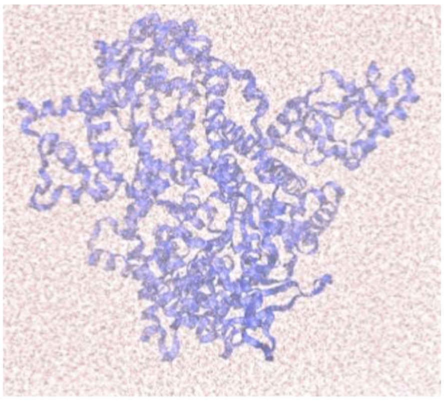
While not always possible, it is important to start the HPC Experiment with a hypothesis on how long the analysis will take to converge. In this case, based on the successful test run, we added too much complexity to the final analysis, resulting in many iterations and time spent trying to make it small enough to converge within the previously established resource limits.

From the resource provider's perspective, among the lessons learned was that it is essential that the resource provider have significant expertise both in CAE *and* HPC. The Rescale team was composed of mechanical engineers and computer scientists with deep domain expertise in their areas. This expertise helped us work with the end user to troubleshoot rapidly and get to results faster for any type of engineering analysis.

Lessons learned by the end user included the fact that running CFD using HPC in the cloud proved to be a viable approach from an engineer's standpoint. However, the lawyer's standpoint (intellectual property questions) and the accountant's standpoint (cost-benefit analysis of using the cloud vs. buying HPC hardware internally) will need to be addressed.

Case Study Authors – Patrick Bourdin and Sunny Manivannan

GROMACS
Molecular Dynamics of the Mutant PI3Kα Protein



"Cloud computing proved to be an extremely easy process as compared to building and maintaining your own cluster."

MEET THE TEAM

End User - Zoe Cournia, Biomedical Research Foundation of the Academy of Athens
Software Provider - GROMACS open source and Al Wegener, Samplify (APAX compression software)
Resource Provider - GRNET-Okeanos IaaS Cloud service: Vangelis Floros, Stefanos Gerangelos, Greek Research and Technology Network S.A.
HPC Expert - Dominique Dehareng, Center for Protein Engineering, University of Liège.

USE CASE

Cancer is a leading cause of death worldwide, accounting for 7.6 million deaths in 2008 according to the World Health Organization. This number is expected to increase to 21 million by 2030. One of the signaling pathways which, when deregulated, becomes centrally involved in several types of cancers, like colon, mammary, and endometrial tumorigenesis, is served by phosphoinositide-3-kinase alpha (PI3Kα). The importance of PI3Kα in cancer is highlighted by the discovery that PIK3CA, the gene encoding the catalytic p110α subunit of PI3Kα, is frequently mutated in human malignancies.

The goal of our project is to gain insights into the oncogenic mechanism of two commonly expressed PI3Kα mutants by studying their conformational changes with Molecular Dynamics (MD) simulations, in comparison with the PI3Kα wild-type (normal, non-cancerous) protein. The utility of cloud computing in performing MD simulations of mutant PI3Kα with Gromacs was examined in this case study.

Gromacs is a versatile package used to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions. GROMACS achieves both extremely high performance on single processors from algorithmic optimizations and hand-coded routines and simultaneously scales very well on parallel machines, http://www.gromacs.org/About_Gromacs.

The cloud computing service provider, GRNET, provided a virtual machine with 8 cores and Gromacs installation. Linear scaling was observed up to the 8 available cores. The process of accessing the cloud computing service, transferring data, running the simulation, and analyzing the results was very effective, but the study was limited by the resources.

~okeanos (http://okeanos.grnet.gr) is an IaaS public cloud that provides a complete range of cloud services such as compute, storage, network, accounting, billing and security. ~okeanos is build on open-source software developed in-house by GRNET, the Greek NREN, with the primary goal of providing public cloud services to the Greek academic and research community. Access to the services is provided through an intuitive, user friendly, web interface, as well as from command line tools.

Programmatically, the ~okeanos software stack offers a set of well documented proprietary REST APIs as well as standard APIs such as OpenStack Compute (Nova) and OpenStack Object Storage (swift compliant). The software is modular, comprising a number of different components that can be independently deployed and exploited. Currently the service is provided on trial basis (public beta) but is expected to be in full production early in 2014. However, ~okeanos is already heavily used by more than 1,700 users who have instantiated more than 100K VMs, validating the stability and efficiency of the architecture and the underlying technology of the service.

Technical Details of the Simulation

The mutant protein E545K PI3Kα was simulated. AMBER99SB-ILDN force field was used to model all protein interactions, and the TIP3P model was used for water. The simulations were performed at constant pressure, temperature, and number of particles (NPT ensemble). The temperature of the system was controlled with the Nose-Hoover thermostat. The pressure of the system was isotropically coupled and maintained at 1.01325 bar using the Parinello-Raman algorithm. Periodic boundary conditions were used. Electrostatic interactions were calculated with the PME method using a 1.4 nm cut-off for the direct space sum. For the van der Waals interactions, the Lennard-Jones potential was smoothly switched off from 0.8 to 1.0 nm. All bonds were constrained using the LINCS algorithm. The simulations were run using a 2-fs integration time step and a total number of 10,000 timesteps, which has been shown to be adequate for benchmark studies. The output coordinates, velocities and energies were saved every 100 steps. The system contained roughly 280,000 atoms (Figure 1). Gromacs 4.6.1 with double precision was used for the simulations.

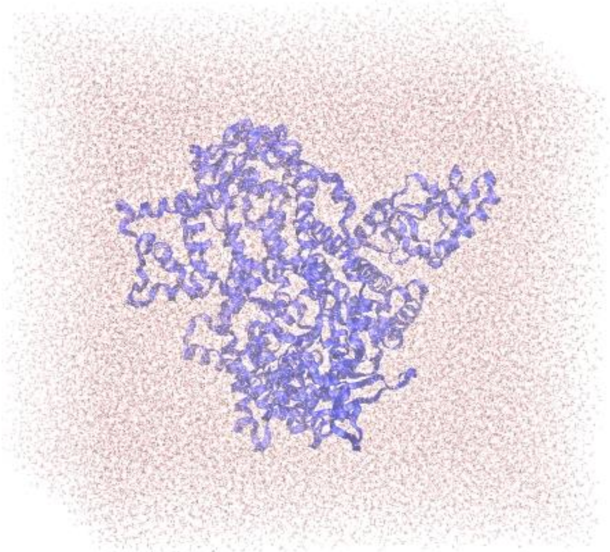


Figure 1: The protein PI3Ka is depicted in ribbons and is placed in a water box, shown as red dots.

Process Overview

1. Definition of the project.
2. Kickoff meeting of the experiment using Skype.
3. The resource providers (~okeanos, https://okeanos.grnet.gr) Vangelis Floros and Stefanos Gerangelos, creates the VM environments with the GROMACS installation.
4. The end user, Zoe Cournia, built a VM machine and transferred the data to the cloud computing service.
5. The end user performed an MD scaling study on 1, 2, 4, and 8 cores.
6. The end user analyzed performance data and communicated the results to the rest of the team.
7. The resource provider setup a cluster consisting of several virtual machines in order to perform the same scaling study on a cluster interconnecting with Ethernet.
8. The APAX compression was handled by the GROMACS team in order to compress the data on a cluster interconnecting with Ethernet.
9. Al Wegener, the software provider, Zoe Cournia, the end user, and Wolfgang Gentzsch, the team mentor, contacted the GROMACS team. A GROMACS source code expert was sought to implement the APAX calls in the code.
10. The team mentor and team expert supervised the experiment very closely and provided useful suggestions and organizational help via BaseCamp.

Scaling Study Results

Data was transferred from the end user's in-house resources to the cloud. The rate of transfer was quite efficient at 1MB/s. Simulations were performed on 1, 2, 4, and 8 cores of the same Virtual Machine (VM) provided by GRNET running GROMACS software. (Output/input files, etc., are available upon request.)

Our results show linear scaling from one up to eight cores on the virtual machine (Figure 2). The absolute timings and speedup of the study are shown in Table 1. The (partially) superscalar speedup may be attributed to the cache/memory architecture of the AMD Opteron 6172 processor.

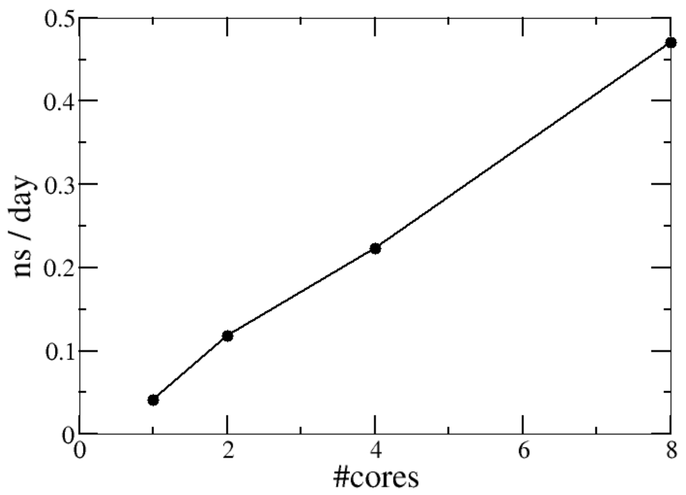


Figure 2: GROMACS performance on the GRNET node.

#cores	Absolute timings (s)	Speedup

1	42051	1.00
2	14652	2.87
4	7432	5.65
8	3671	11.4

Table 1: Results from the Scaling study performed on the GRNET VM.

The underlying hardware where ~okeanos was running was made up of a large number of HP Proliant DL385 G7 nodes. Each node was configured with 2xAMD Opteron 6172, 12 core CPUs (24 cores per node), clocked at 2.1 GHz , 192 GB DDR3 1333MHz main memory, and 8x600GB SAS disks. The local interconnect fabric used is based on 1 Gb Ethernet.

Previously, we performed similar studies on our in-house blade system with 12 cores where the performance was 1/3 better in-house, expectedly.

In another effort, in the past, we have done scaling studies on up to 2048 cores on the CURIE and JUGENE HPC systems for Molecular Dynamics simulations of PI3Ka protein. As a result, a white paper is published online at http://www.drugdesign.gr/white-papers.html.

CHALLENGES

End user perspective - While the cloud computing service was very efficient, the number of cores provided (up to 8) is too limited to simulate large biological systems. With 8 cores we could simulate about 0.5ns /day for the 280,000 atoms of the protein system. For our calculation to be efficient in terms of computing time, we would need at least 64 cores connected with Infiniband in order to reach about 4ns/day in performance. However, this cloud computing service provider only provides Ethernet connections for the 64-core cluster configured as 8 8-core nodes.

Therefore, the challenge was to compress the produced data on the fly so that the bandwidth of the connection is less limiting. For this purpose, we propose a Round 4 HPC-Experiment, where we will work with Samplify to include APAX compression calls within the Gromacs software in order to compress the data that is being communicated by the Ethernet network.

This part of the project was deemed too demanding for the Round 3 HPC-Experiment since the GROMACS team could not contribute at this time. We are still faced with the challenge of attracting a collaborator to implement the APAX technology within Gromacs. Moreover, it was challenging to build a cluster from VM nodes from the service provider. The end user needed to build the cluster under the direction of the service provider, but this was not possible because of OS mismatches. However, the cluster will be built in the next HPC Experiment Round.

Resource provider perspective - ~okeanos has been designed as a general purpose cloud service that aims to satisfy a broad range of applications and use cases. Since the main target group of the service is comprised of research and academic institutes, HPC scientific applications are expected to evolve as one of the major workloads that the final production service should be able to host efficiently. The underlying virtualization software has proven in many cases (and also in this experiment) to be quite robust and easy to use, helping scientists focus on their research and avoid the hassle of managing the underlying infrastructure.
Nevertheless, there are some parts of the software toolkit that need improvement, such as the capability to effortlessly instantiate and manage virtual clusters. On the other hand, many of the high-performance demands of scientific applications can only be satisfied at the hardware level. For these kinds of applications, technologies like Infiniband networks, GPGPUs and accelerators are becoming very relevant. GRNET is taking this into account while designing the next phases of service

expansion especially regarding the underlying physical infrastructure. Seamless synergy of core virtualization software (hypervisors, VMMs, etc) and specialized hardware will be an important factor for the successful adoption of cloud computing in the HPC domain.

BENEFITS

End user perspective

- Gained an understanding of the cloud computing philosophy and of what is involved in using a cloud-based solution for computational work
- Cloud computing is an easy process compared to building and maintaining your own cluster.
- Learned that cloud computing services, which are free and very efficient, do exist.
- Determined that moving computational work to the cloud during periods of full-utilization of in-house compute resources is a viable approach to ensuring analysis throughput.

Resource provider perspective

- Gained important experience from supporting a real, high performance, demanding scientific application.
- Helped identify important gaps and missing functionality for providing optimal cloud platform for high-performance applications.
- Validated the intuitive design, efficiency and user friendliness of online end-user tools.

CONCLUSIONS AND RECOMMENDATIONS

The cloud computing service that was provided was a free and very efficient service. Cloud computing proved to be an extremely easy process as compared to building and maintaining your own cluster. We were provided with a VM with 8 cores and the possibility of building a cluster of up to 64 cores connected via Ethernet network. In these 8 cores, our Molecular Dynamics simulation of the mutant PI3Kα protein scaled linearly.

However, due to the large system we only achieved 0.5 ns /day. In order to simulate large biological systems, the end user needs to verify that the resources and connectivity network will be adequate for the simulation. The end user can choose to simulate smaller systems on cloud computing depending on the available resources and transfer computational work to the cloud during periods of full-utilization of in-house compute resources. We did not observe speed up of the cloud computing simulation, but it did run faster when compared to in-house simulations.

Team communications through BaseCamp were very efficient. Collaborative work over the Internet, using on-line resources like cloud computing hardware and open source software such as Gromacs, is an efficient alternative to in-person meetings and in-house calculation servers.

In the future, scaling tests using an Ethernet connection within a virtual machine cluster will be performed and compared to running on a single node or with an Infiniband network. A future goal is to use the APAX compression method by Samplify in order to evaluate whether the achieved compression can yield speedup benefits for MD simulations utilizing Ethernet connections.

Case Study Authors – Zoe Cournia, Al Wegener, Vangelis Floros, Stefanos Gerangelos, and Dominique Dehareng

HYDROTEC Hydro_AS-2d

Dam Break Simulation Using Hydro_AS-2d on a Remote HPC Cluster

"Even without MPI parallel communication, the speed advantage using a cluster instead of the in-house PC, is remarkable."

MEET THE TEAM

End User & Team Expert - Raimund Mollner, DI, Pöyry Energy GmbH, Vienna, Austria.
Software Provider - Dr. Nujic, HYDROTEC Ingenieur-Gesellschaft für Wasser und Abfall, Aachen Germany.
Hardware Provider - Vladimir Baros and Henrik Nordborg, HSR Hochschule für Technik, Rapperswil, Switzerland.

USE CASE

A 125km long river section, including several existing and construction phase dams, was hydraulically simulated using the software Hydro_AS-2d. Hydro_as-2d is a CFD-software solving the shallow water equation with depth averaged velocities. The uppermost dam is supposed to break.

The calculation speed using the HSR Cluster was compared to the calculation speed using the Pöyry in-house computer.

Simulation on the Pöyry In-house PC

The highly transient simulation C1 was run on the Pöyry in-house PC, which has Windows Server 2008 64bit OS, 6 cores i7 @ 3,33 Ghz and 24GB RAM. The run time for this simulation was 5:03 hours.

Simulation on the Cluster

The same simulation input file C1 was transferred to the cluster in Switzerland via file transfer (via Remote Desktop application which is more reliable and much more convenient for the end user than FTP). Five simulations were carried out using the HPC Cluster on the HSR Hochschule für Technik, Rapperswil, Switzerland. The hydraulic simulation results were identical with the results from the Pöyry-PC.

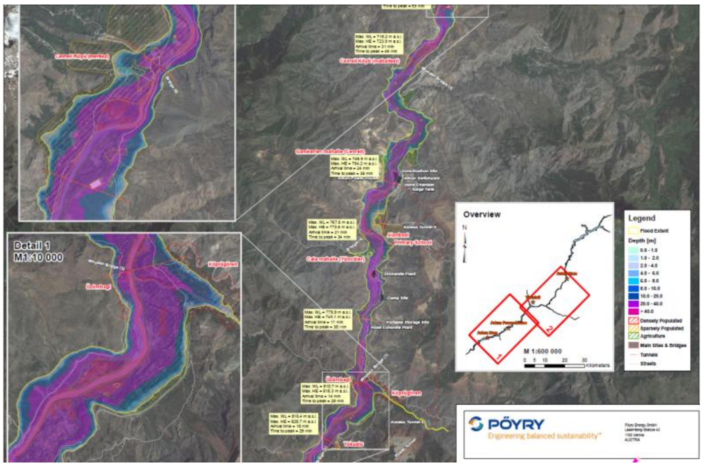


Figure 6: Inundation maps of the different simulation runs are identical.

For the first simulation, only one core was used. This run lasted for 12:57 hours. For further simulations the number of cores was raised to 2, 4, 6 and finally 12 cores. To use more than 12 cores was not possible, due to software limitations. The final run using 12 cores needed 3:17 hours. An overview of the simulation speeds is given in Figure 2.

Figure 7: Run times of the cluster simulations compared with the PC simulation run.

CHALLENGES

End User

- Licensing (using a so-called software protection dongle with encrypted license key)
- Installing the software dongle on the cluster
- Establishing the remote connection to the cluster

Resource Provider

- The software was designed to be used with human interaction, which means that the enter key should be pressed at the end of the simulation. This presents a problem for the cluster software, which presumes that the application has finished execution when it exits. This never happens and the cluster job stays in the state of execution forever. The solution is to define the standard input containing the enter key command. This way the job will exit after it is done.
- Application communication with the license server (the license server is not usually installed on the cluster machine)
- Properly configuring the .ini file which should be placed along with the main application .exe file. The .ini file contains the license server name and communication port along with other data

- The software doesn’t use MPI, so communication between compute nodes is not possible to further speed up the simulation. On the other hand, the end user was able to run many different jobs on the cluster at the same time, which cut the execution time significantly. Every job could run on one machine which, in the case of 32 compute nodes, would mean that 32 jobs could all finish in 3 hours, 17 minutes instead of 4 days and 9 hours.

Software Provider
- The software is protected by a hardware lock HWL (secure software protection dongle). In this case a net-use hardware lock was necessary. The activation is carried out by Dr. Nujic, the author of HYDRO_AS-2D. In this case the activation allowed 50 runs.

BENEFITS

End User
- Getting to know the whole HPC-Process, both locally and remotely.
- Better understanding the capabilities and limitations of the hydraulic software used.

Software Provider
- To know that the software works for an end-user who is using remote HPC systems, including cloud services.

Resource Provider
- Configuration of the remote access for external access (from the Internet to the internal HSR network)
- Making sure running jobs on the cluster and data transfer goes smoothly for the end user

CONCLUSIONS AND RECOMMENDATIONS

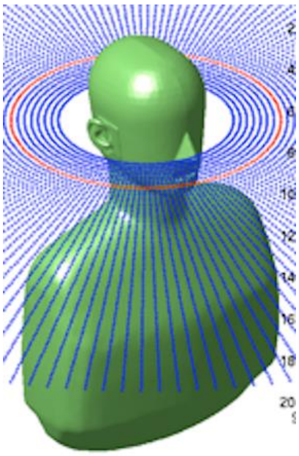
End User
Hydro_AS-2d is able to use more than one core for simulations. Hydro_AS-2d is restricted to using a maximum of one node.
Even without MPI parallel communication, the speed advantage using a cluster instead of the in-house PC , is remarkable. As one can see in Figure 7 the run with 12 cores on the cluster needed 1:45 hours less than the run with 6 cores on the in-house PC. This means, that the cluster simulation ran about 35% faster than the in-house-PC run.
Implementing the MPI technology into Hydro_AS-2d could dramatically improve the simulation speed of these hydraulic 2D-simulations, and would make cluster computing really interesting for Pöyry.

Software Provider
It would be interesting to know how many points (percentage) were wetted during the simulation. Our experience: the more points are wet, the greater the time saving through parallel usage!
The benefit, especially for our customers, lies more in the parallel use of several simulation cases.

Case Study Author - Raimund Mollner

KUAVA Waveller

Simulation of Spatial Hearing



"The main lessons learned in this cloud project were related to using CPU-optimization when compiling the code for cloud simulations."

MEET THE TEAM

End User - Manufacturer of consumer products. The end-user tasks were related to the planning of the simulations and post-processing the simulated data.
Software Provider and HPC Expert – Antti Vanne, Kimmo Tuppurainen, Tomi Huttunen, Kuava Ltd.
HPC Experts – Ville Pulkki, Marko Hiipakka, Aalto University.

USE CASE

A sound emitted by an audio device is perceived by the user of the device. The human perception of sound is, however, a personal experience. For example, the spatial hearing (the capability to distinguish the direction of sound) depends on the individual shape of the torso, head and pinna (i.e. so-called head-related transfer function, HRTF).

To produce directional sounds via headphones, one needs to use HRTF filters that "model" sound propagation in the vicinity of the ear. These filters can be generated using computer simulations, but, to date, the computational challenges of simulating the HRTFs have been enormous due to: the need of a detailed geometry of head and torso; the large number of frequency steps needed to cover the audible frequency range; and the need of a dense set of observation points to cover the full 3D space surrounding the listener. In this project, we investigated the fast generation of HRTFs using simulations in the cloud. The simulation method relied on an extremely fast boundary element solver, which is scalable to a large number of CPUs.

The process for developing filters for 3D audio is long but the simulation work of this study constitutes a crucial part of the development chain. In the first phase, a sufficient number of the 3D head-and-torso geometries needed to be generated. A laser-scanned geometry of a commercially available test dummy was used in these simulations. Next, acoustic simulations to characterize acoustic field surrounding the head-and-torso were performed. This was our task in the HPC Experiment.

The simulations focused on the effect of the acoustic impedance of the test dummy on the HRTFs. Finally, the filters were generated from the simulated data and they will be evaluated by a listening test. The final part was done by the end-user.

Simulations were run via Kuava's Waveller Cloud simulation tool using the system described below. The number of concurrent instances ranged between 6 and 20.

- Service: Amazon Elastic Compute Cloud
  Total CPU hour usage: 371h
  Type: High-CPU Extra Large Instance

  High-CPU Extra Large Instance:
  7 GiB of memory
  20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)
  1690 GB of instance storage

  64-bit platform, I/O Performance: High

One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This is equivalent to early-2006 1.7 GHz Xeon processors

CHALLENGES

Our main challenge was to develop interactive visualization tools for simulation data stored in the cloud.

BENEFITS

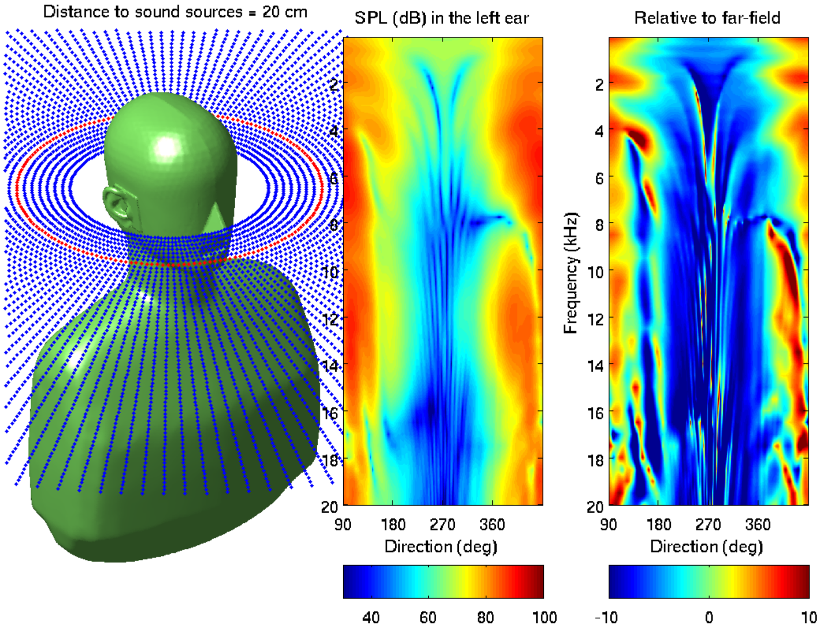
The main benefit resulted from the flexible resource allocation, which is necessary for efficient acoustic simulations. That is, a large number of instances can be obtained for a short period of time.

Other benefits included not having to invest in our own computing capacity. Especially in audio simulations, the capacity is needed in short bursts for fast simulation turnaround times and the time between the simulation bursts while the next simulation is planned – i.e., when no computational capacity is needed – is significant.

CONCLUSIONS AND RECOMMENDATIONS

The main lessons learned during Round 2 were related to using CPU-optimization when compiling the code for cloud simulations. We observed that Amazon did not support all optimization features even though the optimization should be available in the instances used for simulations. The problems were solved (with the kind help of Amazon support) by disabling some of the optimizations when compiling the code.

The man hours accumulated during the experiment included Kuava (50h), and end-user (5h). Total CPU hour usage during the experiment was 371h using High-CPU Extra Large Instance.

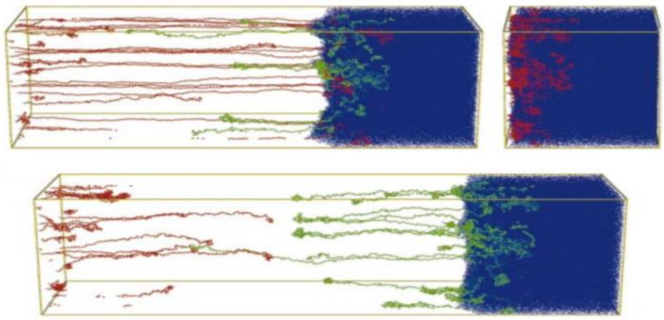


Simulation model (an acoustic test dummy). The dots indicate all locations of monopole sound sources that were used in the simulations. The red dots are the sound sources used in this image. The figure in the middle shows the sound pressure level (SPL) in the left ear as a function of the sound direction and the frequency. On the right, the SPL relative to sound sources in the far-field is shown.

Case Study Author – Tomi Huttunen, Kuava Ltd

LAMMPS

HPC Cloud Performance of Peptide Benchmark Using LAMMPS Molecular Dynamics Package



"HPC software container-based cloud computing is an easy process compared to building and maintaining your own cluster in the cloud."

Figure 1: Simulation snapshots using LAMMPS, studying adhesion dynamics for surface-tethered chains entangled in a polymer melt.

MEET THE TEAM

End User – National Renewable Energy Lab (NREL), Tech-X Research
Software Provider – LAMMPS open source software and Steven J. Plimpton (Sandia National Lab)
Resource Provider – Amazon Web Services (AWS)
HPC Experts – Dr. Scott W. Sides, Senior Scientist, Tech-X Research Boulder, CO, Fethican Coskuner and Ender Guler, The UberCloud.

USE CASE

In order to address realistic problems in the nanomaterials and pharmaceutical industries, large-scale molecular dynamics (MD) simulations must be able to fully utilize high-performance computing (HPC) resources. Many small- and medium-sized industries that could make use of MD simulations do not use HPC resources due to the complexity and expense of maintaining in-house computing clusters.

Cloud computing is an excellent a way of providing HPC resources to an underserved sector of the simulation market. In addition, providing HPC software containers with advanced application software can make the use of these codes more straightforward and further reduce the barriers for entry to small- and medium-sized businesses.

The molecular dynamics package LAMMPS is widely used in academia and some industries. LAMMPS has potentials for solid-state materials (metals, semiconductors) and soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale.

The cloud computing service provider, Amazon Web Services, provided a number of virtual machines each with up to 16 cores for this experiment with different levels of network communication performance.

Technical Details of the Simulation

Figure 2 shows the parallel scaling performance of LAMMPS containers running on an AWS multi-node cluster with each of the nodes having 16 cores available. A simple peptide chain model that is included in the tests for LAMMPS was used for performance scaling. The initial peptide input file only contains 2004 particles, but using the 'replicate' keyword available in LAMMPS the initial simulation cell may be copied in the x,y,z directions an arbitrary number of times.

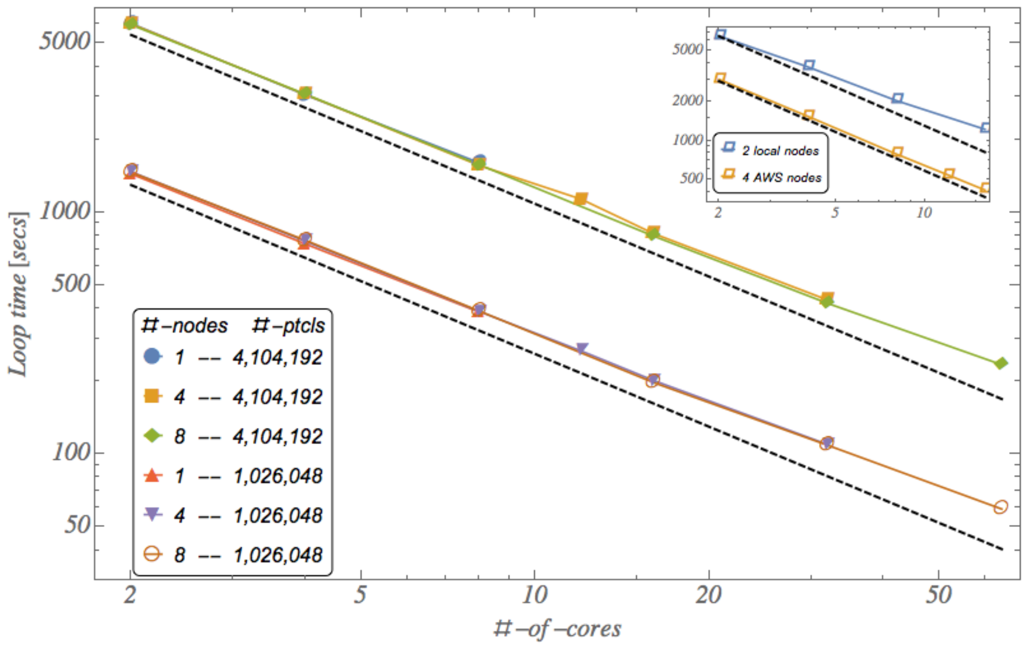


Figure 2: LAMMPS parallel scaling performance on an AWS multi-node cluster with each of the nodes having 16 cores available. Inset upper right: Comparison of the parallel scaling performance between LAMMPS running on the bare-metal 2-node test cluster at Tech-X and LAMMPS containers running on a 4-node remote AWS cluster. The dotted lines indicate the optimal scaling behavior, showing that the performance of the LAMMPS containers running in the cloud is excellent.

The simulations in Figure 2 show two system sizes using ≈ 10^6 and ≈ $4.1*10^6$ particles run for 300 update steps for reasonable timing statistics. The inset in the upper right shows a comparison of the parallel scaling performance for a system with ≈ $2.0*10^6$ particles between LAMMPS running on the bare-metal 2-node test cluster at Tech-X and LAMMPS containers running on a 4-node remote AWS cluster. The dotted line in the main figure and inset is the optimal scaling trend. The main figure shows that the LAMMPS multi-node container performance persists as the number of nodes in the cloud cluster increases. There was degraded performance when the number of processors/node reaches the maximum number of cores available as listed by AWS and is due to hyper-threading. But, there appears to be no degradation of performance as the size of the cluster increased, suggesting that an arbitrary number of processors can be used for HPC molecular dynamics simulations using LAMMPS in the cloud.

Summary of the SBIR project

This cloud experiment was initially funded as part of a Small Business Innovation Research (SBIR) grant. The solicitation called for enabling modern materials simulations in a larger sector of the industrial research community. High performance computing (HPC) is a technology that plays a key role in materials science, climate research, astrophysics, and many other endeavors. Numerical simulations can provide unique insight to physical phenomena that cannot be easily obtained by other means. Numerical simulations complement experimental observations, help in validating models, and advance our understanding of the world. Advances in HPC software development and algorithms are becoming increasingly important in materials science and for industries developing novel materials. According to a recent survey by the US Council on Competitiveness, faster time to market, return on investment, and enabling work that could not be performed by any other means are cited as the most common justifications for using HPC in industry. For instance, Goodyear was able to significantly reduce the time to bring new tires to market through a collaboration with Sandia National Laboratory by leveraging high performance clusters. The oil, aeronautic, and automobile industries are examples of big industries where HPC technologies have been leveraged for decades. The growing penetration of HPC into engineering fields has been fueled by the continued performance improvements of computer chips as well as the emergence of hardware accelerators such as general-purpose graphics processing units (GPUs) and the Intel Xeon Phi co-processor (also known as many integrated core architecture, or MIC).

However, one of most striking features of the US Council on Competitiveness survey, is how underrepresented are the companies that would be most likely to take advantage of soft materials simulations. The biosciences sector accounted for only 5.9% and the chemical engineering sector accounted for only 4.0% of respondents on their use of HPC resources. The Phase I SBIR proposal granted to Tech-X addresses this call and the two issues outlined above, by using an extensible object-oriented toolkit (STREAMM) for linking quantum chemistry (DFT) and classical molecular dynamics (MD) simulations and making this code suite available to take advantage of HPC cloud computing.

Process Overview

11. Kickoff team meeting of the experiment using WebEx.
12. Organization of project tasks, communication and planning through RedMine.
13. The end user, Scott Sides, obtained an AWS account and provided ssh-keys to UberCloud in order to setup a project specific security group that is used to configure the multi-node multi-container environment.
14. A specialized installer was created for LAMMPS and made available to the team.
15. The end user performed an MD scaling study on a 1-node, 4-node, and 8-node cluster.
16. The end user analyzed performance data and communicated the results to the rest of the team.

CHALLENGES

End user perspective - The cloud computing service at Amazon Web Services (AWS) provided high-quality compute nodes with efficient communication networks that enabled the good scaling seen in Figure 2. There is quite a bit of manual setup that needs to be performed by the end-user for AWS. For any cloud computing project, the first step is to create the remote compute instances. One must apply for an account at AWS and use the AWS web interface to navigate to the services for the Elastic Compute Generation 2 (EC2). The 'elastic' refers to the ability to expand or shrink the hardware usage for a particular task at a given time. Then the desired number, type and security settings for the EC2 instances must be selected. For a first-time setup, an ssh-key pair is generated and stored within the user's account information. The web interface instructs the user how to setup their local ssh configuration so that access to any remote AWS instance can be obtained. This procedure is straightforward but again, must currently be done manually. The security group must also be specified manually and is one that is configured by UberCloud in order for the networking modules to function. Now the separate instances must be assembled and configured into a multi-node cluster.

The next steps are to copy setup applications, scripts and configuration files needed to install Docker, pull all needed Docker images, and start the computational images with all of the appropriate network configuration settings. The remote copy requires the DNS addresses generated by the AWS instance startup outlined above and must currently be performed manually. Then one of the compute instances must be designated as the 'Master' node which has two main purposes: (i) to run the 'Consul' container which is part of the framework that manages the network setup for all of the cluster instances and (ii) to provide a remote entry access point for the cluster. When launching simulations on this remote cloud cluster a user executes an SSH login command using the public IP address for the master node (again obtained manually through the AWS web tool) and a password that is automatically generated within the secure container and emailed to the user. These security measures are all part of the networking image layer in the UberCloud simulation containers. However, once these steps are in place, then running on a cloud cluster is much the same as running on an HPC cluster at a university or national lab.

BENEFITS, End user perspective

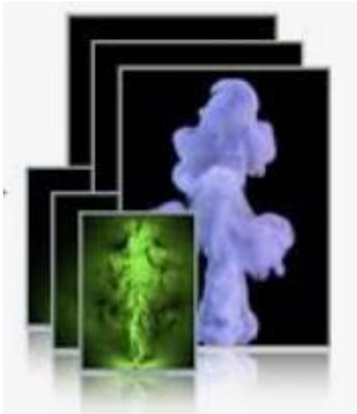
- Gained an understanding of the cloud computing philosophy and of what is involved in using a cloud-based solution for computational work.
- Cloud computing using novel HPC software containers based on Docker is an easy process compared to building and maintaining your own cluster and software environment.
- Developed an effective workflow for constructing additional HPC cloud containers.

CONCLUSIONS AND RECOMMENDATIONS

For the Phase II proposal based on this case study, Tech-X will add additional codes to the UberCloud marketplace for targeted industries and applications including those in nanotech and the pharmaceutical industries. We will also investigate ways to add functionality to our STREAMM framework to streamline the setup steps described in the ‘end-user perspective’ section. We will also check all our current scaling results on the Microsoft Azure cloud platform and compare with AWS and bare-metal. The Azure setup is reported to have ways of streamlining the setup process to make utilizing cloud HPC resources even easier.

Case Study Authors – Dr. Scott W Sides and Wolfgang Gentzsch

MantaFlow
Deep Learning in Computational Fluid Dynamics in the Microsoft Azure Cloud



***Courtesy, Thurey Group, TUM*.**

> *"Applying Deep Learning to highly complex compute intensive CFD by creating a database of synthetic flow descriptors that correlate with a given complex problem, so that the accurate solution could be inferred by a coarse-grid and much less compute intensive solution is one of the great benefits of this method."*

MEET THE TEAM

End-User: Joseph Pareti, Artificial Intelligence Consultant
Software Provider: Nils Thurey and Mengyu Chu, Technical University of Munich, Germany
Resource Provider: Microsoft Azure Cloud
UberCloud Support: Wolfgang Gentzsch, President, The UberCloud
Microsoft Support: Yassine Khelifi, Support Escalation Engineer at Microsoft.

INTRODUCTION: USING ARTIFICIAL INTELLIGENCE TO SPEED UP A CFD APPLICATION

In this work, Convolution Neural Networks (CNN) are used for computing feature descriptors for density and velocity fields in smoke clouds. The CNN learns from a repository of computed results using the MANTAFLOW application. The CNN training, using TensorFlow, determines flow descriptors for density and velocities and a flow similarity score. In addition, the model includes a deformation limiting patch advection with anticipation module which enhances the stability and performance.

When given a coarse simulation, the model associates with a high degree of confidence a more accurate simulation that is retrieved from a database of computed results. When compared with a traditional approach, the CNN-based approach provides much faster time to solution with comparable accuracy as when using a sufficiently fine grid.

There is an interest among CAE users to accelerate the time-to-solution of numerically intensive applications, in order to run a sufficient number of cases for product optimization using a parametric approach. Hence it is expected that more applications will become available that use deep networks to either replace compute intensive modules, such as FluidNet, or replace the entire computation as described in this report. Another reason to select this application for an UberCloud case study was the interest by the developer at TUM to contribute to our work in the form of Q&A, troubleshooting, etc.

Finally, we've made use of Microsoft Azure support for questions related to the configuration of the Data Science Virtual Machine (DSVM). If you are interested to run the application and you need more details than explained in this report, please send us a request at joepareti54@gmail.com or call us at +49 1520 1600 209.

USE CASE

Predicting smoke flows is a common task in computer graphics using CFD models. This can be done in 2D or 3D. Because an accurate calculation of flow properties is time consuming, researchers at the TUM developed new algorithms that benefit from deep learning technologies to dramatically reduce time-to-solution. While this paper is about a specific flow problem, with a specific solver (MANTAFLOW), it should be regarded as a feasibility study, and hence we encourage general purpose CFD users to look *beyond the box*, and come back to us with their specific CFD (or CAE) use case that could also benefit from deep learning or machine learning approaches.

At TUM, Nils Thurey and Mengyu Chu, in their work about Data-Driven Synthesis of Smoke Flows with CNN-based Feature Descriptors, take the following perspective to efficiently realize high-resolution flows (see Figure 1): they propose to use a fluid repository, consisting of a large collection of pre-computed space-time regions. From this, they synthesize new high-resolution volumes. In order to very efficiently find the best match from this repository, they propose to use novel, flow-aware feature descriptor. they ensure that L2 distances in this feature space will correspond to real matches of flow regions in terms of fluid density as well as fluid motion, so that they can very efficiently retrieve entries even for huge libraries of flow datasets.

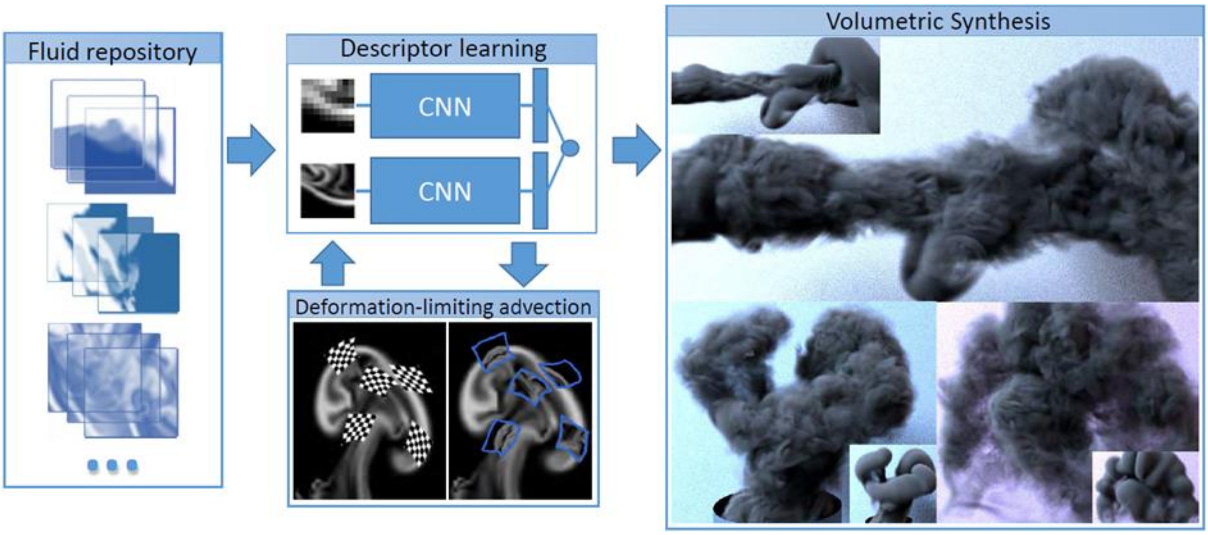


Figure 1: Realizing high-resolution flows efficiently by using a fluid repository, consisting of a large collection of pre-computed space-time regions.

SYSTEM ARCHITECTURE

In this application, we first ***built the MANTAFLOW application*** to calculate the exact data that will be used to train the deep network. There are 2D and 3D datasets. Next, we ***ran MANTAFLOW***. The output data of MANTAFLOW will be fed in ***TensorFlow*** to train the deep network. Once the network is trained we were able ***to make inferences*** and compare "exact" results (i.e. from MANTAFLOW with a fine grid) and results from the deep network. For the interested reader, all steps listed above are explained in detail in the Appendix. The paragraph on Results shows the exact and approximated output for the *smoke_tiled* example.

RESULTS

We achieved the following results:

1. An Azure-based demo version of the application that uses deep learning to significantly reduce time-to-solution in a CFD application.
2. A demo version that was employed for a specific example called the *smoke_tiled problem.*

In Figure 2 we compare the exact solution (right picture) with the approximate solution (left) for one sample:

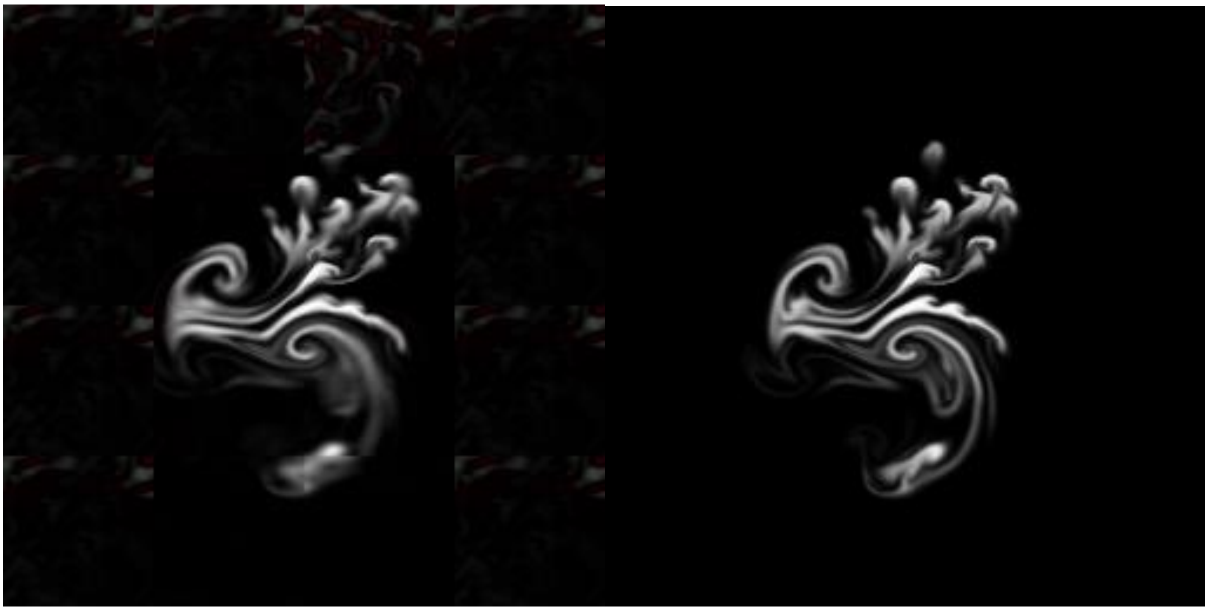


Figure 2: Smoke flow approximated by deep learning (left) vs. accurate MANTAFLOW calculation (right).

PERFORMANCE BENCHMARKING

This is out-of-scope for the current project. On our DSVM, the test cases presented in this report run in a few minutes; the exercise was just intended to validate the application and not to record any performance data.

BENEFITS

There are several R&D organizations working on deep learning tools to accelerate time-to-solution of CAE applications. In CFD, the deep learning approach varies:

1. One could aim at creating a database of synthetic flow descriptors that correlate with a given problem, so that the accurate solution could be inferred by a coarse grid solution (which is obviously less compute intensive). This is the approach taken by TUM in this report.
2. Another way to exploit the power of deep learning is by replacing parts of the computation by means of a deep network that simulates the exact behavior; this is the case of FluidNet where the non-linear Navier-Stokes partial differential equations are solved numerically, while the Poisson pressure correction term is simulated by a deep network instead of being calculated using traditional algorithms like sparse matrix solvers.
3. Or one could accurately calculate a number of cases, and then train the network using those results so that the network can predict the fluid flow based on geometry alone; this is the approach taken by Renumics in a recent UberCloud case study.[4]

[4] Team 211: Deep Learning for Steady-State Fluid Flow Prediction in the Advania Data Centers Cloud

If you are a CAE engineer, you may want to follow the methodology presented in this report and then possibly adapt it to your real problem, or work with us to design a deep learning-based solution.

CONCLUSION & RECOMMENDATIONS

The subject matter of this report is a demonstration how engineering simulations like computational fluid dynamics can benefit from Deep Learning with TensorFlow. Engineers are encouraged to approach us, get inspired for their work, and test this demo version “as-is”, to evaluate how their own environment can benefit from this or a similar approach.

As a next step for interested readers, we suggest to work together in a team (including the customer, Joseph Pareti, UberCloud, Microsoft, and Nvidia) to identify an internal application, scope the project, and create a Statement of Work (SOW): this will specify what algorithm can be used that takes advantage of Deep Learning / Machine Learning. The next step will be a Proof of Concept prior to production ready applications.

Case Study Author – Joseph Pareti and Wolfgang Gentzsch

MantiumFlow

Aerodynamic Simulations for the Silvermine 11SR Sportscar

> *"After logging into the Advania Data Centers cloud, running a CFD case created by MantiumFlow is just a matter of starting it. This makes an engineer's life very easy."*

MEET THE TEAM
End-User/CFD Expert: Andre Zimmer, Managing Director, MantiumCAE
Resource Provider: Jón Þór Kristinsson and Elizabeth Sargent, Advania Data Centers
Cloud Expert: Hilal Zitouni, Fetican Coskuner, Ender Guler, and Burak Yenier, The UberCloud.

ABOUT MANTIUMCAE
Based in Germany, MantiumCAE is an engineering consulting firm dedicated to computational fluid dynamics (CFD) simulations, with a particular focus on aerodynamics, optimization and CFD process automation. They assist manufacturing clients in establishing, enhancing, and optimizing their CFD capabilities and work to create products with greater aerodynamic performance.

As a specialized computer-aided engineering (CAE) consultant, MantiumCAE experiences both large and fluctuating computational demands to work on challenging projects. While browsing for on-demand High Performance Computing (HPC) providers on Cloud 28+, MantiumCAE discovered Advania Data Centers (ADC) and learned about their HPCFLOW service. MantiumCAE reached out to ADC's HPC experts and consulted with them, and subsequently determined that the best approach was to execute a hybrid approach to cloud-based HPC. This allowed them to combine their existing in-house HPC infrastructure with on-demand HPC resources from ADC. The result is a flexible approach which allowed MantiumCAE to make the most out of its existing HPC investments while increasing its ability to scale up HPC resources quickly and efficiently for its customers.

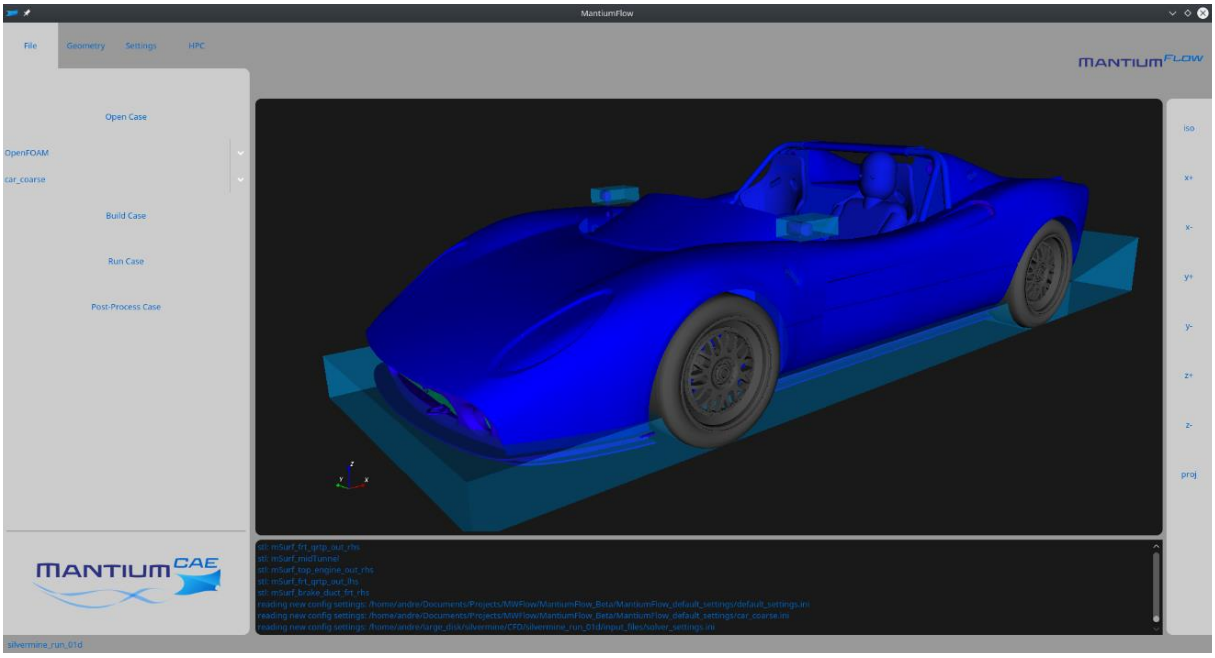
ABOUT ADVANIA DATA CENTERS
Advania Data Centers is a high-density computing technology company headquartered in Reykjavik, Iceland with operations in Sweden, Norway, Germany and the United Kingdom. Through extreme growth, Advania Data Centers now operate one of Europe's largest datacenter campuses in Iceland that is tailor made for high density hosting such as HPC, blockchain technology and high-density compute, all powered by renewable energy. Advania's HPC team consists of experts that oversee the operation of HPC environments and HPC Jobs of their customers, globally leading organizations in manufacturing, technology, science among other industries. Advania partners with industry leaders in HPC such as Hewlett Packard Enterprise, Intel, Nvidia, and UberCloud to deliver next generation

HPC environments such as HPCFLOW – Advania's Bare Metal HPC Cloud – where HPC operators can execute simulations in a fast and efficient manner.

USE CASE

This case study shows how ADC's HPCFLOW computing resources allowed MantiumCAE to create a CFD simulation quickly and efficiently for the Silvermine 11SR sportscar. To achieve this, MantiumCAE set up a CAE computing environment in the Advania's HPCFLOW cloud where simulations could be carried out quickly and efficiently.

A typical external vehicle aerodynamics simulation needs between 2.000 and 10.000 CPU core hours to be processed. Processing this simulation would take weeks to run on a 16-core workstation, but by using the HPCFLOW cloud environment together with MantiumFlow, MantiumCAE is able to deliver results within one business day.

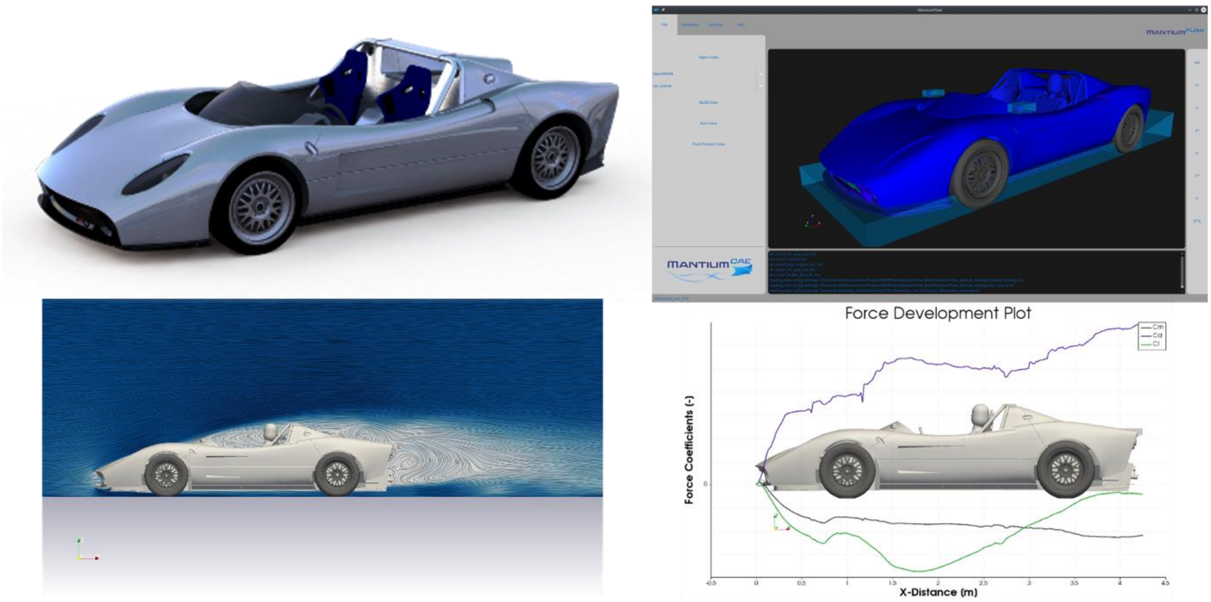


METHOD

In order to successfully create and carry out the CFD simulations for the Silvermine 11SR, MantiumCAE needed the following:

- CFD Engineer with a workstation
- MantiumFlow for the CFD setup
- HPC computing power from ADC
- MantiumFlow for post-processing

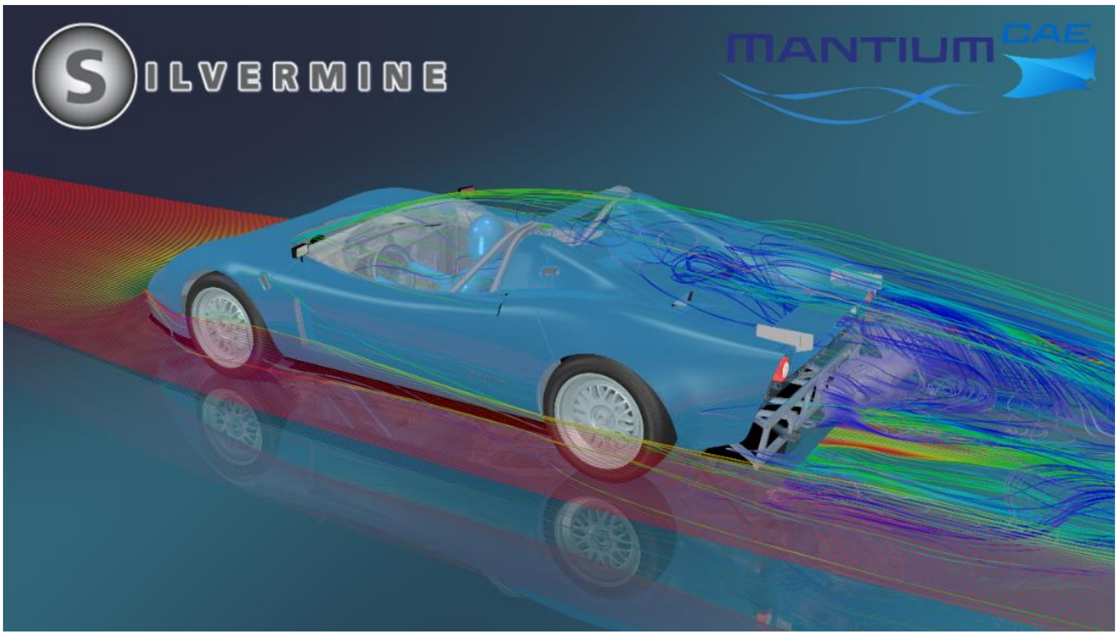
The process of running CFD simulations using HPCFLOW is straight forward. First, the engineer creates the CFD case using MantiumFlow, which automates the setup process and uploads it to ADC's HPCFLOW. The engineer then runs the CFD simulations with a script created by MantiumFlow on the ADC environment.



Afterwards a report containing a series of plots and images is automatically created by MantiumFlow. The almost fully automated approach minimizes user error and ensures that simulations can be repeated. Everything is executed using a desktop-like environment which is easy to use and navigate.

BUSINESS BENEFITS AND NEXT STEPS

By successfully using ADC's HPCFLOW technology, MantiumCAE was able to execute HPC CAE projects on a scale that was previously unattainable, and with a flexibility that allowed them to serve their clients' needs better and faster. This was done without any upfront investment in computers or facilities. MantiumCAE benefitted greatly from the flexibility of the HPCFLOW service, which allowed it to scale its use of HPC resources up and down to meet its changing demands and pay only for what was needed. ADC's HPC nodes proved to be well-suited to CFD, with 8GB of RAM per Intel Xeon E5-2683 v4 core with a total of 256GB of RAM and 32 cores, and we were able to process workloads quickly and efficiently.



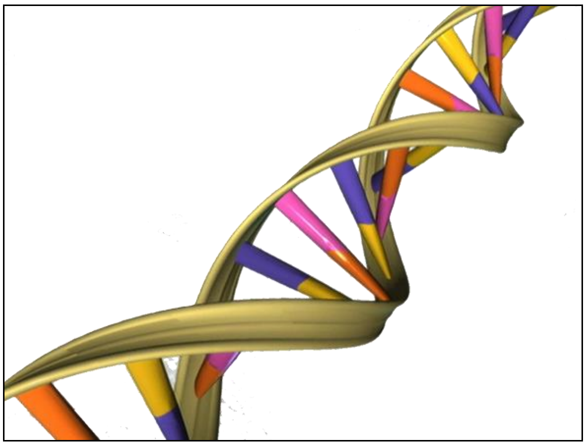
By giving MantiumCAE access to a dedicated HPC engineer for technical support throughout the project process, ADC ensured that there was always someone available to answer questions or troubleshoot problems. They listened to MantiumCAE's needs and provided an excellent level of service and support. This, combined with ADC's low cost per hour, made the experience very positive.

As a result of its work with Advania Data Centers, MantiumCAE has greatly strengthened its ability to be more competitive for challenging projects, without high initial investments and high cost of on-demand resources. This has secured their existing business, opened new markets and positioned them well for future growth.

Case Study Authors – Andre Zimmer and Elizabeth Sargent

NGS SeqWare

Next Generation Sequencing Data Analysis



"A good working relationship with our internal IT group, including Networking and Information Security, as well as familiarity with their processes, was essential for the success of this project."

MEET THE TEAM

End User - A medical devices company with over 20,000 employees and products in diagnostics, medical and biosciences markets.

Software Provider - Brian O'Connor, CEO, Nimbus Informatics, which provides consulting and cloud execution services for workflows that utilize the open source SeqWare application.

Resource Provider - Amazon Web Services (AWS).

HPC/CAE Expert – Ted Pendleton, Carl Chesal, Ian Alderman, James Cuff, Rob Futrick, Cycle Computing.

USE CASE

In this experiment, we explored using cloud computing for next-generation sequencing data analysis. We used SeqWare, an open source framework, running on AWS to perform variant calling. Variant calling is the reporting of differences between input sample DNA and a reference genome, using targeted exome data generated in-house on Ion Torrent's Personal Genome Machine.

Next-generation sequencing (NGS) is a relatively new high-throughput sequencing technology revolutionizing medicine. Its accuracy and breadth of coverage makes it useful in diagnosing and treating hereditary diseases and cancer, as well as infectious disease. However, the advantages of NGS lead to corollary challenges: storing, sharing, and analyzing the data strains disk space, network bandwidth and processor resources.

Cloud computing, though no panacea, can alleviate some of these problems by flexibly providing virtualized compute and storage resources on-demand. Cluster computing, running on top of virtualized infrastructure, as provided by Cycle Computing, can assist in coordinating the compute and storage resources for high-throughput applications, such as processing NGS data. Bioinformatics workflow services, such as those provided by Nimbus Informatics, have emerged. These services, built on top of infrastructures like AWS, promise to provide an end-to-end platform that automates storing, organizing, and processing large amounts of data in the cloud.

Our group has several Mac workstations, and several Linux servers that adequately served the needs of the group before the advent of our NGS work. Although we had unused local server capacity to perform most of the current NGS data analysis, we could see that additional storage and compute capacity would be needed soon.

Our initial NGS plan was to procure a medium-size cluster with storage and high performance computing capabilities. However, management decided against up-front capital expenditure, so we investigated cloud-based options that could be procured on an as-needed basis, which would cost less initially and provide flexibility if project activities changed. Although the strategy had advantages in flexibility, it also assumed:

- Adequate connectivity to cloud resources from our facility
- Cooperation from our IT and security groups to approve and facilitate putting data outside the firewall
- A significant acquisition of skills by our team

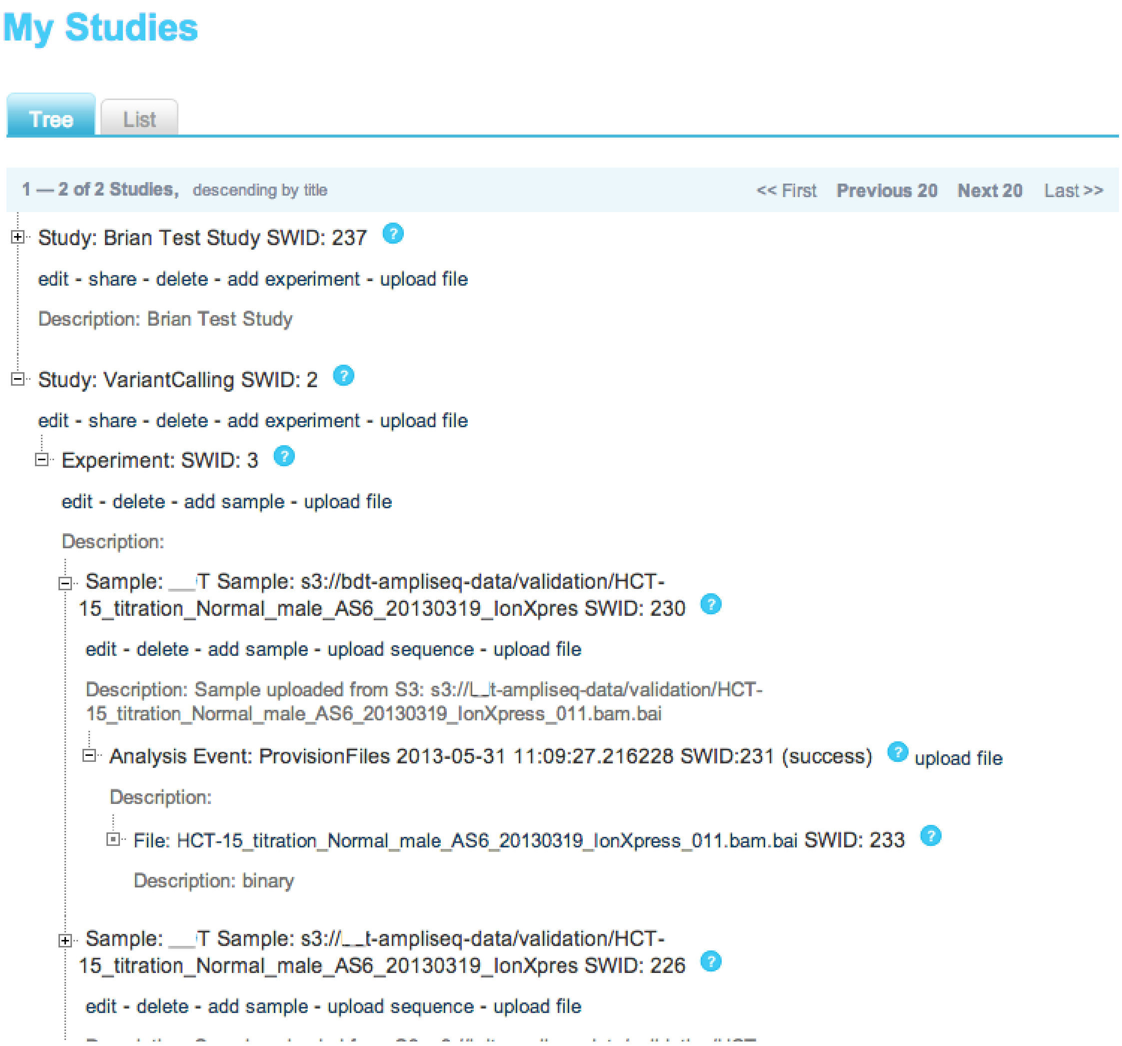


Figure 1: User data is organized hierarchically in SeqWare, with top-level Studies objects that contain one or more Experiments, which contain Samples. Samples map one-to-one to data files, and also have the workflow analysis results attached to them. They keep a detailed record of all analyses performed.

This project with the HPC Experiment is one of the ways our team is developing the necessary skills, and we are investigating options for cloud-based NGS data management and analysis. The project definition phase consisted of consulting with HPCExperiment staff to comment on our initial problem statement. They suggested a team, which began with Cycle Computing as the domain expert. Shortly after that, Tom Dyar learned about SeqWare and Nimbus Informatics, which provides a complete genome-center scale and decided to add the principal Brian O'Connor to the team.

The work plan to achieve the goals included:

- Launch SeqWare AMI and connect to it via SSH through the corporate firewall
- Add ~10 GB NGS sequence raw data samples and annotate within SeqWare MetaDB
- Construct variant calling pipeline, using SeqWare Java library
  - Organize components such as diBayes and ion torrent GATK jar
  - Write/Debug Java code and test on standard BAM file
  - Upload bundle to SeqWare
  - Test on standard BAM file
- Run pipeline on samples, in parallel, and record time to complete, and other metrics
  - Brian O'Connor and Nimbus assisted in providing the team with updated AMI's, accounts on their hosted service, and set up our customized AMI running within Nimbus' infrastructure, so that our custom workflow could easily be run. Cycle Computing was helpful in suggesting ways in which other customers of theirs have performed large-scale sequencing work on AWS.

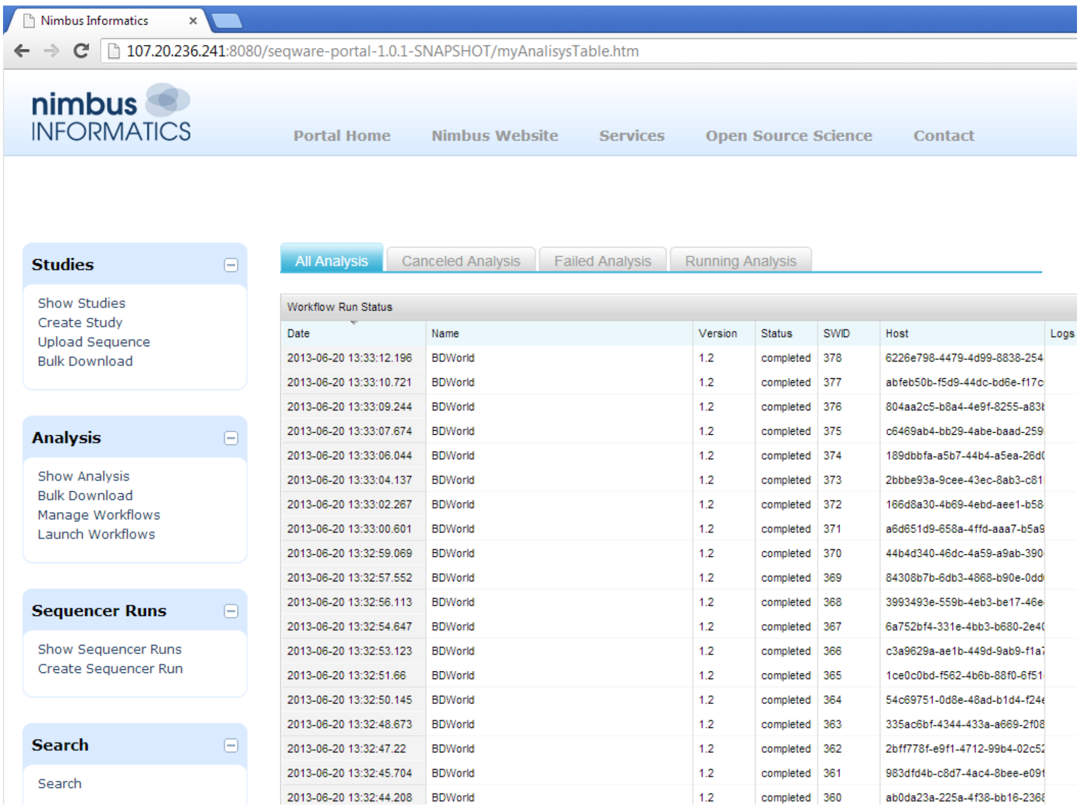


Figure 2: Workflow run status. We ran approximately 30 workflows simultaneously on Nimbus' AWS-based system.

CHALLENGES

The primary challenges we faced revolved around getting good connectivity from our IT group, and approval to proceed with HPCExperiment from our management and corporate legal team. Secondary challenges were technical – learning and using SeqWare.

As our group performed more work on the AWS infrastructure, we encountered several connectivity issues. Typically, we requested ports be opened to support various transfer protocols such as VNC or Remote Desktop Protocol (RDP). Our company also maintains a proxy server through which most traffic passes. The server performs virus and content scanning, which has caused issues such as corrupting large files when downloaded, because the scanning process took so long that the connection timed out. Also, we have adapted to changes made by our information Security group to increase security, such as closing the SSH port 23, which we were using to connect to EC2 instances on AWS.

Finally, we had a globally distributed team with associates in Singapore using a slightly different networking infrastructure that needed modifications. For example, while diagnosing slow connectivity to the Singapore AWS datacenter, we discovered that traffic was being routed first back to the eastern United States before reaching the AWS datacenter. All of these issues typically involved one or several requests to IT, and slowed our progress significantly.

Overall, our management has been supportive of HPC Experiment, but that support is predicated upon sign-off from the Legal department. Since cloud services is a bit of a "hot button" issue, especially since our project involves human DNA data that some might consider highly sensitive, the review process took more time than expected. Multiple individuals with expertise from procurement, information technology, and health information privacy, have all weighed in on the overall project and the pros and cons of performing this type of R&D on AWS and other cloud providers.

SeqWare is a comprehensive open source toolbox, billed as a "genome center in a box," so we expected a considerable learning curve. However, the toolbox is fairly well documented, and we had personal interaction with the principal developer, Brian O'Connor, so our experience overall was positive.

However, we encountered a few difficulties. The first step in using SeqWare is to ingest data into the system, which involves uploading files, and registering them within the metadata database as samples. The samples are then processed by workflows. One option is to use a web interface to the MetaDB system. However, we found the web user interface was slow and had some bugs that made it difficult to work with, so our main interaction with SeqWare was through the command line tools. Yet the command-line tools required invoking very long and complicated commands, which made it difficult to diagnose problems. Once we got some experience with the tools, though, the process was smooth.

To process the samples, SeqWare uses workflows developed in Java. When developing a workflow with the SeqWare Java library, we found that some aspects of working with large files during development need to be sorted out. For example, NGS workflows typically make use of large sequence files used as references. Using the SeqWare tools, it was not apparent how you would transition from a fast develop-compile-debug cycle in which the reference files were not continually re-transferred as part of the workflow packaging process, to a deployed workflow bundle that includes the correct reference file. Although SeqWare generates and uses comprehensive workflow bundles, which ensures a reproducible process, some facilities need to be implemented to reduce the burden on the developer in managing large files. Brian was very helpful, and using our public repository, rewrote and improved our workflow code.

After we had our samples registered, and a workflow developed and tested, we were ready to process our data. The publicly available SeqWare AMI is pre-set to run workflows directly on the EC2 instance without an external cluster, which is a great option for development and testing. However, we wanted to test the scalability of the system by processing the samples in parallel. For this, we utilized the Nimbus Informatics service. Our MetaDB instance and workflow were transferred to Nimbus, and using our account credentials, we were able to initiate workflow runs from the command line of our SeqWare EC2 instance. This configuration and the speed with which Nimbus set it up, underscored the flexibility and maturity of the tools, and was quite impressive. However, once we submitted our jobs via the command line, there was a considerable delay before any samples were actually processed, and the monitoring tools available from the command line (and the web UI) were not very helpful.

BENEFITS

Nimbus Informatics participated in the HPCExperiment’s project to evaluate Nimbus' cloud-based genomics tools. Nimbus Informatics is a provider of hosted computational workflows that enable the analysis of complex, genomics data for customers that lack local HPC infrastructure. In the evaluation, Nimbus and the end-user collaborated on the creation of a sequence variant calling pipeline used to find genomic differences in samples sequenced by the end-user. Like other Nimbus projects, this work leveraged the extensive infrastructure available through the open source SeqWare project (http://seqware.io), which provides workflow and other tools to aid in genomics analysis.

The benefits of this solution are three-fold. First, through partnering with Nimbus, the end-user was able to write workflows using an open source infrastructure. This increased transparency of the engineered solution since both the end-user and Nimbus were able to use the core SeqWare software with complete transparency and few barriers to use.

Second, because of the open nature of the software, the end-users were able to create their own analytical workflows and test them locally. SeqWare, for example, provides virtual machines (VMs) that anyone, including the end-user, can download and use for their custom workflow development. This is a powerful model since most genomics cloud provides are closed source and do not support users creating their own custom workflows.

By providing an open and user-driven solution, Nimbus allowed the end-user to create an analytical workflow that used standard tools but was customized for the particulars of their analytical challenges. The final benefit of the Nimbus solution was the cloud-based hosting of the SeqWare system. While users can choose to build and run their own SeqWare infrastructure on their own hardware, Nimbus adds considerable value by providing the infrastructure on Amazon's cloud transparently to the user. In the case of the trial by the end-user, this meant they could write and test their workflow locally, upload it to the Nimbus infrastructure, and simply "order" workflows on specified samples. Nimbus handled the provisioning of SeqWare infrastructure behind the scenes, monitored workflows, ensured sufficient server capacity was available, and invoiced the end-user for the cloud time used, all without the need for the end-user to interact directly with the cloud. Together, these benefits provided a streamlined and efficient system capable of processing massive amounts of genomic data using an analysis workflow specific to the customer's needs.

CONCLUSIONS AND RECOMMENDATIONS

Since considerable effort was spent working through connectivity issues with IT, a good working relationship with that group, as well as familiarity with their processes, is essential. Recently, our IT department has formed an "R&D IT" sub-group to liaise between R&D associates and the wider IT organization including Networking and Information Security, on a project-specific basis.

These dedicated resources have made a large difference in the efficiency of the request process, and provided a high level of visibility to our projects within the IT management organization. However, our corporation's networking infrastructure is complicated, and there is a tension between security and functionality that will be a continual battle. Also, our group is competing for bandwidth with the rest of the R&D groups and business functions, which will limit our ability to utilize cloud resources going forward. One possibility we are investigating is to purchase our own dedicated bandwidth directly from our Internet vendor. This would give us ample bandwidth for our needs, and allow us to sidestep a lot of the security settings that are in place to protect sensitive corporate data.

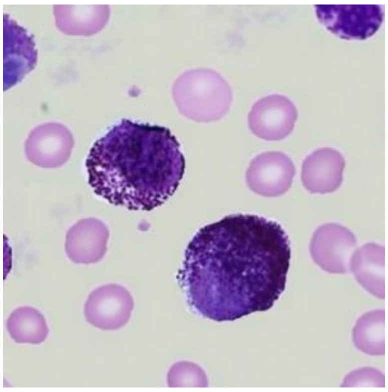
We determined that SeqWare/Nimbus is a comprehensive solution with an attractive open source business model, with promising prospects. We do think that if our group were to use it extensively, we would develop some wrappers around the command line tools to make it user-friendlier. Also, we would work with Nimbus to improve the monitoring capabilities, and push for a better web user interface to the metadata database.

In future rounds of HPC Experiment, we plan to work with Nimbus to explore the processing of large numbers of samples, and the storage, management and analysis of the resulting variant data in the SeqWare Query Engine (SQE). Since SQE is built on open source Hadoop and HBase database systems, there is a vibrant ecosystem of related projects that can be leveraged and integrated to support a wide range of use cases as our group's needs evolve.

Case Study Authors – Thomas Dyar, Senior Software Engineer, Betty Diegel, Senior Software Engineer, Brian O'Connor, CEO Nimbus Informatics

NGS SHRiMP & MAQ

Open Source Clinical Cancer Genomics Pipeline in the Cloud



"UberCloud container images were easily ported between environments by the Operations Engineer."

MEET THE TEAM

End-User: Erinija Pranckeviciene, Bioinformatics Research Scientist, Department of Human & Medical Genetics, Vilnius University
Resource Provider: Amazon AWS
HPC Cloud Experts: Fethican Coskuner, Operations Engineer, UberCloud.

USE CASE

Rapid progress in next generation sequencing is paving the way to individualize cancer treatment based on sequencing of the cancer genome. This process is still evolving and validated open source pipelines to process the enormous amounts of data involved are not available to everyone interested in this field.

Identifying informative genomic variants – potential disease-causing candidates – through the use of next generation sequencing (NGS) data is becoming a routine diagnostic tool. In cancer, exomes of tumor and normal tissues aligned to the reference genome serve as primary data source to start identifying these genomic variants for a better understanding of the disease.

Software programs used to align NGS data to the reference genome are governed by user-defined parameters and map the short-sequenced reads to the reference genome with varying accuracy [4]. Most of the alignment tools require a long time to obtain accurate mappings of the short reads to the reference genome.

Decisions as to which of the alignment tools have to be included into the NGS data processing pipeline depend on many factors such as ease-of-use of tools, their public availability, and their ability to align NGS data from various sequencing platforms (Illumina and AB Solid). Among those factors the time required to accomplish the alignment plays a very important role. The time required for the alignment program to complete the mapping, depends on the size of the NGS data set and scales with available computational resources.

The main goal of this project was to start preparing guidelines on the optimal choices of open source alignment tools to be included into the NGS data processing pipelines by matching these tools to the available computational resources. We started by estimating how much time the alignment tools needed to complete the task of alignment of the cancer data-set to the human reference genome by running these tasks on various configurations of high performance computing (HPC) resources.

At this stage two aligners were chosen: Mapping and Assembly with Qualities (MAQ 0.6.6 and 0.7.1) [1] and Short Read Mapping Package (SHRiMP 2.2.3) [2].

MAQ is one of the most accurate aligners and supports gaped alignments of paired end reads. However, MAQ was programmed in such way that it can't take advantage of available multiple-core processor architectures.

SHRiMP is an exhaustive mapping tool because it finds multiple candidate alignment locations (CALs) and then selects the best hits out of those CALs. SHRiMP can fully utilize and take advantage of the available memory and multiple-core processor architectures. The computational resources used in this project are listed in Table 1.

Table 1: Computing Resources

A) AWS	B) BARE METAL #1	C) BARE METAL #2	D) WORKSTATION AT THE DEPT. OF HUMAN AND MEDICAL GENETICS, VILNIUS UNIVERSITY #3
CPU: Intel® 8 CPU cores (Model is unknown) RAM: 68GB	CPU: Intel Xeon CPU E5-2650, 32 cores RAM: 32GB	CPU: Intel Xeon(R) CPU E5620, 16 cores RAM: 144GB	Allocated resource: CPU: AMD Opteron (tm) Processor 6172, 12 cores RAM: 98GB
OS: Ubuntu 12.10 LTS Server MAQ: 0.6.6	OS: Ubuntu 12.04.2 LTS Server MAQ: 0.6.6 SHRIMP: 2.2.3	OS: Ubuntu 12.04.4 LTS Server MAQ: 0.7.1 SHRIMP: 2.2.3	OS: CentOS 5.10 MAQ: 0.7.1 SHRIMP: 2.2.3

To enable portability of the computation environment (including the aligners, scripts, configuration settings) Linux container images were used. The same image was used in the AWS, Bare Metal #1, and Bare Metal #2 test. The Linux container images were easily ported between environments by the Operations Engineer.

PROJECT OBJECTIVES

Objective 1 – Document how much time it took for MAQ and SHRiMP to align 2 million 75 base-pair (bp) length paired end sequence reads of cancer exome data [3] to the human reference genome. The programs were tested on several HPC configurations summarized in Table 1.

Objective 2 – To compute a complete mapping of the mast-cell leukemia exome data (tumor and germline available in Illumina Genome Analyzer FASTQ format) [3] to the Human Genome Build 37 (GRCh37 hg19).

RESULTS

Total number of paired end reads in a whole sample consisted of 32,971,877 reads in the germline exome and of 37,495,826 reads in the tumor exome. To compare chosen alignment programs in terms of their execution time on different HPC configurations, 2 million paired end reads of the germline exome was used. Durations in which MAQ completed alignments of 2 million reads on tested computing resources are shown in Table 2. The shortest duration was achieved on the HPC configuration **C**. Next, MAQ was tested by using UberCloud Linux container images on HPC architecture C. The alignments of 2 million paired end reads were executed simultaneously on 4 nodes. Each completed in 2 hours and 5 minutes.

SHRiMP can fully utilize multiple-core processor architectures. We were interested in how much the alignment time could be reduced by using the aligner, which takes advantage of available multi-core computing resources. SHRiMP can use as many threads as set by a parameter. Durations in which SHRiMP aligned 2 million reads on different HPC architectures are shown in Table 2.

To summarize the comparative results, we note that MAQ completed the task in 75 minutes and SHRiMP completed the same task in 40 minutes by using 16 threads. Both of them completed the alignment of 2 million paired end reads within similar time range.

Table 2: Running times of the alignment tasks

ALLIGNMENT TASK	**A) AWS**	**B) BARE METAL #1**	**C) BARE METAL #2**	**D) WORKSTATION AT THE DEPT. OF HUMAN AND MEDICAL GENETICS, VILNIUS UNIVERSITY #3**
MAQ (2 mln reads) version 0.6.6	cr1.8xlarge more than 2 days experiment stopped			
MAQ (2 mln reads) version 0.7.1	m1.medium 2 h 20 min	1h 30 min	1 h 13 min	4 h 30 min
SHRiMP (2 mln reads) using 1 thread using 16 threads using 24 threads			 3 h 20 min 1 h 40 min 38 min	

We estimated that the time required for MAQ to complete the mapping of 2 million reads on the architecture C within the virtualization layer takes a little over two hours. Based on this estimate, we predicted that time, required to map exomes of germline and tumor divided into 36 parts of 2 million reads each, should take approximately 36*2= 72 hours. By using 4 computing nodes simultaneously for this task the mapping should complete in 18 hours. The estimated time for SHRiMP using 24 threads to map 2 million reads takes about 40 minutes. Based on this estimate, we predicted that SHRiMP will align the whole sample in 25 hours. A summary of the actual time spent in the mapping of whole sample on the HPC architecture C is shown in Table 3.

We used 24 threads in mapping with SHRiMP. The mappings were performed in batch on a single node. This mapping task completed in 29 hours, which is more or less consistent with previous expectation of 25 hours that were predicted for the whole sample based on the time estimate of mapping 2 million reads (see Table 2).

Mapping by MAQ was performed using UberCloud Linux containers on four nodes. Two nodes were used to map the germline and another two nodes were used to map the tumor sample. Alignment of the whole sample by MAQ completed in 35 hours, which is almost twice as long as expected.

There are several reasons why MAQ had prolonged mapping time. The first is that, in general, reads have different sequence complexity, which influences mapping time. The average actual mapping time of a part of 2 million reads was approximately three hours. The second reason was unexpected interruptions in a batch mapping because of unexpected symbols in some parts of the data. The second reason is minor – the data was cleaned and processed again.

Table 3: Alignment of the whole exome of germline and tumor by using HPC configuration C

Alignment program	Germline sample 0-16 parts x 2million reads	Tumor 0-18 parts x 2million reads
MAQ version 0.7.1 utilized 4 nodes in parallel	In batch: Node1 (part 0-6) From Jun 7 21:02 Till Jun 9 07:42 Node2 (part 12 - 14) From Jun 9 02:38 Till Jun 9 11:57 Realigned: Parts 7,8,9, 10,11,14,15,16	In batch : Node3 (part 0-6) From Jun 7 21:04 Till Jun 9 10:42 Node4 (part 10-16) From Jun 7 21:04 Till Jun 9 07:18 Realigned: Parts 7.8,9 17,18
SHRiMP ver 2.2.3 using 24 threads on 1 node	From Jun 7 22:02 Till Jun 8 11:59 Total 14 hours	From Jun 8 12:57 Till Jun 9 04:00 Total 15 hours

CONCLUSION AND RECOMMENDATIONS

The team gained two particularly useful insights from this project:

Some software programs are created in a such a way that they can't take advantage of modern multiple core architectures. Therefore, a virtualization approach was utilized to perform mapping with simple alignment program MAQ simultaneously on four nodes. Running simple programs in virtualization layers on HPC resources proved to be a very productive use of those resources resulting in a considerable reduction of execution times of simple processes. In NGS data analysis, numerous useful software programs exist that can't use modern HPC architectures. Using these programs to process big data is challenging because of the time required to obtain results. One possible solution to this challenge is to use the UberCloud Linux containers on available HPC architecture.

The predicted mapping times resulting from estimates of mapping times computed on small randomly selected sample of reads are only approximate. This most likely occurs because sequence complexity impacts finding a candidate alignment location of the read in the reference genome.

Alignment of the short reads to the reference genome in NGS exome analysis pipeline is only a single intermediate step in obtaining information on which genomic variants are present in the exome. A second step after the alignment is search and identification of those genomic variants. This step is computationally demanding as well. It would be beneficial to assess a computational intensity and create resource usage guidelines for genomic variant identification tools such as Genome Analysis Toolkit (GATK), Samtools and MAQ.

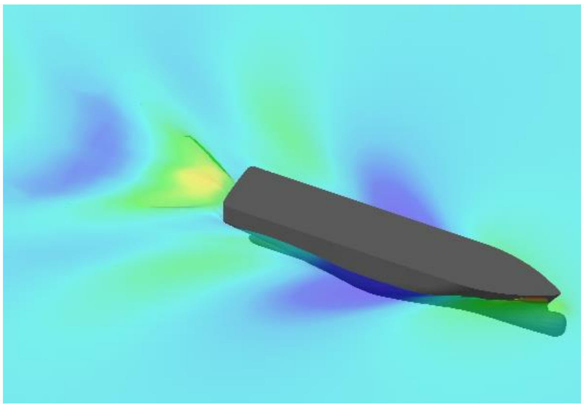
REFERENCES

2. Li, H, Ruan, J, Durbin, R (2008). “Mapping short DNA sequencing reads and calling variants using mapping quality scores.” *Genome Res.*, 18, 11:1851-8.
3. David, M, Dzamba, M, Lister, D, Ilie, L, Brudno, M (2011). “SHRiMP2: sensitive yet practical SHort Read Mapping.” *Bioinformatics*, 27, 7:1011-2.
4. Spector, MS, Iossifov, I, Kritharis, A, He, C, Kolitz, JE, Lowe, SW, Allen, SL (2012). “Mast-cell leukemia exome sequencing reveals mutation in the IgE mast-cell receptor b chain and KIT V654A.” *Leukemia, 26(6):1422-5.* PMID:22173243
5. Hatem, A, Bozda, D, Toland, AE, Catalyurek, UV (2013). “Benchmarking short sequence mapping tools.” *BMC Bioinformatics*, 14:184, p.1-25.

Case Study Author – Erinija Pranckeviciene

NUMECA FINE/Marine

Prediction of Barehull Resistance Curve of a Day-Cruise Ship in the Cloud



"The high accuracy of NUMECA FINE/ Marine, combined with the UberCloud container in the CPU 24/7 Cloud is a ***fantastic tool****. It offers an opportunity for engineers to keep full control & knowledge on their projects in the Marine Hydrodynamic Environment."*

MEET THE TEAM

End User – Costas Carabelas, Managing Director at Costas Carabelas Technical Office (CCTO), ship design office, Athens, Greece
Team Expert – Vassilios Zagkas, SimFWD Engineering Services, Athens, Greece
Software Provider – Aji Purwanto, Business Development Director, NUMECA International S.A.
Resource Provider – Richard Metzler, Software Engineer, CPU 24/7 GmbH
Technology Experts – Hilal Zitouni Korkut and Fethican Coskuner, UberCloud Inc.

SimFWD is a research, development and application company providing engineering services to the transport and construction industries, focusing on CAE technologies such as CFD and FEM applied to Ship Design. SimFWD can provide turnkey solutions to complicated generic problems in a cost effective manner, eliminating the overheads normally associated with a dedicated engineering analysis group or department. SimFWD aims at helping customers develop product designs and processes by supplying them with customized engineering analysis and software solutions.

Costas Carabelas Technical Office (CCTO) is a ship design office based in Athens, Greece. CCTO is actively involved in numerous projects in the marine and offshore sector. CCTO is famous for its design of the Onassis yacht Christina O, https://en.wikipedia.org/wiki/Christina_O.

USE CASE

Objective 1 of this case study was to calculate the calm water resistance of a new small range Cruise Ship concept hull (~90m overall length). The hull-form was developed by CCTO and was specifically designed to combine high capacity and good cruising speed between multiple destinations. Listed below are the main dimensions:

L.O.A.:	87.00	(m)
Breadth:	14.40	(m)
Service Speed:	17.00	(kn)
Draught:	3.900	(m)
Block Coefficient (T=3.90):	0.57	

Objective 2 was for the end-user CCTO to get familiarized with the use of FINE/Marine in an UberCloud application software container and compare the cost benefit in comparison to resources currently in use. The benchmark was analyzed on the bare-metal cloud solution offered by CPU 24/7 and UberCloud. All simulations were run using version 4.1 of NUMECA's FINE/Marine software.

SimFWD provided support to CCTO in setting up the FINE/Marine model and simulation parameters, having as a goal the generation of an initial Power Curve in short time. This shall be a good helping point at this design stage and shall be later enhanced by Self-Propulsion tests in FINE/Marine and verified by model tests.

CHALLENGES AND BENEFITS

The case study was completed without facing any difficulties whatsoever. The entire process right from the access to files in the UberCloud container, running the jobs in the CPU 24/7 Cloud, up to the retrieval of results to a local workstation was very convenient and without any delays. The user-friendliness of the interface was major advantage!

SIMULATION PROCESS AND RESULTS

Computations on the hull-form were performed for 5 different speeds – 13kts, 15kts, 17kts, 19kts, and 20kts. All computations were performed using a fluid domain consisting of approximately 1 million cells except for a speed of 19kts, where a finer mesh containing approximately 2 million cells was additionally computed. Shown below are results for a speed of 19kts:

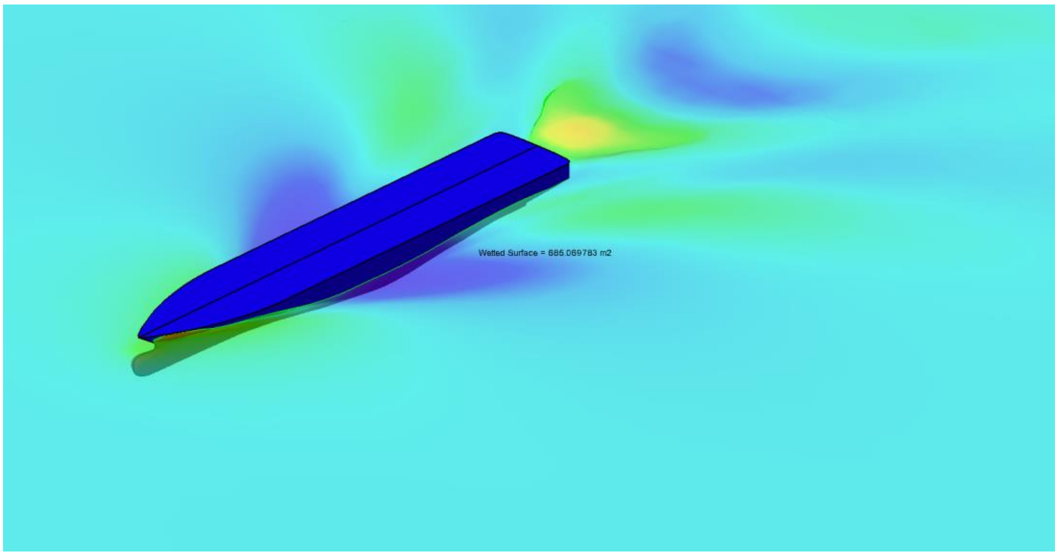


Figure 1: Travelling shot at 19knots: Wetted Surface 676 m².

Parameters | Speeds			19.0 kts
FINE/Marine global, Resistance	**Units**		
Rt (Fx)	[N]		417902
Trim (Ry1)	[deg]		-0.2300
Sink (Tz)	[m]		3.7600
ΔSink (Tz)	[m]		-0.1400

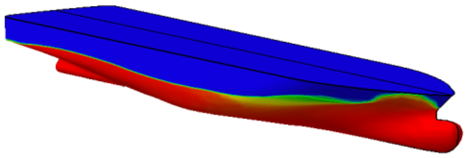
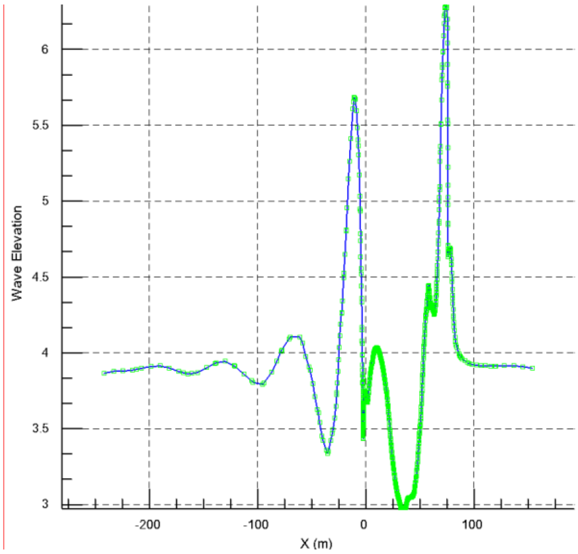



Figure 2: Wave Elevation along hull length X.

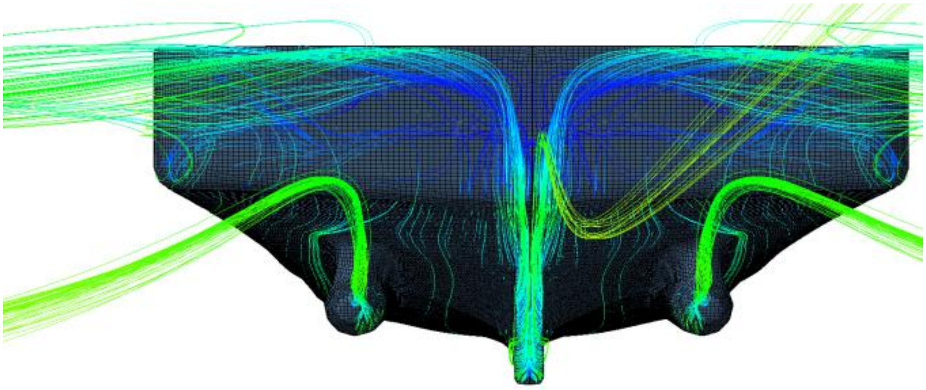


Figure 3: Streamlines colored by the relative velocity.

Hull Pressure Effects

The bow hull pressure has a normal distribution over the most affected regions, bow front and stem near the waterline entrance.

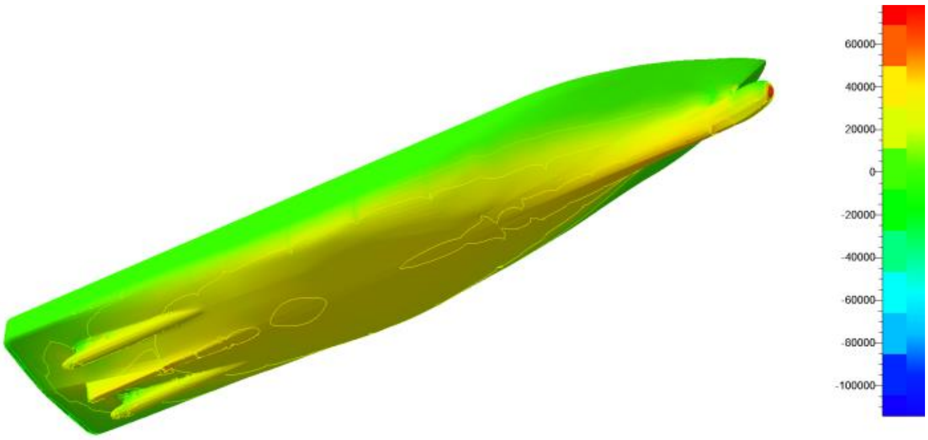


Figure 4: Overall Hull Pressure.

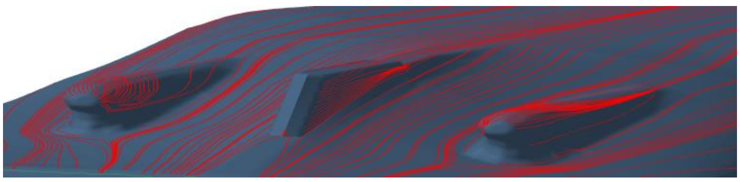


Fig. 5 Flow streamlines on the hull

Following are some details regarding the simulation setup:
Number of time steps : 1500
Number of non-linear iterations : 5
Different numbers of CPUs (10, 16 & 32) were tested on the computation variants. This proved that FINE/Marine is efficient from the scalability point of view.

CONCLUSIONS

The range of computations converged well for all speeds and the overall result was deemed as reliable prediction for the range of bare hull resistance also compared to empirical results and similar designs. This allows CCTO to set the boundaries on their Initial Design process and work towards the next steps with confidence. Results have also given valuable insight on available margin to optimize wave resistance at the bow and streamlines in the after part.

- We showed that the CPU 24/7 HPC bare-metal cloud solution provides performance advantages for NUMECA FINE/Marine users who want to obtain higher throughput or analyze larger, more complex models.
- FINE/Marine provides proven highly dedicated tool for naval architects especially with its C-Wizard, embedded automated full-hex HEXPRESS mesh generator, easiness-to-use, performance and accuracy reducing the engineering and development time and cost.
- CPU 24/7 and UberCloud effectively eliminate the need to maintain in-house HPC expertise.
- The container approach provides immediate access to high performance clusters and application software without software or hardware setup delays.
- The browser-based user interface is simple, robust, and responsive.

APPENDIX: UberCloud Application Containers for NUMECA FINE™/Marine

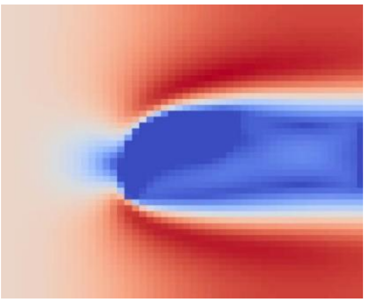
UberCloud Containers are ready-to-execute packages of software. These packages are designed to deliver the tools that an engineer needs to complete his task in hand. In this experiment, the FINE™/Marine software has been pre-installed, configured, and tested, and were running on CPU 24/7's bare metal servers, without loss of performance. The software was ready to execute literally in an instant with no need to install software, deal with complex OS commands, or configure.

The UberCloud Container technology allows wide variety and selection for the engineers because the containers are portable from server to server, Cloud to Cloud. The Cloud operators or IT departments no longer need to limit the variety, since they no longer have to install, tune and maintain the underlying software. They can rely on the UberCloud Containers to cut through this complexity. This technology also provides hardware abstraction, where the container is not tightly coupled with the server (the container and the software inside isn't installed on the server in the traditional sense). Abstraction between the hardware and software stacks provides the ease of use and agility that bare metal environments lack.

Case Study Authors: Costas Carabelas and Vassilios Zagkas

OpenFOAM

Deep Learning for Steady-State Fluid Flow Prediction in the Advania Data Centers Cloud



"The overhead of creating high volumes of samples can be effectively compensated by the high-performance containerized computing environment provided by UberCloud and Advania."

1 MEET THE TEAM

End-User: Jannik Zuern, Renumics GmbH, Karlsruhe, Germany
Software Provider: OpenFOAM open source CFD software
Resource Provider: Advania Data Centers Cloud, Iceland
HPC and AI Experts: Stefan Suwelack, Markus Stoll, and Jannik Zuern, Renumics; Joseph Pareti, AI Consultant; and Ender Guler, UberCloud Inc.

2 USE CASE

Solving fluid flow problems using Computational Fluid Dynamics (CFD) is demanding both in terms of computer processing power and in terms of simulation duration. Artificial neural networks (ANN) can learn complex dependencies between high-dimensional variables. This ability is exploited in a data-driven approach to CFD that is presented in this case study. An ANN is applied in predicting the fluid flow given only the shape of the object that is to be simulated. The goal of the approach is to apply an ANN to solve fluid flow problems to significantly decrease time-to-solution while preserving much of the accuracy of a traditional CFD solver. Creating a large number of simulation samples is paramount to let the neural network learn the dependencies between simulated design and flow field around it.

This project between Renumics GmbH (Karlsruhe) and UberCloud Inc. was therefore established to explore the benefits of additional cloud computing resources that can be used to create a large amount of simulation samples in a fraction of the time a desktop computer would need to create them. In this project, we want to explore whether the overall accuracy of the neural network can be improved when more samples are being created in the UberCloud Container und then used during the training of the neural network. UberCloud kindly provided the cloud infrastructure, a CentOS Docker container with an OpenFOAM installation, and additional tech support during the project development.

3 WORKFLOW OVERVIEW

In order to create the simulation samples automatically, a comprehensive workflow was established.

As a **first step**, random two-dimensional shapes are created. These shapes have to be diverse enough to let the neural network learn the dependencies between different kinds of shapes and their respective surrounding flow fields.

In the **second step**, these shapes are meshed and added to an OpenFOAM simulation case template (Fig. 1). This template is simulated using the steady-state solver OpenFOAM solver simpleFOAM.

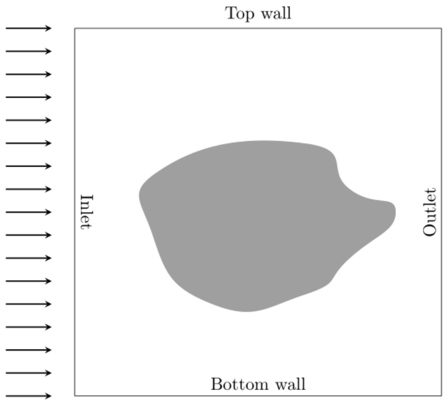


Figure 1: Simulation case setup. The flow enters the simulation domain through the inlet, flows around the arbitrarily shaped obstacle (dark grey shade) and leaves the simulation domain through the outlet.

In the **third step**, the simulation results are Post-Processed using the open-source visualization tool ParaView. The flow-fields are resampled on a rectangular regular grid to simplify the information processing by the neural net.

In the **fourth and final step**, both the simulated design and the flow fields are fed into the input queue of the neural network. After training, the neural network is able to infer a flow field merely from seeing the to-be-simulated design.

In Figure 2, a visualization of the four-step Deep Learning workflow is shown.

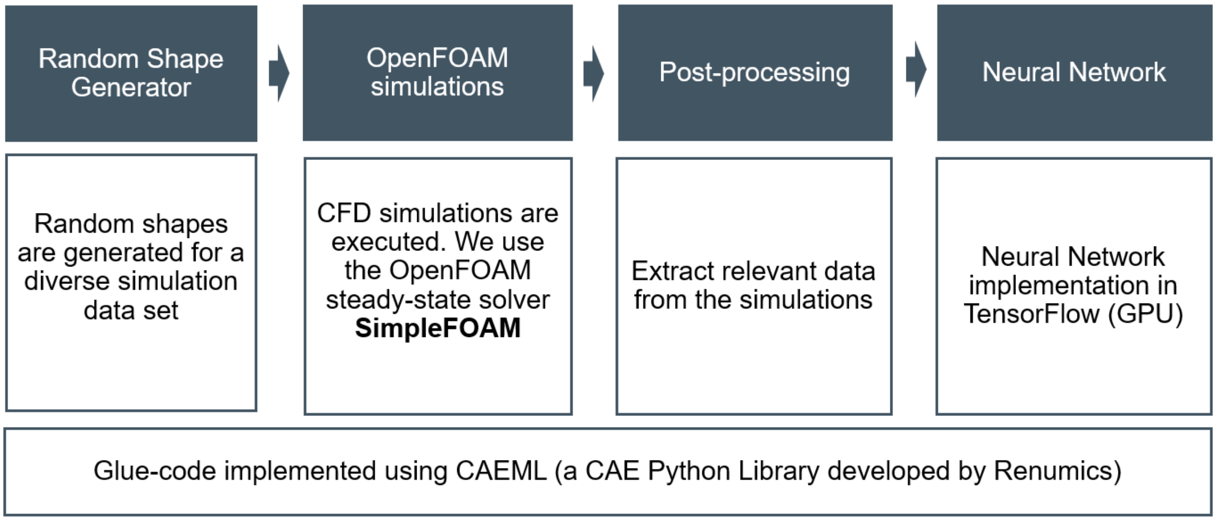


***Figure 2: Deep Learning workflow*.**

Hardware specs

The hardware specs of the Advania Data Centers compute node hosting the UberCloud container are as follows:

- 2 x 16 core Intel Xeon CPU E5-2683 v4 @ 2.10 GHz
- GPU: none
- Memory: 251 GB

The hardware specs of the previously used desktop workstation are as follows:

- 2 x 6 core Intel i7-5820K CPU @ 3.30 GHz
- GPU: GeForce GTX 1080 (8GB GDDR5X memory)
- Memory: 32 GB

4 RESULTS

Time needed to create samples

As a first step, we compared the time it takes to create samples on the desktop workstation computer with the time it takes to create the same number of samples on the UberCloud container. Figure 3 illustrates the difference in time it took to create 10,000 samples.

On the desktop computer it took 13h 10min to create these 10,000 samples. In the UberCloud OpenFOAM container in the Advania Data Centers Cloud, it took about 2h 4min to create 10,000 samples, which means that a speedup of 6.37 could be achieved using the UberCloud container.

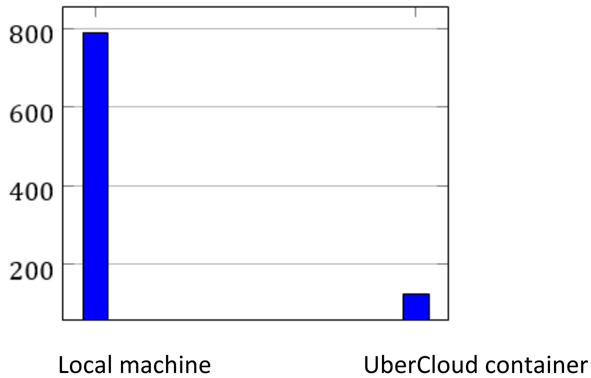


Figure 3: Comparison between Local machine and UberCloud container, vertical axis shows time in minutes.

Neural Network performance evaluation

A total of 70,000 samples were created. We compare the losses and accuracies of the neural network for different training set sizes. In order to determine the loss and the accuracy of the neural network, we first must define, what these terms actually mean.

- Performance for generating the flow field data set and Tensorflow training

Setup	2-D external flow	3-D internal flow
Time for 10.000 simulations	13.2 h	152.5 h
Time for training	23.7 h	48.5 h

- Neural network prediction of flow field

Setup	2-D external flow	3-D internal flow
Time for CFD solver	4.7 s	55.0 s
Time of neural network prediction	3 ms	120 ms
Speedup factor with deep leaning	**1566**	**458**

Figure 4: Performance and speedup of flow simulations with neural network prediction.

Definitions

Loss: The loss of the neural network prediction describes how wrong the prediction of the neural network was. The output, or prediction, of the neural network in our project is a $N \times M \times 2$ tensor since the network tries to predict a fluid flow field with N elements in x-direction, M elements in y-direction, and two flow velocity components (velocity in x-direction and velocity in y-direction). A mean-squared-error metric was used to calculate the loss *l*:

$$l = \frac{1}{2}\sum_{d=0}^{1}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(v_{dij} - \bar{v}_{dij})^2 \tag{1}$$

where v_{dij} denotes the ground-truth velocity component in dimension *d* at the grid coordinates (*i,j*), $\overline{v}_{dij}$ denotes the predicted velocity component at the same position and in the same dimension. The goal of every machine learning algorithm is to minimize the loss of the neural network using numerical optimization schemes such as Stochastic Gradient Descent. Thus, a loss of 0.0 for all samples would mean that every flow velocity field in the dataset is predicted perfectly.

Accuracy: In order to be able to make sensible statements about the validity of the prediction of the neural network, metrics have to be defined that describe the level of accuracy that the neural network achieves. In general, the accuracy of a neural network describes how accurate the prediction of the neural network was. While the loss of a neural network is the metric that is being minimized during training, a small prediction loss does not necessarily mean that the corresponding prediction is physically meaningful. In general, however, a small prediction loss usually corresponds with a high accuracy. Different measurements of how accurate the outputs of the neural network are needed to express the validity of the predictions. A highly accurate prediction should have high values for all formulated accuracy measurements and a low loss at the same time. These accuracies can have values between 0.0 and 1.0, where an accuracy of 0.0 indicates that the prediction of the neural network does not at all coincide with the ground truth flow metric that is examined, and an accuracy of 1.0 means that the prediction coincides perfectly with the ground truth flow metric. Bear in mind that a low loss does not necessary cause high accuracy and vice versa. However, the two measurements are typically correlated.

In this study, two different accuracies were evaluated: Divergence accuracy and Drag accuracy:

- **Divergence accuracy:** Numerical CFD solvers aim to find a solution to the continuity equation and the momentum equation. For an incompressible fluid, the continuity equation dictates that the divergence of the velocity vector field is zero for every point in the simulation domain. This follows the intuition that at no point in the simulation domain fluid is generated (divergence would be greater than zero) or ceases to exist (divergence would be smaller than zero). By design, the Finite Volume Method preserves this property of the fluid even in a discretized form. A data-driven approach should as well obey this rule.

- **Cell accuracy:** The number of correctly predicted grid cells in the two- or three-dimensional grid yields an intuitive metric for how well the neural network predicts fluid flow behavior. As the network will never be able to predict the fluid flow velocity down to the last digit of a floating-point number, the following approach is proposed: If the relative error between the network prediction and the actual flow velocity is smaller than 5%, the respective grid cell is declared as predicted correctly. The cell accuracy can be calculated by counting the number of correctly predicted grid cells and dividing the results by the total number of grid cells.

5 TRAINING RESULTS

The generated samples are divided into the training and validation datasets. The training- and validation loss for different numbers of training samples was evaluated. Concretely, the neural net was trained three times from scratch with 1,000, 10,000, and 70,000 training samples respectively.

The following training parameters were used for all neural network training runs:

- Batch size: 32
- Dropout rate: 0.5
- Learning rate: 5×10^{-4}

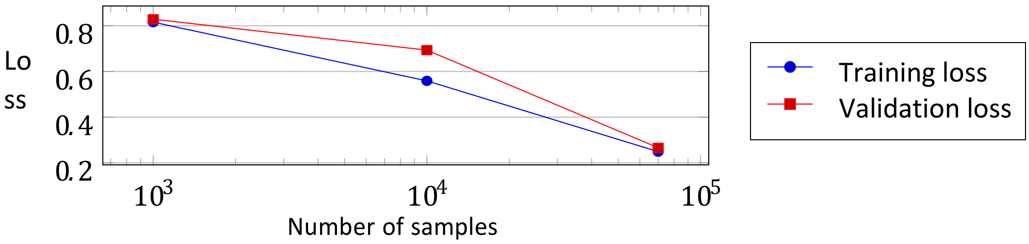


Figure 5: Loss after 50,000 training steps.

It can be observed that both training- and validation losses are lowest for the 70k samples training and are highest for the 1k training samples. The more different samples the neural network processes during the training process the better faster it is able to infer a flow velocity field from the shape of the simulated object suspended in the fluid. The validation loss tends to be higher than the training loss for all tested numbers of samples, which is a typical property of machine learning algorithms. Figure 6 shows the loss after 300,000 training steps:

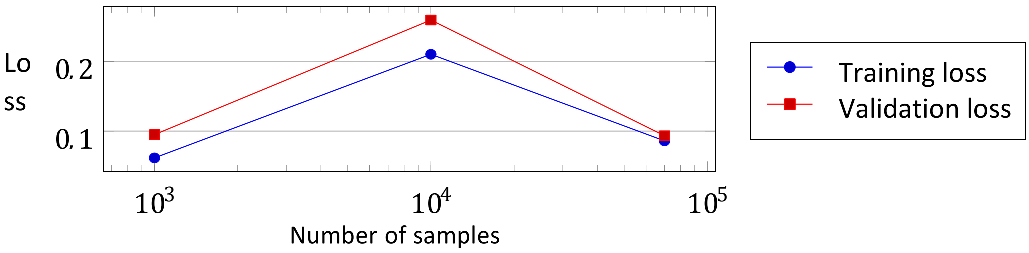


Figure 6: Loss after 300,000 training steps.

Surprisingly, the final training- and validation losses for the 70k samples training session are as low as the losses for the 1k samples training session. Generally speaking, no clear tendency towards lower losses when increasing the set of the training samples could be observed. This result is somewhat surprising since we expected the final losses at the end of the training process to show a similar tendency towards lower losses for higher numbers of samples. We assume that the number of samples does not heavily influence the final loss for extensive training sessions with many hundreds of thousand training steps. Finally, in Figure 7 the divergence and grid accuracies are visualized.

Both the divergence accuracy and the grid cell accuracy show higher values for larger numbers of samples. While the divergence accuracy shows overall high values going from 0.94 for 1,000 samples to 0.98 for 70,000 samples, the grid cell accuracy also increases from a value of 0.53 for 1,000 samples to a value 0.66 for 70,000 samples. To recap: a grid accuracy of 0.66 means that approximately two thirds of all velocity grid cells were predicted correctly within 5% relative error to the correct value.

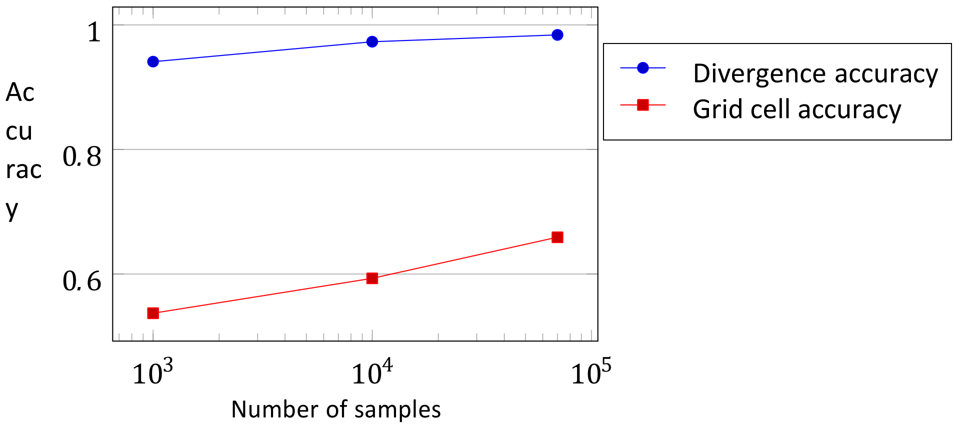


Figure 7: Validation accuracies after training.

Figure 8 illustrates the difference between the ground truth flow field (left image) and the predicted flow field (right image) for one exemplary simulation sample after 300,000 training steps. The arrow direction indicates the flow direction and the arrow color indicates the flow velocity. Visually, no difference between the two flow fields can be made out.

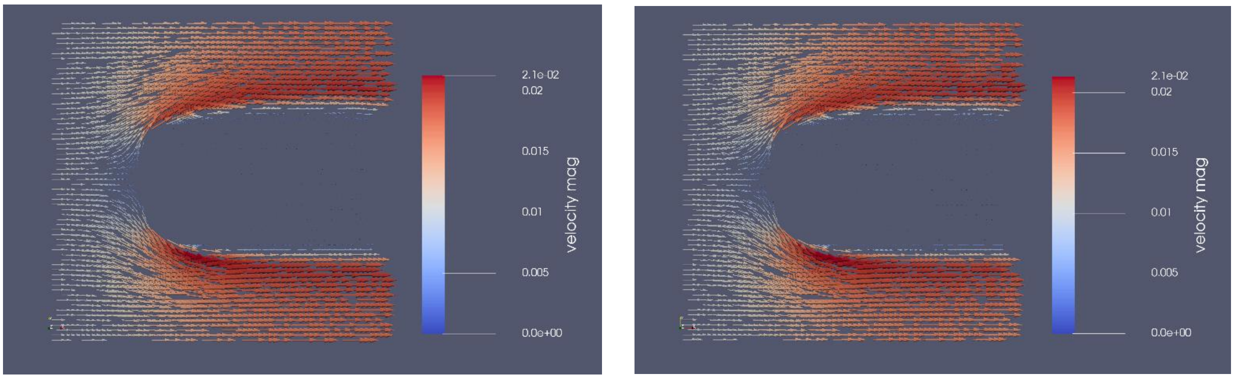


Figure 8: Exemplary simulated flow field (left image) and predicted flow field (right image).

CONCLUSION

We were able to prove a mantra amongst machine learning engineers: *The more data the better*. We showed that the training of the neural network is substantially faster using a large dataset of samples compared to smaller datasets of samples. Additionally, the proposed metrics for measuring the accuracies of the neural network predictions exhibited higher values for the larger numbers of samples. The overhead of creating high volumes of additional samples can be effectively compensated by the high-performance containerized (based on UberCloud Docker) computing node provided by UberCloud on the Advania Data Centers Cloud. A speed-up of more than 6 compared to a state-of-the-art desktop workstation allows creating the tens of thousands of samples needed for the neural network training process in a matter of hours instead of days.

In order to train more complex models (e.g. for transient 3D flow models) much more training data will be required. Thus, software platforms for training data generation and management as well as flexible compute infrastructure will become increasingly important.

Case Study Author – Jannik Zuern, Renumics GmbH

QUANTCONNECT

Quantitative Finance Historical Data Modeling

"This experiment used the cloud for processing large amounts of data in a massively parallel fashion."

MEET THE TEAM

End Users – These are thousands of end users around the world, using the back-testing service through a web browser.

Software Provider - Jared Broad, Founder and CEO of QuantConnect Corporation, a start-up company based in New York City providing an online back-testing service.

Resource Provider - Amazon AWS.

USE CASE

This experiment used the cloud for processing large amounts of data in a massively parallel fashion. We have terabytes of financial data, and can scale unlimited nodes to process it in finite time periods. Users access this cloud through a coding environment in a web browser. A financial back-test is a process of testing algorithms on historical time series data to verify a trading strategy.

Currently it is difficult to process financial data as an individual or on your home PC due to the large volumes of data created each day and the very high expense to purchase financial data. The Software Provider QuantConnect makes the data freely available in a cloud platform, allowing the user access to the data temporarily. It is not uncommon to hear stories of back-tests taking 24-48 hours on users home PCs.

The experiment end-user required extensive processing power to quickly analyze six years of a FOREX tick data, equating to roughly 500MB per simulation. The user would design an algorithm and then perform 10-20 simulations on the dataset, often slightly adjusting the algorithm to optimize variables of different simulations. This problem was ideally suited to a cloud of nodes, where each node processes one long running simulation.

Over the period of two months the user's back-tests were performed and the logs were analyzed. The user's strategy iterated over 10 times during the period, and he performed roughly 1000 hours of compute in total on powerful nodes.

End User - The typical profile of an end-user is either a financial engineer working in a bank or hedge fund, who designs algorithms in his free time, or a general software engineer who has an interest in trading and automating his strategy. They are generally very educated individuals. One such user is

Eyal Zach, a C and C++ embedded systems engineer and quality assurance engineer who designed a FOREX trading system inside QuantConnect.

CHALLLENGES

Amazon EC2 cluster performed well and each node was robust during the trials. Our key limitation was in scaling quickly to user demand as the nodes took several minutes to load an image. This load period was unacceptable so a constant baseline of nodes was required which added to the project cost.

Costs - Using AWS comes with significant cost, which is highly unpredictable and depends on the server demand and complexity of the algorithms we process. QuantConnect made significant innovations to lower the overhead costs. The primary result is an advanced load prediction and machine image scaling algorithm, which learns from historical usage per user and accounts for many environmental factors.

AMI Wait Time - The primary reason for the evolution in the load balancing algorithm was the long delays in machine image spin up time. Amazon currently takes between 3-5 minutes to load the 30GB machine image. This is an unacceptable wait time for an end-user looking to get quick back-test results.

With a fully scaled cloud, running a massively parallel back-test we were able to process 5GB of data within minutes, a process which would normally take 12-24 hours on a home desktop computer.

DLL Upload - The software provider recently launched DLL upload support [1] so registered users could compile the algorithm locally on their PC and upload the DLL with any supporting supplementary files they require. It was a challenging process to ensure the security of the nodes from 100% arbitrary code.

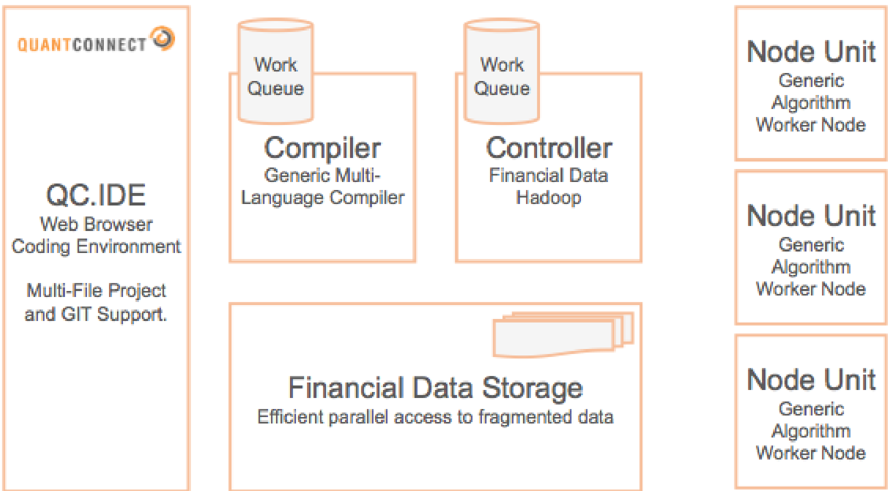


Figure 1: Block Diagram of the QuantConnect platform on AWS.

We solved this problem by making the algorithm worker nodes 100% disposable for paying clients. After a single run the potentially infected nodes are disposed and a fresh set are created for the next user. It takes machine images use scenario to the extreme – potentially only using a worker node for 10 minutes before disposing of it.

CONCLUSIONS AND RECOMMENDATIONS

We recommend there be a global standard version of Hadoop developed for financial data. It is quite a unique problem as the data storage and retrieval of a typical Apache Hadoop is relatively inefficient. Therefore, this experiment used QuantConnect's own proprietary implementation specifically for fragmented time series data and implemented a custom Hadoop controller to splinter the jobs and collect the result data.

There is potential for an easy instantly scalable cloud based from Amazon machine images. Users expect responses within 1-2 seconds and the image turn on time of EC2 is unacceptable for user interaction. There is also potential for a private company to offer an alternative to Amazon Simple Queue Service (SQS). SQS is slow, expensive and has limited data storage capacity. We would gladly substitute Amazon for a private company offering a better product that is in line with SQS API. Alternative open source queues such as RabbitMQ don't use a REST API and are often blocking designs, which require code changes to implement in the platform.

Through fragmenting data and massively parallel analysis, the solution provider was able to speed up time series analysis several orders of magnitude. This has the important limitation that each node unit is ignorant of the past. There is no easy solution to make the nodes aware of the past data or calculations as most efficient formulas are calculated online (on streams of data), which requires the data to appear in order.

Some end users required series analysis (knowledge of past events), so they preferred to use QuantConnect cloud to run 10+ versions of the same algorithm concurrently to optimize algorithm variables.

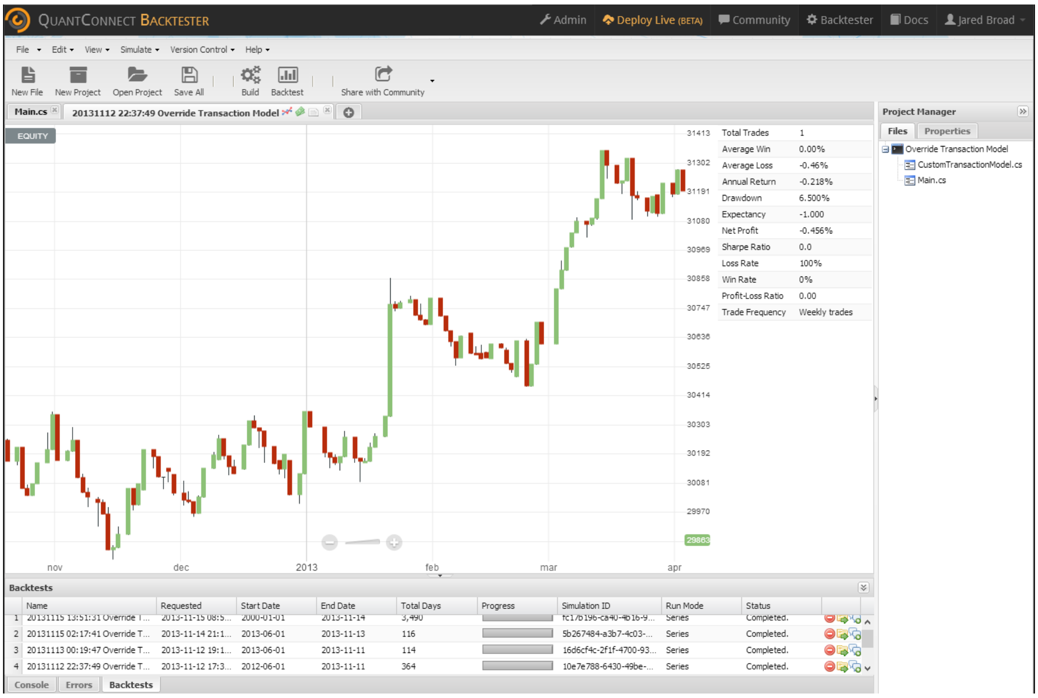


Figure 2: Result of the data analysis is displayed through a rich graphical UI.

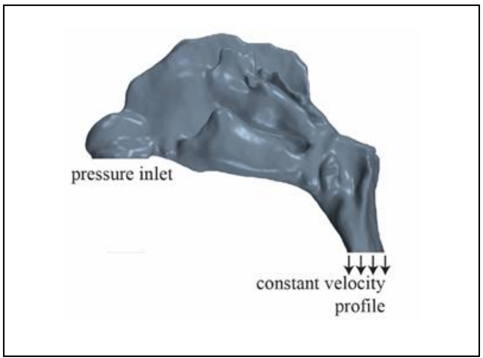
REFERENCE

[1] https://www.quantconnect.com/blog/algorithm-sharing-private-groups-and-dll-upload-support/)

Case Study Author – Jared Broad

Siemens STAR-CCM+

CFD Simulation of Airflow within a Nasal Cavity



> *"Relatively small wall clock time necessary for simulation of one nasal cavity is very promising since it allows short rent times for cloud-based machines as well as software licenses."*

MEET THE TEAM

End Users, Team Experts – Jan Bruening, Dr. Leonid Goubergrits, Dr. Thomas Hildebrandt, Charité University Hospital Berlin, Germany
Software Provider – CD-Adapco providing STAR-CCM+
Resource Provider – Microsoft Azure with UberCloud STAR-CCM+ software container
Technology Experts – Fethican Coskuner, Hilal Zitouni, and Baris Inaloz, UberCloud Inc.

USE CASE

In-vivo assessment of nasal breathing and function is limited due to the geometry of the human nasal cavity, which features several tortuous and narrow gaps. While spatially well resolved, investigation of that complex geometry is possible due to sophisticated methods like x-ray computed tomography (CT) and acoustic rhinometry, there is no sufficient method for assessment of the nasal airflow as of yet. The current gold-standard for objective assessment of nasal function is the rhinomanometry, which allows measurement of the nasal resistance by measuring the pressure drop as well as the volume flow rate for each side of the nose. Thus, a complete characteristics curve for each side of the nose can be obtained.

While high total nasal resistance measured using rhinomanometry correlates well with perceived impairment of nasal breathing, indications may be faulty in some cases. This is caused by several reasons. Firstly, there is no lower limit of "healthy" nasal resistance. In patients featuring a very low nasal resistance, rhinomanometry would always indicate no impaired nasal breathing. However, conditions that feature low nasal resistances as well as heavy impairment of perceived nasal breathing exist (e.g. Empty Nose Syndrome). Furthermore, rhinomanometric measurements allow no spatially-resolved insight on nasal airflow and resistances. It is impossible to determine which region of the nasal cavity poses the highest resistance. This may be the main reason that the role of Computational Fluid Dynamics (CFD) for assessment and understanding of nasal breathing was rapidly increasing within the last years.

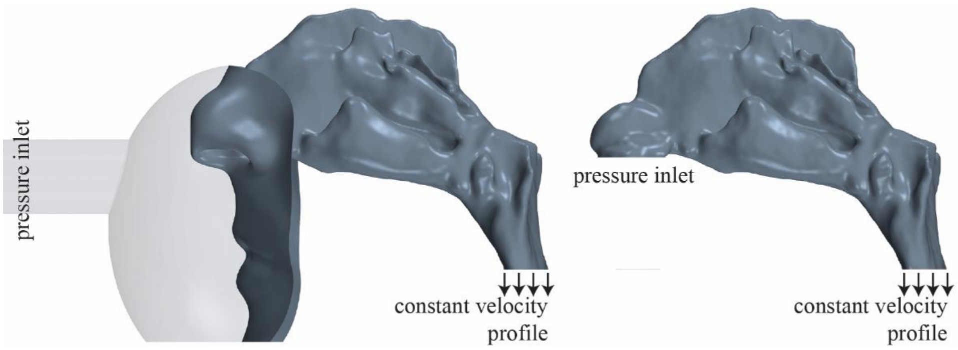
In this study the airflow within a nasal cavity of a patient without impaired nasal breathing was simulated. Since information about necessary mesh resolutions found in the literature vary broadly (1 to 8 million cells) a mesh independence study was performed. Additionally, two different inflow models were tested. However, the main focus of this study was the usability of cloud based high performance computing for numerical assessment of nasal breathing.

Methods
The geometry of the nasal cavity was segmented from CT slice images with a nearly isotropic resolution of 0.263 x 0.263 x 0.263 mm³. The segmentation was performed mostly manually using radio density thresholds. The rough geometry of the segmented nasal cavity was then smoothed and cut at the nostrils as well as the pharynx at height of the larynx.

The truncation at the nostrils and thus neglecting the ambient surrounding of the face is common practice in numerical assessment of nasal breathing. No severe differences in numerically calculated pressure drops and wall shear stress distributions were found when including the ambient compared to geometries truncated at the nostrils. However, no numerical study has been performed yet investigating the change in intranasal airflow while wearing a mask, as it is necessary during the rhinomanometric measurements. Therefore an additional geometry was created, where an oval shaped mask with an outflow nozzle with a diameter of 22 mm was created as well. Therefore, flow differences caused by those two inflow conditions could be evaluated. The mesh independency study was only performed for the truncated models.

Finite Volume meshes were created using Star-CCM+ (v. 11.02). Surfaces were meshed using the *Surface Remesher* option. Different *Base Sizes (1.6 mm, 1.2 mm, 0.8 mm, 0.6 mm, 0.4 mm, 0.3 mm, 0.2 mm)* were used to generate numerical meshes of varying resolution for the mesh independency study. For every *Base Size one mesh featuring a Prism Layer* and one without such a *Prism Layer* was created. The *Prism Layer* consisted of 3 layers with the first layer's height being 0.08 m. Each consecutive layer's height was then 1.2 times the height of the previous layer, resulting in a total *Prism Layer* thickness of 0.29 mm. Thus 14 meshes were created for the mesh independency study.

Steady state simulations of restful inspiration were performed. A negative, constant velocity equaling a volume flow rate of 12 l/min (200 ml/s) at the pharynx was specified. Both nostrils were specified as pressure outlets. Using these boundary conditions, different resistances of the left and right nasal cavity could be taken into consideration. The volume flow rate passing through each side of the nasal cavity would be defined by those resistances. It was not to be expected, that velocities within the nasal cavity would exceed a magnitude of 10 m/s. Thus, Mach numbers were below 0.1 and the inspired air could be modelled as incompressible medium. No developed turbulence can be observed within the nose during restful breathing. However, transitional turbulent regions can be found. To take those transitional effects into account a k-omega-SST turbulence model with a low turbulence intensity of two percent was used. Simulations were considered converged if residuals of continuity and all momentums were below 1.0e-5.



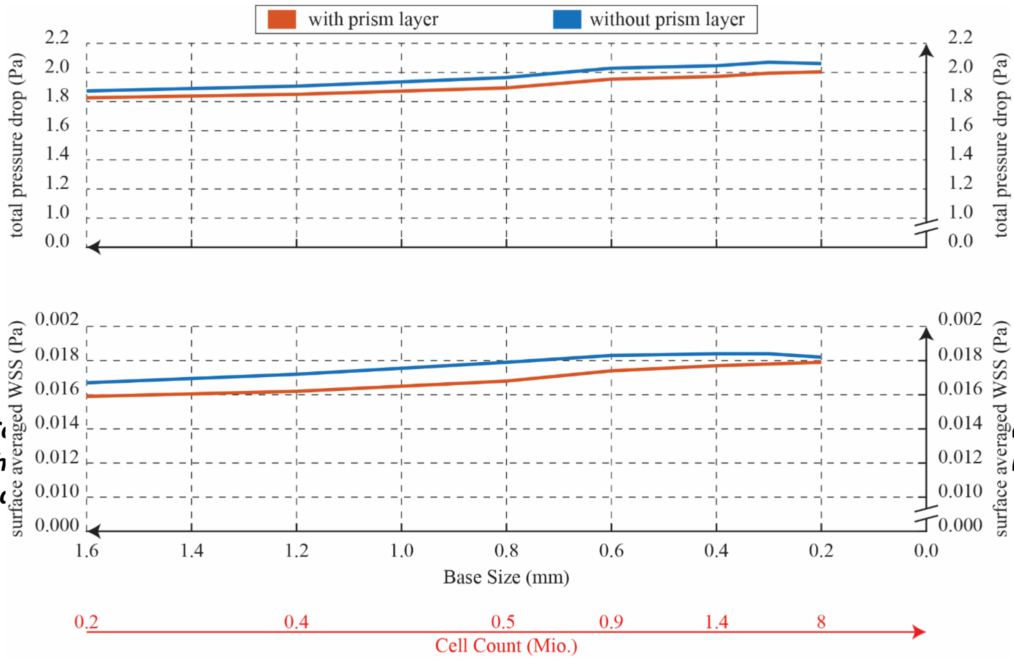


Figure 1: C ***ask***
was attach ***is***
truncated

Figure 2: Total pressure drop (upper panel) and surface averaged wall shear stress (lower panel) calculated on all meshes created for the mesh independency study. The meshes' base size decreases from left to right, while the meshes' cell count increases simultaneously. The total pressure drop as well as the surface averaged wall shear stress increases with an increasing cell count, while meshes featuring a prism layer always resulted in lower values than meshes without a prism layer adjacent to the wall.

Results – Mesh Independency

To determine the resolution sufficient for obtaining mesh independent solutions the total static pressure drop was calculated as well as the surface averaged wall shear stress at the nasal cavity's wall. The total pressure drop across the nasal cavity as function of the numerical meshes' Base Size is shown in the upper panel of Figure 2. The lower the Base Size the higher the calculated pressure drop across the nasal cavity. Calculated pressure drops are higher for meshes not featuring a Prism Layer. However, all calculated values lie within close proximity to each other. The difference between the highest and the lowest calculated pressure drop is 13 percent, while the difference between pressure drops calculated on both meshes with a Base Size of 0.2 mm is only 3 percent. Therefore, the calculated total pressure drop is insensitive to different mesh resolutions. Similar trends can be observed upon evaluation of surface averaged wall shear stress (WSS) as function of the numerical meshes' Base Size. Again, no severe differences in averaged values of wall shear stresses could be discerned.

Therefore meshes generated using a Base Size of 0.4 mm seem suited to correctly calculate integral measures as the total nasal pressure drop and thus the total nasal resistance as well as the surface averaged WSS. To ensure that not only averaged WSS values are mesh independent at a Base Size of 0.4 mm, qualitative and quantitative comparison of WSS distributions were performed. WSS values calculated on meshes with a Base Size of 0.2 mm and 0.4 mm and featuring a Prism Layer were sampled onto the original geometry obtained after segmentation. Thus, a point-wise comparison of WSS values was possible. The correlation between WSS distributions calculated using Base Size of 0.4 mm, and those using a Base Size of 0.2mm were 0.991. Even when no Prism Layer was used correlations were good (0.95).

Results – Incorporation of Ambient

Adding a simplified mask to the nasal cavity yielded in no severe differences in pressure drop. The pressure drop from mask to pharynx was 2.1 Pascal (Pa), while the pressure drop across the truncated model was 2.2 Pa. The division of airflow to both nasal cavities wasn't influenced by incorporation of the mask either. Within the simulation using the simplified mask 56 percent of the air went through the left nasal cavity. Within the truncated model 55 percent of the air went through that side of the nose.

However, WSS distributions as well as streamlines exhibit clear differences as shown in Figure 3. While positions, where high WSS (>0.1 Pa) occur, correlate well, the shape, pattern and size of these regions differ. Especially in the vicinity of the nasal isthmus, downstream of the nostrils, truncating the nasal cavity at the nostrils led to higher wall shear stresses. Incorporation of a simplified mask led to a more chaotic flow within the nasal cavity as well. While streamlines within the truncated model are smooth and perpendicular, those streamlines in the model using a simplified mask show more variations. However, both models show the classical distribution of airflow within the nasal cavity. The highest amount of air passes through the middle meatus. This can be seen within the WSS distributions as well.

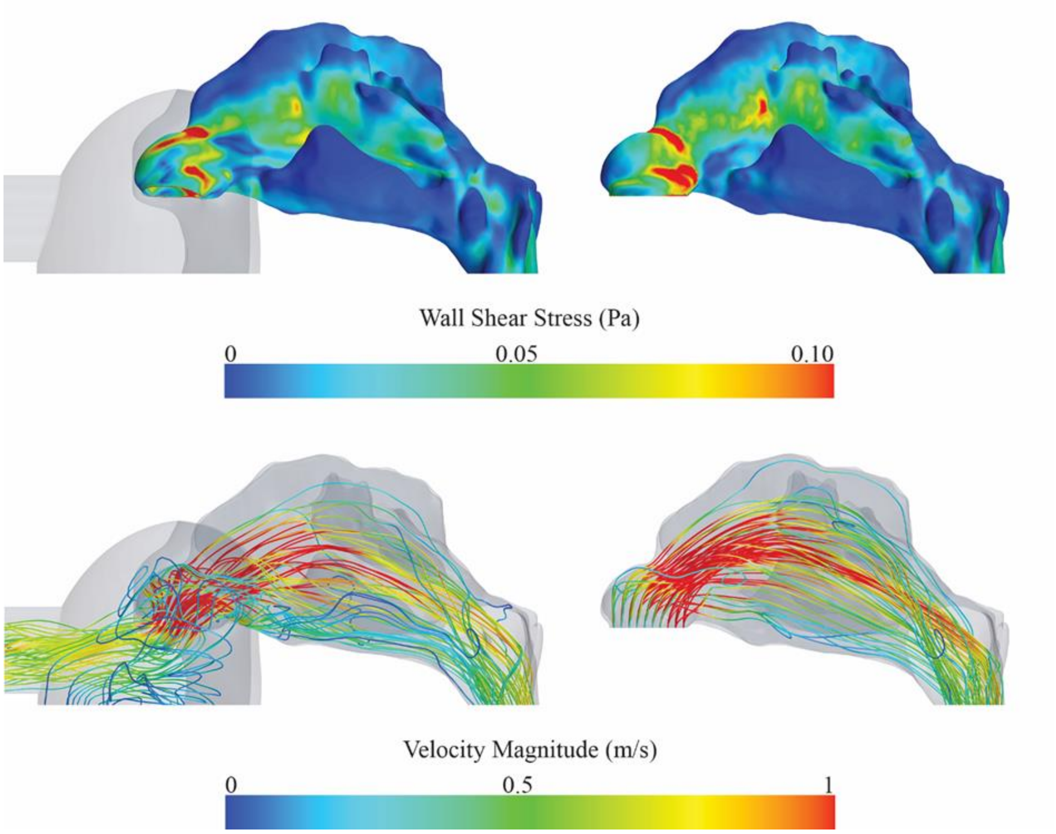


Figure 3: Wall Shear Stress distributions at the nasal cavity's wall (upper panel) and velocity information visualized using streamlines (lower panel). Those distributions are shown for simulations using a simplified mask (left) as well as for simulations, where the nasal cavity was truncated at the nostrils.

Results – Wall Clock Times and Usability of Cloud Based HPC

A dedicated machine within Microsoft's Azure Cloud was used for performing above simulations. This machine featured dual-socket Intel® Xeon® E5 processors with QDR Infiniband and RDMA technology and MPI support, allowing usage of 32 (virtual cores). Thanks to the VD-adapco STAR-CCM+ POD license provided by UberCloud, simulation of the truncated nasal geometry with highest resolution (Base Size of 0.2 mm, ca. 8 million cells) took approximately 8 hours of wall clock time. Simulation of the same geometry with the resolution shown to be sufficient within the mesh independency study (Base Size of 0.4 mm, ca. 0.9 million cells) took a little bit less than one hour of wall clock time. Therefore, within a 24 hour session, 20 or more geometries could be calculated. The simulation of the nasal cavity being attached to a simplified mask (Base Size 0.4 mm, ca. 2 million cells) couldn't be finished within the 12 hour POD time frame. However, estimated by simulation on a local machine, convergence would have been reached after approximately 6 hours of wall clock time. This relatively long duration when compared to both other simulations is due to the fact that no convergence was reached using a steady state solver, demonstrating the necessity to switch to an implicit unsteady solver using steady boundary conditions.

DISCUSSION AND BENEFITS

The overall experience using UberCloud's integrated STAR-CCM+ container environment on the Azure cloud was very convincing.

- Handling of the overall cloud environment was straight-forward and due to the whole setup being browser-based no complications regarding application software requirements occurred. Simulation files were uploaded using the author's institution's OwnCloud Service. However, an upload using Dropbox would have been possible as well, since a Dropbox client was already installed on the machine.

- Simulation speeds were overwhelmingly fast compared to the workstations the authors usually work with. Handling was pretty much the same as on a local computer. An hourly screenshot of the cloud machine's state and emailed log files allowed monitoring of the simulation state without any need to log into the cloud.

- The relatively small wall clock time necessary for simulation of one nasal cavity is very promising since it allows short rent times for cloud-based machines as well as software licenses. Thus, simulation of a patient's nasal cavity as a diagnostic tool might be performed relatively cheap. However, at this time, it is totally unknown what a good nasal airflow is and which airflow patterns and phenomena are related to impairment of nasal breathing. Within the near future, patient-specific computation of nasal airflow might become a relevant diagnostic tool within otolaryngology.

The difference between WSS and velocity distributions within the nasal cavity might indicate that additional research is necessary to better understand how wearing a mask alters the airflow. As of yet, several numerical studies including the ambient were conducted. However, all of these studies used an unobstructed ambient. This preliminary investigation about including the mask resulted in no severe change in the pressure drop across the nasal cavity. This has to be further investigated to ensure that rhinomanometric measurements, where a similar mask is worn by the patient, do not alter the airflow resistance of the nasal cavity by altering the inflow conditions.

Case Study Author – Jan Bruening, Charité Berlin, Germany

Siemens/RedCedar HEEDS

Running Bicycle Optimizations in the Cloud

> *"Trek Bicycle entrusts Rescale to provide secure, financially-competitive, and efficient computing system capable of handling even their most challenging analyses."*

MEET THE TEAM

End User – Mio Suzuki, Analysis Engineer (CFD, Wind Tunnel analysis) with Trek Bicycle Corporation, a bicycle manufacturer whose mission is to build the best bikes in the world.

Resource Provider – Ilea Graedel, Business Development Manager at cloud resource provider Rescale.

USE CASE

Multimodal design exploration is the next frontier in high performance product development. It is not trivial to overcome the challenge of simultaneously evaluating all relevant physics when creating an optimum product design. Trek intends to bring a disruptive shift to a traditional product development approach with an integration of HEEDS, Red Cedar Technology's algorithm based optimization tool. With HEEDS, it is possible to execute large number of design iterations and gain new analytical insights at an unprecedented pace. Continually evaluating large number of designs requires substantial time commitment on hardware. Rather than relying on a single local machine, it is far more efficient to leverage the vast computational power of a large cluster.

This UberCloud Experiment report summarizes how Trek Bicycle uses cloud-ready CD-adapco's STAR-CCM+ and Red Cedar Technology's HEEDS on Rescale's computing platform. This overview describes how Rescale's on-demand resource makes it easy to deploy analysis and view in-progress results. The goal of the simulation was to model drafting patterns observed in cycling racing. In some cycling disciplines, drafting is used as a race tactic to reduce the power consumption of riders. Various configuration scenarios were evaluated using a single CFD simulation file and optimization technique.

Trek uses CD-adapco's STAR-CCM+ for pre/post processing and solving CFD problems. Overset mesh technique was employed to embed four riders in the solution domain (Figure 1). In this simulation, steady-state RANS with k-ω turbulence model was used. It is not a particularly highly resolved model, but Y+ is kept in order of 1.

The use of overset mesh enables one to easily move the embedded object (individual riders, in this case) to and from a particular point in space without the need of re-meshing the entire fluid domain. Thus, for a high-count iterative process, overset mesh is an ideal technique to reduce otherwise taxing portion of the simulation process.

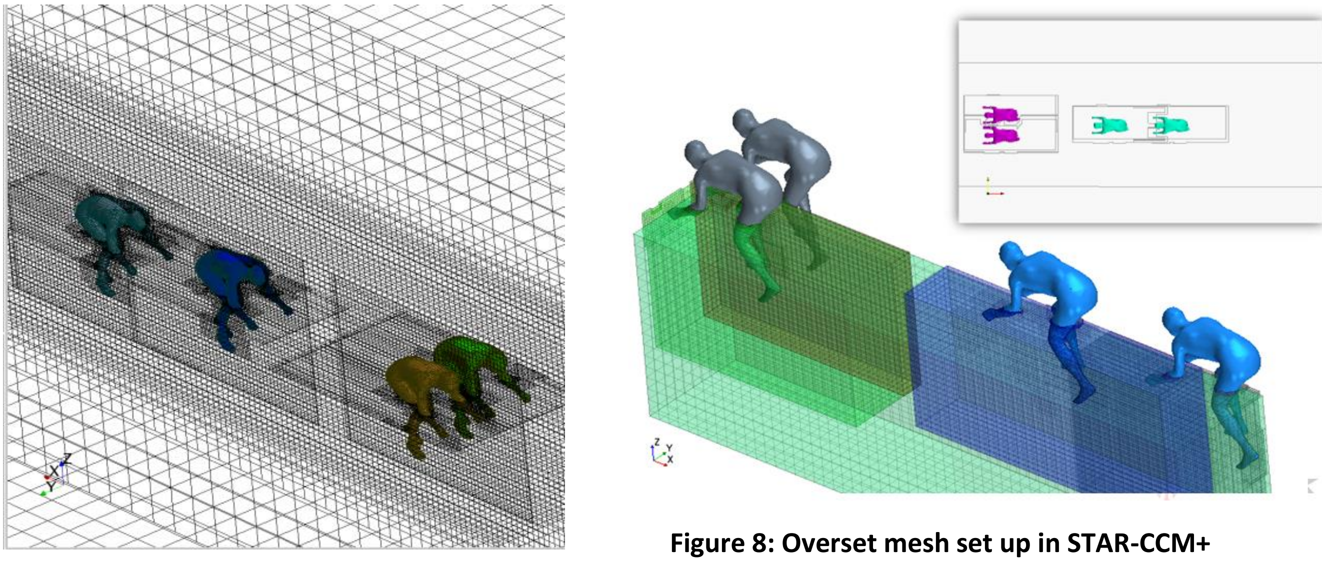


Figure 8: Overset mesh set up in STAR-CCM+

In this model a finite volume trimmer mesh was used to build the fluid volume domain. The inlet wind direction is fixed at 0 degree (head-on wind). Displacement variations of each rider from their starting positions were controlled via HEEDS. The objective of this study was to minimize the overall drag with a simple constraint on the drag. The loose objective and constraint statements were given since the main aim of this initial study was to test the HEEDS plus Rescale set up.

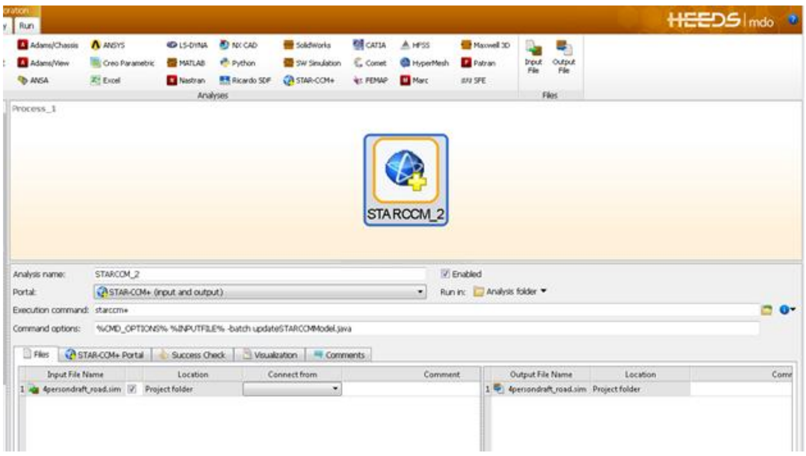


Figure 9: Screen shot of HEEDS set up

Rescale's on-demand cloud resource was used to solve the simulation and to run HEEDS. Rescale's client login is web-browser based. Everything from uploading files to executing the jobs can be done on their browser-based system, which significantly streamlines the set up for end users. One of the advantages of using browser-based system is that users don't have to be bound to a particular operating system. The guided GUI would also be ideal for users who may not be familiar with Linux/Unix command windows.

One particularly useful feature is the embedded terminal window that appears on Rescale's GUI once the submitted job starts. With few mouse clicks, users can see the progress of the simulation that is equivalent to having Linux "tail –f" output on a terminal window. Rescale's system also automatically creates a zipped version of solved CAE files and any other files that were created during post-processing so that the downloading of these files can be expedited.

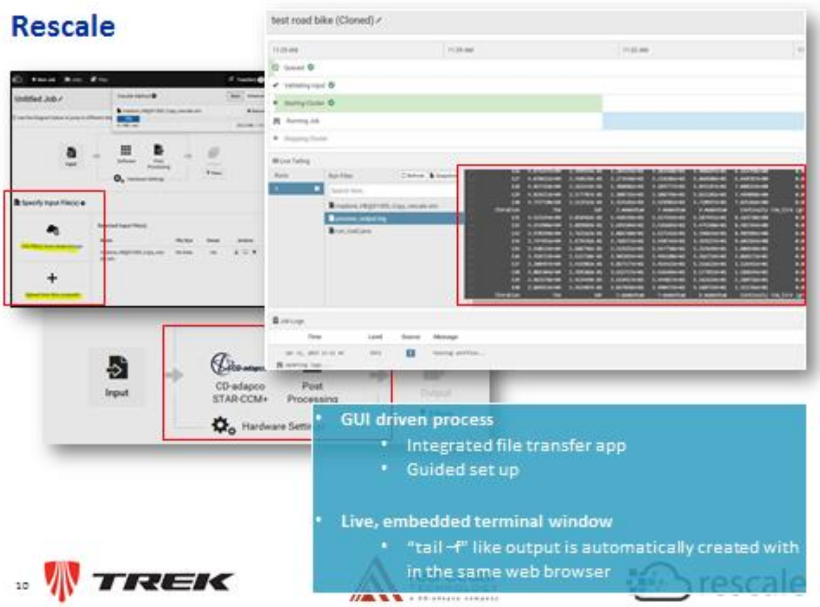


Figure 10: Rescale's web-browser user interface

For optimization, Rescale has a specific GUI to guide the process. During the solving stage, the progress of optimization can be tracked in a manner similar to that described above. Additionally, numbers from design tables can be extracted and relationship among parameter/ objective variables can assessed using 2D, 3D graphs that are available on the GUI. The enhanced progress-tracking feature adds convenience and assurance to users since even the most efficient optimization routine tends to run for a long period of time.

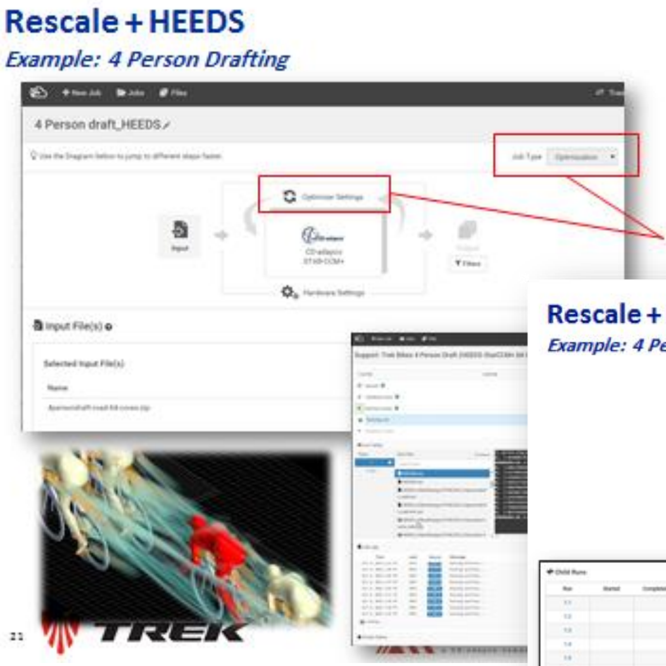
OVERCOMING LICENSE CHALLENGES

When scaling up computational demands, flexible software licensing is essential. STAR-CCM's Power-on-demand (POD) and HEEDS cloud-ready licensing models answer this user demand. POD license is cloud based, and users are free to scale up to the numbers of cores that best suits their needs.

For HEEDS, a process was needed to grab the license that is sitting at Trek's local computer. In order to do that, an appropriate virtual tunnel was set up. Although setting up of the reverse tunnel is trivial, the process may not be intuitive for new cloud users. Having a resource provider with expertise in this type of set up was tremendously helpful.

CONCLUSIONS AND RECOMMEMDATIONS

Smart merger of CAE processes in a single web-based platform is a solution for anyone looking to accelerate the pace of their R&D with instant access to amplified computational power. For Trek, it's a step in a right direction to push the boundaries of bicycling engineering even further.



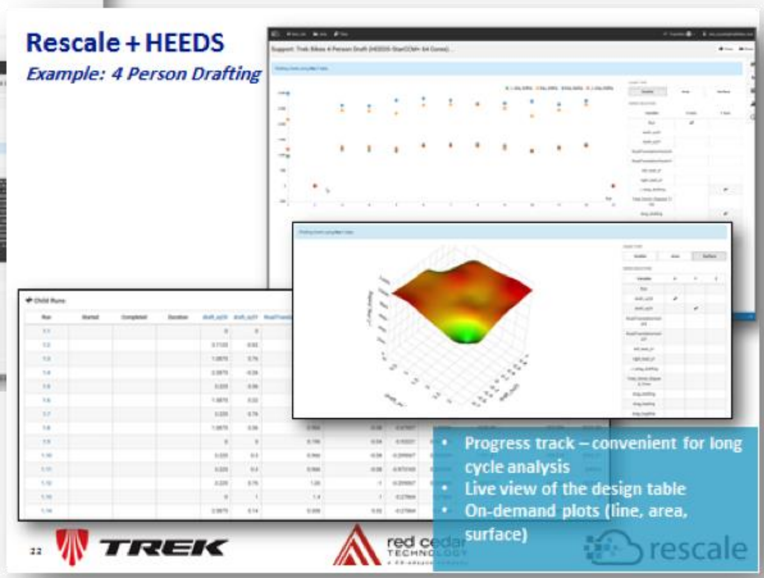


Figure 11: Rescale's optimization mode (above), and integrated graphing capabilities (right).

Accessing Rescale's simulation environment, Trek Bicycle was able to:

- Significantly reduce simulation runtime by using the latest HPC resources available in Rescale's more than 30 advanced, global data centers
- Combine simulation software tools without wasting crucial IT personnel and resources
- Execute multiple simulations in parallel to gain important runtime advantages
- Fully explore the experiment to make informed decisions about drafting techniques related to competitive bicycling
- End-to-end data encryption and private, isolated clusters ensure the highest level of job security

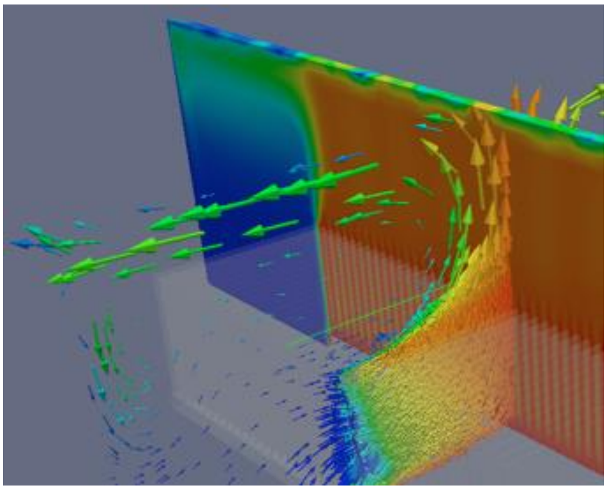
Trek Bicycle entrusts Rescale to provide a secure, financially-competitive, and efficient computing system capable of handling even their most challenging analyses.

Using existing Star-CCM+ and HEEDS licenses combined with the natively integrated solvers, Trek was able to easily couple their complex optimizations on Rescale's cloud simulation platform faster than using internal compute resources. Trek Bicycle continues to expand their simulation capabilities and further push the limits of bicycle design.

Case Study Author – Mio Suzuki

SIMSCALE CFD

Natural and Forced Convection and Thermal Management of Electronics



"SimScale's end-to-end approach hides a large part of the complexity of what it takes to run a numerical analysis code in the cloud."

MEET THE TEAM

End User and CAE Expert- John Flavin is a thermal mechanical engineer at Stored Energy Systems LLC, and an independent thermal engineering consultant. Stored Energy Systems designs and manufactures high power electronic charging systems for industrial back up power plants. CFD analysis is an integral part of the design process for cooling high power electronics.

Software and Resource Provider - David Heiny is a co-founder and head of product at SimScale. SimScale is a Germany based company that develops a web-application enabling engineering simulation within a standard web-browser. The SimScale platform, launched in August 2013 provides its user with access to different FE and FV based solvers for structural, fluid dynamics, thermal and acoustics analysis.

Team Mentor – Gregory Shirin.

USE CASE

The goal of this project was to run a thermal analysis of the cooling behavior of a heat sink for electronics cooling applications such as the Power cabin shown in Figure 1. The passive cooling device transports heat from the heat source to the surrounding air via various thin fins. Figure 2 shows the CAD model of the heat sink after being uploaded to the SimScale platform, which is used for the analysis in this project.

The SimScale platform is a generic CAE environment that runs completely within a standard web-browser. The pre- and post-processing workflow is kept agnostic to the solver that runs in the backend. Currently different Open Source codes for both Finite Element as well as Finite Volume based solvers are available. The number of available solvers and therefore the available functionality on SimScale is constantly growing. The platform is connected to large computing centers where the user can start on-demand computing instances as they are required. Two analyses have been done in this project on the SimScale platform:

- A first, simpler analysis assumes that the heat transfer coefficient is constant on all surfaces of the heat sink. It therefore neglects the local effects of the velocity field around the heat sink on the heat transfer coefficient. This analysis is done using a Finite Element based solver that solves for the temperature field within the heat sink. Figure 2 shows the resulting temperature field within the heat sink.

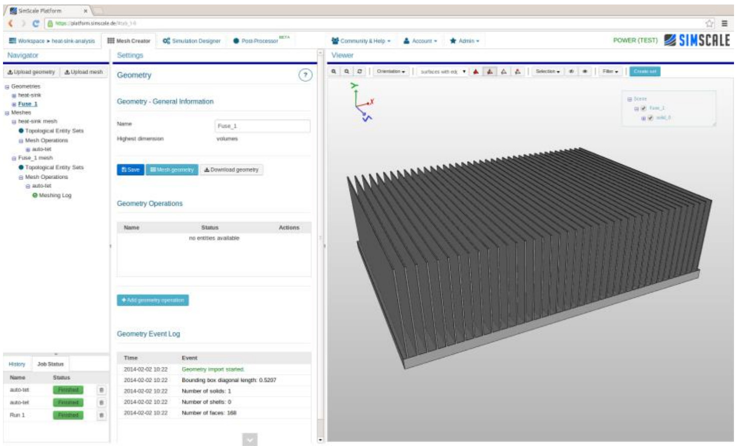


Figure 2: Heat sink uploaded to the SimScale platform.

- The second analysis was done using a Finite Volume based solver that simulates conjugate heat transfer problem and therefore solving both the temperature field within the heat sink as well as the flow field around it. Figure 3 shows the resulting temperature and velocity field in the air domain around the heat sink.

The analyses have been done using the SimScale simulation platform with both online as well as offline post-processing.

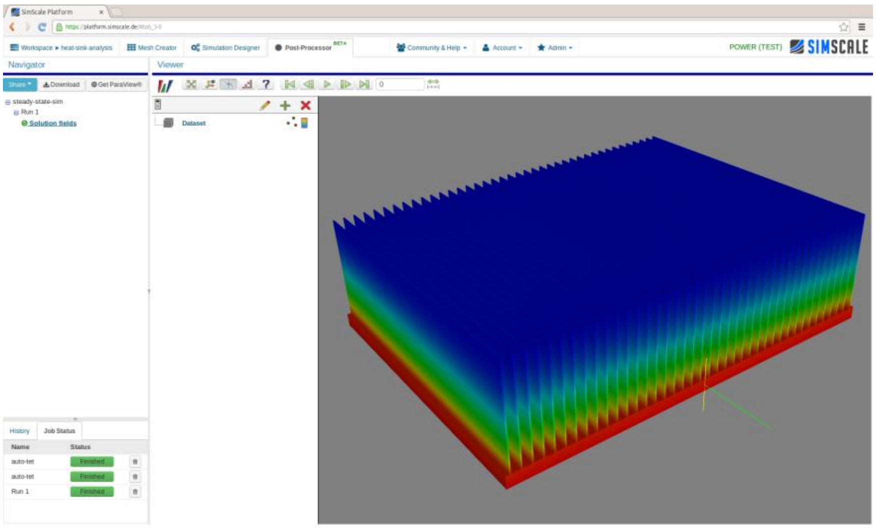


Figure 3: The temperature field of the first analysis.

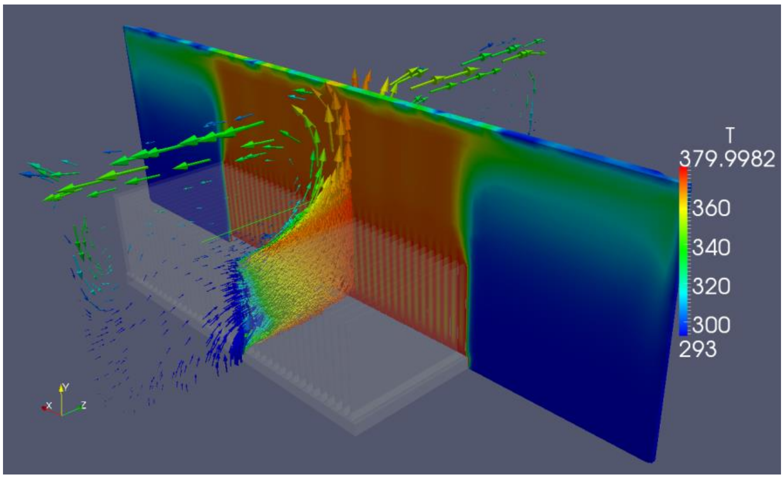


Figure 4: Visualization of the temperature and velocity field of the second analysis.

CHALLENGES

- **Mesh-generation** - The conjugate heat transfer analysis needs a multi-domain mesh which makes the mesh generation more challenging than a single-domain mesh.
- **Domain-specific functionality** -SimScale is a general purpose CAE environment and not specifically tailored for solving convection problems in electronics cooling applications. Therefore it would be great to have more special-purpose functionality in SimScale.

BENEFITS

- **Scalability -** Because the conjugate heat transfer solver is quite resource-intensive, having access to large on-demand computing power is very valuable.
- **Efficiency -** The browser-based user interface of the SimScale platform removes any license or installation issues – you can be productive in a short amount of time.
- **Generic UI -** The fact the same user interface provides access to different solver functionality provides the user with flexibility in terms of how to model a specific application.
- **Cost -** Providing a pay-as-you-go option for solving CFD analysis is a huge paradigm shift in the industry. SimScale offers a cost-effective access to scalable CFD capacity.

CONCLUSIONS AND RECOMMENDATIONS

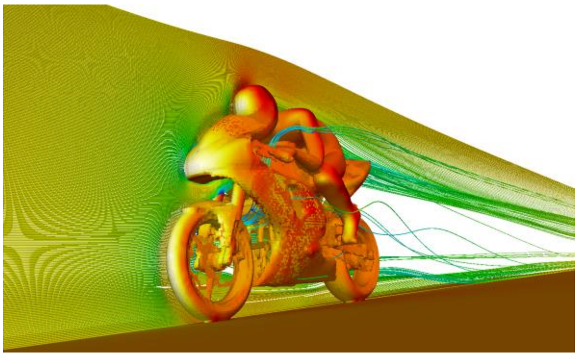
SimScale is an evolving CFD resource that offers a new way for the analysis community to access simulation capacity. The end-to-end approach hides a large part of the complexity of what it takes to run a numerical analysis code in the cloud – the typical challenges of deploying your own code to the cloud did not appear.

For special-purpose application like this project, additional capabilities – such as more boundary conditions – are necessary to ensure a convenient use of the platform.

Case Study Authors – John Flavin and David Heiny

VRATIS SpeedIT

Aerodynamic Simulations in the GPU Cloud



"Flexible licensing which is not dependent on the number of CPU cores but on the number of GPUs reduces the cost of software licensing."

MEET THE TEAM

End-User: Andrzej Kosior, Vratis Ltd.
Resource Provider: Alan Chalker, Ohio Supercomputer Center (OSC).
HPC Cloud Experts: Karen Tomko and David Hudak, OSC.
Team Mentor: Wolfgang Gentzsch, UberCloud.

USE CASE

SpeedIT Flow™ is one of the fastest Computational Fluid Dynamics (CFD) implicit single-phase flow solvers currently available. In contrary to other solutions a Semi-Implicit Method for Pressure Linked Equations (SIMPLE) and the Pressure Implicit with Operator Splitting (PISO) algorithms have been completely implemented on graphics processing unit (GPU). The software is particularly useful in acceleration of time-consuming external and internal flow simulations, such as aerodynamics studies in car and aviation industries.

SpeedIT Flow robust solver technology empowers users by providing accuracy in double precision on fully unstructured meshes up to 11 million cells. Our implementation was validated on standard steady and unsteady industry-relevant problems with RANS turbulence model. Its computational efficiency has been evaluated against the modern CPU configurations using OpenFOAM[5].

Graphics Processing Units provide completely new possibilities for significant cost savings because simulation time can be reduced on hardware that is often less expensive than server-class CPUs. Almost every PC contains a graphics card that supports either CUDA or OpenCL and SpeedIT Flow, supports both platforms. If there are multiple GPUs in the system, independent computing tasks such as in the parameter sweep studies can be solved simultaneously. When cases are solved on GPU the CPU resources are free and can be used for other tasks such as pre- and post-processing.

[5] This offering is not approved nor endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trademarks.

Three most important characteristics of GPUs affecting the performance of numerical simulations are: amount of RAM, number of streaming processors and memory bandwidth. Major GPU manufacturers compete by optimizing these characteristics and every new family of GPU cards outperforms the previous one. Server-class GPUs usually have more RAM, higher memory bandwidth and larger number of streaming processors than desktop and laptop GPUs which have worse characteristics but lower power consumption. While the latter GPUs can accelerate cases of size ca. 1-2 million cells, much larger cases can fit only to server-class cards that are quite expensive.

This is where cloud computing can be helpful. Together with the Ohio Supercomputer Center (OSC) and UberCloud we have prepared a preconfigured, tested and validated environment where SpeedIT Flow is ready to be used. The customer pays for actual consumption only. Real consumption model is particularly important for small companies and research groups with limited budget and for bigger companies who want to reduce the simulation costs. Also, this technology may be particularly interesting for HPC centers, such as OSC, because our software offers better utilization of their resources. The computations may be done on the CPUs and GPUs concurrently. Moreover the power efficiency per simulation, which is an important factor in high performance computing, is comparable for a dual-socket multicore CPU and a GPU.

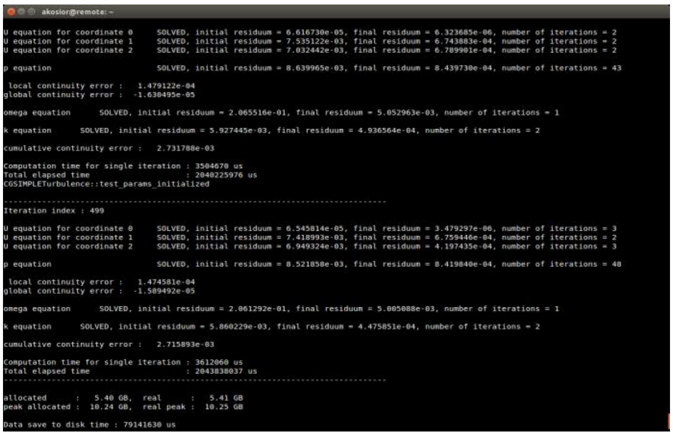


Figure 1: SpeedIT Flow is running from the command line.

Running simulations with SpeedIT Flow is easy. First, the OpenFOAM case must be converted to GPU-friendly format[6]. As this step is not computationally demanding it can be run by the user on the local computer or on the cluster just before the simulation. Next, our solver is executed from the command line and the output file is being generated in a user-friendly format and can be viewed in ParaView - an open-source, multi-platform data analysis and visualization application.

RESULTS

Four OpenFOAM test cases (Figure 2) were run on two different clusters in OSC[7], Oakley with Intel Xeon X5650 processors and Ruby with Intel Xeon E5-2670 v2 processors. Results were compared to SpeedIT Flow which was run on the Ruby using NVIDIA Tesla K40 GPU.

---

[6] Converter is available to download from https://github.com/VratisLtd/OpenFOAM_to_SpeedIT_converters

[7] Full specification can be found at https://www.osc.edu/supercomputing/hpc

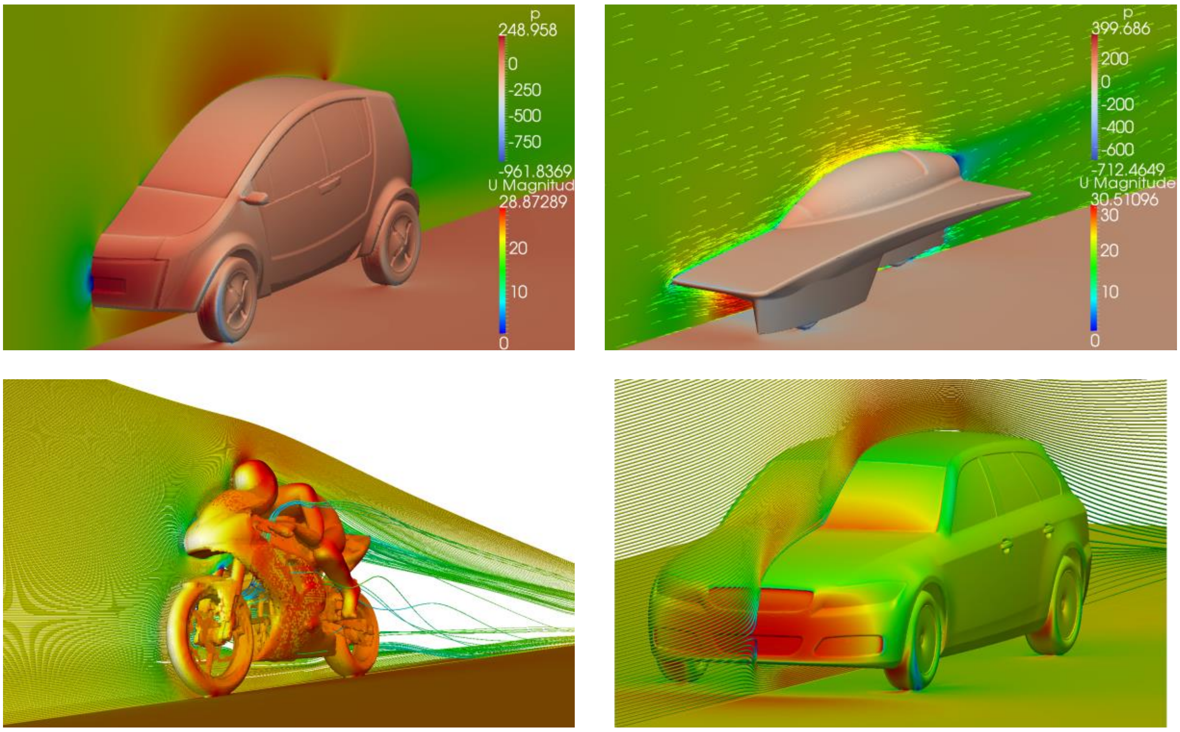


Figure 2: Test cases. From top left: AeroCar and SolarCar - geometries from 4-ID Network; motorBike - geometry from OpenFOAM tutorial; DrivAer - geometry from Institute of Aerodynamics and Fluid Mechanics at TUM. All geometry providers are kindly acknowledged.

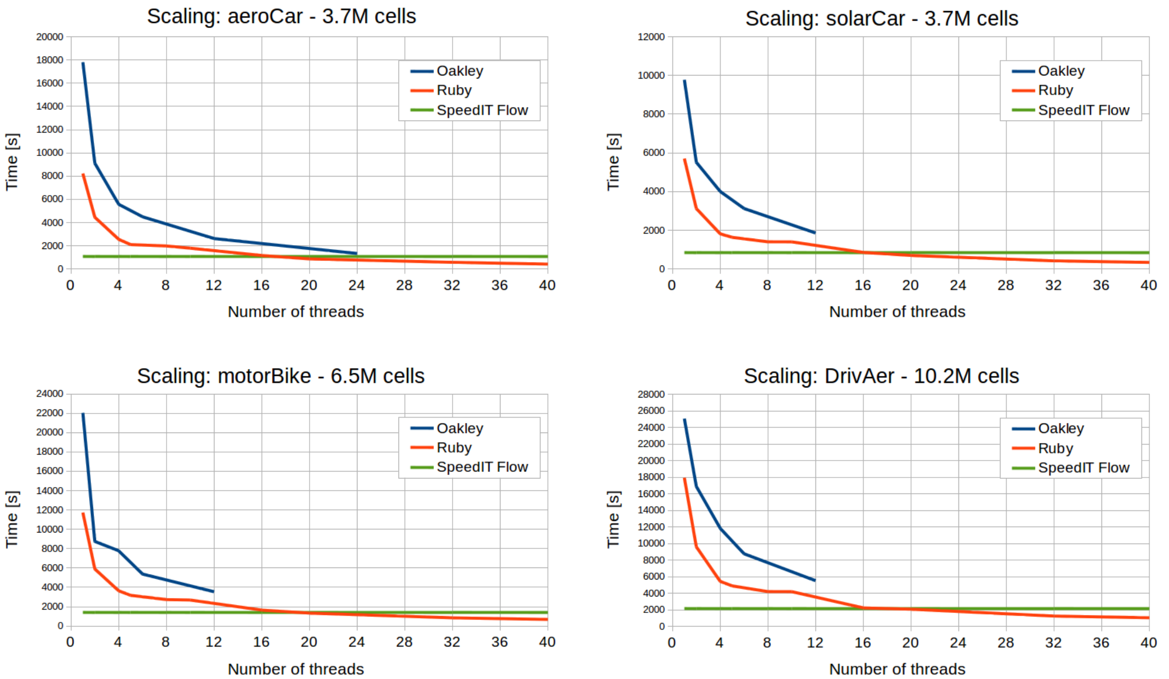


Figure 3: Scaling OpenFOAM results for different test cases on two OSC clusters (Oakley with Intel Xeon X5650 processors and Ruby with Intel Xeon E5-2670 v2 processors), and on a single GPU (NVIDIA Tesla K40), using SpeedIT Flow.

Results of OpenFOAM scaling on different number of cores are given in the Figure 3. The graphs show that SpeedIT Flow is capable of running CFD simulations on a single GPU in times comparable to those obtained using 16-20 cores of a modern server-class CPU. In Figure 4 electric energy consumption per simulation together with scaling for the biggest test case is shown. Again electric energy consumption is comparable to those needed by computations on multicore CPUs.

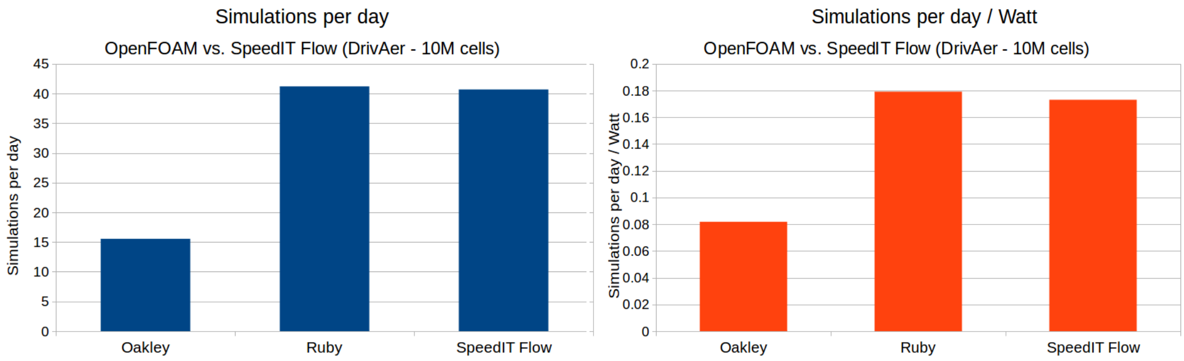


Figure 4: Number of simulations per day (left) and per day per Watt (right) of the DrivAer test case computed on a single node of Oakley (12 cores of Intel Xeon X5650) and Ruby (20 cores of Intel Xeon E5-2670 v2) using OpenFOAM and a single GPU (NVIDIA Tesla K40) using SpeedIT Flow.

CONCLUSIONS

SpeedIT Flow has been developed in the past two years as the result of projects aimed at acceleration of CFD. The biggest challenge was to implement Navier-Stokes solver in CUDA and OpenCL in the efficient way. Supported by our team of engineers and also by the GPU vendors we were able to complete this task. Once this has been done, the tests showed that our software is optimally using graphics card resources meaning that the performance of our solver will raise with every new generation of the graphic cards.

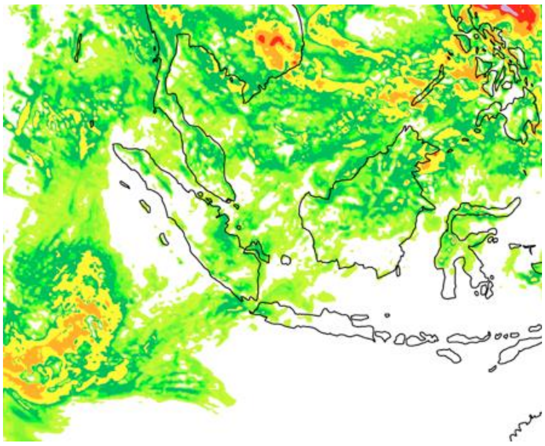
It gives the end-user an alternative way to reduce turnaround times. By taking advantage of GPUs, that are available in most of the systems, the simulation time can be reduced bringing significant cost savings to the production pipeline. Finally, flexible licensing that is not dependent on number of CPU cores but on number of GPUs reduce the costs of software licensing.

With our software resource providers such as OSC and private or public cloud providers can utilize their hardware more efficiently. On a cluster with GPUs CFD simulations can be run at the same time on CPUs as on GPUs. For example for a node with two GPUs and two CPUs with ten cores each three simulations could be run: two on GPUs and one on unused eighteen cores. As shown in our tests the turnaround times and the power consumption per simulation are comparable.

SpeedIT Flow is a new solver that completely changes the paradigm of running CFD calculations. It is an attractive solution for both individual users and HPC centers. It can be also an alternative to new investments in hardware. As the computations on GPUs are comparable to the once on the modern CPUs instead of buying a new CPU-based system the end-user could just furnish an old one with GPUs. This solution should be cheaper than and as effective as the new hardware.

Case Study Author – Andrzej Kosior, Vratis Ltd.

WRF
Weather Research and Forecasting: Performance and Evaluation of WRF



"The primary benefit of participating in the HPC Experiment is that we were able to evaluate the cluster compute instances in AWS."

MEET THE TEAM

End Users - The beneficiary of this research is Temasek Marine Sciences Institute, National University of Singapore. The experiments were conducted by students and faculty from Department of Electrical and Computer Engineering, NUS and the Institute for Infocomm Research, A*STAR, Singapore.

Software Providers - Open source WRF software from UCAR

Resource Provider - Amazon Web Services (AWS)

HPC Expert - Prof. Bharadwaj Veeravali, Department of Electrical and Computer Engineering, NUS and Institute for Infocomm Research, A*STAR, Singapore.

USE CASE

In this experiment, we attempted to evaluate the performance of WRF (Weather Research and Forecasting) open source software on cloud services provided by Amazon Web Services. The WRF software is currently setup to run on a Beowulf-style computing cluster comprised of 12 nodes, each node featuring an 8-core CPU. The cluster is used 24x7. Execution time is 24 hours for a 12 hour weather prediction cycle.

Experiments – We used an 8 node cluster of AWS CC1.4xlarge instances located within a placement group.

CPUs:	2 x Intel Xeon 5570
RAM:	23 GB
Memory Topology: Memory Topology, cont'd:	Socket L#0 + L3 L#0 (8192KB) L2 L#0 (256KB) + L1 L#0 (32KB) + Core L#0 PU L#0 (P#0) PU L#1 (P#8) L2 L#1 (256KB) + L1 L#1 (32KB) + Core L#1 PU L#2 (P#1) PU L#3 (P#9)

	L2 L#2 (256KB) + L1 L#2 (32KB) + Core L#2 PU L#4 (P#2) PU L#5 (P#10) L2 L#3 (256KB) + L1 L#3 (32KB) + Core L#3 PU L#6 (P#3) PU L#7 (P#11) Socket L#1 + L3 L#1 (8192KB) L2 L#4 (256KB) + L1 L#4 (32KB) + Core L#4 PU L#8 (P#4) PU L#9 (P#12) L2 L#5 (256KB) + L1 L#5 (32KB) + Core L#5 PU L#10 (P#5) PU L#11 (P#13) L2 L#6 (256KB) + L1 L#6 (32KB) + Core L#6 PU L#12 (P#6) PU L#13 (P#14) L2 L#7 (256KB) + L1 L#7 (32KB) + Core L#7 PU L#14 (P#7) PU L#15 (P#15) HostBridge L#0 PCI 8086:7010 Block L#0 "sda" PCI 1013:00b8

Table 1: Details of AWS CC1.4xlarge instances.

End-to-end process – Since the goal of this experiment was to analyze the performance of WRF on different configurations of a cloud based computer cluster, we set up an 8 node CC1.4xlarge compute cluster within a placement group at AWS. The installation of WRF was fairly straightforward as it is open source and relatively well documented. We decided to use a MPICH2 cluster configuration to run the jobs in parallel.

The experiments were setup to run up to 16 processes on a single instance and run up to 8 parallel instances/nodes yielding a total of 128 parallel processes. Afterword, the experiments were repeated by using the CPU core pinning option of the hydra MPI manager. Both the best case of all physical cores and worst case of physical/virtual core pairs were documented.

Results - We extracted the performance characteristics of executing WRF on varying number of cluster sizes - from 1 instance to 8 instances, and varied the number of threads/processes in each instance. In general, we observed that the same number of threads running on greater number of nodes had better performance than running on fewer nodes. Figures 2 - 5 show the varying performance of multiple threads in 1, 2, 4 and 8 instances.

The trend of each chart has been consistent whereby threads that run on physical/virtual cores have highest run time regardless of number of instance used. Besides, the practice of core pinning does achieve better performance in comparison to default allocation as the number of thread increases. Further increase of number of threads beyond 32 threads does not reduce the runtimes of WRF.

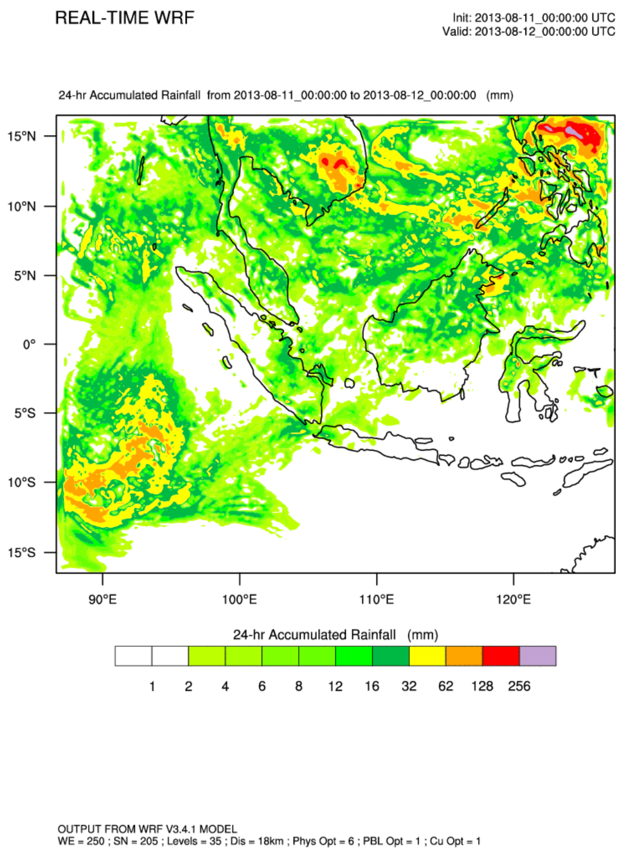


Figure 1: WRF simulation showing 24-hr accumulated rainfall for the Singapore region.

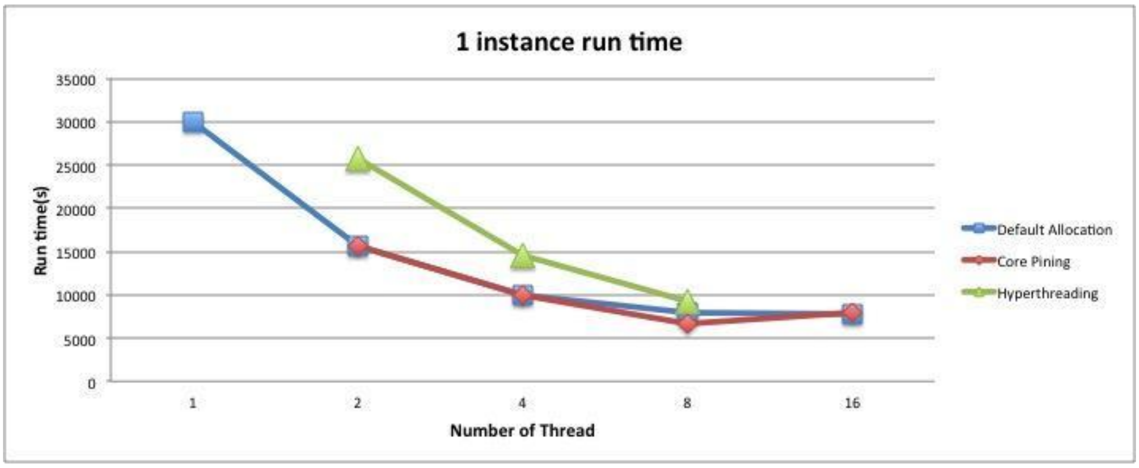


Figure 2: WRF runtimes Vs threads in 1 instance cluster using multiple pinning options.

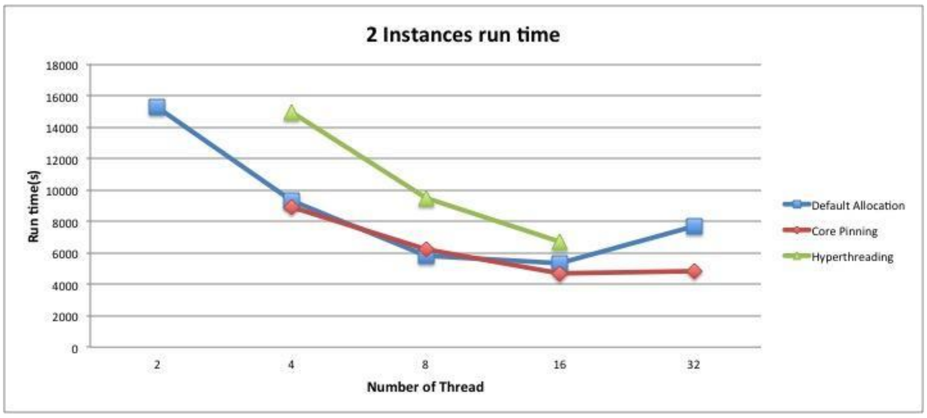


Figure 3: WRF runtimes Vs threads in 2 instance cluster using multiple pinning options.

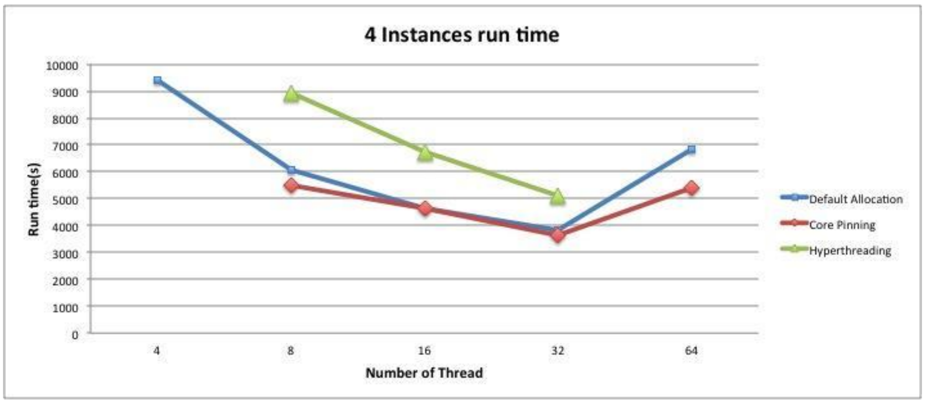


Figure 4: WRF runtimes Vs threads in 4 instance cluster using multiple pinning options.

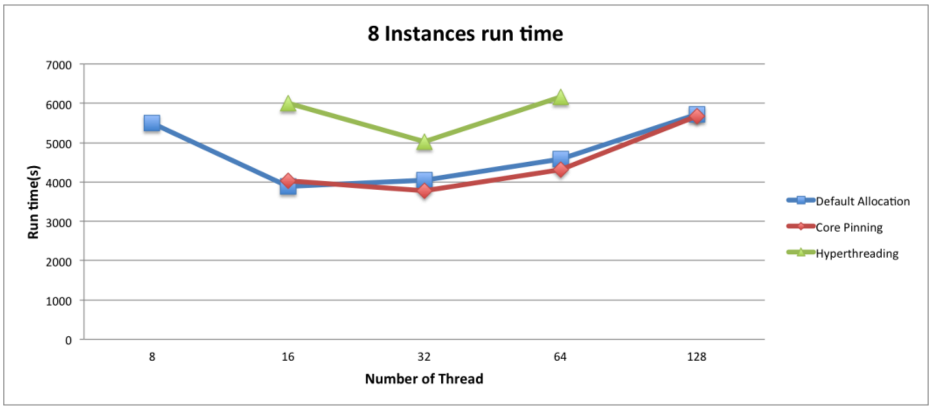


Figure 5: WRF runtimes Vs threads in 8 instance cluster using multiple pinning options.

CHALLENGES

In order to save costs, we implemented a scheme of shutting down the instances when they finished one set of experiments. On restarting the instances, the IP addresses assigned to them were changed by the resource provider. This meant that the cluster had to be reconfigured frequently.

More importantly, this meant that the number of network hops was not a constant. Specifically, if two nodes are in the same network subnet then it is 2 hops (node - to - network switch - to - node). If the same two nodes, on reboot, end up on two different network subnets then it means 4 hops (node - to - network switch - to - router - to 2nd network switch - to node). Figure 6 shows these IP variations. It indicates that there is no pattern in the allocations although all of the nodes are within the same placement group in the AWS computer cluster.

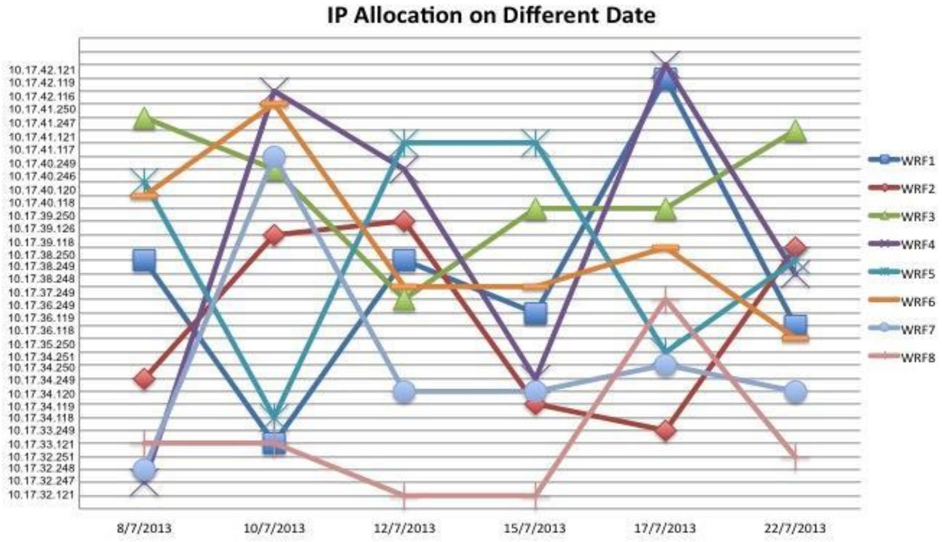


Figure 6: IP assignment variations on instance reboots.

BENEFITS

We were able to evaluate the cluster compute instances in AWS. Technically, the main benefit was to determine the optimum number of instances to use for WRF for a given data set size. Specifically, given a sufficiently high bandwidth, was there is a performance improvement when increasing the number of nodes across which the software was executed? Also, we learned that the ease with which the cluster size on the cloud can be expanded is remarkable, and is of immense use for dealing with performance bottlenecks.

CONCLUSION AND RECOMMENDATIONS

Core pinning is essential because the probability of running threads on virtual cores is higher when the number of threads used is high. We found that running WRF in hyper-threading results in longer run times – therefore, it is undesirable to have virtual cores running WRF. Also, providing additional threads does not speed up compute time. Instead it increases communication latency between the additional cores because the decomposition of the WRF workload is so small that all the threads are running at low utilization.

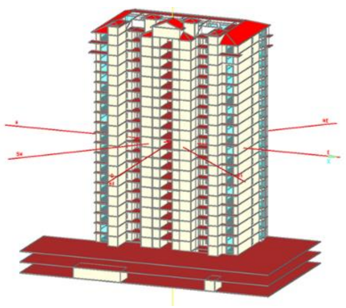
REFERENCES

- WRF Model - How to Compile: http://www.mmm.ucar.edu/wrf/users/wrfv2/install.html
- Getting Started with Amazon EC2 Linux Instances: *Amazon Elastic Compute Cloud*. http://docs.aws.amazon.com:AWSEC2:latest:UserGuide:EC2_GetStarted.html
- Using the Hydra Processing Manager:
- http://wiki.mpich.org/mpich/index.php/Using_the_Hydra_Process_Manager
- ARSC WRF Benchmarking Suite: http://weather.arsc.edu/BenchmarkSuite/

Case Study Author – Prof. Bharadwaj Veeravali

WUFI Plus

Modelling Moisture and Moisture Transfer Within a Residential Condominium Tower in the Microsoft Azure Cloud



"This study also demonstrates that, in general, multi-physics building simulation using HPC Cloud computing is accessible, affordable, and beneficial for our private clients."

1 MEET THE TEAM

End-User: David Pursiano and Robert Simon, Civil Litigation Law Firm PURSIANO BARRY BRUCE LAVELLE LLP, USA
Software Provider: Florian Antretter and Matthias Pazold, Fraunhofer Institute for Building Physics (IBP), Germany
Resource Provider: Microsoft Azure
HPC Expert: Matthias Pazold, IBP, Baris Inaloz and Wolfgang Gentzsch, UberCloud.

2 USE CASE

PURSIANO BARRY BRUCE LAVELLE LLP is a multi-state civil litigation law firm in the US with its main areas of practice in Construction Defect, Product Liability, Eminent Domain and Storm Damage. The subject property is a twin tower (15 and 20 stories), 283-unit residential condominium constructed of cast in place concrete construction with post tension concrete floor slabs and Concrete Masonry Units (CMU) and DensGlass infilled walls. An extensive expert investigation established, among other issues, exterior plaster failure, water intrusion at improper window and roof installations, and high interior humidity levels with apparent biological growth (ABG) observed on interior walls, baseboards and between layers of interior gypsum board at unit partitions. ABG was also observed on interior furnishings within the condominiums. All of the ABG was observed away from obvious sources of water intrusion.

Mechanical testing determined that condominium unit interiors were often under negative pressure, drawing in high-humidity, un-conditioned exterior air. A mechanical engineering evaluation found that the air conditioning units (serving each condominium) were improperly sized to adequately manage humidity.

It was proffered by competing experts that defective exterior plaster was a cause of the high humidity conditions. The plaster was directly applied to the cast in place concrete and CMU block infill. The coating of the plaster was a textured acrylic elastomeric paint. Exterior plaster failure included blistering of the coating system and saponification of the coating and substrate. Moreover, these experts opined that the existing mechanical system for each condominium unit was adequate to handle the dehumidification such that the ABG was caused by the moisture intrusion through the exterior wall system. Their opinion was the negative air pressure was acceptable for 15/20 story towers.

Because of these competing opinions and the inability to field test the experts' hypotheses, PURSIANO BARRY BRUCE LAVELLE LLP chose HPC modelling to determine whether moisture transfer was occurring through the plaster coated exterior walls or the result of negative pressure in the living units. Also, by advancing the modelling, HPC was used to determine the effect of the negative pressure and high humidity in the condominium living environment if left unmitigated.

3 SOFTWARE AND SERVICE PROVIDER: Fraunhofer Institute for Building Physics

3.1 Service Provider: The Fraunhofer Institute for Building Physics and the WUFI® Plus Software

The Fraunhofer Institute for Building Physics (IBP) is a German based applied research institute. The primary focus of the Fraunhofer IBP's work is on research, development, testing, demonstration and consulting in the various specialist areas of building physics.

The WUFI Plus simulation environment is a hygrothermal building simulation software, part of the world's leading building related hygrothermal simulation suite WUFI. The tools of the WUFI family are based on the calculation of the coupled heat and moisture transport across building components, like walls, roofs and floors. They simulate the temporal development of the heat and moisture profiles within a component and the heat and moisture exchange on the component surface. Combining simulations for all components of the building envelope and inner walls with a multi-zone model leads to the hygrothermal building simulation tool, called WUFI® Plus. In addition to the heat and moisture transport across the components, solar radiation through windows, inner heat and moisture sources or sinks, HVAC systems and ventilation are considered. The indoor climate within the zones is calculated with a heat and moisture balance.

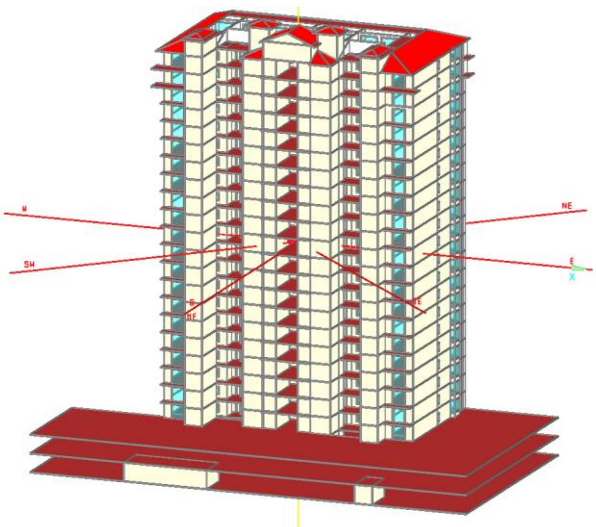


Figure 1: Visualization of the residential condominium tower building model in WUFI® Plus.

4 MODELLING OF THE CONDOMINIUM TOWERS

Modeled hygrothermal processes in the building enclosure

The hygrothermal building component simulation model in WUFI does take the liquid and vapor water transport in building components into account. A material dependent water vapor diffusion resistance factor allows computing the vapor diffusion and solution diffusion. The liquid transport mechanisms capillary conduction and surface diffusion are described by water content dependent liquid transport coefficients for suction and redistribution for each material layer of the building assemblies.

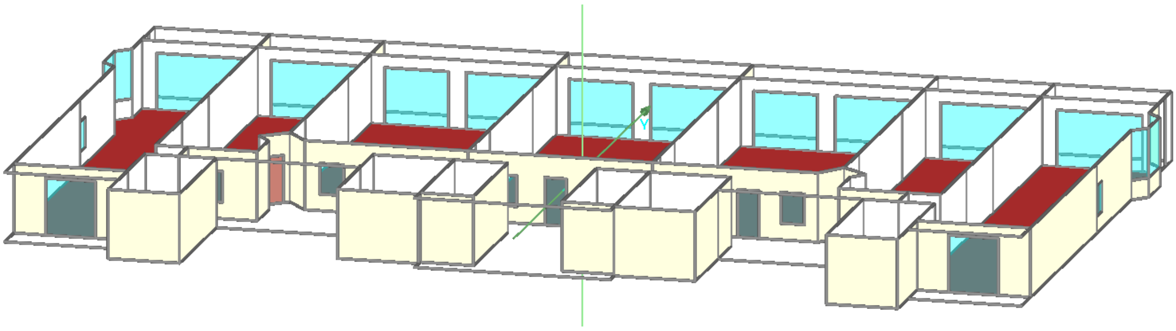


Figure 2: Three-dimensional visualization of the investigated hygrothermal simulation building model.

Preliminary simulation

Preliminary simulation studies enable the assessment of the required degree of detail and the involved uncertainty. They serve to identify the main factors with a reasonable effect on the results including their parametric range to investigate in the next step.

The parameter investigation covers different construction periods to determine the initial moisture content in the building components and in the materials. Each simulation is split in 5 periods: the first period covers the time without any exterior façade finish and results in detailed information about dry-out of built-in moisture. The second period is past the exterior finish on the north elevation, the third period is past finishing the exterior coating on the other walls. In the fourth period the mechanical systems were activated, including the air handling units (cooling and dehumidification) and inner heat and moisture sources were added due to interior construction and finish. The last period starts with the beginning of the usage of the tower, with occupancy and normal operation air handling units.

PARAMETRIC STUDY

Simulation model

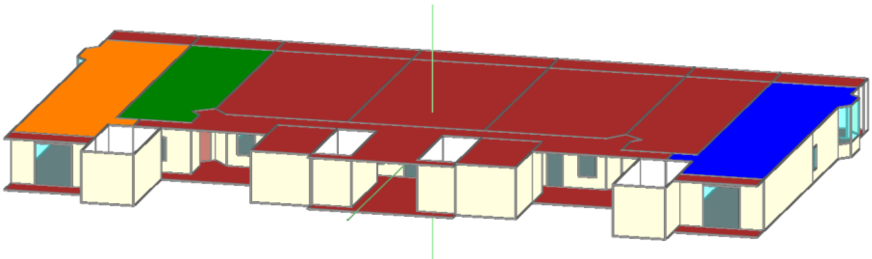


Figure 3: Investigated building model.

The simulation model simulates 3 individual units in the residential condominium tower, as indicated in Figure 3 in orange, green and blue. For each unit the inner climate is simulated as a result of heat and moisture flows via the building components, inner sources and sinks, via ventilation, and the HVAC system. The count of simulated building components:

Unit 1: 1 ceiling; 1 floor; 10 exterior components; 4 interior components; 6 windows; 1 door
Unit 2: 1 ceiling; 1 floor; 5 exterior components; 2 (new) interior components; 2 windows; 1 door
Unit 3: 1 ceiling; 1 floor; 10 exterior components; 4 interior components; 6 windows; 1 door.

In total, 41 “WUFI” opaque components, with up to 7 layers (different building materials) are simulated with the coupled heat moisture transfer algorithm. 17 transparent components (windows) are calculated. For each component detailed results, like water content in the different component materials and surface conditions (surface temperature and relative humidity) are investigated.

Simulation computation time test

With four different compute instances in the Microsoft Azure cloud, a benchmark was done to check the computation times and identify the 'sweet spot' system configuration. Tested were the DS11 (2 cores), DS12 (4 cores), DS13 (8 cores) and DS14 (16 cores) compute instances. The simulation model described above was used. The simulation period for the building model was set to one year. Table 1 shows the elapsed computation time on the different machines.

Table 1: Simulation speed test results.

Simulations		DS11 (2core)	DS12 (4core)	DS13 (8core)	DS14 (16core)	
1		**0:25:10**	0:16:35	0:11:14	0:06:03	
2	simultaneous		0:24:45	0:17:10	0:08:42	elapsed time
			0:12:22	*0:08:35*	*0:04:21*	deduced per simulation
4	simultaneous		(>>25min)	0:34:17	0:16:00	elapsed time
				0:08:34	0:04:00	deduced per simulation
6	simultaneous			*0:56:00*	0:21:07	Elapsed time
				0:09:20	**0:03:31**	deduced per simulation
8	simultaneous				0:25:25	elapsed time
					0:03:11	deduced per simulation
Elapsed time per simulation		00:25:10	00:12:22	0:08:34	00:03:11	

Interpretation of the computation times

The DS11 takes 25 minutes and 10 seconds to simulate one year of the building. The CPU usage was nearly 100%. With 4 cores (DS12) the simulation of the same model needs 16 minutes and 35 seconds. During the simulation of one building model the CPU usage was lower than 100%. Due to this, the simulation was started twice at the same (two simultaneous simulations), to get 100% CPU usage. Both simulations were finished in 24 minutes and 45 seconds. Regarding this and running two simulations at the same time on the DS12 machine it can be concluded that the simulation of one year lasts 12 minutes and 22 seconds on the 4-core machine. Running 4 simulations at the same time (parallel) on the DS12 lasts much longer than 25 minutes, so this configuration does not result in a time benefit.

With DS14 the test was done running the simulation up to 8 times in parallel. Those 8 simulations running in parallel were finished in 25 minutes and 25 seconds. Concluding this, a simulation of one year takes 3 minutes and 11 seconds on the 16-core machine.

Finally, for the parameter study the DS14 with 16 cores was chosen, and set to run 8 simulations at the same time. Because parameter studies consist of independent simulations, we were able to run 16 simulations in parallel on two DS14, or 32 simulations on four DS14, thus reducing the total simulation time from more than one month on the user's workstation to less than two days!

Results

The parametric study resulted in 583 simulations investigating the building behavior depending on modified conditions. With different count of simulated years per simulation (3, 5 or 8) in total 2790 years are simulated. Nearly 36 GB zipped data are stored, containing "the main" simulation results, like the air and material temperatures and humidity, and the material water content, for each hour in each simulated year.

The results in Figure 5 show the coating failure. At a certain value the risk increases. The value is different for perfect workmanship and potential leakage.

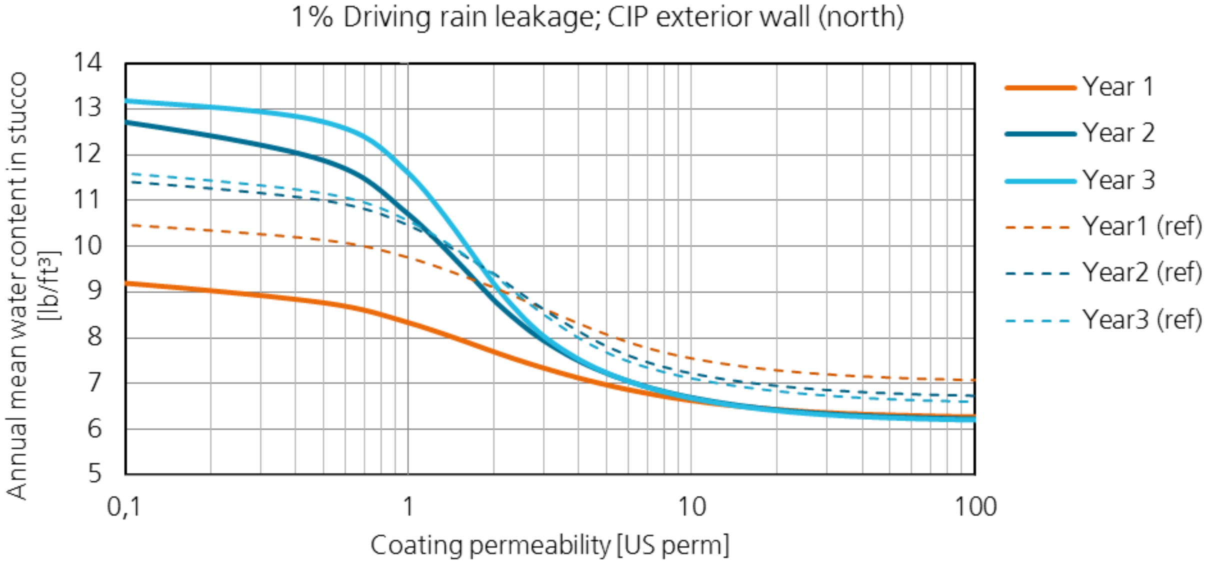


Figure 5: Annual mean water content in Stucco compared to reference case; concrete masonry units (CMU) exterior wall facing north; with 1% driving rain leakage behind coating (non-perfect workmanship). CIP = Cast-in-place concrete.

CONCLUSION

For the specific project it was concluded, that the coating damages are not related to high indoor air relative humidity but to the vapor permeability of the coating and the driving rain leakage behind coating. Indoor climate humidity and mold issues are not related to coating permeability but to HVAC (heating, ventilation and air conditioning) equipment and infiltration of unconditioned and untreated outside air. The impact of single parameters on coating damages, elevated indoor humidity and mold issues can be assessed only with large scale parametric simulation. The simulation models can be used to predict the potential future performance and related risk for renovation measures.

This study also demonstrates that, in general, multi-physics building simulation using HPC Cloud computing is accessible, affordable, and beneficial for private clients. It can be applied for initial design and repair to reduce downstream risk. HPC cloud computing enables the prediction of future performance of buildings by simulation for a broad range of input parameters in a reasonable time period due to the performance benefits with HPC cloud computing. Invasive / destructive building forensics can be reduced. Together with the ability to separate design, material, and workmanship deficiencies, the design process, potential forensic investigations, or litigation can draw huge benefits from utilizing HPC cloud computing with hygrothermal building simulation.

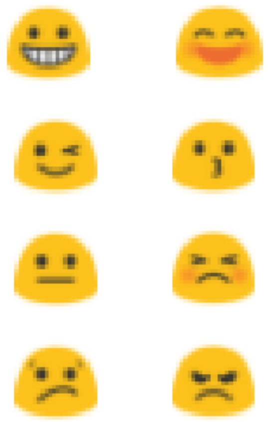
Additional Note

The original case study contains an Appendix with more details about the parameter study. If you are interested please contact wolfgang.gentzsch@theubercloud.com, and we will send the complete case study to you.

Case Study Authors – Matthias Pazold, Dave Pursiano, Florian Antretter, and Wolfgang Gentzsch

Machine Learning Software

Consumer Analytics using Natural Language Processing and Artificial Intelligence in the Cloud



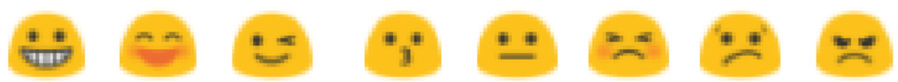
"UberCloud's HPC and Cloud computing capabilities have aided in the processing of large volumes of consumer data in order to build AI models for making better decisions and game-changing strategies."

MEET THE TEAM

End-User/Data Science Expert: Veena Mokal, Data Science Expert, MBA in Business Analytics, Institute of Management Technology, INDIA
Software Provider: Anaconda Python distribution platform
Resource Provider: On-premises systems. And next step: UberCloud Engineering Simulation Platform
HPC Expert: Praveen Bhat, HPC/Python Technology Consultant, India, Wolfgang Gentzsch, UberCloud

USE CASE

In recent years, advancements in internet connectivity have brought significant opportunities to customers and shoppers. Because of these advancements in internet connectivity, rapidly rising e-commerce enterprises have yielded true big data. The enormous popularity of big data on social media allows purchasers to voice their opinions and views on a wide range of topics such as the status of the economy, or to express their **displeasure** with certain items or services, or to express their **satisfaction** with their purchases.



Such numerous consumer opinions and product reviews contain rich and valuable information and recently became important sources for both consumers and business firms. Consumers commonly seek quality information from online reviews before purchasing a product, while many firms use online reviews as important feedbacks of their products, marketing and consumer relationship management. Therefore, understanding the psychology behind online consumer behaviour became the key to compete in today's markets which are characterized by ever-increasing competition and globalization.

Sentiment analysis & text analysis are the applications of big data analysis, which aim to aggregate and extract emotions and feelings from different kinds of reviews. These big data which is growing exponentially are mainly available in an unstructured format, and they are not machine-processable and interpretable. Therefore, the use of the Natural Language Processing (NLP) machine learning technique is essential which focuses on extracting this data and opinions from the huge amount of information present on the web.

Consumer sentiment analysis is one of the popular techniques of discovering the emotion to understand your customer related to your product or service. The scope of this project is to develop a consumer analytics framework considering e-commerce review data using AI machine learning techniques with a Cloud solution capability. This study enables the objective to understand the consumer in real-time and to identify a critical issue affecting the business. As this process is resource-driven, and the data extracted from social media is huge, it requires a higher number of CPU cores and RAM for faster data processing, model building & data visualization. This study explores this relationship through a cloud-based solution infrastructure to speed up computation time and achieve the faster turn-around time required in the model building activity.

PROCESS OVERVIEW

The following defines the step-by-step approach in setting up the model environment for consumer analytics using NLP and Python.

a. Data Pre-Processing & Feature Extraction

The pre-processing of data involves Text cleaning & Feature Extraction:

Text Cleaning: Text cleaning activity is one of the major steps in data pre-processing. In this step punctuations, stop words, URL Links, HTML tags, and special characters like emoticons and emoji are removed and the text converts into lower case. In the final step, text cleaning performs spell checks and corrects it grammatically.

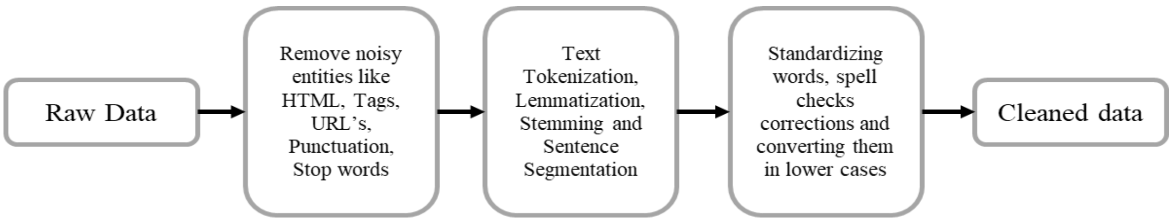


Figure 12: Text cleaning process

Feature Extraction: The cleaned data is then processed to extract features (variables) that help to understand the distribution of the review text. This includes the distribution of characters, numerical values, lowercase words, and average word length. This completes the data pre-processing step.

b. Performing Sentiment Analysis

This section mainly focuses on generating sentiment scores for the review text data. Sentiment score is a function of polarity & subjectivity. Both parameters are extracted from the review text using NLP algorithms to understand the overall sentiment. Typically, the overall sentiment is often inferred as positive, neutral or negative from the sign of the polarity score. Polarity [11] is a floating-point number that lies in the range of [-1,1] where 1 means a positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float that lies in the range of [0,1].

c. Topic Modeling

Topic modelling is the process of identifying topics in a set of documents. It enables search engines on the topics of documents that are important. There are multiple methods of doing this, however, this project includes Latent Dirichlet Allocation (LDA). LDA is a form of unsupervised learning that views documents as bags of words. It works by first making a key assumption: the way a document was generated was by picking a set of topics and then for each topic picking a set of words. To do this it does the following for each document m:

The following algorithm steps are implemented using Python and LDA for topic modelling:

- Assume there are k topics across the whole document
- Distribute these k topics across document m (this distribution is known as α and can be symmetric or asymmetric, more on this later) by assigning each word a topic
- For each word w in document m, assume its topic is wrong, but every other word is assigned the correct topic
- Probabilistically, assign word w a topic based on two things:
  - o what topics are in document m
  - o how many times word w has been assigned a particular topic across all the documents?
- Repeat this process several times for each document to get the list of topics

	Word 1	Word 2	Word 3	Word 4	--	--	--	--	Word n
Topic -1	--	--	--	--	--	--	--	--	--
Topic -2	--	--	--	--	--	--	--	--	--
Topic -3	--	--	--	--	--	--	--	--	--
Topic -4	--	--	--	--	--	--	--	--	--
. . .	--	--	--	--	--	--	--	--	--
Topic -k	--	--	--	--	--	--	--	--	--

Figure 13: Details on topic modelling structure

The above figure shows the representation of topic modelling structure and how the individual topics are related to each of the words in the documents.

d. Predictive modeling

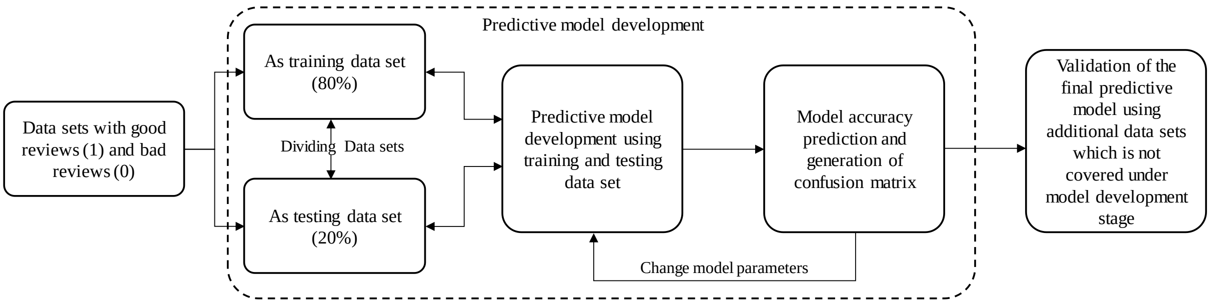


Figure 14: Predictive model development framework

The objective of this phase is to develop a modelling methodology that classifies new input review text into good or bad reviews. The classification accuracy and validation of the model become key criteria for the selection. The predictive model can be developed using both supervised and unsupervised learning methods. This study covers the following predictive modelling techniques to predict the type of reviews (good or bad reviews):

- Naïve Bayes with
    - Gaussian method
    - Multinomial method
    - Bernoulli method
- Logistic Regression model fitting

RESULTS & DISCUSSION

This section details out the results extracted by running the Python scripts developed for the methodology explained. The following provides details on the results derived and insights gained from the analysis.

Make all text lower case

The first pre-processing step which we will do is transform our reviews into lower case. This avoids having multiple copies of the same words. For example, while calculating the word count, 'Analytics' and 'analytics' will be taken as different words.

```
In [17]: df['reviewText'] = df['reviewText'].apply(lambda x: " ".join(x.lower() for x in x.split()))
         df['reviewText'].head()

Out[17]: 0    we got this gps for my husband who is an (otr)...
         1    i'm a professional otr truck driver, and i bou...
         2    well, what can i say. i've had this unit in my...
         3    not going to write a long review, even thought...
         4    i've had mine for a year and here's what we go...
         Name: reviewText, dtype: object
```

Figure 15: Example showing texts converted to lower case

The first step shows all the review texts converted into lower case followed by removing the punctuation marks and stop word removal, unwanted texts like HTML tags, emoticons, white spaces etc. The data set is further subjected to multiple pre-processing steps involved which will further help to cleanse and structure the data sets for further analysis. This includes - extracting the number of words, characters and average word length.

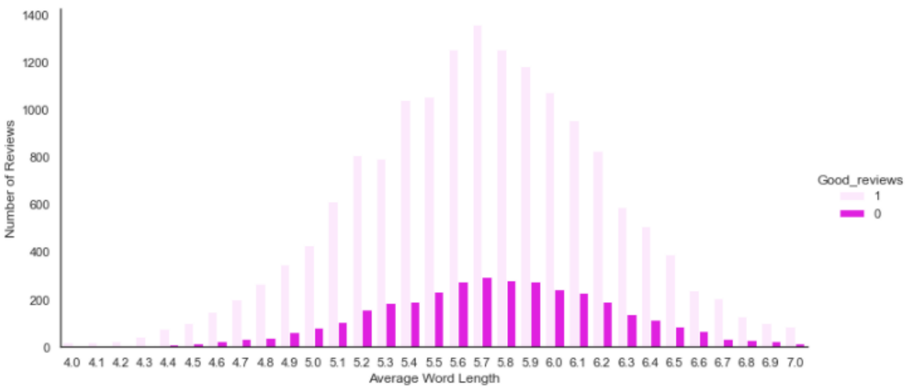


Figure 16: Distribution showing the average word length for good and bad reviews

The above figure shows the average word length for good and bad reviews. From the distribution, the word length for a good review is longer than the bad review and hence the processing time for the good review data is higher when compared to the bad review data.

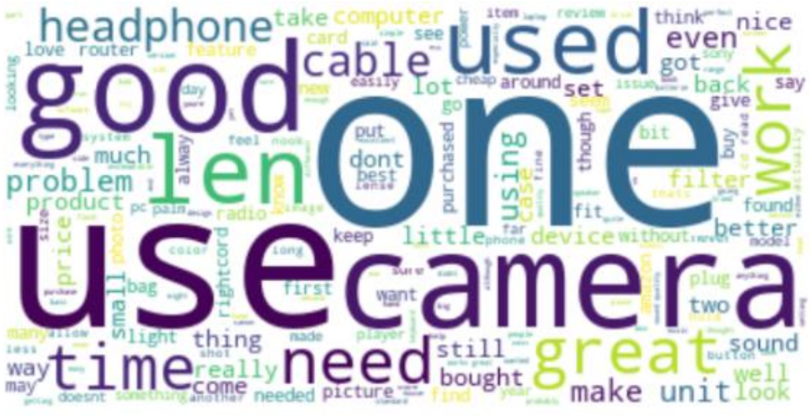


Figure 17: Word Cloud extracted from the processed review data

Figure 17 shows the word cloud which is a visual representation of the word frequency. The more commonly the term appears within the text being analysed, the larger the word appears in the image generated. Word clouds are increasingly being employed as a simple tool to identify the focus of written material.

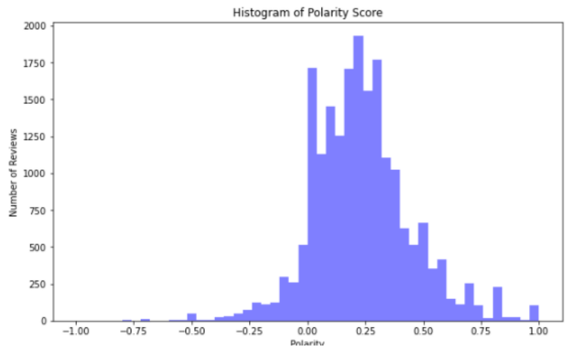


Figure 18: Sentiment scores for the first 2000 reviews

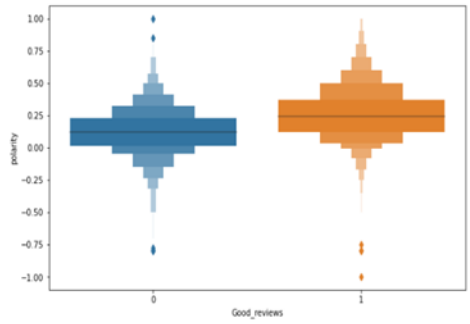


Figure 19: Polarity values for bad (0) & good reviews (1)

Figure 18 shows the sentiment score distribution for the first 2000 reviews where the major portion of the review sentiments shows positive polarity for good reviews. *Figure 19* shows a good review distribution on polarity. Good reviews that have low polarity are categorized in negative sentiments. Bad reviews which have high polarity are categorized in positive sentiments.

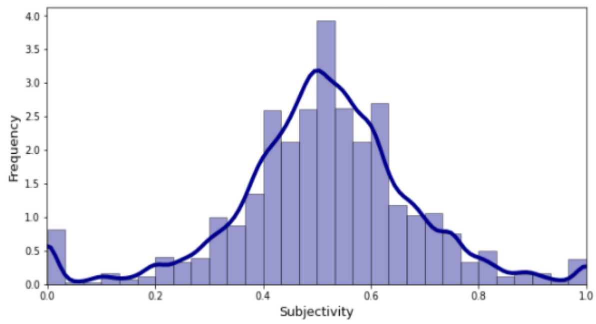


Figure 20: Distribution of subjectivity

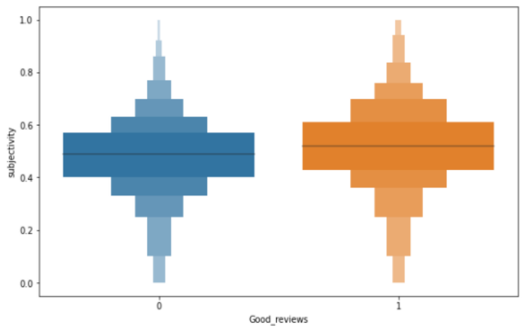


Figure 21: Subjectivity values for good (1) and bad (0) reviews

Figure 20 shows the subjectivity score distribution. *Figure 21* shows a good review distribution on subjectivity. *Figure 22* shows the distribution of subjectivity and polarity for good and bad reviews.

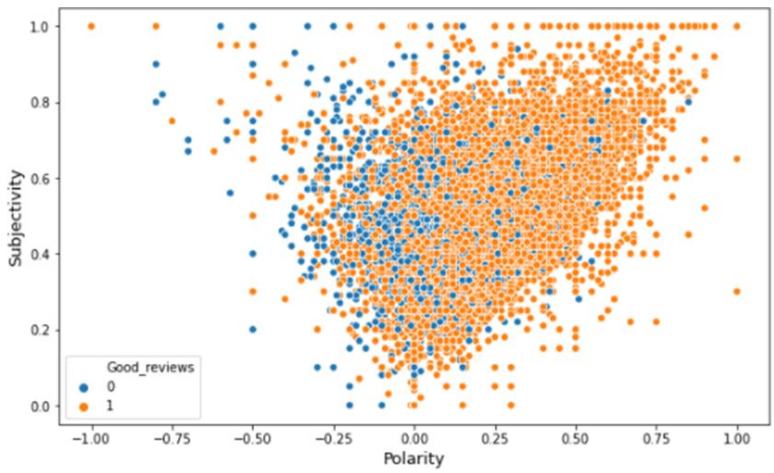


Figure 22: Distribution of subjectivity and polarity for bad and good reviews

The topic modelling activities focus on defining the list of topics from the review data and creating a matrix of topics. The LDA algorithm is used to analyse the topics and generate the probability of occur-ence of topics in a document based on the words. The LDA works with an assumption that each docu-ment is a collection of words, “bag of words”, thus the order of the words and their grammatical role are not considered in the model. **Error! Reference source not found.** provides the LDA equation considering 4 topics.

```
# Let's try 4 topics
ldan = models.LdaModel(corpus=corpusn, num_topics=4, id2word=id2wordn, passes=10)
ldan.print_topics()
```

```
[(0,
  '0.044*"headphones" + 0.026*"sound" + 0.018*"price" + 0.017*"quality" + 0.014*"radio" + 0.013*"bass" + 0.012*"pair" + 0.012
*"music" + 0.011*"use" + 0.010*"volume"'),
 (1,
  '0.030*"cable" + 0.013*"product" + 0.011*"mouse" + 0.011*"use" + 0.010*"power" + 0.010*"tv" + 0.010*"router" + 0.010*"cables"
+ 0.010*"computer" + 0.010*"price"'),
 (2,
  '0.015*"palm" + 0.013*"use" + 0.009*"player" + 0.009*"unit" + 0.007*"device" + 0.007*"case" + 0.007*"software" + 0.007*"tape"
+ 0.006*"cd" + 0.006*"battery"'),
 (3,
  '0.051*"camera" + 0.018*"bag" + 0.017*"lens" + 0.016*"use" + 0.014*"canon" + 0.012*"pictures" + 0.009*"quality" + 0.008*"batt
eries" + 0.008*"flash" + 0.008*"card"')]
```

Figure 12: LDA models with 4 topics considered

A similar approach was followed to extract the topics using both nouns and adjectives and then use the LDA algorithm to analyse the topics. The above exercise is performed for both good and bad reviews separately to generate a list of topics and the associated bag of words for further analysis.

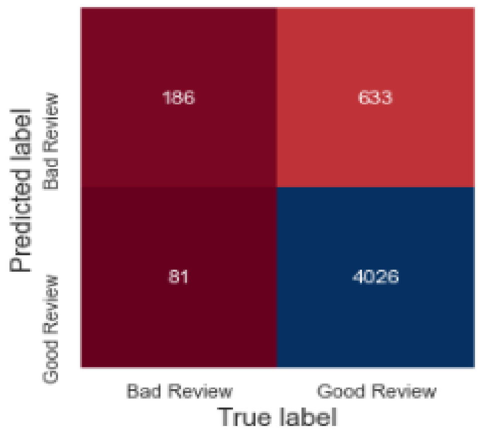


Figure 23: Confusion matrix for logistic regression

The processed data which contains good and bad reviews groups are used for training the predictive models. The review dataset is divided into an 80% training and a 20% testing data group which is used for building the model. The following modelling output is evaluated for model accuracy through a confusion matrix and an accuracy percentage. The Gaussian predictive model shows 71% of prediction accuracy, where around 271 data points are misclassified for the good review and 1142 data points are misclassified for the bad review.

The Multinomial predictive model shows 84% of prediction accuracy where around 749 data points are misclassified as the good review and 21 data points are misclassified as the bad review. The Bernoulli predictive model shows 80% of prediction accuracy where 450 data points are misclassified as the good review and 559 data points are misclassified as the bad review. The Logistic regression predictive model shows 86% of prediction accuracy where 633 data points are misclassified as the good review and 81 data points are misclassified as bad review.
Out of all predictive models, the logistic regression model has a better prediction in terms of accuracy with a lesser misclassification percentage. *Figure 24* shows the comparison done on different predictive modelling methodology.

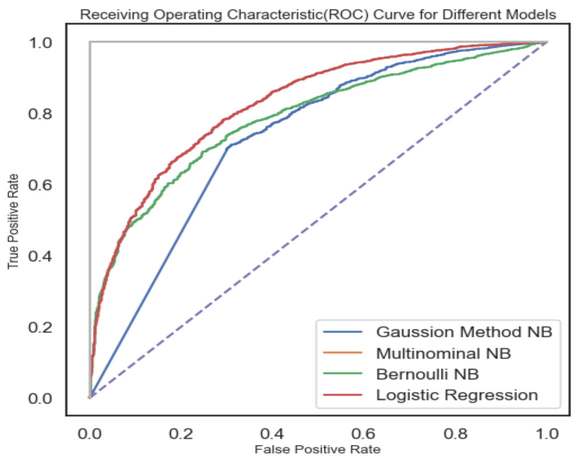


Figure 24: Comparison of different modelling techniques

MODEL VALIDATION
The logistic regression model is validated using the latest review data from the e-com website. The data is then pre-processed using the Python script and analysed using the logistic regression model.

ReviewText	Actual Label	Predicted Label
very secure, we did have to use concrete bolts thought since we were putting it on an outsid	1	1
We bought this for a 55" Vizio and it worked perfectly. I would buy it again and recomm	1	1
Mixed up the review to show how the Nook changed with Updates within 12 hours of openi	0	1
It worked perfectly. Nothing more that I could ask from a home charger. If you need one it w	1	1
I uploaded All of United States and Canada via Garmin and have all the space I need for my G	1	1
Hey, I am a music professional. I make many recordings and service equipment so I know wh	0	0
I have used Maxwell for some time. I liked the price and having used these in the past, you c	1	1
It is a good cord. Simply packaged. Nicely priced.There is not much more to say about a sim	1	1
I'm so sick of most local stores overcharging for cables. I always try to buy my cables online.	1	1
The clear tape is great as a way to replace engraving. It does have a gloss like scotch tape an	0	1
I now own three pair of these, They are a bargain, as I have dozens of earbuds (mostly in ea	1	1
Sure these headphones look a bit wierd. A matter of taste at best.However the sound is gre	1	1
I gave it 5 stars because for the price, these are about the best headphones you can get sans	1	1
Gave these as a stocking stuffer and well received. They were worn during workout, and we	1	1
I have the PortaPro, which I bought when it was selling for $50. The sound is incredible for s	1	1
A good one of these can last 20 years. A bad one doesn't last 20 weeks. Either way, getting th	0	1
I have the DYMO Letra Tag Label maker and I love it. I have one at work and one at home. Ex	1	1
Your standard power strip, with surge protection. Does the job, outlets aren't inaccessible a	1	1
I bought this to replace an orange cord I had in my front yard for Christmas lights. The green	1	1
This cord seems as if it will suffice. It is long enough to reach from the outlet on the outside	1	1

Figure 25: Latest data extracted (Jan-Feb 2021) from e-com portal where actual and predicted labels are compared

Prediction accuracy for good review data has 79 misclassified data out of 4000 good review data points.

HPC PERFORMANCE BENCHMARKING

The NLP – Machine Learning algorithm for e-com reviews is a very compute intensive technique, therefore, to complete the study, we have run a performance analysis using a high-performance desktop machine that has 16 CPU Cores and 32 GB RAM. The performance analysis was conducted to study the computing system requirement to run millions of review data.

CHALLENGES

The challenges faced in the project were related to processing massive volumes of data from different social media and e-commerce websites. The project requirement was to build an AI-driven model to process the reviews from different web sources and analyse the sentiments behind these reviews. Processing review data with local computers was a challenge to handle the data size. This resulted in the need to use a high-performance compute node, where we could load the big volume of data to perform data analysis and develop required steps to pre-process the data.

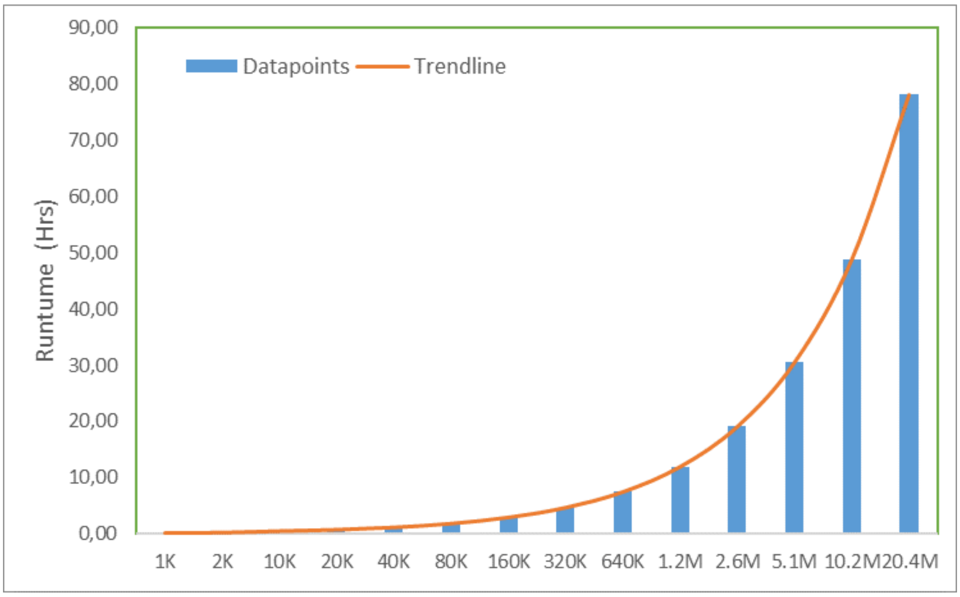


Figure 26: Compute runtime captured for different sizes of review data sets

BENEFITS

1. The HPC cloud computing environment features the Python-based Anaconda platform that aided in data analysis and the construction of predictive models. Dealing with a large volume of data and pre-processing formatting activities was difficult in this project. These tasks demanded a significant amount of computing power. The handling and processing of such massive amounts of data was made possible by cloud HPC.
2. Experiments conducted in the HPC Cloud environment demonstrated the ability to remotely set up and run Big Data analysis as well as build AI models in the cloud. The AI - Machine learning model setup requirements were pre-installed in the HPC container, allowing the user to access the tools without installing any kind of prior set up.

CONCLUSION & RECOMMENDATIONS

- Advanced machine learning technology, such as NLP, is a field of study that examines people's sentiments, attitudes, or emotions toward specific entities. This study addresses the fundamental problem of consumer behavior by using sentiment analysis, sentiment polarity categorization, and high-performance computing containers to speed up the process.
- All of these applications show how sentiment analysis can be a useful resource for analyzing affective information in social platforms, relying not only on domain-specific keywords but also on commonsense knowledge bases that allow for the extrapolation of cognitive and affective information associated with natural language text.
- UberCloud's HPC resource was a good fit for performing NLP analytics and building AL-ML models involving social media and e-commerce big data that could not be executed on a standard workstation. UberCloud's environment aided in speeding up model development activities within a set timeline and completing the project successfully.

REFERENCES

1. Collobert, R. Weston, J. Bottou, L. Karlen, M. Kavukcuoglu, K. Kuksa, (2011). Natural Language Processing (almost) from scratch. Journal of Machine Learning Research 12, pp.2493-2537.

2. Hai, Z. Chang, K. Kim, J. and Yang, C (2014). Identifying Features in Opinion Mining via Intrinsic and Extrinsic Domain Relevance. IEEE transactions on knowledge and data engineering, Vol. 26, No. 3.
3. Hatzivassiloglou, V. & Wiebe, J.M. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. Proc. 18th Conf. Computational Linguistics, pp. 299-305.
4. Pang, B. & Lee, L (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summari-zation Based on Minimum Cuts. Proc. 42nd Ann. Meeting on Assoc. for Computational Linguistics.
5. Mcdonald, R. Hannan, K. Neylon, T. Wells, M.& Reynar, J (2007). Structured Models for Fine-to-Coarse Sentiment Analysis. Proc. 45th Ann. Meeting of the Assoc. of Computational Linguistics, pp. 432- 439.
6. Qu, L. Ifrim, G. & Weikum, G (2010). The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. Proc. 23rd Int'l Conf. Computational Linguistics, pp. 913-921.
7. Yessenalina, A. & Cardie, C (2011). Compositional Matrix-Space Models for Sentiment Analysis. Proc. Conf. Empirical Methods in Natural Language Processing, pp. 172-182.
8. Jin, W. & Ho, H.H (2009). A Novel Lexicalized HMM-Based Learning Framework for Web Opinion Mining. Proc. 26th Ann. Int'l Conf. Machine Learning, pp. 465-472.
9. Qiu, G. Liu, B. Bu, J. & Chen, C (2011). Opinion Word Expansion and Target Extraction through Double Propagation. Computational Linguistics, vol. 37, pp. 9-27.
10. Pang, F. B. & Lee, L. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proc. 42nd Ann. Meeting on Assoc. for Computational Linguistics.
11. https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/

APPENDIX: Summary of the UberCloud Software Containers Suitable for NLP

UberCloud High-Performance Computing Containers are ready-to-execute packages of software. These packages are designed to deliver the tools that an engineer needs to complete his task. The ISV or Open-Source tools are pre-installed, configured, and tested, and are running on bare metal, without loss of performance. They are ready to execute, literally in an instant with no need to install software, deal with complex OS commands, or configure. The UberCloud Container technology allows wide variety and selection for the engineers because they are portable from server to server, Cloud to Cloud. The Cloud operators or IT departments no longer need to limit the variety, since they no longer have to install, tune and maintain the underlying software. They can rely on the UberCloud Containers to cut through this complexity. This technology also provides hardware abstraction, where the container is not tightly coupled with the server (the container and the software inside isn't installed on the server in the traditional sense). Abstraction between the hardware and software stacks provides the ease of use and agility that bare-metal environments lack. Finally, these containers run on top of the UberCloud Engineering Simulation Platform, which is fully automated, self-service, and are easily migrated from any cloud to any other cloud.