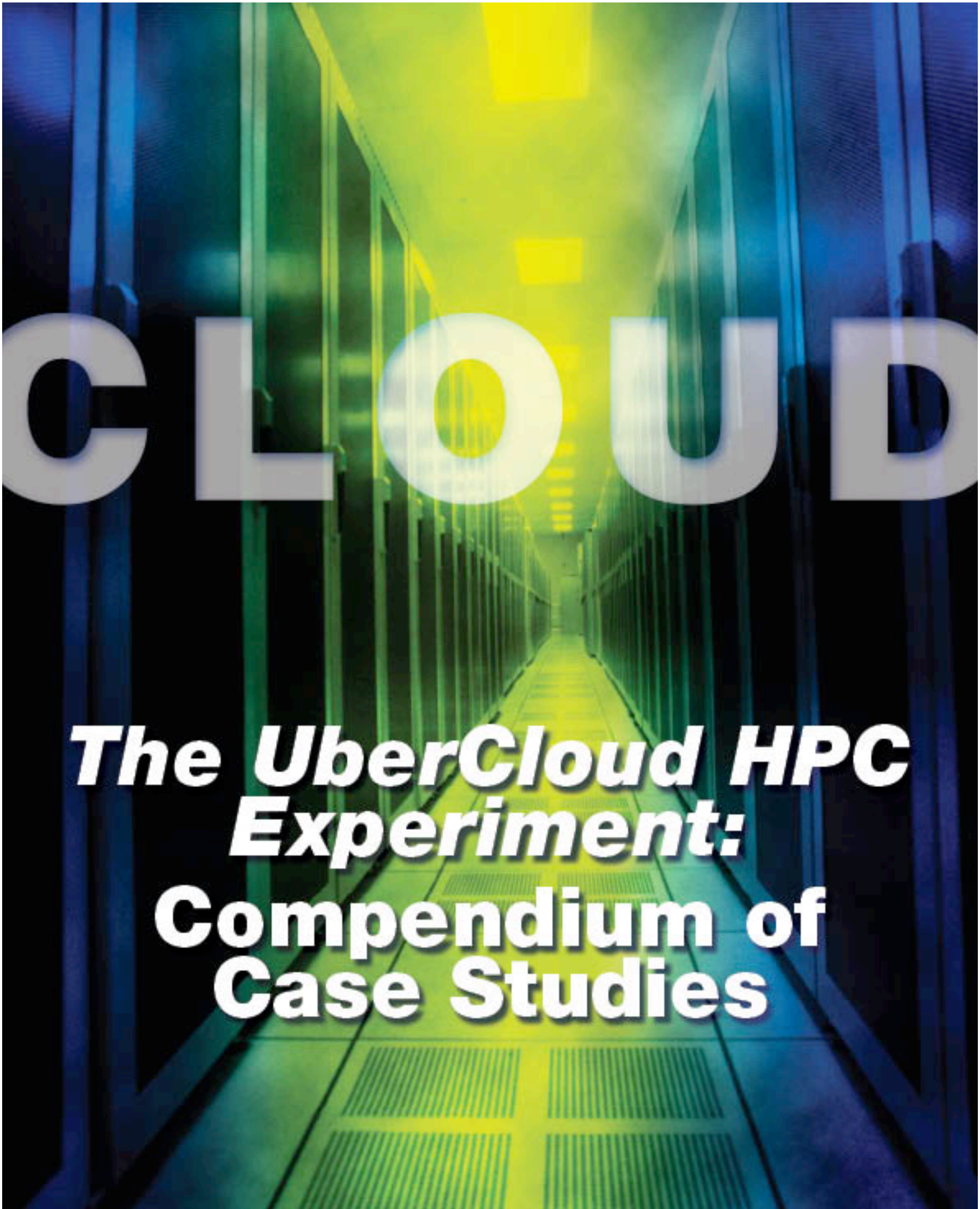


The *UberCloud* HPC Experiment: Compendium of Case Studies

Digital manufacturing technology and convenient access to High Performance Computing (HPC) in industry R&D are essential to increase the quality of our products and the competitiveness of our companies. Progress can only be achieved by educating our engineers, especially those in the "missing middle," and making HPC easier to access and use for everyone who can benefit from this advanced technology.

The UberCloud HPC Experiment actively promotes the wider adoption of digital manufacturing technology. It is an example of a grass roots effort to foster collaboration among engineers, HPC experts, and service providers to address challenges at scale. The UberCloud HPC Experiment started in mid-2012 with the aim of exploring the end-to-end process employed by digital manufacturing engineers to access and use remote computing resources in HPC centers and in the cloud.

In the meantime, the UberCloud HPC Experiment has achieved the participation of 500 organizations and individuals from 48 countries. Over 80 teams have been involved so far. Each team consists of an industry end-user and a software provider; the organizers match them with a well-suited resource provider and an HPC expert. Together, the team members work on the end-user's application – defining the requirements, implementing the application on the remote HPC system, running and monitoring the job, getting the results back to the end-user, and writing a case study.

Intel decided to sponsor a Compendium of 25 case studies, including the one you are reading, to raise awareness in the digital manufacturing community about the benefits and best practices of using remote HPC capabilities. This document is an invaluable resource for engineers, managers and executives who believe in the strategic importance of this technology for their organizations. You can download it at:
http://tci.taborcommunications.com/UberCloud_HPC_Experiment

Very special thanks to Wolfgang Gentzsch and Burak Yenier for making the UberCloud HPC Experiment possible.

This HPC UberCloud Compendium of Case Studies has been sponsored by Intel and produced in conjunction with Tabor Communications Custom Publishing, which includes HPCwire, HPC in the Cloud, and Digital Manufacturing Report.

If you are interested in participating in this experiment, either actively as a team member or passively as an observer, please register at
http://www.hpcexperiment.com

Compendium Sponsor

intel

Media Partners

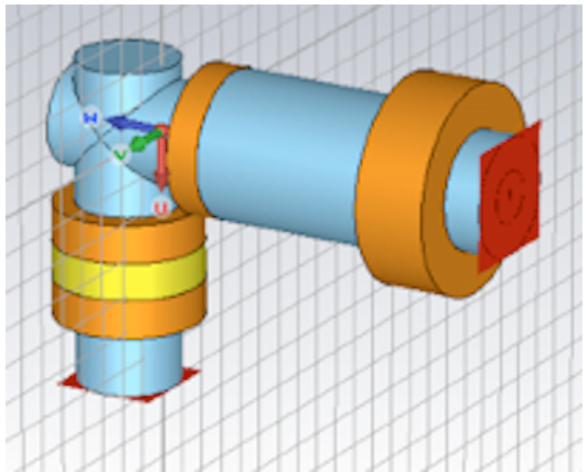
TABOR COMMUNICATIONS

HPC wire

HPC In the Cloud

Digital Manufacturing report

Simulation of a Multi-resonant Antenna System Using CST MICROWAVE STUDIO



"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."

MEET THE TEAM

End User – Dr. Nicolas Freytag
Freytag is a 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.

Software Provider – Dr. Felix Wolfheimer
Wolfheimer is a Senior Application Engineer with CST AG

Resource Provider – Amazon Web Services

HPC Expert – Chris Dagdigian
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.

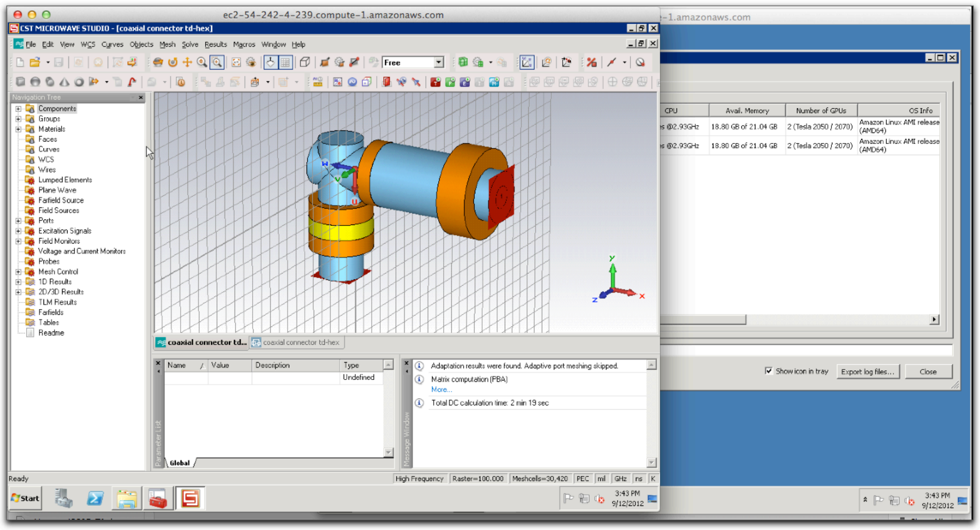
End-user effort: >25h for setup, problems and benchmarking. >100h for software related issues due to large simulation projects, bugs, and post-processing issues that would also have occurred for purely local work.

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.



Front-end and two GPU solvers in action

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.

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 – Felix Wolfheimer and Chris Dagdigian

Thank you for your interest in the free and voluntary UberCloud HPC Experiment.

To download similar case studies go to:
http://tci.taborcommunications.com/UberCloud_HPC_Experiment

If you, or your organization would like to participate in this Experiment to explore hands-on the end-to-end process of HPC as a Service for your business then please register at:
http://www.hpcexperiment.com/why-participate

If you are interested in promoting your service/product at the UberCloud Exhibit then please register at
http://www.exhibit.hpcexperiment.com/