



This case study was created as part of the free and voluntary UberCloud Experiment

Aerodynamics and Fluttering Study of an Aircraft Wing with Fluid Structure Interaction using ANSYS

Summary

- Aero elastic behavior and flutter instability of an aircraft wing in the subsonic incompressible flight speed regime are investigated.
- The Cloud HPC setup helped smoother completion of the simulation runs without re-trails or resubmission.
- The computation requirement for a fine mesh (12 million cells) is too much for a normal workstation. The Cloud HPC setup completed the simulation within acceptable run time (2.25 hrs).
- ANSYS Workbench increased productivity by enabling different iterations in the experiments by varying the simulation models within the workbench environment.



“The whole user experience in the cloud was similar to accessing a website through the browser.”

THE TEAM

- **End-User/CFD Expert:** Praveen Bhat, Technology Consultant, India.
- **Software Provider:** ANSYS, Inc.
- **Resource Provider:** ProfitBricks
- **HPC Expert:** Burak Yenier, Co-Founder, CEO, UberCloud.

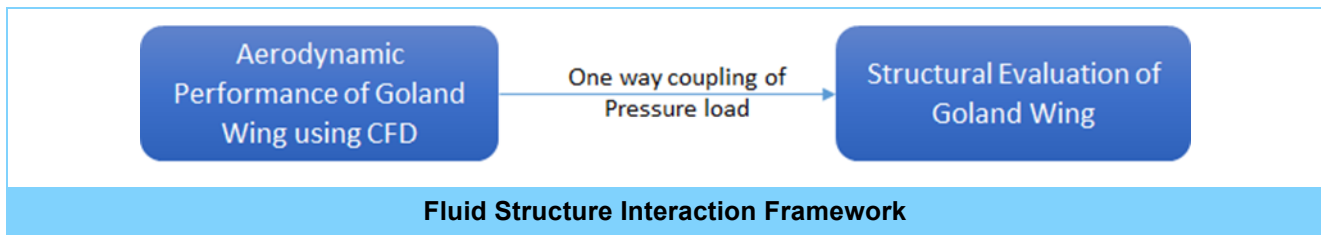
USE CASE

Fluid structure interaction problems in general are too complex to solve analytically and so they have to be analyzed by means of experiments or numerical simulation. Studying these phenomena requires modeling of both fluid and structure. In this case study, aero elastic behavior and flutter instability of aircraft wing in the subsonic incompressible flight speed regime are investigated.

The project involved evaluating the wing aerodynamic performance using the computational fluid dynamics approach. Standard Goland wing is considered for this experiment. The Computational Fluid Dynamics (CFD) models were generated in the ANSYS environment. The simulation platform was built in a 62 core HPC cloud with ANSYS 15.0 modeling environment. The Cloud environment was accessed using a VNC viewer through web browser. ProfitBricks provided the 62 core server with 240 GB RAM. CPU and RAM were dedicated to the single user

and this was the largest instance that was built in ProfitBricks. The ANSYS software was running in UberCloud's new application containers see the Appendix for a short description.

The following flow chart defines the fluid structure interaction framework for predicting the wing performance under aerodynamic loads:



The following defines the step-by-step approach in setting up the Finite Element model using ANSYS Workbench 15.0 Environment.

1. Generate the Golang wing geometry using ANSYS Design Modeler, where the input for the dimension of the wing is the co-ordinate system, which is imported, in the modeling environment as co-ordinate files (*.csv).
2. Develop the CFD model with atmospheric air volume surrounding the Golang wing in ANSYS Design Modeler.
3. Import the CFD model in the Fluent Computational Environment.
4. Define the Model parameters, fluid properties, and boundary conditions.
5. Define solver setup & solution algorithm, mainly related to define type of solver, convergence criteria and equations to be considered for solving the aerodynamic simulation.
6. Extract the pressure load on the wing surface, which is then coupled and applied on the structural wing geometry while solving the structural problem.

The Fluent simulation setup is solved in the HPC Cloud environment. The simulation model needs to be precisely defined with adequate fine mesh

elements around the wing geometry. The following snapshot highlights the wing geometry considered and Fluent mesh models.

The pressure load calculated from the CFD simulation is extracted and mapped on the Golang wing while evaluating the structural integrity of the wing. The following steps define the procedure for the structural simulation setup in ANSYS Mechanical.

1. Golang wing is meshed with ANSYS Mesh Modeler. Hexahedral mesh models were created.
2. The generated mesh is imported in the ANSYS Mechanical Environment where the material properties, boundary conditions etc. are defined.
3. The solution methods and solver setups are defined. The analysis setup mainly involves defining the type of simulation (steady state in this case), output result type (stress & displacement plots, strain plots etc.).
4. The pressure load extracted from the CFD simulation is mapped on the wing structure to evaluate the wing behavior under aerodynamic loads.

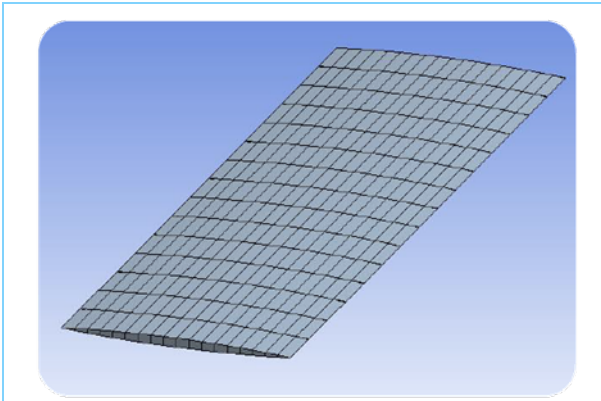


Figure 1: Finite Element Mesh model of Goland Wing

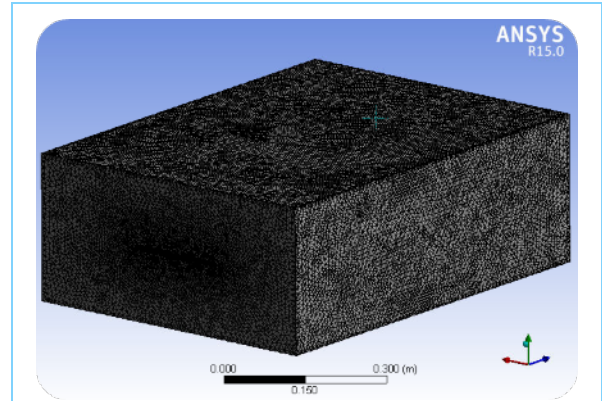


Figure 2: CFD mesh model for the wing geometry with surrounding air volume

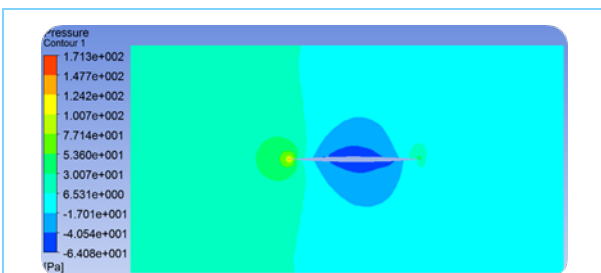


Figure 3: Pressure distribution plot at the mid-section of the wing

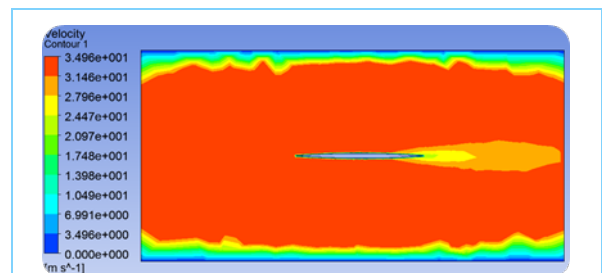


Figure 4: Velocity distribution plot at the mid-section of the wing

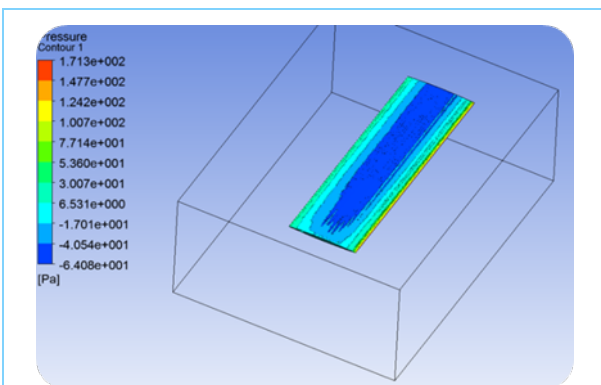


Figure 5: Aerodynamic loads acting on the Wing wall

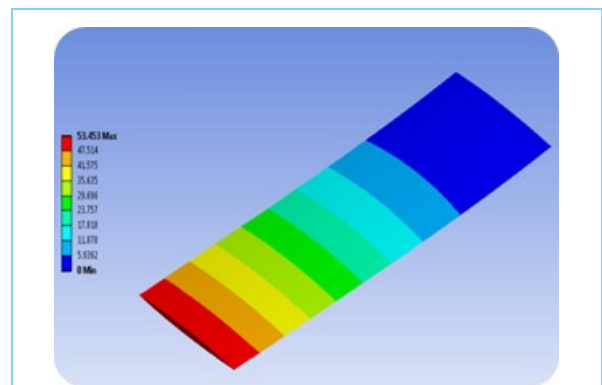


Figure 6: Wing deflection due to aerodynamic load

Figure 3 shows the pressure distribution at the mid-section of the Goland wing. The pressure distribution across the section is uniform. The velocity plot in figure 4 shows that the air velocity varies near the edge of the wing. The air particle velocity is uniform with particles following a streamlined path near the wing wall. Figures 5 & 6 indicate aerodynamic loads on the wing that is calculated based on the pressure distribution on the wing wall. The respective aerodynamic loads are mapped on the wing structure and the deformation of the wing is simulated to evaluate the wing deformation. The wing behavior under

the aerodynamic loads evaluates its flutter stability

HPC Performance Benchmarking

The flutter stability of the aircraft wing study is carried out in the HPC environment that is built on a 62 core server with CentOS Operating System and ANSYS Workbench 15.0 simulation package. The server performance is evaluated by submitting the simulation runs for different

numbers of elements. The higher the element numbers the more is the time required to run the simulation. Using higher core systems can minimize the run time. The following table

highlights the solution time captured for an 8 core system with element numbers ranging between 750K to 12 million.

Table 1: Comparison of solution time (min) for different mesh density

| No. of elements | 8 Core | | 16 Core | | 32 Core | |
|-----------------|----------------------|--------------------|----------------------|--------------------|----------------------|--------------------|
| | Memory utilized (GB) | solving time (min) | Memory utilized (GB) | solving time (min) | Memory utilized (GB) | solving time (min) |
| 750K | 7.92 | 13.00 | 7.92 | 7.00 | 7.92 | 4.00 |
| 2.0M | 9.46 | 66.08 | 9.46 | 35.58 | 9.46 | 20.33 |
| 3.1M | 11.02 | 119.17 | 11.02 | 64.17 | 11.02 | 36.67 |
| 4.3M | 12.55 | 172.25 | 12.55 | 92.75 | 12.55 | 53.00 |
| 5.4M | 14.11 | 225.33 | 14.11 | 121.33 | 14.11 | 69.33 |
| 6.6M | 15.65 | 278.42 | 15.65 | 149.92 | 15.65 | 85.67 |
| 7.7M | 17.21 | 331.50 | 17.21 | 178.50 | 17.21 | 102.00 |
| 9.0M | 18.74 | 384.58 | 18.74 | 207.08 | 18.74 | 118.33 |
| 11M | 20.30 | 437.67 | 20.30 | 235.67 | 20.30 | 134.67 |
| 12M | 21.84 | 490.75 | 21.84 | 264.25 | 21.84 | 151.00 |

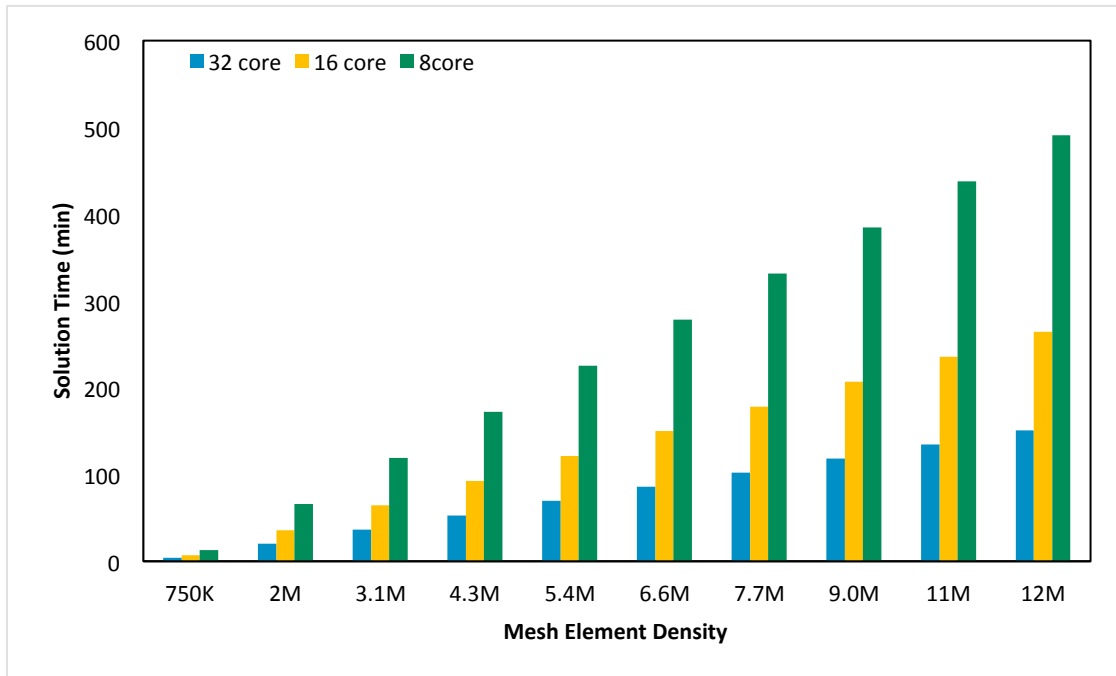


Figure 7: Comparison of Solution time (min) for different element density.

The simulation time reduces considerably with the increase in the number of CPU units. The solution time required for 8 cores with fine mesh model is 3.5 times higher than the time required for a 32 core server with the same mesh model. For a moderate number of elements (~ 750K), the 32

core server performance is 5.2 times better than a normal dual core system with respect to total number of simulation jobs completed in a day.

Person Effort Invested

End user/Team Expert: 100 hours for setup, technical support, reporting & overall management of the project.

UberCloud support: 16 hours for monitoring & administration of host servers and guest containers, managing container images (building & installing container images during any modifications/ bug fixes) and improvements (such as tuning memory parameters, configuring Linux libraries, usability enhancements). Most of the mentioned effort is one time effort and will benefit the future users.

Resources: 1110 core hours for performing various iterations in the simulation experiments.

CHALLENGES

The project started with setting up ANSYS 15.0 workbench environment with Fluent modeling software in 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 from the feedback, the next level of challenge faced was a technical challenge. This involved accurate prediction of flutter behavior of the wing that is achieved through defining appropriate element size to the mesh model. The finer the mesh the higher is the simulation time required and hence the challenge was to perform the simulation within the stipulated timeline.

BENEFITS

1. The HPC cloud computing environment with ANSYS 15.0 Workbench made the process of model generation easier with process time reduced drastically because 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 highly fine mesh (12 million cells) is high which is near to 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 thereby providing an advantage of getting the simulation results within acceptable run time (2.25 hrs).
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 in increasing the productivity in the simulation setup effort and setup thereby providing a single platform to perform end-to-end simulation setup.
5. The experiments performed in the HPC Cloud showed the possibility and gave extra confidence to setup and run 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 that enabled continuous monitoring of the job in progress in the server without any requirement to log-in and check the status.

CONCLUSION & 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, which can be used to perform fluid- structure interaction simulations. 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.

APPENDIX: UberCloud Containers: Brief Introduction

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. 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.

ABOUT UBERCLOUD

UberCloud makes it easy to run your simulations on powerful cloud infrastructure.

No more compromises on mesh quality or model fidelity because of hardware limitations. With UberCloud's flexible software platform and network of cloud partners, you get on-demand access to major providers such as Microsoft Azure, HPE and others. Choose from a variety of secure data centers, and hardware options such as InfiniBand, GPUs etc.

Unleash the full power of your analysis software and boost confidence in your results.

With over 200 technical-computing-as-a-service case studies, UberCloud has the experience, software platform and partnerships required for your success.

Engineers and scientists rely on UberCloud to manage the complexity of cloud and software operations, so they can focus on their analysis.

FOR MORE INFORMATION

2310 Homestead Rd. Suite:C1-301 Los
Altos CA 94024

help@theubercloud.com
www.theubercloud.com/help

