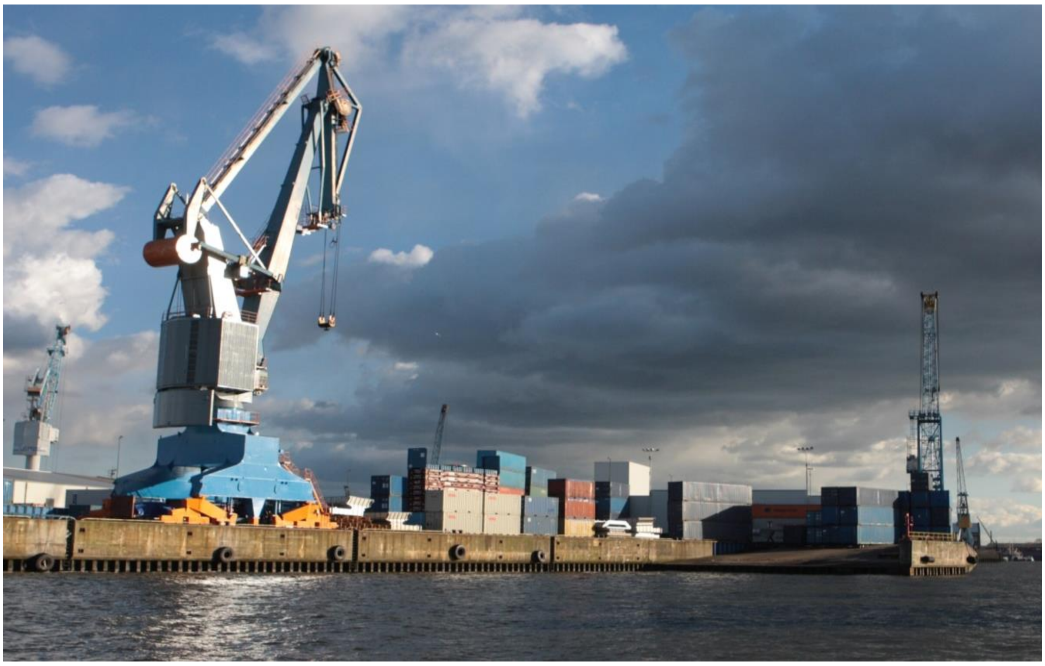
The UberCloud Experiment

Technical Computing in the Cloud



Case Studies 159, 160, 165, and 166:

Computational Fluid Dynamics with ANSYS, CD-adapco, and OpenFOAM Containers in the UberCloud

http://www.TheUberCloud.com

January 21, 2015

Welcome!

The UberCloud* Experiment started in July 2012, with a discussion about cloud adoption in technical computing and a list of cloud computing challenges and potential solutions. We decided to explore these challenges further, hands-on, and the idea of the UberCloud Experiment was born.

We found that especially small and medium enterprises in digital manufacturing would strongly benefit from technical computing in the cloud. By gaining access from their desktop workstations on demand to additional compute resources, their major benefits are: the agility gained by shortening product design cycles through shorter simulation times; the superior quality achieved by simulating more sophisticated geometries and physics and by running many more iterations to look for the best product design. These are benefits that increase a company's innovation and competitiveness.

Tangible benefits like these make technical computing - and more specifically technical computing in the cloud - attractive. But how far away are we from an ideal cloud model for engineers and scientists? In the beginning, we didn't know. We were just facing challenges like security, privacy, and trust; conservative software licensing models; slow data transfer; uncertain cost & ROI; selecting the best suited computing resources; and lack of standardization, transparency, and cloud expertise.

In the course of this experiment, as we followed each of the 170 teams (so far) closely and monitored their challenges and progress, we've got an excellent insight into these roadblocks, how our teams have tackled them. After 60 experiments (and the publication of the first compendium of case studies (sponsored by Intel) we discovered a new open source Linux container technology called Docker. We enhanced Docker with specific features for technical computing, enabling us to reduce or even fully remove many of the cloud computing challenges.

Following the first and second compendium of case studies, we are now proud to present three case studies from Computational Fluid Dynamics based on our new UberCloud Containers documenting the results of the teams, their findings, challenges, lessons learned, and recommendations. **These results present a little revolution in that they prove that this new technology will indeed help to successfully tackle most of the cloud computing roadblocks and thus increase the wider acceptance of cloud computing for engineers and scientists.**

Enjoy reading!

Praveen Bhat, Wolfgang Gentzsch, and Burak Yenier
The UberCloud, January, 2015

**) UberCloud is the online community and marketplace where engineers and scientists discover, try, and buy Computing Power as a Service, on demand. Engineers and scientists can explore and discuss how to use this computing power to solve their demanding problems, and to identify the roadblocks and solutions, with a crowd-sourcing approach, jointly with our engineering and scientific community. Learn more about the UberCloud at: http://www.TheUberCloud.com.*

Foreword

Building a Better HPC Highway with the UberCloud Experiment

Dr. Stephen R. Wheat, General Manager, High Performance Computing Intel Corp.

The UberCloud Experiment owes much of its continuing success to the collaborative model adopted by its originators – Wolfgang Gentzsch and Burak Yenier. Their unique approach is a highly structured form of crowdsourcing that brings together end users, software providers, resource providers, HPC/CAE experts and team mentors to solve a specific problem using HPC in the cloud.

Intel is also involved. Our goal to help the hundreds of thousands of small to medium sized enterprises – especially manufacturers – takes full advantage of the benefits of HPC.

When trying to implement HPC on their own, these companies run into substantial barriers – e.g., the cost of hardware and software, outfitting and maintaining a modern, green data center, and hiring experts in HPC and complex disciplines like CFD and FEA.

Working with cloud service providers introduces yet another level of complexity. Cloud – especially HPC in the cloud – is a maturing space and pioneering users are making good use of the technology. But pioneers inevitably run into major speed bumps and a few arrows along the way. Going it alone can be a problem.

It’s as if a truck driver back in the early days of automotive history decided to build an interstate highway to boost his business. Unless a whole host of other individuals, construction companies and state and federal agencies were involved, the project would most likely fail. Or, if this rugged individualist did manage to build a road, given his limited perspective, he might have wound up with a highway that accommodates 18-wheel tractor trailers, but not automobiles.

Crowdsourcing at Work

The UberCloud approach provides a collaborative environment where a relatively low investment by the participants minimizes the risks, and brings a wealth of group experience to bear on the problem. Everyone involved has a business-driven motive and works together to generate positive results. This is a far cry from a commoditized segment where rival factions are looking to corner the benefits at the expense of the other players.

When you read the case studies in this *Compendium*, you will get a good sense for the symbiotic team environment that generates positive outcomes for everyone – a true win-win situation. This is not to say that crowdsourcing makes finding solutions while working in a cloud environment a piece of cake. The various teams have tackled complex and difficult tasks, ranging from simulating drifting snow to mind-bending genomics research.

In many cases, if the end users were to try and find a solution on their own, they would encounter barrier after barrier until the project failed. By working with team members who offer a broad spectrum of talent and resources, it becomes possible to mix and match cost effective solutions that allow the problem to be resolved. For example, one project might be best addressed with solutions from ANSYS, another with solvers from CD-adapco, and yet another with software from Dassault. In one case, a smaller cloud provider like R Systems might be just the ticket; in another Amazon or Microsoft solutions might be best.

Essentially, The UberCloud brings people out of their silos and into contact with more players who can offer a broader range of solutions.

Lessons Learned

All the participants cited valuable lessons learned. Below are a few comments from Round 3 case studies that will give you a feel for the team members' experience:

- "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… and the accountant's standpoint…will need to be addressed."
- "…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."
- "HPC as a service is most effective in well-established workflows where limited user interaction is required following initial problem definition."
- "Even without MPI parallel communication, the speed advantage using a cluster instead of the in-house PC , is remarkable."
- "…the end user benefited from the HPC provider's knowledge of how to setup a cluster, to run applications in parallel based on MPI, to create a host file, to handle the FlexNet licenses, and to prepare everything needed for turn-key access to the cluster. During the entire process the resource provider stayed competently at the end user's side and provided comprehensive and expedient technical support."

The HPC UberCloud Experiment is well into Round 6 and new teams are constantly signing up to be part of the process in future Rounds. This is an excellent testimonial to the efficacy of the collaborative model the founders have created. They have launched a variation on crowdsourcing that is bringing the benefits HPC as a service to an underserved SME population that, until now, had no way to access this transformative, enabling technology.

Team 159:

Aerodynamic Study of Airfoil

> *"The combination of OpenFOAM & UberCloud Containers enables efficient, effective, and easy performance of complex engineering simulations."*

MEET THE TEAM

End-User/CFD Expert: Praveen Bhat, Technology Consultant, INDIA
Software Provider: ESI Group
Resource Provider: Nephoscale
HPC Expert: Burak Yenier, Co-Founder, CEO, UberCloud.

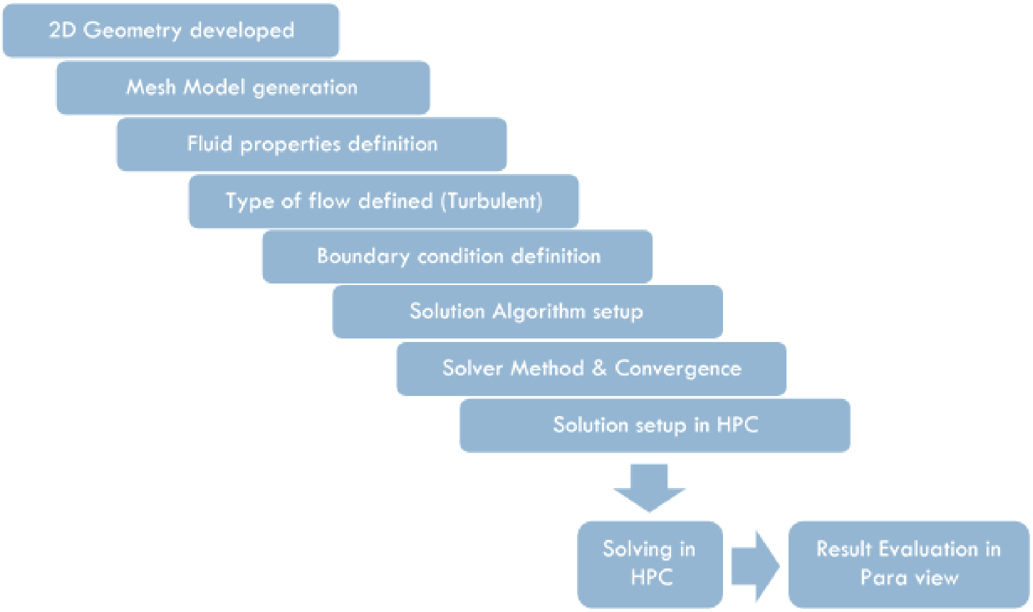


Figure 1: Model Setup flowchart

USE CASE

The aerodynamic study on the 2D airfoil is performed with the incompressible air flow around a 2D airfoil. The model setup includes the geometry preparation where a selected region is model that represents the surrounding air volume with the airfoil profile at the center. The airfoil profile needs to be accurately modeled to capture the variation in the airflow pattern around the airfoil. The model setup is done using the open source software OpenFOAM. The OpenFOAM software is embedded in an UberCloud Container located in the Nephoscale cloud facility. The main objective of this project is to experience the ease-of-use of the UberCloud OpenFOAM container and to evaluate the HPC performance with respect to the accuracy of result prediction and also with respect to the solution time and resource utilization.

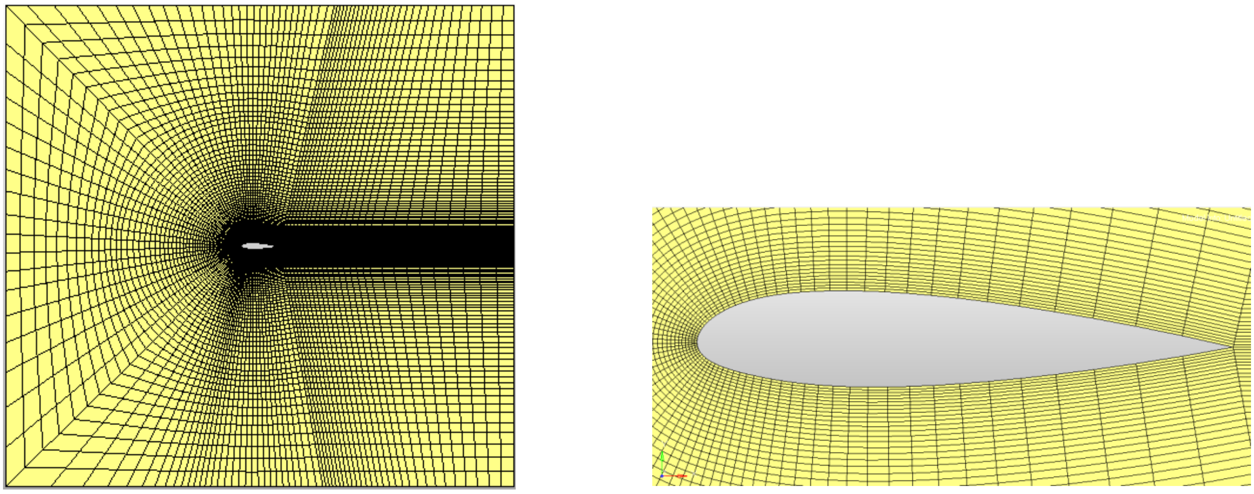


Figure 2: Mesh model for the aerofoil

Process Overview

The meshing density is very fine around the airfoil and also along the path of the trailing edge. The meshes were modeled coarser as it moves away from the airfoil region and the coarseness of the mesh increases near the air volume boundary (air inlet and outlet). The following details describe the steps in the simulation model setup using OpenFoam:

1. The Finite Element mesh model is generated followed by the fluid properties definition. The entire volume surrounding the airfoil is air which is considered as incompressible in nature.
2. The fluid properties are defined as Newtonian fluids with a linear relationship between the shear stress (due to internal friction forces) and the rate of strain of the fluid.
3. Atmospheric air will be turbulent in nature and there is a transition phase from turbulent to laminar in the region near the airfoil. Because of this transition the mesh model needs to be refined accurately near the airfoil region along with defining the turbulence behavior of the air which is captured through a Spalart – Allmaras turbulence model.
4. The next section in the model setup is defining the model boundary conditions and assigning the pressure and velocity initial values. The boundary conditions are assigned where in the airfoil edges are considered as wall. The three sides of the air volume are considered as inlet and the edge following the trailing edge of airfoil is considered as air outlet.
5. The next step in the model development is setting up of the solution algorithm. The problem is solved as steady state and the OpenFOAM solver used for solving this problem is Simple FOAM. The following are the solution parameters for the SimpleFOAM solver: Start time: 0 sec; End time=500 sec; time step= 1sec. The SimpleFOAM solver uses the Gauss-Seidel method for solving. The pressure field is provided with a relaxation factor of 0.3 and the velocity field is assigned a relaxation factor of 0.7, along with the residual parameter which is set at 1.0x10-5. The above parameters define the convergence criteria of the model.
6. The OpenFOAM model developed is then modified for parallel processing where the existing model is divided according to the number of HPC computing nodes.
7. The model is solved in parallel and once the solution is converged, the solved model in the parallel processors is reconstructed to get the final simulation results. The final result is used to view the output of the airfoil simulation and the respective result components are captured using the post-processing software tool Paraview.

The airfoil is surrounded by air volume and the variation in the flow velocity and air pressure near the airfoil section is reviewed. The different plots below show the flow of air and laminar behaviour observed in the airfoil region.

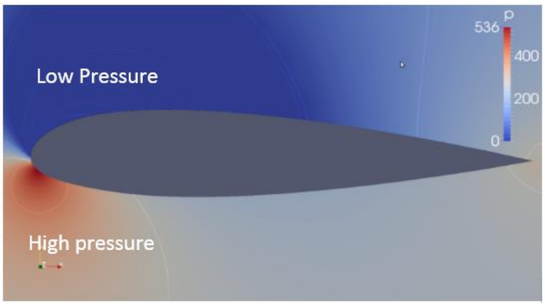


Figure 3:Pressure distribution around airfoil with high & low pressure zone

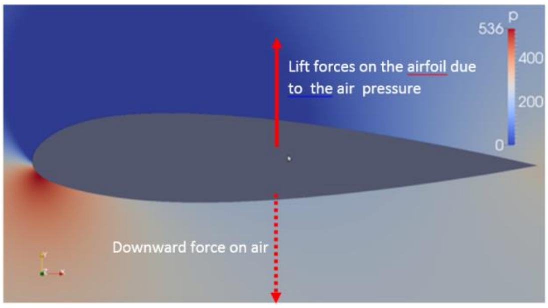


Figure 4: Lift forces represented in the airfoil

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 air plane wing can be considered to be Newton’s third law where in there will be a reaction force in the form of downward force on the air. The life on the airplane wing should be consistent since it is the conservation of the 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 evaluating how the pressure co-efficient value varies with the increase or decrease in the angle of attack.

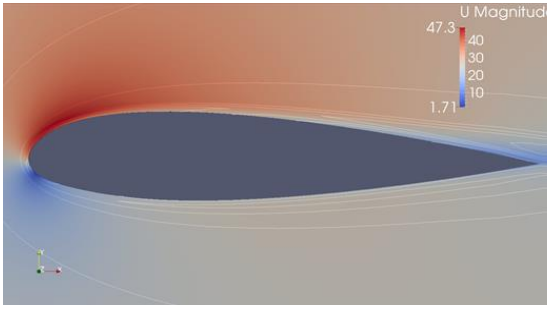


Figure 5: Velocity contour of streamline of air flow around the airfoil

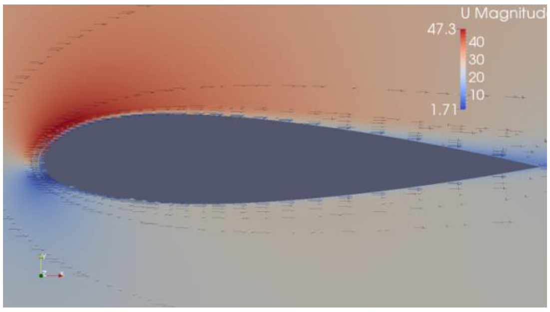


Figure 6: Velocity contour with air flow vectors around the airfoil

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 Figures 5&6. The vector flow path of the air in the airfoil region is also represented where the flow path represents individual air particle flow near the wall boundary of the airfoil.

HPC Performance Benchmarking

The HPC Cloud system is a 32 core system with 32 GB RAM with Ubuntu 12.04. The software installed in the container is OpenFOAM version 2.2 with OpenFoam MPI and Paraview. The model is evaluated for the accuracy of prediction of air flow around the airfoil, with both fine and coarse mesh. The time required for solving the model with different meshes is captured to benchmark the use of HPC performance in solving high density mesh models. The boundary conditions, solution algorithm, solver setup and convergence criteria remain the same for all the models.

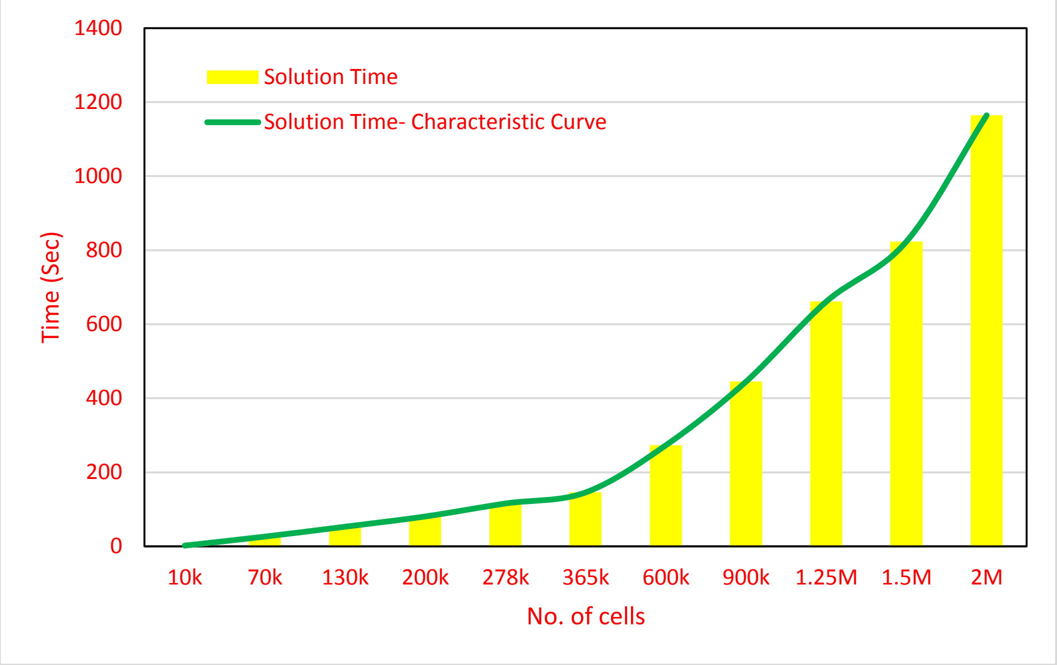


Figure 7: Solution time required in a 4 core configuration

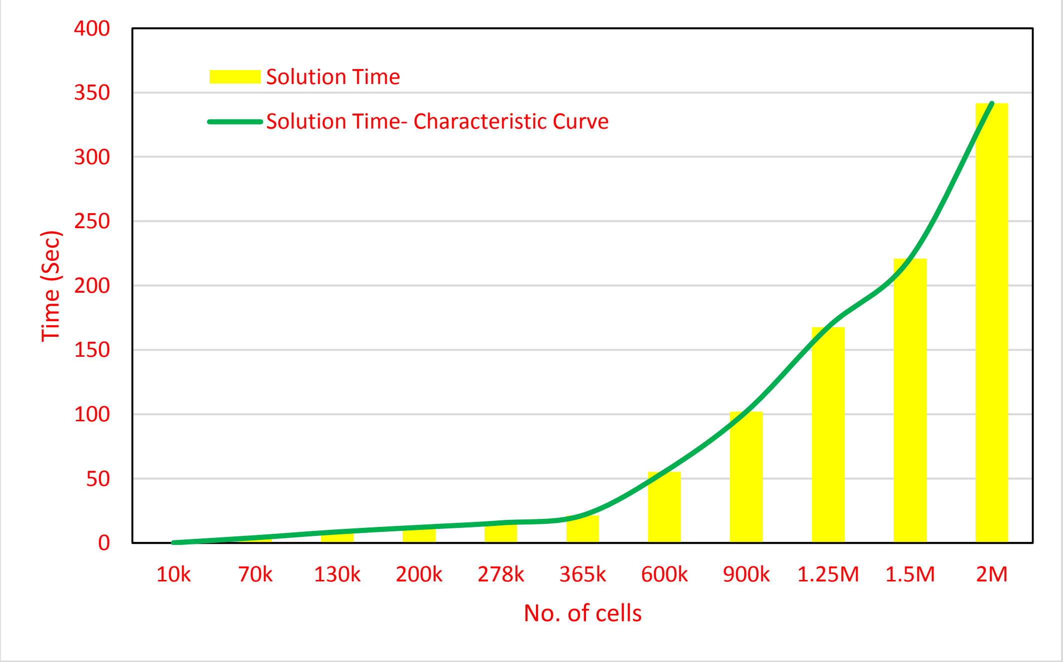


Figure 8: Solution time required in a 32 Core HPC configuration

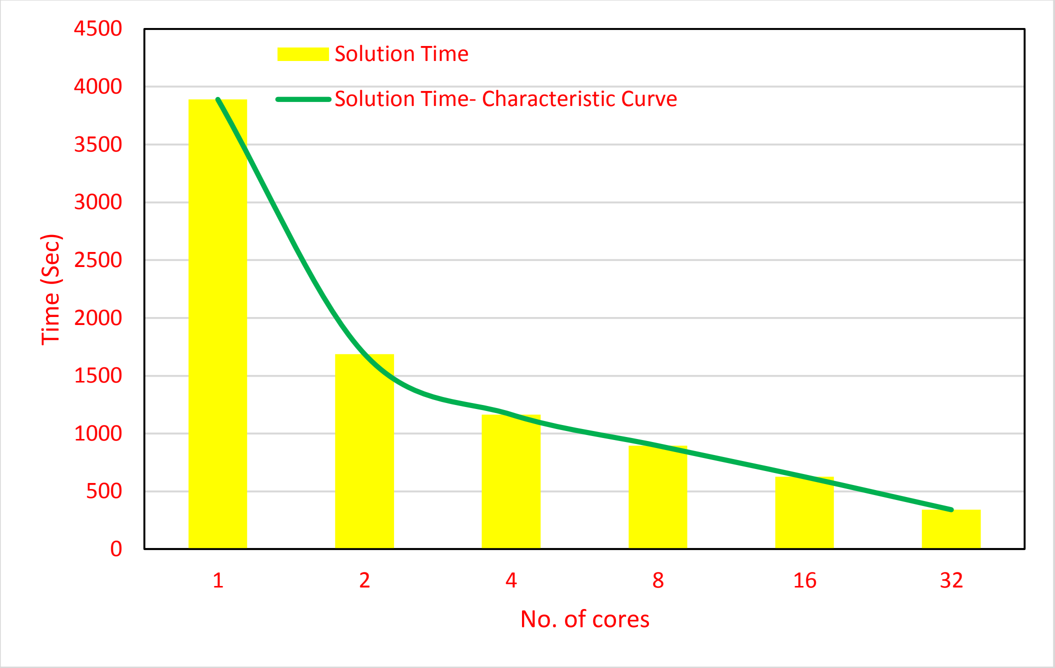


Figure 9: Solution time for a model with 2M elements solved using different HPC core configurations

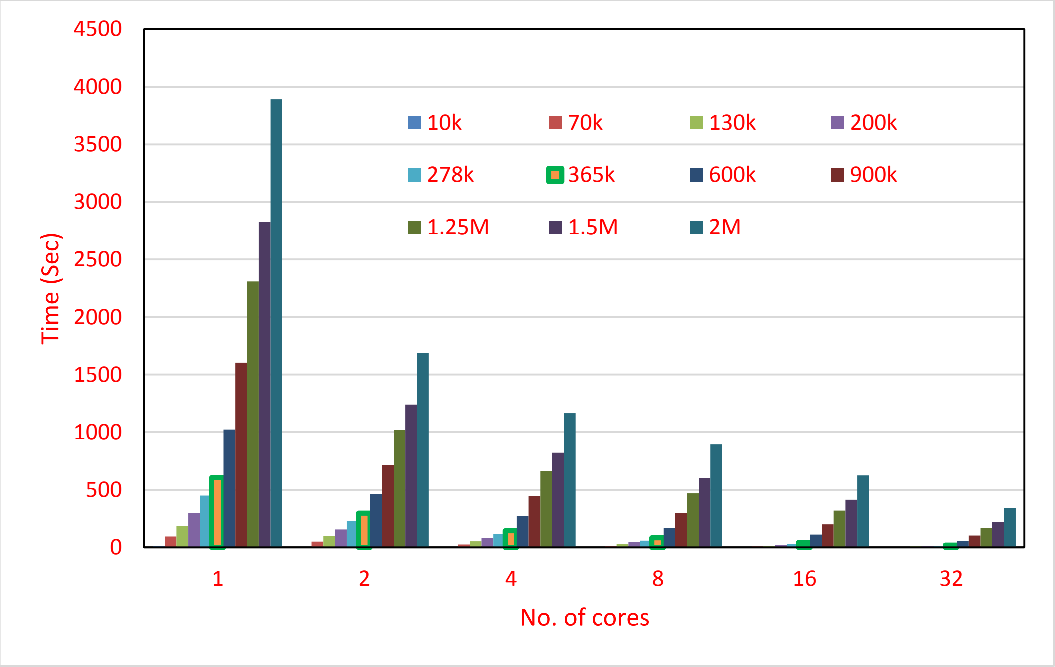


Figure 10: Comparison of solution time for different mesh densities models using different HPC core configurations

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 & 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 the mentioned effort is one time effort and will benefit the future users.

Resources: ~200 core hours for performing various iterations in the simulation experiments**.**

CHALLENGES

The project started testing the installation of OpenFOAM on the HPC 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 level of 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 hence the challenge was to perform the simulation within the stipulated timeline.

BENEFITS

1. The HPC cloud computing environment with OpenFOAM & Paraview made the process of model generation easier with process time reduced drastically along with result viewing & 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 HPC computing resource helped in achieving smoother completion of the simulation runs without re-trails or resubmission of the same simulation runs.
3. The computation requirement for a very fine mesh (2 million cells) is high, which is near to impossible to achieve on a normal workstation. The HPC cloud provided this feasibility to solve very fine mesh models and the simulation time drastically reduced thereby providing an advantage of getting the simulation results within acceptable run time (~30 min).
4. The UberCloud experiments in the HPC Cloud showed the possibility and gave extra confidence in the setup and run of the simulations remotely in the cloud. The different simulation setup tools were installed in the UberCloud Container and this enabled the user to access the tool without any prior installations.
5. With the use of VNC Controls in the Web browser, The UberCloud Container access was very easy with no installation of any pre-requisite software. The whole user experience was similar to accessing a website through the browser.
6. 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 without any requirement to log-in to the server and check the status.

CONCLUSION & RECOMMENDATIONS

1. The selected HPC Cloud environment with UberCloud containerized OpenFOAM on Nephoscale 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. Cloud resources were a very good fit for performing advanced computational experiments that involve high technical challenges and require higher hardware resources to perform the simulation experiments.
3. There are different high-end commercial software applications which can be used to perform virtual simulation. Open source OpenFOAM with HPC UberCloud Containers helped us to solve this problem with minimal effort in setting up the model and performing the simulation trials.
4. The combination of HPC Cloud, UberCloud Containers, and OpenFOAM helped in speeding up the simulation trials and also completed the project within the stipulated time frame.

Team 160:

Aerodynamics & fluttering study on an aircraft wing using fluid structure interaction

> *"The whole user experience in the cloud was similar to accessing a website through the browser."*

MEET THE TEAM

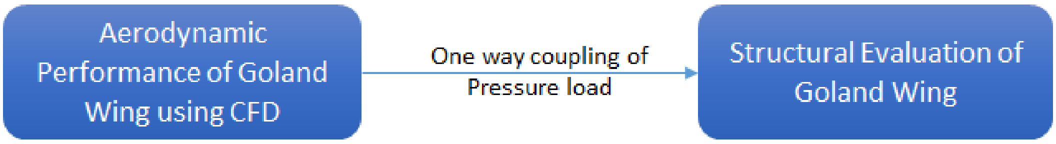
End-User/CFD Expert: Praveen Bhat, Technology Consultant, India.
Software Provider: ANSYS, Inc.
Resource Provider: ProfitBricks
HPC Expert: Burak Yenier, Co-Founder, CEO, UberCloud.

USE CASE

Fluid structure interaction problems in general are too complex to solve analytically and so they have to be analysed by means of experiments or numerical simulation. Studying this phenomena requires modelling of both fluid and structure. In this case study, aero elastic behaviour 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 modelling 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 Goland 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 Goland 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 good amount of fine mesh elements around the wing geometry. The following snapshot highlights the wing geometry considered and Fluent mesh models.

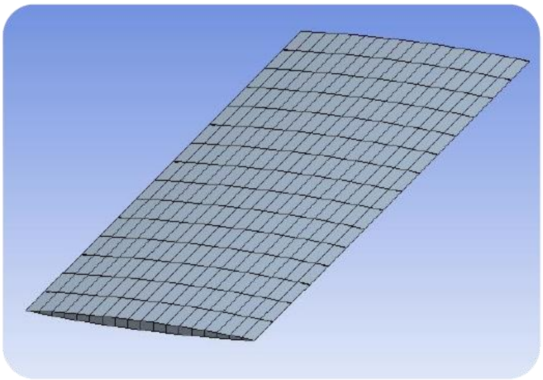


Figure 11: Finite Element Mesh model of Goland Wing

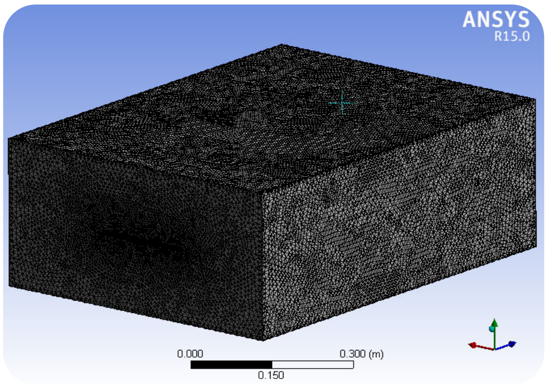


Figure 12: CFD mesh model for the wing geometry with surrounding air volume

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

1. Goland 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 behaviour under aerodynamic loads.

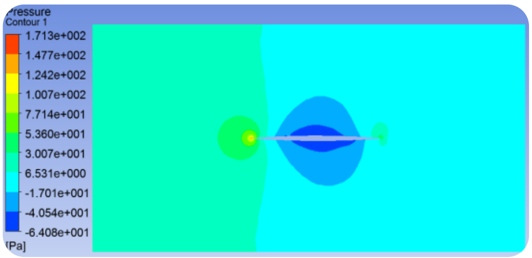


Figure 13: Pressure distribution plot at the mid-section of the wing

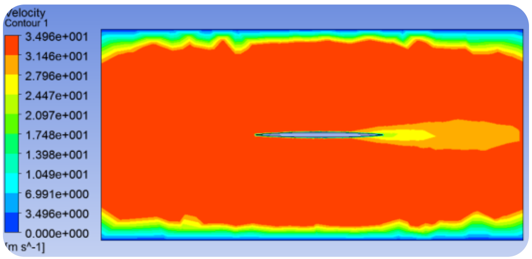


Figure 14: Velocity distribution plot at the mid-section of the wing

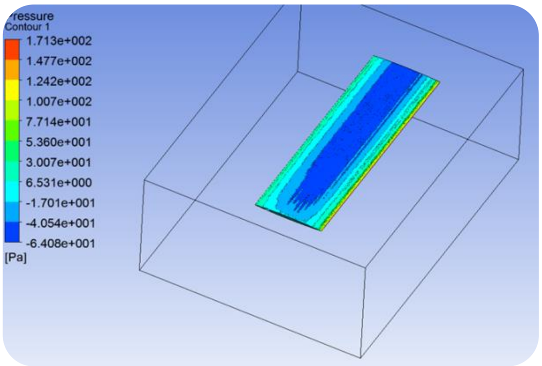


Figure 15: Aerodynamic loads acting on the Wing wall

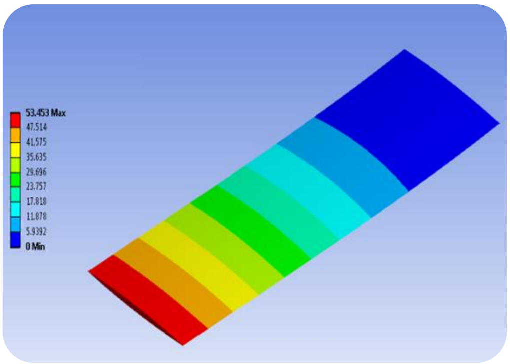


Figure 16: Wing deflection due to aerodynamic load

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 which 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 behaviour 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 which 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. The run time can be minimized by using higher core systems. 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

	8 Core		16 Core		32 Core	
No. of elements	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

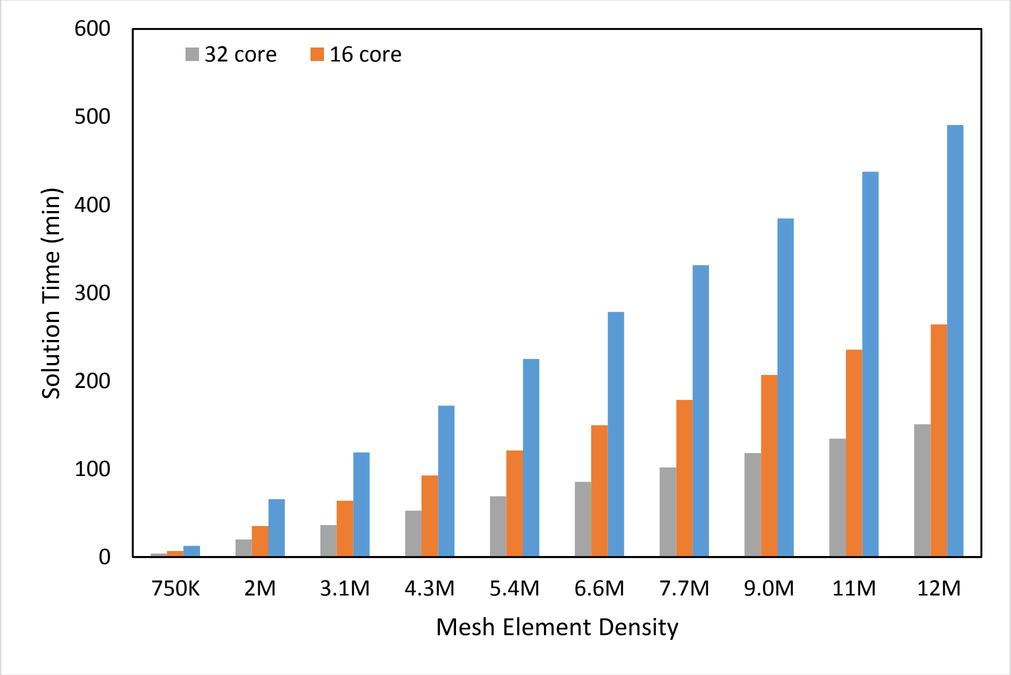


Figure 17: Comparison of Solution time (min) for different element density.

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 modelling 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 behaviour of the wing which 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

7. 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.
8. 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-trails or resubmission of the same simulation runs.

9. 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).
10. 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.
11. 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.
12. 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.
13. 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

5. 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.
6. 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.
7. 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.

Team 165:

Wind Turbine Aerodynamics Study

> *"The HPC cloud provided e 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.
Resource Provider: ProfitBricks
HPC Expert: Burak Yenier, Co-Founder, CEO, UberCloud

USE CASE

With an ever increasing energy crisis occurring in the world it will be 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 the solar energy, the utilization cannot be affected by the climate and weather. A wind turbine is the device which extracts the energy from the wind and converts into electric power.

The case study refers to the evaluation of the wind turbine performance using a Computational Fluid Dynamics (CFD) approach. Standard wind turbine designs are considered for this UberCloud experiment. The CFD models were generated in the ANSYS CFX environment. The simulation platform was built on a 62 core HPC cloud server with ANSYS 15.0 modelling environment. The cloud environment was accessed using a VNC viewer through the web browser. The 62 core server with 240 GB ram installation was at ProfitBricks. The CPU and the 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.

Process Overview

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

1. The standard wind turbine designs which are in the 3D CAD geometry format are imported in the ANSYS Design modeller. The model is modified by creating the atmospheric air volume around the wind turbine design.
2. Develop the CFD model with atmospheric air volume surrounding the wind turbine in ANSYS Mesh Modeller.
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 & solution algorithm. This portion of setup is mainly related to define 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 ANSYS CFX simulation setup is solved in the HPC Cloud environment. The simulation model needs to be precisely defined with good amount of fine mesh elements around the turbine blade geometry. The following snapshot highlights the wind turbine geometry considered and ANSYS CFX mesh models.

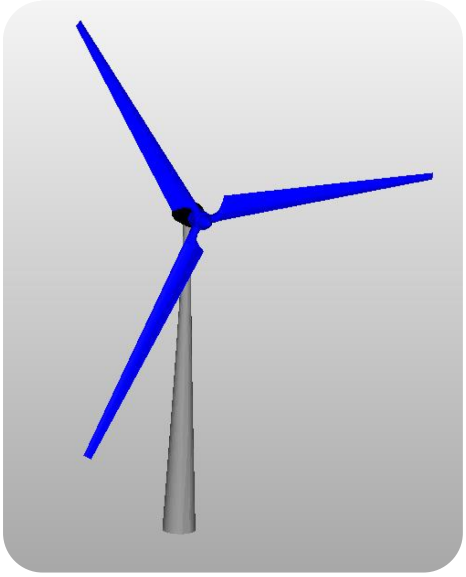


Figure 18: Wind turbine Geometry

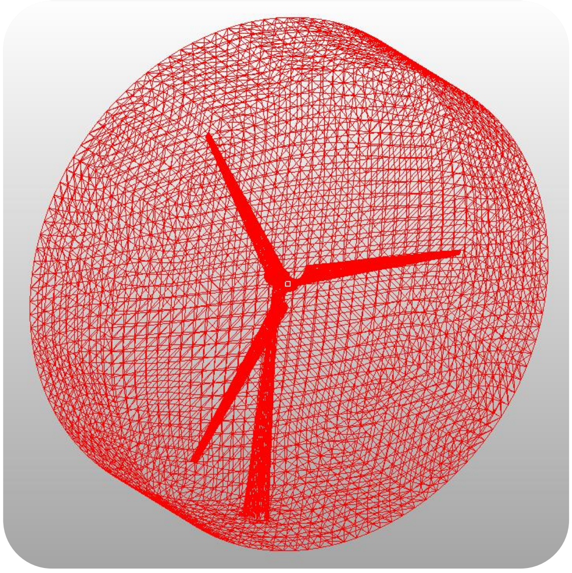


Figure 19: Computation Fluid dynamics model of wind turbine

The CFD simulation is performed to evaluate 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.

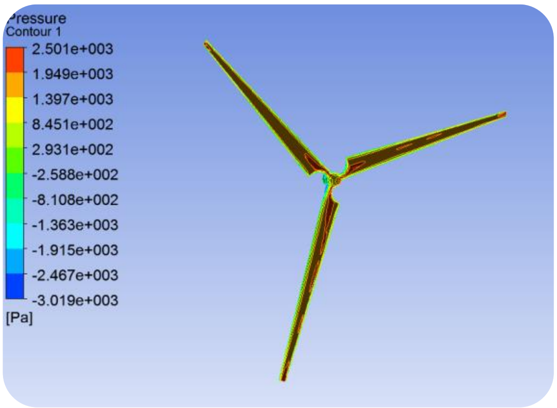


Figure 20: Plot of pressure distribution on the wind turbine blades

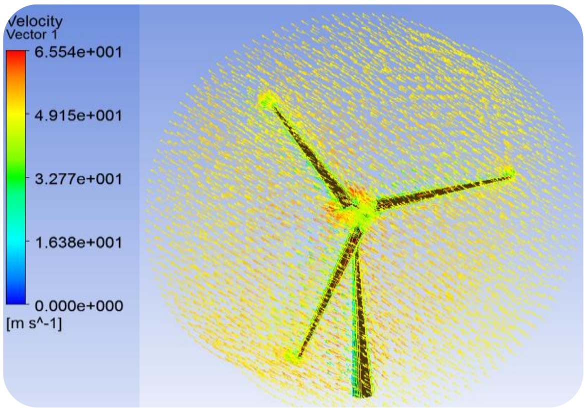


Figure 21: Vector plot of velocity profiles around the wind turbine blades

HPC Performance Benchmarking

The aerodynamic study of wind turbine blades is carried out in the HPC environment which 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 parallel computing environments & 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 has been evaluated for their performance on the 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.

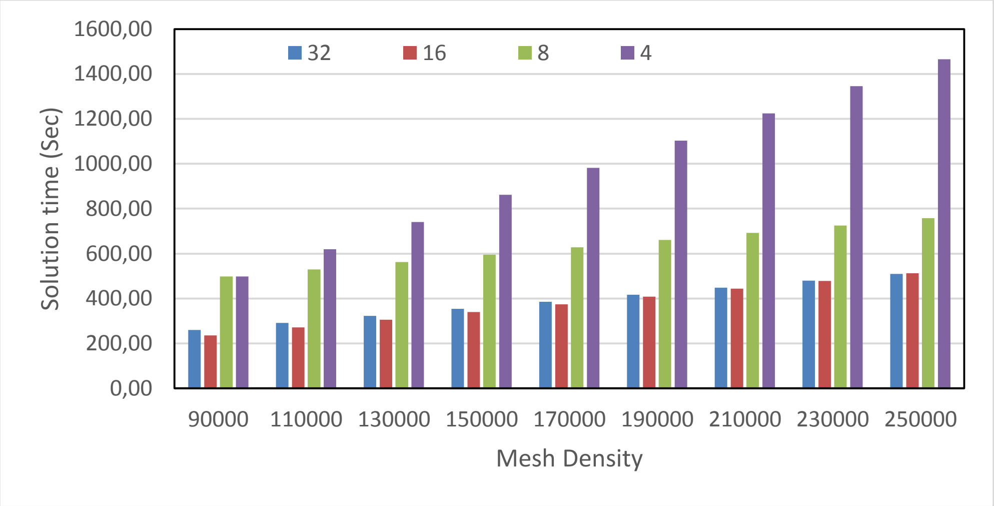


Figure 22: Solution time for different element density using Intel MPI parallel computing platform

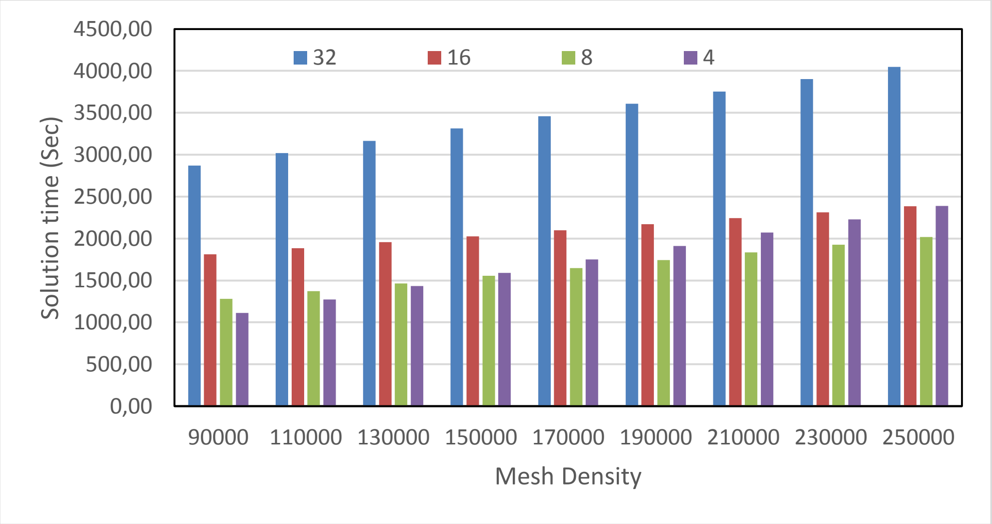


Figure 23: Solution time for different element density using PVM MPI parallel computing platform

Figures 5 & 6 show the plots of solution time required for different mesh density where the simulation models are solved using Intel MPI and PVM parallel computing platform. The Intel MPI parallel computing platform shows a stable performance with the solution time reducing with increases in the number of CPU cores (Ref. Figure 5). The PVM parallel computing platform is highly unstable with higher solution time required for higher CPU cores. The simulation time required for a

32 core configuration is higher than the time required for a 4 core configuration for the same simulation model (Ref. Figure 6).

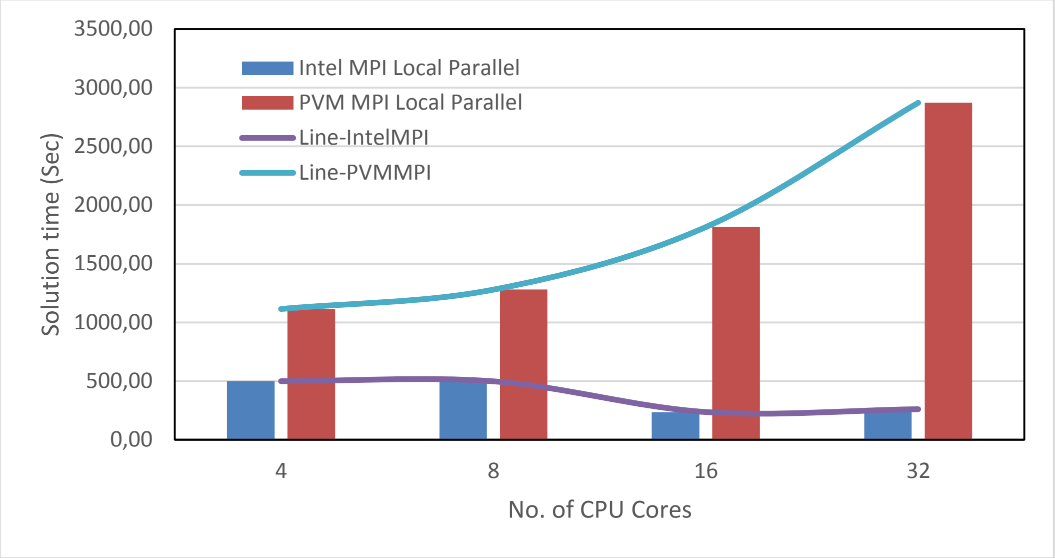


Figure 24: Performance comparison of different parallel computing platforms for a moderate mesh density (90K)

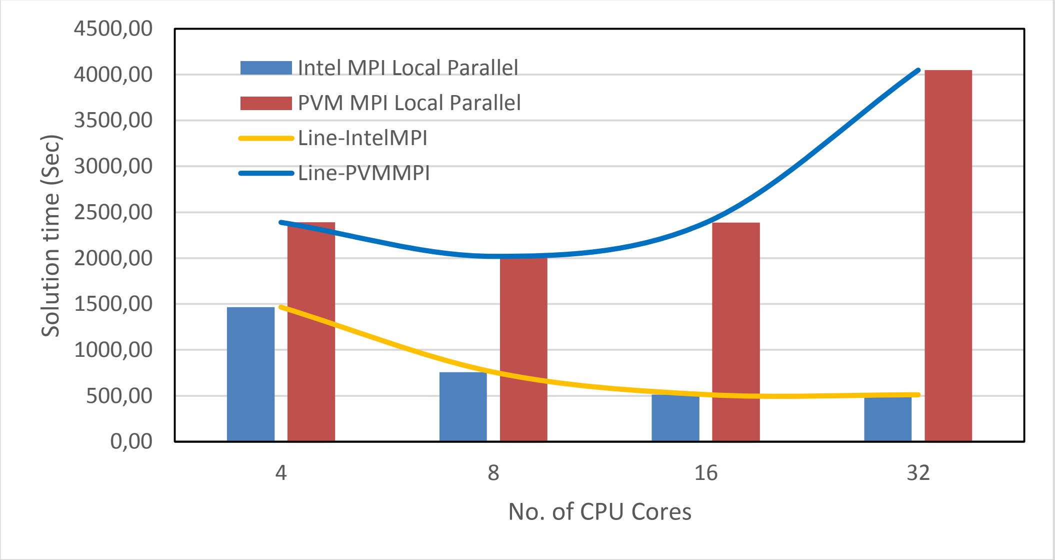


Figure 25: Performance comparison of different parallel computing platforms for a mesh density of 250K

Figures 7 & 8 highlight the HPC performance comparison done with different parallel computing environments. The simulation model was built with mesh densities of 90K and 250K and separate experiments were conducted with the simulation models by submitting the simulation jobs for different CPU cores. The solution time required to solve the simulation model is captured for different parallel computing platforms. The Intel MPI platform shows a better performance as the simulation time reduces with increasing number of CPU cores and is stable when compared to the PVM parallel computing environment.

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 the mentioned effort is one time effort and will benefit the 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 modelling 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 is the simulation time required and hence 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-trails or resubmission of the same simulation runs.
3 The computation requirement for a very fine mesh (2.5 million cells) is high, which is near to impossible to achieve on a normal workstation. The HPC cloud provided this feasibility to solve very fine mesh models and the simulation time drastically reduced thereby providing an advantage of getting the simulation results done within 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 in increasing the productivity in the simulation setup effort and thereby providing a single platform to perform the end-to-end simulation setup.
5 The experiments performed in the HPC Cloud environment showed the possibility and gave extra confidence 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 that enabled continuous monitoring of the job in progress without any requirement to log-in to the server 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 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.

APPENDIX: UberCloud Containers: Brief Introduction

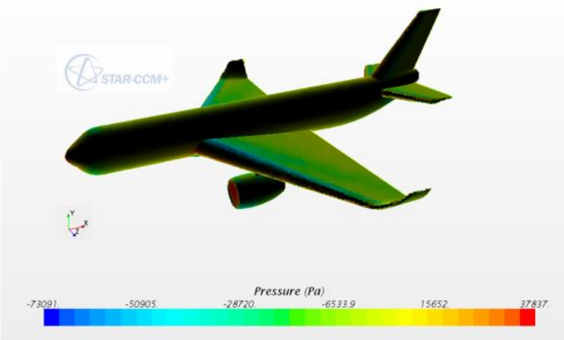
UberCloud Containers are ready-to-execute packages of software. These packages are designed to deliver the tools that an engineer needs to complete his task in hand. The ISV or Open Source tools are pre-installed, configured, and tested, and are running on bare metal, without loss of performance. They 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.

Team 166:

Computational Fluid Dynamics Study on Flight Aerodynamics



> *"Combination of CFD tools & HPC Cloud showcased the possibilities & easiness in performing large system level simulations."*

MEET THE TEAM

End-User/CFD Expert: Praveen Bhat, Technology Consultant, INDIA
Software Provider: CD-adapco, STAR-CCM+
Resource Provider: Nephoscale
HPC Expert: Burak Yenier, Co-Founder, CEO, UberCloud.

USE CASE

Making changes to the airplane configuration can be expensive and sometimes dangerous. There are many reasons to make changes to the airframe of an aeroplane. These include improvements to the airfoil to reduce the profile drag and increase the lift, and wing tip modifications to reduce drag; additions of stores and external components such as landing gear covers; increasing load carrying capacity etc. One method to reduce the expenses is to test the proposed modifications in all possible scenarios. Testing is very time consuming and can be challenging when there is a short time frame defined.

Aerodynamics analysis methods based on computational fluid dynamics (CFD) can reduce testing time by rapidly screening models and pre-selecting only the promising ones for further testing (wind tunnels, scale models & flight testing).

The present case study refers to aerodynamics study of aircraft using CFD approach. Standard aircraft designs are considered for this experiment. The CFD models were developed using CD-adapco STAR-CCM+ simulation software. The simulation platform was built on a 32 core HPC cloud with Star-CCM+ 9.06 modeling environment. The Cloud environment was accessed using a VNC viewer through web browser. The 32 core machine was at Nephoscale cloud. It is a bare metal server and has 64 GB RAM. The machines has 2 CPU's and 8 physical cores on each CPU. The CPU's are Intel® Xeon® CPU E5-26500 @ 2.00 GHZ. CD-adapco STAR-CCM+ code was running in the new UberCloud application container, see the Appendix for a short description.

Process Overview

The following defines the step by step approach in setting up the CFD model in the Star-CCM+ 9.06.

1. The standard aircraft designs which are in the 3D CAD geometry format are imported in the Star-CCM+ modelling environment. The model is modified by creating the atmospheric air volume around the aircraft design.
2. Develop the CFD model with atmospheric air volume surrounding the aircraft using Star-CCM+ pre-processing module.
3. Create the instances /scenes to view the geometry and mesh models.
4. Define the model parameters, fluid properties, and boundary conditions.
5. Define the solver setup & solution algorithm. This portion of setup is mainly related to define type of solver, convergence criteria and equations to be considered for solving the aerodynamic simulation.
6. Define the output results required for comparison plots & graphs.
7. Perform the CFD analysis and review the results.

The Star-CCM+ simulation models are solved in the HPC Cloud environment. The simulation model needs to be precisely defined with good amount of fine mesh elements around the aircraft geometry. The following snapshot highlights the aircraft designs considered with the 3D CAD geometry and Star-CCM+ CFD mesh model.

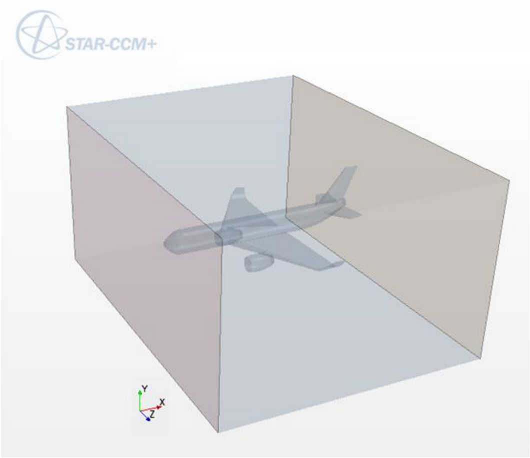


Figure 26: 3D CAD model of Aeroplane with surrounding atmosphere

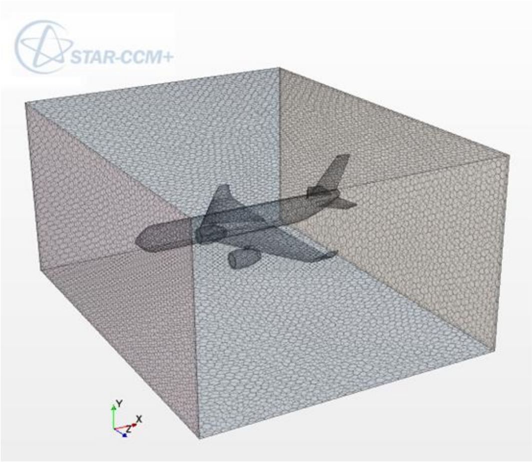


Figure 27: CFD mesh model of aeroplane and atmospheric volume

The CFD simulation is performed to evaluate the pressure distribution and velocity profiles around the aircraft. The aircraft is subjected to an average wind speed of 900 km/hr. The following plots highlight the pressure and velocity distribution around the aircraft structure.

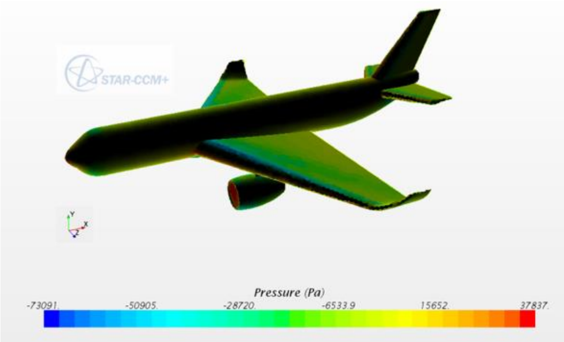


Figure 28: Plot of pressure distribution on the aircraft

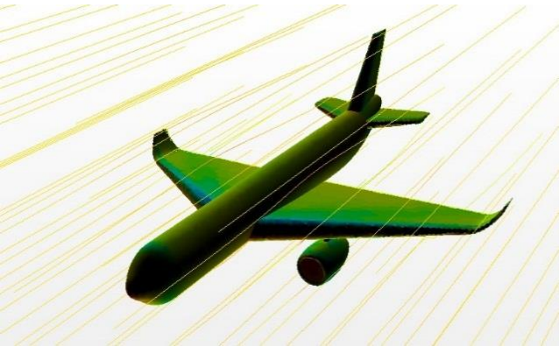


Figure 29: streamline plot of velocity profiles around the aircraft

Figures 3 & 4 highlight the pressure distribution and velocity streamlines around the aircraft. The pressure distribution values are relative to the outlet pressure.

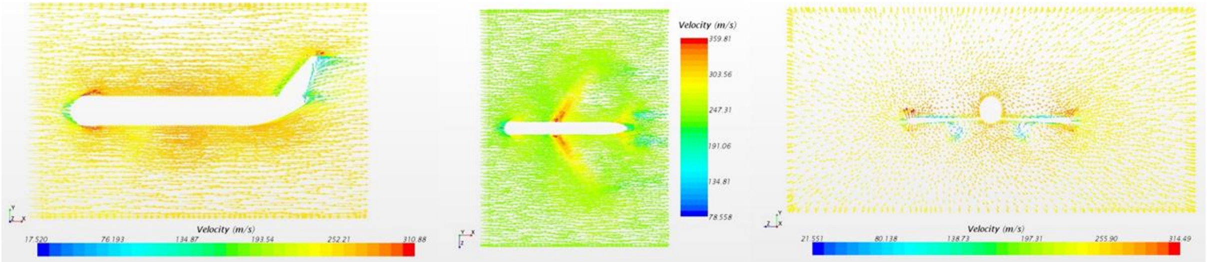


Figure 30: Velocity vector plot along different sections of the aircraft. (a) Side section; (b) Top section; (c) Front section.

Figure 5 highlights the velocity distribution around the aircraft. The section view showcases the turbulence effect of the wind on the aircraft and also the velocity vectors of wind flow. It is observed that the flow becomes laminar as the wind flows around the aircraft.

HPC Performance Benchmarking

The aerodynamic study on the aircraft is carried out on a 32 core server with CentOS Operating System and Cd-adapco Star-CCM+ simulation package. The server performance is evaluated by submitting the simulation runs for different mesh densities. The simulation runs were performed by varying the mesh density and submitting the jobs for different numbers of CPU cores.

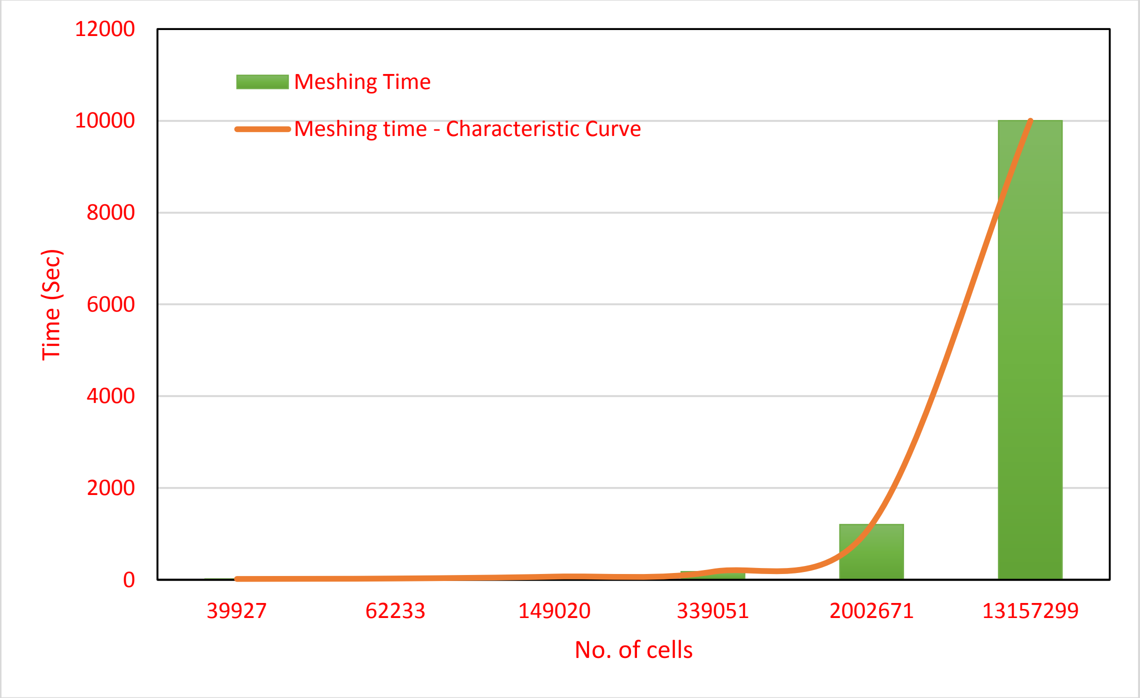


Figure 31: Meshing time computed for different mesh densities for a 32 core configuration

Figure 6 shows the time required for meshing the aircraft model for different mesh densities. The study included meshing of the model for the defined number of elements. The study was performed to understand and evaluate the system capabilities to generate very fine mesh models. On the 32 core configuration the meshing time grows exponential with growing number of elements. The system uses an average of 15% of its RAM to mesh the higher number of cells. Hence highly fine mesh models can be generated through HPC resources which would be not possible to achieve in a normal dual core workstation.

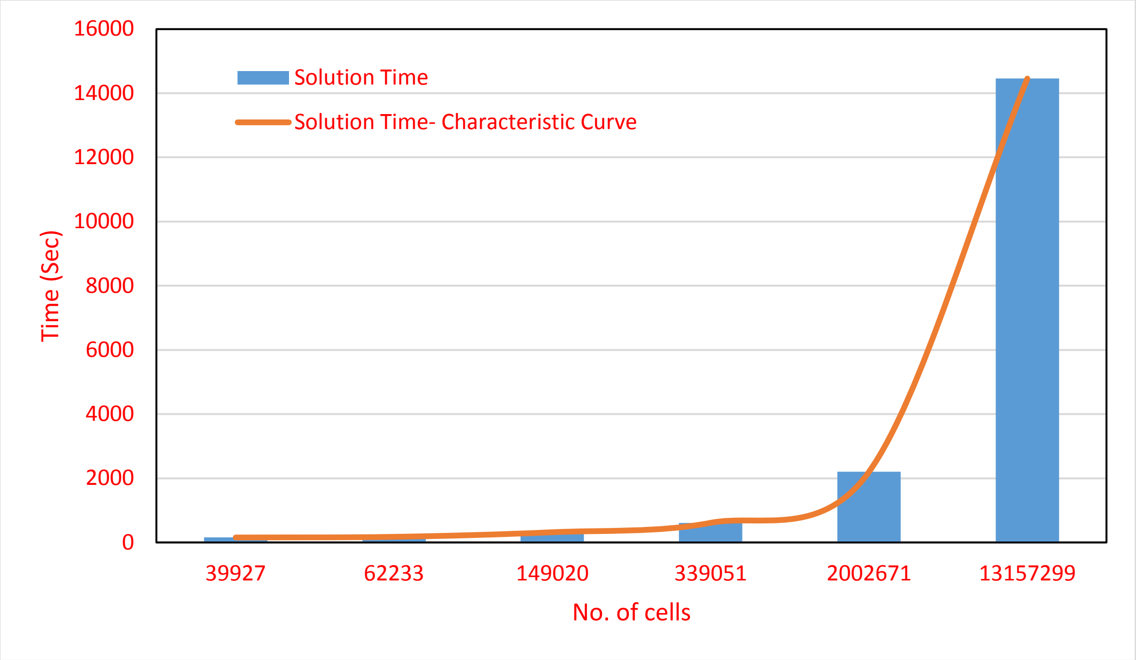


Figure 32: Solution time computed for different mesh densities for a 32 core configuration

Figure 7 shows the runtime required for the simulation runs in a 32 core configuration. On the 32 core configuration the simulation time grows exponential with growing number of elements. Figure 8 shows the simulation time required for solving a model with 149K elements. The time required for solving the model with 32 CPU cores is six times less than the time required for solving the same model in a single core system. The HPC Cloud capability provided the advantage of solving highly fine mesh model in a less time frame.

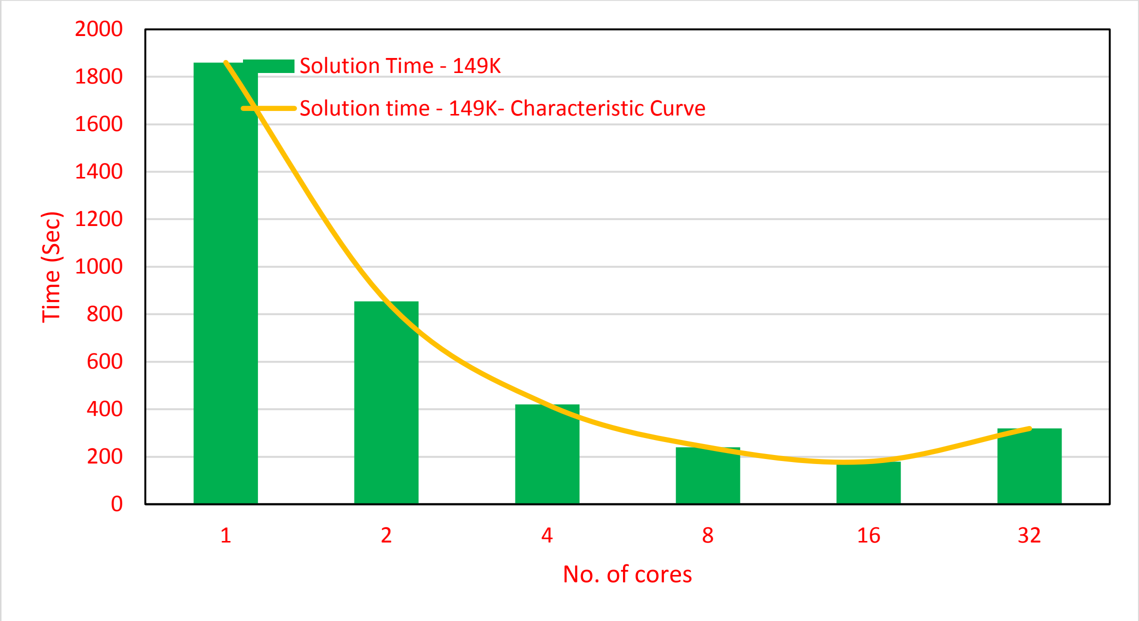


Figure 33: Solution time computed for simulation model with 149K elements for different numbers of CPU cores

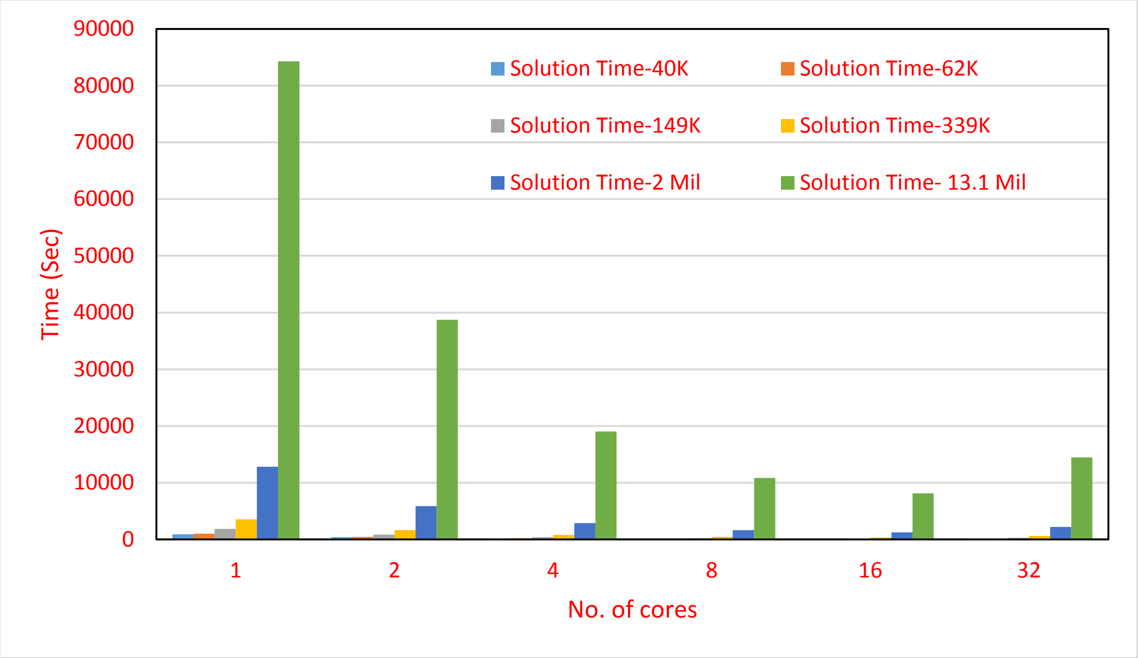


Figure 34: Comparison of solution time required for different mesh densities under different CPU configurations

Figure 9 compares solution times required for different mesh granularity on different numbers of CPU cores. Simulation times for higher numbers of cells at higher core counts are much smaller than solution times required for lower numbers of CPU cores. The combination of CFD tools and HPC Cloud enabled the possibilities and easiness of system level simulations.

Effort Invested

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

UberCloud support: 2 hours on regular system management tasks to build Star-CCM+ image, transfer it on to the server, launch the container with the image, monitoring the server etc.

Resources: ~600 core hours used for performing various iterations in the simulation experiments**.**

CHALLENGES

The project started with setting up the CD-adapco Star-CCM+ 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 level of challenges faced were related to the technical complexity. This involved accurate prediction of aircraft behaviour under aerodynamic loads which is achieved through defining the appropriate element size for the mesh model. The finer the mesh the higher is the simulation runtime required and hence the challenge was to perform the simulation within the stipulated timeline.

BENEFITS

1. The HPC cloud environment with CD-adapco Star-CCM+ made the process of model generation easy, with process times reduced drastically because of the HPC resource.
2. The mesh models were generated for different numbers of cells 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.

3. The high computation requirement for a very fine mesh (13.1 million cells) makes it (almost) impossible to use a normal workstation. The HPC cloud provided this feasibility to solve very fine mesh models, and the simulation time was drastically reduced thereby providing an advantage of getting the simulation results within acceptable run time (~4 hrs).
4. The use of Star-CCM+ helped in performing different iterations on the experiments by varying the simulation models. This further helped in increasing the productivity in the simulation setup effort and 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 the simulations remotely in the cloud. The different simulation setup tools were readily installed in the HPC environment and this enabled the user to access the tools 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 smooth execution of the project with easy access to server resources, and the regular UberCloud auto-update module through email provided huge advantage to the user enabling continuous monitoring of the job in progress without any requirement to log-in to the server and check the status.

CONCLUSION & RECOMMENDATIONS

1. The HPC Cloud with UberCloud containerized Star-CCM+ on Nephoscale cloud resources was an excellent simulation environment for performing advanced computational experiments that involve high technical challenges and require high performance hardware resources to perform the simulation experiments.
2. There are different high-end software applications which can be used to perform complete system modeling. Star-CCM+ with HPC 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 CD-adapco Star-CCM+ helped in speeding up the simulations and completing 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.


UberCloud Experiment

Wolfgang Gentzsch and Burak Yenier

Thank You

http://www.TheUberCloud.com

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

If you, as an end-user, would like to participate in this Experiment to explore hands-on the end-to-end process of on-demand Technical Computing as a Service, in the Cloud, for your business then please register at: http://www.theubercloud.com/hpc-experiment/

If you, as a service provider, are interested in promoting your service/product at the UberCloud Exhibit then please register at http://exhibit.hpcexperiment.com/about/join/

1st Compendium of case studies, 2013: https://www.theubercloud.com/ubercloud-compendium-2013/
2nd Compendium of case studies 2014: https://www.theubercloud.com/ubercloud-compendium-2014/

HPCwire Readers Choice Award 2013: http://www.hpcwire.com/off-the-wire/ubercloud-receives-top-honors-2013-hpcwire-readers-choice-awards/
HPCwire Readers Choice Award 2014: https://www.theubercloud.com/ubercloud-receives-top-honors-2014-hpcwire-readers-choice-award/

In any case, if you wish to be informed about the latest developments in technical computing in the cloud, then please register at http://www.theubercloud.com/. It's free.

Please contact UberCloud at help@theubercloud.com before distributing this material in part or in full.