Heavy Duty Structural Analysis using Abaqus in the Cloud

Frank Ding and Wolfgang Gentzsch

Simpson Strong Tie and The UberCloud

Abstract*: Concrete anchors are critical fastening elements in transferring structural load to concrete foundation. In this case study, cast-in-place mechanical anchors with local reinforcement were modeled using SIMULIA Abaqus. Steel anchor bolt and rebars were modeled using elastic-plastic material, and concrete slab was modeled using concrete damage plasticity model. Because of the large model, contact, and damage failure procedure were involved, parallel computing had to be employed to speed up the solution process. The UberCloud HPC Experiment provided a great opportunity to tap into HPC resource in the cloud.*

Keywords*: Abaqus, Cast-in-place mechanical anchors, Concrete, HPC, Cloud, UberCloud, NICE DCV*

1. Introduction

Concrete anchors are critical fastening elements in transferring structural load to concrete foundation (Widianto, 2008). Various means of anchoring are in use today. One way of differentiating anchor systems is by the way they are installed. Cast-in-place components are secured in the formwork prior to casting. Post-installed anchors are installed in holes drilled into cured concrete base material. Another way to categorize anchor systems is by load transfer mechanism. Load transfer mechanisms are typically identified as mechanical interlock, friction or bond. Headed studs are examples of using mechanical interlock mechanism. Expansion anchors employ friction as load-transfer mechanism. Chemical interlock is the load transfer mechanism employed by bonded anchors.

Figure 1. Example of concrete anchorage in application

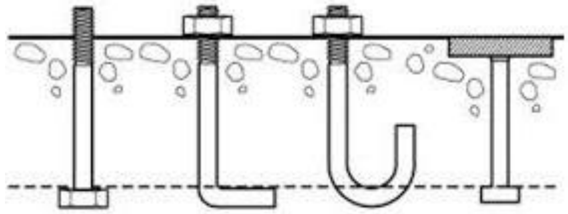

Figure 2. Example of cast-in-place anchors

Ever since the introduction of the strength designs provisions in the 2000 IBC and ACI 318 Appendix D, anchorage to concrete has been a challenge for designers, building officials and manufacturers. With the increasing demand for high-density housing and urban infill projects, designers are now faced with anchoring multi-story wood-framed shear walls to relatively thin elevated concrete slabs (typically referred to as podium slabs). Overturning tension anchorage forces at the ends of shear walls in these projects can routinely be in the 40 kip range and even get as high as 60 kips or more. To meet the challenge, local reinforcement was used to increase the anchoring capacity. Both simulation and testing were employed to verify design concept.

Figure 3. Anchor tensile test with concrete breakout cone

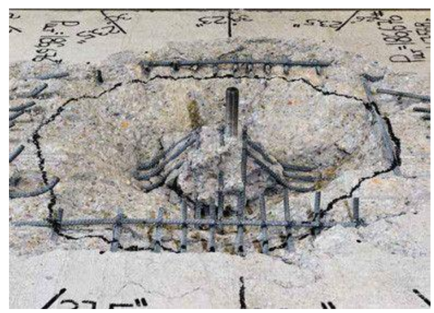

Figure 4. Podium anchorage with local reinforcement

In this study, cast-in-place mechanical anchors with local reinforcement were modeled using Abaqus. Steel anchor bolt and rebars were modeled using elastic-plastic material, and concrete slab was modeled using concrete damage plasticity model. Because of large model, contact, and damage failure procedure were involved, parallel computing had to be employed to speed up the solution process. The UberCloud HPC Experiment provided a great opportunity to tap on the HPC resource in the cloud.

2. The UberCloud Experiment – Abaqus in the Cloud

The UberCloud Experiment started in July 2012 (Gentzsch, 2016), with a discussion about cloud adoption in technical computing and a list of technical and cloud computing challenges and potential solutions. The UberCloud team decided to explore these challenges further, hands-on, and the idea of the UberCloud Experiment was born, also due to the excellent support from INTEL generously sponsoring these experiments!

Especially small and medium enterprises in digital manufacturing strongly benefit from using technical computing in the cloud. By gaining access on demand from their desktop workstations 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; and the cost benefit by only paying for what is really used. These are benefits that increase a company's innovation and competitiveness.

Tangible benefits like these make technical computing - and more specifically technical computing as a service in the cloud - very 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; availability of best suited resources; and lack of standardization, transparency, and cloud expertise. However, in the course of this experiment, as we followed each of the (so far) 188 cloud experiments closely and monitored their challenges and progress, we've got an excellent insight into these roadblocks, how the teams have tackled them, and how we are now able to reduce or even resolve them.

2.1 The Cloud Experiment Team

For the remainder of this paper we concentrate on CAE Cloud experiment #47, Heavy Duty Structural Analysis using Abaqus in the Cloud (Ding, 2013). As it turned out during the cloud experiment, additional expertise in HPC and in remote visualization became necessary which was easily added from the wider UberCloud community of thousands of experts. The team consisted of:

End User – Frank Ding

Frank Ding is the Engineering Analysis and Computing Manager at Simpson Strong-Tie in Northern California. In this experiment he represented the CAE end user focused on creating structural products that help people build safer and stronger homes and buildings. Simpson Strong-Tie is considered a leader in structural systems research, testing and innovation, and one of the largest suppliers of structural building products in the world.

Software Providers – Matt Dunbar, Antonia Arena, Cynthia Underwood

Matt Dunbar is the Chief Architect and CAE technical specialist at SIMULIA Dassault Systems, in its Rhode Island facilities. He represented the Abaqus application level expertise in this experiment.

Antonio Arena is the Solutions Architect at NICE Software and in this project was the network and "middleware" team expert, and Cynthia Underwood is IT consultant and subject matter expert at NICE Software.

Resource Provider – Steve Hebert and Rob Sherrard

Steve Herbert is one of the founders and CEO of Nimbix, located in Texas, which in this team was the provider of cloud-based High Performance Computing infrastructure and applications. Rob Sherrard, is the other co-founder of Nimbix and VP of Service Delivery.

HPC Expert and Team Manager – Sharan Kalwani

Sharan Kalwani is the director of the HPCC group at Michigan State University and in this project was the overall Subject Matter Expert, project management, solution expertise and team lead.

UberCloud Team Mentor - Dennis Nagy

Finally, Dennis Nagy, Principal at Beyond CAE, was the UberCloud Team Mentor for this experiment who took the team through all the CAE Cloud hurdles, with a knowing hand.

2.2 The Use Case: Structural Analysis using Abaqus in the Cloud

In Round 1 of the HPC Cloud experiment, the team recognized that computational use cases could indeed be submitted and executed successfully in the HPC Cloud. The objective of Round 2 – including this cloud experiment - was to explore the following:

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

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

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

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

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

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

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

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

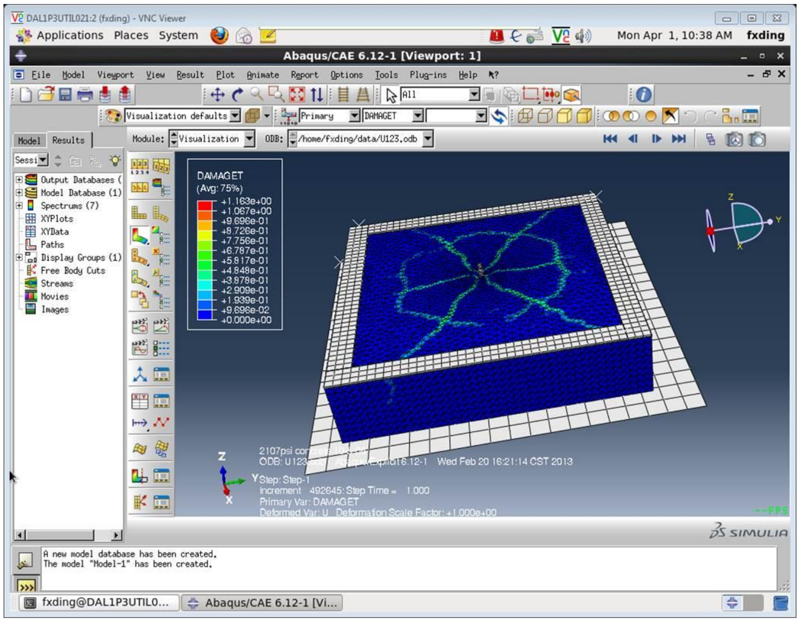

Figure 5. Abaqus/CAE post-process session through DCV

Table 1. Cast-in-Place Mechanical Anchor Concrete Anchorage Pullout Capacity Analysis (FEA stats)

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

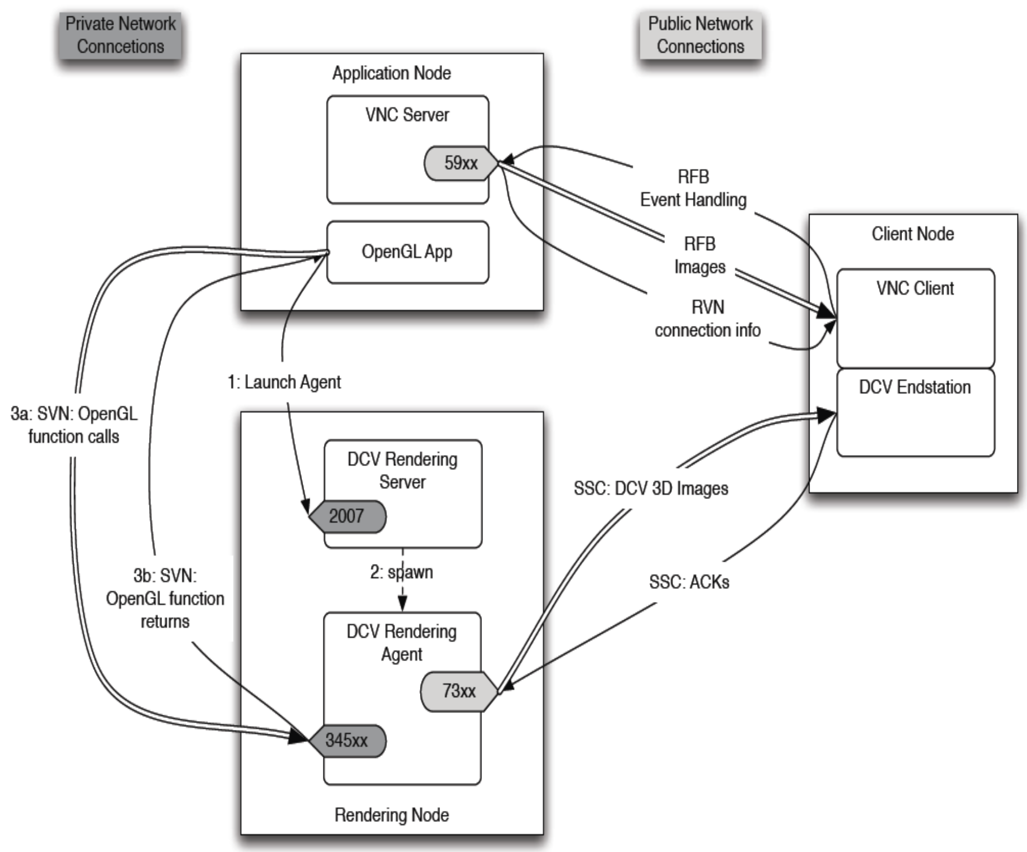

Figure 6. DCV layer setup (post processing underlying Infrastructure, cloud end)

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

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

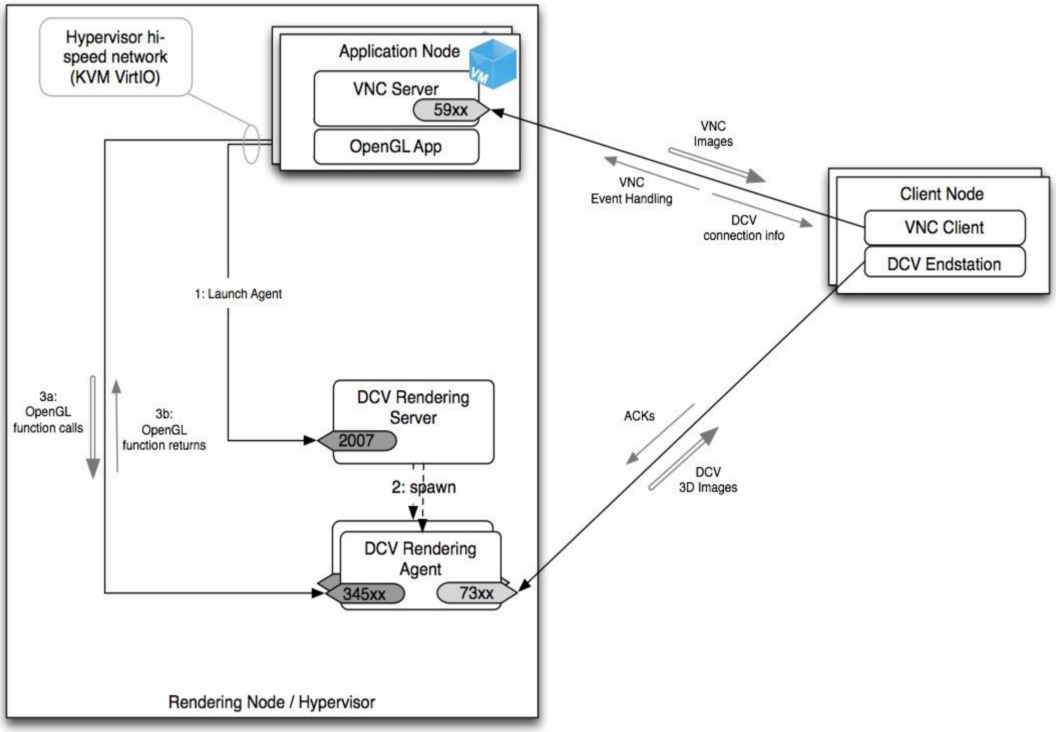

Figure 7. DCV enabled post processing (end user view)

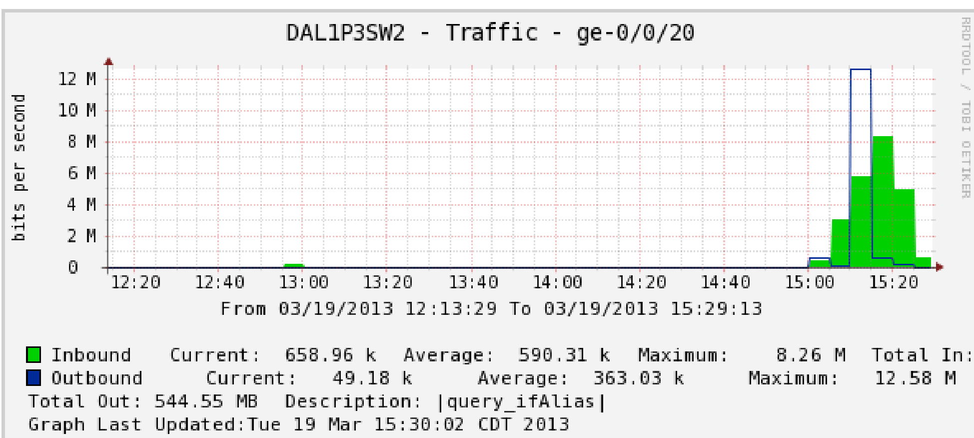

Figure 8. Ingress/egress test results/profile

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

3. Conclusions and Recommendations

Here is a summary of the key results found during our UberCloud Round 2 computing experiment:

- End point internet bandwidth variability: Depending on when it is conducted, a vendor neutral test applet result ranges from 1Mbps ~ 10Mbps. The pipe bandwidth was expected to be 20 Mbits/sec, but when it was shared by the office site using normal enterprise applications such as Exchange server, Citrix, etc., such variation was not conducive to a qualitative end user experience.
- Switching to another pipe (with burst mode of 50 to 100 Mbits/sec): More testing showed that the connection was not stable, and Abaqus /Viewer graphics windows freeze was experienced after being idle for a while. This required local IT to troubleshooting the issue.
- There were no significant differences between Windows or Linux hosted platforms. The NICE DCV/EnginFrame is a good platform for remote visualization if a stable internet bandwidth is available. Some of the parameters for the connection performance:
  - VNC connection line-speed estimate: 1 ~ 6 Mbps, RTT ~ 62 ms
  - DCV bandwidth usage: AVG 100 KB ~ 1 MB
  - DCV Frame Rate: 0~10 FPS, >5 FPS acceptable, >10 FPS smooth
- We tried both Linux and Windows desktop. Because of the bandwidth randomness & variability, it was not possible create a good baseline to compare the performance of the two desktops.
- The graphics cards did not have any impact on the end user experience. However the model size and graphic image pixel size perhaps play a major role and the current experiment did not have enough time to study and characterize this issue. The Abaqus model used in this test case does not put much demand on the graphics card. We've seen only 2% usage on the card.
- There was usually sufficient network capacity and bandwidth at the cloud serving end.
- The "last mile" delivery or capability at the end user site was the most important and perhaps only determining factor influencing the end user experience and perception.
- Beyond the cloud service provider, a local or end user IT support person with network savvy is perhaps a necessary part of the infrastructure team in order to deliver robust and repeatable post-processing visual delivery. This incurs a cost.
- The security aspect could not be tested, as the time and effort required were not sufficient in the time allotted.
- Part of the end user experience learned from Round 1 was to better document the setup which can be found in the Appendix and clearly shows a smooth and easy to follow up flow.

Major single conclusion and recommendations: Any site that wishes to benefit from this experience needs to prioritize the “last mile” issue.

Table 2. End User Experience Observations

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

Table 3. Bandwidth Usage

	Average (KB)	Peak (MB)
Image Quality = 10	450	1.1
Image Quality = 80	750	2.1

4. References

1. F. Ding, M. Dunbar, S. Hebert, R. Sherrard, and S. Kalwani, “Heavy Duty Abaqus Structural Analysis Using HPC in the Cloud,” UberCloud Compendium of case studies, 2013 & 2014.
2. W. Gentzsch and B. Yenier, “The UberCloud Experiment,” 2016, https://www.theubercloud.com/hpc-experiment/.
3. Widianto, J. Owen, and C. Patel, “Design of Anchor Reinforcement in Concrete Pedestals” Structures Congress, 2010, http://ascelibrary.org/doi/abs/10.1061/41130%28369%29225.