ABAQUS High Performance Computing Environment at Nokia

Juha M. Korpela

Nokia Corporation

Abstract: The new commodity high performance computing (HPC) hardware together with the recent ABAQUS performance enhancements, have made it possible to build high performing ABAQUS computing environment cost effectively. Since ABAQUS supports the distributed memory parallel (DMP) hardware, the usage of the cluster type of hardware has become very interesting.

It is an easy task to purchase HPC hardware, but successful utilization of high performing hardware requires also many other issues taken into account. The important issues are for example setting up hardware, operating system and licensing for ABAQUS. Also UNIX user environment settings, ABAQUS installation schemes and batch queue system need to be configured properly. Utilization of ABAQUS in a global enterprise organization creates some additional challenges to the deployment of cluster hardware. There have been many IT projects inside the company that have made it possible to build unified global UNIX/Linux based high performing ABAQUS computing environment.

The ABAQUS/Explicit solver dominates the usage of ABAQUS software as it has proven to be very effective in drop test and quasi-static simulations of mobile devices. The ABAQUS/Standard solver is still used mainly for static simulation cases.

Prior to the ABAQUS version 6.5, all ABAQUS hardware used in the company was shared memory parallel (SMP) machines. The first cluster configurations were tested with version 6.4 in the beginning of year 2004. During the spring 2005, the first cluster installation was taken into production use. Currently the vast majority of computing power is based on the cluster type of hardware.

This paper presents the current state of ABAQUS computing environment set-up at the company.

Keywords: 3DXML, abaqus_v6.env, batch queue, benchmarking, cluster, distributed memory parallel, documentation, DMP, drop test simulation, hardware, High Performance Computing, HP-MPI, HPC, installation, licensing, LAM/MPI, Linux, message passing interface library, modules, MPI, Open MPI, preprocessing, postprocessing, Python, rsync, shared memory parallel, SMP, Sun Grid Engine, SGE, tight integration, UNIX

1. Introduction

At Nokia, the ABAQUS finite element software is used mainly for structural analysis cases. The vast majority of simulation cases are drop test simulations using ABAQUS/Explicit software. The purpose of drop test simulations is to guarantee the mechanical reliability of mobile devices made by Nokia.

1.1 General mechanical simulation working mode

There are three business groups related to mobile devices in the company, see Figure 1. These groups are: Mobile Phones, Multimedia and Enterprise Solutions. A horizontal group called Technology Platforms manages the delivery of technologies and platforms to the other business groups. Those platforms are displays, printed wiring board (PWB) components, audio components, connectors and batteries, for instance.

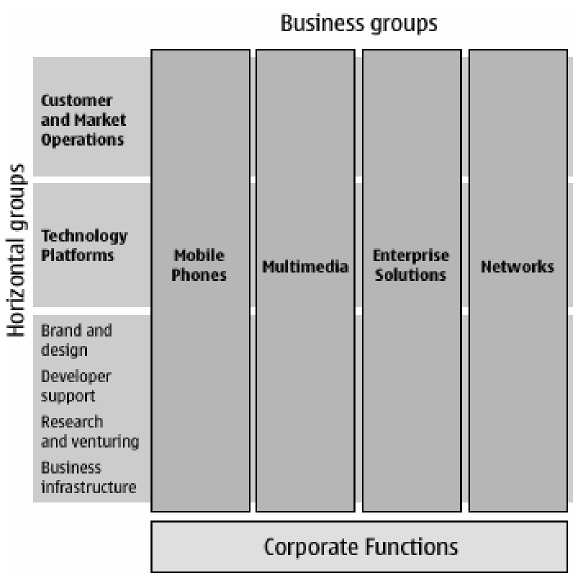

Figure 1. Organizational structure of the company.

Every business group has a group of analysts doing the finite element simulations for mobile devices. Technology Platforms are responsible to deliver finite element analysis related information of platform deliverables to the business groups. This information includes for example finite element models for drop test simulations, implementation guidelines, post-processing scripts and failure criteria for those components.

Drop test handbook contains general guidelines how to do drop test simulation (DTS) of mobile devices. Figure 2 shows a snapshot of the table of contents of a drop test handbook.

TABLE OF CONTENTS

Figure 2. An example of the table of contents of drop test handbook.

1.2 CAD geometry and pre-processing

The geometry of mobile devices is created using 32-bit Windows based CAD-software and finite element models are created mainly using 32-bit Windows based pre-processing tools. The Windows version will soon be updated to 64-bit Windows XP for all users.

There is a component library available where all finite element models of Technology Platforms can be downloaded from. The models are stored in ABAQUS input file format. These input files contain for example the finite element mesh, material models, interaction definitions and output requests.

2. ABAQUS computing environment

ABAQUS/Explicit dominates usage of ABAQUS at the company. About 70% of all jobs are /Explicit jobs and 95% of processor time is consumed by /Explicit jobs. The ABAQUS/Explicit is used for drop test simulations of mobile devices and many kind of quasi-static cases. The ABAQUS/Standard is used mainly for static simulation cases.

Utilization of ABAQUS in a global enterprise organization creates many challenges to the computing environment. It’s not enough just to purchase hardware, but for example user environment settings, ABAQUS installation scheme and batch queue system need to be configured properly. Currently the ABAQUS computing environment is based on Unix/Linux systems. Because /Explicit is the main ABAQUS solver used, the ABAQUS high performance computing environment is optimized for /Explicit jobs.

There have been many IT projects that have made it possible to build unified global UNIX/Linux based high performing ABAQUS computing environment. The following chapters present the most important details of the ABAQUS high performance computing environment.

2.1 ABAQUS documentation

There is a common ABAQUS documentation server available for ABAQUS users within the company, see Figure 3. This server provides documentation of the latest ABAQUS versions. Users have possibility to install the documentation locally on their laptops or workstations also.

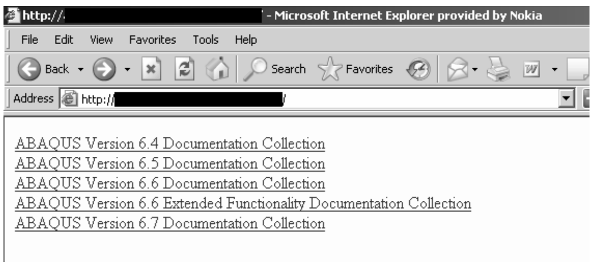

Figure 3. The front page of internal ABAQUS documentation web-server.

2.2 ABAQUS licensing

There is a common wide area network (WAN) ABAQUS license used to serve all geographical research and development (R&D) sites using ABAQUS. R&D sites are located in Europe, North America and Asia.

Internal licensing support team provides tools for license monitoring, license reporting and license maintenance. License monitoring shows the current state of licenses, license reporting shows the long time usage statistics and license maintenance includes for example requests for license file updates.

Figure 4 shows an example of long term usage reporting. These kinds of reports can be used in decision making if there is a need to modify the number of purchased ABAQUS licenses.

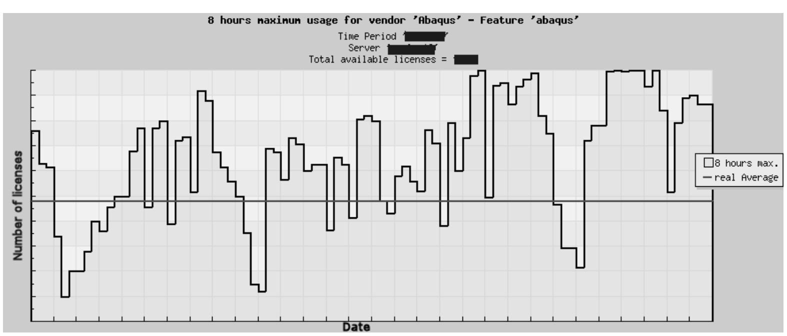

Figure 4. Long term ABAQUS license usage.

2.3 UNIX user environment

Every R&D site have they own user account management system. This means that there are separate user accounts and home directories in every site. This kind of setup requires a lot of co-ordination between sites to get unified UNIX user environment working.

The current UNIX user environment enables unified user environment between different R&D sites. One of the biggest benefits of this kind of unified user environment is that now only few steps are needed to get exactly the same kind of user environment at different R&D sites. There is a web-tool available in order to maintain default shell, own groups and passwords at different R&D sites.

Passwordless ssh configuration is needed for all users to get ABAQUS working in ABAQUS computing environment. It is very important that the passwordless ssh works without user interaction even if there is added new servers in the cluster or if the identity of the servers is changed due to operating system or hardware maintenance. Compared to rsh protocol, ssh provides much more convenient ways to fulfill above mentioned requirements.

The most typical way to connect to UNIX/Linux computing environment is to use Citrix Program Neighborhood application (http://www.citrix.com) from Windows XP laptop or workstation. Citrix provides secure, load balanced connection with session continuity to computing environment. The basic idea of Citrix is that the user can open a Linux desktop (typically KDE) window on his or her Windows computer. The drawback of Citrix is that it cannot handle OpenGL based applications like ABAQUS/Viewer.

All ABAQUS users at the company have a possibility to connect to ABAQUS computing environment using Virtual Private Network (VPN) with laptop or mobile phone wherever Internet connection is available.

2.4 ABAQUS installation

For UNIX and Linux applications there is in use the system where all widely used applications are installed into one place only. These application installations are replicated using rsync command (http://rsync.samba.org) on a daily basis to every site where the application is needed. ABAQUS installation is one of those replicated applications. This kind of replication reduces administration work and guarantees that every user has similar ABAQUS installation. Also all site level abaqus_v6.env settings are identical at every R&D site.

The quality of new ABAQUS versions is checked by running in house made models to make sure that the results between different versions are consistent. This kind of in-house quality assurance has proven to be very important.

2.5 Application settings

When the user’s UNIX environment has been properly set-up and there are applications available, user takes applications into use by using module commands. Module commands are based on open source software package called Modules (http://modules.sourceforge.net).

By using modules it is easy to handle different versions of applications. There is own module for every version of applications. In addition to version control, modules can be used to set environment variables for applications. For example ABAQUS modules used set the PATH variable to point to the correct ABAQUS binaries. Other environment variable examples set by ABAQUS modules are the MPI library and compiler related environment variables. In module files it is also possible to define command aliases for end users. There are aliases for example for checking the status of clusters, licensing information and so on.

The usage of module commands is very simple. There are commands available to check available application version, load desired application and to unload desired application. For example, the Figure 5 lists all available ABAQUS versions.

```
$ module avail -a abaqus
------------------------ modulefiles --------------------------
abaqus/abq67pr4          abaqus/abq663            abaqus/abq634
abaqus/abq67pr3          abaqus/abq656            abaqus/abaqus
abaqus/abq66ef1(default) abaqus/abq644
```

Figure 5. Available ABAQUS modules in the computing environment.

There is one global mechanical simulation module available in the computing environment. In addition to ABAQUS this module contains for example pre- and postprocessing tools and batch queuing application. The most of the users load this module by default during login scripts of UNIX shell, so there is no need to load any module separately. All of the modules are replicated just like ABAQUS installation itself, so it is possible to have exactly the same applications settings for ABAQUS in every R&D site.

2.6 Batch queue system

Sun Grid Engine (http://gridengine.sunsource.net/) has been selected as primary batch queuing system. From ABAQUS application point of view the decision was not ideal. Very likely for example the LSF and PBS Pro could be much more convenient with ABAQUS. However, ABAQUS is relatively small application at the company and there are much bigger user groups in software development (SW) and Electronic Design Automation (EDA) teams and Sun Grid Engine seems to be suitable for those teams. One benefit of SGE is that it is very inexpensive compared to LSF and PBS.

Sun Grid Engine (SGE) installations are separate at every site. However the basic settings are identical like, scheduler, queue and parallel environment definitions. For ABAQUS usage, there are defined queues and parallel environments into SGE which basically are identical at every site. Only the host group definitions containing actual hardware are different at every R&D site. In addition to queue and parallel environment definitions there are defined many kind of computing resources in to SGE installation. This kind of resources are for example free memory of the hosts and license resources for short and long ABAQUS jobs.

The possibility to use wildcards (*) in the names of parallel environments, enables SGE customization at different R&D sites. In the most of the SGE installations, there are defined own parallel environments for distributed memory parallel (DMP) and shared memory parallel (SMP) hardware. In DMP parallel environment one job can be distributed over multiple hosts. In SMP parallel environments, one job stays inside one host.

Even though it is not mentioned in the ABAQUS documentation, ABAQUS recognizes the $PE_HOSTFILE environment variable of SGE. This environment variable defines the hosts where the job will be run i.e. mp_host_list variable of abaqus_v6.env. This makes the usage of ABAQUS together with SGE much easier.

There are build-in SGE commands and in-house scripts available to check the status of parallel environments and running ABAQUS jobs. Script called qstatp shows the status of parallel environments in ABAQUS cluster and script called qstatm shows the information about running and pending jobs, see Figure 6 and Figure 7 for details.

```
$ qstatp
PARALLEL ENVIRONMENT    USED   AVAIL UNAVAIL   TOTAL PENDING
------------------------------------------------------------
mda_dmp                  xxx     xxx       0     xxx      40
mda_short                yyy     yyy       0     yyy       0
mda_smp                  zzz     zzz       0     zzz       0
Available ABAQUS licenses for long jobs  : xxx
Available ABAQUS licenses for short jobs : yyy
```

Figure 6. Status of parallel environments in ABAQUS cluster.

```
$ qstatm
job-ID  prior   name       user         state submit/start at     queue            slots
-----------------------------------------------------------------------------------------
 194333 0.33667 DTS_rules  user_xy      r     01/30/2007 13:40:59 mda.q@host111      24
 194363 0.31549 DTS_isbest user_zz      r     01/30/2007 14:08:07 mda.q@host222      16
 194370 0.35586 disp_test  user_xy      r     01/30/2007 14:15:03 mda.q@host333      32
 194228 0.31567 DTS_rulez  user_yy      r     01/30/2007 11:31:00 mda.q@host444      16
 194372 0.31548 DTS_iscool user_zy      r     01/30/2007 14:18:34 mda.q@host555      16
 194382 0.33662 batspring2 user_yy      qw    01/30/2007 14:24:36                    24
 194384 0.29239 pwb_test   user_xy      qw    01/30/2007 14:27:27                     8
 194385 0.00000 Going_AUC  user_xy      qw    01/30/2007 14:27:58                     8
```

Figure 7. Information about running and pending jobs in ABAQUS cluster.

ABAQUS job is submitted by using ABAQUS queues defined in abaqus_v6.env file. More about abaqus_v6.env settings can be found in chapter 2.7 ABAQUS settings. Typical way to start an ABAQUS job is presented in Figure 8.

```
$ abaqus -j my_job -cpus 816 -que

See $HOME/my_job.log to see detailed information about simulation.
For example, use command:
tail -f -n 100 $HOME/my_job.log

Log-file is moved to current directory after the job completion.

Your job 210935 ("my_job") has been submitted.
```

Figure 8. Typical way to start ABAQUS job at the computing environment.

The previous command submits the job to default ABAQUS queue and processor range option is used, which means that SGE tries to allocate for a job as many processors as possible between 8 and 16. The processor range option is not directly supported by ABAQUS driver, but workaround for that has been created. The processor range option enables more efficient processor usage of the clusters. When this kind of command is used, ABAQUS queue driver will generate the SGE submit script shown in Figure 9.

```
$ cat my_job.sge
#!/bin/bash -l

#This script is used to submit ABAQUS simulation jobs to Sun Grid Engine.
#This script is generated by /path_to_the_abq_files/SGEQueue.env file.

# Use bash as a interpreting shell for the job.
#$ -S /bin/bash

# Export all environment variables.
#$ -V

# Enable notifying time for delete and suspend signals.
#$ -notify

# Give the name for the job.
#$ -N my_job

# Send email at the beginning and the end of the job.
#$ -m be

# Write stdout and stderr to log file.
#$ -o $HOME/my_job.log
#$ -e $HOME/my_job.log

# Request ABAQUS licenses for the job.
#$ -l lic_abaqus=1.5

# Request memory for the job.
#$ -l mf=0.5G

# Request queue for a job.
#$ -q mda.q

# Request parallel environment and slots for the job.
#$ -pe mda_dmp* 8-16

# Source $HOME/.profile to get crossplatform submission to work.
. $HOME/.profile

# Make sure that correct ABAQUS version and binaries are used.
module unload abaqus
module load abaqus/abaqus

# Unset display variable to get rid of ssh authentication error messages.
unset DISPLAY

# Unset ABA_HOME variable to get to get crossplatform submission to work.
unset ABA_HOME

# Run the ABAQUS job.
abaqus python /path_to_the_abq_files/abaJobHandler.py my_job \
         /path_to_my_job/my_job.fs 0 my_job.inp my_job.com

# Transfer log file from $HOME to submit directory and remove .log file from $HOME.
if [ -f $HOME/my_job.log ] ; then scp -q $HOME/my_job.log ${SGE_O_HOST}:${SGE_O_WORKDIR} ; fi
if [ -f $HOME/my_job.log ] ; then rm -f $HOME/my_job.log ; fi
```

Figure 9. Example of Sun Grid Engine submission script of ABAQUS job.

In the previous example, all the lines beginning with #$ marks are options for job submission command of SGE called qsub. The script is self-explanatory regarding the options used. The content of .sge file depends on the name of ABAQUS queue used. The script called abaJobHandler.py is used to start the actual job. This script is needed in order to be able to use local disks of compute nodes. More information about ABAQUS queues and abaJobHandler.py script can be found in chapter 2.7 ABAQUS settings

Finally, ABAQUS queue drivers submits the job using command "qsub my_job.sge". The .log file is written to user's home directory during the simulation. This is useful in the cases where local disks of compute hosts are used, because in .log file there is mentioned the host name and the directory which is used as working directory during the simulation. After the job completion all files are transferred back to job submission directory.

Figure 10 shows the long time reporting of processor usage of one ABAQUS computing cluster at one of the main R&D sites. Similar kinds of reports are available for every ABAQUS computing cluster. These kinds of reports can be used in decision making if there is need for changes in ABAQUS computing resources.

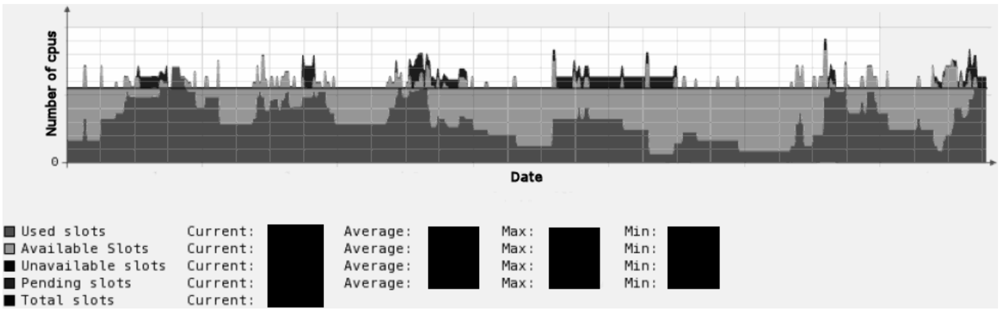

Figure 10. Long time reporting of ABAQUS computing cluster.

2.7 ABAQUS settings

As mentioned previously, all the R&D sites use the same ABAQUS installation and thus the same site level abaqus_v6.env file is in use at every site as well. Thanks to unified computing environment. this kind of arrangement has been working very well. The site level abaqus_v6.env file defines for example ABAQUS queues and on JobCompletion functions for every R&D site.

In the site level abaqus_v6.env file of the ABAQUS the Python class called SGEQueue is derived from default Queue class provided by ABAQUS. SGEQueue class re-writes .com file, writes .sge file and submits the job to SGE. Re-write of .com file is needed to get the processor range option of abaqus command to work, see chapter 2.6 Batch queue system for more details.

ABAQUS queues are divided to dmp (distributed memory parallel) and smp (shared memory parallel) type of queues. If dmp type of queue is used, the job is submitted to cluster hardware. If smp queue is selected, the job is submitted to single machine. There are some additional queue options available. Additional options are: odb, opti, local and hold. Users can combine any of these additional options to dmp or smp queue names, for example “-que dmp_odb” or “-que smp_odb_local”. There is available queue for short jobs also. Figure 11 lists all the ABAQUS queues defined in the site level abaqus_v6.env.

```
$ abaqus -info env
ABAQUS JOB abaqus
ABAQUS Version 6.6-EF1
....
# User defined batch queues
# -------------------------
default=Submits the job to the distributed memory cluster (default).
dmp=Submits the job to the distributed memory cluster (default).
hold=Writes all files required to submit the job, but will not submit the job.
local=Local disks of the compute hosts will be used.
odb=After the job completion, the odb file will be transferred to the user specified location.
opti=Queue for the optimization jobs.
short=Submits the job to the short queue. Time limit is 15 minutes.
smp=Submits the job to the shared memory machines.
```

Figure 11. Available ABAQUS queues.

As mentioned in chapter 2.6 Batch queue system, the script called abaJobHandler.py is used to actually submit the ABAQUS job. This script enables local disk usage. The usage of local disks may be useful in the case of ABAQUS/Standard jobs. By default network file system (NFS) is used.

The onJobCompletion function of abaqus_v6.env file defines the automatic odb file transfer from working directory to user-specified computer and directory. User has to define host name and the directory by using UNIX environment variables. Transfer is done by using either scp or ftp commands. The transfer is started immediately after the job has been completed and none of the computing resources are reserved anymore during the file transfer. This kind of arrangement is very important in the case when the odb file is transferred over the slow network connections. Usually jobs are completed and files are transferred during the night so that odb files are ready for post-processing at the morning.

In the site level abaqus_v6.env file, the double precision is in use by default and HP-MPI delivered with ABAQUS is used as MPI library. The '-prot' option of mp_mpirun_options is useful for checking that correct interconnect is used during simulation. Performance degradation is significant, even if only one host in Infiniband cluster is using Gigabit Ethernet instead of Infiniband.

HP-MPI is tight integrated with SGE. Tight integration is made by using 'qrsh –inherit' command of SGE. However, due to the incompatibility of qrsh and ssh command, wrapper script called qsshm has to be used to get tight integration to work. HP-MPI uses 'qrsh –inherit' as remote command as well.

3. Hardware

The target of the environment settings described above is actually to hide hardware from the end-users. Hardware belongs to different host groups, queues and parallel environments of Sun Grid Engine. This way the users do not have to know anything about underlying hardware. Hardware is selected based on the ABAQUS queue names so it is only required to know how to use ABAQUS queues.

The performance of different hardware platforms and technologies are closely followed to keep the computing environment up-to-date. New interesting hardware is benchmarked by running in-house made models on a regular basis. ABAQUS web-site also provides good hardware benchmark information. The following chapters give an overview of the hardware used for ABAQUS simulations at the company.

3.1 Past

The history of the finite element simulations at the company is not very long. The first finite element simulations were run in the middle of 1990s and the first drop test simulations late 1990s. At that time, the entry level servers and workstations were used for simulations.

A big improvement in the hardware side happened at the end of year 2002 when the IBM p650 8-way SMP servers were implemented at one of the main R&D sites. Quite soon the first HP-UX

Itanium version of ABAQUS was released and small, 2- and 4-way, HP-UX Itanium servers were purchased to many R&D sites.

When the IBM p650 servers were implemented, basically most of the ABAQUS users at the company started to use the same servers. Thus the need for job management system arose. At that time there was Sun Grid Engine batch queue system up and running for EDA and SW user groups and it was decided that ABAQUS is run through SGE as well.

The evolving performance/price ratio of the Linux cluster was noticed during the purchase process of IBM p650 and HP-UX Itanium servers during the years 2002 and 2003. The problem was that there was not any Linux version of ABAQUS available, neither DMP version of ABAQUS.

The first cluster tests with ABAQUS were made in the beginning of year 2004 with version 6.4. At that time, the DMP version of ABAQUS was an undocumented feature, but thanks to high-quality ABAQUS support, the DMP version of ABAQUS started to work. At first there was four 1-way workstations connected together with 100 Mbps Ethernet and later IBM BladeCenter with Gigabit Ethernet was tested. The LAM/MPI (http://www.lam-mpi.org/) was used as MPI library. It was very challenging to get LAM/MPI lamboot universe working in UNIX environment. It was quickly noticed that there is a need for unified UNIX user environment if this kind of cluster solution is selected for a production use. However, at that time the scalability of ABAQUS/Explicit was so poor that it was decided to stay in SMP servers.

During the winter 2004/2005 lot of benchmarking was made with different hardware vendors to compare what is the best platform for ABAQUS/Explicit purposes. The pre-releases of ABAQUS/Explicit v6.5 were used during benchmarking phase. Intel Xeon, Intel Itanium, AMD Opteron and IBM Power5 processors were benchmarked. Both shared and distributed memory hardware was tested. It was quickly noticed that due to the improvements in the /Explicit itself, in the processors and interconnects, the possibility of using cluster hardware became very interesting. With the benchmarked servers it was possible to shorten the simulation time to one tenth compared to servers that were used at that time in production use, see Figure 12.

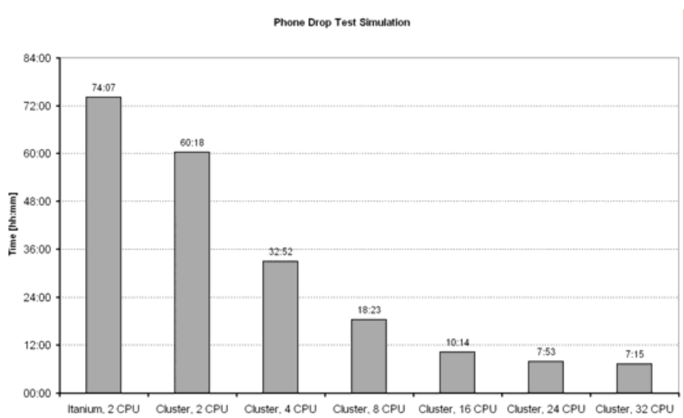

Figure 12. Benchmark results.

Based on the benchmark results, the Intel Xeon based blade servers with 64-bit Linux operating system and Infiniband interconnect were selected and the first cluster hardware for production purposes was purchased during spring 2005.

Thanks to high quality support from hardware vendors, ABAQUS and HP-MPI teams, the set-up of this kind of cluster hardware for ABAQUS/Explicit went relatively smoothly. Some fine tuning of abaqus_v6.env file, Sun Grid Engine, MPI libraries and hardware itself was needed to get ABAQUS/Explicit simulations running on that cluster.

3.2 Present

Currently clusters are the main hardware for ABAQUS simulations. The most of the clusters are based on the blade technology. The clusters are using either Intel Xeon EM64T or AMD Opteron processors. Clusters are based on 2- or 4-core nodes. The Gigabit Ethernet or Infiniband are used as interconnects and there is 1-4 GB memory available per core. Operating system is 64-bit Red Hat Enterprise Linux.

The cluster hardware is used in R&D sites in Europe, North America and Asia. The sizes of ABAQUS clusters vary from tens to more than one hundred processors and in total there are hundreds of modern processors for ABAQUS jobs. Currently there is only couple of big memory 2-4-way HP-UX Itanium servers mainly for /Standard jobs. However, many of ABAQUS/Standard job are ran in cluster hardware also.

In ABAQUS computing environment, all ABAQUS jobs are using shared network file system (NFS) by default as working directory. The scratch directories are local. However it is possible to request local file system for a job as a working directory. This may be useful in the case of ABAQUS/Standard jobs.

In addition to the compute nodes, there are some other nodes as well in the ABAQUS computing environment. There are front-end (login) nodes where the users connect when they want to submit their jobs to clusters. Couples of nodes are reserved for short time ABAQUS jobs, for example datacheck jobs. It is useful to have both hardware and licenses reserved for datacheck jobs to make sure that the users are able to debug models in the case of lack of licenses.

The Infiniband interconnect has proven to be very efficient for the drop test models. There were quite big problems in the beginning to get the Infiniband interconnect reliable. But it seems that the quality of Infiniband drivers has improved a lot and nowadays there are very few Infiniband related problems.

Currently HP-MPI is used as MPI library. HP-MPI is integrated very nicely with ABAQUS and it does not require anything special from the user environment. Very useful feature of HP-MPI is that it is able to select the best available interconnect for ABAQUS jobs. This is very convenient feature when the same ABAQUS installation is used in dissimilar clusters. The Open MPI library (http://www.open-mpi.org) is another interesting MPI library and the development of the Open MPI library is closely followed.

3.3 Future

The server performance development is closely followed and clusters are possibly upgraded when improvements are available and verified by benchmarks. Based on the latest benchmarks the Intel Xeon 5160 (Woodcrest) based blade servers seem to be very interesting option for /Explicit simulation purposes. Hopefully during the upgrade of clusters, it is possible to harmonize the computing hardware also.

It would be preferable that in future, ABAQUS could improve the scalability of clusters to growing number of processors used for a one job. There are many interesting new hardware options coming to high performance computing market. This kind of hardware options are for example accelerator cards for floating point calculations and Cell type of processor architectures (http://www.research.ibm.com/cell/). It is requested that ABAQUS respond rapidly to the new hardware options that are on the horizon.

4. Post-processing issues

The main tool for post-processing is the ABAQUS/Viewer, but there are other post-processing tools used as well. Few workstations with the remote graphics applications located near the computing servers have been implemented to minimize the file transfers from the servers to the local workstations. As mentioned in the chapter 2.7 ABAQUS settings, the automated ODB file transfer system from servers to workstations is available to speed-up file transfers.

The shorter simulation times due to the high performing hardware and software have brought new challenges for post-processing. The Technology Platforms unit provides the post-processing scripts for platform components to enable the automated results extraction and reporting from ODB. These post-processing scripts make the post-processing remarkable easier and faster.

The platform component related results extracted by post-processing scripts are stored to the result database of Technology Platforms. By using the result database, it is possible to compare how the same platform component behaves in the different products and there is also possibility to see the correlation between the simulation and testing results.

5. Thoughts for future development

There is still lot of fine tuning to do with Sun Grid Engine set-up. One of the most important plans is to modify the Sun Grid Engine setup so that in every cluster there are master hosts, which are running pre, package and merging processes of ABAQUS job, but not actual standard or explicit processes. Standard and explicit processes would be run on dedicated compute nodes. This way, the memory requirements of compute hosts could be minimized.

The reliability of clusters seems to be far behind the SMP servers. Better diagnostics tools are required to disable malfunctioning compute nodes from Sun Grid Engine to guarantee reliable computing environment.

Installation of ABAQUS software package is relatively simple task, but there are quite many installations happening during the year. When multiple platforms, maintenance and pre-releases are taken into account there may be tens of ABAQUS installations during the year. That is why it would be good if ABAQUS could simplify the installation procedure, so that multiple platforms could be installed at the same time. Hopefully forthcoming DVD installation media will ease the installation of multiple platforms.

In addition to enhanced installation procedure, it would be preferable to have some kind of common abaqus_v6.env file for every supported ABAQUS platform. This way it would be enough to change abaqus_v6.env setting only in one file and the change would have an effect to all installed platforms.

As the mobile devices are becoming mechanically more complex due to many kinds of mechanisms, there will more drop configurations to simulate. In addition to additional configurations there is clear need to simulate more drop orientations. To be able to simulate all these new configurations and orientations, there will be clear need for more computing power. Also more detailed drop models will require more computing power.

Deployment of 3DXML format (http://www.3ds.com/3dxml/) will enrich the post-processing and reporting capabilities. When 3DXML format will includes essential features for finite element post-processing and reporting, it will allow for mechanical designers an excellent insight into the mechanical behavior of the mobile device under development.

To shorten the lead time of drop test simulations there is a need to better manage and automate the drop test simulation process. The automation covers wide range of issues, for example automated clean-up of geometry, batch meshing and automated post-processing scripts. These kinds of simulation data management activities may have an effect on the existing ABAQUS computing environment also.

6. Summary

The building of high performance computing environment for ABAQUS in a global enterprise organization is a challenging task. Many aspects must be taken into account to be able to build reliable production environment. Utilization of processes described in this paper reduces significantly the administration work related to ABAQUS software.

It has been learned that computing systems based open standards (multiple vendors) is not a silver bullet to high performance computing. In the case of problems, there are easily too many players in the field and it is very difficult to say who is responsible to fix the problems.

The benefit in big company is that it is always easy to find highly talented people with required knowledge to specify and set-up certain parts of computing environment. For example, operating system installations and network set-up can be very complicated if unusual hardware is used. Traditionally, the company has provided state of the art tool for employees. That is true in the area of finite element simulation also.