University of Huddersfield Repository

Kureshi, Ibad

Establishing a University Grid for HPC Applications

Original Citation

Kureshi, Ibad (2010) Establishing a University Grid for HPC Applications. Masters thesis, University of Huddersfield.

This version is available at http://eprints.hud.ac.uk/10169/

http://eprints.hud.ac.uk/

Establishing a University Grid for HPC Applications

Ibad Kureshi

A thesis submitted to the University of Huddersfield
in partial fulfillment of the requirements for
the degree of MSc by Research

September 2010

Abstract

This thesis documents a project undertaken at the University of Huddersfield between October 2009 and August 2010 to setup a High Performance Computing (HPC) resource, which would serve the University's research community by providing a robust computing solution. This thesis will look at all the various kinds of requirements different fields have, with regard to a computing solution, and the tools available to meet these specific needs. This report serves as a manual for any small to medium sized institution that considers setting up a local HPC resource. It looks at all considerations regarding hardware, software, licensing, infrastructure, HR etc for setting up a centralised computing resource with sustainability and robustness being the central aim of the proposed resource. The possibilities of cross-continent and cross-institution collaboration using Clusters and Grid technologies are explored and the method for connecting to the UK eScience community through the NGS is explained.

Acknowledgements

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the "Copyright") and s/he has given The University of Huddersfield the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Contents

Word Count: 29,870

List of Figures

List of Abbreviations

3DsMax	Autodesk 3D Studio Max
ADA	School of Art Design and Architecture
CAE	Computer Aided Engineering
CCLRC	Council for the Central Laboratory of the Research Councils
CENTOS	Community Enterprise Operating System
CFD	Computational Fluid Dynamics
CLS	Computing and Library Services
COTS	Commodity off the Shelf
DL_POLY	Daresbury Laboratory POLY
DOCABS	Department of Chemistry and Biological Sciences
E.T.	Department of Engineering and Technology
EPSRC	Engineering and Physical Science Research Council
FFMD	Force Field Molecular Dynamics
GAMESS-UK	General Atomic & Molecular Electronic Structure System – United Kingdom edition
Gulp	General Utility Lattice Program
HPC	High Performance Computing
HPC-RC	High Performance Computing Resource Centre
HPC-RG	High Performance Computing Research Group
JANET	Joint Academic Network
JISC	Joint Information Systems Committee
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
MATLAB	Matrix Laboratories
Metadise	Minimum Energy Techniques Applied to Defects Interfaces and Surface Energies
MPI	Message Passing Interface
NGS	National Grid Service
NWChem	North Western Chemistry
OSC	Oxford Supercomputing Centre
OSCAR	Open Source Cluster Application Resource
PDE	Partial Differential Equations
QGG	Queensgate Grid
RCUK	Research Councils of United Kingdom
RSH	Remote Shell
SAS	School of Applied Sciences
SCE	School of Computing and Engineering
SGE	Sun Grid Engine
SGEEE	Sun Grid Engine Enterprise Edition
SSH	Secure Shell
STFC	Science and Technologies Facilitation Council
VDT	Virtual Data Toolkit
VO	Virtual Organisation

Section I: Background and Problem Outline

Chapter 1: Introduction

Chapter 1.1: Problem Definition and University Requirements

The University of Huddersfield is classed as a small to medium Higher/Further Educational Institution with a modest research community. The University has a research rating of 52.4% in the Sunday Times University guide (Sunday Times 2010). Since 2008 the University has shifted its focus from being a teaching institution that conducts some research to becoming one of the world's leading research institutions. At a school-level, the university has already begun to achieve four-star ratings (world leading quality in terms of originality, significance and rigour)(RAE Results, 2008).

In the School of Applied Science (SAS) research in molecular dynamics related to (a) biological interference in the creation of crystals leading to improved creation of synthetic materials and (b) the efficiency of new dopants leading to a new generation of nuclear fuel has already pushed the University's computing facilities to the limits of its capabilities. This sort of research is not meant for ordinary desktop computers. Similarly research in the School of Computing and Engineering (SCE) related to (a) Image processing in Security Applications leading to near real time detection of anti-social behaviour; (b) Fluid Dynamic simulations leading to more fuel efficient vehicles and (c) research in fuel materials leading to better fuel types and efficiencies has also pushed the limits of the available computing power. The School of Art Design and Architecture as well as the Department of Informatics encourage their students to pick up commercial work in animation and graphic design so that they may gain professional experience while studying and render farm facilities are absolutely essential for such work.
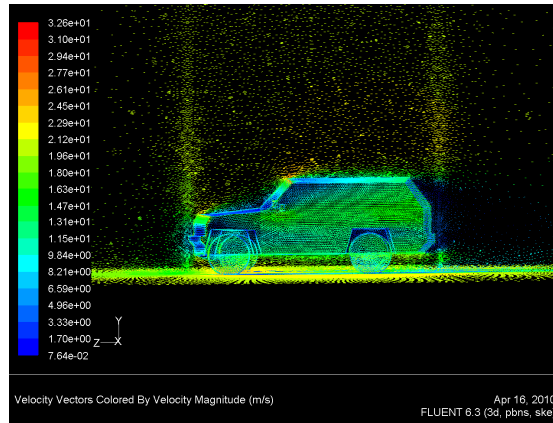
Figure 1: Air Flow Velocity Vector Diagram after Simulation on QGG System (Palmer 2010)

Before investing in such a system or finding an alternate way to provide HPC to the research community, an analysis of each school's requirement is essential. At first glance, it is obvious that the requirements are diverse. The most glaring of differences is the platform for applications. The Windows© versus *NIX divide is evident here: most Art and Design software runs only on Windows© systems, while most of Applied Sciences applications run Linux based codes. The School of Computing and Engineering complicates matters further by using/requiring both platforms equally.

As much of the research being undertaken by the SAS is based on a *NIX platform and incorporates either free/open source packages or has community developed codes that can be compiled on different machines, many SAS researchers have started outsourcing their computing to HPC systems at other universities/research institutions. Students and researchers from SCE and ADA cannot outsource their computing requirements without incurring large costs. Most applications that are being used in Huddersfield are either locked by who can use them/how they can be used or are locked to location by post code. For research in these schools to increase, access to a local cluster/high performance-computing system is essential. Licensing is further complicated as every instructor/research has his own license pool. There is no Department level software pool, let alone a School/University wide pool.

Another layer of complexity is added by the requirements of the Applications. Many programs require large amounts of RAM to be able to

14

complete their tasks while others just need many processors to divide the work up as much as possible. There are application that require many processors but the processors need to cross communicate throughout the simulations (so much so that even gigabit interconnects can possibly lock up with too much traffic). Some applications need to write in excess of 300GB of temporary data to storage devices while they are processing and there are those that don't need to write large chunks of data but still generate so many small chunks of data that writing them to a central storage server would cripple the networking backbone of any system.

From the outset it is obvious that to make a centralised High Performance Computing Resource the interests of all schools will have to be implemented in some shape or form and the resulting system will not only have to be robust and reliable but also diverse and dynamic as it needs to handle the various requirements.

Chapter 1.2: Background to HPC in Education, Research, eScience

High Performance Computing has always been a requirement of research. Since the 1950s military and academic research institutions have heavily invested in large computational facilities to meet the demands of the scientific community. The same principles of large centralised data centres with users getting a time share on highly optimised pieces of hardware continue from the early days with large mainframe computers being replaced by modern day "super computers" and clusters.

High Performance Computing is required in scientific research. Real world modelling and monitoring for simulation and study purposes are very complex tasks that either take very long time to complete, need to record large amounts of data very quickly, storage large amounts of data, perform repetitive calculations or need large amounts of RAM and these factors make personal computers (PCs) ineffective. This is not to imply that there is no place for PCs: users will continue to compile their data and create code/programs on their own machines, as well as assess output. What the HPC system will do is provide a

central computing facility using standardised communications protocols for the actual processing of data/designs. (Sterling et al. 1999)

The Research Councils of UK (RCUK) have funded many projects to improve the country's infrastructure for eScience. In the 1950's the Council for the Central Laboratory of the Research Councils (CCLRC) now known as the Science and Technologies Facilitation Council (STFC) set up one of the first HPC systems in the United Kingdom at the ATLAS computer laboratories. The primary function of these Ferranti and IBM machines was to provide a platform for the Atomic Weapons Establishment. Around the same time Universities like Reading, Oxford, Cambridge, London and Manchester were also setting up their computer facilities. Since then these large institutions, which have a rich tradition of research, have pioneered in the field of HPC and have provided the UK eScience community with access to advance computing technologies.

It was the CCLRC that introduced the Joint Academic Network (JANET) and later upgraded it to super JANET to provide good links between academic institutions in the United Kingdom. This has led to a culture of collaboration between institutions and allowed for smaller institutions to establish good links at a fraction of the cost of commercial Internet links.

To meet the needs of modern day research in 2002 the STFC in collaboration with the University of Edinburgh commissioned an HPC system, known as HPCx, hosted at Daresbury Laboratories. This system remained the UK's primary supercomputing facility for academic research up to 2007. HPCx provided the infrastructure for research in four major fields; Materials and Condensed Matter; Atomic and Molecular Physics; Computational Engineering and Environmental Modelling. Within Atomic and Molecular physics research many tools like GAMESS-UK and DL_POLY were developed which now are intrinsic tools for simulations in Applied Chemistry and Material Physics. World leading work was also done in Computational Fluid Dynamics and many open source tools were developed for the academic community. The two fields mentioned above are also important areas of research within the University of Huddersfield.

Continuing the spirit of collaboration, that saw the birth of JANET, the STFC along with many Universities connected to form the NGS, which allowed member institutions and all academic researchers in the eScience community to connect and share resources, which included the HPCx system (STFC 2007).

In 2007 a newer HPC system was commissioned through JISC funding and this system has taken over the mantle as the most powerful HPC system for academic research in the United Kingdom. HECToR, the new system hosted by the University of Edinburgh is continuing to provide a very important service to the eScience community and is leading to the publication of world leading research (HECToR 2009). The HPCx project is over as of 1st January 2010 but the results gained by research carried out on this HPC system is still being compiled and published (HPCx 2010).

Chapter 1.3: Overview of the NGS

The NGS (formerly known as the National Grid Service) "aims to enable coherent electronic access for UK researchers to all computational and data based resources and facilities required to carry out their research, independent of resource or researcher location." Funded by the EPSRC and JISC the NGS is the collaborative tool of the UK eScience program, which connects researchers from 27-partner/affiliate sites and over 60 Universities, including the University of Huddersfield. The NGS provides an e-Infrastructure to support the computing and data needs of UK researchers.

The NGS was formed out of the Engineering Task Force Production Grid and eventually merged with the Grid Operations Support Centre, which was a UK eScience funded program to provide support for grid users. In essence, the NGS provides free access to large super computers and clusters along with application support and storage services to UK academic researchers through the use of portals and other Shell / Terminal tools. The NGS itself does not own any machines but through collaborating Partner Sites provides users with access to dedicated sites. The backbone of the NGS is based on the existing Joint Academic Network (JANET), which is a UK Government Funded computer network for the use of research and academic institutions. JANET provides a link

17

between UK Universities and the world. The Partner Sites in the NGS provide the hardware and site level support for researchers. Universities like Oxford, Edinburgh and Leeds among others all provide unrestricted access[1] to their clusters. Other academic research organisations like the various sites of the Science and Technologies Facilitation Council (STFC) also provide access to their research machines. Affiliate Sites at the NGS give users access to their resources with some restrictions.

Through the NGS researchers can not only get access to HPC systems but can also use resources to collaborate with researchers in other Universities. The NGS has its own software stack called the Virtual Data Toolkit (VDT) that is based on the Globus Grid Middleware. It uses X.509 certificates to recognise users and resources as members of the NGS virtual organisation (VO). These X.509 certificates are issued to users through the UK eScience council, as it is the recognised Certificate Authority for the United Kingdom. Through the NGS and the eScience council UK based researchers can also connect to EGEE (the European grid) and Teragrid (the US research grid).

Using the JANET backbone the NGS and the University of Manchester have developed a strong Access Grid (AG) backbone in the UK for researchers. Access Grid is the Argonne National Laboratory (US based) tool for Video Conferencing and Net Meetings. A backbone for collaborative work in the United States this system is becoming an important tool for research in the UK (NGS 2009).

Chapter 1.4: Aims of the Project

This project aims to create a robust and sustainable High Performance Computing Resource to serve as research tool at the University of Huddersfield. This tool is hoped to open more doors for research in the University; remove the factor of "computational power/computational time" from any decision making processes in research; and to allow for collaboration not just between researchers with the Department of Engineering (sponsoring this research) but

---

[1] Application/Licensing restricting as well as queue restrictions continue to apply but the clusters job queues have identical priorities for all users NGS or local.

across all disciplines within the University of Huddersfield and other Higher Educational Institutions in the United Kingdom.

To achieve these goals this project will identify the needs of various departments within the University to ascertain the demand and the requirements for a HPC system. The research will investigate the current deployments of Grid and Cluster Technologies within the research and academic institutions in the UK and internationally. After evaluating the current solutions available a Cross-University Grid solution will be designed and implemented. This system should be a combination of specialised clusters and general purpose clusters that cater to the needs of the research community as a whole.

Special provisions will be made to lead the University down the path of Partner status on the NGS so that research and collaboration can be maximised. Connecting with smaller regional colleges would also be beneficial so that the level of education in the community is improved. The proposed Huddersfield Grid should also take into account opportunities for enterprise work and form a benchmark for business/consortium environments.

Chapter 2: Literature Review and Published Experiences

In the following chapter four implementations of grids are analysed and based on the understanding of these grids a local implementation for the University of Huddersfield is designed. The grids differ in their deployment. The First grid to be analysed is a high-throughput cycle stealing grid based on the Condor middleware, implemented at the University of Cambridge is. The second grid is a grid of clusters implemented at the University of Oxford. The third grid is a geographically distributed grid known as the White Rose Grid. Then finally the fourth is the global grid that performs calculations for the Large Hadron Collider. These grids are also looked at as a benchmark because aside from the diverse implementation these institutions are world leading research institutions.

Chapter 2.1: Overview and Analysis of CAMGRID

Dr M Callega et al describe their experiences on establishing a campus grid by deploying Condor in pools to group resources around the University into local HPC systems and then using Condor to group these local pools to form a campus wide grid. The HPC resources at Cambridge are formed by the merger of machines owned by the departments and the centrally administered University Computing Services. The main considerations and technical hitches faced by the CamGrid initiative were stakeholder concerns regarding security policies and due to the federated nature of the various colleges, schools and departments at the university the issues of firewalls and private IP networks residing behind them.

Because of the many different departments contributing their lab machines for this grid the CamGrid is truly heterogeneous, with the three major platforms (Windows©, Linux, and MAC) all represented. X509 based authentication across the grid is not possible as the Condor middleware has no mechanism of cross platform authentication.

The University Computing Services pools of machines, dubbed the Personal Workstation Facility, are desktops that have installations of Windows© XP and SUSE Linux 9.0. These desktops with the use of a controller can reboot

from XP to Linux when idle for long periods and during out-of-hour time periods. Some machines will remain in Windows© XP to provide a Windows© execution environment controlled by Condor. These pools of resources are defined in the "vanilla universe" of Condor, will be available for all users and will use Kerberos for authentication. One terabyte of storage for temporary files is also provisioned for the user's jobs.

There is also a "standard universe" defined in Condor that includes all the federated pools of resources owned by the different Schools and Departments. This universe will not be supported (in the sense of user support) by the CamGrid administration team but will be maintained by the colleges, schools and departments who own and control these resources. By creating a separate universe for these machines, the department's autonomy and control over their own system is not lost and when they require sole use of the system, jobs from the vanilla universe can be stopped. This is similar to the NGS option of affiliate sites.

Two other unsupported universes exist on the CamGrid. These are the Parallel Virtual Machine (PVM), and the Globus environment. While not outlined in the paper the use of eScience certificates in the Globus universe suggest that this environment supports connections out of the University of Cambridge to the NGS (Calleja et al. 2004).

Chapter 2.2: Overview and Analysis of OxGrid

In the paper titled "OxGrid, a campus grid for the University of Oxford", David C. H. Wallom and Anne E Trefethen outline the requirements kept in mind before the various HPC resources that existed within the University were pooled to form a grid. The OxGrid can be defined as mainly a campus grid of clusters.

The primary objective was to provision for large amounts of data, as it was felt that as any institution grows the amount of data generated would increase exponentially. The authors also felt that with the move by the arts, humanities and social sciences into eScience this would become especially true. As data mining is the primary tool of research, provisioning for large amounts of data was the priority. As part of data-mining, the quality of the data is dependent

on the amount of metadata associated and embedded with it and as research spins off from an original dataset replication would also increase thus the system should be provisioned to handle a *"steep increase in the volume of data"*. The other objective was to seamlessly link all the resources within the University and those available externally through the NGS and Oxford Supercomputing Centre (OSC).

Taking a bottom-up approach to the OxGrid, it can be seen that distributed across the University of Oxford are various Clusters and super-computers that are owned either by the colleges, departments or research groups. To link these systems and the NGS and OSC systems to provide a seamless system for end users at the university an OxGrid Control System has been set up. This enables all users in Oxford, whether their college/department/research groups owns a cluster or not, to use the resources. It also saves the university the expense of setting up a centralised data-centre and the cost of running these small, job specific clusters located in research centres falls to the colleges hosting them.
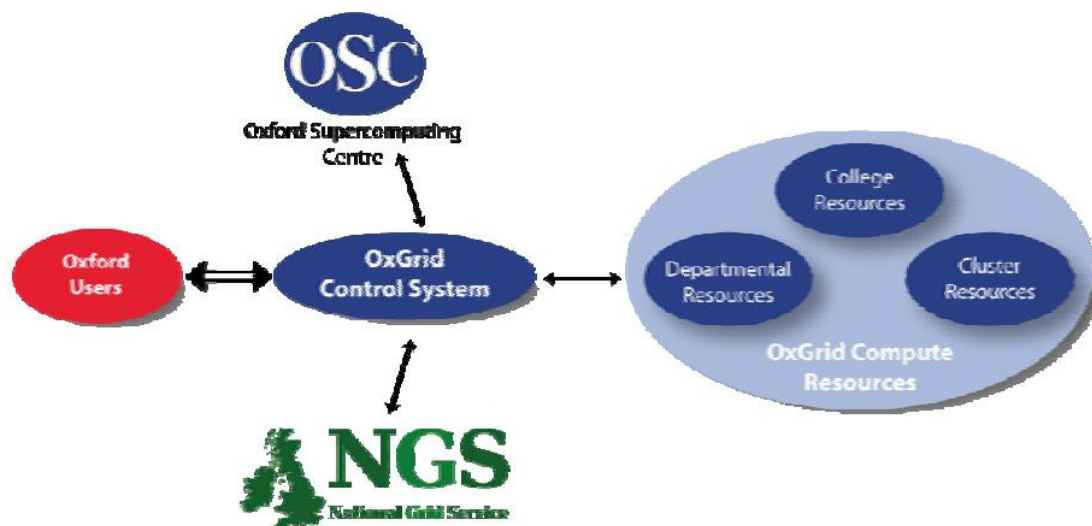
Figure 2: OxGrid Physical Architecture (Wallom & Trefethen 2006)

Linking their clusters together is a Resource Broker which is implemented using Condor-G. Condor is a versatile middleware that can be deployed at a cluster or grid level. Condor-G is a grid level implementation of this resource (Condor is further explained in Chapter 4.4: Condor Overview in

Clusters and Grids). This resource broker receives job requests in the form of scripts which outline the entire hardware requirements and then Condor-G looks at its database to match the job to the appropriate hardware.

To validate users from workstations and desktops within the campus the OxGrid employs the use of X509 certificates which authenticate against a Kerberos domain authentication system. So users within the Oxford campuses can SSH into the system once they get an X509 certificate issued by the OxGrid Control System and the users details are fed into Kerberos authentication system. For users connecting from outside the Oxford University campus (whether they are actual students/staff of Oxford or people from outside the organisation) X509 Certificates issued by the UK eScience council are required. These certificates authenticate using the Globus Grid Toolkit (further explained in Chapter 4.5: Globus Overview) through a service known as Globus MDS using the Grid Laboratory Uniform Environment (GLUE) schema. The figure below outlines the main components of this system.
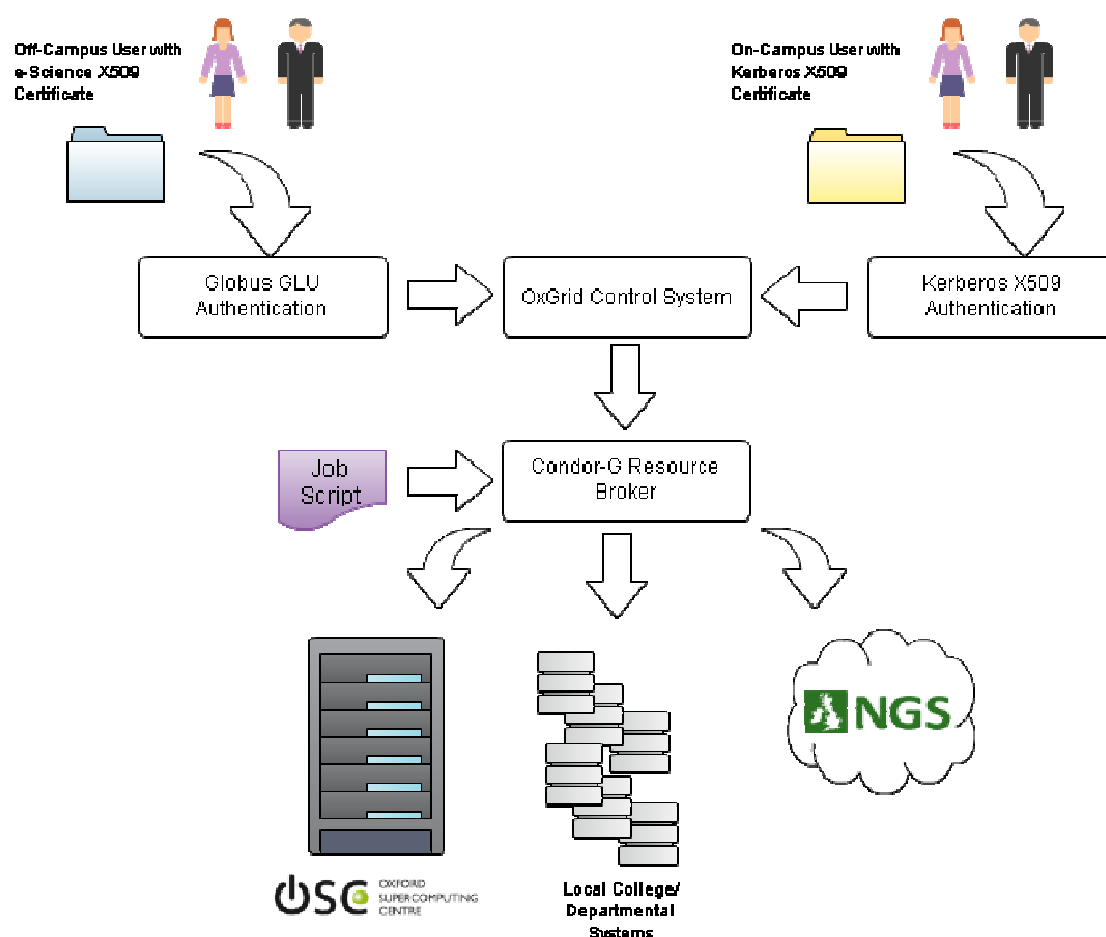
The OxGrid has a locally developed Virtual Organisation Management System so that departments and schools do not lose their priority over their own machines. This VOMS is also the system that lets Condor-G know what resources are available in each cluster and what middleware/job scheduler (e.g. PBS, LFS, SGE or Condor) is installed on each system.

According to the paper to boost the processing resources there is a provision to use Condor Pools to cycle steal from lab machines around the university as well, similar to the CamGrid system. (Wallom & Trefethen 2006)

Chapter 2.3: Overview and Analysis of White Rose Grid (WRG)

The White Rose Grid is a geographically distributed grid across Yorkshire in the north of England. It is formed by the collaboration of the University of Leeds, University of Sheffield and the University of York. These White Rose Universities already combined to form a Virtual Organisation (VO), called the

White Rose University Consortium, and thus the administrative infrastructure required for such a resource sharing initiative are already in place (Padgett et al. 2005).

This sort of system allows the participating Universities to bid for major funding and mega projects as the WRG can show the required man power and infrastructure. The purpose of setting up the WRG was to give the WRG a foothold in industry and eScience research. This is done by focusing on decision support, diagnostics and problem solving environments, and building on the experience already held at the member institutions. The aim is to partner with organisations like Yorkshire Forward to help meet the regional demand for Grid technology and finally to support and enlarge new and growing scientific communities working in cutting edge fields like bio-technology, aerospace, tissue engineering and healthcare.

The WRG has been laid out to work in parallel with the NGS so that the two grids can interoperate seamlessly. Within the WRG there are four nodes comprising three clusters of high performance machines from Sun Microsystems and two Intel processor-based Beowulf systems (the larger one with 256 processors) from Streamline Computing, in total delivering over 450 CPUs with a large file store as integrated computational facilities.

All nodes in the WRG use the same software stack to provide users from each site a uniform computing environment. All the clusters and super-computers use the Sun Grid Engine Enterprise Edition (SGEEE) to manage workloads, resources and policies. The Globus Stack between each node also ensures compatibility with the NGS. One of the two nodes at Leeds performs the function of a Partner site on the NGS.

Aside from the standard SGE and Globus tools available for job submission and information services the WRG have developed specialised portals using tools such as Grid Portal Development Toolkit (GPDK), Apache Tomcat, JetSpeed and GridSphere.

The White Rose model is particularly important in the development of a Huddersfield University grid as the three campuses of the University span an area equal to the White Rose Consortium. The underlying JANET infrastructure between the campuses of the University of Huddersfield is also similar to the WRG. Any social or governmental considerations to be taken before deploying a Grid across the three campuses will be similar to those considerations taken by the WRG, as two sites of the Huddersfield sites overlap with the White Rose nodes (Dew et al. 2003).
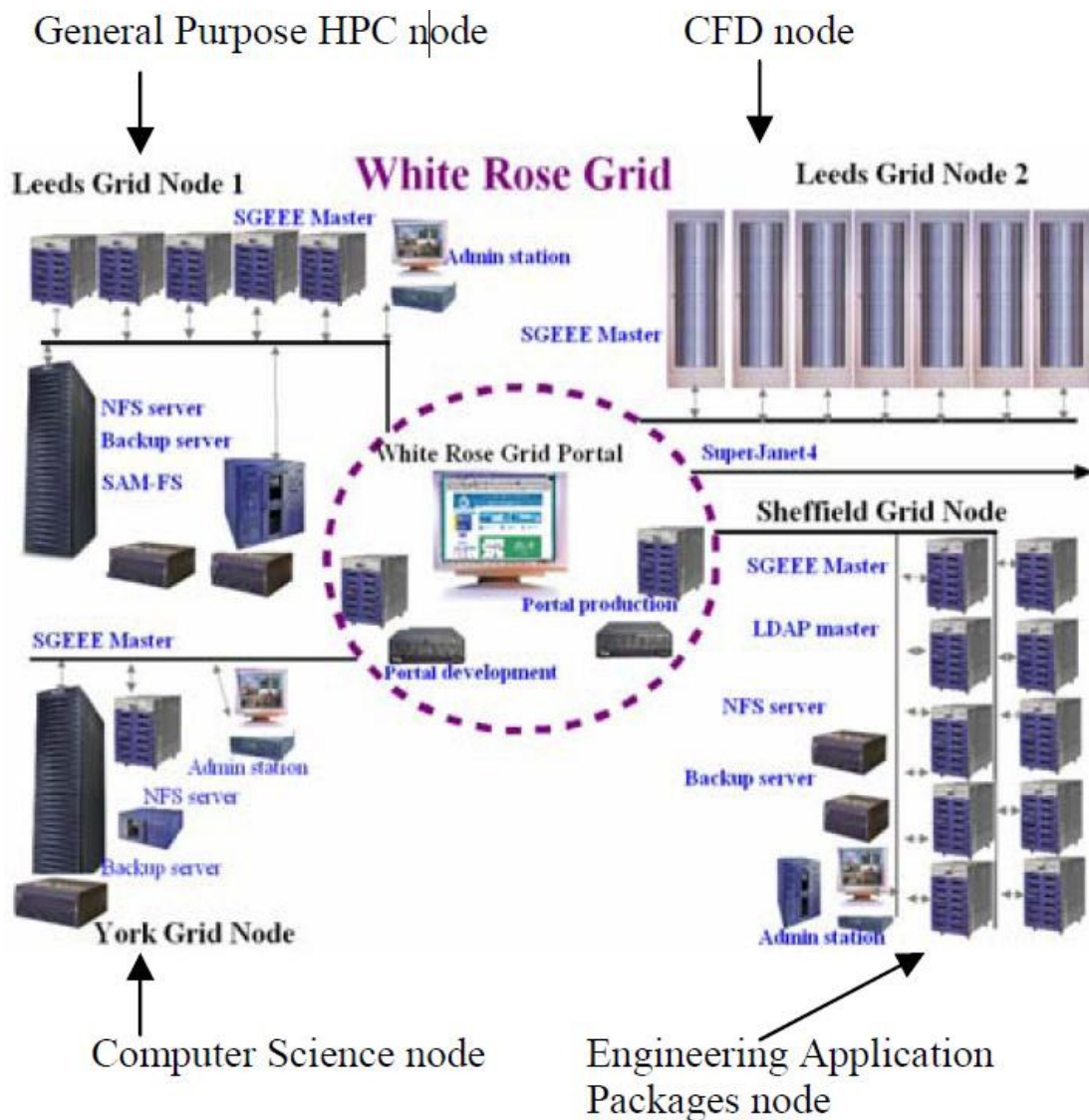
Figure 4: White Rose Grid Architecture (Dew et al. 2003)

Chapter 2.4: Overview and Analysis of LHC Computing Grid (LCG)

The Large Hadron Collider in Geneva, Switzerland is the world's largest super collider and scientists developing and using this system are hoping to recreate moments right after Big Bang in order to observe elements and particles that were last seen at that time. The two main experiments, ALICE (to detect the 'god-particle') and ATLAS (to detect the heavy compounds that existed at the time of the Big Bang) generate data in the peta & exa scales. It would be too costly for any organisation to setup a data centre and to buy a machine to handle such a large scale of data and simulations.
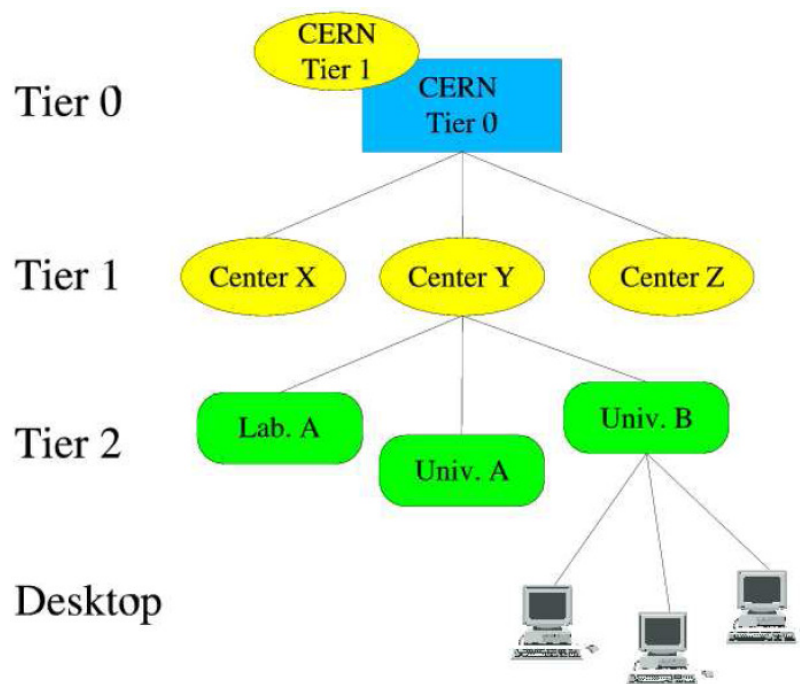
Figure 5: LCG Architecture (Berlich et al. 2005)

To handle the issue of computing power the Large Hadron Collider Computing Grid (LCG) was setup. A multi-tier architecture of collaborative research centres and bodies has enabled scientists and engineers conducting the experiments to easily move the data to the teams of scientists waiting to process this data. The super computer at the LHC is defined as the Tier-0 site and the detection and data collection is carried out at this site. Connected to this site are Tier 1 research centres around the world that collect the data from the Tier-0 site and then divide it to the participating member institutions. These member institutions forming Tier-2 research levels are comprised of specialised labs and

27

Universities. Up till this point the centres making up Tier-1 and Tier-2 sites all have large scale super computers or clusters but the final tier of machines is the desktops of the researchers. This multi-tier architecture is made possible by a workload management system known as MONARC (Models Of Networked Analysis at Regional Centres). The LCG felt MONARC was the best way to manage their simulations as the Globus and Condor Toolkits do not provide the automated selection of target resources.

Several middlewares and combinations of middlewares have been generated due to this project. Aside from Globus and Condor, the LHC project has led to the development of middlewares such as gLite (implemented by EGEE), EDG (European Data Grid), LCG, LCG-2, Unicore, Cactus and AliEn. AliEn is the Alice Environment, which is the middleware specifically, implemented to get data from the ALICE project. This middleware follows a pull architecture rather than a push. In simple terms, after the Tier-0 site has collected data from the ALICE project, the Tier-1 and 2 sites then 'pull' the data that is relevant for their calculations. AliEn thus is a large meta-grid as the data has to be clearly marked for the remote sites to identify the required data and then to pull it.

Project	EDG / DataGrid	LCG	ALICE experiment	EGEE	D-Grid
Project Focus	middleware development and research	middleware deployment, development where needed	building a Grid middleware suitable for experiment's needs	middleware consolidation, deployment, training, support	middleware integration, application support, Grid deployment
Middleware	EDG	LCG-2	AliEn, LCG-2	LCG-2 (EGEE-0), gLite (EGEE-1)	various (gLite, LCG-2, Unicore, Cactus)
Geographical Scope	focus on Europe	world wide	world wide	focus on Europe	Germany
User communities	all interested parties, industry	mainly physicists working at LHC	members of ALICE collaboration	all interested parties, industry	all interested parties, industry
Remarks	Project finished with successful review in March 2004; 9.8 million Euro over a 3-year period. 30 million including national projects of 21 partners	ongoing project, LCG-2 middleware partially based on EDG middleware	small developer base, specialised target audience; web services based middleware	ongoing project, successor of EDG project; 32 Million Euros over 2 years; gLite partially based on LCG-2 and AliEn	ramping up; Expected funding: 300 Million Euros (equally split between research institutions, the German federal ministry of education and research, and industry)
Home	www.eu-datagrid.org	lcg.web.cern.ch/LCG	www.cern.ch/alice	www.eu-egee.org	www.d-grid.de

Figure 6: Grid Middleware work at the LCG (Berlich et al. 2005)

The main lessons can be learned from the LCG experience is that standardisation is important, especially with so many good specialised middleware's around but no single over arching system to integrate. If any middleware development is to be undertaken the emphasis should be on the interoperability of different grid middlewares. Alongside sophisticated features,

a consistent and user-friendly behaviour of grid components is important to end-users of grid systems. This point is particularly important for Huddersfield users as most researchers in Huddersfield are only comfortable in the Windows© XP environment. Support and training play a crucial role in generating a critical mass of users for a grid. (Lamanna 2004), (Berlich et al. 2005)

Section II: Investigation of Problem and the Tools

Chapter 3: Research Methodology

Chapter 3.1: Meeting with the University of Huddersfield Research Community

Before undertaking a project to setup an HPC resource in University, the need for such a resource had to be gauged. The first point of contact for computing in the University is the Department of Computing and Library Services (CLS). In a series of meetings with the CLS Infrastructure Team, Client Consultant for School of Computing and Engineering and later the Manager Data Centre and Network Services it was ascertained that the CLS department tried to introduce HPC at the University but did not get a response from the academic community and as a policy the CLS does not involve itself at School level when it comes to software and departmental requirements. CLS provides the University with the infrastructure and backbone but within each school and department the needs are assessed locally and met with developments funded by 'local' budgets.

In 2009, CLS sent a questionnaire to all departments and researchers to assess what changes or additions the departments would want to the university infrastructure. The replies received mostly stated that the academic community wasn't interested in HPC resources. This response can be explained as anomalous by further analysing the questionnaire. Firstly, the questionnaire was long and tedious, thereby increasing the chances that most users did not in fact complete it. There were some questions relating to many different types of hardware/software/infrastructural changes that a question on High Performance Computing would get lost in the noise. The question regarding HPC was also phrased to ask users if they were interested in "Cluster Technology". The word cluster holds different meaning in statistics, medicine/biology and other fields. It became clear that online questionnaires were ineffective, as the user did not fully comprehend what was being asked.

The best way to get a message across the University of Huddersfield is through email, even though the message often gets 'lost' somewhere along the way. A simple explanation of our project goals was sent across to the various

Directors of Research in all the Schools and Departments across the University. Phrases like "greatly reduce computation time"; "provide faster computing facilities" were used to pass on a sense that our system could perform tasks that ordinary office desktop machines (no matter how new) would never achieve. The response can be described as lukewarm at best. There was no way to tell if the various Directors of Research had taken the email seriously and if they had forwarded it to their research community or made an arbitrary decision when replying. Most directors did not reply. With these results it was decided to engage the research community at a personal level.

Most schools have research open days and the University also holds a bi-annual researchers conference. To get the attention of the researchers in the various departments posters were put up outlining our current work and our hoped outcomes in the researchers' conference. The concept of a central HPC resource was pitched to researchers in between their official presentations on the open-days and during the researchers' conference. As researchers began to show interest, small interviews and briefing sessions were held with them and their supervisors.

Our first response came from the Department of Mechanical Engineering in the form of two researchers, from the Automotive Research Group. They had first contacted this projects supervisor early in the year, as she taught the Parallel Computer Architectures course at the University. One of the researchers expressed the need to get access to an HPC system with large amounts of RAM as his CFD simulations required a highly detailed model and required a high degree of accuracy. He was not able to run these simulations on his office Desktop (A 2.93 GHz Core2Duo with 8GB RAM). The researcher had collected 10 throw away COTS machines to create his own cluster but he didn't have the expertise to make a Linux cluster and he felt that those old machines would not have been able to handle the simulations on a Windows© platform. Despite this, the researcher had spent valuable research time trying to implement a SUSE Linux cluster.

The second researcher had many long running simulations and to overcome this he had managed to get permission to open remote shell (RSH) ports on a handful of lab computers. This enabled him to use ANSYS FLUENTs parallelisation feature to add multiple computers together via RSH to form a crude cluster. While this met his computational needs there were many drawbacks to this approach. These were:

1. The machines in the lab were left vulnerable as opening the RSH ports made the susceptible to hackers and was against the University IT policy

2. His confidential work was being transmitted across an open network unencrypted, as RSH is not secure.

3. Machines in the lab would be busy and lab users would not get access to them.

4. During the period that the simulations were running, the machines would be left unattended and due to a University policy that lab machines cannot be locked. The researchers profile would be at risk. Users could end simulations prematurely by rebooting the unattended machines.

5. The lab in question was located some distance from the researchers office and he would have to walk between the two locations to run simulations, collect data or even debug problems. The University IT policy does not allow remote desktop connections to lab machines.

The reasons mentioned above are the exact reasons a University or Institution which is shifting its focus to become a research-based organisation needs to develop a HPC infrastructure. Many researchers end up spending too much time focusing on aspects that are not related to their work and lose important research time.

During the poster display at the annual researchers conference members of the chemistry faculty from the Department of Chemistry and Biological Sciences (DOCABS), the School of Applied Sciences expressed their interest in

developing an HPC system in the University and were very interested to see what system we could provide. In further meetings it turned out that DOCABS were already registered with the UK eScience council as a local registration authority for Huddersfield and could issue IDs to be used on the NGS. DOCABS had also invested in two small clusters to run simulations locally before submitting to the NGS. The SAPP cluster was made up of 5 Quad Core AMD machines connected using a gigabit interconnect and the ASIM cluster was 8 AMD Dual Core systems with large scratch disks for simulations that generate a lot of temporary data.

Due to space, power and health and safety requirements, DOCABS were unable to house both clusters on their premises. As the different schools around the University are not made to be machine rooms, the "store room" where the SAPP nodes were housed was close to a chemical storage area. With the ASIM machines turned on the heat generated became a health and safety risk. The ASIM cluster also suffered a setback as due to power surges one of the nodes stopped working. To get us started on our project, DOCABS were willing to donate the ASIM cluster provided they could use the resulting system.

In the chemistry department it was learnt that users would simulate problems in molecular dynamics by implementing codes written in FORTRAN. In general the researchers in DOCABS were used to a Linux environment and because of the existing clusters and work on the NGS would be able to quickly adapt to a new system.

During the open-day presentations in the Department of Computing a Senior Lecturer and his researchers presented their work with image recognition and detection algorithms and they expressed their need to speed up the process. Their research involved using codes written in MATLAB and LABVIEW to analyse and detect specific changes between frames in hours of CCTV footage. On a single machine 30 seconds of footage had to be slowed down in the region of greater than 3 minutes per clip this created the problem that while a machine processed one set of data large amounts of data would begin to collect and would overwhelm the system. Parallelisation of the problem by

dissecting frames and distributing frames across several cores would improve the speed, as the operations carried out on each frame are repetitive. Positive results could lead to the creation of a small dedicated cluster that would make this a real-time process.

The lecturer also mentioned that the Department of Gaming and Animation were also looking for possible solutions for a render farm for their 3D world. 3D work being undertaken by the School of Computing involves professional contractual work as well as academic and research projects. The Computing department in Barnsley already owned a MAC based render farm for work undertaken in the Second Life project. The Department at Queensgate want to implement a similar system.

Chapter 3.2: Data-gathering through Analysis of the Universities software pool

A department wise look at the different software packages will shape the architecture of the new system. Within all the software licenses there will be teaching licenses, research licenses and professional licenses. As research is the priority, software that actively supports the research community will be paid special attention.

Department of Engineering

MATLAB: Matrix Laboratory (distributed as MATLAB) is a high-level visual environment for mathematics-based problems. This software is mostly preferred when modelling and designing systems as the interface is more intuitive and user-friendly than FORTRAN or C++. For large complex simulations and evaluations MATLAB provides its Distributed Server and Parallel Computing Toolbox. Users will have to model/design their problems using the Parallel Computing Toolbox to make their problems divisible on a cluster (Mathworks 2010).

Users on their office systems or lab computers can use this software. The Department of Engineering and Technology already has at its disposal 32 Parallel Computing Toolbox's that have been deployed in the Embedded System Laboratory. These will have to be moved around depending on the demand. To simulate the models created in the Parallel Computing

34

Toolbox the files would have to be moved to the cluster and submitted to the MATLAB Distributed Server that will use nodes in the cluster to carry out the simulations.

Currently the University does not hold the Distributed Server license but as per the Systems Group Roadmap, a 32-node licence of the server is scheduled for purchase for the academic year 2010-2011. On the NGS, MATLAB can be found at the Oxford and Leeds nodes but requires special licenses for use.

ABAQUS: This Computer Aided Engineering (CAE) tool is used for Finite Element Analysis (FEA) in the Department of Engineering and Technology (Simula 2009). The Mechanical Engineering subject area holds the license to this tool and uses it in teaching in most of their course pathways as well in research. The Centre for Precision Technology (CPT) at the University of Huddersfield, which works with the National Physics Laboratory (NPL), also uses this tool for much of their ground breaking research and enterprise work.

FEA is the numerical analysis technique used in solving elasticity and structural analysis and has been expanded to calculations in fluid dynamics and electromagnetic or any problem which are expressed as partial differential equations or integral equations. CPT puts ABAQUS to use in part design and testing before it enters the manufacturing process.

The Department of Engineering and Technology holds 40 ABAQUS CAE (designer) licenses, which allow users to make their models and 15 *'tokens'*, each of standard, explicit, foundation and aqua solvers/simulators. They also possess 1 'token' of the Euler-Lagrange, Cosim and Multi-physics solvers. There are also 16,384 tokens for parallel processing in the license pool. While this sounds as if there are many licenses for simulations, there are in fact very few. ABAQUS has a complicated licensing system where a simulation takes 5 tokens of the required solver per simulation. If multiple threads are used then it takes 3 more tokens from the solver and 2 from the parallel bin. This means that with the current pool of licenses a user can only scale to a quad core

simulation with any of the above-mentioned solvers. To complicate matters further, there is a pot of licenses called *'abaqus'* with only 15 tokens in it. This pot restricts the number of instances of ABAQUS which mean that if a user is running a standard simulation across 4 cores he will use 8 parallel licenses, 14 standard solver license *as well as* 14 *'abaqus'* licenses, which will restrict any user for starting another instance of *'abaqus'* solver anywhere in the university. To counter this problem, future licensing will have to be done keeping the HPC system in mind as implementing this tool on the system will be a big asset for the researchers at the University.

On the NGS, ABAQUS is a major tool found at most STFC nodes and is available to all users at the Rutherford Appleton Node.

COMSOL/OPERA 3d: Both these software fall in the category of Finite Element Analysis packages but these are optimised for applications in Electromagnetics and Electronics. The former is part of the MATLAB family (Simula n.d.), while the later is maintained by Cobham Technologies in the United Kingdom (Cobham 2010). At the start of this research, users only had machine locked USB based licenses so this application could not be deployed on the HPC system. Liaising with the High Performance Computing Resource Centre users of these packages have scheduled the purchase of proper cluster licenses and this software will be deployed for the 2010-2011 academic year.

ANSYS FLUENT: FLUENT is a computer aided engineering tool which belongs to the ANSYS family of CAE tools (ANSYS 2010). Mostly the Automotive Research Group and Automotive Design subject area in the Mechanical Engineering course use FLUENT for problems in computational fluid dynamics (CFD). CFD Problems involve a large number of repetitive calculations of how a fluid would flow across a mesh/shape/object (e.g. airflow in a square room with a heater in one corner, or how air would flow round an automobile as it goes round an inclined bend).

Large complex geometries would result in mesh files that are in excess of 4Million Elements and would not open on standard desktops. Those that would load on modern Quad Core/Core2Quad machines will not be able to complete its iterations, as the RAM would be fully occupied by the mesh. Ideally, FLUENT needs several cores with a reasonable amount of RAM to divide its problem. The other issue is that even with such high specs, problems can take anywhere between an hour to several days to finish[2]. FLUENT mesh files do not need to be specially coded as FLUENT has an algorithm to divide the geometry along the principal axis. CAD designs are made in Gambit (a sister tool to FLUENT) and these designs are exported as mesh files that FLUENT takes along with the parameters of simulation to execute the problem.

The Automotive Engineering subject area holds 45 FLUENT licenses along with 45 GAMBIT designer licenses. These licenses are covered by 45 '*fluentall*' license with limits the number of instances of ANSYS software. This means that users can either run up to 45 GAMBIT or 45 FLUENT instances. More '*fluent parallel*' license are required so that simulations can spread across several nodes without eating into the 45 instances mentioned above. For the purpose of testing and usage the 45 licenses are enough to run on the cluster but each spawned node will take 1 license.

FLUENT is the perfect example of an application suited for a cluster. Depending on the simulation the model can require high amounts of RAM (if it has a complex geometry), large amounts of storage (long simulations with big mesh files will require check pointing and this will create large amounts of data), good network interconnects (as distributed cores need to communicate their results of each iteration, the network interconnect comes into play but is not critical as the nodes do not communicate large amounts of data) and several processors (to help divide the job and reduce long run times). For the engineering department, this application

---

[2] Run time depends whether the strictness of the convergence criteria specified is met or the number of iterations hard coded in the simulation file is completed first. Each iterations time depends on the complexity of the mesh, the core speeds, the available interconnect speeds of the system.

will be used, as the benchmark as it meets all the criteria required proving the effectiveness and usefulness of an HPC system as a tool for research in the University.

Department of Chemistry and Biological Sciences (DOCABS)

Force Field Molecular Dynamics Packages: Used to calculate potential energies of systems of particles, DOCABS uses DL_POLY (STFC 2010) and North Western Chemistry (Valiev et al. 2010) as applications for these sorts of simulations. Similar to FEA packages, these applications repeatedly perform PDE and Integral calculations on the lattice of a compound. These software packages generate high levels of network traffic and also create large amounts of temporary/scratch data during their simulations. These software's also require above average (>3GB) of RAM in the system to be able to efficiently complete its simulations.

As this software is open source it is found on many HPC systems across the NGS and can easily be deployed on a local HPC system by compiling it using Fortran77/90 compliers in a Linux environment.

The Scientist at the Daresbury Laboratories writes DL_POLY, which is a major node in the NGS. This software will also be used to benchmark the effectiveness of our HPC system as it exhausts network, data write speeds and processor speeds while simulating and thus will help us find bottlenecks in the system.

Atomic and Molecular Electronic Structure Applications: The Department of Chemistry use GAMESS-UK (CFS 2006) and Metadise (Watson & Oliver 2004) for these applications. In engineering, researchers working in structural analysis use LAMMPS (SANDIA 2010) do to similar simulations.

These codes can be defined as embarrassingly parallel in nature and thus a problem can be divided up into small chunks and be distributed to many cores. As a rule of thumb the more cores present the better. Fast interconnects are required to complete each job. GAMESS-UK in particular can break down problems into tiny jobs and then submit to the

cluster finally collecting all the results from the jobs in the end. As many as 10,000 small jobs can enter the queue on a cluster for processing and can be completed within 20 minutes[3].

3D Design (Product and Transport)

Autodesk Software: The School of Art, Design and Architecture uses Autodesk 3D Studio Max (3dsMAX) and Maya for enterprise work in their Product Design, Digital Media and 3D Design Courses. While Maya and Metal Ray (the 3dsMAX renderer) have Linux versions, the licensing held by the Department restricts the usage to Windows© only (Autodesk 2011).

Typically digital rendering tools require large amounts of RAM and use large input files for execution. These software also generate large output files. A typical movie or animated scene will have many large source movie files (typically 3GB/minute of footage) overlapping each other and many texture, image, transition and audio files that go together to make the scene. All these need to be loaded up, which requires a lot of RAM. The processor then renders the composite frame then collects all the frames together to create a movie. A single core rendering a movie can have run times in months and would need large arrays of memory and fast storage disks.

These image-processing applications are highly parallelisable as each frame is not dependant on the previous frame and does not influence the next frame. Therefore a 300-frame video can be distributed to 10 cores each getting 30 frames and there would be a speed up of a factor of 10.[4]

The Animation Subject Area in the Department of Informatics located at the Barnsley Campus does use a MAC based render farm for its rendering jobs.

---

[3] Based on the observations of jobs submitted to the Eridani Cluster

[4] The actual run time would not decrease by a factor of 10 as the earlier mentioned large files would need to be transmitted across a network interface and this would create a large bottleneck. Parallelisation would be beneficial in videos that have several thousand frames or greater. A typical one-and-a-half-hour feature film encoded in PAL would have in excess of 155,000 frames.

At the UK eScience All Hands Conference in Oxford in December 2009 many users and sites of the NGS were present to show the work they were doing on and for HPC systems. On the development side many Universities are developing workflow programs that help users through the help of portals to connect to HPC resources, find the required system and submit jobs using intuitive web-based graphical tools. A single workflow to manage every user's workflow no matter which application they want to run on the system. Leading research in this field is being done at the University of Manchester. Their work-flow management application is called Taverna (Hull et al. 2006). This sort of work can also be undertaken at the University of Huddersfield by the department of Informatics, provided that the underlying technology is available.

FLUENT is available on the NGS at the Leeds node and as discussed with the administrators from Leeds and Sheffield this node is within 25mi of the University of Huddersfield Queensgate Campus and so our local users can easily scale to this resource without violating the FLUENT EULA. More licenses will be needed to scale up to the larger systems at Leeds.

While the typical eScience applications in bio-chemistry and engineering dominate many of the simulations running on the NGS, world leading research work is also being carried out in the humanities, in fields like criminology, dance, game theory and sociology. At an NGS road-show in York, Dr Luke Rendell from the University of St Andrews presented his findings on social learning behaviours by creating a large massively multiplayer computer game that was run and processed on the NGS (Pennisi 2010).

In June 2010, this experience of setting up a Campus Grid at the University of Huddersfield was presented at the High Performance Computing Symposium (HPCS) in Toronto, Canada. HPCS is Canada's largest conference relating extensively to HPC and is attended by scientists and educational institutions from across the world. During the training sessions and the tour of the SciNET facilities, University of Toronto's HPC Resource Centre, their method of implementing a grid was fully explained to participants. With multiple clusters

in the same data-centre, a common file system was used to link the clusters to a large array of JBODs (Just-a-Bunch-Of-Disks). On the JBODs were node images in Ubuntu, Debian, SUSE, Slackware, CENTOS, Solaris and Windows©. If needed the nodes in a cluster could be rebooted and the controller in the system could distribute a different operating system out to each node. As explained by Niel Bunn of IBM, and one of the three main architects of the system, within 7 minutes the cluster can reboot with any operating system. The limitation is that the whole system needs to switch OS.

Many journal articles also give an insight as to how other institutions are deploying their software in an HPC environment. In the article "The design and implementation of Render Farm Manager based on OpenPBS", authors Jing Huajun and Gong Bin (Huajun Jing & Bin Gong 2008) and in "Grid-based Computer Animation Rendering" Anthony Chong, Alexei Sourin and Konstantin Levinshi explain how Computer based animations can be deployed using OpenPBS and Globus to reduce computation time and hardware stress by distributing loads (Chong et al. 2006). Similarly Petri Kaurinkoskis et al in a paper titled "Performance of a Parallel CFD-Code on a Linux Cluster" show how CFD simulations are being carried out at the Helsinki University of Technology in Finland (Kaurinkoski et al. 2001). Many papers like the ones mentioned above will serve as road-map on how to deploy specialised software and meet the needs of the research community at the University of Huddersfield.

Chapter 4: Tools

In a server environment the choice of which operating system is boundless. Keeping the applications of High Performance Computing in mind still doesn't reduce the number of possible Operating Systems to choose from. IBMs operating system formerly OS/5, Sun Micro Systems Solaris, UNIX, MINIX, Linux, Windows© Server, Apples' OSX-Server are just some of the possible options. Hardware and budget for this project reduces the choices to Solaris, Linux and Windows©. The University has an existing license pool for Windows© and Solaris flavoured operating systems. As the results of the analysis of the University software pool suggested, both a Windows© and a *NIX system would be required.

Chapter 4.1: Which Flavour of *NIX

Though Solaris is widely used across the NGS at several sites and therefore there is support through academic channels, it is felt that its commercial license would make support for users through forums harder, thus putting more load on the management team. Linux is the next option which has support in the University Computing Services, Department of Computing, the eScience community and the World Wide Web in general.

With LINUX, the problem of choosing which flavour should be implemented is a big problem. If the evolution of Linux is looked at since 1992 there are three major distributions: Slackware (late-1992), Debian (mid-1993) and Red Hat (late 1994). While Slackware and Debian are still free, Red Hat (in its pure form) can only be found as Red Hat Enterprise Linux, which is no longer free. Slackware and Debian both are very true to the Linux core and are found to be harder to maintain. Each of these three distributions has free sub-distributions which are well maintained and come with adequate support. These are: openSUSE (Slackware), Ubuntu (Debian), Fedora and CentOS (Red Hat). The biggest problem when choosing a Linux flavour is ensuring that the software will be supported for the life-cycle of the hardware and that the applications running on the OS support it; have been tested on it; or have an active user base for online support. Some flavours of Linux just fizzle out or get bought by companies and then users are stuck, as the freely available repository of applications is no

longer available and users have to resort to compiling every piece of software they need. (Lundqvist & Rodic 2010)

OpenSUSE is the open source community developed version of SUSE Linux Enterprise Server (SLES) owned by Novell. Ubuntu from the Debian stream is a very user-friendly operating system that seems to target Desktop users more as every April and October a new version is released with many innovative features for the home user. After Fedora (Red Hat family) version 9, Fedora appears to go the same route as Ubuntu with more user-desktop based features. While both Ubuntu and Fedora still release server versions openSUSE and CENTOS both concentrate on the server and enterprise class operating systems (DistroWatch 2010).

Detailed testing was done to ensure the OS chosen supported the applications to be run, support the middleware and management tools to be deployed, was easy to manage, would stay maintained, and would support the hardware. On the basis of these criteria the flavour of Linux chosen was CENTOS.

	New Hardware Support	OSCAR 5 Support	OSCAR 5.1 (rc1/b1) Support	OSCAR 6.0 Support	Long Term Support	Application Support	Ease of Installation	Usability Straight out of the Box	Non-Technical User Friendliness
Fedora Core 5		√					√		
Fedora Core 9			√	√			√	√	
Ubuntu 8.10	√			√	√	√	√		√
OpenSUSE 10.1			√			√	√		
CentOS 5	√		√	√	√	√	√	√	

Figure 7: Linux Flavour Choice: Decision Table

Chapter 4.2: Microsoft® HPC Solutions

Microsoft® has also been involved in developing software/operating system solutions for high performance computing and has come up with the Compute Cluster Pack (CCP) and the High Performance Computing Server (HPCS). The Compute Cluster Pack is an old piece of software used to patch Windows© Server 2003 (The standard Microsoft® server OS up till the release of Server 2008) and thus has large support for applications and has been maintained for a long time. Both the CCP and HPCS have an unfortunate restriction found in all Windows© based systems: it can only maintain active

connections with up to 16 systems and no more. A recent (2009) release of HPCS 2008R2 as a free beta software has given very positive results.

This new beta system has removed the problem defined above and thus has enabled scientists to create large clusters on the Windows© platform. The cluster has to be part of an active-directory (AD) and all user management takes place at an AD, level thus utilising Microsoft®'s powerful user management system. Microsoft® SQL Server manages the records of Nodes, Users and Jobs in the system. Using the XML mark-up language users can compile job files and submit to the cluster using a GUI tool from their office desktops or using tools like Power Shell, which give users access to a power full command line interface to interact with the cluster.

While this package takes care of HPC needs, a patch provided by this software can be installed on the new Windows© 7 operating system, which will allow the Windows© 7 nodes to link to the cluster when idle and share their computing power. This provides for a High Throughput Computing system which allows the cluster to virtually grow to the size of all the machines in the organisation on off-days when majority of these desktop systems would be idle (Microsoft® 2008).

Chapter 4.3: Linux Cluster Middleware

On a Linux system there are many ways to develop a cluster. The simplest method is a manual method where each node in a system is installed as a separate machine. One machine becomes the head-node and a job scheduling software is installed on this system. Once each node is installed with the operating system, special rules are created within the firewall and the client tool for the job scheduler is also installed on each system. The applications that the cluster will run need to be accessible and every node should be able to access every users' files. The easiest way to do this is to set the node home folders to mount over the network from the head node through the use of Network File System (NFS) shares. This is a very tedious process as each node has to be manually configured and updated if changes need to be made or a new

application is deployed. This method is only useful if the HPC system is comprised of 3-4 systems.

An easier method of deployment and management is to use a Cluster Middleware which links all the steps above with useful user and administrations based tools like Cluster Command and Control (C-3) Tools, deployment tools, job scheduling tools and user interfaces. Two very popular open-source HPC middleware's for Linux are Rocks and Open Source Cluster Application Resource (OSCAR).

Rocks is a complete cluster-on-a-CD distribution with all the tools needed for a cluster. Based on CENTOS, Rocks can be classified as another flavour of Linux. It allows for easy installation of the head-node system and then an easy deployment of nodes. It incorporates many tools for User and Job management similar to OSCAR. OSCAR is a middleware that comes with similar tools as Rocks, but is an independent application that can be installed on many OS's. Both systems provide an intuitive menu to do all the steps mentioned in the manual method.

It is for this reason the OSCAR has been chosen as the proposed systems middleware as in case the eScience community moves to another operating system that is not similar to CENTOS or Scientific Linux, the same middleware can be used to reduce a layer of complexity as a new system is deployed (Vallee 2010).

Chapter 4.4: Condor Overview in Clusters and Grids

The goal of the Condor® Project is to develop, implement and deploy mechanisms that support High Throughput Computing (HTC) on large collections of geographically distributed computing resources. The Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput keeping in mind social and legal implementations.

Condor is a fully-featured batch system for HPC. Condor provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring,

and resource management. Users can submit their serial or parallel jobs with a specific execution rules and Condor places them into a queue and find the resources to execute the job on. Condor provides a monitoring tool to track job progress, and ultimately informs the user upon completion.

While providing functionality similar to that of a more traditional batch queuing system, Condor's architecture places it between a cluster and a grid middleware. Condor can be used to manage a cluster of dedicated compute nodes and similar to the Windows© HPC 2008s patch on Windows© 7 Condor can effectively harness wasted CPU power from otherwise idle desktop workstations. For instance, Condor can be configured to only use desktop machines where the keyboard and mouse are idle. Should Condor detect that a machine is no longer available (such as a key press detected), in many circumstances Condor is able to transparently produce a checkpoint and migrate a job to a different machine which would otherwise be idle. Condor does not require a shared file system across as it can transfer the job's data files when a job begins execution. As a result, Condor can be used to seamlessly combine all of an organization's computational power into one resource and can be used to share resources between different organisations (Wisc EDU 2010).

Chapter 4.5: Globus Overview

The open source Globus® Toolkit is a popular middleware for grid computing and allows users to harness computing power, databases, and other tools securely online across geographic boundaries without sacrificing local autonomy similar to the Condor model. The package includes software services and libraries to enable users to create, distribute and manage jobs across a wide global network of system. If also provides a complete administration package for resource monitoring, security and usage billing. The Globus Toolkit is used in both commercial and educational settings providing solutions for all the sciences and some humanities.

The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability which can be deployed in any combination. The Globus Toolkit

removes obstacles that prevent seamless collaboration. Its core services, interfaces and protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when.

The Globus Toolkit has grown through an open-source strategy similar to the Linux operating systems. This encourages a large community development and support infrastructure. This leads to greater technical innovation, as the open-source community provides continual enhancements to the product.

Globus emerged due to the needs of scientists and engineers that sought to access scarce high-performance computing resources that were concentrated at a few sites.

"Begun in 1996, the Globus Project was initially based at Argonne, ISI, and the University of Chicago (U of C). What is now called the Globus Alliance has expanded to include the University of Edinburgh, the Royal Institute of Technology in Sweden, the National Center for Supercomputing Applications, and Univa Corporation. Project participants conduct fundamental research and development related to the Grid. Sponsors include federal agencies such as DOE, NSF, DARPA, and NASA, along with commercial partners such as IBM and Microsoft®."

With the Large Hadron Collider at CERN scientists have used the Globus toolkit to spur a revolution in the way science is conducted. Much as the World Wide Web brought Internet computing onto the average user's desktop, the Globus Toolkit is helping to bridge the gap for commercial applications of Grid computing.

The Globus Toolkit works with using a Public-Private key interface implemented using X509 SSL certificates which are generated by Certificate Authorities. This way within a single grid the machines can validate each other and users connecting inwards can be verified. Different CAs can sign agreements with each other allowing users from one grid to harness resources on other grids. An example is how the LHC in Switzerland links with the NGS in the UK and Teragrid in the US.

The Globus toolkit provides different tools to allow for users to connect to the grid and to copy their data on to the grid. Once connected, users submit jobs giving a list of requirements, which the Resource Broker in the Grid uses to pair the job to the hardware/software (Foster et al. 2001).
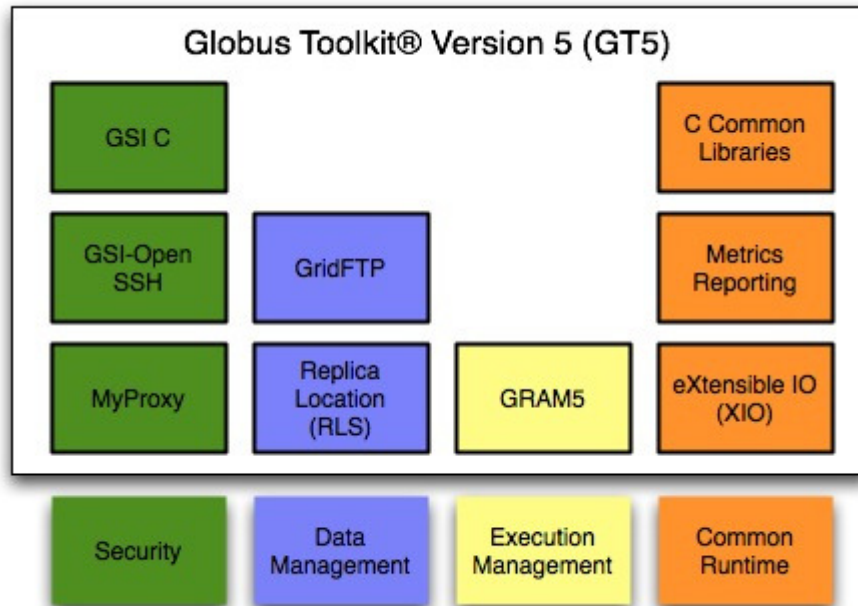

Figure 8: Layers of the Globus Middleware (Foster 2006)

Chapter 4.6: gLite Overview

The gLite distribution is a grid middleware that contains an integrated set of components designed to enable resource sharing. Developed by the Enabling Grids for EScience (EGEE) project, the gLite distribution pulls together contributions from many other projects, including Large Hadron Collider Computing Grid (LCG) and Virtual Data Toolkit (VDT).

gLite middleware is currently deployed on hundreds of sites as part of the EGEE project and enables global science in a number of disciplines, like the STFC in the UK and notably serving the LCG project. Similar to Globus gLite relies on the open source community for development. The gLite Open Collaboration has been established between the EGEE partners involved in the middleware activity as a new framework for the maintenance and future evolution of the gLite middleware, beyond the end of the EGEE series of projects.

"Within the scope of the Collaboration are the following goals:

- *maintain the gLite brand, related names and software products;*

- *coordinate the continued development, promotion and adoption of the integrated set of services which constitute the gLite middleware;*

- *provide other projects with a single interface to the gLite providers;*

- *coordinate the maintenance and evolution of the gLite middleware in response to requirements from its user community (such as resource providers, infrastructure operators, application developers and end-users);*

- *provide the gLite middleware components in an open and accessible manner to the user community; allowing and encouraging community contributions to address problems, port to new platforms, and improve the overall software quality;*

- *achieve interoperability with other Grid infrastructures, preferably through the adoption of established standards, such as those developed by the Open Grid Forum (OGF);*

- *contribute software for deployment within production infrastructures, such as via the Unified Middleware Distribution (UMD) that will be deployed by EGI;*

- *provide community support, for example through mailing lists, discussion forums, help, training and documentation."*

(CERN 2010)

Section III: The System and its Performance Characteristics

Chapter 5: Establishing an HPC System

In order to setup a centralised HPC system at the University of Huddersfield, the first step was to prepare the infrastructure that would support the resource beyond this project. To achieve this goal the following steps had to be taken:

- Gather machines from around the University, buy some hardware from the research budget and then set up a small HPC resource to establish demand. Then to get the support of the university research community for this resource.

- Secure real estate on the Queensgate Campus of the University of Huddersfield to serve as a data-centre for the HPC resource and as an office for the team that will manage the resource. This data-centre and its staff will be hereby known as the High Performance Computing Resource Centre (HPC-RC). The HPC-RC will be responsible for maintaining the HPC system, provide training and tech support to users, liaising between the NGS and the local users' community and promoting the resource to the local academic community. The HPC-RC should be staffed with one Research Assistant and 1-2 post graduate researchers working in HPC.

- Establishing a High Performance Computing Research Group (HPC-RG) whose mandate will be to oversee the research activities in regards to HPC in the University of Huddersfield and manage the policy and staffing of the HPC-RC. The HPC-RG will bid for grants and funding to ensure the sustainability of the HPC-RC as a resource. The HPC-RG will also be the public face of all HPC work carried out in the University and will try to attract researchers and enterprise work. The HPC-RG will need to hire a Post-Doctoral Researcher to complete all its tasks.

- To take over or strongly influence licensing in the University and centralise the process of purchasing licenses. Software being used on the cluster(s) should be controlled by the HPC-RC team and at the start of

year before software is purchased by staff members the input of the HPC team should be taken. To provide the service of licensing, specialised server infrastructure will need to be setup in addition to the clusters.

• Websites for publicity and as a platform for an online knowledge base, a Certificate Authority for local users will also be required and thus adequate infrastructure will be required.

With the following aspects in mind The Queensgate Grid (QGG), a collection of several servers and cluster, has been established to provide an HPC resource. The list of systems running at the time of writing are:

Code Name	URL	Comment
Testbed		32bit cluster for HPC-RC test before rollout
Eridani	eridani.qgg.hud.ac.uk	Main Intel Based Cluster
Tau-Ceti	tauceti.qgg.hud.ac.uk	AMD Cluster donated by DOCABS
Mimosa	storage.qgg.hud.ac.uk	16TB Floating Network Access Storage
Regulus	ngs.qgg.hud.ac.uk	NGS Authentication Node for QGG
Sargas	mech1.hud.ac.uk	Legacy Engineering Flex License Server
Spica	lrc1.hud.ac.uk	HPC-RC Certificate Authority Server
Saiph	lrc2.hud.ac.uk	HPC-RC New Flex License Server
Shaula	lrc3.hud.ac.uk	Legacy Engineering Flex License Server
Bellatrix	bellatrix.qgg.hud.ac.uk	Internal URL for QGG
	qgg.hud.ac.uk	External URL for QGG

Figure 9: Table Showing List of Clusters and Servers forming the QGG

The first step to creating the Queensgate Grid was to set up one or more clusters to meet the immediate needs of the research community and get there and the administration's support.

Introduction to the System

Before the HPC-RC could propose a system to be used by the whole University several operating systems, cluster middleware, software/hardware compatibilities had to be tested. (The results of these can be found in *Chapter 4.1: ; Chapter 4.3:* ; and *Chapter 4.2:* respectively).

System Evolution

The Test-bed system has gone through a series of upgrades to provide for a robust system that can be used for testing and evaluating. The evolution of the system is as follows:

- Keeping to the theme of creating a local HPC resource on a zero budget, the systems that combined to form the Testbed Cluster were throw-away desktop PCs sold under the name of NEC ML-4's. The age of these systems is around 5-7 years old. These are Intel Pentium 4 2.4Ghz Machines with 256MB RAM and 80GB hard drives. The micro-ATX profile of the motherboard and the box like shape of the ML-4 casing meant that they could be easily stacked upon each other and not occupy too much space. As the project started, 8 ML-4 systems were collected and coupled with an old 32-port 100Mbit 3COM hub so that testing could begin. Eventually 9 more systems that were being disposed were scrounged to give the cluster's compute nodes a binary value (i.e. 1 head-node 16 compute nodes). The head-node is an NEC ML-7 which is a Intel Pentium 4 HT 2.8GHz Processor with 512 MB RAM and an 80GB HDD.
- To improve this systems performance and give valuable results with regards to the benefits of HPC systems, the machines RAM was upgraded by purchasing 2GB of RAM per machine making a total of 34GB on the system. Seventeen INTEL Pro/1000 LAN cards, from the Lab teaching pool and a Netgear 32 Port Gigabit Switch, were used to upgrade the Interconnect on the cluster.
- This system remained the primary cluster in the University of Huddersfield from December 2009 up to March 2010 and carried out

FLUENT (SCE CFD Software) and GAMESS-UK (CAE FFMD Software) simulations.

- As of September 2010, this machine has been reduced to the single ML-7 head-node and 4 ML-4 compute-nodes.

- Though several OS, middleware and software combinations have been installed on this system, the stable configuration of this system is CENTOS 5.4 with OSCAR 5.1b2 so that it mirrors the configuration of the major clusters in the system.
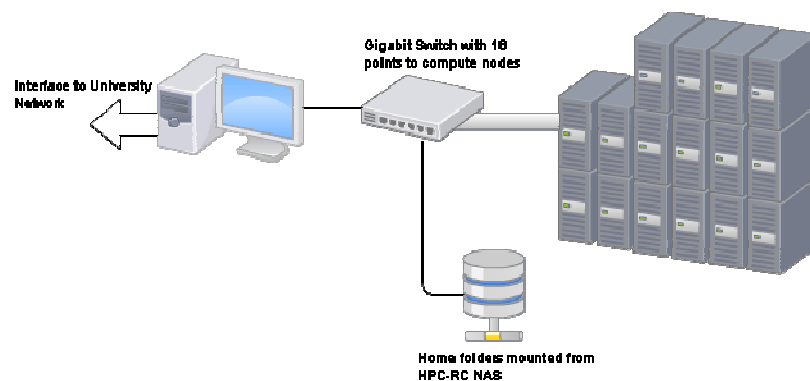
Figure 10: Test-bed Cluster Architecture

Purpose of Cluster

This cluster now performs the role, as the name suggests, of a testbed system on which any major changes or modules to the new clusters can be tested. Major changes such as mounting file systems from Distributed NAS devices or new job schedulers or job scheduling rules can be tested on this before deploying so that the downtime on the main systems is kept to a minimum.

The LINPACK numbers shown in Chapter 6.2: LINPACK Performance established that with the current standard of hardware this cluster is not a High Performance resource; this system can also be used to combine 32GB of RAM which can allow for large items to be loaded on for debugging processes if jobs fail on the other systems.

Figure 11: Test-bed cluster Deployed

Chapter 5.2: The 'DUAL' Boot System

The primary cluster on the Queensgate Grid is the Eridani Cluster. While the system architecture is explained in Chapter 5.3: Eridani architecture and setup, this section explains the basic principles that governed how this system was configured.

The analysis of the SCE, SAS and ADA software pool made it abundantly clear that a cluster based on *NIX systems would not adequately cover the university's requirements. Several pieces of software only ran on the Windows© platform (e.g. 3d Studio MAX for image rendering, Dynamic Studio v3.0 for particle velocimetry), while others had licenses which were platform locked to Windows© platform (e.g. Mental Ray). As this project aimed to establish the need of HPC at the University using existing or open source hardware and software, it was not feasible or possible to make 2 clusters of a decent size which would run the two platforms.

To ignore the requirements of Windows© users was deemed to be detrimental for the project as a large number of researchers would be left out and the initial aim of this project was to prove that there is a demand for HPC systems and therefore the project could not adopt an exclusionary stance. Since ANSYS CFD systems and Ferrari have adopted the Windows© HPC 2008 R2 Platform for their future applications (Baker n.d.), it was agreed that it would be

beneficial to our mechanical engineers if we kept our development road map loosely tied to their applications.

There have been several solutions which allow for imaging of desktop and server computers to contain the Linux compute node files and Windows© workstations executables. Implemented at Blackburn College Carlinville IL, US, this method allows the machines to be deployed in laboratories as normal user workstation and using scheduling software and CPU idle detection scripts machines can be rebooted to join a cluster for computation purposes (Carrigan 2002). Other methods involve imaging both the Windows© Compute Cluster Pack (CCP)/High Performance Computing (HPC) server on all machines along with the Linux compute node software. These systems are either manually rebooted on a time sharing basis (Microsoft® 2007) or are automatically rebooted to Windows© for the job to execute (Bucholtz & Zebrowski 2007). These nodes immediately reboot back into Linux and wait for further instructions.
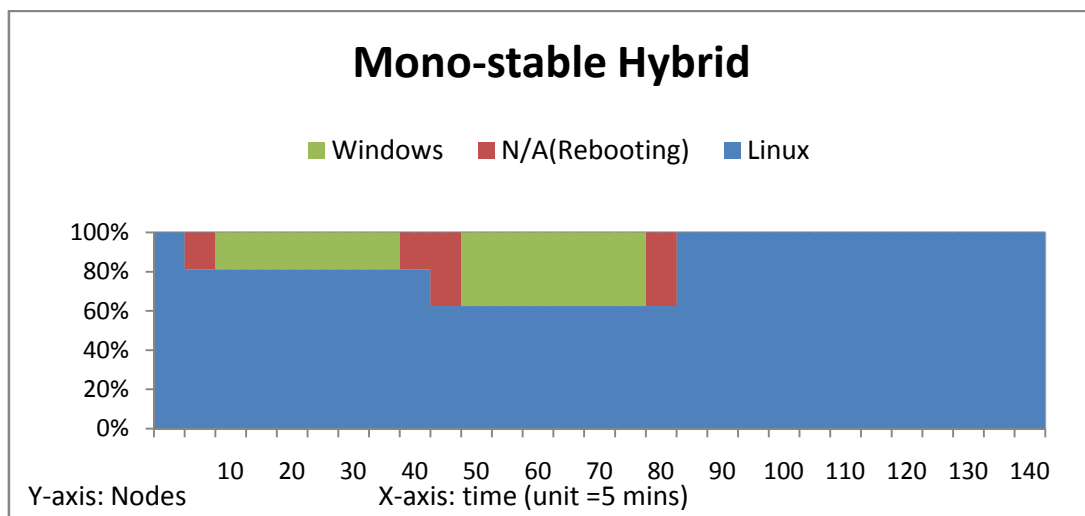
Figure 12: Reboot times in a Mono-stable Hybrid Cluster

As Figure 12: Reboot times in a Mono-stable Hybrid Cluster illustrates, the nodes spend too much time rebooting. From an all Linux, state a job requiring 20% of the nodes executes, forcing the machines to switch to Windows© at the 5 minute marker. Once completed at time 40 these nodes reboot back to Linux where they are instructed to execute another Windows© based job requiring 40% of the nodes. The machines must reboot again. This

56

system also forces users to submit the job in the Linux environment rather than using the Windows© based interface.

The Eridani Bi-Stable Hybrid Cluster

The Beowulf-type Eridani cluster comprises of two head nodes and 16 compute nodes. Windows© Server/ HPC 2008 R2 is deployed on the 'winhead' machine while CentOS/OSCAR is deployed on the 'linhead' machine. Both head nodes must ascertain first whether they have nodes available in their native OS. If not each node must communicate with the other asking it to set a flag in the boot loader and to reboot the machine. Windows© provides an API to detect node status while on the Linux side a script has been implemented that parses the text from PBS and then both operating systems use a script that enables TCP based communication. If the Windows© server (tx-node) requires 2 nodes it will submit 2 'reboot' jobs into the Linux (rx-node) queue and vice versa. This enables the clusters to keep the first come first served scheduler rule. The *'rx-node'* appends the boot loader and reboots an idle compute node.
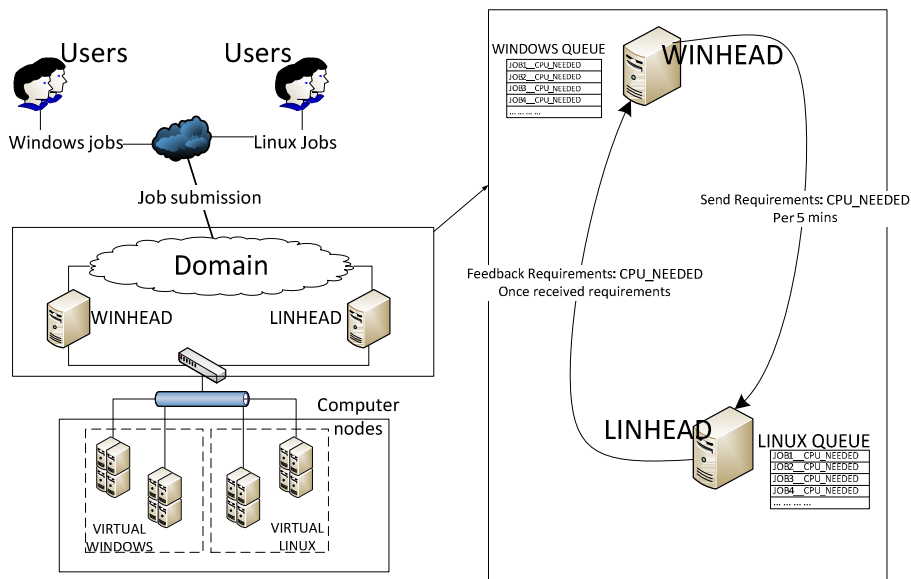
Figure 13: Eridani Cluster, System and Job Scheduler structure

Each compute node therefore has to be carefully deployed using a prescribed method so adequate boot loaders can be set up to enable the reboot. As the Windows© MBR doesn't perform well with Linux, and Windows© cannot make changes to GRUB, a tool called GRUB for DOS has to be implemented. A FAT

partition is required to hold the GRUB and GRUB for DOS boot loaders so that both operating systems can access and modify it.
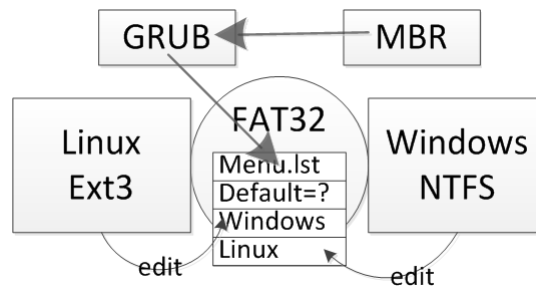
Figure 14: Compute Node Partition Information

It is possible to image the client nodes Windows© first or Linux first. In the Windows© first method the client image is created and only the first half of the hard drive is partitioned in NTFS format. After deploying the image using PXE boot, the Linux image is created in OSCAR. The partition information is given to the *'systemimager'* package to state that the first half of the hard drive is NTFS. Once the image is created the installer scripts within *'systemimager'* are edited to ensure that the Windows© partition is not formatted again. For a Linux first deployment all the partitions can be created as before and the Windows© Server DVD installer is added to the image with an 'autoinstall' script. This will image the Node with Linux and Windows© without any user intervention. Once the installation is complete the HPC pack has to be manually configured to connect to the '*winhead*' machine.

The Bi-Stable Dual-Boot Linux/Windows© system, we have developed and deployed, has allowed the School of Computing and Engineering at the University of Huddersfield to provide a high performance computing resource for both Windows© and Linux based applications. As shown in Figure 15: Throughput of Bi-Stable Hybrid Cluster the throughput of the system improves with this bi-stable arrangement.

This system therefore has met its requirements of being economical by using open source software and the software currently available in the department, and has utilised existing hardware to provide the maximum performance in both Linux and Windows© based environments.

This methodology for establish a bi-stable system was accepted for and then presented as a paper at the UK eScience All Hands Meeting Conference in Cardiff on September 3rd 2010.

Chapter 5.3: Eridani architecture and setup

The Eridani cluster is main workhorse system on the Queensgate Grid and in six months had completed over eighteen and a half thousand jobs even though there have been scattered downtimes amounting to two and a half months. That is approximately 180 jobs/day for four and a half months. Further details of usage can be found in 'Chapter 10.1: Current Software Deployment'

Introduction to the System

The Eridani cluster also sticks to the central aim of establishing an HPC resource at minimal to no cost. After testing and analysis of the software the cluster was deployed based on the considerations in 'Chapter 1.1: Problem Definition and University Requirements'; 'Chapter 1.4: Aims of the Project'; 'Chapter 2: Literature Review and Published Experiences'; 'Chapter 3: Research Methodology'; 'Chapter 4: Tools'. It was made up of systems that became

available from an Engineering Laboratory once teaching ended. This cluster was opened for use to the greater research community so that the benefits of this HPC system could be seen.

Within the first two months of the system being online, the record number of jobs and the demand for the resources warranted upgrades additions and an increase in capacity of the initial system. The feedback the Department of Engineering received lead to the initial system being permanently donated to the HPC-RC and provided for a budget for upgrades.

System Architecture

The Eridani system at the time of writing has 32 compute-nodes with 1 Linux head-node and 1 Windows© head-node. Traffic in and out of the system is controlled using a machine configured as a network address translator (NAT) which by implementing Port Forwarding can selectively give users access to specific resources all on 1 DNS name (i.e. eridani.qgg.hud.ac.uk). All compute-nodes in the system mount their 'home' and 'applications' folder from a floating Network Accessible Storage (NAT).
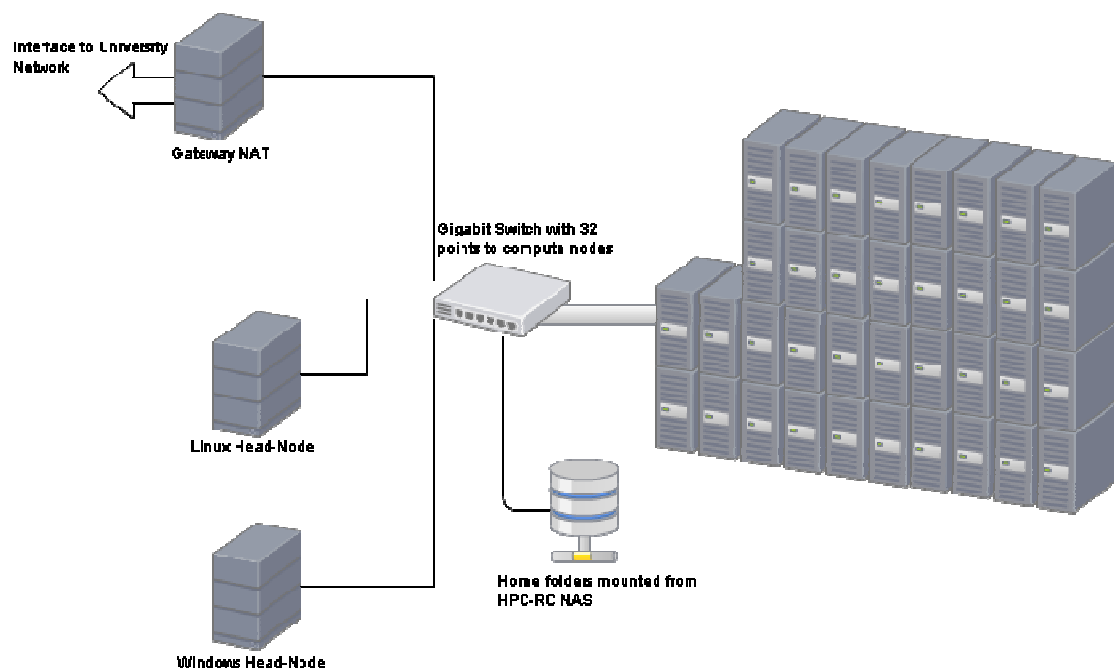
Figure 16: Eridani Cluster Architecture

60

System Evolution

The Eridani Cluster has gone through 3 stages of updates as follows:

- The initial system comprised of 23 machines from Stone Computing UK, which came equipped with 2.33 GHz Intel Core2Quad's on the DQ45 chipset with 2GB of RAM per system. Each system comes equipped with a 250GB Seagate Barracuda SATA HDD which is partitioned according to the rules specified in Chapter 5.2: The 'DUAL' Boot System. A switch that was used in the lab is also used to provide for a gigabit inter-connect.

- As large CFD and MMFD simulations started to tax the system it was observed that the nodes would start filling the swap spaces on the HDD as the physical memory was full. This slowed down the simulations as the HDD write types became a bottleneck. To improve the efficiency of the system, more RAM was bought and the existing 2GB (2x1GB sticks) were recycled into lab computers.

- Due to a robbery, the Linux head node of the system was lost and the system had to be reinstalled and relocated to a safer location, thus emphasising second point mentioned in Chapter 5: Establishing an HPC System for dedicated real estate to house the system.

- As the number of heavy-usage users crossed 25 and the system had completed ten thousand jobs a need for more nodes was felt and for the first time money was invested in dedicated machines for the cluster, adding seventeen more machines to the system with the newer 2.50 GHz processor. One node was fitted with a SATA controller and 8 2TB Seagate Drives were installed to provide for 15TB storage NAS to hold the users home directories.

The final specification at the time of writing is as follows:

Resource	Statistic
Total Systems	37
Total Cores	148
Processing Cores	128
Service Cores	16
Processor	Intel Core 2 Quad Q8200 4M Cache 2.33 GHz 1333FSB or Intel Core 2 Quad Q8300 4M Cache 2.50 GHz 1333FSB
Motherboard	Intel DQ45CB
RAM	4 x Kingston Value 2GB 800Mhz
HDD	Seagate Barracuda 250GB 7200RPM SATA-II
Network	2 x Intel Pro 1000

Figure 17: Table Showing Hardware Configuration of the Eridani Cluster

Figure 18: The Eridani Cluster Deployed

Chapter 5.4: Tau-Ceti architecture and setup

As mentioned in Chapter 3, the Department of Chemistry and Biological Sciences owns two AMD clusters, of which one they could no longer house. This system was given to the HPC-RC to experiment with and was initially labelled the

ASIM (Applied Sciences IBM Machines) cluster. Chemistry still housed their first cluster known as the SAPP cluster (**S**chool of **APP**lied Science). Though powerful, both systems were ageing and kept separately as two distinct clusters were reducing the efficiency of the system.

This reduction of efficiency was due to the fact that as discrete clusters the two systems provided for very specific tasks. The SAPP cluster comprised of 4 nodes with 2 Dual Core AMD Opteron Processors with 4GBof RAM per core. So for tasks which required large amounts of RAM but not much parallelisation these were ideal nodes. But due to the small number of total cores in the system these nodes could not be used for parallelisation above 16 processors, thus leaving the system idle a lot of the time. The ASIM cluster comprised of 8 nodes with single Dual Core AMD Opterons but had large 500GB scratch disks for simulations that needed to write large amounts of data to disk during the jobs life-cycle. Once again the 16 cores were not enough for large simulations and the number of users that needed scratch disks in their system is estimated to currently be less than five. Also, unlike SAPP, where sixteen cores are in 4 nodes, the 8 node distribution of the ASIM cluster added for more network latency.

After 3 months of hosting the ASIM cluster and providing technical and user support to the SAS researchers, the Chemistry Department were agreed to give their SAPP cluster to be integrated within the proposed Queensgate Grid. Space was found in a networking patch room in an engineering complex where adequate cooling and power was available for a 24U rack. Both the ASIM and SAPP nodes were combined keeping the SAPP clusters head-node as the controlling head node for the newly formed Tau-Ceti cluster. This system can now, through specific hardware requests in the TORQUE job scheduler, perform the specialised roles it performed before and has a substantial amount of cores for general parallelisation.
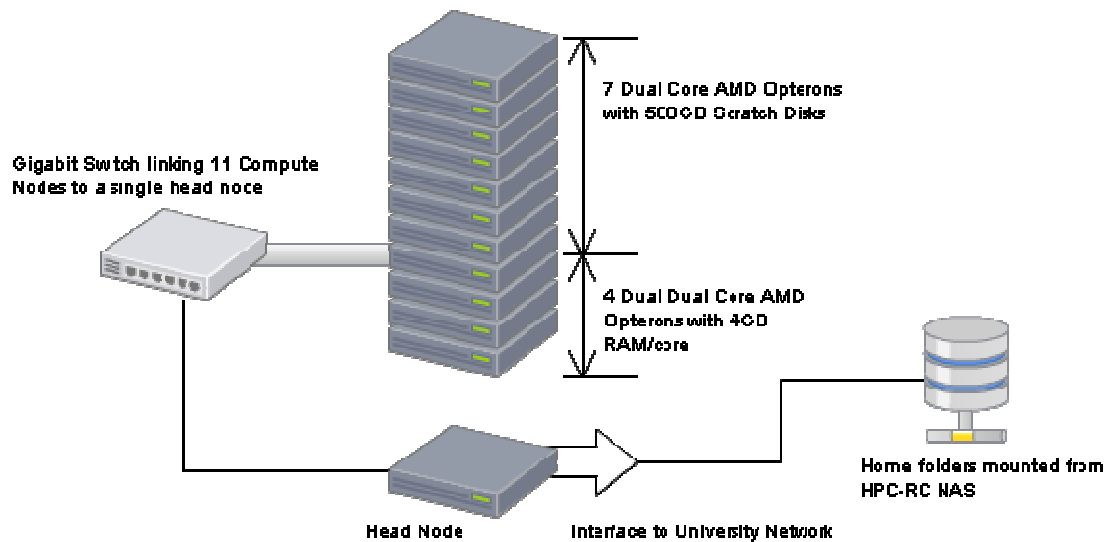
Figure 19: Tau-Ceti Cluster Architecture

There are three different configurations within this system which are as follows

Node Type	Resource	Statistic
	Total Systems	12
	Total Cores	34
	Processing Cores	30
	Service Cores	4
Head Node	Processor	AMD Opteron 2.0Ghz 4C 2-Socket 2U profile
	RAM	2 x 2GB 800Mhz
	HDD	2xSeagate Barracuda 80GB 7200RPM SATA-II RAID-1 Configuration
	Network	2 x Broadcom 1000
SAPP Node	Processor	AMD Opteron 2.0Ghz 4C 1U profile
	RAM	8 x 2GB 800Mhz
	HDD	Seagate Barracuda 80GB 7200RPM SATA
	Network	2 x Broadcom 1000
ASIM Node	Processor	AMD Opteron 2.3Ghz 1C 2 Socket 2U profile
	RAM	2 x 1GB 800Mhz
	HDD	Seagate Barracuda 40GB 7200RPM IDE Seagate Barracuda 500GB 10000RPM IDE
	Network	2 x Broadcom 1000

Figure 20: Table outlining Tau-Ceti Hardware Configuration

Chapter 5.5: QGG Clusters Software Stack

To create a uniform user/software environment across the Queensgate Grid each cluster is configured with some fixed software and services that provide for most user needs but do not affect the systems performance by eating much needed resources. The cluster head-node software stack is as follows:

Layer 0:   **Operating System and Base Services**

The operating system on the head-nodes is CENTOS 5.4 with Linux kernel ver. 2.6.18-164-el5. In the grid environment the entire home folder is mounted over the network. The services being run are mySQL (database management system), Apache (web-services), Postfix (local mail server), SSH (secure remote connection daemon), RSYNC over SSH (backup system).

Layer 1:   **Message Passing Interface**

To allow for the sharing of resources between the head and compute nodes a Message Passing Interface (MPI) is required to work over the TCP/IP gigabit inter-connect. Because there are several different MPI implementations two of the most popular ones in the UK eScience community have been installed. These are OpenMPI version 1.4.1 and two versions of MPICH 1.2 and 2.0.

65

Layer 2:   **User Management**

A HPC-RC coded script that adds users, generates passwords, creates SSH keys for use within the clusters and to access the head-nodes from the outside network is deployed for admin use. The program also emails all the details using the POSTFIX server as soon as it creates the account. The script is also automated to sync user names and passwords and group IDs across all the clusters in the network.

Layer 3:   **Cluster Middleware**

The systems are configured with OSCAR 5.1b2 as the middleware to manage the system. The services gained through this are C-3 Tools (Cluster management tools, e.g.: sys reboot, shutdown, install, execute), TORQUE Job Queuing System (A batch processing system based on openPBS), MAUI (Scheduling software to create rules for TORQUE), GANGLIA (Graphical Monitoring Tool for Administration), JOB MONARCH (Graphical Job and Queue Monitoring Tool for Users).

Layer 4:   **Applications**

A plethora of software from four to five disciplines is deployed across the different clusters. More details on the Applications can be found in Chapter 10.1: Current Software Deployment

Chapter 6: Cluster Statistics

Chapter 6.1: Power Performance

An important part of making a feasible system for the University is to ensure that the systems are not power (i.e. electrical) hungry. In the modern eco-conscience era, all organisations are striving to reduce their carbon foot print and are local for eco-friendly equipment to replace older equipment. Similarly the QGG and its resulting systems should try to be as energy efficient as possible so that the University can meet its targets of carbon neutrality.

Beowulf Clusters are typically more power hungry than server class machines as they come with larger PSU, because commodity-off-the-shelf (COTS) machines need to cater for extra peripherals, graphic cards, multiple hard drives etc. Also in server-class systems, processors are more densely packed with motherboards have multiple sockets for processors and processors hold multiple cores.

Intel and AMD have improved their power consumption greatly. The 42 nm Technology and the on-Chip parallelism and hyper-threading have improved the power to performance ratio even though clock speeds have decreased. These improvements can be clearly seen by the statistics given below. The systems were given max loads with full processor and memory usage as well as network and disk writes while the measurements were taken over the course of a week and averaged out. The systems in question here are the test-bed cluster, the ASIM cluster (Dual Single Core Opterons), and the first generation Eridani Cluster (with just the 2.33Ghz Core2Quads).

Testbed Cluster:	1x2.8 GHz HT P4+ 16x2.4GHz P4 + Network Switch
	17 Cores
	Manufacture Date: 2002
	7A max current
	1.6KWatt max power consumption
	0.41A/core
	94.12W/core
ASIM Cluster:	1x2.7Ghz Core2Quad + 7xAMD 2.3GHz + Network Switch
	18 Cores
	Manufacture Date: 2005
	8A max current
	1.6KWatt max power consumption
	0.44A/core
	88.89W/core
Eridani Cluster:	19x2.33Ghz Core2Quad + Network Switch
	76 Cores
	Manufacture Date 2008
	13A max current
	2.4KWatt max power consumption
	0.17A/core
	31.5W/core

Figure 22: Table Showing Power Consumption with regards to # of Cores

As the figures above show, the newer Intel system is almost 3 times more energy efficient than its predecessor and also trumps the five year old AMD processors. While the AMD and the earlier Intel share the same power consumption per core, in Chapter 6.2: LINPACK Performance it will be seen that

the performance difference of the AMD makes up for the comparatively higher power consumption[5].

Chapter 6.2: LINPACK Performance

"LINPACK is a collection of Fortran subroutines that analyze and solve linear equations and linear least-squares problems. The package solves linear systems whose matrices are general, banded, symmetric indefinite, symmetric positive definite, triangular, and tridiagonal square. In addition, the package computes the QR and singular value decompositions of rectangular matrices and applies them to least-squares problems. LINPACK uses column-oriented algorithms to increase efficiency by preserving locality of reference." (NETLIB 2010)

Using the Basic Linear Algebra Subprogram (BLAS) libraries LINPACK execution statistics are the benchmark for non-vector based super computers. The TOP500.org list of the world's supercomputers is compiled using LINPACK data. Using the hardware configuration stated in Chapter 6.1: Power Performance LINPACK was executed on the compute nodes. The results are as follows

System	# of Cores	GFlop (max)
Testbed	16	1.5
ASIM	14	28
Eridani	32	120

Figure 23: Table Showing Cores to Gigaflop output

These figures tell an astounding tale of how in five years the advent of Hyper Threaded and On-board/On-Chip Parallelism changed the computational power factor. With fewer and slower cores the Gigaflop output increased by a factor of 9 and as seen in Chapter 6.1: Power Performance this was achieved without increasing the power consumption of the system. Most experts would

---

[5] It is an apparent increase in power as the clock speed of the processor has gone down and the cores are newer so it would be expected that the consumption would decrease. But as described in Chapter 6.2: LINPACK Performance AMD managed to increase performance without increasing power consumption.

argue that with one of the systems not possessing HT technology this test/comparison is unfair.

The table below shows the Electrical Power to Computational Power relationship of the hardware configuration defined above. The efficiency field is calculated using what the actual output is as a percentage of the theoretical Gigaflop output.

	GFlops	Efficiency	Current Max	Power Max	Power/GFlop
Testbed Cluster	1.5	~4%	7A	1.6KW	~1KW
ASIM Cluster	28	88%	8A	1.6KW	60W
Eridani Cluster	120	84%	13A	2.4KW	20W

Figure 24: Table Showing Relationship between CPU power and Electrical power

The higher efficiency of the ASIM nodes over the Eridani nodes is due to the fact that with only 8 machines communicating over a gigabit network there was less traffic on the switch and smaller communication overheads as compared to Eridani, which had to communicate between 17 nodes over the network. As every nanosecond counts, it should be mentioned that the densely packed ASIM nodes were all patched into a switch using 1.5m Ethernet cables, while the Eridani nodes were spread out and were patched using cables ranging from 1.5m-20m thus some nodes would have introduced more latency in the system.

The statistics for Eridani given above were calculated on just the 2.33Ghz cores. When the system was upgraded and LINPACK was run again the systems efficiency went down to the 75% region as the faster 2.50 GHz processors in the mix would have to wait for the slower 2.33GHz nodes to catch up. The node count also increased to 33 machines involved in the benchmarking and thus the gigabit interconnects being used for both data and instructions do become a bottle neck. After repeated optimising and testing, an 85% efficiency can be achieved if using either the 64 2.50 GHz cores or 64 2.3 GHz cores.

As a collective system with 128 processing cores the system is at best 75% efficient and has a Gigaflop output rating of approximately 240GFlops.

Chapter 6.3: User Informed Usability Analysis

With over 18,000 jobs being completed on just the Eridani cluster[6] realistic feedback could be gained from the existing users to judge whether the system is effective.

Interviewing the mechanical engineers has revealed that within the first 2 months of the cluster being available the users changed their project deliverables and started to attempt more complex problems and simulations. This sort of major change to PhD project deliverables was also undertaken by a PhD student with only 4 months of research left in his progression. The explanation for this change is that the HPC facility has enabled users to increase the quality of their research output. The course is also considering getting commercial licenses so that they can now carry out enterprise work. As the HPC resource would presumably enable more realistic real-world models for simulations. One mechanical engineer had this to say:

"The QGG has provided us with an opportunity to simulate large and complex flow problems by employing FLUENT (CFD) in parallel which was earlier impossible on single workstations."

The Department of Chemistry and Biological Sciences immediately saw the benefits of an HPC system as the bought 2 cluster and a 10Core Workstation for their users 2-3 years before engineering considered HPC. But the staff and researchers of DOCABS said they immediately felt the benefits of a centralised HPC resource and a grid connecting the various resources as it made connectivity and usability easier. Having one point of contact for technical support, resource accessibility and NGS connectivity simplifies the workflow for

---

[6] No substantial data is available for Tau-Ceti as up until July 2010 it had gone through several re-installations or was kept as 2 discrete clusters. A conservative estimate would be that, including the days when DOCABS only owned the SAPP machines, of 20000 jobs in 28 months. In the 10weeks that the system has existed in its final configuration (with a 2.5 week downtime in the middle) the system has completed over 400 jobs.

researchers and they can concentrate on their work rather than wondering why a node in a cluster has become unresponsive.

All users interviewed stressed the need of a knowledge base and training sessions on cluster usage. While the HPC-RC was good for unexpected trouble, the need for a repository of information regarding the applications and the infrastructure itself was felt.

Infrastructure wise software licenses were limiting many users and therefore the full potential of the Eridani cluster could not be experienced. The end-to-end latency and interconnects bandwidth also became a bottleneck for an embarrassingly parallel program as it would not scale well as nodes were increased. More information can be found in Chapter 9.2: DL_POLY. For this application users were resorting to the NGS where high-speed infiniband interconnects are available. Out of the pool of applications deployed on the Eridani Cluster and the QGG this was the only exception where a decrease in performance was felt on scaling.

Chapter 7: Grid Toolkit

Chapter 7.1: The QGG Grid Mechanism

After the conference in Canada, the SciNET (University of Toronto HPC Grid) system of implementing a simple SSH based grid system with a common file system appealed as a possible initial Grid system for the QGG. As all systems of the QGG are within the same intranet , Grid middleware is not necessary. The famous hour-glass shape that is the basis of the grid infrastructure is not needed for a system where all machines are trusted, maintained by the same administrators and reside behind the same firewall. With a geographically distributed network, security and data privacy are essential. Users need to ensure that the system they are connecting to be the one they are aiming for; their data is residing or passing though locations that are governed by the same privacy laws and that the transfer of data is absolutely secure. Likewise systems need to ensure that the users are who they say they are.
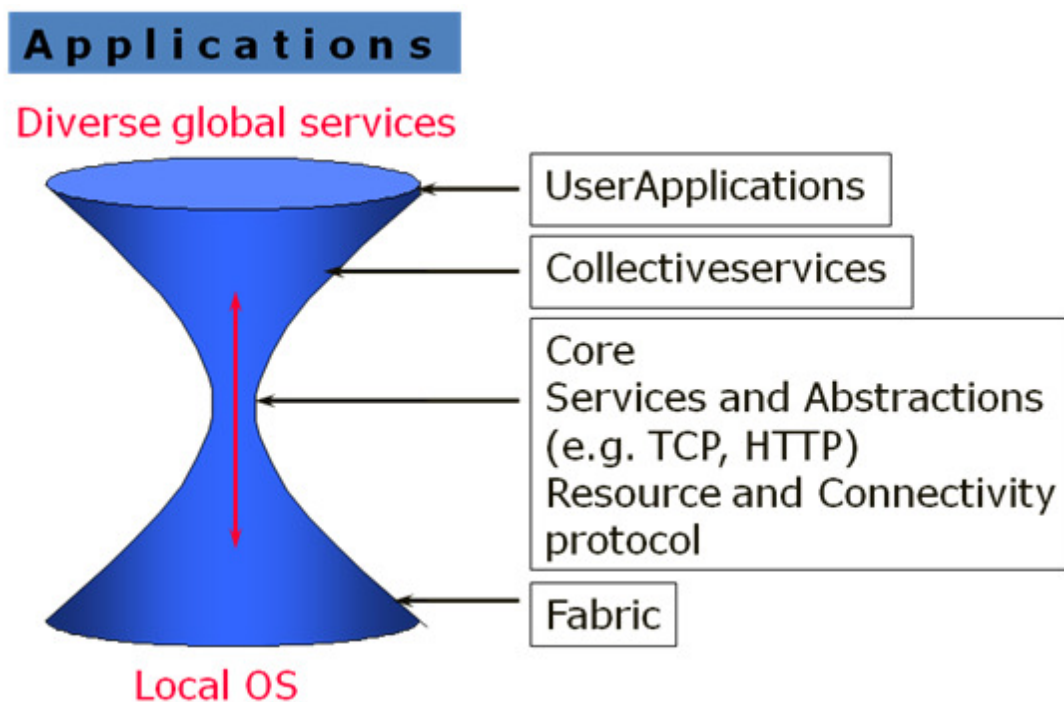
Figure 25: OGSA Hour-Glass (Foster et al. 2001)

For our local internal system, which only has virtually 2 job queues[7], no Grid Based Middleware applications are needed. The primary step to establishing an SSH based grid system is to get all clusters to mount their home folders from a common storage. In this case, the Mimosa NAS server deployed using the FreeNAS flavour of FreeBSD is the central storage device. This 16TB storage device holds all the applications and the home directories of all users on the QGG. All head and compute nodes have a line in their FSTAB file (which controls boot time drive mapping) that states that the /home (main users directory) and /apps (main applications directory) should be picked up over the network. This essentially maps the Mimosa as a folder in the clusters file system. As this is common to all systems, pathways and executables are constant for all users no matter which prompt they are at.

Network security is maintained by not allowing machines other than a central grid control node (Bellatrix) to SSH to the clusters. The Private/Public keys generated by OSCAR for one cluster behave as the "passport" for the grid system as well. To better understand this key system, the OSCAR process should be understood. When a user first logs into his system a script in the *'/etc/profile.d/'* folder executes and creates a private and public SSH-key. A user with a private key that corresponds to a public key residing on a server can log in without having to put in a password. The OSCAR script adds both the users' private and public key to the user's home folder. Now in a stock OSCAR cluster the compute nodes pick their home folder from the head-node so every machine in the cluster has the public key in its record. A user on one node can SSH seamlessly to the next as his private key also moves with him due to the common file system. Technically, if the user was to copy his private key from the server to his office box he can seamlessly SSH in from there as well. In our grid environment as the grid control node and the three head nodes share the same file system users need to only enter their password when connecting to Bellatrix and once authenticated can SSH to the resource they require.

---

[7] Though the Eridani cluster has both the Windows and Linux Job Queues they manage the same resource and the Windows job manager eliminates the need for a Grid Based Resource Broker as all nodes are homogenous and controlled by one queue.

Bash Scripts are the only thing needed to link the C-3 tools across all clusters so that cross-grid instructions can be carried out in one command. These commands include adding users to the user database and propagating that database to all systems. Passwords don't need to be propagated as the SSH-keys replace them but Usernames and Group Names are essential as they are used to manage permissions on file reads and writes.

Up to this point our system closely resembles the SciNET system. The Bellatrix machine has two interfaces. One connected within the University Network the other to a Public IP that can accept connections from outside the University. To secure our network on Bellatrix the SSH daemon only listens on the internal interface. This way no one from outside the University can try to SSH in or brute-force the system. Another Public-Private key set is generated. The Public Key is kept in the users' home folder and the private is given to the user. This system allows users to SSH into the system from within the University intranet. An SSH connection is only made if the user has the correct user-ID and private key. No login-prompt is offered on the internal network so that brute-forcing techniques can't be used.
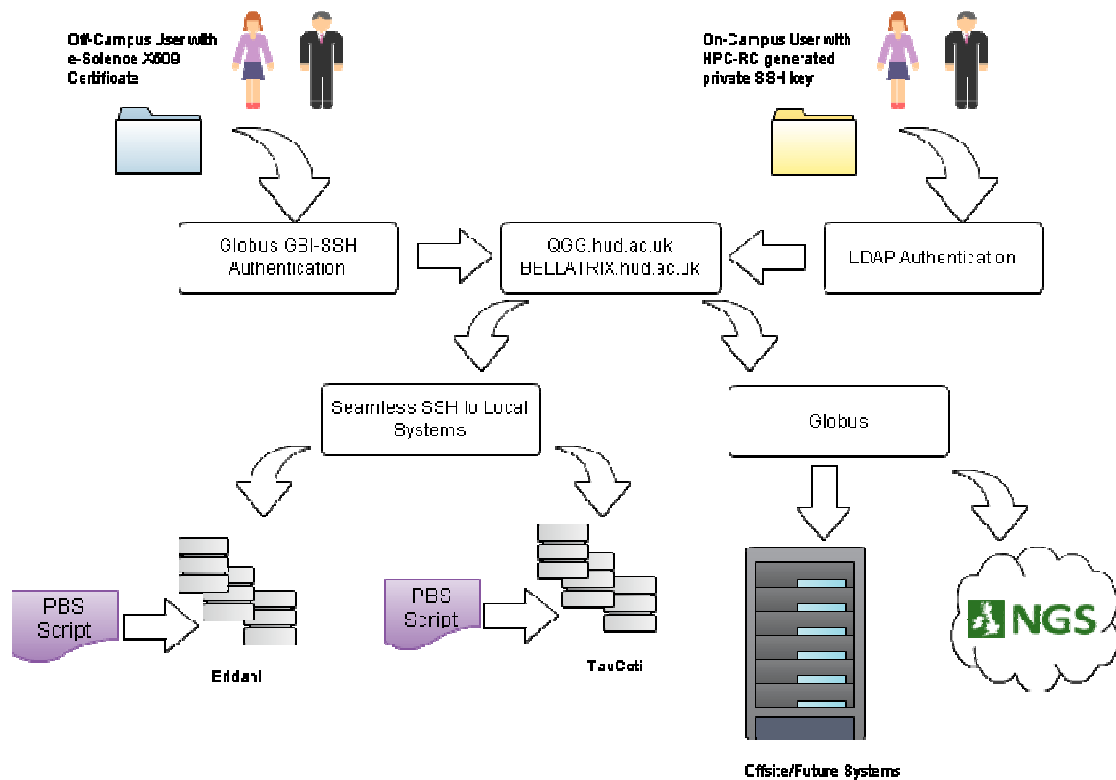
Figure 26: The QGG Workflow

As per terms agreed with Computing Services in a meeting on the 9th of September 2010, the facility of access to the grid from outside the campus is only available for Researchers, Staff and Faculty. To facilitate this and to allow the system to grow and connect to clusters in other campuses and the NGS at large, it was decided to implement the security layer of the grid middleware used by the NGS. Researchers, Staff and Faculty can be issued eScience certificates so all external access authentication can be carried out using the Globus/gLite security services which can be found in the NGS modified software stack known as VDT (Open Science Grid 2010).

The Globus installation on Bellatrix will also enable users to submit to the NGS directly from the QGG head node and will also layout the framework for future local systems or off-site systems at the other Huddersfield campuses to be linked to the existing system.
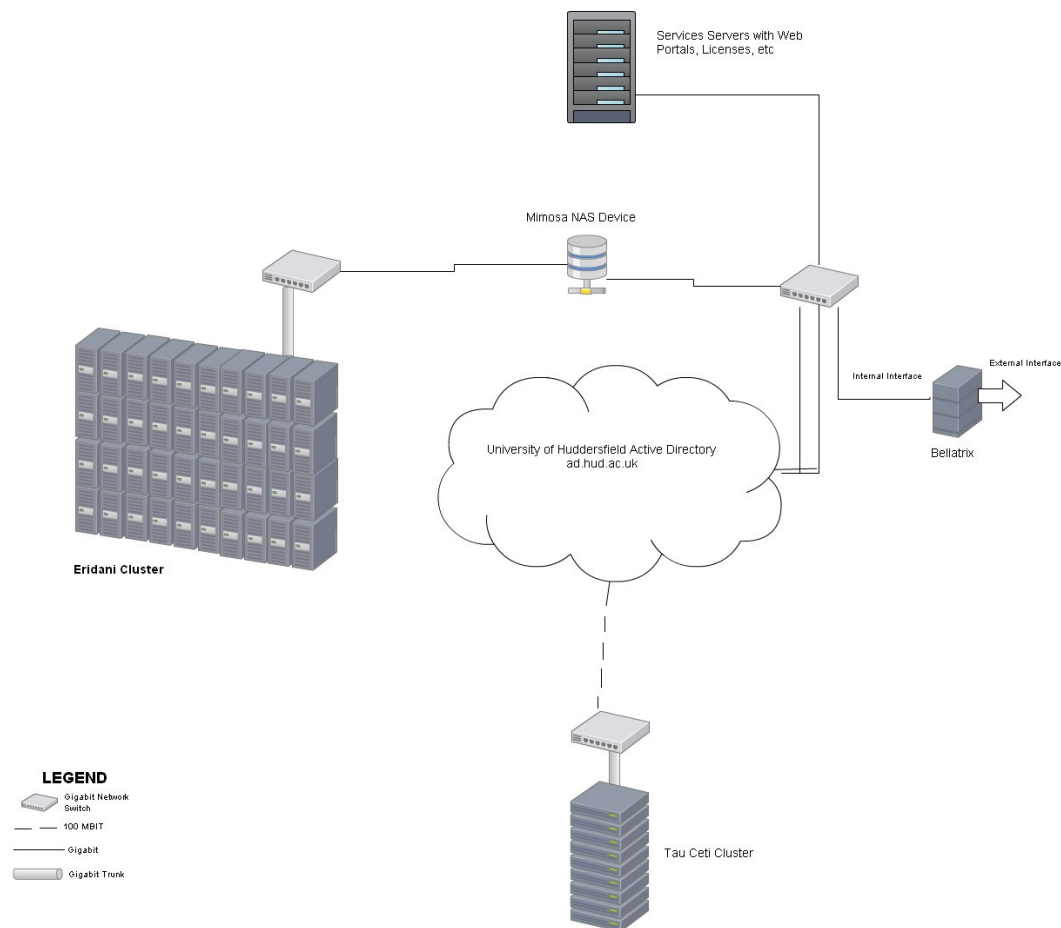
Figure 27: The QGG Architecture

Chapter 7.2: VDT Stack

The Virtual Data Toolkit (VDT) is an ensemble of distributed computing software that can be easily installed and configured. The goal of the VDT is to make it as easy as possible for users to deploy, maintain and use distributed computing software. Ideally, you just type a single-command and you can immediately access distributed resources or provide your resources to others. In reality, it is a bit more work than that, but not much. The VDT is a product of the Open Science Grid (OSG), which uses the VDT as its software distribution. OSG, and therefore the VDT, are funded by the National Science Foundation and the Department of Energy.

The NGS and UK eScience Council use the VDT with specially written scripts that automate the installation and configure them with a local flavour. There are 4 tiers to the NGS-VDT Stack:

Layer M0:   Foundation Services

Layer M1:   Service Offerings

Layer M2:   Application Services

Layer M3:   Pioneer Services

Our initial interest in this software stack is the M0: Foundation services layer which contains the User Authentication and Authorisation. The NGS states: "Services that users wish to access should be secure with clear user separation. The separate issues of user authentication, whether the user is who they claim to be, and user authorisation, whether the user is allowed to make use of the resources, are commonly linked together under the banner of Authentication and Authorisation (often abbreviated as Authn & Authz)." The system in the stack which is being deployed on the QGG is the Grid Security Infrastructure (GSI). The GSI, formerly called the Globus Security Infrastructure, is a specification for tamper-proof communications between software in a grid computing environment. Secure, authenticate-able communication is enabled using encryption techniques involving International Grid Trust Federation (IGTF) recognised X509 digital certificates (NGS 2010a).

The resource broker and user management systems can be later added when the Grid grows and many different types of resources become available.

Chapter 8: System Cost

To fully appreciate the impact of a Beowulf type cluster made of COTS machines, the cost of implementing such a system has to be assessed. The Eridani cluster which falls into the above mentioned classification was upgraded several times, and thus is the newest system even though the initial cluster was made of recycled (though still new, < 1 year old) machines. A breakdown of the costs associated with setting up a machine of these specifications is below:

Equipment	Qty.	Unit Cost	Total Cost
Core2Quad PC, 8GB RAM, 250GB Storage, GigE Lan with 3yr on-site support	37	£700	£25,900
48 Port Network Switch	1	£1100	£1,100
2TB Seagate HDD to be placed in 1 PC to serve as a NAS	8	£82	£576
Shelving to support system	1	250	£250
Misc. (Cabling, Velcro ties, PDU)		500	£500

Total Expenditure: £28,326

Figure 28: Table outlining the Hardware Cost associate with the Beowulf cluster Eridani

The table above establishes that any University can build a HPC system capable of carrying out approximately 250 Billion Instructions per Second in under £30,000. It should be noted that a medium to high-spec computer available in the market, circa 2009-2010, would cost £1,500-£3,000. If £3,000 machines were given to the researchers and staff, the university would have spent the equivalent amount in just 10 users and still would not be able to deliver the equivalent computing power. At £1,500 20 researchers/members of staff would get high-spec machines but not the same computing power as that delivered by investing in the HPC system.

The Open Source operating system and applications further helped to reduce the cost of implementing such a system. The University is also part of the Microsoft® Education Alliance and the Operating System was thus free of charge.

79

Section IV: Results, Justification and Research Outputs

Chapter 9: Case Studies

Chapter 9.1: ANSYS FLUENT CFD

The mechanical engineers at the University of Huddersfield use a package by ANSYS Systems called FLUENT. This is a Computational Fluid Dynamics (CFD) package and is primarily used by researchers and students working in the field of thermodynamics and automotive design. The research involves the modelling of the behaviour or flow of a fluid (air/water) around an object. This helps engineers and designers understand the aerodynamics of the object and can better improve the shape to get better efficiency. This can be better fuel efficiency when working in automotive design and designing cars or optimal room locations of heaters for central heating systems in applications of thermodynamics.

Using a sister tool of FLUENT known as GAMBIT, students and researchers are able to create "mesh" files (large text files that define the shape of the created object as a 3D model). This mesh file along with a script, which defines in sequence what parameters to set and what sort of simulation to run on the mesh, are submitted to the cluster for execution. Depending on the size of the mesh, a job file is created outlining how many resources are to be diverted to execute the simulation. The simulation itself an execution of a series of partial differential equations that evaluate the air flow around an object (ANSYS 2010).

Upon execution, FLUENT gives an output of the calculations it has done and for benchmarking purposes information regarding wall clock time (duration of the simulation in 'actual' time) as well as the CPU time (aggregate sum of the work done by each processor). Before initially benchmarking the system through the FLUENT documentation and the CFD online forums it was learnt that dividing the mesh files between too many cores would be detrimental and instead of speeding up the simulations, it would slow them down as inter-node communication would dominate the time (Jenssen 2001). After experimentation, it was realised that the division should not go beyond 300K mesh element per

core and in the Eridani cluster no more that 1.2M elements should be placed per core. These figures correspond with what users on the CFD support forums also recommend. Our users and testers found that between 800-900K mesh elements per core was the optimal division amount. This number was a good balance between decreasing run time and consumption of licenses.

It should also be noted that while each core can support 1M elements, a 4M elements mesh will not open on a quad-core system. This is because the 1M/core division assumes that the core in question will not be handling I/O or global aggregation of data. In a cluster environment one core behaves as a head-node and handles file and data I/O as well as handles the division and collection of data from each processor involved in the simulation. A 3-3.5M mesh elements file will load on a quad-core system but will not be able to complete all the iterations required in the simulation as handling the I/O and write backs as well as performing the calculation will overwhelm the physical memory of the system.

Below is an excerpt from a simulation of a mesh with over 7M mesh elements divided over 8 Cores:

```
Grid Size

Level   Cells    Faces     Nodes    Partitions
    0  7202290  14780986  1395896          8

 1 cell zone, 47 face zones.

------------------------------------------------------------------------------
ID    Comm.  Hostname       O.S.       PID     Mach ID HW ID   Name
------------------------------------------------------------------------------
n7    hp     node29.Queensga Linux-64  16514   1       7       Fluent Node
n6    hp     node29.Queensga Linux-64  16513   1       6       Fluent Node
n5    hp     node29.Queensga Linux-64  16512   1       5       Fluent Node
n4    hp     node29.Queensga Linux-64  16511   1       4       Fluent Node
host  net    node30.Queensga Linux-64  13638   0       3       Fluent Host
n3    hp     node30.Queensga Linux-64  13908   0       3       Fluent Node
n2    hp     node30.Queensga Linux-64  13907   0       2       Fluent Node
n1    hp     node30.Queensga Linux-64  13906   0       1       Fluent Node
n0*   hp     node30.Queensga Linux-64  13905   0       0       Fluent Node

Selected interconnect: ethernet (intra-machine comm. may use shared memory)
------------------------------------------------------------------------------

Performance Timer for 5000 iterations on 8 compute nodes
  Average wall-clock time per iteration:          13.563 sec
  Global reductions per iteration:                   135 ops
  Global reductions time per iteration:            0.000 sec (0.0%)
  Message count per iteration:                       929 messages
  Data transfer per iteration:                    34.058 MB
  LE solves per iteration:                             6 solves
  LE wall-clock time per iteration:                3.401 sec (25.1%)
  LE global solves per iteration:                      2 solves
  LE global wall-clock time per iteration:         0.006 sec (0.0%)
  AMG cycles per iteration:                            7 cycles
  Relaxation sweeps per iteration:                   436 sweeps
  Relaxation exchanges per iteration:                 68 exchanges

  Total wall-clock time:                       67815.654 sec
  Total CPU time:                             534541.800 sec
```

Figure 29: Excerpt of FLUENT Usage

The above excerpt shows some very promising results. Using the CPU-Time (aggregate duration of work done by each core) and dividing it by the wall-clock time (time taken by the simulations) will give a factor of 'speed-up' introduced by the addition of cores. A 7M element mesh divided across 8 2.5GHz cores gives a speed-up of almost 800% (or 8x faster).

$$\frac{534541.8}{67815.654} = 7.88 \cong 8x \: or \: 800\%$$

The figures above show an almost linear speed up of the simulation and these figures cannot be further improved upon as the cores have to wait for the duration of time when data I/O occurs and then during the time the mesh is divided across the cluster. This is due to the effect of Amdals Law. The above reproducible data reflects how the combination of binary division of the mesh and the optimal element division and minimal cross-network chatter can produce near perfect results. With a larger mesh size, more nodes would need to be introduced and the Gigabit Interconnect would begin to bottle-neck the speedup. If the node count is not increased and more elements are put on each node the speed-up would suffer as each core would run out of its associated memory and would keep writing to its local disk to enhance the amount of RAM available. As HDDs are slow, this would increase the simulation time.

Further tests using 2.3 Ghz cores on the same mesh result in a slight drop in speed-up as the time per iteration/calculation increases. Larger mesh files across more nodes also decrease the speed-up factor, as network overheads start to play a major role. Running the appropriate mesh across 44 2.5GHz cores (the maximum even divisions inside the 45 license limit) leads to a speed up factor of 34 times. This is an efficiency rating of approximately 77%. This number is familiar, as in the LINPACK testing across 64 2.5GHz cores the system reached an efficiency number of approximately 75%. The slightly higher number can be explained by the fact that in the LINPACK test, 16 nodes were communicating with each other while in this case FLUENT had 11 nodes communicating with each other. This similarity in number is due to the fact that the nature of calculations performed by FLUENT is similar to those performed by LINPACK. These results further validate the earlier benchmarking.

DL_POLY is a general purpose serial and parallel molecular dynamics simulation package developed at Daresbury Laboratory by W. Smith, T.R. Forester and I.T. Todorov. The original package was developed by the Molecular Simulation Group (now part of the Computational Chemistry Group, MSG) at Daresbury Laboratory funded by the Engineering and Physical Sciences Research Council (EPSRC). Later developments were also supported by the Natural Environment Research Council through the eMinerals project. The package is the property of the Central Laboratory of the Research Councils.

Two versions of DL_POLY are currently available. DL_POLY_2 is a modified version of the original DL_POLY which has been parallelised using the Replicated Data strategy and is useful for simulations of up to 30,000 atoms on 100 processors. DL_POLY_3 is a version which uses Domain Decomposition to achieve parallelism and is suitable for simulations of order 1 million atoms on 8-1024 processors. Both versions are supplied together under one DL_POLY licence. DL_POLY is supplied to individuals under an academic licence, which is free of cost to academic scientists pursuing scientific research of a non-commercial nature.

DL POLY 2 is a package of subroutines, programs and data files, designed to facilitate molecular dynamics simulations of macromolecules, polymers, ionic systems, solutions and other molecular systems on a distributed memory parallel computer. Though DL POLY 2 is designed for distributed memory parallel machines, with minimum modification the creaters have ensured that can run on the popular workstations. Scaling up a simulation from a small workstation to a massively parallel machine is therefore a useful feature of the package (STFC 2010).

To benchmark this software's performance on the Eridani cluster two models of different sizes underwent a series of simulations on the exact same node. The run times were averaged to give an idea of the effect on performance by scaling up/down. DL_POLY ver. 2.20 complied in GFORTRAN with the OpenMPI libraries is the software platform used for these tests.

The first model is a small cell of MgO with a 5x5x5 scaling of the unit cell. This small cell contains 1500 atoms and 100K steps/iterations are performed on these atoms. The simulation was scaled four times:

Run 1: 2 Cores (on 1 node)

Run 2: 4 Cores (on 1 node)

Run 3: 8 Cores (on 2 nodes)

Run 4: 12 Cores (on 3 nodes)

The graph below is a plot of Speed Up versus Run Number of the observed speed up and what the ideal linear speed up should be.
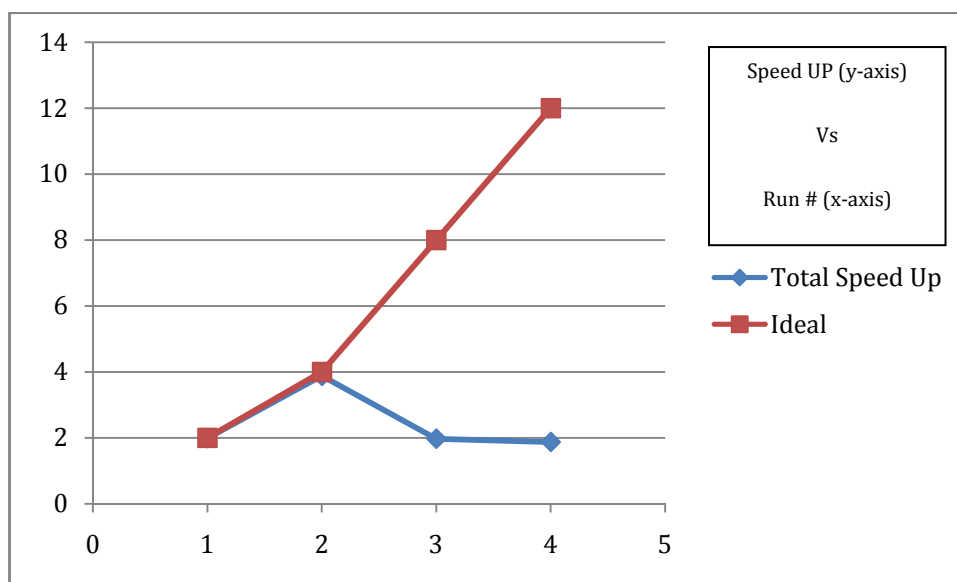
Figure 30: DL_POLY Small Cell Speed Up Graph

The graph in Figure 30: DL_POLY Small Cell Speed Up Graph shows that between Run 1 and Run 2 there is an almost linear speed up. It should be noted that both Run 1 and Run 2 are on the same system and the scale up is on the same processor. As soon as this simulation is ported over the network to one more node and then another the speed slowly declines. This is possibly because of the small number of atoms allocated to each compute node. Each node will not have enough data for calculation and will require a large number of slow global steps, requireing more data passing over the network interconnect. This

85

phenomenon would also be seen in the FLUENT analysis if less than 300K mesh elements were passed to each core.

DL_POLY comes with a useful execution summary that breaks the run time into three parts. "Type I time" is the time taken (in seconds) for the initial input read-in, output streams opened, arrays allocated, MPI initiated and the problem scattered between each core. The graph in Figure 31: DL_POLY: Small Type I Time is expected to rise as more cores and nodes are initiated. More time will be spent initialising and dividing the problem.
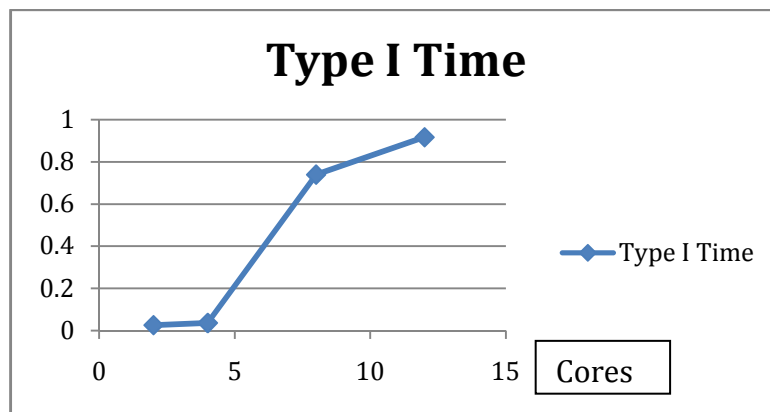
Figure 31: DL_POLY: Small Type I Time

The second graph shows the "Type II time" outputted by DL_POLY. This is the actual run time or production run of the simulations. It is this segment of the simulation that is expected to scale to the cores and decrease in time as the number of cores increases. As Figure 32: DL_POLY: Small Type II Time shows, when the cores are doubled from 2 to 4 there is almost a 50% drop in run time, i.e. for a doubling of the cores there is a corresponding halving of the run time. The third point in the graph is the simulation running across 8 cores on 2 nodes and this does not scale well: the actual run time beings to increase from here on out.

The last "Type III Time" outputted by DL_POLY is the total run time of the simulation. Since the actual simulation time (Type II) is of a much greater magnitude that the loading time (Type I) the overall Type III graph will be similar to the Type II graph. Type III data just holds the additional data of the time to wrap the simulation up, write to disk, close the MPI environment, clear environment variables and exit the program, after the main simulation ends.
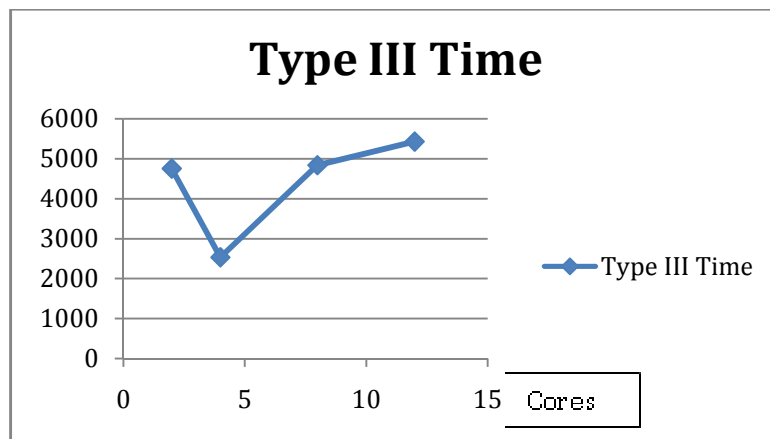
Run #	1	2	3	4
Total Cores	2	4	8	12
Type I Time	0.025	0.035	0.739	0.916
Type II Time	4748.621	2531.104	4832.121	5423.975
Type III Time	4748.673	2531.167	4832.176	5424.03
Total Speed Up	2	3.88	1.98	1.88
Ideal Speed Up	2	4	8	12

Figure 34: Statistical Data for DL_Poly Small

The second set of simulations run on this system is a 10x10x10 scaling of the unit MgO cell with the whole structure containing 12000 atoms. The 4th run in this set is over 16 Cores instead of 12 so as to provide binary division of the dataset. This time it can be seen that in the third run when 8 cores in 2 nodes are used the total speed up almost mirrors the ideal linear speed up. These figures are similar to those in the FLUENT case study. Across 2 nodes the binary division of data and the reduced global steps over the network are highly optimised and therefore the system is able to give an almost ideal response.
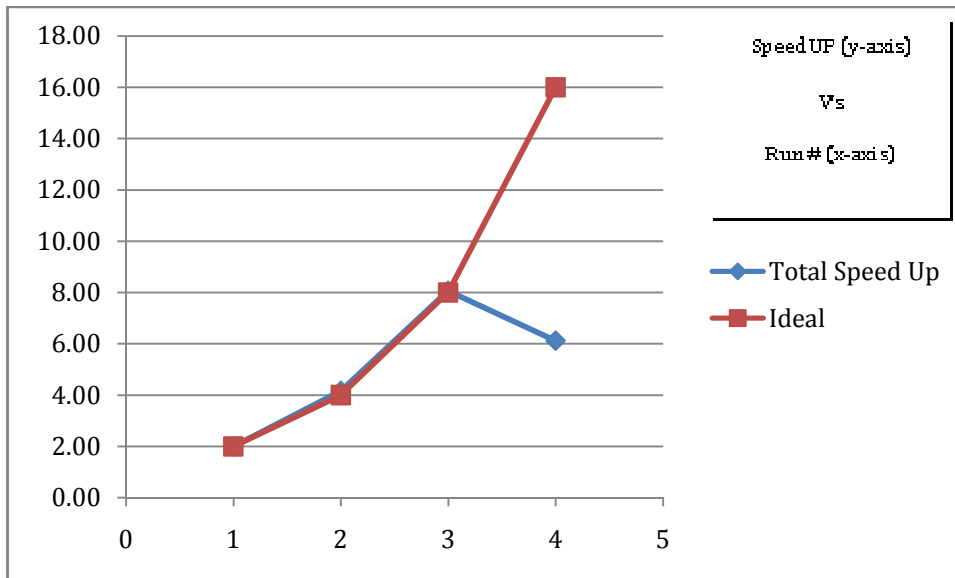
Figure 35: DL_POLY Large Speed Up Graph

Dividing the data beyond 2 nodes once again causes a drop in the speed up. Run times at 16 cores shows some speed up, but this speed up is not close to

the ideal. Beyond 16 Cores the run time begins to increase at a faster rate (not shown in graphs). The statistical data and the three run time graphs are as follows:

Run #	1	2	3	4
Total Cores	2	4	8	16
Type I Time	0.184	0.257	5.769	11.283
Type II Time	9442.154	4367.753	2320.286	3027.289
Type III Time	9442.553	4368.073	2320.608	3027.618
Total Speed Up	2.00	4.16	8.07	6.12
Ideal Speed Up	2	4	8	16

Figure 36: Statistical Data for DL_POLY Large
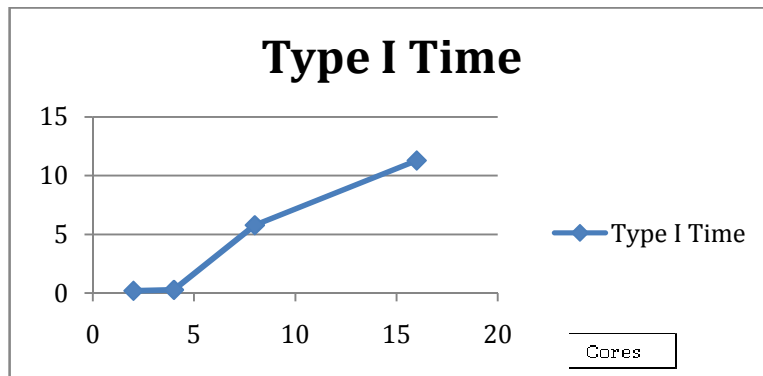
Figure 37: DL_POLY Large Type I Time
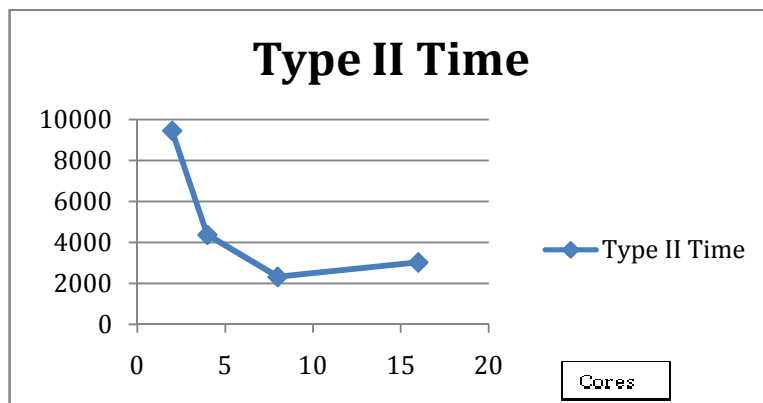
Figure 38: DL_POLY Large Type II Time
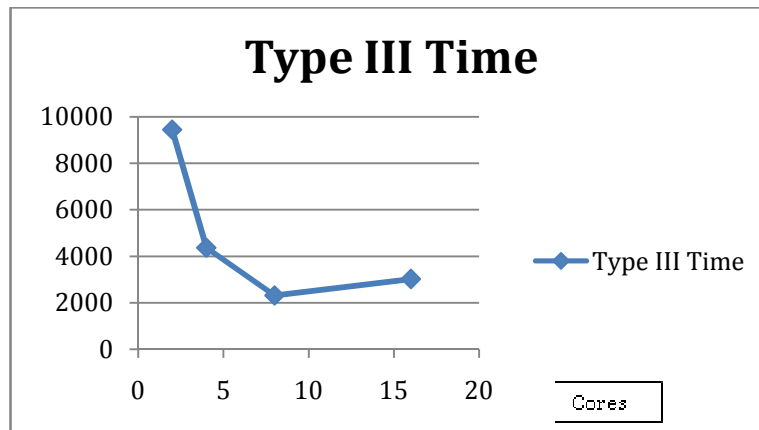
Figure 39: DL_POLY Large Type III Time

This glaring difference in scalability between DL_POLY and FLUENT can be explained by the fact that in FLUENT simulations there are fewer global steps and therefore the network overheads don't come into play[8]. Due to the amount of data generated by DL_POLY there are more frequent write-to-disk operations than FLUENT and this repeated 'attack' on the NAS device slows the overall system.

Chapter 9.3: BLENDER

Blender is an integrated application that is used in the creation of a broad range of 2D and 3D content and was lunched August 1994. Blender provides a broad spectrum of modelling, texturing, lighting, animation and video post-processing functionality in one package. Blender provides cross-platform interoperability, extensibility, an incredibly small footprint, and a tightly integrated workflow. Through its open architecture, Blender comes with a large community support system and knowledge base.

Aimed worldwide at media professionals and artists, Blender can be used to create 3D visualizations, stills as well as broadcast and cinema quality videos, while the incorporation of a real-time 3D engine allows for the creation of 3D interactive content for stand-alone playback (Blender 3D 2010).

---

[8] Based on simulations on the QGG where the availability of the licenses is the ceiling in scalability.

Blender like other video and image rendering software can be described as embarrassingly parallel. As each frame can be defined independently of the previous and next frame the animation can potentially be divided between a large numbers of nodes and thus increase the speed up and throughput of the render. Rendering takes a large amount of memory and so most render farms need high-speed disks and network attached storage so that the minimum amount of tools and libraries have to be loaded in memory during a render. The tools and libraries are picked off the storage devices as and when is needed.

A particularly detailed frame with many hi-level textures will take time to render as the required files need to be called up over the network to make the final result. This process cannot be sped up but to decrease the overall render time a series of frames in an animation can be divided across several cores and machines. A speed up can be seen in just 4 minutes on a high detailed render in an NTSC format. At almost 30 frames per second, four minutes equates to 7200 individual renders on the corresponding frames.

Blender unfortunately doesn't have an automated division method but relies on the user providing a text file which states the division of frames. Autodesk's 3D Studio MAX, the application of choice in the School of Arts, Design and Architecture, comes with it its own middleware known as Back-Burner which handles all the division of the frames across nodes. The parallel version of Back-Burner was deployed during the writing of this report and thus performance statistics of this tool are not available.

```
1       1800    node31.Eridani
1801    3600    node30.Eridani
3601    5400    node29.Eridani
5401    7200    node28.Eridani
```
Figure 40: Sample Frame Division File for Blender

Chapter 10: Applications, Performance, Usage statistics

Chapter 10.1: Current Software Deployment

With the Queensgate Grid up and running, a plethora of software has been deployed to meet the immediate needs of the research community. While licensing restrictions still prevent the entire system from participating in the simulations, the system has surpassed the expectations of the researchers using it and has changed the way many of the PhD researchers approach the simulation aspect of their research.

There have been cases where enterprise work has also been undertaken only because with the new HPC resource it was now possible to provide the level of precision in simulations that is required by industry.

In engineering, Computational Fluid Dynamics problems make the bulk of the systems usage. In chemistry long-running (greater than a month each) simulations in Force Field Molecular Dynamics using NWChem takes a bulk of the system time and keeps nodes booked and busy. 3D Studio MAX as a rendering tool on the Windows© platform is the dominant software from Arts and Design.

Below is a list of software that has been deployed and tested on the Queensgate Grid. While this list is in no way exhaustive it is a list of software that users can get support for from the HPC-RC and find extensive information and tutorials for on the HPC-RC knowledge base (Chapter 12: The Knowledge Base).

Computing & Engineering

- FLUENT – Computational Fluid Dynamics (CFD)
- Abaqus – Finite Element Analysis (FEA)
- MATLAB – Numerical Computing Environment
- COMSOL – Finite Element Analysis (FEA)
- OPERA 3D – Finite Element Analysis for Electromagnetic

Applied Sciences

- DL_POLY – Force Field Molecular Dynamics (FFMD)
- GAMESS-UK – General Atomic and Molecular Electronic Structure System -UK
- NWChem – Force Field Molecular Dynamics (FFMD)
- Amber – Computational Molecular Dynamics (CMD)
- Metadise – Minimum Energy Techniques Applied to Defects, Interfaces and Surface Energies
- Gulp – General Utility Lattice Program
- LAMMPS – Large-scale Atomic/Molecular Massively Parallel Simulator

Arts, Design and Architecture

- Maya – 3D Modeller and Renderer
- 3DsMAX – Autodesk 3D Modeller and Renderer
- Blender – Open Source 3D Modeller and Renderer

Chapter 10.2: Usage Statistics

An open source accounting tool is freely available and can integrate with the job management software TORQUE that is deployed on the cluster. PBS Accounting is a tool which parses the logs generated by the TORQUE queuing system known as Open Portable Batch System (openPBS). The following table is an excerpt of the usage on the Eridani cluster of thirteen users who have been active users since 4th April 2010.

*** Portable Batch System accounting statistics ***
Server Name: Eridani.QGG.hud.ac.uk

*** PBS Per-User Usage Report ***				
User	Group	#Jobs	Wall-Hours	CPU-Hours
oscartst	oscartst	15	0.0142	0.2267
sappdjc2	sappdjc2	11720	2799.3819	3562.04
sappgn2	sappgn2	72	685.0306	685.0306
sapppie	sapppie	54	6100.7806	6105.9214
sengbct	sengbct	54	566.3397	1640.7853
Senggc	senggc	240	6497.0597	6498.7417
Sengik	sengik	55	52.0328	106.1031
sengjoo	sengjoo	78	844.305	967.0097
sengrm	sengrm	6	1.0986	1.0986
sengvm	sengvm	996	6894.2447	13944.1661
sliang	sliang	1100	577.0203	583.3894
u0560509	u0560509	40	33.3558	70.4858
u0651533	u0651533	7	72.7128	145.4256

Total Jobs	14437
Total Wall-Hours	25123.3767
Total CPU-Hours	**34310.424**
Total CPU-Months	**47.65**
Cost @ £0.13	£ 4460.35
Uptime (in Days)	93

Figure 41: Table showing the activity of 13 users from 4-Apr-2010 on the Eridani Cluster

These figures paint a very important picture. If there was any doubt regarding whether an HPC system was effective or whether it was a requirement

for a young research institute, the fact that 13 users can complete fourteen and a half thousand jobs and almost 4 years of computing[9] in just 93 days should prove otherwise. The breakdown of the jobs also gives an interesting statistic. Some users need to perform many small simulations which while not being computationally intensive are impossible to schedule on an ordinary desktop. Looking at user 'sappdjc2' node-hours vs. job ratio, it can be assessed that each job ran an average 18 minutes each, but a mammoth 11,720 jobs were carried out. User 'sapppie' ran only 54 jobs, but averaged out each job ran for 113 hours (almost 5 days).

There is evidence that the lack of licenses and lack of awareness of the availability of this system has lead to under utilisation. The fact that these 93 days also correspond to the University's examination and holidays cycle could also explain the under utilisation. In 93 days of uptime, the system only saw 18% utilisation. This calculation was carried out using the total CPU-Hours executed by the system used as a percentage of the total theoretically available CPU hours. Keeping all the upgrades in mind, this system could have *theoretically* performed 213,504 CPU-hours (711 years) of computing in the same uptime. While obviously this figure is a theoretical ideal, as data reads and writes do not factor into the CPU-hour calculation, this ideal number forms a basis to calculate percentage usage.

With an increase in licensing, to allow for more that 3 FLUENT jobs and 1 Abaqus job to run at a time, and by expanding awareness of the existence of this system this utilisation will increase. For a true reading of utilisation the usage hours should be evaluated over the course of a year with less that 1% downtime. A proper evaluation should be carried out in 2013 and optimistically the system should show a 50% usage.

---

[9] Based on the standard Researcher/Staff Desktop Configuration circa 2009-2010, (P4 HT 3.0 Ghz 1GB RAM).

A major aim of setting up the High Performance Computing Resource Centre and the High Performance Computing Research group was to provide a platform for research and development in parallel computer architectures distributed systems and the workflows on such systems. The following projects were co-supervised by the author.

The Dual-Boot system described above was the result of an Undergraduate Final Year project to develop cluster tools. This project surpassed expectations and was accepted as a paper presentation at the UK eScience AHM 2010.

Another project undertaken by a Masters student was entitled "Investigation of the requirements of Highly Available High Performance Computing: Upgrading the QGG from HPC to HA". The project abstract states:

"This project endeavours to find the best possible solution to providing a highly available solution for high performance computing. The results discovered upon completion can be used to further explore the properties and functionality of clusters and high availability system. The project will be carried out in three phases; the first is an in-depth research of the composition of a HA-clusters and services and how its features and parts affect the clusters operation as a whole. The second phase will be to build and deploy a small Beowulf type HPC cluster, and then upgrading and re-deploying as an HA-cluster.

For the project to be implemented, two middleware software packages will be used. Using Oscar5.1 beta2 will enable the first phase to be achieved. HA-Oscar2.0 will then help in upgrading the first phase in order to achieve the second phase which will then lead to the completion of this projects main aim.

Webmin application will be then used to setup the availability environment. Also, this application will be then used to monitor the cluster functionalities. Moreover, it will be used to monitor the system

when a test will be run. The test will be to force the server node to fail and the standby to take over.

The third phase will be to provide a solution for the current floating NAS device known as Mimosa. Downtime for Mimosa would mean a downtime for the whole grid. A feasibly system will be presented for replication and failover of the Mimosa server to make the whole Queensgate Grid a highly available system."

<div align="right">

Project Undertaken By: M.M.A. El-Desouki

</div>

This project has been able to provide ideas for increasing the reliability of the QGG. The system proposed by the project is being considered to make the NAS device highly available.

To assist ANSYS FLUENT users who are accustomed to the Windows© environment, another project was undertaken to provide a web based workflow. This workflow would allow users to upload their design files (made on their desktop computers), set the FLUENT parameters for the simulation through a GUI interface, specify the computational requirement of the jobs and finally upload it to the grid for execution. The project abstract states:

"The aim of this project is to provide a useful, effective and user-friendly tool for University of Huddersfield FLUENT users. FLUENT users in the University need to use the calculation power of the clusters on the QGG to further their projects and research. For most users using a Linux command line interface is a daunting task and students are not taught the scripting language that FLUENT comes with.

Using web technologies, this project will provide an alternative way to submit and calculate jobs to the cluster without using the FLUENT GUI or manually scripting job and scheme files. By filling different forms and creating or uploading files, general FLUENT users will be able to submit jobs to the Eridani cluster on the QGG. Users who like to script their job files manually (for more control) but do not want to use the Linux interface on Eridani can just upload their files to this portal and

97

the system will submit and execute the files automatically. These web pages will be accessible from a centralized web server that would use Apache and MySQL server via the local network."

Project Undertaken By: Quentin Hossatte
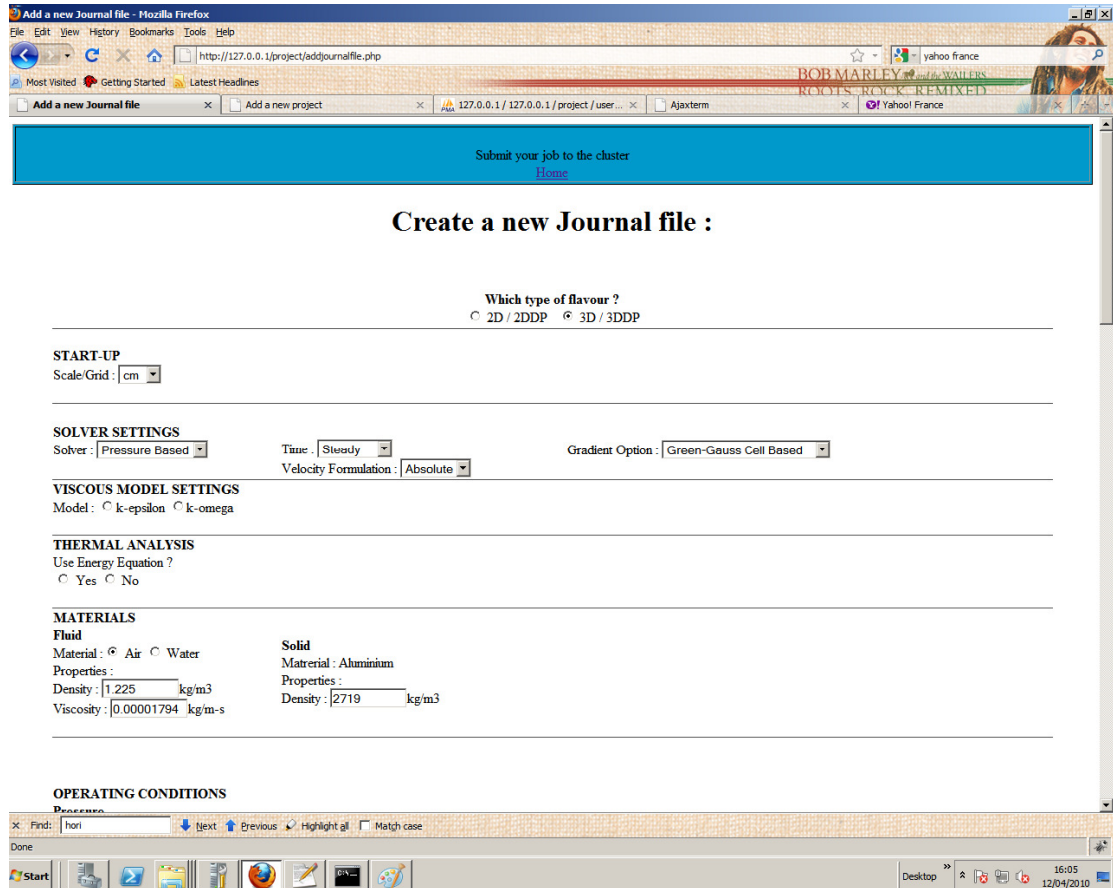
Figure 42: FLUENT Workflow Management System on the QGG

This project too has been well received in the Mechanical Engineering Department and now in collaboration with them this project is growing to provide more functionality and control to the users[10].

---

[10] The initial project just included the popular simulation options and kept the rest of the values set to default.

Section V: Business Model and Sustainability

Chapter 11: System Deployment & Recommended Organizational Structure

Chapter 11.1: QGG Usage Policies

The Queensgate Grid with its component systems like the Eridani and Tau-Ceti clusters, the various servers, the software licenses and the locations where the resources are housed have only been combined together because of the generosity of the various departments and faculty members. Due to this, the HPC-RC has an implied commitment to its stakeholders and to deliver this a standardised process needs to be adopted for all interaction, to ensure uniformity.

Access Policies

The QGG is primarily a research tool for the University of Huddersfield, but is available for all members of staff, researchers and students who wish to use it for academic purposes. Researchers and Faculty members who wish to undertake enterprise work are allowed to do so as per University regulations but the provisos stated in Chapter 11.3: Sustainability should be adhered to.

The Queensgate Grid has been made available to all machines on the Universities wired intranet. As per university policy the "work-from-home" option is only available for staff and researchers and so external access to the QGG is not open to taught students.

Registration Policies

The eScience Council has very strict registration policies for users and this means that the local University Registration Authorities (RA) are required to collect and keep on file information about the applicant. As Huddersfield DOCABS is a recognised RA, a photocopy of the staff/researchers University Identity card along with a print out of the issued key is required to be kept on file. Further to this, the HPC-RC has introduced a form to keep track of the users for justification and revenue tracking. The eScience council only issues

certificates to researchers or academics and these regulations tie in with the University policy of students using local systems only from the wired intranet.

For researchers prior permission from their faculty is required to ensure that the resource is only used when it is necessary. This ensures that in an organisation of more that 30,000 people, there is a chain of identification, thus guaranteeing that the user signing up is actually a member of the University of Huddersfield family.

Software Policies

Part of the registration form asks the user to disclose the title of the project and the application the user intends to use as the licensing for some software does not allow for enterprise work. Some applications have license holders who are not involved with the HPC-RC and have only made the software available to the cluster for their own simulations and thus priority over the licenses is theirs. To counter act this, if a user discloses that he/she is using software X but his/her faculty advisor is not the license holder, then the user must get the approval of the license holder.

Resource Usage Policy

Currently, the QGG operates with a general level of understanding with all users and there is a fair usage policy in place. If a user exceeds the current quota of 50GB storage, a warning email goes to the user for 4 days. After the fourth day the QGG administrators receive an alert and then it is between the administrators and the user as to how to proceed further. A queue limit is in place where jobs running in excess of 1 week are limited to 40% of the cores in the system. Restrictions are also in place for certain software to prevent too many concurrent jobs or too much license usage by the running jobs. This is to ensure that licenses are still available to use the software on campus.

Chapter 11.2: Day-to-Day Management

Currently the High Performance Computing Resource Centre is staffed with volunteer researchers and two faculty members from the HPC Research Group. One Faculty member is from the University of Huddersfield, Department

of Chemistry and Biological Sciences and is thus the eScience RA manager for the university. This manager will usually have 1 to 2 RA Operators located in different schools or in offsite locations (e.g. the other 2 campuses of the University of Huddersfield) to handle day-to-day operations there. One of the RA operators will be in the HPC-RC as the aim is to make the HPC-RC the one-stop shop for anyone interested in high performance computing.

The second faculty member is the HPC-RC Manager and liaises between the HPC-RC and the University. All bids or requests for funding, development and staffing will go through and be led by the HPC-RC manager. Below the HPC-RC manager is the HPC-RC Senior Administrator (he/she can also be the RA Operator of the Centre). This administrator, who can be a post doctoral researcher or a dedicated member of staff, is in charge of the day-to-day activities of the centre. User signups, deployment of new software, deployment of new technologies, installation of hardware, maintenance of the resources and maintaining the online presence of the HPC-RC falls under his/her 's purview. It is recommended for the long term sustainability of the HPC-RC that this position of Senior Administrator becomes a paid position to ensure that there is continuity of staffing and to give the staff running the HPC-RC some legitimacy when enacting policies.

Typically the Senior Administrator will have the support of research assistants or volunteer post-graduate researchers. Any offsite hardware based locations that are integrated into the QGG will require an Administrator to manage those systems who will be answerable to the Senior Administrator at the HPC-RC.

As the HPC-RC cannot exist in isolation within the University of Huddersfield IT infrastructure, a dedicated liaison officer and technician from the Central Computing Services is need to make sure all the goals of the HPC-RC are met and the development of the HPC-RC closely follows the University's own IT development roadmap.

Figure 43: HPC-RC Organisational Chart shows the current organisational chart the HPC-RC follows.
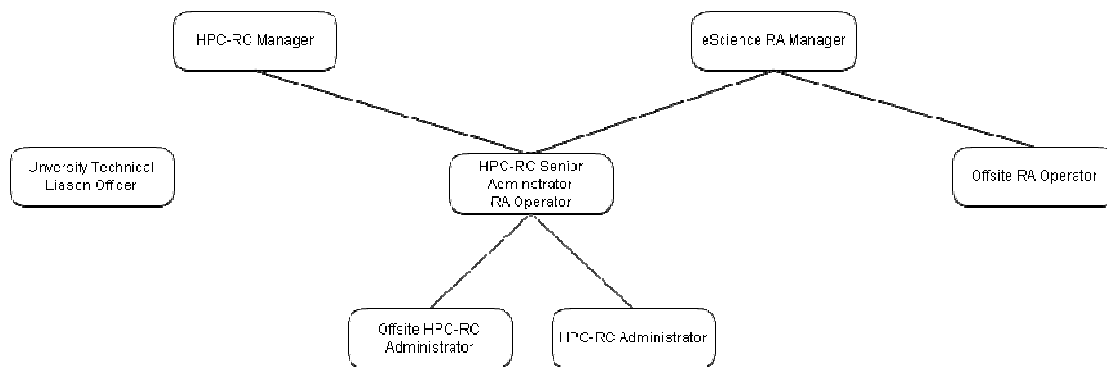
Figure 43: HPC-RC Organisational Chart

Chapter 11.3: Sustainability

With computer hardware it is always a case of playing catch up as the technology develops very quickly. In the case of Eridani, between the time the 2.5 GHz nodes were ordered, paid for and delivered Intel had already released its 'i' series of processors which ran cooler, faster and consumed less energy. The 2.5 Core2Quad systems were discontinued and out of date even before the HPC-RC took delivery for them. This is why a sustainability model is required to keep the QGG alive and to ensure it remains a vital resource.

All system in the QGG should undergo a full replacement two years after the writing of this report. Without going into absolutes, based on the development of desktop computers over the last decade, it appears that desktop computers will not be at the same speed as the Eridani cluster. Researchers at the University of Huddersfield are usually issued 2-4 year old retired-lab machines so if the 50 machines that make up Eridani are retired after two years it will give the engineering department (who have made the initial investment) 50 medium to medium-low spec machines for staff use. The technology in two years will surely be smaller, more powerful and most of all more energy efficient. This last point is key, as the one-time cost of new machines is easily made up by the reduced operating costs. As shown previously, the power consumed by the Test-bed cluster was very high and the HPC-RC was able to operate more cores in the same power rating just by using new machines. Further to this developments in Graphic Processing Units (GPU) have led to desktop-based

clusters equipped with these tools achieving top500 status in the super computing world.

The question of who bears the cost of the upgrade remains unanswered. It should be the aim of the HPC-RG and HPC-RC to garner support from the academic community to bolster funding proposals which can help support the system. Any project on the QGG that can be classified as enterprise work (commercial) or externally funded research projects should allocate some of their revenue/budget for the purposes of the development of the HPC-RC.

Chapter 11.4: Departmental/Schools Recommended Policy Changes

To increase the productivity and efficiency of the QGG, the following Departmental and School level changes are recommended:

- Before software is purchased for research and enterprise work in any school the advice of the HPC-RC should be taken to ensure that if the application is parallelisable, then the licenses do not become a limiting factor later in the project.
- A central repository and a centralised method for purchasing software should be created so that only the best and uniform price is paid. There is also evidence that in the federated system, not just within Schools but even within the departments in the school 'double-buying' of software is taking place.
- Research Groups should first assess whether the HPC centre can provide a computing solution before investing in heavy duty desktops for researchers only to find that when the simulations begin to get complex, the desktop is unable to deliver.
- Schools should encourage cross discipline collaboration, as it would help all parties involved and save the University money by not out sourcing. (e.g. the Humanities developing programs with Computing help)

Chapter 12: The Knowledge Base

As the number of users working on the Queensgate Grid increases, a knowledge base needs to be compiled so that users can easily transition from working on a personal desktop to large scale computing. There also needs to be a system where users can get help from administrators. These targets have in part been achieved in several different ways.

Chapter 12.1: Web presence

The High Performance Research group has a generic web page nested at hud.ac.uk/research, which is maintained by the research office and has information about all the members of the research group and their current research interests and projects. Proposed projects are also advertised on this page to attract further researchers to join the University and the Research Group.

A local intranet site has been created with the address hpc.hud.ac.uk. This website is the base platform for the HPC Centre Staff. The website contains information about the Queensgate Grid, the team, contact information, scheduled updates etc. Several web based applications are also integrated into this website to make it a one-stop-shop for information on joining, connecting to and using the Queensgate Grid.

Figure 44: The HPC website on the University Intranet

Using MediaWiki, a popular knowledge based content management system, several tutorials on the basics of how to use various applications and functions on the grid are uploaded. By giving write permissions to certain researchers or members of staff these pages can be maintained by experts and it is hoped that the wiki will contain specific knowledge on certain applications. Located at hpc.hud.ac.uk/wiki this site takes its inspiration from the NGS wiki and will contain similar tutorials on HPC usage along with aides to help the move from the local grid to the national grid.

Figure 45: The QGG Wiki Site providing users with "how to" documents and tutorials

To tackle the problem of users needing to contact administrators an online helpdesk ticket system has been setup. Using the Freeware package eTicket ver. 1.7.3 from eTicket Support a website has been created at hpc.hud.ac.uk/helpdesk which gives an easy and intuitive interface made in HTML and PHP that allows users to create tickets and threads with issues and feedback which future administrators can use to systematically tackle problems and keep a balanced work load between them. A history is also automatically maintained of faults that may have emerged in the system and this is kept as a knowledge base.
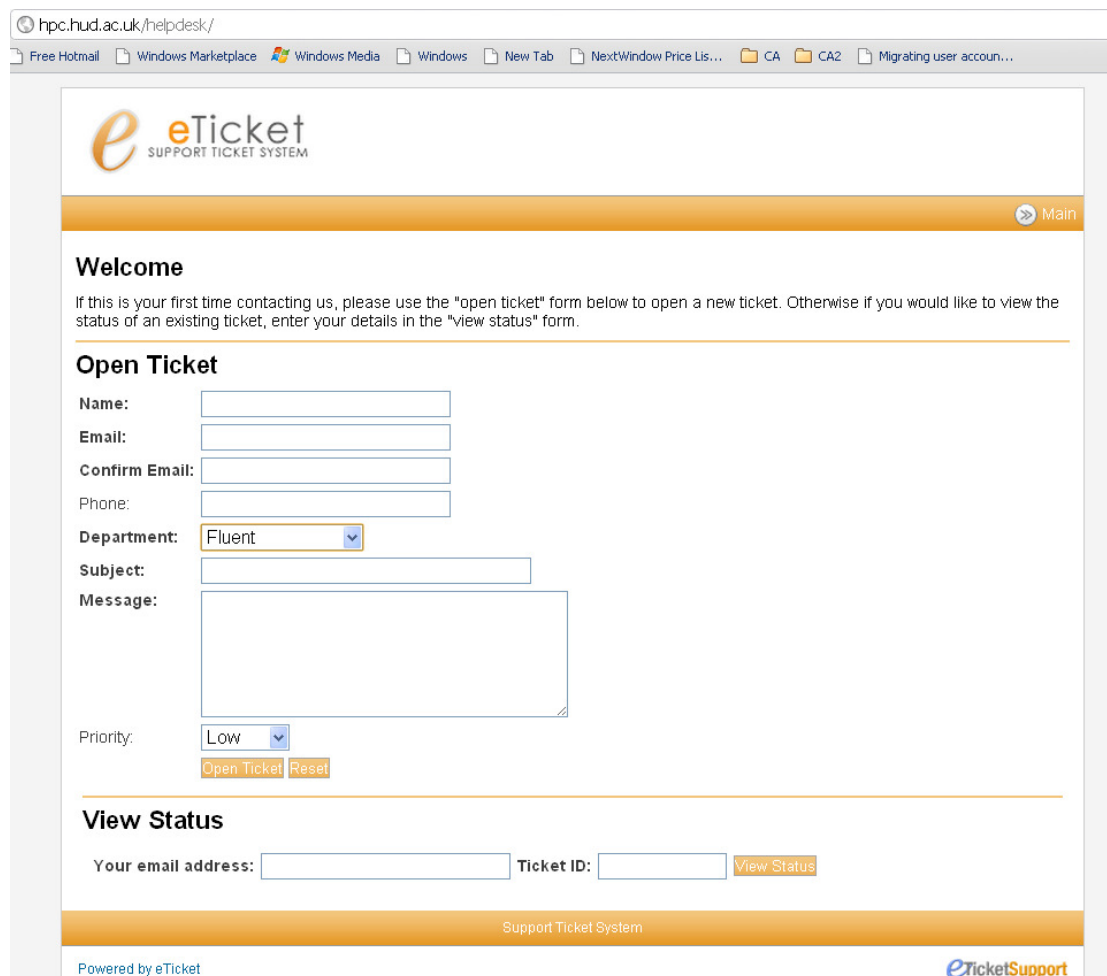
Figure 46: The eTicketing Helpdesk System implemented on the QGG

A Forum using phpBB has also been created for users to get a forum to discuss issues they might have or see threads with other user's issue. As mentioned before hpc.hud.ac.uk/wordpress is a development blog maintained by administrators and though not publically advertised this blog can be used as a tool to understand the development process of the QGG and can be used to debug issues that might arise in the future when the current administrators are not available.

Chapter 12.2: Proposed Workshops for Users

The Parallel Computer Architectures: Clusters and Grids course currently educates twenty undergraduates and sixteen postgraduate students in developing, programming and using clusters and grids. These students are mostly more interested in the development side of HPC and therefore are not end users of the QGG. Researchers, students and faculty members who do make

up user base of the QGG have little or no understanding of SSI systems or Shared/Distributed memory systems.

For such users, once every month there should be a tutorial organised and advertised so that the research community as a whole is aware of the facilities provided by the HPC-RC and users can get over fears of command line interfaces and working in a Linux and distributed environment. Regular sessions will reassure novice users that there is support in case something goes wrong. Beginner sessions can be arranged during months where several new users signup and in other months slightly advanced or NGS related sessions can be arranged.

Currently, no course is giving its students training in using software applications relevant to their field on a cluster level. Researchers and Project students are currently relying on the HPC-RC support staff to train them in their fields' application packages. This system was adequate up to now, because staff members of the HPC-RC have some experience in using the various applications as they themselves are researchers. However, once the HPC Resource Centre is formalized then the staff might not have such personal knowledge and, with new packages being acquired every year, it will be difficult to give knowledgeable advice. While the wiki site will help different experts from each field who have gone through the process should voluntarily give tutorials twice an academic year to new users and pass the mantle on when they move on from the university.

Chapter 12.3: Proposed Staff Development Seminars

More important than training users it is important to carry out staff development seminars to demonstrate to the staff members the usefulness of such a system and to change their way of thinking when it comes to high performance computing and their own research. It has been observed that many advisors limit their students work as they know that the student will inevitably hit the available computing power ceiling. Once the faculty is aware that such facilities are available in-house or can be freely sourced from outside, this will encourage them to push the boundaries of research.

Hosting staff development seminars will also facilitate changes in how many modules are taught. The Academics, will be more likely to include HPC usage of relevant applications in their curriculum so that the courses get more modernised and prepares students for the industry where HPC systems are used for a veriety of simulations.

The cross-discipline collaborative nature of the HPC-RC and HPC-RG will further be enhanced when more academics start to require this system. Essentially, these users will become stake holders in the system and will support the growth and further investment of this resource.

Chapter 12.4: NGS Related Workshops

Special training sessions are being scheduled for the upcoming academic year to train University of Huddersfield users to use tools such as Globus and gLite so that they can maximise the research output and can easily scale to national resources by porting their simulations on the NGS.

The NGS with all its member sites provides support for the many types of software being run at all the sites. This knowledge base is vital for a young university such as the University of Huddersfield as the research output and users experiences can paint an accurate picture as to the usefulness of the applications and then academics at the university can invest with proper informed prior knowledge.

The NGS also provides master classes at local sites to train the local eScience researchers in the tools available on the UK grid. Workflow training in software packages like MATLAB and ABAQUS will greatly help our local research community.

Section VI: Further Work & Conclusions

Chapter 13: Refinement

The Grid based High Performance Computing system deployed, as outlined by this report, has been done on a shoe-string budget. The case studies above have shown the limitations of the deployed systems. To further enhance the system and improve usability many improvements can be made. The following two sections outline some of the glaring changes that are required.

Chapter 13.1: Improvement at Cluster Level

The Eridani cluster requires further investment to further its effectiveness and meet the general needs of the researchers. These improvements are:

- *High Speed Interconnect:* The problem that was observed in Chapter 9.2: DL_POLY2 was that when the system was required to communicate over the network the jobs executed would slow down. The Gigabit internet is not fast enough when there is to much cross-node talk. The solution would be to implement 'Myrinet' or 'Infiniband' interconnects that would allow for cross-node communication at 40Gbps.

- *Improved Head Nodes:* The system would benefit from better quality Head Nodes. Rather than using COTS machines as the Windows© and Linux head nodes it would be better to invest in proper Server Infrastructure. The head nodes can be blades housed in the Grid and Licensing rack, while the COTS machines can all be the compute nodes. This would also enable for the head node to have a higher density of RAM, enabling longer up-times.

- *Increased Number of Nodes:* The rack that currently houses the Eridani cluster and the power supplies to this rack is underutilized. There is a possibility of adding at least 14 more nodes, in the current ITX form factor. This would make a total of 50 compute nodes (if the above mentioned stipulation of removing head nodes is followed). If slightly more compact Mini-ITX form factor machines are used, replacing the

current systems, 100 nodes could be placed on this rack. There are adequate power supplies to enable this upgrade.

- *Identical Processor Specification:* The mix-&-match with the processors on the Eridani cluster, has led to a loss in efficiency. It also means that a single job cannot be scaled across the whole cluster. Identical cores in each machine would optimise and improve the efficiency of the cluster.

- *Optimised Queuing System:* Currently the job scheduler maintains different queues for the different software's. This was done due to license restrictions affiliated with the software. Unfortunately this sort of time management meant that small single core jobs would have to wait as longer multi-node jobs would hold up the queue waiting for resources to become available. Better streamlining of these queues will lead to an improved utilisation level.

Chapter 13.2: Improvement at Grid Level

At a grid level certain changes need to be made to facilitate users and improve the performance of the system.

- *Improved Access Method:* Currently users are unable to log in from Wireless devices. To improve productivity of these users it is important that all new technologies are embraced and access by all interfaces be enabled.

- *Establishment of a Local Certificate Authority:* Student users at the Barnsley and Oldham campuses are currently unable to utilise the HPC facilities at the Queensgate Campus. This is due to the fact that the eScience X509 certificates cannot be issued to students. To meet the needs of students at the remote campuses a local CA should be established to give these users a method to connect to the QGG system.

- *Single Job Submission Interface:* With the Globus deployment users can submit jobs on the NGS from the Bellatrix grid head node. As currently there are not many diverse resources on the QGG the local clusters Job management software's are not linked to the head node. It would be beneficial to eventually connect the locals queues to Globus as well so that

users can just log in once and specify hardware criteria and have the job migrate and execute seamlessly.

- *Workflow Management and Job Submission Portals for all Applications:* To improve usability a single web based portal interface should be deployed to give all the users a uniform, user friendly environment to create and submit job files from.

Chapter 14: Establishment of HUD Grid

The current resources established for the University of Huddersfield by this project are just the stepping stones for HPC and HPC enabled research. With the changes outlined in Chapter 13.1: Improvement at Cluster Level the current clusters might meet the current local demands, but as research evolves this system will be lacking as it is a Beowulf cluster made of COTS systems. To meet the goals of becoming a World Leading institution working in High Performance Computing a more dedicated and specialised machine is required.

Due to the nature of external access to the system, via eScience X509 certificates, it will not be possible to give students, at the other two campuses of the University of Huddersfield, access to the HPC systems at the Queensgate campus. With real-estate being at a premium, it is important to keep a provision open to move some of the existing or future HPC resources to the other campuses. These new constructions will be able to house new machines and expand the resources provided by the HPC-RC. The University of Huddersfield has many researchers who work off-site using HPC systems provided by their host organisations. To improve the productivity of these researchers these HPC systems should be integrated in the existing infrastructure so that they may benefit from more resources. At this point the service will no longer be the Queensgate Grid but a greater University of Huddersfield Computing Grid.

Chapter 14.1: Current Restraints and Requirements

Setting up a new HPC system using specialised hardware and linking all the campuses and resources have many ramifications. The restraints that are currently obvious are:

- *The federated nature of Schools and Departments in the University of Huddersfield*. This makes it difficult to consolidate software's and levy costs, for sustainability.
- *Lack of Space*. Over the last decade the Queensgate campus has grown to accommodate more departments and a larger student body. This has led to a lack of space for any large projects or initiatives and the University has had to purchase more real estate.

- *Listed Buildings*. Purchasing new buildings does not always work out, for a new large project, as a majority of the buildings around the area are listed and therefore no changes can be made that affect the outer facade. A proper data centre will be needed to host a large HPC unit but with the restrictions on construction, proper cooling devices and power transformers will not be accommodated.

- *Stressing the University IT backbone.* When there is sustained activity on the respective clusters, which are distributed across the network, there is a serious risk of crippling the current University Network backbone. Currently there are no high-capacity lines connecting the various buildings on the University Intra-net. Careful expansion of the HPC resources is required that moves in tandem to the Computing services upgrades.

- *Stress on the Super JANET 4 Uplink.* With external systems and users connecting from external sites the Universities JANET connection will also suffer some strain. Most Universities that have been contacted regarding the performance of their uplink on the NGS, have said that their quality of service has not been affected by users submitting jobs to the NGS. It should be noted that most of these institutions, the primary HPC devices are local clusters that are on their local network. In the case of the NGS partner node on the White Rose Grid there is a dedicated Super Janet 4 trunk, so that the universities normal access to the internet is not affected.

- *Weaker Infrastructure at the Smaller Campuses.* The other campuses of the University of Huddersfield do not have the same quality uplinks or internal infrastructure that the Queensgate Campus enjoys. Introducing an HPC system on those sites or getting users to connect to remote sites might stress the IT infrastructure.

Chapter 14.2: Solutions for New HPC System and the Establishment of the HUD-Grid

By meeting 'Partners' on the NGS a set of scenarios has been evaluated for the possible growth of the local system and the possibility of establishing a HUD

Grid. There is an opportunity to collaborate with the STFC Daresbury site as they have a large data centre empty and it is available for use. The Daresbury site used to host HPCx and since it's decommissioning the machine room has been empty. Hosting a new machine at Daresbury will not only be economically viable but will give access to the wealth of knowledge present at the STFC Daresbury Laboratories.

A benefit analysis of the hosting options for a new HPC system is as follows:

Plan		Advantages	Disadvantages
A	HPC equipment procured by the University and located at Queensgate campus of the University	• Daily access to HPC equipment • Faster Access • Visible University Asset	• Limited space available to house HPC • Limitation in possible building alterations – listed buildings • Lack of Infrastructure • Large carbon footprint • 60% of funding will be used to provide infrastructure – power, ventilation, maintenance
B	HPC equipment procured by the University as Part of the University's Data Centre	• Reduced Infrastructure cost (incorporate into the data centre)	• Plans for The University data centre not currently available
C	The University of Huddersfield procures the equipment and accommodate the resources at Daresbury Laboratory (STFC)	• Low infrastructure cost • Low carbon footprint • Experienced STFC staff to provide SW, HW and HPC support	• Proximity- access to hardware remote – requires travelling to DL • Possible strain on the university network • Hosting SLA required
D	Daresbury Laboratory, (STFC) procures the HPC resources and accommodate the equipment	• More processing power – better value for money due to the STFC staff expertise in acquiring HPC SW/HW • Time– fast deployment and utilisation of the HPC • Revenue sharing arrangement for any cycles that OCF sells on the system • University is demonstrating participation in shared services • High-profile strategic partnership with STFC • No Hosting SLA required	• New software licence required to reside on HPC in DL • Proximity- access to hardware remote – requires travelling to DL • After 3 years of HPC at DL – no physical assets available. There is a possibility to request the recovery of HW from DL after 3 years

Chapter 14.3: Sustainability of the proposed system

The sustainability of the new HPC system will be ensured as follows:

- The provision of staff time to support new and existing users will ensure the system is well used, maintaining the need for the resource throughout the lifetime of the grant and beyond.
- Secondly the purchase of software licences for the system will ensure new researchers can migrate to the system easily.
- The links with DL, the National Grid System will ensure the system is well maintained and users possess the tools which enable their research to proceed smoothly.
- HPC group will be applying for external funding through RC and other funding bodies. HPC group will submit a first stage proposal for EPSRC High Performance Computing (HPC) software development call by 15th September.
- Finally as research and enterprise income is generated by the users of the system it will be expected that any such award contains an element of funding for the resource which will be used to upgrade the system and to fund any infrastructure costs incurred by CLS in supporting the system.
- It will reduce costs in purchasing the same software tools and hardware for individual schools that can be centralised, using campus wide licences, and used between schools.
- Resources HW/SW can generate extra income from consultancies and better utilisation of resources - return on investment.

Chapter 14.4: Impact of the proposed system

The proposed system will support the University's submission to the Research Excellence Framework, and potentially impact the economy and society through the research output enabled by this system. The predicted impact is:

- HPC centre will increase research outputs across the university from existing research groups, and encourage collaboration and cooperation leading to new research initiatives.

- HPC centre will support research, enterprise and knowledge transfer activities that has commercial as well as intellectual value.

- Researchers will be more productive – HPC will reduce the simulation and modelling completion time

- It will enable newly established HPC group to expand, increase number, impact and quality of the publications from the HPC users and achieve international and national research excellence.

- Create flagship research and resources – supercomputers.

- HPC software research would influence how Cloud computing is used to provide Software, Infrastructure, Storage, Platform and Applications as a service for business, industry and individual users of IT technology

- HPC centre for Huddersfield University researchers, housed at DL, will reduce carbon footprint for the university by allowing DL to manage the power, cooling and running of the equipment.

- Impact will be demonstrated in the publication of research in high impact, peer reviewed journals, and at national and international conferences. This in turn will enable the leverage of external research funding, particularly from the research councils.

A number of current research projects being run on the local system will benefit with the expansion. These projects will potentially have an impact on renewable and nuclear energy technologies such as:

- SAS project will be using HPC resources for the development of highly novel supramolecular materials containing photophysically active transition metal centres with applications in light harvesting solar energy conversion.

- SAS researchers will use HPC as part of a wider project aimed at understanding biomineralisation and how it can be applied to the development of new materials.

- SAS researchers are currently using high performance computing to evaluate the efficiency of dopants in ZrO_2 and CeO_2 based catalysts and in accessing the viability of ThO_2 as a next generation nuclear fuel.

The HPC centre will impact research projects in School of Computing and Engineering

- Current projects in Automotive Engineering research group are using HPC facilities to improve fuel efficiencies in areas of automotive design and fuel chemistry.
- HPC resources will provide a platform for research and development work in engineering codes and engineering packages for multi-core and multi-computer systems. This research would unify researchers across the School of Computing and Engineering in areas of software engineering, algorithm design, mathematics and mechanical and electronics engineering, and enable creation of new codes and further development of existing codes, leading to commercial tools and packages design.

The HPC centre will impact research projects in Computer Games and 3D Animations

- SCE has a strong reputation in Computer Games. The HPC centre will provide render farm facilities for 3D modelling/ Visualisation with application to Computer Games
- The time to market for Canal Side Studio (Games development) and ADA (product development) will be shortened greatly.

The HPC centre will impact enterprise activities

- HPC will enable the students and researchers to create professional full HD quality. 3D Stereo animations and films where 3D technology is employed within course curriculas - a capability which does not presently exist.

- HPC centre will support the growth and competitiveness of SMEs in Yorkshire to the benefit of local economies in the region.

Chapter 15: Becoming NGS Partners

To promote the research activities of the HPC-RG it was felt that joining the NGS, as a resource provider, would be an important step. According to the roadmap decided for the HPC-RG, it was planned that the first step would be to make remote NGS facilities available to the research community in Huddersfield. Then with adequate resources in hand the HPC-RC would attempt to become an affiliate site on the NGS. After reaching affiliate status an assessment will have to be made to check if becoming Partner members will affect the quality of service currently provided.

The reason for wanting to become a member site on the NGS is the exposure it gives the University of Huddersfield and the High Performance Research group. To be a part of such a large collaborative group as a contributing member will bring positive attention towards the University, and afford possible future research collaboration.

Chapter 15.1: Roadmap to Affiliate Status

To establish the University of Huddersfield as a major centre for High Performance Computing in the United Kingdom, some inroads have been made to becoming an affiliate site on the NGS. There are five major steps in the process to becoming an affiliate site of which two have been fully completed and two of the steps partially complete.

1. *"Contact is established between the prospective site and the NGS. This may be through NGS to site communication or the site completing the site application form."* At the various conferences (IEEE, UK eScience All Hands at Oxford and Cardiff), training sessions (UK eScience RA Operator Training) and the NGS road show members of the HPC-RC expressed their interest to join the NGS and provide resources on the Grid. On the 20th of April 2010 the University of Huddersfield was registered as a site *"progressing to NGS affiliate"*.

2. *The NGS Outreach officer will organise a Roadshow event to give an introduction to the NGS and the services that we offer.* Several staff

development seminars have been scheduled for October 2010 to introduce and give basic training in using HPC systems. After which an NGS road show will be scheduled, to introduce the Huddersfield research community to the possibilities opened up by the NGS.

3. *The site nominates a Campus Champion who will act as the operational level bridge between the user communities, the host institution and the NGS.* Following the acceptance of the University of Huddersfield as a site on the NGS on the 20th of April, this author was appointed as an RA operator on the 21st of April 2010 for the Huddersfield DOCABS Registration Authority and on the 23rd of April 2010 as the Campus Champion for the University of Huddersfield. A "Campus Buddy", who serves as the contact person for the Campus Champion and ensures that a site is ready to become an affiliate/partner, has also been assigned to the University of Huddersfield.

4. *The site makes a decision of which type of resource exchanging member they wish to become, either Partner or Affiliate. This will include the installation of a community specific or general software profile onto their resources. These installation profiles are listed within the NGS Site Level Services document.* As the HPC-RC develops and the system outlined in the bid becomes available the HPC-RG would like to make this resource initially available as an Affiliate Site on the NGS. The following steps are those that are required to meet the conditions for an affiliate site.

The core of the NGS is the resources that the community of users are able to access. Both partners and affiliates run NGS compatible software, and integrate monitoring and support arrangements with the NGS. To affiliate with the NGS an institution or resource provider must:

- Deploy and support the minimum required set of NGS software to enable interoperability with the NGS central services and other NGS sites.
- Provide access to allow NGS monitoring
- Agree to the NGS conditions of use and security practices.

- Sites should accept certificates issued by the UK e-Science Certificate Authority and those

- CAs with which the UK e-Science Programme has reciprocal agreements. The Certificate Revocation Lists are updated on a regular basis.

Once all the conditions are met the NGS runs successive weeks of testing to ensure that all systems on this new site are up to speed. Once these tests are passed the site is registered as an official affiliate site.

5. *Create site specific information sources within both the Grid Operations Centre Database (GOCDB) and the NGS webpage.* This step will be completed after the full extent of the University of Huddersfield participation has been decided. (NGS 2010b)

Chapter 15.2: Feasibility of Partner Status

Partner status of the University of Huddersfield would be an important milestone for the HPC-RG. Many users who require long term access to HPC systems prefer using Partner resources, due to the highly available nature of these services. When these projects publish their findings, it is common etiquette to make a mention of the support of the Partner Site. This publicises the service provided by the Partner site and in turn leads to further collaborations and elevates the institutions research profile.

Before the University of Huddersfield becomes a member site it must ascertain how this loss of autonomy on the resources will affect its own researchers. The priority for the HPC-RC is that the HPC resources should be available at the finger tips of local users. This implies that local users should not have to wait to run their jobs, behind users from other institutions. Special care must also be taken with regards to the clauses presented in the SLA that is required of the Partner sites.

The requirements of a Partner organisation are:

"Partners also contribute significant resources to NGS users at large. A partner must also complete a Service Level Description document which defines

what they provide to the user community. NGS partners are sites that meet the requirements for affiliation with the NGS and in addition:

- Contribute additional services, as agreed with the GOSC Management Board and defined in a Service Level Description (SLD), to NGS approved users or projects.
- Allow additional monitoring and accounting for verification of the services provided.
- Allow inclusion of the SLD services in a national registry, the structure for which is to be decided.
- Services offered may include access to hardware resources, data archives or appropriately licensed software in addition to that required by the NGS provided that they do not adversely affect the operation of the site in question, any other NGS site, or any core services.
- NGS partnership entitles sites to representation on the GOSC Management Board. Initially this may be through direct membership of the board; however, representation through functional or regional consortia is a longer term goal." (NGS 2010c)

Chapter 18: Conclusion

This project has established a complete High Performance Computing solution for the University of Huddersfield. It encompasses, investigation of the current trend, implementation of a definitive solution, and defines the operational procedure for day to day management and customer support. Based on the experience gained by this project, 3 papers have been presented or have been approved for presentation at International Conferences. A referee at one conference gave this feedback:

"This is an interesting case study of rolling out a grid in a campus environment; it will no doubt be of interest to both researchers and IT professionals on many such campuses…"

The ramifications of this project will be seen for months and years to come, as researchers and students publish more and more research that has been made possible by this HPC system. The move to join the NGS and partnership with a prestigious government research body like STFC will raise the profile of the University of Huddersfield. It will also open the doors to large scale international projects. This should propel the University to the forefront of universities doing cutting edge research.

The University's central Research Committee (at the time of printing this thesis) approved a substantial amount of funds for the University to purchase a specialised HPC system in collaboration with the STFC Daresbury Laboratories.

Future research efforts will be directed towards a Grid/Cloud infrastructure combining available resources from University schools and departments, and from neighbouring FE colleges. Further work will be carried out to link the geographically-dispersed campuses in Yorkshire and the University Centre at Blackburn College, Lancashire. This will lead to establishing a Virtual Organization comprising a consortium of small to medium colleges and HE institutions in Lancashire and Yorkshire. It will enable resource sharing such as cluster storage, processing power, instrumentation, dedicated software and hardware, and encourage collaboration between our institutions, establishing a

framework for Cloud infrastructure for education, industry, business and local government.

Section VII: Bibliography

ANSYS, 2010. CFD Flow Modeling Software & Solutions from Fluent. Available at: http://www.fluent.com/ [Accessed October 1, 2010].

Autodesk, 2011. 3D Design & Engineering Software for Architecture, Manufacturing, and Entertainment. Available at: http://usa.autodesk.com/ [Accessed October 1, 2010].

Baker, C., Turnkey Dual-Boot Clusters: Clustercorp and Microsoft® present Rocks+Hybrid, a simple Windows©/Linux Cluster Solution. Available at: http://www.Microsoft®.com/hpc/en/us/community/hpc-forums-blogs.aspx [Accessed June 1, 2010].

Berlich, R., Kunze, M. & Schwarz, K., 2005. Grid computing in Europe: from research to deployment. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research - Volume 44*. Newcastle, New South Wales, Australia: Australian Computer Society, Inc., pp. 21-27. Available at: http://portal.acm.org/citation.cfm?id=1082290.1082294 [Accessed May 29, 2010].

Blender 3D, 2010. Doc:Manual/Introduction - BlenderWiki. Available at: http://wiki.blender.org/index.php/Doc:Manual/Introduction [Accessed October 1, 2010].

Bucholtz, J. & Zebrowski, M., 2007. Dual Boot: WCCS and Rocks Cluster Distribution. Available at: http://www.Microsoft®.com/downloads/details.aspx?displaylang=en&FamilyID=e73a468e-2dbf-4782-8faa-aaa20acb63f8 [Accessed June 1, 2010].

Calleja, M. et al., 2004. CamGrid: Experiences in constructing a university-wide, Condor-based grid at the University of Cambridge. In *Proceedings of UK e-Science All Hands Meeting*.

Carrigan, T., 2002. Setting up an Oscar cluster where the nodes will alsobe. *Open Source Cluster Application Resource*. Available at: http://www.mail-archive.com/oscar-users@lists.sourceforge.net/msg00662.html [Accessed June 1, 2010].

CERN, 2010. EGEE > gLite. Available at: http://glite.web.cern.ch/glite/

[Accessed October 1, 2010].

CFS, 2006. Gamuess-UK. Available at: http://www.cfs.dl.ac.uk/ [Accessed October 1, 2010].

Chong, A., Sourin, A. & Levinski, K., 2006. Grid-based computer animation rendering. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*. pp. 39–47.

Cobham, 2010. VECTOR FIELDS - Software for Electromagnetic Design - Opera-3d. Available at: http://www.vectorfields.com/content/view/26/49/ [Accessed October 1, 2010].

Dew, P.M. et al., 2003. The white rose grid: practice and experience. In *UK eScience-All Hands Meeting*.

DistroWatch, 2010. DistroWatch.com: Put the fun back into computing. Use Linux, BSD. Available at: http://distrowatch.com/ [Accessed October 1, 2010].

Foster, I., 2006. Globus Toolkit Version 4: Software for Service-Oriented Systems. *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS 3779, pp 2-13.

Foster, I., Kesselman, C. & Tuecke, S., 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, (15(3)), 200-222.

HECToR, 2009. HECToR » What is HECToR and why is it special? Available at: http://www.hector.ac.uk/abouthector/hectorbasics/ [Accessed October 1, 2010].

HPCx, 2010. HPCx Home Page. Available at: http://www.hpcx.ac.uk/ [Accessed October 1, 2010].

Huajun Jing & Bin Gong, 2008. The design and implementation of Render Farm Manager based on OpenPBS. In *Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on*. Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on. pp. 1056-1059. Available at: 10.1109/CAIDCD.2008.4730744 [Accessed September 30, 2010].

Hull, D. et al., 2006. Taverna: a tool for building and running workflows of services. *Nucleic acids research*, 34(suppl 2), W729.

Jenssen, C.B., 2001. *Parallel computational fluid dynamics: trends and applications : proceedings of the Parallel CFD 2000 Conference*, Gulf Professional Publishing.

Kaurinkoski, P. et al., 2001. Performance of a Parallel CFD-Code on a Linux Cluster. In *Parallel computational fluid dynamics: trends and applications: proceedings of the Parallel CFD 2000 Conference*. p. 107.

Lamanna, M., 2004. The LHC computing grid project at CERN. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 534(1-2), 1–6.

Lundqvist, A. & Rodic, D., 2010. GNU/Linux distro timeline. Available at: http://futurist.se/gldt/ [Accessed October 1, 2010].

Mathworks, 2010. MathWorks United Kingdom - MATLAB Distributed Computing Server - MATLAB. Available at: http://www.mathworks.co.uk/products/distriben/ [Accessed October 1, 2010].

Microsoft®, 2007. Dual Boot White Paper. Available at: http://www.Microsoft®.com/downloads/en/results.aspx?freetext=Dual -boot&displaylang=en&stype=s_basic [Accessed June 1, 2010].

Microsoft®, 2008. Windows© HPC Server 2008 Technical Overview. *Microsoft®.com*. Available at: http://www.Microsoft®.com/downloads/en/confirmation.aspx?FamilyI d=7A4544F0-81F2-4778-8A59-35C43BA49875&displaylang=en [Accessed October 1, 2010].

NETLIB, 2010. LINPACK. Available at: http://www.netlib.org/linpack/ [Accessed October 1, 2010].

NGS, 2009. Who we are and what we do | NGS. Available at: http://www.ngs.ac.uk/aboutUs [Accessed October 1, 2010].

NGS, 2010a. NGS software stack | NGS. Available at: http://ngs.ac.uk/site-level-services/ngs-software-stack [Accessed October 1, 2010].

NGS, 2010b. Becoming a member | NGS. Available at: http://ngs.ac.uk/member-sites/becoming-a-member [Accessed October 1, 2010].

NGS, 2010c. NGS Joining Procedure. Available at: http://ngs.ac.uk/member-sites/becoming-a-member [Accessed October 1, 2010].

Open Science Grid, 2010. Virtual Data Toolkit. Available at: http://vdt.cs.wisc.edu/ [Accessed October 1, 2010].

Padgett, J., Djemame, K. & Dew, P., 2005. Grid-based SLA management. *Advances in Grid Computing-EGC 2005*, 1076–1085.

Palmer, R., 2010. Vehicular Design Efficiency of a Range Rover.

Pennisi, E., 2010. Conquering by Copying. *Science*, 328(5975), 165.

SANDIA, 2010. LAMMPS Molecular Dynamics Simulator. Available at: http://lammps.sandia.gov/ [Accessed October 1, 2010].

Simula, 2009. SIMULIA > Products. Available at: http://www.simulia.com/products/abaqus_fea.html [Accessed October 1, 2010].

Simula, Multiphysics Modeling and Simulation Software - COMSOL. Available at: http://www.comsol.com/ [Accessed October 1, 2010].

Sterling, T.L. et al., 1999. *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters* 2nd ed., Cambridge, Massachusetts: The MIT Press.

STFC, 2007. History of scientific computing facilities at CCLRC. *Rutherford Appleton Laboratories, STFC*. Available at: http://www.w3c.rl.ac.uk/pasttalks/CCLRC_Computer_History/cclrc_computing_history.html [Accessed October 1, 2010].

STFC, D., 2010. DL POLY. *STFC Computational Science and Engineering Department - Molecular Simulation*. Available at: http://www.cse.scitech.ac.uk/ccg/software/DL_POLY/ [Accessed October 1, 2010].

Sunday Times, 2010. Profile: University of Huddersfield - Times Online. Available at:

http://www.timesonline.co.uk/tol/life_and_style/education/sunday_ti
mes_university_guide/article4772036.ece [Accessed October 1, 2010].

University of Huddersfield, 2008. RAE Results. *University of Huddersfield*. Available at: http://www2.hud.ac.uk/research/research/RAE_Results.php [Accessed August 14, 2010].

Valiev, M. et al., 2010. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications*, 181(9), 1477-1489. Available at: http://www.sciencedirect.com/science/article/B6TJ5-502V6YP-2/2/2d40ddac658b388d1681698f0642d5e5 [Accessed October 1, 2010].

Vallee, G., 2010. OSCAR Project – Trac. Available at: http://svn.oscar.openclustergroup.org/trac/oscar [Accessed October 1, 2010].

Wallom, D.C. & Trefethen, A.E., 2006. Oxgrid, a campus grid for the university of oxford. In *Proceedings of the UK e-Science All Hands Meeting*.

Watson, G. & Oliver, P., 2004. Computational Solid State Chemistry Group. Available at: http://people.bath.ac.uk/chsscp/group/programs/programs.html [Accessed October 1, 2010].

Wisc EDU, 2010. Condor Project Homepage. Available at: http://www.cs.wisc.edu/condor/ [Accessed October 1, 2010].

Section VIII: Appendix

A: QGG Job Queue Setup

```
#
# Create queues and set their attributes.
#
#
# Create and define queue workq
#
create queue workq
set queue workq queue_type = Execution
set queue workq resources_max.cput = 10000:00:00
set queue workq resources_max.ncpus = 128
set queue workq resources_max.nodect = 32
set queue workq resources_max.walltime = 10000:00:00
set queue workq resources_min.cput = 00:00:01
set queue workq resources_min.ncpus = 1
set queue workq resources_min.nodect = 1
set queue workq resources_min.walltime = 00:00:01
set queue workq resources_default.cput = 10000:00:00
set queue workq resources_default.ncpus = 1
set queue workq resources_default.nodect = 1
set queue workq resources_default.walltime = 10000:00:00
set queue workq resources_available.nodect = 32
set queue workq enabled = True
set queue workq started = False
#
# Create and define queue bburnq
#
create queue bburnq
set queue bburnq queue_type = Execution
set queue bburnq resources_max.cput = 10000:00:00
set queue bburnq resources_max.ncpus = 32
set queue bburnq resources_max.nodect = 16
set queue bburnq resources_max.walltime = 10000:00:00
set queue bburnq resources_min.cput = 00:00:01
set queue bburnq resources_min.ncpus = 1
set queue bburnq resources_min.nodect = 1
set queue bburnq resources_min.walltime = 00:00:01
set queue bburnq resources_default.cput = 10000:00:00
set queue bburnq resources_default.ncpus = 1
set queue bburnq resources_default.nodect = 1
set queue bburnq resources_default.walltime = 10000:00:00
set queue bburnq resources_available.nodect = 16
set queue bburnq enabled = True
set queue bburnq started = True
#
# Create and define queue fluentq
```

```
#
create queue fluentq
set queue fluentq queue_type = Execution
set queue fluentq resources_max.cput = 10000:00:00
set queue fluentq resources_max.ncpus = 28
set queue fluentq resources_max.nodect = 7
set queue fluentq resources_max.walltime = 10000:00:00
set queue fluentq resources_min.cput = 00:00:01
set queue fluentq resources_min.ncpus = 1
set queue fluentq resources_min.nodect = 1
set queue fluentq resources_min.walltime = 00:00:01
set queue fluentq resources_default.cput = 10000:00:00
set queue fluentq resources_default.ncpus = 1
set queue fluentq resources_default.nodect = 1
set queue fluentq resources_default.walltime = 10000:00:00
set queue fluentq resources_available.nodect = 7
set queue fluentq enabled = True
set queue fluentq started = True
#
# Create and define queue chemq
#
create queue chemq
set queue chemq queue_type = Execution
set queue chemq resources_max.cput = 10000:00:00
set queue chemq resources_max.ncpus = 128
set queue chemq resources_max.nodect = 32
set queue chemq resources_max.walltime = 10000:00:00
set queue chemq resources_min.cput = 00:00:01
set queue chemq resources_min.ncpus = 1
set queue chemq resources_min.nodect = 1
set queue chemq resources_min.walltime = 00:00:01
set queue chemq resources_default.cput = 10000:00:00
set queue chemq resources_default.ncpus = 1
set queue chemq resources_default.nodect = 1
set queue chemq resources_default.walltime = 10000:00:00
set queue chemq resources_available.nodect = 32
set queue chemq enabled = True
set queue chemq started = True
#
# Set server attributes.
#
set server scheduling = True
set server default_queue = workq
set server log_events = 64
set server mail_from = adm
set server query_other_jobs = True
set server resources_available.ncpus = 128
set server resources_available.nodect = 32
set server resources_available.nodes = 32
```

```
set server resources_max.ncpus = 128
set server resources_max.nodes = 32
set server scheduler_iteration = 60
set server node_check_rate = 150
set server tcp_timeout = 6
set server pbs_version = 2.1.8
```

B: User Creation Script

```
#!/bin/bash

NEW_USERS="/root/users.txt"
EMSG="/tmp/emilmessage.txt"
EMSG2="/tmp/ngsmail.txt"
HOME_BASE="/home/"

cat $NEW_USERS | \
while read USER COMMENT EMAIL NGS
do
 export PASSWORD=`apg -a 0 -n 1`
# export ENCPASSWD=`mkpasswd -m md5 $PASSWORD`
 useradd -c $COMMENT -p $PASSWORD -m -d $HOME_BASE$USER $USER
 echo "Dear "$COMMENT > $EMSG
 echo " " >> $EMSG
 echo "These are your login details to the Queensgate Grid (@ qgg.hpc.hud.ac.uk) : " >> $EMSG
 echo "userid: "$USER >> $EMSG
 echo "password: "$PASSWORD >> $EMSG
 echo $PASSWORD
 echo " " >> $EMSG
 echo "Please give atleast 1 hour for your account to sync before login in" >> $EMSG
 echo "This is an automated email so please do not hit reply" >> $EMSG
 echo "If you are facing any difficulties call on ext 1855 or email i.kureshi@hud.ac.uk" >> $EMSG
 echo "To change your password login and type passwd; then follow the instructions" >> $EMSG
 echo " " >> $EMSG
 echo "To get the recommended portable toolkit for cluster use paste this address" >> $EMSG
 echo "in your browser: http://hpc.hud.ac.uk/hpc/files/QGG-Student-Toolkit.zip" >> $EMSG
 mail -s "Login Details to Queensgate Cluster" "$EMAIL" < $EMSG
 export PASSWORD="0"
 echo 0 > $EMSG

 echo "Dear "$COMMENT > $EMSG2
 echo " " >> $EMSG2
 echo "    According to the request submitted to the HPC Centre you have expressed " >> $EMSG2
 echo "the need to use the NGS to assist you in your simulations. As the NGS is an " >> $EMSG2
 echo "external body you will have to register for an eScience certificate and then " >> $EMSG2
 echo "for time on the NGS. As the UoH HPC Centre is also the access point to the " >> $EMSG2
 echo "NGS we will help you every step of the way to get your credentials. " >> $EMSG2
 echo " " >> $EMSG2
 echo "To begin the process please visit the link below (in FIREFOX or IE <= v6 only): " >> $EMSG2
 echo "  https://ca.grid-support.ac.uk/cgi-bin/pub/pki?cmd=getStaticPage&name=index " >> $EMSG2
 echo "Chose Huddersfield (DOCABS) as your RA. " >> $EMSG2
 echo "Once you receive a confirmation email from the eScience Council please " >> $EMSG2
 echo "schedule a time for an appointment with the RA Operator/Manager that is " >> $EMSG2
 echo "specified in the email. You will be required to bring some documents to the " >> $EMSG2
 echo "HPC office to complete the process. " >> $EMSG2
 echo " " >> $EMSG2
 echo "After your certificate is issued you will have to complete the NGS registeration " >> $EMSG2
 echo "and ask for computing time and storage space. This can be done from: " >> $EMSG2
 echo "https://uas.ngs.ac.uk/apply.php " >> $EMSG2
 echo " " >> $EMSG2
 echo "Please feel free to contact us if you require any further assistance along the way" >> $EMSG2
```

```
if [[ "$NGS" == "y" ]]; then
  mail -s "Access to the National Grid" "$EMAIL" < $EMSG2
fi
echo 0 > $EMSG2

done
```

C: Eridani Node Configuration

guitemp.qgg.hud.ac.uk np=4 GUI all
node01.Queensgate-CLS np=4 C23 all
node02.Queensgate-CLS np=4 C23 all
node03.Queensgate-CLS np=4 C23 all
node04.Queensgate-CLS np=4 C23 all
node05.Queensgate-CLS np=4 C23 all
node06.Queensgate-CLS np=4 C23 all
node07.Queensgate-CLS np=4 C23 all
node08.Queensgate-CLS np=4 C23 all
node09.Queensgate-CLS np=4 C23 all
node10.Queensgate-CLS np=4 C23 all
node11.Queensgate-CLS np=4 C23 all
node12.Queensgate-CLS np=4 C23 all
node13.Queensgate-CLS np=4 C23 all
node14.Queensgate-CLS np=4 C23 all
node15.Queensgate-CLS np=4 C23 all
node16.Queensgate-CLS np=4 C23 all
node17.Queensgate-CLS np=4 C25 all
node18.Queensgate-CLS np=4 C25 all
node19.Queensgate-CLS np=4 C25 all
node20.Queensgate-CLS np=4 C25 all
node21.Queensgate-CLS np=4 C25 all
node22.Queensgate-CLS np=4 C25 all
node23.Queensgate-CLS np=4 C25 all
node24.Queensgate-CLS np=4 C25 all
node25.Queensgate-CLS np=4 C25 all
node26.Queensgate-CLS np=4 C25 all
node27.Queensgate-CLS np=4 C25 all
node28.Queensgate-CLS np=4 C25 all
node29.Queensgate-CLS np=4 C25 all
node30.Queensgate-CLS np=4 C25 all
node31.Queensgate-CLS np=4 C25 all
node32.Queensgate-CLS np=4 C23 all

D: TauCeti Node Configuration

tcnode01.tauceti.qgg.hud.ac.uk np=4 all
tcnode02.tauceti.qgg.hud.ac.uk np=4 all
tcnode03.tauceti.qgg.hud.ac.uk np=4 all
tcnode04.tauceti.qgg.hud.ac.uk np=4 all
tcnode05.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode06.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode07.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode08.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode09.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode10.tauceti.qgg.hud.ac.uk np=2 Msd all
tcnode11.tauceti.qgg.hud.ac.uk np=2 Msd all

E: SSH Key Generation

```sh
#!/bin/sh

user=`whoami`
home=`getent passwd $user`
if test "$?" != "0"; then
   home=`getent passwd | egrep "^$user\:"`
fi
home=`echo $home | awk -F: '{print $6}' | tail -1`
if [ "$user" == "nobody" ] ; then
   echo Not creating SSH keys for user $user
elif [ `echo $home | wc -w` -ne 1 ] ; then
   echo cannot determine home directory of user $user
else
   # echo the home directory for user $user is $home
   # echo cd $home
   if ! cd $home ; then
      echo cannot cd to home directory $home
   else

      file=$home/.ssh/id_dsa
      type=dsa
      if [ ! -e $file ] ; then
         echo generating ssh file $file ...
         ssh-keygen -t $type -N '' -f $file
      fi

      file=$home/.ssh/identity
      type=rsa1
      if [ ! -e $file ] ; then
         echo generating ssh file $file ...
         ssh-keygen -t $type -N '' -f $file
      fi

      file=$home/.ssh/id_rsa
      type=rsa
      if [ ! -e $file ] ; then
         echo generating ssh file $file ...
         ssh-keygen -t $type -N '' -f $file
      fi

      id="`cat $home/.ssh/id_dsa.pub`"
      file=$home/.ssh/authorized_keys2
      if ! grep "^$id\$" $file >/dev/null 2>&1 ; then
         echo adding id to ssh file $file
         echo $id >> $file
      fi
```

```
        id="`cat $home/.ssh/identity.pub`"
        file=$home/.ssh/authorized_keys
        if ! grep "^$id\$" $file >/dev/null 2>&1 ; then
           echo adding id to ssh file $file
           echo $id >> $file
        fi

        id="`cat $home/.ssh/id_rsa.pub`"
        file=$home/.ssh/authorized_keys2
        if ! grep "^$id\$" $file >/dev/null 2>&1 ; then
           echo adding id to ssh file $file
           echo $id >> $file
        fi

        # echo chmod 600 $home/.ssh/authorized_keys*
        chmod 600 $home/.ssh/authorized_keys*

   fi
fi
```

F: QGG SSH Configuration

```
#       $OpenBSD: sshd_config,v 1.73 2005/12/06 22:38:28 reyk Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
#Protocol 2,1
Protocol 2
#AddressFamily any
ListenAddress 10.4.88.72
#ListenAddress ::

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 768

# Logging
# obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6

#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys
```

For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
similar for protocol version 2
#HostbasedAuthentication no
Change to yes if you don't trust ~/.ssh/known_hosts for
RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no

Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

GSSAPI options
#GSSAPIAuthentication no
GSSAPIAuthentication yes
#GSSAPICleanupCredentials yes
GSSAPICleanupCredentials yes

Set this to 'yes' to enable PAM authentication, account processing,
and session processing. If this is enabled, PAM authentication will
be allowed through the ChallengeResponseAuthentication mechanism.
Depending on your PAM configuration, this may bypass the setting of
PasswordAuthentication, PermitEmptyPasswords, and
"PermitRootLogin without-password". If you just want the PAM account and
session checks to run without PAM authentication, then enable this but set
ChallengeResponseAuthentication=no
#UsePAM no
UsePAM no

Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY
LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE
LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL
#AllowTcpForwarding yes

```
#GatewayPorts no
#X11Forwarding no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#ShowPatchLevel no
#UseDNS yes
#PidFile /var/run/sshd.pid
#MaxStartups 10
#PermitTunnel no
#ChrootDirectory none

# no default banner path
#Banner /some/path

# override default of no subsystems
Subsystem     sftp   /usr/libexec/openssh/sftp-server
```

G: QGG GSI-SSH Configuration

```
Port 2222
ListenAddress 161.112.232.42
Protocol 2
PermitRootLogin no
RSAAuthentication yes
PubkeyAuthentication no
PasswordAuthentication no
ChallengeResponseAuthentication no
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
UsePAM yes
X11Forwarding yes
UsePrivilegeSeparation  yes
Subsystem      sftp   /usr/local/VDT/globus/libexec/sftp-server
```

H: QGG Hosts File

```
127.0.0.1      localhost.localdomain localhost
10.4.88.72     qgg.hud.ac.uk   qgg
::1            localhost6.localdomain6 localhost6

10.71.56.134   tauceti.qgg.hud.ac.uk tauceti
10.4.88.77     qgc.qgg.hud.ac.uk eridani
10.4.88.76     storage.qgg.hud.ac.uk storage
```

I: Eridani Hosts File

Do not remove the following line, or various programs
that require network functionality will fail.
127.0.0.1   localhost.localdomain localhost
192.168.0.2 linhead.Queensgate-CLS linhead oscar_server nfs_oscar pbs_oscar
::1   localhost6.localdomain6 localhost6
10.4.88.72   qgg.hud.ac.uk qgg
192.168.0.202   storage.qgg.hud.ac.uk storage
These entries are managed by SIS, please don't modify them.
192.168.0.101   node01.Queensgate-CLS   node01
192.168.0.102   node02.Queensgate-CLS   node02
192.168.0.103   node03.Queensgate-CLS   node03
192.168.0.104   node04.Queensgate-CLS   node04
192.168.0.105   node05.Queensgate-CLS   node05
192.168.0.106   node06.Queensgate-CLS   node06
192.168.0.107   node07.Queensgate-CLS   node07
192.168.0.108   node08.Queensgate-CLS   node08
192.168.0.109   node09.Queensgate-CLS   node09
192.168.0.110   node10.Queensgate-CLS   node10
192.168.0.111   node11.Queensgate-CLS   node11
192.168.0.112   node12.Queensgate-CLS   node12
192.168.0.113   node13.Queensgate-CLS   node13
192.168.0.114   node14.Queensgate-CLS   node14
192.168.0.115   node15.Queensgate-CLS   node15
192.168.0.116   node16.Queensgate-CLS   node16
192.168.0.117   node17.Queensgate-CLS   node17
192.168.0.118   node18.Queensgate-CLS   node18
192.168.0.119   node19.Queensgate-CLS   node19
192.168.0.120   node20.Queensgate-CLS   node20
192.168.0.121   node21.Queensgate-CLS   node21
192.168.0.122   node22.Queensgate-CLS   node22
192.168.0.123   node23.Queensgate-CLS   node23
192.168.0.124   node24.Queensgate-CLS   node24
192.168.0.125   node25.Queensgate-CLS   node25
192.168.0.126   node26.Queensgate-CLS   node26
192.168.0.127   node27.Queensgate-CLS   node27
192.168.0.128   node28.Queensgate-CLS   node28
192.168.0.129   node29.Queensgate-CLS   node29
192.168.0.130   node30.Queensgate-CLS   node30
192.168.0.131   node31.Queensgate-CLS   node31
192.168.0.132   node32.Queensgate-CLS   node32
192.168.0.251   guitemp.qgg.hud.ac.uk   guitemp

J: TauCeti Hosts File

```
127.0.0.1      localhost.localdoamin localhost
192.168.0.50 head.tauceti.qgg.hud.ac.uk head oscar_server nfs_oscar pbs_oscar
10.4.88.76     storage.qgg.hud.ac.uk storage
10.4.88.72     bellatrix.hud.ac.uk bellatrix
161.112.232.42  qgg.hud.ac.uk qgg
10.71.76.134   eridani.qgg.hud.ac.uk eridani
::1            localhost6.localdomain6 localhost6

# These entries are managed by SIS, please don't modify them.
192.168.0.51      tcnode01.tauceti.qgg.hud.ac.uk    tcnode01
192.168.0.52      tcnode02.tauceti.qgg.hud.ac.uk    tcnode02
192.168.0.53      tcnode03.tauceti.qgg.hud.ac.uk    tcnode03
192.168.0.54      tcnode04.tauceti.qgg.hud.ac.uk    tcnode04
192.168.0.55      tcnode05.tauceti.qgg.hud.ac.uk    tcnode05
192.168.0.56      tcnode06.tauceti.qgg.hud.ac.uk    tcnode06
192.168.0.57      tcnode07.tauceti.qgg.hud.ac.uk    tcnode07
192.168.0.58      tcnode08.tauceti.qgg.hud.ac.uk    tcnode08
192.168.0.59      tcnode09.tauceti.qgg.hud.ac.uk    tcnode09
192.168.0.60      tcnode10.tauceti.qgg.hud.ac.uk    tcnode10
192.168.0.61      tcnode11.tauceti.qgg.hud.ac.uk    tcnode11
```

K: Sample Mounting Configuration from NAT

```
LABEL=/1            /              ext3   defaults     1 1
LABEL=/boot1        /boot             ext3   defaults     1 2
tmpfs            /dev/shm          tmpfs  defaults     0 0
devpts           /dev/pts          devpts gid=5,mode=620  0 0
sysfs            /sys           sysfs  defaults     0 0
proc            /proc           proc   defaults     0 0
LABEL=SWAP-sda3     swap             swap   defaults     0 0
storage:/mnt/qgg_nas/users_home/home /home   nfs   rw,bg        0 0
storage:/mnt/qgg_nas/apps /apps           nfs   rw,bg        0 0
```

L: Eridani NAT Configuration

```
Chain INPUT (policy ACCEPT 27455 packets, 2732K bytes)
 pkts bytes target    prot opt in    out    source          destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in    out    source          destination
5971K 4387M ACCEPT    all -- eth0  *    0.0.0.0/0        0.0.0.0/0
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.2      tcp
spts:1024:65535 dpt:8080 state NEW
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.2      tcp
spts:1024:65535 dpt:3490 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    all -- *    eth1   0.0.0.0/0        0.0.0.0/0        state
NEW,RELATED,ESTABLISHED
5032K 3315M ACCEPT    all -- eth1  *    0.0.0.0/0        0.0.0.0/0        state
NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:8081 state NEW
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:5800 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9893 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:5969 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9892 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:5970 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9794 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9087 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9088 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:9089 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:1856 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:8677 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:6729 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:5801 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:5999 state NEW,RELATED,ESTABLISHED
   0    0 ACCEPT    tcp -- eth1 eth0   0.0.0.0/0        192.168.0.1      tcp
spts:1024:65535 dpt:443 state NEW,RELATED,ESTABLISHED
```

Chain OUTPUT (policy ACCEPT 21024 packets, 1545K bytes)
 pkts bytes target   prot opt in   out   source   destination

Chain PREROUTING (policy ACCEPT 685K packets, 47M bytes)
 pkts bytes target   prot opt in   out   source   destination
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:3490 to:192.168.0.2:3490
81412 4233K DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:80 to:192.168.0.2:80
  66 3960 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:22 to:192.168.0.2:22
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:2201 to:192.168.0.251:22
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:8081 to:192.168.0.1:80
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:5800 to:192.168.0.1:5800
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9893 to:192.168.0.1:9893
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9892 to:192.168.0.1:9892
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:5969 to:192.168.0.1:5969
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:5970 to:192.168.0.1:5970
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9794 to:192.168.0.1:9794
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9087 to:192.168.0.1:9087
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9088 to:192.168.0.1:9088
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:9089 to:192.168.0.1:9089
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:1856 to:192.168.0.1:1856
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:8677 to:192.168.0.1:8677
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:6729 to:192.168.0.1:6729
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:5801 to:192.168.0.1:5801
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:5999 to:192.168.0.1:5999
   0   0 DNAT   tcp -- eth1 *   0.0.0.0/0   0.0.0.0/0   tcp
spts:1024:65535 dpt:443 to:192.168.0.1:443

Chain POSTROUTING (policy ACCEPT 82949 packets, 4326K bytes)
 pkts bytes target   prot opt in   out   source   destination

```
 679K   46M MASQUERADE  all  -- *    eth1   0.0.0.0/0        0.0.0.0/0

Chain OUTPUT (policy ACCEPT 1543 packets, 95155 bytes)
 pkts bytes target    prot opt in    out    source           destination
```

M: Sample Job Submission Script

```
#####################################
### Job Submission Script      ###
# Change items in section 1       #
# to suit your job needs          #
#####################################
# Section 1: User Parameters      #
#####################################
#
#!/bin/bash
#PBS -l nodes=2:ppn=4
#PBS -m abe
#PBS -M i.kureshi@hud.ac.uk
#PBS -N belachew_trial
#PBS -o stdout.out
#PBS -e stderr.err
#PBS -q fluentq
#
#####################################
# Section 2: Enviornment Variables  #
# State your executable path       #
# and any license info            #
# eg:                    #
# export LM_LICENSE_FILE=7241@mech1 #
#####################################
export LM_LICENSE_FILE=7241@10.4.56.8
export FLUENTLM_LICENSE_FILE=7241@10.4.56.8


#####################################
# Section 3: Executing Commands     #
#####################################

/apps/Fluent.Inc/bin/fluent 2d -g -ssh -t8 -cnf=$PBS_NODEFILE -i
/home/sengik/fluentest/fluent.in
```

N: The Cambridge Grid Group

University of Cambridge › Department of Physics › High Energy Physics

Search

GRIDPP

Status of the LCG farm

CamGrid Status

The Cambridge Grid Group

Bruce Beckles, Frederic Brochu, Santanu Das, Karl Harrison, Mark Hayes, Karl Jeacle, Chris Lester, Andy Parker

Computing grids supported

We are operating two clusters: one in the Cavendish Laboratory involved in the LCG project and another one located at the CMS building, running GLOBUS 2.4. Another on-going project aims to interconnect these two clusters with other Cambridge University resources in a campus-wide grid.

Associated projects:

- The GANGA project
- ATLAS Data Challenges
- LHCb Data Challenges:
  - CHEP paper on LHCb DC production system
- Multicast for data replication across Grid sites.
- CamGrid

Publications

- GANGA documentation:
  - K. Harrison et al., "Ganga: a user-Grid interface for ATLAS and LHCb", in Proceedings of "e-Science All Hands Meeting 2004", Nottingham, 31st August - 3rd September 2004
  - K. Harrison et al., "Ganga: a user-Grid interface for ATLAS and LHCb", in Proceedings of the 2003 Conference for Computing in High Energy and Nuclear Physics, La Jolla, California, 24th-28th March 2003
- F.M. Brochu et al., "EDG integration and validation in the framework of ATLAS Data Challenges", in Proceedings of "e-Science All Hands Meeting 2004", Nottingham, 31st August - 3rd September 2004

Contact: cam-grid AT hep.phy.cam.ac.uk

O: CERN Grid Café

P: White Rose Grid

Q: OxGrid

R: Abaqus

S: Fluent

T: Autodesk

U: MATLAB Distributed Computing Server

V: MATLAB Parallel Computing Toolbox

W: COMSOL

COMSOL   HOME PRODUCTS SHOWCASE EVENTS SUPPORT COMMUNITY COMPANY

Running COMSOL in parallel and cluster mode

Browse by Category

Solution Number:	1001
Title:	Running COMSOL in parallel and cluster mode
Platform:	All Platforms
Applies to:	All Products
Versions:	3.5, 3.5a, 4.0, 4.0a
Created:	November 21, 2006
Last Modified:	September 22, 2010
Categories:	Solver, Mesh
Keywords:	solver memory parallel smp cluster

Problem Description

This solution describes how you enable parallelization of COMSOL.

Solution

Introduction

COMSOL 3.5a and later supports parallel computations on computers with multiple processors under the shared-memory parallelization model (SMP, somtimes called multicore or multithreading), as well as distributed parallelism to nodes in a cluster. Version 3.5a supports parametric runs submitted to clusters, version 4.0 and later has distributed cluster computing support.

Below are some tips and troubleshooting on how to use multithreading and cluster distribution .

Cluster distribution, Windows and Linux

In the COMSOL Installation and Operations Guide, the installation and operation of COMSOL on clusters is outlined. A step-by-step tutorial slideshow is attached below for both Windows and Linux. Also, two example models and documentation are attached: One example of a distributed parametric sweep (one parameter per node), and one with a distributed solver.

Troubleshooting cluster installations

If you get error messages, make sure that the compute nodes can access each other over tcp/ip and that all nodes can

X: Blender