COVER SHEET

Experiences with a Grid Gateway Architecture Using Virtual Machines

David Bannon[10], Rajesh Chhabra[6], Paul Coddington[7,8], Daniel Cox[7], Frank Crawford[1], Rhys Francis[4], Gerson Galang[7], Graham Jenkins[10], Marco La Rosa[9], Steve McMahon[2], Terry Rankine[5], Robert Woodcock[3], Ashley Wright[6]

1. Australian Centre for Advanced Computing and Communications, Australian Technology Park, Eveleigh, NSW 1430, Australia
2. Australian National University Supercomputer Facility, Canberra, ACT 0200, Australia
3. CSIRO, ARRC, 26 Dick Perry Ave, Kensington, WA 6151, Australia
4. CSIRO, 700 Collins Street, Docklands, VIC 3008, Australia
5. iVEC, ARRC, 26 Dick Perry Ave, Kensington, WA 6151, Australia
6. Queensland University of Technology, Brisbane, QLD 4001, Australia
7. South Australian Partnership for Advanced Computing, University of Adelaide, SA 5005, Australia
8. School of Computer Science, University of Adelaide, SA 5005, Australia
9. School of Physics, University of Melbourne, VIC 3010, Australia
10. Victorian Partnership for Advanced Computing, 110 Victoria Street, Carlton South, VIC 3053, Australia

Abstract

The Australian Partnership for Advanced Computing (APAC) began developing the APAC National Grid in 2004. The APAC Grid integrates several partner sites, most of which have multiple compute resources. Different APAC grid application projects require different grid middleware systems, including GT2, GT4 and LCG. In order to provide these different systems to interface to different resources at each site, it was decided to provide a single, standard grid gateway machine at each site, and to use Xen to provide a number of virtual machines to run the different grid middleware stacks, as well as other services such as grid portals and data management. In this paper we discuss the design of this system, and our experiences in deploying and using virtual machines on a single grid gateway machine for interfacing to multiple clusters at a site.

1. Introduction

The Australian Partnership for Advanced Computing (APAC) provides advanced computing and grid infrastructure and services for Australian researchers. APAC is a partnership of six state-based organisations representing the universities in the state, CSIRO (Commonwealth Scientific and Industrial Research Organisation) and the Australian National University in Canberra. APAC is a federally funded body and operates a large national supercomputer facility, currently a 1928 processor SGI Altix system that ranked 26th in the June 2005 Top 500 list. Each partner serves its own community, and each has different structure, requirements and obligations. Importantly, from a Grid perspective, each partner has different hardware, different software stacks and different security and usage policies.

APAC instigated a grid computing program at the start of 2004, and this has led to the establishment of the APAC National Grid [1], which currently interfaces to 12 different sites, providing a total of over 3 PBytes of data storage and 4500 CPUs across more than 25 parallel computers. Grid Computing is seen as a means to make considerably more High Performance Computing (HPC) capacity available to research groups that need very high capacity and, at the other end of the scale, it is seen as a means of allowing users who would not, in the normal course of their research, use HPC at all. It does this by hiding the complexity and diversity of the various HPC systems behind a standard Grid interface, usually based on the Globus Toolkit in one form or another.

The APAC Grid program is broadly divided into two components, Infrastructure Projects and Application Support Projects. The Application Support projects have focused on delivering discipline or application-specific tools, services and portals to a group of nominated scientific

fields, consistent with Australia's overall research interests and expertise. These include astronomy, bioinformatics, chemistry, earth systems science, geoscience, and high-energy physics. The Infrastructure Projects have been charged with uniting Australia's existing and diverse research HPC facilities. Unlike some grids being established around the world, the APAC Grid is not being funded from the top down, it is being built from existing resources up.

The APAC Grid currently provides a functional grid infrastructure, mostly based on the emerging world standards for grid computing, but along the way it has been necessary to develop a number of innovative solutions to problems that were encountered, some expected and some not. Chief amongst these innovations is the concept of a single Grid Gateway machine at each site, which supports multiple grid middleware stacks by using Xen Virtual Machines [11]. The Gateways are a very effective way to solve some of the problems that are particular to the APAC Grid, but are likely to be of more general interest.

Gateways are attractive in a general Grid for a number of reasons, including (but not limited to) being able to support a number of different grid middleware stacks at the same time, allowing 'risky' grid systems to run without risking a large cluster's stability, and the ability to clearly identify potential points of security risk.

The use of Virtual Machines to implement Gateways is a significant step. It can be demonstrated that most of the APAC Grid Gateway infrastructure would not be practical, cost effective or manageable with out the use of virtualisation. Further, Virtual Machines provide a very useful test environment, an existing machine can be duplicated in a matter of minutes to try out a new idea or technology. There is also a substantial saving in capital, power and floor space. Additionally, the dangerous idea of 'spare' machines left running but forgotten about can avoided, one less target for hackers.

The combination of Gateway Machines and Virtualisation has made it easier to roll out a consistent and technically correct system to remote sites where varying levels of skills and experience with this sort of technology are available. Also, the self-contained nature of virtual machines and the one job one machine approach has allowed a high level of quality control and system dependability.

2. Architecture of the APAC National Grid

The design of the APAC National Grid confronted a major issue when selecting grid middleware. Amongst the APAC Grid users there are several groups that have been working on various Grid solutions long before the APAC Grid came into existence in early 2004, and had already developed or were using existing software based on Globus Toolkit (GT) 2.4. Other groups were involved with with different international activities that were deploying different standards and middleware stacks, including NorduGrid and LCG. A number of people were excited about the web services capabilities of GT3. At the time GT4 was expected soon, and much discussion was taking place about the superior design it exhibited and the desirability of moving our efforts there.

It was apparent very early in the program that the necessity of supporting this very diverse group would make for a different grid than many of the others in existence at the time. It was decided that the APAC Grid needed to initially provide a GT2 implementation for existing applications and early adopters, and then move to a GT4 and/or gLite based implementation, with the likelihood of a significant period of overlap where many or all of these different flavours of grid middleware would need to be supported.

2.1. Grid Gateways and Virtualization

Systems Administrators at a number of partner sites expressed concern about the prospect of installing multiple versions of the Globus Toolkit and related grid middleware onto system critical head nodes or management nodes of multiple clusters under their control. Our grid deployment tests early in 2005 indicated that some grid middleware represented a significant threat to the stability of such machines, particularly some of the GT4 beta releases. Further, the development nature of the grid work meant that machines used for this purpose were often being rebooted as systems staff tried different software stacks and kernels. Characteristics such as this are totally unacceptable to good systems practice and system administrators at partner sites were unwilling to deploy grid middleware at their sites unless an alternative was found. There was also concern about opening up the firewalls protecting the clusters in order to enable access to the large number of ports required by the grid middleware (particularly GT2). Also, some grid middleware was only validated on limited or indeed fully specified versions of the underlying operating systems, and partners were unenthused about having to convert clusters to a different operating system in order to install such software.

It became apparent that a separate machine, apart from the cluster head or management node, was needed to be the grid face of a cluster. By providing a separate, standalone machine to host the grid middleware, most of these problems could be controlled. We therefore adopted the approach of a grid gateway machine that would host all grid middleware. This machine would be similar to a head or management node but would be reserved for Grid use only.

Early tests identified a number of incompatibilities and potential conflicts between installs of various versions of Globus. Along with the need to install VDT with the GT2 install, this meant that we really required separate machines

for each different version of the Globus Toolkit.

It was suggested that the use of virtual machines based on VMWare was a way to consolidate these separate machines back into one physical box, but cost and license restrictions on VMWare raised doubts about the viability of this approach. Xen was receiving some publicity at the time, and after some initial investigation and tests, a decision was made to base the APAC Grid Gateways on Xen.

The APAC Grid gateway machines use a physical computer supplied by APAC to each partner. The machine is located at each partner's site and managed by local Systems Administrators. The hardware configuration for the gateway machinemachines is an IBM dual 2.8GHz Intel Xeon system with 4GB RAM, 300GB SCSI mirrored disk, dual power supplies and at least 5 GigE NICs. The NICs are allocated as one for management, two (one for communications internal to the site, and one for external) for NGdata (the data management virtual machine) and two shared among the other virtual machines.

The Gateway machines currently use Xen 3 and run CentOS 4.3 clients, with the exception of some VMs running Scientific Linux in order to run software such as LCG and VOMRS that is only supported on that OS. Most of the Virtual Machines and the underlying Xen Dom0 use Red Hat's RPM (Redhat Package Manager) to install and maintain the images.

2.2. Virtual Machines on the Gateways

APAC needed to support three, and potentially more, middleware stacks in order to meet the requirements of the APAC grid application projects. A number of other functions could also sensibly be separated out onto their own Virtual Machines. Once the roles are separated, the management can be also separated, allowing for an improved build and deploy structure.

In this model the middleware, combined with Xen, exists as a complete machine, and makes the concept of a gateway possible. One physical box provides, using virtual machines, all the middleware stacks connecting to the resources offered at a given site.

The APAC National Grid (NG) identified the following Virtual Machines (VMs) as being needed at some or all of the sites. The capabilities and installation of these VMs is carefully defined within the APAC Grid and sites are intentionally restricted in variations allowed.

NG1 – GT2 and VDT
Provides a Globus Toolkit 2 (GT 2.4.3) system, based on the Virtual Data Toolkit (VDT). Capable of accepting jobs and submitting them to one or more associated clusters, using PBS (or the local queuing system) client tools. Can run GridFTP.

NG2 – GT4
Provides a Globus Toolkit 4 (currently GT4.0.2) system capable of accepting jobs and submitting them to one or more associated clusters, using PBS (or the local queuing system) client tools. Can run GridFTP.

It is much easier if the NG1 and NG2 VMs mount user file systems using e.g. NFS, which avoids many problems with data staging and with Globus assuming that it can write information to the users home directory.

NGLCG - LCG Compute Element (CE)
This VM is deployed to enable the Australian High Energy Physics community to effectively utilize the resources of the APAC National Grid along with the resources of the Worldwide LHC Computing Grid (WLCG) [10]. It provides an LCG 2.6 interface to the available computing resources with the tools required to access LCG storage resources and data catalogues.

The LCG CE VM is built on Scientific Linux 3.0.6, and consists of job management and compute resource interface tools (VDT-1.2), workload management and authorisation components (LCAS, LCMAPS) originally from the European Data Grid project and tools developed as part of the LCG/gLite middleware projects (APEL, RGMA, GLUE, data management and file catalogue tools).

This VM differs from the NG1 and NG2 VMs in the integration of its components as a part of a larger whole. Specifically, effective use of all of the facilities of the VM requires the LCG user interface component to interact with the LCG resource brokering component which mediates user access to compute and data resources based on job descriptions written in the Condor ClassAd language.

The LCG CE middleware assumes that it interfaces to only one cluster. Deployment of the VM in the Australian model (one gateway per site, which may have multiple clusters) required some alterations to the PBS job manager and the scripts used to gather information and populate the information system. Other than these small modifications, the deployment of the CE in a VM was no different to installation on a dedicated host.

NGData – Data management services
A GT4 (i.e. NG2) based VM that provides standardised and encapsulated data staging services, handles management of grid data transfers at grid sites, and enables high performance data transfers, using GridFTP and client software including SRB, uberftp and scp. Data can be transferred to and from it directly, and since it mounts a site's cluster file system, this data is available to compute jobs submitted to that cluster (e.g. from NG1 or NG2). End users or end user applications can also use it to manage data staging by submitting data staging jobs to it. These jobs could be submitted directly from a site's cluster or through the grid job submission system. It could also be part of the data staging mechanism for GT4 jobs submitted to a site through the NG2 virtual machine.

NGPortal – Web Portals
NGPortal is based on the NG2 VM with Tomcat, Java and GridSphere. It does not queue jobs to the compute resources (e.g. clusters) itself, it instead forwards jobs to a site's NG1 or NG2 VM. This has the added the benefit that a given portal is able to submit jobs to any site, using either GT2, GT4, or a combination of the two. The main purpose is for hosting Grid Portals, however it also provides a consistent environment for the software developers creating the Grid Portals.

By creating a core standard environment at each participating site we managed to decrease the time taken in troubleshooting problems by accessing a collective knowledge base of the system administrators and software developers at different sites. This makes it easier for developers to be sure their products will work when deployed at the many sites. We discovered early in the Grid Portal development process that a robust Grid Portal environment relied heavily on having the right combination of the various versions for Tomcat, Java and GridSphere toolkits. Currently the following additional software is supplied on top of what the NG2 VM provides Java SDK 1.4.2, Apache Tomcat 5.5.12 and Apache HTTP 2, GridSphere 2.1.1 and GridPortlets 1.2.

This VM also produces the Gridmap file used by the other VMs in the Gateway. It obtains data from VOMRS and software that we have developed called the AuthTool, which allows a user to log into a secure web page on NGPortal, and create a direct mapping for their certificate to a local site user.

Grid – grid job submission node
A VM very similar to NG2 but without the capability of directly submitting jobs to the clusters (i.e. the GRAM is not run). Provides an appropriate platform for users who wish to use GT4 command line client tools to launch jobs, and is trusted by the other APAC Grid sites.

VMs for testing and development
Additionally, sites may have 'test' or 'dev' versions of the above that are not subject to strict compliance and uptime requirements. Offering production services increases the restrictions placed on the changes made to services, subsequently there is a need for active development boxes.

Users are able to use the testing machines if they wish and the stable machines, NG2, NG1, are not affected. The development services are not guaranteed to work, however the prototyping style of development gives the user a system to play with that can be offered before the service is fully developed. With any software cycle, feedback is essential, and these development and testing machines provide faster iterative interaction between developers and users.

VOMRS – A Virtual Organisation Manager
A Scientific Linux system that runs VOMRS. VOMRS is derived from VOMS and provides a means of mapping Grid users to local users where the Grid user does not have a local account. Groups or projects are organised into Virtual Organisations (VOs) that are under control of nominated project leaders. The grid application project leaders can accept or remove users from their own project and control their activities within it. Currently there is one VOMRS server for the APAC Grid.

MyProxy – Grid Proxy Servers
There are two MyProxy virtual machines, located at QUT and VPAC to provide a level of resilience. They issue proxies, or time limited certificates, to allow jobs to run and interactions to take place between third parties on the users behalf.

Currently all of the job management VMs (NG1, NG2, and NGLCG) rely on Gridmap files (as acquired from NGPortal) to map grid credentials to a local user, but we are aiming to move to using GUMS/PRIMA [2, 3] processes soon.

Figure 1 illustrates the APAC National Grid architecture, with each APAC partner site hosting a single grid gateway machine to interface to its compute resources (most sites have more than one resource). Each grid gateway machine runs a subset of the standard VMs listed above. Usually sites run a single instance of each VM to interface to all their compute resources, however some sites run one instance per compute resource, since this makes it easier to manage systems that do not share user file systems.

3. Experiences with Gateways and VMs

Initially it was intended that the VMs at each APAC partner site would be kept up to date by distributing new VM images. The plan was for a central group to build VM images and ship these to Partner Sites where local Systems Administrators would make necessary localisations. Xen is very suited to this approach, the VM is built, the various components installed and configured and then the VM can be stopped. A copy of the VM Image can easily be brought up elsewhere and will almost certainly be fully functional.

However, this approach proved difficult for a number of reasons :

- GT4 wants to know things like the name of the host at build time and it is then hardwired into the Globus installed system.
- Local Systems Administrators and security authorities were not happy with the "black box" approach and needed to understand the build process and what was contained in the image.
- The list of necessary changes to the image at deployment time grew beyond what was a reasonable task. As every site had a different list it was deemed impractical

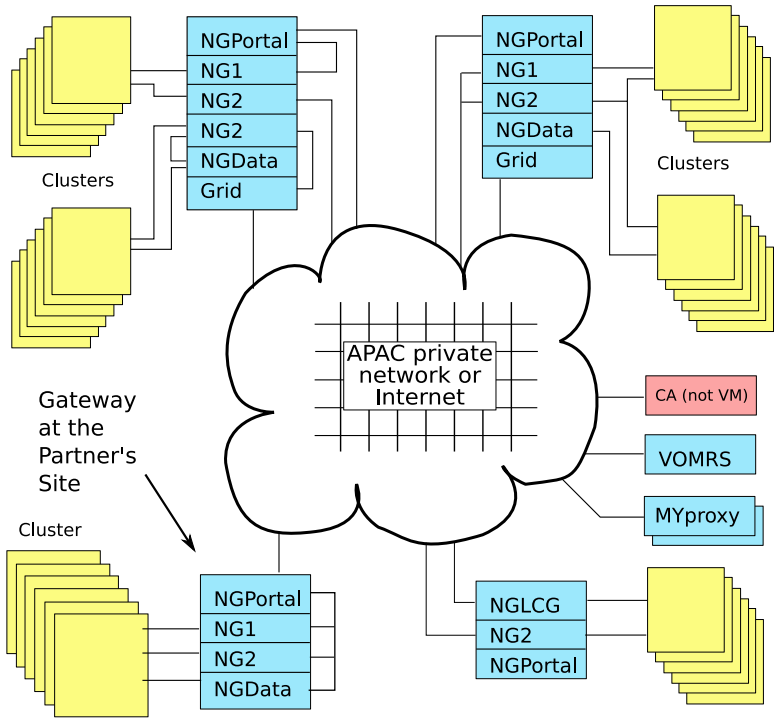

Figure 1. APAC Grid Architecture

to maintain a universal script and locally maintained scripts would need updating with almost every release.

- It was found that updates to the VMs were frequent but usually minor and did not justify a complete rebuild.

The GT4 build was, after several other methods were rejected, standardised as being RPM based. All updates and alterations to the VM are made with RPM. Each RPM package deals with configuration files and scripts by placing a new but renamed version in place allowing a local Systems Administrator the opportunity to compare the old with the new, and apply, selectively if necessary, the necessary changes. The same RPMs and scripts to configure them can be utilized to build a new VM from scratch in only a very few steps and minimal time.

Due to the success of this approach with the GT4 VM, the other VMs are now being built in a similar way. At the time of writing, new updated versions of the install scripts and RPMs are being released every couple of weeks. Updates at each site are coordinated so that the same VMs at different partner sites provide a uniform interface and functionality to users. The grid gateway administrators at each partner site meet weekly via Access Grid to coordinate updating and deployment of the VMs.

3.1. Advantages of the Gateway and VM Approach

The use of gateway machines and VMs resolve the following issues:

- A limitation on the number of systems that need grid middleware installed and managed within the APAC Grid sites, thus reducing overall grid management overheads.

- Grid connectivity can be made available for applications that are dependent on middleware such as LCG that is not supported at all sites, by simply adding a required VM at a particular sites.

- Enhanced security, as many grid protocols and associated ports only need to be opened between the various gateway machines, rather than directly to all the clusters at each site. Only the local gateway needs to interact with the site systems. Since the gateways are local to the site, security in and out of the gateways can be closely monitored. All interactions that happen between the gateways and the sites (log monitoring, migrating data, etc) are kept internal to that site. Security and firewall issues do not go away, but they can be reduced using a gateway approach.

- Different grid toolkits can be incompatible with each other, however each of these can run in their own virtual machines.
- Providing support for the roll-out and control of production grid configuration through the implementation of standardised grid support across all APAC Grid sites.
- Providing support for production and development grid interfaces and local experimentation without significant hardware investment, through a virtual machine implementation where different services and different quality of service are provided on separate installations. Virtualization has allowed machines to be restarted/rebuilt/updated (or even crash) without interrupting other grid services or middleware. With this level of separation, running experimental, or highly unstable code is possible.
- Monitoring and benchmarking – as processes are isolated from each other, it is relativly easy to measure the resource usage and performace of a specific middleware component.
- Cost – it is estimated that the APAC grid would need to provide between three and five times as many physical machines to provide a similar level of isolation between applications if virtualisation had not been used. Apart from the initial capital cost, long term maintance, power and physical space must be considered.

Many of these issues are applicable to many other projects and implementations.

3.2. Modifications Required by the Gateway Approach

As much as possible, the APAC Grid has endeavoured to minimise changes to its Grid Middleware. This is to ensure compatibility with other grids and future releases of the middleware itself. However, the VM-based grid gateway model has required some modifications, primarily due to certain aspects of the middleware assuming that it is being run on the head node of a cluster. Some modifications to grid middleware are to be expected in just about any Grid situation, and we believe the number required to support our gateway architecture is not excessive, and the modifications are, for all practical purposes, undetectable by end users.

Globus Toolkit

One major issue is that GT4 expects to be installed on the management node of a PBS based cluster. As such it expects to be able to watch PBS's log file to see a job's progress through the queuing system. Some partner sites have choosen to export their log files via NFS (read only) and others use a tool, pbs-telltail, initially written by Damon Smith (VPAC) and rewritten to provide greater reliability by Graham Jenkins (VPAC). 'pbs-telltail' runs on a PBS management node and sends relevent entries to the appropriate gateway.

Considerable modifications have been made to the Globus PBS module, pbs.pm but its difficult to say that these have been because of the Xen VM architecture, in most cases they have been made to meet the APAC Grid's other business needs.

Generic Information Provider

VDT uses the Generic Information Provider as a tool to generate LDAP based grid information that is used as an input to MDS2. The GIP comes with scripts that can be used to communicate with a number of information sources. Most of the APAC partner sites use PBS as their resource manager and the GIP's PBS script can be used to publish PBS dynamic information.

However, GIP assumes that it will be installed on the head node of the cluster where it has direct access to the local information sources also running on the cluster. When installed on a Gateway VM the GIP scripts need to be modified to make them query a remote PBS server.

The naming convention used for the computing elements (which represent queues) in the GLUE schema will also not work with the gateway system. Since a number of clusters might be running behind the gateway node, the names of the computing elements should reflect the names of the clusters these queues are running on, however this is straightforward to deal with.

Nimrod/G

Nimrod/G [4] is a specialised parametric modelling system that lets the user run distributed parametric modelling experiments. Many APAC users make use of Nimrod to run a wide range of scientific and engineering appliations.

For a Nimrod/G experiment to get executed, it requires Nimrod/G Agents to run on the actual resource which will execute the job. These resources are usually the compute nodes of the cluster. The Agents need to communicate the status of the job back to the Nimrod server. Since most compute nodes of clusters belong to a private network and cannot access any machines outside of this network, a Nimrod/G proxy needs to be started on the head node of the cluster. The proxy serves as a data relay for communication between the Agents and the Nimrod server. The Nimrod/G developers redesigned the way this was done to ensure it could function in the gateway model. Another way around this problem is to run NAT on the head node of the cluster, which is a common approach for providing the compute nodes with access to external machines.

3.3. Issues with Using Xen

Previous tests on Xen has shown that it has a very low overhead in general [12], and in particular a low overhead for running grid middleware [7, 6]. We have not found any performance problems apart from the few issues mentioned in this section.

Xen 2 vs Xen 3

Originally the APAC virtual machines were built using Xen 2.0.7 technology, which had an older kernel (2.6.11) and had a few issues that were hindering performance (particularly I/O performance). This led us to upgrade to Xen 3, which is more stable, provides a newer kernel, and dedicated hardware allocation (particularly for NICs, which helps with I/O performance).

Network Throughput

Extensive testing of the network interfaces, particularly under high load has demonstrated a tendancy to crash the network stack on a VM. This was initially detected in Xen 2, systems upgraded to Xen 3 can also suffer the same thing but at much higher load levels. The same hardware and a software stack lacking only Xen is not subject to the same problem. Examination of the underlying cause has proven difficult because practical network links between sites are not routinely capable of delivering the throughput necessary to cause the problem.

Threads and CPU management

Xen 3.0 selects CPUs or hyperthreads for a VM on creation and does not dynamically switch them to the least busy processor on the dual processor gateway machine. This could cause problems, for example if NGPortal is sending jobs to NG2 and these two VMs are both assigned to the same CPU, or if NGData is doing a large data transfer. One approach to this problem is to specify a particular CPU or hyperthread to each VM, for example on a dual processor system with hyperthreading the gateway OS will show 4 CPUs that can be allocated, and a possible allocation would minimize the problems mentioned above would be CPU0 reserved for Dom0 only; CPU1 shared by NG1 and NG2; CPU2 NGPortal; CPU3 NGData.

The Native POSIX Thread Library (NPTL) is emulated in Xen with a warning message displayed when first used, so it is normally disabled. But this severely affects the performance of multi-threaded processes such as Java applications. It is possible to rebuild glibc in order to avoid this problem, and XenSource provide downloads for Red Hat Enterprise Linux containing replacement glibc RPMs.

Memory issues

32-bit Xen 3.0 makes only 3328MB available from the 4GB of RAM installed. Compiling the hypervisor with Physical Address Extension (PAE) available in Xen 3 and the Linux Dom0 kernel with 64GB high memory support enables the extra memory without having to use 64-bit. This is necessary to be able to allocate 512MB to the VMs running Java and still have a development version of the VM running on the same machine.

We have found that the Sun JDK (1.4 and 1.5, including the -server VM) often uses much more memory than specified (using -Xmx), resulting in very high swap usage which increases over time. Sun JDK 1.6 (still in beta) and BEA JRockit are being looked at as an alternatives to try to reduce this problem.

4. Related Work

A very similar approach to grid deployment, using a single grid gateway machine hosting multiple virtual machines, has been adopted by Grid-Ireland [7, 8, 9]. However their system places different components of the LCG middleware onto different virtual machines on the gateway, whereas our approach aims to support multiple grid middleware stacks, including GT2, GT4 and an LCG compute element, as well as additional VMs for things such as grid portals. Another difference is that the Grid-Ireland gateways machines are centrally managed and remotely installed, whereas in our approach each site installs, manages and configures its own gateway machine and associated VMs. We only became aware of the Grid-Ireland work after we had come up with the same basic idea of using gateways and VMs for the APAC National Grid.

Hardt et al. [5, 6] have also explored the idea of implementing an LCG system using multiple VMs on a single machine.

5. Conclusion

When the architecture of the APAC National Grid was being designed in 2004, it became clear that multiple grid middleware stacks needed to be deployed in order to support the different grid applications projects, however this needed to be done with quite limited systems administration resources and grid expertise at each APAC partner site. There were also significant concerns about firewall issues, security and robustness of the grid middleware, and the amount of effort it would require to install the different grid middleware stacks on the multiple clusters managed by most partner sites.

These considerations led to experimentation with the concepts of a single grid gateway server that would provide the grid interface to a site's compute resources. The availability of virtual machine technology meant that gateways could be partitioned into separate VMs for each middleware stack.

Implementing such a system has taken significant time and effort, but probably less than would have been needed

to deploy the grid using a more standard approach. The most significant difficulty has been the assumption, built into many grid middleware services, that they reside on the compute clusters. We have raised this issue with the developers of some of the middleware, and they agree that this is an inappropriate assumption. It has been reasonably straightforward to develop workarounds to address the various problems caused by this assumption.

Our experience has been that this approach is a fundamentally sound idea, in that the APAC Grid sites how provide simultaneous access to the same resources through GT2, GT4 and LCG middleware and services, in contrast to many other existing grids. Adding a new middleware stack is a reasonably straightforward exercise now that the underlying architecture has been deployed at each site. The benefits of this approach include reduced cost, complexity and management overhead, and increased flexibility and security.

Our experience with using Xen virtual machines has been so positive that other VMs (e.g. for web portals and data management services) have been defined, and many APAC partners are now using VMs for their own local services. We have experienced a few problems with using Xen in a grid gateway environment, however these are relatively minor and there are workarounds for most of them. Overall we believe that the use of Xen virtual machines to provide multiple grid middleware interfaces and services through a single grid gateway machine has enabled the Australian National Grid to provide users with a functional and flexibile environment, while reducing the amount of systems support required to deploy and manage the grid infrastructure and services.

6. Acknowledgements

Funding for the APAC National Grid Program was provided by the Australian federal government under its Strategic Infrastructure Initiative.

We would like to thank the systems adminstrators at each APAC partner site who helped to deploy and troubleshoot the grid gateways at each site, the members of the Compute Infrastructure, Information Infrastructures and Portals projects who helped develop and test the VMs, and the members of the Grid Applications projects who provided requirements for the system, helped test it, and gave invaluable feedback and suggestions.

References

[1] APAC National Grid, http://www.grid.apac.edu.au/

[2] Grid User Management System (GUMS), http://grid.racf.bnl.gov/GUMS/

[3] PRIMA (PRIvilege Management and Authorization), http://computing.fnal.gov/docs/products/voprivilege/prima/prima.ht

[4] Nimrod: Tools for Distributed Parametric Modelling, http://www.csse.monash.edu.au/ davida/nimrod/

[5] Marcus Hardt, Ruediger Berlich, Xen Grid Site - the Art of Consolidation, http://public.eu-egee.org/files/xen-grid-in-a-box-fzk.pdf

[6] Marcus Hardt, Ruediger Berlich, Xen: Scientific Use Cases and Performance Comparisons, http://www.ep1.rub.de/ ruediger/xen-grid-in-a-box-ukuug.pdf

[7] Stephen Childs, Brian Coghlan, David O'Callaghan, Geoff Quigley, John Walsh, A single-computer Grid gateway using virtual machines, Proc. AINA'05, Taiwan, March, 2005.

[8] Stephen Childs, Brian Coghlan, David O'Callaghan, Geoff Quigley, John Walsh, Deployment of Grid gateways using virtual machines, Proc. EGC'05, Amsterdam, February, 2005.

[9] Brian Coghlan, John Walsh, and David O'Callaghan, The Grid-Ireland Deployment Architecture, Proc. EGC'05, Amsterdam, February, 2005.

[10] LHC Compute Grid, http://lcg.web.cern.ch/LCG/

[11] University of Cambridge Computer Laboratory, Xen, http://www.cl.cam.ac.uk/Research/SRG/netos/xen/

[12] P.Barham *et al.*, Xen and the Art of Virtualization, Proc. 19th ACM Symposium on Operating Systems Principles, ACM (2003).