Introduction to the Cluster Grid – Part 2

James Coomer, Solutions Technology Group, UK

Charu Chaubal, Grid Computing

Sun BluePrints™ OnLine—September 2002

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 816-7765-10
Revision A, 9/12/02
Edition: September 2002

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95045 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Grid Engine, Sun Grid Engine Enterprise Edition, Sun HPC ClusterTools, SunVTS, Sun Management Center, Sun ONE, Sun Cluster Runtime Environment, Java, Sun HPC ClusterTools, Sun StorEdge, Solaris JumpStart, Sun Cluster, Sun Fire, CacheFS, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95045 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque enregistree aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Sun BluePrints, Sun Grid Engine, Sun Grid Engine Enterprise Edition, Sun HPC ClusterTools, SunVTS, Sun Management Center, Sun ONE, Sun Cluster Runtime Environment, Java, Sun HPC ClusterTools, Sun StorEdge, Solaris JumpStart, Sun Cluster, Sun Fire, CacheFS, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Please
Recycle

Introduction to the Cluster Grid – Part 2

This article is a follow up article for the Sun BluePrints™ OnLine article titled "Introduction to the Cluster Grid – Part 1", which provided a description of a cluster grid and the architecture of the Sun Cluster Grid stack.

This article takes the next step by describing the Sun Cluster Grid design phase. The Sun Cluster Grid design process involves information gathering followed by design implementation.

Information gathering involves defining the type and extent of service provision and the type and mix of applications to be supported by the Sun Cluster Grid. The former impacts the access and management tiers, while the latter primarily impacts the design of the compute tier.

This article is intended for IT professionals, system administrators, and anyone interested in understanding how to design a Sun Cluster Grid.

Information Gathering

This section describes the services that can be implemented in the three logical tiers of the Sun Cluster Grid. Use this information in the information gathering stage when planning your Sun Cluster Grid.

Service Provision

The choice of which cluster grid services to provide depends on various factors such as the requirements for security, service availability, manageability and scalability. The only essential component of a Sun Cluster Grid is the distributed resource management (DRM) software, which provides a single point of access for job submission, and controls the compute environment for the cluster grid. Additional service provisions can be implemented in the tiers as follows:

- **Access tier—**Authentication, web-based access, administration
- **Management tier**—High availability, health monitoring, install management, hardware testing, NFS, license key management, backup management
- **Compute tier—**MPI runtime environment, other runtime libraries

For each tier, the various services are discussed and reasons for providing (or not providing) the services are given.

Access Tier

The authentication schemes vary widely between implementations. In many cases, the cluster grid will run under an existing authentication scheme, so a new authentication scheme need not be implemented as a cluster grid service. In such cases, access to the cluster grid service can still be restricted by the administrator. The Sun™ Grid Engine (SGE) software integrates with the authentication services by automatically checking user credentials at job submission time. Access to the cluster grid can be restricted by explicitly denying (or enabling) user or group access to Sun Grid Engine software, or just certain SGE queues.

If no web-service provision is needed, all access to the cluster by users is usually through a SGE submit host. In some cases, the administrator designates users' desktop machines to be submit hosts. Alternatively, the submit hosts could be physically under the control of the administrator (for example, in a secure data center) and accessible to users using commands such as `telnet`, `ssh`, and the like. The latter case would apply, for example, if the cluster grid is always accessed remotely, or if the cluster grid exists under its own authentication scheme.

Management Tier

At a minimum, this tier includes the SGE master. Other services can be implemented to provide increased reliability, availability, and to simplify management.

Health monitoring services are provided by Sun™ Management Center (SunMC) software. A minimal installation of SunMC software does not require agents to be installed on any hosts. In this case, the SunMC server still reports *alive* status for hosts on the network using SNMP ping. Installing the SunMC agent is particularly useful in maximizing the availability of management nodes that provide vital cluster grid services, and of NFS servers, or large SMP nodes. The agents report detailed information and can email the administrator to warn of low disk space, high swapping rates, or hardware failures and errors, and can be programmed to take prescribed corrective measures automatically. In a compute intensive environment, the decision to employ SunMC agents on smaller compute nodes depends on the perceived benefits given the inevitable small load that the agent introduces.

In large compute environments, particularly when based on large numbers of thin-node compute hosts, the Solaris Jumpstart™ environment should be used to facilitate installations. Custom scripts can be written to perform complete automated installations of new compute servers. Where large numbers of identically configured systems exist, Solaris™ Flash software becomes very efficient, allowing a single execution server image to be copied onto new servers in minutes. This is discussed in more detail in “Solaris Jumpstart and Flash Software” on page 16.

Both the Sun Grid Engine, Enterprise Edition software and the open-source version of Grid Engine provide facilities for multi-departmental control. These should be chosen rather than Sun Grid Engine standard edition if advanced accounting and share-based resource allocation is needed. If there are plans to allow access to Sun Grid Engine software for external regional, national, or global grid users, then the share-based scheme enables external use of the cluster grid to be tightly controlled by the local administrator.

The archiving and backup strategy usually does not encompass all the available storage in a cluster grid. Thin node hosts in the compute tier usually just hold temporary data associated with running jobs and are usually excluded from backups. If the cost of backup hardware and software is to be minimized, users may be allocated a limited storage space for home directories and a larger *working directory* for day-to-day use, which is not backed up.

For business critical implementations of the cluster grid, some high availability can be built in. High availability features can be implemented through HA clustering software such as Sun™ Cluster 3.0 software. The DRM and NFS services are usually the first candidates to be supported by an HA solution.

Compute Tier – Application Support

The user applications that are to be supported by the cluster grid strongly affect the design of the compute tier. Both current and future applications should be characterized as well as possible. For each application that is to be supported, at least approximate answers to the following questions should be gathered:

- Is the application a single-threaded, multi-threaded, or multiprocess application?
- What data access patterns are expected?
- What are the memory requirements?
- What is the average runtime?
- If the application is multiprocess, which message passing approach is implemented?
- How does the application scale?

Also, information on how the applications are used from day-to-day can be important. For example, some applications in development and research environments require multiple test runs before final submission. In this case, it might be wise to provide some machines dedicated for interactive use.

Often, the cluster grid workload falls into one of the following categories or some superposition of them:

- Throughput—Characterized by maximizing the execution rate of large numbers of independent, serial jobs.
- On demand—Characterized by maximizing day-to-day utilization while enabling high priority jobs to execute on demand.
- Highly Parallel—Characterized by minimizing the execution time for relatively small numbers of multiprocess jobs scaling beyond ~10 processors.

Design Implementation Decisions

In this section, the information gathered is translated into recommendations and requirements for the Sun Cluster Grid design. For each tier of the cluster grid, the impact of the information gathered is assessed.

Access Tier

At a minimum, the access tier has to be sized to support logins, telnet sessions, and job submissions (running simple binaries). Non-interactive jobs are executed through the queueing system by submitting a simple batch shell script which acts as a wrapper to the executable, and can be used to pass information to the queuing system, and to perform simple setup or cleanup tasks.

If users' own workstations are used to submit jobs, this represents a negligible load. However, if a single system is required to support hundreds of remote logins, it would be wise to ensure some dedicated resources.

Administrative access to the cluster grid is provided through the Sun Grid Engine and SunMC software. Both can be administered through a command line or GUI interface. Sun Grid Engine binaries and the GUI are supported on both Solaris™ and Linux operating environments. Binaries can also be obtained for other operating systems from the opensource site (see “Related Resources” on page 27).

The SunMC console, based on Java™, is supported on both SPARC systems running Solaris Operating Environment software (versions 8, 7 and 2.6) and Intel-based systems running Microsoft Windows 2000, NT 4.0 (with Service Pack 3 or 4), 98 and 95. For SPARC systems, the minimum system requirements for running the SunMC console are: Ultra 1 (or equivalent), 256 Mbyte RAM, 130 Mbyte Swap. For MS Windows Intel-compatible systems, the minimum requirements for running the SunMC Console are: 300 MHz Pentium, 256 Mbyte RAM, 35 Mbyte free disk space.

If web-based access is to be implemented, for example, using the Sun ONE portal server, the server must be sized appropriately for the expected load; this should take into account the sessions characteristics and headroom for future scalability.

Management Tier

The decisions to be made when designing the management tier depend on which services are to be provided, and the expectations for future scalability. For small cluster grids with a minimal service provision (DRM only), a single processor

machine might suffice. As discussed in the previous article titled "Introduction to the Cluster Grid – Part 1", the master host functionality for Sun Grid Engine software is provided primarily through two daemons. Moving beyond a dedicated dual processor machine for the SGE master, therefore, results in limited performance enhancement. If a multiprocessor machine (more than two processors) is employed in the cluster grid as the SGE master, it would be appropriate for this machine to provide other services. For example, an eight-way server could act as an SGE master host, a Sun MC server, NFS and backup server as well as supporting computational tasks.

The load from the SunMC server is caused by normal management operations, including periodic data acquisition, alarm rule processing, alarm annuciation, alarm action execution, and processing of client requests. The generated load is proportional to the rate at which data is gathered, the amount of data gathered, the number of alarms detected and the number of user requests. The percentage of CPU resources consumed depends on the number and type of modules loaded on the system, the configuration of these modules, and the computational capacity of the host system. In general, even on low-end machines with a comprehensive suite of modules loaded and high management activity, the agent should never consume more than a fraction of the CPU resources. As with CPU consumption, the memory consumed by an agent depends on multiple factors.

The primary considerations are the number of modules loaded and the amount of information being monitored by these modules. Loading many management modules on an agent inevitably increases its footprint requirement. Similarly, agents managing hosts with large disk arrays or other highly scalable assets probably require more virtual memory, as the sheer volume of management information passing through them increases. In general, a base agent with the default set of modules loaded will be under 10 Mbyte in size, and under typical operation will only require 50–60% of this to be resident in physical memory.

NFS server sizing is a complex topic beyond the scope of this document. Obviously access to data files is a primary consideration, and the design is dictated by the application and type of work being done. The following list summarizes the other various elements of a cluster grid which might require a shared file system.

- Sun Grid Engine software—By default, the SGE directory structure is shared across the cluster grid so that all execution, submit, and administration hosts access the same physical database. Non-default arrangements are discussed in "Sun Grid Engine Software Installation Considerations" on page 13.
- User's Home Directories—As with nearly all DRMs, the input files and executables are arranged by the user, usually in their home directory or some working directory. When the job is submitted, by default the execution host must be able to access the files over a shared file system. Methods to minimize file-sharing network traffic are discussed in"File Sharing" on page 11.
- Sun HPC ClusterTools™ 4.0 binaries—Two installation methods are available involving either a distributed install or a centralized install.

- License Key server, application servers—Accessing application binaries from a shared location is beneficial because only one version of these files needs to be maintained for upgrades, bug fixes, and so forth.
- Installation servers should have access to sufficient disk space to hold multiple Solaris images, software images, and flash archives, taking into account any RAID implementations.

Compute Tier

The decision of which hardware to implement in the compute tier is based primarily on maximizing performance/price. The overall hardware profile should closely match the user application profile. The hardware profiles pertaining to the throughput and highly parallel environments are at opposite extremes.

- Throughput—Large numbers of thin nodes. Often the key requirement is to maximize the number of processors per unit volume rather than individual processor performance.
- Highly Parallel—Depending on various attributes of the application, either a smaller number of large SMP nodes or a large number of thin nodes supported by a cluster runtime environment (CRE) will be appropriate.

A typical mixed load cluster grid for an academic site, for example, might consist of:

- Distributed memory, network of workstations (NOW) interconnected with specialized high bandwidth low-latency interconnect and Ethernet.
- A number of low- to mid-range independent servers for general tasks, interactive use, large memory serial jobs, and so on.
- A large SMP system to support large OpenMP applications or other message passing applications, which benefit from ultra low-latency communications.
- Workstations in student labs with standard fast Ethernet connections to be used at night, on weekends, and holidays for smaller one-CPU jobs.

Direct attached disk space on compute nodes that are part of a NOW cluster is usually used purely as scratch space and caching.

In a throughput environment, the compute tier should provide the resource to meet demand on some timescale. In FIGURE 1, the compute tier is sized to complete the submitted jobs on a daily basis. While the rate of job submission peaks during working hours, the available compute power enables jobs to be completed by the start of the next working day.

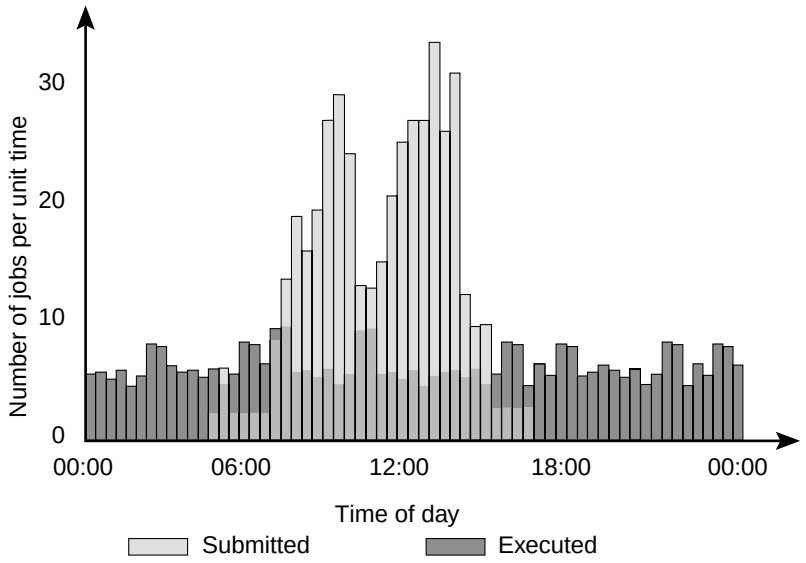

FIGURE 1 Example of Daily Profile for Cluster Workload

Memory requirements and cache — Some applications benefit a great deal from large cache processors. In such cases, the aim is to maximize the proportion of the active data that is retained in cache, giving an order of magnitude lower access times than memory resident data.

Networking Hardware

Three interconnect types should be considered: serial, Ethernet and specialized low-latency interconnects.

Serial

A serial network allows the system administrator to gain console access to all the machines in a cluster. For large environments, this is a tremendous convenience, allowing almost complete control over all systems from a single remote location. The use of a terminal concentrator gives the administrator access to multiple console ports over the Ethernet network.

Ethernet

The network load in a cluster grid will originate from a number of activities:

- MPI or PVM message passing communications at runtime for parallel applications in a distributed NOW
- NFS traffic from various sources such as the following:
  - Cluster grid services accessing binaries, spool files, and so on
  - User directories being accessed at runtime for executables, input files, and so on
  - Sun Grid Engine communications
  - Data transfer from backups and installs

The load generated by this traffic (especially if no MPI traffic is involved) is handled satisfactorily with standard Ethernet or gigabit Ethernet. Techniques for reducing network load by minimizing file sharing are covered in the section, “File Sharing” on page 11”. Ethernet capacity can be scaled through the use of multiple Ethernet cards. Shared file systems, for example, can be implemented through dedicated interfaces to separate NFS traffic from other network traffic.

Specialized Interconnects

Particular care should be made to avoid MPI traffic and other standard Ethernet traffic mixing if high communication intensity is a feature of the applications. One way to avoid this is to invest in a specialized low-latency interconnect for handling message passing communications. Alternatively, MPI applications may be run within large SMP machines where the interprocess communication takes place across the ultra-low latency specialized backplane.

Typical latencies for MPI communications between nodes over standard or gigabit Ethernet are over 100 microseconds. The maximum achievable bandwidth over gigabit Ethernet is around 700 Mbits per second. For many parallel applications, these latencies and bandwidths introduce a severe bottleneck to the calculations, and such specialized interconnects alleviate the bottleneck.

A Myrinet network is composed of interfaces, links, and switches. For Sun machines, the PCI interface is used with drivers tuned for Sun architecture, available from Myricom. Myricom supports a loadable protocol module (PM) for the Sun HPC ClusterTools 4.0 software. The PMs are used by Sun HPC ClusterTools software to carry the traffic between processes to exploit the low latency and high data rates of Myrinet. The Myrinet interconnect can reduce latencies by an order of magnitude, and can give higher aggregate bandwidths for compute clusters. Use of these specialized interconnects results in an added advantage of reduced load on the host processor as the message routing is processed in hardware on the Myrinet interface card.

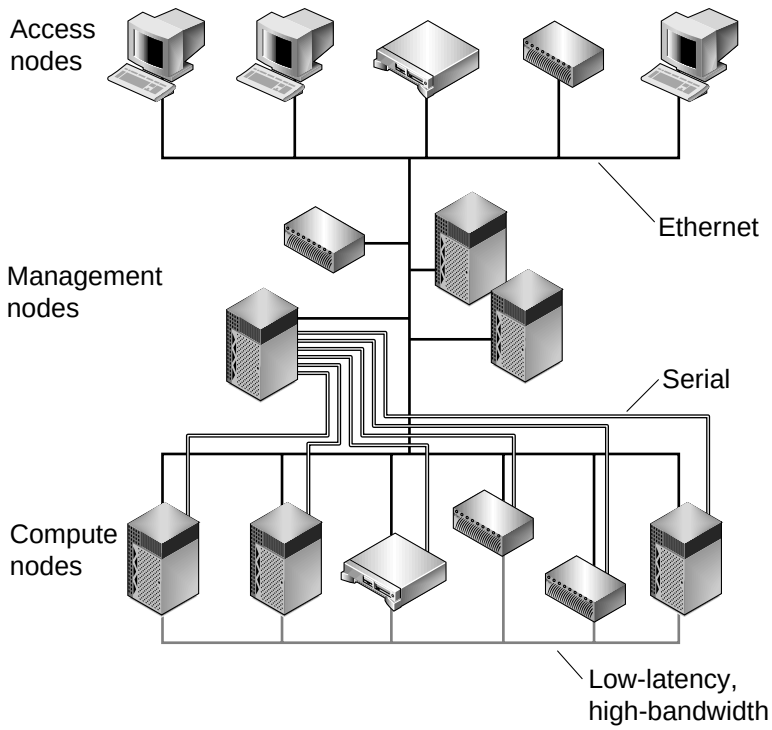

FIGURE 2 Three Types of Cluster Grid Interconnects

Storage

Full treatment of storage options is beyond the scope of this document, so a summary of the major options is outlined here.

If a large SMP server is part of the cluster grid compute tier, it might be appropriate for this server to provide the NFS service to the cluster grid. This ensures that the applications running on the large SMP have access to fast, direct storage, and provides some headroom for scaling the NFS service.

Choices of storage arrays depend very much on budget, availability, future scalability, and expected access patterns. Top-of-the-line disk arrays feature large caches, hardware RAID, and are compatible with SAN implementations. In some cases, the applications perform large volumes of random reads and writes and are better suited to a JBOD array (non-RAID storage implementations) with maximum spindle count.

Installing a Cluster Grid

In this section, the installation of the cluster grid software stack is discussed, covering major issues, options and salient points particular to the cluster grid environment. For detailed step-by-step installation instructions for the individual software elements of the stack, refer to the appropriate documentation that is supplied with the product.

The previous section described evaluating which services to employ and which applications to support. The impact of that information on the tiers of the cluster grid was evaluated. The cluster grid installation process involves making further decisions at a lower level, and might involve using some advanced installation options.

Solaris Installation Considerations

Solaris 8 operating environment installations should be performed according to the requirements of the software you plan to run on each node. If thin node compute hosts are used, the disk configurations are often configured simply to supply scratch space, space for spool files, and core dump and crash files. Disk partitioning, in such cases, might simply include `/`, `/var` file systems and swap.

Solaris Jumpstart software can be used to great advantage in installing a large cluster grid. In addition to installing the basic operating environment on each node, it can also configure prerequisites for the various components, for example, the services port and admin user for Sun Grid Engine software. Some software components, such as the SunVTS™ diagnostic software, can also be installed directly from the Jumpstart post-install script using the `pkgadd` command, while others would require the use of a custom post-install scripted procedure. The setup of Solaris Jumpstart environment is beyond the scope of this document, but this article provides guidelines for specific procedures in the sections that follow.

File Sharing

The topic of how to share files across the cluster grid is by nature extremely complicated and dependent on the particular environment and priority of considerations. It boils down to a balance between performance and manageability. In this article, this issue is divided into two parts: sharing of binaries, and sharing of data.

Sharing of Binaries

Two elements of the cluster grid software stack provide the installation options of centralized or distributed installs. For ease of management, a centralized install is preferable. In cases where minimizing network traffic is a priority, distributed installs should be considered.

Sun HPC ClusterTools 4.0 software provides the option at install time of performing a distributed install or installing all binaries on a single master host. However, the Sun Grid Engine install script only supports binaries to be installed on a file system that must be shared across the SGE master host, compute tier hosts, and access nodes. In the next section, a customized installation is outlined that results in a bare-minimum sharing of grid engine files. If a centralized installation for both tools is chosen, they can be combined in the same file system, reducing the number of NFS shares.

In addition to the components of the stack, the other important binaries are obviously the ones belonging to the applications which run on the compute hosts. If the applications will not change frequently, then installing them locally on each compute host is a possibility. However, this must be done with extreme care, since this can lead to management difficulties when it comes to patches, upgrades, and so on.

A Sun technology that can play an important role here is the Sun CacheFS™ software. This allows you to set up a local cache for an NFS-mounted file system, which is automatically and transparently updated anytime the remote file system changes. This must be used only for remote file systems that change infrequently (read-mostly), otherwise performance can degrade due to excessive network overhead. In this situation, there would be an application server on which all applications are placed. This directory would then be shared through NFS and locally mirrored with CacheFS software. In normal operation, the application binaries would be invoked from the cache, but anytime the binaries are updated, the local cache on each compute host is automatically updated on the next invocation. Refer to the *Solaris Advanced Administration Guide* for details on CacheFS software.

Sharing of Data

The essence of computing is in the data files, both input and output. Although one could come up with methods for compute hosts to access data exclusively through stage-in and stage-out, some form of shared storage is usually inevitable. Apart from using a dedicated SAN (Storage Area Network), NFS is the most feasible way to share data files.

One way to maximize performance is to use a dedicated network for NFS. This could simply be regular Fast Ethernet, or it could be higher-performing Gigabit Ethernet. This network should be isolated from all other traffic, including and especially ordinary non-compute traffic (email, internet access, and so forth).

An important point to note is that Sun Grid Engine software by default expects home directories to be shared by compute hosts and submit hosts. If the same physical files are not accessible by both, care must be taken so that no submitted job makes any assumptions about files in the home directory, but only makes reference to files that are known to be accessible to the compute hosts.

In specific instances, CacheFS software can be used to speed up access to shared data directories. For example, in many biotechnology applications, pattern matching is done repeatedly on a common set of database files that are updated infrequently (once a week or less). In this case, putting these database files into a single shared file system and using CacheFS software on the compute hosts can cut down dramatically on execution time. Again, if updates to the database file system becomes too frequent, performance can be worse than without CacheFS software. Obviously, care must be taken so that output files are never created in the cached directory, but rather in another location.

Sun Grid Engine Software Installation Considerations

For a detailed examination of the Sun Grid Engine installation process refer to the Sun Grid Engine documentation and man pages provided with the software. This article augments the installation documentation by discussing important options for SGE installation.

Installation of the Sun Grid Engine master host is interactive, requiring input to questions asked along the way. It is straightforward once the prerequisite conditions, such as existence of the admin user and registration of the services port for the `commd` daemon, have been met.

Installation of the compute hosts (*exec hosts*) is also done with a script, which is by default interactive. However, this can be time consuming for large clusters, and there are several options to hasten this procedure:

- The install script `install_execd` can be invoked with the `-auto` flag. This causes all the installation questions to be automatically answered with default values, which is usually acceptable. This can further be incorporated into scripted install procedures, which do other things in addition to simply registering the host with the `qmaster` daemon.
- In the `util` directory of the SGE distribution, there is a script called `install_cluster.sh` that can be used to automatically install the SGE on all hosts specified on the command line. This script requires remote root `rsh` or `ssh` access privileges from the current host to all the candidate hosts. Consult the script for more details.

An option to Sun Grid Engine software that should strongly be considered is the use of local spool directories for compute hosts. By default, each compute host uses a subdirectory in the SGE distribution root to read and write information about jobs as they are running. This can result in a considerable volume of network traffic to the single shared directory. By configuring local spool directories, all that traffic can be redirected to the local disk on each compute host, thus isolating it from the rest of the network and reducing the I/O latency. The path to the spool directory is controlled by the `execd_spool_dir` variable; it should be set to a directory on the local compute host that is owned by the admin user, and that ideally can handle intensive reading and writing (for example, `/var/spool/sge`).

By default, the Sun Grid Engine software distribution is installed on a file system that must be shared across the SGE master host, compute tier hosts, and access nodes. Usually, this does not cause a significant performance issue. However, in cases where extremely high simultaneous access to binaries occurs (such as when launching extremely large parallel jobs), or where NFS traffic needs to be kept to a minimum for some other reason, the SGE can be installed locally on each compute host. In this case, rather than sharing the entire SGE distribution directory, it is sufficient to share only the `$SGE_ROOT/$SGE_CELL/common` directory, where `$SGE_ROOT` is the path to the SGE root and `$SGE_CELL` is the name of the SGE cell specified during the `qmaster` installation (typically, the cell name is `default`). The files that are then shared are mostly static configuration files, and no NFS traffic is incurred when binaries are invoked.

Note – This setup should always be used in conjunction with the SGE local spool directories as explained above. For submit and admin hosts, you can choose to install the binaries locally on all, or else have a central location from which those hosts can share the files. In all cases, it is recommended that the `path name` be the same for all hosts, so that `$SGE_ROOT` is the same, regardless of the actual location of the files.

Sun Management Center Installation Considerations

SunMC agents introduce a minimal ambient computational load on the host system. Therefore, if SunMC software is to be installed in the cluster grid, the benefits of comprehensive health monitoring of execution hosts must be balanced against the inevitable reduction in CPU and memory resources available for user applications.

If it is decided that the SunMC agents would put too much of a burden on the execution hosts, you can still use SunMC software to monitor those hosts without agents. In this case, the monitoring would be a simple SNMP ping, and the only thing that is tracked is whether or not the host is alive (accessible). For many

environments, this can be sufficient; the administrator can choose to ignore other problems, and simply inspect a malfunctioning system manually or use the testing suite to perform periodic tests.

MPI Runtime Environments

A heterogeneous compute tier can provide a platform for OpenMP, MPI, threaded and serial applications. Furthermore, it might be convenient to subdivide resources available for MPI applications through the use of *partitions*. An MPI job submitted to the Sun HPC ClusterTools CRE is launched on a predefined logical set of nodes or partition that is currently *enabled*, or accepting jobs. A job will run on one or more nodes in that partition, but not on nodes in any other enabled partition.

Partitioning a cluster allows multiple jobs to execute concurrently, without the risk that jobs on different partitions will interfere with each other. This ability to isolate jobs can be beneficial in various ways. For example, if a cluster contains a mix of nodes whose characteristics differ—such as having different memory sizes, CPU counts, or levels of I/O support—the nodes can be grouped into partitions that have similar resources. Jobs that require particular resources then can be run on suitable partitions, while jobs that are less resource dependent can be relegated to less specialized partitions. The system administrator can selectively enable and disable partitions.

Managing a Cluster Grid

In many cluster grid environments, large numbers of identically configured systems exist, providing opportunities to minimize administration time by taking advantage of diagnostic and automated installation tools.

SunVTS Software

SunVTS software can be used to perform hardware testing on a new node before the node is put into production. It can also be used on a regular basis for routine hardware checkups, or used to investigate a malfunctioning node. Many separate tests are included in the SunVTS application. Each test is a separate process from the SunVTS kernel. Tests are provided for processor, memory, network, communication, storage and peripheral devices.

When SunVTS software is started, the SunVTS kernel automatically probes the system kernel to identify which hardware devices are installed, and displays the testable devices in the SunVTS UI. This provides a quick check of the hardware configuration, and only those tests applicable to that system are displayed. During testing, the hardware tests send the test status and messages to the SunVTS kernel through interprocess communication protocols. The kernel passes the status to the user interface and logs the messages.

Cluster Console Manager

In the compute farm environment, tasks that require simultaneous command line input to multiple clients, such as patch installs and software upgrades, can be performed using the Cluster Console tool set that is provided with Sun HPC ClusterTools software. The Cluster Console Manager (CCM) enables you to issue commands to all nodes in a cluster simultaneously through a graphical user interface.

The CCM offers three modes of operation: `cconsole`, `ctelnet` and `crlogin`. The `cconsole` interface is of particular use as it provides access to each node's console port through terminal concentrator links. To use this tool, the cluster nodes must be connected to terminal concentrator ports and these node/port connections must be defined in the `hpc_config` file. Operations performed at the `ok` prompt such as configuring the boot PROM parameters, booting, and initializing operating system installations, can use this tool.

The `ctelnet` and `crlogin` uses `telnet` and `rlogin` respectively to log you in to every node in the cluster. Each of these modes creates a command entry window, called the *common window*, and a separate console window, called a *term window*, for each node. Each command typed in the common window is echoed in all term windows (but not in the common window). Every term window displays commands that you issue as well as system messages logged by its node.

Solaris Jumpstart and Flash Software

Solaris Jumpstart software should be used to perform installs at least for the compute tier of the cluster grid. If new hosts are added in the compute tier, or reinstalls are necessary, a well-configured Jumpstart environment will vastly reduce the management time for these tasks. The Jumpstart environment allows the administrator to set the Solaris install type according to the characteristics of the Jumpstart client.

The development of post-install scripts can further speed the install by performing the required system configuration tasks automatically following the Solaris install. This is particularly useful in the case of compute tier servers, which usually have relatively simple configurations in large number.

For a compute host, which is to be integrated in an existing Sun Grid Engine environment, the following tasks are required (in addition to the Solaris installation):

- Performing simple configuration tasks such as populating files like `/etc/hosts` or `/etc/system,` adding the SGE administrator user, and so fourth.
- Mounting directories for Sun Grid Engine binaries, user home directories, libraries, and executables.
- Execution of setup scripts such as `install_execd` for a Grid Engine execution host. In this case, it is necessary to register the new host as an admin host with the master node prior to the installation of an exec host.

In addition, it may be useful to perform some testing, in which case the installation of SunVTS, possibly followed by the automated execution of selected hardware testing scripts, can be performed.

Sun Grid Engine Software

The built-in logging and accounting capabilities of Sun Grid Engine allow administrators to keep track of compute jobs that run on the cluster grid. The software keeps a record of every job that has been run by SGE, including details such as start time, duration, CPU, memory and I/O consumption, user statistics, and, in the case of SGEEE, project and department information. The built-in SGE command `qacct` can be used to display summary information of resource consumption based on such criteria as user, project, resources requested, and time period. For more intricate sort criteria, or third-party accounting or analysis tools, you can access the accounting record directly. By default, it is kept in a file called `accounting` in the `$SGE_ROOT/$SGE_CELL/common` directory that is stored in a flat file database (the man page for the `accounting` command gives the format). A *how to* on the Grid Engine Project web site discusses how to rotate this file using a utility script provided with the software.

Cluster Grid Example Implementations

In this section, a sample Sun Cluster Grid implementation is outlined, demonstrating how to perform the following tasks:

- Translate site gathered information into grid architectural features.
- Take advantage of features that enable the management of heterogeneous environments in terms of the compute tier.
- Configure the MPI environment, OpenMP, serial, and interactive queues to maximize utilization.
- Take advantage of the inherent scalability of the architecture, and the flexibility with respect to changing user demands.
- Consider various network architectures to enhance the grid environment.

Scenario 1—General HPC Environment

This example represents a multi-departmental academic site that caters to 200 users with differing computational requirements. The information gathering stage is outlined below in terms of service provision and the three tiers.

Service Provision

In this example, a highly available service is not required, but system health monitoring is considered advantageous. The authentication scheme for the new cluster grid should integrate with the existing University scheme so that users can use their existing user names and passwords.

It is expected that the cluster grid will be configured to allow the compute tier to scale four-fold in terms of CPU count without further enhancement of the management tier.

A new storage facility will be associated with the cluster grid giving users significant disk space (at least 2 Gbytes per user) on high performance storage arrays. The total storage capacity can be increased within the year.

The DRM service is not currently required to implement a fair-share scheme across the departments, but this might be a future requirement.

Access Tier

Users are competent with UNIX and expect to have `telnet`, `rlogin`, and `ftp` access to the cluster grid storage system to configure jobs. Because the authentication scheme is to be integrated with the university's existing authentication scheme, the access nodes will be added to that scheme. A maximum of 50 users are expected to be simultaneously accessing cluster grid services.

A desktop machine with a monitor is required by the administrator for accessing the cluster grid management software.

The HPC resource must cater to mixed workloads including throughput jobs, highly parallel jobs as well as large memory and medium-scaling threaded or OpenMP jobs.

Management Tier

Besides the DRM, there is a requirement for the file storage to be served out to the compute tier using NFS. This storage should be fully backed up. Some health monitoring service should also be installed.

Compute Tier

The HPC resource must cater to mixed workloads including throughput jobs, highly parallel jobs as well as large memory and medium-scaling threaded or OpenMP jobs. The applications are roughly characterized in FIGURE 4. There are large numbers of serial jobs and a number of shared-memory parallel applications (OpenMP and threaded). A few OpenMP applications scale to 20 processors, while the high scaling jobs of 20 and above processors are MPI.

Data access patterns are highly varied but in this case, one key serial application is typified by very high read intensive I/O. Application runtimes for serial applications are typically 2–4 hours. The parallel application users have requested the ability to run for 24 hours on occasion, but day-to-day usage will include runtimes about 2–12 hours.

Design Implementation

In this section, the information gathering results are translated into hardware and software implementation decisions (FIGURE 3).

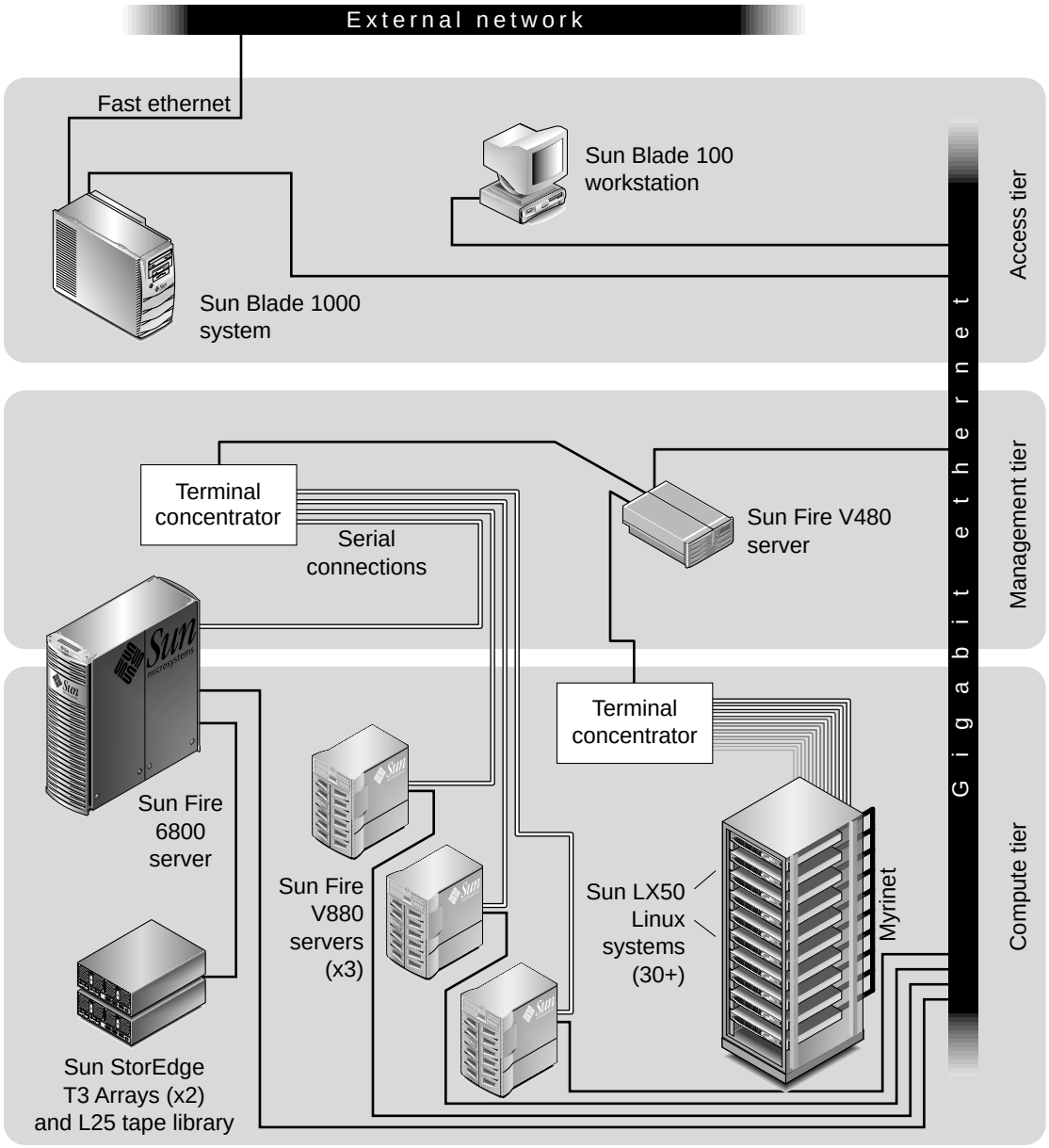

FIGURE 3 Grid Hardware and Network Infrastructure for Scenario 1

Access Tier

Users will be allowed login access to one dual-CPU node (a Sun Blade™ 1000 workstation) in order to access the cluster grid file system and submit jobs. These machines must cater for file transfers to and from the storage system. A maximum of 50 users are expected to be simultaneously accessing cluster grid services.

A Sun Blade 100 is supplied for the administrator's use.

Management Tier

A four-way Sun Fire V480 server is used for the SGE and SunMC server. The internal disks are mirrored for added resilience. this configuration should allow for substantial future scaling of the compute tier. The NFS service is provided by the Sun Fire 6800 server which is to be used also in the compute tier. The Sun Fire 6800 server will also provide full backup service through the use of a Sun StorEdge L25 tape library.

Compute Tier

The applications are characterized in FIGURE 4. There are large numbers of serial jobs and a number of shared-memory based parallel applications (OpenMP and threaded), as well as MPI applications beyond 40 processors.

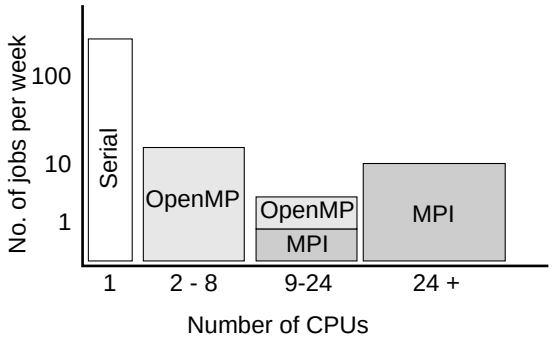

FIGURE 4 Approximate Workload for Example Site

A single Sun fire 6800 server is provided primarily to supply compute power for those OpenMP jobs that scale beyond 8 threads. It may also provide a platform for those MPI jobs that benefit from the SMP architecture. Smaller threaded jobs and serial jobs requiring greater than 1 Gbyte memory will be catered to by three Sun Fire V880 servers.

Price and performance considerations usually tend towards distributed memory systems for those appropriate applications. In this example, while applications scaling up to 24 processors must be catered to using a shared memory system, the high scaling MPI applications can take advantage of a distributed memory system such as a network of workstations (NOW). A Linux-based cluster, comprising x86 architecture processors packaged in a large number of dual-processor enclosures caters to those MPI jobs scaling beyond 24 processors. The Linux cluster is interconnected with Myrinet, Gigabit Ethernet, and serial network.

Shared storage across the cluster grid is hosted by a Sun Fire 6800 server on two Sun StorEdge™ T3 disk arrays implementing hardware RAID 5. The choice to host the shared storage here was by the desire to host storage direct attached (and therefore lowest access latency) on the largest SMP server, which is likely to be the host which accesses the storage most frequently.

Networks

Serial, gigabit Ethernet and Myrinet interconnects all feature in the example solution as shown in FIGURE 3. Gigabit Ethernet interconnects the management server with all compute hosts. A dedicated Myrinet interconnect provided a low-latency connection between the nodes of the Linux cluster.

Software

This section describes the software as it is applied to the example implementation.

Sun Grid Engine

In this example, the SGE is sufficient (rather than SGEEE) because fair-share scheduling across multiple departments is not required. If resource allocation based on departments or projects becomes a future requirement, the upgrade to the Enterprise Edition is a straightforward implementation.

The SGE queue structure must cater to serial, OpenMP and MPI jobs.

Queues are implemented on a calendar basis across the cluster grid. In the daytime, short 2- or 4-hour queues are enabled on all machines covering parallel environments and serial queues. Each evening, a 12-hour queue is enabled in place of the daytime queues. On weekends, 24-hour queues are enabled for jobs with long runtimes.

Three parallel queues for OpenMP jobs exist:

- 20-slot PE on Sun Fire 6800 server
- 8-slot PE on Sun Fire V880(a) server
- 4-slot PE on Sun Fire V880(b) server

A total of 12 slots are available for serial jobs on the Sun Fire V880 servers (four slots on the Sun Fire V880(b) and 8 slots on the Sun Fire V880(c)).

CacheFS software is implemented on one of the Sun Fire V880 servers. The CacheFS file system encompasses the subdirectory, which holds the data that is prone to high read-intensive activity by one key serial application. A Sun Grid Engine queue is configured on this machine to run jobs requesting this application.

On the management host, four slots are allocated for users to run interactive jobs (leaving approximately four processors continuously available for system processes, DRM and SunMC.)

MPI Environments

MPICH is installed across the Linux cluster to provide the runtime environment for MPI applications. MPICH can be tightly integrated with Sun Grid Engine.

Scenario 2– Minimal Cluster Grid Implementation

This scenario (FIGURE 5) demonstrates a minimal cluster grid implementation that uses the Sun Grid Engine resource manager to control a number of single processor machines. The following information pertains to the requirements placed upon the cluster grid.

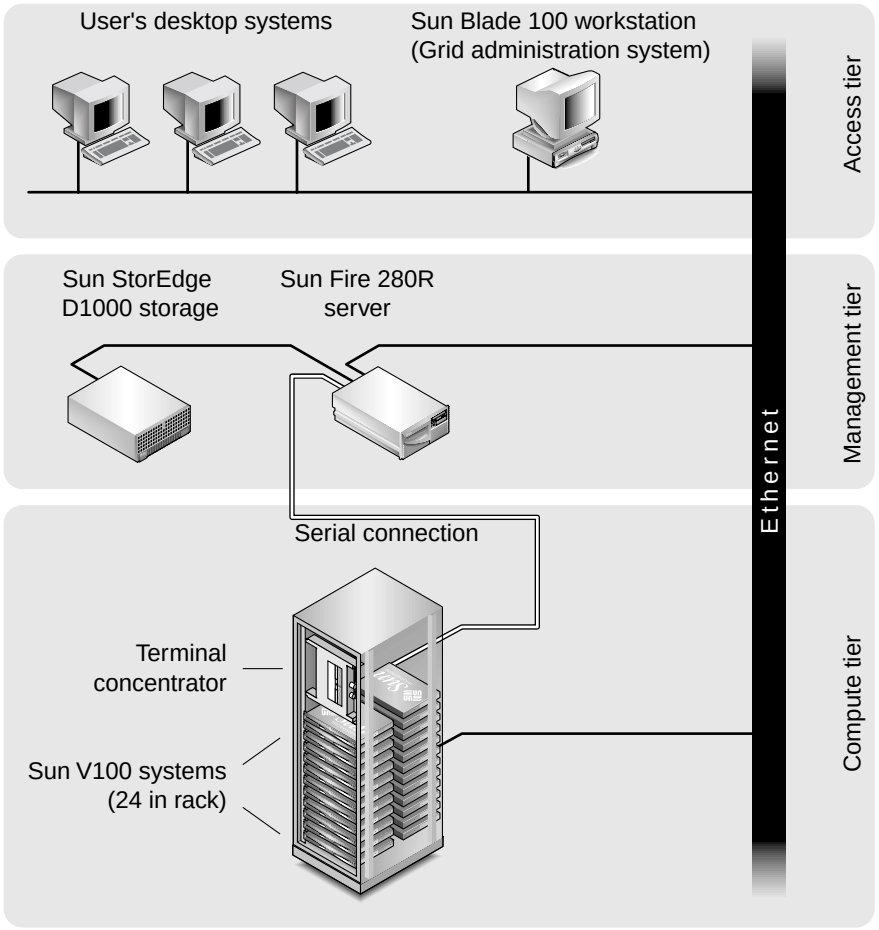

FIGURE 5 Grid Hardware and Network Infrastructure for Scenario 2

Service Provision

The service provision requirement calls for a queueing system to optimize the utilization of a new compute farm. The new service should impose minimal disruption to users, and take into account moderate scaling of the compute layer for future growth. Storage should be integrated within the new system.

Access Tier

Users currently work using desktop workstations and these should be integrated into the access tier to allow access to file storage, job submissions, and so on. A maximum of 30 users are expected to simultaneously access the cluster grid.

Minimal health monitoring of the system and administrative access is needed from an existing administration workstation.

Management Tier

DRM and NFS services must be provided. An existing SunMC server will be used to monitor any servers which are critical to the grid service.

Compute Tier

The HPC resource must cater to serial jobs only. A small number of floating point intensive applications are used. Jobs have memory requirements of approximately 500 Mbytes, with approximately ten percent of the jobs requiring up to 2 Gbytes. Average runtimes are around 30 minutes with little variation.

Design Implementation

In this section, the information gathering results are translated into hardware and software implementation decisions.

Access Tier

The user's existing desktop workstations will be registered with SGE as submit hosts, allowing users to submit applications without using `telnet`, and `rlogin` type commands. Access to the cluster grid file system where users have file space will be enabled using NFS.

Administrative access will be provided by registering the existing administration workstation as an admin host with SGE.

Management Tier

The management tier service is provided by a single 2-way Sun Fire 280R with a Sun StorEdge D1000 array directly attached. A SunMC agent will be installed on the 280R server so it can be monitored by the existing SunMC server. The file space will be NFS shared across both the Compute Layer and the users desktops. The network for the cluster grid (between compute and management layer) will be isolated from the office traffic through the use of an additional gigabit Ethernet interface which will connect the 280R to the compute layer only using a switch.

Serial connections to the compute tier will be installed along with a terminal concentrator to allow console access by the administrator.

Compute Tier

The compute tier is composed of 20 Sun Fire V100 servers, each configured with 1 Gbyte of RAM. An additional four Sun Fire V100 systems will be configured with 2 Gbyte of RAM.

Software

The SGE queue structure will be simple. This simplicity will be reflected in a fast scheduling loop which will improve throughput if very large numbers of jobs are submitted over a short time span. By default, the SGE installation results in single slot queues on single processor machines. Those jobs requesting greater than 1 Gbyte of RAM will be automatically scheduled to use the 2 Gbyte compute nodes. Further adjustments to time limits on these queues or tuning of the scheduler should be considered during the early stages of implementation.

About the Authors

James Coomer is a certified systems engineer in Sun's Solution Technology group, and is based in Sale, Manchester, England. He has five years experience in high performance computing, and currently works closely with UK academic institutions on high performance and grid computing. James received a M.Phys in theoretical physics at Lancaster University in 1997, and went on to gain a Ph.D. in theoretical quantum chemistry at Exeter University.

Charu Chaubal is an engineer in the Grid Computing group for Sun Microsystems, Inc. He has been working on implementations of grid technology, for customers and for demonstration projects, for the last two years. He has also developed and delivered training courses on grid computing, and performed technical marketing for grid technology products. Charu received a Bachelor of Science in Engineering from the University of Pennsylvania, and a Ph.D. from the University of California at Santa Barbara, where he studied the numerical modeling of complex fluids.

Related Resources

For additional information about the topics discussed in this article, refer to the following web sites:

- `http://www.sun.com/software/gridware/whitepapers.html` – This site is a source for the latest Sun Grid white papers.
- `http://www.sun.com/gridengine` – This site provides access to Sun Cluster Grid white papers, case studies, software downloads, and more.
- `http://gridengine.sunsource.net` – This site, sponsored by Sun and hosted by Colabnet, is a source for continued collaborative development of the Grid Engine project.
- `http://www.sun.com/solutions/hpc/communitysource` – Sun's HPC ClusterTools community source code access site, where Sun HPC ClusterTools source code is freely available through the Sun Community Source Code License (SCSL) model.
- `http://www.sun.com/oem/products/vts/index.html` – Sun's SunVTS site where you can freely download the SunVTS software and documentation.
- `http://www.myri.com` – Myricom's web site that provides links to Myricom products, services, and company information.

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com`[SM] web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive, or search for a specific book title or subject. The URL is `http://docs.sun.com/`.

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine web site at: `http://www.sun.com/blueprints/online.html`.