5-2009

Grid Computing in a Healthcare Environment: A Framework for Enterprise Design and Implementation

Charles Brust
Mayo Clinic, brust.charles@mayo.edu

Grid Computing in a Healthcare Environment: A Framework for Enterprise Design and Implementation

Charles Brust[1]
Mayo Clinic
brust.charles@mayo.edu

ABSTRACT

In this paper, we present an overview of grid computing, examine the potential uses for grid computing in healthcare environments, and propose a decision framework for healthcare information systems in large healthcare organizations. The proposed framework extends the Enterprise Desktop Grid architecture to allow healthcare systems to validate whether a grid computing system is an appropriate choice for their system, and to enhance decision making regarding the architecture of such a grid.

Keywords

Grid Computing, Healthcare Grids, Healthcare IT Infrastructure

INTRODUCTION

Grid computing - the tying together of several computers to allow them to perform work as a single larger entity - has gained significant ground in recent years as a method of harnessing large amounts of computing power for a relatively small cost. The benefits of grid computing have been adopted through many businesses, including graphic arts and film, publishing, e-commerce, internet search, energy producers, and pharmaceuticals (Joch, 2004), but the uptake of the technology in healthcare has lagged behind. Grid computing has many potential applications when applied to the healthcare setting. These include the use of the grid technology for such diverse applications as genetically-specialized medicine, 3D modeling and rendering, body simulation for surgery or diagnostics, drug interactions, genomic research, epidemic modeling, and research data mining (Groen & Goldstein, 2006). For these and many other applications, a healthcare organization may opt to use a grid of lower-cost servers to grind through the necessary calculations instead of purchasing a high-end system. In that manner, there are significant cost-savings that can be realized – a major benefit in today's world of low-reimbursement healthcare management practices. Further, grids will bring increased reliability and scalability to medical applications, while retaining a low cost-per-unit threshold. The need for reliable, scalable infrastructure solutions continues to increase in the healthcare arena, and the ability to meet these needs in a constrained budget will allow the necessary expansion of technology solutions to continue. In this paper, we will propose a framework by which healthcare enterprises might make determinations regarding the feasibility of grid computing for their environments, and regarding the correct grid format to use for a particular application. Further, this paper will extend previous work by the authors which defined a model for grid computing in large healthcare systems.

OVERVIEW OF GRID COMPUTING

Grid Computing Defined

Grid computing can be defined as the use of multiple computing resources in parallel to arrive at the desired output results more quickly than would have been possible with a single system (Gagliardi, Jones, Grey, Bégin, & Heikkurinen, 2005). In some cases, grid computing has been performed by dedicated systems, tied together by specialized software for such tasks as large-scale computations or for 3-D animation rendering. In other situations, the processor time has been culled from "spare" CPU cycles on desktop systems in an enterprise, or throughout the world (i.e. SETI@ Home). In either situation though, the basic premise remains the same: use many low-cost computing devices together to do the work that once would have required large dedicated systems (Gentzsch, 2002). Computational grids generally utilize specific software that allows the nodes to negotiate which work units will be completed by each system, and in what manner the grid will aggregate the results for presentation. In this way, enterprises that need large amounts of computing power have the ability to purchase multiple

[1] Surendra Sarnikar contributed to this paper as well. He is not listed as an author because he served on the conference committee for the MWAIS 2009 conference.

lower-cost systems as nodes, rather than investing in large super-computer type systems. Additionally, there is the added benefit of redundancy with the grid model, as the loss of any given node will not substantially reduce computational power, unlike the monolithic supercomputer model where operating system or hardware failures can cause total loss of processing for extended periods.

Two basic designs are available for grid computing: dedicated systems, or scavenged CPU systems. In either model, the grid creates in essence, a virtual supercomputer, where the work units can be processed in a massively parallel fashion. In a dedicated system, the computer resources within the grid members are used only for the functions of the grid, whereas in a scavenged CPU system, the computers are used for other functions on a day to day basis, and their "spare" CPU cycles are then redirected to do useful work for the grid. One version of the scavenged CPU model would be the Enterprise Desktop Grid. In this scenario, the workstations within an enterprise are used as the nodes for CPU scavenging, thus creating a ready-made installation base, with the added advantage of increased (though not total) control over the nodes.

Characteristics of Grid Technologies

Varying grid technologies have differing technological characteristics. In this section, we summarize nine different characteristics of various grid technologies.

Computing Power refers to the amount of processing power afforded by the grid technology. In general, all grid configurations are designed to provide a large amount of processing power for applications.

Intra-grid Network Bandwidth refers to the amount of bandwidth that connects each node in the grid. While dedicated grids have a high intra-grid bandwidth, scavenging grids are connected by local area networks and Internet.

Public Network Bandwidth refers to the bandwidth over which grid services are offered to end consumers. Both dedicated and scavenged CPU grid services are typically offered over lower bandwidth local area or public networks such as the Internet.

Data Storage Capability refers to the amount of data storage capability provided by the grid. Dedicated grid architectures are configured to provide high data storage capacity. While scavenged CPU grids can be configured to provide high data storage capabilities. Typical configurations of scavenged CPU grids such as Enterprise Desktop grids and volunteer computing grids do not include high data storage capabilities.

Scalability refers to the ability to add grid nodes in an expedited manner if workloads warrant. All the above discussed configurations support the quick addition of nodes to scale up the grid capabilities.

Reliability is the measure of any single node running the software, and whether a failure of that node will have significant effects on the application as a whole – high reliability would translate to a need for high per-node uptimes. All the above discussed grid architectures support high reliability configurations.

Availability: Availability measures application uptime, but on a grid-wide basis. Here, a node could go offline without severely altering the responsiveness of the application. Due to the high redundancies built into grid architecture, most grids provide high availability and uptime.

Manageability is the measure of how easy it might be for an administrator to ensure that the grid continues to function on a day-to-day basis, taking into account application upgrades, operating system patching, etc. as well as the ability to examine statistics from the nodes and the grid as a whole to determine how well they are performing. In current configurations, dedicated grids have a much higher degree of manageability, as the infrastructure is under the direct control of the administrators.

Cost per work unit measure examines to what level an institution may expect to see infrastructure investments take in order to implement that type of grid. Dedicated grids generally will have a high cost per work unit, as the hardware must be purchased only for that application. Scavenged CPU grid types will exhibit a low cost per work unit, owing to the nature of the hardware involved.

MEDICAL APPLICATION OF GRID TECHNOLOGY

In today's medical science, there is a large need for high-throughput computing, and as the medical technology advances, that need for computing power grows exponentially. Today, much of the compute power needs fall into the research-oriented areas of medicine, but if current trends continue, physicians will soon be requesting genomics, gene sequencing and other highly specialized tests for day-to-day clinical diagnoses, as well as utilizing these genomic markers to create customized treatments for cancer and other diseases (Donachy, Harmer, & Perrott, 2003). Areas in medicine that are using or planning to use grid technologies include genomics and bioinformatics, image rendering and storage for radiological functions such as

CT scans, MRIs, and ultrasounds, the electronic medical record (EMR), and even drug development through human systems simulation – here the grid would act as a human body simulator, and would react physiologically like a human to gauge reactions to and side effects from new medications. Liu, Zhou, & Documet (2005) also advocates the use of grids as a possible solution to clinical image data storage and recovery. In that scenario, Liu et al. notes that clinical images have long been a major diagnostic tool, but with the move to digital imaging, the potential of image loss increases, as well as the problems around the archival of these images. The use of grid computing in this arena could hold potential for allowing for higher availability of the images, by the elimination of single points of failure within the infrastructure, as well as assisting in the archive of the data and its timely retrieval. Further, there are scenarios where data could be linked across institutions to create national healthcare grids, allowing research and decision support models to be created. With this broader scope, there may be opportunity to add home-based systems into the grid to support monitoring or other long-term care needs.

Requirements Framework for Medical Informatics Applications

Medical applications described above have varying technological needs. We use the technology characteristics of various grid architectures outlined in the previous section to describe the technology requirements of medical applications. Table 1 summarizes the high-level technology requirements and capabilities for the various medical applications of grid technology. For purposes of clarity, the classifications denote the amount of resource that an application would need to run properly in a production medical environment. As an example, Genomics applications are ever-changing, as different report styles, etc. are created, whereas EMR databases will see little change over time with regard to the actual application. The cost per work unit measure in this context examines how fiscally valuable the data from an application may be to a medical institution, and to what level they may expect to see infrastructure investments take in order to implement that application. This may be equated to the amount of time that is required to see a return on investment (ROI) for the infrastructure.

Table 1. Need Thresholds for Various Medical Applications

	Computing Power	Network Bandwidth (intra-grid)	Network Bandwidth (public)	Data Storage	Scalability	Reliability	Availability	Manageability	Cost per Work Unit
Medical Imaging	High	High	Medium	High	Medium	High	High	Medium	High
Databases and the Electronic Medical Record	Low	High	High	Medium	High	Medium	Medium	Low	Low
Genomics	High	Low	Medium	Med - High	Medium	Medium	Medium	High	Medium
Home Healthcare	Low	Low	Low	Low	Low	Medium	Medium	Low	Low
Simulation and Modeling	High	High	High	High	Medium	Medium	Medium	Medium	Medium
Personalized Prescription and Dosing	Medium	Low	Low	Low	Medium	High	High	Medium	Low
Clinical Decision Support	Low	Low	Low	Medium	High	Medium	Medium	Low	Low

AN ENTERPRISE GRID FRAMEWORK FOR HEALTHCARE

As the need for grid technologies grows, it becomes necessary to examine the methods by which such a system could be architected for a given enterprise application. In order to satisfy the requirements of such a diverse set of applications typically run in a large healthcare organization, it is necessary to develop a customized grid that is optimized to handle the requirements for each application. This may mean that multiple grids are needed within an organization, depending on the

application load. Before presenting the framework for healthcare information systems to architect grid technology, we begin by defining a healthcare information system.

Healthcare Information Systems Defined

For the purposes of this framework, a healthcare information system will encompass the hardware, software, and dataset inherent in the care of a particular patient or research scope. For example, a healthcare information system might be as simple as the application and related data used to perform eye examinations in a single-provider clinic, or as complex as a research set which includes all the CT scans taken for patients from European descent exhibiting a particular set of symptoms. This framework will take the breadth of these definitions into play in describing the application of grid technology to assist in solving the related problems of accessing and processing this data.

Proposed Framework

The purpose of this framework is to allow healthcare organizations to optimize their grid architecture choices. The grid arena is complex, and incorrect choices in architecture can severely limit grid performance or utility. The aim of this framework then is to guide these decisions so that each implementation can achieve its full potential.

To design a grid, one must first consider the application to be run there. As previously discussed, various applications will have vastly different computing needs. The application discussion must also examine the grid type (high work-unit throughput, collaboration, etc.) Further, an enterprise must also consider the question of dedicated or CPU-scavenged grid nodes. If, as in most organizations, workstation utilization is low, then this excess capacity can be siphoned for more useful work.

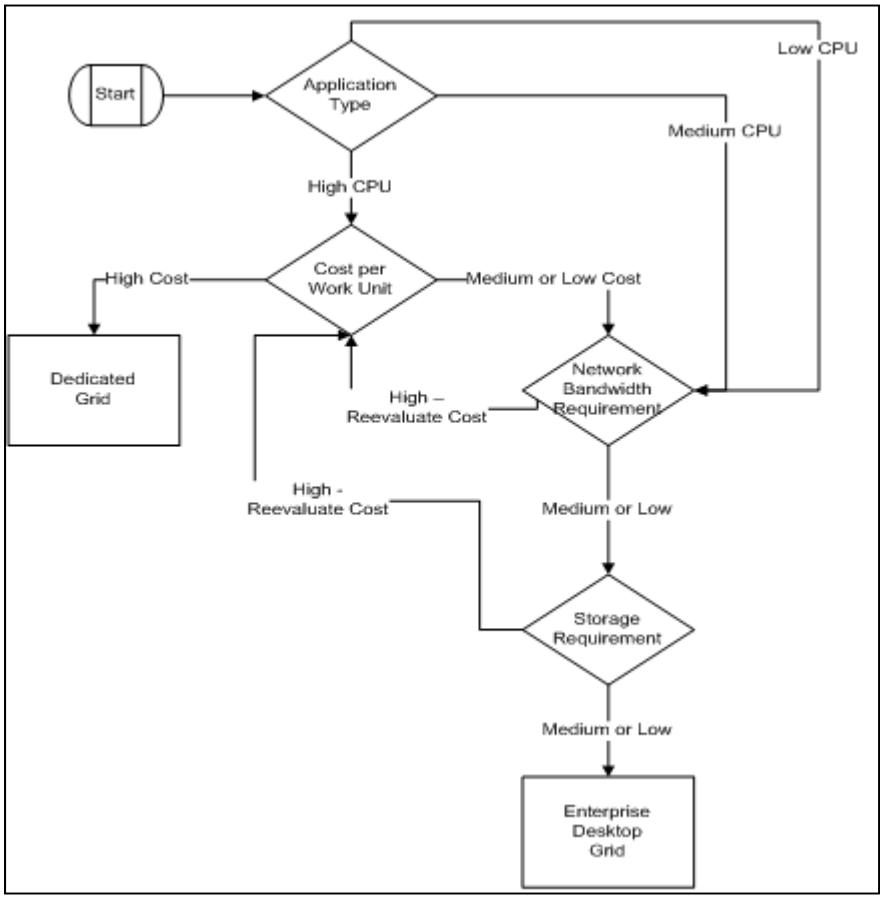

Figure 1. General Framework

In the proposed framework (Figure 1), we describe a method by which an enterprise can make correct choices around the type of grid to deploy for any given application. First, the enterprise must examine the type of application being proposed.

In general, if the application is projected to have high CPU utilization (continual), there is a likelihood that a dedicated grid will be needed to allow for sufficient throughput. This is potentially overridden by the available funding (as described by the cost per work unit) – if this funding is not available at a high level, then it is possible to continue to deploy a high-CPU need application in an Enterprise Desktop Grid, though the throughput will potentially be reduced, thus translating to a slower result return for the application.

An application with high CPU utilization, and which is allowed to have a high cost per work unit is then recommended to be implemented as a dedicated server grid. In this fashion, the application will have unfettered access to all the server resources, along with any related infrastructure such as network and storage. This scenario will maximize the throughput for the application.

As most implementations will not have unlimited funds to implement the grid, the more likely scenarios are that the cost per work unit is limited, thus pushing decisions to tend toward a lower-cost installation. This then feeds to the next decision point of network bandwidth: high bandwidth requirements will translate directly to higher costs per work unit, due to the increased funds needed to implement large pipeline networks. If this is a requirement for the application's function, then the cost per work unit allotment must be reexamined, as it will not be possible to fall into a low cost scenario. If on the other hand, bandwidth requirements are lower, then the application is still trending toward Enterprise Desktop Grid deployment potential.

Storage requirements are similar to network, in that if those needs are high, the cost per work unit is increased significantly, and again, the cost per work unit allowance must be reconsidered. Again, if this is increased, the best scenario is to install a dedicated grid. If instead, the storage requirements are reduced (or a lower cost implementation is allowed), then an Enterprise Desktop Grid is the optimal solution.

Several items in the requirements framework above are not discussed in the solution, as they do not directly affect the selection, either because the grid types are equivalent across implementations, or because the grid type selection will inherently direct the potential scope, rather than the reverse.

Scalability could be used as a decision point in determining grid type, but as it really would be a function of cost at that point, it has not been included in this framework as such. Instead, scalability will be much easier in an Enterprise Desktop Grid, as there is a larger pool of pre-installed workstations to draw from; in contrast to the dedicated grid which would require the purchase and installation of added server hardware (thus increasing costs and time-to-implement) to scale upward.

Reliability is inherently lower in an Enterprise Desktop Grid, as the workstation users have the capability of powering systems off at the end of the day, stopping the grid application service, etc. In a dedicated grid, each individual server is a more controlled environment, as their function is only to process work units, without user interaction.

Availability for the Enterprise Desktop Grid has the potential to actually be higher than that of a dedicated grid, as there will be a much larger number of nodes in the grid. This then allows for multiple systems to work on results in parallel or to duplicate work unit assignments, taking the results from the first responding node. Thus, the individual workstation outages that may happen are less important due to the vast potential scale. In a dedicated grid, monetary concerns will likely limit the number of nodes, and thus limit parallelism and redundant work. As that is the case, the outage of a server node will have a more detrimental effect on the grid as a whole.

Manageability is normally not taken into account when looking at cost per work unit calculations, as the costs are soft (administrator time, downtime for a node, etc.) Dedicated grids will have a lower management footprint, simply due to the lower number of nodes involved. Enterprise Desktop Grids will thus have a higher manageability time commitment, as the number of nodes is vastly increased.

Figures 2 and 3 step through the decision framework for example applications.

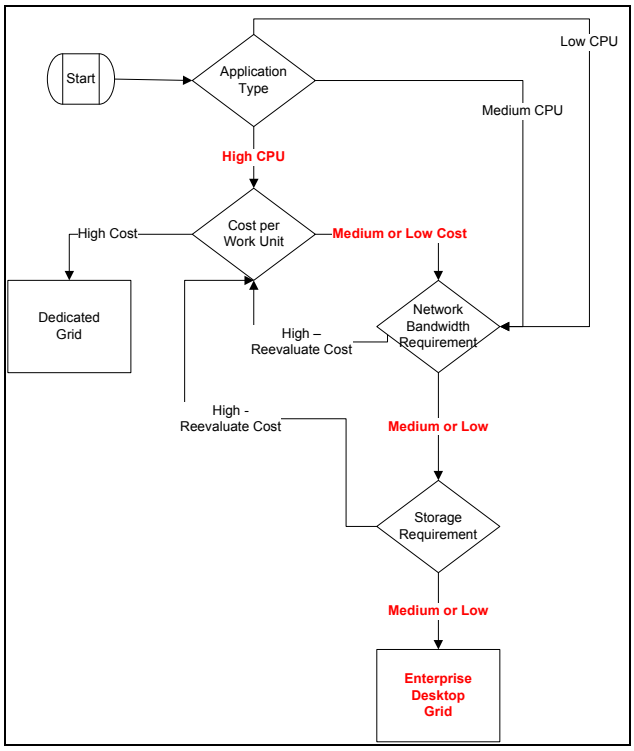

Figure 2. Genomics

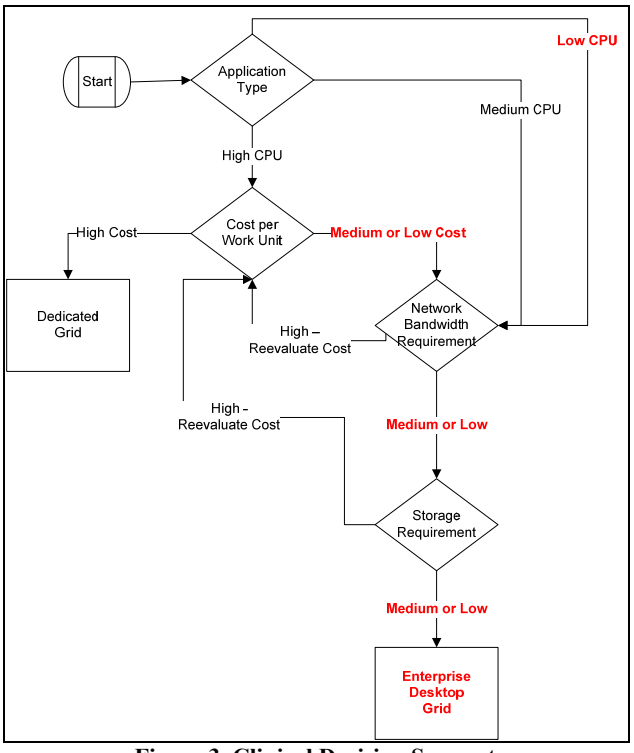

Figure 3. Clinical Decision Support

LIMITATIONS AND FUTURE WORK

As with any model, there are assumptions made as to infrastructure, funding, etc. that may not be realistic in practice. These include the availability of unlimited staff time and infrastructure (including monetary concerns), the ability of the institution to define the needs for grid resources within their enterprise, and the availability of underutilized workstation / server hardware to compose the nodes of the grid. Further, this framework assumes that the grid architecture model proposed by the authors in an earlier work has been accepted, and is utilized in this architecture. Also, this framework is optimized for utilization in CPU-sharing grids, rather than in instances of data distribution or knowledgebase grids. The framework must be extended in future work to be optimized to include other grid styles. Finally, this framework is limited as it is somewhat subjective in the definitions. This will be corrected in future revisions by utilization of an AHP-style framework model, which will allow for mathematical definitions to be applied to the decision points.

REFERENCES

1. Chien, A., Calder, B., Elbert, S., and Bhatia, K. "Entropia: Architecture and Performance of an Enterprise Desktop Grid System," Journal of Parallel and Distributed Computing (63:5) 2003, pp 597-610.
2. Donachy, P., Harmer, T.J., and Perrott, R.H. "Grid Based Virtual Bioinformatics Laboratory," Proceedings of the UK e-Science All Hands Meeting (2003)) 2003, pp 111-116.
3. Gagliardi, F., Jones, B., Grey, F., Bégin, M.E., and Heikkurinen, M. "Building an infrastructure for scientific Grid computing: status and goals of the EGEE project," Philosophical Transactions: Mathematical, Physical and Engineering Sciences (363:1833) 2005, pp 1729-1742.
4. Gentzsch, W., and Computing, D.G. "Grid Computing, A Vendor's Vision,".
5. Groen, P.G., D. "Grid Computing, Health Grids, and EHR Systems," in: Virtual Medical Worlds Monthly, 2006.
6. Joch, A. "Grid Gets Down to Business," in: Network World, 2004.
7. Liu, B.J., Zhou, M.Z., and Documet, J. "Utilizing data grid architecture for the backup and recovery of clinical image data," Computerized Medical Imaging and Graphics (29:2-3) 2005, pp 95-102.