



**IGNITED MINDS**  
Journals

*Journal of Advances in  
Science and Technology*

*Vol. III, No. VI, August-  
2012, ISSN 2230-9659*

**REVIEW ARTICLE**

**IMPORTANCE OF GRID COMPUTING AND  
ITS CONCEPT**

# Importance of Grid Computing and its Concept

Shelly Bhutani

Research Scholar of Singhania University, Rajasthan, India

## INTRODUCTION

Grid computing involves the sharing, selection, and aggregation of a wide array of resources that may include supercomputers, mainframes, storage systems, data sources, distributed applications and management systems. If one considers the internet as a network of communication, grid computing can be considered a network of computation.

Grid computing is an extension of scalable and distributed computing concepts. Its objective is to harness the resources of unused CPU cycles of computers to perform computational tasks at a faster rate. The idea of using the unused CPU cycles originated in the early 1970s with the linking of computers by networks. In 1973, scientists at the

the emergence of peer-to-peer networks and computational grids.

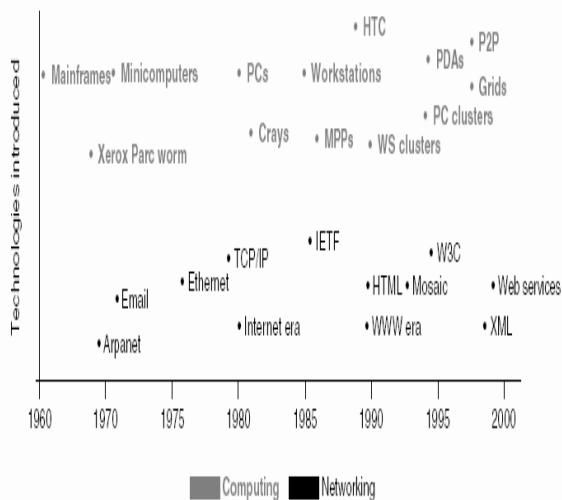
Though the terms “peer-to-peer computing” and “grid computing” have been used interchangeably by some, Buyya and Chetty make an interesting distinction between them: P2P computing is concerned with sharing low end systems such as PCs connected to the Internet for amassing computing power or share contents as done in Napster and Gnutella. On the other hand grid computing is concerned with aggregating the distributed clusters of computers using well defined protocols and standards.

While grid computing was initially used for scientific and academic research, the past two years have seen a significant increase in the use of grid computing by commercial institutions.

Shell uses grid-enabled infrastructure for applications that involve interpretation of seismic data. The grid is expected to cut processing time of seismic data while improving the quality of the output. Further, because of the open standards used, the grid is expected to provide easy integration of existing software.

Bank One is using grid computing to boost performance of analytics for its Chicago based trading floor. Instead of using a super computer to process its massive risk analytics, Bank One uses a lot of small processors and the computing responsibilities have been distributed across them. By doing so, Bank One expects to make significant cost savings on hardware and also utilize the unused processing cycles of the distributed resources JP Morgan Chase Investment Bank (JPMCI) is offering its business units online access to raw computing power via an enterprise technology infrastructure, which pools the financial firm’s flexible processing resources for compute-intensive applications. The project code-named “Compute Backbone” uses grid computing technology to improve process service levels and to cut costs by charging the business units for their processing, with peak and off-peak pricing Steven Neiman, head of JPMCI’s high performance computing initiative.

For others in the financial industry, grid computing enables the migration to cheaper computing solutions, such as running Linux on Intel. Charles



XEROX Palo Alto Research center developed a worm program that replicated itself in the memory of each of the 100 Ethernet connected computers. The worms used the idle resources to perform shared computations for rendering realistic computer graphics.

The popularity of the Internet and the technological advances in the last decade has taken the distributed computing concept to a new level. Computers are now hooked together in clusters for high-performance computing. These clusters have resulted in the emergence of peer-to-peer (P2P) networks and computational grids. Figure provides a timeline of the advances in networking and computation leading to

Schwab converted an existing wealth-management application and grid enabled it using the Globus Toolkit with IBM eServer running RedHat Linux. By running Linux, a free open-source operating system, Charles Schwab expects to save significant capital expenditure on hardware, such as expensive UNIX servers. The project has helped Charles Schwab reduce the processing time of certain transactions from four minutes to fifteen seconds.

Novartis A.G., the Switzerland based drug and pharmaceutical giant uses grid computing technology to speed up its drug research in a cost-effective way. The grid links about 2700 PCs delivering 5 teraflops of computing power. Manuel Peitsch, head of Novartis AG's department of Informatics and Knowledge Management explains, "If you look at the desktop PCs in a typical corporation, probably 90 percent of computing cycles are unused. Just by capturing unused cycles on the PCs we have already got, we have created a 5 teraflops supercomputer. We have avoided the expenses of buying an HPC system, building another computer center, and taking on the people to support it. We invested roughly \$400,000 in grid software licensing and figure we have saved at least \$2M based on the 2700 seats we have currently. We expect to realize more savings of this nature in the future as our grid expands."

Other organizations adopting grid services include the Mayo clinic, which is developing a system for linking its medical database with vast external public and private data sources in order to develop more-effective patient treatments Pratt & Whitney, a Connecticut based flight technology company, has been using grid technology to model jet engines and gas turbines.

The rest of this paper discusses the need for grid computing, grid concepts and components, grid architecture and standards, a case study of adoption of grid computing, and future trends in grid computing.

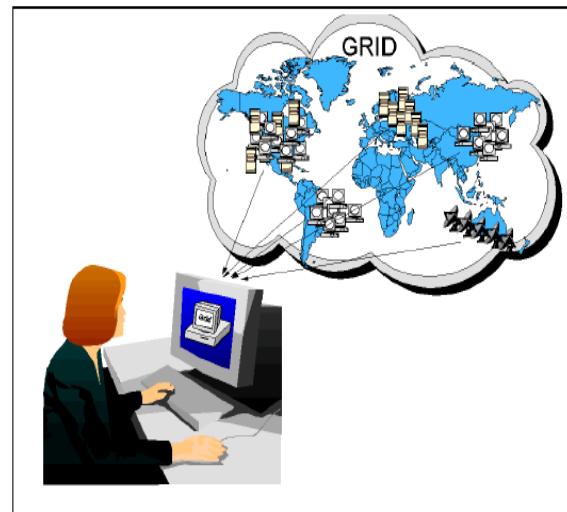
### Why should one care about grid computing?

In most organizations there is an enormous amount of computational processing power which is unused. There is also usually a huge amount of unused storage capacity residing on these machines. Grid computing provides a framework for exploiting these underutilized resources.

Another important grid computing contribution is to enable and simplify collaboration among different organizations. This collaboration does not concern file sharing alone. It concerns direct access to computers, software and other resources. Heterogeneous systems distributed globally can work together to create an image of a virtual computing system offering a variety of virtual resources.

The users of the grid can be divided and grouped together based on their functional areas to form virtual organizations (VOs). These virtual organizations may

have their own sets of policies and rules. According to Foster, Kesselman and Tuecke in their paper, "The Anatomy of the Grid" ,



**Figure: Simple view presented to the user after virtualization of the distributed resources**

"VOs vary tremendously in their purpose, scope, size, duration, structure, community, and sociology. Nevertheless, careful study of underlying technology requirements leads us to identify a broad set of common concerns and requirements. In particular we see a need for highly flexible sharing relationships, ranging from client-server to peer-to-peer; for sophisticated and precise levels of control over how shared resources are used, including fine-grained and multi-stakeholder access control, delegation, and application of local and global policies, for sharing of varied resources, ranging from programs, files, and data to computers, sensors, and networks; and for diverse usage modes, ranging from single user to multi-user and from performance sensitive to cost-sensitive and hence embracing issues of quality of service, scheduling, co-allocation, and accounting."

They further state that current distributed computing technologies do not address the concerns discussed above. According to them:

- CORBA and Enterprise Java are concerned with resource sharing within a single organization but not across organizations;
- Open Group's Distributed Computing Environment (DCE) supports resource sharing across sites but are too cumbersome and inflexible; and
- Storage Service Providers and Application Service Providers allow organizations to outsource storage and computing requirements, but only in constrained ways.

Hence, comprehensive technologies with flexible frameworks and standards are needed to address the concerns discussed above. Years of research on grid computing have now resulted in a set of protocols and rules that address the concerns discussed above. These technologies include security algorithms that support management of resources across multiple locations, protocols for querying information across multiple locations and protocols for configuring resources and machines that form part of the grid.

The last few years has seen a renewed interest in grid projects around the world. Vendors such as IBM and Oracle are positioning grid computing as an important cog in their computing strategies. Grid computing is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation and many others.

## **GRID CONCEPTS AND COMPONENTS**

A grid consists of many resources. These resources are sometimes addressed by different names such as "nodes", "resources", "members", "donors", "clients", "hosts", "engines". "Computational Grid", "Scavenging Grid" and "Data Grid" are other terms used in the grid computing world. In a computational grid, resources, usually high performance servers, are set aside for computing power. In a scavenging grid, resources, usually desktop machines, are scavenged for unused CPU cycles and other resources. In a data grid, the main focus is to provide access to data. Users are unaware of the location of data and are only concerned with access to data. This section discusses the meaning and application of concepts such as "Computation", "Storage", "Communications", "Jobs", "Scheduling", etc., to the realm of grid computing. The various software components that are used in grid computing are also discussed.

### **Grid Concepts**

- **Computation:** The computing cycles provided by the processors on the grid.
- **Storage:** The data storage present in the grid. The storage may refer to the primary storage such as memory attached to the processor or it may be secondary storage such as hard disk drives and other permanent storage media such as networked file systems. Data may also be stored on storage devices that span several machines in a grid. A unifying file system may be used to provide a single uniform name space for this storage. By having a unifying file system, the user can reference the data in the grid without regard to its exact location. Grids may also have an independent scheduler that can select the appropriate storage devices based on usage patterns and storage needs.

- **Communications:** This refers to communication within the grid and communication external to the grid. Internal communication within the grid is essential, when one considers the fact that many jobs in the grid have to access data that reside on multiple machines. The criticality of the bandwidth available for this internal communication depends on the amount of data that has to be transferred between different machines on the grid. External communication refers to access to the Internet. If an organization's grid has to communicate with other grids, then external communication becomes critical. Communication between grids that are spatially and geographically distributed usually takes place through the Internet. Redundant communication paths are sometimes needed in a grid to provide a fault tolerant system to guard against network failure or excessive data transfer.

- **Software and licenses:** Grid computing provides the opportunity of installing specific software on only a few machines. If a job requires the software to be used, the job can be sent to the machine in which the software is installed. By doing this, an organization can save significant expenses on licensing fees.

- **Special equipment, capacities, architectures, and policies:** A grid may contain platforms with different architectures, operating systems, devices, capacities, and equipment. All these attributes must be considered before assigning jobs to the grid. Some software may run only on Windows machines while others may run only on Linux machines. Further, some machines may be used only for number crunching financial research because of the hardware specifications of those machines. Some machines may be used only for search purposes. By connecting each of these machines to the Internet using separate high speed connections, instead of sharing a single connection, the bandwidth is increased and would facilitate faster search results. Hence the grid administrator should ensure that policies and rules are available to schedule jobs based on the capacities and purpose of the resources.

- **Jobs and Applications:** The term "application" is used to refer to the highest level of a piece of work on the grid. An application may be broken down into jobs and sub-jobs. Terms such as transaction, work unit, and submission are commonly used as equivalents for job. The jobs in an application may be programmed to execute in parallel on different machines in the grid. Some jobs may have specific dependencies and hence may execute in a sequential order with the output produced by one job being used as the input for a certain job. The grid should have provisions to collect and appropriately

assemble the results of the jobs to produce the required output for the application .

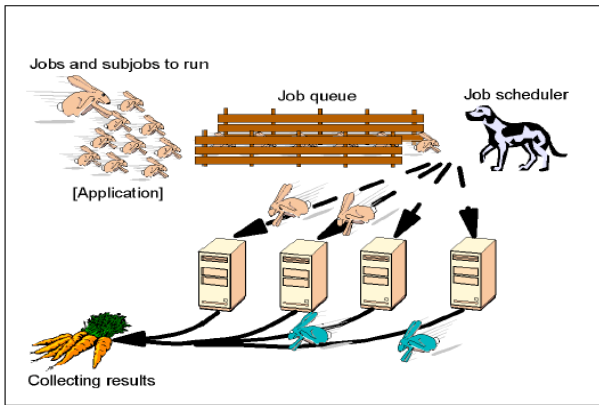


Figure : Jobs and Applications

(Source: Berstis)

- Scheduling, Scavenging and Reservation:** A grid may contain a “job scheduler” that schedules a job based on the availability and appropriateness of the resources. The term “resource broker” may also be used as an equivalent to “job scheduler”. A “scavenging grid” system is one where a machine reports its idle time to the grid management node. The management node would assign to this idle machine the next job that meets the constraints of the machine’s resources. But the downside is, if this machine becomes busy with a local non-grid job, then the grid job gets delayed or suspended. This can be overcome by having “dedicated” machines on the grid that are not preempted by local non-grid work. In some cases, grid resources may be “reserved” in advance for a designated set of jobs. “Reservation” can aid in meeting deadlines and also guarantee quality of service. Some of the ‘reserved” resources may be “scavenged” for their idle cycles to run low-priority jobs during a reservation period. For maximum efficiency, a combination of “scheduling”, “scavenging” and “reservation” may be used. The type of software used for this purpose is discussed later in this section.

- Intragrid and Intergrid:** The simplest grid may consist of a few machines having the same hardware architecture and operating system and connected on a local network. Choosing application software for these machines is simple since they have the same type of hardware and operating system. In most cases, a grid would include machines from other departments and also the machines may be heterogeneous dissimilar hardware architecture and operating systems. An “Intragrid” usually refers to machines from different departments within a single organization forming the grid “Intergrid” on the other hand refers to a grid that crosses organization boundaries .

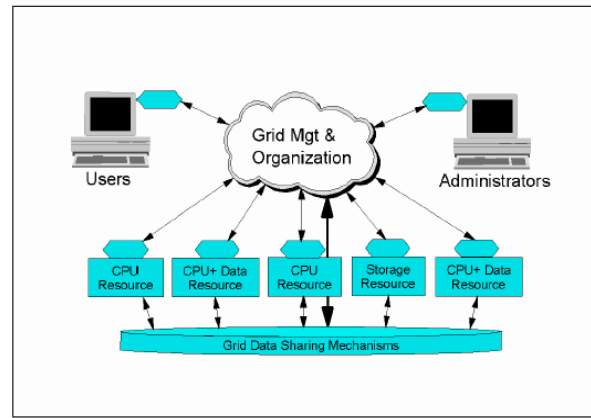


Figure : Intragrid

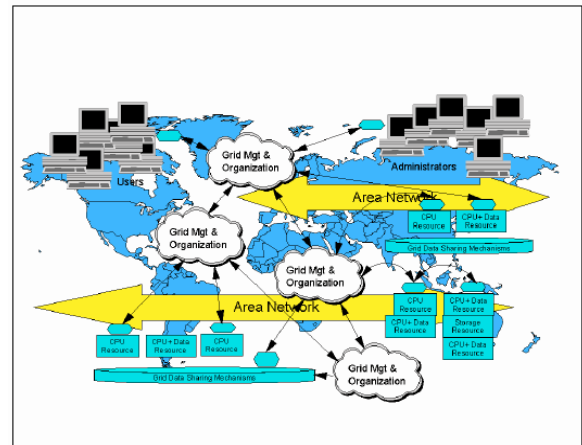


Figure : Intergrid

(Source: Berstis)

**Grid Software Components**

- Management Components:** Every grid system must have some kind of management software. This software is used to keep track of resources available to the grid and also to keep track of the different users on the grid. Other jobs of this software include measurement of: capacities of the resources, utilization rate of the resources, traffic congestion and bottlenecks, overall usage patterns of resources. Such software also generates statistics, provides for recovery from grid failures, finds alternate ways to get the jobs done.

- Donor and Submission Software:** A machine (donor) that contributes resources to the grid must be enrolled as a member of the grid and must have software installed that allows the grid to better manage its resources. The software installed on the donor machine can help to monitor its resources and also to send the resource information to the grid management software. For example, in a “scavenging” grid this information is used to inform the grid management software of the availability of idle time in a machine. Also, this software is used to accept jobs from the grid and

execute it on the donor machine. Any member machine in a grid can submit jobs to the grid. But, in some grid systems, specialized software called "submission software" may be installed on individual machines to allow the machines to submit jobs to the grid.

- **Job Scheduling Software:** Job scheduling software is used to assign a specific job to a machine on the grid. This assignment is usually based on some scheduling algorithm. For example, a priority algorithm with priority queues may be used to schedule jobs. As grid resources become available, the jobs from higher priority queues are scheduled to be executed first. Other information such as grid traffic and network outages is taken into account before scheduling jobs.

- **Communications Software:** A grid system may contain communication software to help jobs communicate with each other. An application that consists of many sub-jobs may require some of these sub-jobs to communicate information with each other. The sub-jobs should be able to locate other sub-jobs and transfer information to them. For example, communication software that follows the open standard Message Passing Interface (MPI) can be used for this kind of communication.

## REGERENCES

- Lovas, R., Dózsa, G., Kacsuk, P., Podhorszki, N., Drótos, D. (2004) 'Workflow Support for Complex Grid Applications: Integrated and Portal Solutions', Proceedings of 2<sup>nd</sup> European Across Grids Conference, pp.129-138.
- Ludtke, S., Baldwin, P. and Chiu, W. (1999) 'EMAN: Semiautomated Software for High-Resolution Single-Particle Reconstruction', Journal of Structure Biology, Vol. 128, pp. 146–157.
- Ma, T. and Buyya, R. 2005 'Critical-Path and Priority based Algorithms for Scheduling Workflows with Parameter Sweep Tasks on Global Grids', Proceeding of the 17<sup>th</sup> International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2005), IEEE CS Press, pp.251-258.
- Mello, R. F., Filho J. A. A., Senger, L. J., Yang, L. T. (2007) 'RouteGA: A Grid Load Balancing Algorithm with Genetic Support', Proceedings of the 21st International Conference on Advanced Networking and Applications, (AINA 2007), IEEE CS Press, pp.885-892.
- Quan, D.M., Kao, O. (2005) 'On Architecture for an SLA-aware Job Flows in Grid Environments', Journal of Interconnection Networks, Vol. 6, No. 3, pp.245-264.
- Quan, D.M., Hsu, D. F. (2006) 'Network based resource allocation within SLA context', Proceedings of the GCC2006, pp. 274-280.
- Quan, D.M., Altmann, J. (2007) 'Business Model and the Policy of Mapping Light Communication Grid-Based Workflow Within the SLA Context', Proceedings of the International Conference of High Performance Computing and Communication (HPCC07), pp.285-295.
- Quan, D.M. (2007) 'Error recovery mechanism for grid-based workflow within SLA context', Int. J. High Performance Computing and Networking, Vol. 5, No. 1/2, pp. 110-121.
- Quan, D.M., Altmann, J. (2007) 'Mapping of SLA-based Workflows with light Communication onto Grid Resources', Proceedings of the 4th International Conference on Grid Service Engineering and Management (GSEM 2007), pp.135-145
- Quan, D.M., Altmann, J. (2007) 'Mapping a group of jobs in the error recovery of the Grid-based workflow within SLA context', Proceedings of the 21st International Conference on Advanced Networking and Applications, (AINA 2007), IEEE CS Press, pp.986-993.
- Rewini, H. E. and Lewis, T. G., 1990, Scheduling parallel program tasks onto arbitrary target machines. Journal of Parallel and Distributed Computing, 9, 138-153.
- Shahid, A., Muhammed, S. T. and Sadiq, M., 1994, GSA: scheduling and allocation using genetic algorithm. Paper presented at the Conference on European Design Automation, Paris, France, 19-23 September.
- Sahai, A., Graupner, S., Machiraju, V. and Moorsel, A. 2003 'Specifying and Monitoring Guarantees in Commercial Grids through SLA', Proceeding of the 3rd IEEE/ACM CCGrid2003, pp.292–300.
- Sarkar, V., 1989, Partitioning and Scheduling Parallel Programs for Multiprocessors. MIT Press, Cambridge, MA.

- Singh, M. P. and Vouk, M. A. (1997) Scientific Workflows: Scientific Computing Meets Transactional Workflows,

<http://www.csc.ncsu.edu/faculty/mpsingh/papers/databases/workflows/sciworkflows.html>

Spooner, D. P., Jarvis, S. A., Cao, J., Saini, S. and Nudd, G. R. (2003) 'Local Grid Scheduling Techniques Using Performance Prediction'