Grid Computing

Kousik Dasgupta
Lecturer, Dept. of Computer Science and Engineering

1. What does it mean?

It is a form of distributed computing that involves coordination and sharing of computing, application, data storage or network resources across heterogeneous and geographically dispersed organisation.

As, standardization of communications between heterogeneous systems created the Internet explosion. The emerging standardization for sharing resources, along with the availability of higher bandwidth, is the driving force behind the evolutionary step towards grid computing.

2. What can grid computing do?

a) Explore underutilized resources :

The simplest use is running existing application on different machines.

In many places it is seen that most machines remain idle for most amount of time, even the servers remain idle. Grid provides a framework for exploiting these resources efficiently. Often, machines may have enormous unused disk drive capacity. Grid computing, more specifically, a "data grid", can be used to aggregate this unused storage into a much larger virtual data store. An organization may have occasional unexpected peaks of activity that demand more resources. If the applications are grid enabled, they can be moved to underutilized machines during such peaks.

Thus providing efficient resource balancing.

b) Provide parallel CPU capacity:

The potential for massive parallel CPU capacity is one of the most attractive features of a grid.
Such computing power is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation,and many others.

But this has some barriers:

The **first barrier** depends on the algorithms used for splitting the application among many CPUs.
The second barrier appears if the parts are not completely independent; this can cause contention.

c) Access to additional and specialized resources:

For example, if a user needs to increase his total bandwidth of Internet to implement a data mining search engine. This can be split among grid machines that have independent connections to the Internet. Thus the total searching capability is multiplied. If the machines had shared the connection to the Internet, there would not have been an effective increase in bandwidth.
Some machines on the grid may have special devices, which can be used by others who do not have it. The grid also can enable more elaborate access, potentially to remote medical diagnostic and robotic surgery tools with two-way interaction from a distance.

d) Virtual resources and virtual organization:

Another important grid computing contribution is to enable and simplify collaboration among a wider audience.
Sharing which starts with data (data grid) in the form of files or databases.
Sharing is not limited to files, but also includes many other resources, such as equipment, software, services, licenses, and others. These resources are **"virtualized"** to give them a more uniform interoperability among heterogeneous grid participants.

e) Reliability :

High-end conventional computing systems use expensive hardware to increase reliability. The machines also use duplicate processors with hot pluggability so that when they fail, one can be replaced without turning the other off. Power supplies and cooling systems are duplicated. All of this builds a reliable system, but at a great cost, due to the duplication of high-reliability components.

But Grid computing allows us to see an alternate approach to reliability that relies more on software technology than expensive hardware.

Thus, if there is a power or other kind of failure at one location, the other parts of the grid are not likely to be affected. Grid management software can automatically resubmit jobs to other machines on the grid when a failure is detected.

In critical, real-time situations, multiple copies of the important jobs can be run on different machines throughout the grid.

Such grid systems uses **"autonomic computing."** This is a type of software that automatically heals problems in the grid, perhaps even before an operator or manager is aware of them.

3. Resources provided by Grid.

a) Computation :

The most common resource provided by the processors of the machines are the computing cycles. The processors can vary in speed, architecture, software platform, and other associated factors, such as memory, storage, and connectivity.

There are **three primary ways** to exploit the computation resources of a grid.

The **first and simplest** is to run an existing application on an available machine on the grid rather than locally.

The **second** is to use an application designed to split total work in such a way that the separate parts can execute in parallel on different processors.

The **third** is to run an application that needs to be executed many times on many different machines in the grid.

b) Storage :

The second most common resource provided in a grid is data storage. Each machine provides some quantity of storage for grid use, even if temporary. Storage can be memory attached to the processor or it can be "secondary storage" using hard disk drives or other permanent storage media.

Cache Memory attached to a processor is fast but is volatile. It would best be used to cache data to serve as temporary storage for running applications. Secondary storage in a grid can be used to increase capacity, performance, sharing, and reliability of data. Many grid systems use mountable networked file systems, such as Andrew File System (AFS), Network File System (NFS), Distributed File System (DFS), or General Parallel File System (GPFS). Which offers varying degrees of performance, security and reliability features to suit the requirement of the grid.

More advanced file systems on a grid can automatically duplicate sets of data, to provide redundancy for increased reliability and increased performance.

Data striping can also be implemented by grid file systems, When there are sequential or predictable access patterns to data, this technique can create the virtual effect of having storage devices that can transfer data at a faster rate than any individual disk drive. This is an important tool for multimedia data streams or when collecting large quantities of data at extremely high rates from CAT scans.

c) Software and License :

The grid may have software installed that may be too expensive to install on every grid machine. Using a grid, the jobs requiring this software are sent to the particular machines on which this software happens to be installed. When the licensing fees are significant, this saves expenses for an organization.

Some software licensing arrangements permit the software to be installed on all of the machines of a grid but may limit the number of instances that can be simultaneously used at any given instant. License management software keeps track of how many concurrent copies of the software are being used and prevents more than that number from executing at any given time.

d) Communication:

The rapid growth in communication capacity today makes grid computing practical. Therefore, it should not be a surprise that another important resource of a grid is data communication capacity.

Redundant communication paths are sometimes needed to better handle potential network failures and excessive data traffic.

4. Scheduling, Reservation and Scavenging:
Grid systems includes a job **"scheduler"** of some kind that automatically finds the most appropriate machine on which to run any given job that is waiting to be executed. Schedulers react to current availability of resources on the grid.
In a **"scavenging"** grid system, any machine that becomes idle would typically report its idle status to the grid management node. This management node would assign to this idle machine the next job that is satisfied by the machine's resources.
As a further step, grid resources can be **"reserved"** in advance for a designated set of jobs. Such reservations operate much like a calendaring system used to reserve conference rooms for meetings. This is done to meet deadlines and guarantee quality of service.

5. Types of Grid :
The **simplest grid** consists of just a few machines, all of the same hardware architecture and same operating system, connected on a local network. This kind of grid uses homogeneous systems so there are fewer considerations and may be used just for experimenting with grid software.
Machines participating in the grid may include those from multiple departments but within the same organization and heterogeneous systems. Such a grid is also referred to as an **"Intragrid."**
A grid may grow to cross organization boundaries, and may be used to collaborate on projects of common interest. This is known as an **"Intergrid".** The security and scalability increases with increase in size of grid.

6. Grid Construction :
Deployment grid practically does not need any extra Hardware only thing required is proper planning and security consideration.
Normally the machines in a grid are connected in LAN or WAN.

7. Grid Software :
Grid software consists of following components:

i) Management components:
Any grid system has some management components.
First, there is a component that keeps track of the resources available to the grid and which users are members of the grid. This information is used primarily to decide where grid jobs should be assigned.
Second, there are measurement components that determine both the capacities of the nodes on the grid and their current utilization rate at any given time. This information is used to schedule jobs in the grid.
Third, advanced grid management software can automatically manage many aspects of the grid. This is known as "autonomic computing," or "recovery oriented computing."

ii) Donor Software :
Each machine contributing resources typically needs to enroll as a member of the grid and install some software that manages the grid's use of its resources in the grid.
Some sort of identification and authentication procedure must be performed before a machine can join the grid. The grid system makes information about the newly added resources available throughout the grid.
The donor grid software must be able to receive the executable file or select the proper files from others.
The Job is executed and the output is sent back to the requester.

iii) Submission Software:
Usually any member machine of a grid can be used to submit jobs to the grid and initiate grid queries.
However, in some grid systems, this function is implemented as a separate component installed on "submission nodes" or "submission clients."

iv) Schedulers:
Most grid systems include some sort of job scheduling software. This software locates a machine on which to run a grid job that has been submitted by a user.

v) Communication:

A grid system may include software to help jobs communicate with each other. For example, an application may split itself into a large number of subjobs. Each of these subjobs is a separate job in the grid. The open standard Message Passing Interface (MPI) and any of several variations is often included as part of the grid system for just this kind of communication.

8. Grid from a users perspective:

Enrolling and installing grid software

- Enroll as Grid user
- Install Grid Software
- Authentication

Logging onto the grid

- Log in with User ID or Grid login ID in the Grid.
- Query Grid and Submit Job.

Query and Submitting Jobs

- Job submission which include I/P Data and executable programs.
- Execute the Program on Donating Machine.
- Result is sent to user

Monitoring progress and recovery

- User query the Grid system to see the progress of job.

A job may fail due to a

Programming error: The job stops in the middle with some program fault. This report can be sent back to submitter and subsequent actions can be taken.
Hardware or power failure: The machine or devices being used stop working in some way. These jobs can be given to suitable machines in the grid by grid management software.
Communications interruption: A communication path to the machine has failed or is overloaded with other data traffic. Grid management software takes proper action for it.
Excessive slowness: The job might be in an infinite loop or normal job progress may be limited by another process running at a higher priority or some other form of contention in that machine of the grid. So the job can be given to some other node in the grid.

9. Grid: from an Administration perspective:

Planning: One should start by deploying a small grid first, to test its installation and management, before having to confront more complicated issues involved with a large grid.
Installation: First, the selected grid system must be installed on an appropriately configured set of machines. These machines should be connected using networks with sufficient bandwidth to other machines on the grid.
Machines should be configured and connected to facilitate recovery scenarios.
Any critical data bases or other data essential for keeping track of the jobs in the grid, members of the grid, and machines on the grid should have suitable backups.
After installation, the grid software may need to be configured for the local network address and IDs.
Once, the grid is operational, there may be application software and data that should be installed on donor machines as well.

10. Managing, enrollment of donors and users

Certificate authority :

The primary responsibilities of a certificate authority are:

- Positively identify entities requesting certificates.
- Issuing, removing, and archiving certificates.
- Protecting the certificate authority server.
- Maintaining a namespace of unique names for certificate owners.
- Serve signed certificates to those needing to authenticate entities.
- Logging activity.
- Resource management.
- Data sharing.

11. Grid from an Application Developer's perspective:

To develop a:

- Applications that are not enabled for using multiple processors but can be executed on different machines.
- Applications that are already designed to use the multiple processors of a grid.

- Applications that need to be modified or rewritten to better exploit a grid.

12. Conclusion:

To conclude it should be noted that not every application is suitable or enabled for running on a grid. Some kinds of applications simply cannot be parallelized. For others, it can take a large amount of work to modify them to achieve faster throughput.

The configuration of a grid can greatly affect the performance, reliability, and security of an organization's computing infrastructure.

For all of these reasons, it is important for us to understand how far the grid has evolved today and which features are coming tomorrow or in the distant future.

The Globus toolkit is a set of tools useful for building a grid. Its strength is a good security model, with a provision for hierarchically collecting data about the grid, as well as the basic facilities for implementing a simple, yet world-spanning grid. Globus will grow over time through the work of many organizations that are extending its capabilities. More information about Globus can be obtained at http://www.globus.org.

Open Grid Services Architecture (OGSA) is an open standard at the base of all of these future grid enhancements. OSGA will standarize the grid interfaces that will be used by the new schedulers, *autonomic computing agents, and any number of* other services yet to be developed for the grid.

More information about OGSA can be obtained at http://www.globus.org/ogsa