2011 2nd International Conference on Networking and Information Technology
IPCSIT vol.17 (2011) © (2011) IACSIT Press, Singapore

High Speed Internet Access using Grid Computing

R Dinesh and C.Ajit Nataraj

SSN College of Engineering, Kalavakkam,Chennai,India-603 110

1. Introduction

Grid computing is all about to create the illusion of a simple yet large and powerful self managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. A 'resource' may be computing power, data storage, analysis/visualization tools, scientific hardware (a spectrometer, telescope and so on) or any other remote resource. The Grid aims to bring all these together -- in a virtual sense -- allowing large or small scale resource sharing and collaboration.

The standardization of communications between heterogeneous systems created the Internet explosion. The uniformity for sharing resources, along with the availability of higher bandwidth, are driving a possibly equally large evolutionary step in grid computing.

In a nutshell the grid computing is all about using the wasted CPU cycles in the network as one virtual machine, which can compete with supercomputer speed.

2. Genesis of Grid Computing

Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated. In this view of grid computing, computing becomes pervasive and individual users gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system.

3. GRID Oranization

The Grid architecture has a layered approach and each layer is defined in their, own behavioral manner to work independently. On integration it works as one complete well-organized grid model.

3.1 Fabric:

Fabric is the lowest layer in grid architecture. Unlike in normal computer architecture where the lowest layer represents logic gates, the fabric is an abstract layer which represents local computing resources such as storage, networking and computational resources.

3.2 Connectivity:

Connectivity layer connects several fabrics into one giant node of “fabric”. Connectivity layer provides secure connections and is implemented using network protocols such as Internet protocol (TCP/IP) and application protocols (DNS), etc.

3.3 Resource:

Resource layer deals with management of many connectivity layers. Resource layer can be information protocols used to obtain information about configuration, load and usage policies, and management protocol that negotiate the policies for handling resource requirement and operations.

3.4 Collective:

Collective layer consist of the protocols of interactions among several different resources. This layer includes directory services, accounting payment, collaboration services, and scheduling services to name a few.

3.5 Application:

Application layer is the highest layer in grid computing architecture. This layer calls other layers to perform desired actions. Application layer is simply the program we are working with to solve our problems.

4. Grid's & TYPES

The three primary types of grids and are summarized below:

- **Computational grid**
- **Scavenging grid**
- **Data grid**

5. Grid Model Proposed

We are using the **Scavenging Grid** for our implementation as large numbers of desktop machines are used in our Grid and later planning to extend it by using both Scavenging and data Grid. Figure2 gives an idea about the Grid that we have proposed.

6. Problems Due to Multiple Downloading

While accessing Internet most of us might have faced the burden of multiple downloading and in particular with downloading huge files i.e., there can be a total abrupt system failure while a heavy task is assigned to the system. The system may hang up and may be rebooted while some percentage of downloading might have been completed. This rebooting of the system leads to download of the file once again from the beginning, which is one of the major problems everyone is facing today.

7. Employing the Globus Architecture IN Proposed Grid

While planning to implement a Grid project, we must address issues like security, managing and brokering the workload, and managing data and resources information. Most Grid applications contain a tight integration of all these components.

The Globus architecture represents a multiple-layer model. The local services layer contains the operating system services and network services like TCP/IP. In addition, there are the services provided by cluster scheduling software (like IBM Load Leveler) -- job-submission, query of queues, and so forth. The cluster scheduling software allows a better use of the existing cluster resources. The higher layers of the Globus model enable the integration of multiple or heterogeneous clusters.

The Globus Toolkit (GT) was developed by Global Alliance, a division of Global Grid Forum. Global Alliance comprises of R&D research groups based at several universities such as the University of Chicago, the University of Edinburgh and the University of Southern California. GT is the de facto standard for grid computing [2] and it is comprised of 3 main services:

- **The core services**
- **Security services**
- **Data/Resource Management**

8. Grid Security Infrastructure (GSI)

GSI provides elements for secure authentication and communication in a grid. The infrastructure is based on the SSL protocol (Secure Socket Layer), public key encryption, and x.509 certificates. For a single sign-on, Globus add some extensions on GSI. It is based on the Generic Security Service API, which a standard API is promoted by the Internet Engineering Task Force (IETF).

These are the main functions implemented by GSI:

- Single/mutual authentication
- Confidential communication
- Authorization
- Delegation

9. Grid Resource Allocation Manager (GRAM):

GRAM is the module that provides the remote execution and status management of the execution. When a job is submitted by a client, the request is sent to the remote host and handled by the gatekeeper daemon located in the remote host. Then the gatekeeper creates a job manager to start and monitor the job. When the job is finished, the job manager sends the status information back to the Client and terminates.

Global Access to Secondary Storage (GASS) GRAM uses GASS for providing the mechanism to transfer the output file from servers to clients. Some APIs are provided under the GSI protocol to furnish 138 Introduction to Grid Computing with Globus Secure transfers. This mechanism is used by the globusrun command, gatekeeper, and job manager.

10. Monitoring and Discovery Service (MDS)

MDS provides access to static and dynamic information of resources. Basically, it contains the following components:

- Grid Resource Information Service (GRIS)
- Grid Index Information Service (GIIS)
- Information Provider
- MDS client

11. Grid Resource Information Service (GRIS)

GRIS is the repository of local resource information derived from information providers. GRIS is able to register its information with a GIIS, but GRIS itself does not receive registration requests. The local information maintained by GRIS is updated when requested, and cached for a period of time known as the time-to-live (TTL). If no request for the information is received by GRIS, the information will time out and be deleted. If a later request for the information is received, GRIS will call the relevant information provider(s) to retrieve the latest information.

12. Grid Index Information Service (GIIS)

GIIS is the repository that contains indexes of resource information registered by the GRIS and other GIISs. It can be seen as a grid wide information server. GIIS has a hierarchical mechanism, like DNS, and each GIIS has its own name. This means client users can specify the name of a GIIS node to search for information.

13. Security

Security is a much more important factor in planning and maintaining a grid than in conventional distributed computing, where data sharing comprises the bulk of the activity. In a grid, the member machines are configured to execute programs rather than just move data. This makes an unsecured grid potentially fertile ground for viruses and Trojan horse programs. For this reason, it is important to understand exactly which components of the grid must be rigorously secured to deter any kind of attack. Furthermore, it is important to understand the issues involved in authenticating users and properly executing the responsibilities of a Certificate Authority.

14. 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

15. 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. In the simplest cases, it may just blindly assign jobs in a round-robin fashion to the next machine matching the resource requirements. However, there are advantages to using a more advanced scheduler. Some schedulers implement a job priority system. This is sometimes done by using several job queues, each with a different priority. As grid machines become available to execute jobs, the jobs are taken from the highest priority queues first. Policies of various kinds are also implemented using schedulers. Policies can include various kinds of constraints on jobs, users, and resources. For example, there may be a policy that restricts grid jobs from executing at certain times of the day.

More advanced schedulers will monitor the progress of scheduled jobs managing the overall work-flow. If the jobs are lost due to system or network outages, a good scheduler will automatically resubmit the job elsewhere. However, if a job appears to be in an infinite loop and reaches a maximum timeout, then such jobs should not be rescheduled. Typically, jobs have different kinds of completion codes, some of which are suitable for re-submission and some of which are not.

16. Monitoring Progress and Recovery

A grid system, in conjunction with its job scheduler, often provides some degree of recovery for sub jobs that fail. A job may fail due to a:

- Programming error: The job stops part way with some program fault.
- Hardware or power failure: The machine or devices being used stop working in some way.
- Communications interruption: A communication path to the machine has failed or is overloaded with other data traffic.
- 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.

17. Communications

A grid system may include software to help jobs communicate with each other. 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

18. Accessing the Intranet Grid

When any user wants to access our proposed Intranet Grid in order to download multiple files over the Internet, then he should follow certain procedures that we consider necessary for the security of our Grid. The main Requirements for Processing in Grid Environment are:

- **Security:** Single sign-on, authentication, authorization, and secure data transfer.
- **Resource Management:** Remote job submission and management.
- **Data Management:** Secure and robust data movement.
- **Information Services:** Directory services of available resources and their status.
- **Fault Detection:** Checking the intranet.
- **Portability:** C bindings (header files) needed to build and compile programs

19. Existing Algorithm for Globus Architecture

[1]. Create security proxy via GSI services
[2]. Access a MDS-GIIS server
[3]. Search for required machine(s)
[4]. Rank the machine list based on a scheduling policy
[5]. Prepare the data
[6]. Transfer the data to the target machine by using GASS services
[7]. Prepare a RSL document
[8]. Submit the program using GRAM services
[9]. React to status changes from GRAM
[10]. Get results via GASS

Here, we have got the resources available in the Network which is automatically done by have the Globus Toolkit in the server. When we want to download a file this information has to be matched with the client module and then the downloading has to be carried out in the clients. For this we have added some modules to the Grid Architecture.

Added module:

[1]. Get the Information about files to be downloaded.
[2]. Match the files with appropriate Machines.
[3]. Store files in common database.Retrieval of data from database is done after proper authentication.

20. Pproposed Algorithm for Intranet Proposed Grid

Steps to perform multiple downloading in the Grid, The host details are got from the server of the LAN in order to identify the various hosts.

The host information is got whenever needed on the priority queue basis.

//module for downloading files

[1]. Start lookup // look for file size and resource information
[2]. Declare nres, nfile // no of resources available and no of files
[3]. Input nres, nfiles
[4]. Input size // the file size
[5]. Initialize P1 → res info // store the resource information in priority queue P1 with highest system configuration as priority
[6]. Initialize P2 → file size // store the file information in the priority queue P2 with maximum file size as priority
[7]. If condition (nfiles == nres) // check whether the no of resources is equal to no of files
[8]. Initialize counter
[9]. For (counter =1; counter <= nres; counter++) // initialize the loop to assign the files.
[10]. Assign the 1st file of P2 to the 1st node in P1. // first node will be node with highest configuration and first file will be the file with maximum size.
[11]. Start processing // files directed to the appropriate system for accessing their wasted CPU cycles.
[12]. Loop
[13]. Else:
[14]. Start timer
[15]. Delay → 1 min
[16]. Collect incoming files // the files that the user clicked to download in this duration.
[17]. Assign the files → P2
[18]. Goto step 8
[19]. Goto step 1
[20]. End // when the user exits from proposed Grid.

21. Conclusion

Grid computing was once said to be fading out but due to the technological convergence it is blooming once again and the Intranet Grid we have proposed adds a milestone for the Globalization of Grid

Architecture, which, leads to the hasty computing that is going to conquer the world in the nearest future. By implementing our proposed Intranet Grid ***it is very easy to download multiple files very fast and no need to worry about the security*** as we are authenticating each and every step taking place in our Grid and in particular user to access the database. Further implementations could be carried out in the nearest future.

22. References

[1]. P Plaszczak, R Wellner, Grid computing, 2005, Elsevier/Morgan Kaufmann, San Francisco

[2]. Francesco Lelli, Eric Frizziero, Michele Gulmini, Gaetano Maron, Salvatore Orlando, Andrea Petrucci and Silvano Squizzato. The many faces of the integration of instruments and the grid. International Journal of Web and Grid Services 2007 – Vol. 3, No.3

[3]. Fran Berman, Geoffrey Fox, Anthony J. G. Hey Grid computing, 2005, Elsevier/Morgan Kaufmann, San Francisco

[4]. Pawel Plaszczak, Richard Wellner, José Cardoso Cunha, Omer Rana, Lizhe Wang, Wei Jie, Jinjun ChenFundamentals of Grid Computing: Theory, Algorithms and Technologies BY Frédéric

[5]. Advances in grid computing -- EGC 2005: European Grid Conference, ... Peter Sloot

[6]. Grid computing: the new frontier of high performance computing-Lucio Grandinetti

[7]. Distributed data management for grid computing-Michael Di Stefano