

# LARGE FILE ENCRYPTION AND DECRYPTION USING GRID ENGINE

Mrs. G.Hemalatha , HOD/MCA Dept, Paavai College of Engineering , Pachal.<sup>1</sup>

E-mail: hem\_vigbs@yahoo.co.in

Dr. M.V. Srinath, Prof/CSE Dept, Mahendra Engineering College, Mahendrapuri.<sup>2</sup>

E-mail: sri\_induja@rediffmail.com

## Abstract

"Distributed" or "Grid" computing in general is a special type of parallel computing which relies on complete computers connected to a conventional network interface, such as ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus. In the commercial world, grid aims to maximize the utilization of an organization's computing resources by making them shareable across applications (sometimes called virtualization) and, potentially, provide computing on demand to third parties as a utility service. It enables the regulation of supply and demand for resources, provides an incentive for resource owners to participate in the grid and motivates the users to trade-off between deadline, budget, and the required level of quality-of-service. Computing is about making large amounts of compute power available for applications and users. Collaborative development of Java Grid Engine technology provides the proper development framework to ensure that Grid Engine technology meets the requirements of the largest number of users.

Grid Engine is helpful in three elementary ways. It can:

- optimally place computing tasks and balance the load on a set of networked computers
- allow users to generate and queue more computing tasks than can be run at the moment
- ensure that tasks are executed with respect to priority and to providing all users with a fair share of access over time.

Grid Engine presents users a seamless, integrated computing capability. Grid Engine is used to support a wide variety of requirements; for instance, where users start many interactive and batch tasks as in product design or financial simulations; where sets of repetitious tasks are run as in software; where large numbers of users are placing jobs on limited resources as in education environments.

## Keywords:

Grid Engine, Heterogeneous, Jini Technology

## 1.0 Introduction

Grid Computing is about making large amounts of computing power available for applications and users. Collaborative development of Java Grid Engine technology provides the proper development framework to ensure that Grid Engine technology meets the requirements of the largest number of users.

Grid engine is helpful in three elementary ways. It can:

- Optimally place computing tasks and balance the load on a set of networked computers
- Allow users to generate and queue more computing tasks than can be run at the moment
- Ensure that tasks are executed with respect to priority and to providing all users with a fair share of access over time.

Grid Engine presents users a seamless, integrated computing capability. Grid Engine is used to support a wide variety of requirements, for instance, where users start many interactive and batch tasks as in product design or financial simulations, where sets of repetitious tasks are run as in software where large numbers of users are placing jobs on limited resources as in education environments; and where users are launching parallel applications across massive numbers of processors for applications such as weather simulation.

## 2.0 Overview

Grid Computing is an advanced technology of distributed computing. A Grid is a collection of computers, storage and other devices which are joined together by any means of communication like internet and which can be used to manage information and solve their problems among themselves. In spite of this JINI technology also implemented in this concept.

## 3.0 Definition

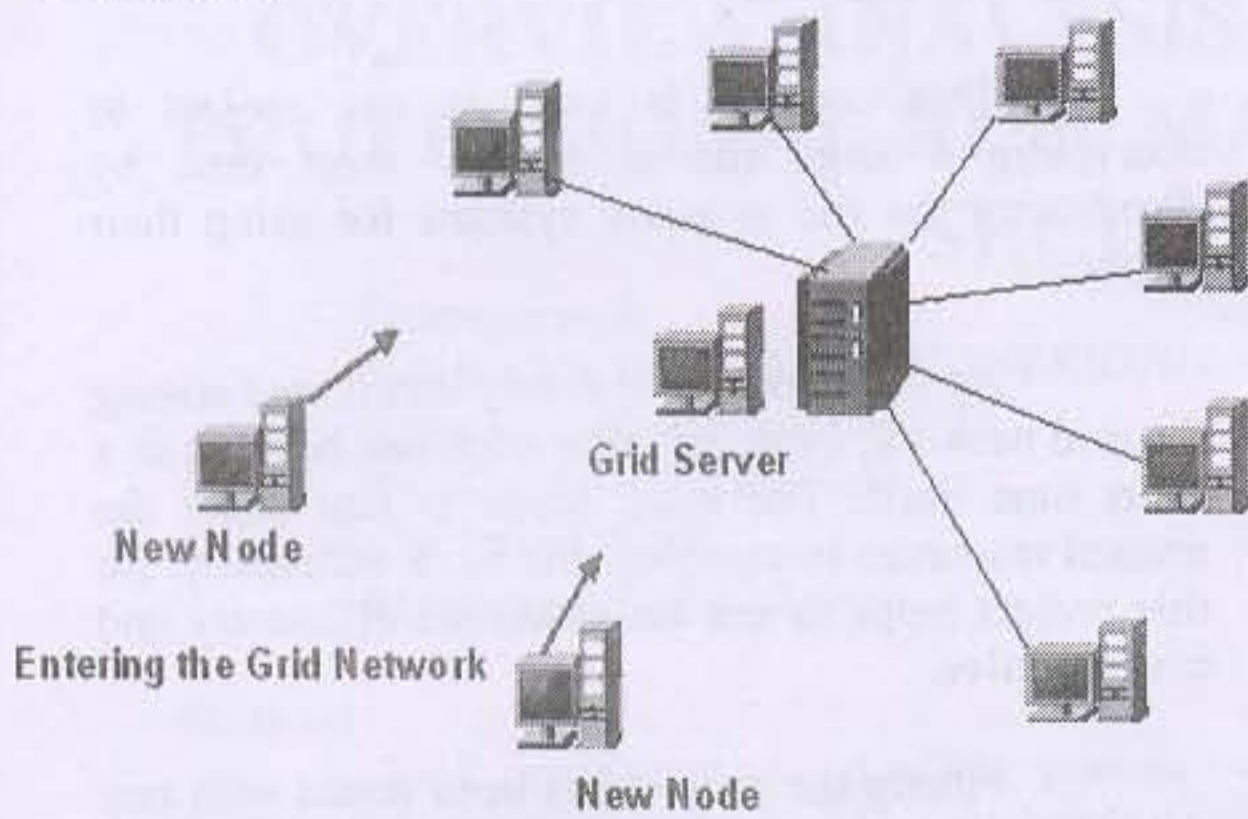
Grid Computing allows using the unutilized resources of other systems. This is achieved by distributing the workload of the system to the other systems in order to use their un-used resources such as their memory, Processor, etc which results in balancing the workload, decreasing the network traffic, bandwidth, etc. This concept is used in our project to encrypt and decrypt large file in a very short time by distributing the file to many systems for using their resources.

## 4.0. Scope

As the workload is evenly distributed among the grid network, even the large work can be done in a short time itself. The main scope is that using the unused resources to complete the work efficiently. This project helps to use the resources efficiently and cost effective.

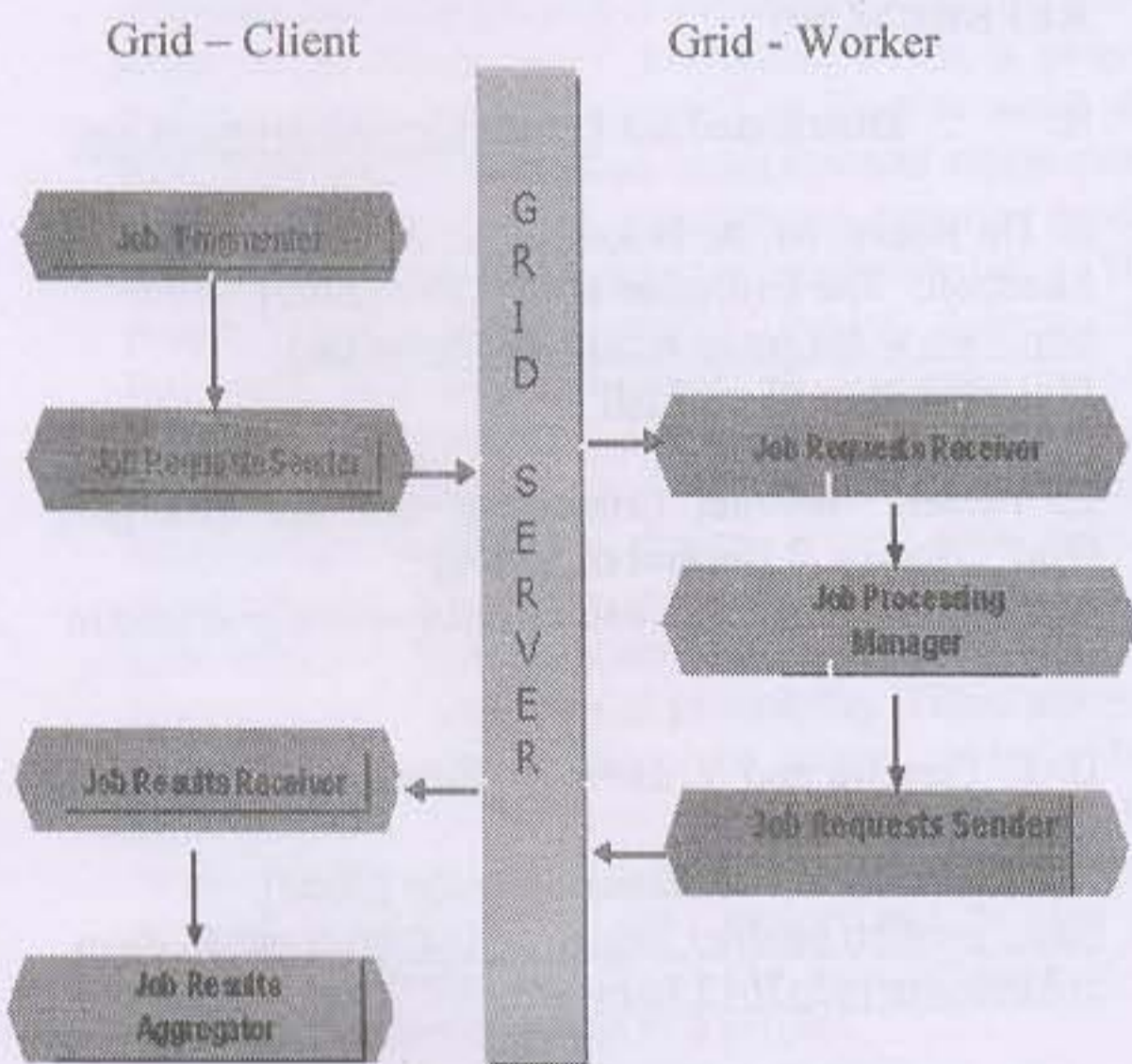


**5.0 Grid Engine Methodology**  
**Grid Nodes**



**Fig 1 Grid Nodes**

**Technical Overview Diagram**



**Fig 2 Grid Architecture**

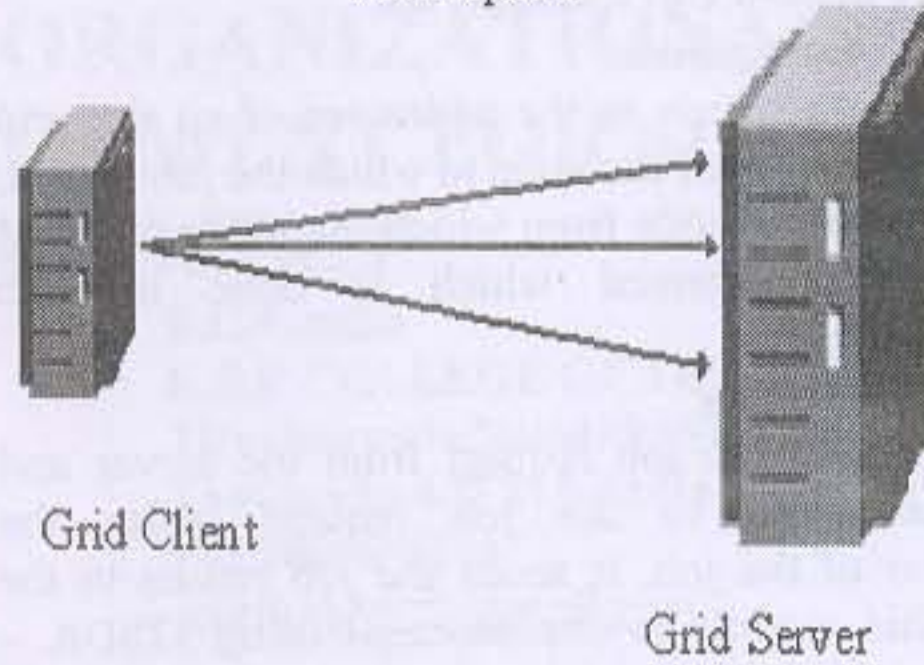
In the Grid Network, all the Systems connected are called the Grid Nodes. When a node wants to do the job, that node becomes the client and all other nodes are called as the Grid-Worker

**Grid - Client:**

The Grid-Client is the node which wants to do the job. It divides the job into many job requests and send it to the Grid-Server. The Client will also collect the job results from the Server and it will aggregate those collected job results and forms it into a whole end result.

This module is implemented using J2SDK – mainly using the classes in Java Util Package, etc. The following technical architecture represents the Grid – client working process.

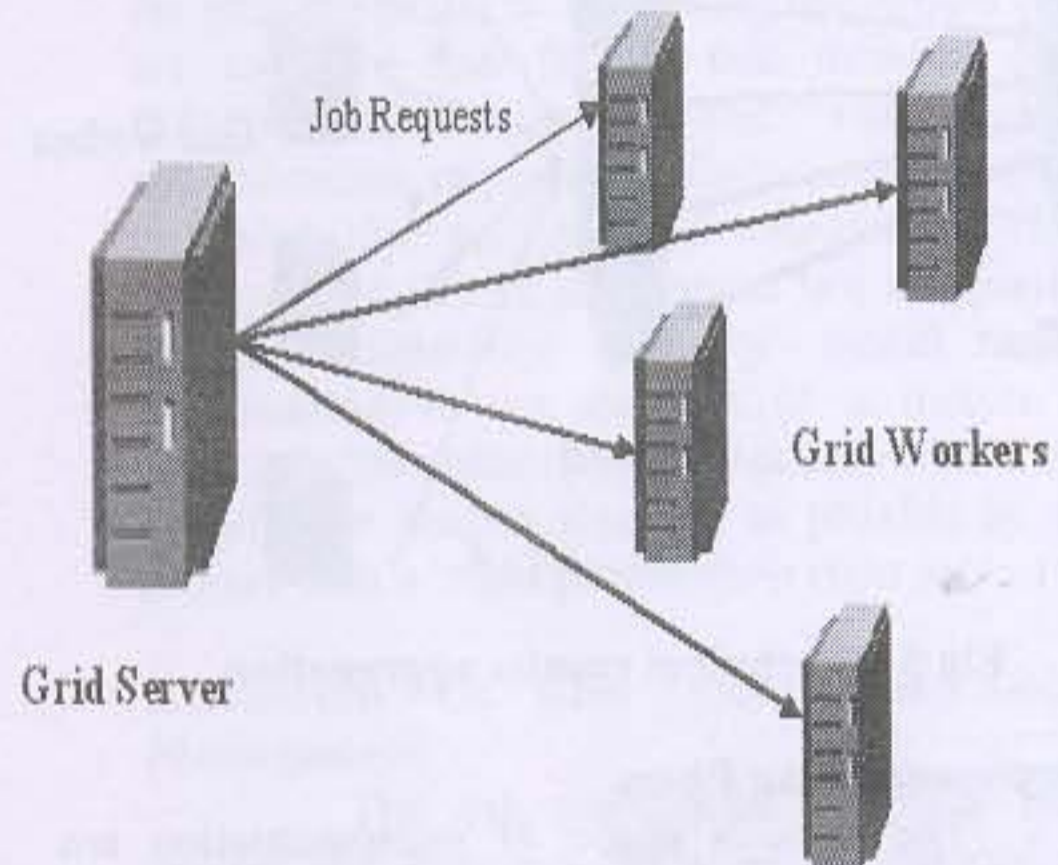
Job Requests



**Fig 3 Technical client architecture**

**Grid- Server:**

The following technical architecture represents working process of the Grid – server.



**Fig 4 - Technical server architecture**

In the Grid network, there will be a centralized Server and it comprise of many modules such as:

- **Security:**  
 In the Security module, the systems which are entering into the grid network should be authenticated and should be authorized. This can be achieved by using any encryption and decryption technique like SSL (Secured Socket Layer).
- **User Interface:**  
 In the User Interface module, the page or the interface is to be designed for the user to access the grid using HTML, JSP and Java AWT.
- **Workload Management:**  
 Monitors the workload such as the processing cycles of each nodes present in the grid network. This module is implemented using J2SDK – mainly using the classes in Java Util Package, etc
- **Job Scheduler**  
 Job Scheduler will schedule the job requests and maintains a queue in according to the workload manager. This module is implemented using J2SDK – mainly using the classes in Java Util Package, etc
- **Resource Management**  
 Monitors all the available resources and the status of the nodes present in the grid network. This



module is implemented using J2SDK—mainly using the classes in Java Util Package, etc

- **Data Management**

Data's such as the addresses of all the grid nodes, the address of the node to which the job is sent, the address of the node from which the job is received are to be maintained which is done in data management

- **Grid - Worker**

It receives the job request from the server and works according to the job request. After the completion of the job, it sends the job results to the server. This module is implemented using J2SDK – mainly using the classes in Java Util Package. The following technical architecture represents the working process of the Grid - worker

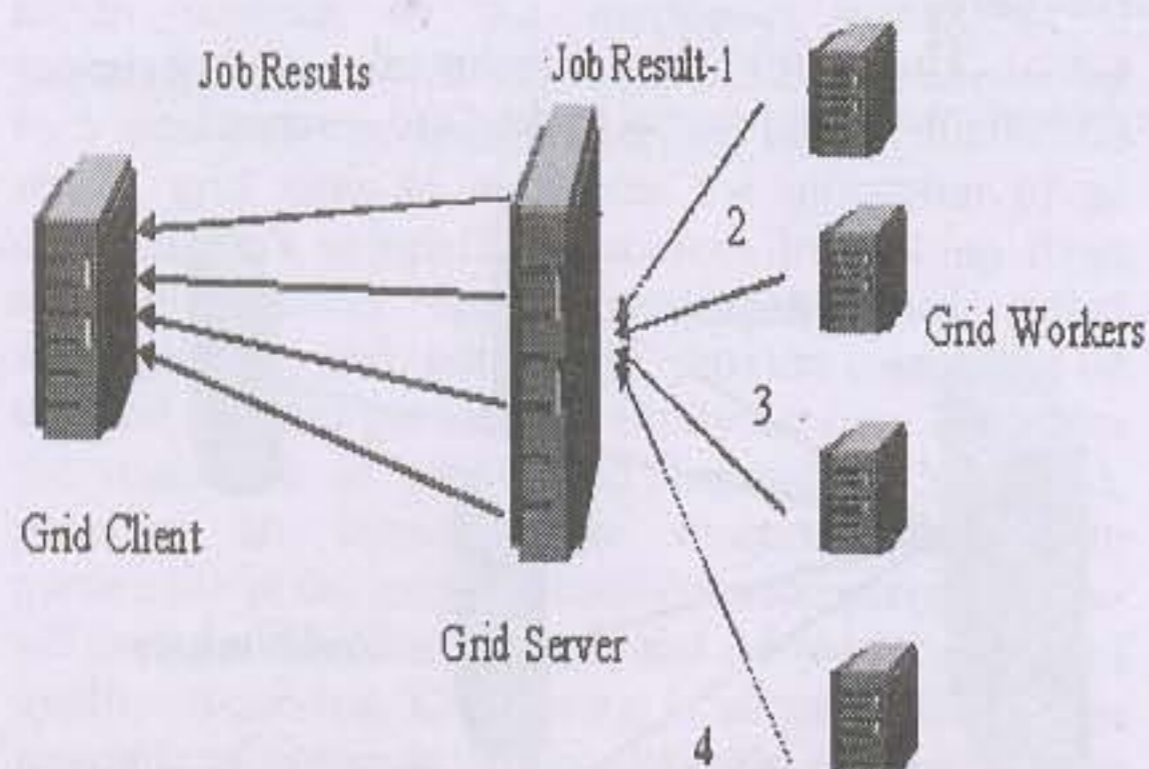


Fig 5 - Technical results aggregation

### 6.0 Implementation Phase

The various phase of implementation are given below :

**Phase I:**

The server is to be started. The File which is to be processed is opened in the client system. Now the Client will divide that file into job requests and send it to the server.

**Phase II:**

The server will be monitoring the Clients and Workers and when it receives the job request from the clients, it will distribute it to the workers according to their workload and resources.

**Phase III:**

The worker will receive it and do that job according to the job request. After the work gets over, it sends back to the server and it will send it to the respective client.

The Client will collect all the results from the server and aggregate it to form the end result.

### 7.0 CONCLUSIONS

Grid Computing allows using the unutilized resources of other systems. This is achieved by distributing the workload of the system to the other systems in order to use their un-used resources such as their memory, Processor, etc which results in

balancing the workload, decreasing the network traffic, bandwidth, etc.

This concept is used in our project to decrypting a large file in a very short time by distributing the file to many systems for using their resources.

As the workload is evenly distributed among the grid network, even the large work can be done in a short time itself. The main scope is that using the unused resources to complete the work efficiently. So this project helps to use the resources efficiently and cost effective.

Finally the project has been tested with real data and everything worked successfully. The project has fulfilled all the objectives identified. The user can obtain many advantages from this project.

### REFERENCES

A. Distributed.net [2002] <http://distributed.net/>

B. De Roure, M. A. Baker, N. R. Jennings and N. R. Shadbolt, The Evolution of the Grid [2002] <http://www.ecs.soton.ac.uk/~nrj/download-files/evolution-of-grid.pdf>

C. Foster. "Internet Computing and the Emerging Grid". *Nature*. 7 December. [2000] <http://www.nature.com/nature/webmatters/grid/grid.html>

D. L. Ferreira and V. Berstis. "Fundamentals of Grid Computing" IBM Redbooks. IBM Publications Center. [2002]. <http://publibb.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp3613.html>

E. Sun Microsystems, Sun Cluster Grid Architecture [2002]. <http://www.sun.com/software/grid/SunClusterGridArchitecture.pdf>

F. S. Tanenbaum. Modern Operating Systems. Upper Saddle River, New Jersey. Prentice-Hall. [2003.]

G. S. Tanenbaum. Computer Networks. Fourth Edition. Upper Saddle River, New Jersey. Prentice-Hall. [2002.]