

# Development of GRID Portal

B. Dhana Lakshmi

Department of Computer Science and Engineering  
ARYABHATA Institute of Technology & Science  
Jawaharlal Nehru Technological University  
On Srisailem Highway, R R Dist, Hyderabad, India

**Abstract:** The objective of this paper is to develop a framework for submitting the jobs and integrates those jobs to available systems by using a meta-scheduler. The meta-scheduler being used is the GridBus broker and the middleware being used is Globus. The GridSphere Portal is used to submit the jobs to the GridBus broker which in turn submits jobs to systems running Globus middleware by using scheduling algorithms and results are retrieved from them. Grid may consist of normal and scheduled PCs and it can also increase the number of systems.

**Keywords:** grid; meta scheduler; resource broker; middleware; grid sphere.

## 1. INTRODUCTION

A grid [1] is a collection of distributed resources that facilitates the sharing, distribution and aggregation of services depending on their performance and quality-of-service measurements. A resource on a grid could be any entity that provides access to a service. This could range from servers to databases, scientific instruments, applications and the like.

Users make use of resource brokers [2] to know the transparency of heterogeneous resources in the field of Grids and distributed systems. Meta-scheduler is one that is specially designed to schedule jobs across various grid middleware. The various grid middleware accept these jobs from the meta scheduler, process them, and return the results back to the meta-scheduler.

A meta-scheduler schedules work across a number of clusters or grids, each of which has its own independent scheduling solutions. It accepts the different application level local schedulers and employs the scheduling policies between different applications. Thus, this paper will be used to submit and execute the jobs from different applications by giving the results to applications.

## 2. PROPOSED SCHEME

In this scheme, the main objective is to integrate the GridSphere Portal with grid middleware's by using a meta scheduler. User submits his job through the GridSphere Portal which is submitted to the meta-scheduler.

The meta scheduler that we can use is GridBus Broker which is responsible for submitting the job to the middleware. Globus will act as a middleware. Once the jobs are run on the middleware the results are returned to the meta-scheduler which in turn returns to the portal and then to users to check the output.

The user can submit the job to the middleware of his/her choice and he provides the path of the application file which has to be run on the middleware and he can also specify the middleware on which he wants his job to be running. The application file is the actual application to be run and the resource file consists of all the resources available and the credentials required for accessing these resources.

## 3. GRID SPHERE PORTAL

It is developed using the GridSphere Portal Framework, which consists architecture for deploying “pluggable” web applications using the Portlet model that provides development environment for easily creating new portlet applications and offers a core set of portlets for the management of portlets, users, groups and layouts.

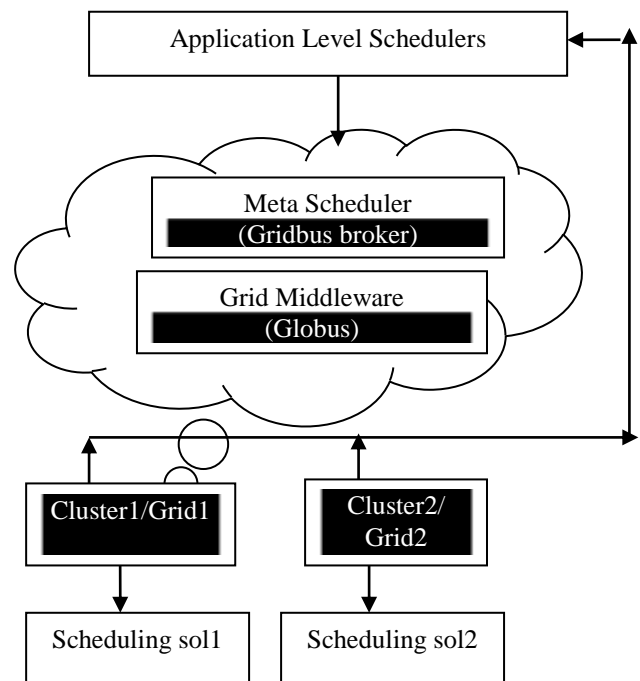


Figure 1. Gridsphere Portal

#### 4. SYSTEM DESIGN

System design is implemented by using the following components.

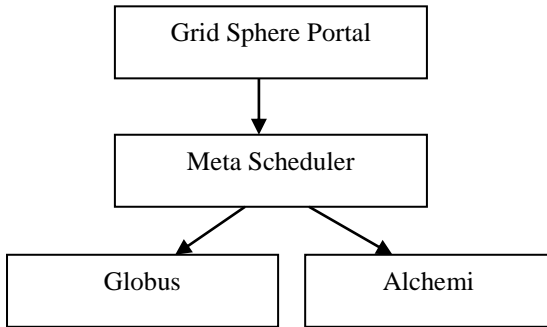


Figure 2. Proposed Design Architecture

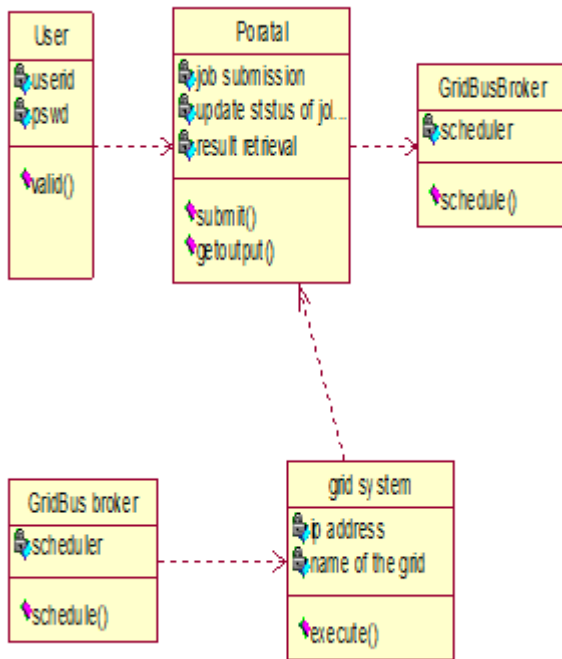


Figure 3. Gridsphere Portal Class Diagram

#### 4.1 Gridbus Broker

It is used to support [2] both computational and data grid applications and its architecture have importance on simplicity, and extensibility in executing the grid applications. Transparency is maintained in running grid nodes. Set the proxy certificates before running the broker and the device groups have to be manually targeted before running the broker as the jobs will not be dispatched otherwise.

When running the broker, it is needed to input the resource description, which specifies the available resources and describes their attributes. The broker is pointed to the location of the resource description file which contains the description of the resources that are to be used by the broker for executing the grid applications.

This file may be modified to specify the resources the user has access to. Each run of the broker creates a separate dimension.

#### 4.2 Broker Properties Configuration

The broker is configured in two ways, either by providing it configuration values or by creating an object to broker. Resources are classified into three types as compute, storage and service resources. Compute resources are servers to which the user's jobs are submitted for execution.

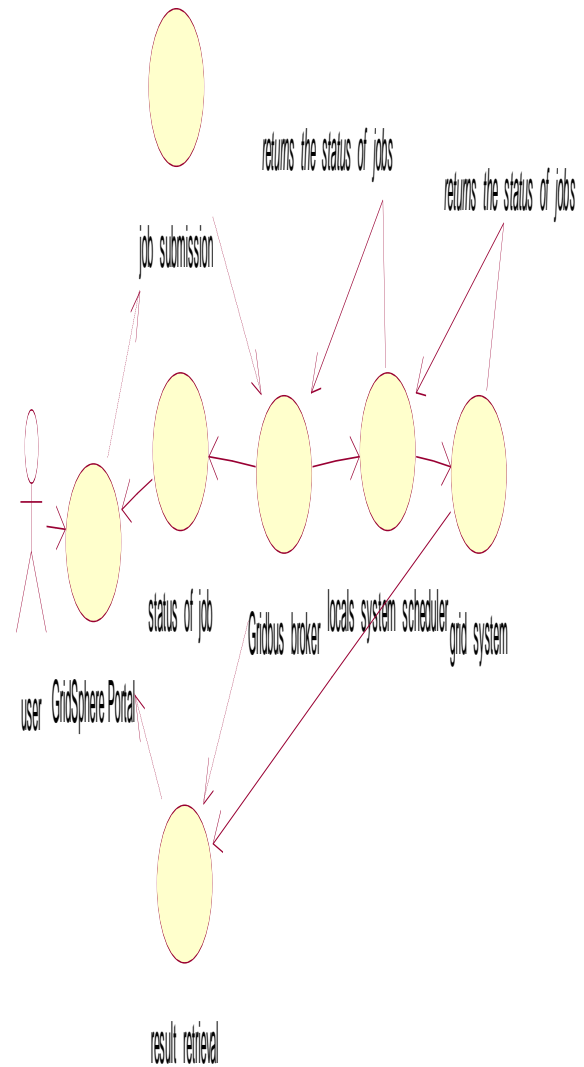


Figure 4. Gridsphere Portal Usecase Diagram

Storage resources are used to store the results of execution and hence can be considered as data sinks and which provide generic services that can be used by the broker. A service resource can be of two types-information services and application services.

Application services provide applications hosted on nodes that can be accessed as a service. Information services are entities which provide information about other resources or services.

Storage resources are used to store the results of execution and hence can be considered as data sinks and which provide generic services that can be used by the broker. A service resource can be of two types-information services and application services.

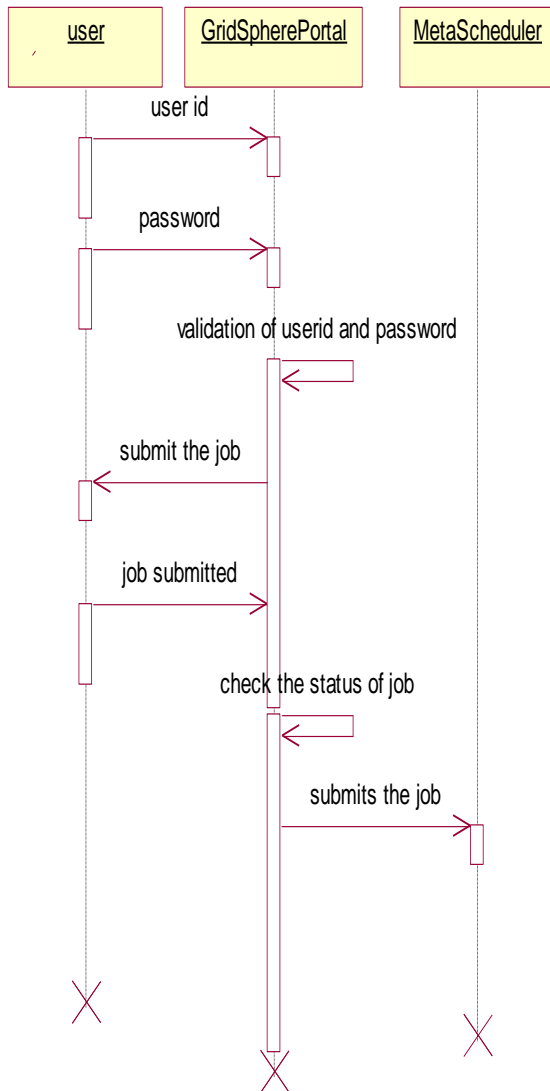


Figure 5. Gridsphere Portal Sequence Diagram1

A compute resource is associated with a domain that can take two values local and remote. Local resources could be the local computer or a cluster, on which the broker runs. Remote compute resources are used to represent nodes on the grid which have a job submission interface accessible via a network.

### 4.3 Broker Properties Configuration

The broker is configured in two ways, either by providing it configuration values or by creating an object to broker. Resources are classified into three types as compute, storage and service resources. Compute resources are servers to which the user's jobs are submitted for execution.

Application services provide applications hosted on nodes that can be accessed as a service. Information services are entities which provide information about other resources or services. A compute resource is associated with a domain that can take two values local and remote. Local resources could be the local computer or a cluster, on which the broker runs. Remote compute resources are used to represent nodes on the grid which have a job submission interface accessible via a network.

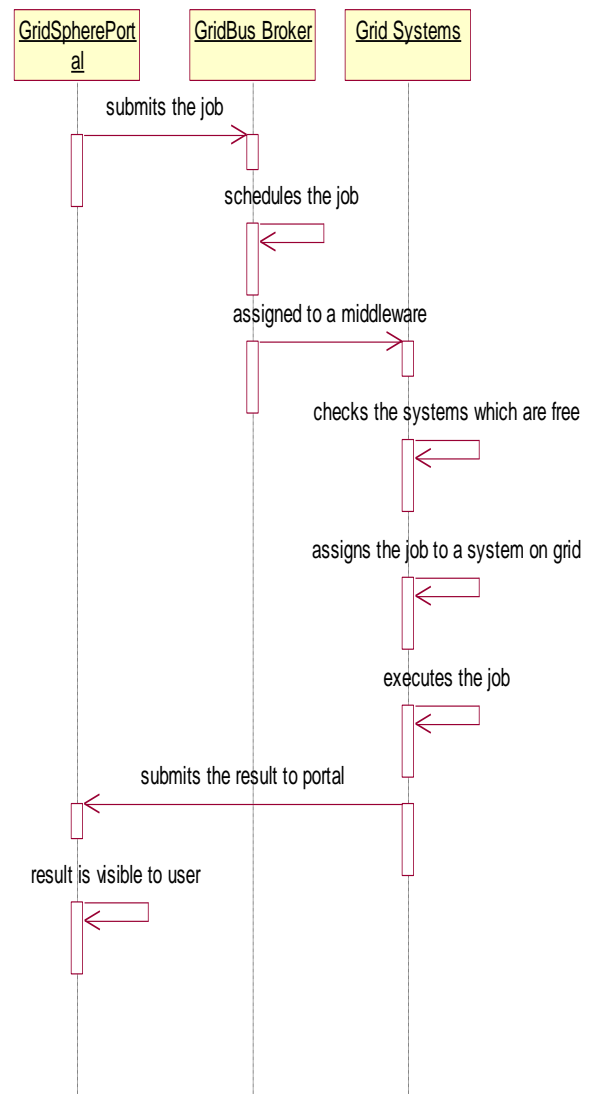


Figure 6. Gridsphere Portal Sequence Diagram2

### 4.4 Resource Description

This is needed by the Broker, which is used to describe two types of entities resources and to access the resources.

## 4.5 Broker Entities

### 4.5.1 Farming Engine

It is the central component which maintains the overall state of the broker at all times. It is the glue that binds all the components together. It acts as a container for the job and server collections.

It is the component that interacts with external applications and initiates the scheduling. The farming engine can be considered as the broker's in-memory database, as it holds the broker's current state at any point of time.

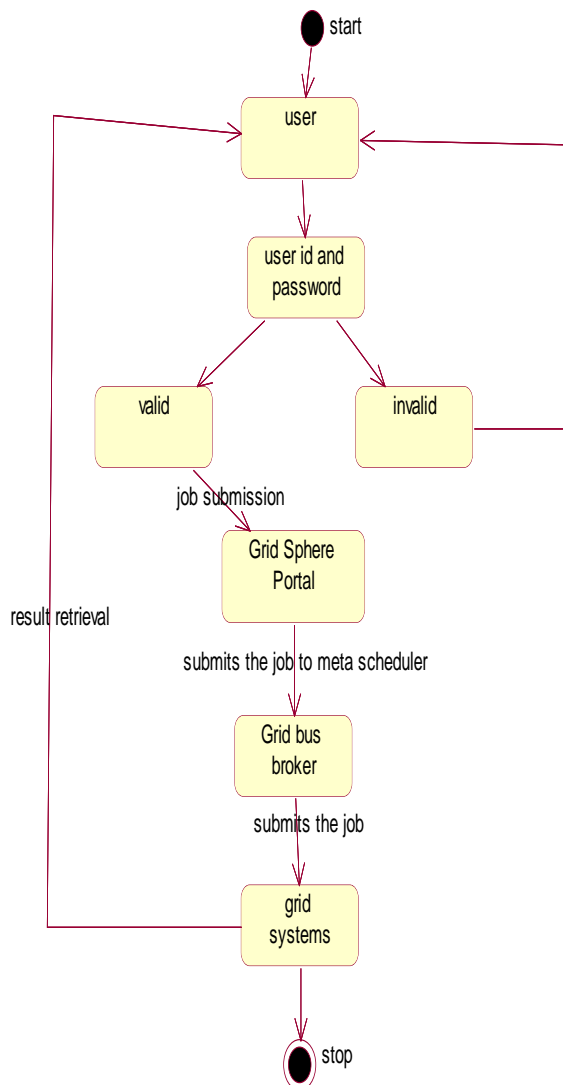


Figure 7. Gridsphere Portal Activity Diagram

### 4.5.2 Scheduler

Mapping of jobs to nodes in a grid platform is done by an autonomous component [1] which is known as a scheduler. It can schedule jobs based on metrics which do not depend on the underlying platform.

### 4.5.3 Job

It is an abstraction for a unit of work assigned to a node. It consists of variables and a task. A variable [1] consisting of a single value for a job, is called a single variable. A task is the description of what has to be done by the job.

It is composed of a set of commands. Job monitor is responsible for monitoring the execution of all jobs submitted to the node corresponding to the middleware.

### 4.5.4 Data Hosts & Data File Objects

These are nodes on which data files have been stored. These objects store the details of the data files that are stored on them such as their path on the disk and the protocol used to access them.

The data host objects also maintain a list of the compute resources sorted in the descending order of available bandwidth from the host. Data file objects store attributes of input files that are required for an application such as size and location. A data file object links to the different data hosts that store the file. Overall, the broker is designed to be a loosely coupled set of components working together.

### 4.5.5 Algorithm Steps

Algorithm for submission and monitoring cycle for job sent to a Globus node consists of following steps..

1. The scheduler submits a job to a server.
2. The server puts the job in its local job buffer and informs to server manager.
3. The server manager creates a middleware specific agent and sends it to remote machine.

### 4.5.6 Realization as a Framework.

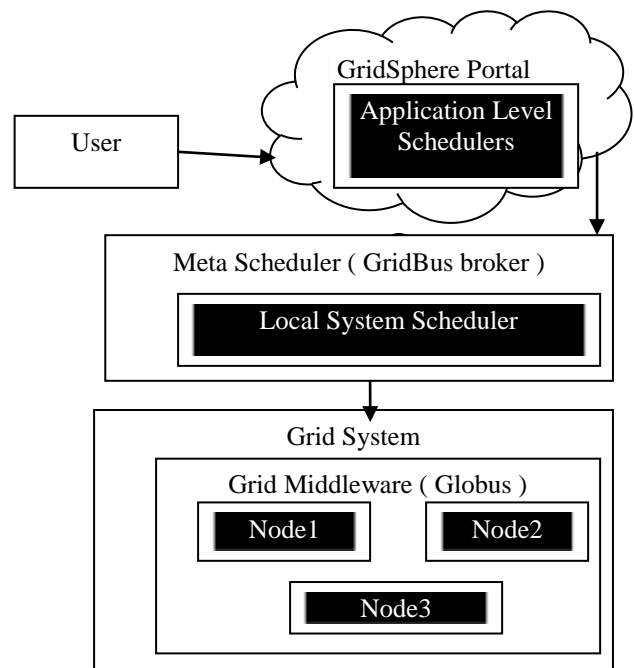


Figure 8. Basic Design Model



### 5.3 Monitor Portlet

After submitting the job the user can check or monitor the status of the job through this portlet. It gives the status about each job and the resource on which the job is running. The status of the job could be whether the job is waiting or the job is waiting or the job is running or the job is done or the job has failed.

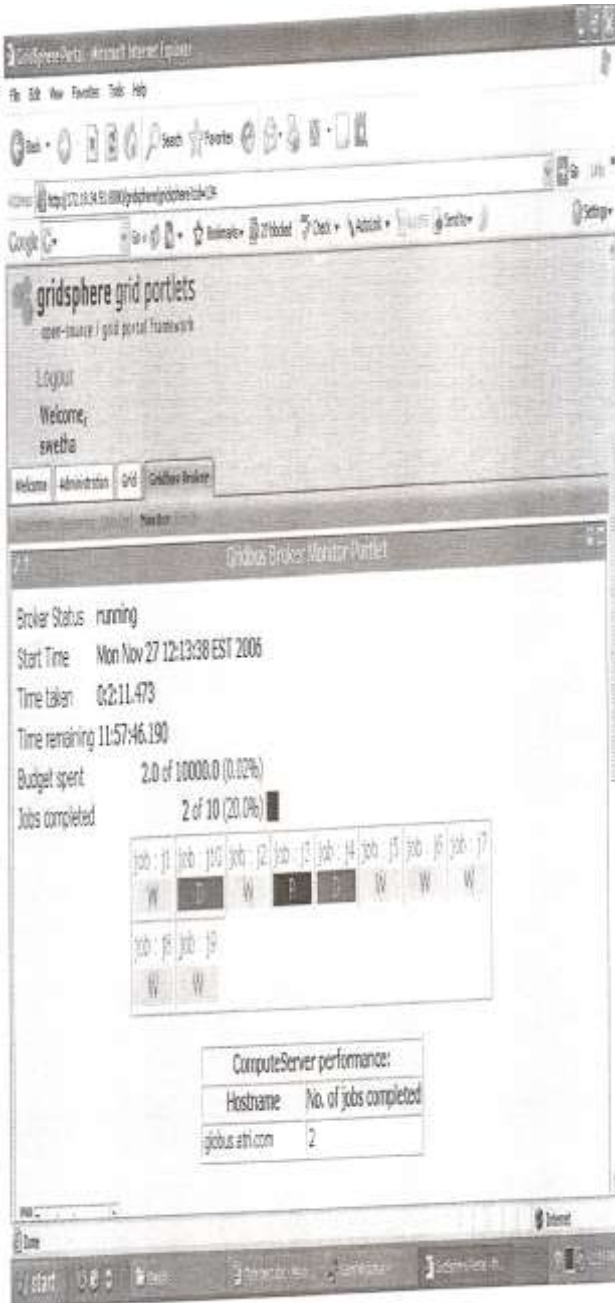


Figure 9: Monitor Portlet

Figure 11. Monitor Portlet

### 5.4 Result Portlet

Once the execution of all jobs are done the GridBus Broker returns the results as a link to the result file on the result portlet and then the user can download the output files and check for the results.

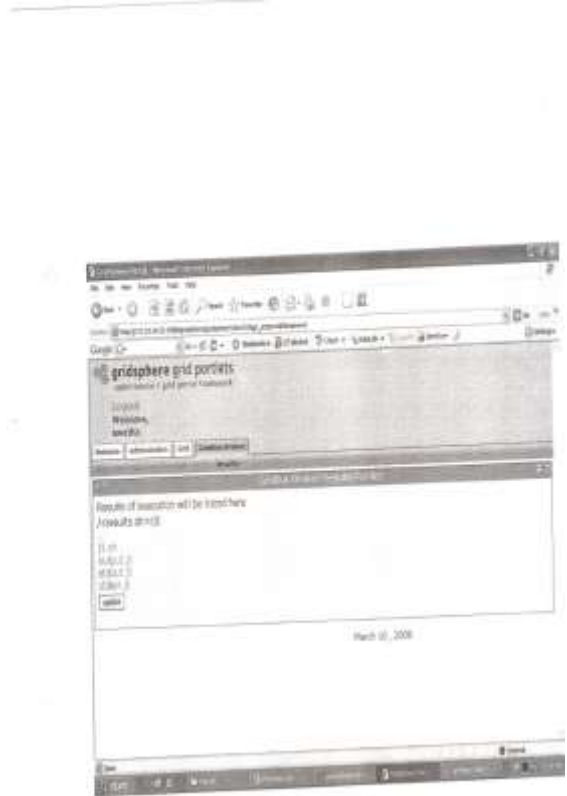


Figure 10: Result Portlet

Figure 12. Result Portlet

### 5.5 Login Page

A GridSphere Portal Login page is as shown below and this is the entry point for users and then users can start developing their own portals.

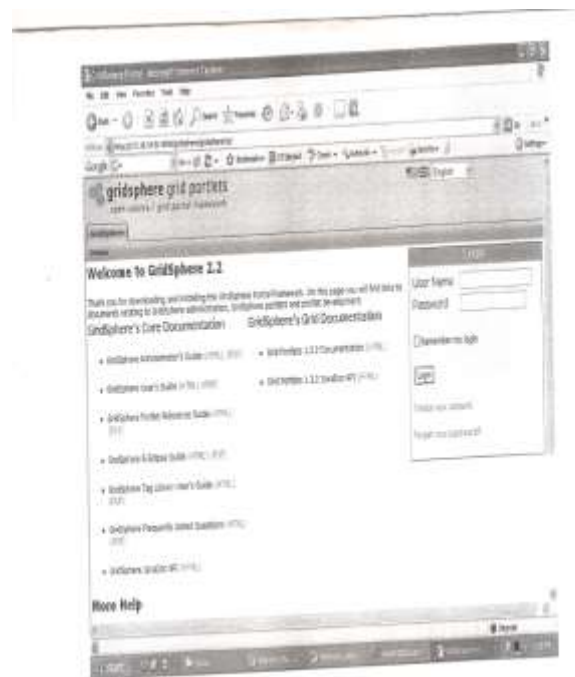


Figure 14: Login Portlet

Figure 13. Login Page



## 5.6 User QOS Portlet

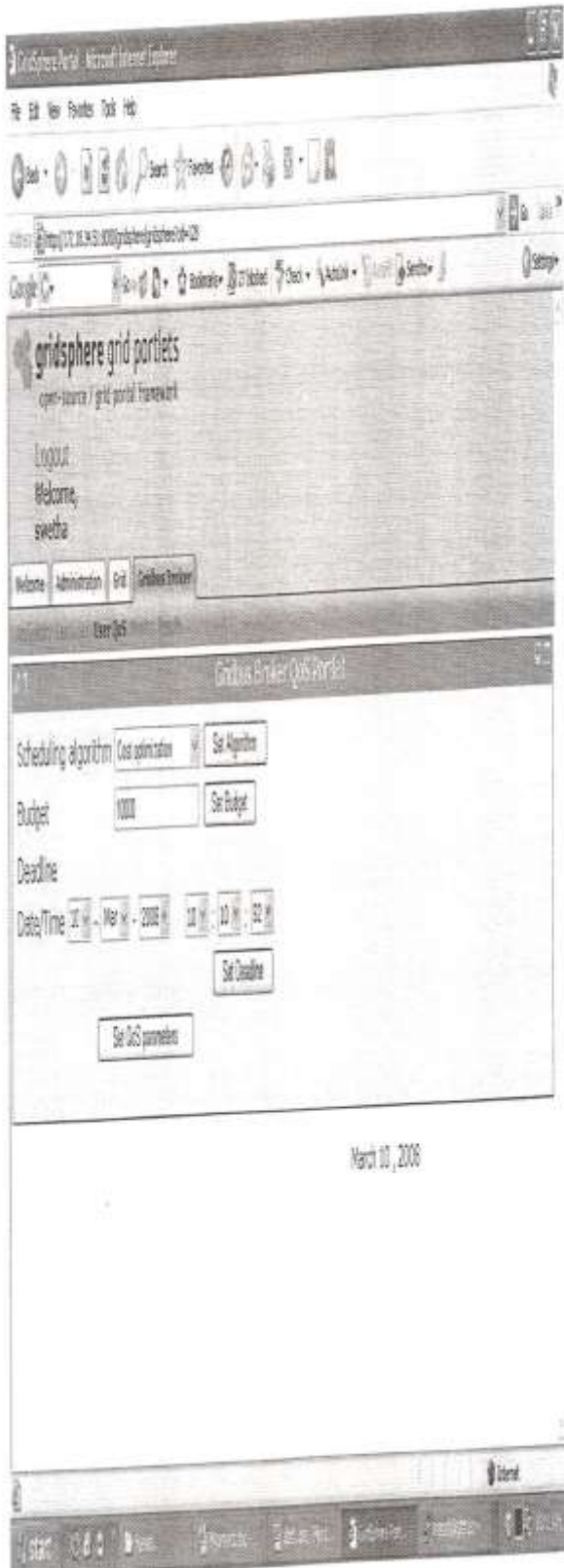


Figure 14. User QOS Portlet

## 5.7 Deadline

This is used to set the deadline for finishing the job. The deadline is set in the form of Date and Time.

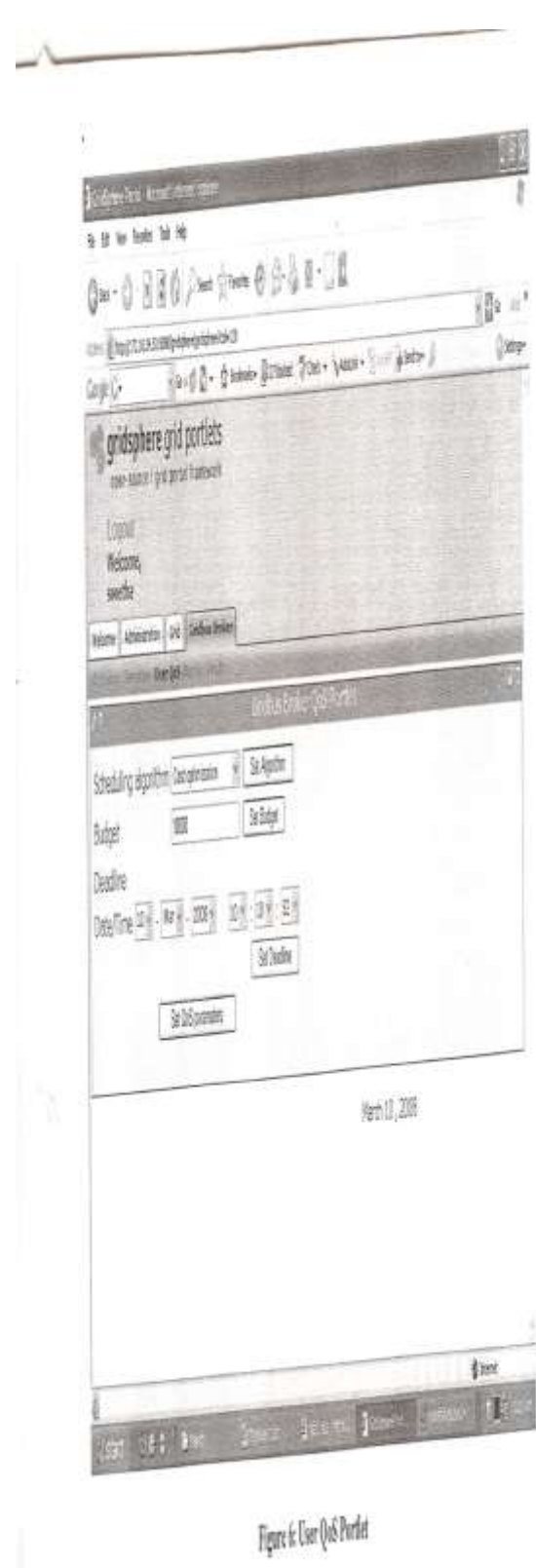


Figure 15. Deadline

## 6. CONCLUSION

This framework model attempts to explain the design of the GridSphere Portal, GridBus Broker and Globus.

## 7. ACKNOWLEDGMENT

This work was supported as Research Project implementation at “Osmania University”, Hyderabad, India.

## 8. REFERENCES

- [1] The Gridbus Grid Service Broker and Scheduler (2.0) User Guide <http://www.cloudbus.org/reports/gridbus-broker-guide2.pdf>
- [2] <http://phditsiamuresearch.pbworks.com/f/017.pdf>
- [3] <http://www.rimtengg.com/coit2007/proceedings/pdfs/69.pdf>
- [4] <http://gridcomputers.blogspot.com/>
- [5] High-Performance Computing On-Demand | The Scientist Magazine® <http://www.the-scientist.com/?articles.view/articleNo/15271/title/High-Performance-Computing-On-Demand/>
- [6] International Referred Research Journal ISSN- 0975-3486 VOL.I <http://www.ssmrae.com/admin/images/d378a351843796b8be2df2e4f0ef5f46.pdf>
- [7] An Integration of Global and Enterprise Grid Computing: Gridbus <http://www.cloudbus.org/reports/gridbus-xgrid.pdf>
- [8] An Analysis of MIPS Group Based Job Scheduling Algorithm with ... <http://ijcsi.org/papers/IJCSI-8-6-3-335-340.pdf>
- [9] Cluster and Grid Computing [http://www.dcc.fc.up.pt/~ines/aulas/0910/MAPi/scheduling\\_gc.ppt](http://www.dcc.fc.up.pt/~ines/aulas/0910/MAPi/scheduling_gc.ppt)
- [10] J. Tourino, M. Martin, J. Tarrío, and M. Arenaz, « A grid portal for an undergraduate parallel programming course,» IEEE Trans. Educ., vol.48, no. 3, pp.391-399, Aug.2005.
- [11] A. Andronico, R. Barbera, A. Falzone, P. Kunszt, G. L. Re, A. Pulvirenti, and A. Rodolico, « Genius : A simple and easy way to access computational and data grids, » *Future Gener. Comput. Syst.*, vol. 19,no. 6,pp.805-813, 2003.
- [12] R. Berlich, M. Kunze, and K. Schwarz, R. Buyya, P. Coddington, and A. Wendelborn, Eds., « Grid Computing in Europe : From research to deployment, » in *Proc. Australasian Workshop Grid Computing e-Research (AusGrid2005)*, Newcastle, Australia, 2005, vol. 44, pp. 21-27, ACS.