Resource Selection in Web-Based Grid

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Thapar University, Patiala

By:
VINEET BEHRA
(80631026)

Under the supervision of:
Mrs. Damandeep Kaur

MAY 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Resource Selection in Web-Based Grids"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mrs. Damandeep Kaur. and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Vineet Behra)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mrs. Damandeep Kaur)
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(SEEMA BAWA)
Professor & Head
Computer Science & Engineering. Department
Thapar University
Patiala

(R.K.SHARMA)
Dean(Academic Affaris)
Thapar University,
Patiala.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my guide **Mrs. Damandeep Kaur**, Lecturer, Computer Science and Engineering Department for immense help, guidance, stimulating suggestions and encouragement all the time with this thesis work. This work would have not been possible without her encouragement. She always provided a motivating and enthusiastic atmosphere to work with; it was a great pleasure to do this thesis under her supervision. The successful completion of this thesis is a direct consequence of the moral and material support extended by **Centre of Excellence in Grid Computing**, TU throughout this thesis.

I am equally grateful to **Dr. Seema Bawa** , Professor and Head, Computer Science and Engineering Department, **Dr. Maninder Singh**, Professor, Computer Science and Engineering Department, **Ms. Inderveer Chana**, Lecture, Computer Science and Engineering Department. My greatest thanks are to my parents who bestowed ability and strength in me to complete this work. I am deeply indebted to my parents and friends for their inspiration and ever encouraging moral support, which enabled me to pursue my studies.

I would also like to thank all the staff members and PhD Scholars Ms Anju Sharma and Ms. Shashi Bhanwar who were always there at the need of the hour and provided with all the help and facilities, which I required for the completion of my thesis.

I am also very thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct–indirect help, cooperation, love and affection which made my stay at Thapar University memorable.

VINEET BEHRA
(80631026)

Abstract

Grid is comprised of a large number of heterogeneous resources like computer, printer situated at geographically different places. For this reason, selecting suitable best resources becomes a crucial activity. To provide the user a quality of service by meeting the user's demand, we need an efficient resource selection algorithm. This thesis we describes the resource-ranking algorithm for resource selection in web based grids using page rank technique. This algorithm describes how to select Best Resources from a given list of resources after resource discovery.

Grid Portal provides an easy way to access Grid Services hosted on a Grid. It provides user-friendly interface to users, so user can easily use the Grid Services. In this thesis, we describe our experience in building a web based Grid Portal, named Global Grid Portal. We describe the Resource selection & then we describe in detail through using ranking algorithm that show how to select Best Resources through ranking algorithms.

Table of Content

List of Figures

Figure No. Title Page No.

List of Snap-Shot

Snap-shot No. Title Page No.

Organization of Thesis

The **first chapter** briefly introduces Grid Computing concepts. It describes various types of Grid, and Architecture of Grid. It also describes introduction about different types of Middleware and comparison among them.

The **second chapter** of the thesis is devoted to the literature survey of Grid Portal, Resource Discovery and Resource Selection.

In the **third chapter** the problem has been formulated and it describe in detail about proposed approach to solve this problem.

The **fourth chapter** describes implementation detail and provides the experimental result of resource selection approach.

Finally, thesis has been concluded in the **fifth chapter** along with mentioning the scope for future.

CHAPTER 1: Introduction

The term 'Grid" refers to heterogeneous Environment where resources are in distributed manner and it provides on demand access to those resources. It benefits organizations by permitting them to offer unused resources on existing hardware and software. The term "Grid" was coined in the mid1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. The concept of computational Grid has been inspired by the 'electric power Grid', in which a user could obtain electric power from any power station present on the electric Grid irrespective of its location, in an easy and reliable manner. When additional electricity is required, just plug into a power Grid to access additional electricity on demand, similarly for computational resources plug into a Computational Grid to access additional computing power on demand using most economical resource [1].

1.1 Grid Computing:

The term "Grid" was coined in the mid1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. The concept of computational Grid has been inspired by the 'electric power Grid', in which a user could obtain electric power from any power station present on the electric Grid irrespective of its location, in an easy and reliable manner. When we require additional electricity we have to just plug into a power Grid to access additional electricity on demand, similarly for computational resources plug into a Computational Grid to access additional computing power on demand using most economical resource [2].

Rajkumar Buyya defined the Grid as:

"Grid is type of parallel and distributed system that enables the sharing, selection and aggregation of geographically distributed resources dynamically at run time depending on their availability, capability, performance, cost, user quality of self service requirement[3]."

Another important facility provided by Grid Computing is its heterogeneity. It enables and simplifies collaboration among a wider audience. While distributed computing this collaboration is achieved it to some extent. In a Grid Computing takes these capabilities to a wider & open extent. Every heterogeneous resource in Grid makes the image of a large virtual computing system.

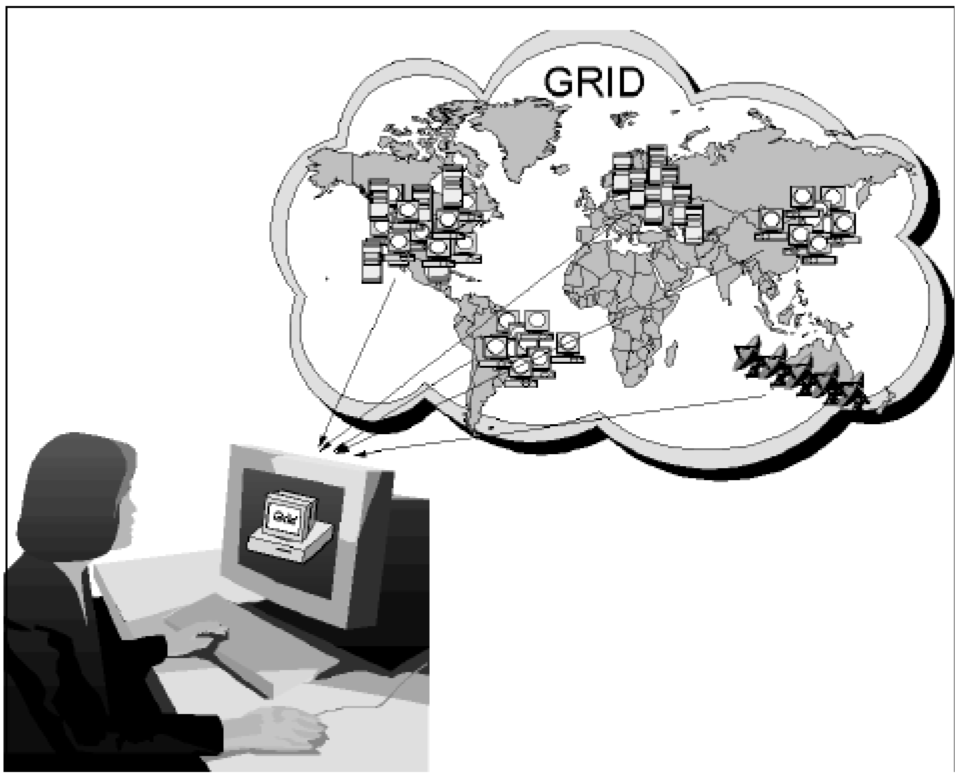


Figure 1-1 The grid virtualizes heterogeneous geographically disperse resources

1.1.1 Types of Grid

Grid computing can be used in a variety of ways to address various kinds of application requirements. Often, the type of solutions categorizes Grids that they best address. Ofcourse, there are no hard boundaries between these Grid types and often Grids may be a combination of two or more of these [4].

Grids can be classified on the basis of two factors:

- Scale
- Functionality

On basis of scale they can be further classified as:

- Global Grid
- Enterprise Grid
- Cluster Grid

Global Grids

It is collection of Enterprise and Cluster Grid as well as other geographically distributed resources. all the resources are agreed upon global usage policies and protocols to enable resources sharing.

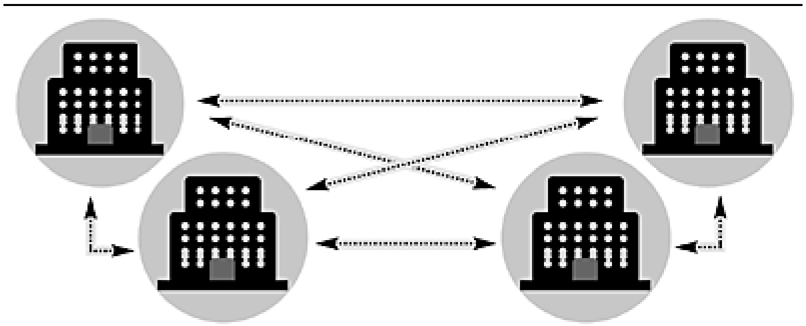


Fig1.3 Global Grids [4]

Cluster Grid

It is the simplest form of Grid. It provides a compute service to the group or department level.

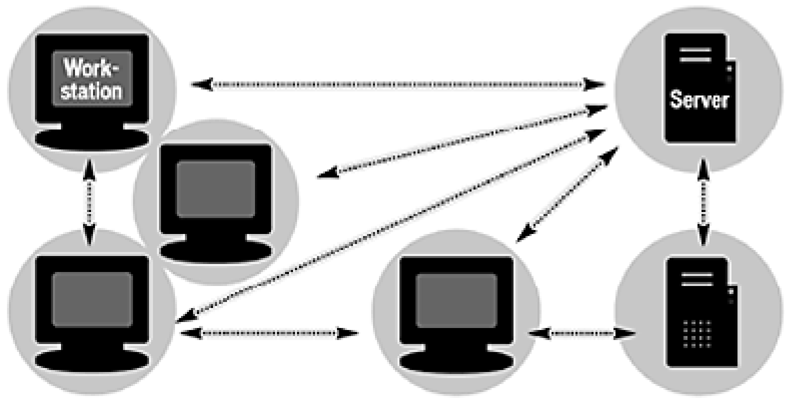


Fig 1.1 Cluster Grid [4]

Enterprise Grids

Enterprise Grids enable multiple project or department to share recourses with in enterprise or campus and not necessary have to address security and other global policy management issues associated with global Grid.

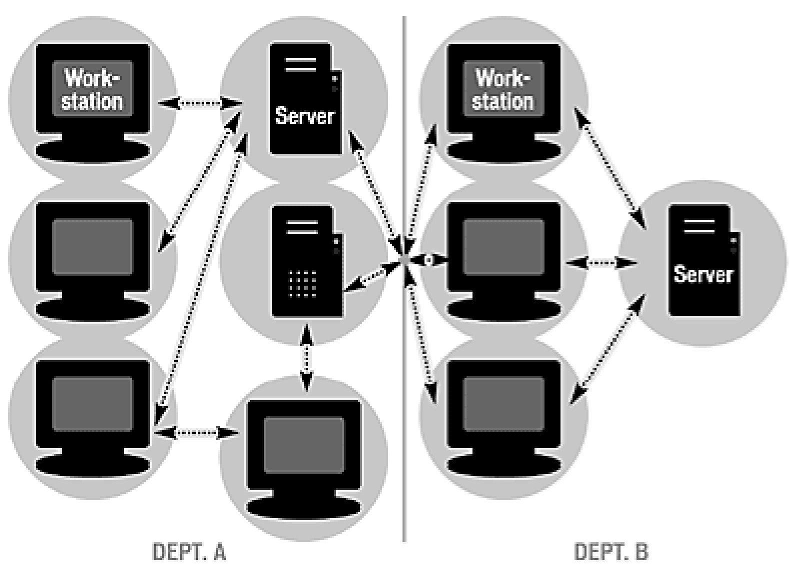


Fig1.2 Enterprise Grid [4]

On basis of Functionality

• Computational Grids

• Data Grids

• Communication Grid

Computational Grid

By analogy, we adopt the term computational grid for the infrastructure enable the in-creases in computation discussed above. A computational grid is a hardware and software infrastruc- capabilities.

By analogy, we adopt the term computational grid for the infrastructure that will enable the increases in computation. A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities[5]. When we talk about computing cycles provided by the processors of the machine on the grid it's a computational resource and when we use it in Grid its called computational Grid. In computational Grid our main concern area is computational power provided by different processors. The processors can vary in speed, architecture, software platform, and other associate factors, such as memory, storage and connectivity. A computational grid combines the processing power from a distributed collection of systems. Garuda project, SETI@home grid, Tera Grid these are the well known grid for providing computational power.

Computational grids can be recognized by these primary characteristics[6]:

_ Made up of clusters of clusters`

_ Enables CPU scavenging to better utilize resources

_ Provides the computational power to process large-scale jobs

_ Satisfies the business requirement for instant access to resources on demand

The primary benefits of computational grids are a reduced Total Cost of Ownership (TCO) and shorter deployment life cycles. Besides the SETI@home grid, the World Community Grid™, the Distributed Terascale Facility (TeraGrid), and the UK and Netherlands grids are all different examples of deployed computational grids. The next generation of computational grid computing will shift focus towards solving real-time computational problems[6].

Data Grid

Data Storage is the second most common resource used in any Grid. A grid providing an integrated view of data storage is sometimes called a “data grid.” Each machine on the grid usually 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.

Communication Grid

The rapid growth in communication capacity among machines today makes grid computing practical, compared to the limited bandwidth available when distributed computing was first emerging. Therefore, it should not be a surprise that another important resource of a grid is data communication capacity. This includes communications within the grid and external to the grid. Communications within the grid are important for sending jobs and their required data to points within the grid. Some jobs require a large amount of data to be processed, and it may not always reside on the machine running the job. The bandwidth available for such communications can often be a critical resource that can limit utilization of the grid[6].

1.1.2 Characteristics of Grids

- **Heterogeneity**: A Grid involves a multiplicity of resources that are heterogeneous in nature and might span numerous administrative domains across wide geographical distances.
  - Resources are heterogeneous
  - Resources are administratively disparate
  - Resources are geographically disparate
  - Users do not have to worry about system details (e.g., location, operating system, accounts).
  - Resources are numerous.
  - Resources have different resource management policies.

- Resources are owned and managed by different, potentially mutually distrustful organizations and individuals that likely have different security policies and practices.

- **Scalability**: Generic definition of scalability can be given as ability of a solution to some problem (computer application or product) to continue to function well as the problem (or its context) increases in size or volume. In term of Grid it might grow from few resources to millions. This raises the problem of potential performance degradation as a Grids size increases. Consequently, applications that require a large number of geographically located resources must be designed to be extremely latency tolerant.
- **Dynamicity or Adaptability**: In a Grid, a resource failure is the rule, not the exception. In fact, with so many resources in a Grid, the probability of some resource failing is naturally high. The resource managers or applications must tailor their behavior dynamically so as to extract the maximum performance from the available resources and services.

1.1.3 Grid Architecture

Grid Architecture is often describes in terms of “layers”. Each layer provides some kind of specific function.

.

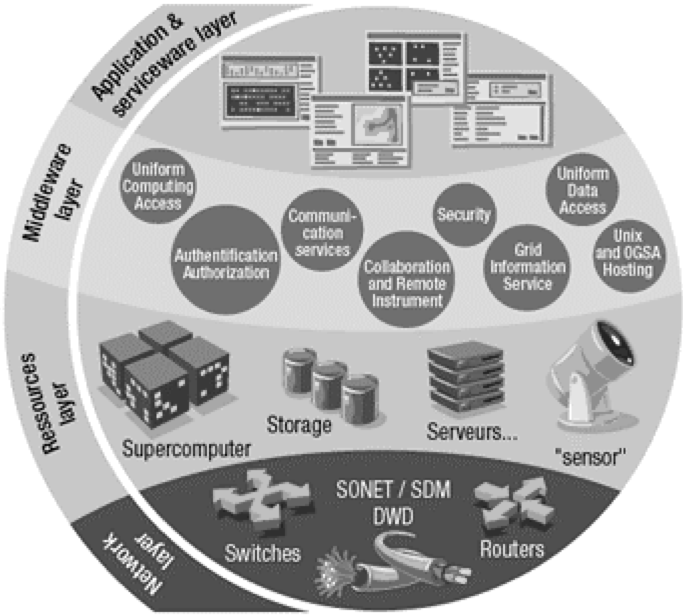
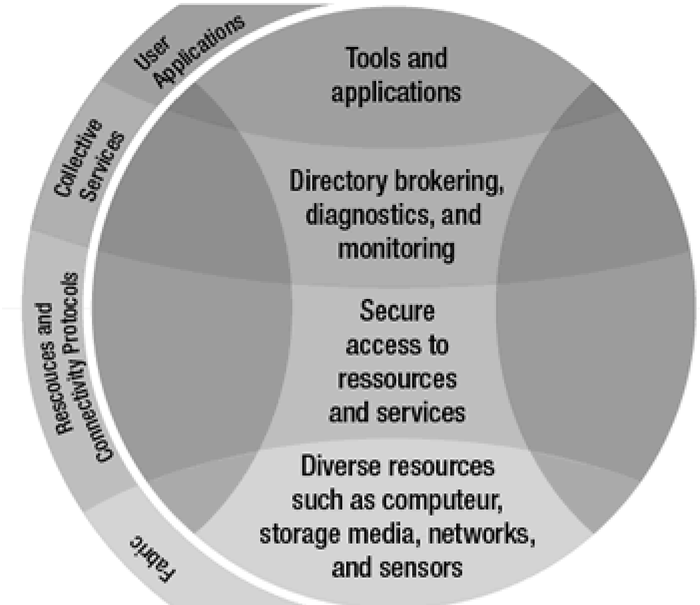


Fig x grid layered Architecture [7]

In the layered architecture the lowest layer is the network layer, which connects Grid resources. In network layer we have all networking devices like switches, routers etc. it's a hardware centric. Above the network layer lies the resource layer. In resource layer we have actual grid resources like computer, storage system, sensors and many more and all they are connected to the network. Above this layer is middleware layer or it's a " brain of the Grid! " its provides the tools that enable the various resources to participate in the Grid. It makes the Grid or it combined the all element together. After this the highest layer of the structure is application layer. It is user centric. User interacts with Grid through application layer. It handle the general management functions such as measuring the amount a particular user employs the Grid, all billing activities, and it keeps the records of all resource ,who is using resources and by who is providing.



Grid layered architecture [7]

There are other ways to describe this layered structure. For example, experts like to use the term **"fabric"** for the physical infrastructure of the Grid: things like computers and networks. It includes all the physical devices like computers, storage media, networks and sensors. Above this layer there is resources and connectivity protocols layer. Resource and connectivity protocols layer handle all the network transactions between different computers and Grid resources. The property of grid is it's heterogeneity that means all the resources are in distributed manner. We use Internet for connecting them, On this network all the resources contributing to the Grid must recognize Grid-relevant messages and ignore the rest. It's all possible because of communication protocols, it's allow the resources to communicate with each other, enabling exchange of data, and authentication protocols provide secure mechanisms for verifying the identity of both users and resources. Above this layer we have collective services layer. It also based on protocols. There are many protocols like information protocols, which obtain information about the structure and the state of the resources on the Grid, and management protocols which deals access to resources in a uniform way. Collective services include[7]:

- updating directories of available resources,

- brokering resources (which like stockbroking, is about negotiating between those who want to "buy" resources and those who want to "sell")
- monitoring and diagnosing problems
- replicating data so that multiple copies are available at different locations for ease of use
- providing membership/policy services for tracking who is allowed to do what and when.

The upper most layer is application layer. Consider a user application that needs to analyze data contained in several independent files. It will have to[7]:

- **obtain the necessary authentication credentials** to open the files (resource and connectivity protocols)
- **query an information system and replica catalogue** to determine where the files are and which computational resources can do the data analysis (collective services)
- **submit requests to the fabric** - the appropriate computers, storage systems, and networks - to extract the data, initiate computations, and provide the results (resource and connectivity protocols)
- **monitor progress** of the various computations and data transfers, notifying the user when analysis is complete, and detecting and responding to failures (collective services).

1.1.4 Advantages of Grid Computing

Some of the advantages of Grid Computing are listed below [8][9]:

• Seamless and secure access to large number of geographically distributed resources.

• Reduction in average job response time may occur but an overhead of limited network bandwidth and latency exists.

• Provides users around the world with dynamic and adaptive access to unparalleled levels of computing.

• With the infrastructure provided by the Grid, scientists are able to perform complex tasks, integrate their work and collaborate remotely.

• Grids can lead to savings in processing time.

• Efficient, effective, and economic utilization of available resources.

• Increased availability and reliability of resources.

• Shared access (by multiple users) to large amounts of data.

• Improved methods for collaborative work.

• Unprecedented Price-to-Performance ratio.

1.2 Grid Portal

Grid Portals are the Gateway to Grid Application. The Term “Grid Portal” inspired from the term “Web Portal” like yahoo, amazon, rediff. Grid Portal provides the interface between the grid resource and grid users. Campus Grid Portal is a web based portal. Web based accessing is the most common way to explore the information published in the network. So we believe that for the grid user, a web-based interface is a more appropriate system to enable grid participation. With a web-based grid portal the user can access and execute the grid resources anytime and anywhere through standard web browsers. Still, it requires no software downloads or configuration changes on the client side [10].

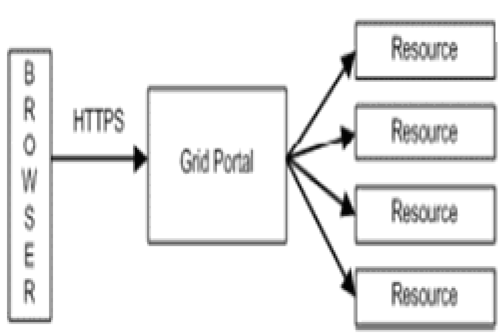


Fig 1. General Portal design

1.2.1 Types of gird portal

1. User Portal
2. Application Portal
3. Horizontal vs Vertical Portal
4. Information Portal

1. User Portals :

User portal provides general grid services like single sign-on, job submission etc. in this portal user can customize his own desires.

2. Application Portal

It is for scientific user, they interact with it and wish to harness the computational power on the grid to execute complex application tasks in a problem solving environment(PSE).

3. Horizontal vs Vertical Portals :

Horizontal Portals:

In this type of portal information provided is wide but not deep. Example yahoo portal.

Vertical Portal :

In this portal, it provides detailed information on a specific topic. Example : http://www.premierleaguecricket.in provides very specific information about the IPL T20 cricket match.

4. Information Portals:

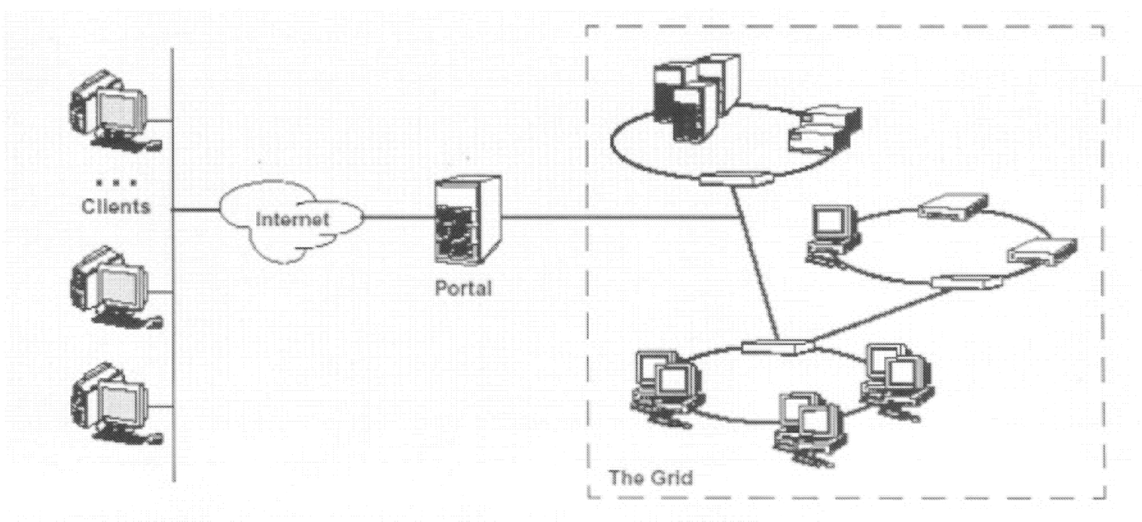
Portals which Provides the user with information or links to information. Example yahoo.com

1.2.2 Why Web Services:

Using web services we can publish our application to rest of the world. Web services use XML to code & decode our data & it use SOAP to transport it between different locations on the Web. The main benefit using web services is we can exchange our data between different application and different platform. A portal is a web based application that commonly provides single sign on facility to its user. If u considers various activities that are part of Grid Portal, they are Security Services (single sign on), Remote File Management, Access to information services, Application interface.

1.2.3 Portal Computing and Grid

Portal is the gateway to Grid. Today, portals can be used to access a single server, a cluster of resources, or a even a complex grid, creating opportunities for more organization and their users.



Portal and Grid [11]

Indeed, employing portal technology in HPTC environments offers the ability for users to work more effectively by allowing complex jobs to run with greater efficiency and at lower cost[11]. Multiple application can run across cluster of servers that are combined into a unified Grid Computing system, giving users easy access to complex applications and sophisticated computing resources. Today the Grid Portal plays an important role on

Grid Environment. Number of people do lots of work on the development of user-friendly interface for the Grid.

There are difference portals supported grid, for example WebSphere Portal, WebLogic Portal, Sun One Portal, Some open source software developers issue Grid portals to use freely for end-users as well. Among these portals, GridSphere is popular with supporting of Grid Portlets which can access distributed remote resources via Globus Toolkit middlewareb[12]. The UCLA[13] Grid Portal provides a single web interface to those computational clusters that have joined the UCLA Grid. Additionally, the UCLA Grid Portal can directly access some clusters outside of the UCLA Grid, including clusters on the TeraGrid. The UCLA Grid Portal Provide all Grid Services like Resource Discovery, job Submittal, file transfer, File management and editing, visualizationb[13].

In Grid Environment User interact with Portals. Portal is the interface between the user and Grid Services. Once user has login to the portal he can access the other services hostel on Grid. Example: Resource Discovery, Resources Selection, Job management, job submission etc. In resource management mainly we have two task resource discoveries and another one is resource selection. In Resource discovery, the user gives a detail description about the desired services that should be satisfied by the candidate resource. The resource discovery mechanism then returns a set of the resources for the given description. Resource discovery is the process by which a node can become aware of the attributes and capabilities of other nodes in the network. There are several projects which have addressed the resource discovery problem in the Grid. The Monitoring and Discovery System (MDS3) is the Information Services component framework of the Globus Toolkit Version 3.0 (GT3) [14]. It provides information about Grid resources for use in resource discovery, selection, and optimization. Condor-G [15] uses a centralized mechanism for maintaining the resource information; it uses a collector that listens for advertisements of resource availability [16]. The most interesting aspect of the resource management area is the selection of the correct resource from the Grid resource pool, based on the service-level requirements. Resource selection involves a set of factors, such as application minimal requirements, application run time, and resource access policies. In addition, resource selection must consider uncertainties associated with each resource and answer questions related to resource reliability, prediction error probability, and cost error probability to access a resource [17]. Decision theory and multi agent system, MASK (multi agent system broker) [18] is one of the one resource selection method. The other is policy-Based resource selection [19].

2.1 Resource Selection

2.1.1 Distributed Resource Selection in Grid Using Decision Theory:

Resource selection in grid must consider several factors and meet user objectives. In decision theory, it combines the probability and utility theories. Past evidence is used to calculate resource reliability. Utilities are associated with user preference. Specific models calculate job execution time predictions and verify resource access restrictions. The decision model groups this information and selects the best machine to run a job. The prediction module investigated here, PredCase, can give the user guarantees about when his job will finish and allows advance job scheduling. PredCase presents a small error prediction percentage and improves case retrieval performance. The decentralized restriction certification model, POLAR, can verify the resource workload, SLA parameters, and user access, achieving better performance compared with some other approaches. Experimental results demonstrate that the selection maximizes user utility. Different machines can be chosen if information about user preference or past evidence is changed. The middleware MASK is completely distributed and all model executions are parallel. These characteristics offer significant advantages to grid. They allow solving scalability problems in real time[20].

2.1.2 An Overview of Condor ClassAds and Matchmaking

A ClassAd (Classified Advertisement) [21] is a mapping from *attribute names* to *expressions.* Attribute expressions can be simple constants or a function of other attributes. A protocol is defined for *evaluating* an attribute expression of one ClassAd with respect to another ClassAd. For example, the expression "other.size > 3" in one ClassAd evaluates to **true** if the other ClassAd has an attribute named "size" and the value of that attribute is an integer greater than three. ClassAds can be used to describe arbitrary entities. In the current context, they are used to describe resources and user requests[23].

Conventional Condor matchmaking [22] takes two ClassAds and evaluates one with respect to the other. Two ClassAds *match* if each ClassAd has an attribute named "requirements" that evaluates to **true** in the context of the other ClassAd. A ClassAd can also include an attribute named "rank" that evaluates to a numeric value representing the quality of the match. When matchmaking is used for resource selection, the matchmaker evaluates a ClassAd request with every available resource ClassAd and then selects a resource that both matches the request and returns the highest rank[23]

2.1.3 Set matching Algorithm

The set-matching algorithm evaluates a set-extended ClassAd request against a set of resource ClassAds and returns a resource set that has highest rank. It comprises two phases. In the *filtering* phase, individual resources are removed from consideration based on individual expressions in the request. For example, individual expressions "other.os==redhat6.1 && other.memory>=100M" would remove any machine with an OS other than Linux Redhat v6.1, and/or with less than 100 Mb of memory. A Postfix expression can also be used in this phase, as discussed above. A set-matching implementation can index ClassAds to accelerate such filtering operations. In the *set construction* phase, the algorithm seeks to identify a resource set that best meets application requirements. As the number of possible resource sets is large (exponential in the number of resources available), it is not typically feasible to evaluate all possible combinations. Instead, we use the following greedy heuristic algorithm to construct a resource set from the resources remaining after Phase 1 filtering[23].

```
. CandidateSet = NULL;
BestSet=NULL;
LastRank = -1; Rank = 0;
while (ResourceSet != NULL)
{
Next = {X : X in ResourceSet && for all Y in ResourceSet,
rank(X+CandidateSet) > rank(Y+CandidateSet); }
```

```
ResourceSet = ResourceSet - Next;
CandidateSet = CandidateSet + Next;
Rank = rank(CandidateSet);
If (requirements(CandidateSet)==true && Rank > LastRank)
BestSet=CandidateSet;
LastRank=Rank;
}
if BestSet ==NULL return failure
else return BestSet
```

In narrative form, the algorithm repeatedly removes the “best” resource remaining in the resource pool (with “best” being determined by the rank of the resulting resource set formed) and adds it to the “candidate set.” If this “candidate set” has higher rank than the “best set” so far, the “candidate set” become the new “best set”. This process stops when the set of resources in the resource pool is exhausted. The algorithm returns the “best set” that satisfies the user’s request, or failure if no such resource set is found. As with other greedy algorithms, this set-matching algorithm is not guaranteed to find the best solution if one exists. The set-matching problem can be modeled as an optimization problem under some constraints. It is known that this problem is NP-hard under some situations. Hence it is difficult to find a general algorithm to solve this problem efficiently, especially when the number of resources is large. Our work provides an efficient algorithm with complexity O(N2) with rank computation as the basic operation[23].

2.1.4 Constraint-satisfaction-based matching.

Chuang Liu *et. al* recently proposed the Redline matching system: an alternative approach for doing resource selection in the Grid [24]. In this framework, the matching problem is first transformed into a constraint satisfac6 tion problem2, the set of constraints are then checked to make sure that no conflicts occur, and finally existing constraint solving technologies [25] (such as integer programming) are used to solve the transformed problem. Similar to Condor matchmaker, the Redline matching system is based on symmetric description of resource and request (*i.e.,* the same description syntax is used to describe both resources and requests). However, comparing to ClassAd, the Redline language is more expressive. It supports both gang-matching and set-matching capabilities. A common requirement among these systems is the symmetric syntactic description of resources and requests properties. As illustrated in the previous example in Section 1, it is difficult to introduce new concepts or characteristics into the system. Moreover, in the Grid environment, where resources and users span multiple organizations, it may be difficult to guarantee that resources and requests will use the same attribute names, and that the semantics of the same attributes are interpreted the same way by both resource providers and consumers[26].

2.2 Site Rank Apporach:

Site Rank [27], user-driven ranking of computational Grid resources. SiteRank enables Grid users to easily construct and adapt ranking functions that:
(i) Take as arguments performance metrics derived with the low-level benchmarks of GridBench [28]; the selection of these metrics can be done manually or semi-automatically by the end-user, through the user interface of GridBench.
(ii) Combine the selected metrics into a linear model that takes into account the particular requirements of the application that the user wishes to execute on the Grid (e.g., memory vs. floating-point performance bound). Using a ranking function, Grid users can derive rankings of Grid resources that are tailored to their specific application requirements [27].

2.3 Page Rank Approach:

The essential idea behind PageRank [29] is that if page A has a link to page B, then a is implicitly conferring some kind of importance to B. intuitively, a page has high rank if it has many back links or it has a few highly ranked back links. Like Google use page rank for web pages, we want to use same ranking algorithm for our resources selection. In PageRank, if a page links to another page this means that it is casting a vote as an indication that the other page is good. If many pages link to a page then that page has more votes and its worth should be higher [30].

2.3.1 What is PageRank?

PageRank is Google's method of measuring a page's "importance." When all other factors such as Title tag and keywords are taken into account, Google uses PageRank to adjust results so that sites that are deemed more "important" will move up in the results page of a user's search accordingly[30].

A basic overview of how Google ranks pages in their search engine results pages (SERPS) follows:

1) Find all pages matching the keywords of the search.
2) Rank accordingly using "on the page factors" such as keywords.
3) Calculate in the inbound anchor text.
4) Adjust the results by PageRank scores.

In reality, it’s slightly more complex and we’ll discuss this in more depth later, but for now the above description serves our purposes. It’s worth noting that PageRank is a multiplier and is not just simply added to the score. Thus, if your page had a PageRank of zero, it would rank at the very end of the SERPS[30].

2.3.2How is PageRank determined?

The Google theory goes that if Page A links to Page B, then Page A is saying that Page B is an important page. PageRank also factors in the importance of the links pointing to a page. If a page has important links pointing to it, then its links to other pages also become important. The actual text of the link is irrelevant when discussing PageRank[30].

2.3.3 How PageRank is calculated

On a simple level, we can tell quite a lot about how PageRank is calculated. This is because when Google was just a university research project, the creators of PageRank published a paper that detailed a formula for calculating it. This formula is now more than a handful of years old, and we suspect that it has changed somewhat since then. However, for detailing the over-riding principles of how PageRank works, it is as accurate today as the day it was written[30].

PR(A) = (1-d) + d (PR(T1) /C(T1) + ... + PR(Tn) /C(Tn))

Where PR(A) is the PageRank of Page A (the one we want to work out) .

D is a dampening factor. Nominally this is set to 0.85

PR(T1) is the PageRank of a site point ing to Page A

C(T1) is the number of links off that page

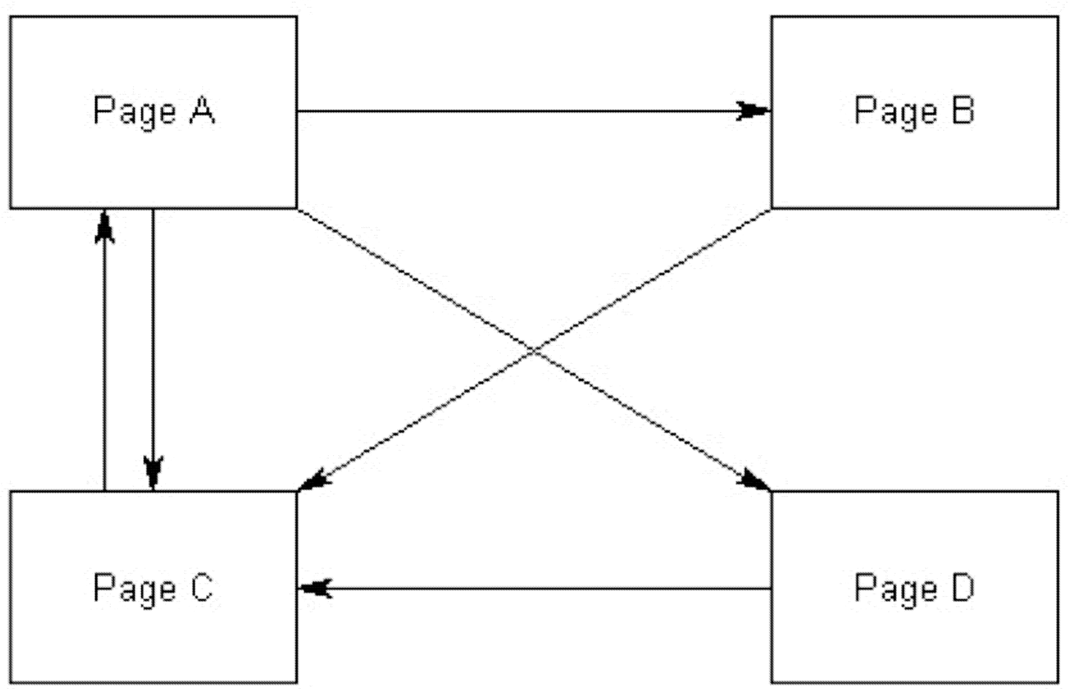
PR(Tn) /C(Tn) means we do that for each page point ing to Page A

Source: The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin and Lawrence

Page, http://www-db.stanford.edu/~backrub/google.html

This really is important. So to be totally clear, let's put some numbers to it (these numbers are made up for demonstration purposes only, and have no relevancy to any particular page). Say Page B has a PageRank value of 5 and has a single link on it pointing to Page A. Page A's PageRank is improved by a proportion of Page B's value of 5 (Page B doesn't lose anything, but Page A gains). If Page B has two links, that PageRank improvement would be split, and Page A would only gain half the PageRank that it did before.

Now put the formula out of your mind for a moment, as it's easier to understand how it works using a diagram. Let's say we have a hypothetical set of pages imaginatively titled Page A, Page B, Page C and Page D. They link to each other as shown below:



To begin with, in our example at least, we don’t know what the page’s starting PageRanks are. There’s nothing special about the number that we pick to start with (in fact, if you read forward to the section on convergence – you can see that we can start at any number we want). Since in the last version of this paper we performed the calculations by setting these values to 1 – we’re going to set them to zero this time around in order to prove that it doesn’t matter what the starting values are.

Next, we perform the necessary calculation to obtain the PageRank for each page. The rules are:

1. We take a 0.85 * a page’s PageRank, and divide it by the number of links on the page.
2. We add that amount on to a new total for each page it’s being passed to.
3. We add 0.15 to each of those totals.

The first calculation is easy. Because we’ve started at zero – 0 * 0.85 is always 0. So each page gets just 0.15 + 0. Meaning each page now has a PageRank of 0.15. Clearly we’re not done – we want to show the importance of each page based upon links, and they’re all the same; so we need to run the calculation again. Page A links to pages B, C and D. Page A’s PageRank is 0.15 so it will add 0.85 * 0.15 = 0.1275 to the new PageRank scores of the pages it links to. There are three of them so they each get 0.0425. Page B links to page C. Page B’s PageRank is 0.15 so it will add 0.85 * 0.15 = 0.1275 to the new

PageRank score of the pages it links to. Since it only links to page C, page C will get it all. Page C links to Page A, all 0.1275 passes to page A. Page D links to Page C. Again all 0.1275 passes to page C.

The new totals for each page them become:

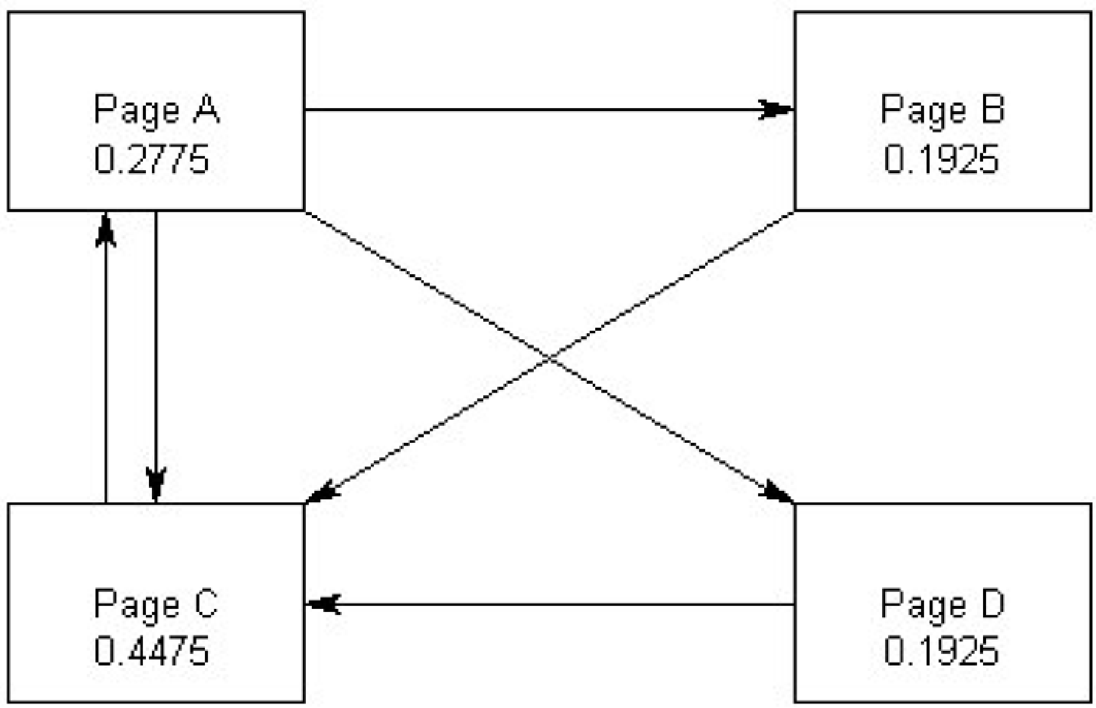
Page A: 0.15 (base) + 0.1275 (from Page C) = 0.2775

Page B: 0.15 (base) + 0.0425 (from Page A) = 0.1925

Page C: 0.15 (base) + 0.0425 (from Page A) + 0.1275 (from Page B) + 0.1275 (from Page D) = 0.4475

Page D: 0.15 (base) + 0.0425 (from Page A) = 0.1925

So we've got:



As suspected, Page C is the most important. If we take a quick look at these raw values we can see something about the number of links pointing out from a page. Look at Page A, which has a link from a high PageRank page (Page C), which has only one outbound link. Then look at Page B and D; both share links from a high PageRank page (Page A), with three outbound links. The number of links significantly alters the way PageRank is distributed[30].

In the proposed approach, resource selection is done through a portal using web services. It's a global grid portal. The term global means it will support heterogeneous resources as well as it consumes services from different middleware like Alchemi, Globus and Condor. Web services are used because it supports interoperability so Campus Grid Portal serves as an interface between user and different middleware. Resources Discovery relates to discover all the suitable resources available on Gird as per clients requirement. For resource discovery it consumed the WSDL document is consumed from different middleware which include description of various resources available on different IntraGrid. The major benefit of using web services is to make our portal to dynamic. In small interval it automatically refreshes the Consumed WSDL documents. Security is an important issue in developing Grid portal. It assures not only no leaking information of registered users and job running results, but also no attacking to the Grid system by network hack.

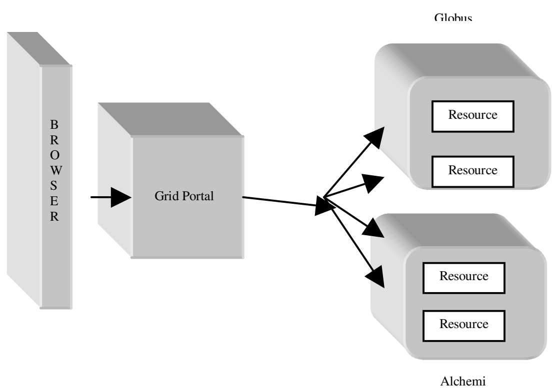
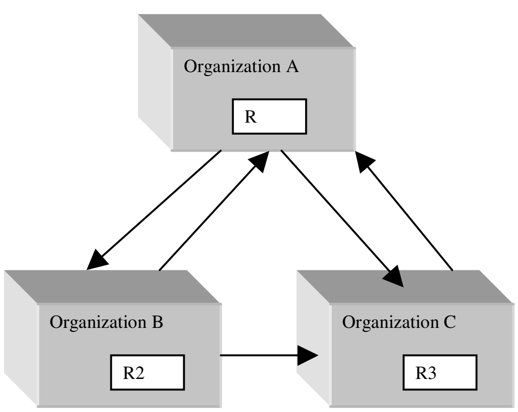


Fig 2: Proposed Portal Design

Using the portal, resource selection is performed using resource database obtained after resource discovery done as per user's requirement. Resource selection relates to choose suitable resources as per clients requirement. Grid technology allows resource sharing among several entities, but selecting the best resource to run a specific job remains one of its main problems. The challenges for the best resource selection involve analysis of several factors such as prediction time to run a job, and cost to use resources [31]. In the proposed work a page ranking algorithm is used to solve this problem. When users want to select any resources from resources pool first he selects his choice such as he want to run job in highly available resources with high execution speed. The other option is cost effective, to select an economical approach so the cost will low comparatively to highly available resource with high execution speed. The third one is Economical approach, where users get low cost and low resource available. And according to our choice we use ranking algorithm to select any resources. Resource in the World Wide Web (WWW) is the web pages which comprises of text- and hypertext-based entities that are independent an uncontrolled [32]. Resource can obtain higher ResourceRank score, if many users from different organizations submit jobs to that resource [33]. The ResourceRank is computed by adding up ResourceRank score of each resource provided by the organization.

The algorithm used to calculate the rank score. ResouceRank is actually a back link calculation [30] where in order to obtain the score of a resource, we need to get the ResourceRank score of users that currently vote or use that resource.



RR(A) = 0.15(base) + 0.1275/2(B) + 0.1275 (C) = 0.341225

RR(B) = 0.15(base) + 0.1275/2 (A) = 0.21375

RR(C) = 0.15(base) + 0.1275/2 (A) + 0.1275/2 (B) = 0.2775

Fig 4 Show the organization and their Resource Rank

Assuming that there are three virtual organizations; A, B, and C. with one user and one resource each. An arrows from organization A to B implies user from organization A has submitted a jobs to resource in organization B while an arrow from organization B to C implies user from organization B has submitted a job to resource in organization C. ResourceRank score of organization A is obtained by calculating the backlink of other user's ResourceRank; in this case is the user in the organization C who submits jobs to resource in the organization A. The base value is the initial value of each resource where there is no incoming vote or outgoing vote. User's ranking value depends on the ResourceRank of the organization. Organization A provides the most quality and reliable resources since there are many users' votes for resources in organization A. The final calculation indicates that the higher the ResourceRank score, the quality and the more reliable the resources provided by the organization [33].

CHAPTER 5: Implementation detail and Experimental results

In order to implement our proposed resource ranking approach, we developed a Web Based Portal named Global Grid Portal. We have implemented the application using .NET framework.

5.1 Case Study

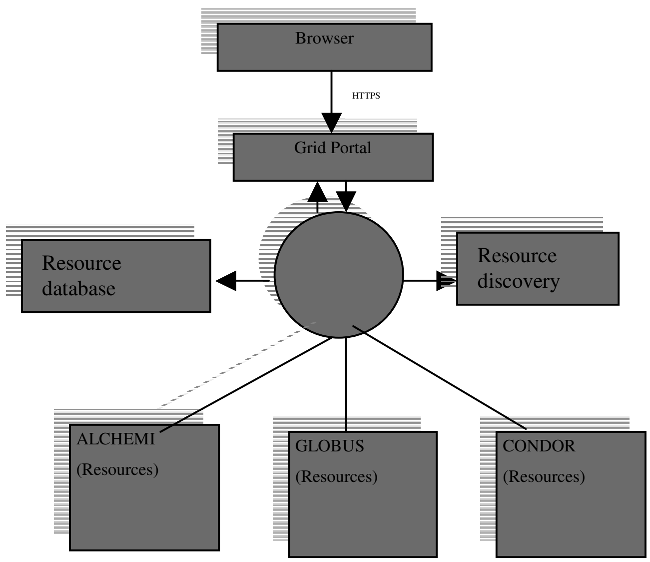


Fig 5: Global Grid Portal

Campus Grid Portal it's a Grid portal, where user use the portal through web browser. First of all if he authorized to use the services of portal he login the member area on portal then according to his choice about fast execution time, economical or cheaper cost, portal provides the list of all available resources. When users want to use the services provided by portal he first login through user name and password. After that he is able to use the other services of portal like Resource discovery, Resource Selection, job submission etc.

Name	Type	Architec-ture	Operat-ing System	Proces-sing Speed (GHz)	Proces-sing cost /sec	Used by Organiza-tion
R1	Mach-ine	Intel	Wind-ows	1.8	5000	A,B,C,D
R2	Mach-ine	AMD	Linux	2	4000	A,C
R3	Mach-ine	Intel	Linux	1.6	3000	B,C,D
R4	Mach-ine	AMD	MAC OS	1.2	1000	D
R5	Mach-ine	AMD	Solaris	0.8	500	C

Table 7: Resource Discovered by Portal

Above table show the discovered resources. After getting the all discovered resource our main concern area is resource selection in our portal by providing three facilities to users.

1. High performance

   Constraints: Processing Speed > = 1.6

2. Cost Effective Solution

   Constraints: Processing Speed >=1.2 and

   Processing Cost <=3000

3. Most Economical Solution

Constraints: Processing Cost <= 1000

According to users point of view he select his best suitable option. And according to his selection & our resources ranking algorithm users get available resources. If user request for:

Name	Type	Arch.	O.S.	Processing Speed & cost	Section criteria
User1	job	-	-	-	High Performance
User2	job	AMD	Linux	-	Most Economical Solution

Table 8: Resource Request by User1 and User2

Above Example if user select High Performance will get following resources and their resource rank according the Resource Rank algorithm.

Name	Architecture	Operating System	Processing Speed (GHz)	Processing cost /sec	Used by Organization	Resource Rank
R1	Intel	Windows	1.8	5000	A,B,C,D	0.3837
R2	AMD	Linux	2	4000	A,C	0.2562
R3	Intel	Linux	1.6	3000	B,C,D	0.32

Table 9: Resource Rank by ResourceRank algorithm[18]

These are the calculation to obtain the ResouceRank for each resource.

1. We take a 0.85 * a Resource's ResourceRank, and divide it by the number of links on the Reoources.
2. We add that amount on to a new total for each Resource it's being passed to.
3. We add 0.15 to each of those totals.

The first calculation is easy. Because we've started at zero – 0 * 0.85 is always 0.

So each Resource gets just 0.15 + 0. Meaning each Reasource now has a ResourceRank of 0.15. Clearly we're not done – we want to show the importance of each Resource based upon used by organization, and they're all the same; so we need to run the calculation again.

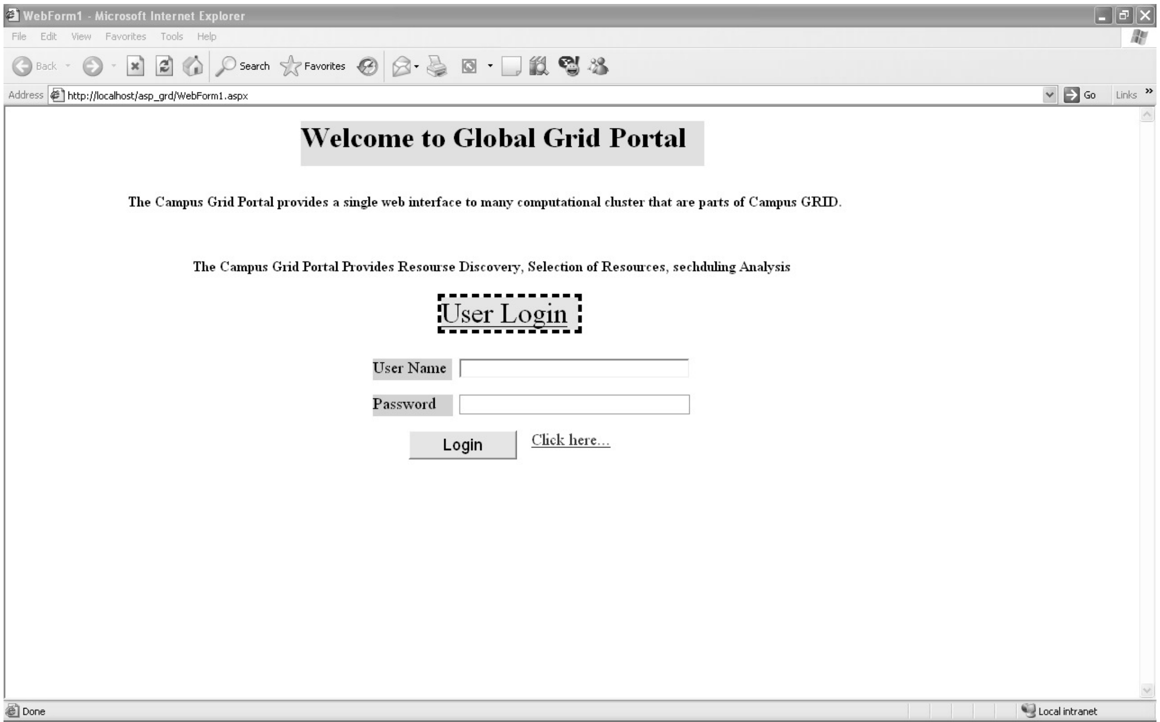
In other words, rank denoted the goodness of the Resource. We add ResourceRank into the Resource discovery table. Hence, ResourceRank becomes a new constraint that must be considered when selection of resources is performed. These are the following step for selecting any resource from all discovered Resources.

1. After resource discovery the matchmaking is strictly based on user's selection. And provide the all available resources.
2. Resource who has high ResourceRank will provide to its user.

Requestor interacts with Campus Grid Portal by placing their requirements or specifications on targeted resources as constraints. And get the best suitable Resources according to his/her Requirements.

5.2 Experimental Results:

The application starts with taking the user input. The first login page asks to user to submit his user name and password for authentication purpose.



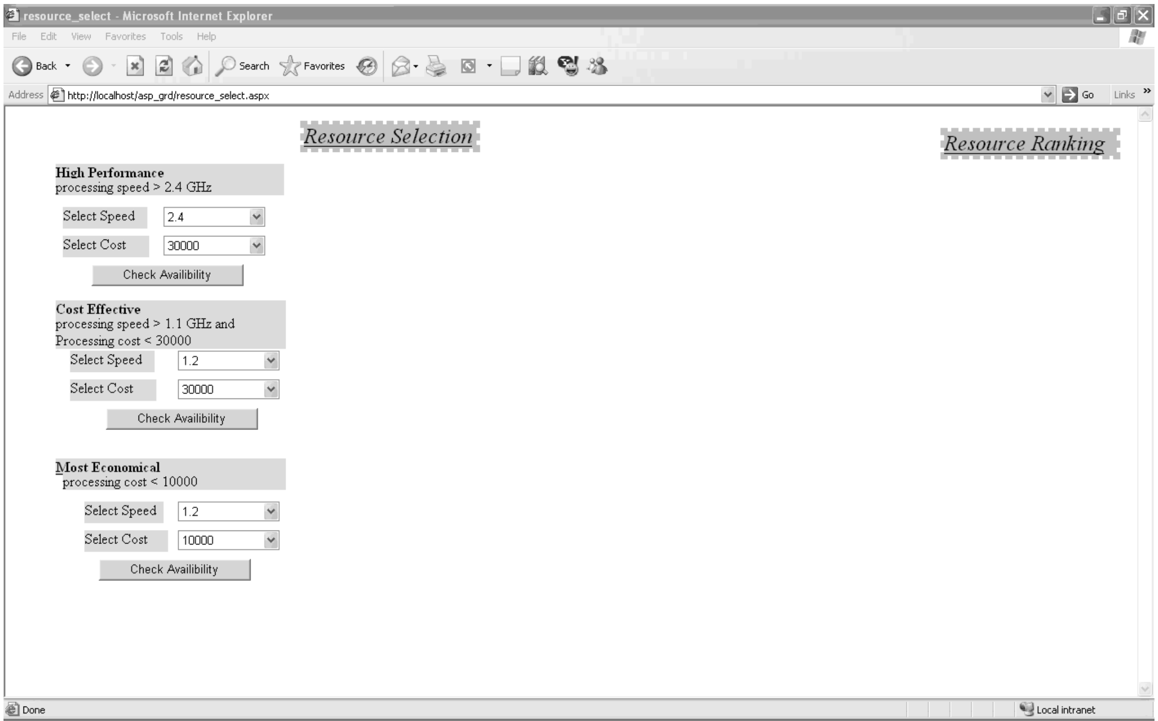
Snap-shot 5.1 : Login Page of Global Grid Portal

Above snap-shot show the login page of Global Grid Portal.

After the authentication process is done. User is ready to use other services hosted on grid. After login, users get the information about all the discovered resources attached with Grid. The below snap-shot show the all discovered resources.

Snap-shot 5.2: Resource Discovery in Global Grid Portal

Once the users get the all discovered resource now he click the resource selection and enter into the next page.



Snap-shot 5.3 Selection of resources

Above is the resource selection page where user can select his choice according to his application. According to user choice if user selects the option high performance he get the following resources with the resource rank of all those resources.

Snap-shot 5.4: resource and their resource rank according to high performance

Above snap-shot show the high performance resources. If users select high performance then he will get all the available resources according to his choice. And simultaneously he gets the resource rank of all the best suited resources.

If user selects the cost effective he will get all the result according to his/her choices. The below snap-shot show the all the best suited resources according to users choice.

Snap-shot 5.5: resource and resource rank according to cost-effective

CHAPTER 6: Conclusion and Future work

This chapter discusses the conclusions of the work presented by this thesis. The chapter ends with a discussion of the future direction this work will take.

6.1 Conclusion

This thesis provides the introduction to Grid Computing. We describe the Resource Selection on Grid. We presented a new approach for Resource Selection that is resource ranking. We describe the technique how to select best resources among all distributed resources according to user's requirement. We apply our approaches on the Grid Portal, name Global Grid Portal. Where we show the technique how to select best resources using resource ranking algorithm. We implement the resource rank algorithm to select best resources as per user's requirement.

6.2 Summery of contributions

- On this thesis we show the new technique to select the best resources after resource discovery.
- We use resource ranking approach to solve the problem of selection the best resources.
- We provide the facility to user to select the resource according to his/her choice.
- We apply our resource selection algorithm to the Campus Grid Portal.

6.3 Future Scope

The work performed in this thesis can be used as one of the major parameter while developing the grid portal. This is because it provides the user friendly environment and Choice to user to select his resources according to his choice.

A further extension to this work would be use to enhance the portal services. After resource selection we can use scheduling analysis. We can submit our job to various middleware.