Chapter I

Introduction
CHAPTER 1

INTRODUCTION

This chapter introduces the context of the research presented in this thesis. It first provides a high-level overview of advance reservation of the resources and scheduling of grid computing for e-governance applications. Then it briefly presents the inspiration for data grids, the objective, the scope of the research work and the primary contributions. This chapter ends with a discussion on the organization for the rest of this thesis.

1.1 Grid Computing and Architecture

The vision of grid computing was introduced by Ian Forster and Carl Kesselman with the publication of their book “The Grid: Blueprint for a New Computing Infrastructure” in July 1998 [50]. The idea is to virtualize computing, with the goal of creating a utility computing model over a distributed set of resources.

Within a single computer, standard elements including the processor, storage, operating system, and I/O exist. The concept of grid computing is to create a similar environment, over a distributed area, made of heterogeneous elements including servers, storage devices, and networks – a scalable, wide-area computing platform. The figure 1.1 helps to visualize the concepts.
The grid middleware is the software that handles the coordination of the participating elements. It is analogous to the operating system of a computer. A grid service is a special service that contributes to avail the grid infrastructure to users. It is analogous to an operating system component such as the file system or the memory manager.

The grid middleware is organized in a "grid service" layered stack for performing various operations. The grid services provide users with standard and uniform interfaces over all the sites participating to the grid. A grid resource is a component of the system that provides or hosts services and may enforce access to these services based on a set of rules and policies defined by entities that are authoritative for the particular resource.

Typical resources in grid environments might be a computer providing compute cycles or data storage through a set of services it offers. Access to
resources may be enforced by a resource itself or by some entity that is located between a resource and the requestor, thus protecting the resource from being accessed in an unauthorized fashion.

No matter what the hardware and software solutions are used to create a grid participating site, users can always use a transparent and standard interface to access the available resources. Also many different grid services can be used for performing the same task. Such grid services publish their interfaces that can be therefore used by other grid services as well as by higher-level user applications.

In this chapter overview of grid computing and its architecture and components are illustrated. In the academic as well as in the commercial world there are several definitions of grid computing [14]. However, they all focus on the need of virtualizing a set of distributed resources in order to provide a scalable, flexible pool of processing and data storage that can be used to improve efficiency. Another key of grid computing is the promotion of standards that allow interoperability of systems provided by many vendors and modular integration to create complex systems. Grid computing helps to create a sustainable competitive advantage by the way of streamlining product development and allowing focus to be placed on the core business.

The grid computing market is in a relatively early stage [121] [24], but now it is the time to initiate grid-related developments for several reasons, particularly the following ones:
The need for communication and efficient interconnection is becoming important in order to have predominance in the market.

Emerging applications are significant, coming from increasingly important markets, such as energy and oil, financial services, government, life sciences, and manufacturing.

The infrastructure to support these applications is currently underserved.

The potential market size is substantial (including hardware and software associated with grid deployments).

Investment commitment and development focus from the industry’s largest computing players, including HP, IBM, Microsoft, and Sun, it is an indicator that this is a growth market.

Increased deployment of blade servers coincides with the related view of blade server vendors that clusters and grids are ways of moving forward.

There is increasing pressure for enterprise IT organizations to cut costs and increase utilization of existing infrastructures.

Web services are distributed software components that provide information to applications, through an application-oriented interface. Grid environments enable the creation of virtual organizations and advanced web services.

Referring to grid computing as "The Grid", is not necessarily the most appropriate or accurate reference. Although it is convenient for introducing a high-level discussion, there is not one single "Grid". Instead, there are many grids, some
private, some public, some distributed worldwide and even some local to one room.

1.1.1 Grid Computing

In the following section some fundamental definitions in order to provide a common knowledge base for this thesis is given. There are many definitions for grid computing. The following three are quite comprehensive:

- The grid is an aggregation of geographically dispersed computing, storage, and network resources, coordinated to deliver improved performance, higher quality of service, better utilization, and easier access to data.

- The grid enables virtual, collaborative organizations, sharing applications, resources and data in an open, heterogeneous environment. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs.

- The grid promotes standard interfaces definitions for services that need to inter-operate in order to create a general distributed infrastructure to fulfill user's tasks and provide user level utilities.

The common denominator for all grid infrastructures is the network layer. Since it is the common network fabric that connects all of the resources in a given grid, its significance is amplified. Distributed grid systems demand high-speed connectivity and low latency.
A grid application can be defined as an application that operates in a grid environment.

From an application perspective, there are two types of grids: computing grids and data grids. The majority of the early grid deployments have focused on computation, but as data grids provide easier access to large, shared data sets, data grids are becoming more and more important.

A computing grid [50] is a collection of distributed computing resources, within or across sites, which are aggregated to act as a unified processing virtual supercomputer. These compute resources can be either within or across administrative domains. Collecting these resources into a unified pool involves coordinated usage policies, job scheduling and queuing characteristics, grid-wide security, and user authentication. The benefit is faster, more efficient processing of compute-intensive jobs, while utilizing existing resources.

Computing grids have typically higher latencies than clusters. However, they provide for more computing power due to the CPU aggregation. Compute grids also eliminate the drawback of tightly binding specific machines to specific jobs, by allowing the aggregated pool to most efficiently serve sequential or parallel jobs with specific user requirements.

A data grid [30] provides for wide-area, secure access to significant amount of data. Data grids enable the management and efficient usage of data stored in different formats as well as in distributed locations. Much like computing grids, data grids also rely on software for secure access and usage policies. However grid
storage solutions assume a quite important role in this context. Data grids can be deployed within one administrative domain or across multiple domains. In such cases grid software and policy management become critical. Data grids reduce the need to move, replicate, or centralize data, translating into cost savings. Initial data grids are being constructed today, primarily serving collaborative research communities [26] [130] [89] [92]. Software vendors and large enterprises are currently investigating data grid solutions and services for business applications [24].

The evolution from computing grids to data grids is an important factor in moving grid technology from education, research and development to large enterprise. This transition is an indicator that the market, in addition to the technology, is maturing. From a networking perspective, the impact of data grids will include a tighter integration of storage protocols and high-performance networking.

The first stage of grid computing is a cluster [32] and is shown in figure 1.2. Commonly the definition of a grid includes terms like "distributed" and "heterogeneous".

A cluster file system is shared among the nodes of a computer firm. Local Resource Management Systems (LRMSs) provide submission, scheduling, and dispatch services for local clusters. Clusters are often defined as collections of homogeneous servers aggregated for increased performance. Clusters are widely
used in the manufacturing domain for things like simulation-based testing and evaluation.

Figure 1.2 A computing clusters

The majority of new cluster servers has Gigabit Ethernet interfaces [33], and can range from a handful to literally thousands (in research environments) of servers [122]. As a result, high-density Gigabit Ethernet support is necessary. In addition, low-latency switching is also critical in maintaining application performance across the fabric of a cluster.

Clusters are critical in the evolution of grid computing. This is because clusters need to be interconnected in order to move to the next phase known as Intra-Grids (in analogy with Intranets). Interconnecting separate clusters enables the creation of enterprise and inter-departmental grids as depicted in figure 1.3.

The creation of Intra-Grids puts additional constrains on the controlling software layer, or middleware, and the underlying network layer. The middleware
must now have a better understanding of resource allocation because of additional complexity introduced by resources-sharing relationships. Intra-Grids will evolve in a very controlled fashion. For example, two or three clusters may be interconnected between departments within an enterprise to increase processing capacity and share data sets. A good example of this can be found at Magna Steyr in Austria, where clusters of design IBM workstations were interconnected via LSF [68] to increase the computing power dedicated to the batch simulation of the assembly process and the clashes detection [24]. Because the relationship between clusters is still within one enterprise domain, things like security and authentication, although important, are not as critical as in later phases.

Extra-Grids are essentially clusters and/or Intra-Grids that are connected between geographically distributed sites within or between enterprise organizations. The two important distinctions here include geographic distribution and inter-enterprise relationships.

Figure 1.3 Intra/Extra-Grids
Clusters are interconnected in a LAN to form Intra-Grids, or on the WAN to form Extra-Grids. Now that processing and/or data can be shared between two different organizations, authentication, policy management, and security become critical requirements that the middleware must address. Multi-site load balancing, topology discovery, and application awareness are also important to ensure performance.

![Figure 1.4 The evolution of the Grid](image)

The final stage in the evolution of grid computing is the Inter-Grid and is shown in figure 1.4. This is the most advanced stage because it embodies two of the primary visions for grid computing: utility computing infrastructure and grid services/service providers. It should be noted that Inter-Grids do exist today in the research and development world [31] [29]. Examples of Inter-Grids (both data and computing Grids) are the WLCG for the High Energy Physics research
community, Teragrid [123] in USA for e-Science, EGEE [25] in Europe. An Inter-
Grid hosts hundreds or even thousands of users.

1.1.2 Grid Computing vs. Distributed Computing

Grid computing poses many issues that are generally not present in
distributed computing. Here is a list of differences between Grid and distributed computing.

- Grid computing does not generally focus on one specific type of application
  but it is supposed to provide a computing infrastructure similar to the one
  offered by an operating system. Even though the Web is considered to be a
  predecessor of grid computing, it is a good example of distributed computing
  focusing on information handling.

- Resources are numerous and of many kind.

- Resources are owned and managed by different, potentially mutual distrustful
  organizations and individuals.

- Resources are potentially faulty and managed with different degree of fault
  tolerance. A grid infrastructure is therefore highly dynamic. Resources can
  appear and disappear while the infrastructure has to guarantee the given quality
  of service promised to users.

- There are different security requirements.

- Local management policies are different and need to be honored.
• Grid environments are strongly heterogeneous, e.g., they have different CPU architectures, run different operating systems, and have different amounts of memory and disk. Ideally, a running application should be able to migrate from platform to platform.

• Heterogeneous, multi-level networks connect resources. Firewall policies can be different and quite strict from site to site. Resources can also be connected using different technologies.

• Resources are geographically separated (on a campus, in an enterprise, on a continent).

• Sociological and cultural factors influence the success of a grid computing infrastructure.

• A global name space needs to be enforced in a grid for data access. More generally, resources need to be described globally with a unified vision.

• One of the most challenging issues in a grid is that grid services need to be able to scale to support a very high number of installations.

• Grid infrastructures need to be extensible in order to accommodate smoothly new technologies and new user needs.

• Persistency should be honored by the system. Applications need to read and write data. Such data can be spread around the world and need to be accessed efficiently. Furthermore, many protocols for data access exist and should be supported in order for instance to accommodate the needs of legacy applications.
Table 1.1 summarizes the differences between conventional distributed environments and Grids.

<table>
<thead>
<tr>
<th>Conventional Distributed Environments</th>
<th>Grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>A virtual pool of computational nodes</td>
<td>A virtual pool or resources</td>
</tr>
<tr>
<td>A user has access (credential) to all the nodes in the pool</td>
<td>A user has access to the pool but not to individual nodes.</td>
</tr>
<tr>
<td>Access to a node means access to all resources on the node.</td>
<td>Access to a resource may be restricted.</td>
</tr>
<tr>
<td>The user is aware of the capabilities and features of the nodes.</td>
<td>The user has little or no knowledge about each resource.</td>
</tr>
<tr>
<td>Nodes belong to a single trust domain</td>
<td>Resources span multiple trust domains.</td>
</tr>
<tr>
<td>Elements in the pool 10-100, more or less static.</td>
<td>Elements in the pool 10-100, dynamic.</td>
</tr>
</tbody>
</table>

Table 1.1 Differences between conventional distributed environments and Grids

Complexity management is another issue that spans different sectors: system installation, configuration, support and control, heterogeneity in policies for resource usage, monitoring in large scale, different failure modes and high number of components etc.

1.1.3 Grid Architecture

An architecture is a formal description of a system, defining its purpose, functions, externally visible properties, and interfaces. It also includes the
description of the system’s internal components and their relationships, along with
the principles governing its design, operation, and evolution [71] [54].

A service is a software component that can be accessed via a network to
provide functionality to a service requester. A general, clearly defined grid
architecture that follows the definition above does not exist. This is due to the fact
that it is difficult to try to force homogeneity on distributed groups of
collaborators. The goal of the CoreGrid project [29] is to provide such a general
description of grid architecture.

A Virtual Organization (VO) is a set of individuals and/or institutions
defined by specific applications, data and resources sharing rules [54].
The figure 1.5 provides a graphical view of the layers that define grid architecture and the function that each layer provides.

Application and service-aware software defines the highest layers of the
grid architecture. This includes portals as well as development toolkits. Applications vary from use case to use case in either academia or industry, depending on the problem, as do the portals and development toolkits supporting the applications.

Service-awareness provides many management-level functions including billing, accounting, and measurement of usage metrics, all very important topics to track as resources are virtualized for sharing among different users, departments, and companies.
The Middleware layer provides the protocols that enable multiple elements (servers, storage, networks, etc.) to participate in a unified grid environment. The middleware layer can be thought of as the intelligence that brings the various elements together through software and control.

The middleware enables virtualization that transforms computing resources into a ubiquitous grid. As shown in figure 1.6, the middleware is also organized in layers (similar to the 7 OSI layers in the Internet protocol stack), following the description given in [54], the Middleware Application layer provides the interface between the lower middleware layers and the user’s applications; the middleware resource and connectivity protocols facilitate the sharing of individual resources. Protocols at these layers are designed so that they can be implemented on top of a
diverse range of resource types (Middleware Fabric layer). Global services define the Collective layer, so called because it involves the coordinated ("collective") use of multiple resources.

The Resource (or Fabric) layer defines the grid system. It is made up of the actual resources that are part of the grid, including primary servers and storage devices. The Network layer is the underlying connectivity for the resources in the grid.

This thesis mainly focuses on the advance reservation of the resources and job scheduling of a data grid. This thesis aims to improve resource utilization and user satisfaction by considering novel job scheduling and reservation management strategies.

Figure 1.6  The layered Grid Middleware architecture and its relationship to the Internet protocol architecture. Figure taken from “The Anatomy of the Grid” [54]

This thesis mainly focuses on the advance reservation of the resources and job scheduling of a data grid. This thesis aims to improve resource utilization and user satisfaction by considering novel job scheduling and reservation management strategies.
1.2 Motivation

Most of the consuming amenities and needs are manual, complicated and time-consuming process. To overcome these drawbacks e-governance applications are established.

1.2.1 The Need for e-governance

World economies have recognized Information Technology (IT) as an effective tool in catalyzing the economic activity, in efficient governance and in developing human resource. They have, therefore, made significant investments in it and successfully integrated it with the development process, thereby reaping the benefits to their society. In India also these developments have impacted the industrial, education, service and government sectors and their influence on various applications is increasingly being felt of late. As the era of digital economy is evolving, the concept of governance has assumed significant importance. Some of the e-governance applications which will be benefited by the effect of information technology are illustrated in this thesis. As a scenario, the application of citizen applying for new water connection is chosen in this work. To implement these types of applications grid technology take an important role. In this technology advance reservation and scheduling are the two concepts chosen for this work. Thus with this grid technology implementing an e-governance solution will lower the cost of developing, deploying, managing government solutions and providing better services to the citizens.
1.2.2 The Need for Advance Reservation and Scheduling

One of the primary goals of grid computing [50] [32] is to share access to geographically distributed heterogeneous resources in a transparent manner. There will be many benefits when this goal is realized, including the ability to execute applications whose requirements exceed local resources and the reduction of job turnaround time through workload balancing across multiple computing facilities. The development of data grids and the associated middleware has therefore been actively pursued in recent years. However, many major technical hurdles stand in the way of realizing these benefits. Among the myriad research issues to be addressed is the problem of distributed resource management and job scheduling for data grids.

In most grid systems, submitted jobs are initially placed into a queue if there are no available resources in the local resource manager or scheduler. However, each grid system may organize a different scheduling algorithm that executes jobs based on different parameters, such as submission time, number of resources, and duration of execution. Therefore, there is no guarantee as to when these jobs will be executed. Several researchers have proposed the need for advance reservation [60] [126] [3] to address these issues. Common resources that can be reserved or requested are compute nodes, storage elements, network bandwidth or a combination of any of those.

In general, reservation of the above mentioned resources can be categorized into two: immediate and advance. However, the main difference between these
two reservations is the starting time. Immediate reservation acquires the resources to be utilized straight away, whereas advance reservation defers their usage later in the future.

Advance reservation can be useful for several applications such as parallel applications, where each task requires multiple compute nodes simultaneously for execution, workflow applications, where each job may depend on the execution of other jobs in the application. Hence, it needs to wait for all the dependencies to be satisfied before it can be executed and multimedia or soft real-time applications, such as video conferencing and player, where they need to have a certain amount of bandwidth to ensure a smooth broadcast of video and audio over the network.

The suitable approach to this problem begins with defining advance reservation of the resources with scheduling architecture that consists of autonomous local schedulers that schedules access to computer systems and grid schedulers, paired with local schedulers that send jobs to local schedulers and migrate jobs between grid schedulers. It is important that grid scheduling be distributed for scalability and fault tolerance and it is important that local schedulers have control of local resources so that grid scheduling will be accepted by the owners of the computer systems. Second, the algorithm for reserving the resources in advance and migrating jobs between grid schedulers are described. Third, the advance reservation and grid scheduling algorithms by simulating compute servers, networks, and schedulers are evaluated. Then several key
performance metrics are used during these simulations and used these metrics to compare the performance of the algorithm.

The experiments show that the hybrid-core algorithm has lower turnaround times than the other and that these times are less, even if no advance reservation and grid scheduling is performed.

1.3 Problem Specification

One of the primary goals of grid computing is to share access to geographically distributed heterogeneous resources in a transparent manner. There will be many benefits when this goal is realized, including the ability to execute applications whose requirements exceed local resources and the reduction of job turnaround time through workload balancing across multiple computing facilities.

Following are the issues involved in all e-governance projects

a) The number of users of the system is enormous.

b) As time progresses, the number of applications will increase. Hence the system has to provide facilities for handling large loads.

c) All e-governance applications must strictly adhere to specifications otherwise it is liable for legal prosecutions.

d) The hardware and software heterogeneity exists in all spheres of e-governance. Hence there is a need for extensive integration.

e) It requires common platform availability.
So the present research gives a scheduling approach along with advance reservations to overcome these problems. To improve the resource utilization it gives a scheduling solution within the same reservation block and backfilling with other independent jobs.

1.4 Objectives of this Research

The main objective of this research is to devise a new hybrid-core algorithm for resources over grid. The hybrid-core algorithm includes the advance reservation and the scheduling to implement the e-governance applications in the grid environment. While preserving competitive-scheduling performance more precisely, the proposed hybrid-core algorithm, the achieved time and cost could be comparable with the best possible time and cost of other algorithms.

1.5 Scope of the Research

The GRID is a technology for using enormous amounts of computing power and data storage, a possibility to share expensive computational resources.

Grids can be categorized with a three stage model of departmental grids, enterprise grids and global grids. These correspond to a firm initially utilising resources within a single group i.e. an engineering department connecting desktop machines, clusters and equipment. This progresses to enterprise grids where non-technical staff's computing resources can be used for cycle-stealing and storage. A global grid is a connection of enterprise and departmental grids that can be
commercial or collaborative. Grids offer a way to solve grand challenge problems such as protein folding, financial modeling, earthquake simulation, and climate/weather modeling. Grids offer a way of using the information technology resources optimally inside an organization. They also provide a means for offering information technology as a utility for commercial and noncommercial clients, with those clients paying only for what they use, as with electricity or water.

Grid computing is being applied by the National Science Foundation's National Technology Grid, NASA's Information Power grid [135], Pratt & Whitney, Bristol-Myers Squibb Co., and American Express.

One of the most famous cycle-scavenging networks is SETI@home [136], which was using more than 3 million computers to achieve 23.37 sustained teraflops (979 lifetime teraflops) as of September 2001. As of March 2008, Folding@home [137] had achieved peaks of 1,502 teraflops on over 270,000 machines. The European Union has been a major proponent of grid computing. So there is a lot of scope and necessity for efficient advance reservation and scheduling algorithms for the resources geographically distributed in this area of research.
1.6 Contributions

This thesis makes the following contributions towards research in advance reservation and scheduling for grid systems:

- This thesis identifies and proposes the design and development of e-governance architecture with the support of advance reservation and scheduling. To realize this, a distributed framework called the grid architecture for e-governance framework is developed.

- This thesis proposes an advance reservation concept to secure or guarantee resources prior to their executions in grid environment. In the grid framework advance reservation can be applied first to reserve the resources in advance.

- This thesis proposes a scheduling approach to improve the resource utilization after the advance reservation.

- This thesis develops a new hybrid algorithm which is named as “hybrid-core” based on time and cost. The hybrid architecture is designed which deals with the combination of scheduling and advance reservation.

1.7 Organization of the Thesis

The rest of this work is organized as follows. The second chapter describes recent works to give an insight into the latest research advancements in systems related to advance reservation and scheduling in grids.

The third chapter presents a new architecture and design of e-governance applications and this architecture was made as to support advance reservation and
scheduling. It discusses the use of real-world e-governance application models and strategies.

The fourth chapter addresses the topic of modeling and scheduling of task graphs. This explains an advance reservation algorithm to secure resources prior to their executions.

The fifth chapter explains the scheduling algorithm and it is applied in the grid architecture for e-governance applications using grid simulator.

The sixth chapter presents the hybrid architecture for e-governance applications. It discusses the effect of both the advance reservation and scheduling techniques by proposing the new hybrid-core algorithm.

The seventh chapter deals with the discussions on the results obtained in the grid architecture for e-governance applications.

The conclusions and the recommendations for future enhancements are summed up in the eighth chapter.