CHAPTER 1

INTRODUCTION

1.1 OUTLINE

This chapter explains briefly about the focus of research work presented in this thesis. First, it explains the concepts of grid computing. Then it describes the motivation for the research work. After that, it presents the major contributions of this research work. Finally, this chapter gives the details about the organization of this thesis.

1.2 GRID COMPUTING

Grid (Czajkowski et al, 1998), is a kind of open standard distributed infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collection of resources that are belonging to diverse organizations across the globe ensuing users quality of service requirements (QoS). The primary objective of grid computing is to enable cross organizational resource sharing to solve the problems in advanced science and engineering. This objective is not fulfilled by the existing internet architecture since the internet acts as an environment for information sharing, whereas grid acts as a platform for resource sharing. Hence, the grid architecture needs to evolve new components, functionalities of each component and the interaction between them (Foster et al 2001). With the confidence gained in using grid computing, many organizations started to realize their own grid environment. This leads to a problem of interoperability while trying to
combine or integrate them. Without interoperability, collaboration in grid is treated as a bilateral sharing relationship and cannot be applied to others. For example, electric appliance from United States (US) need not work automatically in India. Hence, grid solutions focused towards a specific grid need not always work for other. To achieve this, Grids must be interoperable by means of standards. Standardization means defining a common way to define, implement and interacting with other components. Hence, the Global Grid Forum (GGF) has been evolved as a body to design the standards (GGF 2003).

Grid applications often involve large amount of data and/or computing resources that require secure resource sharing across organizational boundaries. This makes grid application management and deployment a complex undertaking. Consequently, plenty of middleware are proposed to provide support for security, communication, information and resource allocation in a grid environment. Grid middleware provides users with seamless computing ability and uniform access to resources in heterogeneous grid environment. For that, the grid middleware systems must be built on top of grid architecture and standards. Globus Alliance, a community of organizations and individuals developing fundamental technologies behind the grid, along with many other communities, developed Globus Toolkit (GT) (Foster et al 1997). It is an open source software toolkit used for building grid systems and applications. It provides an open source set of services and software libraries that supports the development of grid systems and applications. Though grid middleware is used to build grid infrastructure, it lacks in some of the mandatory components such as matchmaker, scheduler that provides unified environment for application execution. Hence, grid meta-schedulers are evolved.
Grid meta-schedulers orchestrate and allocate the resources that are belonging to multiple administrative domains in a way that the heterogeneity is transparent from the end-users. Clients submit their applications to the meta-scheduler. Then it gathers both static and dynamic resource information of grid sites by querying the information component of underlying grid middleware. Next, it does match-making of the available resource information against the application requirements to find the suitable resources. After that it schedules the applications to these resources on a best-effort basis without any guarantee. Subsequently, the meta-scheduler uses the data management component of grid middleware to transfer the input and executable files to the executing resource. Later, it uses the execution component of grid middleware to start its execution and gather the results. Thus, the grid meta-scheduler provides a collaborative computing environment by unifying the islands of grid environments.

1.3 MOTIVATION

Grid is a technology that tries to realize a logical super computer by aggregating loosely coupled heterogeneous resources over geographically distributed locations to meet the requirements of advanced science and engineering applications. Managing resources in a grid environment poses following five challenges (Czajkowski et al, 1998): autonomy, diversity, policy, manageability and controllability. The autonomy problem arises due to resources, which are participating in grid are managed by diverse organizations at different geographic locations. Hence, it is not possible to have a unique usage policy or security mechanisms for a grid on the whole. The diversity issue is a by-product of autonomy due to the fact that different sites may exploit their own local resource managers like Portable Batch System (PBS), Load Sharing Facility (LSF). The resource management system must provide an application specific infrastructure across various sites
by extending domain specific policies. This will lead to manageability issues while allocating resources from multiple sites. Finally, the controllability arises when the application’s requirements and resource behaviour vary during execution. Many research works have been conducted to address these problems either individually or collectively. But the root that produces these challenges is sharing of resources belongs to multiple institutions. Resource sharing in grid environment is a serious issue since the participating resources belong to different administrative domains and are allowed to take part in more than one Virtual Organization (VO) (Foster et al. 2001). Hence, they may have multiple sharing rules and relationships in different VOs. Moreover, the resource providers clearly specify what can be shared, who is allowed to access and when to share. Since these rules and relationships framed based on the nature, locality and business hours of an organization, they should not be violated during the operation of grid. If they are violated, the resource providers cannot maintain both the accepted resource share and service level in other virtual organizations.

Currently, if a resource provider willing to contribute resources towards grid environment, there is no standard way to specify their sharing rules and relationships available in literature. These sharing relationships and rules need to be expressed as Resource Usage Policies (RUP). Further, a mechanism that gives assurance to the resource providers that their local resources are never overrun is still lacking in the literature. It is important to facilitate the resource providers to precisely specify their desired level of sharing. To obey the desired usage of resource providers, these rules must be considered while selecting the resources for application execution. Hence, it is mandatory to integrate these RUPs with grid meta-schedulers to realize a controlled grid resource sharing. Further, there is no such facility to record and monitor the accepted service level during the application execution. For instance, a grid meta-scheduler schedules the application to a resource
provider by computing the capability and availability on a best effort basis. This assurance is not obeyed, if the resource provides lesser capability than the application’s requirements. This situation eventually leads to failure of the application execution. Hence, it is essential to ensure the availability of resource till the end of execution, monitor the assurance and to penalize the violator in case of falling below the assured level. Though grid has numerous resources, the required software stack or a particular application environment is not always available in these resources. The existing grid infrastructure is not flexible. It cannot be modified according to user requirements so that it is not possible to satisfy all the user application requirements. The above said limitations are not sorted out by any of the conventional grid meta-schedulers.

This research work concentrates on the above mentioned research issues. For that, it explores three major areas: RUPs, Service Level Agreements (SLA) (Czajkowski et al, 2004) and integration of grid and virtualization. The primary objectives of RUPs are to propose a new policy specification language using their own vocabularies, interoperability between different policy specification languages and eXtensible Access Control Markup Language (XACML) based RUP expression and enforcement. It is evident that there is no standard policy specification language to express the RUPs. Hence, it is difficult to integrate resource providers having their RUPs expressed using different policy specification language. Further, each resource provider has their own RUPs with them. Hence, whenever a grid meta-scheduler tries to refer a RUP, it needs to send a request to the resource provider to gather policy information. This will create an unnecessary delay in match-making. So this research work proposes a system to create and manage the RUPs. Also, this system has a policy editor to express the RUPs in an eXtensible Markup Language (XML) format and eliminates the interoperability issues. The vocabularies that are used in the proposed RUPs are extended from Grid Laboratory Unified Environment (GLUE) schema. In
addition, these RUPs are stored in a centralized policy repository located in a grid meta-scheduler thus eliminates requesting the resource providers to gather the policy information.

The exploration of grid SLAs yields that existing systems have three components: SLA Creation, SLA monitoring and SLA enforcement. The SLA creation module initiates the agreement formation by negotiating with the potential resource providers. In current state of art, the negotiation starts with the identification of template that is most suitable for application’s requirements. These templates do not satisfy the application’s requirements completely. If suitable templates are not available with the resource providers, then closer one is selected for negotiation. This will reduce the customer satisfaction and SLA acceptance rate. Also, the negotiation objectives expressed other than in the form of resource usage policies need the presence of provider during negotiation. This limitation introduces a barrier while automating the negotiation. In addition, the negotiation protocol should support multiple negotiation strategies rather than confined to single strategy. This thesis proposes a negotiation framework that expresses the negotiation objectives as RUPs. It also eliminates the template based negotiation. Since the RUPs are located within a meta-scheduler, the unnecessary negotiation delay is avoided. The SLA monitoring module observes the accepted service level by monitoring the job execution. But the existing research works do not concentrate on job or process level monitoring. Hence, this thesis proposes architecture that supports job/process level monitoring of resource parameters during execution. The SLA enforcement module is used to identify the violated objective/ violator and penalize them. In this thesis, all these three modules are implemented as grid services and grouped under Grid SLA Management Architecture (GSMA).
To provide dynamic execution environments to the applications, the grid must be integrated with virtualization technology. The grid architecture does not have the vision to integrate virtualization technologies with grid infrastructure. Hence, there is no specific APIs and services are proposed by grid architecture to support virtualization. But, several services that support creation, monitor and deletion of virtual machines are proposed in our CARE (Center for Advanced Computing Research and Education) Resource Broker (CRB). The above said components need to be integrated with the grid meta-schedulers to discover the required quality and quantity resources as per the requirements of an application and to realize a controlled grid resource sharing. The detailed operations of this integrated CRB are explained by the following example. Consider the scenario, where an user U submits the following job request to the meta-scheduler that match-makes the job request with the available resource providers: “Need 16 CPUs of PARAMPADMA and 32 CPUs of a PC cluster with MATLAB-7.0 for six hours between 2012/12/12 06:00p.m and 2012/12/15 06:00p.m and a network connection providing a bandwidth of 100 Mbit/s between them. The input file (of size 40 Giga Byte (GB)) and the executable should be transferred from Space Application Center (SAC), Ahmadabad to the execution site. From all resource providers (pair of PARAMPADMA, computer cluster and SAC Ahmedabad) choose the one with the highest network bandwidth among them”. After submitting the above said application requirement, the grid meta-scheduler identifies the resource providers matching the application’s requirements. The possible outcomes of matchmaking may be:

**Exact Match:** All the requirements of a particular job request are exactly matches with the available resource parameters. This match occurs, if the resource has the same capability as that of request.
Plug-in match: All the requirements of a particular job request are exactly or over matches the available resource parameters. This match occurs if the resource has equal or greater capability than that of request.

Subsume match: All or part of the requirements of a particular job request are under matches with the available resource parameters. This match occurs if the resource has lesser capability than that of request.

In addition, the above mentioned request clearly states three parts: Hardware, software and QoS requirements. Apart from these requirements, it also specifies the temporal parameters, such as the time at which the job requires resources, spatial connections like bandwidth and delay, external information, such as location of input file and finally the choice of resources like highest bandwidth, when multiple numbers of resources are selected. In order to schedule the above posted job request, first the orchestrator (i.e. grid meta-scheduler) has to filter out the potential resources based on their dynamic information (RUP), such as available period of resources, load and then their static information, such as number of CPUs, OS, RAM size. After that it has to use some negotiation strategy to get their final commitments and finally execute the job over these committed resources. If no resources are available in the exact or plug-in region, then it creates the application specific execution environments as virtual machines over the potential resources in the subsume region and submit the job over them. After the job submission, the application’s runtime parameters are monitored against the accepted service level. If any violation found in accepted level, then the violator get penalized.

1.4 CONTRIBUTIONS

This thesis discusses potential avenues in grid computing for integrating resource usage policies and SLAs to realize a controlled grid resource sharing. In addition, it also explores the various issues related to
scheduling, negotiation, monitoring and enforcement due to the inclusion of policies. This proposed architecture is integrated with our meta-scheduler, CRB as a value addition component. This integrated architecture bridges the above said gaps and assures a controlled grid resource sharing. The additional overhead incurred due to the inclusion of policies by varying both the number of policies and the resources are also evaluated.

In brief, the summarized contributions of this research work are:

a) Policy Management System (PMS) that gives provision to the resource owners to express their desired usage scenarios as resource usage policies,

b) Deviation based Resource Scheduling (DRS) algorithm that aids in SLA guided resource scheduling,

c) Grid SLA Management Architecture (GSMA) that manages negotiation, creation, monitoring and enforcement of SLAs,

d) Comparing and justifying the advantages of the proposed architecture against various existing implementations in terms of scheduling, negotiation, monitoring and enforcement,

1.5 THESIS ORGANIZATION

The rest of the thesis is organized as follows: Chapter two highlights several research works closely related to this thesis. These research works have been grouped into three subsections based on their focus. For example, the research works related to resource usage policies have been organized in a subsection called Policy Management Systems. In addition, the shortcomings of these research works that seed the problem statement for this
thesis are also discussed. Brief justifications on the proposed research works to overcome those shortcomings are also presented.

Chapter three presents a proposed architecture that supports the management of resource usage policies and SLAs in grid environment. This architecture explains the vocabulary and schema for resource usage policy expression. It also explains a way to add, modify and delete the resource usage policies. In addition it also explains the generic negotiation framework for SLA creation, monitoring framework for observing the committed Service Level Objectives (SLO) and enforcement framework for any violation of SLOs.

Chapter four describes the DRS algorithm that orders and schedules the resources based on application’s requirements. The major advantage of DRS is that it also handles the additional scheduling scenarios arise due to the inclusion of virtualization technology.

Chapter five explains the architecture and components of our CRB. In addition, it also explains the integration of PMS, DRS and GSMA with CRB.

Chapter six describes several experiments conducted in order to evaluate the performance of SLA enabled matchmaking system against conventional matchmaking system. It also justifies the impact of inclusion of resource usage policies and SLAs in resource selection. In addition it also compares the proposed components against the existing implementations.

Chapter seven concludes the research work and opens up new avenues for further scope of research.