CHAPTER 6

ONTOLOGY BASED MATCHMAKING SYSTEM IN VIRTUALIZATION ENABLED RESOURCE BROKER (CARE)

6.1 GENERAL

This chapter describes the limitations observed in conventional Grid architecture and the integration of virtualization techniques to address them. The integration of virtualization technologies in Grid infrastructure allows scheduling user’s application to the potential physical resources even if they do not possess required application environment by creating virtual resources as ‘leases’ in them. In such cases, the matchmaking mechanism implemented in conventional Grid scheduler needs to be extended to determine potential resources to host virtual resources. To facilitate this, the Grid scheduler must be provided with information about capabilities of running virtual resources in it and the willingness of resource providers to support creation of resource leases. In this work, we propose an intelligent Grid resource brokering mechanism in which Grid resource information and other usage policies are represented using ontology. Such semantic representation can be exploited to make scheduling inferences using rule based reasoners. It helps the Grid scheduler in making a scheduling decision when potential resources need to be determined to create virtual resources in it for application execution.
6.2 MOTIVATION

Grid is highly dynamic with respect to application requirements. When the scope of Grid gets wider, applications from different domains are accessing it. This puts burden over the resource providers to re-configure the execution environment of the resources to meet the application environment which is always not possible due to the autonomous nature of participating resources. Hence, there are occasions when Grid scheduler fails to locate potential resources due to non-availability of execution environment in them. The emergence of virtualization technologies integrated with Grid infrastructure addressed the above said issue by creating virtual resources in the remote potential resources and dynamically deploy the required execution environment. With the help of virtualization technology adopted in Grid, a Grid client can lease a Grid resource, deploy Grid based application, and customize a complete execution environment resulting in on-demand provisioning of resources. In such a scenario, the Grid scheduler requires appropriate mechanisms for Virtual Machine deployment, creation of virtual clusters, and application execution in the virtual clusters. Existing Grid schedulers lack mechanisms for dynamic creation of virtual Grid resources in remote physical hosts to meet the application execution. For instance, conventional Grid schedulers fail to address the following scenarios:

- Application requires a number of CPUs that shall be satisfied by a single cluster.
- Application requires a number of CPUs that cannot be met by a single cluster,
- Application requires a completely different software environment that no cluster in the Grid can provide
In order to address these limitations, the Centre for Advanced Computing Research and Education (CARE) of Anna University Chennai has recently developed a Resource Broker called CARE Resource Broker (CRB) to support dynamic creation of virtual machines and virtual clusters, and dynamic preparation of execution environment with on the fly installation and configuration of Grid resources. However, this feature puts additional burden in Grid resource management wherein the resource broker has to manage additional information related to the capabilities of physical resources to host virtual resources, their willingness to support such virtualization technologies, and type of virtualization software. Further, CRB uses keyword based matchmaking mechanism that exactly matches the requested resource requirements against the available ones. We extend ontology based Grid resource discovery mechanism to support the scheduling scenarios addressed by CRB. Thus, the extended solution must describe additional resource information which is used to create virtual resources in physical resources. The extended matchmaking mechanism also verifies resource usage policies during resource discovery.

6.3 CARE RESOURCE BROKER

CARE Resource Broker (CRB) is a Grid metascheduler built over Globus middleware. It addresses several scheduling issues encountered when virtualization technology is integrated with Grid. CRB supports creation and management of virtual resources in remote physical resources, and capable of deploying Grid middleware and other related software on the fly. This feature allows CRB to make several application decisions that conventional Grid schedulers cannot do. CRB obtains user’s requirement request for running an application in Grid resources. It then aggregates Grid resource information and discovers suitable physical resources that exactly match with the
requirements. If no such resource is discovered, the broker determines the reason.

- If it is due to non availability of sufficient computing nodes in a cluster, it then identifies a physical cluster (A) that comes closely with the request. It then determines another cluster (B) and suggests formation of required number of CPUs in the form of virtual machines in B. The virtual machines will then be added to Cluster B as computing nodes of A to meet the application requirements.

- If it is due to non availability of execution environment such as simulation tools, operating system and other libraries in any of the cluster. The broker then determines potential physical cluster that can host virtual cluster that can meet the application requirements.

Thus, the resource broker makes several such scheduling decisions in order to schedule application to Grid resources while improving throughput of the Grid system. The matchmaking system implemented in CRB is based on XML information obtained from the underlying middleware and thus supports keyword based resource discovery. Hence, in this work, in order to improve the efficiency of resource management with respect to representation and discovery, the information obtained from middleware, policy information represented using a standard WS-Policy language is converted into ontology representation, and construct knowledge base. Algernon inference engine is used for efficient information retrieval from ontology representation and thus complement the scheduling decisions of CRB.
6.4 SEMANTIC COMPONENT FOR CRB

The central part of the semantic component is the Grid Resource Ontology template. An ontology template is a domain specific ontology that provides hierarchy of concepts along with properties to define their characteristics. The structuring of Grid Resource Ontology template is motivated by the fact that any resource can be modelled as an instance of a specific concept provided that the resource can be described using the properties defined in that concept. Protégé is an ontology editor used for creating ontology using Web Ontology Language. The ontology template is shown in Figure 4.3 that shows several Grid resource concepts such as type of machines, operating system, and resource usage policies and resource information as some of the properties of those concepts.

Once the ontology template is created, knowledge base can be built with the instances and the specific property instantiations. Together the ontology and the knowledge base make up a semantic repository. The semantic description component shown in Figure 4.2 is responsible for constructing knowledge base of Grid resources. Here, this description module has to obtain information from three different sources, the middleware, Virtual Resource Service running in Grid resource that supports virtualization, and resource usage policy expressed in standard WS-Policy language.

When a resource is registered in the Grid, its information can be obtained using Grid resource monitoring tool like MDS of Globus Middleware. With this information, instance of appropriate resource concept in the ontology template is created for every computing resource in the Grid. Virtual Resource Service (VRS) is implemented in CRB to aggregate information related to virtual resource creation and management. In order to create a virtual resource in a physical resource, the following information is needed.
* Kernel
* Image file
* Device Driver Configuration
* Root Device
* LVM Device
* Free LVM disk space
* Total LVM disk space
* Total RAM size allocated to LVM
* Free RAM size available in LVM
* Node Count

Figure 6.1 Architecture of Semantic Component for CARE Resource Broker
This information is not obtained by Grid middleware. Hence, a separate service was written to gather such information and report to Grid scheduler. The semantic description component interacts with this service and obtains this information. Finally, the resource usage policy is expressed in WS-Policy language which is basically an XML document. This policy file expresses the parameters for our prototype implementation namely percentage of hard disk space, and RAM contributed to Grid, number of CPUs to be contributed, willingness to support creation of virtual resources, availability, and budget.

The semantic description component implements a WS-Policy XML parser and extracts the policy information. At this point, the description module possesses complete required information with which it can construct knowledge base of Grid resources. Protégé-OWL APIs is used to dynamically create instance of a particular concept and also to assign values to appropriate properties in the resource ontology template. With these features, the resource information of the entire Grid environment aggregated by the semantic description component can be populated in ontology template thereby creating a knowledge base.

6.5 SEMANTIC DISCOVERY

This component constitutes the actual resource brokering mechanism and it implements matchmaking mechanism. This component accepts user’s application requirements expressed in the form of Job Submission Description Language. The request expresses the hardware and software requirements such as Operating System, Number of CPUs required, and Software libraries. The power of brokering algorithm heavily relies on ontology reasoner used. In our prototype implementation, we have used Algernon inference engine for information retrieval from ontology knowledge base. In Chapter 4, we proposed a similar algorithm but that considers only
physical resource information for matchmaking. However, in this case, the brokering algorithm must consider resource usage policy information, Virtual resource information and physical resource information while discovering a suitable resource for a job. Hence, based on the user’s resource request, the broker forms an Algernon based query and executes in knowledge base. The query also determines semantic relationship between the request and available ones. However, if no such resource found in the first iteration of query over knowledge base, the broker constructs an alternate query to determine physical resources that match all requirements but possesses lesser number of CPUs than that requested. If such resources are available, it can then suggest creating as many virtual machines still required in one of the clusters and connect them as computing nodes to another cluster. If it still fails, the broker constructs yet another query to determine physical resources that exactly matches the hardware requirements but not the software. If such resources are available, the broker suggests creating virtual cluster as requested by the user’s request. In the last two cases, the broker verifies whether the cluster possesses required virtualization software and willing to support creation of virtual machines. Algernon inference engine can model these three individual brokering scenarios into three concurrent ‘path’ and execute all the three ‘paths’ in one go over knowledge base.

In this chapter, the semantic description of Grid resource component was extended to represent hardware information of a computational resource in ontology knowledge base. Accordingly, the semantic discovery component was extended to perform matchmaking to determine potential computational resource in which virtual resources can be created. In the next chapter, an interoperability system is devised to aggregate Grid resource information across several Grid infrastructures running different Grid middleware.