IBIMA Publishing
Journal of Cloud Computing
http://www.ibimapublishing.com/journals/JCC/jcc.html
Vol. 2013 (2013), Article ID 149313, 15 pages
DOI: 10.5171/2013.149313

Research Article

Efficient Resource Provisioning by Means of Sub-Domain Based Ontology and Dynamic Pricing in Grid Computing

Abdul Khalique Shaikh[1], Aminul Haque[2], Saadat M. Alhashmi[3] and R. Parthiban[4]

[1]School of IT, Monash University Malaysia Campus, Bandar Sunway, Malaysia

[2]Department of Comp Sc. & Engg. Daffodil International University, Dhaka, Bangladesh

[3]College of Engineering & Computer Science, Abu Dhabi University, UAE

[4]School of Engineering, Monash University Malaysia Campus, Bandar Sunway, Malaysia

Correspondence should be addressed to: Abdul Khalique Shaikh; shaikh@monash.edu

Received 12 January 2013; Accepted 2 June 2013; Published 29 October 2013

Academic Editor: Tianyong Hao

Abstract

Grid computing was introduced with the provision of delivering high performance computing through connecting distributed heterogeneous resources over the Internet. However, due to the varied nature of Grid applications and lack of suitable resource provisioning tools and mechanisms, a major part of Grid resources remains unutilized. Therefore, provisioning mechanisms must understand how the resources are defined for different applications to obtain maximum accessibility and to increase resource utilization. In this paper, we present a sub-domain based ontology resource provisioning mechanism to increase the utilization of Grid resources. The use of a sub-domain based ontology structure enables us to provision resources in terms of computational context. As the Grid targets resources across distributed domains and ownerships, inspiring mechanisms are essential to increase sharing of resources among the providers. We, therefore, extend our model to measure its effectiveness in an economic system's point of view. We evaluate the significance of using dynamic pricing over static pricing to deal with the dynamic nature of the Grid. The results show improved success probability and system's profit compared to that of traditional resource provisioning mechanisms.

Keywords: Resource Provisioning; Semantic; Ontologies; Profit; Economic Grids; FreePastry; GridSim.

Introduction

The importance of high performance computing such as Grid computing for scientific applications has been broadening day by day. The purpose of a Grid computing is to aggregate physical and logical computing resources to execute and solve problems in large scale structural testing, sensor analysis, weather forecasting and drug design (Lacks and Kocak, 2009). Users want to execute their

Cite this Article as: Abdul Khalique Shaikh, Aminul Haque, Saadat M. Alhashmi and R. Parthiban (2013), "Efficient Resource Provisioning by Means of Sub-Domain Based Ontology and Dynamic Pricing in Grid Computing," Journal of Cloud Computing, Vol. 2013 (2013), Article ID 149313, DOI: 10.5171/2013.149313

applications (jobs) in Grids by maintaining their job deadlines and budgets. On the other hand, providers have their own objectives such as maximizing profit through economically efficient resource provisioning. Prior to submit jobs on the Grids, appropriate resources are selected and allocated by a scheduler or Grid broker. Resources in a Grid system are highly distributed, dynamic in nature and under the control of different virtual organizations typically driven by different rules and policies (Caminero et al., 2007). Ian Foster states (Iamnitchi et al., 2003) that unlike global identification of resources in decentralized systems, it is extremely difficult to define global naming scheme for attribute based resource identification in Grid computing environment. Therefore, it is highly probable that the same resources might be published with different names and it could be possible to skip some relevant resources in syntax-based technique. Therefore, negotiating resources through brokers becomes hard. Due to the usage of fixed schema between users' requirements and providers' availability, a higher number of resources remain unutilized in Grid. Due to the lack of suitable coordination schemes between users and providers, jobs are failing in finding relevant resources. In other words, due to the tight coupling between user requirements and resource availability in a syntax-based matching, a higher percentage of jobs submitted by users is rejected (Neocleous et al., 2007). This rejection leads to a minimization of resource utilization and consequently affects the profit made by the providers.

To overcome the above challenges, firstly, the usage of semantic technology is considered, because semantic matching helps minimize the tight coordination between resource providers and users. Therefore, a semantic-based resource discovery mechanism has been developed by extending two existing Grid computing ontologies – *Operating System* and *Processor Architecture* from (Vidal et al., 2009). After developing two ontologies, we have computed semantic similarity values between the concepts defined in (Andreasen et al., 2003). By using this approach, users can also find the relevant resources, if even an exact match is not available for their jobs. We establish a decentralized network in FreePastry simulator (Druschel et al., 2012) using the Pastry protocol and create Grid computing entities in GridSim (Buyya and Murshed, 2002). Details about the usage of both simulators will be explained in Section 3.

We have further extended our model towards developing a sustainable economic system. The development incorporates a pricing mechanism to encourage resource providers to contribute their resources to the Grid. Grid entities are typically regarded as self-interested and are driven by their own parameter models. Economic-based approaches are found efficient to control and regulate the behavior of these entities help constructing a consistent computing platform (Buyya et al., 2000). Realizing the distributed and dynamic nature of the Grid, in this paper, we consider supply and demand driven pricing where resource costs change dynamically to reflect on the dynamic changes in supply and demand. We find that, by integrating this pricing mechanism, our sub-domain based ontology model demonstrates significant improvement in resource utilization as well as profit made by the providers. This work is an extension of our previous work (Shaikh et al., 2011), towards the better utilization of Grid resources where we used Chord decentralized protocol with existing ontologies. However, the Pastry protocol provides better results and overcome current limitations in terms of overall communication overheads.

Our main contributions in this paper include, (i) the design and development of an efficient decentralized resource provisioning mechanism by using sub-domain based ontology structure and (ii) the incorporation of economic parameters to understand the system's effectiveness in terms of a competitive economics' system.

In Section 2, we present some works related to centralized and decentralized resource provisioning mechanisms and economic-based distributed resource collaboration. The description of our proposed framework, the working principle of the Pastry protocol, mapping ontologies in resource discovery and pricing mechanism are explained in Section 3. Section 4 provides an experimental evaluation of our work. Finally, the paper presents the conclusion with future directions in Section 5.

Related Work

Existing real Grid systems such as CONDOR (Frey et al., 2002), Globus (Schopf et al., 2006), gLite (Kretsis et al., 2009) utilize centralized and hierarchical approaches for resource management. These approaches provide a significant advantage of considering the simplicity of finding Grid resources on a central point. However, due to the centralized nature, these approaches suffer from scalability, low fault tolerance and single point of failure. Due to the above limitations, we consider decentralized approach.

A significant amount of research has been carried out to improve Grid resource management by using decentralized approach in both academia and industry. The approach fits for a large scale Grid system and helps increase scalability, load balancing, heterogeneity and fault resilience (Buyya et al., 2000). Most of the decentralized approaches use a basic DHT technique to publish and subscribe Grid resources. For instance, a DHT-based approach has been introduced in (Albrecht et al., 2008). Authors of (Albrecht et al., 2008) have designed and implemented a Scalable Resource Discovery service for Wide-area distributed system (SWORD). SWORD allows users to describe desired resources as a topology of interconnected groups with required inter and intra groups. Authors claim that the service built on a top of DHT automatically inherits the DHT's self configuration, self healing and scalability. However, Kim et al., (Kim et al., 2009) mention that SWORD network latency is affected for high bandwidth demanded applications. By considering the above issues, an extended version of DHTs such as, Chord and Pastry are proposed in a P2P networking systems, which could also be helpful in Grid environment. Ranjan et al., (Ranjan et al., 2007) present decentralized resource discovery services for large scale federated Grids that utilize a P2P spatial publish/subscribe index using Chord protocols. A fixed schema has been used between providers resources and users request. Authors of the paper (Ranjan et al., 2007) consider the cost of resources at the time of publishing and budget by users at the time of subscriptions. Their results show that the percentage of successful queries is affected significantly due to the increase in query rate under average queue size. However, the impact of economic parameters in their results is ignored.

We analyze that all above research work are based on syntactical approaches, where there is a high possibility of missing relatively close resources that could affect utilization of resources. By adding the features of semantic, we can overcome this issue. Hence, we deploy our model in a sub-domain based decentralized environment. The paper (Vidal et al., 2007) has proposed semantic Grid architecture to describe and discover resources. Authors show some comparisons between semantic and non-semantic results on extremely low scale where a domain-based Ontology is used and show some relationship between concepts of ontologies. However, a domain–based Ontology can lead to wrong decision in the matchmaking process. Therefore, in this paper, we propose sub-domain based ontology structure. Another domain-based ontology approach named "OntoSum" has been proposed in (Li., 2010) for efficient resource information integration and services which improves the search expressiveness, efficiency, quality and scalability in a decentralized way. However, the results of OntoSum are not based on real ontology concepts, and authors used artificially generated semantic data for experiment.

We present some works related to economic-based distributed resource management below:

In Grid computing, the potential of economic models was mainly emphasized in 2002 through a discussion about the need of economic-based resource management (Buyya et al., 2002). However, this discussion about different economic models was only based on their hypothetical suitability for the Grid, not based on any experimental evidence. Since the introduction of various economic models in the Grid (Buyya et al., 2002), an extensive research has been conducted to understand the effectiveness of the models for distributed resource collaboration (Haque et al., 2011).

Commodity Market Model (CMM) is one of the most widely proposed economic models in the Grid (Haque et al., 2011). The model's main strength is drawn from its ability to maintain equilibrium between resource supply and demand. Maintaining supply and demand by regulating price behavior ensure a high probability to deliver requested QoS (Quality of Service) for users as well as to increase system performance. The main principle behind this model is to determine an equilibrium/spot price at which the aggregated supply and demand of the market can be diminished. For example, if demand for a resource exceeds its supply at a particular state, the price of that resource increases in such a way so that the demand function shifts to a point closer to the available supply. Various techniques are used to determine the spot price in the literature (Stuer et al., 2007). Richard et al., identify the suitability of CMM for maintaining market equilibrium and minimizing communication overhead (Wolski et al., 2001). Realizing the suitability of CMM, in this paper, we integrate a supply and demand driven pricing mechanism in our sub-domain based decentralized model.

Therefore, the main focus of this paper is to combine semantic features and economic services in a decentralized resource provisioning system towards enabling adaptive and robust resource management architecture. To the best of our knowledge, existing literature have not focused on the combination of these two significant approaches.

The Proposed Framework

This section describes our proposed framework that shows how overlay network is built, and how Grid and network entities are linked together. It further shows how resources are published by providers and are searched by users. It also explains the working behavior of the Pastry protocol with the principles of semantic and economic-based systems.

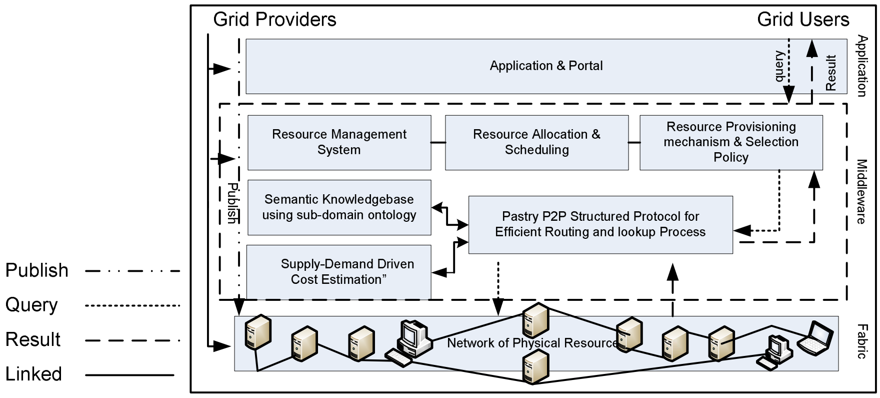

Grid Providers
Grid Users
Application & Portal
Query
Result
Application
Resource Management System
Resource Allocation & Scheduling
Resource Provisioning mechanism & Selection Policy
Publish
Semantic Knowledgebase using sub-domain ontology
Pastry P2P Structured Protocol for Efficient Routing and lookup Process
Supply-Demand Driven Cost Estimation"
Middleware
Network of Physical Resources
Fabric
Publish
Query
Result
Linked

Fig. 1: A Layered Architecture of the Proposed Framework

Fig. 1 indicates that providers publish Grid resources through the application layer which provides an interface to reside resources to the fabric layer. Resource Management System (RMS) with the help of P2P overlay (Pastry) at middleware layer facilitates to distribute the resources according to the scheduling policy defined by providers. When users query the resource requirements for their Gridlets (an application is decomposed into many Gridlets) through an application layer then Gridlets adapts the Pastry routing mechanism to reach the target node. The selection of resources is based on semantic knowledgebase and cost estimation module that are maintained at core middle layer. The details of mapping of ontologies and dynamic price mechanism with the Pastry protocol are defined as follows:

Pastry P2P Protocol

The Pastry protocol (Rowstron and Druschel, 2001) is scalable, and self administered that was initially designed for wide area P2P applications such as global data sharing & storage, group communication and naming purpose. However, we utilize this protocol in a Grid environment as a Grid computing resembles with P2P architecture in the context of resource-sharing environments (Karaoglanoglou and Karatza, 2009). A main reason for choosing the Pastry protocol is that Pastry reduces communication overheads compared to other P2P protocols such as Chord (Lua et al., 2005). First, we build decentralized overlay network in the FreePastry simulator using the Pastry protocol. After creating P2P overlay network, we develop nodes and insert Grid resources under these nodes. Grid resources are created in GridSim (Buyya and Murshed, 2002) and integrated into FreePastry to carry out simulations. The reason of using two simulators in our proposed model is GridSim provides the platform for creating Grid resource related parameters which are much closure to a real Grid environment. On the other hand, FreePastry provides decentralized overlay network with efficient routing and location mechanism. Along with the basic resource characteristics, we add three extra parameters in Grid resources such as resource architecture, resource operating system and resource cost. The Pastry assigns a 128-bit identifier *Node Id* for each node in hexadecimal format. Each node maintains a leaf set, a routing table, and a neighborhood set that carries latest information about other nodes (Rowstron and Druschel, 2001) and keeps track of its immediate neighbors. As the routing mechanism is concerned, a node can route a message/query to its numerically closest nodes. In Pastry, the total routing are less than ($log_B N$) steps under normal operation (N is a number of nodes and $B= 2^b$ where b = number of bits used for the base of the chosen identifier with a typical value 4). For instance, if we build a Pastry network of 20 nodes and assume that only 7 nodes occupied Grid resources and generate the following resource IDs randomly:

Table 1: Resource Nodes with Generated IDs

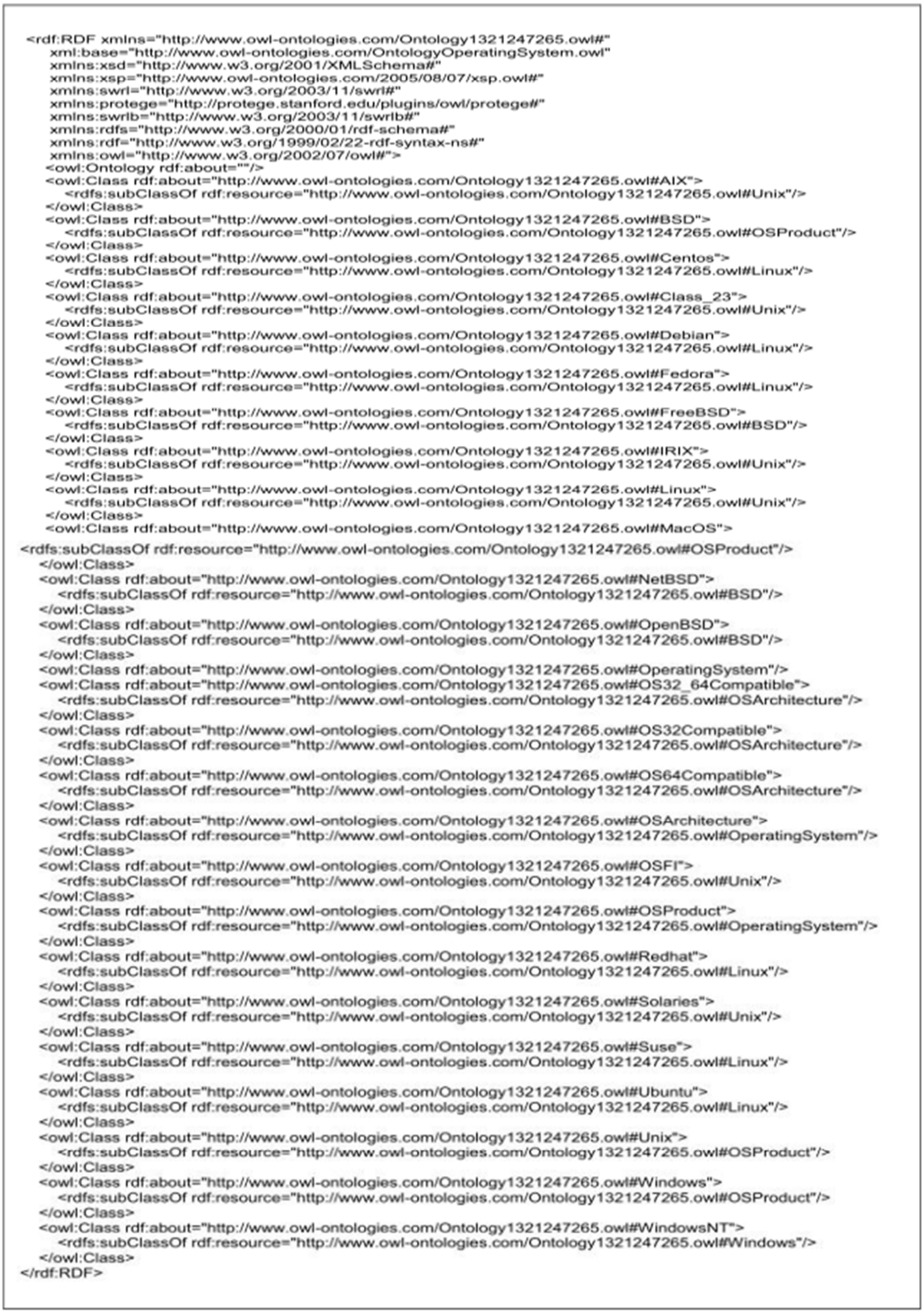
Resource Node	Generated ID	Resource Node	Generated ID
RNode1	2924C1	RNode2	3212D4
RNode3	3637B4	RNode4	3803F2
RNode5	3835C4	RNode6	3836A5
RNode7	3932C5		

With the help of generated IDs as mentioned in Table 1, the resource nodes are resided in a circular space and users are able to find resources for their jobs. Now, it is assumed that the user is looking for a resource that is resided on node 3836A5. The look up process begins from the closure of users' node. According to the Pastry routing algorithm, a query can efficiently route in a highly distributed network with a minimum number of hops. As in above scenario, the job finds a target node within 4 hops. When a query travels from one node to another node in the network, it completes one hop. When the job reaches the destination node 3836A5 then comparison process starts. In case, an exact requirement for the job is not taken rather it will go for semantic matchmaking based on the semantic threshold value set by the user. If it matches the semantic threshold value with the resource's semantic similarity value, the job is submitted otherwise the job will be rejected from this particular resource, and it will move forward to another available resource. Details about measuring the semantic similarity are discussed in sub-section 3.2. In the next section, we will describe the semantic deployment in decentralized Grids.

Mapping of Ontologies in a Decentralized Resource Provisioning Mechanism

This section explains the mapping of ontologies that we have used for Grid resource provisioning in a decentralized Pastry environment. A sub-domain based ontology approach is useful to identify the relationship between Grid resources (Chen and Tao, 2008). For this purpose, we have extended two Grid computing resource Ontologies such as *Operating System* and *Processor Architecture* from paper (Vidal et al., 2009) using Ontology editor & knowledge base framework protégé (Standford, 2011). Ontology is a set of concepts and relationship between concepts (Liangxiu and Berry, 2008) that provides meta information which describe semantic data (Fensel, 2004).

The OWL code of extended developed Ontology for *Operating system* is mentioned as follows:



```
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1321247265.owl#"
     xml:base="http://www.owl-ontologies.com/OntologyOperatingSystem.owl"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
     xmlns:swrl="http://www.w3.org/2003/11/swrl#"
     xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
     xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#AIX">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#BSD">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSProduct"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Centos">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Class_23">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Debian">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Fedora">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#FreeBSD">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#BSD"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#IRIX">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux">
      <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
    </owl:Class>
    <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#MacOS">
<rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSProduct"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#NetBSD">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#BSD"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OpenBSD">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#BSD"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OperatingSystem"/>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OS32_64Compatible">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSArchitecture"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OS32Compatible">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSArchitecture"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OS64Compatible">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSArchitecture"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OSArchitecture">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OperatingSystem"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OSFI">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#OSProduct">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OperatingSystem"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Redhat">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Solaries">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Suse">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Ubuntu">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Linux"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Unix">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSProduct"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#Windows">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#OSProduct"/>
   </owl:Class>
   <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1321247265.owl#WindowsNT">
     <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1321247265.owl#Windows"/>
   </owl:Class>
</rdf:RDF>
```

Fig. 2: OWL Code for Operating System Ontology

Before using this Ontology in provisioning Grid resources, we need to measure the relationship between concepts of Ontology. The degree of relationship between Ontology concepts is known as semantic similarity. No standard procedure is available to measure the semantic similarity. However, a survey paper (Schwering, 2008) compares and contrasts the various models to measure the semantic similarity distance between several ontological concepts. For our framework, we select the semantic measurement formula based on the network model because the network model measures similarity based on the notion of the distance (short path algorithm). Andreasen et al., (Andreasen et al., 2003) derive conceptual similarity using the notion of "similarity graph". In this, Ontology is represented as a graph with concepts as nodes and relationships connecting these concepts as edges. Authors from the same papers introduce the following equation and use a function, *sim* (*x, y*) to measure the degree of similarity, which is proportional to the common concepts x and y share.

$$\boldsymbol{Sim(x,y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1-\rho)\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|}} \quad (1)$$

In Equation (1), ρ is a factor that determines the degree of influence of generalization of Ontology concepts. The value of ρ is between 0 and 1. If the value of ρ is 1, that means perfect generalization, with each and every concept defined properly and 0 means very poor generalization. We set the average generalization value of $\rho = 0.5$. $\alpha(x)$ is the set of nodes reachable from x and $\alpha(x) \cap \alpha(y)$ *i*s the reachable nodes shared by x and y. $Sim(x,y) = 0$ means x and y are entirely dissimilar and $Sim(x,y) = 1$ means fully similar. To compute semantic similarity for extended Ontology concepts, we use Equation (1). The semantic similarity values range is as follows:

$$0 < Simthr \leq 1 \quad (2)$$

A sub-domain based decentralized resource provisioning model is able to find out the relationship between a resource properties and a job requirement in case of exact match is not possible. In this way, job success probability can be enhanced to increase the utilization of resources. Before explaining the experimental results, we describe the cost estimation process for resources.

A Dynamic Pricing Mechanism

As mentioned earlier, the main focus of our pricing mechanism is to determine a spot (equilibrium) price based on current supply and demand functions in the environment. Whenever there is a change in supply and/or demand, our mechanism is invoked to update its current supply and demand. A resource can request for the current spot price to the price determinator before the resource negotiates with users with the most updated cost. We use linear algorithm for our price determination process. According to linear equilibrium theory, the demand and supply functions are given as,

$Q^D = -aP + b$

$Q^S = cP + \alpha$

Where Q^D refers to the quantity demanded at any specific time and Q^S is for supply; *a*, *b* and *c* are the scalar parts where *a*, *c* are the change in demand and supply respectively, *b* is the *current_demand* which is defined by the number of available Gridlets in the market still looking for resources. The values of *a*, *c* are ranges from 0 to 1. The negative sign in the demand function presents the relationship between price (*P*) and demand, which is, an increase in price will induce a decrease in the quantity demanded and vice versa. In supply function, α refers to the shift in supply which can be manipulated as,

$\alpha = (\mathit{initial\_supply} - \mathit{current\_supply}) / \mathit{initial\_supply}$

$\alpha = 1 - (\mathit{current\_supply} / \mathit{initial\_supply})$

Again, *a* is determined by calculating *current_demand* over the *total_demand* and *c* is calculated as *current_supply* over the *total_supply*. *Current_supply* is defined by the number of nodes available to serve Gridlets. Now, if we want to know the price at which total supply and demand at any given state diminishes, we need to solve the supply and demand functions for *P* when $Q^D = Q^S$. If P^* be our *spot price*, we get,

$$P^* = (b - \alpha) / (a + c) \qquad (3)$$

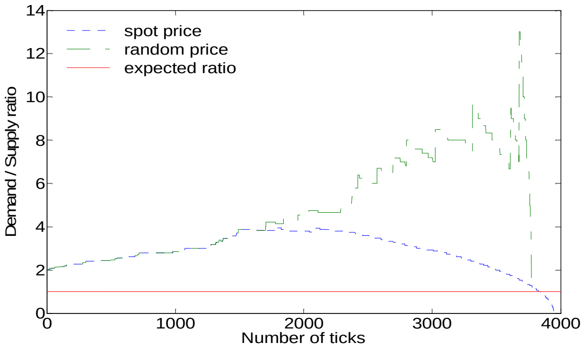


Fig. 3: The Effect of Spot Prices over 100 Jobs and 50 Nodes

At every turn, *cpuCost* of a resource is computed using current *spot-price*. Our main reason for using such dynamic pricing is to maintain equilibrium between supply and demand, and thus, to increase providers' profit. In order to ensure that the defined *spot price* is working for pushing the market into equilibrium, we conduct a basic experiment with 100 Gridlets and 50 nodes. We have this figure appeared in our earlier work (Haque et. al., 2012). The effect of *random price* and *spot price* in the market is shown in Fig. 3.

The effect of *spot prices* is observable in Fig. 3. Because of the effect, the trend for the demand-supply ratio is smothering compared to the trend using *random prices*. On the other hand, because the random prices are not affected by supply and demand, the trend fluctuates randomly. The experimental evaluation of our pricing scheme is appended in the following section.

Experiment

In this section, we explain the details of our experimental evaluation. We integrated two discrete event-based simulators i.e. GridSim and FreePastry to measure the effectiveness and efficiency of both Grid entities and network related performance matrices.

First, a decentralized overlay *EuclideanNetwok* is developed in FreePastry environment. After that, we create 512 resources using GridSim and active the overlay *network* by assigning the resources over the nodes (one resource is assigned to one node). Parameters related to the network, nodes, jobs and resources are shown as follows:

Table 2: Resource Configuration

Parameters	**Relevancy**	**Values**
Network Bandwidth	Providers	100 mbps
Number of Machines per resource	Providers	4-10 with step of 2
Number of Processor per machine	Providers	5-80
MIPS Rating of a PE (Processing Element)	Providers	350-450
Operating System (OS) ontology	Providers/Users	25 concepts
Processor Architecture ontology	Providers/Users	31 concepts
Cost-per-sec (G$- Grid Dollar)	Providers	1.0-2.0
Job length (MI- Million Instructions)	Users	1000-10000
Job Deadline (Sec)	Users	10- 22
Job Budget (G$)	Users	15-40
File size (KB)	Users	300
Output File size (KB)	Providers/Users	400
Semantic Threshold	Providers/Users	0-1

We keep the size of our network constant, which is of 512 nodes and vary the number of jobs between 100 and 500 with step 100 to understand the effect of different supply and demand. A node can consist of several machines and one machine could have several PEs. Both 31 and 25 concepts of *Operating System* and *Processor Architecture* have been considered for both jobs and resources to maintain consistency. Our Equation 3 provides price, which is the *Cost-per-sec* in Table 2. However, the original job execution cost depends on job length and MIPS rating of the node where the job is intended to be executed. We use the same formula to operate this as used in GridSim, that is,

$$cpuCost = job\text{-}length * (cost\text{-}per\text{-}sec / MIPS\text{-}rating\ of\ n) \qquad (4)$$

This *cpuCost* is then compared with the job-budget to facilitate the matchmaking process. Again, the job-deadline is compared with *cpuTime* to know whether the node is able to execute the job within its deadline. The *cpuTime* is computed as follows.

$$cpuTime = gl\text{-}length / MIPS\text{-}rating\ of\ n \qquad (5)$$

To maintain the statistical significance of our simulation, we have used *Random Uniform Distribution* to generate random values between the ranges (Table 2). To ensure the consistent sample generation, pseudorandom approach (generator is seeded with a particular value) is used.

Simulative Study

This section first explains the simulation results obtained for sub-domain ontology based resource provisioning. This mainly compares the results between semantic and traditional non-semantic approaches.

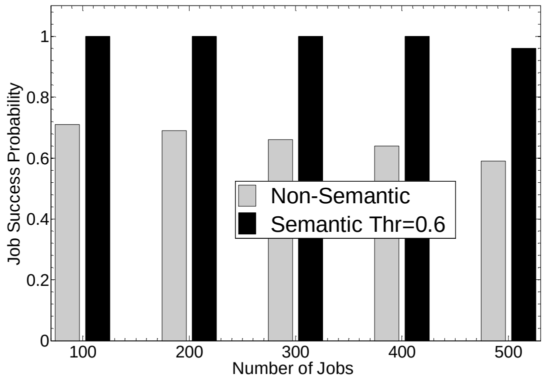


Fig. 4: Comparison for Success Probability

Fig. 4. shows the relationship of job success probability with a number of jobs for semantic and non-semantic cases. The result shows that a significant number of jobs are rejected under non-semantic case compared to the semantic. This rejection is due to the tight coupling between user requirements and resource availability. As a result, the resources are not utilized well. As we increase the number of jobs, the performance for semantic approach remains almost constant. The reason is that semantic matching helps removing the tight coupling between users and resources that helps enhance the job success probability.

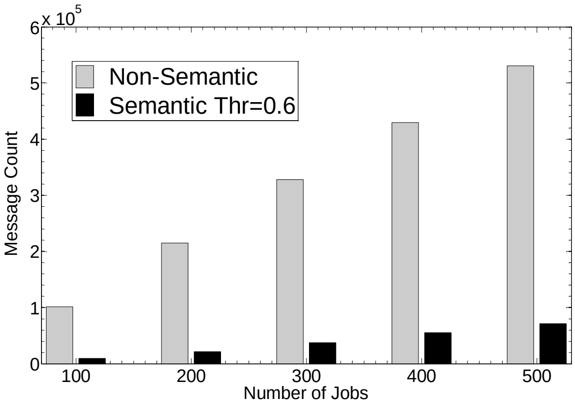


Fig. 5: Comparison for Communication Overhead

Fig. 5. shows the relationship between message count with the number of jobs. In terms of semantic approach, communication overhead is significantly lower compared to that of the non-semantic. As the probability of being matched for a user and a resource is extremely low in non-semantic, the users in this case keep exploring to find out their suitable resources. This results in exchanging a large amount of messages in the network. It is also observed that the ratio of increasing message is much higher in non-semantic compared to the semantic when demand increases.

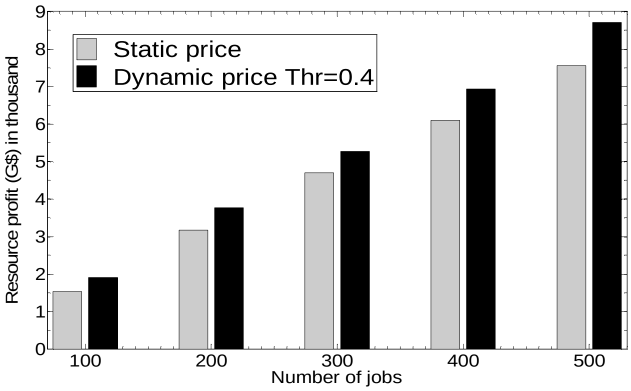


Fig. 6: Profit Comparison

Static Price and Dynamic Price in the Sub-Domain Based Ontology Model: A Combined View

Fig. 6. illustrates the comparison for profit earned by the resources in case of static and dynamic prices. Because of using dynamic pricing, the system not only maintains equilibrium between supply and demand but also ensures a better profit for resources. As we increase the number of jobs (demand) in the environment, the profit tends to be higher. For 500 jobs the difference between the results for static and dynamic prices are the highest. When the demand is 100, due to the low demand compared to the supply (512), the spot prices generated by the Equation 3 are lower. These results for the resources to be provisioned with low costs make it hard by the resources to maximize their profit. However, as the demand increases, the spot prices keep increasing accordingly. Therefore, the system is unable to utilize users' budgets, which makes the profit higher. We can predict the scenario when demand will exceed the supply (512) from Fig. 6. On the other hand, as the pricing behavior remains unchanged in terms of static scenario, there is not much difference in profit for a particular scenario. Table 3 explains the profit for different semantic threshold values.

Table 3: The Effect of Dynamic Price on Resource Profit

Semantic thr./ No of jobs	0.4	0.6	0.8	1.0
100	1.54–1.90 (Static-Dynamic)	1.48–1.93	1.42–1.62	0.87–1.08
200	3.16–3.77	3.07–3.45	2.88–2.89	1.77–2.00
300	4.70–5.27	4.52–4.56	4.14–3.92	2.60–2.73
400	6.10–6.94	5.77–5.91	5.03–4.94	3.36–3.35
500	7.55–8.70	6.96–7.20	5.87–5.74	3.88–3.86

Table 3 presents the comparison for profit in terms of static and dynamic pricing for different semantic threshold values. We have already explained the result for 0.4 semantic threshold values in (Fig. 6.) We can observe from Table 3 that as we increase the semantic threshold, the difference in profit between static and dynamic scenarios becomes lower. In other words, as we move towards non-semantic, the difference decreases. The reason is that in terms of non-semantic, the job success probability starts decreasing (Fig. 4.); therefore, the aggregated potential of dynamic price over the static price is low. We can further observe, in terms of higher threshold (0.8, 1.0) and higher demand, the static pricing is performing better. Because of higher demand, the spot price increases. This increment compounding with the higher threshold makes the job success probability extremely low for dynamic pricing. For static pricing, on the other hand, the job success probability does not move down tremendously as it is not increasing the resource cost over time. This is the reason why static pricing generates higher profit in these particular regions.

Conclusions and Future Work

Due to the low resource utilization in Grid computing, the expected revenue of providers is affected. Hence, there is a need of a comprehensive mechanism for resource provisioning so providers could get maximum benefits. In this paper, we presented a decentralized resource provisioning model by using a sub-domain based ontology structure to increase the utilization of resources in an economic Grid. We extended two existing Grid resource ontologies and utilized them in a P2P decentralized Pastry environment. We also evaluated the significance of dynamic pricing over static pricing to deal with the dynamic nature of the Grid. Our results showed that by integrating this pricing mechanism in the sub-domain based model, resources are making better profit compared to conventional resource provisioning mechanisms. We believe that our findings in this paper would help in better understanding the diverse application models in the context of dynamic Grid computing environment. As a future work, we would like to implement our proposed model in a real Grid environment to measure the real time adaptability of our model.

References

Albrecht, J., Oppenheimer, D., Vahdat, A. & Patterson, D. A. (2008). "Design and Implementation Trade-Offs for Wide-Area Resource Discovery," *Acm Transactions on Internet Technology,* 8, 1-44.

Andreasen, T., Bulskov, H. & Knappe, R. (2003). From Ontology over Similarity to Query Evaluation, *2nd Colognet-Elsnet Symposium-Questions and Answer: Theoretical and Applied Perspective, Amsterdam, Holland.* 39-50.

Buyya, R., Abramson, D., Giddy, J., Campus, C. & Melbourne, A. (2000). An Economy Driven Resource Management Architecture for Global Computational Power Grids, *The 7th International Conference on Parallel and Distributed Processing Techniques and Applications* (Pdpta 2000), June 26-29 2000 Las Vegas,Usa.

Buyya, R. & Murshed, M. (2002). "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," *Concurrency and Computation: Practice and Experience,* 14, 1175-1220.

Buyya, R., Stockinger, H., Giddy, J. & Abramson, D. (2002). "Economic Models for Resource Management and Scheduling in Grid Computing," *Concurrency and Computation: Practice and Experience,* 14, 1507-1542.

Caminero, A., Sulistio, A., Caminero, B., Carrión, C. & Buyya, R. (2007). "Extending GridSim with an Architecture for Failure Detection," *13th International Conference on Parallel and Distributed Systems* (Icpads 2007). 1-8.

Chen, L. & Tao, F. (2008). "An Intelligent Recommender System for Web Resource Discovery and Selection," *Intelligent Decision and Policy Making Support Systems.*

Druschel, P., Haeberlen, A., Hoye, J., Iyer, S., Mislove, A., Nandi, A., Post, A. & Singh, A. (2012). Free Pastry Software [Online]. Available: http://www.freepastry.org/FreePastry/ [Accessed Jan 20 2012].

Fensel, D. (2004). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, *Springer Verlag.*

Frey, J., Tannenbaum, T., Livny, M., Foster, I. & Tuecke, S. (2002). "Condor-G: A Computation Management Agent for Multi-Institutional Grids," *Cluster Computing,* 5, 237-246.

Haque, A., Alhashmi, S. M. & Parthiban, R. (2011). "A Survey of Economic Models in Grid Computing," *Future Generation Computer Systems,* 27, 1056-1069.

Haque, A., Alhashmi, S. M. & Parthiban, R. (2011). "An Inspiration for Solving Grid Resource Management Problem Using Multiple Economic Models," *Economics of Grids, Clouds, Systems, and Services.* Lecture Notes in Computer Science Volume 7150, 2012, Pp 1-16.

Iamnitchi, A. & Foster, I. (2003). A Peer-to-Peer Approach to Resource Location in Grid Environments, *International Series in Operations Research and Management Science. Springer.*

Karaoglanoglou, K. I. & Karatza, H. D. (2009). "Performance Evaluation of a Resource Discovery Scheme in a Grid Environment Prone to Resource Failures," *Parallel & Distributed Processing, 2009. Ipdps 2009. Ieee International Symposium On,* 1-8.

Kim, J., Chandra, A. & Weissman, J. B. (2009). "Using Data Accessibility for Resource Selection in Large-Scale Distributed Systems," *Ieee Transactions on Parallel and Distribution System*, 20, 788-801.

Kretsis, A., Kokkinos, P. & Varvarigos, E. (2009). "Developing Scheduling Policies in Glite Middleware," 9th *Ieee/Acm International Symposium on Cluster Computing and the Grid. Ieee Computer Society,* 20-27.

Lacks, D. & Kocak, T. (2009). "Developing Reusable Simulation Core Code for Networking: The Grid Resource Discovery Example," *Journal of Systems and Software,* 82, 89-100.

Li., J. (2010). "Grid Resource Discovery Based on Semantically Linked Virtual Organizations," *Future Generation Computer Systems,* 26, 361-373.

Liangxiu, H. & Berry, D. (2008). "Semantic-Supported and Agent-Based Decentralized Grid Resource Discovery," *Future Generation Computer Systems,* 24, 806-812.

Lua, E. K., Crowcroft, J., Pias, M., Sharma, R. & Lim, S. (2005). "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *Ieee Communications Surveys and Tutorials,* 7, 72-93.

Neocleous, K., Dikaiakos, M., Fragopoulou, P. & Markatos, E. (2007). "Failure Management in Grids: The Case of the Egee Infrastructure," *Parallel Processing Letters,* 17, 391-410

Ranjan, R., Lipo, C., Harwood, A., Karunasekera, S. & Buyya, R. (2007). Decentralised Resource Discovery Service for Large Scale Federated Grids, *e-Science and Grid Computing, Ieee International Conference* on, 10-13 Dec. 2007 Banglore India. 379-387.

Rowstron, A. & Druschel, P. (2001). "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Springer,* 329-350.

Schopf, J. M., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'arcy, M. & Chervenak, A. (2006). "Monitoring the Grid with the Globus Toolkit MDS4," *Journal of Physics,* 46 521-525

Schwering, A. (2008). "Approaches to Semantic Similarity Measurement for Geo-Spatial Data: A Survey," *Transactions in Gis,* 12, 5-29.

Shaikh, A. K., Alhashmi, S. M. & Parthiban, R. A (2011). "A Semantic Decentralized Chord-Based Resource Discovery Model for Grid Computing," *Proceeding of 17th Ieee International Conference on Parallel and Distributed Systems* (Icpads 2011), 7-9 Dec. 2011. 142-148.

Standford. (2011). Ontology Editor & Knowledge Base Framework, [Online]. *Standford University Uk.* Available: http://protege.stanford.edu/ [Accessed 12 Dec 2011]

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. &

Balakrishnan, H. (2003). "Chord: A Scalable Peer-To-Peer Lookup Protocol for Internet Applications," *Networking, Ieee/Acm Transactions on,* 11, 17-32.

Stuer, G., Vanmechelen, K. & Broeckhove, J. (2007). "A Commodity Market Algorithm for Pricing Substitutable Grid Resources," *Future Generation Computer Systems,* 23, 688-701.

Vidal, A. C. T., Jos, F., Silva, D. S. E., Kofuji, S. T. & Kon, F. (2007). Semantics-Based Grid Resource Management, *5th International Workshop on Middleware for Grid Computing, Newport Beach, California. Acm.*

Vidal, A. C. T., Silva, F. J. S., Kofuji, S. T. & Kon, F. (2009). "Applying Semantics to Grid Middleware," *Concurrency and Computation: Practice and Experience,* 21, 1725-1741.

Wolski, R., Plank, J. S., Brevik, J. & Bryan, T. (2001). "G-commerce: Market Formulations Controlling Resource Allocation on the Computational Grid," *Proceeding of the15th Ieee International Conference on Parallel and Distributed Processing Symposium*, April 2001 San Francisco, Ca Usa. 1-8.