Invited Paper:

An Economic View of Indirect Reputation Management for Grids

Javier Echaiz* Jorge R. Ardenghi
Institute of Computer Science and Engineering, Universidad Nacional del Sur
Bahía Blanca – Buenos Aires – República Argentina
{je,jra}@cs.uns.edu.ar

Abstract

Scientific collaboration are becoming interdisciplinary, and scientists are working in informal collaboration to solve complex problems that require multiple types of large resources. An option is a computational grid. A computational grid is a distributed infrastructure that appears to the end user as one large computing resource across organization boundaries. Grid technologies enable large-scale sharing of resources within formal or informal consortia of individuals and/or institutions, usually called virtual organizations. In these settings, the discovery, characterization, management, and monitoring of resources, services, and computations can be challenging due to the considerable diversity, large numbers, dynamic behavior, and geographical distribution of the entities in which a user might be interested.

Trust is one of the biggest concerns in the grid resource management field. Grid systems can employ reputation mechanisms in order to provide this essential trust, but not usually without incurring in certain additional costs that negate the potential performance gains offered by grid computing technologies. Moreover, current reputation mechanisms are not appropriate for resource management in large-scale systems (generally used in P2P).

In this paper, we present a new reputation model for resource management based on a economy model. Also we demonstrate how it can by employed to add trust into algorithms for grid scheduling. Finally, we simulate the proposed resource management algorithm in order to verify its effectiveness.

Key Terms: *grid resource management, economic models, reputation models.*

1 Introduction

Needs of computational resources is increasing day by day because of the universe of computationally intensive tasks. For carrying out certain scientific experiments, simulations, problem solving, capacity of single stand alone machine is inadequate. Hence all organizations end up buying dedicated resources like supercomputers and mainframes with exorbitant costs. During the last two decades the development of low cost powerful microprocessors and high speed computer networks have promoted a change in the computing paradigm. Today, big mainframes have been replaced by low cost and highly powered collaborative infrastructure that would make effective utilization of computational resources owned by the organization. However, this potential cannot be fully utilized unless the users are able to *transparently* access these resources. By transparency, we mean that the users should be able to access any resource without worrying about (and indeed, without being aware of) its physical location. This has been the main motivation behind evolution of Resource Management Systems (RMS).

The concept of grid computing has grown far beyond its original intent of linking supercomputing sites. Through a web browser, a command line interface, or a graphical desktop, users are able to view and select all the grid resources and services in a virtual infinite machine room [29]. Building a grid requires the development and deployment of a number of services, including those for: resource discovery [47, 10, 24], scheduling configuration management [43, 17], security [37, 36], and payment mechanisms [1] in an open environment.

The complexity of the design of grid brokerage system lies in the functionalities that such a system should implement and in the characteristics that it should have. These are:

- Ability to cope with the different ownerships of resources and jobs.

*Supported by a fellowship of the "Consejo Nacional de Investigaciones Científicas y Técnicas" (CONICET), Argentina.

- Ability to define what optimal resource selection means in regard to each actor (resource owners, job owners, global welfare of the grid, optimal occupation of the resources, etc.) and offer different solutions to different definitions of optimality.
- Ability to cope with the highly dynamic characteristics of grid computing offering flexibility, scalability and fault tolerance.
- Ability to interface with existing and foreseen grid solutions.
- Ability to incentivise resource owners to put their resources in the grid and jobs owners to submit the jobs to the grid rather then buying more resources on their own when an advantage is foreseen.
- Ability to cope with different scenarios ranging from pure cooperation to pure competition.
- Capability to be implemented in small steps, allowing the incremental composition of simple services into more complex ones.

In grid resource management, various approaches [28, 33, 26] have been used in several projects. Among these, the management model based on computation economy [18, 13] has been widely recognized because of its improvement of the efficiency of management and simplification of the complexity of resource scheduling.

The economy is an old, historical, sociological, and mature system influencing many areas of human life. Science describes and analyzes this system by the research areas of political economics and business economics. Economy is a driving factor which motivates technological developments out of others as human needs, both based on humans yearning.

Economic theory says that given a perfectly competitive market, if the supply and demand functions are homogeneous, continuous and obey Walras Law of Markets, then there exists a equilibrium price point for the entire market. In the area of computation economic models, lots of research has been investigated, which covers from economic models [15] and security [35, 39] to GridBank [4]. Nevertheless, few works are focused on the indirect reputation problem of grid resource management. To facilitate resource evaluation, the concept of indirect reputation is proposed in this paper. This idea reflects the essence of the resource, that is, whether the resource can be treated as a reliable object. This concept applied to the *Commodity Market Model* [16] is investigated in the rest of this paper.

The rest of this paper is organized as follows. The next section introduces the background and related work, summarizing the key ideas behind economic-based market models. Section 3 provides the concept of indirect reputation and its components. Section 4 describes the computation of the proposed model. Section 5 proposes an indirect reputation resource scheduling algorithm. Section 6 presents a simulation environment in order to analyze and validate the performance of the indirect reputation model compared with other approaches. Finally, Section 7 presents the conclusions and future work.

2 Background and Related Work

Market methods, sometimes called *market oriented programming* in combination with Computer Science, are used to solve the following problems which occur in real scheduling environments ([19]):

- The site autonomy problem arises as the resources within the system are owned by different companies.
- The heterogeneous substrate problem that results from the fact that different companies use different resource management systems.
- The policy extensibility problem means that local management systems can be changed without any effects for the rest of the system.
- The co-allocation problem addresses the aspect that some applications need several resources of different companies at the same time. Market methods allow the combination of resources from different suppliers without further knowledge of the underlying schedules.
- The online control problem is caused by the fact that the system works in an online environment.

The supply and demand mechanisms provide the possibility to optimize different objectives of the market participants under the usage of costs, prices and utility functions. It is expected that such methods provide high robustness and flexibility in the case of failures and a high adaptability during changes.

Next, the definitions of market, market method and agent will be presented briefly.

A *market* can be defined as a virtual market or from an economical point of view as follows: *"Generally any context in which the sale and purchase of goods and services takes place."* [41]. The minimal conditions to define a virtual market are: *"A market is a medium or context in which autonomous agents exchange goods under the guidance of price in order to* *maximize their own utility."* [41]. The main aspect is that autonomous agents exchange voluntarily their goods in order to maximize their own utility.

A *market method* can be defined as follows: *"A market method is the overall algorithmic structure within which a market mechanism or principle is embedded."* [41]. It has to be emphasized that a market method is an equilibrium protocol and not a complete algorithm.

The definition of an *agent* can be found in [41]: *"An agent is an entity whose supply and demand functions are equilibrated with those of others by the mechanism, and whose utility is increased through exchange at equilibrium ratios."*. Details about the general equilibrium and its existence can be found in [45].

The two key players driving the grid marketplace, like in the conventional marketplace, are: Grid Service Providers (GSPs) [9] providing the traditional role of producers and Grid Resource Brokers (GRBs) [27] representing consumers. The grid computing environments provide necessary infrastructure including security, information, transparent access to remote resources, and information services that enable us to bring these two entities together [14]. Consumers interact with their own brokers for managing and scheduling their computations on the grid. The GSPs make their resources Grid enabled by running software systems (such as Globus) along with Grid resource Trading Services/Servers (GTS) to enable resource trading and execution of consumer requests directed through GRBs. The interaction between GRBs and GSPs during resource trading is mediated through a Grid Market Directory (GMD) [14].

A number of approaches have been proposed using economic models to address resource usage and incentives in a grid [21, 42, 30, 11]. Particularly, a well designed market-based resource allocation mechanism provides incentives for participation by ensuring that all the actors in the system maximize their utility and do not have incentives to deviate from the designed protocol.

Economy-based resource management and scheduling in computational grids was proposed and evaluated in [12]. An important limitation to that work is the lack of consideration given to data while scheduling jobs on remote resources. [42] aimed to extend the deadline and budget constrained cost and time minimization algorithms proposed in [12] to data grids by removing that limitation. These two algorithms are compared with the proposed indirect reputation model in Section 6. Here, we briefly mention some other typical algorithms of economic models.

The WALRAS method is a classic approach by translating a complex, distributed problem into an equilibrium problem [8]. One of the assumptions is that agents do not try to manipulate the prices with speculation, which is called a *perfect competition*. To solve the equilibrium problem the WALRAS method uses a *double auction*. During that process all agents send their utility functions to a central auctioneer who calculates the equilibrium prices. A separate auction is started for every good. At the end, the resulting prices are transmitted to all agents. As the utility of goods may not be independent for the agents, they can react on the new equilibrium prices by re-adjusting their utility functions. Subsequently, the process starts again. This iteration is repeated until the equilibrium prices are stabilized.

The WALRAS method has been used for transportation problems as well as for processor rental. The transportation problem requires to transport different goods over an existing network from different start places to different end places. The processor rental problem consists of allocating one processor for different processes, while all processes have to pay for the utilization.

Another application example for market methods is the Enterprise [32] system. Here, machines create offers for jobs to be run on these machines. To this end, all jobs describe their necessary environment in detail. After all machines have created their offers the jobs select between these offers. The machine that provides the shortest response time has the highest priority and will be chosen by the job. All machines have a priority scheme where jobs with a shorter run time have a higher priority.

Reputation has been applied to a wide variety of systems but currently, the research of trust models are mainly focused on the area of electronic business (buyers and sellers reputation), most products, such as eBay [20], OnSale [34], Sporas and Histos [46], perform comprehensive evaluation according to the history data of the bargainers that are judged and formed by the satisfaction degree of users. The results of this approach are local to the bargainers and vary with different bargainers. This approach relies on a centralized system to store and manage trust ratings. Another example of this kind of model is the C-Net information portal, which maintains an editors ranking on products and resellers. However individual user responses are not integrated in a correction of the editors ranking.

In the area of grid resource management, researchers try to solve the trust problem [5, 38] in a similar way and have achieved some results [46, 6]. But those works basically migrate the evaluation methods from the area of electronic business to the area of grid or P2P computing.

In the next sections we present our scheduling method to add trust into algorithms for grid job scheduling.

3 Indirect Reputation Model

3.1 Definition of Trust and Reputation

Trust is a psychological state in our society. The capabilities of an individual (or organization) are so limited that we must depend on and cooperate with others in order to achieve various goals of our daily life and businesses. This interdependence on each other makes trust arise as one basic social glue unit, which enables us to collaborate with others without fear, and lets us use trust as a key element for successful conflict resolution.

The notion of trust and reputation are complex subjects related to a firm belief in attributes such as reliability, honesty, and competence of the trusted entity. There is a lack of consensus in the definition of them. The definition of trust and reputation that we will use in this paper is adopted from [22, 7].

Trust is the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subject to the entity's behavior and applies only within a specific context at a given time.

The direct reputation of an entity is an expectation of its behavior based on other entities' observations or information about the entity's past behavior at a given time.

The definition of indirect reputation that we will use in this paper is as follows: the indirect reputation of an entity is an expectation of its behavior based on a third party organization's observations or information about the entity's past behavior at a given time.

Like trust and direct reputation, the indirect reputation is a dynamic value and spans over a set of values ranging from very trustworthy state to very untrustworthy state. The indirect reputation is built on past experience by a third party organization's observations and given for a specific context. When making indirect reputation-based decisions, entities can rely on the third party organization for information pertaining to a specific entity. While direct reputation is a private value between any two entities and varies with different entities, indirect reputation is a unique and public value in the grid administration domain.

3.2 Components of the Indirect Reputation

The indirect reputation algorithm is built on the Resource Usage Record (RUR) [23, 4]. RUR is a platform-independent record (XML document) for resource accounting data[1]. The RUR was developed as a co-operative effort between the Global Grid Forum and the GridBus Project. It is considered a common grid middleware format that reflects the accounting principles involved in any kind of financial transaction. The record may contain information about CPU time, storage, or memory used (Table 1).

Resource Data	**User Data**
host name / IP address	host name / IP address
certificate name	certificate name
host type	
local job ID	
wall clock time + price	
user CPU time + price	**Job Data**
system CPU time + price	job ID
main memory + price	application name
secondary storage + price	job start date
I/O channels + price	job end date
total price	

Table 1. Resource Usage Record

Let x a resource identifier (resource certificate name) defined by RUR, then the resource indirect reputation $R_r(x)$ is defined as follows:

$$R_r(x) = \alpha \cdot R_{r,cur}(x) + \beta \cdot R_{r,new}(x)$$

$$R_{r,new}(x) = \chi \cdot \frac{\sum_i (b_{io} - b_i)}{\sum_i b_{io}} + \delta \cdot \frac{\sum_i (t_{io} - t_i)}{\sum_i t_{io}} \quad i = 1, 2, \ldots$$

In the equations above, $\alpha, \beta \geq 0$, $\alpha + \beta = 1$, $\chi, \delta \geq 0$, $\chi + \delta = 1$, $b_{io} \geq b_i$, $t_{io} \geq t_i$, $R_{r,cur}(x)$ means the current value of resource indirect reputation, that is, the result of latest evaluation. $R_{r,new}(x)$ is the result of current evaluation, which includes two factors: time and cost. And b_{io} is the budget value of each record and b_i is the actual cost. Similarly, t_{io} is the estimative time and t_i is the actual cost. χ and δ are the contribution rate of the cost and time to the indirect reputation respectively. Finally, $0 \leq R_r(x) \leq 1$, where $R_r(x) = 1$ means the best indirect reputation and $R_r(x) = 0$ means the worst one.

[1] XML allows RUR to be extended by each site to include site-specific information.

In the equation about $R_{r,new}$, the actual cost will surpass the budget if $b_{io} < b_i$ under certain resource usage. Similarly, the time expended will surpass the budget if $t_{io} < t_i$. Both of these two cases belong to the *breach of contract* class. After having broken a contract, the update method of resource indirect reputation is defined as follows:

$$R_r(x) = (1 - \varepsilon) \cdot R_{r,cur}(x) \qquad 0 \leq \varepsilon \leq 1$$

The indirect reputation is a function that varies with time. If at certain time period greater than the evaluation period, there is a resource x that doesn't produce the new Usage Record, then its indirect reputation function is as follows:

$$R_r(x) = (1 - \varphi) \cdot R_{r,cur}(x) \qquad 0 \leq \varphi \leq 1$$

Obviously, if a resource doesn't provide service for the grid, its indirect reputation decreases as time goes by.

4 Reputation Computation

4.1 Reputation Coefficient, Initial Value and Update

The coefficient value embodies the corresponding weight. To reflect the more important contribution of the new-arriving scheduling to the indirect reputation, the value of β should not be small. If $\chi > \delta$ then the proposed system concerns more about the scheduling cost, otherwise the system will give more relevance to the actual finish time of a resource.

The choice of initial indirect reputation value is closely related to the resource's behavior in the grid. If a resource finds its indirect reputation value is smaller than the initial value because of breach of contract, it may choose to register again after having logged out. Setting 0 as the initial indirect reputation value may solve this problem, but on the other hand, this may greatly depress the new-coming users and resources. The approach that we have adopted is to delete a record of breach of contract after having dealt with it, which has no effects on the next evaluation. Besides, the indirect reputation value will be unconditionally smaller than the initial indirect reputation value.

There are two approaches that can be used to update the value of the indirect reputation: the event-driven approach and the time-driven approach. Since the indirect reputation evaluation is not sensitive to the features of performance and real-time, we believe that the time-driven approach is more appropriate. The deleting strategy we have adopted is as follows: deleting its record if a resource has not logged in during a certain time period (two months, for example). Deleting the record of the resource provider means deleting all the records of the resources it currently owns.

4.2 Algorithm Implementation

Let τ be the latest evaluating time, $\Delta\tau$ be the inter-evaluating time, and $R_{ini}(r_i)$ be the initial value of the resource r_i, then the pseudo-code of the evaluation algorithm is as follows:

```
while (CurrentTime = τ + Δτ)
 collect RUR, price and cost data into
   REM database
 for all r_i do
  NewRep := IRE(r_i, τ + Δτ)
  if NewRep = -1 then
   R_r(r_i) := max[(1 − φ)R_r,cur(r_i), R_ini(r_i)]
   if NewRep ≥ 1 then
    for i:=1 to NewRep do
     R_r(r_i) := (1 − ε)R_r,cur(r_i)
     R_r(r_i) := max[R_r(r_i), R_ini(r_i)]
    endfor
   else R_r(r_i) := α · R_r,cur(r_i) + β · NewRep
 endfor
 τ := CurrentTime;
 R_r,cur := R_r(r_i)
 ReleaseResults
endwhile
```

The indirect reputation evaluation function is defined as follows:

```
function IRE(r_i, τ, τ + Δτ)
 count := 0
 S := RUR(r_i, τ + Δτ)
 if S = ∅ then
  if TimeofFree + Δτ ≥ TimeofDecrease
   then
    TimeofFree := 0
    return -1
  else for all records in S do
   if (b_io ≥ b_i) and (t_io ≥ t_i) then
   return χ · (Σ_i (b_io − b_i)) / (Σ_i b_io) + δ · (Σ_i (t_io − t_i)) / (Σ_i t_io)
 for all records in S do
  if (b_io ≤ b_i) or (t_io ≤ t_i) then
   count := count + 1
   return count
  endif
endfunction
```

Where `TimeofDecrease` is the period in which the indirect reputation value of certain resource decreases because this resource has not been scheduled yet.

`TimeofFree` is the total time counted during certain resource has not been scheduled, and `count` is the counter for the records of breach of contract.

5 Indirect Reputation-Based Resource Scheduling Algorithm

In order to satisfy the goal of minimizing the cost and time, we have designed an indirect reputation-based resource scheduling algorithm that can be expressed as follows:

1. Sort the resources that meet the task requirement according to the indirect reputation value firstly, then select m resources from the sorted list;
2. For the candidate list (m in length) obtained in step 1, sort it by increasing cost and then select n resources as the candidate resource set;
3. For each resource in candidate resource set (n in length), calculate the next completion time for an assigned job, taking into account previously assigned jobs.
4. Sort resources by next completion time.
5. Assign one job to the first resource for which the cost per job is less than or equal to the remaining budget per job.
6. Repeat all previous steps until all the jobs are assigned.

6 Analysis of Simulation Experiments

Before showing the experiments we will summarize the key idea behind the algorithms used for comparison [15]:

Cost minimization. Jobs are executed in the schedule that causes least expense while keeping the execution time within the provided deadline.

Time minimization. Jobs are executed in the fastest time possible with the budget for the execution acting as the constraint.

The algorithm that implements the scheduling of a set of jobs according to the above objective functions is depicted in [42]. This algorithm is based on the Min-Min heuristic discussed in [31].

To facilitate the comparison between our indirect reputation-based algorithm, time-minimization algorithm and cost-minimization algorithm [15, 42], we employ Nimrod/G 3.2.0 for GNU/Linux as the simulation experiment platform. Nimrod/G is a resource management system for scheduling computations on globally distributed resources with varying QoS [3, 14, 2].

The simulation environment is composed of 100 nodes and each node provides a CPU resource in either 2.0GHz or 2.5GHz (reasonable value for current technologies, e.g. Intel Core 2 Duo desktop processors). Besides, each node is not only a resource provider but also a task host. The agent residing in the node administers the resource and performs the model strategy. Every task in the system owns running time of 100s by estimate and a budget of 2800G$[2]. To acquire precise initial indirect reputation value, we adopt random tasks scheduling strategy and control the running time greater than 50 evaluation periods, assuring that the indirect reputation evaluation has been performed at least 50 times.

In the simulation experiment, we mainly focus on the following aspects: throughput per time-unit, turnover, rate of resource utilization and average finish time of task. The price of resources is classified by indirect reputation ranking and price of 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 G$/s corresponds to indirect reputation level (0, 0.1), [0.1, 0.2), . . ., [0.9,1) respectively. In the indirect reputation-based scheduling algorithm, parameters m and n are assigned as 5 and 3 respectively. Our experiment results are achieved during 4 periods (/hour), these results are depicted in Tables 2 to Table 5.

As shown in Table 2, compared with the cost-minimization algorithm and the time-minimization algorithm, using the indirect reputation-based algorithm, the system throughput has 5.5% enhancement and 1.8% enhancement respectively. In Table 3, time-minimization gains the maximal transaction volume and indirect reputation-based decreases a little, while the cost-minimization algorithm decreases greatly. In Table 4, the cost-minimization algorithm achieves the average cost of 2378G$, while the indirect reputation-based cost is 2482G$ and the time-minimization cost is 2545G$. As shown in Table 5, the indirect reputation-based is the best in terms of both resource utilization and rate of completed tasks, and its average finish time is close to the time-minimization algorithm (1.2 seconds behind).

According to the results presented, the indirect reputation-based algorithm is the best one in terms of throughput, rate of resource utilization and rate of completed tasks. While in the aspects of turnover, average task finish time and average task cost, the proposed indirect reputation-based is a tradeoff between the time-minimization one and the cost-minimization

[2] G$ is the grid currency [4].

one.

Algorithm name	1	2	3	4
Indirect Reputation	3241	3319	3261	3326
Cost Minimization	3106	3051	3140	3088
Time Minimization	3233	3281	3309	3257
Standard deviation	61.84	118.4	71.11	99.98

Table 2. *Scheduling times per time-unit*

Algo name	1	2	3	4
Indirect Rep	8101	8247	7929	8341
Cost Min	7239	7431	7143	7609
Time Min	8611	8361	8251	8491
Standard dev	566.23	414.16	465.37	385.32

Table 3. *Transaction volume in different periods (G$)*

Algorithm name	1	2	3	4
Indirect Reputation	2501	2486	2433	2509
Cost Minimization	2332	2437	2276	2465
Time Minimization	2526	2550	2495	2608
Standard deviation	86.2	46.3	92.2	59.8

Table 4. *Average task cost (G$)*

Algorithm name	Util(%)	Com(%)	Fin(s)
Indirect Reputation	87.4	98	83.6
Cost Minimization	84.3	92	92.5
Time Minimization	85.2	97	82.4
Standard deviation	1.30	2.62	4.50

Table 5. *Resource utilization, completed rate and average finish time*

7 Conclusions and Future Work

Due to the expected scale of the grid computing systems, we need to develop highly distributed and extensible resource management frameworks for such systems. The results of simulation experiments presented in this paper validate the efficiency of the indirect reputation evaluation model. It is feasible to establish an indirect reputation architecture for the resources and their service partners in the grid. This way, we could not only constrain the behavior of Grid Service Providers [9] and Grid Service Containers [40] but also enhance the Quality of Services of the grid.

In the near future, we plan to extend our indirect reputation architecture to multiple grid systems and take up more resources, such as credit for the GridBank [4], into our evaluation architecture. It is interesting to review reputation systems for P2P with the purpose of adopting its characteristics to grid computing [44] [25]. Also we want to integrate the evaluation mechanisms employed in real life into our evaluation model, in order to help the grid computation economics become more practical.

References

[1] J. H. Abawajy. Grid accounting and payment architecture. In M. H. Hamza, editor, *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN'04)*, pages 82–87, Innsbruck, Austria, Feb. 2004. IASTED/ACTA Press.

[2] D. Abramson, R. Buyya, and J. Giddy. A computational economy for grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computer Systems*, 18(8):1061–1074, Oct. 2002.

[3] D. Abramson, R. Giddy, and L. Kotler. High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid? In *Proceedings of the 14th International Conference on Parallel and Distributed Processing Symposium (IPDPS-00)*, pages 520–528, Los Alamitos, May 2000. IEEE.

[4] Alexander Barmouta and Rajkumar Buyya. GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 245.1, Washington, DC, USA, 2003. IEEE Computer Society.

[5] F. Azzedin and M. Maheswaran. Evolving and Managing Trust in Grid Computing Systems, June 2002.

[6] F. Azzedin and M. Maheswaran. Integrating Trust into Grid Resource Management Systems. In *Proceedings of the 2002 International Conference on Parallel Processing (31th ICPP'02)*, Vancouver, Canada, Aug. 2002. Univ. of Toronto.

[7] F. Azzedin and M. Maheswaran. Towards Trust-Aware Resource Management in Grid Computing Systems. In *CCGRID*, pages 452–457. IEEE Computer Society, 2002.

[8] N. R. Bogan. Economic allocation of computation time with computation markets. Technical Report MIT-LCS//MIT/LCS/TR-633, MIT, 1994.

[9] M. Bogdanski, M. Kosiedowski, C. Mazurek, and M. Wolniewicz. Grid service provider: How to improve flexibility of grid user interfaces? In P. M. A. Sloot, D. Abramson, A. V. Bogdanov, J. Dongarra, A. Y. Zomaya, and Y. E. Gorbachev, editors, *Computational Science - ICCS 2003, International Conference, Melbourne, Australia and St. Petersburg, Russia, June 2-4, 2003. Proceedings, Part I*, volume 2657 of *Lecture Notes in Computer Science*, pages 255–263. Springer, 2003.

[10] A. Bradley, K. Curran, and G. Parr. Resource discovery and management in computational Grid envi-

ronments. *International Journal of Communication Systems*, 19(6):639–657, 2006.

[11] J. Brunelle, P. Hurst, J. Huth, L. Kang, C. Ng, D. C. Parkes, M. Seltzer, J. Shank, and S. Youssef. EGG: An Extensible and Economics-Inspired Open Grid Computing Platform. In *Proceedings of the 3rd International Workshop on Grid Economics and Business Models (GECON 2006), Singapore*, 2006.

[12] R. Buyya. *Economic-based distributed resource management and scheduling for Grid computing.* Ph. D thesis, Monash University, Melbourne, Australia, Apr. 2002.

[13] R. Buyya, D. Abramson, and J. Giddy. Grid Resource Management, Scheduling and Computational Economy, Mar. 2000.

[14] R. Buyya, D. Abramson, and J. Giddy. Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid, 2000.

[15] R. Buyya, J. Giddy, and D. Abramson. An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications, July 2000.

[16] R. Buyya and S. Vazhkudai. Compute Power Market: Towards a Market-Oriented Grid. In *CCGRID*, pages 574–581. IEEE Computer Society, 2001.

[17] L. Chunlin and L. Layuan. Utility Based Multiple QoS Guaranteed Resource Scheduling Optimization in Grid Computing. In *ICCTA*, pages 165–169. IEEE Computer Society, 2007.

[18] K. Czajkowski, I. Foster, N. Karonis, S. Martin, W. Smith, and S. Tuecke. A Resource Management Architecture for Metacomputing Systems. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 62–82. Springer Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.

[19] K. Czajkowski, I. T. Foster, N. T. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. A Resource Management Architecture for Metacomputing Systems. In *IPPS/SPDP '98: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing*, pages 62–82, London, UK, 1998. Springer-Verlag.

[20] eBay. e-commerce. http://www.ebay.com.

[21] Ferguson, D. and Yemini, Y. and Nikolaou, C. . Microeconomic algorithms for load balancing in distributed computer systems. In *8th International Conference on Distributed Computing Systems*, pages 491–499, 1988.

[22] T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3(4), 2000.

[23] GridForum. OGSA Resource Usage Service Working Group (RUS-WG). https://forge.gridforum.org/projects/rus-wg.

[24] Javier Echaiz and Jorge Ardenghi. Extending an SSI Cluster for Resource Discovery in Grid Computing, 2006.

[25] R. Jurca and B. Faltings. Reputation-based pricing of p2p services. In *P2PECON '05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 144–149, New York, NY, USA, 2005. ACM.

[26] Y.-S. Kee, H. Casanova, and A. A. Chien. Realistic Modeling and Synthesis of Resources for Computational Grids. In *SC'2004 Conference CD*, Pittsburgh, PA, Nov. 2004. IEEE/ACM SIGARCH. UCSD.

[27] A. Kertesz and P. Kacsuk. A Taxonomy of Grid Resource Brokers. In *Distributed and Parallel Systems - Cluster and Grid Computing (Proceedings of the 6th Austrian-Hungarian Workshop on Distributed and Parallel Systems (DAPSYS)), EDITOR = Kacsuk, P., Fahringer, T., Nemeth, Zs.*, page 10, Innsbruck (Austria), September 2006.

[28] K. Krauter, R. Buyya, and M. Maheswaran. A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing. *Softw, Pract. Exper*, 32(2):135–164, 2002.

[29] S. L. Infrastructure for Science Portals. *IEEE Internet Computing*, pages 71–73, 2000.

[30] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman. Tycoon: An implementation of a distributed, market-based resource allocation system. *Multiagent Grid Systems*, 1(3):169–182, 2005.

[31] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund. Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems. *Journal of Parallel and Distributed Computing (JPDC)*, 59(2):107–131, Nov. 1999.

[32] T. Malone, R. Fikes, K. Grant, and M. Howard. *Enterprise: A market-like task scheduler for distributed computing environments*, pages 177–205. North-Holland, 1988.

[33] J. Nabrzyski, J. M. Schopf, and J. Weglarz, editors. *Grid Resource Management: state of the art and future trends.* Number ISOR 64 in International series in operations research & management science. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands, 2003.

[34] OnSale. e-commerce. http://www.onsale.com.

[35] R. Saadi, J.-M. Pierson, and L. Brunie. The Chameleon: A Pervasive Grid Security Architecture. In *ICNS'07 (International Conference of Networking and Services), Athens, Greece, 19/06/2007-25/06/2007*, http://www.computer.org, June 2007. IEEE Computer Society.

[36] Security Working Group. Enterprise Grid Security Requirements. Technical Report Version 1.0, Enterprise Grid Alliance, July 2005.

[37] F. Siebenlist. Grid security: requirements, plans and ongoing efforts. In *Proceedings of the 2003 ACM workshop on XML security (XMLSEC-03)*, pages 38–38, New York, Oct. 2003. ACM Press.

[38] M. P. Singh. Trustworthy Service Composition: Challenges and Research Questions. In R. Falcone, K. S. Barber, L. Korba, and M. P. Singh, editors, *Trust,*

Reputation, and Security, volume 2631 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2002.

[39] S. Song, K. Hwang, and Y.-K. Kwok. Trusted Grid Computing with Security Binding and Trust Integration. *Journal of Grid Computing*, 3(1-2):53–73, 2005.

[40] H. Sun, W. Liu, T. Wo, and C. Hu. CROWN Node Server: An Enhanced Grid Service Container Based on GT4 WSRF Core. In *GCC Workshops*, pages 510–517. IEEE Computer Society, 2006.

[41] P. Tucker and F. Berman. On market mechanisms as a software technique, 1996.

[42] S. Venugopal and R. Buyya. An economy-based algorithm for scheduling data-intensive applications on global grids, 2004.

[43] C. Weng, M. Li, and X. Lu. An Online Scheduling Algorithm for Assigning Jobs in the Computational Grid. *IEICE Transactions on Information and Systems*, e89-d(2):597, Feb. 2006.

[44] K. Wongrujira and A. Seneviratne. Monetary incentive with reputation for virtual market-place based p2p. In *CoNEXT '05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 135–145, New York, NY, USA, 2005. ACM.

[45] F. Ygge. *Market-Oriented Programming and its Application to Power Load Management*. Ph. D thesis, Lund University, Sweden, 1998.

[46] G. Zacharia and P. Maes. Trust Management through Reputation Mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

[47] C. Zhu, Z. Liu, W. M. Zhang, W. Xiao, Z. ning Xu, and D. Yang. Decentralized Grid Resource Discovery Based on Resource Information Community. *Journal of Grid Computing*, 2(3):261–277, 2004.