Economically Enhanced Resource Management for Internet Service Utilities

Tim Püschel[1 2], Nikolay Borissov[2], Mario Macías[1], Dirk Neumann[2], Jordi Guitart[1], and Jordi Torres[1]

[1] Barcelona Supercomputing Center - Technical University of Catalonia(UPC),
c/ Jordi Girona 29, 08034 Barcelona, Spain
{tim.pueschel,mario.macias,jordi.guitart,jordi.torres}@bsc.es
[2] Institute of Information Systems and Management (IISM)
Universität Karlsruhe (TH), Englerstr. 14, 76131 Karlsruhe, Germany
{pueschel,borissov,neumann}@iism.uni-karlsruhe.de

Abstract. As competition on global markets increases the vision of utility computing gains more and more interest. To attract more providers it is crucial to improve the performance in commercialization of resources. This makes it necessary to not only base components on technical aspects, but also to include economical aspects in their design. This work presents an framework for an Economically Enhanced Resource Manager (EERM) which features enhancements to technical resource management like dynamic pricing and client classification. The introduced approach is evaluated considering various economic design criteria and example scenarios. Our preliminary results, e.g. an increase in achieved revenue from 77% to 92% of the theoretic maximum in our first scenario, show that our approach is very promising.

1 Introduction

Many web applications have strongly varying demand for computing resources. To fulfil these requirements sufficient resources have to be made available. In some cases it is possible to run certain tasks at night to achieve a more even usage, however in many cases it is not feasible to let users wait a long time for results. This leads to a situation in which utilization is very high during certain peak times while many resources lay idle during other times. Due to high competition on global markets many enterprises face the challenge to make use of new applications and reduce process times on one side and cut the costs of their IT-infrastructures on the other side [5].

In light of this challenge the idea of utility computing gained interest. Utility computing describes a scenario where computer resources can be accessed dynamically in analogy to electricity and water [19]. The more resource providers offer their resources or services, the more likely it is they can be accessed at competitive prices. Therefore it is important to attract more providers.

However providers will only offer their services if they can realize sufficient benefit. With state-of-the-art technology, this assimilation is hampered, as the local resource managers facilitating the deployment of the resources are not designed to incorporate economic issues (e.g. price).

In recent times, several research projects have started to develop price-based resource management components supporting the idea of utility computing. Those approaches are devoted to scheduling by utilizing the price mechanism. Clearly, this means that technical issues such as resource utilization are ignored for scheduling. In addition, resource management is much more comprehensive than just scheduling. For example Service-Level-Agreement (SLA) management is also part of resource management that is often omitted in economic approaches. This plays a role when deciding which already ongoing jobs to cancel in overload situations to maintain system stability.

To improve performance in the commercialization of distributed computational resources decisions about the supplied resources and their management should be based on both technical and economic aspects [12]. Hence, this paper is an interdisciplinary work taking into account aspects from computer science and economics. We will explore the use of economic enhancements such as client classification and dynamic pricing to resource management.

Technical resource management systems typically offer the possibility to include priorities for user groups. In purely price-based schedulers it is not possible to distinguish important from unimportant partners, as only current price matters for the allocation. We will motivate that client classification should be integrated into economically enhanced resource management systems. Essentially, there are two main reasons to do so: First, client classification allows the inclusion of long-term oriented relationships with strategically important customers so-called credential components. Second, client classification can be used as an instrument of revenue management, which allows skimming off consumer surplus. The main contribution of this paper is to show how technical parameters can be combined into an economically enhanced resources management that increases revenue for the local resource sites.

2 Motivational Scenarios

2.1 Organisation Selling Spare Resource Capacity

An organisation offers different web services for internal use. However the utilization of the resources is uneven. During certain times there is only a low load, the resources are almost idle, and at other times users have to wait for their results very long. Therefore the organisation decides to buy more capacity on the market when needed and sell its spare capacity when the load is low. When accepting jobs users from within the organisation should always be preferred. To provide and communicate a good dynamic evaluation of resources, market mechanisms are also used for internal users. This also gives internal users an incentive to run their jobs during times of low utilization. Internal users also receive a significant discount compared to external users.

2.2 Resource/Service Provider with Preferred Customers

A big service provider maintains a data center whose resources are sold. The service provider already has a number of clients but still has a high spare capacity. Therefore he joins a marketplace to find new clients and optimize capacity utilization. To maintain the good relations with the current clients and encourage regular use of its services the provider offers a preferred client contract. Preferred clients receive a discount on the reservation price and soft preference when accepting jobs. Soft preference means that their bids are increased by a certain amount for the winner determination. When offering the same price the preferred client is chosen over the external client, but the standard client has the chance to outbid the preferred client. Prices should be calculated dynamically based on utilization, client classification, estimated demand and further pricing policies of the provider.

3 Economically Enhanced Resource Management

To improve performance in the commercialization of distributed computational resources and increase the benefit of service providers we propose the introduction of various economic enhancements into resource management. This chapter first describes the objectives and requirements for the enhancements, then follows a description of the key mechanisms that are to be integrated in the EERM. The chapter ends with a description of the architecture and the components of the EERM.

3.1 Objectives and Requirements

The main goals of these enhancements are to link technical and economical aspects of resource management and strengthen the economic feasibility. This can be achieved by establishing more precise price calculations for resources, taking usage of the resources, performance estimations and business policies into account. The introduced mechanisms should be be able to deal efficiently with the motivational scenarios given earlier. This means they have to feature *client classification*, different types of *priorities* for jobs from certain clients, *reservation* of a certain amount *of resources* for important clients, and *dynamic calculation of prices* based on various factors. In addition to these requirements the system should also offer quality of service (QoS) and be able to deal with situations in which parts of the resources fail. To adapt to different scenarios and business policies of different situations it should be highly flexible and configurable via policies.

When designing mechanisms various economic design criteria [4], [20] should be considered. These following criteria apply to the respective features as well as the overall system and the market mechanisms it is embedded in.

Individual Rationality. An important requirement for a system is that it is individual rational on both sides, i.e. both providers and clients have to have a benefit from using the system.

Simplicity and Computational Costs. While the enhancements introduce some additional factors and they should not introduce any unnecessary complexity. Similarly client classification, quality of service and dynamic pricing add some additional computational complexity, however they should not add any intractable problems and its benefits should outweigh its costs.

Revenue Maximization. A key characteristic for resource providers is revenue maximization or more general utility maximization.

Incentive Compatibility. Strategic behaviour of clients and providers can be prevented if a mechanism is incentive compatible. Incentive compatibility means that no other strategy results in a higher utility than reporting the true valuation.

Efficiency. There are different types of efficiency. The first one considered here is pareto optimimality. An allocation is considered pareto optimal if no participant can improve its utility without reducing the utility of another participant. The second efficiency criterion is allocative efficiency. A mechanism is called allocative efficient if it maximizes the sum of individual utilities.

3.2 Key Features

The motivational scenarios and the further requirements lead to four key features the of the EERM presented in this work.

Quality of Service. The first feature is quality of service. This can be broken down into two aspects. The first aspect is to assure adequate performance during normal operation of the resources. Overload situations can lead to reduced overall performance [17] and thereby can result in breaking QoS agreements between the provider and clients. Thus it is necessary to have a mechanism that ensures that jobs will not be accepted if they result in an overload situation. The second aspect of quality of service regards situations in which parts of the resources fail. To be able to fulfill all SLAs even in situations of partial resource failure it would be necessary to keep an adequate buffer of free resources. Where this is not feasible there should at least be a mechanism that ensures that those SLAs that can be kept with the available resources are fulfilled. This can be done by suspending or cancelling those jobs that can not be finished in time due to the reduced availability of resources.

Job Cancellation. Related to QoS is the feature automatic job suspension and cancellation. It is needed to ensure quality of service in situations where problems arise, i.e. parts of the resources fail or the estimations of the utilization were to optimistic. Cancellation of lesser important jobs to free capacity for incoming jobs with higher importance, i.e. jobs from a client with a higher classification or a jobs that deliver significantly more revenue is also possible.

Dynamic Pricing. Another enhancement is dynamic pricing based on various factors. [21] shows an approach for a pricing function depending on a base pricing rate and a utilization pricing rate. However the price can depend not only on current utilization but also on other factors such as projected utilization, client classification, and projected demand. Pricing should also be contingent on the demand on the market. This feature can be either implemented in the EERM or in a dedicated component responsible for trading. This agent would request a price based on factors including utilization of the resources and client classification. from the EERM and then calculate a new price taking the situation on the market into account.

Client Classification. Earlier giving clients different privileges was mentioned, e.g. by discriminating on factors like price and quality of service. This part describes the factors that can be used to differentiate various client classes.

Price Discrimination. Price discrimination or customer-dependent pricing is one way to differentiate between different classes of clients. One idea to achieve this is introducing Grid miles [16] in analogy to frequent flyer miles. Clients could be offered a certain amount of free usage of the resources or a 10% discount after spending a certain amount of money.

Reservation of Resources. For certain users it may be very important to always have access to the resources. This class of users could be offered a reservation of a certain amount of resources. One option is to reserve a fixed share of resources for a certain class of users another possibility is to vary this share depending on the usage of the system.

Priority on Job Acceptance. Another option is to use a client priority on job acceptance. When the utilization of the system is low jobs from all classes of clients are accepted but when the utilization of the resources rises and there is competition between the clients for the resources, jobs from certain clients are preferred. There are two types of priorities: strict priorities and soft priorities.

- *Strict priority* means that if a job from a standard client and a client with priority compete for acceptance, the job from the client with priority always wins. Jobs from clients with priority are always preferred, thus there is no real competition between the different classes of clients.
- *Soft priority* means jobs from clients with priority are generally preferred but standard clients have the chance to outbid clients with priority. Thus soft priority is essentially a discount on the reservation price or bid that may only apply in certain situation, i.e. when utilization exceeds a certain threshold.

Quality of Service. Another factor where differentiation for classes of clients is possible is quality of service. For some classes of clients quality of service is offered, for others not. Offering different levels of quality of service for different classes of clients is also possible. An example for this would be offering different risk levels [7].

4 Architecture of the EERM

This part describes the architecture of the EERM that was designed based on the requirements and key features. It includes a description of the components with the aid of sequence diagrams. An overview of the architecture of the EERM can be seen in Fig. 1.

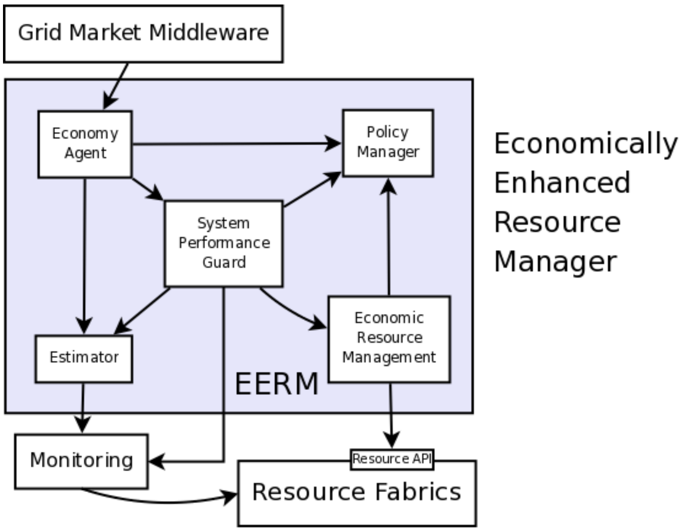

Fig. 1. EERM Architecture

The EERM interacts with various other components, namely a Grid Market Middleware, a Monitoring component and the Resource Fabrics. The Grid Market Middleware represents the middleware responsible for querying prices and offering the services on the Grid market. The Monitoring is responsible for monitoring the state and the performance of the resources and notifying the System Performance Guard in case of problems. Additionally data collected by the Monitoring is used by the Estimator component to for its predictions. Resource Fabrics refers to Grid Middlewares such as Condor [15] or Globus [9].

Economy Agent. The Economy Agent is responsible for deciding whether incoming jobs are accepted and for calculating their prices. These calculations can be used both for negotiation as well as bids or reservation prices in auctions. Figure 2 shows the sequence of a price request. First the Economy Agent receives a request from a market agent. Then it checks whether the job is technically and economically feasible and calculates a price for the job based on client category, resource status, economic policies and predictions of future job executions from the estimator component.

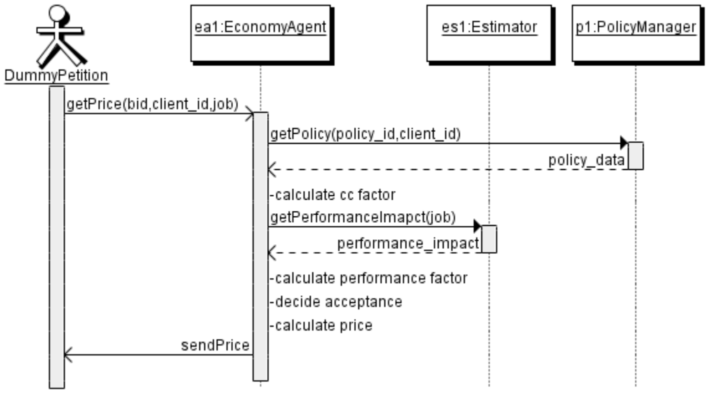

Fig. 2. Sequence Diagram Price Request

Estimator. The Estimator component calculates the expected impact on the utilization of the resources. This is important to prevent reduced performance due to overload [17], furthermore the performance impact can be used for the calculation of prices. This component is based on work by Kounev et al. using online performance models [13],[14].

System Performance Guard. The System Performance Guard is responsible for ensuring that the accepted SLAs can be kept. In case of performance problems with the resources it is notified by Monitoring. After checking the corresponding policies it determines if there is a danger that SLAs cannot be fulfilled. It then takes the decision to suspend or cancel jobs to ensure the fulfilment of the other SLAs and maximize overall revenue. Figure 3 shows a typical sequence for such a scenario. Jobs can also be cancelled when capacity is required to fulfil commitments to preferred clients.

Policy Manager. To keep the EERM adaptable the Policy manager stores and manages policies concerning client classification, job cancellation or suspension, etc. Policies are formulated in the Semantic Web Rule Language (SWRL) [11]. All features of the EERM require the respective components to be able to communicate with the Policy Manager and base their decisions on the corresponding policies. A simple example from a pricing policy in SWRL is the following rule which expresses that if the utilization is between 71% and 100% there is a surcharge of 50:

$$Utilization(?utilization) \wedge InsideUtilizationRange(?utilization, "71\% - 100\%")$$

$$\Rightarrow SetSurcharge(?utilizationsurcharge, "50")$$

Economic Resource Management. The Economic Resource Management is responsible for the communication with the local resource managers and influences the local resource management to achieve a more efficient global resource use.

5 Evaluation

The evaluation of this proposal consists of three parts. First an example scenario is described, then the results are presented and last the economic design criteria [4],[20] are considered.

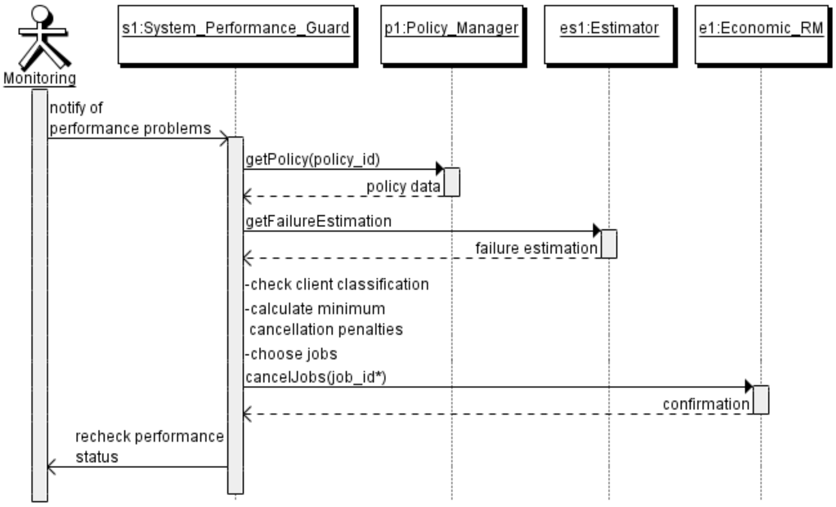

Fig. 3. Sequence Diagram System Performance Guard

5.1 Example Scenario

The first part of the evaluation considers an example scenario. For this evaluation there were no actual jobs executed. The policies were manually applied and revenue, utilization and utilities calculated. The job information given in Table 1 and the following assumptions were used.

The total capacity is 100 and the jobs given in Table 1 will be available during the run. The system receives information about jobs one timeslot before they become available. The Gold-client only uses the provider when he is assured access to a certain capacity. In this case he is guaranteed a total capacity of 60 units, while being able to launch new jobs of up to 30 units per period.

Using this scenario as a basis four policies are evaluated. The following policies represent the some of the basic options - no enhancements, client classification with fixed reservation, dynamic pricing with reservation prices, and client classification with strict priority:

- Case I is an example without any EERM. In this case any job is accepted if there is enough capacity left to fulfil it.
- Case II features a simple form of client classification, there is a fixed reservation of 60% of the capacity for the Gold-client. There is no System Performance Guard.
- Case III features the EERM with utilization-based pricing and the System Performance Guard, but without client classification. The policy is to only accept jobs that offer a price higher than 1 currency unit per capacity unit and time slot if the job results in utilization over 80%. The unit price is determined by dividing the price by the total capacity used by a job over all periods.
- In case IV the EERM with client classification and the system performance guard is used. The policy is to accept only jobs from the Gold-client if the job would result in utilization higher than 70%. This policy which can be applied to the first motivational scenario when replacing "Gold-client" with "Internal".

To evaluate the ability of the EERM to adapt to problems with the resources there is also a second scenario. It includes a reduced capacity of 70 in t=7 and a capacity of 60 in t=8 due to

Table 1. Example scenario

Job	Start Time	End Time	Capacity/t	Client Class	Price	Penalty
A	1	3	55	Standard	330	-82.5
B	1	5	24	Standard	180	-30
C	1	7	20	Standard	140	-17.5
D	2	4	20	Standard	120	-30
E	3	5	15	Standard	90	-22.5
F	3	8	20	Standard	120	-15
G	4	7	20	Standard	160	-40
H	4	9	15	Standard	135	-22.5
I	5	10	30	Standard	180	-22.5
K	5	8	30	Standard	240	-60
L	6	8	30	Standard	90	-11.25
M	6	9	12.5	Standard	50	-6.25
O	8	10	20	Standard	90	-15
P	9	10	21	Standard	84	-21
Q	9	10	30	Standard	90	-15
R	2	6	30	Gold	375	-187.5
S	5	8	30	Gold	300	-150
T	7	10	7.5	Gold	75	-37.5
U	7	9	20	Gold	150	-75
V	9	10	20	Gold	100	-50

unpredicted failure of resources. In the cases without the System Performance Guard it is assumed that the SLAs of jobs that end in periods that are overloaded due to the reduced capacity fail and that jobs that continue to run for more time can catch up and fulfil their SLAs. To assess the benefits of the EERM for providers and clients we compare their utility without and with EERM. Utility functions depend on the preferences of providers and clients. For the provider revenue is obviously a key criteria. Another factor that has to be considered is utilization. To allow for maintenance and reduce the effects of partial resource failure it is not in the interest of the provider to have a utilization that approaches 100%. In this example an average utilization of 80% is considered optimal. The following utility function for the provider considers the revenue per utilization while taking into account the optimal average utilization of the resources:

$$u_{provider} = \frac{revenue}{n * averageutilization} * \frac{1}{1 + |0.8 - \frac{averageutilization}{100}|}$$

For the client utility the idea is to weight the capacity needed by a job with its importance. The importance of a job for a client is expressed in the price per capacity unit the client is willing to pay for it. This results in the following client utility function, where P is the set of prices per unit (i.e. 2.5, 2, 1.5, 1):

$$u_{client} = \sum_{l \in P} unitprice_l * \frac{allocatedjobcapacity_l}{totaljobcapacity_l}$$

5.2 Results

Table 2 shows the results of applying these 4 cases to the scenario given in Table 1. A* is the allocation with maximum revenue that could be achieved if all job information was available before the first period. The table includes information about the completed jobs, the realized revenue, the average utilization of the providers resources, the provider utility, the client utility, and the proportion of the revenue in comparison with the A*. Cases II, III and IV perform significantly better than the standard case I without any economical enhancements. Although the average utilization is lower than in case I the yielded revenue as well the provider's and the client's utility is higher.

Table 2. Result of the four cases

Case	Completed	Revenue	Avg Load	$u_{provider}$	u_{client}	Revenue Proportion
I	A, B, C, G, H, L, M, O , P, Q	1349	89.7	1.37	3.10	76.65%
II	C, D, M, O, R, S, T, U, V	1400	71	1.81	3.33	79.55%
III	A, B, D, G, H, K, M, O, P, Q	1479	84.7	1.67	3.25	84.03%
IV	A, G, H, R, S, T, U, V	1625	73.5	2.08	3.87	92.33%
A*	A, E, G, O, Q, R, S, T, U, V	1760	81	2.15	3.53	100%

Table 3 shows the results of applying the policies to the scenario with failure of parts of the resources in t=7 and t=8. It can be seen that the improvement in cases III and IV is even more significant. In this situation case II is worse than case I regarding revenue, provider utility and client utility. This is caused by the lack of the System Performance Guard and thereby higher cancellation penalties of the jobs of the Gold-client. Case III delivers a significantly better utility for providers and clients. However since it doesn't include client classification it does not address the needs of the Gold-client. Case IV again delivers the best results regarding revenue, provider utility and client utility. Figure 4 shows the revenue proportion in the standard case (a) and with partial resource failure (b). In both cases III and IV the benefit of the System Performance Guard can be clearly seen, they deliver of 83.61% and 84.79% compared to the 56.56% of the standard case I.

Table 3. Result with partial resource failure

Case	Completed	Failed	Revenue	Avg Load	$u_{provider}$	u_{client}	Revenue Proportion
I	A, B, H, L, M, P, Q	C, G	901.5	79.7	1.13	2.28	56.56%
II	D, P, R, M, T, U, V	C, S	786.5	66.2	1.04	2.23	49.34%
III	A, B, D, G, H, K, P, Q	M	1332.75	74.95	1.69	3.05	83.61%
IV	A, P, R, S, T, U, V	G, H	1351.5	71.2	1.74	3.31	84.79%
A*	A, E, P, Q, R, S, T, U, V		1594	71.2	2.06	3.11	100%

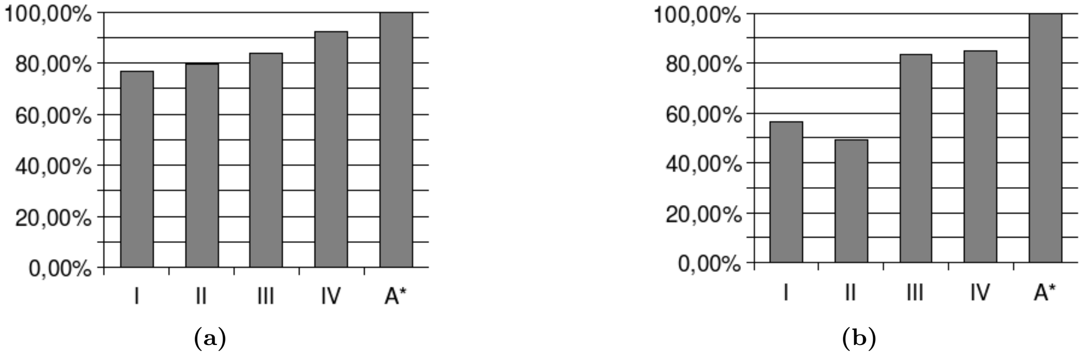

Fig. 4. Proportion of maximum revenue in the four cases

5.3 Economic Design Criteria

In cases III and IV of the example both sides have a clear benefit from using the respective mechanisms, hence *individual rationality* is achieved in these cases. The proposed mechanisms introduce some additional *complexity* but this is encapsulated in the EERM. They, however, do not introduce any NP-hard problems into the mechanism and the additional *computational cost* is limited. Another key characteristic for resource providers is *revenue maximization.* In the given example the policy of case IV delivers the highest revenue, provider utility and client utility among the options.

Whether the criterion of *incentive compatibility* is fulfilled depends on the market and classification mechanisms the EERM is embedded in. For the example a simple pay as you bid scheme is assumed. As this means that clients set their bids lower than their evaluation to achieve a benefit incentive compatibility is not given in the example. If, however, embedded in an allocation and pricing mechanism that is incentive compatible the EERM serves to give more precise valuations for the jobs.

Efficiency also depends on the mechanisms the EERM is embedded in. For the example we used a scenario where the provider decides whether jobs are accepted. Obviously the provider is mainly concerned with maximizing its own utility. Therefore it does not guarantee the maximum total utility. The EERM can also be embedded in efficient market mechanisms serving to give a more precise valuations of the jobs.

6 Related work

There is related work covering different aspects of our work. [8] discusses the application of economic theories to resource management. [1] presents an architecture for autonomic self-optimization based on business objectives. Elements of client classification such as price discrimination based on customer characteristics have been mentioned in other papers [16], [3]. They did however not consider other discrimination factors. [6] describes data-mining algorithms and tools for client classification in the electricity grids but concentrate on methods for finding groups of customers with similar behaviour. An architecture for admission control on e-commerce websites that prioritizes user sessions based on predictions about the user's intentions to buy a product is proposed in [18]. [2] presents research on how workload class importance should be considered for low-level resource allocation.

One approach to realize end-to-end quality of service is the Globus Architecture for Reservation and Allocation (GARA) [10]. This approach uses advance reservations to achieve QoS. Another way to achieve autonomic QoS aware resource management is based on online performance models [13], [14]. They introduce a framework for designing resource managers that are able to predict the impact of a job in the performance and adapt the resource allocation in such a way that SLAs can be fulfilled. Both approaches do not consider achieving QoS in case of partial resource failure. Our work makes use of the second approach and adds the mechanism of Job Cancellation.

The introduction of risk management to the Grid [7] permits a more dynamic approach to the usage of SLAs. It allows modelling the risk that the SLA cannot be fulfilled within the service level agreement. A provider can then offer SLAs with different risk profiles. However such risk modelling can be very complex. It requires information about the causes of the failure and its respective probabilities. Clients need to have the possibility to validate the accuracy and correctness of the providers risk assessment and risks have to be modelled in the SLAs.

7 Conclusion and Future Work

In this work various economical enhancements for resource management were motivated and explained. We presented a mechanism for assuring Quality of Service and dealing with partial resource failure without introducing the complexity of risk modelling. Flexible Dynamic Pricing and Client Classification was introduced and it was shown how these mechanisms can benefit service providers. Various factors and technical parameters for these enhancements were presented and explained.

Furthermore the preliminary architecture for an Economically Enhanced Resource Manager integrating these enhancements was introduced. Due to the general architecture and the use of policies and a policy manager this approach can be adapted to a wide range of situations.

The approach was evaluated considering economic design criteria and using an example scenario. The evaluation shows that the proposed economic enhancements enable the provider to increase his benefit. In the standard scenario we managed to achieve a 92% of the maximum theoretically attainable revenue with the enhancements in contrast to 77% without enhancements. In the scenario with partial resource failure the revenue was increased from 57% to 85% of the theoretical maximum.

The next steps will include refinement of the architecture as well as the implementation of the EERM. During this process further evaluation of the system will be done, e.g. by testing the system and running simulations. Another issue that requires further consideration is the generation of business policies for the EERM. Further future work in this area includes the improvement of price calculations with historical data and demand forecasting; the design of mechanisms for using collected data to determine which offered services deliver the most revenue and concentrate on them; the introduction of a component for automatic evaluation and improvement of client classification.

8 Acknowledgments

This work is supported by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contract TIN2004-07739-C02-01 and Commission of the European Communities under IST contract 034286 (SORMA).

References

1. Sarel Aiber, Dagan Gilat, Ariel Landau, Natalia Razinkov, Aviad Sela, and Segev Wasserkrug. Autonomic self-optimization according to business objectives. In *ICAC '04: Proceedings of the First International Conference on Autonomic Computing (ICAC'04)*, pages 206–213, 2004.
2. Harley Boughton, Pat Martin, Wendy Powley, and Randy Horman. Workload class importance policy in autonomic database management systems. In *POLICY '06: Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 13–22, 2006.
3. Rajkumar Buyya. *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. PhD thesis, Monash University, 2002.
4. Donald E. Campbell. *Resource Allocation Mechanisms*. Cambridge University Press, London, 1987.
5. Nicholas G. Carr. The end of corporate computing. *MIT Sloan Management Review*, 46(3):32–42, 2005.
6. Gianfranco Chicco, Roberto Napoli, and Federico Piglione. Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions on Power Systems*, 21(2):933– 940, 2006.
7. Karim Djemame, Iain Gourlay, James Padgett, Georg Birkenheuer, Matthias Hovestadt, Odej Kao, and Kerstin Voß. Introducing risk management into the grid. In *The 2nd IEEE International Conference on e-Science and Grid Computing (eScience2006)*, page 28, Amsterdam, Netherlands, 2006.
8. Donald F. Ferguson, Christos Nikolaou, Jakka Sairamesh, and Yechiam Yemini. Economic models for allocating resources in computer systems. pages 156–183, 1996.
9. Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, 1997.
10. Ian Foster, Carl Kesselman, Craig Lee, Bob Lindell, Klara Nahrstedt, and Alain Roy. A distributed resource management architecture that supports advance reservations and co-allocation. In *Proceedings of the 7th International Workshop on Quality of Service (IWQoS 1999)*, pages 62–80, London, UK, 1999.
11. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. Technical report, W3C Member submission, 2004.
12. Chris Kenyon and Giorgos Cheliotis. Grid resource commercialization: economic engineering and delivery scenarios. *Grid resource management: state of the art and future trends*, pages 465–478, 2004.
13. Samuel Kounev, Ramon Nou, and Jordi Torres. Autonomic qos-aware resource management in grid computing using online performance models. In *The 2nd International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2007)*, Nantes, France, 2007.
14. Samuel Kounev, Ramon Nou, and Jordi Torres. Building online performance models of grid middleware with fine-grained load-balancing: A globus toolkit case study. In *The 4th European Engineering Performance Workshop (EPEW 2007)*, Berlin, Germany, 2007.
15. Michael J. Litzkow, Miron Livny, and Matthew W. Mutka. Condor - A hunter of idle workstations. In *Proceedings of the 8th International Conference of Distributed Computing Systems*, 1988.
16. Steven Newhouse, Jon MacLaren, and Katarzyna Keahey. Trading grid services within the uk e-science grid. *Grid resource management: state of the art and future trends*, pages 479–490, 2004.
17. Ramon Nou, Ferran Julià, and Jordi Torres. Should the grid middleware look to self-managing capabilities? In *The 8th International Symposium on Autonomous Decentralized Systems (ISADS 2007)*, pages 113–122, Sedona, Arizona, USA, 2007.
18. Nicolás Poggi, Toni Moreno, Josep LLuis Berral, Ricard Gavaldà, and Jordi Torres. Web customer modeling for automated session prioritization on high traffic sites. In *Proceedings of the 11th International Conference on User Modeling*, Corfu, Greece, 2007.
19. Michael A. Rappa. The utility business model and the future of computing services. *IBM Systems Journal*, 43(1):32–42, 2004.
20. Peter R. Wurman. *Market structure and multidimensional auction design for computational economies*. PhD thesis, University of Michigan, 1999. Chair-Michael P. Wellman.
21. Chee Shin Yeo and Rajkumar Buyya. Pricing for Utility-driven Resource Management and Allocation in Clusters. In *Proceedings of the 12th International Conference on Advanced Computing and Communications (ADCOM 2004)*, pages 32–41, Ahmedabad, India, 2004.