

Imprecise Computation Grid Application Model for Flexible Market-based Resource Allocation

Kyong Hoon Kim, Rajkumar Buyya
Grid Computing and Distributed Systems Lab
Dept. of Computer Science & Software Eng.
University of Melbourne, Australia
E-mail: {jysh, raj}@csse.unimelb.edu.au

Jong Kim
Dept. of Computer Science and Engineering
Pohang University of Science and Technology
San 31, Hyoja-dong, Pohang, Korea
E-mail: jkim@postech.ac.kr

Abstract

Market-based resource management is becoming an emerging issue as the utilization of Grid computing is growing rapidly, particularly in the business field. In this paper, we provide a new Imprecise Computation Grid application model for flexible market-based resource management. Each job in a Grid application has two parts: mandatory part for the minimum quality and optional part for additional computations. This application model can be applied to QoS-related Grid applications and used in adaptive resource management. We also provide scheduling algorithms for resource allocation of IC Grid applications. Simulation results show that better utility is achieved when users specify both mandatory and optional requirements.

1. Introduction

The concept of Grid computing has gained popularity from scientific computations over geographically distributed systems [1]. Most studies in Grid computing have been conducted on issues, such as resource allocation, resource management, security, and Web services. In recent years, market-based resource management approaches have been proposed [2], as Grid technologies have been implemented in various fields, not only in scientific computations but also in businesses. In market-based Grid environments, users allocate *budgets* (also called *bids*, *values*, and *utilities*) for their jobs. Thus, resource providers offer their services depending on users' valuation of their jobs and its current commitments.

Several market-based resource management approaches have been proposed. Chun and Culler [3] propose a heuristic priority assignment scheme based on task value per unit running time. Irwin, et al. [4] introduce a discount rate for future reward and propose a scheduling heuristics called

FirstReward to balance the risk of future costs against rewards of a task. QoPS [5] uses an admission control to guarantee the deadlines of previously accepted jobs.

LibraSLA [6] is proposed based on a proportional share allocation technique. It extends Libra [7] by enhancing pricing and penalty functions. Lai, et al. [8] design and implement a distributed market-based resource allocation system called Tycoon, which is also based on proportional share. In another recent work [9], several profit-based scheduling and admission control algorithms that consider load, user impatience, number of resources, price, and resource uncertainty are evaluated.

One important issue in market-based Grid resource management is to meet the deadlines of jobs given their budgets. Users provide their specified budgets that they will pay if their jobs meet the deadlines. The deadline models in previous works can be categorized into two types: *hard* and *soft*. In the hard deadline model, a job gives some penalty if it does not meet its deadline. To the contrary, a job with soft deadline provides a diminished budget or utility even if it exceeds the deadline. The penalty function, such as linear decreasing function, is used in the soft deadline model.

However, traditional hard and soft deadline models are insufficient to model many recent applications, such as multimedia and QoS-related ones. The Imprecise Computation (IC) model [10] has been proposed for flexible scheduling and much research has been done on scheduling IC real-time tasks [11, 12]. A job in the IC model consists of two parts: a *mandatory* part and an *optional* part. The mandatory part is required for an acceptable result, while the optional part refines and improves the quality of the job in proportion to the amount of execution time before the deadline.

None of these previous works has considered the IC application model for market-based Grid environments. As resource management plays an important role in market-based Grids, a new flexible Grid application model is needed. In

this paper, we apply the IC model to Grid application model for the purpose of flexible market-based resource management and modelling of target Grid applications, such as distributed multimedia services. We also provide a resource allocation scheme with several scheduling algorithms.

The remainder of this paper is organized as follows. In Section 2, we describe the IC Grid application model and list examples of its applications it can support. Section 3 explains the proposed resource allocation scheme and scheduling algorithms. We show simulation results in Section 4 and conclude the paper in Section 5.

2. Imprecise Computation Grid Application Model

In the proposed IC Grid application model, a Grid application τ_i is composed of p_i independent IC jobs. Each job of a Grid application τ_i is defined by the following parameters.

- **Hard deadline (d_i):** The Grid application specifies the relative hard deadline d_i from the submission time, within which at least the mandatory part of the job must be completed.
- **Length of mandatory part (m_i):** The mandatory part of an IC job requires m_i MIs (Million Instructions).
- **Length of optional part (o_i):** The optional part of an IC job requires o_i MIs.
- **Mandatory budget (b_i^m):** If the mandatory part of a job is finished before its deadline, the application user pays the mandatory budget b_i^m to the resource.
- **Budget function for optional execution ($b_i^o(l)$):** The budget for the optional part of a job is defined as a function of the length of executed optional part l ($0 \leq l \leq o_i$).

In market-based Grids, users assign a portion of budget for their applications to be run on the Grid. If the quality of the application results satisfies the user's requirement, the resource obtains the specified budget from that user. Since the mandatory part is the minimum requirement of an IC job, the resource site will receive a minimum budget b_i^m .

In case of optional parts of jobs, the budget depends on how much optional parts are executed before the deadline. The budget (as for reward in [12]) for the optional part is determined by the function $b_i^o(l)$ of executed optional part length l . This budget function is generally a non-decreasing function, so that higher budget is payable for executing more optional parts. Figure 1(a) shows the budget function of an IC job's optional execution, while Figure 1(b) shows the budget function of an entire job depending on both its

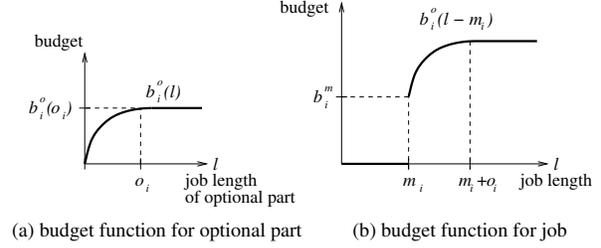


Figure 1. Budget Function for IC Job

mandatory and optional part execution. The budget is given according to the amount of executed job length before the deadline. Thus, the maximum budget of an IC Grid application τ_i is $p_i \times (b_i^m + b_i^o(o_i))$.

The proposed IC Grid application model can be applied to many Grid applications. Possible applications and targets of the IC Grid application model are:

- **Flexible use of Grid resource:** The IC Grid application model specifies an application's behavior in two respects: mandatory and optional parts. When the resource site is overloaded, it can be adopted in a manner that only mandatory parts of jobs are executed.
- **Fine-grained QoS specification:** A user can describe a Grid application's QoS in more detail using the proposed IC Grid application model. The minimum required quality is specified by the mandatory part. Users also pay their budgets according to the quality of the application results serviced by resource providers.
- **Penalty estimation based on application's error or quality:** When all optional parts of an IC Grid application are executed, it is said to be *precise* [10], which indicates that the result contains no error. Therefore, the result error or quality of an IC Grid application can be used to the system penalty estimation.
- **Multimedia services in the Grid:** One of the target applications of the IC application model is multimedia services. While the mandatory part of an application job corresponds to a minimum quality of a multimedia content, the optional part enhances the multimedia quality.
- **Iteration-sensitive applications:** The proposed model is suitable for iteration-sensitive problem solving techniques, such as probabilistic algorithms and genetic algorithms. These algorithms show more accurate results in proportion to the number of iterations. The mandatory part can be defined as the number of iterations for the minimum accuracy of the algorithms.

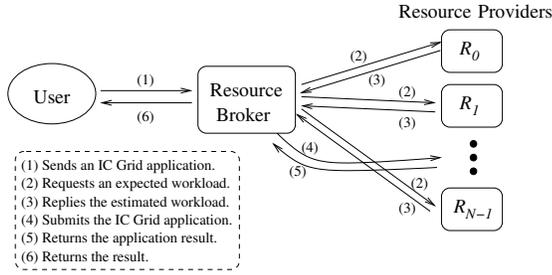


Figure 2. Resource Allocation Procedure

3. Resource Allocation and Scheduling Scheme

In this section, we provide a resource allocation scheme for the IC Grid application model. In addition, we provide possible approaches to schedule IC Grid applications in resource sites.

3.1. Workload-based Resource Allocation

In the Grid, a *resource broker* is in charge of resource allocation of Grid applications. Figure 2 shows an IC Grid application execution procedure through the resource broker. A user submits a new IC Grid application to the resource broker (step 1). When the resource broker receives the application, it requests the estimated amount of workload that each resource provider can complete (step 2). After all estimated workloads are received, the resource broker allocates the Grid application to the resource site which can complete more parts of applications (step 3, 4). When the application is completed, the resource broker sends the result from the resource site to the user (step 5, 6).

The amount of workload that each resource provider can complete by the new IC Grid application is defined as the difference between the estimated budget of the schedule including this new application and that of the original schedule without the application. Resource providers can produce different workloads for a given IC Grid application due to many aspects, such as system capacity, current load, and so on. We will suggest several scheduling algorithms and workload estimation in the following subsections.

3.2. Scheduling Scheme of IC Grid Applications

The proposed scheduling of IC Grid applications is based on *Mandatory First* policy in order to support the minimum qualities of as many applications as possible. The scheduler or dispatcher selects a job that requires mandatory part execution first. As long as there is no waiting job requiring mandatory part execution, it executes optional parts

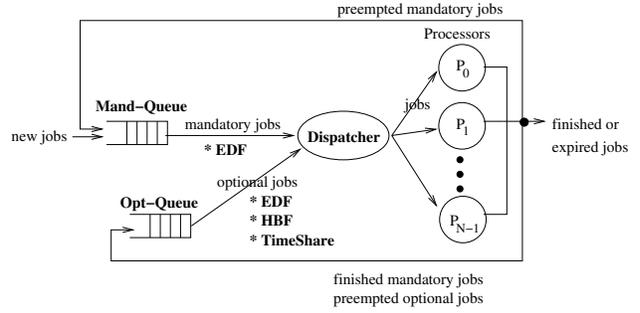


Figure 3. IC Grid Application Scheduling Scheme within a Resource

of jobs. Figure 3 shows the scheduling scheme in a resource site. We assume that a resource site manages two common waiting queues for multiple processing elements: *Mand-Queue* for jobs requiring mandatory part execution and *Opt-Queue* for jobs with optional part execution.

When a new IC job arrives at the resource site, the scheduler allocates the job to a free processor. If all processors are busy, the scheduler preempts a job whose priority is lower than the new job. The preempted job is sent to either *Mand-Queue* or *Opt-Queue*. If the new job cannot preempt a job, it is inserted in *Mand-Queue* and waits to be run.

When a job finishes its execution, the resource site receives the budget according to the job's budget function. When the mandatory part of a job completes, it is inserted to *Opt-Queue*. On the contrary, the job which finishes its optional part is returned to the user. Once the deadline of a job expires, the resource site does not execute the job any more and returns the result at the time to the user.

3.3. Priority Assignment Schemes

The priority assignment scheme for mandatory part jobs is EDF (Earliest Deadline First) in order to meet the deadlines of jobs. Thus, the dispatcher selects the mandatory part jobs with the earliest deadline in *Mand-Queue* if there is an idle processing unit. We also provide three scheduling schemes for jobs with optional parts: EDF, HBF, and TimeShare. In Section 4, we show simulation results of those schemes.

- *EDF*: Since all jobs have their hard deadlines, EDF policy can show a good performance in terms of the number of jobs meeting their deadlines.
- *HBF (Highest Budget-rate First)*: HBF assigns higher priorities to jobs with higher budget rates. The instant budget rate of a τ_i 's job is defined as the derivative of $b_i^o(l)$: $\frac{db_i^o(l)}{dl}$, where l is the current executed job length.

Since the instant budget rate changes for a short time, it cannot be used as a metric of priority. Thus, we define the *budget rate* of an job as $b_i^o(o_i) - b_i^o(0)$ and selects the job with the highest budget rate.

- *TimeShare*: It is a time sharing scheme between remaining jobs. Suppose that there are p processors available and n_o jobs requiring optional part execution. If $p \geq n_o$, each job is executed on one processor. If $p < n_o$, however, $\lceil n_o/p \rceil$ or $\lfloor n_o/p \rfloor$ jobs are executed on each processor in time sharing mechanism.

3.4. Workload Estimation

The budget estimation of a given application set is obtained by pre-scheduling of the application set. First, the finish times of all jobs with mandatory part execution are calculated by EDF policy such that each job in the priority order is assigned to the processor with the shortest service time. It requires $O(n_r \times n_\tau)$ time complexity, where n_r and n_τ are the number of processors and that of IC jobs respectively. Next, service times of all jobs requiring optional part execution can also be calculated similarly.

If the resource provider receives a workload request of a new IC Grid application, it calculates the budget by the current application set and that of the new application set including the new IC Grid application. If resource site is unable to complete the mandatory part of any job due to the lack of capacity or its current commitment, it will not accept the new application.

4. Simulation Results

In this section, we simulate the proposed resource allocation policies using the GridSim toolkit [13, 14]. We simulate six resources, each having different system performance with a total of 144 processors as shown in Figure 4(a). The generated workload consists of 1000 IC Grid applications. Each application parameters are generated as shown in Figure 4(b). The interarrival time between two consecutive applications is generated by Poisson distribution with a mean time of 5 minutes. The simulated budget functions are linear functions with different budget rates, as shown in Figure 4(c).

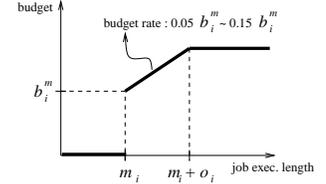
In the simulation results shown in Figure 5, we vary the deadline laxity from 10% to 80% more than the average execution time. The proposed resource allocation schemes based on Mandatory First policy are denoted as **MF-EDF**, **MF-HBF**, and **MF-TimeShare**, each having different optional part scheduling algorithms: EDF, HBF, and TimeShare, respectively. In order to compare the previous hard deadline application model, we simulate **Hard-M+O** with

	# of processors	MIPS
R1	32	500
R2	16	750
R3	16	1000
R4	24	1000
R5	32	1250
R6	24	1500

(a) Resource characteristics

p_i	1 ~ 16
m_i	600,000 MIs ~ 7,200,000 MIs
o_i	$0.5m_i \sim 4m_i$
b_i^m	$0.001 * m_i$
d_i	10%, 20%, 40%, 60%, 80% more than average execution time

(b) IC Grid application parameters



(c) Budget function

Figure 4. Simulation Environments

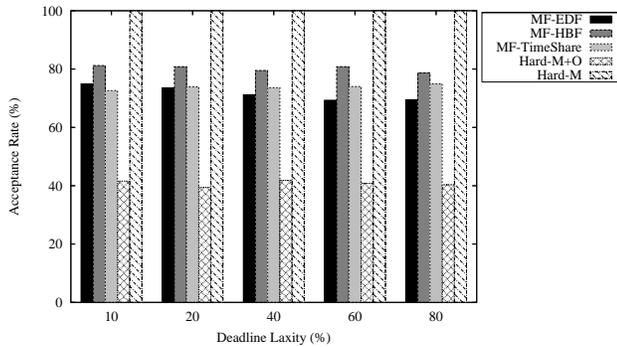
the computation requirement with both mandatory and optional parts, and **Hard-M** with only mandatory parts. Both schemes use EDF policy to schedule applications.

As indicated in Figure 5(a), **Hard-M+O** only accepts less than half of applications. The proposed schemes, however, accept about 80% applications due to their flexible scheduling policy. Since **Hard-M** only executes the mandatory part of applications, it shows 100% acceptance rate but provides low profit. As shown in Figure 5(b), the proposed schemes provide more profits to the system due to their flexible use of resource based on the proposed IC Grid application model. In overloaded environments (in case of 10% and 20% of deadline laxity), proposed algorithms show better performance compared to hard deadline models.

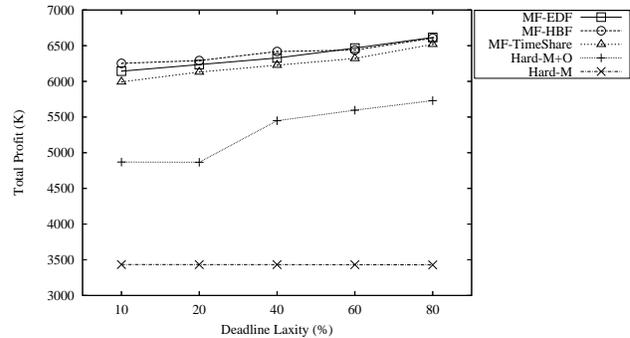
Next, we evaluate the impact of budget rate to total profit. In Figure 6, deadline laxity of each application is randomly generated from 10% to 50%. The budget rate of optional part is varied from the average $0.05b_i^m$ to $0.3b_i^m$. In lower budget rates, **Hard-M+O** produces even less budget than **Hard-M** because it schedules all the optional parts for low budgets. The proposed schemes show better performance due to its flexible scheduling.

5. Conclusions

In this paper, we proposed a new IC Grid application model for flexible market-based resource management. An IC Grid application consists of multiple independent jobs with mandatory and optional parts. If the mandatory part of an IC job meets the deadline, the mandatory budget is given. The optional part produces an additional budget in proportion to the amount of execution, which is specified by the budget function for the optional part. The proposed model



(a) Application acceptance rate



(b) Total profit

Figure 5. Simulation Results w.r.t. Deadline

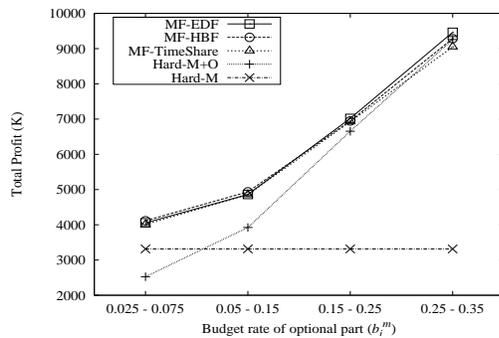


Figure 6. Profit w.r.t. Optional Budget Rate

can be applied to many QoS-related Grid applications. We also provided a resource allocation scheme based on three scheduling algorithms in the resource site. Simulation results show that the system utility is enhanced when users specify their mandatory and optional requirements.

References

- [1] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3):200–222, 2001.
- [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 Journal*, 18(8):1061–1074, 2002.
- [3] B. N. Chun and D. E. Culler. User-centric performance analysis of market-based cluster batch schedulers. In *Proc. the 2nd International Symposium on Cluster Computing and the Grid (CCGrid 2002)*, pages 22–30, Berlin, Germany, May 2002.
- [4] D. E. Irwin, L. E. Grit, and J. S. Chase. Balancing risk and reward in a market-based task service. In *Proc. the 13th International Symposium on High Performance Distributed Computing (HPDC 13)*, pages 160–169, Honolulu, USA, June 2004.
- [5] M. Islam, P. Balaji, P. Sadayappan, and D. K. Panda. Towards provision of quality of service guarantees in job scheduling. In *Proc. the 6th International Conference on Cluster Computing (CLUSTER 2004)*, pages 245–254, San Diego, USA, September 2004.
- [6] C. S. Yeo and R. Buyya. Service level agreement based allocation of cluster resources: handling penalty to enhance utility. In *Proc. the 7th International Conference on Cluster Computing (CLUSTER 2005)*, Boston, USA, September 2005.
- [7] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya. Libra: a computational economy-based job scheduling system for clusters. *Software: Practice and Experience*, 34(6):572–590, 2004.
- [8] K. Lai, B. A. Huberman, and L. Fine. Tycoon: a distributed market-based resource allocation system. Tech. Report cs.DS/0404013, HP Lab, 2004.
- [9] F. I. Popovici and J. Wilkes. Profitable services in an uncertain world. In *Proc. Supercomputing '05*, Washington, USA, November 2005.
- [10] J.-Y. Chun, J. W.-S. Liu, and K.-J. Lin. Scheduling periodic jobs that allow imprecise results. *IEEE Trans. on Computers*, 19(9):1156–1173, 1990.
- [11] J. W.-S. Liu, K.-J. Lin, W.-K. Shih, A. C.-S. Yu, C. Chung, J. Yao, and W. Zhao. Algorithms for scheduling imprecise computations. *IEEE Computer*, 24(5):58–68, 1991.
- [12] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Optimal reward-based scheduling for periodic real-time tasks. *IEEE Trans. on Computers*, 50(2):111–130, 2001.
- [13] R. Buyya and M. Murshed. GridSim: a toolkit for the modeling and simulation of distributed management and scheduling for Grid computing. *Concurrency and Computation: Practice and Experience*, 14:13–15, 2002.
- [14] A. Sulistio, G. Poduvaly, R. Buyya, and C. K. Tham. Constructing a Grid simulation with differentiated network service using GridSim. In *Proc. the 6th International Conference on Internet Computing (ICOMP '05)*, Las Vegas, USA, June 2005.