Li Zhi-jie, Wang Cun-rui / International Journal of Engineering Research and Applications IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.1353-1358

Resource allocation optimization based on load forecast in computational grid

Li Zhi-jie, Wang Cun-rui
(School of Computer Science and Engineering, Dalian Nationalities University, Dalian Liaoning 116600, China)

ABSTRACT
This paper presents a grid resource allocation strategy based on load forecast for optimizing user's execution time in a proportional resource sharing environment. The problem of multiple users competing for computational resource is formulated as a multi-player game. The goal of each grid user is to complete its tasks as quickly as possible within the budget constraint. Through finding the Nash equilibrium solution, a profile of user optimal bid is produced to allocate resource. In particular, a load forecasting method for grid resource price is proposed using sequential game. The experimental results show that the proposed allocation based on load forecast using sequential game outperforms the allocations using other three forecasting methods in terms of resource processing time.

Keywords - **About five key words in alphabetical order, separated by comma**

I. INTRODUCTION
Grid computing emerged as an important new field, it complement rather than compete with existing distributed computing technologies, because of their focus on dynamic, cross-organizational sharing. Grid technology distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation [1].
Resource allocation and management [2] in grid environments is a complex undertaking. This paper is concerned with optimal allocation of computational resources (CPU time) in grid computing. Many researches [3~8] have explored economic theory in managing grid resource, since grid is a heterogeneous and distributed environment. The economic model of proportional resource sharing is a good way of managing large-scale sharing resource in an organization. In this model, the percentage of resource share allocated to the user application is proportional to the bid value in comparison to other users' bids. Several research systems have explored the use of proportional resource sharing model from trading resources to managing resources [9~12]. This paper is focus on resource allocation optimization in the proportional sharing economic model.

In addition, the models in Game Theory [13], [14] are very suitable for solving the problem of competitive activities. Sequential game is a very important concept in dynamic game. This paper formulates the problem of resource allocation optimization with multiple users competing for computational resources as a multi-player game of perfect information. In order to obtain the reasonable resource price, the sequential game is used to calculate the current resource load.

Nowadays, the typical systems that have explored the use of different game models for managing resources contain: GameMosix [15] is a game-theoretic middleware. Selfish behaviors are modeled by "friendship relationships" in that computers will help each other only when they have established friendship relationships before. P. Ghosh [16] devised a framework for unifying network efficiency, fairness, utility maximization and pricing using Nash Bargaining Solution (NBS). D. Grosu [17] designed a load balancing system based on the Vickrey-Clarke-Groves (VCG) mechanism in which each computer optimizes its "profits" by considering the payment and cost involved in handling a job. Y. K. Kwok [18] presented a hierarchical game theoretic model of the grid by taking machine selfishness into account. J. Bredin [19] proposed a game-theoretic formulation of multi-agent resource allocation that minimizes an agent's execution time with a fixed budget constraint. Based on the work of J. Bredin, similar policy that minimizes an agent's execution cost with a fixed deadline constraint is presented by R. T. Maheswaran [20]. But both J. Bredin and R. T. Maheswaran consider only evaluated resource load as parameter, not real resource load. Hence, they cannot obtain reasonable resource price. This can in turn lead to less efficient resource allocation.

How to forecast the future resource load? A simple and efficient way is "sequential game". This paper presents a resource allocation strategy for time optimization using sequential game. This strategy makes grid users play second game using the real resource load from first game to form the final resource price and the optimal bid of each user.

II. Grid Resource Allocation Model

The system model follows that presented in [19]. There are N grid users competing for a computing resource with fixed finite capacity. These grid users are given a job of completing a sequence of tasks of different types by purchasing resource from grid resource. The resource is allocated using the proportional resource sharing mechanism, where the partitions depend on the relative bids sent by the grid users. Because the resource availability time is set to zero, the completion time for a task is equal to *ETC* (expected time to compute) of the task. One criterion used to optimize performance is the sum of *ETC* of all tasks in a job. The sum of *ETC*, $\sum_{i=1}^{K} ETC_i$, is the job execution time. It is assumed that there are K types of resources and that each grid user only needs to complete a task of a particular type at most once. The grid user's job is a sequence $\{q_k^i\}_{k=1}^K$, where q_k^i is the size of the kth type of task for the ith grid user. Let c_k^i be the capacity of the grid resource chosen by the ith grid user to complete its task of type k. The ith grid user receives resource proportional to its bid relative to the sum of all bids,

$$r_k^i = c_k^i\left(\frac{b_k^i}{B_k^i}\right) = c_k^i\left(\frac{b_k^i}{b_k^i + B_k^{-i}}\right) \qquad (1)$$

where b_k^i is the amount per second that the ith grid user bids for resource, the grid resource receives bids totaling B_k from the set of grid user, A_k, and $B_k^{-i} = \sum_{j\in A_k, j\neq i} b_k^j$. The grid users are assumed to have perfect knowledge (i.e. B_k^{-i} is known) about the states of prices of various resources. Since b_k^i is independent from B_k^{-i}, ($B_k^{-i} + b_k^i$) is substituted for B_k. Then, the time taken by the ith grid user to complete its task of type k is

$$t_k^i = \frac{q_k^i}{r_k^i} = \frac{q_k^i(b_k^i + B_k^{-i})}{c_k^i b_k^i} \qquad (2)$$

and the expense to the grid user is

$$e_k^i = t_k^i \cdot b_k^i = \frac{q_k^i(b_k^i + B_k^{-i})}{c_k^i} \qquad (3)$$

III. Optimal Problem for Resource Allocation

In this section, the i superscript is dropped in all variables except B_k^{-i}. The time optimization problem of grid user is to complete its job as quickly as possible with a finite budget, E_i. The optimization problem can be described below:

$$\min \sum_{k=1}^{K} t_k^i \quad , \qquad s.t. \quad \sum_{k=1}^{K} e_k^i \le E_i \qquad (4)$$

Problem (4) can be solved using Lagrange methods, and the Lagrange is

$$L = \sum_{k=1}^{K} t_k + \lambda\left(\sum_{k=1}^{K} e_k - E_i\right) \qquad (5)$$

Substituting for t_k and e_k into equation (5) and taking partial derivatives with respect to b_k, there is

$$\frac{\partial L}{\partial b_k} = -\frac{q_k B_k^{-i}}{c_k (b_k)^2} + \lambda \frac{q_k}{c_k} = 0 \Rightarrow \lambda = \frac{B_k^{-i}}{(b_k)^2} \qquad (6)$$

Note that $B_k^{-i} > 0$ implies λ >0 for all jobs. Thus there exists the following relationship between any two bids, j and k:

$$b_k = b_j \sqrt{B_k^{-i} / B_j^{-i}} \qquad (7)$$

Incorporating the constraint, there is

$$\frac{\partial L}{\partial \lambda} = \sum_{k=1}^{K} \frac{q_k (b_k + B_k^{-i})}{c_k} - E_i = 0 \qquad (8)$$

Substituting for $\{b_k\}_{k=2}^K$ in terms of b_1 using the relationship in equation (6), there is

$$\frac{q_k}{c_k}(b_k + B_1^{-i}) + \sum_{k\neq 1} \frac{q_k}{c_k}\sqrt{\frac{B_k^{-i}}{B_1^{-i}}} b_1 + \sum_{k\neq 1} \frac{q_k}{c_k} B_k^{-i} - E_i = 0 \qquad (9)$$

Introducing the following variables,

$$\alpha_1^i = E_i - \sum_{k\neq 1} \frac{q_k^i}{c_k^i} B_k^{-i}, \quad \beta_1^i = \frac{q_1^i}{c_1^i}, \quad \gamma_1^i = \sum_{k\neq 1} \frac{q_k^i}{c_k^i}\sqrt{B_k^{-i}}$$

Solving Equation (9) for b_1^i, there is

$$b_1^i = \frac{\alpha_1^i - \beta_1^i B_1^{-i}}{\beta_1^i + \frac{\gamma_1^i}{\sqrt{B_1^{-i}}}} \tag{10}$$

Equation (10) can be expressed as:

$$b_1^i = \frac{(\alpha_1^i - \beta_1^i B_1)^2}{2\gamma^2}(-1 + \sqrt{1 + \frac{4\gamma^2 B_1}{(\alpha_1^i - \beta_1^i B_1)^2}}) \tag{11}$$

where $B_1 \in (0, \frac{\alpha_1^i}{\beta_1^i})$ and $b_1^i = 0$ otherwise. The b_1^i expressed in Equation (11) is the optimal bid for the first task or current task of the *i*th grid user given the demand of the resources for the tasks in its job.

IV. Figures and Tables Load Forecast Based Allocation Optimization

This section describes how the sequential game can be used to forecast the future resource load. The game result is a Nash equilibrium. Intuitively, Nash equilibrium is a kind of "stalemate" in which no one is interested in changing, given no change of others. It distinguishes from general equilibrium theory by its focus on equilibrium analysis between not only consumer and supplier, but also consumer and consumer. Obviously, Nash equilibrium matches well with real-life.

Inherent in the settings what to be considered is the competition among grid users attempting to gain access to limited computational resources. With the bid-based proportional resource sharing mechanism, the performance of each grid user is affected by the actions of all other grid users. The autonomy of grid users creates an environment where each grid user is acting to better its own utility. In this context, utility u_i is defined as the reciprocal of job execution time. The attempt to find an operating point calls for Nash equilibrium. A Nash equilibrium solution is a set of bids where no grid user can gain advantage by unilaterally changing its bid, i.e.

$$(\mathrm{b}^i)^* = \arg\max_{b^i} \ \mathrm{u}_i(\mathrm{b}^i; \mathrm{B}^{-i}) \qquad i = 1, 2, ..., n$$

A sequential game consists of many stage games. This paper focuses on the sequential game with two stage games. The second game takes the result of the first game as condition, so an equilibrium different from the first game is produced. Let *G* be a stage game, *G(1)* be first game, *G(n)* be *n*th game, $B^{G(1)}$ be the resource price produced by first game, $B^{G(n)}$ be the resource price produced by second game.

First the evaluated resource price load *B* is used in time optimization allocation for the first game, solve the equation: $B = \sum_{i=1}^{N} b_i(B)$, i.e.

$$B = \sum_{i=1}^{N} (\frac{(\alpha^i - \beta^i B)^2}{2{\gamma^i}^2}(-1 + \sqrt{1 + \frac{4{\gamma^i}^2 B}{(\alpha^i - \beta^i B)^2}})) \tag{12}$$

Then, substitute the real resource price load $B^{G(1)}$ obtained from the first game for the *B* in equation (12), solve equation: $B^{G(1)} = \sum_{i=1}^{N} b_i(B^{G(1)})$, i.e.

$$B^{G(1)} = \sum_{i=1}^{N} (\frac{(\alpha^i - \beta^i B^{G(1)})^2}{2{\gamma^i}^2}(-1 + \sqrt{1 + \frac{4{\gamma^i}^2 B^{G(1)}}{(\alpha^i - \beta^i B^{G(1)})^2}})) \tag{13}$$

From equation (13), we get the second resource price $B^{G(2)}$ of second game based on the real current resource load. Repeat this process, we will get the *(n-1)*th resource price $B^{G(n-1)}$. Using equation (14), the final user bid can be calculated. At the resource, we would like to generate a set of bids that form a Nash equilibrium with respect to the strategies of the N grid users:

$$\left\{ b^i = \max\left\{ 0, \frac{(\alpha^i - \beta^i B^{G(n-1)})^2}{2{\gamma^i}^2}\left(-1 + \sqrt{1 + \frac{4{\gamma^i}^2 B^{G(n-1)}}{(\alpha^i - \beta^i B^{G(n-1)})^2}} \right) \right\} \right\}_{i=1}^{N} \tag{14}$$

V. Allocation Algorithm of Load Forecast-Based Allocation

(1)Grid resources register themselves with Grid Information Service (GIS). This resource registration process is similar to GRIS (Grid Resource Information Server) registering with GIIS (Grid Index Information Server) in Globus system.

(2)Grid user query GIS for resource discovery. The GIS returns a list of registered resources and their contact details. Grid user sends events to grid resource with request for resource configuration and properties. These grid resources respond with dynamic information such as resources capability, zone, availability, load, and other configuration parameters.

(3)Grid users submit bidding functions containing three coefficients (α, β, γ) according to the dynamic information of the desired grid resource. Grid resource form the current resource price by using a bisection search to find the bidding point $B=\sum_{i=1}^{K} b^i(B)$ and send back it to grid users.

(4)Grid users submit bidding functions again according to the resource current load $B^{G(i)}$. Grid resource forms the resource price $B^{G(i+1)}$ and sends back a feedback pair $(B^{G(i+1)}, b_i / B^{G(i+1)})$ to grid user,

(5) If convergent condition is satisfied, then the algorithm ends and output the final resource price $B^{G(i+1)}$, which denotes the congestion for that current time slot and the resource rate received. Otherwise, go to step (4).

VI. EXPERIMENTS AND RESULTS

The main aim of the experiment is to demonstrate the effectiveness of load forecast-based allocation strategy whose performance needs to be evaluated under different scenarios such as varying the number of users with different requirements. It is hard to perform performance evaluation involving multiple users in a repeatable and controllable manner for different scenarios due to dynamic nature of grid environment. Therefore, this work simulates a grid environment based on a Java-based discrete-event grid simulation toolkit called GridSim. The toolkit provides facilities for modeling and simulating grid resources and grid users with different capabilities and configuration. To simulate application scheduling in GridSim environment requires the modeling and creation of GridSim resources and applications that model tasks.
Resource modeling: first, CPUs (also called Processing Elements (PEs)) are created with different MIPS (Million Instructions Per Second) rating as SPEC-like ratings. Then, one or more PE are put together to create a machine (a single CPU node). Similarly, one or more machines are put together to create a grid resource. Thus, the resulting grid resource can be a single processor, shared memory multiprocessors (SMP), or a distributed memory cluster of computers. Grid resources are modeled and simulated as many as 5 with different characteristics such as number of PE, speed of processing, time zone, etc. These grid resources are managed by proportional sharing mechanism. The resources capability (i.e. total PE MIPS rating) is defined as a MIPS rating multiple number of PE. It varied with normal distribution from 0.5 to 1.5. For every grid resource, local workload is estimated based on typically observed load conditions, but all resources are initially no loads. The network communication speed between user and resource are defined in terms of data transfer bandwidth rate 100Mbps. However, to simplify the experiment setups, all resources and users have the same set of network properties.

User modeling: a task is a tiny grid application that contains all information related to task execution management details such as tasks processing requirements, expressed in MIPS, disk I/O operations, the size of input files that help in computing execution time of remote resource and the size of output files. Grid users are modeled as many as 20 that are competing for resource. Each user consists of 4 tasks with variation of ± 1. Each task is heterogeneous in terms of task length and input files size. Task length is expressed in such a way that they are expected to take at least 20,000 MI (Million Instructions) with a random variation of 0 to 10% on the positive side. This 0 to 10% random variation in task length is introduced to model heterogeneous in different tasks.

To evaluate the performance of the proposed load forecast based resource allocation, we compare the load forecast using sequential game with other three common forecasting methods, such as one-step-ahead (OSA), interval mean (IM) and history mean (HM), in terms of resource processing time. The varying curves of resource processing time using four methods are given from Figure 1 to Figure 4.

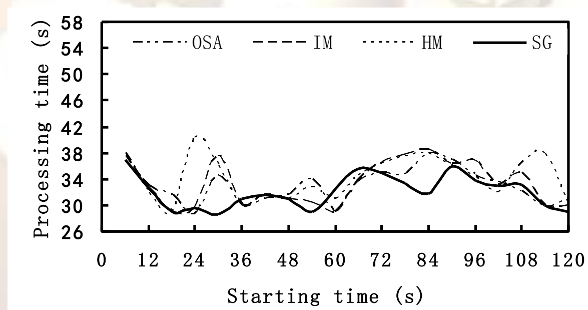

Fig. 1 Resource processing time comparison of four load prediction methods on the resource with low-mean and low-variance load

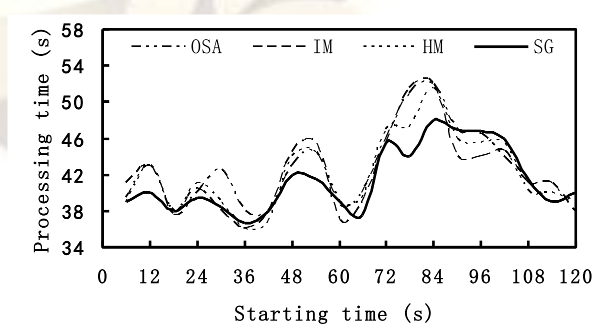

Fig. 2 Resource processing time comparison of four load prediction methods on the resource with low-mean and high-variance load

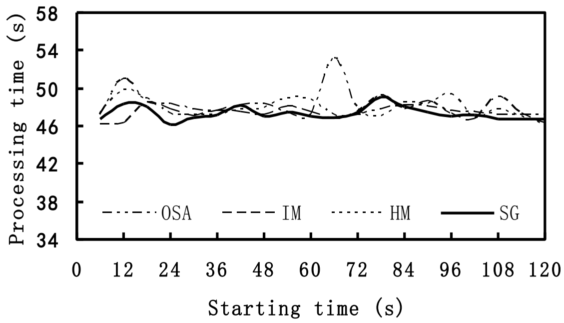

Fig. 3 Resource processing time comparison of four load prediction methods on the resource with high-mean and low-variance load

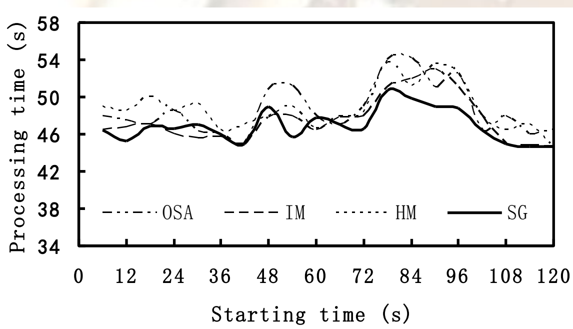

Fig. 4 Resource processing time comparison of four load prediction methods on the resource with high-mean and high-variance load

Under four kinds of load background, the resource processing time using four methods have great difference. For low-mean load, the resource processing time is relatively short; for high-mean load, the resource processing time is longer; for low-variance load, the resource processing time changes little; for high-variance load, the resource processing time changes great. The result of load forecast using sequential game is best. This is because that after continuous game, every user has much information about others, which is helpful to forecast resource load. OSA method is not very stable due to its forecast mainly aims at short-term changing, not long-term changing. IM belongs to the method of forecast using mean value; hence its effect is not good for paroxysmal load changing. HM method produces the worst result. It is because that this method contains larger time scale, therefore will be fit for long-term load forecasting.

From above performance comparisons, we can get an important conclusion: the proposed allocation based on load forecast using sequential game outperforms the allocations using other three forecasting methods in terms of resource processing time.

VII. Conclusion

In grid environment, resource load prediction is a crucial and difficult problem affecting resource allocation optimization due to heterogeneity and dynamic nature. In response to this issue, this paper proposes a resource allocation strategy that uses sequential game method to predict resource load for time optimization in a proportional resource sharing environment. The problem of multiple users bidding to compete for a common computational resource is formulated as a multi-player dynamic game. The goal of each grid user is to complete its tasks as quickly as possible within the budget constraint. Through finding the Nash equilibrium solution of the multi-player dynamic game, a set of user optimal bids is produced to partition resource capacity according to proportional sharing mechanism. The experimental results show that the proposed allocation based on load forecast using sequential game outperforms the allocations using other three forecasting methods in terms of resource processing time. Hence, the load forecast based allocation as described in this work maybe a good choice to achieve resource allocation optimization in grid environment.

References

I. Foster, C. Kesselman, S. Tuecke, The anatomy of the Grid: Enabling scalable virtual organizations, *International Journal of High Performance computing Applications*, 2001, 15(3): 200-222.

2. K. Krauter, R. Buyya, M. Maheswaran, A taxonomy and survey of grid resource management system, *Softw.: Pract. Exp*. 32 (2) (2002) 135-164.

3. K. Subramoniam, M. Maheswaran, M. Toulouse, Towards a micro economic model for resource allocation in grid computing systems, *Proceedings of the 2002 IEEE Canadian Conf. on Electrical & Computer Engineering*, Manitoba, Canada, June 15, 2002.

4. An Economic-based Resource Management Framework in the Grid context, *Proc. CCG 2005*, Cardiff, Wales, UK, May 9–12, 2005.

5. R. Buyya, H. Stockinger, J. Giddy, D. Abramson, Economic models for management of resources in peer-to-peer and grid computing, *Proceedings of the International Conference on Commercial Applications for High-performance Computing*, SPIE Press, 2001, pp.13-25.

6. R. Buyya, D. Abramson, J. Giddy, A case for economy grid architecture for service oriented grid computing, *Proc. HCW* 2001, pp. 776-790.
7. K.-M. Chao, R. Anane, J.-H. Chen, R. Gatward, Negotiating Agents in a Market-Oriented Grid, *Proc. CCGrid* 2003.
8. R. Wolski, J. S. Plank, J. Brevik, T. Bryan, Analyzing market-based resource allocation strategies for the computational grid, *International Journal of High Performance Computing Applications* 15 (3) (2001) 258-281
9. U. Maheshwari, Charge-based proportional scheduling, *Technical Memorandum MIT/LCS/TM-529, MIT Laboratory for Computer Science*, April 1995.
10. I. Stoica, H. Abdel-Wahab, K. Jeffay, et al, A proportional share resource allocation algorithm for real-time-shared systems, *Proc. RTSS 1996, Washington, D C*, 1996, pp. 288-299.
11. B. Chun, D. Culler, Market-based proportional resource sharing for clusters, *Technical Report CSD-1092*, University of California, Berkeley, USA, January 2000.
12. Chunlin Li, Layuan Li, Competitive proportional resource allocation policy for computational grid, *Future Generation Computer Systems* 20 (2004) 1041-1054.
13. M.J. Osborne, A. Rubinstein, A Course in Game Theory, *MIT Press*, 1994.
14. R. Mahajan, M. Rodrig, D. Wetherall, et al, Experience Applying Game Theory to System Design, *Proc. ACM* SIGCOMM 2004.
15. D.E. Volper, J.C. Oh, M. Jung, GameMosix: Game-Theoretic Middleware for CPU Sharing in Untrusted P2P Environment, *Proc. PDCS* 2004, San Francisco, CA.
16. P. Ghosh, N. Roy, S. K. Das, et al, A Game Theory Based Pricing Strategy for Job Allocation in Mobile Grids, *Proc. IPDPS 2004*, Santa Fe, USA, IEEE Computer Society Press, April 26-30, 2004, pp. 26-30.
17. D. Grosu, A.T. Chronopoulos, Algorithm Mechanism Design for Load Balancing in Distributed Systems, *Proc. CLUSTER* 2002, Chicago, USA, September 23-26, 2002, pp. 445-450.
18. Y.K. Kwok, Shan-Shan Song, Kai Hwang, Selfish Grid Computing: Game-Theoretic Modeling and NAS Performance Results, *Proc. CCGrid* 2005, Cardiff, UK, May 9-12, 2005, pp. 349-356.
19. J. Bredin, R.T. Maheswaran, Çagri Imer, et al, A Game-Theoretic Formulation of Multi-Agent Resource Allocation, *Proceedings of the 4th International Conference on Autonomous Agents*, Barcelona, Spain, 2000, pp. 349~356.
20. R.T. Maheswaran, T. Başar, Nash Equilibrium and Decentralized Negotiation in Auctioning Divisible Resources, *Group Decision and Negotiation* 12(5) (2003) 361-395.