Journal of Computer Science 8 (4): 538-544, 2012
ISSN 1549-3636
© 2012 Science Publications

Integrated Resource and Cost Management Scheme for Computational Grids

[1]Manavalasundaram, V.K. and [2]K. Duraiswamy
[1]Department of CSE, Velalar College of Engineering and Technology, Erode, India
[2]Department of CSE, K.S.R. College of Technology, Tiruchengode, India

Abstract: Problem statement: Autonomous decision making and resource scheduling are the main objectives of market-life computational grid. Resource providers and consumers make the scheduling decisions with cost and incentive factors. The two objectives are to maximize the success rate of job execution and to minimize fairness deviation among resources. The challenge is to develop a grid-scheduling scheme that enables individual participants to make autonomous decision while producing a desirable emergent property in the grid system. **Approach:** An incentive-based scheduling scheme is presented to utilize a peer-to-peer decentralized scheduling framework a set of local heuristic algorithms and three market instruments of job announcement, price and competition degree. The incentive based scheme is enhanced with priority based pricing schemes. The resource availability, job priority and network delay are used for the cost and incentive decisions. **Results:** The performance of this scheme is evaluated via extensive simulation using synthetic and real workloads. The system achieves efficient cost and incentive optimization for both providers and consumers. **Conclusion:** The approach outperforms other scheduling schemes in optimizing incentives for both consumers and providers, leading to highly successful job execution and fair profit allocation.

Key words: Computational grids, economic scheduling, resource sharing, provider and consumer

INTRODUCTION

Grid computing, which aims at enabling wide-area resource sharing and collaboration, is emerging as a promising distributed computing paradigm (Parashar and Lee, 2005). Based on how computational jobs are scheduled to resources, computational grids can be classified into two types: controlled and market-like grids. Both the types involve sharing and collaboration among resource providers and resource consumers and the scheduling schemes can be either centralized or decentralized. The key difference between the two lies in who makes scheduling decisions. In a controlled grid, the grid system decides when to execute which job on which resource. In a market-like grid, such decisions are made by each resource provider/consumer, but all the individual participants utilize some market instruments such as price to achieve the grid system wide objectives.

This work focuses on the scheduling problem in market like computational grids. In particular, it addresses the issues of optimizing incentives for both resource consumers and resource providers so that every participant has sufficient incentive to stay and play, leading to a sustainable market. The main challenge, phrased as a scheduling problem, is to schedule jobs of consumers to resources of providers to optimize incentives for both parties. Most importantly, such objectives should be realized not by an omnipotent scheduler, but rather, the scheduling scheme should be autonomous. That is, each participant makes decisions on its own behalf and the individual economic behaviors of all participants work together to accomplish resource scheduling, with optimized incentives being an emergent property of the grid system. Does such a scheme exist at all? The answer is not obvious.

Formulation of the above scheduling problem and investigation of market instruments and algorithms are done. Identification of the successful-execution rate of jobs as the incentive for consumers and the inverse of fairness deviation as the incentive for providers is made. As even a sub problem of the formulated scheduling problem is NP-complete, we develop a scheduling scheme (called IB) using local heuristics is done. Job announcement, Competition Degree (CD) and price are defined and used as market instruments. Four heuristic algorithms, local to each participant, are developed to utilize the market instruments and to optimize the incentives. Performance evaluation is conducted via extensive simulations, utilizing both

Corresponding Author: Manavalasundaram, V.K., Department of CSE, Velalar College of Engineering and Technology, Erode, India

statistically generated workloads and real workloads. The results show that the proposed IB scheme outperforms other schemes in optimizing incentives for both consumers and providers.

Problem formulation: We define a market-like computational grid as a quadruple G= (R, S, J, M). The grid G consists of a set of m resource providers R = {R0,..., Rm-1} and a set of k resource consumers S = {S0 .... Sk-1}. Over a time period T, a set of n jobs J = {J0,., Jn-1} are submitted to the grid by the consumers, scheduled by the scheduling scheme M and executed by resources of the providers. The scheduling scheme M should employ market instruments to allow each provider and each consumer to make the scheduling decision autonomously. That is, each provider Ri can decide whether it would offer its resource and each consumer Sj can decide whether it would use a certain resource to execute its jobs.

Consumers and jobs: In this work, computation-intensive jobs are considered and all communication/networking overheads are ignored. All jobs are independent of one another (Padala *et al*., 2003). The k consumers altogether have n jobs to execute in time period T. The consumers first submit job announcements to the computational grid. A job announcement includes the information of job length and job deadline. Job length is an empirical value assessed as the execution time of the job on a designated standard platform. Job deadline is a wall clock time by which a consumer desires a job to be finished, expressed as a number between 0 and T. Thus, a job with length = 10 and deadline = 100 means that the job’s execution takes 10 time units on a designated standard computer and it must be finished 100 time units after the common base time 0.

Providers and resources: From the scheduling viewpoint, each resource provider is modeled with three parameters: capability, job queue and unit price. Capability is the computational speed of the underlying resource, expressed as a multiple of the speed of the standard platform. The job queue of a resource provider keeps an ordered set of jobs scheduled but not yet executed. Each job, once it is executed on a resource, will run in a dedicated mode on that resource, without time-sharing or preempting. A provider charges for a job according to its unit price and job length. Unit price refers to the price that the resource offers for executing a job of unit length. When a provider with capability 5 bids to execute a job of length 20 at a unit price of 2 and if the consumer accepts the bid and decides to send the job to run there, the job will take 20/5 = 4 units of time to complete, generating a profit of 2×20 = 40 for the provider.

Incentives for consumers and providers: Intuitively, consumers are attracted to a grid, because it offers high quality of computational service at low cost. This could lead to many potential metrics of consumer incentives. However, a fundamental incentive requirement is that a grid should have a high successful-execution rate of jobs, where a successful job execution means that a job is executed without missing its deadline. When this rate is too low, even if the cost is zero (as in the case when a grid is advertising funded), the consumers will lose faith in the grid and quit it. Therefore, we choose the successful-execution rate of the grid system as the incentive for consumers.

Related work: Much attention has been devoted to the area of scheduling in distributed computing (Lai *et al*., 2005). However, to the best of our knowledge, there is still no work investigating effective scheduling to optimize incentives for both consumers and providers, utilizing market information. Many previous research projects focused on optimizing traditional performance metrics, like system utilization, system load balance and application response time in controlled grids. They did not consider market-like grids, where providing sufficient incentives for participants is a key issue.

Enterprise is a task scheduler for distributed market like computing environments. The work shows the effectiveness of a bidding model for a decentralized scheduling framework. Spawn is a market-based computational system that utilizes idle computational resources in a distributed network of heterogeneous computer workstations. The auctions employed by Spawn are sealed-bid second-price auctions. Buyya *et al*. (2005) identify the distributed resource management challenges and requirements of economy-based grid systems and discuss various representative economy-based systems. They also present commodity and auction models for resource allocation (Abdelkader *et al*., 2008). The evaluation results of computational and data grid environments demonstrate the effectiveness of economic models in meeting users’ QoS requirements (Abdelkader *et al*., 2009). A consumer initiated bid model is chosen in this work.

CompuP2P (Gupta *et al*., 2006) is an architecture for enabling Internet computing, using Peer-To-Peer (P2P) networks for sharing of computing resources. The work focuses on modeling pricing with the game theory and microeconomics to deal with selfish behavior and proves that its model guarantee the incentive for all the providers to share resources and not to cheat.

Enterprise tries minimizing the completion time of jobs. Spawn aims at the fairness of resource allocation: the number of CPU slots bought is proportional to the amount of funding. Nimrod/G is a resource management and scheduling system based on the parameter sweeping system. Nimrod and Nimrod/G built with Globus toolkit. Resources can be associated with prices and jobs can be given budgets. The authors do not focus on economic feature and give no further explanation and implementation of their economic idea over Nimrod/G. Libra (Sherwani *et al*., 2004) is an expansion of Nimrod/G for cluster computing. Its objective is to maximize the successful-execution rate under the constraint of budget. Performance evaluation shows its improvement in the rate of accepted jobs compared with FIFO. Unlike most related work that considers performance objective only for resource consumers, First Reward (Irwin *et al*., 2004), a value-based heuristic task scheduling scheme for a market based grid setting, tries maximizing the profits of providers.

Partial results of our incentive-based scheduling work are reported in (Zhu *et al*., 2004; Xiao *et al*., 2005). Zhu *et al*. (2004) consumers assign budgets to jobs and choose providers according to the claimed completion time. No price or CD mechanisms are investigated. In (Xiao *et al*., 2005), the impact of CD is studied. It does not formulate the dual-objective scheduling problem, develop a complete scheduling scheme, evaluate performance in detail, or provide quantitative comparison with related work, as what the current work does.

The incentive-based scheduling scheme: An incentive-based scheduling scheme IB is proposed here with heuristics, employing a P2P decentralized scheduling framework. The scheme is characterized as follows: (1) Each consumer or provider autonomously makes scheduling decisions, (2) All scheduling algorithms are local to a resource provider and (3) Three market instruments, job announcement, price and CD, are used.

Peer-to-Peer scheduling framework: Our scheduling framework takes advantage of the P2P technology, utilizing its characteristics of decentralization and scalability. A central server is far from robust and the maintenance is costly. Apart from that, as every participant in the computational grid is autonomous and acts individually. A decentralized scheduling infrastructure is more favorable. Furthermore, owing to the dynamics of grid environments, players may enter or leave at any time. A P2P network can handle such dynamics.

The computational grid G has several portals, via one of which a provider can join the grid. On entering, the provider gets the information of designated neighbors from the portal and then connects into the P2P network.

A consumer submits a job announcement to the computational grid via one portal. Then, the job announcement spreads throughout the P2P network, similar to query broadcast in an unstructured P2P system. The providers that receive a job announcement may bid for the job. Realization of the complete competition among all the providers based on two considerations is desired. Firstly, the job execution time is sufficiently long such that the overhead of executing them on remote computers becomes relatively negligible. Thus, all the providers should have an equal chance to compete for any job, without considering the geographical locations. Secondly, the number of providers will not be too large, (typically not more than several hundred), for a provider represents an administrative domain, within which local scheduling policies are employed. It is well known that blind-flooding-based broadcasting is a fatal weakness of unstructured P2P networks. Many investigators (Liu *et al*., 2004) have studied building overlay networks, whose topology closely matches the topology of physical networks. Once an overlay network with the desirable characteristic is built, an efficient broadcasting mechanism with good performance can be constructed.

The P2P scheduling infrastructure enables the effective interactions between consumers and providers and jobs are scheduled as a result. Scheduling scheme of steps is that a single job goes through in the scheduling scheme M. All jobs from consumers follow the same steps:

Step 1: A consumer submits a job announcement to the computational grid and the job announcement is broadcast to all the providers.

Step 2: Each provider, upon receiving a job announcement, estimates whether it is able to meet the deadline of the job. If yes, the provider sends a bid that contains the price for the job directly back to the consumer; otherwise, the provider ignores the job announcement.

Step 3: After waiting for a certain time, the consumer processes all the bids received, chooses the provider who charges the least and sends the job to the selected provider.

Step 4: The provider who receives the job inserts it into its job queue. When the job is finished, the provider sends the result to the consumer.

The value of the parameter-waiting interval in step 3 should try not to miss any potential bid and also to make decisions as soon as possible. In the experiments conducted in this work, the average execution time is chosen as the waiting interval for synthetic workloads and 10 sec for real workloads.

Both are rather conservative values so that the performance evaluation results will not be favorably skewed.

Incentive-based scheduling algorithms: The incentive based scheduling algorithm is designed with the criteria such as job levels, local schedule information and dynamic price assignment. Four algorithms have been designed in this work for providers. The job competing algorithm describes how a provider bids when receiving a job announcement in step 2. The heuristic local scheduling algorithm is responsible for arranging the execution order of jobs in the job queue of a provider. It starts when a provider receives a job in step 4. The price-adjusting algorithm and the CD-adjusting algorithm help a provider in dynamically adjusting its unit price and CD properly over the period of its participation in the computational grid.

Job competing algorithm: As a result of the decentralized scheduling framework, providers make decisions based on local, imperfect and delayed information, which often puts them in a dilemma.

Things get more complex when more jobs are involved. There are two extreme attitudes for providers to compete for jobs. One is aggressive. It means that a provider never considers the unconfirmed jobs when estimating whether it is able to meet job deadline. This is a risky one, but chances often accompany risks. The other is conservative. It means that a provider always keeps the unconfirmed jobs in the job queue for consideration for a certain time. This attitude will never lead to deadline missing but may lose potential chances and, thus, profits. Different competition attitudes will result in different allocations of profits. To study the impact of competition attitude, a parameter by name CD is defined a real number from 0-1. A provider will insert unconfirmed jobs into its job queue at the probability of 1-CD.

Every time a provider receives a job announcement, it starts the job competing algorithm. The algorithm is stated as follows: Its time complexity is O(q), where q is the number of jobs in the job queue:

Step 1: The provider estimates whether it is able to meet the job deadline.
Step 2: The provider offers a price for the job.

The pseudo code is given as follows:

```
price ← p * $L_s$;
if reordered then
    price ←$\chi$ *
endif
```

Here, p is the unit price of the provider, Ls is the job length of job s and χ is a decimal slightly larger than 1. When the variable reordered is set to true, the price is raised. Generally, jobs are enquired in the order of their arrival. To meet job deadlines, some jobs may be inserted into the job queue ahead of foregoing jobs, which indicates that the deadlines of these jobs are somewhat tight and the jobs need to be given higher priority. Thus, it is reasonable to charge more for them. On the other side, a tight deadline also increases the possibility of failing to meet it. Providers raise the price to reduce the chance of being chosen to some extent.

Step 3: The provider sends the price as a bid and inserts the job at the place that the variable insert place indicates at the probability of 1-CD. If the provider chooses to insert and the job does not come after a certain time, it deletes the job from its job queue. The duration of keeping an unconfirmed job should be as short as possible but long enough to guarantee not to delete offered jobs.

Heuristic local scheduling algorithm: Once the penalty model is introduced, providers must take some measures to minimize the loss. What a provider can do is to arrange the execution order of jobs in its job queue. We call it local scheduling. On calculating the penalty of all the possible permutations of jobs to find out the one with the least penalty is NP-complete, a heuristic approach is applied. The approach is based on the heuristic rule that when a job is inserted, the relative order of the jobs in the origin queue is unchanged. Every time a provider is offered a job that is not kept in the job queue, it starts the heuristic local scheduling algorithm. The algorithm is needless for providers whose CD is equal to 0, because they always keep unconfirmed jobs. The heuristic local scheduling algorithm is described with the following pseudocode. Its time complexity is O (q2):

```
insert_place ←$P_q$;
penalty ← calculate the penalty of inserting the job at $P_q$;
for I ← q - 1-0 do
    $penalty_i$ calculate the penalty of inserting the job at $P_i$;
    if $penalty_i$ < penalty then
        penalty ← $penalty_i$;
        insert_place ← $P_i$;
    end if
end for
insert the job at insert_place
```

Price-adjusting algorithm: As our performance objective for providers is the fair allocation of profits, it involves all the providers. It is almost impossible to be realized if every provider just behaves based on the local information. Inevitably, all the providers need to know some global information. In the algorithm of this work, it is assumed that every provider is informed with the aggregated capability of all the providers in the computational grid. The information can be acquired when a provider enters the grid via a portal and is updated in the same way that a job announcement is forwarded.

In a certain period of time, every commodity has a predominant price in the market. For a commodity like CPU cycles, such a price is easier to determine, because commodities of this kind do not have great difference in quality. We call the price as market price and it acts as a directive. When entering the grid, a provider gets the market price from a portal and sets it as the initial unit price. Then, every time a provider is offered a job or deletes an unconfirmed job, it starts the price-adjusting algorithm. The algorithm is stated as the following pseudocode and the time complexity of this algorithm is O (1):

```
$r1 \leftarrow L_o/L_T$
$r2 \leftarrow C / \sum_{0\le j\langle m} C_j$
if offered a job then
    if r1 > r2 and p <= $P_M$ then
        $p \leftarrow \alpha * p$;
    end if
else // delete an unconfirmed job
    if r1 < r2 and p >= $P_M$ then
        $p \leftarrow \beta * p$
    endif
endif
```

L_O, which is the offered job length, is the aggregated length of jobs offered to the provider. L_T, which is the total job length, is the aggregated length of jobs whose announcements are received by the provider. $\sum_{0\le j\langle m} C_j$ is the aggregated capability of all the providers. The offered job length and the total job length rewind when the total capability is updated. In addition, C and p are the capability and unit price of the provider, respectively, PM is the market price, α is a decimal above 1 and β is a positive decimal under 1. The price-adjusting mechanism in this work simple and intuitive: just to make prices different and it differentiates the chances of providers to be chosen and eventually realize the fair allocation of profits. Furthermore, the algorithm skillfully avoids endless increase or decrease in unit price. Thus, the price will fluctuate around the market price, which is acceptable for both consumers and providers.

Providers can choose not to adjust price every time one job is offered or not but start the algorithm every several jobs. However, if so, the providers are slow to react to the market. The fairness will be degraded accordingly.

Competition-degree-adjusting algorithm: Like human beings, providers have diverse behavior. Thus, providers with various CDs coexist in a computational grid. The more conservative ones are relatively less competitive than the more aggressive ones. They always keep unconfirmed jobs in their job queues and tend to lose potential jobs because of being unable to bid. Most likely, these jobs are offered to the more aggressive ones. As a result, fairness among all the providers is hard to achieve. Moreover, the jobs that could have been done by the conservative ones may bring the aggressive ones not only profit but also penalty, of course, which results from deadline missing. A wise provider, whether a conservative or an aggressive one, should never hold its attitude toward competition if things like that happen. It will adjust its CD according to the situation that it perceives. Thus it is the main objective of the CD-adjusting algorithm. The following pseudo code describes the algorithm and the time complexity of this algorithm is O (1):

```
//Every time the penalty increases
if $R_p$ >= $TH_p$ and CD >= $\varepsilon$ then
    CD $\leftarrow$ CD - $\varepsilon$;
endif
//Every time a certain interval such as 1 day
if $R_p < TH_p$ and $R_J$ >= $TH_J$ and CD <= 1 -$\varepsilon$ then
    CD $\leftarrow$ CD+$\varepsilon$;
endif
```

Here, Rp is the ratio of penalty to profit and R_J is the ratio of jobs that the provider does not bid for. TH_P and TH_J are thresholds for them, respectively. If one rate gets above its threshold, CD is adjusted accordingly at the step of ε. As can be seen, the check of Rp is not only timelier but also prior. The reason is that the rate of penalty to profit is a more obvious index to providers. Thus, Rp is checked every time and the penalty increased, whereas R_J can be checked regularly at a little longer interval such as 1 day.

Resource and cost management scheme: The cost and incentive estimation scheme for computational grids is implemented using the J2EE environment. The system is designed as three applications. They are grid server, resource provider and consumer. The grid server application is designed to handle the authentication and scheduling operations. The resource provider application is designed to provide shared resources to other nodes. The consumer application is used to access the resources. The applications are interconnected using the remote method innovation techniques. The resource provider allocates the resources to the consumer with reference to the scheduling scheme provided by the grid server.

The cost estimation and incentive estimation scheme is designed with the priority information. The supply and demand factor is used for the cost estimation process. The cost is increased due to the demand factor and the incentive is increased with reference to the supply factor. The priority factor is decided by the provider and the consumer during the resource request process. The proposed scheme also considers the network delay factors.

Grid server: The grid server application is designed to carry out the administrative operations. The user management and authentication tasks are handled by the grid server. This system integrates the scheduling process in the grid server application with the support of the autonomous information from the provider and n consumer applications. The resource allocation is carried out in the grid server application.

Consumers: In this study, only consideration of computation-incentive jobs is made, where all communication/networking overheads can be ignored .All jobs are independent of one another. The K Consumers altogether have n jobs to execute in time period T. The Consumers first submit job announcement to the computational grid. A Job announcement includes the information of job length and job deadline. Job length is an empirical value assessed as the execution time of the job on the designated standard platform. Job deadline is a Wall clock time by which a consumer desires a job to be finished expressed as a number between 0 and T. Thus, a job with length = 10 and Deadline = 100 means that the job’s execution takes 10 time units on a designated standard computer and it must be finished 100 times units after the common base time 0.

Providers: From the scheduling viewpoint, each resource provider is modeled with three parameters: Capability, job queue and unit price. Capability is the computational speed of the underlying resource, expressed as a multiple of the speed of the standard platform. The job queue of a resource provider keeps an ordered set of jobs scheduled but not yet executed. Each job, once it is executed on a resource, will run in a dedicated mode on that resource

Without time-sharing or preempting, a provider charges for a job according to its unit price and jobs length. Unit price refers to the price that the resource offers for executing to its unit price and job length.

CONCLUSION

We formulate job scheduling in a sustainable market-like computational grid as a double-objective optimization problem to optimize incentives for both consumers and providers. As the problem is at least NP-complete, development of an incentive-based scheduling scheme IB with heuristics, using a P2P decentralized scheduling framework is done, this scheme has the following features: (1) Each consumer or provider elaborate makes scheduling decisions, (2) All scheduling algorithms are local to a resource provider and (3) Three market instruments, that is, job announcement, price and CD, are employed and the former two circulate in the grid.

REFERENCES

Abdelkader, K., J. Broeckhove and K. Vanmechelen, 2008. Commodity resource pricing in dynamic computational grids. Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, Mar. 31-Apr. 4, IEEE Xplore Press, Doha, pp: 422-429. DOI: 10.1109/AICCSA.2008.4493568

Abdelkader, K., J. Broeckhove and K. Vanmechelen, 2009. Economic-based resource management for dynamic computational grids: Extension to substitutable CPU resources. Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, May 10-13, IEEE Xplore Press, Rabat, pp: 1-6. DOI: 10.1109/AICCSA.2009.5069288

Buyya, R., D. Abramson and S. Venugopal, 2005. The grid economy. Proc. IEEE, 93: 698-714. DOI: 10.1109/JPROC.2004.842784

Gupta, R., V. Sekhri and A.K. Somani, 2006. CompuP2P: An architecture for internet computing using peer-to-peer networks. IEEE Trans. Parallel Distributed Syst., 17: 1306-1320. DOI: 10.1109/TPDS.2006.149

Irwin, D.E., L.E. Grit and J.S. Chase, 2004. Balancing risk and reward in a market-based task service. Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, Jun. 4-6, IEEE Xplore Press, pp: 160-169. DOI: 10.1109/HPDC.2004.1323519

Lai, K., L. Rasmusson, E. Adar, L. Zhang and B.A. Huberman, 2005. Tycoon: An implementation of a distributed, market-based resource allocation system. Multiagent Grid Syst., 1: 169-182.

Liu, Y., Z. Zhuang, L. Xiao and L.M. Ni, 2004. A distributed approach to solving overlay mismatching problem. Proceedings of the 24th IEEE International Conference on Distributed Computing Systems, Mar. 26-26, IEEE Xplore Press, pp: 132-139. DOI: 10.1109/ICDCS.2004.1281576

Padala, P., C. Harrison, N. Pelfort, E. Jansen and M.P. Frank *et al*., 2003. OCEAN: The open computation exchange and arbitration network, a market approach to meta computing. Proceedings of the 2nd International Symposium on Parallel and Distributed Computing, Oct. 13-14, IEEE Xplore Press, pp: 185-192. DOI: 10.1109/ISPDC.2003.1267662

Parashar, M. and C.A. Lee, 2005. Special Issue on Grid Computing. Proc. IEEE, 93: 479-484. DOI: 10.1109/JPROC.2004.842744

Sherwani, J., N. Ali, N. Lotia, Z. Hayat and R. Buyya, 2004. Libra: A computational economy-based job scheduling system for clusters. Software Practice Exper., 34: 573-590. DOI: 10.1002/spe.581

Xiao, L., Y. Zhu, L.M. Ni and Z. Xu, 2005. GridIS: An incentive-based grid scheduling. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, Apr. 4-8, IEEE Xplore Press, pp: b65-b65. DOI: 10.1109/IPDPS.2005.237

Zhu, Y., B. Carragher, R.M. Glaeser, D. Fellmann and C. Bajaj *et al*., 2004. Automatic particle selection: Results of a comparative study. J. Struct. Biol., 145: 3-14. PMID: 1506566