Adaptation of Optimal Methods In Resource Utilization, Job Scheduling In Grids

Jhansi Lakshmi Vaddelle
Assistant professor,
Dept of CSE, SRK Institute of Technology,
Enikepadu – 521108

Y.Surekha,D.Sree Lakshmi
Assistant Professors,
Dept of CSE, PVP Siddhartha Institute of Technology
Kanuru - 520007

Abstract— **A Optimal Grid should have Three characterstics : It should not allow resource providers to increase prices continuously, No provider should reject to provide resources to certain type of jobs due to risk associated in executing them, Resource Providers and Consumers are allowed to take autonomous decisions. There will be no force on providers regarding the price for which they should offer their resources, but at the same time the price cannot be increased to any extent by providers because if no consumer is selecting him to offer his resources he should definitely get the price down. The two objectives identified are to schedule all type of jobs for execution and to minimize fairness deviation among resources. We present a scheduling scheme, which utilizes a peer-to-peer decentralized scheduling framework. Here the Greedy nature of provider is taken into account. we need not adjust competition degree explicitly , the provider automatically decreases his competition degree depending upon his capacity and number of jobs he is offered and automatically increases competition degree when jobs offered is more than his capacity. Here we give different levels to jobs to be executed depending upon the application type to which they belong and resource quota is given for different type of jobs while allocating resources by a provider so that starvation of particular type jobs can be avoided.**
Keywords- Optimal Grid, scheduling, Greedy, competition degree;

I. Introduction

Grid computing (or the use of a computational grid) is applying the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data.

Generally nature of providers is to maximize their profits. We consider this nature of providers and allow provider to increase resource prices to some extent so that he will not get dissatisfied and leave the grid. At the same time we impose limitation on the extent to which he increases the resource price and make him to allocate resource quota to different types of jobs so that he cannot escape from handling critical job requests. Here if competition degree is more for a provider means that he will accept jobs even if he is not able to deliver that job in scheduled time. Provider will do so to get more profit. Therefore he accepts number of jobs even though it's not feasible for him to complete them in time. The provider accepts number of jobs and completes those jobs in time for which he gets more profit. This generally happens in all grids because the nature of providers is generally greedy and they want to maximize their profits always. But if they are allowed to do so no consumer will stay in the grid and finally all consumers will leave the grid and grid will collapse. But this will not happen here because as the provider observes that no one is requesting him for resources he will automatically reduce his competition degree and will not accept a job until he really has the feasibility to complete it with in stipulated time.

Similarly if competition degree of provider is less then number of jobs not getting bidded will be more, because provider accepts a job only if it can be completed in time. It means if no provider is free then a consumer does not receive a bid to complete his job from any provider and if this frequently happens number of jobs getting not bidded at all will increase and this leads to consumer dissatisfaction and makes them to leave the grid. This leads to collapse of grid. Therefore competition degree should not be a constant it should get adjusted according to situations so that there will be both consumer and provider satisfaction and the grid sustains for more time.

When the provider is receiving a number of jobs to schedule to his resources he will generally increase per unit price of his resources. But when a provider fails to execute jobs in time gradually number of consumers opting him as provider gets reduced and therefore his price alsogetsreduced.
According to how they schedule computational jobs to resources, computational grids can be classified into two types: controlled and market-like grids. Both 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.

All type of jobs submitted by Consumers in the grid should be executed by resource providers offering their resources so that all consumers get their jobs executed at optimal price. Resource Provider should be independent to set costs for his resources. At the same time price of Provider will be controlling his cost and Competition degree by himself due to competition from Other providers in the Grid. As Resource Quota is allocated for different type of jobs to be executed in the grid some type of jobs will not be starved without being offered resources by any Provider. As a result we get Optimal Methods for Resource Utilization and Job Scheduling in our grid.

II. THE PRICE, RESOURCE RESTRICTIVE SCHEDULING SCHEME

All type of jobs submitted by Consumers in the grid should be executed by resource providers offering their resources so that all consumers get their jobs executed at optimal price. Resource Provider should be independent to set costs for his resources. At the same time price of Provider will be controlling his cost and Competition degree by himself due to competition from Other providers in the Grid. As Resource Quota is allocated for different type of jobs to be executed in the grid some type of jobs will not be starved without being offered resources by any Provider. As a result we get Optimal Methods for Resource Utilization and Job Scheduling in our grid.

We propose a Price, Resource Restrictive scheme employing a P2P decentralized scheduling framework. The scheme is characterized as follows:

1) It should not allow resource providers to increase prices to any extent.
2) No provider should reject to provide resources to certain type of jobs due to risk associated in executing them.
3) Resource Providers and Consumers are allowed to take autonomous decisions.

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. Aside 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 will 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. When 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 provider that receives a job announcement may bid for the job. We want to realize the complete competition among all the providers based on two considerations. First, 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, no matter where their geographical locations are. Second, 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 researchers 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.

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 checks if it has any resources allocated to the
Job of that particular type. If not go to step 6.

Step 3. Each provider, upon receiving a job announcement checks its competition degree.

- If competition degree is less then it 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.
- If competition degree is more provider accepts the job even if its' not feasible to complete the job in time.

Step 4. 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 5. 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 missing any potential bid and also making decisions as soon as possible. In our experiments, we choose the average execution time as the waiting interval or synthetic workloads and 10 seconds for real workloads. Both are rather conservative values so that the performance evaluation results will not be favorably skewed.

Step 6: End

III. ALGORITHMS

A. Resource Restriction Algorithm

Jobs to be executed are of different types each consuming different type of resources , resource time and Input/output time. Depending upon requirements for their execution jobs are categorized in to different types. For each and every job type resource quota that is percentage of resources to be utilized for a particular job out of all the resources available with the provider is calculated for all providers. So, when a particular job announcement is received by the provider he checks if any resource quota is available for execution of the job whose announcement is received and if so the provider may or may not bid for the job depending upon his competition degree value.

Step 1: Different types of jobs J1,J2….Jn that can be handled by providers in grid are listed.
Step 2: calculate Resource Quotas Q1,Q2,Q3...Qn for every particular type of job.
Step 3:for all providers calculate amount of resources for particular job type.
Quota for job type Ji=(resources with provider) * Qi
Step 4: End

B. Optimal Scheduling Algorithm

When a particular job announcement is received by the provider , he will identify the type of the job and check whether any resource quota is available for that particular job type. If resources for that type of job are not available then the provider will not bid for that job. If resource quota is available for that type of job then competition degree cd is checked . If competition degree is less then provider will bid for that job only if he can complete the job within deadline. If competition degree is more than provider will bid the job even if it is not possible for him to complete the job in time.

Step 1: Check if Resource Quota is available for the received job type. If no go to step 6.
Step 2: if cd < 0.5 then
Go to step 3
Else
Go to step 4
Step 3: find if the job can be completed in deadline.
If (yes) then
Job is accepted.
Else if job can be completed by rescheduling all jobs of provider such that none of already accepted jobs misses deadlines then job is accepted.
Else
Job is rejected .Go to step 6
Step 4:Accept job even if it cannot be completed in deadline.
Step 5:compute total-price = (length of job) x (unit price of provider)
Step 6:Send bid to consumer.

C. Optimal Cost Restriction Algorithm

Generally a greedy provider increases his resource cost as number of jobs being offered to him is more, but in our optimal scheduling scheme as the provider observes that he is receiving less number of job requests compared to other providers with relatively same capacity as him he automatically reduces his resource cost. Inevitably, all the providers need to know some global information. In our algorithm, we assume 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.

//Over a time period t
Step 1: find total execution time of all the jobs executing in the grid let it be E_t.
Step 2:If (E_t x Cp)/(Ap) < E_p then // jobs less offered than available capacity.
Unit price = unit price-x
Else
Unit price = unit price + x
Where E_t =execution time of all the jobs in the grid.
Cp=capacity of provider.
Ap=capacity of all providers.
E_p=Execution time of jobs allocated to provider P.
Step 3 :End

D. Greedy 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, we design the Greedy algorithm to adjust competition degree .

Step 1 : G1= length of jobs offered to provider/ length of jobs whose announcements are received by provider

Step 2 :G2= Capacity of the provider / sum of Capacities of all providers

Step 3: IfG1 >G2 then
Increase competition degree by an amount x.
CD=CD + x
// as number of jobs received by provider are more than his capacity.
Step 4: If G1< G2 then
Decrease Competition Degree.
CD=CD-x.
Step 5: end.

If number of jobs received by a provider is more than his capacity then the provider will increase his unit price per resource as he is having number of consumers opting him as provider. If number of jobs received by a provider is less than his available capacity even when there is sufficient number of consumers in the grid then he automatically reduces his unit price per resource to attract consumers towards him .

IV. CONCLUSIONS

Here we are not considering network overheads that have to be faced when sending job broadcasts to different providers and when receiving bids from different providers by consumers and also when sending jobs and receiving results from providers. This is a important factor consuming most of amount of time and leading to delays .For successful implementation of above algorithms we need to give a correct estimate of amount of time a resource will be needed which again requires a correct estimation algorithm for that purpose. Estimation of type of particular job to be executed is also a critical task depending upon which the resources will be scheduled for the execution of that job. If all the above factors are satisfied that is no network delays, resource needed time is estimated correctly, job type is identified correctly then our algorithms works efficiently for all real workloads.

ACKNOWLEDGMENT

I express my deep sense of gratitude to one and all who extended their professional help in making my theme into reality and techful paper

REFERENCES

[1] A. Anderson, A. Nadalin, B.Parducci, D. Engavatow, H.Lockhart, M. Kudo,P. Humenn, S. Godik, S. Abderson, S. Crocker, and T. Moses. eXtensibleAccess Control Markup Language (XACML) Version 1.0. Technical report,OASIS, 2003.

[2] Jorge Andrade and Jacob Odeberg. HapGrid: a resource for haplotype reconstructionand analysis using the computational Grid power in Nordugrid.*HGM2004: New Technologies in Haplotyping and Genotyping*, April 2004.

[3] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne,J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification (wsagreement).Technical report, Global Grid Forum, 2005.

[4] A. Barmouta and R. Buyya. Gridbank: A grid accounting services architecture(gasa) for distributed systems sharing and integration. In *Int. Parallel andDistributed Processing Symposium (IPDPS'03)*, Nice, France, 2003. IEEE.

[5] Proc. IEEE, special issue on grid computing, M. Parashar andC.A. Lee, eds., vol. 93, no. 3, pp. 479-714, 2005..Foster and C. Kesselman, "Globus: A Metacomputing InfrastructureToolkit," Int'l J. High-Performance Computing Applications,vol. 11, no. 2, p. 115, 1997.

[6] [M. Litzkow, M. Livny, and M. Mutka, "Condor: A Hunter of IdleWorkstations," Proc. Eighth Int'l Conf. Distributed ComputingSystems (ICDCS '88), pp. 104-111, 1988.

[7] J. Chapin, D. Katramatos, J. Karpovich, and A. Grimshaw,"Resource Management in Legion," Future Generation ComputerSystems, vol. 15, nos. 5-6, pp. 583-594, 1999.

[8] F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman,S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen,N. Spring, A. Su, and D. Zagorodnov, "Adaptive Computing on the Grid Using Apples," IEEE Trans. Parallel and DistributedSystems, vol. 14, no. 4, pp. 369-382, Apr. 2003.

[9] R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economy,"
Proc. IEEE, vol. 93, no. 3, pp. 698-714, 2005.

[10] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/G: AnArchitecture of a Resource Management and Scheduling Systemin a Global Computational Grid," Proc. Fourth Int'l Conf. High-Performance Computing in the Asia-Pacific Region (HPC Asia), 2000.

[11] O. Regev and N. Nisan, "The POPCORN Market: An OnlineMarket for Computational Resources," Proc. First Int'l Conf.Information and Computation Economies (ICE '98), pp. 148-157, 1998.

[12] Wolski, J.S. Plank, T. Bryan, and J. Brevik, "G-Commerce:Market Formulations Controlling Resource Allocation on theComputational Grid," Proc. 15th Int'l Parallel and DistributedProcessing Symp. (IPDPS '01), p. 8, 2001.

[13] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B.A. Huberman,"Tycoon: An Implementation of a Distributed, Market-BasedResource Allocation System," Multiagent and Grid Systems, vol. 1,no. 3, pp. 169-182, 2005.

[14] P. Padala, C. Harrison, N. Pelfort, E. Jansen, M.P. Frank, and C.Chokkareddy, "OCEAN: The Open Computation Exchange and Arbitration Network, a Market Approach to Meta Computing,"
Proc. Second Int'l Symp. Parallel and Distributed Computing,pp. 185-192, 2003.

[15] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra:A Computational Economy-Based Job Scheduling System forClusters," Software: Practice and Experience, vol. 34, no. 6,pp. 573-590, 2004.

[16] D.E. Irwin, L.E. Grit, and J.S. Chase, "Balancing Risk and Rewardin a Market-Based Task Service," Proc. 13th Int'l Symp. High-Performance Distributed Computing (HPDC '04), pp. 160-169, 2004.

[17] R. Gupta, S. Sekhri, and A.K. Somani, "CompuP2P: An Architecturefor Internet Computing Using Peer-to-Peer Networks,"IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 11,pp. 1306-1320, Nov. 2006.

[18] F. Yun, C. Jeffrey, C. Brent, S. Stephen, and V. Amin, "SHARP: AnArchitecture for Secure Resource Peering," Proc. 19th ACM Symp.Operating Systems Principles (SOSP '03), pp. 133-148, 2003.

[19] Y. Zhu, L. Xiao, L.M. Ni, and Z. Xu, "Incentive-Based P2PScheduling in Grid Computing," Proc. Third Int'l Conf. Grid andCooperative Computing (GCC '04), p. 209, 2004.[20] L. Xiao, Y. Zhu, L.M. Ni, and Z. Xu, "GridIS: An Incentive-BasedGrid Scheduling," Proc. 19th IEEE Parallel and Distributed ProcessingSymp. (IPDPS '05), p. 65, 2005.

AUTHORS PROFILE

V Jhansi Lakshmi received MCA degree in year 2003 from Acharya Nagarjuna University, and currently doing M.Tech at JNTU- Kakinada University.
She is having work experience of 7 years in teaching field.She is currently working as a Asst. Professor in SRK Institute of Technology, Vijayawada.
uthors Profile …