A Two Way Scheduling Approach for Effective Resource Scheduling in Grid

Sandeep Kumar
Student, M.Tech (CSE), Department of CSE/IT
Lovely Professional University,
Phagwara, Punjab, India.

Manisha Bhardwaj
Asstt. Professor, Department of CSE/IT
Lovely Professional University,
Phagwara, Punjab, India.

***Abstract*— Grid computing is one of the major computational area in which number of resources, resource providers are associated to share the services and to achieve high utilization of the resources. In such case the major challenge is about the sequence of providing the services to the customers. In this paper we have studied the different scheduling techniques used by the earlier researches as well as a new scheduling technique is suggested. The presented scheduling mechanism will not only provide the better utilization of resources but also avoid the major scheduling problem called starvation. The scheduling criteria is based on two approaches one is best first search and other is the dynamic change in priority of the process.**

***Keywords*- Resource, Grid, Scheduling, Prioritization, Best First Search**

I. INTRODUCTION

In most organizations, computing resources are underutilized. Most desktop machines are busy less than 25% of the time (if we consider that a normal employee works 42 hours a week and that there are 168 hours per week) and even the server machines can often be fairly idle. Grid computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage. The easiest use of Grid computing would be to run an existing application on several machines. The machine on which the application is normally run might be unusually busy, the execution of the task would be delayed. Grid Computing should enable the job in question to be run on an idle machine elsewhere on the network. From this point of view, Grid Computing looks like a great answer to numerous problems without prodigious management requirements. Convenience does not always imply simplicity and grid enabled applications have to meet two prerequisites: first, the application must be executable remotely and with low additional operating cost; second, the remote machine must meet any special hardware, software, or resource requirements needed to run the application. Hence, remote computing is not only a matter of sending jobs to distant machines but also choosing the correct machine that has all the required software and hardware. Even though Computing Power (CPU) can be assumed to be the most commonly shared resource, it belongs to a large set of underutilized resources that are to be shared on the network such as storage capacity, software, internet connection and the like which also need to be considered.

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.

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.

A. Grid Computing

Computers have been proven to be very efficient to solve complex scientific problems. They are used to model and simulate problems of a wide range of domains; for instance medicine, engineering, security control and many more. Although their computational capacities have shown greater capabilities than the human brain to solve such problems, computers are still used less than they could be. One of the most important reasons to this lack of use of computational power is that, despite the relatively powerful computing environment one can have, it is not adapted to such complicated computational purposes.

Many of the basic ideas behind the Grid have been around in one form or other throughout the history of computing. For example, one of the "novel" ideas of the Grid is sharing computing power. Nowadays, where most people have more than enough computing power on their own PC, sharing is unnecessary for most purposes. But back in the sixties and seventies, sharing computer power was essential. At that time, computing was dominated by huge mainframe computers, which had to be shared by whole organizations. In 1965 the developers of an operating system called Multics (an ancestor of Unix, which in turn is an ancestor of Linux - a popular operating system today) presented a vision of "computing as a utility" - in many ways uncannily like the Grid vision today. Access to the computing resources was envisioned to be exactly like water, gas and electricity - something which the client connects to and pays for according to the amount of use. Ironically, "utility computing" is all the rage again these days, and used more or less as a synonym for the Grid by some people. So, yes, there is a certain amount of "reinventing the wheel" going on in developing the Grid. However, each time the wheel is reinvented, it is reinvented in a much more powerful form, because computer processors, memories and networks improve at an exponential rate which is associated with Moore's law. Because of the huge improvements of the underlying hardware (typically more than a factor of 100x every decade), it is fair to say that reinvented wheels are qualitatively different solutions, not just small improvements on their predecessor.

B. Scheduling in Grid

The conventional resource management schemes are based on relatively static model that have centralized controller that manages jobs and resources accordingly. These management strategies might work well in those scheduling regimes where resources and tasks are relatively static and dedicated. However, this fails to work efficiently in many heterogeneous and dynamic system domains like Grid where jobs need to be executed by computing resources, and the requirement of these resources is difficult to predict. The complex, heterogeneous and dynamic systems present new challenges in resource management such as: scalability, adaptability, reliability and fault tolerance. This poses numerous research questions like[11]:

Grid is a dynamic environment where location, type and performance of components are constantly changing; hence their computational requirement varies over time. How to adapt to such conditions when computational capacities fluctuate in high ratio? One way to adapt to the changes as the availability of resource may fluctuate due to connection/disconnection of computing resources is to use autonomous entities.

In Grid environment, computing nodes can join and leave the system at any time without any dedicated commitment, another issue concerned for GRID management is how to tolerate such failures and recover from them when crashes occur? How to re-allocate task and resource automatically? One approach for solving this problem is to use robust entities.

How to manage the intricacy, complexity, performance analysis in such dynamic system where nothing is predictable? For manageability it is useful to use intelligent entities.

II. LITERATURE SURVEY

A work is presented by Raj Kumar Buyya to combine multiple markets across the grid environment. It was a grid architecture the include market agent, market server, market resource broker, market trader and the scheduler. It allow the scheduling and negotiation also. David E. Irwin presented bid market environment along with value added scheduling. In this paper opportunity and the risk factor is included with scheduling process. An incentive based grid scheduling was proposed by Lijuan Xiao. The author implemented the approach in P2P decentralized network. It provides a better resource sharing machismo with fair allocation to the customer

Chia-Hung also present the distributed grid market for job scheduling. It was based on market agents to control the workflow. It will also check for confliction of resources. It uses shorest job first scheduling for job assignment. Li Chunlin presented a utility based job assignment scheme in market grid. It gives an iterative approach to serve user requrest based on utilization. It also correlate the pricing along with resource allocation.

Kyong Hoon Kim presented a new model for market based resource allocation in grid network. It uses the QOS based adaptive resource allocation. The scheduling machinism is used for the resource allocation. (2006) Kai Shen presented a cost and opportunity based algorithm to perform resource scheduling. It is the extension of budget based algorithms. The method covers the concept of opportunity along with risk assessment (2007)

Furo-cho presented a economic scheduling system based on market model. It uses the lexicographic and Euclidean Distance algorithm for scheduling. The algorithm will improve the quality of service for job scheduling. (2007) Zhu Tan also presented a market-based Grid resource allocation mechanism. This paper present a novel Stable Continuous Double Auction (SCDA). the SCDA delivers continuous matching, high efficiency and low cost, allied with low price volatility and low bidding complexity.(2007) Lijuan Xiao present an incentive-based scheduling scheme, which utilizes a peer-to-peer decentralized scheduling framework, a set of local heuristic algorithm. it allow resource providers and resource consumers to make autonomous scheduling decisions, and both parties of providers and consumers must have sufficient incentives to stay and play in the market.(2008). Qi Weiyi presented a new model for resource sharing called DSM(Differential Service Model). It uses the Strictness Factor, Greediness Factor and Credit Score to describe the characteristics of the users and resources, like service requirement and service level. A series of related strategies, including resource evaluating, job risk exposure rating, operation adjustment etc., have been developed to conduct the market players' behavior. (2008)

Bing Tang presented a dynamic resource price-adjusting (RPA) strategy in computational Grid. he new algorithm improves the performance of Grid in aspect of increasing the task accomplishment ratio and solving the problem of load imbalance.(2008) Laiping Zhao presented an approach according to customer interest. It will schedule the resources if a person like it and offer such resources to him. It improve the HRED algorithm using hierarchical mechanism and resource selection mechanism, which cooperate with resource providers' trust degree, making it more suitable, practical and credible for Grid market.(2009)

Bin Wu work on dynamic and autonomic characteristics of grid market. He considers resources set availability and can be extended easily. It utilizes statistics method to describe resource availability and improves the availability of resources set effectively. Then a resources composing based batch scheduling algorithm framework is proposed.(2009). Ang Li, Nianming Yao presented Min-Min scheduling algorithm. This paper consider both cost and makespan, and focus on scheduling parallel tasks from Grid users considering a commodity market. (2010) Amritava Chaudhuri present an effective scheduling of job and resource plays an essential role to optimize and enhance the quality of services provided to the service consumers by the service providers. It provides optimal model for scheduling of grid entities among an assortment of conventional methods of job and resource scheduling and concurrent execution on multiple market based economic scheduling models.(2010)

III. PROPOSED WORK

The effectiveness of the cache operation is based on a property of computer programs called locality of reference or locality principle. Analysis of programs shows that most of the program time is spent on executing many instructions repeatedly. These instructions may be a simple loop, nested loops, or a few procedures that repeatedly call each other. Many instructions in localized areas of the program are executed repeatedly during some period of time, and the remainder of the program is accessed relatively infrequently. This is referred to as locality of reference.

Locality of reference happens in two ways: temporal and spatial. Temporal locality means that a recently referenced memory word will be referenced again very soon. Spatial locality means that memory words close to recently referenced memory word will be referenced soon.

Temporal locality happens when server executes iterative loops and calls to subroutines. Spatial locality is seen when the server performs operations on tables and arrays.

The memory circuitry is designed to take advantage of the locality of reference. The temporal locality suggests that whenever a word is first needed, it should be brought to cache where it will hopefully remain until it is needed again. The spatial locality suggests that instead of fetching just one word from the main memory to the cache, several words adjacent to the needed word are also fetched into the cache.

The proposed work is about to define a cache data replacement scheme based of frequency analysis. According to this scheme most frequent data items will be kept in cache. As the most required data items are in cache itself the system will improve the efficiency as well as will improve the hit ratio. The system is presenting an efficient and reliable data replacement scheme in cache memory.

To work with the proposed system we have to simulate the service allotment in grid market based architectural environment in a programming environment. Here the work is basically about the improvement over the process allotment. The presented work will show the access to the cpu with access time and the relative parameters.
As the architecture build up the next work is to define any of the existing data replacement algorithm in cache memory. The work is here based on the access of data from internal cache to cpu. The analysis will be done on this exsiting approach.

Now the proposed work will be implemented where we will a list to store the frequencies of the dataset. Now according to the user access the frequency of data items will be changed. And the data values in cache with low frequency will be replaced by the high frequency data items.

The presented scheduling mechanism is given as under

```
Step 1:  Intialize the Page Cache of Size N
Step 2:  Perform User Input Page called Pagei
Step 3:  if count(inputpage)<=N
         {
         Include the Page in Queue
         }
Step 4:  else
         {
         If Pagei   BelongsTo  Queue
         {
         Frequency(Pagei)=Frequency(Pagei)+1
         }
Step 5   Else
         {
```

5.1 Find the Low Frequency Page From Cache called LFrequPage

5.2 If Count(LFrequPage)=1

{

Replace this page by pagei

}

5.3 Else

{

Find the Recently Visited Page called RecVisited from LFrequPage and replace it in cache by pagei

}

}

Step 6 Exit

}

IV. CONCLUSION

The presented work is about to improve the effectiveness of the grid system along with reliability. In this system we will keep the most frequent data items in cache by estimating the data frequency. As the most required data items are keep in the cache itself, it will improve the hit ratio and improve the reliability of data access. The system will give the better service allotment such way that the starvation will not occur over the system.

REFERENCES

[1] Mehdi Bahrami and Ahmad Faraahi, "AGC4ISR,New Software Architecture for Autonomic Grid Computin", 2010 International Conference on Intelligent Systems, Modelling and Simulation 978-0-7695-3973-7/10 © 2010 IEEE

[2] H.M. Faheem, "Accelerating Motif Finding Problem using Grid Computing with Enhanced Brute Force", ISBN 978-89-5519-146-2

[3] Inderpreet Chopra, "Analysing the need for Autonomic Behaviour in Grid Computing", 978-1-4244-5586-7/10/ 2010 IEEE

[4] Peng Zhang Ming Chen Peng-ju He, "The Study of interfacing Wireless Sensor Networks to Grid Computing based on Web Service", 2010 Second International Workshop on Education Technology and Computer Science, 978-0-7695-3987-4/10 © 2010 IEEE

[5] Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood and Alan Oxley, "Hybrid Resource Allocation Method for Grid Computing", Second International Conference on Computer Research and Development 978-0-7695-4043-6/10 © 2010 IEEE

[6] Zhen-chun HUANG, Shi-feng SHANG, "Design and Implementation of SCO-GADL – a Scientific Computing Oriented Grid Workflow", IEEE 2010.

[7] Manoj Kumar Mishrai, Raksha Sharmai, "A Memory-Aware Dynamic Job Scheduling Model in Grid Computing", 201O International Conference On Computer Design And Appliations (ICCDA 2010) 978-1-4244-7164-51© 2010 IEEE

[8] Naidila Sadashiv, "Cluster, Grid and Cloud Computing: A Detailed Comparison", The 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. SuperStar Virgo, Singapore

[9] M Victor Jose, "Object Based Grid Architecture for Enhancing Security in Grid Computing", Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011) 978-1-61284-653-8/11/ ©2011 IEEE

[10] CCITT Recommendation, X.509: The Directory – Authentication Framework. 1988.

[11] Chan, S., Grid Security at NERSC/LBNL, Globus Security Workshop, 2004. http://grid.ncsa.uiuc.edu/gw04-security/GW04-SecWkshp-nersc.ppt

[12] Dierks, T. and Allen, C., The TLS Protocol Version 1.0, RFC 2246, IETF, 1999.

[13] Globus Simple CA, http://www.globus.org/security/simple-ca.html, 2004.

[14] Peter Gutmann, "Plug-and-Play PKI: A PKI your Mother can Use", Usenix Security Symposium, 2003.