

A Review: Process Scheduling Approach for Starvation Improvement with Time Delay Analysis in Grid Resources Allocation

Er. Himanshu Jain¹, Kavita Khatkar²

Student, CSE, JCDV, Sirsa, India¹

Asst Professor, CSE, JCDV, Sirsa, India²

Abstract: This paper is about the study of the Grid Resources Allocation for tackle the problem of the Starvation and scheduling the resources efficiently for reduces the time delay in processing. The work has been proposed for improve the steps of scheduling during resources allocation. The existing proposed work is the FCFS (First Come First Serve) scheduling mechanism which has been improved by implementing the priority rule. In existing work a deadline gap is been analyzed to set the prioritization. In this research work, the cache contents are maintained based on the current requirement and the previous history. The work is very similar to the recommender system that will analyze the frequency of the data requirement in previous history. The system will perform the improvement over the best first algorithm along with delay analysis. The system is about to reduce the delay and to minimize the starvation over the process allotment.

Keywords: FCFS, Grid Computing, Resource Allocation, Scheduling, Starvation .

I. INTRODUCTION

Grid computing 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.

Grid computing requires the use of software that can divide and farm out pieces of a program to as many as several thousand computers. Grid computing can be thought of as distributed and large-scale cluster computing and as a form of network-distributed parallel processing. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration (in which case it is also sometimes known as a form of peer-to-peer computing). Job Scheduling is choosing the most suitable resource for a job to complete its execution either in terms of waiting time, turnaround time or cost. Scheduler is used to manage the jobs and resources. Scheduler performs two main functions; First scheduler selects the appropriate computational resource for the job and then assigns the resource to the jobs. Job Scheduling is done in order to make the efficient use of resources and for seamless execution of jobs. The main objective of scheduling is to reduce the completion time of an application by properly allocating the jobs to the processors.

Need of Grid

1. A biochemist exploits 10, 000 computers to screen 100,000 compounds in an hour.
2. 1,000 physicists worldwide pool resources for peta-op analyses of petabytes of data.
3. Civil engineers collaborate to design, execute, & analyze shake table experiments.

4. Climate scientists visualize, annotate, & analyze terabyte simulation datasets.
5. An emergency response team couples real time data, weather model and population data.
6. A multidisciplinary analysis in aerospace couples code and data in four companies.
7. A home user invokes architectural design functions at an application service provider.
8. Scientists working for a multinational soap company design a new product.
9. A community group pools members PCs to analyze alternative designs for a local road.

Grid Classification:

A grid can be classified into three categories:

- a. Computational grid,
- b. Data grid and
- c. Service grid.

The categories of grid systems has been shown in the Grid. A computational grid provides access to huge pool of shared processing power suitable for high throughput applications and computation intensive computing. Depending on how the capacity is utilized, computational grid is classified into Distributed Supercomputing grid and High Throughput grid. A supercomputing grid executes applications on multiple machines in parallel to reduce the computation time of a task.

A data grid provides an infrastructure for synthesizing information from data archives such as digital libraries or data warehouses that are distributed in a wide area of network. European data grid project and Globus are data grid initiatives, working on developing large scale data

organizations. A service grid provides services that cannot be accomplished by any single machine. The service grid further categorized into on-demand grid, collaborative grid and multimedia grid systems.

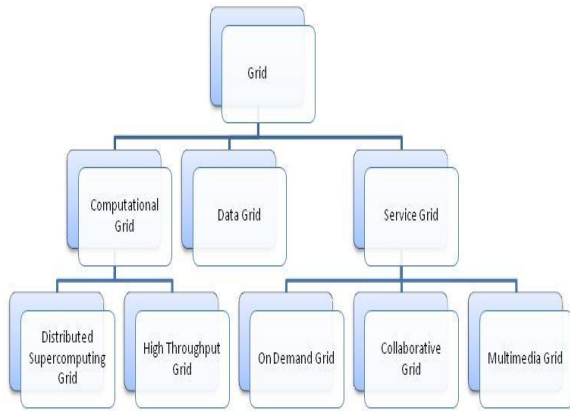


Figure 1 Grid System Hierarchy

Grid Architecture Description

The goal in describing the Grid architecture is not to provide a complete enumeration of all required protocols (and services, APIs, and SDKs) but rather to identify requirements for general classes of component. The result is an extensible, open architectural structure within which can be placed solutions to key VO requirements. The architecture and the subsequent discussion organize components into layers. Components within each layer share common characteristics but can build on capabilities and behaviors provided by any lower layer. In specifying the various layers of the Grid architecture, there is need to follow the principles of the hourglass model. The narrow neck of the hourglass defines a small set of core abstractions and protocols (e.g., TCP and HTTP in the Internet), onto which many different high-level behaviours can be mapped (the top of the hourglass), and which themselves can be mapped onto many different underlying technologies (the base of the hourglass).

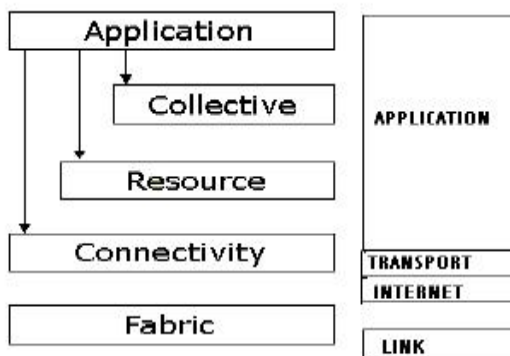


Figure 2 Architecture and the subsequent discussion organize components into layers.

Advantages of Grid Computing

1. It can solve larger, more complex problems in a shorter time.
2. Easier to collaborate with other organizations
3. Make better use of existing hardware
4. No need to buy large six figure SMP servers for applications that can be split up and farmed out to smaller commodity type servers. Results can then be concatenated and analyzed upon job(s) completion.
5. Much more efficient use of idle resources. Jobs can be farmed out to idle servers or even idle desktops.
6. Upgrading can be done on the fly without scheduling downtime. Since there are so many resources some can be taken offline while leaving enough for work to continue.

II. LITERATURE REVIEW

Author has been explained the different types of computing along with its architecture. The last decade has seen a substantial increase in commodity computer and network performance, mainly as a result of faster hardware and more sophisticated software. A number of teams have conducted experimental studies on the cooperative use of geographically distributed resources unified to act as a single powerful computer. This new approach is known by several names, such as meta computing, scalable computing, global computing, Internet computing, and more recently Grid computing. In this article, we define grid computing, first, we review the “Need of Grid computing”. The author has been explained the Grid architecture, in which protocols, services, application programming interfaces, and software development kits are categorized according to their roles in enabling resource sharing. Finally, we discuss benefits and application of grid computing [1].

Author has been explained the concept of the Grid and the different steps of the grid. They explained that the Grid computing is concerned with the sharing and use of resources in dynamic distributed virtual organizations. The dynamic nature of Grid environments introduces challenging security concerns that demand new technical approaches. In this brief overview we review key Grid security issues and outline the technologies that are being developed to address those issues. We focus on works done by Globus Toolkits to provide security and also we will discuss about the cyber security in Grid [2].

Author has been assimilated the knowledge about the Grid Computing. They explained that the Grid Computing has become another buzzword after Web 2.0. However, there are dozens of different definitions for Grid Computing and there seems to be no consensus on what a Grid is. This paper describes about Grid Computing. In various ways we are trying to explain grid computing along with its architecture and the standards available for grid computing. Then at last we have discussed about the benefits of grid computing [3].

Author explained the concepts of the Grid Computing and Cloud Computing. Cloud Computing is not a completely

new concept; it has intricate connection to the relatively new but the established Grid Computing paradigm, and other relevant technologies such as utility computing, cluster computing, and distributed systems in general. The service oriented, loose coupling, strong fault tolerant, business model and ease use are main characteristics of cloud computing. Grid computing in the simplest case refers to cooperation of multiple processors on multiple machines and its objective is to boost the computational power in the fields which require high capacity of the CPU. In grid computing multiple servers which use common operating systems and software have interactions with each other. Grid computing is hardware and software infrastructure which offer a cheap, distributable, coordinated and reliable access to powerful computational capabilities. This paper strives to compare and contrast Cloud Computing with Grid Computing from various angles and give insights into the essential characteristics of both [4].

III. PROBLEM FORMULATION

In the case of computational grids the consumers and providers share their resources and schedule the decision. The work is to increase the success rate of job execution and to minimize the fairness deviation among resources and in this work we propose to achieve both this target simultaneously. The proposed work is very much inspired from a recommendation system. In this system the new decision making will be taken place respective to delay analysis based thresholding mechanism. The proposed work is about to keep only the most required data items in cache. The requirement depends on the frequency of the reuse of data item. The work is about to identify the frequency of each data items from previous history. For this analysis the improved inverted list is suggested here. It will also predict the group of data items that should be kept in a sequence to get better optimization. The system will perform the improvement over the best first algorithm along with delay analysis. The system is about to reduce the delay and to minimize the starvation over the process allotment.

Causes of Starvation:

Starvation is caused by failure to allocate some resource to a process, so to find the causes we must inspect the policies which the system uses in handling resources. Here are some possibilities.

- Processes hand on resources to other processes without control. If processes queue for a resource, and the resource is always handed on to the next process in the queue, it is essential that every process awaiting the resource must be placed in the queue.
- Processes priorities are strictly enforced. If a process of worse priority requires a resource in competition with a constant stream of processes of better priority, it might wait for ever.
- Random selection is used. If processes awaiting service are not queued, but an arbitrary process is selected

whenever the resource becomes available, it is possible for some processes to wait for a very long time.

- Not enough resources. This is commonly the real problem, so far as physical resources are concerned, though as its solution costs money it might be a hard one to solve.

IV. OBJECTIVES

The proposed system will cover the following objectives:

1. To Analyze the Existing Grid Resources Allocation Process.
2. To Understand the Concept of the Starvation in Processes Allocation under Grid.
3. To Design an effective algorithm for reduce the time delay.
4. To develop an efficient job scheduling algorithm for resource allocation.
5. To implement the proposed algorithm to see the results outcome.
6. Generate Results.

Algorithm:

MET (Minimum Execution Time): MET assigns each task to the resource with the best expected execution time for that task, no matter whether this resource is available or not at the present time. The motivation behind MET is to give each task its best machine. This can cause a severe load imbalance among machines. Even worse, this heuristic is not applicable to heterogeneous computing environments where resources and tasks are characterized as consistent, which means a machine that can run a task faster will run all the other tasks faster.

V. PROPOSED METHODOLOGY

The hypothesis defines the research question around which the complete research is performed. The hypothesis actually setup a direction for the researcher to perform is work in an organized way. During the research the answer of this research question are driven positively or negatively. In this proposed work we also setup up some research question.

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 existing approach. Starvation is a problem that occurs when some high priority process always get CPU quantum and some other process never get executed.

It will not occur if the algorithm will be design properly because all the processes get time in order they come, there is no priority for them, but as far as the non-

preemptive shortest job is concerned, if shortest jobs continue arriving, long running processes will always be waiting for CPU time and hence the starvation problem.

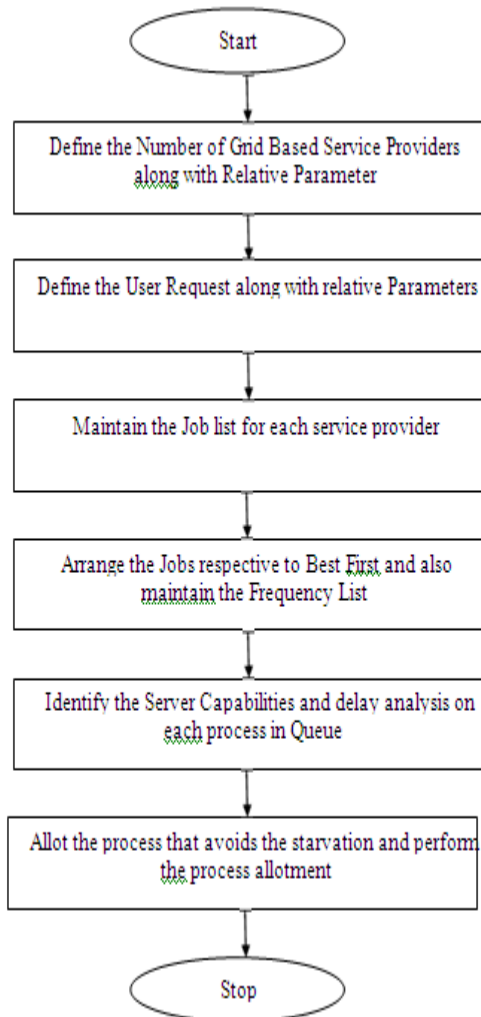


Fig 3 Grid Resource Allocation

The proposed work is about to define a cache data replacement scheme based on 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. The proposed work will be implementing 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. If starvation is possible in a system, then a process which enters the system can be held up for an arbitrary length of time. To avoid starvation, it is often systems resources to be shared fairly. It isn't much use because it doesn't define what mean by fairly, so it's really more of a description of the problem than a contribution to its solution. Its more

constructive to investigate the paths by which a system can reach a livelocked state, and try to control them.

Solution:

Cures for starvation are in general based on means of ensuring that the conditions for starvation cant happen. Here is a selection.

- There must be an independent manager for each resource, which must manage all allocations of its resource; this will guarantee that processes don't just pass resources around between themselves without making them available for general allocation.
 - Strict priorities should not be enforced. A poor priority should be regarded as a weak claim, but not an overridable claim. There are at least two ways to achieve this end : - Improve the priority of a waiting process with time; then even a process with poor priority which has waited for a long time will eventually be able to compete successfully with a newly-arrived process of much better basic priority.
 - Avoid random selections, uncontrolled competition, etc.
- It is very unusual for random resource allocation techniques to have any intrinsic merit. After all, if a random technique will work, it doesn't matter which process receives the resource, so you might as well queue them and give the resource to the process at the head of the queue. If anything, the queue overhead is less, and you win by the closer approximation to a functional system.
- Provide more resources. This is the only satisfactory solution to continued congestion when demand approaches supply.

VI. PRINCIPLES OF GRID COMPUTING

Grid computing operates on these technology principles.

a. Standardization

IT departments have enjoyed greater interoperability and reduced their systems management overhead by standardizing on operating systems, servers, storage hardware, middleware components, and network components. Standardizing also helps reduce operational complexity in the data center by simplifying application deployment, configuration and integration.

b. Virtualization

Virtualizing IT resources means that applications are not tied to specific server, storage, or network components and can use any virtualized IT resource. Virtualization occurs through a sophisticated software layer that hides the underlying complexity of IT resources and presents a simplified, coherent interface used by applications and other IT resources.

c. Automation

Because of the potentially large number of components—both virtual and physical—grid computing demands large-scale automation of IT operations. Each component requires configuration management, on-demand provisioning, top-down monitoring, and other management tasks. A grid management solution must ensure that infrastructure cost savings do not evaporate as a result of hiring additional staff to manage the grid. IT

administrators need a top-down view from the end-user or application level so they can effectively measure service levels and proactively resolve problems. Combining these capabilities into a single, automated, integrated solution for managing grids gives organizations a maximum return on their grid investment.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have been proposed the Grid Resources Allocation basics and the objectives has been defined for handle the starvation problem and time delay as well. The proposed work's implementation part is not explained in this paper.

The implementation part will be covered in the next paper, which will demonstrate the real working of proposed algorithm.

REFERNCES

- [1] Priyanka C A, "Survey Paper on Grid Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 2013 (Software Engineering).
- [2] Neha Mishra, Ritu Yadav, "SECURITY ISSUES IN GRID COMPUTING", International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1, February 2014.
- [3] Shruti N. Pardeshii , Chitra Patil² ,Snehal Dhumale, "Grid Computing Architecture and Benefits", International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013 I ISSN 2250-3153.
- [4] Prabha Sharma, "Grid Computing Vs. Cloud Computing", International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 6 (2013), pp. 577-582.
- [5] Yang Bo, Wang Xun, "Research on the performance of grid computing for distributed neural networks", International Journal of Computer Science and Network Security, VOL.6 No.4, April 2006.
- [6] Harmeet Kaur, Kamal Gupta, "Challenges in Grid computing", International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 2 Issue3 pp 141-144 June2013
- [7] Yuvraj S. Sase , Pratik A.Yadav, "Big Data Implementation Using Hadoop and Grid Computing", International Journal of Innovative Research in Science, Engineering and Technology, Volume 3, Special Issue 4, April 2014
- [8] Raksha Sharma, Vishnu Kant Soni, "A Survey of Job Scheduling and Resource Management in Grid Computing", World Academy of Science, Engineering and Technology Vol:4 2010-04-22.
- [9] Seyyed Mohsen Hashemi, Amid Khatibi Bardsiri, "Cloud Computing Vs. Grid Computing", VOL. 2, NO.5, MAY 2012
- [10] Kumaresh.V.S, Prasad.S, "Multilevel Queue-Based Scheduling for Heterogeneous Grid Environment", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012 ISSN (Online): 1694-0814 www.IJCSI.org
- [11] Sandeep Kaur , Sukhpreet kaur, "Efficient Load Balancing Grouping based Job Scheduling Algorithm in Grid Computing", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 4, July – August 2013