INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

A Review of Scheduling Algorithms in Grid Computing

Mamta Sharma[1] , Manish Mahajan[2]

[1]M.Tech, Chandigarh Engineering Collage, Landran
sharma.mamta0801@gmail.com

[2]Chandigarh Engineering Collage, Landran
cec.manish@gmail.com

Abstract: *Grid computing is aggregate power of distributed resources. Grid computing is basically used to coordinate and share the information. Grid computing is the form of distributed computing. In the grid computing the scheduling is used. In the grid computing the task scheduling is used, the main goal of grid task scheduling is to achieve the high throughput. It also helps to match the application with the available resources. Grid computing provides the non trivial services to the users. In this review paper, we use the scheduling algorithms to achieve these goals. as the grid computing is the latest technology, so the people has to face the many challenges in this field. To overcome these challenges or the problems we use the scheduling algorithms. The grid computing algorithms are of two types. The first one is known as the static computing and the other one is known as the dynamic computing. In the grid computing complexity is the biggest issue. As the size of the grid is increased, the complexity of the system is also increased. To solve this problem, we use the scheduling algorithms.*

Keywords: *Grid computing, scheduling, security, scheduling algorithms.*

1. INTRODUCTION

Grid computing is the new era of the computer applications. It is currently the big application area. Grid computing is used to aggregate the power of distributed resources. It also used to provide the non trivial services to the users. Grid computing is a type of distributed computing that enables the creation of a computational infrastructure. It uses the coupling wide area distributed resources, databases, storage servers, high speed networks, super computers and clusters for solving large scale, massive and complex problems. Grid computing is generally compared with the cluster computing. But the cluster computing is different from the grid computing in physical logical and technical environment.

Characteristics of Grid Computing:

Heterogeneity:

Grids computing involve the heterogeneity. It allows incorporating varying software and hardware resources spread across different domains.

A wide spectrum of Resources:

The grid resources incorporate computational resources, data storage, communication links, software, licenses, special equipment, supercomputers, and clusters. The Grids promise to provide consistent, dependable, transparent access to these resources.

User Centric: Grids lay the entire focus on the end user. This means that the specific machines are that are used to execute an application are chosen from user's point of view.

The grid computing algorithms are of two types.

- Static
- Dynamic

Grid computing came into being and is currently an active research area [1].

Types of Grid Computing:

Grid computing has the many types. The grid computing has the many uses, according to their uses it divides into many parts.

Computational Grid: These grids provide secure access to huge pool of shared processing power resources. These resources are suitable for high throughput applications.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Data Grid: Data grids provide an infrastructure to grid computing. It used to support data storage, data discovery, data handling, data publication, and data manipulation of large volumes of data. The data is stored in various heterogeneous databases and file systems.

Collaboration Grid: collaboration is possible using the grid. For instance, persons from different companies in a virtual enterprise can work on different components of a CAD project without even disclosing their proprietary technologies.

Network Grid: A Network Grid provides fault tolerant and high performance communication services. Each grid node works as a data router between two communication points, providing data caching and other facilities to speed up the communications between such points.

Utility Grid: This is the ultimate form of the Grid, in which not only data and computation cycles are shared but software or just about any resource is shared. The main services provided through utility grids are software and special equipments. For instance, the applications can be run on one machine and all the users can send their data to be processed to that machine and receive the result back.

Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated. In this view of grid computing, computing becomes pervasive and individual users gain access to computing resources needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on are. Though this vision of grid computing can capture one's imagination and May indeed someday become a reality, there are many technical, business, political, and social issues that need to be addressed. If we consider this vision as an ultimate goal, there are many smaller steps that need to be taken to achieve it. [2]
Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated. In this view of grid computing, computing becomes pervasive and individual users gain access to computing resources as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system.

Grid computing could be defined as any of a variety of levels of virtualization along a continuum. Exactly where along that continuum one might say that a particular solution is an implementation of grid computing versus a relatively simple implementation using virtual resources is a matter of opinion. But even at the simplest levels of virtualization, one could say that grid-enabling technologies are being utilized.

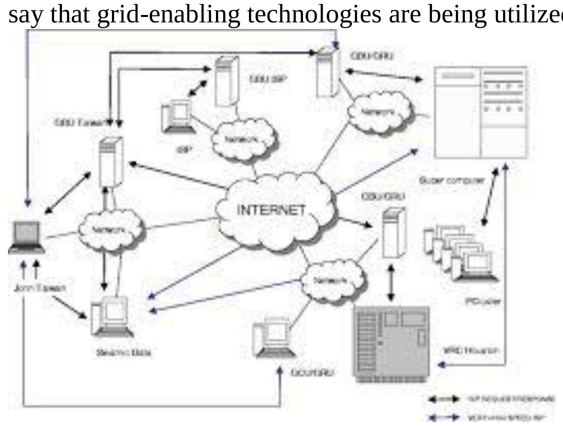

Figure 1: Grid computing

Early implementations of grid computing have tended to be internal to particular company or organization. However, cross-organizational grids are also being implemented and will be an important part of computing and business optimization in the future.
The distinctions between intra organizational grids and inter organizational grids are not based in technological differences. Instead, they are based on configuration choices given: Security domains, degrees of isolation desired, type of policies and their scope, and contractual obligations between users and providers of the infrastructures. These issues are not fundamentally architectural in nature. It is in the industry's best interest to ensure that there is not an artificial split of distributed computing paradigms and models across organizational boundaries and internal IT infrastructures. Grid computing involves an evolving set of open standards for Web services and interfaces that make services, or computing resources, available over the Internet. Very often grid technologies are used on homogeneous clusters, and they can

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Add value on those clusters by assisting, for example, with scheduling or provisioning of the resources in the cluster. The term grid, and its related technologies, applies across this entire spectrum.

Benefits of grid computing:

Exploiting underutilized resources: One of the basic uses of grid computing is to run an existing application on a different machine. The machine on which the application is normally run might be unusually busy due to a peak in activity. The job in question could be run on an idle machine elsewhere on the grid.
Parallel CPU capacity: The potential for massive parallel CPU capacity is one of the most common visions and attractive features of a grid. In addition to pure scientific needs, such computing power is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation, and many others. The common attribute among such uses is that the applications have been written to use algorithms that can be partitioned into independently running parts. A CPU-intensive grid application can be thought of as many smaller sub jobs, each executing on a different machine in the grid. To the extent that these sub jobs do not need to communicate with each other, the more scalable the application becomes.
Virtual resources and virtual organizations for Collaboration: Another capability enabled by grid computing is to provide an environment for collaboration among a wider audience. In the past, distributed computing promised this collaboration and achieved it to some extent. Grid computing can take these capabilities to an even wider audience, while offering important standards that enable very heterogeneous systems to work together to form the image of a large virtual computing system offering a variety of resources The users of the grid can be organized dynamically into a number of virtual organizations, each with different policy requirements. These virtual organizations can share their resources collectively as a larger grid. Sharing starts with data in the form of files or databases. A *data grid* can expand data capabilities in several ways. First, files or databases can span many systems and thus have larger capacities than on any single system. Such spanning can improve data transfer rates through the use of striping techniques. Data can be duplicated throughout the grid to serve as a backup and can be hosted on or near the machines most likely to need the data, in conjunction with advanced scheduling techniques.
Access to additional resources in addition to CPU and storage resources, a grid can provide access to other resources as well. The additional resources can be provided in additional numbers and/or capacity. Some machines may have expensive licensed software installed that users require. Users' jobs can be sent to such machines, more fully exploiting the software licenses.
Resource balancing A grid federates a large number of resources contributed by individual machines into a large single-system image. For applications that are grid-enabled, the grid can offer a resource balancing effect by scheduling grid jobs on machines with low utilization, this feature can prove invaluable for handling occasional peak loads of activity in parts of a larger organization. This can happen in two ways. An unexpected peak can be routed to relatively idle machines in the grid. [3]
Reliability: High-end conventional computing systems use expensive hardware to increase reliability. They are built using chips with redundant circuits that vote on results, and contain logic to achieve graceful recovery from an assortment of hardware failures. The machines also use duplicate processors with hot plug ability so that when they fail, one can be replaced without turning the other off. Power supplies and cooling systems are duplicated. The systems are operated on special power sources that can start generators if utility power is interrupted. All of this builds a reliable system, but at a great cost, due to the duplication of expensive components.

2. LITERATURE SURVEY

Fangpeng Dong and Selim G. Akl discuss about the grid computing. As popularity of the Internet and the availability of powerful computers and the high speed networks as low-cost commodity components are changing the way we use computers today. These technical opportunities have led to the possibility of using geographically distributed resources to solve large-scale problems in science, engineering, and commerce. Recent research on these topics has led to the emergence of a new paradigm known as Grid computing. Grid computing is used to aggregate the power of widely distributed resources, and provide non-trivial services to users. To achieve this goal, an efficient [4] Grid scheduling system is an essential part of the Grid. In this paper author discuss the lots of things. First, the architecture of components

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

involved in scheduling is briefly introduced to provide an intuitive image of the Grid scheduling process. Then various Grid scheduling algorithms are discussed from different points of view, such as static vs. dynamic policies, objective functions, applications models, adaptation, QoS constraints, and strategies dealing with dynamic behavior of resources, and so on.

Leyli Mohammad Khanli, et.al discuss about the resource matching problem in grid computing. The grid infrastructure provides a way to execute applications over autonomous, distributed and heterogeneous nodes by secure resource sharing among individuals and institutions. Typically, a user can submit jobs to a grid without necessarily knowing where it will be executed. The grid resource management system used to distribute such jobs among a heterogeneous pool of servers. [5]It tries to optimize the resource usage. It provides the best possible quality of service. The resource matching problem is an important task in the Grid environment which involves assigning resources to tasks in order to satisfy task requirements and resource policies. This contribution presents algorithms, methods, and software for a Grid resource manager, responsible for resource brokering and scheduling in Grids. The broker selects
Computing resources based on actual job requirements and a number of criteria identifying the available resources, with the aim to minimize the turnaround time for the individual application.
Javelin (Neary et al., 2000) is discussed about the Java based infrastructure for internet-wide parallel computing. In this case, three key components of Javelin system are used. These systems are clients or applications, hosts, and brokers. A client is a process seeking computing resources, a host is a process offering computing resources, and a broker is a process that coordinates the allocation of computing resources. The Javelin system can be considered a computational Grid for high throughput computing. It has a hierarchical machine organization where each broker manages a tree of hosts. Resources are the simple fixed objects with a tree namespace organization. The resources are simply the hosts that are attached to a broker. Any host that wants to be the part of Javelin contacts Javelin BNS system, a Javelin information backbone that maintains the list of available brokers. The host then communicates with brokers, chooses a suitable broker, and then becomes part of the broker-managed resources. Thus the information store is a network directory implemented by Javelin BNS. Hosts and brokers update each other as a result of scheduling work. Thus, Javelin uses demand resource dissemination. The broker manages the host-tree or resource information through a heap like data structure. Resource discovery uses the decentralized query based approach since queries are handled by the distributed set of brokers. In Javelin the burden of the subtask management and monitoring is layered on the client side. The client should also look for a broker that matches its requirements. Thus the clients are thick. Grid-JQA tries to keep the clients as thin as possible and refers the subtask management and monitoring, and the matching to the broker of the system

3. PURPOSED WORK

Due to the grid computing is a dynamic and uncertainly environment, the grid system needs a resource management mechanism which provides a set of available appropriate resources for requesters. [6] There is a framework for QoS in Grid computing, called the Globes Architecture for Reservation and Allocation (GARA), which enables programmers and users to specify and manage end-to end QoS for Grid-based applications. It also provides a uniform mechanism for making QoS reservations for different types of Grid resources, such as processors, networks and storage devices. The main drawback of GARA is its inability to support subtask management, which is one of Grid's main goals. There are two more drawbacks in GARA: the topology of domain should be known in advance and also the resources can not publish themselves. Grid-JQA is going to address the subtask management while the topology information and resource publishing are handled management, formulating the matching problem as an object placement problem dynamically [7]. QoS Guided Weighted Mean Time Min-Min Max-Min selective heuristic for QoS based task scheduling. In fact this work adds QoS parameters to weighted Mean Time Min-Min Max-Min Selective (WMTS) heuristic [8]. In this method, first the Meta tasks divided into high and low QoS groups that the high QoS group contains the tasks with high requirement QoS and the other one includes the tasks with low QoS requirements. In mentioned work after mapping all tasks with high Qos requirements, the tasks with low QoS group are mapped. A Meta scheduling structure that called CARE Resource Broker (CRB) is used. For resource management they are proposed Virtual Resource Management Protocol (VRMP). In this method scheduler is responsible for choose a suitable

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

resource between available matching resources, and Virtual resource manager is responsible for formation of virtual cluster and virtual machine Grid broker selection strategies are used to resolve the broker selection problem in multiple grid scenarios[9]. They also proposed two different resource aggregation algorithms (SIMPLE and CATEGORIZED aggregation algorithms) that used for broker selection. Efficient resource selection in every resource management approach is one of the major challenges. There is one problem about matching in grid environment. Following model explains the problem:

- The problem input
- The problem output: Matching the best resource for each task
- The problem purpose: Minimizing turnaround time

There are two managing models:

- Local
- Global

There are many solution provides for this kind of problems. These are looking on the matching problem; we bring about the following solutions:
First, resources introduce themselves to all clients and each client decides where to send its
Request, thus matching is accomplished by the client. Second, the clients send their request to all resources and each resource decides which request to reply, therefore matching is done in the resources. Third, the resources introduce themselves to a broker and each client sends its request to the broker, then the broker decides how to match the requests with the resources. Here there are two possible methods, (a) after the broker finds the best matched resource; the resource and
The client negotiates with each other directly. (b) The broker interferes in the process until end of job, in order to guarantee the quality of service and perform subtask management. In this work, we implement the third mentioned solution (b). The advantages of the broker management till end are: 1. for guarantying the QoS parameters, the broker should monitor till the results return to the user.
2. May be all requested resources are not free, so application can not start execution, but in our work as soon as one resource is free, the execution of user's task can be started. Gradually, the remained tasks can be executed when the other resources are released.
3. Adaptation is possible only in this method that the broker monitors till the end, as duplicate execution of strong tasks which are being executed in weak resources.
4. By using the broker, the user can be thin client that is the users can execute their application by PDA or Mobile handset considering the user want the power of processing like supercomputer. Because the broker manages for matching, monitoring, sending task, receiving results, and managing failed tasks, the user only sends the request and waits until it receives the results.
5. Managing failed tasks is simpler than the other methods. Because the broker can assign more priority to failed tasks and matches them as soon as possible.
6. Fault tolerance is possible

4. CONCLUSION AND FUTURE WORK

The concept of computational grids, and grid computing in general, is being studied by researchers in many fields, including high-performance computing, networking, distributed systems, and web services. It is an extensive introduction to what a computational grid actually consists of, and what is required to implement it. The modeling of computational grids with heterogeneous resources is just beginning to be explored. [10] The evaluation of scheduling algorithms focused on efficiency rather than robustness. The construction of actual grids for industrial and scientific work has been undertaken by many companies and scientific groups. In future, we plan to implement our resource manager as a part of the Globus Toolkit and to do various experiments to measure the efficiency of the resource manager and job duplication. Also, we will investigate ways to select the best value of the threshold.

REFRENCES

[1] S. Parastatidis, J. Webber, P. Watson, and T. Rischbeck, A Grid Application Framework based on Web Services Specifications and Practices, http://www.neresc.ac.uk/ws-gaf/A%20Grid%20Application%20Framework%20based%20on%20Web%20Services%20Specifications%20and%20Practices%20v1. 0.pdf, 2003.
[2] Foster, C. Kesselman, The Grid: Blueprint for a new Computing Infrastructure, Morgan Kaufmann Publishers, San Francisco, CA, 1998.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

[3] Open Grid Service Infrastructure Primer, Global Grid Forum, http://www.ggf.org, August, 2004.

[4] Fangpeng Dong and Selim G. Akl, Scheduling Algorithms for Grid Computing: State of the Art and Open Problems

[5] Leyli Mohammad Khanli et.al, Grid-JQA: A QoS Guided Scheduling Algorithm for Grid Computing

[6] Foster, I.; Kesselman, C.; et al. (1999). A Distributed resource management architecture that supports advance reservation and co-allocation. In proceedings of the international workshop on Quality of Service, page 27-36

[7] Khanli, L. M. & Analoui, M. (2006a). "Grid-JQA - a new architecture for QoS-guaranteed grid computing system", Proc. of the 14th Euromicro Conference on Parallel, Distributed and Network-based processing, France, PDP2006, Feb 15-17.

[8] Sameer Singh Chauhan & R. C. Joshi, "A Heuristic for QoS Based Independent Task Scheduling in Grid Environment", 5th International Conference on Industrial and Information Systems (ICIIS'10), Jul 29 - Aug 01, 2010a, India pp. 102-106.

[9] Ludwig, S.A. & Reyhani, S.M.S. (2006). Semantic approach to service discovery in a Grid Environment, Future Generation Computer Systems 4 (1)1_13.

[10] Shoukat Ali, Howard Jay Siegel, Muthucumaru Maheswaran, Debra A. Hensgen, Sahra Ali. Task Execution Time Modeling for Heterogeneous Computing Systems. Heterogeneous Computing Workshop, 2000: 185-199.