

Optimizing Job distribution for load reduction

Manik Rakhra¹, Nitin Umesh²

Student, Department of Computer Science¹

Asst. Prof., Department of Computer Science²

Lovely Professional University, Jalandhar, Punjab, India^{1,2}

Abstract: - In today world multiple resources distributed geographically. Optimization is most critical area in cloud computing environments. Resource allocation and time scheduling is an area that need to be taken under high consideration especially if our motive of improvement is performance enhancement. For many jobs I/O is an exploring resource. Most of the data from the grid is going to be online system. The allocation of the processor and I/O resources must be synchronized for the effective performance. For this a tree scheduling policy to accommodate parallel jobs to manage the background workload .A static space time sharing policy is implemented to estimate the planned performance.

Keywords: - CPU, I/O, SS, MPL

I. INTRODUCTION

Development in computational and transmission technologies has made feasible to combine multiple clusters of services to develop large scale distributed system. Grid services and the supporting grid technology make the secure sharing and access to the member of the virtual organization. A computing system enables a platform for all IT resources (servers, storage, network devices, data stores and application). It enables hardware and software assets providing flexibility and collaboration in handling demands. This prevents the need to buy a new infrastructure for the organization with low investments. A service provider makes computing resources available to the customer, as required, and charges them for specific usage rather than charge a flat rate. The word utility is used to make an analogy to other services, such as water, electrical power, etc. Virtualization a computing environment means that the various hardware and the software resources are viewed and managed as a pool, this provides improved utilization of resources. The focus of this paper is to deploy storage resources compute and network to provide them on demand of the consumer [1]. The goal for the problem is that the scheduling of the task and the resource allocation on enterprise computing is a complex problem and also a necessary for improving the advantages. Enterprise grid computing system gained popularity it is increasingly being used for sequential and parallel access. To satisfy high scalability, throughput and policy it must bring job and data in close [2].

The resource allocation and scheduling are addressed in this paper. Based on the allocation of the resources the jobs are scheduled with the CPU research and development organizations, distributed worldwide participate in one or more virtual organizations [2]. A virtual organization (VO) is group of resource providers to achieve a common goal. Dynamic load and availability of the resources need methods for exploring their status continually [3]. The execution and simulation results:-

- 1) Allocate grid resources to a set of applications ;
- 2) An iterative approach to perform a optimized scheduling on heterogeneous resources ;
- 3) Improve the completion time of workflow in combination with job execution

We also compare the static space- time sharing policy against the proposed scheduling. The result shows that the proposed policy performs much more desirable than the static space sharing policy [4].

II. BACKGROUND

In this we formulate the job scheduling problem. Related work is also discussed.

2.1 Distributed Resource Federation Infrastructure

Enterprise computing system consists of the resources that provide services to the consumers within that enterprise managed by a single organization. It also facilitates the consumers to leverage the ready to use services or, enables to choose the required services from the service catalog. The self-service interface should be user friendly. Existing grid infrastructure deployments simply couple a cluster of dedicated machines with existing grid infrastructure [5]. The grid federation consists of clusters of the resources that are dispersed over multiple administrative domains. The proposed frame work defines a relation between the resources and the power. We believe that the collection of many resources is not sufficient to get the high quality performance; an orderly scheduling of job must be done to attain a good performance.

2.2 Resource Scheduling Problem

Suppose given a system of with multiple clusters, each cluster with P sharable workstations and D disks we want to design an online scheduling policy that schedules a set of jobs that occur into the system with aim of reducing the average job execution time and increasing the system usage. Job scheduling is generally composed of two steps that are mutually dependent on each other:-

- The task allocation (space sharing)
- The scheduling of the tasks over time (time sharing)

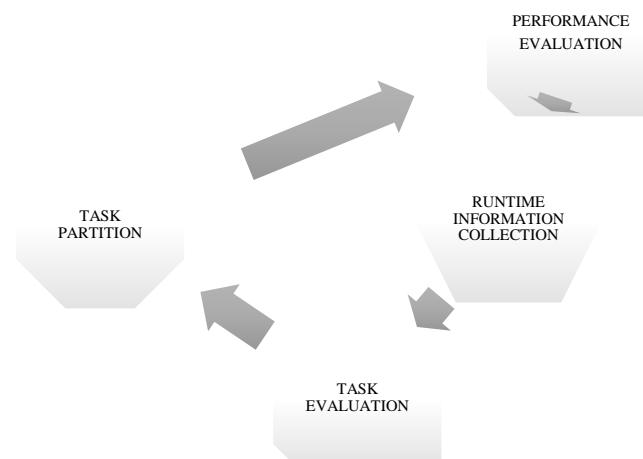
The processors are divided into disjoint sets and each application executes in an isolate manner on one of these sets [6] processor are shared over time by executing different application on the same processor in the time sharing policies during different time intervals. Enterprise follows hierarchy method strategy with assumption that most workers can get it by a single machine. This leads to heterogeneous resources. Therefore, an enterprise grid computing scheduling policy must take in account the resources mixed policy when allocating the resources to jobs. Parallel jobs require computing facilities and data access and storage to be executed. When large number of users attempts to execute a job on a parallel enterprise system of application, computational and storage requirement are issues which make the resource scheduling problem difficult in these system. The problem of coordinated jobs and data scheduling is very complex one [7]. In large scale computing system, there is no resource allocation method that addresses how to solve the problem for online synchronization of processor and I/O resources. There is very little work done in coordinated allocation of processors and I/O resources that addresses the problem for online allocation of processor. Static space sharing can lead to the resource fragmentation problem that effect system performance. We assume that a shared heterogeneous system in which a combination of the CPU-I/O jobs arrive to the system in a stochastic manner. To combine the processor and I/O resource scheduling is the goal of our work. The load balancing concept is used in the tree scheduling. The process how to balance the time and space by allocation task to the processors on the standard buffer processing unit that is allocated to the jobs and the system by values that are generated by a certain method. Space sharing can be static or dynamic. In static space sharing for the whole life time of a job the allocation of processor remain fixed [8].

As if a job is given 5 processors, all the processors are fixed to that particular job till the job get executed and then only the processors can be allocated to another job. All the jobs in the queue have to wait for a very large time. In dynamic the processor allocation is not fixed as dynamic policies assume that the distribution of processors from one job to another doesn't causes much overhead. A space sharing policy is implemented [8].

III. SYSTEM ARCHITECTURE.

It is possible to couple dispersed enterprise resources as a part of one large enterprise system with the large scale growth of networks and their connectivity.

Figure1.1 shows an abstract model of enterprise computing system architecture. Clusters of the processors P are grouped together on the basis of their speed, architecture and physical locations. A set of workstation of similar architecture is grouped into clusters in our method. An assumption is made that enterprise grid computing system is a self-possessed to the set of processors connected to the local area network (LANs), where from further the LANs are connected to the WAN.



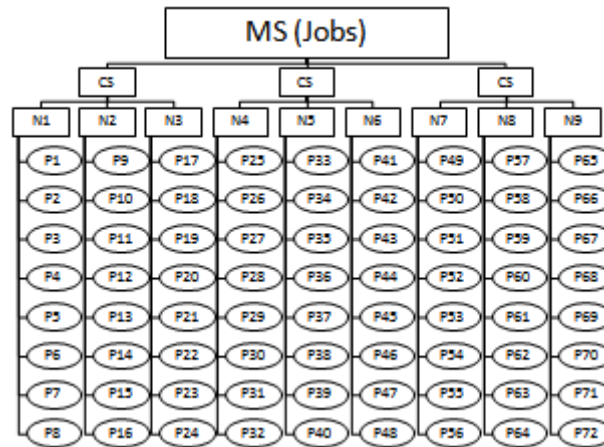


Figure 1.1: Grid Computing System Architecture

Processor on different processor can communicate by sending message over the network. We refer the resulting system architecture as the cluster tree. The system resources are in the form of L levels deep tree. Where $L \geq 1$ as shown in figure1. The number of children of a node I a cluster tree is referred to as its splitting factor. At the top there is a system scheduler (SS) with which the three local scheduler are attached. The internal node represents the higher level structure which acts as a mediator for the node below it. A node in a pending state is said to be neutral state. Multiprogramming level (MPL) is used to control the tasks among which the processors are being shared at a given time. All processors have different processing speed the MPL of a given processor is evaluated by:-

$$MPL = \frac{\text{speed of CPU} * \text{basic MPL}}{\text{lowest processor speed}}$$

The hierarchy of this scheduled model contains a recursive structure for the internal nodes. Jobs are submitted to the system scheduler where they are placed in the job wait queue until a placement decision is made. Application in the enterprise system maintains their data in the databases or the file system. These applications reside on the networked storage. Networked storage means the SAN or NAS architectures. A request from a job is sent to the data scheduler in the I/O subsystem, which further divide the request on multiple I/O nodes for greater performance. If the response of the request is not processed immediately then the request is forwarded in the waiting queue. When the I/O response device becomes available the scheduler processes the pending request from the waiting queue by FCFS policy. When the job and the I/O request are located on the different clusters we say it as a remote request and when they are located on the same clusters we say it as a local request. The concept of virtualization is also used in enterprise so that the resources must be provisioned for a specific task as on demand. It provide the compute and storage facilities. With this we can scale out or scale in the resources depending upon the requirements. With this the allocation of the resources, load balancing and the data management services that control data access and movement around a network.

Algorithm for the adaptive hierarchical scheduling policy:-

```

If state(neutral) == true then
    Sent request for computation to parent node
If level (root) == true then
    End if
If (QUEUE(job) == ∅) then
    Backlogs the request
Else
    Perform job/task assignment
Endif
Perform job/task assignment
Endif
Else
    Request computation from parent node
Endif(level(root)≠true)^(level(leaf)≠true)
If (QUEUE(job) == ∅)then
    End if
If(QUEUE(RFC) == ∅)then
    End if
    
```

When a non- node is in neutral state, by sending a request it initiates the self-scheduling by sending a request for computation (RFC) message to the parent node. If the parent node is in the neutral state at the time of receiving the, it in turn generate its own RFC and send it to its parent on the next level of the cluster. The whole scenario continues till the RFC reaches the system scheduler (SS).

Cloud Computing Services

In term of architecture many attempts are made to classify various cloud computing services. We can also call this approach as “everything as service”(XaaS) .It basically follow a layered approach.

Software as a service: - software as a service is the top most layer of the cloud computing stack which is directly consumed by the end user. It is the capability provided to the consumer to use the service provider's applications. Application is accessible from various client devices via a thin client such as a web browser. The billing is basically dependent on the application usage. The more we use the application the more the cost and vice versa. This service is basically on the top. We can also call this service as a final service offered to the end user such as the Google docs or Microsoft office live. It reduces the need for infrastructure because storage and compute powers can be provided dynamically.

Platform as a service: - Platform as a service provides with an application or development platform. The layers of platform as a service can be further decomposed into programming environments and execution environment. It allow user to build web based applications using a Google infrastructure. With app engine there are no servers to maintain. Google app engine provides platform for consumers to deploy or create their own applications. It allows dynamic allocation system resources for applications based on the actual demand. It provides python and java environment to create applications. The best source of the platform as a service as a service is Microsoft azure platform. It builds applications in java and PHP using eclipse and other tools.

Infrastructure as a service: - This lowest level provides the underlying resources, i.e., storage, computing and networking on which the platform as a service and software as a service rely on. The resources basically refer to the physical resources i.e. the servers but these often are virtualized. Amazon elastic compute cloud (EC2) is an IaaS model that provides resizable compute capacity on a pay per use basis. It basically allows the users to hire the virtual compute on which they run their own applications. Atmos with its web access mechanism is the ideal infrastructure for storage as a service.

IV. CONCLUSION

In the grid computing allow user to access resources and scattered around the organization which can improve the performance of the application at large scale there are required large amount of processing power that can be vary with the demand of the I/O. In this paper we represent the combined CPU and I/O resource scheduling policy and also compare its performance with static space, time sharing policy under workload parameters. You are able to determine that the performance improves if both the processors and I/O subsystem are explicitly co scheduled. With the help of time sharing CPU requirement can be increase the system utilization and minimize the response time

REFERENCE

- [1] Allcock W, Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S. The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. *J network computing Appl* 2001;23:187-200.
- [2] Thain D, Bent J, Arpaci-dusseau A, Arpaci-Dusseau R, Livny M. Gathering at the well: creating communities for grid i/o. In: *Proceedings of supercomputing*, Denver, Colorado, November 2001.
- [3] Jang Uk In, "Policy based scheduling and resource allocation for multimedia communication on grid computing environment" Member of IEEE
- [4] J.H. Abawajy, school of engineering and information technology deakin university Australia "adaptive hierarchical scheduling for enterprise grid computing system",
- [5] *Gore grid today*, vol 4, august 15, 2005
- [6] Abawajy JH an efficient adaptive scheduling policy for high performance computing. *int j Future Gener Comput syst* 2008;22(5).
- [7] Desprez F, Verneis A Simultaneous scheduling of replication and computation for data intensive application on grid; 2005
- [8] Luyang Dong, Bin Gong A Hierarchical Scheduling Policy for Large-scale Rendering.



Mr. Manik received his B.Tech degree in information technology from lovely professional university, Phagwara, Punjab, India in 2012. He is currently pursuing M.Tech in computer Science and Engineering from Lovely professional university, Phagwara, Punjab, India. His research area in database is to optimize the performance.

Mr Nitin Umesh has done his M.S. from IIIT Allahabad. He has done short term research from IISC Bangalore (department of supercomputing and research education). at present he is working as assistant professor in lovely professional university Jalandhar