Volume 4, Issue 2, February 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Resource Allocation on grid Computing System in Secured Environment

Manik Rakhra[1], Nitin Umesh[2]
Student, Department of Computer Science[1]
Asst. Prof., Department of Computer Science[2]
Lovely Professional University, Jalandhar, Punjab, India [1, 2]

Abstract: Scheduling is the method by which the number of processes, threads, or data flows given the access to system. In the load balancing system effectively target with the quality of services. Scheduling algorithm arises from the requirement for the most modern system to perform the multiplexing and multitasking. Scheduling mainly depends on the throughput it define that the number of processes that complete their execution per unit time. In case turnaround time is the total time between the submission of processes and its completion and response time amount of time take when the request is submitted until the first response is produced. In the current work we are going to propose the system which can execute N number of jobs on 72 processors which can take less time and work more efficiently as compared to existing systems. To achieve this goal the proposed system uses to approaches which are time sharing and space sharing. Time sharing approach helps to balance the load of number of jobs on processors and also helps to allocate that job the processor can execute according to its capacity which results in getting less weight time for the jobs. A novel data base in encryption scheme for enhances data sharing inside a database while preserving data privacy.

I. Introduction

Scheduling basically provides the efficient way to use the resources and perform load balancing in computing system. In case of multiple jobs to be executed at a time we need to perform multitasking and each thread is executed with some priority. It needs to schedule the job and time On the other hand waiting time depends on the equal processor remain in the ready queue. Mainly scheduling can be categorized into the three forms. Schedulers mainly divide into long term scheduling when the admission scheduler decide and divide the jobs into ready queue and to execute the program. In this process scheduler set the currently executing process either it is authorized or delay by the long term scheduler. Scheduling is depend upon the degree of concurrency to be supported at any one time whether it high and low amount of processes are executed concurrently by splitting the input and output, CPU processes is to be handled. Long term scheduler is responsible for controlling the degree of multiprogramming. In the modern operating system is used to make sure that real time process get enough CPU time to finish their task. In the case medium term scheduling it schedule temporarily remove the processes from the main memory and place them on the secondary memory. In the medium term scheduling use the concept of paging in the medium term scheduler decide to swap out a process which has active for some time or process which has low priority which is called page faulting. To remove this error use the process of swapping the process back in later when more memory is available when process has been unblocked and no longer waiting for resource [2]. Cluster has become more common for parallel applications. The main benefits of clusters are the availability, scalability and performance. Since personal computers have become much easier to acquire, clusters provide an alternative to supercomputers that eliminates the need for expensive equipment and sophisticated maintenance techniques [3]. Clusters consist of hundreds and thousands of standard CPUs. A cluster is a group of loosely coupled computers that work closely together as a single computational facility. Clusters are composed of interconnected SMP nodes in order to provide increasing processing power [10]. Clusters are arranged so that the speed and reliability provided by single computer can be improved. They are cost effective compared by the single computer.

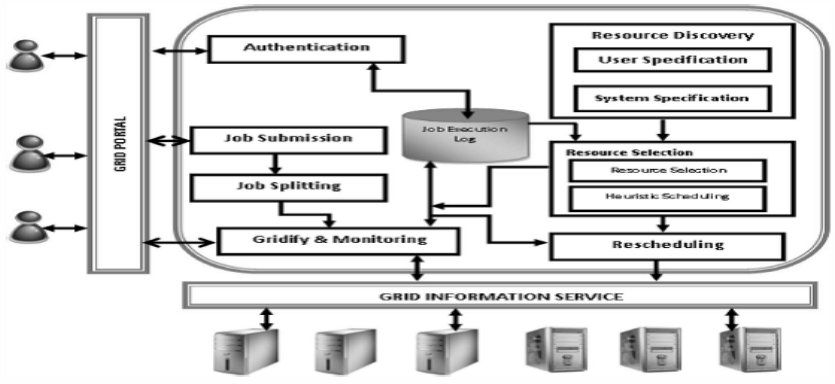

Figure 1. Job execution Life cycle for desktop Grid

II. Job Scheduling Problems:

Distributed Resource Federation Infrastructure: Grid computing comprise of resources spread across an enterprise and provide services to users with enterprise and managed by single organization. They can be deployed within large corporations that have a global presence though they are limited to single enterprise [10]. Efficient mechanisms that loosely couples disparate resources and allocate them to user jobs are important components of the enterprise grid computing.

Existing grid infrastructure deployments simply couple a distributed resource manager with a cluster of dedicated machines. The system defined a grid computing that consists of cluster resources that are distributed over multiple administrative domains, managed and owned by different organizations having different resource management policies. It has no functions for accessing remote data repositories and for optimizing on data transfer. In contrast, the proposed framework defines bindings between computing power and data resources [11]. The described a system which is matches data and jobs [7]. However, many applications will require not just the discovery of data but also of more arbitrary types of resources as well. In the aggregation of many resources is not enough to get good performance careful scheduling of the jobs must be employed to achieve the best performance possible.

Moreover, the users are expected to express relationships between the jobs and the data. In the defined infrastructure together with resource allocation and scheduling that offers a single, cohesive management environment that allocates the shared resources across geographically dispersed sites for all mission-critical enterprise applications, services, and workloads.

Resource Scheduling Problem: The scheduling problem of interest is defined as the process of making scheduling decision involving resources and services over multiple sites. Specifically, given a system composed of multiple clusters, each cluster with P sharable work stations and D disks. In the design an online scheduling policy that schedules a set of J jobs that arrive into the system in a stochastic manner with the objectives of minimizing the average job completion times and maximizing the system utilization [3]. Job scheduling is generally composed of at least two interdependent steps: the allocation of tasks to the processors is called a space sharing and scheduling of the tasks overtime is called timesharing [2].

III. Proposed Method

The proposed policy used the demand driven approach in which all the processor demand for the job according to their BPU sum in every step it creates demand for job form the main system. In order to take into account dynamicity of computing resources and share the system resources both spatially and temporally. It also combine self scheduling approach divide all the jobs in equal partition according to the BPU sum that is given in the system [1]. Whole process mapping is done and find the waiting time and response time of the job. In the system all the jobs distribute equally so it improves the fault tolerance and scalability of the system. With the help of this approach improve the overall resource utilization [7]. Performance of the scheduling policy is studied with the help of work load parameters. There are various form of space sharing approaches where the processor are partitioned into the disjoint sets and each application executes in isolation on one of these sets. In the time sharing policies processor are shared over time by executing different applications on the same processors during the same time intervals. Space sharing and time sharing approach are orthogonal to each other [10]. As this process is running already N number of operations so it may be possible that it might get mapped or clashed with internal schedule. So we may derive a function or some external parameter. Attach to sub schedule [5]. In this procedure computing resources are under the assumptions that most of the workers get by with basic machine, some user require more power and small number of workers like engineers, planners need the fastest machines with two processors and lots of memory. In fact heterogeneous resources in grid computing scheduling policy must take into account the resources heterogeneous when allocating resources to jobs.

Flow for the proposed system:

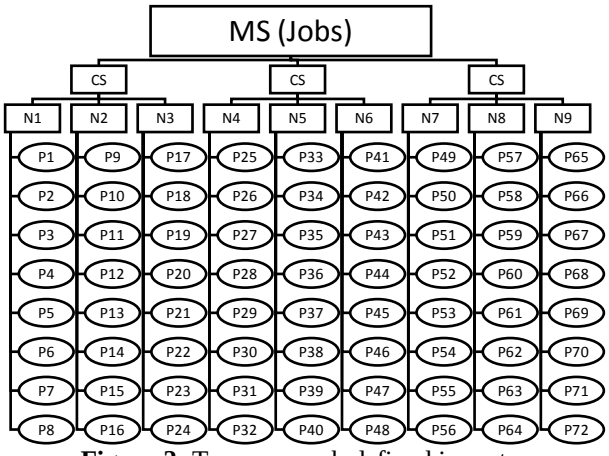

Figure 2: Tree approach defined in system

To achieve this goal the proposed system uses to approaches which are time sharing and space sharing. Time sharing approach helps to balance the load of number of jobs on processors and also helps to allocate that job the processor can execute according to its capacity which results in getting less weight time for the jobs. After this the time sharing technique execute jobs which are allocated jobs according to job sharing techniques and result in producing less response

time then the existing systems [2]. The space sharing technique also allows splitting the job on different processors if one processor is not able to full the requirements of the job then the job will be split on the different processors which makes job to be executed in less time. The proposed system also uses the demand driven approach to make the results of the system more efficient. In the demand driven approach if the processor is in idle state then he will demand its parent for the jobs and if the parent does not have the job then he will demand the job from his parent will makes the system to wait less for the jobs [3]. In the work load model all tasks of jobs have equal service demand. Job cumulative service demand is dividing into maximum jobs and each job will have a demand of minimum time. This work load shows the advantage of space sharing policy. The adaptive scheduling used for Heterogeneous Multi-cluster System can be framed using following steps:

Job Selection: Job selection policy is used to select the jobs in the queue. The global scheduler consist the jobs in the queue. The aim of scheduling policy is to carry the job from the queue in some manner. So we use First Come First Serve policy. It is one of the simple policies and it has less overhead as compare to other policies. It implements just one queue which holds the tasks in order they come in. The job is served in arrival order.

I) Selecting Site: The Site/Cluster is selected on which the job runs. The Most-fit policy is used to select the cluster. The Most-fit policy is used to reduce resource fragmentation by choosing the appropriate cluster which waste less number of processor and by taking care of the other jobs in the queue.

Each cluster which has enough processors for the waiting job, the system performs a series of simulated activities, to measure how many immediate subsequent allocations can follow the allocation decision. After each cluster has been checked, the system selects the cluster with the largest number of immediate subsequent allocations to perform current job allocation.

If there is no single site having enough free processors Multi-site execution co-allocation will be used, this policy tries to run a parallel job across several sites. However, a parallel job might take much longer execution time when running across site boundaries. This is because the speed and bandwidth of inter-site network is usually much slower and less than those of intra-site network. The Largest-First selection is done in case of Multi-site execution. The system first sorts the sites in the computing grid into decreasing order according to the number of available processors BPUs. Then, the system repeatedly picks up a site according to the sorted order until the total number of available processors on all selected sites is larger than or equal to the requirement of the waiting parallel job.

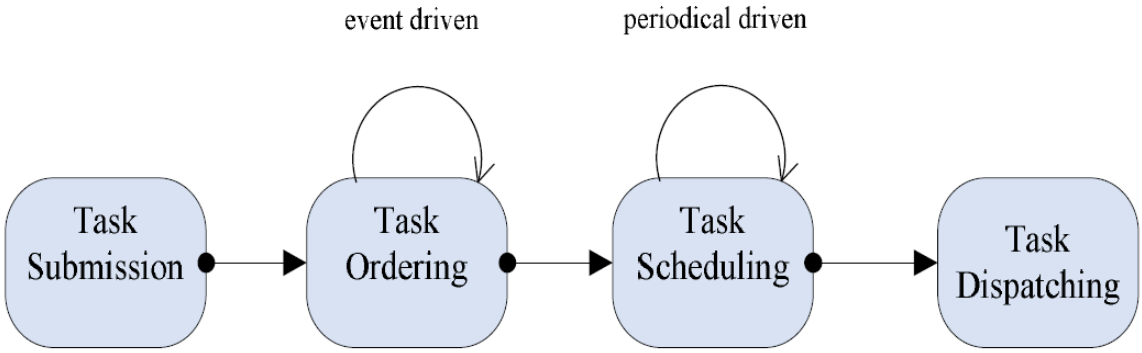

Figure.3: The flow of rendering scheduling system

- **Round Robin**: Each process is assigned a time interval, called its quantum, which it is allowed to run. If the process is still running at the end of the quantum, the CPU is pre-empted and given to another process. If the process has blocked or finished before the quantum has elapsed, the CPU switching is done when the process blocks, of course [2]. All the scheduler needs to do is maintain a list of run able processes. When the process uses up its quantum, it is put on the end of the list. The only interesting issue with round robin is the length of the quantum. Switching from one process to another requires a certain amount of time for doing the administration saving and loading registers and memory maps, updating various tables and lists, flushing and reloading the memory cache in the system.

IV. Conclusion and Future work

In this proposed system to develop such a system which can execute N number of jobs on 72 processors which can take less time and work more efficiently as compared to existing systems. its signifies that it speed up the execution of the job and decrease the waiting time. Processing time for any job will be decease by using this system. This system results in more calculation in less time. In future we will simulate the current process to scrutinize the results

References

1. J.H. Abawajy "Adaptive hierarchical scheduling policy for enterprise grid computing systems"[
2. Jose Orlando Melendez, Shikharesh Majumdar "Utilizing Opaque Resources for Revenue Enhancement on Clouds and Grids"
3. J. H.Abawajy ,S. P.Dandamudi "Parallel Job Scheduling on Multi cluster Computing Systems"
4. Sivarama P. Dandamudi, Samir Ayachi in "Performance of Hierarchical Processor Scheduling in Shared-Memory Multiprocessor Systems"

5. Gang Chen Ke Chen, Jin Xiang Dong, "A database in encryption scheme for enhanced security and easy sharing"
6. Congfeng Jiang, Cheng Wang, Xiaohu Liu, and Yinghui Zhaoin "A Survey of Job Scheduling in Grids"
7. Thyagaraj Thanalapati and Sivarama Dandamudi in "Efficient Adaptive Scheduling Scheme for Distributed Memory Multi computers"
8. Uyang Dong,Bin Gong,Yan Ma, Yi Hu in" A Hierarchical Scheduling Policy for Large-scale Rendering"
9. Young-Chul Shim in "Performance evaluation of scheduling schemes for NOW with heterogeneous computing power"
10. Jang Uk In, Soocheol Lee, Seungmin Rho, and Jong Hyuk Park in "Policy-Based Scheduling and Resource Allocation for Multimedia Communication on Grid Computing Environment"
11. Helen Karatza in "Scheduling in Distributed Systems"

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

Mr NitinUmesh has done hisM.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.

