

Hierarchical Structure based Job Group Resource Allocation Scheduling Model

P. K. Suri

Dean (Academic and R&D),
Chairman & Professor (CSE/IT/MCA),
H.C.T.M., Technical Campus, Kaithal, Haryana,

Sunita Rani

Assistant Professor,
Department of CSE & IT,
B.P.S.M.V., Khanpurkalan, Sonapat, Haryana,

ABSTRACT

In grid environment, the resource management system schedules the jobs to specific resources with trying to minimize some objective functions. This paper considers the hierarchical structure of grid with multiple level resource schedule model which divides the tasks into group according to minimum execution time assigns and these groups to the resources. This grouping based algorithm is compared with non- grouped jobs and obtained better results in terms of completion time, expected performance and expected execution time than the non- grouped jobs.

Keywords

Hierarchical Structure, job groups, expected performance, completion time.

1. INTRODUCTION

A resource management system in grid computing matches requests to resources, schedules the matched resources, and executes the requests using the scheduled resources. Grid resources are the entities such as processor, disk space, memory space, network bandwidth etc. that are managed by the resource management system.

Resource allocation in grid environment is the assignment of available resources to various jobs. There are a number of approaches to solving resource allocation problems. The grid can be expressed as an undirected graph with the nodes to which the system resource belongs [1]. The weights on undirected graph represent the physical distance between nodes.

Due to the heterogeneous environment and dynamic characteristics of resources and node in grid, resources have different speed to execute task and jobs/nodes have different execution time on different resources. However, for the performance concerns, grouping of jobs are performed. Grouping of jobs is based upon minimum execution time of job/task on a resource, those jobs/tasks have minimum execution time on particular resource are placed in a group for that resource, m number of tasks are divided into n number of groups and assigned to n number of resources by the resource management system. These groups are assigned to the resources for execution of tasks and resources are controlled by RMS as central unit and structure of grid turns into hierarchical structure or tree structure from undirected graph.

When the request arrives for task execution, the resource management system groups the jobs and assigns these groups to different resources for execution. Each resource gets only one job/task group for the execution. When all of the job/task groups are completely executed then the entire task is completed and their results are returned to the resource management system.

When the request for task execution arrives to the resource management system and tasks are individually (or non-

grouping state) executing, each task will be assigned to the resource for execution and after completion result will be returned back to resource management system. This will be repeated for all jobs which increase the overhead and data exchange process for each job (assignment of job to resource, sending job to resource and getting output from resources) and increases the communication time as compared to the grouped jobs in which all these are done for a group which are less in numbers than all jobs. Simulation results show the better performance of grouped jobs as compared to the non-grouped jobs in terms completion time, expected execution time and expected performance.

The remainder of the paper is organized as follows. Section 2 presents related work, Section 3 describes the architecture of hierarchical grid. Proposed model is presented in section 4 and section 5 gives the experiments and performance analysis. Conclusion is described in section 6.

2. RELATED WORK:

Grid resource allocation is one of the critical functions affecting the performance of a Grid, because the number of jobs and amount of the required resources are massive and quick responses to users are necessary in a real grid computing, [2] proposed a Hybrid Resource Allocation method based on the Least Cost Method (LCM) and Divisible Load Theory (DLT) method. The Hybrid Resource Allocation method divides the jobs into tasks of equal size and allocates the tasks to available processor using the Least Cost Method.

Effective iterative model for optimal workload allocation is developed [3] and this model is for load allocation to processors and links for scheduling divisible workload applications. This resource allocation model with load originating processor acts as an economic model.

The Bandwidth-aware scheduling schedules jobs in grid systems by taking into consideration of their computational capabilities and the communication capabilities of the resources [4]. The job grouping approach is also used in the framework where the scheduler retrieves information of the resources processing capability. The scheduler selects the first resource and groups independent fine-grained jobs together based on chosen resources processing capability. These jobs are grouped in such a way to maximize the utilization of the resources. After grouping all the jobs sends to the corresponding resources whose connection can be finished earlier which implies that the smallest request issued through the fastest connection giving best transmission rate or bandwidth. Hierarchical architecture based on small-world cluster and Small World Resource Discovery (SWRD) algorithm used in education resource grid have high search success rate and low query cost [5]. Grouping based fire grained job scheduling algorithm [6] light weight jobs are grouped as coarse grained jobs. This groping based algorithm utilized resourced efficiently, improved processing, reduced total processing time, reduced execution time of the jobs and

maximized the utilization of the resources. Dynamic job grouping based scheduling algorithm group the jobs according to MIPS of the resource and selects resource in first come first serve order [7].

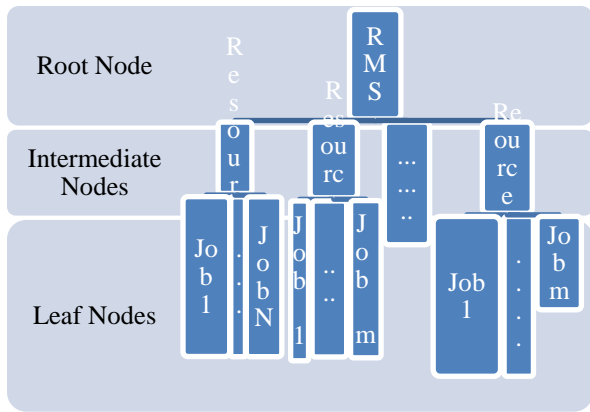
3. ARCHITECTURE OF HIERARCHICAL GRID

Now grid adopts hierarchical structure of extended tree which is managed by RMS (Resource Management System) on upper node which supervises all the nodes below.

In Hierarchical/tree structured grid system the upper level/ root of the grid structure is RMS, middle level/ intermediate nodes of the grid structure are resources with job groups are attached and lower level/leaf nodes of the grid structure are jobs. The branches or edges are communication link which connects the leaves to the root or can say that jobs are connected to RMS via communication link or channel [8]. When the request arrives for task execution, the resource management system groups the jobs and assigns these groups to different resources for execution. Each resource gets only one job group for the execution. When all of the job groups are completely executed then the entire task is completed and their results are returned to the resource management system. The resource is involved in a data exchange process.

Architecture of hierarchical grid [9] is represented by graph formalized as $G = (V, E)$, where V is the vertex set, E is Edge set. $V(G) = \{V_R, V_I, V_L\}$ and $E(G) = \{E_1, E_2, E_3, \dots, E_n\}$

V_L is the leaf node, mainly made up of group of jobs, are the bottom of hierarchical grid. V_I is intermediate node which is resources or resource nodes. V_R is the root of hierarchical grid which is main controller of grid structure.



4. PROPOSED MODEL

4.1 Problem Definition

As grid has a tree structure, each resource has a given processing speed and each communication channel has a data transmission speed (bandwidth). The RMS is spent time on generation and assignment of task group, sending them to the resources, receiving the results and integrating them into entire task.

When the RMS receives a service request from a user, the task is divided into a set of groups (with minimum execution time on a particular resource) that are executed in parallel. Each group contains a portion of the task. The RMS assigns these groups to available resources for execution. After the resources complete the execution of the assigned task groups, they return the results back to the RMS, and then the RMS

integrates the received results into the entire task output requested by the user [10].

Task execution time is random variable that is depends upon many factors such that the speed of resource to which it is allocated for the execution and data transmission speed on communication channel.

4.2 Notations

m	Number of tasks
n	Number of resources
ET_{kj}	Execution Time of a task group j on resource k
r_k	Processing speed of resource k
ET_j	Execution Time of a task j
N	Number of tasks in a group
d_j	Amount of data that is transmitted for task group j (input data from the RMS to the resource, and output data from the resource to the RMS).
s_k	Data communication speed (taken as random variable).
CMT_{kj}	Communication Time of task group j on resource k
CTG	Completion Time of a task group
CT	Completion Time of entire task
μ	average number of task group completed per unit time
P_i	Performance
EP	Expected Performance
EET	Expected Execution Time

4.3 Proposed Algorithm

- Step 1: Input Number of tasks, m and Number of resources n
- Step 2: Generate random execution time (ET) of each m tasks on each resource
- Step 3: Find minimum execution time of each task on a resource
- Step 4: Make groups of tasks according to the minimum execution time of tasks on resource
- Step 5: Generate random processing speed of resource k, r_k
- Step 6: Execution Time of a group $\sum_{j=1}^N ET_j = e_j$
- Step 7: $ETG_{kj} = e_j / r_k$
- Step 8: Generate random d_j, s_k
- Step 9: $CMT_{kj} = \frac{d_j}{s_k}$;
- Step 10: $CTG_{kj} = \frac{e_j}{r_k} + \frac{d_j}{s_k}$
- Step 11: $CT = CTG_1 + CTG_2 + \dots + CTG_n$
- Step 12: for k groups, P_i (Performance) = $1 / CTG_i$
- Step 13: Expected Performance = $\sum_{i=1}^n [(CTG_i) - (CTG_{i-1})] \cdot P_i$

Step 14: Expected Execution Time= $1/\sum_{i=1}^n [(CTG_i) - (CTG_{i-1})].P_i$

Step 15: END

5. EXPERIMENTS AND RESULTS

Total completion time of all tasks will be sum of completion time of all tasks groups,

$$CT = CTG_1 + CTG_2 + \dots + CTG_n;$$

and execution of each group is mutually independent and let average time to execute one task group be β then average number of task group executed per unit time be $\mu=1/\beta$. So, total completion time of task group follows a negative exponential distribution.

Furthermore, the average of completion time of each these n groups is the same, so the sum of completion time of these groups is also said to be Erlang-n distributed.

Thus, Erlang-n distributed random variable is the sum of n independent and identically distributed exponential distributions. Its density function can be shown as:

$$h_n(t) = \frac{1}{(n-1)!} \cdot \left(\frac{1}{\beta}\right)^n \cdot t^{n-1} \cdot e^{-\frac{t}{\beta}};$$

Where, $\mu = 1/\beta$ and μ is average number of task group completed per unit time.

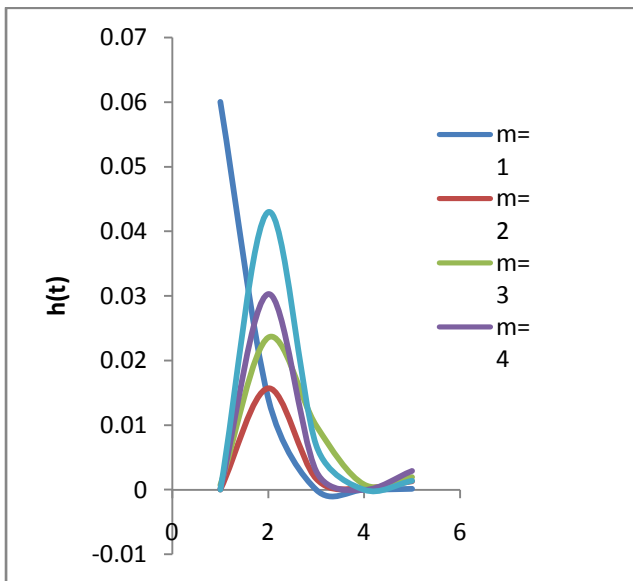


Figure 1: Completion Time follows the Erlang distribution

In this paper, hierarchical structure based job group resource allocation scheduling model has proposed, so their performances need evaluation. Simulation is performed using MATLAB. A comparison is made between two algorithms that based upon hierarchical structure, one of them is based upon grouping of tasks and other one is non-grouped tasks or can say that individual execution of tasks on resources considering various objective parameters i.e., completion time, expected performance and expected execution time.

Firstly, simulation is performed for completion time. Here, number of tasks and number of resources are input and completion time of entire tasks is noted down. In figure 2 shows that if tasks are allocated in groups then completion time is less as compared to when tasks are allocated individually to resources.

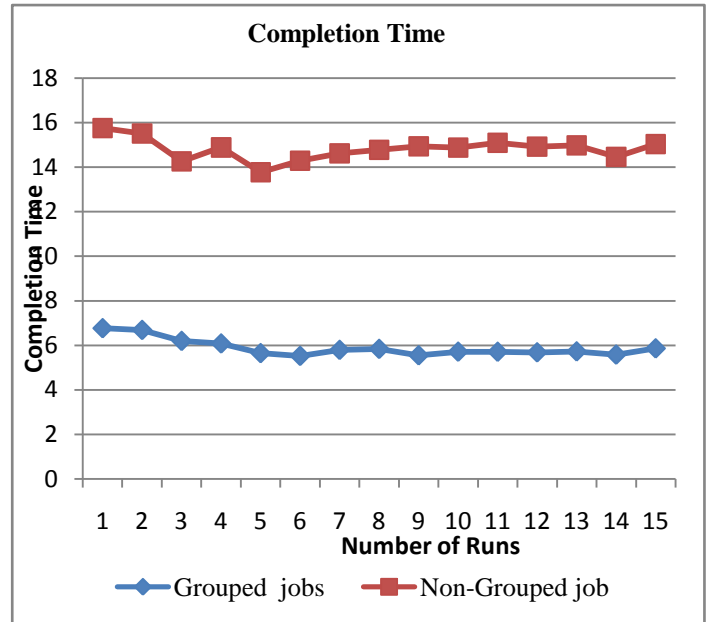


Figure 2: Completion Time of Grouped and Non-Grouped Jobs

Secondly, simulation is performed for expected performance. Here, number of tasks and number of resources are input and expected performance of entire tasks is noted down. In figure 3 shows that if tasks are allocated in groups then expected performance is high as compared to when tasks are allocated individually to resources.

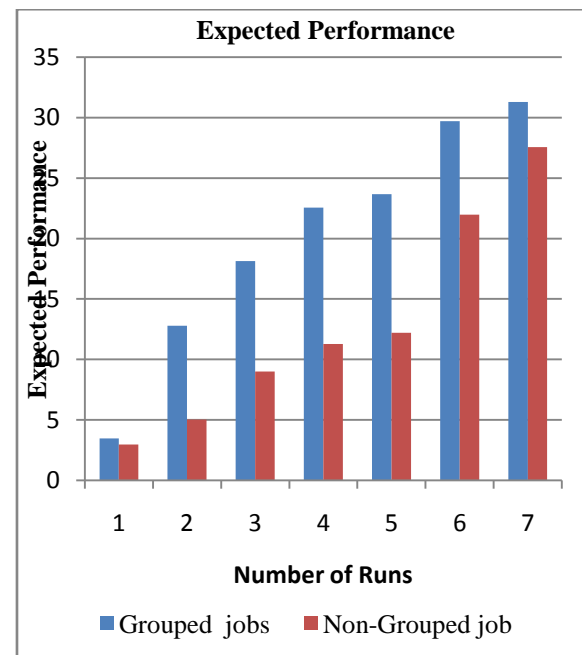


Figure 3: Expected Performances of Grouped and Non-Grouped Jobs

Thirdly, simulation is performed for expected execution time. Here, number of tasks and number of resources are input and expected execution time of entire tasks is noted down. In figure 4 shows that if tasks are allocated in groups then expected execution time is low as compared to when tasks are allocated individually to resources.



Figure 4: Expected Execution Time of Grouped and Non-Grouped Jobs

6. CONCLUSION

Resource Allocation is one of the important issues to be solved in grid computing environment. In this paper, a hierarchical structured based job grouped resource allocation model is proposed and compared with non-grouped jobs considering some objective parameter i.e. completion time, expected performance and expected execution time. Simulation results demonstrate the efficiency and effectiveness of the proposed model. The proposed model reduces the completion time, expected execution time and increases expected performance. In future work load balancing and network delay will be considered to make more effective and more efficient model in grouped jobs for resource allocation.

7. REFERENCES

- [1] Manfu MA, Jian WU, Shuyu Li, Dingjian CHEN, Zhengguo HU, A Grid-distance Based Scheduling for Grid Resource Management, Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA'05),2005.
- [2] 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, 2010.
- [3] G. Murugesan, Dr. C. Chellappan, "An Economic-based Resource Management and scheduling for Grid Computing Applications", IJCSI International Journal of Computer Science Issues, Vol.7, Issue 2, No 5, March 2010.
- [4] Ng Wai Keat, Ang Tan Fong, Ling Teck chaw, Liew Chee Sun, "SCHEDULING FRAMEWORK FOR BANDWIDTH-AWARE JOB GROUPING-BASED SCHEDULING IN GRID COMPUTING", Vol.19(2), Malaysian Journal of Computer Science, 2006.
- [5] Yan Ma, Bin Gong, Lida Zou, Resource Discovery Algorithm Based on Small-World Cluster in Hierarchical Grid Computing Environment, 2008 Seventh International Conference on Grid and Cooperative Computing, 2008.
- [6] Quan Liu, Yeqing Liao, "Grouping-Based Fine-grained Job Scheduling Grid Computing", IEEE, 2009.
- [7] Nithiapidary Muthuvelu, Junyang Liu, "A Dynamic Job Grouping- Based Scheduling for Deploying Application with Fine-Grained tasks on Global Grids, Australasian Workshop on Grid Computing and e- Research, vol. 44, AusGrid -2005.
- [8] Wang Kun, Guo Qiang, Zhang Hengwei, Wang Jindong, "HRMF-G: A Grid-Based Hierarchical Resource Management Framework", 2011.
- [9] Yan Ma, Bin Gong, Lida Zou, Resource Discovery Algorithm Based on Small-World Cluster in Hierarchical Grid Computing Environment, 2008 Seventh International Conference on Grid and Cooperative Computing, 2008.
- [10] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," *Software— Practice and Experience*, vol. 32, no. 2, pp. 135–164, 2002.
- [11] Yuan-Shun Dai, Gregory Levitin, "Optimal Resource Allocation for Maximizing Performance and Reliability in Tree-Structured Grid Services", IEEE TRANSACTIONS ON RELIABILITY, VOL. 56, NO. 3, pp. 444-453. SEPTEMBER 2007
- [12] Rajkumar Buyya and Srikumar Venugopal, "A Gentle Introduction to Grid Computing and Technologies", CSI Communication, Computer Society of India, pp. 9-19, July, 2005.
- [13] Klaus Krauter, Rajkumar Buyya, and Muthucumaru Maheswaran, "A Taxonomy and Survey of Grid Resource Management Systems"
- [14] P.K.Suri, Sunita Rani, "Resource Management in Grid Computing: A Review", "Global Journal of Computer Science and Technology: E Network, Web & Security", Online ISSN: 0975-4172 & Print ISSN: 0975-4350, Volume 13 Issue 17 Version 1.0, 2013, pp. 7-13.