International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-2, Issue-5, September 2014

Distributed and Grid Computing: An Analytical Comparison

Rohit Saxena, Ankur Kumar, Anuj Kumar, Shailesh Saxena

Abstract- **CPU utilization is an important aspect of distributed and grid computing environment. The computing nodes can be overloaded, i.e., they can have more jobs than their capacity such that no more jobs can be associated to them and in that case, the load from the overloaded node can be shifted to other nodes those are under loaded(i.e. doing little work or sitting idle). For this, load balancing is required. In load balancing the workload is redistributed among the computing nodes of the system. This improves the job response time and CPU utilization. Dynamic load balancing schemes operate on the decisions that based on the current state of the system. They do not require the previous state of the system for making the load balancing decisions.**
In this paper, we present an analytical comparison of the various dynamic load balancing schemes in distributed and grid computing environment. This comparison depicts which scheme is better in distributed environment and which is better in grid environment on a particular quality metrics.

Keywords- **Distributed Computing Environment, Grid Computing Environment, Dynamic Load Balancing, Throughput, Response Time, Fault Tolerant, Network Overhead.**

I. INTRODUCTION

Distributed Computing Environment is a collection of computing resources shared among active users. In this environment, a number of workstation or computing nodes (nodes) are connected through a communication network to form a large loosely coupled distributed computing environment [1].
Grid Computing Environment consists of computer systems which are dispersed geographically to share the computing resources in a heterogeneous environment. The increasing demands of the computational resources have lead to the requirement for solutions that are more flexible. Grid connected computers are basically a combination of computing resources applied to a common task, usually to a scientific technical or business problem that requires a major number of processing cycles of the need to process huge amount of data [19].

Manuscript received September 11, 2014.
Rohit Saxena, Computer Science Deptt., UPTU/ SRMSWCET, Bareilly, India, 9458702650, (e-mail: saxena.rohit83@gamil.com).
Ankur Kumar, Computer Science Deptt., UPTU/ SRMSWCET, Bareilly, India, 9458702635, (e-mail: saxenaanksrms@gmail.com).
Anuj Kumar, Computer Science Deptt., UPTU/ SRMSWCET, Bareilly, India, 9458702580, (e-mail:anurom2001@gmail.com).
Shailesh Saxena, Computer Science Deptt., UPTU/ SRMSWCET, Bareilly, India, 9458705791, (e-mail: shaileshgla@gamil.com).

The amount of processing time needed to execute all process assigned to a processor is called workload of a processor [1]. When the demand for computing power increases, the load balancing problem becomes important. Load balancing is the process of distribution of workload among the computing nodes to ensure maximum CPU utilization & minimum response time without additional or faster hardware. Basically, load balancing approach is broadly classified as: static and dynamic. In *static approach*, load balancing decisions are based on the assumed information about the state of the system, job resource requirement and communication time [20] while in *dynamic approach,* the knowledge about the job behavior or the global state of the system, i.e. load balancing decisions are based on the current status of the system [20].
In the section II, we discuss the dynamic load balancing (DLB) strategies and different schemes for dynamic load balancing. In section III, we discuss the distributed & grid computing environments and their structures. In section IV, we describe the behavioral performance of the various schemes on the basis of quality metrics through the tabular representation. In the last section, we conclude our review.

II. DYNAMIC LOAD BALANCING

As stated earlier, in dynamic approaches e.g. [2], [3], [4], [5], [6], [7], [8], [9], [18], load balancing decisions are based on the current state of the system. In this approach, tasks move dynamically from an overloaded node to an under loaded node to provide faster services. The advantage is to react according to the changes in the state of the system. Finding a dynamic solution is much more complicated than finding a static one, dynamic load balancing can produce a better performance because it makes load balancing decisions based on the current load of the system [2],[16]. For this reason, we will focus our attention on dynamic load balancing schemes in this paper.
As shown in fig. 1, there are three different strategies [20] used in dynamic load balancing schemes -

a. *Transfer Strategy*: It helps in decision making of job transfer on the basis of the current status of the computing nodes, either overloaded or under-loaded. It also describes the amount of job which is to be transferred.

b. *Location Strategy*: Its enables the load balancing scheme through decision making for the selection of destination node using the shortest distance and the size of the job.

c. *Information Strategy*: It is the knowledge center of a dynamic load balancing schemes that provides the location and transfer strategies at each node along with the necessary information needed to make load balancing decisions.

In a distributed (and grid) computing environment, DLB can be carried out in two different schemes: distributed and non-distributed. In a *distributed scheme* [4],[6],[8],[10],[11],[12],[18], the DLB algorithm is executed by all computing nodes in the system. All the computing nodes share the responsibility of load balancing among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative. In a *cooperative (global-queue) scheme*, all node works together to achieve the global objective of improving the system's overall global response time. In a *non-cooperative (local-queue) scheme*, each node work independently towards the local goal, e.g. to improve local tasks' response time.
In a *non-distributed scheme*, the responsibility of load balancing is either taken on by a single computing node or some computing nodes but not by all nodes. Non-distributed based dynamic load balancing can take two schemes: centralized and semi-distributed. In *centralized scheme, e.g.* [13],

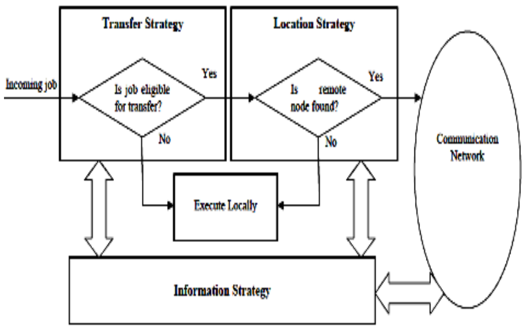


Fig 1. Interaction among components of dynamic load balancing schemes

[14], the DLB is executed only by any single computing node i.e. central node, of the system. The central node is only responsible for load balancing of the whole system. Other nodes interact with the central node only.
In a *semi-distributed scheme*, e.g. [15], nodes of the systems are divided into clusters. Load balancing within each cluster is centralized. A central node is selected to take the responsibility of load balancing within the cluster. Load balancing of the whole system is achieved through the cooperation of the central nodes of each cluster, i.e. the responsibility is distributed among the central nodes of each cluster.
In *sender initiated scheme*, load balancing activity is initiated by an overloaded node (sender) trying to send a job to an under loaded node (receiver) [17].
In *receiver initiated scheme*, load balancing activity is initiated by an under loaded node (receiver), which tries to get a job from an overloaded node (sender) [17].

III. COMPUTING ENVIRONMENT

Distributed and grid computing environment contains a collection of computing nodes spread across the network. The distributed & grid environments have different structure (or organization of computing nodes) which is well-suited for them to give better results on certain quality metrics. In distributed computing environment, mesh topology [21] is more efficient for load balancing schemes due to the maximum connectivity among the nodes directly or in-directly as shown in fig. 2

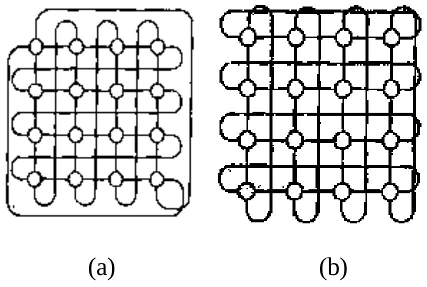


Fig.2. Mesh Network (a) with wrap around connections between processors in adjacent rows & columns (b) with wrap around connections between processors in same row and column.
Similarly, the hierarchical structure of the grid environment is divided into two types of groups: *physical* and *logical* groups. As shown in fig. 3, the physical group can have any topology such as star, mesh etc. Each physical group has a group coordinator (GC) responsible for decision making for load balancing and interconnection with the GC of other physical groups via the logical groups.

IV. PERFORMANCE ANALYSIS

For the study of various schemes, we have assumed that in every circumstance the load balancing is performed successfully. The analysis of the behavioral performance of various load balancing schemes are presented in table 1 on the basis of following quality metrics:

a. *Throughput*: It is the amount of data moved successfully from one node to another in a unit time period.

b. *Response Time*: It is the time taken by an environment to respond while particular load balancing scheme is under operation.

c. *Network Overhead*: It determines the amount of network overhead (or messages) involved while implementing load balancing scheme.

d. *Fault Tolerance*: It is ability of the system to withstand in the event of failure. If a load balancing schemes continue operating properly in the event of some failure then is fault-tolerant.

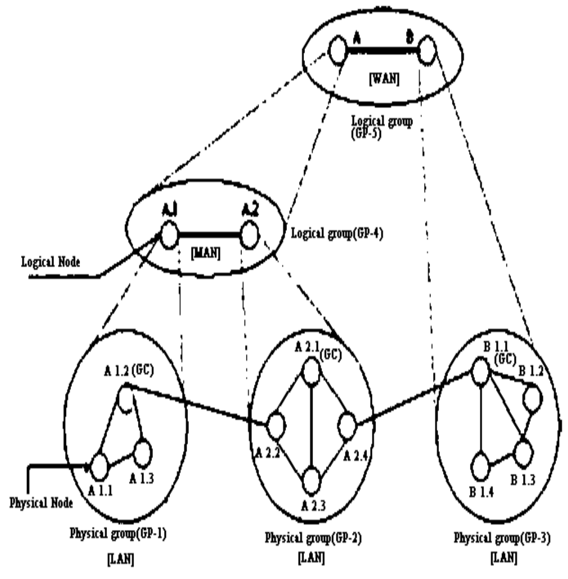
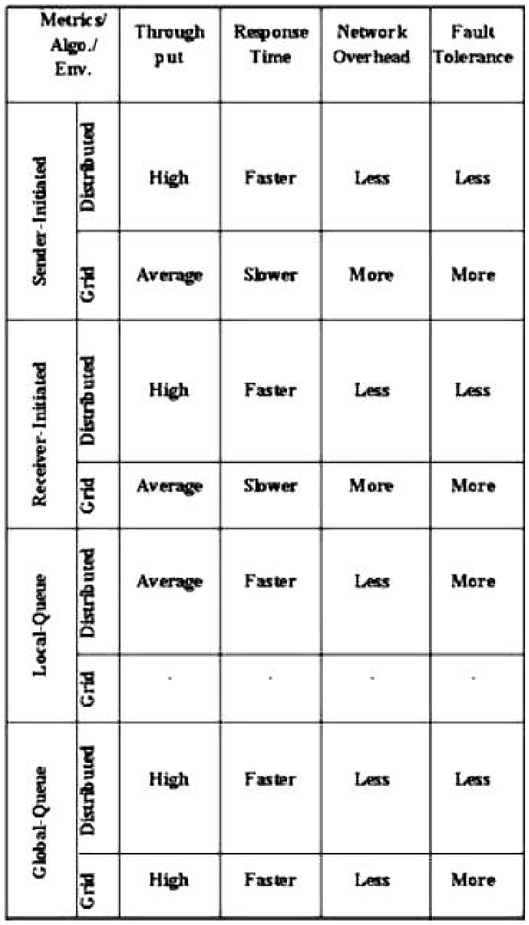


Fig. 3. Hierarchical structure of Grid Environment

Table 1. Comparative Analysis of different dynamic load balancing schemes in Distributed & Grid environments



Metrics/ Algo./ Env.		Through put	Response Time	Network Overhead	Fault Tolerance
Sender-Initiated	Distributed	High	Faster	Less	Less
	Grid	Average	Slower	More	More
Receiver-Initiated	Distributed	High	Faster	Less	Less
	Grid	Average	Slower	More	More
Local-Queue	Distributed	Average	Faster	Less	More
	Grid	-	-	-	-
Global-Queue	Distributed	High	Faster	Less	Less
	Grid	High	Faster	Less	More

V. CONCLUSION

From the study and analysis of the above mentioned load balancing schemes of distributed and grid computing environments, we conclude that the sender-initiated and receiver-initiated schemes have higher throughput, faster response time and less network overhead for the distributed computing environment than grid computing environment. But for both the schemes, the grid computing environment is more fault-tolerant. The local-queue scheme can not be implemented on grid computing environment as each physical group has a group coordinator (fig. 3) that is responsible for load balancing and communication with other group coordinators of other physical group via logical group. The behavior of both the computing environments for global-queue scheme is almost same although the grid computing environment remains more fault-tolerant than distributed computing environment. It means that the DLB schemes give higher throughput and faster response time in distributed computing environment but these are less fault-tolerant in grid computing environment.

REFERENCES

[1] Abhijit A. Rajguru, S.S. Apte, “A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August 2012.
[2] D.L. Eager, E.D. Lazowski, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," IEEE Trans. Software Eng., vol. SE-12, no. 5, pp. 662-675, May 1986.
[3] A. Karimi, F. Zarafshan, A. b. Jantan, A. R. Ramli and M. I. Saripan,"A New Fuzzy Approach for Dynamic Load Balancing Algorithm," International Journal of Computer Science and Information Security," vol. 6 no. 1, pp. 001-005 , October 2009.
[4] J. A. Stankovic and I. S. Sidhu, "An Adaptive Bidding Algorithm for Processes, Cluster and Distributed Groups," in Proc. 4th Int. Conf. Distributed Compu. Sys., pp. 49-59, 1984.
[5] J. Stankovic, “Simulations of Three Adaptive, Decentralized Controlled, Task Scheduling Algorithms,” Computer Networks, Vol. 8, No. 3, pp. 199-217, June 1984.
[6] A. Barak and A. Shiloh, “A Distributed Load-balancing Policy for a Multicomputer,” Software-Practice and Experience, Vol. 15, No 9, pp. 901-913, September 1985.
[7] B. Blake, “Assignment of Independent Tasks to Minimize Completion Time,” Software-Practice and Experience, Vol. 22, No. 9, pp. 723-734, September 1992.
[8] D. Evans, D. and W. Butt, “Dynamic Load Balancing Using Task-Transfer Probabilities,” Parallel Computing, Vol. 19, pp 897-916,1993.
[9] Z. Zeng and B. Veeravalli, "Rate-based and Queue-based Dynamic Load Balancing Algorithms in Distributed Systems," Proc. of 10th Int. Conf on Parallel and Distributed Systems, pp. 349-356, July 2004.
[10] R. Mirchandaney and J. Stankovic, “Using Stochastic Learning Automata for Job Scheduling in Distributed Processing Systems, Journal of Parallel and Distributed Computing, Vol. 3, pp. 527-552, 1986.
[11] J. Stankovic, “Bayesian Decision Theory and Its Application to Decentralized Control of Task Scheduling,” IEEE Transactions on Computers, Vol. C-34, No. 2, pp. 117-130, ,February 1985.
[12] D. Grosu and A. T. Chronopoulos," Noncooperative Load Balancing in Distributed Systems," Journal of Parallel and Distributed Computing, vol. 65, no. 9, pp. 1022-1034, Sept. 2005.
[13] L. Ni, and K. Hwang, K., “Optimal Load Balancing in a Multiple Processor System with Many Job Classes,” IEEE Transactions on Software Engineering, Vol. SE-11, pp. 491-496, May 1985.
[14] Y. Chow and W. Kohler, “Models for Dynamic Load Balancing in Heterogeneous Multiple Processor System,” IEEE Transactions on Computers, Vol. C-28, pp. 354-361, , May 1979.
[15] I. Ahmed and A. Ghafoor, "Semi-Distributed Load Balancing for Massively Parallel Multicomputers," IEEE Trans. Software Eng., vol. 17, no. 10, pp 987-1004, October 1991.

[16] A. Goscinski, “Distributed Operating Systems,” Addison-Wesley, Sydney, 1991.
[17] N. Shivaratri, P. Krueger ,M. Shinghal, “Load Distributing for Locally Distributed Systems” , Ohio State University.
[18] R. M. Bryant and R.A. Finkel. “ A Stable Distributed Scheduling Algorithm” in Proc 2nd International Conference Dist. Comp. pp 323-341 April, 1981.
[19] S. Kumar, N. Singhal, “A Study on the Assessment of Load Balancing Algorithms in Grid Based Network” Int. Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231 – 2307. Volume 2, Issue-1, March 2012.
[20] A.M. Alakeel, “ A Guide to Dynamic Load Balancing in Distributed Computer Systems” Int. Journal of Computer Science and Network Security, Vol 10, No 6, June 2010.
[21] Quinn M. J., “Parallel Computing: Theory and Practices” Tata McGraw-Hill Education Pvt Ltd.
[22] Shailesh Saxena, Mohd Zubair Khan & Dr. Ravendra Singh, “Performance Analysis in Distributed System of Dynamic Load Balancing using Fuzzy Logic” SCET(IEEE) Conference, 2012, Xi’an,China

Mr. ROHIT SAXENA has completed B.Tech from M.J.P.Rohilkhand University, Bareilly and M.Tech from UPTU, Lucknow. Currently, he is working with SRMS Women’s College of Engineering & Technology, Bareilly.

Mr. ANKUR SAXENA has completed B.Tech & M.Tech from UPTU, Lucknow. Currently, he is working with SRMS Women’s College of Engineering & Technology, Bareilly.

Mr. ANUJ KUMAR has completed his M.Sc from Pant Nagar University and is pursuing Ph. D from Gurukul Kangri University, Haridwar. Currently, he is working with SRMS Women’s College of Engineering & Technology, Bareilly.

Mr. SHAILESH SAXENA has completed B.Tech & M.Tech from UPTU, Lucknow. Currently, he is working with SRMS Women’s College of Engineering & Technology, Bareilly