OPTIMIZE PERFORMANCE LOAD BALANCING TECHNIQUES: USING BINARY VOTE ASSIGNMENT GRID QUORUM (BVAGQ): A SYSTEMATIC REVIEW

A.Fairuzullah, A Noraziah, Ruzaini Abdullah Arshah, Azila Che Fauzi

Faculty of Computer System and Software Engineering
Universiti Malaysia Pahang
Lebuhraya Tun Razak, 26300 Gambang, Malaysia

{fairuzullah,noraziah,ruzaini}@ump.edu.my, azilaainul@gmail.com

Abstract. This paper present load balancing technique in a heterogeneous environment allows the usage for geographically widely distributed and multi-owner resources to solve large-level application, usage of load balancing algorithms was important to keep maintaining the balance of workload between emerged infrastructures like grid. This replication generally referred as mechanism to improve availability and performance in distributed databases especially handling fragmented database replication becomes demanding issue. Intended in this paper we address various kinds of load balancing algorithms for the heterogeneous network like grid, especially Binary Vote Assignment Grid Quorum (BVAGQ) and to identify various metric and gaps between them. Many load balancing algorithms are already implemented which work against various issues like heterogeneity, scalability, etc. Different load balancing algorithms for the grid environment work on various metrics such as make span, time, average resource utilization rate, communication overhead, reliability, stability, and fault tolerance. However the aim is to find improved query response time and overall throughput as compared to other scheme.

Key words: Grid Computing, Load Balancing, Distributed Computing, Resource Management, Fault Tolerance

1. Introduction

1.1 Inspiration for Computational Grids

Grid computing is manifested as a wide scale distributed infrastructure which allows large scale resource sharing and synchronized problem solving in a dynamic, diverse network. Numbers of resources are interconnected and work independently by cooperating with each other. Workload represents the amount of work to be performed where all resources have different processing speed. A grid environment can offer a resource balancing results by scheduling grid jobs properly. As the requirements of resource-intensive distributed applications grow, the need for improved over-all throughput and scalability are growing as well. This paper present a new design a cost-effective way to address these application demands is to employ load balancing services based on distributed database that able to handle tiny fragmentation and finally preserve data consistency based on the Binary Vote Assignment on Grid Quorum (BVAGQ) – Query Load balancing. We address this issue how to build to an improved scheduling and efficient load balancing algorithm across the grid may lead to improve the overall system performance with less execution time. Load balancing is required to fairly distribute the tasks across various resources so as to increase the computation and minimum task execution time. In a grid some nodes may be heavily loaded while some may be idle or say under loaded. So a better load balancing algorithm is about to prevent from the condition where some resources are over burdened with work and some are not fully utilized or say free (A,Noraziah et al, 2012) but this disparity between the rate at which scientific applications can calculate results and the rate at which the applications can store their data onto persistent storage especially hard disks is an unavoidable issue for high-end computer systems (Bin Dong et al, 2012), However since apperance of cloud computing has been observed very recently as a new promising paradigm that delivers IT services as computing utilities for companies, academic computing and enterprises. It has caused an influence in IT industries. According to IBM, a cloud is a pool of simulated computer resources of which variety of different workloads are hosted, and allowing them to be deployed and scaled-out through the rapid provisioning of virtual or physical machines; supports superfluous, self recovering, highly scalable programming models and resource usage monitoring in real time to enable rebalancing of distribution when needed. By breaking down, the physical barriers exist in isolated systems, and automate the management of the group of systems as a single form. Cloud computing is an instance of an ultimately simulated system and a natural evolution for data centers that utilized automated systems management, workload evenness, and virtualization technologies. A cloud infrastructure can be a cost optimized model for delivering information services, lessening IT management complexity, encouraging innovation, and escalating responsiveness through real-time workload balancing, though it comes with a price. These successes reveal powerful cloud capabilities that could be leveraged to deliver services faster than any of these users could have achieved if they had to build out their own infrastructure. Despite these successes, Cloud computing enables shared servers to issue resources, software and data for joint services on demand with high interoperability and scalability. We present a result of case study on the system with nonetheless, there are several technical difficulties that need to be resolved before these benefits can be fully realized, which comprise system reliability, resource provisioning, efficient resources consuming etc. Load-balancing is a necessary mechanism among them to improve the service level agreement (SLA) and better use of the resources. Unfortunately, the servers' capability differs a lot in practice and is complicated to record in ordered positions in a server farm, which will result in non resource-aware load-balancing algorithms to circulate workloads fairly. (Sripanidkulchai et al, 2010).

Heterogeneity:

Heterogeneity refers to the use of different technologies and management policies that exists in both of computational and network resources.

Autonomy:

It refers to autonomous because the multiple owned organizations that share Grid resources, a site are viewed as an autonomous computational entity.

Scalability:

Problems involved when a grid grows from few of resources to millions. Better fault tolerant service and quality capability required.

Dynamicity:

Resource failure is possible it can be due to some hardware of software problem or connection disturbance. So to adopt a dynamic behavior to deal with such circumstances is important.

Resource balancing:

Balance the workload on millions of resources itself a challenge. Fair distribution and proper migration policies needs to be implemented.

Reliability and Management:

To keep the data in reliable form there are other issues involved that need to be handled.

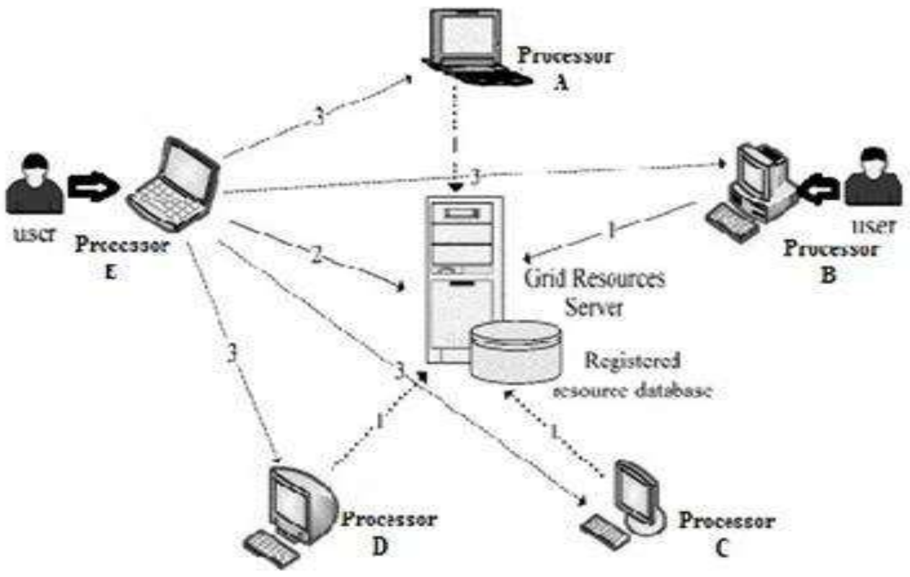

Figure 1.1 [Computational Grid Environment]

The main motivation is to share a load balancing algorithm is used to fully exploiting the unused resources, and has the possibility substantially increasing the Efficiency of resources usage, to enhance the performance and speed of system with no wastage of time. They are also important for the purpose of sharing of computational results, fulfilling the periodic computational needs and overall to meet the goal of balancing the workload among resources.

2. RELATED WORKS

Many load balancing algorithms have been proposed in this field. It is demanding to achieve load balancing in grid systems than in traditional distributed computing environment because of various issues and its dynamic nature. Many schemes presented are based on centralized structure. All of them suffered from significant deficiencies, such as scalability issues. The commonly used techniques for task allocation in grid environment are static and dynamic load balancing (Dobber et al, 2009). Several works have been done on dynamic load balancing approach. A load balancing model based on tree representation of a grid is proposed. In (Yagoubi et al, 2007) a hierarchical load balancing strategy which utilizes a task-level load balancing is presented**.** In (Dobber et al, 2009), Menno et al the effectiveness of dynamic load balancing and job replication by using trace-driven simulations method is analyzed and evaluated.

They proposed a solution to users of parallel application and distributed environment that weather to use DLB or JR. Agent-based approaches have tried to provide load balancing in cluster of machines (Junwei Cao, et al,2003) concerned on load balancing while developing parallel and distributed computing applications. & when the issues of cross-domain and large-scale arrangement comes then emergence in computational grid extends the problem. So In this author proposed work with an agent-based grid management infrastructure which is coupled with a performance-driven task scheduler that has been enhanced for local grid load balancing. In (Shah et al, 2007) represented two job migration algorithms, which are MELISA (Modified ELISA) and LBA (Load Balancing on Arrival).

These differed in the way load balancing is carried out and has proved its efficiency in reducing the response time on large and small-scale heterogeneous Grid environments, authors proposed a decentralized grid model as a collection of clusters and then introduced a dynamic load balancing algorithm (DLBA) which performs intra-cluster and inter-cluster (grid) load balancing. (Yajun Li et al, 2009) in presenting a hybrid strategy for load balancing in grid environment which takes the advantage of two approaches average based, instantaneous approach by merging them. A new replicated model technique a new decentralized algorithm (Azzon et al, 2010) proposed algorithm at Meta scheduler we introducte the cluster or resource level. (Jasma et al , 2010) proposed a fault optimal load balancing algorithm by recognizing the available challenges in grid environment. To ensure the reliability in distributed grid environment, fault tolerance should be high related to the previous model which is Meta Scheduler. In a grid like environment where thousands of computing nodes connected to each other, reliability of each and every resource cannot be guaranteed.

Hence, it's necessary in order to eliminate the probability of failure is in grid computing. Main goal is to prevent, from condition where some processors are overloaded with a set of tasks while others are less loaded or free that the result is been proof with the proposed fault optimal load balancing model realiablity algorithms.

2.2 LOAD BALANCING APPROACHES

a. **Static load balancing**- the amount of processors is fixed, it is assumed that some priori information exist, but if any change occurs in problem size, the fixed amount of processors may not be sufficient, and in some circumstances all processors cannot be employed all the time. So it required some strategy which deal with such circumstances and overcome this problem. Round-robin, simulated annealing, randomized are some of techniques for static load balancing. It leads to use of dynamic load balancing (Lalitha Hima Bindu.et al, 2011)

b. **Dynamic load balancing-** adjustment is made by algorithms to the distribution of work among computing nodes at run-time. They utilize current and recent load information when making distribution decisions, and they do continuous observation of the load on all the processors and when the load imbalance achieves some predefined level, the redistribution of work is ended (Bote-Lorenzo et al, 2004).

The Static method is very convincing because of its clarity and minimized runtime overhead. However, it has disadvantage which reckons that the characteristics of the computing resources and communication network are all known in advance and will be constant. Such assumptions cannot be applied to grid environment.

c. **Parameters-**There are basically three important parameters which determine that which load- balancing strategy will be employed

Who takes the decision for load balancing?

What type of information is required for making the load balancing decision?

Where the decision about load balancing is made?

2.3 STRATEGIES FOR LOAD BALANCING

2.3.1 Centralized & Decentralized Strategy:

Centralized Strategy-In centralized approach (Dobber et al, 2009) only one node in the distributed system functions the role of the main or central controller. This main node has global view on the load information of all nodes connected to it, and decides how to assign jobs evenly to each of the nodes. While the rest of the nodes function as slaves.

Decentralized- all nodes in the distributed system are taking each part in making the load balancing decision. It is commonly agreed that distributed algorithms are more scalable and tolerate faulty better.

2.3.2 Sender-Initiated & Receiver-Initiated Strategies:-

Sender-initiated: In sender initiated strategy, congested nodes attempt to transfer work to under-loaded nodes. Sender-initiated policy works well than the receiver-initiated strategy at low system loads to moderate system loads. The reason behind it is the probability of discovering a lightly-loaded node is higher than that of finding a heavily-loaded node.

Receiver-initiated- In this type, less-loaded nodes look for heavily-loaded nodes from which work may be accepted similarly, at high system loads; the receiver-initiated policy works better since it is much unchallenging to find a heavily-loaded node.

2.3.3 Global & Local Strategies:-

Global Strategy: - The load balancer uses the achievements profiles of all available nodes. Global or local policies both reply the question of what information will be used to make a load balancing decision in global code. For global schemes, balance load speed is faster compared to a local scheme since all workstations are considered at the same time.

Local Strategy:-In local scheme workstations are divided into distinct groups. The advantage in a local scheme is that performance profile information is only traded within the group.

2.3.4 Co-operative & Non-co- operative:-

a. **Co-operative strategy-** is one in which load is shared by other node, in other words nodes co-operate with each other. & on the other side, if they don't reflects **non co-operative** strategy behavior. It takes their decision own to balance load. These are the main strategies used in load balancing mechanism.

2.4 VARIOUS ISSUES

Dynamic load balancing may consider following issues, however it need to collect and maintain information about available nodes (Lalitha Hima Bindu.et al, 2011).

a. ***Process transfer issue:*** It determines whether to execute a process locally or remotely.

b. ***State information exchange issue:*** It determines how to exchange the collected load information among various nodes.

c. ***Load estimation issue**:* This policy specifies the issue regarded to estimate the workload of a particular node of the system.

d. ***Migration issue**:* Main job of this policy is to transfer the load from one state to another. It determines total number of times of the migrating process.

3.COMPARISON OF DIFFEENT LOAD BALANCING ALGORITHM IN GRID BASED ON VARIOUS METRIC/ISSUES

A comparison has been shown in the following table for different load balancing algorithms based on various metric such as communication overhead means to message traffic while communicating , load balancing time, scalability, heterogeneity etc;

Algorithm Matrix	Agent Based Approaches for Load Balancing	Fault Tolerance Optimal neighbor load balancing	Dynamic Load Balancing algorithm in Grids	Decentralized Load Balancing algorithm in Grid
Communication Overhead	More	More	Less	More
Make Span	Less	Average	More	Less
Load Balancing Time	Less	Less	More	Less
Scalability	Scalability	Scalability	Scalability	Scalable
Average Resource/ Utilization rate	Average	Improved	More	More
Fault Tolerance	Integrated	very high	Integrated	Integrated
Reliability	Integrated	high	Integrated	Integrated

Figure 1.2- (Comparison of load balancing algorithm)

There are various techniques to balance the load of Grid Computing, Replication Database and cloud computing. Some of which are discussed in this paper

Year	Authors	Model/Technique	Advantages	Disadvantages
2013	Rajan, R., & Jeyakrishnan, V	Honey Bee Foraging Algorithm	The process of honeybees finding the food and alarming others to go and eat the food. First forager bees go and find their food. After coming back to their respective beehive	As the server gets heavy or is overloaded, the bees search for another location i.e. client is moved to any other virtual server.
2013	A.K. Sidhu, S. Kinger	Throttled Load Balancing Algorithm	This algorithm makes use of identity of virtual machines. Client requests the ID of virtual machine	Throttled load balancing algorithm returns that ID to the user
2013	S. Mohana Priya, B. Subramani	Ant Colony Optimization Technique	In this technique, a pheromone table was being designed which was updated by ants as per the resource utilization and node selection formulae. Ants move in forward direction in search of the overloaded or under loaded node. As the overloaded node is traversed, then ants move back to fill the recently encountered	encountered under loaded node, so a single table is updated every time
2013	Suresh M., Shafi Ullah Z., Santhosh Kumar B	Role Based Access Control (RBAC):	RBAC is a technique used to reduce the load of the cloud. In this, a role is assigned to each user so that limited applications of the cloud can be accessed by their respective number of users.	So by this approach, the resources are restricted to the users
2013	A. Kaur,N. Bansal	Resource Allocation Scheduling Algorithm (RASA)	In this algorithm, virtual nodes are created first. Then the expected response time of each virtual node is found	Then according to the least loaded node criteria, efficient virtual node is found and ID of that node is returned to the client. In this, Min-Min and Max-Min

				strategies are followed. If number of resources available are odd, then Min-Min strategy is applied else Max-Min strategy is applied.

Load Balancing Methods	*Parameters*	*Merits*	*Demerits*
1. Honey Bee Foraging Technique	Execution Time, Overheads, Throughput	Maximize throughput, Low overheads	Low Priority Load
2. Task Scheduling Algorithm based on Load Balancing	Response Utilization, Performance, Response to Request Ratio	Maximize Response Utilization, Increased Performance	Doesn't improve response to request ratio
3. Throttled Load Balancing Algorithm	Load Movement Factor, Communication Cost, Network Delay	High Load Movement Factor	High Communication Cost, High Network Delay
4. Ant Colony Optimization	Performance, Resource Utilization, Fault tolerance, Scalability	Performance increased, Resource utilization high, Fault tolerance excellent, Scalability Good	Complex Network
5. RBAC	Performance, Resource Utilization, Energy, CPU Burst Time	High Performance, High Resource Utilization, CPU Burst time decreases	Response Time increases
6. RASA	Performance, Execution Time	Performance increases, Execution time decreases	Less Fault Tolerance

4.0 Conclusion

This paper presents a comparative survey of load balancing algorithms in grid environment. The accepted techniques of load balancing in grid environment with their importance, combinations and variations have been discussed. With the achievement of (BVAG-CQ) Grid application performance remains a challenge in dynamic grid environment, especially quick query time. Resources are presented to Grid, and can be removed from Grid at any moment. The main objective of load balancing algorithm is to achieve high performance in grid environment by optimal usage of geographically distributed and heterogeneous resources. So such an algorithm which efficiently manage and balance the workload also according to working capacity of processor and minimized the execution time and increase the global throughput of system, is required in such an unpredictable environment of grid. However, accepting the importance of all the aforesaid areas, to put forward a future direction of work, this research would next focus on finding optimal approach for better performance of applications running in grid.

Acknowledgement

The author is thankful the support given by Universiti Malaysia Pahang Fundamental Research Grant FRGS.

References

A, Noraziah., Azila Che Fauzi, A., Zin, N. M., & Herawan, T. (2012). Binary vote assignment grid quorum for managing fragmented database. *ICICA'12 Proceedings of the Third international conference on Information Computing and Applications* (pp. 584-591). Chengde, China: ACM.

2. Azzon, I. A., & Down, D. G. (December,2010). Decentralized Load Balancing for Heterogeneous Grids. *Department of Computing and Software* (pp. 1-6). Perundurai, Erode, India: McMaster University.

3. Bote-Lorenzo, ,. M., Dimitriadis, ,. Y., & Gómez-Sánchez, E. (febraury 13-14,2004). Grid Characteristics and Uses: A Grid Definition. *Grid Computing Lecture Notes in Computer Science Volume 2970* (pp. 291-298). Santiago de Compostela, Spain: Springer Berlin Heidelberg.

4. Buyya, R., Abramson, D., & Giddy, J. (April 2001). A Case for Economy Grid Architecture for Service-Oriented Grid Computing. *Proceedings of the 15th International Parallel and Distributed Processing Symposium* (pp. 776 - 790). San Francisco, California, USA: 10th IEEE International Heterogeneous Computing Workshop.

5. Cao, J., Spooner, D. P., Jarvis, S. A., Saini, S., & Nudd, G. R. (22-26 April 2003). Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling. *Parallel and Distributed Processing Symposium, 2003. Proceedings. International,C&C Research Laboratories, NEC Europe Ltd.,* (pp. 1-10). Sankt Augustin, Germany: IEEE.

6. Dobber, M., Mei, R. v., & Koole, G. (2009, Febraury). Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment:A Comparison. *Parallel and Distributed Systems, IEEE Transactions on, 20*(2), 207 - 218 .

7. Dong, B., Xiuqiao, L., Qimeng, W., Xiao, L., & Ruan, L. (October,2012). A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers. *ELSEVIER, Journal of Parallel and Distributed Computing, 72*(10), 1254–1268.

8. Jasma, R. N. (26-28, November,2010). A Fault Tolerance Optimal Neighbour Load Balancing. *2010 International Conference on Computational Intelligence and Communication Networks* (pp. 428-433). Bhopal, India: IEEE Computer Society.

9. Lalitha Hima Bindu.P., V. R. (8-10 April 2011). Perspective study on resource level load balancing in grid computing environments . *Electronics Computer Technology (ICECT), 2011 3rd International Conference on (Volume:6)* (pp. 321 - 325). Kanyakumari : IEEE Computer Society.

10. Rajan, R., & Jeyakrishnan, V. (2013, December). A Survey on Load Balancing in Cloud Computing Environments. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(12).

11. Rathore, N., & Chana, I. (11-14 Dec. 2011). A Cognitive Analysis of Load Balancing and Job Migration Techniques. *World Congress on Information and Communication Technologies* (pp. 77 - 82). Mumbai: IEEE Computer Society.

12. S. Mohana Priya, B. S. (May, 2013). A new approach for load balancing in cloud computing. *International Journal of Engineering and Computer Science, ISSN: 2319-7242 Volume 2 Issue 5* , 1636-1640.

13.Shah, R., Veeravalli, B., & Misra, M. (2007, December). On the Design of Adaptive and Decentralized Load Balancing Algorithms with Load Estimation for Computational Grid Environments. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 18*(12), 1675 - 1686.

14.Sidhu, A., & Kinger, S. (March-April, 2013). Analysis of Load Balancing Techniques in Cloud Computing. *International Journal of Computers & Technology,Volume 4 No. 2, ISSN 2277-3061*, 737-741.

15.Sripanidkulchai, K., Sahu, S., Ruan, Y., Shaikh, A., & Dorai, C. (April 2010). Are Clouds Ready for Large Distributed Applications. In News (Ed.), *ACM SIGOPS Operating Systems Review. 44 issue 2*, pp. 18-23. New York: ACM.

16.Y. Fang, F. W. (2010). A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. *Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, Vol. 6318*, 271-277.

17.Yagoubi, B., & Medebber, M. (7-9 Nov.2007). A Load Balancing Model for Grid Environment. *Computer and information sciences, 2007. iscis 2007. 22nd international symposium on* (pp. 1-7). Ankara: IEEE.

18.Yajun, L., Yuhang, Y., & Rongbo, Z. (30-31 May 2009). A Hybrid Load Balancing Strategy of Sequential Tasks for Computational Grids. *Networking and Digital Society, 2009. ICNDS '09. International Conference on* (pp. 112 - 117). Guiyang, Guizhou: IEEE.