International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 757-765
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

Dynamic Load Balancing Policy with Communication and Computation Elements in Grid Computing with Multi-Agent System Integration

Bakri Yahaya, Rohaya Latip, Mohamed Othman, and Azizol Abdullah
Department of Communication Technology and Network,
Faculty of Computer Science & Information Technology,
Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia
bakriy@gmail.com, [rohaya, mothman, azizol]@fsktm.upm.edu.my

ABSTRACT

The policy in dynamic load balancing, classification and function are variety based on the focus study for each research. They are different but employing the same strategy to obtain the load balancing. The communication processes between policies are explored within the dynamic load balancing and decentralized approaches. At the same time the computation processes also take into consideration for further steps. Multi-agent system characteristics and capabilities are explored too. The unique capabilities offered by multi-agent systems can be integrated or combined with the structure of dynamic load balancing to produce a better strategy and better load balancing algorithm.

KEYWORDS

Dynamic load balancing, policy, communication, computation, multi-agent system, grid.

1 INTRODUCTION

Manuscript must be typed in two columns. All text should be written using Times Roman 12 point font. Do not use page numbers. There are many load balancing techniques proposed in grid computing environment such as randomized load balancing, round robin load balancing, dynamic load balancing, hybrid load balancing, agent based load balancing and multi-agent load balancing. Round robin and randomized load balancing are considered as basic, simple and easy to implement but not for dynamic, hybrid, agent and multi-agent load balancing. These load balancing methods has undergone an improvement or new ones introduced in the grid load balancing solution.

The load balancing goal is to fully utilize the computing power from multiple hosts without the disturbing the user, regardless of the number of hosts available in the background and aims to improve the overall performance. Besides, load balancing aims to ensure that the workload is fairly distributed among the nodes and that none of the nodes are overloaded or under loaded. Basically, there are two load balancing strategies to consider off, which are called static load balancing and dynamic load balancing.

Static load balancing makes the balancing decision at compile time and it will remain constant. Meanwhile the dynamic load balancing makes more informative decisions in sharing the system load based on runtime state. Comparatively, dynamic load balancing have the potential to provide better performance than static load balancing.

Dynamic load balancing which are based on runtime state needs to process the collected information with firm procedures. The balancing procedures are placed in the dynamic load balancing policy. It contains of a set of rule referred by the system to run and to employ dynamic load balancing for better performance.

System and network performance issues have been explored previously by many researchers. Some look into the resource management, scheduling strategy and load balancing strategy which aim to improve the performance of grid computing [7,8,9,11]. This paper will discuss about the dynamic load balancing in grid computing and multi-agent system (MAS). The paper is organized as follows. In Section 2, we discuss the dynamic load balancing policy, communication and computation elements. Section 3, we carry on the discussion with multi-agent system. The implementation strategy in section 4 and conclusion in the research is discussed in section 5.

2 RELATED WORKS

Dynamic load balancing can be classified into as the centralized approach and the decentralized approach. The Centralized approach is managed by central controller that has a global view of load information in the system which is used to decide how to allocate jobs to each node. In the decentralized approach all joint nodes are involved in making the load balance decision [1]. Dynamic load balancing are based on redistribution of tasks among the available processors during execution time [2]. It transfers the tasks from overloaded processors to the under loaded processor [4]. Under the High Level Architecture (HLA) environment, [15] the redistribution of tasks based on information collected in monitoring interval and during the execution time also. Therefore, generally none of the nodes are heavily loaded.

2.1 Dynamic Load Balancing Policy

The decisions to balance the workload are based on the setup policy. Dynamic load balancing considers and involves four policies [2,3] which consists of transfer policy, selection policy, location policy and information policy. The dynamic load balancing algorithm proposed in [4] takes into consideration the load estimation policy, process transfer policy, state information exchange policy, priority assignment policy and migration limiting policy. Table 1 describes the Dynamic Load Balancing Policy.

Although the policy structure used are diverse but they apply the same strategy to implement the proposed dynamic load balancing solution. It is still involved with information, selection, location and transfer policy which act as the basic policy. These policies work closely pertaining to each unique role and share the decision made to the related or needed policy for the subsequent process. Policy processes or communications process will be discussed in detail in 2.2 and computations process involved in dynamic load balancing will be discuss in 2.3.

Table 1. Dynamic Load Balancing Policy.

No	Policy	Function
1	Transfer Policy	- Need for load balance to initiate. - Determine the condition under which a task should be transferred.
2	Selection Policy	- Select job to transfer. - Select a task for transfer.
3	Location Policy	- Determine under loaded node. - To find a suitable transfer partner. - To check the availability of the service(s) required for proper execution of migration.
4	Information Policy	- Containing all needed information. - To decide the time when the information about the state of other hosts in the system is to be collected

2.2 Dynamic Load Balancing Policy Communication

Policies in the dynamic load balancing need to communicate with each other to determine the processes involved. They need to share the information or decision made to ensure that the subsequent process is able to start. Figure 1 portrays the interaction or communication path among the policy in the dynamic load balancing algorithm.

The incoming jobs are directed to the *transfer policy* to determine whether it should be transferred or not and it is based on various criteria such as workload value and computing power. In other word, it determines the need for the load balancing. If load balance is needed, so the decision will be sent to the selection policy. If not, the jobs will process locally. Receiving the information from transfer policy indicate the *selection policy* to commence the job selection for transference or migration. The decisions made by selection policy are then directed to the location policy for further process.

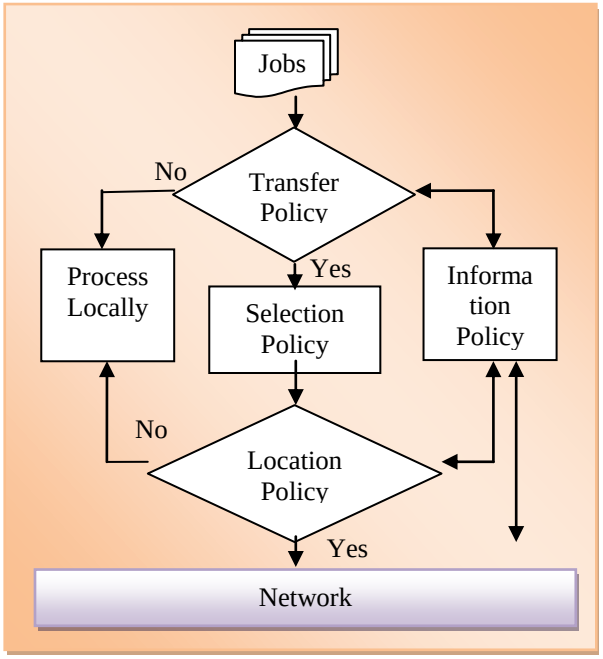

Figure 1: Policy Interaction in Dynamic Load Balancing

The *location policy* is responsible to determine the under loaded node, to find a suitable transfer partner and to check the service availability. If location policy manage to fulfill all the requirement then the particular jobs will be migrated. Otherwise, the local processor will be appointed to process the jobs.

In brief, the information policy plays a big role or possess high responsibility in dynamic load balancing. Information

policy provides the transfer policy and location policy with the necessary information in order for them to build their decision.

2.3 Dynamic Load Balancing Policy Computation.

As discussed in paragraph 2.2, the dynamic load balancing policy need to talk to other policies involved in the system to ensure the perfect decision can be achieved. Usually, it will be influenced by the computation elements.

The incoming jobs are directed to the *transfer policy*, which decides whether to transfer the jobs or to process the workload received. In this case, the employed policy has to have a concrete value or rules to make a particular comparison. The maximum workload value for individual machine or node, called delimit processing workload, can be used in order to make a decision. However, when the average processor utilization [12] is used as an indicator for the dynamic load balancing implementation, it will provide a different set of decisions, based on the environmental setup.

In [13], they computed an estimation of the lowest possible running time under Dynamic Load Balancing strategy using the measured calculation times in the original run to get the average point for further processes. Grid computing in Computational Fluids Dynamic (CFD) [14] uses the concepts of migrating or transferring from one parallel task to another to balance the loads in computers. In this implementation, they are based on equal distributions of parallel task size and one can balance loads by actually moving parallel tasks among available parallel compute nodes. These implementation gives more room to user-level load balancing and eventually let the system to run the balancing or to balance the system without going through centralized load management or data migration among parallel tasks. Thus, from the discussion, it can be conclude that the element of calculation is widely used to make a decision on workload transfer or sharing issues.

The *selection policy* which receives a decision from the *transfer policy* will commence the specific function or activity. In this case, the selection policy will refer to the list or logs file about workload available, makes a selection and transfers the decision to the *location policy*. The simplest mathematical operation engaged by the transfer policy is values or numbers comparison. It compares the list of workload with its own capability or delimitation boundary, to select or choose which workloads to migrate.

Consequently, the *location policy* will trigger and begin to search available computing nodes. There are several strategies which can be used or were explored previously to cater these issues. Based on HLA environment [15], the research proposed several calculations regarding the computational and communication weightage to find a partner or neighbor to migrate or move the workload. They find the arithmetic mean of current load in the whole system, compare with local load and search for a partner to share the workload of an overloaded node. The suitable partner will be evaluated under the communication criteria's latency value to ensure that the workload

migration or workload sharing is beneficial to the overall system performance.

Another strategy [2] used by other researchers are Random, Threshold and Shortest Queue Length. In *Random strategy*, no prior information exchange between the hosts. The task is transferred to a remote host selected in a random fashion. However, it is always a possible that the transfers are done without any improvement, and the randomly selected receiver may already be in an overloaded state.

Meanwhile, in the *Threshold Strategy* algorithm, there is a slight improvement over the randomized algorithm where the location policy avoids useless task transfers by polling a host (selected at random) to determine whether, by transferring a task, its queue length exceeds *T*. If not, the task is transferred to the selected host, which must execute the task regardless of its state when the task actually arrives. Otherwise, another host is selected at random and is polled. To keep the overhead low, the number of polls is limited by a parameter called the *poll limit.* If no suitable receiver host is found within the poll limit, then the host at which the task was submitted must execute the task.

In the random and threshold strategy or approaches, the probability of finding the best transfer partner for a particular task is very low. But in the *Shortest Queue Length* a number of hosts (which is also the poll limit) are selected at random and polled to determine their queue length. The host with the shortest queue length is selected as the destination for task transfer. The destination host will execute the task regardless of its queue length when the transferred task arrives.

Therefore, overall we can see that, there are variety of calculations or computations with different strategies involved in Dynamic Load Balancing (Policy) Computation, starting from the workload entrance until the sharing or migrating workload in the setup environment.

3 MULTI-AGENT SYSTEM

An agent is a computer system that is capable of independent action on behalf of its user or owner. The Agent can figure out for itself what it needs to do in order to satisfy its design objectives. Basically agent is developed to provide services to a particular system. The communication is done by exchanging messages through a computer network. Some of the agents are developed to interact cooperatively and some interact competitively between the agents.

The agents in multi-agent system hold several characteristics such as autonomy, local views, cooperation, social ability, reactivity, proactivity, goal oriented and decentralized. Agents are developed with layer structure and it consists of [11] communication layer, coordination layer and local management layer. The communication layer provides an agent with interfaces to heterogeneous networks and operating systems. It will receive the request and then explain and submit to the coordination layer to decide the suitable action according to its own knowledge. The local management layer performs functions of an agent for local grid load balancing.

A Multi-agent system is composed of multiple intelligent agents that has the ability to interact or communicate, collaborate and negotiate among them. The cooperation between agents permits them to accomplish a common goal. For management and scheduling to be effective, such system must develop intelligent and autonomous decision making techniques [11]. Concordantly the agent itself should also poses intelligence on their own role. In connection with that, agent can make decisions without direct orders or interference. This social ability enables the agent to get assistance from other agents for tasks that are difficult to handle independently. This allows a multi-agent system to have the capability to solve problems which are difficult or impossible for individual agents or monolithic system to solve. The structure of generic multi-agent system is illustrated in Figure 2.

Recently, many load balancing approaches in grid have been suggested, using Multi-Agents. By using Multi-agent system [5], brings to the fore an efficient dynamic load balancing scheme to retrieve and provide the agent-based services. The proposed dynamic load balancing is implemented using new definitions of models and policies on load data collection and agent migration. They employed a credit-based index model to decide which agent needs to be migrated using the credit value. They utilized the load data collection policy, agent selection policy and destination selection policy to enable the dynamic load balancing. The structure proposed complies with FIPA and Figure 3 shows the FIPA agent management reference model.

In [6] prediction-based dynamic load balancing has been proposed, using multi-agent system that predicts the load of the agent based on the predicted data and measured data. They also employ the data collection policy, agent selection policy and migration policy to enable the dynamic load balancing. The proposed scheme succeeds in avoiding unnecessary agent migration and reduced the communications overhead.

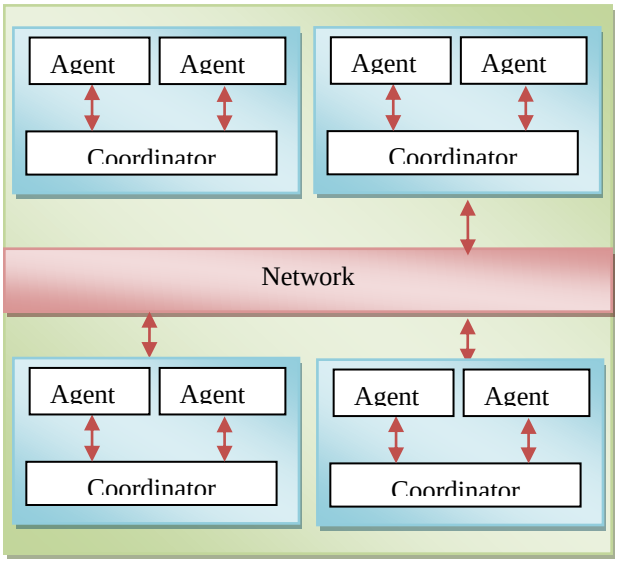

Figure 2: Structure of Multi-Agent System

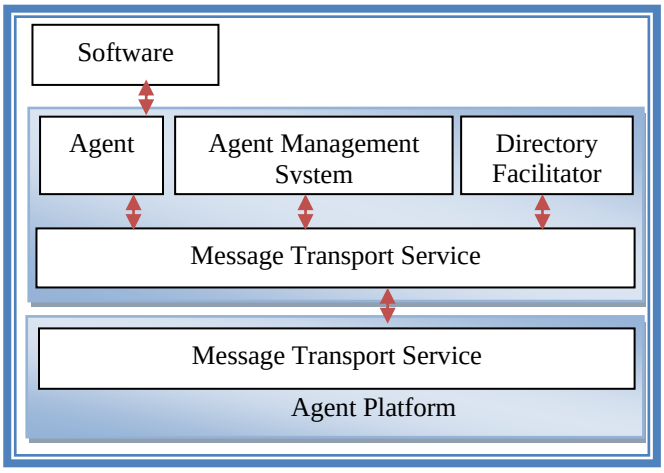

Figure 3: The FIPA Agent Management Reference Model

4 IMPLEMENTATION STRATEGY

The information policy contributes a lot to the decision making process in the dynamic load balancing and hold the authority in that sense. Besides, we can conclude that information policy has a big implication on performance in grid computing through accurate, suitable and efficient parameter use. In fact, the information policy are connected directly to the information directory called index load, profiling or task profiling that does the workload management too. Index load issues [5,6] explored the weightage strategy using the credit-based index model in considering the load balancing factor. But this paper will explore a different method based on agent using the computing power information with an adapted method.

Agents will be developed with multifunction capabilities due to the role embedded into them. They will be in 2 (two) statuses which are as leader of the computing element or as worker of the computing element. The agent will determine what they are and automatically turn themselves into the determined status or role. If the agent is a leader, it will auto-notify the workload system manager. The agent itself has the capability to communicate among the agent and performs the information exchange.

In this paper load balancing function will be implemented at the global and local grids. In the global grid the load balancing decision will be made by workload system manager which sits at the top of the grid described in Figure 5. It makes the decision based on computing element power or index. This is to allocate the correct load value to the correct computing elements which are the leaders in the local grid. This is based on information provided by the computing element leaders to the workload system manager. Then, the computing element leader will decide how to distribute the load according to the worker node computing power available. The worker node will auto-notify the computing element leader on its computing power information if there are any changes to its states since its last update. This will also reduce the communication overhead compared to the polling method.

This implementation will utilize all the discussed policies for the dynamic load balancing which are transfer policy, selection policy, location policy and information policy. The transfer policy is combined with the selection policy, to be known as migration policy as depicted in Figure 6. This will reduce the internal communication between the policies in the agent. The migration policy will be the main door way for receiving data. As it is already holding data, it will analyze the load and decide whether to process locally or remotely. The decision made by migration policy will submit to the location policy to look for a processing partner. The information policy will play a role as information collector to supply information for the agent to make a decision. The proposed strategy is planned to comply with FIPA model.

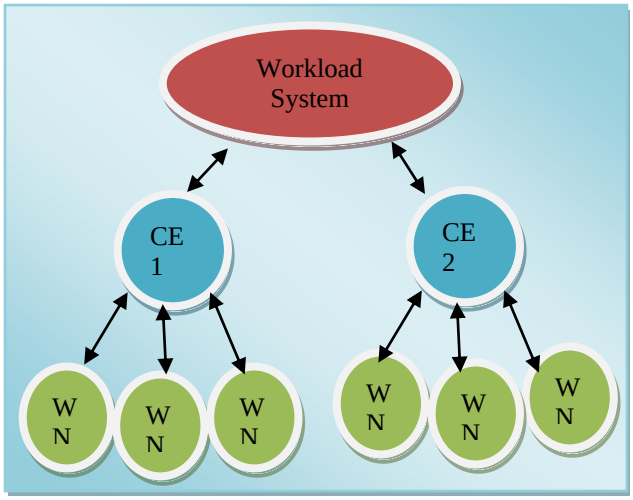

Figure 5: Grid Structure Environment

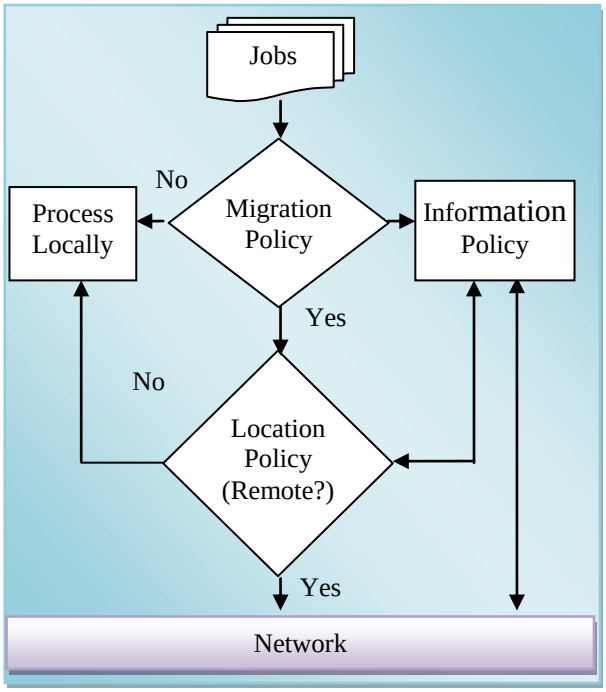

Figure 6: Proposed Policy Interaction in Dynamic Load Balancing

5 CONCLUSION

Multi-agent system (MAS) capabilities are featured broadly in various discipline of study. The successes of integration triggered more researchers to explore these opportunities. The unique capabilities offered by multi-agent systems can be integrated into or combined with the structure of dynamic load balancing communication and computation elements to produce a better strategy in producing a better dynamic load balancing algorithm with multi-agent system. In the future, this paper will be extended to explore the High Level Architecture (HLA) environment and its component, the Run Time Infrastructure (RTI), for the study of Dynamic Load Balancing and the algorithm development that will be embedded into the agent and to implement the simulation. The simulation will be developed using the JAVA programming language.

6 REFERENCES

1. Kai Lu, Riky Subrata, Albert Zomaya, "An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grids Sites" IEEE, pp. 311-319(2006)
2. Rupam Mukhopadhyay, Dibyajyoti Ghosh, Nandini Mukherjee, "A Study On The Application Of Existing Load Balancing Algorithms For Large, Dynamic, Heterogeneous Distributed Systems" ACM: pp 238-243 (2010)
3. Janhavi B., Sunil Surve, Sapna Prabhu, "Comparison Of Load Balancing Algorithms In A Grid" IEEE, pp. 20-23 (2010)
4. Amith Chhabra, Gurvinder Singh, Sandeep Singh Waraich, Bhavneet Sidhu, Gaurav Kumar, "Qualitative Parametric Comparison of Load Balancing Algorithms in Parallel and Distributed Computing Environment" WASET 16, pp.39-42(2006)
5. Yong Hee Kim, Seungwok Han, Chang Hun Lyu, Hee Yong Youn, "An Efficient Dynamic Load Balancing Scheme for Multi-Agent System Reflecting Agent Workload" IEEE, pp. 216-222 (2009).
6. Byung Ha Son, Seong Woo Lee, Hee Yong Youn, "Prediction-Based Dynamic Load Balancing Using Agent

Migration for Multi-Agent System" IEEE, pp. 485-490 (2010).
7. Masaya Miyashita, Md. Enamul Haque, Noriko Matsumoto, Norihiko Yoshida "Dynamic Load Distribution in Grid Using Mobile Threads" IEEE, pp. 629-634 (2010).
8. M. Bohlouli, M.Analoui, "Grid-HPA: Predicting Resource Requirements Of A Job In The Grid Computing Environment" WASET 45, pp. 747-751(2008)
9. Jaehwan Lee, Pete Keleher, Alan Sussman, "Decentralized Dynamic Scheduling Across Heterogeneous Multi-core Desktop Grids" IEEE, (2010).
10. Foundation for Intelligent Physical Agents http://www.fipa.org/
11. Jun Wei Cao, Daniel P. Spooner, Stephen A. Jarvis, Graham R. Nudd, "Grid Load Balancing Using Intelligent Agents" Future Generation Computer System 21, pp. 135-149(2005).
12. Albert Y. Zomaya and Yee-Hwei The, "Observations on Using Genetic Algorithms for Dynamic Load-Balancing", IEEE Transactions on Parallel and Distributed Systems, Vol 12, No 9, pp. 899-911(2001).
13. Menno Dobber, Ger Koole and Rob van der Mei, "Dynamic Load Balancing for a Grid Application", Springer-Verlag, LNCS 3296, pp. 342-352(2004).
14. R.U. Payli et al, "DLB – A Dynamic Load Balancing Tol for Grid Computing", Parallel *CFD* Conference, Grand Canaria, Canary Islands, Spain, (2004).
15. Robson E. De Grande, Azzedine Boukerche, "Dynamic Balancing of Communication and Computation Load for HLA-Based Simulations on Large-scale Distributed Systems", J. Parallel Distributed Computing 71, Elsevier, pp. 40-52(2011).