RESOURCE MANAGEMENT ON DYNAMIC LOAD BALANCING BY SCHEDULING WITH COMPUTATION ELEMENT CONSIDERATION THROUGH POLICY REFINEMENT IN GRID.

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 proper resource management strategy in the scheduling algorithm will produce an improved dynamic load balancing result. The computation element and policy refinement has a big impact to the grid computing performance. The combination of these components supported by a sound strategy embedded can reduce the makespan and improve the load balancing distribution result. The development tools most suitable for this study are the GridSim with Java language.

KEYWORDS

Resource Management, Dynamic Load Balancing, Policy, Computation Element, Scheduling, Grid.

1 INTRODUCTION

Resource management is a mechanism for locating and allocating computational resources that available in the entire set of grid setup. The resources that have to be managed include the physical resources and also logical resources. Other resources that can be managed are memory, disk and networks. Resource management is also responsible for scheduling and job submission.

Certainly, there are specific roles played by the resource management including maintaining the resources and environment, monitoring the states of resources and achieving the highest performance with regards to the environment limitation. At the same time resource management also react with any changes in resource and its environment. This response capability allows resource management to initiate the routine operations such as system initialization, processing and completion.

Resource management in grid can be divided into three steps. There are identifying requirements and resource specification, matching up the requirements to resources based on scheduling or load balancing algorithm and allocating the workload or task to the resources and also monitoring the whole process to obtain the information needed.

Hence we can say that through the resource management dictation, grid computing provider can ensure the aggregation of resource availability, task requirement for processing and enable the strategy to meet the performance criterion and achieve great workload sharing or fairness workload distribution.

According to [1] load sharing and load balancing are two different things. Load sharing algorithm strives for avoiding any unshared states where some computers lie idle while tasks contend for service at other computers. Meanwhile, load balancing algorithm also aim to avoid unshared states but attempt to equalize the loads at all computers.

Managing resources in dynamic balancing environment require great emphasis on computation elements. Normally, computation elements are embedded into specific policies. This paper’s focus is to explore and propose the resource management strategy in the scheduling algorithm under dynamic load balancing conditions.
The rest of this paper is organized as follows. Section 2 illustrates the related works. The implementation strategy discussed in Section 3 and followed by Section 4 for the conclusion.

2 RELATED WORKS

Scheduling is a method of deciding the commitment between resources and task requirement. There are a number of studies for load balancing through the scheduling algorithm.

2.1 Scheduling For Load Balancing

Scheduling is the core part of grid resource management system [2]. The system receives job requests and matches it to the best resource based on proposed scheduling. There are many studies that demonstrate a variety of scheduling concepts and strategies to achieve the targeted performance.

Double layer of scheduling strategy in load balancing has been proposed in [3]. This work focuses on issues of scientific workflow collaboration and interoperability between the same or different Scientific Workflow Management System. The system considers the load balancing engines and balanced utilization of bottom resources. The load model is formulated by utility functions and quantifies resource load, engine load and related attributes. Hence, this strategy the engine selection and resource mapping had been set properly.

But anyhow, this study is limited to experimental conditions and environments only. Furthermore, the proposed algorithm is meant for situations where there exists high volume of unbalanced requests within heterogeneous environment.

Min-Min scheduling algorithm has been improved by [2] and was called the Dmm algorithm. Dmm used the probability methods called Expected Time to Compute (ETC) to sort the jobs. Sorted jobs will be sent to the appropriate list that has been segmented according to the strategy adopted. At the end, the Min-Min algorithm is applied to the remaining segments.

Dmm algorithm has been applied at the production level and has achieved good results. It has shortened the execution time effectively, balancing the load of the system and has improved scheduling efficiency.

Another work in scheduling algorithm of load balancing done by [4] which focused to improved the polling scheduling algorithm. The main strategy for this work is to apply dynamic policy management framework to the load balance schedule algorithms.

There are two improvements which have been adopted into the proposed algorithm. They are Global Load Map (GM) to elevate efficiency in the bi-driven algorithm and Dynamic Determination Threshold to determine the threshold $M$ dynamically through the dynamic scheduling algorithm. Another component introduced in this paper is Algorithm Analysis. This component showed a significant difference between the polling algorithm and the proposed algorithm. The differences are due to the use of dynamic threshold based on prediction, feedback and also the Global Map. As a final result, the proposed algorithm shows that there is no significant difference to the polling algorithm performance when running on light-load conditions. But it is contributing a significant difference when working on converse conditions.

Paper [5] focused in scheduling strategy for high throughput computing in a grid environment. The proposed algorithm considers overall performance of resource to assign tasks. Through this research, Min-Min and Max-Min algorithm has been employed for comparison. The standard deviation of minimum completion time results for both algorithms are compared and the best algorithm results were assigned the tasks. Consequently, the proposed algorithm outperforms both original algorithms.
From the discussion above we found that there are many strategies to produce a schedule for load balancing. Among them are layering concepts or hierarchical based, probability computations with selected attributes, policy refinement and traditional scheduling algorithm improvements. So, it is wide open to choose the preferable strategy, technique and methods to employ the load balancing through scheduling. Anyhow, some of the researchers consider more towards the dynamic load balancing adoption.

2.2 Dynamic Load Balancing

Generally dynamic load balancing development consider four policies [6,7] which consists of transfer policy, selection policy, location policy and information policy. But, we can use a different policy name with different policy role based on the targeted objective. As seen on [8] that 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 from one to another implementation [6,7,8] but they share the same objective which is to enable the algorithm development and to achieve a better overall performance.

Nevertheless, it is still involved with information, selection, location and transfer policies which act as the basic policy for dynamic load balancing scheme. These policies work closely pertaining to each unique role and share the decision made to the related or needed policy for the subsequent processes. Generally, most of the scheduling algorithm that focuses on load balancing shares the pattern of dynamic load balancing strategies implicitly or explicitly.

2.3 Computation Element

Computation element or the calculation strategy in grid computing mostly refers to the resource management component related to the workloads or tasks and resources. However, there are other resources being referred such as network bandwidth, memory capability, CPU speeds and etc.

The variety of available resources in grid computing environment makes the calculation elements more meaningful in order to find the segregation of resources. Based on [10], the competency rank introduced on resources, logically create different sites. The sites ranking are created based on the number of requests that they do not respond to, and ranked as low competency, middle competency and high competency. At the same time jobs priority is also used to segregate jobs. These strategies enable the grid to achieve the targeted performance.

Loads or sometimes called jobs, tasks and workload are the data that needs to go through the processing part. Many researchers used the load size to calculate the average load, average load for cluster, average load for system and etc. Subsequently, the load values can be use to contribute to the threshold as used in [11, 12, 13, 17, 18].

<table>
<thead>
<tr>
<th>No</th>
<th>Policy</th>
<th>Function</th>
</tr>
</thead>
</table>
| 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                              |

Table 1. Dynamic Load Balancing Policy.
Meanwhile, the loads and computation element also contribute to the Resource Index Element. The index is normally used to provide lists or ranks of the available resources for the resource management to make further decisions. The next discussion will explain about resource index element.

2.4 Resource Index Element

Resource index elements are very important to the dynamic environment in grid computing as more resources available and the resource selection becomes more challenging. The challenge to make a selection for appropriate resources becomes more critical when we add more parameters into the selection criteria’s.

Some researchers used the resource speed to make a selection or used the information to generate more valuable conclusion [14, 10]. Some used the prediction methods to suggest the resources to be selected. In [14] the Prediction of Execution Time or PET has been implemented and able to provide a good result for the resource selection purposes.

This method employed several layers comprises of listing up the available resources, calculating the prediction execution time based on specific policies and then splitting up the resources into dedicated ranges. Consequently, some of the ranges are assigned with a number of machines and some none.

If the range of resource speed is too wide, then the number of resources becomes more in the list. Otherwise, if the range of resource speed is small, then more resources are available in the list. Anyway, it's not a total solution when the bandwidth capability does not support bigger traffic and the resource pool under the same cluster.

At the same time, there is another study which uses the execution time for resource selection. This strategy only considers the load index calculation with two values which are total estimated job execution time and remaining estimated execution time. These values are used to obtain the load index [19]. However, it is only suitable for certain environments such as neighboring cluster.

Another strategy used in resource index is through load level determination. The load level is calculated then compared to the threshold value to classify the resource state. If the resource load level is more than the threshold value, the respective resource is labeled as overload and will not be consider for the following steps [15,16].

Some papers used the upper boundary and lower boundary in conjunction with the threshold value. In this case the upper boundary is denoted as threshold value or overloaded state. The lower boundary value means that the respective resource or machine is under loaded [17,18]. In addition some of the strategies employ the statistical calculation with variance calculation as in the case of [9] and official standard deviation formula. These are the common strategies used by many researchers nowadays.

3 IMPLEMENTATION STRATEGIES

There are several components that have been discussed earlier in this paper. The components are resource management, dynamic load balancing, scheduling, computation element and a little bit about the policy in dynamic load balancing.

3.1 Resource Management and Load Balancing

Resource management has the authority to orchestrate the other components in the environment. It will look into the overall processes and manage the procedures given in the policy. Here, the resource manager will be known as the Workload System Manager or WSM. As the structure setup for the grid in this research is hierarchical, hence there will be an intermediate level resource acting as a cluster manager. Figure 1 describes the structure discussed.
In this paper the load balancing function will be implemented at the global grid, where the load balancing decision will be made by WSM, supported by weighted resource methods. It is a combination of Prediction Execution Time (PET) and weighted value specified by [11]. The proposed method responsible to determine the rank of resources based on the policies specified by the selection mechanism:

All of the selection mechanism attributes listed below will be considered as needed. However, more attention should be given to the matchmaking model mechanism because of its known constraint.

### 3.2 Policy Structure

The resource management and load balancing will reside on the migration policy. The migration policy will be the main doorway 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 the migration policy will be submitted to the location policy to look for a processing partner. On the other hand, the information policy will collect the information attributes and supply them to the migration and location policies.

Technically, the computation element will use the common standard deviation in conjunction with a few other policies or formulas to determine the resource threshold. It is very important to the algorithm to ensure the limitation boundary for performance achievement and load balancing fairness. Therefore, the computation elements have to consider not only one but multiple parameters.

The computing element or CE will act as a leader to worker nodes denoted as WN. At the same time the CE will work closely with WSM, supported by the information policy acting as a bridge between them. The WN 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.
The policy structure to be used in the proposed study is depicted in Figure 2.

![Figure 2: Dynamic Load Balancing Policy](image)

4 CONCLUSION

The scheduling algorithm will be developed using Java language based on GridSim platform. The use of computation elements comprises of Prediction Execution Time (PET) and computing power ranking are believed to improve the WN selections, reduce the makespan and enhance the distribution of the loads. In addition, the adjustments of the threshold value and distribution cycle interval will also affect the overall performance.

5 REFERENCES