Journal of Computer Science 10 (12): 2553-2563, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.2552.2563 Published Online 10 (12) 2014 (http://www.thescipub.com/jcs.toc)

RESOURCE BROKER MANAGEMENT BY ADOPTING SELF-ADAPTIVE MULTI-INSTANCE BROKER SCHEDULING IN GRID COMPUTING

Bakri Yahaya, Rohaya Latip, Azizol Abdullah and Mohamed Othman

Department of Communication Technology and Network,
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

Received 2014-08-18; Revised 2014-10-20; Accepted 2014-11-24

ABSTRACT

A grid resource broker seeks to assign the appropriate jobs to the appropriate resources as part of resource management in the multi-grid environment. Multi instances of the broker system provides multiple instances of brokers to simultaneously process jobs between multiple resources in a hierarchical cluster grid environment. In this study, the multi-instance broker is developed using grid resource broker taxonomy properties. The number of broker instances to be used for each processing session is determined by calculating resources, computing power and workload. The Self-Adaptive Multi-Instance Broker Scheduling algorithm SAMiB was tested against iHLBA algorithm through four types of scenarios containing various mixes of background load and CPU speed. The SAMiB algorithm has achieved a decrease of 14.93% in makespan time for 2000 jobs, proving the suitability of the multi-instance broker concept for the hierarchical cluster grid environment.

Keywords: Grid Computing, Resource Management, Grid Broker, Multi-Instance Broker, Scheduling

1. INTRODUCTION

The interpretation of the broker made in the grid resource monitoring system has encouraged a dedicated exploration of resource brokers. As a result, Kertesz and Kacsuk (2007a; 2007b; Kertesz *et al*., 2009; Kandagatla, 2003) had elaborated on the taxonomy of grid resource brokers more specifically to increase the researchers' understanding of the resource broker. Kertesz and Kacsuk (2007a) claimed that although the existing grid middleware provides the function to choose the environment for the user's task to run, but in reality they still aren't supporting automated discovery and selection. Afgan (2004) has explained that the automated discovery and selection issues were supposedly solved by the grid broker. E. Afgan proposed that brokers have to be equipped with several matched resource suggestions to process the jobs, but be subjected to user options to make a decision.

This study is motivated by the invention of the multi-broker and the automation issues from the broker perspective in grid computing. Many studies have introduced new capabilities of multi-broker extensions to serve in the multi-domain grid environment. For example, research by Roy and Nandini (2011) explored the advantages of agent systems in developing their enhanced resource brokers. The framework for trust management in multi-broker for resource selection in grid computing was explored by Varalakshmi *et al*. (2007). Research on the scheduling, evaluation technique

Corresponding Author: Bakri Yahaya, Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

for multiple grid scenarios by Rodero *et al*. (2010) which had proposed the "best Broker Rank" for the broker selection modules and finally the research on the integration of the web portal, resource brokering subsystem, multi-grid manager centre and multi-grid resource modules for the new resource broker architecture was done by Yang and Hu (2010). The mentioned examples had successfully shown that more grid brokers can be applied in the operation of multiple grid environments and that more than one broker can assist in the single grid environment.

The same strategies discussed above will be used in the hierarchical cluster, grid environment, but compounded to different methods to seek the usability and effect to overall grid performance. The automation issue will be refined and employed to improve the selection strategy in evaluating the grid facilities. Two (2) selected algorithms called improved Hierarchical Load Balancing (iHLBA) and Self-Adaptive Multi-Instance Broker Scheduling Algorithm (SAMiB) will be used to facilitate the experiment.

This study's contribution is to clarify the adoption of multi-instances of the broker in cluster, grid, to enhance the multi-instance broker usability through the concept of self-determination and to define the method for automated multi-instance broker implementation. Finally, this study also aims to seek a significant performance gap by way of the multi-instance broker against the selected algorithm. The remaining part of this study is organized as follows. Section 2 discusses the related work of multi-broker invention, the concept and the structure proposed to determine the practical way to develop the multi-instance broker in cluster grid environment. Section 3 discusses the proposed multi-instance broker, the automation and the strategies employed to determine the number of broker instances to be generated. The calculation and processes involved presented under section 4 and section 5 discusses the experiment design. The results and discussion are presented in section 6. Finally, the research conclusion and future work are discussed in section 7.

2. RELATED WORK

Studies by Roy and Mukherjee (2011), explored the advantages of agent systems to develop their enhanced resource brokers. As reported, the uniqueness of the developed brokers is their ability to perform resource brokering activities for a batch of jobs that are concurrently executed in a grid environment. Another unique characteristic of this enhancement, is a method for minimizing the execution time of the batching level. This experiment was undertaken in a test bed environment, where Java was used to develop the agent-based system and all nodes were running Linux OSes. The strength of this research was founded on the concurrent execution of brokers and job run time manipulation which considers the batching level strategies. The benefits of these strategies can be seen at the cumulative processing run time.

The framework for trust management in multi-broker for resource selection in grid computing was explored by Varalakshmi *et al*. (2007). A reputation-based trust management architecture that supports the choice of service provider based on their trust values available on the fly through brokers was introduced. The trust parameters used were the number of transactions, satisfaction-level and cost of transactions. The suggested architecture insists on multiple brokers in each domain. As a result, the performance of without-trust model has been surmounted by the performance of with-trust model. In other words, the trust values introduced in this research can be mapped into ranking methods. The higher value of rank means more suitable matching nodes or destinations were found. It is one of the most chosen strategies in the selection rule in grid computing and proven to help in improving the grid performance.

The next research focuses on the scheduling, evaluation technique for multiple grid scenarios which was done by Rodero *et al*. (2010). In particular, this research also consists of the suggestion on "best Broker Rank" for the broker selection modules. This study is based on the ranking methods with double layer filtering. It had increased the accuracy of broker selection to be mandated with job processing. Therefore, it is worthy to conclude that this strategy can improve the efficiency of broker selections too.

Another successful research on resource broker was done by Yang and Hu (2010). The web portal, resource brokering subsystem, multi-grid manager centre and multi-grid resource modules have been integrated and is known as new resource broker architecture. This new architectural design has enabled users to communicate well with the system through the web portal's facilities. Acting as a gateway, the web portal assists in submitting jobs to the resource broker. The best features provided here are the abilities to achieve higher-performance computing by way of workflow execution and monitoring the status of a grid or multi-grid.

The Hwang *et al*. (2010) incorporates safety issues under the resource broker studies with two risk-aware

strategies. These two strategies are “self-insurance” and “risk performance” which have similar functions, but different objectives. The “self-insurance” strategy is broker-driven based that provides a replacement component or resources regarding any failures. Contrarily, “risk performance” is a user-driven based strategy, ensuring the user security requirement.

As to conclude on the research above, the objective of the studies mostly lead to interoperability between grid, the suitable broker numbers of grid and the broker selection issues. Those inventions and strategies shown in **Table 1**, aims to improve the performance and stability of the grid. Furthermore, the implementation of multi-broker in the multi - grid environment has given for an idea of the Multi-Instance Broker in grid computing.

3. MULTI-INSTANCE BROKER

The Multi-Instance Broker consists of two (2) components which are the Self-Adaptive Multi-Instance Broker Manager and the Broker Instance entity. The Adaptive Multi-Instance Broker Manager is responsible to decide on a suitable number of broker instances to be used in the processing. The second component of multi-instance broker is the broker instance entity which is responsible to generate the broker instances according to the result notified by the Self-Adaptive Multi-Instance Broker Manager and also for aiding the workload processing by implementing the scheduler policies.

3.1. Multi-Instance Broker Properties

Bound to the multi-instance broker properties as shown in **Table 1**, the multi-instance broker is generated as a brokerage service extender. The attributes of grid broker are replicated into the broker instance to enable the broker function. The broker instances will receive new jobs handled by the grid system and execute the related processes. Through this invention, the next broker instance is ready to receive new jobs without considering the completion of the current processing job. **Figure 1**, portray the structure of multi-instance broker and the framework studies.

The multi-instances of the broker work as a swarm of instances in harmony. They are embedded with parallel processing methods and share the same pool of resources or destinations. However, the status of occupied computing elements that has been selected by the prior broker instance, will be tagged as busy or unavailable. This successfully prevents the competition of broker instances to seize the computing element.

Table 1 shows the Multi-Instance Broker's properties used in the development of the processing component in the hierarchical grid structure. All of the properties also portray the scope of this research.

3.2. Multi-Instance Broker Characteristics and Framework

Table 2 lists out the Multi-Instance Broker characteristics applied in this research. The combination of the adopted Multi-Broker characteristics and the newly introduced characteristics, complement and enhance the overall Multi-Instance Broker characteristics.

The Multi-Instance Broker properties and characteristics that have been discussed were taken as the guideline criterion and implemented in the new framework of the Hierarchical Cluster Grid environment as depicted in **Figure. 1**. At the same time **Figure. 1** also shows the location of the Multi-Instance Broker in the framework.

The Performance Information Storage entity is responsible for recording and keeping all of the performance information items. However, this study will not discuss the Performance Information Storage entity in detail , but focuses more on the Multi-Instance Broker creation, experimentation and the performance regarding makespan time.

Table 1. Multi-instance broker properties

Type	Categories	Details
Job model	Job TYPE	Parallel
Data movement	Automatic	-
Scheduling model	Architecture	Hierarchical
	Matchmaking	Dynamic
	Scheduling methods	Grid-oriented
Resource broker	Multi-instance broker	Automatic

Table 2. Multi-instance broker characteristic

Item	Characteristic	Details
1	The self-determination methodology in broker instances number suggestion.	To avoids the grid user involvement in selecting the number of broker instances for workload processing. By Kertesz *et al*. (2009)
2	Batching mode workload submission and used the concurrent processing style.	The multi-broker has the ability to perform resource brokering activities for a batch of jobs that are concurrently executed. By Roy and Mukherjee (2011).
3	Ranking method to classified the resources and broker instances suggestion.	Using the ranking method to categorised the resources and broker instances to provide a suitable option for selection. By Varalakshmi *et al*. (2007).
4	The ranking methods with double layer filtering. By Hwang *et al*. (2010; Lee *et al*., 2011)	Introducing two (2) level filtering selection, for example in the resource or the node selection for more suitable selection. By Rodero *et al*. (2010)
5	The multi-grid manager centre and multi-grid resource modules.	Introducing two (2) entities such as Adaptive Multi-Instance Broker Manager and Broker Instance to segregate the function and to accelerate the processing. By Yang and Hu (2010).
6	To avoid the competition among broker instances for resource selection. By Buyya and Murshed (2002).	Broker instances, has to be designed without broker competition for resource selection because it can lead to the deadlock condition. (Adopted characteristics)
7	To remove the occupied resources from the available or ready resource.	The broker instances will not consider the occupied resources, thus, will result more suitable destination selection. (The new introduced)
8	To prevent on workload submission delays	The highest limit broker instance number introduced is to prevent the grid system to hold the workload at longer times. After all, this will cause the grid queue system burden to increase and disrupt the grid processing performance. (The new introduced)

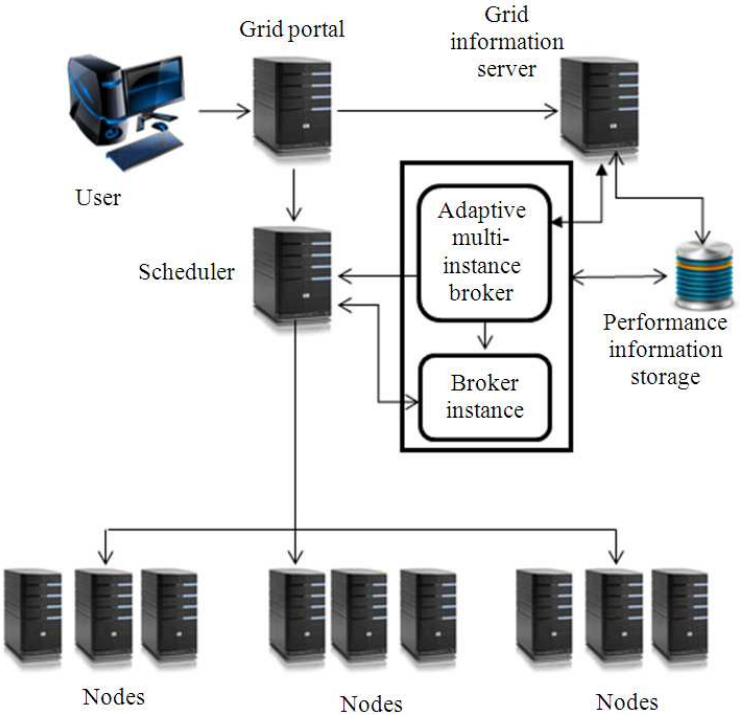

Figure. 1. Hierarchical cluster, grid environment framework

4. MULTI-INSTANCE BROKER CALCULATION AND PROCESS

There are two (2) components supporting the Multi-Instance Broker which are the Self-Adaptive Multi-Instance Broker Manager and Broker Instance. The manager needs to determine the suitable number of broker instances to generate while all of the processes will run by the broker instance.

4.1. Broker Instances Number Determination Strategy Weightage

Determining the number of broker instances is the main role of the first entity called the Self-Adaptive Multi-Instance Broker Manager. After the Self-Adaptive Multi-Instance Broker Manager has decided on the number of broker instances to be used, then notification will be sent to the broker instance entity to produce the broker instances accordingly. This is to ensure that the processing activity is able to commence on time. The Self-Adaptive Multi-Instance Broker Manager serves in self-adaptive heuristic mode. This makes the Self-Adaptive Multi-Instance Broker complies with the dynamic and adaptive running concept. But, how exactly is the number of broker instances determined.

Ideally, there are two (2) items considered in determining the broker instance number to be appointed for each session. They are the resource and workload information of the grid facilities and user request. The information of the resource items is the total computing power value, total background load value and the average computing power value. The average computing power value is used to find the weightage for the resource components. Meanwhile, the total workload number derived from the workload information becomes the indicator to determine the weightage for the workload item. **Table 3**, shows the range of the resource computing power and workload adopted in this research. Finally, both of the weightage are added to obtain the final weightage value to be mapped for broker instance number determination. Nevertheless, both of the ranges are pre-determined and become the rules in a policy respectively.

4.2. Multi-Instance Broker Calculation

This section presents the calculation or formulas involved in determining the number of broker instances. The computing power value and the background load value are captured by the grid system while initializing the grid environment. The total computing power is the sum of the CPU speed from all of the resources. The background load of the computing element in percentage form shows the current running load that may come from the operating systems' activities or updating process which uses the internet connection. The total background load is a cumulative value of all resources. The average computing power of the computing element in **Table 3**, are in Million Instruction Per Second (MIPS) form. The workload represents the number of workload used in each of the sessions Equation 1 to 3:

$$ACPCE = \frac{1}{n}\sum_{i=1}^{n}(a_i - b_i) \quad (1)$$

$$BCMT = \left(\frac{1}{s}\sum_{r=1}^{s} MTr\right) * j \quad (2)$$

$$CMT = j * BCMT \quad (3)$$

The first Equation is meant for the calculation of Average Computing Power of Computing Elements (ACPCE). The n is referred as the total number of computing elements, a is for the CPU speed and b for the background load for each computing element or machine i.

The second Equation is used to calculate the bench Mark of Calculated Makespan Time (BCMT). Parameter s represents the total simulation runs, the MT stands for makespan time for each running session and parameter j is for total workload number used for simulation respectively. The second Equation is only useable if the makespan time data has been recorded.

Table 3. Computing power of resources and workload range

Type	Range	Weight
Average computing power of computing element	4000-5000	0.2
	3000-3999	0.4
	2000-2999	0.6
	1000-1999	0.8
Resource broker	1000-2000	0.2
	2001-3000	0.4

The third Equation is used to calculate the Calculated Makespan Time (CMT) and the result is applicable as a comparison to the current makespan time or simulation running session.

The range method used in this research is to define the weightage of calculation items respectively. The range strategies have been applied by Chang *et al*. (2011) in declaring the Prediction of Execution Time (PET) for all of the CPU. Then, the PET value has been allocated into several levels of ranges against CPU speed. The adopted methodologies used by Hwang *et al*. (2010; Chang *et al*., 2011; Lee *et al*., 2011) has been reused in this research to complement the multi-instance broker invention.

4.3. Self-Adaptive Multi-Instance Broker Scheduling Algorithm (SAMiB)

The Self-Adaptive Multi-Instance Broker Scheduling Algorithm (SAMiB) has been developed to improve the iHLBA makespan time performance. Thus, the SAMiB which was developed with a broker instance strategy focuses to manage the creation of multi-instances of the broker and implementation. The SAMiB algorithm which was developed with the multi-instance broker has self-adaptive capabilities to environmental changes. SAMiB also considers the background load utilization accumulated from several resource items, namely the CPU, memory and network bandwidth that was originally introduced by the iHLBA algorithm. The processing limitation is controlled by the threshold that holds the upper limit value which manages the balancing of the load distribution. The simulation algorithm steps as shown in **Figure 2**.

The multi-instance broker concept has been introduced for this research to assist the resource broker management in the hierarchical cluster grid structure. This invention is found to be interesting and has various advantages to be highlighted. Below are the justification of the invention and its benefit. Firstly, the multi-instance broker development concept was based on replication and extension characteristics of the original resource broker and to employ similar capabilities. Certainly, the duplication will not drop any functionality or features that the resource broker has. Hence, the multi-instances of the broker should be able to work as smoothly as its parent.

Therefore, job processing through this innovation will run in parallel among the broker instances. Thus, more computing elements or nodes will be able to be selected in minimum time and will be working simultaneously. This will prevent the delay time of selecting the computing elements from becoming longer. As discussed earlier, the multi-instances of the broker will work as a swarm and in parallel in a harmonious strategy. The fact is, the occupied computing element will not be listed and is tagged as busy in the next selection process until the current processing completes. Hence, there will be less competition among the broker instances to choose the appropriate computing element. This also prevents or reduces the possibility of the grid system to suffer from a case of deadlock**.**

5. EXPERIMENT DESIGN

Table 4, lists down the simulation parameter properties of the grid computing environment derived from the iHLBA algorithm experiment. Meanwhile, this research uses a number of parameters for filtering purposes and decision making as shown in **Table 5**.

All of the detailed calculations and mathematical formulas referred to the research work done by Lee *et al*. (2011) for the iHLBA algorithm. **Table 6**, contains the background load composition randomly generated and following the specifications based on research needs. These specifications become the research treatment for the algorithm featured on four (4) different groups with different configurations of background load composition. Similarly, the CPU is also based on different configurations too.

The background load is the accumulated value of the CPU, memory and network bandwidth utilization determined at the initial stage of the simulation. In this case, the background load utilization value is assumed to be unchanged throughout the simulation session. But, when the submitted job processing is finished, the computing element load will reset to the initial stage value. However, the simulation experiment scenarios used are as in **Table 6**.

There are four (4) types of resource mixes used in this research which are Type A, Type B, Type C and Type D. Each of the types has different sets of configuration on the percentage of background load that are cumulatively less than ten (10), cumulatively more than ten (10) and the CPU speed which represents the power of the computing elements. The background load for each computing element is contributed by the CPU, memory and the network bandwidth utilization. The background load configuration used in this research in accordance to the study done by Yahaya *et al*. (2013).

Table 4. Computing power of resources and workload range

Number	Parameter	Value
1	Size of task (MI)	300000-500000
2	Number of nodes per cluster	10
3	Number of clusters	10
4	Processor speed (MIPS)	500-5000
5	Memory size (MB)	500-1000

Table 5. Parameters for filtering purposes and decision making

Number	Parameter	Value	Detail
1	BGL CPU	% Utilization	Background load for CPU
2	BGL Mem	% Utilization	Background load for memory
3	BGL Net	% Utilization	Background load for network.
4	Load	CE Load	Current load hold by the each computing unit.
5	ACL	Cluster load	Average cluster load in percentage
6	AL	System load	Average system load.
7	Sigma	Standard deviation	The workload distribution value
8	Threshold	Simulation upper limit	Setting up the limitation of simulation.

Table 6. Experiment scenarios with different background load composition

Resource mixed	% of background load >10	% of background load <10	CPU speed
Type A	60	40	Random
Type B	30	70	Random
Type C	50	50	Controlled randomness
Type D	30	70	Controlled randomness

6. RESULTS AND DISCUSSION

Simulation properties discussed in the previous sections are the guidelines for environment implementation and are also responsible for controlling the simulation boundaries. There are two (2) scheduling algorithms called improved Hierarchical Load Balancing (iHLBA) and Adaptive Multi-Instance Broker Scheduling (SAMiB) used to run the simulation in this research. **Figure 2**, describes the SAMiB algorithm. The experiments undertaken were made to comply with the simulation properties. As the simulations progressed, the algorithm was not generating constant results, therefore this research was based on the average result.

The experiment undertaken at this point is for two thousand (2,000) jobs only. Both of the chosen algorithms had run all resource types explained earlier. Each of the resource mix encompasses various computing power and background load properties. Generally, the SAMiB performance has surpassed the iHLBA performance of makespan time.

The SAMiB algorithm has a decreased makespan time by 13.58% over the iHLBA algorithm for the Type A resource mix. This resource mix type consists of random computing power, but posses 60% higher background load and 40% lower background load. Meanwhile, through the resource mix for Type B, SAMiB has shown an improved performance against the iHLBA compared to the performance for Type A. Under the Type B resource mix, the SAMiB shows a decline of makespan time to 13.67%. This is due to the composition of a low background load which is at 70% and a high background load of only 30% from the total resource. **Figure 3 and 4**, depicts the results for two thousand (2000) jobs over different resource mix types.

The SAMiB algorithm had obtained good performance in the Type C resource mix segment and has produced a decrease of 14.93% in makespan time over the iHLBA algorithm. This resource mix has a balanced background load ratio between the lower and the higher range of background load and equipped with the second highest computing power which contributed to a better outcome. The final running experiment for two thousand (2,000) jobs is for the Type D resource mix. The Type D resource mix segment was equipped with the highest computing power compared to other type resource mixes, but has approximately 70% lower background load composition and only 30% of the resources in higher background load.

User submits a request to the grid system.
Grid system initializes the infrastructure and workloads.
Multi-Instance Broker Manager
Decide the number of broker instances to be used
Broker Instance Entity
Do the broker instances creation
Broker instances ready to aid the processing
Job Check
If null-Update all parameters.
Check the job again
If null-Terminate the processing session.
Else-Process the job.
Else-process the job.
Select the job.
Check and run the scheduler policy
Check the average cluster load on entire setup.
Compare with the threshold value.
Remove overload cluster.
Select the cluster with highest computing power.
Select the computing element with highest computing power
Job send to the selected computing power
Update the local average cluster load.
Change the computing element status to busy.
Repeat until end of job.

Figure. 2. The adaptive multi-instance broker scheduling algorithm

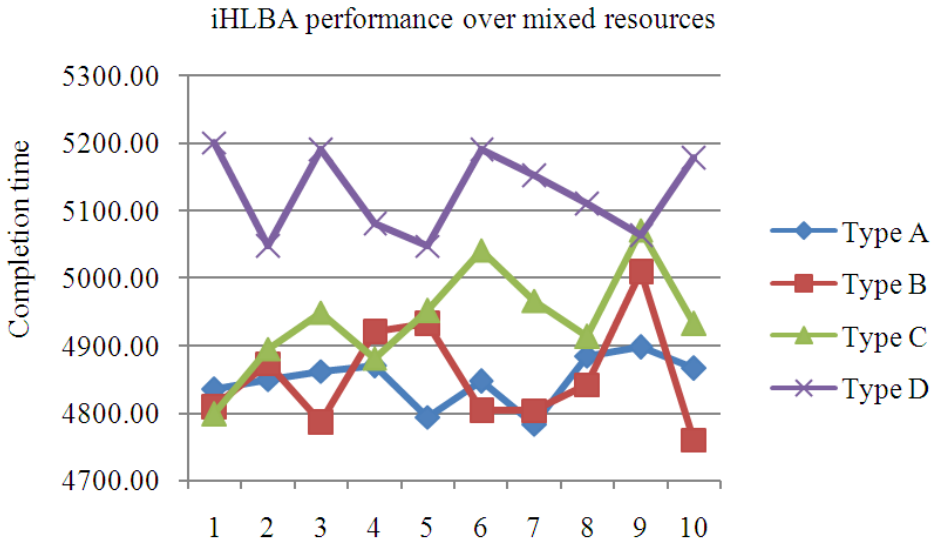

Figure. 3. iHLBA performance based on 2000 jobs over different resource type

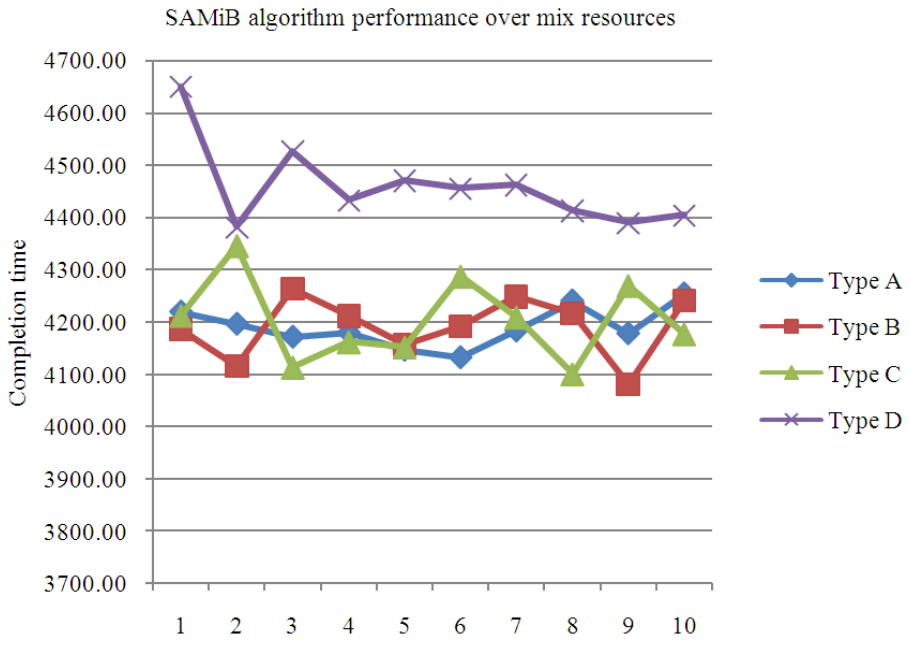

Figure. 4. SAMIB performance based on 2000 jobs over different resource type

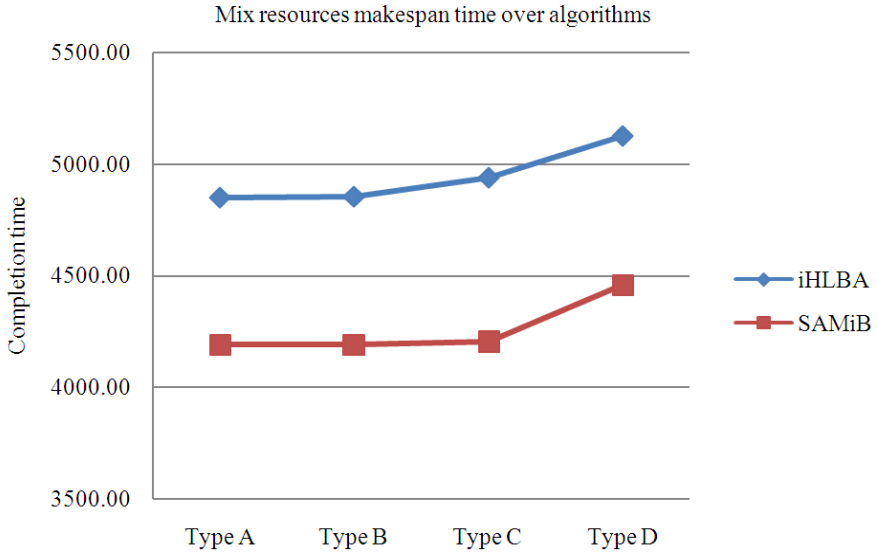

Figure. 5. iHLBA and SAMiB performance comparison over multiple mix resources for 2000 jobs

The SAMiB generate less makespan time at 13.02% over the iHLBA but shows a decrement of makespan time compared to the Type C resource mix. This was caused by 40.6% of the computing element in the range of 4000 MIPS to 5000 MIPS posses higher background load close to 30% of the utilization.

Consequently, the SAMiB has surpassed the performance of iHLBA under the two thousand (2,000) jobs segmentation. The results show that SAMiB algorithm had successfully overcome the iHLBA performance in each of the mix resource types. The introduction of multi-instance of the broker through a

SAMiB algorithm in hierarchical cluster, grid structure has produced better performance over the iHLBA algorithm in the 2000 jobs segmentation. Based on this observation, the conclusions that can be made are that the dispersion of the background load is the obstruction for the algorithm to achieve a better performance in the experiment for 2000 jobs. Although the resource mix has a bigger composition of computing power, however the allocation and dispersions of higher background load on the computing power contribute to the fluctuation of the results. The performance comparison of iHLBA and SAMiB over multiple mix resources for 2000 jobs is shown in **Figure 5.**

7. CONCLUSION

The main focal point is to develop a multi-instance broker concept to extend the capabilities of the grid resource broker and its execution in the hierarchical cluster grid environment. The investigation concludes by initiating experiments that compare the two algorithms with mixed resources, the comparison of the background load allocation and the makespan time among algorithms. It is also highlighted that the SAMiB has better capabilities for processing the workload compared to iHLBA algorithm.

The exact properties that have been used to substantiate the research are described in section 4.1. At this point the SAMiB has accomplished to overcome the iHLBA algorithm for experiment of two thousand (2,000) jobs. Based on the results achieved, this proves to be a positive outcome and is pivotal in this research. It is also interesting to note that the diversified background loads have contributed to the differences in simulation results altogether. This signifies the impact of various compositions of the background load itself.

This research has succeeded to clarify the adoption of the multi-instance broker concept in the hierarchical cluster grid environment. The modification or innovation called the multi-instance broker has enabled the grid broker to be replicated with the extension of the service capabilities. From the perspective of the automation concept, the multi-instances of the broker are fully equipped with self-adaptive methods which enable the system to determine and produce the appropriate number of broker instances to be used in a simulation session.

Finally, the concept of the Multi-broker which has been applied in multiple domains can be adopted into the environment of the cluster grid. Upon concluding this study, it has been agreed that this research does not consider the grid scalability in its implementation; therefore future research would consider investigating the performance capability of SAMiB to process a more sizable amount of job allocation.

8. ACKNOWLEDGEMENTS

The author would like to largely thank the Department of Communication Technology and Network, Faculty of Computer Science and Information Tehnology and namely Dr. Rohaya Latip for her valuable guidance.

9. ADDITIONAL INFORMATION

9.1. Funding Information

Financial support given to this research by the Malaysian Public Service Department.

9.2. Author's Contributions

Bakri Yahaya: PhD candidate and writer.
Rohaya Latip: Main supervisor.
Azizol Abdullah: Committee members.
Mohamed Othman: Committee members.

9.3. Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

10. REFERENCES

Afgan, E., 2004. Role of the resource broker in the grid. Proceedings of the ACM-SE 42nd Annual Southeast Regional Conference, Apr. 02-03, Huntsville, New York, pp: 299-300. DOI: 10.1145/986537.986608

Buyya, R. and M. Murshed, 2002. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. Concurrency Comput. Pract. Exp., 14: 175-1220. DOI: 10.1002/cpe.710

Chang, R.S., C.F. Lin and J.J. Chen, 2011. Selecting the most fitting resource for task execution. Future Generat. Comput. Syst., 27: 227-231. DOI: 10.1016/j.future.2010.09.003

Hwang, J., J. Park and J. Altmann, 2010. Two risk-aware resource brokering strategies in grid computing: Broker-driven Vs. User-driven methods. Proceedings of the 4th International Conference on Information Systems, Technology and Management, Mar. 11-13, Springer, Bangkok, Thailand, pp: 187-197. DOI: 10.1007/978-3-642-12035-0_19

Kandagatla, C., 2013. Survey and taxonomy of grid resource management systems. University of Texas, Austin.

Kertesz, A. and P. Kacsuk, 2007a. A Taxonomy of Grid Resource Brokers. In: Distributed and Parallel Systems: From Cluster to Grid Computing, Kacsuk, P., T. Fahringer and Zsolt Németh (Eds.)., Springer US, ISBN-10: 978-0-387-69857-1, pp: 201-210.

Kertesz, A. and P. Kacsuk, 2007b. Grid meta-broker architecture: Towards an interoperable grid resource brokering service. In: Euro-Par 2006: Parallel Processing, Lehner, W., N. Meyer, A. Streit and C. Stewart (Eds.)., Springer Berlin Heidelberg, ISBN-10: 978-3-540-72226-7, pp: 112-115.

Kertesz, A., Z. Farkas, P. Kacsuk and T. Kiss, 2009. Grid interoperability by multiple broker utilization and meta-broking. In: Grid Enabled Remote Instrumentation, Davoli, F., N. Meyer, R. Pugliese, S. Zappatore (Eds.)., Springer US, ISBN-10: 978-0-387-09662-9, pp: 303-312.

Lee, Y.H., S. Leu and R.S. Chang, 2011. Improving job scheduling algorithms in a grid environment. Future Generat. Comput. Syst., 27: 991-998. DOI: 10.1016/j.future.2011.05.014

Rodero, I., F. Guim, J. Corbalan, L. Fong and S. M. Sadjadi, 2010. Grid broker selection strategies using aggregated resource information. Future Generat. Comput. Syst., 26: 72-86. DOI: 10.1016/j.future.2009.07.009

Roy, S. and N. Mukherjee, 2011. Efficient resource management for running multiple concurrent jobs in a computational grid environment. Future Generat. Comput. Syst., 27: 1070-1082. DOI: 10.1016/j.future.2011.04.008

Varalakshmi, P., S. Thamarai and M. Pradeep, 2007. A multi-broker trust management framework for resource selection in grid. Proceedings of the 2nd International Conference on Communication Systems Software and Middleware, Jan. 7-12, IEEE Xplore Press, Bangalore, pp: 1-6. DOI: 10.1109/COMSWA.2007.382492

Yahaya, B., R. Latip, A.H. Abdullah and M. Othman, 2013. Workload utilization dissemination on grid resources for simulation environment. Proceedings of the 3rd International Conference on Research and Innovation in Information Systems, Nov. 27-28, IEEE Xplore Press, Kuala Lumpur, pp: 463-468. DOI: 10.1109/ICRIIS.2013.6716754

Yang, C.T. and W.J. Hu, 2010. Design and Implementation of a Multi-Grid Resource Broker for Grid Computing. In: Data Driven e-Science: Use Cases and Successful Applications of Distributed Computing Infrastructures, Lin, S.C. and E. Yen, (Eds.)., Springer US, ISBN-10: 978-1-4419-8013-7, pp: 251-262.