A Journal of Software Engineering and Applications, **2012, 5, 149-156**
doi:10.4236/jsea.2012.512b029 Published Online December 2012 (http://www.scirp.org/journal/jsea)

Fuzzy Logic–Based Scheme for Load Balancing in Grid Services

Tarek Helmy*, Hamdi Al-Jamimi, Bashar Ahmed, Hamzah Loqman

Information and Computer Science Department, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31216, Mail Box, 413, Saudi Arabia; "On leave from College of Engineering, Department of Computers Engineering & Automatic Control, Tanta University, Egypt".
Email: helmy@kfupm.edu.sa

Received 2012

ABSTRACT

Load balancing is essential for efficient utilization of resources and enhancing the responsiveness of a computational grid, especially that hosts of services most frequently used, i.e. food, health and nutrition. Various techniques have been developed and applied; each has its own limitations due to the dynamic nature of the grid. Efficient load balancing can be achieved by an effective measure of the node's/cluster's utilization. In this paper, as a part of an NSTIP project # 10-INF1381-04 and in order to assess of FAQIH framework ability to support the load balance in a computational grid that hosts of food, health and nutrition inquire services. We detail the design and implementation of a proposed fuzzy-logic-based scheme for dynamic load balancing in grid computing services. The proposed scheme works by using a fuzzy logic inference system which uses some metrics to capture the variability of loads and specifies the state of each node per a cluster. Then, based on the overall nodes' states, the state of the corresponding cluster will be defined in order to assign the newly arrived inquires such that load balancing among different clusters and nodes is accomplished. Many experiments are conducted to investigate the effectiveness of the proposed fuzzy-logic-based scheme to support the load balance where the results show that the proposed scheme achieves really satisfactory and consistently load balance than of other randomize approaches in grid computing services.

Keywords: Fuzzy Logic; Load Balancing; Grid Computing.

1. Introduction

Grid computing can be considered as a type of parallel and distributed systems that enables the selection, distribution, and aggregation of resources dynamically at run time depending on their availability, capability, and performance. It focuses on large-scale resource sharing and distributed system integration for the sake of effective utilization and high-performance orientation. Indeed, at the service level of the grid, software infrastructure, the workload and resource management are essential provided functions. In turn, distributed systems have become a natural setting in various environments either for business or academia [1,14,17-20]. In like these systems settings, tasks arrive to the different nodes of the cluster randomly. This random fashion leads to a non-uniform distribution of the workload among different nodes of the cluster. Loading imbalance is harmful to the whole system performance in terms of the mean response time of tasks and resources utilization. Therefore, load balancing tries to increase system throughput by making all processors as busy as possible. The objective can be achieved by two ways: either by avoid sending more tasks to the overloaded nodes and keep checking the nodes' states periodically, or taking a migration decision to migrate some tasks from the overloaded nodes to the lightly loaded nodes for the sake of improving the system performance as a whole. The load balancing dilemma in a distributed computing environment has been addressed extensively by many studies and researches. Based on that, many algorithms have been introduced to tackle the load balancing problem. All kinds of these algorithms fall into one of the following two categories: dynamic or static [2, 18]. Dynamic load balancing algorithms exploit the information describing the state of the system in order to improve the accuracy of load balancing decision. Therefore, the dynamic algorithms are appropriate to be utilized in practical situations. However, this kind of algorithms should have the ability to collect the information about

*Corresponding author.

the current state of the system, to store the collected information and to analyze them. Thereby, the dynamic algorithms may cause additional overhead computations for the system. On the other hand, the static algorithms don't exploit such information. In static algorithms, the decisions are taken based on a priori knowledge of the system. Thus, the static algorithms don't have the ability to deal with the dynamic changes of such environments. An efficient load balancing scheme is needed for grid computing environment. The effective scheme needs to know the global system's state such as the distribution of the workload. However, the load model of the distributed systems tends to change dynamically, and hence, it is very complicated to model the system accurately. Thus, precise and fast load balancing scheme is an essential factor in increasing the efficiency and performance of grid computing components. However, many factors lead to ambiguous information about the clusters/nodes states, i.e. lack of shared memory among independent clusters. This ambiguity leads to uncertainty in decision for load balancing. Another important aspect is that the states of the clusters/nodes can be changed rapidly. Therefore there is always some amount of uncertainty in the state information used for making a decision. Hence it is necessary that the decision making process deals with these uncertainties. Fuzzy logic is one of the methods that deal with the uncertain information [4,5,13,15,16]. In this paper, in order to tackle the problem of load balancing in the grid computing environment where uncertainty is unavoidable, we utilize a Fuzzy Inference System (FIS)-based approach to model those contributed factors to determine the nodes' states and thereby the clusters' states. In some previous studies, both length of the ready queue and CPU utilization have been used to provide a good load indicator. In the case of multiplicity of resources, a good measure can be considered like a linear combination of the length of all resource queues. In this paper we use four parameters as input variables for the sake of more accurate fuzzy-logic based approach. We consider a continuous stream of independent jobs which arrive to the system and are stored into a single job queue. We assume that all jobs are not preemptable, and each job is described by its ID, submission time, deadline time, an estimation of its execution time and the number of resources it requires. The computing grid is composed of many independent clusters of homogenous processing machines. Because of the lack of a mechanism in the computing platform to notify configuration changes to the system, we assume to have a static number of resource instances available to cluster's processing node. The rest of the paper is organized as follows: Section 2 describes technical background about fuzzy logic concepts. Section 3 introduces the related work. The proposed fuzzy-logic based scheme is demonstrated in Section 4. The implementation details of the proposed scheme are described in Section 5, while Section 6 introduces the simulation environment and discusses the obtained results. Section 7 concludes the paper and introduces some trends for future work.

2. Technical Background

In this section, we introduce a brief background about the concepts used in this paper. Fuzzy logic methodology can be utilized in solving the problems that are complex to be analyzed quantitatively or in the case of natural phenomena that are not easy to be modeled mathematically [6]. IF-THEN rules are used in the FIS where the output is concluded from the fired rules based on the given inputs [3,4]. The system parameters are modeled as linguistic variables based on expert knowledge; also the corresponding membership functions are designed for each parameter. Therefore, fuzzy logic theory can be employed even for the nonlinear systems suffering from uncertainty and complexity. That complex uncertain system can be modeled effectively based on fuzzy logic without need for complicated mathematical models [5]. A general FIS engine mainly involves the following components as shown in Figure 1:

- *The fuzzifier* receives the inputs which are the node's information in the proposed model case. The fuzzifier maps the given numerical inputs to fuzzy sets and linguistic variables. Where the defined linguistic variables can be matched with the fuzzy rules premises that are predefined in the rule base of the specified application.
- *The rule base* includes a set of linguistic rules designed in the form (if-then rules). These rules define the consequent of the model in terms of the given linguistic variables such as low, moderate and high. In addition, it specifies the type and number of the used membership functions of input and output parameters.
- *The fuzzy inference engine* is considered the central part of the fuzzy system. In this stage, in order to produce the intended output, the inference engine is applied to a set of rules included in the fuzzy rule base. This procedure involves many steps as follows: *first*, it matches the linguistic variables of the input with the rules' premises. *Second*, it activates the matched rules in order to deduce the resultant of each fired rule, and *finally* it combines all consequents by using fuzzy set union in order to generate the final output which is represented as fuzzy set output.
- *The defuzzifier:* as described in the previous phase, the output is produced as a linguistic variable, which is fuzzy and can be interpreted in different ways. Therefore, the fuzzy set output in this stage is converted to a

crisp output, which is a numerical value.

3. Related Work

The issues related to load balancing in the distributed systems have been addressed broadly by many studies and researches. In the literature, many studies have proposed the using of fuzzy logic theory to solve the load balancing problem. Since the load balancing problem can be considered as NP-hard [6], in this case heuristics are sought to tackle this problem.

Nejad et al. [8] proposed a fuzzy logic-base load balancing in centralized distributed system. The efficiency of proposed load balancing is studied in OPNET simulation environment. The lengths of sent packet and service rate in each node have been considered non-constant. The current load of the system and waiting time for the last processed task in each node are considered as the fuzzy controller inputs. Based on the weight assigned for each node, a percentage of existing tasks is assigned to that node. The result of the experiments in the states of constant and variable nodes, and constant and variable tasks, indicates that this algorithm performs much better than both static and dynamic algorithms in terms of throughput, drop rate and response time.

Bey et al. [9] introduced a model that can be used for predicting the future CPU load in a dynamic environment. The proposed model is for single-step-ahead CPU load prediction. The multiple local Adaptive Networks-based Fuzzy Inference Systems (ANFIS) predictors were used to build the introduced prediction model. In turn, the predictors controlled via the Naïve Bayesian Network (NBN) inference between clusters states of CPU load time points obtained by C-means clustering process. The ANFIS models were used to carry out short-term accurate and mid-term reliable prediction. The selection was based on the basis of divide-and-conquer principle that divides a CPU load time series into several clusters. Afterwards, those clusters can be used separately. Indeed, in that case, the ANFIS system behaved accurately and gave a reasonable performance.

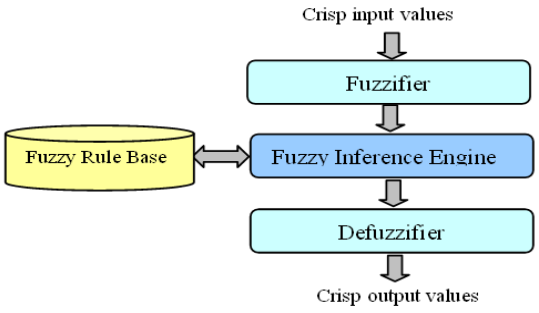

Figure 1. The general architecture of the fuzzy logic inference engine.

Revar et al. [10] focused on analyzing load balancing requirements in a grid environment and proposed design and implementations of innovative load balancing system for grid environment using machine learning. Their technique balances the load dynamically. The technique uses initial load information stored in the database at the initial level. When load imbalance takes place, the current load information is collected and stored as raw data. Afterwards, various machine learning algorithms have been used to process and analysis the recorded data. As a final step, the rules automatically generated by data mining techniques and used for migrating jobs for load balancing.

K. Ming and C. Hsun [11] designed and implemented a load-balancing system based on fuzzy logic control in loosely coupled distributed system. The run-queue length and CPU utilization were used as input variables. They used six workstations running different UNIX OS to build a heterogeneous computing environment. These workstations are located on different workstations and connected by communication network. Based on their experimental results, the fuzzy logic model for load balancing decreased effectively the amount of communication traffic of the network and provided substantial enhancement in the overall performance.

Rantonen et al. [12] designed and implemented a fuzzy expert system for load balancing in a symmetric multiprocessor environment. The novelty of their algorithm is the use of the algorithm for load balancing only on demand, instead of using the algorithm periodically. Thereby the computations overhead can be reduced that makes the algorithm more fair and fast. Two parameters are considered as input to the proposed model: the number of threads per processor and the sum load of ready queues. They claimed that, the evaluation results proved the best load balance using different techniques.

To the best of our knowledge, none of the previous proposed fuzzy logic-based techniques has used more than two parameters to evaluate the workload. In this paper, to evaluate the node's workload, we use four parameters (Node's ready queue length, burst time, CPU utilization, and the available resources needed to accomplish the assigned tasks), for the sake of more accuracy. Moreover, two fuzzy-logic based models are used in different levels which are node-level and cluster-level.

4. The Proposed Fuzzy-Logic-Based Load Balancing Scheme

In the proposed scheme, the load balancing task is designed to be performed at two levels: *i)* at the cluster level by global manager, and *ii)* at the node level by local manager. The grid consists of many clusters, where sev-

eral nodes constitute one cluster. In each cluster, there is a local manger to evaluate the states of the nodes belonging to its cluster and to perform local load balancing in its own cluster. The nodes of each cluster only communicate with the local manager. Assuming that, the task allocated to each cluster/node will be processed in that cluster/node and will not be relocated. Global manager communicates with the local manager of each cluster. The states of nodes in each cluster are sent to the global manager, to specify the state of the corresponding cluster. The structure of the fuzzy logic-based scheme is demonstrated in Figure 2. In this scheme, the workload state for the nodes and the clusters are evaluated. This evaluation is carried out in the grid and the cluster levels by the global and the local managers respectively. Based on the generated states, a newly arrived task will be assigned to the lightly loaded cluster. Then, based on the states of the nodes, the arrived tasks will be assigned to the lightly loaded node/nodes. Deciding the state of the nodes is made by fuzzy logic-based model, and thereby the clusters states are evaluated. Four parameters as mentioned above are collected from the current information tasks and used as inputs to the fuzzy model. The process of load balancing is handled in different stages as followings:

4.1 Data Collection Stage

Fuzzy logic-based scheme needs data about each node to determine the state of that node. The parameters collector collects the data about each node from the table contains the information about the current tasks. This collected information is then passed to the FIS that performs the needed analysis and takes a decision about the node's state. All the nodes sates are sent from time to time to the local manager that carries out the local balancing. To exchange the data, each of the nodes notifies its manager about its load state periodically. That is, to specify the current state of the grid, clusters and nodes, the balancing data in the system, is updated without any request. Each local manager does the task of collecting the node's data, in its group, by using the data collector. Then, based on the data received from the nodes, the fuzzy logic scheme evaluates the workload of the nodes, determines their states, and sends the results to the local manager. Afterwards the local manager stores the workload states of the nodes and sends a copy to the global manager. Based on the data received from local managers, the global manager calculates the workload of each cluster and decides about its global state in general, and also sends the clusters states to the grid manager.

4.2 Specifying the Workload Sate stage

In this stage, the workload state of each node is determined and the decision on load distribution is taken. We consider four different inputs variables to the fuzzy logic model, namely: the node's ready queue length, the estimated burst time of the tasks assigned to the node, the CPU bound for each task, and the available resources for each node. Three fuzzy subsets have been considered for each input to determine the fuzzy rules in order to get the output.

4.3 Distribution Satge

The grid manager, based on the inferred states of different nodes and clusters, ranks the clusters and nodes based on their availability. In that sense, the distributor routes the arriving tasks to the appropriate cluster and node based on the rank. In this stage, the grid distributor utilizes the workload information about the different clusters to decide straightforwardly which cluster is lightly loaded. It then distributes the new arriving tasks to the clusters/nodes based on their workload states collected in advance. Finally, the local manager inside the target cluster forwards the arrived tasks to the lightly loaded nodes based on their inferred states.

5. Implementation of the Proposed Scheme

Using Matlab's FIS toolbox, we implemented two fuzzy logic based models. The first model to evaluate the node's workload based on four parameters explained above while the second fuzzy model is to evaluate the clusters workload based in the nodes states that belong to the target cluster. Following, we will detail the design and implementation of each model by introducing its inputs, rules, output, and accuracy measure when training and validating each model.

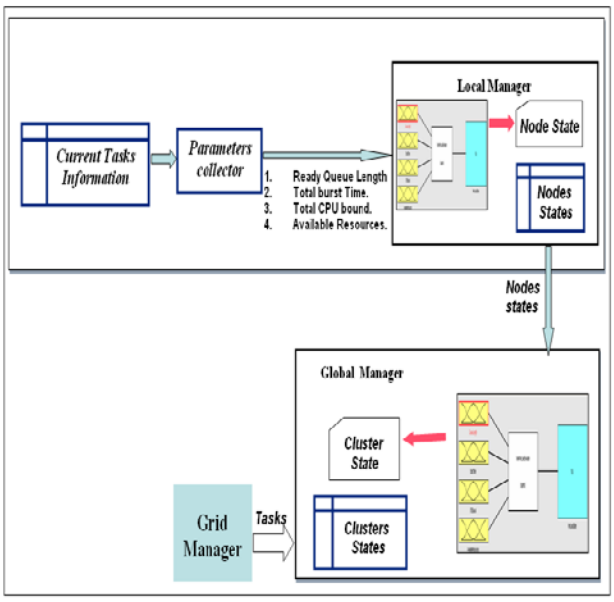

Figure 2. The architecture of the proposed scheme

5.1 Fuzzy Logic Model – Node Level

Each input variable is represented by five Membership Functions (MFs) of Gaussian type except the last parameter only three MFs are used. For example, when the length of queue is 45, this value has around 0.05 degree of membership to the function as short and about 0.75 degree of membership to the function as moderate and also has zero degree of membership to the function is long. Thus, we can classify the queue length of 45 as moderate. For all the inputs the ranges differ from one to another. For instance, the range of queue length is [0 -100]. However, when the queue length is greater than the maximum in the range which is 100, the MF is defined as very long because the degree of membership to the function very long is 1. The MFs for the rest of the input variables operate in a similar way. In general, the fuzzy rule base includes a set of linguistic rules. Those rules represent the communitarians between the different input parameters, where each parameter is represented by linguistic variables. That is, the rules promises represent the input parameters, where the consequent of each rule indicates the resultant output. The rules are triggered by mapping the input parameters to the input functions which might be one or more. In our system, based on the given inputs to the FIS and the degree membership of each, 375 rules are obtained; two of them are shown in Figure 3.

For example, in rule1, if the number listed in the ready queue is low, and the CPU utilization is small, and the available resources for the corresponding node to achieve the assigned tasks are high, then the fuzzy output which indicates the node state, would be very light. While in rule 2, if the ready queue length is high and the CPU is very heavily utilized and the resources needed are not too much, then the node state would as heavy loaded node and no more tasks will be send to it. The output of the fuzzy model represents the workload value of the corresponding node based on the given parameters. For training and testing the proposed fuzzy model to evaluate the node's workload, we generated a dataset consists of the input parameters and has one output represents the workload based on the given inputs. The dataset consists of six hundred data samples. To build the fuzzy logic based model, the dataset is divided into training set and validation set where 70 % of the samples are used for training the model and 30% are used for validation. Figure 4 shows the actual and predicted output for both training and testing stages. It is obvious from Figure 4 that no much difference between the actual and predicted workload for each node. The accuracy measure for the model is measured in terms of error measure, where the Root Mean Square Error (RSME) for training and validation are 0.000291 and 0.00398 respectively.

5.2 Fuzzy Logic Model – Cluster Level

Similarly for this model, the input variables are represented by five MFs of Gaussian type. Since the input value is the node's sate, the five MFs representing the load state are 'very light', 'light', 'moderate', heavy' and 'very heavy'. The range of each parameter is between 0 and 1 representing the workload for each node assuming that we have twenty nodes in each cluster. The output of the fuzzy model represents the workload state of the corresponding cluster based on its nodes states. We trained and evaluated the proposed fuzzy logic–based model at the cluster level by using generated dataset which consists of 300 data points. Each data point has 4 inputs represents the cluster's nodes states and one output which indicates the workload of the cluster. 70% of the samples were used for training the model, and 30% of the samples have been used for validation. Figure 5 shows the actual and predicted output for both training and testing stages. It is obvious from Figure 5 that, no much difference between the actual and predicted workload for each cluster. The accuracy measure for the model is measured in terms of error measure, where RSME were as follows: 0. 0.00095 for training and 0.00329 for validation respectively.

6. Simulation and Experimental Results

In the simulation, we considered that a grid composed of four independent clusters of homogenous processing nodes. Because of the lack of a mechanism in the computing platform to notify configuration changes to the system, we assume to have a static number of resource instances available to each cluster's processing node. In addition, we consider that the number of nodes equally distributed among the different clusters where each cluster consists of twenty nodes. We consider a continuous stream of independent tasks which arrive to the system and are stored into a single job queue. All the arrived tasks are not preemptable, and each task is described by five attributes: the task ID, task's submission time, task's deadline time, an estimation of its execution time, and it's CPU-bound. Moreover, a dataset contains information about 5000 tasks was generated and used. In the first run of the simulation, a

1. If (ReadyQueueLength is VeryShort) and (BurstTime is VeryLow) and (CPUBound is VeryLow) and (AvailableResources is Low) then (WorkloadSatae is out1mf1)
2. If (ReadyQueueLength is VeryHigh) and(BurstTime is VeryHigh) and (CPUBound is VeryHigh) and (AvailableResources is High) then (WorkloadSatae is out1mf375)

Figure 3. Rules for fuzzy inference system

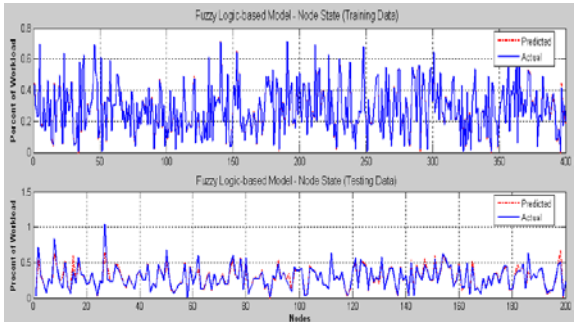

Figure 4. The fuzzy model – Node state (training and testing accuracy measures)

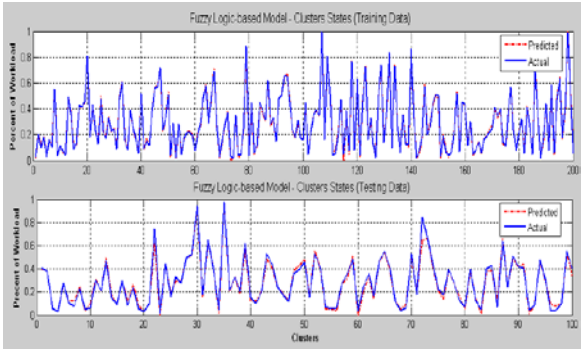

Figure 5. The fuzzy model – Cluster state (training and testing accuracy measures)

random algorithm has been used to assign around 800 tasks to the available clusters and nodes randomly without assessing the clusters and nodes workload states. That is, all the jobs entered the system in a uniform way.

Figure 6 and 7 demonstrate the distribution of the workload among the different clusters and various nodes. It is obvious form the charts that the workload was not distributed in a fair manner where some nodes have workload more than other. It is noticeable, in the different clusters the workload was not equally distributed among the different nodes in each cluster. For instance, in the first cluster the nodes 2 and 5 have the heaviest workload, whereas the nodes 1 and 20 in the same cluster have much less workload. In the same way, among the different clusters, the workload was not distributed in a balanced way as shown in Figure 7 which shows the total number of tasks assigned to each cluster.

In Figure 7, the second and third clusters have almost equal workload. Similarly, the cluster 1 and 4 have almost identical load. Nevertheless, the number of tasks assigned to cluster 2 almost double the number of tasks assigned to the first cluster. In the same way, the workload assigned to the forth cluster equal the half of the work assigned to the third cluster. After that, we have considered the case with an initial workload of 800 tasks that have been assigned randomly to different clusters and nodes. Then the proposed algorithm was used to evaluate the state of each node and thereby to evaluate the clusters workload state. Based on the available information, the arriving tasks are assigned to the clusters and nodes that have the lower load.

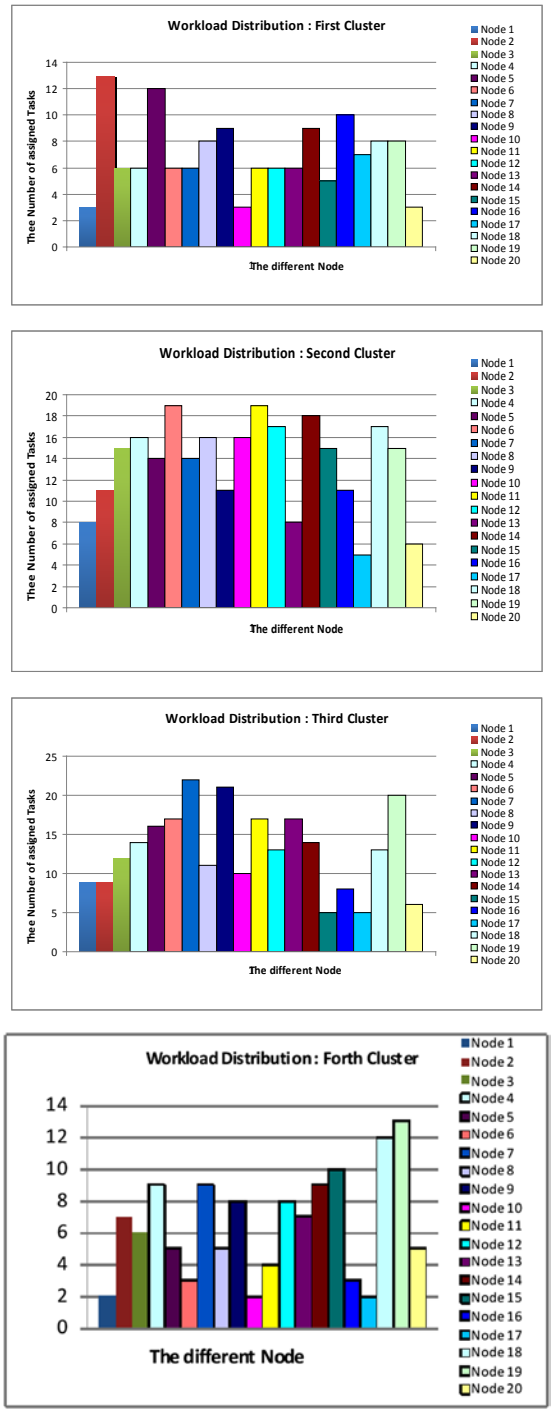

Figure 6. The workload of nodes in different clusters at a certain amount of time using random distribution algorithm (after assigning 800 tasks)

Figure 8 shows the system state after using the proposed algorithm to evaluate the workload states of the clusters, and to distribute the new tasks. Clearly the tasks are distributed in a balanced way among the different clusters; where the number of tasks assigned to each cluster doesn't vary much form the others. Moreover, the distribution of workload among the different nodes inside the same cluster is reasonable.

Even though some nodes may be assigned more work, most of the nodes have almost identical workload. For instance, Figure 9 demonstrates the workload states of the nodes related to the second and third cluster. It is obvious the workload is distributed in approximately balanced manner among the different nodes related to this cluster. In order to evaluate the performance of the proposed scheme, the load balancing of the overall system was monitored during different points of time. Figure 10 shows the states of the different clutters during different times. Obviously form Figure 10, during observed intervals of time the workload was distributed in almost balanced way. With the passing of time, despite of the steady increase of the arriving tasks, the workload was distributed in a balanced way. That is, the clusters have almost identical workload. Comparing with the random algorithm, shown in Figures 6 and 7, the proposed fuzzy logic-based scheme achieves more efficiency in distributing the arriving tasks based on the workload states of the available nodes and clusters.

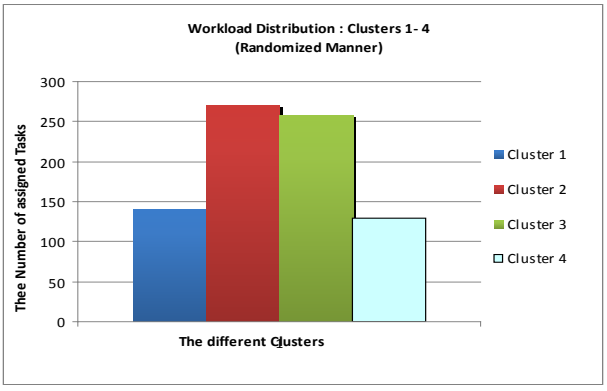

Figure 7. The workload distributed randomly among different clusters at a certain amount of time (after assigning 800 tasks).

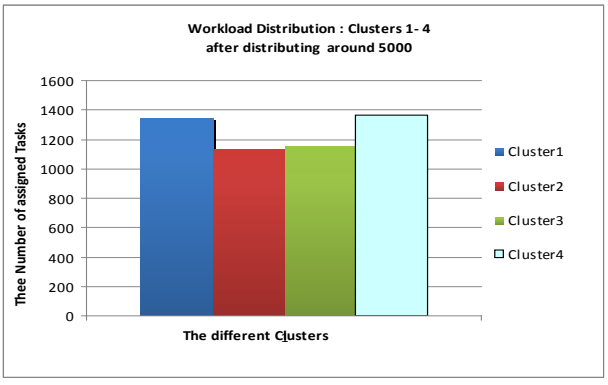

Figure 8. The workload states of the clusters after distributing the tasks by applying the proposed algorithm

7. Conclusion and Future Work

Fuzzy logic systems have the ability to deal with the imprecision and uncertainties surrounded the input variables and their relationships. In this paper, we presented a new fuzzy logic-based scheme for dynamic

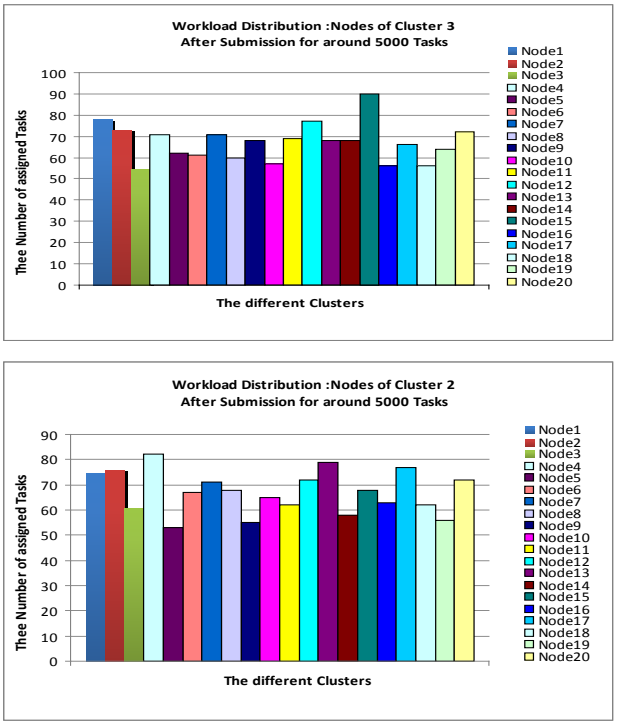

Figure 9. The workload states of the different nodes related to clusters 2 and 3 respectively after distributing the tasks

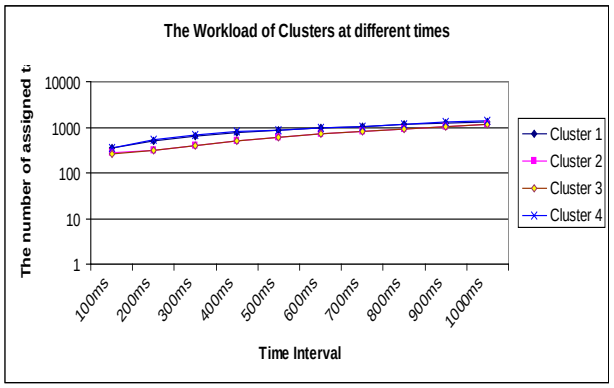

Figure 10. The clusters states at different times

load balancing in the grid computing services. Four input variables have been considered in the proposed scheme to evaluate the workload state of the presented node. We evaluated the performance of the proposed scheme in terms of its ability to keep all nodes and clusters of the overall system in a balanced way. The simulation results show that the proposed scheme achieves really satisfactory and consistently load balancing than in other randomized approaches. As a future work, it is possible to use clustering for the nodes in the same cluster based on their states since enormous number of nodes may impair the proposed scheme performance. Another goal is to embed this scheme into the proposed framework of the NSTIP project # 10-INF1381-04 in order to validate the scalability of the proposed framework and to assess its ability to support the load balance in a grid that hosts of food, health and nutrition inquire services.

8. Acknowledgment

We would like to thank King Fahd University of Petroleum and Minerals for providing the computing facilities. The authors would like to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project No.10-INF1381-04 as part of the National Science, Technology and Innovation Plan.

REFERENCES

[1] C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” International Journal of High Performance Computing Applications, vol. 15, issue3, pp 200–222, 2001.

[2] S. Sharma, S. Singh, and M. Sharma, "Performance Analysis of Load Balancing Algorithms," World Academy of Science, Engineering and Technology, vol. 38, 2008.

[3] L. A. Zadeh, “Fuzzy Logic, Neural Networks, and Soft Computing, Communications of the ACM, vol-37, Issue-3, pp: 77-84, Mar. 1994.

[4] L. A. Zadeh, “From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions”, Int. J. Appl. Math. Computer Sci., Vol.12, No. 3, pp. 307-324, 2002.

[5] H. T. Nguyen, N. R. Prasad, C. L. Walker, and E. A. Walker, “A First Course in Fuzzy and Neural Control”, CRC Press, 2002.

[6] B. Kosko, Neural Networks and F u q Systems: A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, New Jersey, 1992.

[7] L. S. Cheung. Load Balancing in Distributed Object Computing Systems, M. Phil. Thesis, Department of Electrical and Electronic Engineering, the University of Hong Kong, August 2001

[8] S. R. Moosavi-Nejad, S.S. Mortazavi, and B. V. Vahdat, “Fuzzy based Design and tuning of distributed systems load balancing controller,” The 5th Symposium on Advances in Science and Technology (SASTech), 2011.

[9] K. B Bey, F. Benhammadi, Z. Gessoum and A. Mokhtari, "CPU Load Prediction Using Neuro-Fuzzy and Bayesian Inferences", Journal of Neurocomputing Vol.74, pp. 1606-1616, May. 2011.

[10] A. Revar, M. Andhariya, D. Sutariya, "Load Balancing in Grid Environment using Machine Learning - Innovative Approach, " International Journal of Computer Applications (0975 – 8887), Volume 8– No.10, October 2010

[1] K. V. Yu, and Chih-Hsun Chou, “A Fuzzy-Based Dynamic Load-Balancing Algorithm,” http://jitas.im.cpu.edu.tw/2004-2/4.pdf

[2] M. Rantonen, Tapio Frantti, and Kauko Leivisk, "Fuzzy expert system for load balancing in symmetric multiprocessor systems ", Expert Systems with Applications Journal, Vol. 37, Issue 12, pp. 8711-8720, December 2010

[3] L. Cheung and Y. Kwok, “On load balancing approaches for distributed object computing systems,” The Journal of Supercomputing, vol. 27, pp. 149-175, 2004.

[4] E. Saravanakumar and Gomathy Prathima,” A novel load balancing algorithm for computational grid, “International Journal of Computational Intelligence Techniques, ISSN: 0976–0466 & E-ISSN: 0976–0474 Volume 1, Issue 1, 2010, PP-20-26

[5] K. Lu, R. Subrata, and A.Y. Zomaya, “An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grid Sites,” Proc. 25th IEEE Int’l Performance Computing and Comm. Conf. (IPCCC ’06), 2006.

[6] Y. Li, Yuhang Yang and Rongbo Zhu, “A Hybrid Load balancing Strategy of Sequential Tasks for Computational Grids,” 2009 IEEE International Conference on Networking and Digital Society, pp.112-117.

[7] B. Yagoubi and Y. Slimani, “Task Load Balancing Strategy for Grid Computing,” Journal of Computer Science 3 (3): 186-194, 2007.

[8] L. M. Khanli1 and Behnaz Didevar, “A New Hybrid Load Balancing Algorithm in Grid Computing Systems, ”Journal of Computer Science Vol-2 No 5 October, 2011.

[9] D. Ramesh and A. Krishnan, “Hybrid Algorithm for Optimal Load Sharing in Grid Computing, Journal of Computer Science 8 (1): 175-180, 2012.

[10] T. Helmy, F. Al-Otaibi, “Dynamic Load-Balancing Based on a Coordinator and Backup Automatic Election in Distributed Systems“, International Journal of Computing & Information Science, pp. 37-45, Vol. 9, No. 1, April 2011.

