A New Fuzzy-based Job Scheduling Algorithm for Cluster Computing

Behzad Azizpour
Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran

Mehdi Effatparvar
ECE Department, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Mohammad Sadeq Garshasbi
Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran

ABSTRACT
Scheduling is the process of improving the performance of a parallel and distributed system. Cluster systems are part of distributed systems. Cluster systems refers to the concept of run parallel jobs that can be run simultaneously on several processors. In this paper, introduce a method based on fuzzy logic for scheduling Parallel jobs on cluster systems. The main objective is to achieve performance and power improvement. The results of the simulations indicate our introduced method is better than comparison with the algorithm FCFS and SJF.

General Terms
Cluster Systems, Scheduling, Fuzzy Logic.

Keywords
Job Scheduling, Cluster Computing, Idle Time, Utilization.

1. INTRODUCTION
Parallel job scheduling is an important topic in the field of research. The main issue is how to share the processors available in the existing parallel computing environment among tasks submitted by users or processes. There are different parallel computing environments first, the parallel computers in which the systems come under are desktops and laptops with multi-core chips, second are the grids in which large scale heterogeneous distributed shared computing environment, third are the web servers, in which large scale latency sensitive online services are provided and finally the virtualization, in which the resource management is performed inside and among multiple virtual machines [1-5]. It is easily understood that the parallel computing environments are existing and rather continues to future technologies[3,4].

In multiple processing, multiple processors work together to implement a program. The major used of these systems for problems solving in models and engineering science. (Such as applied physics, nuclear, Geology and Seismology, mechanical engineering, electrical engineering, mathematics, etc.). Today not only scientific problems solve that require parallel processing also some commercial applications require fast computers. Many of these applications require the processing of large volumes of Informations are complex. Some of these programs can be huge databases and data mining operations[5-8]. Oil exploration, medical imaging and diagnosis and etc.

Clusters offer computational power used to solve problems with large computational requirements. Proper scheduling is fundamental to performance in distributed systems. The most important aspect of a distributed system is the scheduling algorithm. The scheduling algorithm is responsible for the allocation of processors to jobs and determines the order in which jobs will be executed on processors[6,9,10].

In general, task scheduling is presented in two forms: static and dynamic[11]. In static scheduling algorithms, all information needed for scheduling, such as the structure of the parallel application, the execution times of individual tasks and the communication costs between tasks, must be known in advance. There are several techniques to estimate such information. Static task scheduling takes place during compilation time before running the parallel application. In dynamic scheduling, however, tasks are allocated to processors upon their arrival, and scheduling decisions must be made at run time[8-12].

In this paper, introduce a method based on fuzzy logic for scheduling Parallel jobs on cluster systems. The main objective is to achieve performance and power improvement. Our method minimize the response time and idle time, also it maximum the utilization. Results of the simulations indicate minimum response time and idle time in comparison with FCFS and SJF the algorithms. In addition, maximum utilization in comparison with FCFS and SJF the algorithms.

This study is divided into the following sections: In section 2 an overview of Fuzzy Logic. Section 3 presents System setup in detailed. The proposed method is described in Section 3.2. Results of the study are analyzed in Section 4. Finally, Section 5 presents the conclusions.

2. INTRODUCTION TO FUZZY LOGIC
Classical logic only permits propositions having a value of truth or falsity. The notion of whether 1+1=2 is absolute, immutable, mathematical truth. However, there exist certain propositions with variable answers, such as asking various people to identify a color. The notion of truth doesn't fall by the wayside, but rather a means of representing and reasoning over partial knowledge is afforded, by aggregating all possible outcomes into a dimensional spectrum[11,12].

Both degrees of truth and probabilities range between 0 and 1 and hence may seem similar at first. For example, let a 100 ml glass contain 30 ml of water. Then we may consider two concepts: Empty and Full. The meaning of each of them can be represented by a certain fuzzy set. Then one might define the glass as being 0.7 empty and 0.3 full. Note that the concept of emptiness would be subjective and thus would depend on the observer or designer. Another designer might equally well design a set membership function where the glass would be considered full for all values down to 50 ml. It is essential to realize that fuzzy logic uses truth degrees as a

mathematical model of the vagueness phenomenon while probability is a mathematical model of ignorance[12].

In this paper, introduce a method based on fuzzy logic for scheduling Parallel jobs on cluster systems. The main objective is to achieve performance and power improvement.

3. SYSTEM SETUP

The system model assumed for this work explained in section 3.1. The policy the system must follow for scheduling the parallel job named as NewFuzzy Based Scheduling and described in section 3.2.

3.1 System Model

The number of jobs submitted to scheduler is *m*. The scheduler node N_0 contains the details of all the jobs which are arranged in a queue. Each parallel job composed of a set of dependent tasks. There are totally N_{n+1} number of nodes. The node N_0 is scheduler or head node which is responsible for assigning the work or job to the worker nodes from N_1 through N_n. The scheduling algorithm implemented in head. The system for the computer cluster assumed to satisfy the following conditions:

1. There are N_{n+1} number of nodes and are fully connected.
2. Communication and computation can be executed simultaneously. This can be accomplished by following the non-blocking send and non-blocking receive protocol for communication purpose.
3. Before scheduling a particular job, the system assumed to know the runtime of that job.
4. Nodes are homogeneous.
5. Tasks or jobs are independent.

It was assumed that the communication between the nodes will take place with the help of message passing interface environment.

Also, any node is consist of following section:

1. Routing Table
2. Load Index
3. Cost Table
4. Fuzzy Controller
5. Scheduller

As in Fig. 1 illustrated, our system model is involved Routing table, Load index, Cost table and a fuzzy controller, which manages scheduling of system. The routing table presents the communication links among nodes in the system. Load index indicates the load of its related node. In order to determine the node status as a sender, receiver or neutral by using fuzzy controller and based on fuzzy rules, we need a cost table that provides the nodes communication costs and the number of heavy loaded nodes. The cost table is obtained by using load index and routing table while the number of heavy loaded nodes can be extracted from the cost table.

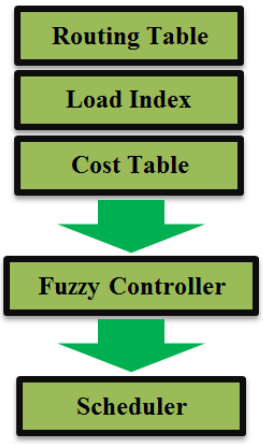

Fig 1: System Model based Fuzzy in any node

3.2 NewFuzzy Based Scheduling

Load index value based on a given threshold is classified into three categories and is defined between 0 to s and threshold is s. three Fuzzy sets (Fig.2) are used to describe the load index value: lightly loaded, moderate loaded, heavy loaded. Variables for load index take grade values of Fuzzy variables are uncertainties and depends on network situation it can be changed.

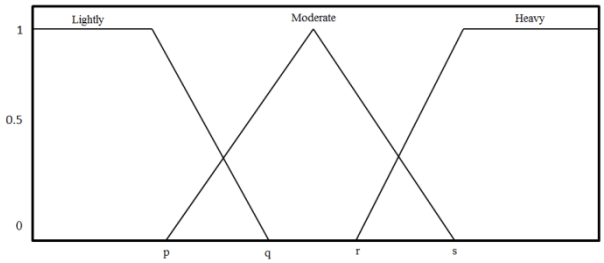

Fig 2: Fuzzy index load chart

Assuming fuzzy rules base are as follows:

$$\mu_{\text{lightly load}}(\text{load}) = \begin{cases} 1 & \text{load} < p \\ \dfrac{q - \text{load}}{q - p} & p \le \text{load} \le q \\ 0 & \text{load} > q \end{cases}$$

$$\mu_{\text{moderate load}}(\text{load}) = \begin{cases} 0 & \text{load} < p \quad \text{load} > s \\ \dfrac{\text{load} - p}{q - p} & p \le \text{load} \le q \\ 1 & q < \text{load} < r \\ \dfrac{s - \text{load}}{s - r} & r \le \text{load} \le s \end{cases}$$

$$\mu_{\text{heavyload}}(\text{load}) = \begin{cases} 1 & s < \text{load} \\ \dfrac{s - \text{load}}{s - u} & r \le \text{load} \le s \\ 0 & \text{load} < r \end{cases}$$

Assuming sender initiated scheduling algorithm, the proposed knowledge base is as follows:

Rule [1]:
If (load is lightly load)
then (status node is reciver)

Rule [2]:
If (load is moderate_load) and
(no heavy load nodes is more)
then (status node is reciver)

Rule [3]:
If (load is heavy load)
then (status node is not reciver)

Scheduler base on nodes status send jobs to nodes, so if node status is lightly load or moderate load then it can recive jobs for execution else it can not recive any job for execution. therefor there is two status for nodes:

- Reciver
- No reciver

If a node is reciver then scheduler can give to it jobs, else if a node is no reciver then scheduler can not give to it jobs.

4. EVALUATION OF SIMULATION RESULTS

In this section, we present and discuss the experimental results of the proposed scheme. All simulations were performed using MATLAB software. We evaluated the performance of our proposed scheme in comparison with SJF (Short Job First) and FIFO (First in First out) algorithms in a Cluster system.

The simulation results showed two instances: when the numbers of tasks are more, and when the numbers of tasks are less.

When the number of jobs is 20, Fig 3 shows processors idle time average by applying FIFO, SJF and NewFuzzy Based algorithm for job scheduling on Cluster system.

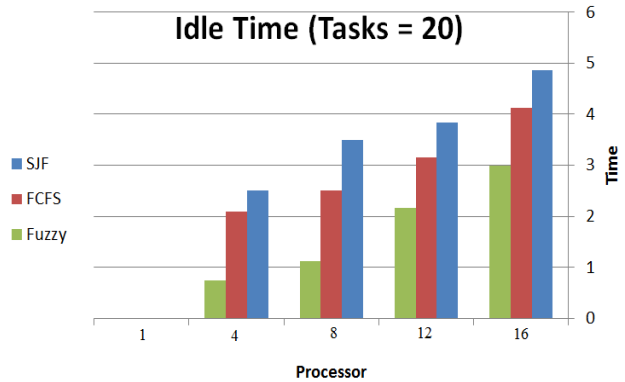

Fig 3: idle time when the number of tasks is 20

When the number of jobs is 2000, Fig 4 shows processors idle time average by applying FIFO, SJF and NewFuzzy Based algorithm for job scheduling on Cluster system.

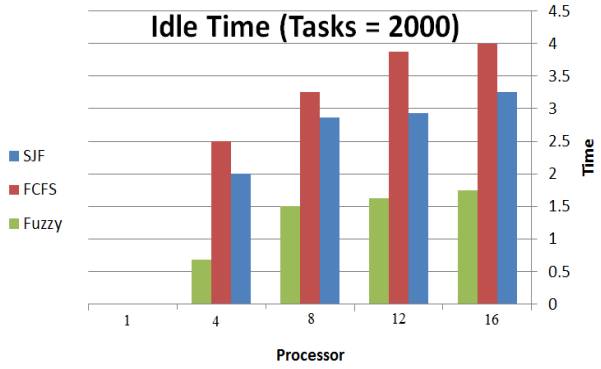

Fig 4: idle time when the number of tasks is 2000

The vertical axis represents time, and the horizontal axis represents the number of processors.

Fig 5 shows the system utilization when the number of tasks is 20. Also, Fig 6 shows the system utilization when the number of tasks is 2000.

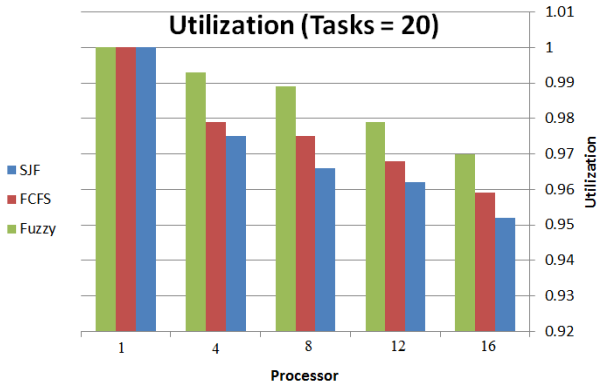

Fig 5: system utilization when the number of tasks is 20.

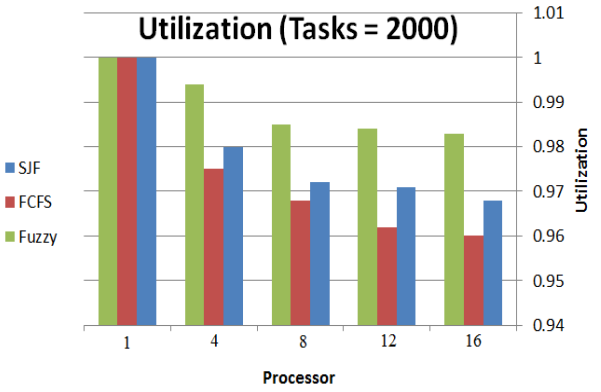

Fig 6: system utilization when the number of tasks is 2000.

Obtained results in large and small scales indicate that our proposed method can provide similar results in different scales, and proves the robustness of the proposed method in different scales.

5. CONCLUSION

In this study, we proposed the Fuzzy based algorithm for jobs scheduling in Cluster. The proposed method found a better solution for assigning jobs to the Cluster system. This method reduces the idle time average and provides maximum utilization compared to same algorithms. The method proposed in this paper was compared with FIFO and SJF algorithms. The results of simulations indicate that our method is better compared with the FIFO and SJF algorithms.

6. REFERENCES

[1] A.Neela madheswari, R.S.D.Wahida banu, Static and Dynamic Job Scheduling with Communication aware policy in Cluster computing, Elsevier, Computers and Electrical Engineering 39 (2013) 690–696.

[2] Ananth Grama, Georage Karypis, Anshul Gupta, Vipin Kumar, "Introduction to parallel computing", Published by Pearson Education, 2009.

[3] Tran, Van Hoai, "Task Scheduling for Parallel Systems ", Faculty of Computer Science and Engineering HCMC University of Technology, 2009-2010.

[4] Jasbir, Gurvinder, "Improved Task Scheduling on Parallel System using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 39– No.17, February 2012.

[5] Frachtenberg E, Schwiegelshohn U. New challenges of parallel job scheduling. In: Proceedings of the 13th IEEE international conference on Job scheduling strategies for parallel processing, 2007.

[6] Franke H, Pattnaik P, Rudolph L. Gang scheduling for highly efficient distributed multiprocessor systems. In: Sixth Symposium on the Frontiers of massively parallel computation, 1996.

[7] Zhou B.B. Job Relocation: A Method to Enhance the Performance of Gang Scheduling on Clusters of PCs. In: Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003, p. 582-586.

[8] Ryu K.D, Pachapurkar M, Fong L.L. Adaptive memory paging for efficient gang scheduling of parallel applications. In: Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium, 2004.

[9] Barbosa da silva F.A, Scherson I.D. Improving Throughput and Utilization in Parallel Machines through Concurrent Gang. In: Proceedings of the 14th IEEE International Symposium on Parallel and Distributed Processing, 2000.

[10] Jette M.A. Performance characteristics of Gang scheduling in multiprogrammed environments. In: Proceedings of the ACM/IEEE Conference on Super Computing, 1997.

[11] Song F, Static and dynamic scheduling for effective use of multicore systems. Thesis Report, University of Tennesse, Knoxville, Dec 2009.

[12] Gonzalez-Dominguez J, Taboada G. L, Fraguela B.B, Martin M.J, Tourino J. Automatic mapping of parallel applications on multicore architectures using the Servet benchmark suite, In: Computers and Electrical Engineering, 2012, 38, p. 258-269.