IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Deadline aware fuzzy scheduler for parallel computing environment

Vandana[1*], A. Chhabra[2]

[1] Dept. of CET, Guru Nanak Dev University, Amritsar, India
[2] Dept. of CET, Guru Nanak Dev University, Amritsar, India

**Corresponding Author: vandananayar897@gmail.com, Tel.: 896810060*

Available online at: www.ijcseonline.org

Accepted: 03/Jun/2018, Published: 30/Ju/2018

Abstract: Parallel computing has been gaining much attention in recent years due to the fact that many real-world applications have become more complex and dynamic. Due to advances in parallel computing, the user base of cloud computing is increasing hugely and therefore there is need of scheduling algorithms in order to schedule all those resources provided by parallel servers. It has been found that the scheduling techniques developed so far have some limitations; i.e. no one is perfect in every case. The majority of the existing techniques have neglected one of these issues: the most of the existing researchers have neglected the use of the deadline constraint of jobs while designing the fuzzy membership functions for efficiently scheduling the jobs in parallel environment. The majority of existing researchers who have designed fuzzy based job scheduling techniques have neglected the use of job co-allocation while scheduling the jobs on available servers. In this paper a new rule based and deadline aware fuzzy scheduler have been proposed which outperforms the existing schedulers.

Keywords: Deadline; scheduling; Co-allocation; Fuzzy logic; Fuzzy rules.

1INTRODUCTION

Various researchers are showing a keen interest in scheduling strategies lately, reason being the increase in resources in organisations. To take advantage of these resources, we use co-allocation technique that allows jobs to be distributed across different servers Scheduling and resource allocation in grid environments are considered NP-hard problem as complexity of scheduling increases. Fuzzy logic have proven to be effective for finding optimal or near optimal solutions. They help in improving parameters such as makespan, speedup, flowtime, convergence etc.

A. Grid Environment

A grid is a scalable computing infrastructure spread over a wide area and comprising of resource provider, resources and users. Resources include power, network components, printers, CPUs etc. It can be imagined as a distributed system with workloads that are non-interactive and involves a huge number of files. The grid computers are dispersed geographically. One grid can be used for a particular application or for various applications. The jobs are submitted to the grid by users. These jobs are scheduled by the Resource Management System (RMS) and processed jobs or results are sent back to the user. Grid is perceived as a single system. The grid size can vary from small to large. The small grids are limited to network of computer within one corporation and large grid can spread across many components and networks. The resources in a grid are dynamic and belong to different domains. A resource can be active or inactive in a grid. Assigning jobs manually is impossible. Hence scheduling in grids is one of the most challenging issues in grid computing.

B. Scheduling

The method by which specified work is assigned to the resource by some means, to complete a particular task is called scheduling. Schedulers keep the resources busy by allowing multiple users to share the systems resources in an effective manner. It is due to scheduling that computers are able to multi-task within a single CPU. Distributed computing has grabbed much attention in the recent years as it is reliable, highly scalable and has low cost [2]. The complexity of scheduling increases due to heterogeneity of the processing and communication resources, which leads to NP-Hard problems [1]. For such problems, there aren't any algorithms that may produce optimal solutions. This brings us to fuzzy logic based algorithms that provide near optimal solutions within reasonable time.

C. Coallocation

From the last few years, distributed memory multiprocessors and clusters have gained a lot of attention. Multi-cluster systems are made up of multiple geographically distributed clusters and hence provide large computation power as compared to single cluster. Due to

large groups being able to share the multi-cluster, job turnaround time is reduced and system utilisation becomes high. This leads to larger job sizes to be possible as jobs are allowed to use processors in multiple clusters simultaneously i.e. to employ co-allocation [37].
Restriction to size of job component and to number of job components improves the performance of co-allocation. The co-allocation mechanism should posses two main features: a) it should be highly distributed and b) it should be robust to any changes. Jobs can be co-allocated due to fragmentation that occurs naturally within each cluster. This can be explained by taking an example. Suppose a job is waiting in a cluster's ready queue. It might happen that this job requires more nodes than are available presently on its cluster, but in collection there are enough nodes in the multi-cluster to accommodate the job. Then the job is called co-allocated if it was mapped onto nodes that were borrowed from other clusters.
Also the execution time of the co-allocated job itself as well as that of the other co-allocated job can be affected by the amount of communication produced by each co-allocated job.

D. Fuzzy logic
Fuzzy logic starts with and builds on a set of user-defined human language rule.The fuzzy system changes these rules to their mathematical equivalents that simplifies the job of system and results in much more accurate representation of the way system behaves in the real world. A fuzzy membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. Sometimes the input space refers to universe of discourse.

Rest of the paper is organized as follows Section II contain the related work of job scheduling algorithms for parallel environment., Section III addresses proposed methodology. Section IV contains results and discussion for existing and proposed methodology, and Section V concludes research work with future direction.

Section-2

RELATED WORK

Jun Wang et al. (2018) [1] in their work proposed a new power aware job scheduler by applying fuzzy rule based control method which improve power efficiency . **NirmalaH** et al. (2016) [2] proposed a scheduling algorithm EDZL which schedules the task sets based on the priority generated by fuzzy inference system. Mohammad Shojafar et al. [2015][3] suggested a hybrid approach called FUGE that is based on fuzzy theory and genetic algorithm that aims to perform optimal load balancing taking into account the execution time and cost. Zhijia Chen et al. [2015] [4] presented a dynamic resource scheduling method based on fuzzy control theory. Results show that this approach improves the resource scheduling efficiency and quality of service in cloud computing. John Yen et al. [2018][5] developed a fuzzy scheduler to facilitate searching for a feasible schedule. It has proved to be a good technique for finding schedule. Deepa N K et al. [2014][8] presented a fuzzy logic based self adaptive job replication scheduling(FSARS) algorithm. It uses security demand of job as parameter.It gives best results against resource failure. Fuzzy membership functions were used to transform security conditions to fuzzy sets. Dr.Girijamma H A et al. [2014][9] presented a scheduling algorithm that schedule real time jobs using fuzzy logic. M.M.M. Fahmy [2010][11] proposed an algorithm based on fuzzy logic for non periodic jobs on soft real time single processor system. Shatha J. Kadhim [2010] [12] proposed a rule based scheduling algorithm in design and implementation to overcome the limitations of well known algorithm.

Section-3

METHODOLOGY

1. In the starting phase, we initialize parallel configuration and load the parallel workload.
2.In this phase deadline time is evaluated by using the equation "Deadline time= arrival time+ k* burst time" where 'k' is a random number, using a suitable tool like MATLAB.
3. In this phase, fuzzification of job's burst time, deadline, server's speed and resources is done by using fuzzy if-then rules.
4.In this phase, the output of the previous phase is updated in a table called fuzzy value table.
5. In this phase, jobs are assigned using backfilling according to fuzzy rules.
6.Finally jobs are scheduled onto the servers and values for the make span and flow time are returned.

Table 1 Final fuzzy table

Job Length	Speed of server	resources	Deadline value	Fuzzy values
0.2	0.2	0.2	0.1	0.5
0.2	0.2	0.3	0.1	0.4
0.2	0.3	0.2	0.1	0.3
0.3	0.2	0.2	0.1	0.5
0.3	0.3	0.2	0.1	0.5
0.3	0.2	0.3	0.1	0.4
0.3	0.5	0.5	0.1	0.3
0.3	0.5	0.2	0.4	0.2
0.3	0.3	0.3	0.4	0.5
0.5	0.5	0.5	0.4	0.2
0.2	0.5	0.5	0.4	0.2
0.2	0.3	0.5	0.4	0.3
0.2	0.5	0.3	0.4	0.5
0.5	0.5	0.2	0.4	0.3
0.5	0.5	0.3	0.4	0.2
0.5	0.2	0.2	0.4	0.2
0.5	0.3	0.3	0.4	0.5
0.2	0.5	0.5	0.6	0.3
0.2	0.3	0.3	0.6	0.3
0.3	0.3	0.5	0.6	0.4
0.5	0.3	0.5	0.6	0.5
0.5	0.2	0.5	0.6	0.3
0.2	0.5	0.2	0.6	0.3
0.3	0.2	0.5	0.6	0.4
0.3	0.5	0.3	0.6	0.3
0.5	0.3	0.2	0.6	0.3
0.5	0.2	0.3	0.6	0.3

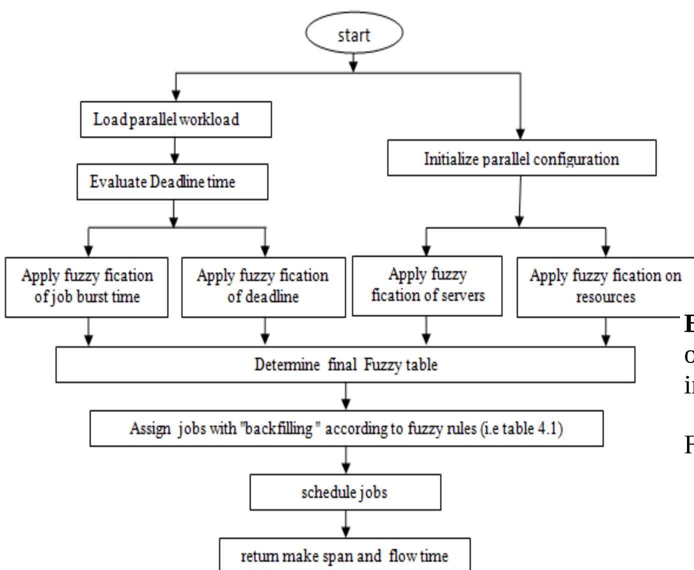

Fig 1. Flowchart of the proposed technique

Section-4

RESULTS

For experimentation and implementation the proposed technique is evaluated using MATLAB tool . The evaluation of proposed technique is done on the basis of following parameters i.e. Make-span and flow time.

1)Scalability of jobs

A. Make span-refers to the total length of the schedule. It can be explained as:

Make span= Max(T_k) (where T_k is the finishing time task k).

Table2. Make Span analysis(in sec)

Job size	MS exiting	MS proposed
100	154454	85731
200	245452	190992
400	474126	301766
500	507495	308148
600	505863	376391

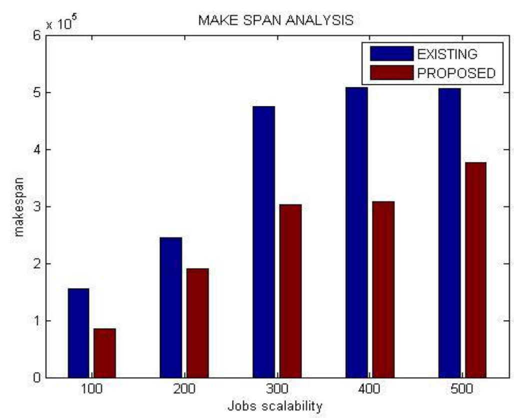

Fig 2. Make span analysis

B. Flow time - It is defined as the sum of finalization time of all tasks. The definition of flow time can be translated into a simple formula:

Flow time=$\sum T_k$ (where T_k is finishing time of task k)

Table3. Flow time analysis (in sec)

No of jobs	FT existing	FT proposed
100	97833	67785
200	202194	126541
400	211377	150219
500	340560	198112
600	383440	242260

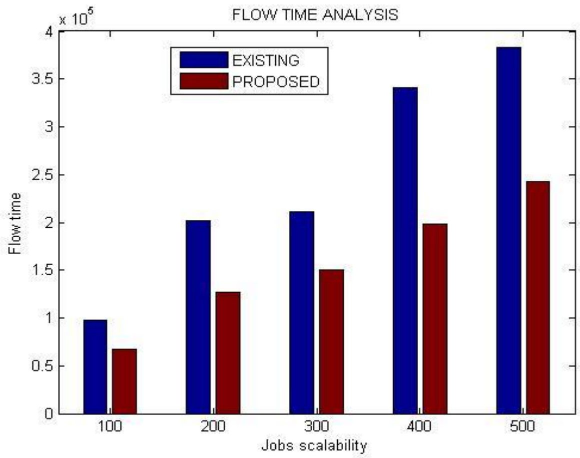

Fig 3 Flow time analysis

The above observation are taken by varying the size of the job sets like 100, 200, 400, 500, 600 and keeping the number of processors fixed.

Fig2 and Fig3 shows the comparison between the existing and the proposed algorithm, and it is clear that the value of proposed algorithm is less than the existing algorithm therefore proposed algorithm outperforms the existing algorithm in terms of make span and flow time respectively.

2.Scalability of servers

A. Make span

Table 4. Make span analysis (in sec)

Resource configuration (128)	MS existing	MS proposed
64, 64	842827	407616
32, 32, 32, 32	335578	234656
16, 16, 32, 64	396344	227426

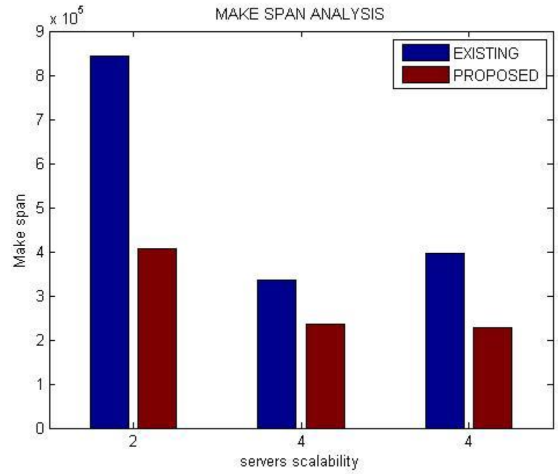

Fig 4 make span analysis

2)Flow time analysis

Table 5 Flow time analysis (in sec)

Resource configuration(128)	FT of existing work	FT of proposed
64, 64	802827	326894
32, 32, 32, 32	345578	293524
16, 16, 32, 64	386344	593138

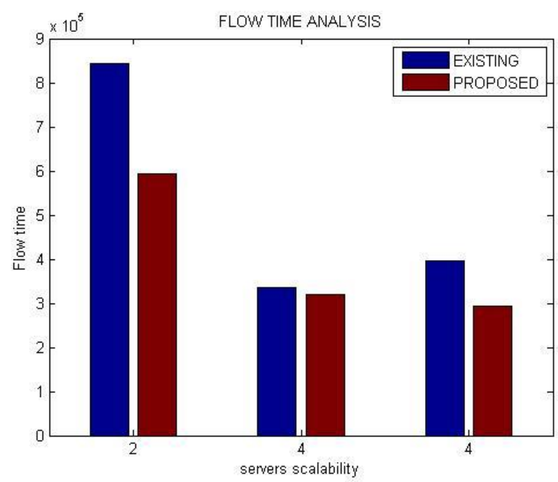

Fig 5Flow time analysis

Fig 4 and Fig 5 reveals that the proposed technique provides significant results when the provided resources to the servers are changed. It is observed that for similar resourcces if we reduce the number of servers then the flow time increased at rapid rate. However, the proposed technique outperforms existing technique in every configuration in terms of make span and flow time respectively.

Section-5

CONCLUSION

This paper has proposed a deadline aware fuzzy scheduler. A novel fuzzy rule base is designed to allocate the user jobs between available servers. To efficiently utilize the resources the concept of backfilling is also used. It has been designed and implemented in Matlab tool 2013.The comparison of existing and proposed schedulers has been done in terms of make span and flow time. Results show that proposed techniques outperforms the existing technique.

REFERENCES

[1] Jun Wang ,Dezhi Han1, Ruijun Wang ”A new rule-based power-aware job scheduler for Supercomputers” Springer Science& Business Media, LLC, part of Springer Nature 2018.

[2] NirmalaH , Dr.Girijamma H A “Priority based Scheduling algorithm using Fuzzy Technique” 2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions.

[3] Zhijia Chen, Yuanchang Zhu, Yanqiang Di, and Shaochong Feng “A Dynamic Resource Scheduling Method Based on Fuzzy Control Theory in Cloud Environment” Journal of Control Science and Engineering Volume 2015.

[4] John Yen, Jonathan Lee, Nathan Pfluger, and Swami Natarajan “Designing a Fuzzy Scheduler for Hard Real-Time Systems” https://ntrs.nasa.gov/search.jsp?R=19930020348 2018-05-17.

[5] Oğuzhan Ahmet Arık & M. Duran Toksarı “Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects” International Journal of Production Research, 2017 https://doi.org/10.1080/00207543.2017.1388932.

[8] Giovanni Mazzuto, Maurizio Bevilacqua and Filippo E. Ciarapica “A heuristic scheduling algorithm based on fuzzy logic and critical chain project management” Int. J. Project Organisation and Management, Vol. 9, No. 4, 2017.

[9] M.M.M. Fahmy “A fuzzy algorithm for scheduling non-periodic jobs on soft real-time single processor system” Elsevier 2010.

[11] Shatha J. Kadhim, Kasim al-aubidy “Design and Evaluation of a Fuzzy-Based CPU Scheduling Algorithm” January 2010.

[12] Rajani Kumari, Vivek Kumar Sharma, Sandeep Kumar “Design and Implementation of Modified Fuzzy based CPU Scheduling Algorithm” International Journal of Computer Applications. 2013

[13] Matthew T. Ogedengbe, Moses A. Agana “New Fuzzy Techniques for Real-Time Task Scheduling on Multiprocessor Systems” International journal of computer trend and technology 2017.

[14] Nirmala H, Dr.Girijamma H A “Fuzzy Scheduling Algorithm for Real –Time multiprocessor system” International Journal of Scientific & Engineering Research, Volume 5, Issue 7, July-2014.

[15] Zhijia Chen, Yuanchang Zhu, Yanqiang Di, and Shaochong Feng “A Dynamic Resource Scheduling Method Based on Fuzzy Control Theory in Cloud Environment” Journal of Control Science and Engineering Volume 2015.

[16] Er. Inderpal Singh, Er. Aman Arora “Fuzzy Based Improved Multi Queue Job Scheduling For Cloud Computing” International Journal of Advanced Research in Computer Science, May-June, 2015.

[17] S. Saravana Kumar “Improved CPU Utilization using Advanced Fuzzy Based CPU Scheduling algorithm (AFCS)” International Journal of Scientific Research in Science, Engineering and Technology 2016.

[18] Jian li, meili tang, wenhai shen “Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing” MDPI 2017.

[19] Ming; Yi, John; Zeng, Edward ”fuzzy based scheduling algorithms”

[20] Dr. G. Geetharamani, C. Daniel Sundar Rajan, J. Arun Pandian “Fuzzy Based Task Scheduling For Heterogeneous Distributed Computing” International Journal of Innovative Research in Science Engineering and Technology Vol. 6, Issue 12, December 2017.

[21] Anindita Kundu “A New Approach for Task Scheduling of Cloud Computing Using Fuzzy” International Journal of Innovative Research in Computer Science & Technology ,Volume-3, Issue-2, March 2015.

[22] Shatha J. Kadhim and Kasim M. Al-Aubidy “Design and Evaluation of a Fuzzy-Based CPU Scheduling Algorithm”

[23] Vikash Goswami,Dr.R.K.Shrivastava “HMM and Fuzzy Logic Based Algorithm for Efficient Task Scheduling and Resource Management in Cloud Systems” International Journal of Mathematics Trends and Technology (IJMTT) February 2018.

[24] Prerna Ajmani and Manoj Sethi “Proposed Fuzzy CPU Scheduling Algorithm (PFCS) for Real Time Operating Systems” International Journal of Information Technology 2013

[25] Carsten Franke, Joachim Lepping, and Uwe Schwiegelshohn “On Advantages of Scheduling using Genetic Fuzzy Systems”

Authors profile

Miss vandana pursued Bachelor of computer science from Guru Nanak Dev University, Amritsar in 2016. She is currently pursuing Masters of computer science in department of CET, Guru Nanak Dev University, Amritsar. Her main research work focuses on cloud and parallel computing.

Mr A. Chhabra is currently pursuing PhD and currently working as Assistant professor in department of CET , Guru Nanak Dev University of Amritsar. He has published more than 35 research papers in reputed international journals. His main research work focuses on cloud computing. He has 14 years of teaching experience and 6 years of Research experience.

