

REVIEW OF VARIOUS STRATEGIES FOLLOWED IN ORDER TO ALLOCATE JOB IN PARALLEL ENVIRONMENT

Malkeet Kaur¹, Karan Mahajan²

¹*Malkeet Kaur* Research Scholar of M. Tech in CSE, Global Institute of Management & Emerging Technologies, Amritsar, Punjab, India

²*Karan Mahajan* Assistant Professor of CSE department, Global Institute of Management & Emerging Technologies, Amritsar, Punjab, India

Abstract: The jobs can be simple or complex in nature. The processing of jobs through single processor system consume more time. In order to resolve the problem parallel systems can be resolved. The parallel system involve group of processors that could be same or distinct in nature. The same type of processors within the parallel system is known as array environment and distinct processor within the system is known as federated system. The purpose of this paper is to present the comparison of such system along with performance evaluation of each technique utilized in it. Efficient hybrid approach of ant and honey bee algorithm can be extracted out of the proposed paper.

KEYWORDS: Job, Single Processor, Multiprocessor, Array, Federated, Distinct Processor

1. INTRODUCTION

The job presented to the system can be simple or complex in nature. The simple job which is not modular not required to be subdivided and distributed. Cluster environment is the one of the environment in which multiple servers can be operated together. These servers can either be loosely connected or tightly connected. If the servers are tightly connected then changes in one server bring about a change in other servers also. If the servers are tightly connected then high overhead will be associated with them. Servers that are connected loosely can reduce the overhead since changes to one system may not require changes in other servers. Single quorum device cluster is the most widely used cluster type and are also known as the Standard Quorum Cluster .In this type of clusters ,multiple nodes with one or more cluster disk arrays, also called the cluster storage, and a connection device, that is, a bus. Each disk in the array is owned and managed by only one server at a time.

The disk array also contains the quorum resource. The job which is complex needed to be divided into modules. These jobs which are complex may have to be processed by the use of multiprocessor environment. Such environment is known as multiprocessor environment. (Singh 2014)[1]The complexity of job hence specifies the requirement of amount of processor required to yield the output within specified interval of time. The description of parallel system is better understood by the use of Flynn's classification.

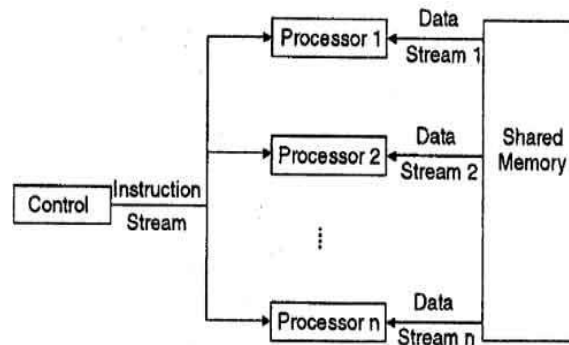


Figure 1: Showing Parallel System

Job allocation in parallel system is complex. Legion of techniques are devised by researcher. Genetic algorithm is one such technique which can be used to optimally allocate job to server within the cluster. Combination of distinct servers is termed as Cluster. Clusters when combine together Grid is formed. Genetic algorithm is search technique used to seek actual or approximate solution to given problem. The optimality of Gas depends greatly on size of population. The smaller problems if utilizes genetic algorithm than performance degrades. Genetic algorithm is considerable useful for larger problems. In case Genetic algorithm is applied higher cost of evaluation is encountered. So a modification to existing GAs has to be made to obtain better results. The problems which can be solved through the use of Genetic algorithm are generally distributed. The algorithms handling complex problems require GAs design and simulation strategies. (Munawar and Wahib 2008)[3]

2. LITERATURE SURVEY

The critical part of parallel system is mechanism of distributing jobs within the parallel systems. Legions of algorithms utilized for this purpose. The algorithms which take part in the parallel system are described in this section.

(Systems et al. 2013)[4]In this job scheduling algorithm job is distributed to the processor within the system on the first come first serve basis. This algorithm may or may not yield optimal solutions. The allocated processor release the job when the job burst time finishes. This algorithm is strictly non primitive in nature. It is hence rarely utilized in the parallel environment.

(Delisle and Delisle, n.d.)[5]In this scheduling time is shared among multiple jobs. The time sharing system utilizes time quantum. The processor is switched among the processors based on time quantum. The process continues until all the jobs finish execution. The time sharing system involves states such as waiting, active and ready. The time scheduling on parallel system can be implemented using local scheduling. The processing node has processors associated with them. Threads ready to be executed are placed within first come first serve buffer. When the processor is available thread is fetched from the queue and executed. The time sharing environment generally adopt pipeline concept for executing instructions concurrently within uniprocessor systems.

(Yuce et al. 2013)[6]This algorithm utilizes better features of Ant Colonies algorithm along with high distance capabilities. The Honey bee algorithm utilizes optimal path finding along with distance coverage. It utilizes the foraging system associated with

honey bees to find the path out of available alternatives. This algorithm converges much faster as compared to existing ant colony algorithm.

(Silva et al. n.d.)[7]This scheduling produces optimal result as compared to all other parallel scheduling algorithms. This algorithm is primitive in nature. Hence deadlock never occurs within the system. The tasks that form a job are grouped together and then scheduled in this approach. The job priority is also considered in this case. The job with highest priority is executed at first place in this case in the early system this approach produces optimal results. But nowadays more advanced algorithms exists which produces better result as compared to this algorithm.

3. COMPARISON OF PARALLEL COMPUTING ALGORITHMS

PAPER	ALGORITHMS USED	PARAMETERS	PROS	CONS
(Gabaldon, Lerida, Guirado & Planes, 2015) [2]	Genetic algorithm	Make span Flow time QOS	Improved quality of service	Genetic algorithm shows indifferent convergence rate
(Wu, Ishikawa, Zhu, Xia, & Member, 2016) [17]	IGA	Energy Cost	Better energy and cost conservation policy is suggested	Convergence rate is slow hence may not be suitable for all environment
(Wen, Wang, Wu, & Xie 2015) [16]	ACO	Downtime Migration time	Optimal solution is obtained with high convergence rate	Large distances cannot be handled
(Zhu, Zhang, Li, & Liu, 2015) [8]	Work Flow Scheduling	Cost Downtime Migration time	Overhead is reduced considerably	No Hybrid approach associated with meta heuristic is suggested hence not applicable to multiple environments
(Yuce et al. 2013) [6]	Honey Bee	Flow time Make span	Convergence rate is faster Large distance can be handled	Optimal solution is not produced all the times

4. OBJECTIVE OF STUDY

The objective of this literature is to describe pros and cons of various techniques which are used so that evaluation mechanism can be devised from which algorithms having best performance can be extracted. The algorithms studied provide optimal performance along with

1. Energy conservation by load balancing.
2. Enhance rate of convergence.
3. Enhance performance independent of distance.
4. Determine optimal solutions by extracting all set of possibilities in allocation
5. Minimum power consumption while execution of job in parallel system.

6. METHODOLOGY

Methodology assigned with ant and honey bee algorithm is analysed. The job allocation procedure can be hybridised in order to obtain optimal schedule. The methodology of hybridized approach is listed as under

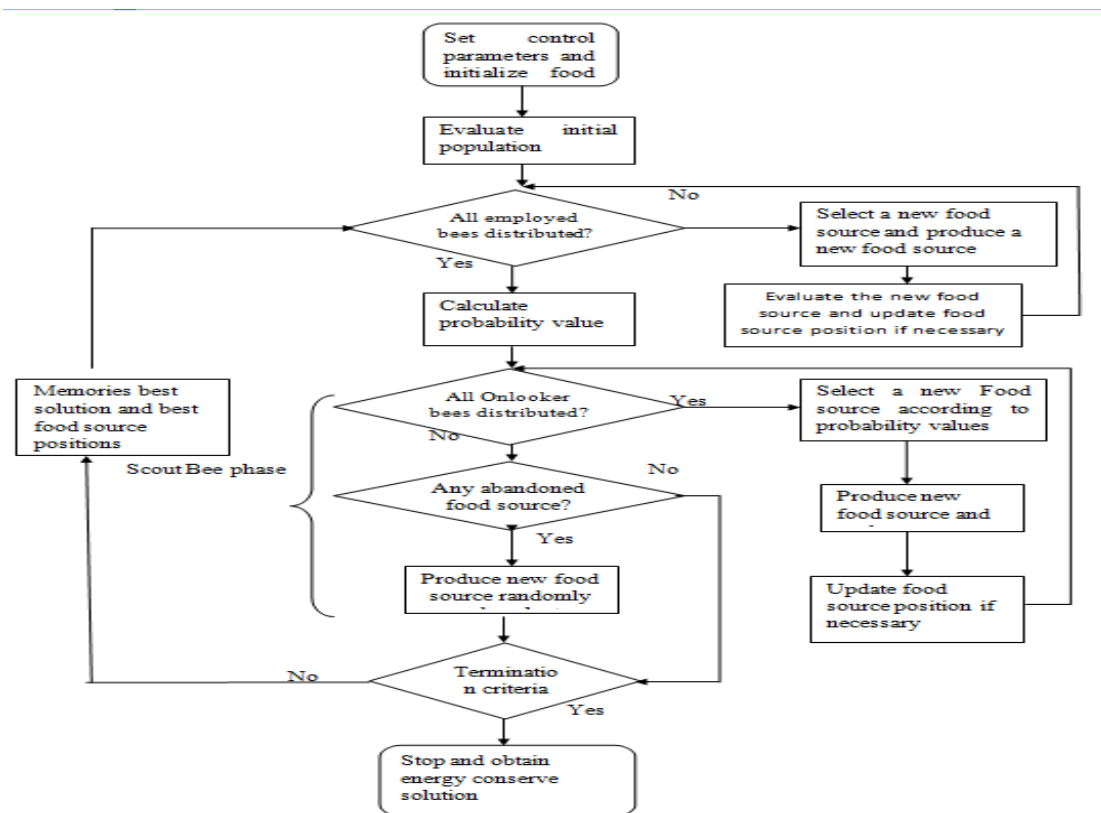


Figure 2: Methodology to be considered for enhancement of job scheduling

5. CONCLUSION

The proposed system conducts the review of various algorithm utilized in the parallel system. The objective of work is to discover algorithm which can be utilized in almost every situation. The comparison table is also presented showing performance of each algorithm. The comparison shows that single algorithm is not enough to be utilized in simulative environment hence algorithms have to be merged to obtain best possible solution. Hence in the future for enhanced performance multiple algorithms has to be merged in order to form hybrid algorithm.

6. REFERENCES

1. Singh, Rachhpal. 2014. "Task Scheduling in Parallel Systems Using Genetic Algorithm" 108 (16): 34–40.
2. Gabaldon, E, J L Lerida, F Guirado, and J Planes. 2015. "Multi-Criteria Genetic Algorithm Applied to Scheduling in Multi-Cluster Environments." *Journal of Simulation* 9 (4). Nature Publishing Group: 287–95. doi:10.1057/jos.2014.41.
3. Munawar, Asim, and Mohamed Wahib. 2008. "A Survey : Genetic Algorithms and the Fast Evolving World of Parallel Computing," 897–902. doi:10.1109/HPCC.2008.77.
4. Systems, I J Intelligent, Mohammad Sadeq Garshasbi, Mehdi Effatparvar, and Ardabil Branch. 2013. "High Performance Scheduling in Parallel Heterogeneous Multiprocessor Systems Using Evolutionary Algorithms," no. October: 89–95. doi:10.5815/ijisa.2013.11.10.
5. Delisle, Pierre, and Pierre Delisle. n.d. "Parallel Ant Colony Optimization : Algorithmic Parallel Ant Optimization : Algorithmic Models Models and Colony Hardware Implementations and Hardware Implementations."
6. Yuce, Baris, Michael S Packianather, Ernesto Mastrocinque, Duc Truong Pham, and Alfredo Lambiase. 2013. "Honey Bees Inspired Optimization Method: The Bees Algorithm." *Insects* 4 (4): 646–62. doi:10.3390/insects4040646.
7. Silva, Fabrício A B, Ernesto P Lopes, Eliana P L Aude, Flavio Mendes, T C Júlio, Henrique Serdeira, Mario Martins, and Walfredo Cirne. n.d. "Response Time Analysis of Gang Scheduling for Real Time Systems."
8. Zuo, L., Shu, L. E. I., & Dong, S. (2015). A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing, 3.
9. Carretero, Javier, Fatos Xhafa, and Ajith Abraham. 2007. "Genetic Algorithm Based Schedulers for Grid Computing Systems." *International Journal of Innovative Computing, Information and Control* 3 (6): 1–19.
10. Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2). IEEE: 182–97. doi:10.1109/4235.996017.
11. Wang, Meihong, and Wenhua Zeng. 2010. "A Comparison of Four Popular Heuristics for Task Scheduling Problem in Computational Grid." *2010 6th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2010*, 3–6. doi:10.1109/WICOM.2010.5600872.
12. Dror G. Feitelson, Larry Rudolph, Dror G. Feitelson, and Larry Rudolph. 1995. "Parallel Job Scheduling: Issues and Approaches." *Jsspp* 949: 1–18. doi:10.1.1.47.545.
13. Xhafa, Fatos, and Ajith Abraham. 2008. "Meta-Heuristics for Grid Scheduling Problems." ... for Scheduling in Distributed Computing ..., 1–37. doi:10.1007/978-3-540-69277-5_1.

14. Hamscher, Volker, Uwe Schwiegelshohn, Achim Streit, Ramin Yahyapour, Rajkumar Buyya, and Mark Baker. 2000. "Evaluation of Job-Scheduling Strategies for Grid Computing." *Grid Computing* 1971: 191–202. doi:10.1007/3-540-44444-0.
15. Karatza, Helen D. 2000. "SCHEDULING GANGS IN A DISTRIBUTED SYSTEM" 7 (1): 15–22.
16. Wen, W., Wang, C., Wu, D., & Xie, Y. (2015). An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment. <http://doi.org/10.1109/FCST.2015.41>
17. Wu, Q., Ishikawa, F., Zhu, Q., Xia, Y., & Member, S. (2016). Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters, 1374(c). <http://doi.org/10.1109/TSC.2016.2616868>
18. Karatza, Helen D. 2000. "SCHEDULING GANGS IN A DISTRIBUTED SYSTEM" 7 (1): 15–22.
19. Kousalya, A, and R Radhakrishnan. 2015. "A Comparative Study of Parallel Job Scheduling Algorithms in Cloud Computing" 6 (3): 2687–90.
20. Maruthanayagam, D, and R UmaRani. 2010. "Enhanced Ant Colony Algorithm for Grid Scheduling." *International Journal of Computer Technology and Applications* 1 (1): 43–53.
21. Mishra, Anurag, Shakti Mishra, and Dharmender Singh Kushwaha. 2011. "An Improved Backfilling Algorithm : SJF-BF" 05 (01).
22. Nedunchezian, R, and P.Vivekanandan. 2010. "A Fast Genetic Algorithm for Mining Classification Rules in." *Soft Computing* 1 (1): 10–20.
23. Sharma, Shyna, Amit Chhabra, and Sandeep Sharma. 2015. "Comparative Analysis of Scheduling Algorithms for Grid Computing," 349–54.