Volume: 08 Issue: 01 Pages: 3009-3013 (2016) ISSN: 0975-0290

Stochastic Scheduling Algorithm for Distributed Cloud Networks using Heuristic Approach

Mamatha E.
Department of Mathematics, GITAM University, Bengalore, india
Email: sricsrmax@gmail.com
Saritha S.
Department of Mathematics, GITAM University, Bengalore, india
CS Reddy
School of Computing, SASTRA University, Thanjour, India
Email: csreddy@cse.sastra.edu

----------**ABSTRACT**----------
Rule based heuristic scheduling algorithms in real time and cloud computing Systems employ for resource or task scheduling since they are suitable to implement for NP-complete problems. However, they are simple but there is much room to improve these algorithms. This study presents a heuristic scheduling algorithm, called High performance hyper-heuristic scheduling algorithm (HHSA) using detection operator, to find better scheduling solutions for real and cloud computing systems. The two operators - diversity detection and improvement detection operators - are employed in this algorithm to determine the timing to determine the heuristic algorithm.. These two are employed to dynamically determine a low level heuristic that can be used to find better solution. To evaluate the performance of this method, authors examined the above method with several scheduling algorithms and results prove that Hyper Heuristic Scheduling Algorithm can significantly decrease the makespan of task scheduling when compared with all other scheduling algorithms. A novel high-performance hyper-heuristic algorithm is proposed for scheduling on cloud computing systems to reduce the makespan. This algorithm can be applied to both sequence dependent and sequence independent scheduling problems.

Keywords - **Heuristic Algorithm, Scheduling tasks, cloud computing, diversity detection**

Date of Submission: Aug 05, 2016 Date of Acceptance: Aug 15, 2016
----

1. INTRODUCTION

Since Rule based algorithms are simple and easy to implement there is lot of scope to improve the performance of these algorithms especially in cloud environments. Traditional methods and algorithms give results in small scale environments[1, 3, 8, 15,18], however due to the advent of computer and other internet technologies there is a lot of scope to improve these algorithms to improve the performance in large scale[2, 9]. Drastic use of millions of servers in the recent past has reduced the usage of traditional scheduling techniques. The resource management at this scale is the concerned issue. Scheduling is responsible for arbitrate of resources and is at the centre of resource management [4]. The issue of efficiency at this rate and developing model of consumption of cloud providers needs new approaches and techniques to be applied to the age old problem of scheduling.

Cloud computing has allowed the consumption of services over internet with subscription based model. Based on the level of abstraction, we have different models of cloud computing like Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). This model of service consumption is extremely suitable[5] for many workloads and cloud computing has become highly successful technology[6, 12]. It allows its users to pay for what they use and also remove the upfront infrastructure cost [7]. Cloud service providers receive resource requests from a number of users through the use of virtualization [11]. It is very essential for cloud providers to operate very efficiently in multiplexing at the scale to remain profitable. The increase in the use of cloud computing has risen to develop massive data centers with very large number of servers [10]. The resource management at this scale is the concerned issue.

Scheduling is responsible for arbitrate of resources and is at the center of resource management. The issue of efficiency at this rate and developing model of consumption of cloud providers needs new approaches and techniques to be applied to the age old problem of scheduling. Virtual machine is the primary unit of scheduling in this model [13]. In this study, we deal with problem of virtual machine scheduling over physical machines. We aim to understand and solve the various aspect of scheduling in cloud environments. Specifically, we leverage various fine grained monitoring information in making better scheduling decisions [14]. We used

Volume: 08 Issue: 01 Pages: 3009-3013 (2016) ISSN: 0975-0290

learning based approach of scheduling in widely different environments. There is increasing concern over energy consumption by cloud data centers and cloud operators are focusing on energy savings through effective utilization of resources.

SLAs is additionally essential for them for the execution of uses running. We propose calculations which attempt to minimize the vitality utilization in the server farm appropriately keeping up the SLA guarantees[16,17]. The calculations attempt to utilize less number of physical machines in the server farm by alterably rebalancing the physical machines in light of their asset use. The calculations likewise do a streamlining of virtual machines on a physical machine, diminishing SLA infringement.

Scheduling is very important for cloud provider to achieve efficient resource pooling and elasticity. Scheduling problem becomes very relevant in cloud computing scenarios where the cloud service providers have to operate at very much efficient to be competitive and take advantage at scale. The wide acceptance of cloud computing means data centers with many more machines and the usage model is much different than traditional clusters, like hour boundaries, auction based prices are to name a few. Thus scheduling in cloud data center is more challenging than traditional cluster schedulers [19]. Also, these data center run many different kinds of applications with varying expectations from infrastructure. Resource usage patterns in traditional data centers are have less variance in than the unpredictability faced by cloud data centers. The way in which the operating system scheduler try to minimize the utilization of resources on a single machine, similarly cloud schedulers also try to minimize the utilization of data center as a whole. It is clear that in such conditions [20], the role played by the scheduler is very important in achieving maximum utilization without reducing application performance.

Until now we are still actively looking for possible solutions to enhance the performance of information systems for computation, analysis, and storage. Since distributed and parallel computing was widely used to enhance the performance of a variety of computer systems, several models and ideas have been proposed for different approaches and congenital restrictions in different eras. Whatever may be the consideration it is for, the way to efficiently utilize computer resources is a pivotal research issue. Among them, scheduling is indispensable in the success of increasing the performance of the system [21]. With the advent of computer and other internet technologies, paradigms of cloud computing have been successfully used on several information systems in recent years. Although cloud computing systems nowadays provide a better way to carry out the submitted tasks, most job and task scheduling problems on cloud computing systems are still either NP-hard or NP-complete. Unfortunately, these rule-based scheduling algorithms are inappropriate for large-scale or complex scheduling problems because the results of these scheduling strategies are usually far from optimal.

To develop more efficient scheduling algorithms for cloud, some recent reviews were, which encompass non-meta heuristic and Meta heuristic scheduling algorithms, such as earliest-finish-time-based algorithm and genetic algorithm. Also given in was a comparison sheet to classify the scheduling algorithms discussed there by methods, parameters, and factors of scheduling. In addition, the method presented uses feedback information to estimate the earliest finish time to dynamically adjust the resource allocation.

2. EXISTING WORK

The main objective of the project is to develop more efficient scheduling algorithm for cloud. Since distributed and parallel computing was widely used to enhance the performance of a variety of computer systems, several models and ideas have been proposed for different approaches and congenital restrictions in different eras. Whatever may be the consideration it is for, the way to efficiently utilize computer resources is a pivotal research issue. Among them, scheduling is indispensable in the success of increasing the performance of the system. The wide acceptance of cloud computing means data centers with many more machines and the usage model is much different than traditional clusters, like hour boundaries, auction based prices are to name a few. Thus scheduling in cloud data center is more challenging than traditional cluster schedulers.

The small scheduling problems that is problems for which all the solutions can be checked in a reasonable time by using classical exhaustive algorithms running on modern computer systems. In comparison, with the large scale scheduling problems, like the problems for which not all the solutions can be examined in a reasonable time by using the same algorithms running on the same computer systems.

These observations make it easy to understand that exhaustive algorithms will take a prohibitive amount of time to check all the candidate solutions for large scheduling problems because the number of candidate solutions is simply way too large to be checked in a reasonable time. As a result, researchers have paid their attention to the development of scheduling algorithms that are efficient and effective, such as heuristics. Work process is utilized with the robotization of systems where by documents and information are gone between members as per a characterized arrangement of tenets to accomplish a general objective.

A work process administration framework is the particular case that oversees and executes work processes on figuring assets. Work process Planning: It is a sort of worldwide undertaking booking as it spotlights on mapping and dealing with the execution of between

Volume: 08 Issue: 01 Pages: 3009-3013 (2016) ISSN: 0975-0290

ward assignments on imparted assets that are not specifically under its control. The creators arrange and survey hyper-heuristic methodologies into the accompanying four classifications: in view of the arbitrary decision of low level heuristics, insatiable and puckish, meta heuristic-based, and those utilizing learning instruments to oversee low level heuristics.

The hyper heuristics can be used to operate at a higher level of abstraction. Meta heuristic techniques are expensive techniques that require knowledge in problem domain and heuristic technique. Hyper heuristic technique does not require problem specific knowledge. In order to solve hard computational search problems the hyper heuristic techniques can be used. The hyper heuristic techniques can be operated on the search space of heuristics.

DISADVANTAGES

- The FIFO scheduler will list the first job and apportion all the assets it needs.
- This indicates that the second job must wait until the first job is finished.
- The result is apparently a very long response time for the second job.
- Algorithms is in that the results attained by these algorithms may be far from ideal or even suitable.
- Two important disadvantages that take the researchers into contemplation of modifying the search approach of these traditional scheduling algorithms.

3. ANALYSIS AND DEVELOPMENT OF ALGORITHM

The essential thought of the proposed calculation is to utilize the differences location and change recognition administrators to adjust the escalation and expansion in the pursuit of the arrangements amid the meeting methodology. The proposed calculation, called hyper-heuristic booking calculation (HHSA).The parameters max and ni, where max signifies the greatest number of cycles the chose low-level heuristic calculation is to be run; ni the quantity of emphases the arrangements of the chose low-level heuristic calculation are not moved forward.
Line 2 reads in the tasks and jobs to be scheduled, i.e., the problem question. Line 3 initializes the population of solutions Z = fz1; z2;:::; zNg, where N is the population size. Online 4, a heuristic algorithm Hi is randomly selected from the candidate pool H = fH1;H2; :::;Hng.

By estimating and foretelling the processing speed and accomplishment time of each mechanism and each job, HHSA can create feasible solutions that optimize the make span not taken into consideration by the other schedulers. In addition, our results show that HHSA can finally provide better solutions in terms of make span. Hyper-heuristic algorithms can preserve a high search diversity to surge the chance of finding better solutions at later iterations while not increasing the estimation time. The improvement detection operator, the diversity disclosure operator is used by HHSA to decide "when" to variation of the low-level heuristic algorithm. This machinery implies that the higher the temperature, the higher the occasion to escape from the local search space to find better results. The outcomes of ACO are similar to those of the suggested algorithm for the first three datasets. This is different from the hybrid heuristic algorithm, which runs more than one low level algorithm at each reiteration, thus needing a much longer addition time.

Hyper-heuristics are abnormal state issue free heuristics that work with any arrangement of issue ward heuristics and adaptively apply and join them to tackle a particular issue. This could be because of the way that variations of differential advancement, which we chiefly use as essential heuristics because of their aggressive execution and basic arrangement, unequivocally rely on upon the populace circulation. Hyper-heuristics may be viewed as an extraordinary manifestation of hereditary programming, the key instinct fundamental research around there is that, for a given sort of issue, there are regularly various clear heuristics as of now in presence that can function admirably (however maybe not ideally) for specific sorts of examples of that kind of issue. Maybe it is conceivable to consolidate those current heuristics into some more expand calculation that will function admirably over a scope of issues.

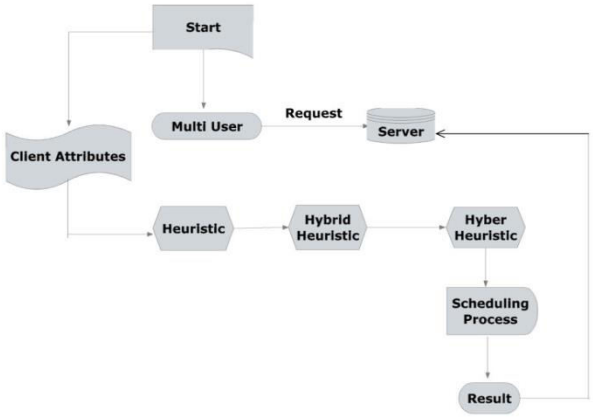
Features of the Algorithm

- Simple and easy to implement
- Some rule-based deterministic algorithms can catch acceptable solutions quickly.
- Most of them are well-matched to each other.
- All the clarifications can be patterned in a rational time by using classical comprehensive algorithms.

4. MODULE DESCRIPTION

A simple random method is used to select the low-level heuristic Hi from the candidate pool H. The diversity detection operator is used by HHSA to decide "when" to change the low-level heuristic algorithm Hi. This mechanism implies that the higher the temperature, the higher the opportunity to escape from the local search space to find better solutions. The timer is fixed at startup and end up mode of an application.

Volume: 08 Issue: 01 Pages: 3009-3013 (2016) ISSN: 0975-0290

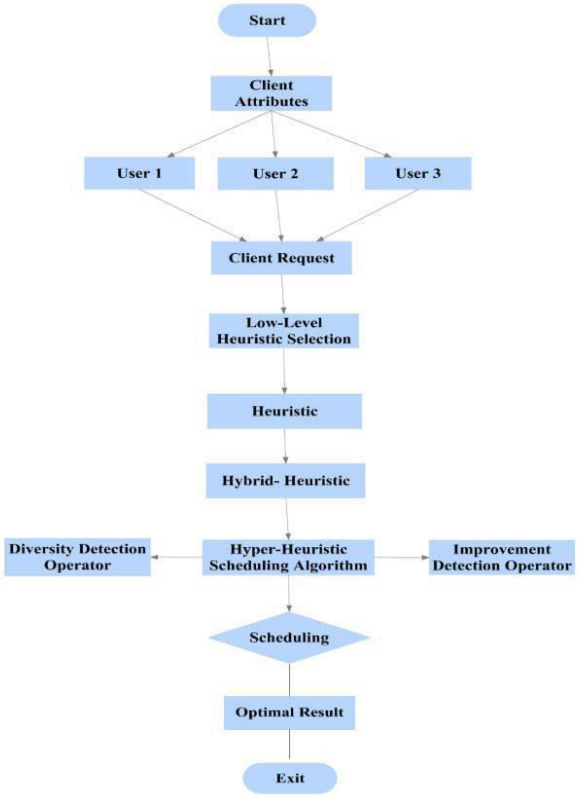


System Architecture Diagram

Time-based consists in setting a timer that schedules the time instant when the Scheduled task has to be performed. The timer is fixed at startup mode of an application. Hyper-heuristic algorithms can then maintain a high search diversity to increase the chance of finding better solutions at later iterations while not increasing th e computation time. A time- based rejuvenation policy intends to identify the optimal time to rejuvenate with respect to one or more performance indices. The VMM does not degrade, and therefore, it is only necessary to keep memory of the age that was reached at the workload changing point.

Hyper-heuristic is a search method to learn automatically and recurrently by the machine to make it learn, in which the process follows combination, generation and/or adaption a number of modest heuristics to resourcefully find solution for computational search problems. Prime motto to study the hyper-heuristics is to develop a system that can deal any problem rather than solving a particular single problem. It may have multiple heuristics from this one can choose to solve the problem, so that all heuristic scheduling have good enough to solve problem and to rectify its weakness.

The idea of implementation is simple and consistently develops algorithms by conjoining the weakness and strength by compensating are known to be heuristics. When a problem instance is given, the high-level method selects which low-level heuristic should be applied at any of the given time, based upon the current problem state, or the search stage.



Flow Diagram of hyper-heuristic scheduling

5. CONCLUSION

The proposed calculation utilizes two identification administrators to naturally focus when to change the low level heuristic calculation and an annoyance administrator to tweak the arrangements acquired by every low-level calculation to further enhance the booking results regarding make compass. As the reenactment results demonstrate, the proposed calculation cannot just give preferred results over the customary tenet based booking calculations, it likewise beats the other heuristic planning calculations, in illuminating the work process booking and Hadoop guide errand booking issues on distributed computing situations.

Moreover, the recreation results show further that the proposed calculation unites speedier than the other heuristic calculations assessed in this study for the vast majority of the datasets. To sum things up, the essential thought of the proposed "hyper heuristic" calculation is to influence the qualities of all the low level calculations while not expanding the processing time, by running unrivaled one low-level calculation at every cycle. This is in a general sense not the same as the alleged cross breed heuristic calculation, which runs more than one low level calculation at every cycle, consequently obliging an any longer processing time.

With the joining of inherited programming into hyper-heuristic examination, another level of philosophies are found that we have termed 'heuristics to make heuristics'. These philosophies give wealthier heuristic interest

Volume: 08 Issue: 01 Pages: 3009-3013 (2016) ISSN: 0975-0290

spaces, and in this way the chance to make new approaches for dealing with the fundamental combinatorial issues. On the other hand, they are all the more difficult to execute, when appeared differently in relation to the more excellent heuristic to pick heuristics", in light of the fact that they require the deterioration of the available existing heuristics, and the design of a fitting framework. There is significantly more to be done in this class of techniques.

REFERENCES

[1] M. Pinedo, Scheduling: Theory, Algorithms and Systems. Prentice Hall, Inc., 1995.

[2] P. Chr´etienne, E. G. Coffman, J. K. Lenstra, and Z. Liu, Eds., Scheduling Theory and Its Applications. John Wiely & Sons Ltd, 1995.

[3] Chandra Sekhar Reddy, C., and K. Ramakrishna Prasad. "The Study State Analysis of Tandem Queue with Blocking and Feedback." *International Journal of Advanced Networking & Applications* 1.1 (2009).

[4] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," European Journal of Operational Research, vol. 187, no. 3, pp. 985–1032, 2008.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[6] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," Proceedings of the Grid Computing Environments Workshop, 2008, pp.1-10.

[7] Mamatha, E., C. S. Reddy, and Ramakrishna Prasad. "Mathematical Modeling of Markovian Queuing Network with Repairs, Breakdown and fixed Buffer." *i-Manager's Journal on Software Engineering* 6.3 (2012): 21.

[8] K. Bilal, M. Manzano, S. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 64–77, 2013.

[9] Mamatha, E., C. S. Reddy, and K. R. Prasad. "Antialiased Digital Pixel Plotting for Raster Scan Lines Using Area Evaluation." Emerging Research in Computing, Information, Communication and Applications. Springer Singapore, 2016. 461-468.

[10] Mamatha, E., C. S. Reddy, and S. Krishna Anand. "Focal point computation and homogeneous geometrical transformation for linear curves." *Perspectives in Science* (2016).

[11] M. R. Garey and D. S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness. New York: W.H. Freeman and Company, 1979.

[12] C. W. Tsai and J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," IEEE Systems Journal, vol. 8, no. 1, pp. 279–297, 2014.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, 1983.

[14] Mamatha et al. An Efficient Line Clipping Algorithm in 2D Space, in press, International Arab Journal of Information Technology,

[15] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[16] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol.1, no. 1, pp. 53–66, 1997.

[17] Mamatha, et al. "Performance evaluation of homogeneous parallel processor system of Markov modeled queue." i-Manager's Journal on Software Engineering 3.2 (2008): 58.

[18] J. Bła˙zewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. We¸glarz, Scheduling Computer and Manufacturing Processes. Springer-Verlag New York, Inc., 2001.

[19] Reddy, C. S., et al. "Obtaining Description for Simple Images using Surface Realization Techniques and Natural Language Processing." *Indian Journal of Science and Technology* 9.22 (2016).

[20] Reddy, P. S, et al. "Finite Element Analysis of Thermo-Diffusion and Diffusion-Thermo Effects on Convective Heat and Mass Transfer Flow Through a Porous Medium in Cylindrical Annulus in the Presence of Constant Heat Source." International journal of applied mathematics and mechanics 6.7 (2010): 43-62.

[21] Y. T. J. Leung, Handbook of Scheduling: Algorithms, Models and Performance Analysis. Chapman & Hall/CRC, 2004.