International Journal of Computer Systems (ISSN: 2394-1065), Volume 02– Issue 06, June, 2015
Available at http://www.ijcsonline.com/

Cloud-Scheduling Algorithm using Hyper heuristic Approach: Study and Implementation

Amirkhan Pinjari, B. R. Mandre

Department of Computer Engineering,
SSVPS's BS Deore College of Engineering, Dhule, 424005, India

Abstract

Cloud computing is a parallel and distributed system. Simple and easy to implementation of Rule-based scheduling algorithms widely used in many cloud computing systems. Results of these scheduling strategies are usually far from optimal. However Performance of this scheduling algorithm is improved by using heuristic algorithm. The paper present Heuristic scheduling algorithm is a Hyper-heuristic scheduling algorithm to find better scheduling solutions for cloud computing systems. Algorithm uses two operators. The algorithm implemented and compared with different algorithms in a simulated environment using a CloudSim. Result show that hyper heuristic algorithm reduces makespan as compared with other algorithm.

Keywords: *Cloud Computing, scheduling, Evolutionary Algorithm, CloudSim.*

I. INTRODUCTION

Cloud computing is define as parallel and distributed system consist of a collection of different inter-connected and also virtualized computers presented as one or more unified computing resources [2].NIST define the cloud as Cloud computing also explain as an model which enable convenient, omnipresent and on-demand network access to a shared pool of computing resources [3].The scheduling is important factor in the success of improving the performance of a computer system. With the advance internet technologies, cloud computing have been successfully used on several information systems in recent years. Cloud computing systems providing better way to carry out the submitted tasks in terms of execution, scalability, and flexibility, most job and task scheduling problems on cloud computing are NP complete. Mostly scheduling algorithms for cloud are rule based .They are widely used on today's cloud computing systems because of easy to implement. Unfortunately, these scheduling algorithms are inappropriate for processing large or complex scheduling problems because the results of these scheduling strategies are usually far from optimal. Therefore there is plenty of ways to improve scheduling algorithms in cloud computing [1].

Many of researchers are attracted toward heuristic algorithm from different research domain for improving performance scheduling on cloud computing system Hybrid heuristic is an extended version of heuristics which combines two or more heuristics into a single heuristic algorithm to leverage their strengths for scheduling but Instead to explore search space problem solutions, hyper-heuristics will automatically produced algorithm which solves a problem more efficiently. Global optimum is always found with heuristics. However it provides at least one solution whenever the algorithm stops. Workflow scheduling is a big issue in the area of computing. Basically issue related to the mapping of each task to an appropriate resource and allowing the task to satisfy some performance constraints. Workflows can execute in Grids but due to complexity of environment in Grids, the execution of workflows in clouds is more favorable. Cloud services such as storage will compute and bandwidth is available at much lower costs.

The basic idea of the algorithm is to increase the strengths of heuristic algorithms like simulated annealing genetic algorithm, particle swarm optimization by integrating them into a single algorithm. Two operators—diversity detection and improvement detection—are used by algorithm to determine the timing to employ the low-level heuristic (LLH) algorithm. The advantage of algorithm is, it is high performance reducing makespan and faster than the other heuristic algorithms. Computation cost is low[1].

The rest of the paper is organized as follow- Section 2 begins with a brief introduction to the traditional Scheduling algorithm, followed by the various scheduling algorithm on cloud computing. Section 3 describes in detail the hyper heuristic algorithm .Simulation results on CloudSim discussed in Section 4. Section 5 gives Conclusions and Future scope

II. SCHEDULING METHOD FOR CLOUD

Many of cloud system are using traditional scheduling algorithms to manage the computing resources like Hadoop uses FIFO as the default scheduling algorithm. Although the FIFO scheduler can achieve a high utilization but sometimes leads to a long response time when the number of jobs to be run is beyond the capacity of computing resources. Facebook developed Hadoop Fair Scheduler (HFS) and improved it with Delay Scheduling Yahoo developed Capacity Scheduler to deal with the

same scheduling problem but capacity Scheduler can also achieve fairness, but it requires a more sophisticated control over the cluster

In 2010,Suraj Panday, Linlin Wu, Siddeshwara Mayura Guru and Raj Kumar Buyya[4] represent algorithm that based on Particle Swarm Optimization (PSO) is a swarm-based intelligence algorithm influenced by the behavior of animals. Heuristic algorithm used by Authors to minimize the total cost of execution of application workflows on Cloud computing. Algorithm achieves least three times cost savings as compared to BRS mapping for workflow.

Zhangjum Wu, Zhiwei Ni, Lichuan Gu and Xiao Liu [5] represent: Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling. Authors proposed a revised discrete version of PSO (RDPSO) which is base on concept of set-based that is use in this paper. The key issue of DPSO is to define the position, velocity of particle and define their operation rules and equation of motion of discrete variables

In 2011, Sawant Shailesh [6] presents ―AGenetic Algorithm for Virtual Machine Resource in a Cloud. Usually, when scheduling is performed by traditional algorithm load imbalance and high number of VM migrations will occurred. Author used genetic algorithm to remove problem on both load imbalance and high migration costs with applying utilized historical data and current state of system.

Xiaofeng Wang, Chee Shin Yeo, Jinshu Su and Raj kumar Buyya[7],represent Reputation and Look-ahead Genetic Algorithm for optimizing makespan and reliability. Experiments show that the RD reputation improves the reliability of an application while Look ahead provides better solutions than existing list heuristics.

Eugen Feller, Louis Rilling and Christine Morin [8] present the new workload algorithm using Ant colony optimization for better conserving the energy. Nature-inspired approach used for solving the dynamic workload placement problem.

Rajarathinam Jeyarani, N. Nagaveni and Vasanth Ram present new scheduling algorithm that provide better and efficient Virtual Machine Provision ing by Self Adaptive Particle Swarm Optimization. Focus paper is on infrastructure as a Service (IaaS) model of cloud.

Jiandun Li, Junjie Peng, Zhou Lei and Wu Zhang[9] proposed the hybrid energy-efficient scheduling approach for private cloud . paper explore couples of characteristics related to workflow scheduling in private clouds Hybrid algorithm is based on pre-power technique and least-load-first algorithm.

Zhangjun Wu, Xiao Liu, Zhiwei Ni, Dong Yuan and Yun Yang [10] propose a package based random scheduling algorithm as the candidate service-level scheduling algorithm for market-oriented business model. Algorithm strategy includes for both service-level and task-level for scheduling stage.

In 2012, H. M Fard, R. Prodan, J. J. D Barrionuevo and T. Fahringer [11] introduce new framework with heuristic algorithm for multi-objective static scheduling in the scientific workflows of heterogeneous environment. Algorithm use constraints to specify by user with each objectives and approximates are optimal solution with the help of double strategy named as, maximized and minimized distance to constraint vector for dominant solutions.

Timur Keskinturk, Mehmet B. Yildirim and Mehmet Barut [12] introduce a mathematical model that minimizes average relative percentage of imbalance paper focus on the problem; imbalance with sequence-dependent setup times in a parallel-machine environment by minimizing average relative percentage. Further, authors present a mathematical model to derive the goal; minimize total relative imbalance by parallel-machine problem with sequence dependent setups.

In 2013, new meta-heuristic algorithm called Intelligent Water Drops algorithm was presented by S. H. Niu, S. K. Ong and A.Y. C Nee [13].Algorithm is customized for solving the problem of Multi-objective job shop scheduling. The aim of this scheduling is to allocate resources to a set of tasks while satisfying several constraints and objectives. The IWD algorithm is inspired by the movement of natural water drops that flowing n lakes and seas.

Salid Abrishami, Mahmoud Naghibzadeh and Dick H. J. E pema[14] proposed Deadline-Constrained Workflow Scheduling Algorithm for Infrastructure Service. In this paper, execution time is minimized by algorithms while maintaining the user defined deadline. Two different types of PCP are implemented. First is Cloud Partial Critical Path and second with Deadline Distribution.

Jing Huang, Kai Wu, Lok Kei Leong, Seungbeom Ma, and Melody Moh[15] proposed A Tunable Workflow Scheduling Algorithm Based on Particle Swarm Optimization for Cloud Computing. in this paper, a tunable fitness function is proposed by authors which give different weights to minimization of cost and makespan minimization.

In order to reduce the impact of performance variation of public Cloud resources in the deadlines of workflows, Rodrigo N. Calheiros and Rajkumar Buyya[16] proposed a new algorithm, called EIPR, which consider the behavior of Cloud resources during the scheduling process and also applies replication of tasks in order to increase the chance of meeting application deadlines .

Ranjit Singh and Sarbjeet Singh [17] introduce the score based deadline constrained algorithm. The concept score represent the capabilities of hardware resources which is used in algorithm. This value is used in allocation to the resources of various tasks in the application of workflow. Algorithm allocates resource to workflow application which is reliable and reduces the execution cost and completes the workflow application within user specified deadline.

III. HYPER HEURISTIC ALGORITHM

A Hyper-heuristic is a methodology for selecting or generating heuristics to solve hard computational search problems [18]. The basic idea of algorithm is to use two operator diversity detection and improvement detection to balance the intensity and diversification in the search of the solutions during process. The algorithm is below

Input: Cloudlet
Output: Makespan.

1. Start
2. Set up the parameter
3. Input the scheduling problem
4. Initialize the population at solution $Z= \{z_1, z_2, \dots \dots z_N\}$
5. Randomly select the heuristic algorithm H_i from Candidate pool H
6. While the termination criterion is not met
7. Update the population of solution Z by using selected the algorithm H_i
8. F_1=improvement_ Detection (z)
9. F_2=Diversity detection
10. If $\psi(H_i, F_1, F_2)$
11. Randomly select a new H_i
12. Z=Perturb (Z).
13. End While
14. End

Algorithm reads the tasks and jobs to be scheduled then initializes the population of solutions $H = \{z_1, z_2, \dots \dots z_N\}$ where N is the population size. Heuristic algorithm H_i is randomly selected from the candidate pool improvement detection operator denoted F1 and by the diversity detection operator denoted F2 to decide whether to select a new LLH or not.

A. *The Improvement Detection Operator*

The low-level heuristic H_i from the candidate pool is selected by random method. This operator only checks to see whether the solutions found by , i.e., BSFMK [Best for Makespan] are being improved or not. If the selected cannot improve the BSFMK after a row of iterations, the improvement detection operator will return a false value

$$F_1 = \begin{cases} \text{false}, & \text{BSFMK is not improved after } t_{ni} \text{ iteration} \\ \text{true}, & \text{Otherwise} \end{cases} \quad (1)$$

B. *The Diversity Detection Operators*

When to change the low-level heuristic algorithmH_i is decided by this operator It works as the diversity of the initial solution $D(z_0)$ will be used as a threshold. If the diversity of the current solution $D(z)$ is less than the threshold ω, this operator will return false, and algorithm will then randomly select a new low level heuristic, as given below:

$$F_2 = \begin{cases} \text{true}, & D(z) > \omega, \\ \text{false}, & \text{Otherwise} \end{cases} \quad (2)$$

C. *The Perturbation Operator*

Perturbation operators of Hi itself, the presented algorithm will perturb the solutions obtained by Hi before they are passed on to the newly selected LLH. Candidate solutions created by the low level heuristic Hi can be perturbed by the simulated annealing mechanism by assigning a different temperature to each individual to balance the intensification and diversification of the search.

The perturbation temperature T_k is defined

$$(T_k) = \begin{cases} T_{max} \cdot \dfrac{\emptyset_{max} - \emptyset}{\emptyset_{max}} \cdot \dfrac{f(Z_{best}) - f(z_k)}{f(Z_{best}) - f(Z_{worst})} & \text{, if } H_i \in P \\ T_{max} \cdot \dfrac{\emptyset_{max} - \emptyset}{\emptyset_{max}} \cdot \dfrac{1}{N} & \text{Otherwise} \end{cases} \quad (3)$$

Where T_k ($1 < k < N$) denotes the temperature assigned to the k_{th} individual, T_{max} the maximum temperature, $\emptyset_{max}$ the maximum number of iterations LLH will be performed, and $\emptyset$ the number of iterations LLH is actually performed. Also, $f(\cdot)$ is the fitness function z_k z_{best} and z_{worst} are the finesses of the k_{th} solution, the best fitness of the population, and the worst fitness of the population.

The algorithm was implemented in CloudSim which provide simulation environment of cloud computing using java and compare with other scheduling algorithm. Comparison of algorithm is taken on basis of makespan, is completion time of task

IV. SIMULATION ON CLOUDSIM

CloudSim is a new toolkit that generalized, and extensible simulation framework allows simulation and experimentation Cloud computing and application services [19]. CloudSim is a simulation tool for creating cloud computing environment. CloudSim is used as the simulator in solving the workflow scheduling problem. CloudSim can be used to construct a data center with a set of virtual machines as the resource. Each task of a workflow can be started by assigning it to a VM once its parent tasks are completed.

A. *Simulation Description*

- Virtual machine - is a implemented software of a computer virtually that executes programs like a physical machine.
- Cloudlet-Cloudlet is a Input job or task in the Environment. It stores all information encapsulated in it and the ID of the VM running it.

TABLE 1: SIMULATION RESULT

CLOUDLET	ALGORITHM			
	HHSA	PSO	Max_MiN	FIFO
JOB_10	1562	14688	15062	3250
JOB_15	1641	39979	8938	21422
JOB_30	1703	28469	8390	11031
JOB_60	3281	17907	9453	4203
JOB_90	3343	21078	9110	4860

The empirical analysis was conducted on an I-Ball PC with 2.0 GHz Intel Core-2 duo CPU and 2 GB of memory running windows XP and using CloudSim to construct four virtual machines in single data center. The cloudlets described above are given as Input to algorithm and output is taken for different no. of cloudlet is shown in table 1. Performance Comparison of the all algorithm is shown in table no.1on basis of Makespan. To comparing the performance of algorithm for the workflow scheduling problem, we compare it with two traditional rule-based algorithms and one heuristic algorithms, namely, FIFO, max-min, Particle swarm optimization.

Table 1 show makespan calculated by each algorithm for different algorithm implemented in this paper. Hyper heuristic algorithm showing much less makespan than other and it is much better than other algorithm. It is faster than PSO and traditional algorithm. Max_min and FIFO are showing better performance than PSO. From the result, it is clear that hyper heuristic is makespan reducing high performance algorithm than other heuristic algorithm.

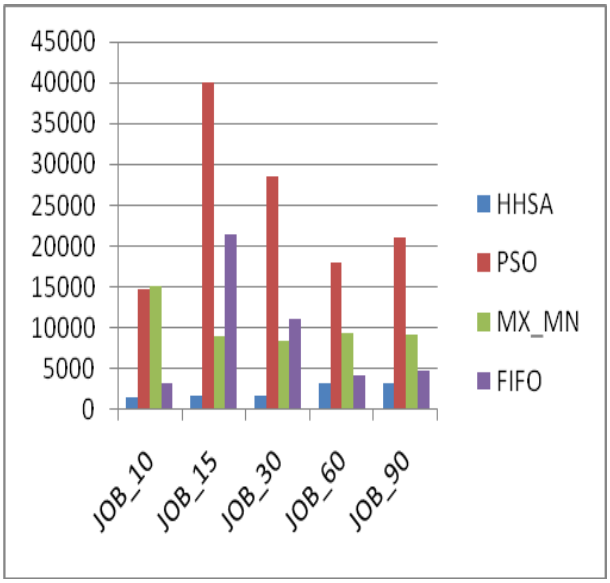

Figure 1.Makespan of all Algorithm

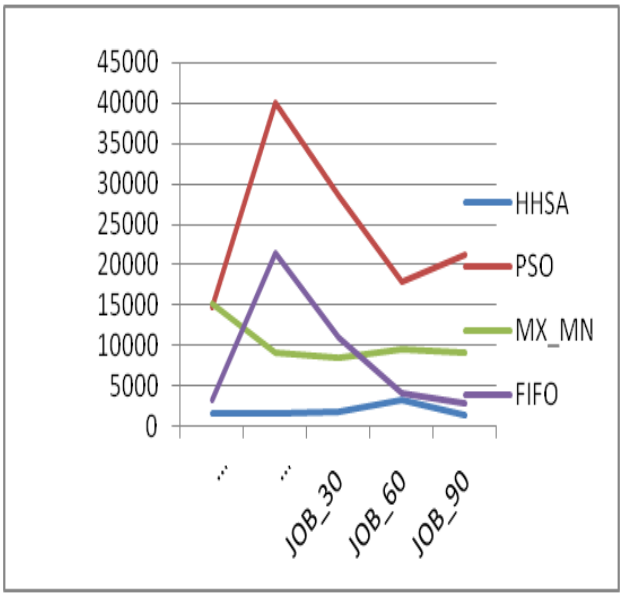

Figure 2. Performance of Algorithms

V. CONCLUSION

Paper presents a high-performance makespan reducing heuristic algorithm to find better solutions for scheduling in cloud computing systems. The algorithm uses two operators to automatically determine when to change the low-level heuristic algorithm and a perturbation operator to fine-tune the solutions obtained by candidate pool algorithm to further improve the scheduling results in terms of makespan. As the simulation results show that hyper–heuristic algorithm provides better results than the traditional scheduling algorithms and also outperforms the other heuristic scheduling algorithms. The simulation results prove that presented algorithm faster than the other heuristic algorithms and computational cost is also low. This is fundamentally different from hybrid heuristic algorithm, which requiring a much longer computation time. Hyper heuristic can be applied to different cloud computing systems to enhance the performance of scheduling problems.

REFERENCES

[1] Wei-Cheng Huang, Meng-Hsiu Chiang Chun-Wei Tsai, "A Hyper-Heuristic Scheduling Algorithm," IEEE TRANSACTIONS ON CLOUD COMPUTING, vol. 2, pp. 236-250, April-June 2014.

[2] Rajkumar Buyyaa, Chee Shin Yeoa, Srikumar Venugopala, James Broberg , Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Elsevie, pp. 3-18, Dec 2008.

[3] The NIST Definition of Cloud Computing, "NIST Special Publication 800-145," Sept 2011.

[4] Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," IEEE International Conference, Pg. 400-407, 2010.

[5] Zhangjun Wu, Zhiwei Ni, Lichuan Gu,Xiao Liu, "A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling," Computational Intelligence and Security IEEE International Conference, pp. 184-188, 2010.

[6] Shailesh Sawant , "A Genetic Algorithm Scheduling Approach for Virtual Machine Resources in a Cloud Computing Environment," Computing Environment. Master's Project, 2011.

[7] Xiaofeng Wang, Chee Shin Yeo, Rajkumar Buyya, Jinshu Su, "Optimizing Makespan and Reliability for Workflow application with reputation and look-ahead genetic Algorithm," Future Generation Computer Systems, vol. 27, no. 8, pp. 1124-1134, March 2011.

[8] Eugen Feller, Louis Rilling, Christine Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds," 12th IEEE International Conference, Pg.., pp. 26-33, November 2011.

[9] Jiandun Li, Junjie Peng ,Zhou Lei , Wu Zhang, "An Energy-efficient Scheduling Approach Based on private clouds," Journal of Information & Computational Science, vol. 8:4, pp. 716-724, 2011.

[10] Zhangjun Wu, Xiao Liu, Zhiwei Ni Dong Yuan and Yun Yang, "A market-oriented hierarchical scheduling strategy in cloud workflow systems," The Journal of Super Computing, Springer US., vol. 63, no. 1, pp. 256-293, March 2011.

[11] H. M. Fard, R. Prodan, J. J. D Barrionuevo and T. Fahringer, "A Multi-Objective Approach for Workflow Scheduling in Heterogeneous Environment Cluster," Cloud and Grid Computing 12th IEEE International Conference, pp. 300-309, 2012.

[12] Timur Keskinturk, Mehmet B. Yildirim and Mehmet Barut, "An Ant Colony Optimization Algorithm for Load Balancing in Parallel Machines with Sequence-Dependent Setup Times.," Computer and Operations Research, vol. 39, no. 6, pp. 1225-1235, 2012.

[13] S. H. Niu, S. K. Ong and A. Y. C Nee, "An Improved Intelligent Water Drops Algorithm for Achieving Optimal Job Shop Scheduling Solution," Engineering ApplicationsofArtificial Intelligence26, pp. 2431-2442, 2013.

[14] Saeid Abrishami, Mahmoud Naghibzadeh and Dick H. J. E Pema., "Deadline-Constrained Workflow Scheduling Algorithm for Infrastructure as a Service.," Journal Future Generation Computer Systems, vol. 29, no. 1, pp. 158-169, 2013.

[15] Kai Wu, Lok Kei Leong, Seungbeom Ma, and Melody Moh Jing Huang, "A TunableWorkflow Scheduling Algorithm Basedon Particle Swarm Optimization for Cloud computing," The International Journal of Soft Computing and Software Engineering , vol. 3, no. 3, pp. 351-358, March 2013.

[16] Rodrigo N. Calheiros and Rajkumar Buyya, "Meeting Deadlines of Scientific Workflows in," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2013 2013.

[17] Ranjit Singh and Sarbjeet Singh, "SCORE BASED DEADLINE CONSTRAINED WORKFLOW SCHEDULING ALGORITHM FOR CLOUD SYSTEMS," International Journal on Cloud Computing: Services and Architecture , vol. 3, December 2013.

[18] Matthew Hyde, Graham Kendall Edmund K. Burke, "A Survey of Hyper-heuristics," University of Nottingham, Nottingham, Computer Science Technical Report NOTTCS-TR-SUB-0906241418-2747, 2009.

[19] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Roseand ,Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," SOFTWARE – PRACTICE AND EXPERIENCE, pp. 23-50, 2011.