Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 2, February 2015, pg.130 – 135

RESEARCH ARTICLE

An Enhanced Hyper-Heuristics Task Scheduling In Cloud Computing

[1]R. Priyanka, [2]M. Nakkeeran

Department of Computer Science and Engineering, India

[1] preethi.me13@gmail.com, [2] nareeksm@gmail.com

ABSTRACT -- Cloud computing is a promising and up-coming technology which allows the users to pay as they require. Cloud computing endow with a proficient and an enhanced way to perform the jobs which were submitted by the users in various aspects such as openness, elasticity, and scalability. Owing to the minimalism and lenience of implementation a variety of rule based algorithms are widely used in cloud computing environment. The necessitate for a scheduling algorithm arises from the requirement for most up to date systems to perform multitasking. Job scheduling is one of the core and challenging issues in cloud environment. The modern heuristics move towards the scheduling on cloud computing and it has spellbound several researchers from different research domains. The multiplicity revealing and enhancement revealing operators are used in the proposed algorithm to determine which low-level heuristic is to be used in finding better solutions for job scheduling.

Keywords -- Cloud Computing, Scheduling, Multiplicity Revealing, Enhancement Revealing, Heuristics

I. INTRODUCTION

Cloud computing is the way of using inaccessible servers on the internet to handle, store and process data instead of using a workstation. Cloud computing is better hooked on to three categories: infrastructure as a service, platform as a service, software as a service. Iaas provides servers and storage on demand with the consumer paying accordingly. Paas allows the users to build and develop the applications within a providers framework. Saas enables the customers to use an application on demand through the browser.

Cloud computing allow the users to access the applications and data from any computer from any location at any time because they are stored on a remote server. It trim down the need for companies to acquire top of the line servers and hardware or engage users to run them since it is all maintained by a third party. There is no need for purchasing any software licences for every user because the software servers are stored and executed remotely by the cloud. By means of centralizing bandwidth, storage, processing and memory in an offsite environment for a charge, cloud computing can significantly minimize the costs. The cloud can also store data therefore companies do not have to residence servers and databases themselves.

In cloud computing, *Scheduling* is the progression of captivating decisions regarding the allocation of available capacity and/or resources to jobs and/or customers on time. Millions of user share cloud services by submitting their millions of computing task to the cloud computing environment. Scheduling of these millions of task is a clash to the cloud environment. The scheduling crisis in cloud makes it difficult to work out, predominantly in the case of large composite jobs like workflows. At the same time, the scheduling strategies focus on throughput, efficiency, space, cost of time and improve the class of service of the entire cloud computing environment.

Scheduling process in cloud is divided into three stages namely; *Resource discovering and filtering, Resource selection, Task allocation.* In *Resource discovering and filtering* the datacenter broker discovers the resources present in the network system and collects status information about the resources. In *Resource selection* the target resource is selected based on the requirements of task and resource. This is a deciding stage. In t*ask allocation*, the task is allocated to selected resource.

II. JOB SCHEDULING IN CLOUD

The needs of job scheduling in cloud computing are load balance, quality of service, economic principles, best running time, throughput. With computing systems being shifted to cloud-based systems progressively, one of the main characteristics is that it works on a pay-as-you-use basis. Several studies attempted to define the scheduling problem on cloud systems as the workflow problem, which can be further classified into two levels: service-level (platform layer and static scheduling) and task-level (unified resource layer and dynamic scheduling). Different from grid computing, the user can install their programs on the virtual machines (VMs) and determine how to execute their programs on the cloud computing system. For these reasons, although both grid computing and cloud computing are heterogeneous, the key issues they face are very different. A good example is the cost and latency of data transfer on these environments. That is why some studies added more considerations to their definitions of scheduling on cloud. For instance, a couple of studies, used directed acyclic graph (DAG) to define the scheduling problem on cloud. The basic idea is to use the vertices of a DAG to represent a set of tasks and the edges between the vertices to represent the dependencies between the tasks.

III. HYPER HEURISTICS

Hyper-heuristics aim to discover some algorithms that are capable of solving a whole range of problems, with little or non-direct human control. Heuristic techniques are often referred to as "search algorithms". The problems are solved by discovering a solution from the set of all possible solutions for a given problem, which is regarded as the "search space". Non-deterministic search methods such as "local search methods", "evolutionary algorithms", "simulated annealing", and other search algorithms offer an alternative approach to an exhaustive search to solve complicated computational problems within a sensible amount of time. These methods guarantee for finding a solution at any time, but it may not be optimum.

Hyper-heuristics must positively influence the selection of heuristics. The optimized heuristics for a given problem should compute high quality solutions. The learning point should refine the algorithms, so that the algorithm solutions subsequently meet the needs of the training set and problems of a certain class can be solved more efficiently. The response mechanism should move towards optimum algorithm solutions in the workspace, as it guides the selection of heuristic. The Algorithm Selection Problem represents in a three-dimensional coordinate system namely the relationship between a problem instance, an algorithm solution and its performance. Comparatively, the two-level model offers a clear separation between the optimization of an algorithm and the optimization process of a specific problem.

IV. RELATED WORK

[3] In *"Hierarchical Queue-Based Task Scheduling"*, they present a queue-based task scheduling algorithm which aims to achieve a minimum completion time of a job that is being scheduled so far as well as for arresting the load balancing of all virtual machines. A global queue is adopted to store all the new arriving jobs. This paper proposes a novel scheduling algorithm to optimize the utilization of all resources and achieve load balancing at the same time and to detect the status of all virtual machines in real time. Based on the history of new coming jobs, the algorithm tries to dispatch all the unscheduled jobs actively into the global queue to avoid long idle time on the virtual machines. But there is poor performance for short jobs or tasks performing

frequent I/O operations and we cannot know how much time a job has remaining. Long running jobs can be starved for CPU.

[5] In *"Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm*" they establish a macroscopic scheduling model with cognition and decision components for the cloud computing, which considers both the requirements of different jobs and the circumstances of computing infrastructure, then propose a job scheduling algorithm based on Multi-Objective Genetic Algorithm (MO-GA), taking into account of the energy consumption and the profits of the service providers, and providing a dynamic selection mechanism of the most suitable scheduling scheme for users according to the real-time requirements. Genetic algorithm is a search heuristic that mimics the process of natural evolution based on a population of candidate solutions. Each chromosome represents a scheduling result, and an evaluation operator (fitness) is called to evaluate the offspring. With the increase of the arrival rate, more applications are rejected. This is because that all the cloud becomes saturated and busy at the high arrival rate, with no ability to accept the new arrival applications. Certain optimization problems cannot be solved by means of genetic algorithm due to poorly known fitness functions which generate bad chromosome blocks.

[7] In "*Context-Aware Job Scheduling for Cloud Computing Environments*", aims at rationalising the resource utilisation in a cloud computing environment, while leading to significant improvement of quality of service. Context can be defined to be a user's physical, social, emotional informational state. It can be described as situations where the individual or machine is immersed. In this scenario, context awareness is the ability to sense and react to situation variations towards better operations. This approach has been used by software to provide better integration with the environment and reduce the need for user's input. They apply context awareness as a tool to automate adaptation of the processing of jobs in the Cloud Computing environment based upon variations of the local environment. For that, they exploit the concept of window of opportunity, wherein the delivery of certain information is more relevant to the user. Thus, the context processing relates to the ability of sensing variation of local context and computing its context distance to a certain condition. This processing is trivial for some aspects of context, such as location and time. This paper do not combine local context information to coordinate the job execution.

V. THE PROPOSED ALGORITHM

A. An Enhanced Hyper-Heuristic Scheduling Algorithm

A high-performance enhanced hyper-heuristic algorithm is proposed for scheduling the jobs on cloud computing systems to shrink the makespan. From the pool of candidate heuristics, one of the heuristic algorithms will be picked by the low-level heuristic (LLH) selection operator as the heuristic algorithm that is to be performed. Two revealing operators one for diversity revealing and one for enhancement revealing are proposed for the proposed algorithm to regulate the effectiveness to employ the low-level heuristic algorithm. The suggested algorithm can be useful to both sequence-dependent and sequence-independent scheduling problems. To assess its performance, the proposed algorithm and numerous other scheduling algorithms are applied in the CloudSim. The basic idea of the proposed algorithm is to use the multiplicity revealing and enhancement revealing operators to balance the escalation and diversification in the search of the solutions during the convergence process. As far as the proposed algorithm described herein is concerned, the low-level heuristic candidate pool consists of simulated annealing, genetic algorithm, particle swarm optimization, and ant colony optimization. The selected enhanced hyper-heuristic algorithm will then be performed repeatedly until the termination criterion is met. More specifically, the selected LLH will evolve the solution for iterations by using the determine function to balance the escalation and diversification of the search directions, which in turn rely on the information provided by the enhancement detection operator.

B. The Enhancement Revealing Operator

A simple random method is used to select the low-level heuristic Hi from the candidate pool H. According to our observation, the best so far makespan (BSFMK) for both SA and GA could continue to improve the results at early iterations (e.g., less than 200 iterations), but it is difficult to improve the results at later iterations (e.g., after 800 iterations), especially when the search directions converge to a small number of directions. If the selected H_i cannot improve the BSFMK after a row of ϕ_{ni} iterations, the improvement revealing operator will return a false value to the high level hyper center to indicate that it should pick up a new LLH. The

improvement revealing operator will return a false value in three cases: the maximum number of iterations ϕ_{max} is reached, the number of iterations ϕ_{ni} the solutions are not improved is reached, and when the stop condition is reached.

C. The Multiplicity Revealing Operator

In addition to the improvement revealing operator, the diversity revealing operator is used by HHSA to decide "when" to change the low-level heuristic algorithm H_i. The diversity of the initial solution $D(A_0)$ will be used as a threshold Θ, i.e., $\Theta = D(A_0)$. The diversity of the current solution $D(AZ)$ is computed as the average of the task distances between individual solutions. If a task in two different individuals is assigned to the same VM, the task distance is 0; otherwise, the task distance is 1. If the diversity of the current solution $D(A)$ is less than the threshold Θ (the diversity of the initial solution), this operator will return false, and HHSA will then randomly select a new LLH.

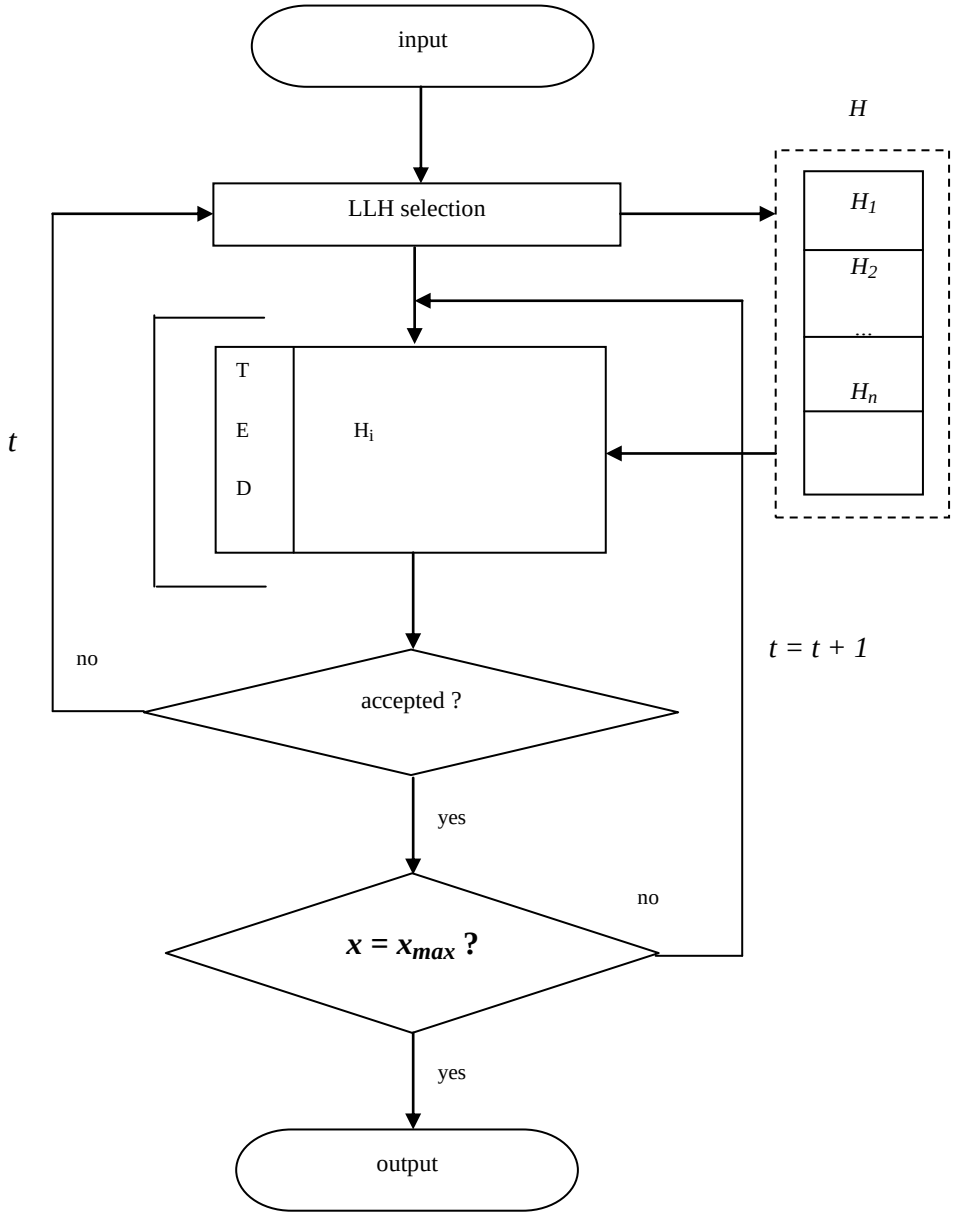

Fig 1 Proposed model

D. Algorithm: Enhanced Hyper Heuristics Scheduling

[1] Set up parameters.
[2] **Input** the scheduling problem.
[3] Initialize the population of solutions $A= \{a_1,a_2, ..., a_N\}$.
[4] Randomly select a heuristic algorithm H_i from the candidate pool H.
[5] While the termination criterion is not met
    a. Update the population of solutions A by using the selected algorithm H_i.
    b. F_1=enhancement_revealing(A).
    c. F_2=multiplicity_revealing(A).
    d. If $\Psi(H_i, F_1, F_2)$
        i. Randomly select a new H_i.
        ii. A=Perturb (A).
    e. End.
[6] End.
[7] **Output** the best so far solution as the final solution.

VI. CONCLUSION

The proposed algorithm uses two revealing operators on impulse to define when to variant the low-level heuristic algorithm and a perturbation operator to fine-tune the answers gained by each low-level algorithm to more expand the scheduling effects in expressions of makespan. The proposed algorithm can not only deliver better results than the traditional rule-based scheduling algorithms, it furthermore overtakes the other heuristic scheduling algorithms, in resolving the workflow scheduling and map-task scheduling difficulties on cloud computing environments. The simple indication of the projected "enhanced hyper-heuristic" algorithm is to influence the possessions of all the low-level algorithms. The future scope is to understand the type of job failures with the intention of improving the dependability of the essential cloud infrastructure from the perspective of cloud providers. Further, the focus is to explore the potential for failure prophecy and incongruity revealing in cloud applications in order to avoid wastage of resources by jobs that fail in due course.

REFERENCES

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

[2] Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, and ChuSing Yang, "A Hyper-Heuristic Scheduling Algorithm for Cloud", IEEE Transactions on Cloud Computing, January 2014.

[3] Wanqing You, Kai Qian, and Ying Qian, "Hierarchical Queue-Based Task Scheduling", Journal of Advances in Computer Networks, Vol. 2, No. 2, 2014.

[4] Christina Delimitrou and Christos Kozyrakis, "Paragon: QoS-Aware Scheduling for Heterogeneous Datacenters", In Proc. of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Houston, March 2013.

[5] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.

[6] Santana-Perez, I. , Ontology Eng. Group, Univ. Politec. de Madrid, Madrid, Spain , Perez-Hernandez, M.S. , "A semantic scheduler architecture for federated hybrid clouds", Cloud Computing (CLOUD), IEEE 5th International Conference on 2012.

[7] Marcos D. Assunc¸ ˜ao, Marco A. S. Netto, Fernando Koch, Silvia Bianchi, "Context-Aware Job Scheduling for Cloud Computing Environments", Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing.

[8] S. Abrishami, M. Naghibzadeh, "Deadline-Constrained Workflow Scheduling in Software as a Service Cloud", Scientia Iranica, Volume 19, Issue 3, June 2012.

[9] Nanduri. R, Maheshwari. N, Reddyraja. A, Varma. V, "Job Aware Scheduling Algorithm for MapReduce Framework", Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference.

[10] M. Rahman, X. Li, and H. Palit, "Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment," Proceedings of the IEEE International Symposium on Parallel and Distributed Processing Workshops, 2011, pp. 966–974.

[11] Bala and I. Chana, "A survey of various workflow scheduling algorithms in cloud environment," Proceedings of the National Conference on Information and Communication Technology , 2011, pp. 26–30.

[12] Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference.

[13] Mohsen Amini Salehi and Rajkumar Buyya, "Adapting Market-Oriented Scheduling Policies for Cloud Computing", Algorithms and Architectures for Parallel Processing, Springer, 2010, pp 351-362.

[14] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, Ion Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling", Proceedings of the 5th European conference on Computer systems, ACM, 2010.

[15] D. Laha and U. Chakraborty, "An efficient hybrid heuristic for makespan minimization in permutation flow shop scheduling", The International Journal of Advanced Manufacturing Technology, vol. 44, no. 5-6, pp. 559–569, 2009.

