AN EFFICIENT VM SCHEDULING ALGORITHM TO MINIMIZE THE MAKESPAN AND MAXIMIZE THE PROFIT

AWS FADHIL IBRAHIM

FSKTM 2019 27

UNIVERSITI PUTRA MALAYSIA

BERILMU BERBAKTI

AN EFFICIENT VM SCHEDULING ALGORITHM TO MINIMIZE THE MAKESPAN AND MAXIMIZE THE PROFIT

By:

AWS FADHIL IBRAHIM

Thesis submitted to the School of Graduate Student, Universiti Putra Malaysia, in fulfillment of the requirement for the degree of Master of Computer Science

2019

COPYRIGHT

DEDICATION

This Thesis is dedicated to:

The sake of Allah. My Creator and my Master.

.My great teacher and messenger, Ahmad (May Allah bless and grant him).

Who taught us the purpose of life.

My beloved Father and Mother for their countless Love, Support and Encouragement.

My Brother and Sister,

And all my friends,

For

Their Endless Patience and Support

ABSTRACT

Abstract of this thesis is presented to the Senate of Universiti Putra Malaysia, in fulfillment of the requirement for the degree of Master of Computer Science

AN EFFICIENT VM SCHEDULING ALGORITHM TO MINIMIZE THE MAKESPAN AND MAXIMIZE THE PROFIT

By:

AWS FADHIL IBRAHIM

2019

Supervisor: MR. AHMAD ALAUDDIN ARIFFIN

Faculty: Computer Science and Information Technology

Virtual Machines (VMs) in Cloud systems are scheduled to host based on the usage of instant resource, namely hosts that are equipped with the highest RAM that is available. This is done without taking their long-term and overall utilization into account. The main issue of VM scheduling is that, it lowers the performance of a system. It is used to schedule tasks for better utilization of resources by allocating certain tasks to particular resources at a particular time. VM scheduling in cloud means to select the most suitable and outstanding resource attainable for execution of tasks or to appoint computer machined to task in such a method that the fulfillment time is

reduced as workable. In this study, we focus on improving the scheduling performance of VM, namely on the cost and makespan. The process of VM scheduling includes three main processes. The first process is VM cluster formation based on the characteristics such as CPU, Memory and bandwidth. The second process is hyper analytical task scheduling algorithm, and based on the scheduled task, the policy based profit maximization algorithm was proposed in final process. The comparison of the performance of proposed work is analyzed through some empirical results. The findings have demonstrated that the proposed work has decreased the task scheduling makespan significantly and gives high profit compared with the other scheduling algorithms.

ABSTRAK

Virtual Machines (VMs) di dalam sistem cloud dijadualkan untuk menjadi host berdasarkan penggunaan sumber segera, dilengkapi dengan RAM tersedia yang tertinggi. Ini dilakukan tanpa mengambil kira penggunaan jangka panjang dan keseluruhan. Isu utama penjadualan VM adalah ia menurunkan prestasi sistem. Ia digunakan untuk menjadualkan tugas-tugas untuk penggunaan sumber yang lebih cekap dengan memperuntukkan tugas-tugas tertentu kepada sumber-sumber terpilih pada masa tertentu. Penjadualan VM dalam cloud bermaksud memilih sumber yang paling cekap dan sesuai untuk menjalankan tugas, atau untuk memperuntukkan tugas pada mesin komputer, untuk mengurangkan masa yang diambil untuk penyelesaian tugas tersebut. Algoritma penjadualan tugas yang efisien diperlukan untuk meningkatkan prestasi sistem. Dalam kajian ini, kami menumpukan perhatian untuk meningkatkan prestasi penjadualan VM, terutamanya ke atas kos dan makespan. Proses penjadualan VM terdiri daripada tiga proses utama. Proses pertama adalah pembentukan kluster VM berdasarkan ciri-ciri seperti CPU, memori dan jalur lebar. Proses kedua adalah algoritma penjadualan tugas hyper analytical, dan berdasarkan tugas yang dijadualkan, algoritma profit maximization telah dicadangkan dalam proses terakhir. Perbandingan prestasi kerja yang dicadangkan dianalisis melalui beberapa hasil empirik. Hasil penemuan telah menunjukkan bahawa kaedah yang dicadangkan telah menurunkan penjadualan tugas makespan dan memberikan keuntungan yang tinggi berbanding dengan algoritma penjadualan yang lain.

ACKNOWLEDGEMENT

To my Lord Allah Almighty, I am thankful for the blessings and virtues, and for reco11cile, strength, patience, courage, and determination he gave me to complete this work to the fullest, Alhamdulillah.

I would like to extend my gratitude to MR. AHMAD ALAUDDIN ARIFFIN, for his supervision, advice, and guidance from the very early stage of this project as well as giving me extraordinary experiences throughout the work. Above all and the most needed, he provided me with unflinching encouragement and support in various ways.

My warmest gratitude goes to all of my family members, especially my father, my mother who always believed in me, gave me all the possible support, and being patient with me for years, providing me with everything, just to make me focus on my goals.

I am also thankful for my brother and sister for their support and concern about my study, and their willing to provide me with any support I need.

Finally, I must extend my sincere thanks to the Ministry of Higher Education in Iraq. None the less, my gratitude to the Malaysian people in general for their perfect hospitability in their green land during my study period.

APPROVAL

This thesis was submitted to the Faculty of Computer Science and Information Technology of Universiti Putra Malaysia and has been accepted as partial fulfillment of the requirement for the degree of Master of Computer Science.

The members of the Supervisory Committee were as follows:

Supervisor: Mr. Ahmed Alauddin Ariffin

Department of Communication Technology and Network

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

Date and Signature: _____

Assessor: Dr. Fahrul Hakim Ayob

Department of Communication Technology and Network

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

Date and Signature: _____

DECLARATION

I declare that the thesis is my original work, except for the quotation and citations, which have been duly, acknowledge. I also declare that it has not been previously, and is not concurrently, submitted for any other degree at Universiti Putra Malaysia or any other institution.

AWS FADHIL IBRAHIM

Date: ----------

LIST OF TABLES

Table 4. 1 List of PM used for experiments 46
Table 4. 2 Clustered PM 48
Table 4. 3 VM Request 49
Table 4. 4 Scheduling Time 50
Table 4. 5 Profit Value 51

LIST OF FIGURES

Figure 3. 1 System Architecture.....25
Figure 3. 2 Implementation Work Flow.....44
Figure 4. 1 Clustered PM 49
Figure 4. 2 Scheduled VM 50
Figure 4. 3 Makespan Comparison 51
Figure 4. 4 Profit Comparison.....52

LIST OF ABBREVIATIONS

VM	Virtual Machine
PM	Physical Machine
CPU	Central Processing Unit
RAM	Random Access Memory
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
Amzaon S3	Amazon Simple Storage Service
Amazon EC2	Amazon Elastic Compute Cloud
DC	Data Center
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony
IWO	Invasive Weed Optimization
RASA	Resource-Aware-Scheduling algorithm
HHSA	Hyper-Heuristic Scheduling Algorithm
F1	Improvement Detection
F2	Diversity Detection
F3	Perturbation Operator

TABLE OF CONTENTS

ABSTRACT ..... **IV**

ABSTRAK ..... **VI**

ACKNOWLEDGEMENT ..... **VII**

APPROVAL ..... **VIII**

DECLARATION ..... **IX**

LIST OF TABLES ..... **X**

LIST OF FIGURES ..... **XI**

LIST OF ABBREVIATIONS ..... **XII**

CHAPTER 1 ..... **1**

INTRODUCTION **1**

1.1 OVERVIEW 1
1.2 PROBLEM DEFINITION 4
1.3 OBJECTIVE 6
1.4 PROJECT SCOPE 6
1.5 MOTIVATION 6
1.6 ORGANIZATION OF THESIS 7

CHAPTER 2 ..... **8**

LITERATURE REVIEW **8**

2.1 OVERVIEW 8
2.3 SUMMARY 21

CHAPTER 3 ..... **22**

METHODOLOGY **22**

3.1 OVERVIEW 22
3.3 PROPOSED ALGORITHM 34
3.3.1 VM Cluster Formation 34
3.3.2 Task Scheduling Algorithm 35
3.3.3 Profit Maximization Algorithm 38
3.4 IMPLEMENTATION DETAILS 38
3.4.1 Netbeans 39
3.4.2 CloudSim 39
3.4.3 Implementation Work Flow 44

CHAPTER 4 ..... **46**

RESULT AND DISCUSSION **46**

4.1 OVERVIEW 46
4.2 SIMULATION RESULTS 46
CHAPTER 5 ..... **53**
CONCLUSION AND FUTURE WORK **53**
5.1 CONCLUSION 53
5.2 FUTURE WORK 53
REFERENCE **55**
APPENDIX **61**
SIMULATION CODE **61**

CHAPTER 1

INTRODUCTION

1.1 Overview

While cloud computing is not considered as a recent technology, it however, as a concept, symbolizes the collaboration between a number of computers together with services, through a network which enables the users to experience a more powerful service. Numerous recent technologies and studies have been done to realize this concept.

Cloud computing, as defined by the National Institute for Standard Technology (NIST) [1], is a pay-per-use model which assists a shared computing resources pool on the access of an available, on-demand, and fitting network, which can be provisioned swiftly, and released with minimal interaction of service provider and management effort.

There are three service model types offered by cloud computing, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Iaas enables the provisioning of computing resources, namely computation resources, network, and storage as services. The services offered through Iaas model by cloud computing are identical to supplying users with hardware resources, for example, Amazon EC2 and Amazon S3. In this computing type, the most vital concept is virtualization. Virtual machines on limited hardware are created through utilization of virtualization technology to gain higher service user number. Virtualization technology enables users to have their own personal hardware which is

able to independently function while not hampering with other users. In addition, hardware virtualization also grants users in requiring numerous environments of hardware. On the other hand, the PaaS model enables the delivery of solution stacks and/or computing platforms by cloud providers, which usually consist of operating system, database, web server, and programming language execution [2]. The third model, SaaS, enables software application to be delivered as services which is utilized on a SaaS vendor managed infrastructure [3]. There are four ways of clouds deployment [1], which are:

Private cloud: This type of cloud infrastructure is designated for an organization exclusively. This might possibly be overseen by the organization or a third party, and can be accessed either on or off the premise.

Community cloud: Instead of working solely for an organization, this cloud infrastructure works by enabling a number of organizations or a community with a common set of areas of concerns, such as security requirements, mission, compliance considerations, and policy, to share it. This cloud infrastructure may be overseen by the organizations or a third party, and can be accessed either on or off the premise.

Public cloud: This cloud infrastructure is owned by the organization that markets cloud services, and provides the service to a large industry group or general public.

Hybrid cloud: This cloud infrastructure consists of a combination between two or more clouds that can include public, community, or private. The clouds remain as unique bodies, but are linked to one another through proprietary or standardized technology which facilitates the portability of data and application, including cloud bursting, which assists in load-balancing between clouds.

Cloud facilitates the computational resources provisioning by on-demand models through virtual machines (VMs), which are set up in a geographically distributed provider data centers. Based on the Pay-Per-Use model, the users share the customer side computational resources. The system performance is significantly impacted by the resource distribution and proper scheduling [4].

Tasks or processes are scheduled based on a set of given requirements and algorithm used, acting as a balancing scenario. Scheduling is divided into two types, which are static and dynamic. The first type, static, schedules the tasks in known environment, which means that there is already a set of information gathered on it, namely the tasks complete structure, mapping of resources before execution, and also the task execution time estimation. On the other hand, in the second type of scheduling, dynamic, relies on both the tasks submitted to cloud atmosphere, and also the current conditions of computer machines and systems, before scheduling them.

Most services in cloud computing is provided by virtualization technique, as a majority of cloud providers comply with the user requests by utilizing virtual machines (VM). Thus, the VM scheduling efficiency is a crucial task in cloud computing [5]. VM scheduling helps in determining on how each resources of physical machines are allotted to launch a VM instance [6]. Based on the requirement fulfilled with the resources requested, namely RAM, Bandwidth, and memory, VM requests are scheduled to the Physical Machines (PM) from a specific Data Center (DC).

In the recent years, cloud computing has evolved in an improved approach to execute the tasks submitted, especially on the flexibility, responsiveness, and scalability. On cloud computing, a majority of task scheduling problems and jobs are mainly either NP-complete or NP-hard. In the recent cloud computing systems, a

majority of rule-based scheduling algorithms, namely deterministic and and exhaustive algorithms are widely utilized as they are easy, and straightforward to implement. However, the rule-based algorithms are not inadequate to be used for any multifaceted or important scheduling troubles, due to the inconsistent results. Thus, this opens an opportunity for scheduling algorithms improvement for schemes of cloud computing, which in return, has attracted researchers of different areas, mainly on the application of modern heuristics. Two or more heuristics are combined in a more comprehensive variants of heuristics, to become a solitary heuristic algorithm, which enables a better scheduling power. The proposed algorithm, hyper heuristics-based algorithm, however, differs from the hybrid heuristic-based scheduling algorithms, which can only function through one or two heuristics. Hyper heuristics-based algorithm works by combining various heuristic algorithms together. The proposed algorithm is particularly essential in leveraging heuristic algorithm strengths, namely Invasive Weed Optimization, Artificial Bee Colony, Ant Colony Optimization, and Particle Swarm Optimization.

The current study mainly aims to contribute on group available PMs into number of clusters before the allocation. Besides that, this study also aims to provide the efficient VM scheduling algorithm that could minimize the make-span of the task, and finally find the maximum profit for providers based on the policy.

1.2 Problem Definition

Cloud computing has paradigm that schedules the VM with minimum make-span and maximum profit, and it is considered as an optimization problem. We need to firstly consider both make-span and resource utilization in order to define the VM scheduling problem. Make-span is the amount of time needed by the resource to carry out the allocation of all VMs. Utilization of VM is defined as how well the resources are used in the cloud to maximize the profit for providers. The efficient scheduling algorithm should schedule the VM with minimum make-span and maximum profit. In this work, the VM scheduling is considered as a problem optimizer, and it is solved through multiple optimization technique.

How to schedule the virtual machine in cloud to provide improved solution in terms of make-span and the profit?

To solve the problem, an efficient VM scheduling algorithm is needed to provide a better VM scheduling in cloud to reduce the make-span. This study introduces novel policy based service model, which aims at maximizing the profit of cloud providers.

The VM scheduling problem can be defined as follows:

Find an optimal solution to schedule a given set of VMs $VM = \{VM_1, VM_2, \cdots, VM_N\}$ to a given set of physical machines $PM = \{PM_1, PM_2, \cdots, PM_M\}$

There are M heterogeneous physical machines for dispatching the VM requests.

A VM request set $VM_{Req} = \{R_1, R_2, \cdots, R_r\}$ is the set of input requests. The VM request represents $R_i = (C, M, B)$ where C is the number of cores, M is the memory and B is the Bandwidth of VM.

1.3 Objective

This study mainly aims to develop an efficient scheduling algorithm, that effectively minimize the scheduling time i.e., make-span and maximize the profit for cloud providers.

1.4 Project Scope

The proposed VM scheduling works on single cloud environment not for multi or distributed environment.

1.5 Motivation

As the hardware resources in the cloud computing system are limited, this could contribute to a lower level of performance. Another issue that needs to be addressed is how to allocate and boost utilization in cloud computing. While various scheduling algorithms have been introduced [19,21], only a number of them fit the cloud, as a majority of them demand a comprehensive prior information of tasks, and do not suit dynamic network environments. This is due to the fact that service providers can only provide the virtual machines and hardware, while users are the ones that

determine the types of tasks executed. The definite workload of a VM generally cannot be assessed by the systems before the operation begins, thus, VM scheduling has to be carried out using limited information. Besides that, the VMs' configuration can either be fixed in an IaaS system or stays the same. While these types of systems are definitely simpler in both implementation and maintenance, they however, are not able to fully comply with the demands of the users. In addition, numerous studies that have implemented single scheduling and fixed data sets in evaluating their methods' performance, and the major drawback of this practice is that, it will not be able to ensure that the system can function efficiently for a prolonged time frame. Thus, to solve the issue of VM scheduling in cloud computing, this study proposes a hyper heuristic-based scheduling algorithm.

1.6 Organization of Thesis

The thesis work is composed of five chapters including the Introduction as chapter 1. It explains overview of cloud computing, problem definition, objective and scope.

Chapter 2 describes the related work; this section covers the existing scheduling algorithms.

Chapter 3 presents the methodology of the project, focusing on overview, and evaluation metrics of proposed algorithm and implementation details.

Chapter 4 is concerned with the evaluation for this project, and includes the details of the parameters and simulation results.

Finally, Chapter 5 provides the conclusion and future work.

Reference

[1] P. Mell and T. Grance,(2009) "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Information Technology Laboratory, Technical Report Version 15, 2009.

[2] Bazarbayev, S., Hiltunen, M., Joshi, K., Sanders, W.H., Schlichting, R.(2013) "Content-based scheduling of virtual machines (VMs) in the cloud". In: Proceedings of 33rd IEEE International Conference on Distributed Computing Systems (ICDCS 2013), pp. 93–101. IEEE (2013)

[3] Archana Pawar, Deepak Kapgate (2014) "A Review on Virtual Machine Scheduling in Cloud Computing", IJCSMC, Vol 3, Apr 2014, pg 928-933

[4] Karan D. Prajapati, Pushpak Raval, Miren Karamt and Potdar(2013) "Comparison of Virtual Machine Scheduling Algorithms in Cloud Computing", International Journal of Computer Applications (0975 – 8887) Volume 83 – No 15, December 2013

[5] Varma, M.K., Choi, E.(2016) "Study and comparison of virtual machine scheduling algorithms in open source clouds". In: The 11th International Conference on Future Information Technology, 20–22 April 2016

[6] S.Sotiriadis, N.Bessis, C.Amza, R.Buyya (2016) "Vertical and horizontal elasticity for dynamic virtual machine reconfiguration", IEEE Trans. Serv. Comput. PP (99) (2016), doi: 10.1109/TSC.2016.2634024. 1–1

[7] K. M. Elsayed and A. K. Khattab (2006), "Channel-aware earliest deadline due fair scheduling for wireless multimedia networks", Wireless Personal Commun., vol. 38, no. 2, pp. 233–252, 2006.

[8] W. H. Kohler (1975), "A preliminary evaluation of the critical path method for scheduling tasks on multiprocessor systems", IEEE Trans. Comput., vol. C-24, no. 12, pp. 1235–1238, Dec. 1975.

[9] R. Ingalls and D. Morrice (2000), "PERT scheduling with resources using qualitative simulation graphs", in Proc. Simul. Conf., 2000, vol. 1, pp. 362–370.

[10] H.-J. Sch€utz and R. Kolisch (2012), "Approximate dynamic programming for capacity allocation in the service industry", Eur. J. Oper. Res., vol. 218, no. 1, pp. 239–250, 2012.

[11] D. Shi and T. Chen (2013), "Optimal periodic scheduling of sensor networks: A branch and bound approach", Syst. Control Lett., vol. 62, no. 9, pp. 732–738, 2013.

[12] S. M. Johnson (1954), "Optimal two- and three-stage production schedules with setup times included", Naval Res. Logistics Quart., vol. 1, no. 1, pp. 61–68, 1954.

[13] M. R. Garey, D. S. Johnson, and R. Sethi (1976), "The complexity of flowshop and jobshop scheduling", Math. Oper. Res., vol. 1, no. 2, pp. 117–129, 1976.

[14] Saeed Parsa and Reza Entezari-Maleki (2009), "RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.

[15] Arash Ghorbannia Delavar, Mahdi Javanmard, Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi,(2012) "RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012

[16] El-Sayed T.El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy (2012) "Extended Max-Min Scheduling Using Petri Net and Load Balancing" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012

[17] Linlin Wu, Saurabh Kumar Greg, (2014) "SLA based Resource Provisioning for Hosted Softwareas-a-Service Applications", IEEE Trans. Services Computer., vol. 7, no. 3, pp. 465–485, Jul. 2014.

[18] Dario Bruneo (2013) "A Stochastic Model to Investigate Data Center Performance and QoS" in IaaS Cloud Computing Systems Proc. 32nd IEEE Int.Conf. Comput. Commun., 2013, pp. 2148–2156.

[19] A. Shahina Banu and W. R. Helen (2014) "Self Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud". EEE Trans. Autom. Sci. Eng., vol. 12,no. 1, pp. 309–323, Jan. 2014.

[20] Jianying Luo, Lei Rao, and Xue Liu (2014) "Temporal Load Balancing with Service Delay Guarantees for Data Center Energy Cost Optimization", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 3, pp. 775–784, March 2014.

[21] Haitao Yuan, Jing Bi (2014) "CAWSAC: CostAware Workload Scheduling and Admission Control for Distributed Cloud" Data Centers vol. 2, no. 1, january-march 2014

[22] Wenhong Tian (2009) "A Toolkit for Modeling and Simulation of Real-time Virtual Machine Allocation in a Cloud" Data Center Resource scheduling in infrastructure as a service (IaaS). Future Gener. Comp. Sy., vol. 25, no. 6, pp. 599–616, 2009.

[23] Yeo, C.S.,& Buyya,R.(2004)."Pricing for utility-driven resource management and allocation in clusters". In Proceedings of the ADCOM(pp.32–41). Ahmedabad, India.

[24] Maglaras, C.,& Meissner, J.(2006)."Dynamic pricing strategies for multi product revenue management problems". Manufacturing & Service Operations Management,8(2),136–148.10.1287/msom.1060.0105

[25] Eren,S.S.,& Maglaras, C.(2009)."Monopoly pricing with limited demand information". Journal of Revenue and Pricing Management, 9(1-2), 23–48.

[26] C. Tsai et al.(2014) "A hyper-heuristic scheduling algorithm for cloud" IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 236–250, Feb. 2014.

[27] Selvarani S, Sadhasivam G (2010) "Improved cost-based algorithm for task scheduling in cloud computing". In: Proceedings of the IEEE international conference on computational intelligence and computing research, pp 1–5

[28] Lin C, Lu S (2011) "Scheduling scientific workflows elastically for cloud computing." In: Proceedings of the IEEE international conference on cloud computing, pp 746–747

[29] Blum C, Roli A (2003) "Metaheuristics in combinatorial optimization: overview and conceptual comparison". ACM Comput Surv35(3):268–308

[30] Ferrandi F, Lanzi PL, Pilato C, Sciuto D, Tumeo A (2010) "Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems." IEEE Trans Comput Aided Des Integr Circuits Syst 29(6):911–924

[31] Lu W, Zhiliang W, Siquan H, Lei L (2013) "Ant colony optimization for task allocation in multi-agent systems". China Commun 10(3):125–132

[32] Bai L, Hu YL, Lao S, Zhang WM (2010) "Task scheduling with load balancing using multiple ant colonies optimization in grid computing". In: Proceedings of the international conference on natural computation, pp 2715–2719

[33] Lu X, Gu Z (2011) "A load-adapative cloud resource scheduling model based on ant colony algorithm" In: Proceedings of the IEEE international conference on cloud computing and intelligence systems, pp 296–300

[34] Zuo X, Zhang G, Tan W (2014) "Self-adaptive learning PSO based deadline constrained task scheduling for hybrid IaaS cloud". IEEE Trans Autom Sci Eng 11(2):564–573

[35] Garcia-Nieto J, Olivera A, Alba E (2013) "Optimal cycle program of traffic lights with particle swarm optimization." IEEE Trans Evol Comput 17(6):823–839

[36] Wen X, Huang M, Shi J (2012) "Study on resources scheduling based on ACO allgorithm and PSO algorithm in cloud computing." In: Proceedings of the international symposium on distributed computing and applications to business, engineering science, pp 219–222

[37] Xie X, Wu P (2010) "Research on the optimal combination of ACO parameters based on PSO." In: Proceedings of the international conference on networking and digital Society, vol 1, pp 94–97

[38] Kruekaew B, Kimpan W (2014) "Virtual machine scheduling management on cloud computing using artificial bee colony", Proceedings of the International MultiConference of engineers and computer scientists, vol 1, pp 12–14

[39] Kimpan, W., & Kruekaew, B. (2016). "Heuristic Task Scheduling with Artificial Bee Colony Algorithm for Virtual Machines". n Soft Computing and Intelligent

Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems, 2016 Joint 8th International Conference on (pp. 281-286). IEEE.

[40] L.D. Dhinesh Babu,P.Venkata Krishna(2013),"Honey bee behavior inspired load balancing of tasks in cloud computing environments",Applied Soft Computing, Vol. 13,2013;pp.2292-2303

[41] R. Jena, (2017) "Task scheduling in cloud environment: A multi-objective ABC framework", Journal of Information and Optimization Sciences, vol. 38, no. 1, pp. 1-19, 2017.

[42] Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013). User-priority guided min-min scheduling algorithm for load balancing in cloud computing. 2013 National Conference on Parallel Computing Technologies, PARCOMPTECH 2013.

[43] Devipriya, S., & Ramesh, C. (2013). Improved max-min heuristic model for task scheduling in cloud. Proceedings of the 2013 International Conference on Green Computing, Communication and Conservation of Energy, ICGCE 2013, 883–888.

[44] Zheng, Z., Wang, R., Zhong, H., & Zhang, X. (2011). An approach for cloud resource scheduling based on parallel genetic algorithm. ICCRD2011 - 2011 3rd International Conference on Computer Research and Development, 2, 444–447.

[45] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology, 10, 340–347.

[46] Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Computing, 18(2), 829–844.

[47] Tawfeek, M., El-Sisi, A., Keshk, A., & Torkey, F. (2015). Cloud task scheduling based on ant colony optimization. International Arab Journal of Information Technology, 12(2), 129–137.

[48] Wkh, P., Ri, E., Vr, D., Lqihulru, W., Zloo, F., Ljqruhg, E. H., … Wkh, R. (n.d.). * , ,1752’8&7,21.

[49] Wei, L., Zhang, X., Li, Y. Y., & Li, Y. Y. (2012). An Improved Ant Algorithm for Grid Task Scheduling Strategy. Physics Procedia, 24, Part C(0), 1974–1981.

[50] Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011). Cloud task scheduling based on load balancing ant colony optimization. Proceedings - 2011 6th Annual ChinaGrid Conference, ChinaGrid 2011, 3–9.

[51] Raghavan, S., Sarwesh, P., Marimuthu, C., & Chandrasekaran, K. (2015). Bat algorithm for scheduling workflow applications in cloud. 2015 International Conference on Electronic Design, Computer Networks and Automated Verification, EDCAV 2015, 139–144.

[52] Sadasivam, G. S., & Selvaraj, D. (2010). A novel parallel hybrid PSO-GA using MapReduce to schedule jobs in Hadoop data grids[1] G. S. Sadasivam and D. Selvaraj, "A novel parallel hybrid PSO-GA using MapReduce to schedule jobs in Hadoop data grids," Proc. - 2010 2nd World Congr. Nat. Biol. Inspi. Proceedings - 2010 2nd World Congress on Nature and Biologically Inspired Computing, NaBIC 2010, 377–382.

[53] Raju, R., Babukarthik, R. G., Chandramohan, D., Dhavachelvan, P., & Vengattaraman, T. (2013). Minimizing the makespan using Hybrid algorithm for cloud computing. Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013, 957–962.