Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,
IMECS 2014, March 12 - 14, 2014, Hong Kong

Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony

B. Kruekaew and W. Kimpan

Abstract—**Resource scheduling management design on Cloud computing is an important problem. Scheduling model, cost, quality of service, time, and conditions of the request for access to services are factors to be focused. A good task scheduler should adapt its scheduling strategy to the changing environment and load balancing Cloud task scheduling policy. Therefore, in this paper, Artificial Bee Colony (ABC) is applied to optimize the scheduling of Virtual Machine (VM) on Cloud computing. The main contribution of work is to analyze the difference of VM load balancing algorithm and to reduce the makespan of data processing time. The scheduling strategy was simulated using CloudSim tools. Experimental results indicated that the combination of the proposed ABC algorithm, scheduling based on the size of tasks, and the Longest Job First (LJF) scheduling algorithm performed a good performance scheduling strategy in changing environment and balancing work load which can reduce the makespan of data processing time.**

Index Terms—**Artificial Bee Colony, Cloud Computing, Scheduling Management, Virtualization Machine**

I. Introduction

RECENTLY, internet communication has become one of the most important factors in daily life activities. Consequently, internet is the center of data sharing. Due to the large amount of increasing users, the data is often changed or modified. This causes the heavy task scheduling for computers while less services can be processed to serve numerous users. However, this problem can be solved by using new technology of Cloud computing which focuses on networking and computing, storage, and data service resources.

Cloud computing is an emerging concept of information technology service. It consists of both infrastructure as well as the application services and focuses on types of users' requirements. The users can identify their needs through the software in Cloud.

Cloud computing [1], [2] is a terminology used in describing various computing concepts that involve a large number of computers connected through a real-time communication network such as the internet. Cloud computing is the result of evolution and adoption of existing technologies and paradigms. Its goal is to allow the users to take advantages from these technologies without the requirements of deep knowledge or expertise. The Cloud aims to reduce costs and to provide the ease of resource management. Virtualization is the principle technology for Cloud computing. It provides the agility in operations, and reduces cost by increasing infrastructure utilization. On the other hand, Cloud computing automates the process through which the users can utilize resources on-demand.

Virtualization [3], [4] is critical to Cloud computing because it delivers services by providing a platform for optimizing complex Information Technology resources in a scalable manner. The main aim of the virtualization is an ability to run the multiple VMs on a single machine by sharing all the resources that belong to the hardware.

The problem of load balancing services occurs when the services from the users make a request to access to the same server while other servers have no request from the services. This phenomenon is called distributed load imbalance system. In this situation, it can be solved by scheduling the tasks or the services before using the system. Therefore, a good task scheduler can increase the performance of resource utilization and can reduce the makespan of assigned tasks which is called distributed load balance system.

Task scheduling in distribution system such as Cloud is used for balancing work load. It requires some conditions for example, stability of the system, makespan of work, ability to adapt to the environment changing, and etc. Those conditions in task management are similar to the behaviors of Artificial Bee Colony algorithm (ABC algorithm). ABC is inspired by the behavior of a honey bee colony in nectar collection. Bee colony based on the social behavior has the ability to adapt to the environment changing.

The suggestion for enhancing the performance of cloud task scheduling is to reduce the makespan of a given task set and increate performance of resource utilization by using improved ABC algorithm. It can be used to solve the problem and the appropriate value can be found. Moreover, it has the ability to adapt to the environment changing [5]-[7].

The structure of this paper is organized as follows: the section II presents an overview of the related work. The Artificial Bee Colony is presented in section III. Section IV presents Artificial Bee Colony inspired scheduling and load balancing algorithm. Section V is the experimental results and discussion. Finally the conclusion and future work are presented in section VI.

Manuscript received November 21, 2013; revised December 3, 2013.
B. Kruekaew is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520, Thailand (e-mail: s3650853@kmitl.ac.th).
W. Kimpan is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520, Thailand (e-mail: knwarang@kmitl.ac.th).

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Related Work

Scheduling and load balancing techniques are crucial for implementing the efficient parallel and distributed applications in making the appropriate use of parallel and distributed system. Task scheduling can increase the efficiency of Cloud computing if it is able to work under the reasonable resources [4], [8], [9].

Li *et al.* [10] proposed the tree structure method in order to solve Job-Shop Scheduling in supporting the needs of a complex and flexibility to Job-Shop Scheduling problem. It also appropriately managed the specific features of model management and scheduling algorithm by using Reusable Scheme and Multi-Agent system. Later, Fang *et al.* [11] presented task scheduling in Cloud computing considering the requirements of the users and load balance in Cloud environment when the users increase and change the environment. The scheduling strategy was simulated by using CloudSim toolkit package.

Calheiros *et al.* [12], [13] developed the CloudSim simulation for modeling and simulation of virtualized Cloud-based datacenter environments, including dedicated management interface for VMs, memory, storage, and bandwidth. Simulation-based approaches can evaluate Cloud computing system and application behaviors offering significant benefits.

Mondal *et al.* [14] proposed troubleshoot of load balance in Cloud computing using Stochastic Hill Climbing. In 2011, Hsu *et al.* [9] presented a focus on energy efficiency in datacenter by considering the task scheduling to physical server and reducing energy in the system. The criterion used to measure the performance of an energy loss was to prepare computer exceeds the requirement of the job. Experiment strategies for allocating server to a sequence of jobs were a largest machine first heuristic, a best fit method, and a mixed method. The experiment results indicated that all three algorithms waste less energy consumption in over-provision.

In 2010, Hu *et al.* [15] introduced the scheduling strategy on load balancing of VM resource in Cloud computing environment by adopting a tree structure using Genetic algorithm for scheduling. It considered previous data and the current state of work in advance to the performance behavior of the system which can solve the problem of load imbalance in Cloud computing.

In 2012, Wei *et al.* [16] presented Genetic algorithm for scheduling in Cloud computing to increase the system performance. Li *et al.* [17] proposed a Load Balancing Ant Colony Optimization (LBACO) Algorithm. ACO was use to schedule on load balancing and reduce makespan in Cloud.

Karaboga *et al.*, in 2012 [7], presented ABC algorithm which was used to solve the problem and find the most appropriate value within environment changing. ABC algorithm based on the swarm-based optimization algorithm and it is used to solve the optimization problems in searching, electrical engineering, data mining, and etc. In the same year, Bitam *et al.* [18] proposed Bee Life algorithm which was used for scheduling in Cloud computing. Bee Life algorithm is inspired by the behavior and reproduction of bee to find food source. The algorithm evaluated the performance of the resources and it has the aim to reduce time and complexity of work.

Mizan *et al.* [19] presented job scheduling in Hybrid cloud by modifying Bee Life algorithm and Greedy algorithm to achieve an affirmative response from the end users and utilize the resources.

III. Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga [5], [6], [20], [21]. It is the method to find the appropriate value. This method is inspired by the foraging behavior of honey bees. In ABC model, there are three kinds of honey bee to search food sources, which include scout bees search for food source randomly, employed bees search around the food source and share food information to the onlooker bees, and onlooker bees calculate the fitness and select the best food source. In the nature, bees can extend themselves over long distances in multiple directions. After scout bees find the food source and return to the hive, they compare the quality of food source and go to the dance floor to perform a dance known as “waggle dance”. The waggle dance is the communication of bees to shares the information about direction of the food source, distance from the hive, and the nectar amount of the food source. While sharing information, bees evaluate the nectar quality and energy waste. After sharing information on the dance floor, onlooker bees select the best food source and then scout bees will return to the food source to bring nectar back to the hive. A pseudo code of ABC algorithm is described in Fig. 1.

Begin

Initialize the population of the scout bees, generate randomly scout bees into the food sources and calculate the fitness values.

Repeat

- Each the employed bees search around the food sources and update the new fitness, if the new fitness is better than the old values.
- Select employed bees and recruit onlooker bees to search around the food sources and calculate on their fitness values.
- Choose the onlooker bees with have the best fitness value.
- Send scout bees into the food sources to discover new food sources.

Until (Stopping criterion is not met)

End

Fig. 1. Basic of ABC algorithm.

IV. Artificial Bee Colony Inspired Scheduling and Load Balancing Algorithm

Cloud computing provides a dynamic resource pool of VM according to different requirements from the users or the system. The routing of services which request to the diverse servers, depend on the Cloud management policies based on load of individual server. Scheduling management on Cloud computing is different degrees of loads on each and every VM. This may lead to different load among the VMs. Load balancing techniques are effective in reducing the makespan and response time.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Let $VM = \{VM_1, VM_2, VM_3, \ldots, VM_N\}$ is a set of N virtual machines and $Task = \{task_1, task_2, task_3, \ldots, task_K\}$ is a set of K task to be scheduled and processed in VM. All the machines are unrelated but are paralleled. Scheduling is non-preemptive which means that the processing of the tasks on VMs cannot be interrupted. The flowchart of the VM scheduling and load balancing using ABC algorithm is shown in Fig. 2.

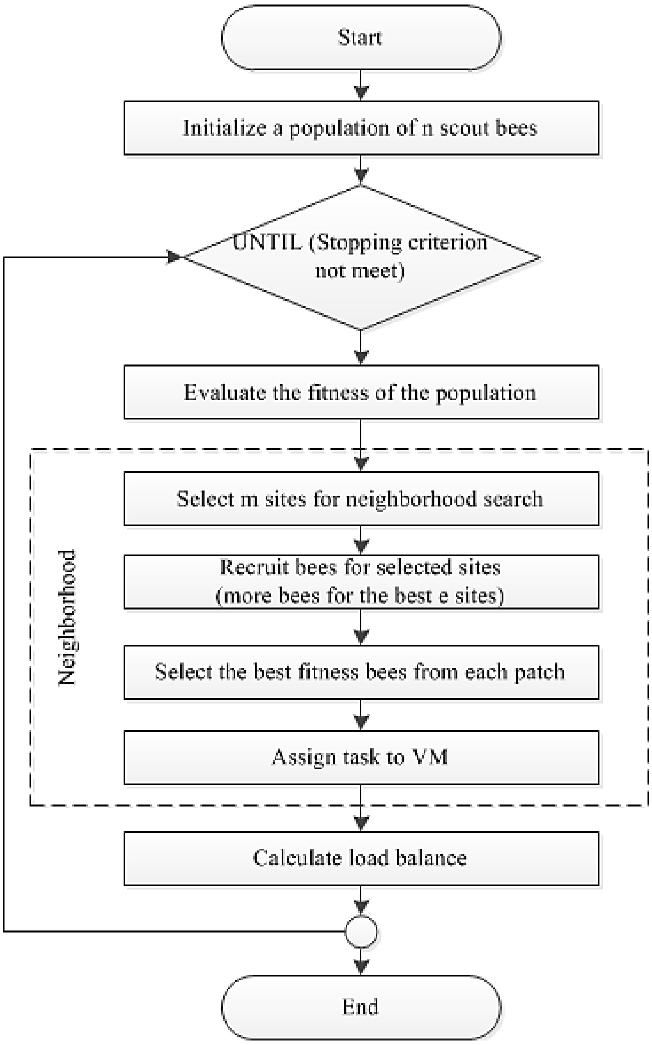

Fig. 2. Flowchart of the VM scheduling and load balancing using ABC algorithm.

A. Initialize Population

At the beginning, the initial n scout bees are placed randomly in VMs on Cloud computing and n is the number of scout bees.

B. Evaluate the Fitness of the Population

Fitness is calculated based on (1).

$$fit_{ij} = \frac{\sum_{i=1}^{n} task\_length_{ij}}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (1)$$

Where fit_{ij} is the fitness of the bees population of i in VM_j or capacity of VM_j with bee number of i. $task\_length$ is the length of the task that has been submitted in VM_j and $capacity_j$ is the capacity of VM_j calculating based on (2).

$$capacity_j = pe\_num_j \times pe\_mips_j + vm\_bw_j \quad (2)$$

Where $capacity_j$ is a capacity of VM_j, pe_num_j (processing element) is the number processor in VM_j, pe_mips_j is million instructions per second of each processor in VM_j, and vm_bw_j is the network bandwidth ability of VM_j.

C. Select m Sites for Neighborhood Search

Scout bees with the highest fitness are chosen as “Selected Bee” and the sites visiting by them are chosen from neighborhood of m VMs.

D. Recruit Bees for Selected Site

Send more bees to neighborhood of the best e VM, then evaluate the fitness based on (3).

$$fit_{ij} = \frac{\sum_{i=1}^{n} task\_length_i + InputFile\_length}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (3)$$

Where $InputFile\_length$ is the length of the task before execution.

E. *Select the Best Fitness Bees from Each Patch and Assign Task to* VM_j

For each round, the bee with the best fitness will be chosen to assign task in VM_j.

F. Calculate Load Balance

After submitting tasks to the under loaded VM_j, the current workload of all available VMs can be calculated by using the information that received from the datacenter. Thus, Standard Deviation (S.D.) is calculated in order to measure the deviations of load on VMs. Standard deviation of load can be defined as (4):

$$S.D. = \sqrt{\frac{1}{n}\sum_{j=0}^{n}(X_j - \bar{X})^2} \quad (4)$$

Processing time of a VM:

$$X_j = \frac{\sum_{i=1}^{k} task\_length_i}{capacity_j} \quad (5)$$

Mean of processing time of all VMs:

$$\bar{X} = \frac{\sum_{j=1}^{n} X_j}{n} \quad (6)$$

Where $S.D.$ is Standard deviation of load, n is number of all VM. X_j is processing time of VM_j which can be calculated based on (5) and $\bar{X}$ is mean processing time of N virtual machines which can be calculated based on (6).

If the S.D. of the loaded VM is less than or equal to the mean, then the system is in a balance state. On the other hand, if the S.D. is higher than the mean, then the system is in an imbalance state.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. Experimental Results

A. Implementation Environment

According to the algorithm described above, the simulation using CloudSim-3.0.1 Tools will be addressed. The experiments consist of 10 datacenters and 100-700 tasks under the simulation platform. The parameters setting on the CloudSim-3.0.1 is shown in Table I.

TABLE I
PARAMETERS SETTING OF CLOUD SIMULATOR

Type	Parameter	Value
Datacenter	Number of Datacenter	10
	Number of Host	5
	Type of Manager	Space_shared, Time_shared
	Number of PE per Host	2 - 4
	Bandwidth	2000
	Host Memory (MB[a])	2048 - 10240
	Datacenter Cost (The cost of using processing in this resource)	10
Virtual Machine (VM)	Total number of VMs	30 - 210
	MIPS[b] of PE[c]	1000 - 2000
	Number of PE per VM	1
	VM Memory (MB)	512 - 2048
	Bandwidth (Bit)	1000
	Type of Manager	Time_shared
Task	Total number of Tasks	100 - 700
	Length of Task (MI[d])	5000 - 20000
	Number of PE per requirement	1
	Type of Manager	Space_shared

[a]MB = Megabyte, [b]PE = Processing Element, [c]MIPS (Million Instructions Per Second) is a measure of the processing speed of the computer, MI[d] is Million Instruction.

B. Parameters Setting of ABC Algorithm

The parameters setting of improved ABC algorithm is shown in Table II.

TABLE II
PARAMETERS SETTING OF IMPROVED ABC ALGORITHM

Symbol	Parameter	Value
n	Number of scout bees	1000
m	Number of sites selected out of n visited site	5
e	Number of best site out of m select site	1
nep	Number of bees recruited for best e site	800
nsp	Number of bees recruited for other (m-e) selected sites	200

C. Experimental Results

The evaluation of the performance of the proposed method can be described and compared with the scheduling algorithm as the followings:

- First Come First Serve (FCFS) is considering the sequence of arrival of the job.
- Shortest Job First (SJF) is considering the size of the job by selecting the shortest job first.
- Longest Job First (LJF) is considering the size of the job by selecting the longest job first.

There are two experiments; the first experiment is the comparison of the average makespan with the number of requests changing, the second phase is the comparison of the average makespan with the number of VMs changing.

In the first experiment, the comparison of the average makespan with the number of requests changing from the different scheduling algorithm is performed. The number of VMs is fixed as 50 VMs and the number of requests gradually increased between 100 – 700 tasks. The x axis shows the effect on performance of increased requests as shown in Fig. 3.

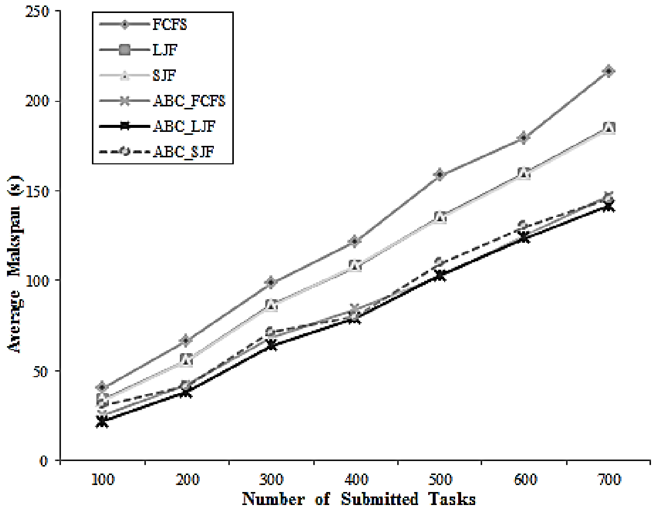

Fig. 3. Comparison of average makespan among FCFS, LJF, SJF, ABC_FCFS, ABC_LJF, and ABC_SJF algorithm based on the fixed Number of VMs and the increased number of the requests.

Fig. 3 shows the average makespan of the scheduling using improved ABC algorithm with FCFS, LJF, and SJF (ABC_FCFS, ABC_LJF, and ABC_SJF), comparing with the original scheduling (FCFS, LJF, and SJF) which was based on 50 fixed VMs and the number of increasing requests. The experimental results indicated that, while the number of requests was increasing, the average makespan consequently increased. It can be concluded that ABC_LJF performed the effective results in the optimal scheduling on the loaded system.

In the second experiment, the comparison of the average makespan with the number of VMs changing from the different scheduling algorithm is performed. The number of tasks was fixed as 500 tasks and gradually increased the number of VMs from 30 to 210 VMs. The x axis shows the effect on performance in increasing the system size (VMs).

Fig. 4 shows the average makespan of the scheduling using ABC_FCFS, ABC_LJF, and ABC_SJF comparing with FCFS, LJF, and SJF which was based on the fixed number of tasks and the number of increasing VMs. It can be concluded that ABC_LJF performed better than all methods and its performance is more prominent in scalability.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

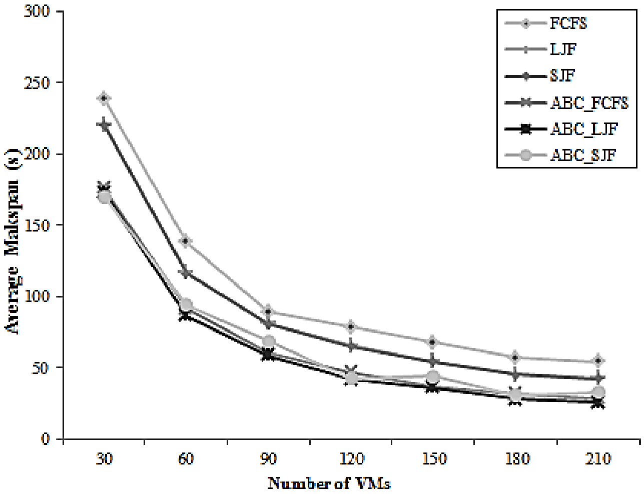

Fig. 4. Comparison of average makespan between FCFS, LJF, SJF, ABC_FCFS, ABC_LJF, and ABC_SJF algorithm based on the fixed number of the requests and the increased number of VMs.

VI. Conclusion and Future Work

This paper presents ABC optimization algorithm which can solve the Virtual machine scheduling management under the environmental changing of the number of VMs and requests on Cloud computing. Even in changing environment, Cloud computing needs to be operated in a stable system. Therefore, ABC algorithm is suitable for Cloud computing environment because the algorithm is able to effectively utilize the increased system resources and reduce makespan. The experimental results illustrated that the proposed methods of ABC_LJF performed effective results than all methods and its performance is more prominent in scalability. Under the circumstance of increasing or decreasing the number of servers, the load balancing algorithm should be done by using ABC_LJF algorithm in order to maintain system stability and scheduling and to prevent the system crash. For further studies, the preemptive Virtual machine scheduler operating with independent and heterogeneous tasks on Cloud computing will be focused.

References

[1] K. Tang, J. M. Zhang, and C. H. Feng, “Application centric lifecycle framework in cloud,” *IEEE 8th Int. Conf. on e-Business Engineering (ICEBE)*, 2011, 19-21 Oct. 2011, pp. 329-334.

[2] 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,” in *Communications of the acm*, vol. 53, no. 4, April 2010, pp. 50-58.

[3] K. Dutta, “A smart job scheduling system for cloud computing service providers and users: modeling and simulation,” in *1st Int’l Conf. on Recent Advances in Information Technology RAIT-2012*, 2012.

[4] S.Sindhu, and S. Mukherjee, “Efficient task scheduling algorithms for cloud computing environment,” in *High Performance Architecture and Grid Computing, Communications in Computer and Information Science*, 2011, vol. 169, 2011, pp. 79-83.

[5] D. Karaboga, and B. Basturk, “Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems,” in *IFSA 2007*, LNAI 4529, 2007, pp. 789–798.

[6] D. Karaboga, and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” in *Applied Soft Computing*, 2008, pp. 687–697.

[7] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: artificial bee colony (ABC) algorithm and applications,” in *Artificial Intelligence Review 2012, Springer Science Business Media B.V. 2012*, March 2012.

[8] S. T. Maguluri, R. Srikant, and L. Ying, “Stochastic models of load balancing and scheduling in cloud computing clusters,” in *Proc. IEEE Infocom*, 2012, pp. 702-710.

[9] Y. C. Hsu, P. Liu, and J. J. Wu, “Job sequence scheduling for cloud computing,” in *Int. Conf. on Cloud and Service Computing (CSC 2011)*, 2011, Dec. 2011, pp. 212-219.

[10] N. Li, Z. Hong, and H. Xiaoting, “Dynamic integration mechanism for job-shop scheduling model base using Multi-agent,” in *2009 Int. Conf. on Information Management, Innovation Management and Industrial Engineering*, 2009, vol. 4, Dec. 2009, pp. 421-425.

[11] Y. Fang, F. Wang, and J. Ge, “A task scheduling algorithm based on load balancing in cloud computing,” in *Web Information Systems and Mining, Lecture Notes in Computer Science 2010*, Vol. 6318, 2010, pp. 271-277.

[12] R. N. Calheiros, R.Ranjan, C. A. F. D. Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software : Practice and Experience,* Vol. 41, No.1, Jan. 2011, pp. 23-50.

[13] R. N. Calheiros, R.Ranjan, C. A. F. D. Rose, and R. Buyya, “CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services,” *arXiv preprint arXiv: 0903.2525*, 2009.

[14] B. Mondal, K. Dasgupta, and P. Dutta, “Load balancing in cloud computing using Stochastic Hill Climbimg-A soft computing approach,” in *Procedia Tachnology*, vol. 4, 2011, pp. 783-789.

[15] J. Hu, J. Gu, G. Sun, and T. Zhao, “A scheduling strategy on load balancing of virtual machine resources in cloud computing environment,” in *3rd Int. Symp. on Parallel Architectures, Algorithms and Programming(PAAP)*, 2010, 18-20 Dec. 2010, pp. 89-96.

[16] Y. Wei, and L. Tian. “Research on cloud design resources scheduling based on genetic algorithm,” in *2012 Int. Conf. on Systems and Informatics (ICSAI 2012)*, 2012, May 2012, pp. 2651-2656.

[17] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, “Cloud task scheduling based on load balancing Ant Colony Optimization,” in 6^{th} *Annual ChinaGrid Conf.*, 2011, Aug. 2011, pp. 3-9.

[18] S. Bitam, “Bees life algorithm for job scheduling in cloud computing,” in *Conf. on Computing and Information Technology (*ICCIT *2012)*, 2012, pp. 186-191.

[19] T. Mizan, S. M. R. A. Masud, and R. Latip, “Modified bees life algorithm for job scheduling in hybrid cloud,” in *Int. Journal of Engineering and Technology(IJET)*, 2012, vol. 2, no.6, June 2012, pp. 974-979.

[20] D. Karaboga, “An Idea based on honey bee swarm for numerical optimization,” in *Technical Report TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[21] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The Bees Algorithm-A novel tool for complex optimisation problems," in *Proc. of IPROMS 2006 Conf.*, 2006, pp. 454-461.