Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

Novel Artificial Bee Colony Algorithm Based Load Balance Method In Cloud Computing

Linlin Tang, Jeng-Shyang Pan, Zhifang Wang, Kaiqiang Ma and Hongnan Zhao
Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen China
hittang@126.com;

Zuohua Li
School of Civil and Environment Engineering, Shenzhen China
lizuohua@hit.edu.cn

Received February 2016; revised February, 2017

Abstract. *with rapid development of cloud computing and the Internet, load balance techniques are more and more important than ever. A good scheduling algorithm is an important method to solve the load balance problems. A new load balance algorithm based on ABC algorithm is proposed here. It can be seen as a new scheduling method based on swarm intelligence algorithm. Good experimental results proved by simulation tools called cloudsim has shown the efficiency.*
Keywords: Cloud Computing; Load Balance; Artificial Bee Colony

1. **Introduction.** Cloud computing has recently emerged as a new paradigm of hosting and delivering services over the Internet [1, 2]. It is a new way in which people needn't compute on local computers, but on centralized facilities operated by third-party compute and storage utilities. And the computing is transferred from "endpoint" to "cloud" [3]. More and more corporations have participated in this area since 2007, such as Microsoft, Google, Amazon and so forth. All these companies have provided the cloud computing products themselves respectively. One of the definition for the Cloud Computing can be described as below.

A large-scale distributed computing paradigm[13] that is driven by the economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platform and services are delivered on demand to external customers over the Internet [4].

Based on different types of service, three scenarios where Clouds are used can be given: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Virtualization, parallel computing, grid computing and distributed computing can be seen as the important techniques to cloud computing [5, 6].

Load balance problem belongs to combination optimization research. It can also be called the task scheduling problems[14, 15]. An algorithm based on the Artificial Bee Colony[20, 21] is proposed and the experimental results which have been done on Cloudsim [7, 8], a cloud simulation toolkit, are used to show its efficiency. Other method based on such swarm intelligence algorithms has been paid attention for its good performance [16]. Our work is organized as follows: in section 2 related works are given and our proposed algorithm is shown in section 3. Experimental results are proposed in section 4.

Conclusion and some future work is described in section 5. Acknowledgment is in section 6.

2. Related work.

2.1. Methods and Algorithms for Load Balance Problems. As shown in the above introduction part, the rapid development of the computer network and Internet applications show that the load balance in cloud computing has been widely used to improve the system response time and enhance utilization of system resources[9]. Generally speaking, some important and famous development time and events can be summarized as follows.

(1)1996-1999: Finding opportunities: starting stage
(2)2000-2003: Developing in enterprises[10]
(3)2003-2005: Network upgrading
(4)2006-2008: Load Balance upgrading: application development
(5)2006: Google CEO Eric Schmidt first put forward the notion of cloud computing on SES San Jose 2006
(6)After 2006, cloud computing as an important part of load balance has been widely spread

Load balance algorithm is related to consumer requirements and system efficiency. It is worth to say that the load balance problem is an NP problem, and most of the optimization algorithms for it are sub-optimal algorithm[11, 12]. In fact, performance of an algorithm is restrained by many factors, such as the complex network environment, different performance of nodes, servers, hosts and so on.

2.2. Cloudsim Experimental Platform. Cloudsim is cloud computing imitating software developed by the University of Melbourne and Gridbus in 2007. Its basic frame is shown in the following figure 1.

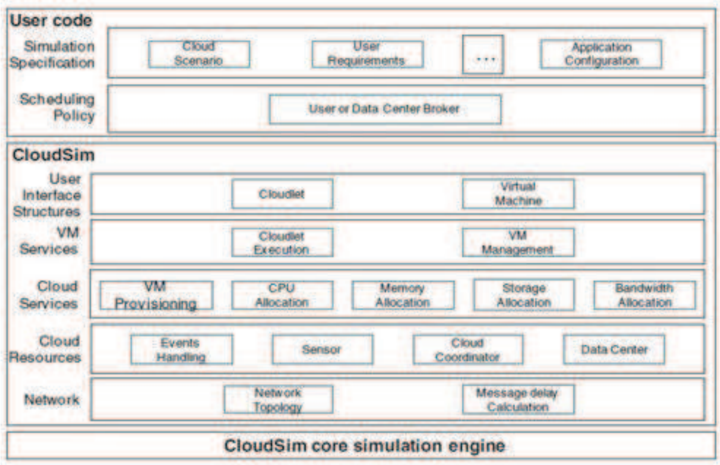

FIGURE 1. Basic frame of CloudSim

Generally speaking, there are three layers in CloudSim, including the cloudlets layer, the VMs layer and the datacenter layer. The cloudlets layer is provided by service providers to users for simulating apps. The second layer is the VMs layer which offers virtual machines to process tasks. The last one is the datacenter layer which is the cloud computing system.

3. OUR PROPOSED METHOD.

3.1. **Parameter Settings.** In order to describe the whole designed systems, the VMs and the cloudlets should be established firstly. So, their parameters based on the actual system should also be set in advance. Parameters of a VM show the characteristics of the VM, such as memory, storage, bandwidth and so forth. A task, which has a pre-assigned instruction length and amount of data, is represented by a cloudlet.

The parameters of a VM can be described as follows:

1) The identification of the VM is defined as *vmid*. It is the unique symbol of the VM that is used to distinguish different VMs in the system.
2) $MIPS$ are the millions of instructions per second which is used to show the computing speed. It is a fundamental variable to measure the speed of CPUs.
3) The image size is defined as *size*. It is the size of storage in a VM and is used to set available space.
4) The ram describes VM memory. As long as the computer is running, operation data will be transferred to the memory. When the tasks are completed, the results will be sent from the ram.
5) The BW is described for the bandwidth. It represents that how much data can be transmitted in a fixed period, and it also shows capacity of transmitting data in the transport pipeline.
6) The number of *cups* is pesnumber.
7) The virtual machine manager name is VMm.

The parameters of cloudlet can be described as follows:

1) The cloudlet is marked as *id*. It is the unique symbol of a cloudlet in the overall system, so that we can use it to distinguish different cloudlets.
2) The length of a cloudlet is recorded as *length*. It refers to the size of a cloudlet that if a VM can finish the applied task depending on.
3) The size of a file is described as $filesize$.
4) The size of an output file is *outputsize*. After tasks are completed, it figures the size of the cloudlets that will be output.
5) The utilization of a cloudlet is $utilizationModel$.

3.2. **Mathematical Model.** The load of virtual machines discussed in this paper is expressed by the number of cloudlets and the capacity of VMs [17]. As shown in the following formular (1).

$$LOAD = MIPS * NumCPU + BW \tag{1}$$

As shown in the above formula, $LOAD$ is the capacity of CPU, the MIPS gives millions of instructions per second, the $NumCPU$ illustrates the number of cups, and the BW presents the bandwidth.

In fact, capacity of a VM's load is based on the MIPS, the number of CPU and the bandwidth.

First of all, we will give the definition of load balance. It is based on the variance of the tasks lengths sum and is scheduled by every VM shown as below.

$$P_i = F_i / \sum_{j=1}^{vmNum} F_j \tag{2}$$

In the above formular (2), P_i is the percentage of a VM tasks length in all VMs. F_i is the length of scheduled tasks of a VM that describes the load.

At last, we introduce a judgment of load balance as shown in the following formulation (3).

$$bla = \sqrt{\sum_{j=1}^{vmNum}((\sum_{i=1}^{vmNum} P_i)/vmNum - P_j)^2/vmNum} \quad (3)$$

Here, the $vmNum$ is the number of all VMs and the variance represents average rate about the load of a VM accounting to the load of all VMs.

3.3. **Tasks Scheduling Algorithm.** Tasks scheduling algorithm is aimed at searching the better solutions for a tasking assignment system at one time. In other words, it gives a locally optimal solution. It means that the algorithm cannot have access to the optimal solutions for all problems, whereas it can produce a wide range of optimal approximate solutions for the global optimal solutions [18, 19].

The scheduling algorithm in this paper is represented as follows:

1) **Initialization**: Initialize all parameters of VMs and cloudlets. Based on the actual environment which we need in our projects, they are set.
2) **Selection**: According to the rate ($n\%$), $n\%$ of all VMs are chosen as a sample to implement the NO.i task. At this step, we will choose the optimal VM in all selected VMs by formulation (2). Because of less VMs, the efficiency can be accepted.
3) **Iteration**: After choosing a VM, due to the randomness of the VM selected in the Step 2, here, you can utilize the iteration formulation for searching a more efficient VM to implement the task. The iteration formulation which we can find a more optimal solution according to it can be shown in the following formulation (4).

$$LOAD_i(t+1) = LOAD_i + \phi(LOAD_i(t) - LOAD_k(t)) \quad (4)$$

In the above formulation, the iterative times is represented by t, and ϕ is a random number between 0 and 1. Through the iteration, a better solution to scheduling the task to VMs will be got.
4) **Judging overload**: After computing "bla", we can judge if the current system is overloaded or not. According to the result, we decide whether we will use the limited formulation to balance the system or not. If the system is balanced, then go to the Step 6. Otherwise go to the Step 5.
5) **Limited**: If the result in the Step 4 presents the current system is overloaded, we will use the formulation below for renewing the system to schedule the task to another VM. The limited formulation is as show in the following formulation (5).

$$LOAD_i = LOAD_i + r * (LOAD_{max} - LOAD_{min}) \quad (5)$$

6) **Judging balance**: If the system is not balanced, then judge the loop ends and based on the result make a decision to decide whether to go to Step 3 or not. Otherwise the system distributes the tasks directly.
7) **Judging end**: By judging whether there is another task, we will go back to the Step 2 or stop the process.

Generally speaking, the tasks are scheduled to a better VM. Its advantage is that the system can be avoided to be overloaded; the algorithm makes all tasks scheduled more efficiently.

The whole process can be shown in the following figure 2.

In our algorithm, the balance criteria are aimed at implementing the program under the circumstances that can be controlled. It can keep balance of the system in the current environment. At last, when all VMs are overloaded and there are remnant cloudlets not to be processed, tasks will be scheduled based on a random method.

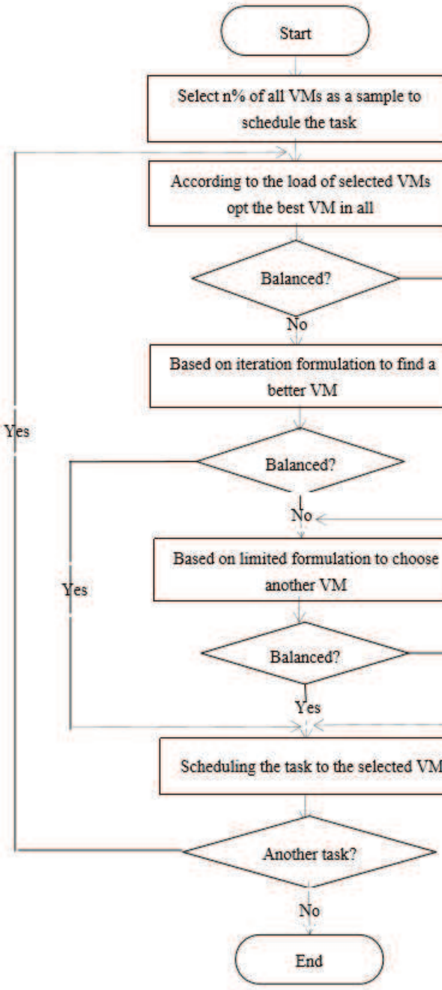

Figure 2. The proposed ABC algorithm based load balance method

4. **EXPERIMENTAL RESULTS AND ANALYSIS.** To show the efficiency and the accuracy of our proposed ABC algorithm based on load balance method, the CloudSim is used for testing and giving some experimental results.

In the **first experiment**, 350 cloudlets, 30 VMs and 750 cloudlets, 70 VMs are used in the experiments respectively. Some results are described in the following figure 3. The detailed experimental results are shown in the following tables corresponding to the above two curves in figure 3.

The above graph illustrates that there is a fluctuation between 10000 and 50000. It is that because the algorithm is an intelligent and random algorithm. The intelligent algorithm which does not produce a constant value is a dynamic random algorithm.

The **second experiment** gives the results when the number of cloudlets and the number of VMs change simultaneously. As shown in the following figure 4. And the table 2 shows the detailed data in this experiment.

In the above figure, when the ratio of cloudlets and VMs is fixed in every group, with the number of cloudlets and VMs increasing simultaneously, the lines all fluctuate. The worst line sharply climbs between the first and fifth group. After that until the tenth group, it has a fluctuation from the value 20000 to 70000. Because of the intelligent

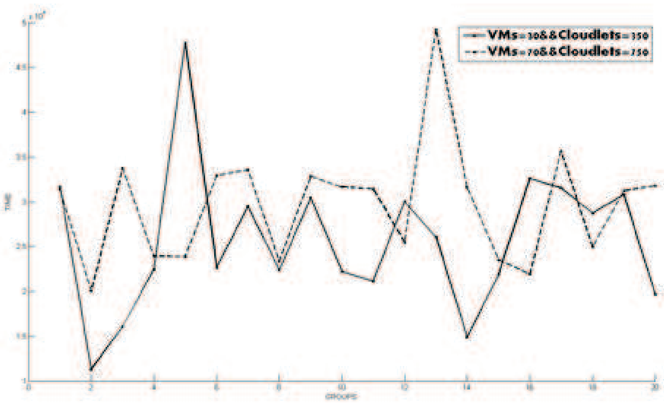

Figure 3. The results in the first group

Table 1. Detail data for experiment 1

(a) 16 times testing "finish time" results for 70 VMs and 750 Cloudlets

31349.53	20083.28	33729.2	23945.36	23914.7
32920.51	33552.31	23413.74	32866.8	31647.99
31413.04	25476.38	49233.06	31578.18	23515.54
21933.93	35610.36	24949.06	31209.13	31800.49

(b) Results for 30 VMs and 350 Cloudlets

31643.31	11309.4	16030.74	22470.5	47741.43
22659.38	29474.47	22358.58	30478.46	22208.69
21142.57	30029.56	26066.7	14929.83	21855.76
32606.21	31522.05	28731.22	30811.19	19660.03

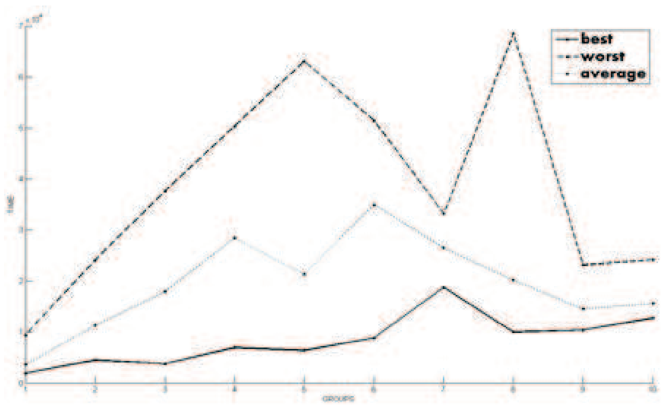

Figure 4. The results in the second group

algorithm, the worst line shows some random results. However, the best line remains stable during the first and sixth group and then there is a slight fluctuation. Finally, a

TABLE 2. Parameters and results in the experiment 2

(a)

	1	2	3	4	5
VMs	10	20	30	40	50
Cloudlets	100	200	300	400	500
Best	1957.91	4529.31	3808.55	6929.32	6401.9
Worst	9432.57	24110.44	37572.82	50335.75	63061.89
Average	3632.91	11347.72	17918.12	28429.51	21287.72

(b)

	6	7	8	9	10
VMs	60	70	80	90	100
Cloudlets	600	700	800	900	1000
Best	8836.42	18710.71	10001.59	10409.98	12736.28
Worst	51454.2	33145.77	68567.37	23156.78	24230.93
Average	34977.3	26458.91	20223.36	14524.97	15589.29

gentle grow occurs in the average line till to the fourth group. Furthermore, the average line decreases slowly after a fluctuation.

5. CONCLUSIONS.

Based on the ABC algorithm, we propose a load balance method to solve the load problems on the Internet. Experimental results prove its efficiency.

A load balance method based on the ABC algorithm for cloud computing has been proposed in this paper. This is a good method that the ABC algorithm is used in solving the load balance problems. It can schedule the tasks to reasonable VMs and keep the system under the balanced circumstance. Due to the lots of experiments, we can get a conclusion that the algorithm can complement the load balance and meet the requirements from the clients.

Additionally, through the cloudsim toolkit, we can simulate the actual system and test the algorithm. The cloudsim is an efficient tool to test the algorithm in cloud computing environment. To to find a more efficient scheduling method is our future work.

ACKNOWLEDGMENT. The authors would like to thank for the support from the project NSFC (National Natural Science Foundation of China) with the Grant number: 61202456, the project NSFC with the Grant number: 61201399, the Innovation Fund of Harbin Institute of Technology with the Grant number: HIT. NSRIF. 2015087,the two Shenzhen science and technology projects with the Grant number: JCYJ20150403161923527, JCYJ20160608161351559, and the Shenzhen science and technology techniques research project with Grant number: JSGG20150330103937411. In addition, authors will also thank for the support from the China Scholarship Council (CSC) for No. 20150612512..

REFERENCES

[1] H. F. Zhu, T. H. Liu, D. Zhu, and H. Y. Li, Robust and Simple N-Party Entangled Authentication Cloud Storage Protocol Based on Secret Sharing Scheme. *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 2, pp. 110-118, April 2013.

[2] B. R. Chang, H. F. Tsai, and C.-M. Chen, Evaluation of Virtual Machine Performance and Virtualized Consolidation Ratio in Cloud Computing System. *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, No. 3, pp. 192-200, July 2013.

[3] Q.Zhang, L. Cheng, R. Boutaba, Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, vol. 1, No. 1, pp. 7-18, 2010.

[4] I. Foster, Y. Zhao, I. Raicu, et al, Cloud computing and grid computing 360-degree compared. *Proc. of Grid Computing Environments Workshop*, pp. 1-10, 2008.

[5] L. M. Vaquero, L. Rodero-Merino, J. Caceres, et al., A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, vol. 39, No. 1, pp. 50-55, 2008.

[6] Y. Jadeja, K. Modi, Cloud Computing-Concepts, Architecture and Challenges. *Proc. of the International Conference on Computing and Electronics and Electrical Technologies*, pp. 877-880, 2012.

[7] R. Buyya, R. Ranjan, R. N. Calheiros, Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *Proc. of International Conference on High Performance Computing and Simulation*, pp.1-11, 2009.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, et al.. 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, pp.23-50, 2011.

[9] S. Das, H. Viswanathan, G. Rittenhouse. Dynamic load balance through coordinated scheduling in packet data systems INFOCOM 2003. *Proc. of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 1, pp. 786-796, 2003.

[10] T. D. Braun, H. J. Siegel, N. Beck, et al.. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing*, vol. 61, no. 6, pp.810-837, 2001.

[11] J. Canon, P. Alexandrino, I. Bessa, et al.. Genetic diversity measures of local European beef cattle breeds for conservation purposes. *Genetics Selection Evolution*, vol. 33, no. 3, pp. 311-332, 2001.

[12] L .Jijian, H. Longjun, W. Haijun, Prediction of vibration response of powerhouse structures based on LS-SVM optimized by PSO.*Engineering Sciences*, vol. 2, pp. 009, 2011.

[13] A. Kazem, A. M. Rahmani, H. H. Aghdam, A modified simulated annealing algorithm for static task scheduling in grid computing, Computer Science and Information Technology, *Proc. of ICCSIT'08*, pp. 623-627, 2008.

[14] Yulan J, Zuhua J, Wenrui H, Multi-objective integrated optimization research on preventive maintenance planning and production scheduling for a single machine. *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 954-964, 2008.

[15] S. Pandey , L. Wu, S. M. Guru, et al., A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. *Proc. of IEEE International Conference on Advanced Information Networking and Applications*, pp. 400-407, 2010.

[16] X. Hua, J. Zheng, W. Hu, Ant colony optimization algorithm for computing resource allocation based on cloud computing environment, *Journal of East China Normal University (Natural Science)*, vol. 1, no. 1, pp. 127-134, 2010.

[17] L. D. Babu, P. V. Krishna. Honey bee behavior inspired load balance of tasks in cloud computing environments. *Applied Soft Computing Journal*. vol. 13, no. 5, pp. 2292-2303, 2013.

[18] P. W. TSai, J. S. Pan, B. Y. Liao,et al.. Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081-5092, 2009.

[19] Vaclav Snasel, Lingping Kong, Pei-Wei Tsai, and Jeng-Shyang Pan. Sink Node Placement Strategies based on Cat Swarm Optimization Algorithm. *Journal of Network Intelligence*, vol. 1, no. 2, pp. 52-60, 2016.

[20] P. W. Tsai, W.-L. Wang, J. F. Chang, Z.-S. Chen, and Y.H. Zhang. Utilizing ABC and Time Series Model in Investigating the Influence of Adding Monitoring Indicator for Foreign Exchange Rate Forecasting. *Proc. of ICGEC 2016*, pp. 7-9, 2016.

[21] P. W. Tsai, K. F. Liu, X. S. Xue, J.-F. Chang, C.-L. Tu, Vac Istanda, and C.-F. Wu, Comparative Analysis of Stock Portfolio Construction by IABC and GARCH with the ISCI in Taiwan Stock Market. *Proc. of IIHMSP 2016*, pp. 21-23, 2016.