An Energy-Efficient Scheme for Cloud Resource Provisioning Based on CloudSim

Yuxiang Shi
College of Computer Science
Zhejiang University
Hangzhou 310027, China
shiyuxiang220@gmail.com

Xiaohong Jiang
College of Computer Science
Zhejiang University
Hangzhou 310027, China
jiangxh@zju.edu.cn

Kejiang Ye
College of Computer Science
Zhejiang University
Hangzhou 310027, China
yekejiang@zju.edu.cn

***Abstract*—Cloud computing has recently received considerable attention. With the fast development of cloud computing, the datacenter is becoming larger in scale and consumes more energy. There is an emergency need to develop efficient energy-saving methods to reduce the huge energy consumption in the cloud datacenter. In this paper, we achieve this goal by dynamically allocating resources based on utilization analysis and prediction. We use "Linear Predicting Method" (LPM) and "Flat Period Reservation-Reduced Method" (FPRRM) to get useful information from the resource utilization log, and make M/M/1 queuing theory predicting method have better response time and less energy-consuming. Experimental evaluation performed on CloudSim cloud simulator shows that the proposed methods can effectively reduce the violation rate and energy-consuming in the cloud.**

***Keywords*-cloud computing; energy efficiency; resource prediction; M/M/1 model;**

I. Introduction

Recently, the development of cloud computing has made great impact on Information Technology [1]. With the fast development of cloud computing, the datacenter is becoming larger in scale and consumes more energy [2]. The cloud brokers always try to make full use of the hardware and get more profit from the datacenter. So the method that can reduce energy-consuming, make more efficient use of the hardware, and optimize the system performance is widely discussed [3–7]. A good resource scheduling and management scheme is the fundamental for on-demand resource allocating, performance optimizing, load balancing and energy saving. In this paper, we will discuss the methods to make resource utilization prediction, based on which, a new cloud resource provisioning scheme is presented to achieve the goal of energy efficiency.

The cloud offers all kinds of services, and the web service takes a large part. The resource utilization of web applications has obvious characteristics of time-regularity and special patterns. The resource utilization log contains much useful information that can help to improve predicting accuracy. We can combine this useful information with resource predicting methods to reduce the violation rate.

The main contribution of this paper is to use "Linear Predicting Method" (LPM) and "Flat Period Reservation-Reduced Method" (FPRRM) to get some useful information from the utilization log, and also improve the M/M/1 Queuing Theory Predicting Method (MMQMPM) with better response time during the rapidly growing period and reduce the reserved resource in steady sequence. And in the experiment part, the paper uses modified *CloudSim* [8] cloud simulator to check the results of these two improvements of MMQMPM.

II. Related work

A. Processing Ability Scaling

Dynamic Voltage and Frequency Scaling (DVFS) is one of the most commonly used power reduction techniques in high performance processors. DVFS can vary the frequency and voltage of a microprocessor to reduce the energy-consuming [9]. Many power-reducing schemes use predicting methods to predict the next-time CPU utilization and set the frequency and voltage of the CPU following the predicted value to save the spare computing power. In CMOS chips, the energy-consuming contains two main parts, one is the static consuming and the other is dynamic consuming. And the dynamic part takes a large part in the total energy-consuming. The dynamic energy-consuming is related to the voltage and frequence of CPU. The relation can be expressed in formula (1) [10].

$$P = c \times V_{dd}^2 \times f \tag{1}$$

Here, P is the CPU's dynamic power, c is a constant, V_{dd} is the the voltage of CPU and f is the frequence. And normally, the frequence is in a direct ratio with voltage of CPU. So the formula (1) can be reduced to formula (2).

$$P = c^{'} \times f^3 \tag{2}$$

The $c^{'}$ is a constant. Then, the dynamic energy-consuming follows the cubic model. In the part of Result and Analysis, this energy model is used to estimate the energy-consuming of the algorithm.

B. M/M/1 Queuing Model Predicting Method (MQMPM)

The main prediction scheme in this paper, the MQMPM, is based on continuous-time birth and death process, M/M/1 queuing theory which is about single waiter Markov queuing

978-0-7695-4516-5/11 $26.00 © 2011 IEEE
DOI 10.1109/CLUSTER.2011.63

model and Formula Little in Queuing Theory. The rate of web service requirement follows the Poisson Process; it is the same as the customer arriving rate in M/M/1 queuing model. So, the M/M/1 queuing model is fit for the web service modeling. Here, we supposed that the customer arrival rate is the same in short period. So, the probability distribution of the states in continuous-time birth and death process follows the hypergeometric distribution. So, the average number of customers in the system can be got from the formula (3) [11].

$$E[N] = \rho/1 - \rho \tag{3}$$

Here, the average number of customers is $E[N]$ and $\rho = \lambda/\mu$, λ is the customer average arrival rate, μ is the system average service rate. The Formula Little claims that the average number of customers in the system is equal to the average customer arrival rate multiply the average duration of each customer. We can represent Formula Little as formula (4):

$$E[N] = \lambda \times E[T] \tag{4}$$

Here, $E[N]$ is the average number of customers, $E[T]$ is the average duration each customer spends in the system, λ is the customer average arrival rate. Then, the necessary service rate can be claimed as formula (5). So, the predicted service rate can be got by knowing the average duration of each customer and the average customer arrival rate.

$$\mu = \lambda + 1/E[T] \tag{5}$$

The λ can be considered as the web requirement rate, μ means the service rate to fit for task and $E[T]$ is the response time of the web service. So, the resource utilization can be predicted by formula (5). Here, λ can be estimated as the average web requirement rate in recent period. The predicted resource utilization μ contains extra resource reservation to ensure the system is stable in continuous-time birth and death process [12].

This predicting method has some shortcomings. One is that this method has bad predicting performance when the sequence continues to increase rapidly. The predicted value curve is delayed some time compared with the real utilization sequence. And another shortcoming is that the resource reservation in flat and smooth period is too much. Aiming these two points, this paper offers two improvements of the MQMPM.

III. Improvement Methods

A. The Linear Trend Predicting Method (LTPM) Combined Algorithm

The Figure 3 shows that the trend of the predicted sequence, especially during the rapidly increasing period, is delayed by a phase. This is because we use the average task arrival rate to predict the next-time resource utilization and the average rate may lower the high utilization. But in other smooth changing sequences, the predict results of MQMPM are perfect. To overcome the predicted value delaying in MQMPM, the LTPM is used to improve the predicting method. The LTPM responds very fast in the continuous increasing sequence. It can reduce the predicting delay of MQMPM dramatically. But because this rapid respond characteristic, it will cause predicting jolt in little fluctuation period. So, this method cannot be used directly in utilization predicting. But the advantages and disadvantages of these two methods can complement each other.

Then, we can set a combining factor α to combine the two predicting methods together, when the resource utilization is continuous increasing, use the LTPM; and during other time, it will use MQMPM. This method can be presented as (6):

$$P(t) = (1 - \alpha) * P_{mml(t)} + \alpha * P_{line(t)} \tag{6}$$

$P_{mml(t)}$ and $P_{line(t)}$ are the predicted values of MQMPM and LTPM separately. The α contains two parts, one is range factor β_{range}, the other is length factor θ_{length}. The α can express as formula (7):

$$\alpha = \beta_{range} + \theta_{length} \tag{7}$$

β_{range} is related to the increasing change range. If the range is bigger, this factor is larger; when the range is smaller this factor is less. This rule is because if the increasing change range is big, the LTPM needs fast response to change, the LTPM will take more contribution to the next time prediction. θ_{length} is related to the trend duration. The paper supposes that the trend duration follows the normal distribution with the mean and standard deviation of trend length in the utilization log, the θ_{length} is equal to the probability of the normal distribution. So, when the trend duration is close to the average length, factor θ_{length} is bigger; if the trend duration is much shorter or longer than the average, perhaps this duration is just jolt or cannot continue anymore, so the factor θ_{length} is relatively small.

And also, there is also some other useful and advanced information in the utilization log which can improve the predicting accuracy. When using the above method to make predict, you can double check the fault points. Some special patterns can be found there. Sometimes, these special pattern can cause many predicting violations. Then they can be added into the predicting method. The patterns can be found by hand or by some data mining method.

B. Flat Period Reservation-Reducing Method in MQMPM

Again in Figure 3, during the time from 190 to 418, the task required resource is relative stable and the predicted value is much high than the real resource utilization. Although the high resource utilization predicted value does not make any violation, there is a lot of unnecessary power-consuming.

So, another shortcoming of the MQMPM is that the predicted value is relatively high when the utilization sequence is flat and smooth. So we can reduce this high predicted value to a lower level.

First of all, we have to find the flat and smooth period in the resource utilization sequence. The standard deviation in a short-time utilization sequence before the predicted point can be used to judge whether the sequence is flat. If the standard deviation is smaller than the predetermined threshold value, then the sequence can be considered as flat and smooth. The second thing is to predict the resource utilization. If the sub-sequence is considered as flat and smooth period, the predicted value is the last-time utilization before the predicting point. And if the standard deviation shows that the sequence is not flat any more, the predict method changes back to MQMPM. This improvement can decrease the flat period sequence predicted value, then the method can get more energy-reducing.

IV. The Experiment

A. The CloudSim Platform

The experiment is on the CloudSim cloud simulator which is a framework for modeling and simulating the cloud computing infrastructures and services [8]. The CloudSim simulator has many advantages: it can simulate many cloud entities, such as datacenter, host and broker. It can also offer us a repeatable and controllable environment. And we do not need to take too much attention about the hardware details and can concentrate on the algorithm design. The simulated datacenter and its components can be built by coding and the simulator is very convenient in algorithm design.

The main parts which relate to the experiments in the article and the relationship between them are shown in Figure 1.

- **CloudInformationService:** It is an entity that registers, indexs and discovers the resource.
- **Datacenter:** It models the core hardware infrastructure, which is offered by Cloud providers.
- **Datacenter Broker:** It models a broker, which is responsible for mediating negotiations betweeen SaaS and Cloud providers.
- **Host:** It models a physical server.
- **Vm:** It models a virtual machine which is run on Cloud host to deal with the cloudlet.
- **Cloudlet:** It models the Cloud-based application services.
- **VmAllocation:** A provisioning policy which is run in datacenter level helps to allocate VMs to hosts.
- **VmScheduler:** The policies required for allocating process cores to VMs. It is run on every Host in Datacenter.
- **CloudletScheduler:** It determines how to share the processing power among Cloudlets on a virtual machine. It is run on VMs. [8]

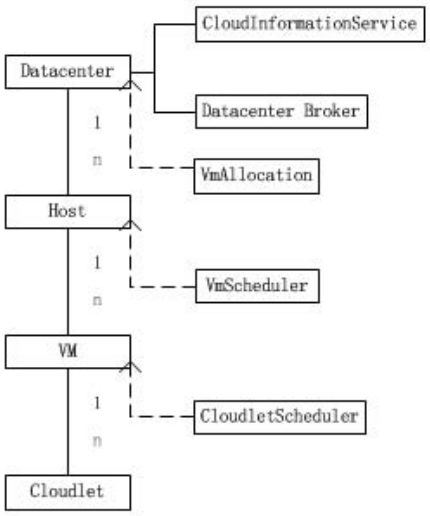

Figure 1. The Main Parts and Relationship of CloudSim.

Table I
The main API of Class *Predictor*.

Methods	return value	function
getPredictedVal	double	Implement the predicting algorithm and its return value is the need of resource.
getPower	double	Get the power of resource.
setResponseTime	void	Set the standard service response time.

And in order to implement the virtual machine (VM) dynamic resource allocation, some modifications are made on the CloudSim simulator. A predictor class is added in CloudSim. It offers the predicting algorithm designer a unified interface to implement their algorithm on CloudSim. The main API is listed in Table I. The algorithm designer can inherit the abstract class *Predictor* and implement the necessary methods.

Besides, some modifications are made in class *VM*, *powerhost*, *VmScheduler* and *CloudletScheduler*. The dynamic resource allocation can be implemented through these changes. The work flow diagram of improved Cloudsim is shown in Figure 2. First of all, the predictor will predict the needed resource and modify the resource allocation of virtual machine. And at the same time, the utilization model class, a simulated resource utilization generator, will generate a simulated resource utilization of the task (cloudlet). Then, the simulated virtual machine will process the task with the allocated resources. At last, the simulated resource utilization which is from utilization generator will send to the predictor. This information will help to do the next-time predicting. After these modifications, our experiment can be simulated on this cloud simulator more appropriate.

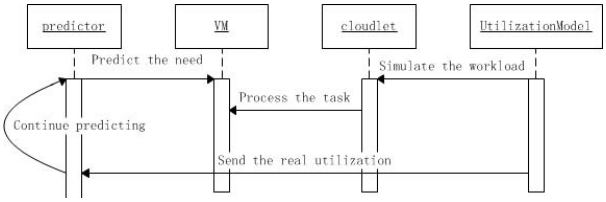

Figure 2. The Work Flow Diagram of Improved CloudSim.

B. The Experiment Design

Because the industrial web service utilization log is difficult to get, so we use the minute's utilization log of our university's BBS [1] in one day as the test data. And simulate these two algorithms with modified CloudSim cloud simulator. In the experiment, we consider the server of the BBS is a virtual machine in one host of a datacenter. Its utilization of CPU is from the real utilization log file of the BBS server. Then we make predicted value to compare with the real data and check the violation rate and power consuming.

In the CloudSim simulator, we create one datacenter, one cloud broker, one host, one virtual machine (VM) and one cloudlet which is the task in the VM. And the resource utilization of the cloudlet is from the utilization log of real server. The result is outputted to the console and GUI dash board.

C. Result and Analysis

The first experiment is about the pure MQMPM. The predicted value and the real value are presented in Figure 3. Seeing the result carefully, we can find there is a delay in the predicted sequence compared to the real utilization sequence. And in some continuous increasing periods, there exists some predict faults. With MQMPM, the fault rate is 2.06%. The Figure 3 can be compared with the results of improved algorithms.

The second experiment is about the combined algorithm of LTPM and MQMPM. To be clearer, we only use 200 continuous points in this experiment. The result of the combined algorithm is in the Figure 5. And compared with the result of pure MQMPM in Figure 4 which is the magnified view of Figure 3 during these 200 points, this method reduces some violations caused by the delayed utilization trend. We can see this improvement from the Figure 5 clearly. For example, during the continuous increasing sequence between time 114 to 120, the response and predicting accuracy are much better than pure MQMPM. In this 200-continuous-points period, the violation rate of pure MQMPM is 6.5%; and the improved method is 4.0%. So, the improvement of the pure MQMPM is obvious.

The third experiment is the improvement on reducing the flat sequence predicted value. Table II shows the violation rate and dynamic power-consuming of MQMPM and the second improved algorithm. The predicted sequence and the real sequence are represented in Figure 6. We can compare the Figure 3 and Figure 6 together, the effect is dramatic. From time 137 to 141 and from time 190 to 418, the resource reservation is reduced to a relatively low level. Here, we use cubic model to measure the energy-consuming of these two methods. The improve method can reduce the dynamic power of the CPU of virtual machine about 10.98% and the violation rate is the same, around 2.06%.

[1]CC98 BBS at Zhejiang University (Intranet Only): http://www.cc98.org/

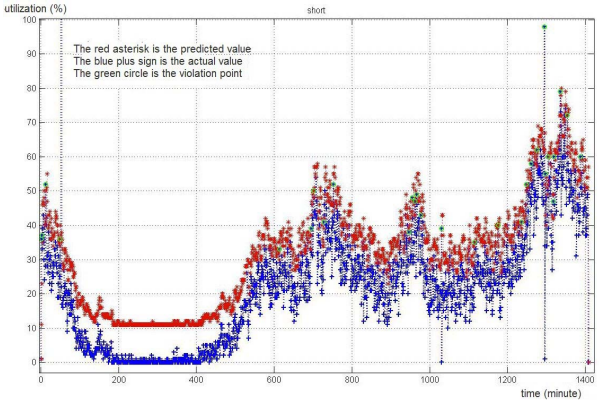

Figure 3. The Result of Pure MQMPM.

Table II
THE COMPARE OF MQMPM AND ITS IMPROVED METHOD.

	MQMPM	**Improved Method**
Violation Rate	2.06%	2.06%
Energy-consuming	9.229α[1]$* 10^7$	$8.216\alpha * 10^7$

[1] α is a power factor.

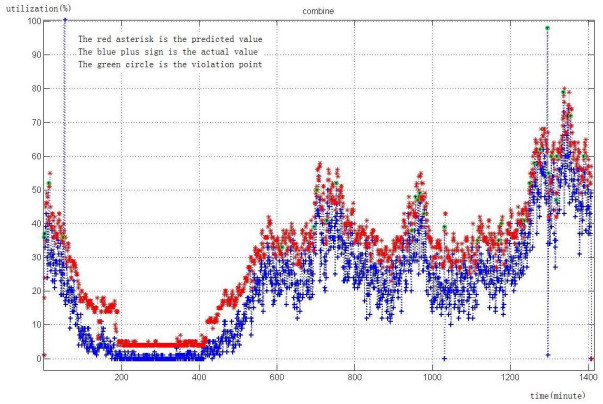

Figure 6. The Result of Reducing the Flat Sequence Predicted Reservation Improvement.

V. CONCLUSION

The MQMPM is a good method to predict the resource utilization and make the reservation. But it does still have

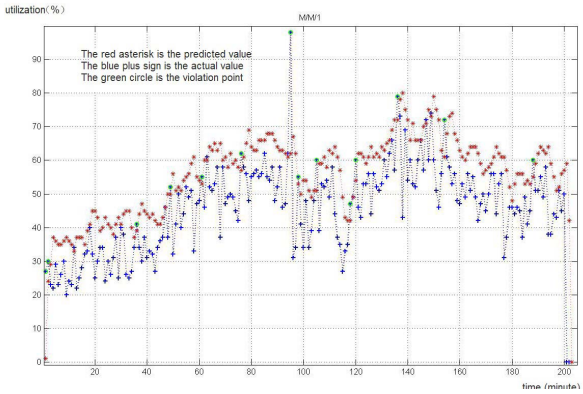

Figure 4. The Result of Pure MQMPM With the Test Set in Experiment II.

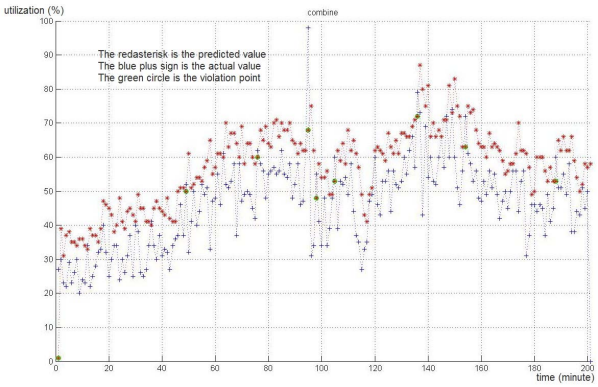

Figure 5. The Result of Combined Algorithm of LTPM and MQMPM.

improving points. The combination method of The Linear Trend Predicting Method and MQMPM can make better response in continuous increasing period. The Flat Period Reservation-Reducing Method in MQMPM can reduce the unnecessary high resource reservation in flat and smooth period. The experiment result is obviously good.

And in the future, we can implement the algorithms in openNebula, an open source cloud manager, and Xen, a virtual machine manager, and test it in real cloud environment. Otherwise, more advanced and efficient machine learning methods and idea can be used to find the useful information in the utilization log. Then, this information can be used to help predict future resource utilization. The predicted method with machine learning in server utilization log is really a promising area. We can make more appropriate resource allocation and reservation.

ACKNOWLEDGMENT

This work is funded by the National 973 Basic Research Program of China under grant NO.2007CB310900. The author Kejiang Ye is supported by the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, "Virtual machine based energy-efficient data center architecture for cloud computing: A performance perspective," in *Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications*, 2010, pp. 171–178.

[3] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live migration of multiple virtual machines with resource reservation in cloud computing environments," in *Proceedings of the 2010 IEEE International Conference on Cloud Computing*, 2011, pp. 267–274.

[4] K. Ye, X. Jiang, S. Chen, D. Huang, and B. Wang, "Analyzing and modeling the performance in xen-based virtual cluster environment," in *12th IEEE International Conference on High Performance Computing and Communications*, 2010, pp. 273–280.

[5] K. Ye, X. Jiang, Q. He, X. Li, and J. Chen, "Evaluate the performance and scalability of image deployment in virtual data center," *Network and Parallel Computing, LNCS 6289*, pp. 390–401, 2010.

[6] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in *Proceedings of the 2008 IEEE International Conference on Cluster Computing*, 2008, pp. 99–106.

[7] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive, memory compression," in *Proceedings of the 2009 IEEE International Conference on Cluster Computing*, 2009, pp. 1–10.

[8] R. Calheiros, R. Ranjan, A. Beloglazov, C. De 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, pp. 23–50, 2011.

[9] K. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services," in *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, 2009.

[10] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power cmos digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.

[11] P. Purdue, "The m/m/1 queue in a markovian environment," *Operations Research*, vol. 22, no. 3, pp. 562–569, 1974.

[12] C. Lin, "The performence estimation in computer network and system," *Tsinghua University Press*, pp. 25–64, 2001.