LETTER *Special Section on Cloud and Services Computing*

Dynamic Consolidation of Virtual Machines in Cloud Datacenters*

Han-Peng JIANG[†], **Ming-Lung WENG**[†], ***Nonmembers***, ***and*** **Wei-Mei CHEN**[†a)], ***Member***

SUMMARY Now that the subject of green computing is receiving a lot of attention, the energy consumption of datacenters has emerged as a significant issue. Consolidation of Virtual Machines (VMs) reduces the energy consumption since VM live migration not only optimizes VM placement, but also switches idle nodes to sleep mode. However, VM migration may negatively impact the performance of the system and lead to violations in SLA (Service Level Agreement) requirements between end users and cloud providers. In this study, we propose a VM consolidation mechanism that reduces the energy consumption of datacenters, eliminates unnecessary migrations, and minimizes the SLA violations. Compared to previous studies, the proposed policy shows a reduction of 2% to 3% in energy consumption, 13% to 41% in VM migration frequency, and 15% to 50% in SLA violations.

key words: *cloud computing, virtual machine, consolidation, SLA, datacenter management*

1. Introduction

With the rapid growth of cloud computing, many cloud service providers have emerged to offer several convenient services. Google Drive and DropBox provide users with universal access to data and documents. Google App Engine and Windows Azure have released application programming interfaces (APIs), which programmers can use to access cloud resources. Amazon EC2 provides flexible computing capacity and makes web-scale computing easier for developers. In addition, no matter what type of service cloud providers supply, they have to guarantee that the quality of service that cloud users obtain is as good as they paid for.

A critical benchmark of the quality of service for cloud-based providers, such as the CPU capability and the amount of memory space, is specified in the form of a Service Level Agreement (SLA) between providers and customers. Providers have to offer compensation when they cannot meet SLA requirements. In terms of cost considerations, cloud providers minimize datacenter costs (including electrical usage and hardware resources) under the constraint of guaranteeing that performance meets SLA requirements.

Virtualization technology integrates heterogeneous physical machines into one virtualized resource pool and creates virtual machines which service cloud users. VM consolidation is a technique that aggregates workloads and switches idle nodes to low-power mode, thereby reducing the overall energy consumption of the datacenter [2], [4], [5]. Using live migration, the datacenter manager can easily balance the load among different nodes, without stopping the VMs' processes. However, VM migration may negatively impact the performance of the system and lead to violations in SLA.

Owing to the fact that the resource demands of virtual machines (VMs) change during execution, the reduction of energy consumption is now one of the main issues in cloud datacenters. It is important that the datacenter manager manages VMs with an ability to consider future loading trends. Troung et al. [4] measured the power consumption of a server running in different states, and proposed a green scheduler for energy savings in cloud computing. Optimization of the allocation of VM placements can reduce the number of active servers, resulting in greater energy savings [6], [7]. In this regard, Beloglazov et al. [2] proposed several heuristics for the dynamic consolidation of VMs based on historical data of VM resource usage.

The goal of this study is to provide an IaaS (Infrastructure as a Service) system provider which can reduce the energy consumption of datacenters while meeting SLA requirements. This study proposes an autonomic consolidation technique that can control the average CPU capability to reduce the VM migration frequency and guarantee SLA requirements. The remainder of this paper is organized as follows: Sect. 2 presents the system model we used in this study, Sect. 3 presents details of the mechanism that we propose, and Sect. 4 evaluates the proposed mechanism by employing simulations using CloudSim. Conclusions will be provided in Sect. 5.

2. System Model

A cloud provider typically consists of a collection of servers offering virtualized resources and a datacenter manager which uses these resources to create VMs to process user tasks. The datacenter manager creates and allocates VMs to computing nodes, and balances the workload between nodes to guarantee SLA requirements.

To standardize the quality of service, cloud providers and consumers sign an agreement termed the SLA to define the level of service being sold. In terms of cloud computing resources, the SLA requirement refers to the CPU capability of the VM in this study. In our model, the CPU capability

Manuscript received October 22, 2013.

[†]The authors are with the Department of Electronic Engineering and Computer Engineering, National Taiwan University of Science and Technology, Taiwan.

*This work is partially supported by National Science Council under the Grant NSC 101-2221-E-011-039-MY2.

a) E-mail: wmchen@mail.ntust.edu.tw (Corresponding author)

DOI: 10.1587/transinf.E97.D.1727

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers

of the server is represented in units of some Million Instructions per Second (MIPS).

2.1 Power Model

Recent studies have shown that the power consumption of a server which applies the DVFS technique has an almost linear relationship with CPU utilization. The DVFS technique allows the CPU to switch the frequency and voltage levels to low-performance levels when the workload is low, and dynamically adjust them to high-performance levels when it is high. Therefore, the power consumption of the server at a fraction u of CPU utilization, $P(u)$, is defined as follows:

$$P(u) = (P_{\max} - P_{\text{idle}})u + P_{\text{idle}},$$

where $P_{\max}$ and P_{idle} are the power consumption at maximum and idle CPU utilization, respectively. CPU utilization of the server changes with time, and the relationship between utilization and time is denoted by $u(t)$. The energy consumption of a server over a period of time is defined as an integral of the power consumption over that period, as follows [1]:

$$E_i = \int_{t_0}^{t_1} P_i(u_i(t))dt.$$

We also considered the energy consumption of the switching server when it is turned ON and OFF. The switching energy consumption of server i is defined as

$$e_i = S_i(T_i^{\text{ON}} P_i^{\text{ON}}) + T_i^{\text{OFF}} P_i^{\text{OFF}}),$$

where T_i^{ON} and T_i^{OFF} are the duration of time that server i turns ON and OFF, P_i^{ON} and P_i^{OFF} are the power consumption of server i when it turns ON and OFF, and S_i is the number of the switching instances for server i. Then the energy consumption of the datacenter of N servers is computed by $E = \sum_{i=1}^{N}(E_i + e_i)$.

For the cost of migration, since migration negatively impacts the performance of the VM and hurts the SLA, the migration time T_m is estimated by the ratio of the amount used memory space and the network bandwidth.

2.2 SLA Violation Metrics

Generally, end users can not estimate the computing capability they need accurately. For the safe reason, the amount of reserved capability usually exceeds that of consumption. In this study, we assume that users always request computing resources under the amount specified in the SLA requirements.

A cloud provider has responsibilities to ensure that users are offered QoS in the SLA. Thus we consider the metric for the SLA violation in terms of the rate of violation of a resource provision. A violation is said to occur when a server cannot offer enough resources to a VM. The amount of unfinished workload is the difference between the VM resource demand and the amount of resources that the server provides. The SLA violation rate of a datacenter with N servers over a period of time $[t_0, t_1]$ is calculated as follows:

$$V_{\text{SLA}} = \frac{\int_{t_0}^{t_1} \sum_{i=1}^{N}(r_i(t) - a_i(t))dt}{\int_{t_0}^{t_1} \sum_{i=1}^{N} r_i(t)dt}$$

where $r_i(t)$ is the resource demand of VM_i at time t, $a_i(t)$ is the resource provided at time t.

3. Prediction-Based VM Allocation Mechanism

In this section, we present the structure of the datacenter management system that we employed. When a VM enter the cloud system in the first time, the initialization process of the system will allocate resource to the VM with the amount specified in the SLA. Figure 1 represents the process of the datacenter management system which includes three main parts: the predictor, monitor, and allocator (in Fig. 2).

(1) VM Predictor

Workloads of applications that are processed on the VM vary with time, and uncertain resource requirements result in the difficulty of load balancing. However, the behavior and resource demand of an application may be similar over a period of time. The predictor collects historical resource usage data and outputs an estimated value. The management system allocates the VM with resources under the estimated value. The estimated value is defined as

$$U_P = \frac{\sum_{i=1}^{R} \text{Record}_i}{R}(1 + \sigma)$$

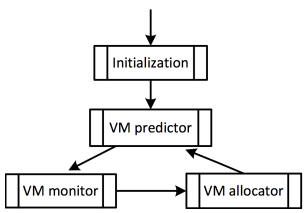

Fig. 1 The process of the datacenter management system.

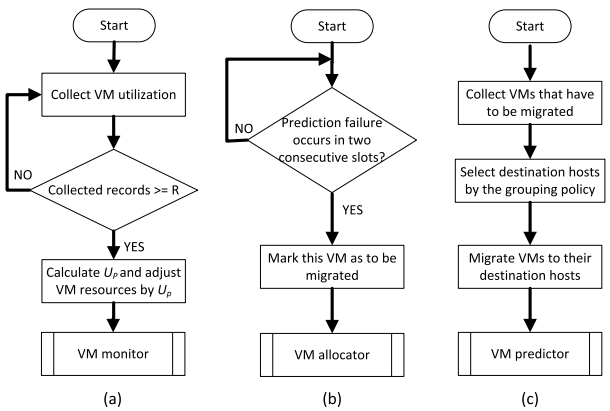

Fig. 2 Three main parts of the datacenter management system: (a) VM predictor, (b) VM monitor, and (c) VM allocator.

where Record$_i$ is the historical usage, R is the number of usage records, and σ is the standard deviation of the historical usage. In fact, we reserve more resources for U_P with a safe factor $1 + \sigma$ of previous usages. With the estimated value, the system restricts VM resource usage to under U_P. For a VM, its workload is completed only when its demand is lower than the estimated value. Unfinished workloads appear when the demand of the VM exceeds the estimated value. In this situation, the server did not offer resources demanded during some previous periods and it has to catch up the progress later under the limitation of U_P when it has more computing capability or VM's demand is less than U_P.

(2) VM Monitor

When the management system limits VM resources whereas the VM's demand exceeds the estimated value, the system should relocate and move the VM to a new node to honor the SLA. On the contrary, when the VM demand is much lower than the estimated value, the system should lower the resources earmarked for that VM in order to load other VMs onto that node or move this VM onto other node. A monitor is used to detect the unexpected situations, such as overloading or underloading. Instead of moving or readjusting a VM immediately, the system leaves room for the possibility for the VM to return to the expected situation. To detect unloading of a VM, we introduce a tolerance factor τ, $0 < \tau < 1$. The underloaded state occurs if $U/U_P < \tau$ and the overloaded state occurs if $U > U_P$, where U is the actual usage. Furthermore, the prediction failure is reported when one of the above two states happens. In our system, if the prediction failure occurs in two consecutive time periods, the monitor informs the predictor to re-calculate U_P and passes the related information to the VM allocator.

(3) VM allocator

The VM allocation policy consists of two parts: (1) selecting VM based on records by VM monitor and (2) deciding target hosts for migration. Instead of searching for overloaded nodes, we focused on locating VMs which are in the prediction failure state and need to be migrated. In this study, we suggest a grouping policy to speed up the process of selecting the VM migration destination. The policy groups servers into two status types: *active* and *sleep*. The policy will search the active group first, and if there are no nodes in the active group that satisfy the VM, it will then go on to search the sleep group. Besides, the minimum growth in server power consumption can be easily found by locating the server which has the minimum value of ρ, where ρ is the ratio of the computing capability to the power consumption of a server.

4. Performance Evaluation

We compared our mechanism with the IQR-MMT policy of VM placement [2], which is a very efficient consolidation management. Compared to the pure DVFS policy which does not migrate VMs, IQR-MMT can reduce datacenter energy consumption by 45.6% on energy consumption of the datacenter with an SLA violation rate of only 0.10%.

Table 1 Power consumption of server growth with CPU utilization.

Utilization (%)	00	10	20	30	40	50	60	70	80	90	100
Power (W)	75	78	84	89	94	100	105	109	112	115	117

To evaluate the efficiency of the proposed mechanism, we used two different sets of workloads for VMs: one consisted of workloads with different durations for the VM, and the other comprised workloads with different behaviors for the VM. The duration of a VM is the time between it starting and finishing its jobs. The behavior of a VM is the behavior of its resource demand requests. To simplify experimental results, we only created a single-core VM in simulations. For the parameters, because there is a tradeoff between the decreasing number of SLA violations and increasing datacenter energy consumption, we set the number of records (R) as 10 and the tolerance factor (τ) at 0.85 in our experiments.

We employed the CloudSim simulator [3] to simulate a large-scale datacenter and evaluate the proposed resource allocation algorithms. CloudSim 3.0 is used to simulate a datacenter consisting of 500 heterogeneous IBM x3200 M2 servers. At most, 1000 VMs ran concurrently. Every server had a dual-core Xeon 3.0GHz CPU, 4GB of memory, and a network bandwidth of 1Gbit/s. The relationship between server power consumption and CPU utilization is presented in Table 1. Every VM consists of a 1GHz CPU, the same capability as an Amazon EC2 small instance type, and 50MB of memory, the minimum size in a KVM VM.

We used CPU utilization records, a monitoring infrastructure for PlanetLab [2], [8] and provided by the CoMon project. The utilization records represent the CPU usage of the server for one day, and the recording interval was 5 minutes. We use a Poisson distribution to simulate a user entering the cloud system and accessing VMs [9]. When a user enters and requests a VM, the system will create a VM for the user, randomly choose a record, and assign it to the VM. A VM requests CPU computing resources according to the assigned workload which varies with time and the workload of the server also varies with time.

We divided workloads into two classes according to their behavior: stable and unstable types. An unstable type indicates that the demand for resources changes violently and frequently. A stable type is the opposite of this case. We classified the workload according to the rule: a workload for which the amount of variation in usage is under 15% and over 75% of the duration is classified as a stable type. If the variation is otherwise, it is classified as an unstable type. Figure 3 presents an example for stable workloads. In this case, the usage difference between two consecutive time slots, $u(t + 1) - u(t)$, is smaller than 15% and this situation occurs over 75% of its executive duration.

Table 2 presents results for different workload types. The proposed mechanism shows a large improvement when the workload is unstable. This is because the proposed pol-

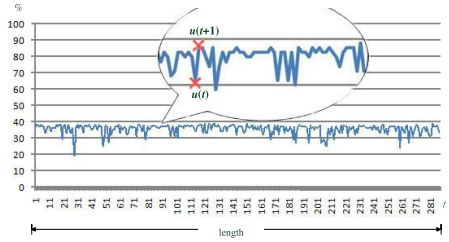

Fig. 3 A stable workload.

Table 2 Comparisons with different workload types.

	IQR-MMT		New	
	Unstable	Stable	Unstable	Stable
Number of migrations	18108	9608	10554	8265
Overall SLA violation (%)	0.122	0.070	0.101	0.035
Energy consumption (kWh)	65.288	66.251	63.617	64.982
Number of instructions completed (millions)	9951.33	10703.22	10184.04	11105.69

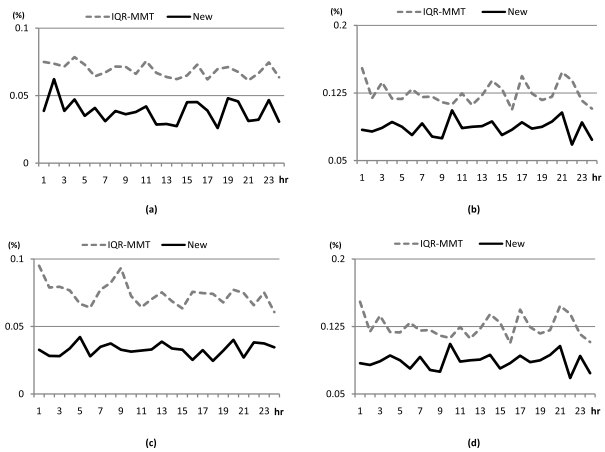

Fig. 4 SLA violations over 24 hours: (a) for all long type workloads, (b) for all short type workloads, (c) for all stable type workloads and (d) for all unstable type workloads.

icy does not migrate VMs frequently while the IQR-MMT policy immediately moves a VM when the server is overloaded. Meanwhile, our mechanism reduces datacenter energy consumption and completes more instructions than the IQR-MMT policy for both workload types.

We also classified VM workloads into two duration types: short and long. The duration of short workloads takes 1 to 6 hours to finish the job, and for long workloads, 6 to 24 hours. The amount of SLA violations for different workloads over a 24-hour period is shown in Fig. 4. We have that the SLA violation for our mechanism is always better than that of IQR-MMT.

5. Conclusion

Cloud providers improve datacenter performance using virtualization techniques to implement VM consolidation and switch idle servers to energy-saving mode and reduce energy consumption. In this study, we propose a VM consolidation mechanism that reduces the energy consumption of datacenters, eliminates unnecessary migrations, and minimizes the SLA violations. We used CloudSim Toolkit 3.0 as a simulation platform and evaluated the proposed policy using workloads traced from PlanetLab VMs. Compared to the IQR-MMT policy [2], the proposed policy shows a reduction of 2% to 3% in energy consumption, 13% to 41% in VM migration frequency, and 15% to 50% in SLA violations.

References

[1] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing," Future Generation Computer Systems, vol.28, no.5, pp.755–768, 2011.

[2] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, vol.24, no.3, pp.1397–1420, 2012.

[3] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software-Practice & Experience, vol.41, no.1, pp.23–50, 2011.

[4] T.V.T. Duy, Y. Sato, and Y. Inoguchi, "A prediction-based green scheduler for datacenters in clouds," IEICE Trans. Inf. & Syst., vol.E94-D, no.9, pp.1731–1741, Sept. 2011.

[5] G. Katsaros, J. Subirats, J.O. Fitó, J. Guitart, P. Gilet, and D. Espling, "A service framework for energy-aware monitoring and VM management in clouds," Future Generation Computer Systems, Available online 20 Dec. 2012.

[6] Y.C. Lee and A.Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," J. Supercomputing, vol.60, no.2, pp.268–280, 2012.

[7] C. Li and L. Li, "Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and SaaS provider in cloud environment," J. Supercomputing, vol.65, no.2, pp.866–885, Feb. 2013.

[8] K.S. Park and V.S. Pai, "CoMon: A mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Operating Systems Review, vol.40, no.1, pp.65–74, 2006.

[9] H. Yu, D. Zheng, B.Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," Proc. 1st ACM SIGOPS/EuroSys European Conference on Computer Systems, pp.333–344, 2006.