International Journal of Grid and Distributed Computing
Vol. 9, No. 11 (2016), pp.323-336
http://dx.doi.org/10.14257/ijgdc.2016.9.11.27

Power and QoS Aware Multi-level Resource Coordination and Scheduling in Virtualized Servers

Congfeng Jiang[1, 2], Jingling Mao[1,2], Dongyang Ou[1,2], Yumei Wang[1,2], Xindong You[1,2], Jilin Zhang[1,2], Jian Wan[1,2]

[1]*Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China*
[2]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
cjiang@hdu.edu.cn, {maojingling20, oudongyang1, wyumei1}gmail.com, {youxindong, jilin.zhang,wanjian}@hdu.edu.cn

Abstract

Modern cloud data centers are virtualized for resource multiplexing and services consolidations. Virtual machines (VMs) residing in the same server cluster share the same hardware resources and power supply while they may have different QoS requirements for their services and applications. Moreover, the power consumption of the server cluster is highly dynamic since different virtual machines have different workloads due to different services requests. Cluster level power and QoS coordination is crucial for data center level energy efficiency coordination as well as high quality service provisioning. In this paper we propose the power and QoS aware multi-level resource coordination and scheduling in virtualized servers, i.e., the cluster-level power control layer, the VMs resource allocation layer, and the QoS optimization layer. This three-level controlling framework schedules and allocates hardware resource for QoS guarantee and cluster level power management. We use dynamic frequency scaling for QoS mitigating when power budget changes. The experiment results show that the proposed multi-level coordinated control architecture consumes 5.36% and 6.96% less power for web servers and computing intensive virtual machines, respectively while it can guarantee the response time of web server and execution time of computing tasks no more than those without the proposed controlling approach.

Keywords*: Resource scheduling, Virtual machines, Power aware scheduling, QoS*

1. Introduction

In modern cloud data centers, the high density server racks increase the power density and thus pose urgent need for power capping as well as dynamic power control in both rack level and data center level. Most of these cloud data centers are typically operated by energy from fossil fuels and other dirty energy. In these data centers, the hardware resources are usually in low or middle utilization level and cause a huge waste of energy. Such low hardware utilization is primarily due to the randomness of the tasks arrived at the data center, in other words, the number of the tasks per unit time is uneven in the data center. Therefore, virtualization is used for resource multiplexing and services consolidations to reduce power and energy consumption. To further reduce the carbon footprints, renewable energy and liquid cooled systems are also introduced into modern data centers [1, 2]. However, since many virtual machines (VMs) share the same hardware, interference among various VMs results in not only resource contending but also conflicts in Quality of Service (QoS) guarantee for different VMs. Consequently,

ISSN: 2005-4262 IJGDC

coordination between power and QoS requirement is crucial for not only energy reduction but also high quality service provisioning in data center level.

Traditionally, power and QoS are treated independently in two different dimensions in servers or clusters. System-level control strategy designed for the QoS is aimed to meet the requirement of the application QoS while reducing power consumption as much as possible. On the other hand, the control strategy designed for power management treats power as the first control objective while maximizing the QoS in a best-effort manner. In virtualized environment, it's more challenging to coordinate between power and QoS requirements due to the interference among various VMs sharing the same hardware components. Therefore traditional single strategy cannot provide guarantee for control of both QoS and cluster level power consumption because the single strategy is lack of good portability on virtualized cluster system.

Various hardware related approaches have been proposed to increase data center energy efficiency at different levels [3, 4], including circuits and chips [5, 6, 7], memory [8, 9], disk [10, 11], and network traffic routing [12, 13]. In modern data centers server virtualization and consolidation are actively deployed for power and energy savings. Usually over-commitment or over-subscription approaches are used to further reduce energy related costs more aggressively. At the same time, control theory has been successfully applied to the cluster power saving of enterprise servers. Lefurgy *et al*. [14] have studied and shown that control theory is superior to the commonly used heuristic method because it has more accurate power control and better control performance. Wang and Chen [15] proposed a MIMO control algorithm for the cluster power control. Their work is different from ours in that they can only control power consumption but cannot provide QoS guarantees.

Sharma *et al*. [16] applied control theory to control the QoS of the service request. Chen *et al*. [17] proposed a feedback controller to manage the response time of server cluster. Wang *et al*. [18] proposed to control response time for the virtualized server. Kephart *et al*. [19] proposed a coordinating strategy to achieve the trade-off control of power and performance in non-virtualized single server. Although they have used control theory to manage system performance and reduce power consumption, they do not provide guarantee of power protection in case of power budget changes.

Control theory has been used to manage QoS indicators of virtualized servers. However, all the above studies are not designed for virtualized servers saving power. Wang *et al*. [20, 21] proposed a cluster control architecture (Co-Con) for coordinating independent power and performance control loop and a two-layer control architecture (PARTIC) based on control theory. They only analyze the cluster power and performance from two tiers while our three-level controlling framework schedules and allocates hardware resource for QoS guarantee and cluster level power management. In our previous work[22], we use feedback control to achieve power aware job Scheduling with QoS guarantees. In response to these shortcomings of existing single control strategy of power or QoS, we propose the power and QoS aware multi-level resource coordination and scheduling in virtualized servers, i.e., the cluster-level power control layer, the VMs resource allocation layer, and the QoS optimization layer. This three-level controlling framework schedules and allocates hardware resource for QoS guarantee and cluster level power management. Our work has three main contributions:

Firstly, we propose the framework of power and QoS aware multi-level resource coordination and scheduling in virtualized environment using control theory. The multi-level control framework consists of three layers: the cluster-level power control layer, the VMs resource allocation layer, and the QoS optimization layer. In our framework, the cluster-level power control layer is the primary control layer, the VMs resource allocation layer is the secondary control layer, and the QoS optimization layer is the third control layer from bottom to top logically. This three-level controlling framework schedules and

allocates hardware resource for QoS guarantee and cluster level power management. We use dynamic frequency scaling for QoS mitigating when power budget changes.

Secondly, we design the control model hierarchically by traversing each control level and their coordination. Our multi-level control architecture considers the impact of each level and improves the control performance while it's able to achieve better flexibility and scalability. Therefore, our architecture has better portability for a typical virtualized cluster system.

Thirdly, we analyze the relationship between each control layer in the multi-level control system, the mutual influence variables (such as CPU frequency, CPU time slice allocation, *etc*.) involved in different levels, the relationship between control periods of each control layer, and their coordination. We also analyze the effectiveness of the proposed control framework in experiments, including the request response time for real-time tasks, execution time for intensive tasks, and the total power consumption of the cluster. The experiment results show that the proposed multi-level coordinated control architecture consumes 5.36% and 6.96% less power for web servers and computing intensive virtual machines, respectively while it can guarantee the response time of web server and execution time of computing tasks no more than those without the proposed controlling approach. The results show that our multi-level coordinated control architecture and all control layers achieve desired control performance on cluster-level power savings and QoS guarantee for different tasks request in virtualized servers.

The remainder of the paper is organized as follows. In Section 2, we propose the framework of power and QoS aware multi-level resource coordination and scheduling in virtualized environment. In Section 3, we present the experiment results and analysis. In Section 4, we conclude the paper and give some directions for future work.

2. Multi-level Control Architecture

We design power and QoS aware multi-level control architecture and the power and QoS aware resource coordination and scheduling based on the proposed multi-level control architecture as shown in Figure 1.

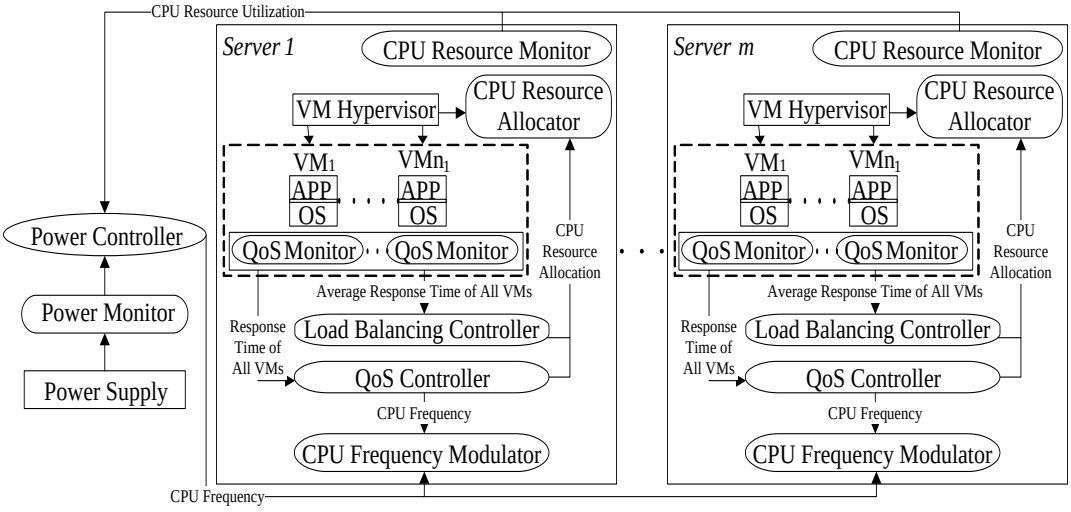

Figure 1. Power and QoS Aware Multi-level Control Architecture

Our power and QoS aware multi-level control framework mainly consists of three layers: the cluster-level power control layer, the VMs resource allocation layer, and the QoS optimization layer. This three-level controlling framework schedules and allocates hardware resource for QoS guarantee and cluster level power management. We use dynamic frequency scaling for QoS mitigating when power budget changes especially in

virtualized server clusters where the cluster workload fluctuates significantly. We describe the details in the following subsections.

2.1. Cluster Level Power Control Layer

The cluster level power control layer shown in Figure 2 is the primary control layer of our multi-level control architecture. The cluster level power control layer is mainly to control the cluster power consumption. In the cluster level power control layer, we set cluster power as the control target. First, we need to consider the limited conditions of the cluster power control. Obviously, the total power of a cluster must be less than the maximum available power of the cluster. Moreover, according to the feedback response time of each server, we split the cluster power into each server on its demand. We use CPU frequency scaling in each server to achieve power re-distribution, so that the cluster power can be used fully and effectively and we may achieve an efficient power saving. In order to reduce power consumption of the server cluster, we can either improve the performance of a component in the cluster or reduce the power of a part of the cluster. Based on the feedback of the CPU resource usage from the VMs resource allocation control layer and the QoS optimization control layer in each server, we adjust the CPU frequency of each server on demand through dynamic voltage and frequency scaling (DVFS) to dynamically control the overall power of all servers in the cluster.

The main design principle of the cluster level power control layer is as following: In the cluster power control procedure, it provides an interface through power controller. According to the feedback response time and CPU utilization of each server obtained from the last control period, we provide the CPU utilization of each server in last control period to power controller as the power weight value of each server allocated by the controller. Intuitionally, the server whose CPU utilization is higher than the average can be allocated more power, otherwise, the server will be allocated less power by adjusting the CPU frequency of each server through DVFS. So that the cluster power controller can control the overall power of all servers in a cluster dynamically. With the limitation of hardware power and CPU frequency, it should be a reasonable allocation strategy for the power resource in the cluster, so the cluster power can be fully and effectively used and it can guarantee the QoS and save power effectively as well.

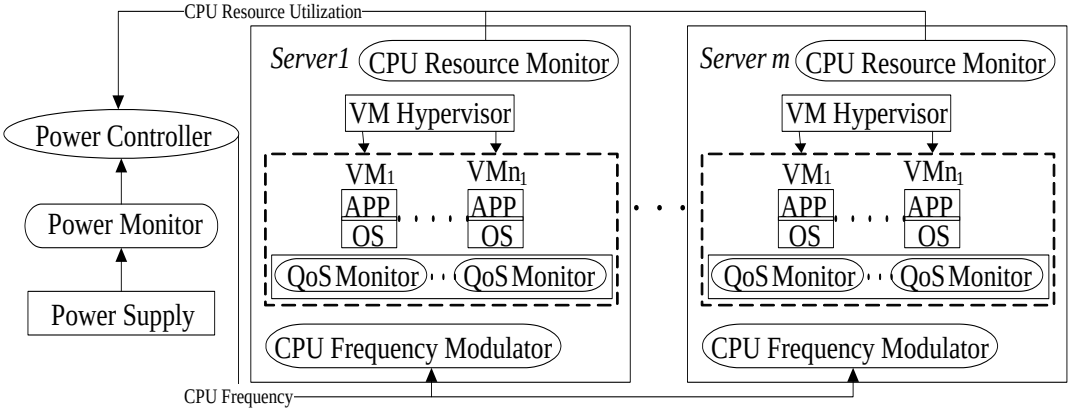

Figure 2. Cluster Power Control Architecture

In our framework, we consider the CPU power consumption as the main power consumption in the cluster assuming that the power consumption of other components is constant. In our control loop, the controlled variable is *the power* and the manipulated variable is the *CPU frequency*. Meanwhile, in the entire cluster level control system, the power control layer will be called periodically. The control period is determined by

weighing response time, DVFS overhead time, the settling time before control system reaches a steady level, and other factors.

2.2. VMs Resource Allocation Layer

The VMs resource allocation layer is to control the VMs resource allocation in each host. It achieves the load balancing among VMs via adjusting the CPU time slice allocation which is assigned to all of the VMs on a server, so that the VMs on the same server can roughly have the same relative response time. The relative response time of the VM is the ratio of the actual response time over maximum allowed response time of each VM, namely

$$rres_{VM}(k,i,j) = res_{VM}(k,i,j) / maxRes_{VM}(i,j) \quad (1)$$

where $rres_{VM}(k,i,j)$ is the relative response time of the *j*th VM on sever *i* in the *k*th control period; $res_{VM}(k,i,j)$ is the average actual response time of the *j*th VM on sever *i* in the *k*th control period; $maxRes_{VM}(i,j)$ is the maximum allowed response time of the *j*th VM on sever *i*.

For the convenience of description, we describe the VMs CPU allocation control of one server as an example to introduce the VMs load balancing control layer of the control architecture in Figure 3.

The VMs resource allocation control layer shown in Figure 3 is the secondary control layer of the multi-level control architecture. The main design principle of the VMs load balancing control layer is as following: In the process of the VM resource allocation, it provides an interface through VM load balancing controller. According to the feedback response time of each VM obtained from the last control period and the load balancing principle, it assigns the CPU time slice to each VM on demand, which is to assign different CPU allocation weights to different VMs on the server by CPU resource allocator. So that the VM load balancing controller can control the load of all VMs on the server dynamically to maintain the load balancing among all VMs on the same server. With the limitation of the CPU frequency and its power, it's a reasonable allocation strategy for CPU resource allocation in the server. So the VMs load across different servers and the QoS can be adjusted while the cluster power is being fully and effectively used. The load balancing control layer runs periodically like the power control layer. The control period of the control layer is determined by concurrent application requests in the configuring period. According to the actual situation, the control parameters and the order of system control input and output will be changed for the different workloads. Due to the different experimental environment, control order and control parameters of the system model need to be measured again. So we redesign the system model determining algorithm to avoid the uncertainty of the order of system control input and output, and present the new method to calculate the system control parameters. For the optimized system model, the effect parameters are determined by the actual value of the last control period dynamically.

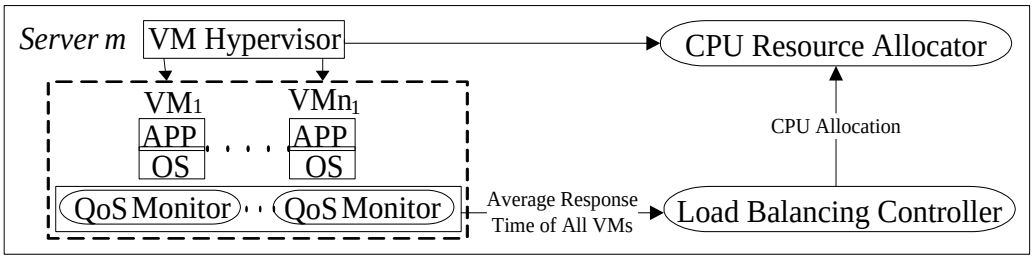

Figure 3. VMs Resource Allocation Control Architecture

2.3 QoS Optimization Layer

The QoS optimization layer is mainly to provide QoS guarantee. It is the third control layer of the multi-level control architecture based on the QoS control of each VM. We set the response time for real-time tasks (or execution time for intensive tasks) as the indicator of QoS in the proposed multi-level control architecture. The QoS optimization layer controls the QoS by adjusting CPU resource allocation (CPU time slice) assigned to VMs, so that each VM can achieves the desired QoS (response time for real-time tasks or execution time for computing intensive tasks) requirements. Thus, fully and rationally use CPU resources on a physical server also can improve the efficacy of the server while the desired response time of all VMs will be satisfied. We design the QoS controller for each VM on each server. For the convenience of description, we describe the QoS optimization control of one VM as an example to introduce the QoS control layer in Figure 4.

The main design principle of the VM-level QoS control layer shown in Figure 4 is as following: In the procedure of the QoS control, it provides an interface through QoS controller. According to the feedback response time of each VM obtained from the last control period, it assigns the CPU resource to each VM on demand, which is to assign different CPU allocation weights to different VMs on the server so that the QoS controller can dynamically control the CPU resource allocation of the server. The main control components in QoS control layer comprise server-level QoS controller, CPU resource allocator, CPU frequency modulator, and VM-level QoS monitor which is primarily responsible for monitoring application response time. The controlled variable (the average response time of all web requests) and the manipulated variable (the desired CPU resource allocation) of the QoS control layer interact to form a control loop. Meanwhile, in the entire control system, the QoS (i.e., response time) control layer on each VM will be run periodically like other control layer.

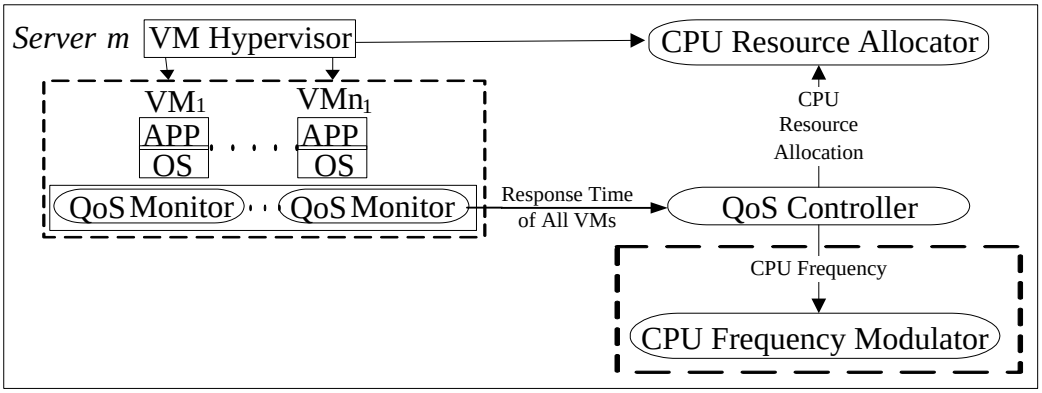

Figure 4. QoS Control Architecture

2.4. Multi-level Control Coordination

For the above proposed power and QoS aware resource coordination and scheduling multi-level control architecture, there is an interdependent relationship among all control layers.

The secondary VMs load balancing control layer and the third QoS control layer interact with each other. We need to coordinate these two control layers. First, the VMs resource allocation control layer tries to minimize the difference between the response time of each single VM and the average response time of all VMs, while the average response time is controlled by QoS control layer and it converges to the desired average response time of all VMs. Second, VMs resource allocation control layer and QoS control layer both have the corresponding scheduling and allocation control on CPU time slice of servers. The QoS controller will change the server's CPU frequency by CPU frequency modulator based on the feedback response time of QoS control layer in each control period, which is to change the total amount of available CPU resource on the server. The change of the total CPU resource will affect the CPU resource allocation on VMs resource allocation control layer. Therefore, in order to minimize the effect of QoS control layer on VMs resource allocation control layer, the control period of QoS control layer should be longer than the settling period of VMs resource allocation control layer. That is, in order to ensure that all VMs in QoS control period T_{VM} have the same response time, the control period of QoS controller should be longer than the settling control period of VMs resource allocation controller. Therefore it can ensure that the VMs resource allocation control layer can reach to a stable state in the QoS control period.

When analyzing the interaction among cluster power control layer and the other two control layers, VMs resource allocation control layer and QoS control layer are regarded as a whole performance control layer. The CPU frequency controlled by the power controller will have a direct effect on application performance of all VMs on the server. At the same time, the CPU resource allocation controlled by the performance control layer may also affect system power consumption. Obviously, if we cannot effectively coordinate the power control layer and performance control layer, these two control layers may have conflict. In order to achieve the desired control function and the stability of system, the settling period of the primary control layer (power control layer) should be longer than the secondary control layer (performance control layer). Thus in a control period of the primary control layer, the secondary control layer can maintain its stable state, which can decouple between the two control layers. So each control layer can be designed independently.

3. Performance Evaluation

3.1 Experimental Setup

The testbed in this paper includes a server cluster and a client. There are two physical servers in the cluster named as *Server1* and *Server2*. All physical hosts run with CentOS 6.2 (the kernel version is Linux 3.5.7 with Xen 4.2.1). Each server is configured with AMD Opteron 6272 (2.1GHz, 16-cores, 16MB L3 cache), 1600MHz 64GB DDR3 memory, 300GB SAS hard drive, and Intel Gigabit Ethernet. Each host connects with each other through an Ethernet switch. The client is used as load generator to send a large number of HTTP requests to *Server1* and *Server2*, and receives the power data from power analyzer.

We configure 16 VMs on *Server1* and 25 VMs ($VM_{1,1}, VM_{1,2}, ..., VM_{1,25}$) to *Server2*. We configure LAMP(Linux, Apache httpd web server, MySQL database, and PHP web pages) software stack on each VM. We run an Apache web server as a virtual server on each VM and the Apache server answers the HTTP requests with a dynamic Web page written in

PHP. The PHP file is running a task of inserting 1 million records to the database. The client load generator is the benchmarking tool (ab) of Apache HTTP server. We setup multiple ab instances on the client and each instance will send web requests to the Apache server inside the VMs. We define *Server1* as the IO-intensive web server. At the same time, we run CPU compute-intensive tasks on *Server2*. Our experiment runs a prime number calculation program on *Server2* which calculates the prime number from 100000 to 1000000.

3.2 Control Verification for Real-time Tasks

For the assessment of the control capability of the control architecture for real-time tasks (web requests), we compare the monitoring results of the following four test environments:

Test environment I: the general cluster without control architecture;

Test environment II: only cluster power control architecture;

Test environment III: add QoS control architecture based on the test environment II;

Test environment IV: add VMs load balancing on the basis of test environment III, which is to construct the complete multi-level control architecture on the cluster.

According to above test environments and the data collection, we get the results in Figure 5 to Figure 7. Figure 5(a) shows that when the server desired power need to be reduced from 240W to 220W at 500s, and to be increased from 220W to 240W at 1000s, the single power control layer can control on reducing cluster power effectively by reducing server CPU frequency. Figure 5(b) indicates that in the case that the desired cluster power needs to be reduced, if we only have the power control layer, it cannot effectively control the response time of each VM converge to the desired response time , i.e., Server 1 is 700ms and server 2 is 800ms).

In Figure 6, compared with no control architecture environment, the two-layer control architecture consists of power control layer and QoS control layer can significantly control the average response time of the server in the desired range. Compared with no control architecture environment, the complete multi-level control architecture proposed in this paper can effectively control the web request response time within the desired range when the workload of the cluster changes. Moreover, comparing the control layer which is lack of VMs resource allocation control and the complete multi-level control architecture, it shows that VMs resource allocation control layer has little effect on the control of the complete control architecture on the web response time when the concurrency level of web requests has changed.

In Figure 7, compared with no control architecture environment, when cluster workload changes, the complete multi-level control architecture proposed in this paper can reduce the cluster power (include just two servers' power) by 5.36% through controlling resource scheduling. The complete multi-level control architecture is better than the control architecture which is lack of VMs load balancing control layer to control the total power saving in the cluster. Therefore, the VMs load balancing control layer in multi-level control architecture has a great impact on the control of the total power saving in cluster.

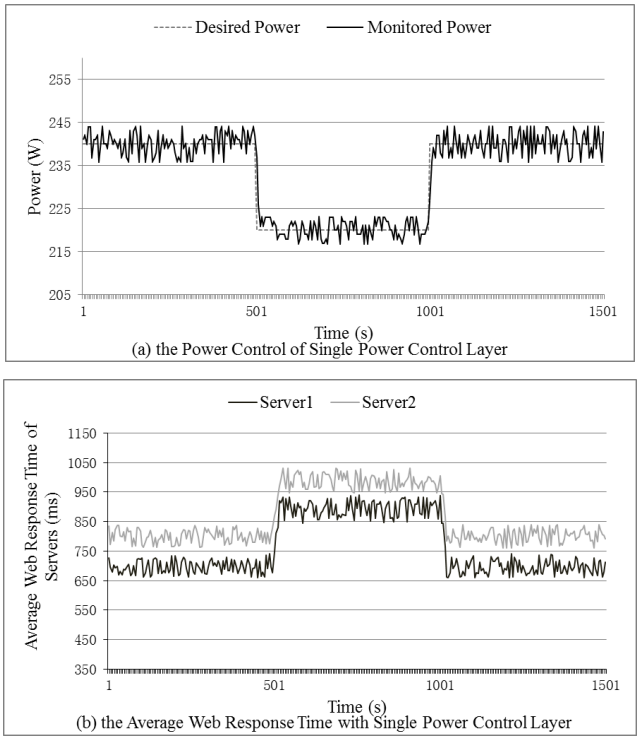

Figure 5. The Power Control Capability Assessment of Single Power Control Layer

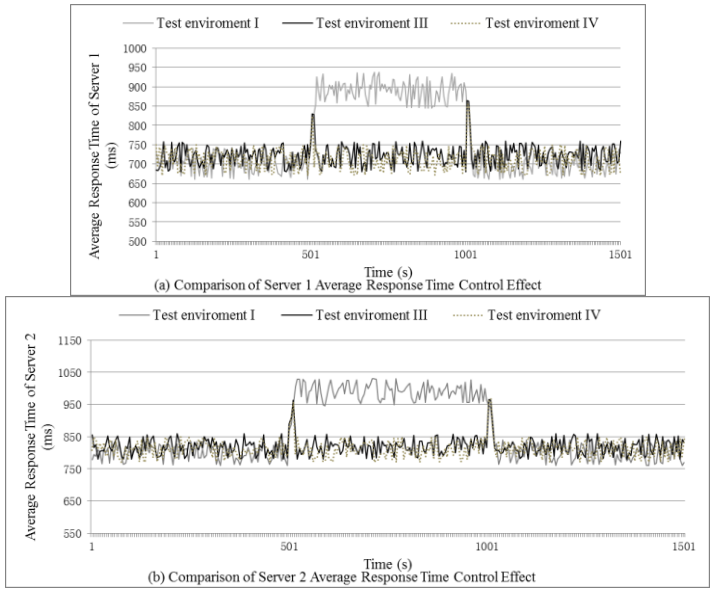

Figure 6. Comparison of Average Response Time Control

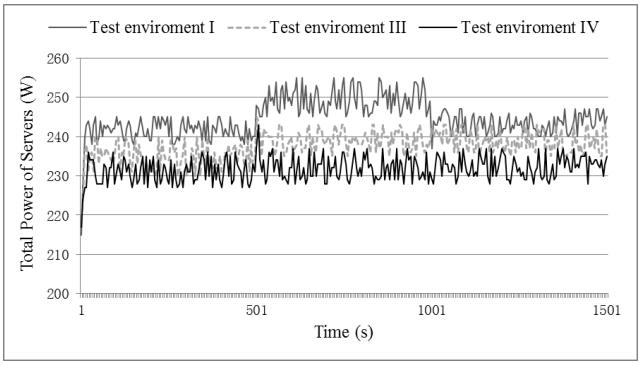

Figure 7. Comparison of Cluster Level Power Control Effectiveness

3.3 Control Verification for Computing Intensive Tasks

For the assessment of the control capability of our framework for computing intensive tasks, we compare the monitoring results of the following four test environments:

Test environment I: cluster without control architecture;

Test environment II: cluster level power control and QoS optimization control on the cluster;

Test environment III: add VMs resource allocation control based on test environment II, which is to construct the complete multi-level control architecture on the cluster.

In addition to the test environment configuration in section 3.1, we also have the following configurations: on server 1, $VM_{2,1}, VM_{2,2}, \ldots\ldots, VM_{2,4}$ are running a PHP program, $VM_{2,5}, VM_{2,6}, \ldots\ldots, VM_{2,8}$ are running two PHP programs, $VM_{2,9}, VM_{2,10}, \ldots\ldots, VM_{2,12}$ are running three PHP programs, and $VM_{2,13}, VM_{2,14}, \ldots\ldots, VM_{2,16}$ are running four PHP programs; on server 2, $VM_{1,1}, VM_{1,2}, \ldots\ldots, VM_{1,5}$ are running a prime calculation program, $VM_{1,6}, VM_{1,7}, \ldots\ldots, VM_{1,10}$ are running two prime calculation programs, $VM_{1,11}, VM_{1,12}, \ldots\ldots, VM_{1,15}$ are running three prime calculation programs, $VM_{1,16}, VM_{1,17}, \ldots\ldots, VM_{1,20}$ are running four prime calculation programs, and $VM_{1,21}, VM_{1,22}, \ldots\ldots, VM_{1,25}$ are running five prime calculation programs.

The results are shown in Figure 8.

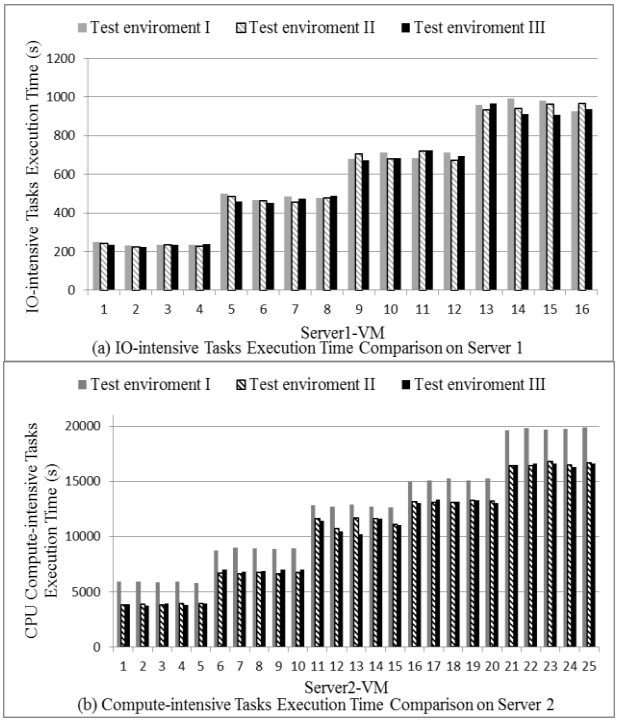

Figure 8. Intensive Tasks Execution Time Comparison

Figure 8(a) shows that, for IO intensive tasks, because the multi-level control architecture in this paper controls QoS and power by controlling CPU frequency and allocating resource, the multi-level control architecture has no significant effect on the execution time control of the IO intensive tasks. In Figure 8(b), we can see that the multi-level control architecture has significant control effect on the execution time control of the CPU computing intensive tasks. Compared with the no control architecture environment, the control architecture which is lack of VMs resource allocation control layer can still effectively reduce application execution time. The execution time is reduced by 16.36%. Combined with the conclusions in Section 3.2, the QoS control layer in this paper can effectively control the execution time of computing-intensive applications to be reduced as much as possible.

Compared with the no control architecture environment, the complete multi-level control architecture proposed in this paper can effectively control the execution time of compute-intensive applications and make it be reduced as much as possible. The execution time is reduced by 17.51%. However, comparing the control architecture which is lack of VMs resource allocation control layer and the complete multi-level control architecture, there is a certain difference among the control effects of these two control architectures on the average execution time. The reducing extent of the average execution time in complete multi-level control architecture is 1.15% higher than the control architecture which is lack of VMs resource allocation control layer. Therefore, the VMs resource allocation control layer of the multi-level control architecture has a certain effect on reducing the execution time of the intensive applications.

The results of the control effect of the multi-level control architecture on the intensive tasks' execution effect are shown in Figure 9. Compared with no control architecture environment, in order to minimize the application execution time and fully utilize the CPU resource to achieve power savings, the multi-level control architecture proposed in this paper can meet the QoS (application execution time)

demand while effectively reducing the total power of all servers through controlling the host CPU utilization.

Comparing the control architecture which is lack of VMs resource allocation control layer and the complete multi-level control architecture, there is a certain difference among the saving effects of these two control architectures on the total power of all servers. The control architecture without VMs load balancing control layer just reduces the cluster power by 6.96%. Obviously, the complete multi-level control architecture is better than the control architecture without VMs load balancing control layer to control the cluster power saving. Thus, the VMs load balancing control layer of the multi-level control architecture has a great impact on the control of cluster power saving. The control architecture with the VMs load balancing control layer can achieve the better saving effect.

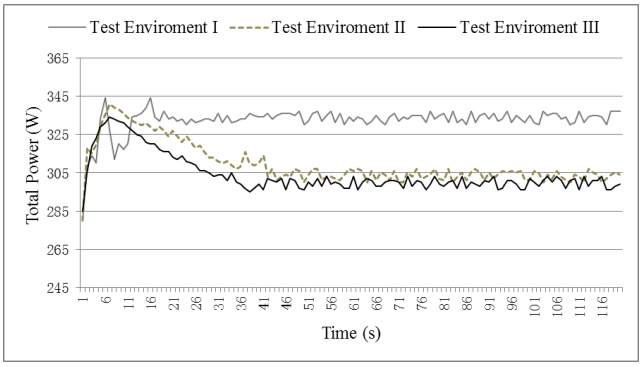

Figure 9. Total Power Consumption

4. Conclusion

In this paper, we design and propose the power and QoS aware multi-level resource coordination and scheduling control architecture for better flexibility and scalability. Furthermore, we design and build testbed, and evaluate the control effect of the proposed multi-level control architecture. The experiment results show that the proposed multi-level coordinated control architecture consumes 5.36% and 6.96% less power for web servers and computing intensive virtual machines, respectively while it can guarantee the response time of web server and execution time of computing tasks no more than those without the proposed controlling approach. The evaluation results of our experiments show that the multi-level coordinated control architecture and all control layers have a good effect on cluster-level power saving and QoS controlling for different tasks request in virtualized servers.

For the primary cluster power control layer, we can expand its control architecture and join other drivers to make the model more presize; for VMs load balancing control layer, we may further consider the optimization of hierarchical VM configuration on the same server.

Acknowledgements

This work is supported by Natural Science Foundation of China (No.61472109, 61003077, 61402140, 61572163), and Natural Science Fund of Zhejiang Province (LY14F020045, LY16F020018, Y1101092).

References

[1] I. Goiri, W. Katsak, K. Le, *et al.* , "Designing and managing datacenters powered by renewable energy", IEEE Micro, vol. 34,no. 3 (**2014**), pp.8 -16.
[2] L. Li, W. Zheng , X. Wang , *et al*., "Data center power minimization with placement optimization of liquid-cooled servers and free air cooling", Sustainable Computing: Informatics and Systems, 2016.
[3] P. Pande, F. Clermidy, D. Puschini, I. Mansouri, P. Bogdan, R. Marculescu, A. Ganguly, "Sustainability through massively integrated computing: are we ready to break the energy efficiency wall for single-chip platforms?", Proceedings of the Design, Automation and Test in Europe, Grenoble,France, (**2011**),Mar 14-18.
[4] Z. Zhang and S. Fu, "Characterizing power and energy usage in cloud computing systems", Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science , Athens, Greece, (**2011**), Nov 29-Dec 1.
[5] S. Deb, K. Chang, X. Yu, *et al*., "Design of an energy efficient CMOS compatible NoC architecture with millimeter wave wireless interconnects", IEEE Transactions on Computers, vol.62, no.12(**2012**),pp.2382-2396.
[6] D. Fan, Z. Tang, H. Huang, Hailin Huang, G. R. Gao, "An energy efficient TLB design methodology", Proceedings of the 2005 international symposium on Low power electronics and design, San Diego, California, USA, (**2005**), Aug 8-10.
[7] R. Rodrigues, A. Annamalai, I. Koren, *et al*., "Reducing energy per instruction via dynamic resource allocation and voltage and frequency adaptation in asymmetric multicores", Proceedings of 2014 IEEE Computer Society Annual Symposium on VLSI, Tampa, FL, USA, (**2014**), July 9-11.
[8] M. Bi, S. Chandrasekharan, and C. Gniady, "IAMEM: interaction-aware memory energy management", Proceedings of 2013 USENIX Annual Technical Conference, San Jose, CA, USA, (**2013**), Jun 26-28.
[9] Q. Deng, D. Meisner, A. Bhattacharjee, *et al*., "CoScale: coordinating CPU and memory system DVFS in server systems", Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture, Vancouver, BC,Canada, (**2012**), Dec.1-5.
[10] D. Tiwari, S. Vazhkudai, Y. Kim, *et al.* , "Reducing data movement cost using energy-efficient active computation on SSD", Proceedings of the The 11th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, (**2013**), Feb.12-15.
[11] Y. Zhang, J. Liu, M. Kandemir, "Software-directed data access scheduling for reducing disk energy consumption", Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), Macau, China, (**2012**), Jun 18-21.
[12] R. Krishnaswamy, V. Nagarajan, K. Pruhs, *et al*., "Cluster before you hallucinate: approximating node-capacitated network design and energy efficient routing", Proceedings of the 46th Annual Symposium on the Theory of Computing , New York, USA, (**2014**), May 31- Jun 3.
[13] Y. Ohsita, and M. Murata, "Optical data center networks; architecture, performance, and energy-efficiency, Handbook on Data Centers, Springer, New York, (**2015**).
[14] C. Lefurgy, X. Wang, M. Ware, "Power capping: a prelude to power shifting", Cluster Computing, vol.11, no.2(**2008**), pp.183-195.
[15] C. Clark, K. Fraser, S. Hand, *et al*. "Live Migration of Virtual Machines", Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, Boston, USA, (**2005**), May 2-4.
[16] V. Sharma, A. Thomas, T. Abdelzaher , *et al*. "Power-aware QoS Management in Web Servers", Proceedings of the 24th IEEE Real-Time Systems Symposium, Cancun, Mexico, (**2003**), Dec.3-5.
[17] Y. Chen , A. Das , W. Qin , *et al*., "Managing Server Energy and Operational Costs in Hosting Centers", Sigmetrics, vol. 33, no.1(**2005**),pp. 303-314.
[18] Y. Wang, X. Wang, M. Chen , *et al*., "Power-Efficient Response Time Guarantees for Virtualized Enterprise Servers", Proceedings of the 2013 IEEE 34th Real-Time Systems Symposium, Barcelona, Spain, (**2008**), Dec.1-3.
[19] J. Kephart, H. Chan , R. Das , *et al*., "Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs", Proceedings of the Fourth International Conference on Autonomic Computing, Jacksonville, Florida, USA, (**2007**), June 11-15.
[20] X. Wang , Y. Wang , "Coordinating power control and performance management for virtualized server clusters", IEEE Transactions on Parallel and Distributed Systems, vol.22,no.2(**2011**), pp.245-259.
[21] Y. Wang, X. Wang, M. Chen, *et al*. PARTIC: Power-Aware Response Time Control for Virtualized Web Servers[J]. IEEE Transactions on Parallel and Distributed Systems, vol. 22, no.2(**2011**), pp.323-336.
[22] C. Jiang, X. Xu, J. Wan, J. Zhang, X. You, R. Yu, "Power Aware Job Scheduling with QoS Guarantees Based on Feedback Control", Proceedings of 18th IEEE International Workshop on Quality of Service(IEEE IWQoS 2010), Beijing, China, (**2010**), June 16-18.

Authors

Congfeng Jiang received his Ph.D. Degree from Huazhong University of Science and Technology, Wuhan, China, in 2007. He is currently an associate professor in School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. He is with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China, and the Grid and Services Computing Lab and the Cloud Technology Research Center. His current research interests include big data processing, information retrieval, document recognition and analysis. He is the founded co-chair of international workshop on high performance data intensive computing (HPDIC) and international workshop on performance aspects of cloud and service virtualization (CloudPerf). He is the guest editor of some international journals, including Elsevier Future Generation Computing Systems, Springer Information System Frontiers, and Springer Cluster Computing.

Jingling Mao is a master student in School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. She is with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China, and the Grid and Services Computing Lab and the Cloud Technology Research Center. Her current research interest is power aware computing systems.

Dongyang Ou is now a master student in School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. He is with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China, and the Grid and Services Computing Lab and the Cloud Technology Research Center. His current research interest is parallel processing of big data.

Yumei Wang is a master student in School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. She is with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China, and the Grid and Services Computing Lab and the Cloud Technology Research Center. Her current research interest is energy efficiency optimization in data centers.

Xindong You is an Associate Professor in the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. He received his Ph.D. Degree from Northeast University in 2007. Her research is cloud storage and green computing.

Jilin Zhang received his Ph.D. Degree from the School of Computer Science and Technology, University of Science and Technology Beijing, in 2009. He is currently is an Associate Professor in the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His research is in the areas of big data analytics and cloud computing.

Jian Wan received his Ph.D. Degree in Computer Science from Zhejiang University, China, in 1996. He is now a professor of School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. His research interest includes big data, cloud computing, and services computing.

