Energy Efficient Resource Allocation and Scheduling in Cloud Computing Platform

Shuen-Tai Wang, Ying-Chuan Chen, Yu-Ching Lin

***Abstract*—**There has been renewal of interest in the relation between Green IT and cloud computing in recent years. Cloud computing has to be a highly elastic environment which provides stable services to users. The growing use of cloud computing facilities has caused marked energy consumption, putting negative pressure on electricity cost of computing center or data center. Each year more and more network devices, storages and computers are purchased and put to use, but it is not just the number of computers that is driving energy consumption upward. We could foresee that the power consumption of cloud computing facilities will double, triple, or even more in the next decade. This paper aims at resource allocation and scheduling technologies that are short of or have not well developed yet to reduce energy utilization in cloud computing platform. In particular, our approach relies on recalling services dynamically onto appropriate amount of the machines according to user's requirement and temporarily shutting down the machines after finish in order to conserve energy. We present initial work on integration of resource and power management system that focuses on reducing power consumption such that they suffice for meeting the minimizing quality of service required by the cloud computing platform.

***Keywords*—**Cloud computing, energy utilization, power consumption, resource allocation.

I. Introduction

CLOUD computing [1]-[3] is the access to computers and their functionality through different network modes like public, private and hybrid. It has emerged as a favorable technology with the advent of virtualized resources [4] for satisfying the computationally intensive requirements of users. The cloud computing centers usually employ Virtual Machine (VM) technologies [5], [6] to support dynamic provisioning of computing services for consolidation and environment isolation purposes. Cloud computing also delivers infrastructure, platform, and software as services, which are made available to consumers as subscription-based services under the pay-per-use model, where the term service is used to reflect the fact that they are provided on demand. In industry, these services are referred to: Infrastructure as a Service (IaaS), a provision model in which an organization outsources the equipment used to support associated operations, such as storage, hardware, servers and network components; Platform as a Service (PaaS) new application platform that abstracts infrastructure, and provides users a set of application development frameworks. It also offers a high level integrated environment to test, build and deploy applications; Software as a Service (SaaS) allows users to use application software in the cloud. It is a new cloud computing model in which a provider hosts some specific application programs that users can access, execute and control via the Internet. However, cloud computing centers hosting cloud applications running on VMs or physical machines consume huge amounts of electrical energy. Currently, it is estimated that cloud computing centers consume 0.5% of the world's total electricity usage [7] and if current demand continues, is projected to quadruple by 2020. It has received great interests in how to design a high-performance, low-power computing facility in cloud computing research community. Many [8]-[10] have realized the concept of reducing energy consumption. In [11], the authors gave the idea to minimize energy consumption guaranteeing service level agreements. They developed many models to profile the power consumption based on computation intensive, data intensive and communication intensive. Reference [12] gave the idea to power consumption evaluation. They analyzed the power consumption can be decreased by adopting virtualization technology. They stated that employing the consolidation strategy, power overhead is less as compared to regular deployment. Reference [13] analyzed that the cloud computing resources were provisioned to different VM instances allocated to users for specific period of time. They proposed combinatorial auction based allocation which described that user's request is taken into account while making provisioning decisions and VM allocation.

The most popular methodology involved both resource allocation and scheduling, and it was studied by using the improved techniques and algorithms. Resource allocation and scheduling are the key technology of cloud computing, which utilizes the computing resources in virtualized environment to speed the execution of complicated jobs. Resource allocation needs to consider many factors, such as load balancing and power consumption. And scheduling favorable resource nodes to execute a job in cloud computing must be considered, and they have to be properly selected according to the properties of the user's computing job. In particular, cloud resources need to be allocated not only satisfy quality of service requirements specified by users but also to reduce power consumption [14].

In this paper, a methodology is presented that provides efficient energy enhancements within a scalable cloud computing architecture. Using power-aware scheduling techniques and variable resource management, overall power consumption will be vastly reduced in cloud computing platform with minimal performance overhead.

The rest of this paper is organized as follows. Section II lists the related works. Section III gives some details of the implementation. Section IV shows the experimental results.

S.T. Wang, Y.C. Chen and Y.C. Lin are with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: stwang@nchc.org.tw, ycc0301@nchc.org.tw,1203043@nchc.org.tw).

Section V discusses future work and concludes.

II. Background

This section gives a brief review about the background which mainly considers the energy efficiency of resources in cloud computing.

A. Green Computing

Green Computing [15] is the study and practice of using IT resources more efficient. Today, Green Computing is an effort to raise awareness on the subject of energy usage associated with information technology. The effort is to not only to reduce cost, but also to lower the impact on our environment. The number of activities is mentioned to achieve this goal. Unlike hardware vendors tending to invent lower power devices, here, we induce two issues: Power management and efficient computing, and try to find out what else we can do in the cloud computing platform.

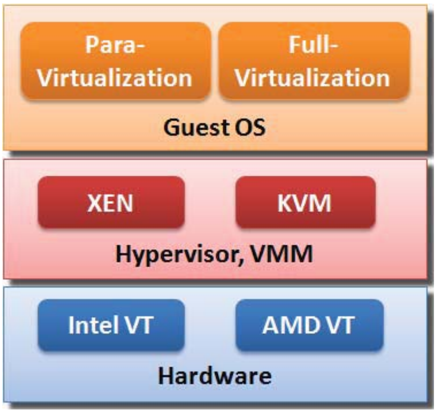

Fig. 1 Virtualization Architecture

B. Virtualization Architecture

Fig. 1 shows the normal virtualization architecture. The hardware layer is the physical resources are divided as many virtual resources by the VMM (Virtual Machine Monitor), and these virtual resources are assigned to each VM by application requirements. Furthermore, the VMM provides each Guest OS a set of virtual interfaces that constructs VMs, and acting as a bridge to connect between physical and virtual devices. Instructions are delivered to hardware layer from virtual device, which receives results from VMM. Each VM will be execute independently, although physical resources are shared. The main different between Hypervisor and VMM is that the monitor can be run above hardware layer directly with better performance than VMM, such as VMware's ESX [16] and Xen [17]. In the Guest OS layer, it includes two virtualization types [18]: Full-virtualization and para-virtualization, which can be both combined with hardware assisted virtualization. Full-virtualization means that all devices are closed to the physical machine and accessed through light weight virtual drivers to offer better performance. Para-virtualization means that the Guest OS is simulated by modified kernel of Linux, but related devices are not emulated.

C. VM

A VM is an operating system or application environment that is installed on software, which imitates dedicated hardware. VM provides its own virtual hardware, including virtual CPU, hard drive, memory, network interface, and other devices. The virtual hardware devices provided by the VM are mapped to real hardware on the physical machine. To implement a VM, software developers design a software layer to physical machine to support the virtualization architecture. By providing one or more efficient virtual system, VM has extended multi-processing system and become multi-environment system as well. There are many kinds of VM now, we choose the KVM (Kernel-based Virtual Machine) [19] to build our virtualization platform. KVM is an open source software with GPL, developing by Qumranet. KVM provide full-virtualization solution for Linux on x86 hardware containing virtualization extensions with AMD-V or Intel-VT, and KVM was encased in mainline Linux OS kernel over version 2.6.20. We can run multiple VMs running unmodified Linux or Windows images by using KVM. Each VM has private virtualized hardware devices, and it was like a process in queuing system of Linux, so user can directly kill the VM by control command.

D. IPMI

IPMI stands for Intelligent Platform Management Interface. [20] The IPMI is a common interface for the administration and management of servers. It is implemented by a BMC (Baseboard Management Controller) of a motherboard. It can be operated independently of the OS and allows users to control a system remotely even the monitored machine is powered off but connected to a power source. Many high performance computing leading companies have wrapped this protocol into their commercial cluster packages, and integrate remote console, such as IBM Cluster System Management and Integrated Lights-Out on HP rack-mount servers. With this enhancement, cluster administrators can know the detailed hardware health more easily. The most interesting feature is that IPMI operates independently of the Linux distribution. Administrators can manage compute nodes remotely even in the absence of management utility as long as Ethernet and power cables connected. IPMI can also work when OS begins its operation and provide other advanced features when worked with IPMI management tool.

III. Implementation

Fig. 2 is the system architecture of our implementation. The core of the implementation is Cloud Management Wizard, which wrapped the major components are: Resource Allocator, Job Scheduler, Power Saving Trigger, and Remote Power Controller. These components are as follows.

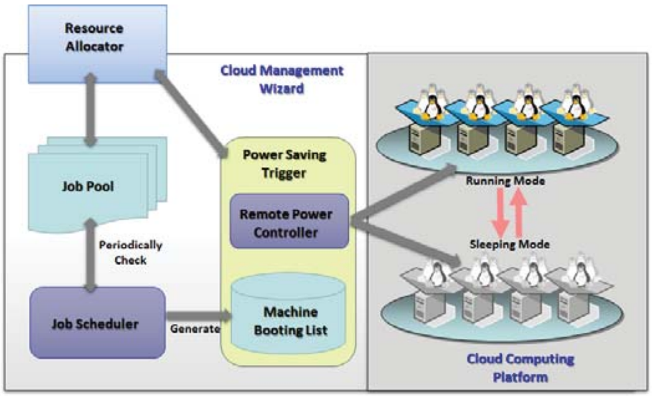

Fig. 2 System Architecture

A. Resource Allocator

For resource management of cloud infrastructure, we developed a resource allocator providing control over virtualization requests from user to guarantee the fairness of using the physical machines, priority escalading, and resource allocation. In virtualized environment, the optimization of current allocation of VMs is carried out in two steps: At the first step, we select the physical machine and the VMs on such physical machine that need to be migrated, at the second step chosen those VMs are placed on the right physical machines using optimal algorithm from job scheduler. On the other hand, in order to avoid unexpected failure during booting or execution procedure, arranging enough spare computing machines is necessary. This resource manager can offer an environment variable for administrator to specify how many physical machines should be woken up in case of booting failure. Once compute machines started booting process, the client daemon of resource allocator will become active when OS finished its booting process. When the total number of CPU is enough for virtualization request, then VMs will start automatically. When all users' virtualization requests finished and job pool is empty, then the resource allocator will start doing its function again. The selected machines will receive commands and enter sleeping mode at once.

B. Job Scheduler

Once the resource containers are given to the user, the application makes a scheduling decision. In many cases, the virtualization request consists of multiple jobs to which the allocated resources are given. We developed a Job Scheduler to schedule the actual jobs. The scheduler is responsible for assigning preferred resources to a particular job so that the overall computing resources are utilized effectively. It also has to make sure each job is given adequate amount of resources, or its fair share.

In our work, the scheduling decision of job scheduler is based on optimal Round-Robin algorithm, the allocation policies are:

1) Round-Robin by Racks: This policy allocates VMs in a round-robin manner across racks.
2) Round-Robin by Machines: This policy allocates VMs in a round-robin manner across physical machines with a preference for selecting all available machines in a rack.
3) Round-Robin by Hybrid: This policy combines Racks and Machines in a round-robin manner.

The scenario of scheduling is: when users submit their job into the Job Pool, it will be queued as 'W' state at first. Scheduler will calculate the how many physical processors that job needs for finding the physical machine satisfying the given constraints by Round-Robin by Hybrid policy. All information will be recorded in Machine Booting List. This list acts as platform load history provider for Power Saving Trigger. According to load history information, trigger can smartly decide which kind of Sleeping Mode the physical machine should enter. If recent load were lower than average, cold shout-down command will be used for the physical machine.

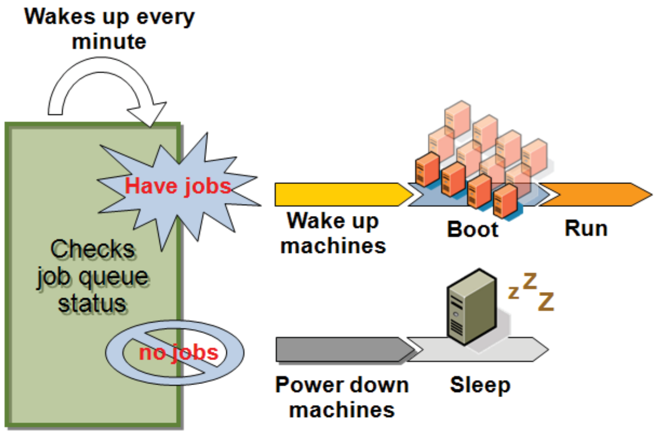

Fig. 3 Power Saving Trigger Workflow

C. Power Saving Trigger

The Power Saving Trigger provides commands for suspend, hibernate and a hybrid suspend using the kernel's native suspend/resume functionality from remotely. There are also mechanisms to add some power-related events to customize pre- and post-suspend actions. As shown in Fig. 3, users can submit their jobs to job pool anytime, even if the computing resource is not available at the time the job is submitted. The submitted jobs are queued and then wait for the computing resource to become available. When the machines become available, the resource allocator will fetch the applicable jobs, parses the requirements, and remotely powers on the correct number of physical machines by IPMI protocol. After the job completes, the Trigger will power the physical machines down. So this module currently relies on checking the job pool and then decides to shut down which machines when no new job is submitted. By powering off the idle physical machines, it can significantly save more energy than always keeping all machines running.

D. Remote Power Controller

In order to reduce the power consumption during booting physical machine, it was adopted the entering sleep mode instead of cutting off power supply. Because hibernate or suspend the machine could offer great energy saving since no power is consumed in this state. Moreover, we have to modify distributive shell by wrapping IPMI protocol to Suspend-to-Disk/Ram for better remote control in this development. There are three methods in Remote Power Controller to control from running mode to sleeping mode. They are e-suspend, e-hibernate and e-shutdown commands. Table I shows the comparison. These power-off methods can fit the Linux based cloud computing platform regardless of the hardware variations.

TABLE I
POWER-OFF STRATEGIES IN POWER SAVING TRIGGER

Command Name	e-shutdown	e-suspend	e-hibernate
Wrapped Mechanism	Power Off via IPMI	Suspend via DSH	Hibernate via DSH
Response time	Quick	Medium	Slow
Need motherboard support	Yes	Yes	Yes
Need power source connected	Yes	Yes	No
Need additional IP address	Yes	No	No
Data & System Integrity	Low	Medium	High

IV. RESULT ANALYSIS

This section describes the result analysis, including the experimental setup, energy evaluation on real facilities, and energy consumption analysis.

A. *Experimental Setup*

Table II shows the specification information of the experimental platform named Formosa 3. Formosa 3 [21] is a 64 bits high-performance Beowulf cluster located within Southern Business Unit of the National Center for High Performance Computing (NCHC) [22]. It consists of 76 IBM X3550 M3 servers as its compute nodes. This self-made cluster was designed and constructed by the 'HPC Cluster Group' at NCHC for cloud service. Each node has two Intel Xeon x5660 2.8 GHz processors and 96 GB of DDR3 SDRAM. All nodes were connected on InfiniBand high speed network and a private subnet with 1000 Mbits/s Gigabit Ethernet.

TABLE II
FORMOSA 3 CLOUD CLUSTER SPECIFICATION

CPU	Intel Xeon x5660 six cores 2.8GHz
Hard Disk	80GB SSD
Memory	48GB DDR3 Registered ECCSDRAM
Network	4x QDR 40Gb Infiniband and Gigabit Ethernet
Operating System	CentOS 6.3
Hypervisor	Kernel-based VM

To determine the total power used by Formosa 3, we refer to the manufacturer’s specification. Theoretically, total power consumption produced by this platform is as Table III.

TABLE III
DEVICE WATTAGE RATINGS FOR FORMOSA 3 CONFIGURATION

	Total Device	Watts per Device	Total Watts
IBM X3550M3	80	411	32880
ARA Stor A316i	4	1050	4200
Nortel BayStack 5510 Gigabit Switch	6	135	810
Voltaire ISR 9288 InfiniBand Switch	1	2340	2340

The maximum wattage for the active devices is estimated at 40,230W = 40.23KW. From its inception, the Formosa 3 has embarked on open source movement.

B. *Energy Evaluation*

The power for server racks in Formosa 3 are offered by PDU (Power Distribution Unit) in machine room. We setup the electricity monitor on each rack to measure the ampere value per hour. Table IV shows the platform arrangement and daily energy consumption measured in Formosa 3. We divided three groups for four racks: the rack 1 and rack 2 contained pure compute machines, so energy intake of these two racks was higher than the rest. All network switches were set in the rack 3, and this rack consumes least electricity in rating. The Rack 4, holding both compute and storage nodes, consumes medium electricity. Average ampere multiplied by 220 voltage will get the average watt per hour of those racks. To emulate the power consumption, we developed a recorder that monitors actual VMs and investigates different workload scenarios.

TABLE IV
AVERAGE WATTS PER DAY

	Staged Hardware	Quantity	Average Ampere	Average Watts
Rack 1	Physical Machine	30	33.56	742.34
Rack 2	Physical Machine	30	31.35	684.16
Rack 3	IB Switch	1	10.33	225.97
	Gigabit Switch	6		
	Login Node	4		
Rack 4	Storage Node	4	22.44	464.37
	Physical Machine	12		

C. *Power Consumption Analysis*

There are three cases arranged for power consumption evaluation. They are: (1). Power consumption of physical machines were always powered on, (2). Power consumption of physical machines were adopted e-shutdown command, (3). Power consumption of physical machines were adopted e-hibernate command. Case 1 represents the regular cloud computing platform that all its machines are always keeping ready. Case 2 ~ 3 are special scenes while adding power management function to the platform. From the log data of one month, 563 virtualization requests completed during the experiment time. By running different job workloads through three scenarios, we show that our work (i.e. Case 2 and 3) can have a large positive impact on power conservation. Table V summarizes power consumption of three cases.

TABLE V
AVERAGE ENERGY USAGE (IN WATTS)

	no shutdown	e-shutdown	e-hibernate
Rack 1	2678.2	2241.6	2445.7
Rack 2	2335.3	1994.6	2134.6
Rack 3	586.7	563.3	581.2
Rack 4	1129.4	704.5	811.9
Total	6729.6	5504	5973.4
	Case 1	Case 2	Case 3
Saved	0	**1225.6**	**756.2**

V. Conclusion

Cloud computing is becoming more matured over the last few years, and green computing is the future development trend and main research object. Reducing power consumption is an increasingly important issue in cloud computing, more specifically when dealing with a large-scale cloud in a production-run platform. In this paper, we propose an improved resource allocation and scheduling methodology that are short of or have not well developed yet to reduce energy utilization. The experimental results show that our approach has immense potential as it offers significant improvement.

While current approach helps us to reduce the energy waste, it needs to be refined to provide more elastic features. Our implementation currently relies on checking job pool and then decides to shut down the related physical machines only when no new jobs were submitted. A more accurate checking mechanism should be based on user job individually. Once user job finished, those job-finished compute machines should keep waiting for a specific period of time, and then resource allocator marked them as off-line state, and finally being powered off. How to avoid new jobs submitted to those machines which happened to do shutdown procedure is tricky. The ability to make those changes will require advanced modification of scheduling service. And design other scheduling algorithms will be also our future work.

References

[1] Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." 2008 Grid Computing Environments Workshop. IEEE, 2008.
[2] Armbrust, Michael, et al. "Above the clouds: A Berkeley view of cloud computing." (2009).
[3] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications 1.1 (2010): 7-18.
[4] Goth, Greg. "Virtualization: Old technology offers huge new potential." IEEE Distributed Systems Online 8.2 (2007): 3.
[5] Meyer, Richard A., and Love H. Seawright. "A virtual machine time-sharing system." IBM Systems Journal 9.3 (1970): 199-218.
[6] Goldberg, Robert P. "Architecture of virtual machines." Proceedings of the workshop on virtual computer systems. ACM, 1973.
[7] W. Forrest, "How to cut data centre carbon emissions?" Available: http://www.computerweekly.com/Articles/2008/12/05/233748/how-to-cut-data-centre-carbon-emissions.htm, Sep. 2016.
[8] Stemm, Mark. "Measuring and reducing energy consumption of network interfaces in hand-held devices." IEICE transactions on Communications 80.8 (1997): 1125-1131.
[9] Younge, Andrew J., et al. "Efficient resource management for cloud computing environments." Green Computing Conference, 2010 International. IEEE, 2010.
[10] Tripathi, Tanvi, and Sunita Gond. "An Efficient Approach of Power Consumption in Cloud using Scheduling of Resources." International Journal of Modern Education and Computer Science 7.7 (2015): 60.
[11] DuBois, Paul. MySQL: the definitive guide to using, programming, and administering MySQL 4. Sams, 2003.
[12] Johnson, Bruce, and Joel Webber. Google web toolkit. Addison-Wesley, 2007.
[13] Hickson, Ian, and David Hyatt. "Html5." W3C Working Draft WD-html5-20110525, May (2011).
[14] Buyya, Rajkumar, Anton Beloglazov, and Jemal Abawajy. "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges." arXiv preprint arXiv:1006.0308 (2010).
[15] Liu, Liang, et al. "GreenCloud: a new architecture for green data center." Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session. ACM, 2009.
[16] VMware virtualization, Available at: http://www.vmware.com/, Sep. 2016.
[17] Xen hypervisor, Available at: http://www.xen.org/, Sep. 2016.
[18] Chen, Wei, et al. "A novel hardware assisted full virtualization technique." Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for. IEEE, 2008.
[19] Kivity, Avi, et al. "KVM: The Linux virtual machine monitor." Proceedings of the Linux symposium. Vol. 1. 2007.
[20] IPMI, Intelligent Platform Management Interface, Available at: http://www.intel.com/design/servers/ipmi/, Sep. 2016.
[21] NCHC Formosa 3 Cloud Cluster, Available at: http://formosa3.nchc.org.tw/, Sep. 2016.
[22] NCHC, National Center for High-performance Computing, Available at: http://www.nchc.org.tw, Sep. 2016.