GreenCloud: A New Architecture for Green Data Center

Liang Liu[1], Hao Wang[1], Xue Liu[2], Xing Jin[1], WenBo He[3], QingBo Wang[1], Ying Chen[1]

IBM China Research Laboratory[1], McGill University[2], University of New Mexico[3]

{liuliang, wanghcrl}@cn.ibm.com[1], {xueliu}@cs.mcgill.ca[2], {wenbohe}@cs.unm.edu[3]

ABSTRACT

Nowadays, power consumption of data centers has huge impacts on environments. Researchers are seeking to find effective solutions to make data centers reduce power consumption while keep the desired quality of service or service level objectives. Virtual Machine (VM) technology has been widely applied in data center environments due to its seminal features, including reliability, flexibility, and the ease of management. We present the *GreenCloud* architecture, which aims to reduce data center power consumption, while guarantee the performance from users' perspective. *GreenCloud* architecture enables comprehensive online-monitoring, live virtual machine migration, and VM placement optimization. To verify the efficiency and effectiveness of the proposed architecture, we take an online real-time game, Tremulous, as a VM application. Evaluation results show that we can save up to 27% of the energy when applying *GreenCloud* architecture.

Categories and Subject Descriptors

C.3 SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS

General Terms

Management

Keywords

Green Cloud Computing, Virtualization, Power Saving

1. INTRODUCTION

Recently, *cloud computing* [2] has attracted considerable attention. Cloud computing is believed to become one of the most important future computing and service paradigm. As stated in [3], a *cloud* is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers. By this means, customers will be able to access applications and data from a "cloud" anywhere all over the world on demand. In other words, the Cloud appears to be a single point of access for all the computing needs of consumers.

Though cloud computing technology is not mature for the mass market yet, service providers are actively seeking to develop cloud computing platforms for consumers and enterprises to facilitate the on-demand access regardless of time and location. For example, Amazon Elastic Compute Cloud (EC2) [20] provides a virtualized computing environment which hosts different kinds of Linux-based service. Another example is Microsoft Live Mesh [21], which provides a centralized storage for applications and data, so users could access all the information through a Web-based Live Desktop or his own devices with Live Mesh software installed.

Internet Data Center (IDC) is a common form to host cloud computing. An IDC usually deploys hundreds or thousands of blade servers, densely packed to maximize the space utilization. Running services in consolidated servers in IDCs provides customers an alternative to running their software or operating their computer services in-house. The major benefits of IDCs include the usage of economies of scale to amortize the cost of ownership and the cost of system maintenance over a large number of machines. With the rapid growth of IDCs in both quantity and scale, the energy consumed by IDCs, directly related to the number of hosted servers and their workload, has been skyrocketed [8]. A recent Internet Data Center report estimated the worldwide cost on enterprise power consumption exceeds $30 billion in year 2008 and likely to even surpass spending on new server hardware. The rated power consumptions of servers have increased by 10 times over the past ten years [1]. This surging demand calls for the urgent need of designing and deployment of energy-efficient Internet data centers.

Many efforts have been made to improve the energy efficiency of IDC [9-18], including network power management [26], Chip-Multiprocessing (CMP) energy efficiency [10], IDC power capping [16], storage power management solutions [11] etc. Among all these approaches, Virtual Machine (VM) technology begins to emerge as a focus of research and deployment. Virtual Machine (VM) technology (such as Xen[28], VMWare[32], Microsoft Virtual Servers[33], and the new Microsoft Hyper-V technology [34] etc), enables multiple OS environments to co-exist on the same physical computer, in strong isolation from each other. VMs share the conventional hardware in a secure manner with excellent resource management capacity, while each VM is hosting its own operating system and applications. Hence, VM platform can facilitate server-consolidation and co-located hosting facilities [4][5][7].

Virtual machine migration, which is used to transfer a VM across physical computers, has served as a main approach to achieve better energy efficiency of IDCs. This is because in doing so, server consolidation via VM migrations allows more computers to be turned off. Generally, there are two varieties [28]: regular migration and live migration. The former moves a VM from one host to another by pausing the originally used server, copying its memory contents, and then resuming it on the destination. The latter performs the same logical functionality but without the need to pause the server domain for the transition. In general when performing live migrations the domain continues its usual activities and from the user's perspective—the migration should be imperceptible. It shows great potential of using VM and VM migration technology to efficiently manage workload consolidation, and therefore improve the total IDC power efficiency [24].

GreenCloud is an IDC architecture which aims to reduce data center power consumption, while at the same time guarantee the performance from users' perspective, leveraging live virtual machine migration technology. A big challenge for *GreenCloud* is to automatically make the scheduling decision on dynamically migrating/consolidating VMs among physical servers to meet the workload requirements meanwhile saving energy, especially for performance-sensitive (such as response time-sensitive) applications, e.g. online gaming servers. Hence, a real-time VM consolidation is needed. An important aspect of this work is to utilize the live migration feature of Xen to implement our *GreenCloud* architecture, which guarantees the real-time performance requirement as well as saves the total energy consumption of the IDC. In the design of *GreenCloud* architecture, we address several key issues including when to trigger VM migration, and how to select alternative physical machines to achieve optimal VM placement. To verify the effectiveness and efficiency of our approach, we built an exploratory system which monitors comprehensive factors in the data center, and intelligently schedule the workload migration to reduce unnecessary power consumption in the IDC. We take an online real-time gaming service, Tremulous **Error! Reference source not found.**, as a VM application on each VM. So when our system is triggered to balance performance and power, players enjoying the games hardly notice that their game server workloads are being or have been migrated.

The rest of this paper is organized as the follows. We first briefly summarize the state-of–the-art power management solutions and virtualization technologies for IDCs in Section 2. In Section 3, we introduce the overview of our *GreenCloud* architecture. A heuristic algorithm to achieve optimal VM migration is studied in Section 4. We present our prototype implementation of the *GreenCloud* architecture and performance evaluation results in Section 5. Finally, we conclude the paper and point out future research directions in Section 6.

2. RELATED WORK

In this section, we present work most pertinent to the discussion of this paper in the field of power management, virtualization technologies, and Cloud Computing.

2.1 Could Computing

Cloud Computing, which refers to the concept of dynamically provisioning processing time and storage space from a ubiquitous "cloud" of computational resources, allows users to acquire and release the resources on demand and provide access to data from processing elements, while relegating the physical location and exact parameters of the resources. As the user could see, Cloud Computing means scalability on demand, flexibility to meet business changes and easy to use and manage.

Therefore, the number of emerging Cloud Computing platforms has increased, including EC2 [20] and Microsoft Live Mesh [21]. Moreover, Google has also published the Google App Engine [22] allows a user to run Web applications written using the Python programming language, and it also provides a Web-based Administration Console for the user to easily manage his running Web applications. Sun has unveiled Sun network.com (Sun Grid) [23] which enables the user to run different kinds of applications, such as SUN Solaris application. As Microsoft has presented the Azure service platform, Azure is designed to provide a wide range of internet services that can be consumed from both on-premises environments and the Internet [36]. The Azure Services Platform uses a specialized operating system, Windows Azure, to run its "fabric layer" — a cluster hosted at Microsoft's datacenters that manages computing and storage resources of the computers and provisions the resources (or a subset of them) to applications running on top of Windows Azure. Windows Azure, which has been described as a "cloud layer" on top of a number of Windows Server systems, which use Windows Server 2008 and Hyper-V to provide virtualization of services [35].

For cloud computing platforms, both power consumption and application performance are important concerns. The *GreenCloud* architecture we present in this paper is an effective method to reduce server power consumption while achieving required performance using VM technologies.

2.2 Power Management in IDC

There are extensive researches on server and IDC power management. Generally, we could categorize these individual solutions into four categories in accordance with their different features [1]. The first one is the power management of the objectives and constraints, which deals with the tradeoffs between performance and the energy saving, such as whether transient power budget violation is allowed or not, with or without additional performance constraints. The second could be viewed as the solutions concerned with the scope and granularity. For instance, some of these solutions will have the best performance at the embedded-level, while others are more efficient at the rack-level, or datacenter-level. If we compare the different management policies deployed at hardware, we will notice that even these solutions are limited at the lower level; it will have better access to the system components and smaller time granularity than the solutions of the software level. The third category is specified by the approaches utilized. Such as the local server approach, the distribution scheduling, or maybe the virtual machine consolidation. The last type of the solutions is the options that used by the power management solutions. These options include the DVFS, the system components turning On/Off, the sleeping method and etc.

Dynamic Voltage/Frequency Scaling (DVFS) is one of the key knobs adjusting the server power states. Horvath et al. [27] have studied how to dynamically adjust the server voltages to minimize the total system power consumption at the same time meet end-to-end delay constraints in a Multi-tier Web Service environment. Heo et al. [6] later studied how to combine the DVFS together with server ON/OFF to further decrease total power consumption. In [16], the authors studied the power capping solutions which ensures that the system will not violate the given power threshold. In [25], Barrsos et al, studied how to use Chip Multi-Processor (CMP) to achieve power management. In [26], Nedevschi et al. studied how to maximize the network power saving via sleeping and rate-adaption. After these individual solutions which attack different aspects of IDC power management problem are studied, Raghavendra et al. have suggested the coordination architecture to regulate different individual approaches in multi-level power management in [1].

2.3 VM Power Management & Migration

In IDC, there are two kinds of Virtualization technologies that are studied a lot recently. One is *full-virtualization* technology, such as VMWare [32]. Full-virtualization, otherwise known as native virtualization, uses a virtual machine that mediates between the guest operating systems and the native hardware. VMM mediates between the guest operating systems and the bare hardware. Certain protected instructions must be trapped and handled within the hypervisor because the underlying hardware isn't owned by an operating system but is instead shared by it through the hypervisor. On the other hand, *para-virtualization* is a very popular technique that has some similarities to full virtualization. This method uses a hypervisor for shared access to the underlying hardware but integrates virtualization-aware code into the operating system itself. This approach obviates the need for any recompilation or trapping because the operating systems themselves cooperate in the virtualization process. A typical para-virtualization product is Xen [28]

While various management strategies have been developed to effectively reduce server power consumption by transitioning hardware components to lower-power states, they cannot be directly applied to today's data centers that rely on virtualization technologies. In [41], Chen et al. have proposed ON/OFF control strategies to investigate the optimization of energy saving with desired performance levels.

Nathuji et al. [24] have proposed an online power management to support the isolated and independent operation assumed by VMs running on a virtualized platform and globally coordinate the diverse power management strategies applied by the VMs to the virtualized resource. They utilize the "Virtual Power" to represent the 'soft' versions of the hardware power state, to facilitate the deployment of the power management policies. In order to map the 'soft' power state to the actual changes of the underlying virtualized resource, the Virtual Power Management (VPM) state, channels, mechanisms, and rules are implemented as the multiple system level abstraction.

In the early research, the Collective project [29], has designed VM migration as a tool to provide mobility to users who work on different physical machines at different times. This solution aims at the process of transferring an OS instance through slow links and long time spans. With a set of enhancement work to reduce the image size, it will stop the running of the VM during the migration duration. Zap [31], which implement the partial virtualization technology to enable the migration of process domains, using a modified Linux Kernel. Recently, researchers have noticed the performance deterioration brought out by the traditional VM migration, which may lead to service unavailable during the period of the migration, which could not be acceptable in a performance-sensitive computing environment. To address this challenge, NomadBIOS [37], which is a virtualization and migration technology built on top of the L4 microkernel [38], implements pre-copy migration to achieve very short best-case migration downtimes. Later, with the research of live migration conducted by Clark, the latest version of Xen now supports the live migration of VM [30][28].

3. BACKGROUND & DESIGN OVERVIEW

The availability of inexpensive networking equipment, coupled with new standards for network cabling, led to use a hierarchical design in data centers environments for the ease of management. In the hierarchical architecture of data centers, the redundancy in routing and storage is necessary for fault-tolerance. Hence, there are multiple choices to select physical servers to host a given VM. We will seek a cost-efficient solution to do a VM migration.

3.1 Live Migration

For performance-sensitive applications, VM live migration offers great benefits we attempt to optimize the utilization of available resources (e.g., CPU). In VM live migration, a VM is moved from on physical server to another while continuously running, without any noticeable effects from the point of view of the end users. During this procedure, the memory of the virtual machine is iteratively copied to the destination without stopping its execution. The halt of around 60–300 ms is required to perform the final synchronization before the virtual machine begins executing at its final destination, providing an illusion of seamless migration. However, with the traditional VM migration technology, which stops the running VM during the migration, will cause the failure to meet the Service Level Agreement (SLA) guarantees, especially in the response time sensitive computing.

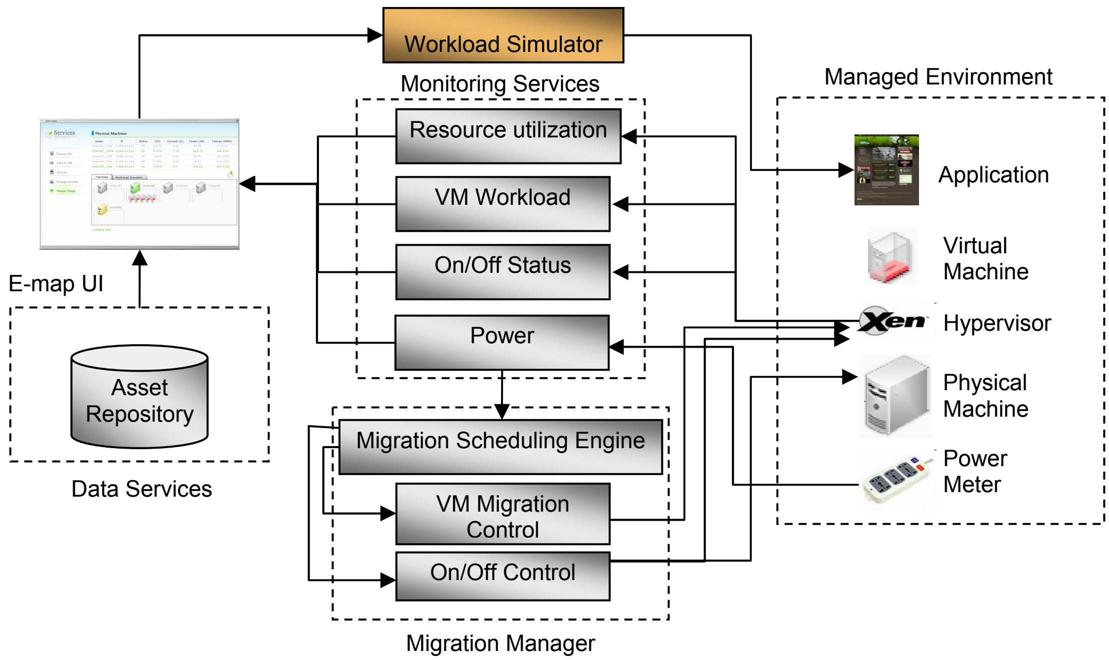

Figure 1. GreenCloud Architecture

3.2 Performance Metric

In this paper, we investigate the power efficiency and effectiveness for online gaming applications hosted in data center environment, achieved by live migration technology. We know that total power consumption on a chip consists of two parts, static and dynamic power dissipation. The static power dissipation is primarily caused by various leakage currents, and the dynamic power dissipation is proportional to workload, e.g. CPU utilization. Due to the existence of static power dissipation, a server consumes considerable amount of power even if it is idle with power on. According to [41], a server with zero workload consumes about 60% of its peak power. One of straight forward way to save power is to consolidate workload first and then turn the unnecessary devices (e.g. idle devices) off.

During the procedure of VM live migration, one may concern whether or not the performance of service from the end user point of view will be sacrificed. Among these performance metrics, round trip time (RTT) is an essential performance concern for online gaming applications. Generally, the acceptable quality for an online game requires RTT less than 600–800 ms. Hence, live migration technology is feasible to address workload consolidation for online gaming applications.

3.3 GreenCloud Architecture

As discussed above, cloud computing platform as the next generation IT infrastructure enables enterprises to consolidate computing resources, reduce management complexity and speed the response to business dynamics. Improving the resource utilization and reduce power consumption are key challenges to the success of operating a cloud computing environment. To address such challenges, we design the *GreenCloud* architecture and the corresponding *GreenCloud* exploratory system. The exploratory system monitors a variety of system factors and performance measures including application workload, resource utilization and power consumption, hence the system is able to dynamically adapt workload and resource utilization through VM live migration. Therefore, the *GreenCloud* architecture reduces unnecessary power consumption in a cloud computing environment. Figure 1 demonstrates the *GreenCloud* architecture and shows the functions of components and their relations in the architecture.

Monitoring Service monitors and collects comprehensive factors such as application workload, resource utilization and power consumption, etc. The *Monitoring Service* is built on top of IBM Tivoli framework and Xen, where the IBM Tivoli framework is a CORBA-based system management platform managing a large number of remote locations and devices; Xen is a virtual machine monitor (VMM). The *Monitoring Service* serves as the global information provider and provides on-demand reports by performing the aggregation and pruning the historical raw monitoring data to support to intelligent actions taken by *Migration Manager*.

Migration Manager triggers live migration and makes decision on the placement of virtual machines on physical servers based on knowledge or information provided by the *Monitoring Service*. The migration scheduling engine searches the optimal placement by a heuristic algorithm, and sends instructions to execute the VM migration and turn on or off a server. A heuristic algorithm to search an optimal VM placement and the implementation details of *Migration Manager* will be discussed in Section IV. The output of the algorithm is an action list in terms of migrate actions (e.g. Migrate VM1 from PM2 to PM4) and local adjustment actions (e.g. Set VM2 CPU to 1500MHz) [40].

Managed Environment includes virtual machines, physical machines, resources, devices, remote commands on VMs, and applications with adaptive workload, etc.

E-Map is a web-based service with Flash front-end. It provides a user interface (UI) to show the real-time view of present and past system on/off status, resource consumption, workload status, temperature and energy consumption in the system at multiple scales, from high-level overview down to individual IT devices (e.g. servers and storage devices) and other equipment (e.g. water- or air-cooling devices). *E-map* is connected to the *Workload Simulator*, which predicts the consequences after a given actions adopted by the *Migration Monitor* through simulation in real environment.

Workload Simulator accepts user instructions to adapt workload, e.g. CPU utilization, on servers, and enables the control of *Migration Manager* under various workloads. Then, *E-Map* collects the corresponding real-time measurements, and demonstrates the performance of the system to users. Therefore, users and system designers will verify the effectiveness of a certain algorithm or adjust parameters of the algorithm to achieve better performance.

Asset Repository is a database to store the static server information, such as IP address, type, CPU configuration, memory setting, and topology of the servers.

The *GreenCloud* IDC management framework is running and accessible to IBM internal staffs and customers. They can view up-to-date status of resources, configure their applications, allocate resources, and experience the live management system.

4. VM LIVE MIGRATION

4.1 Algorithm

We plug a heuristic algorithm in Migration Scheduing Engine to search optimal placement of a virtual machine on a physical machine to minimize the total cost. The cost includes the possible migration cost and the execution cost thereafter. The algorithm provides an open interface for users to define their own cost function depending on users' requests and system specification. In this paper, we present take the PM cost, the VM status and the VM migration cost as the inputs of searching algorithm. Next, we first present the notations and definitions used in the algorithm. First, p represents a placement of a VM on a PM. Then, $LB(p)$ means the lower bound of the cost for all the placements reachable from p by a single VM live migration decision. Formally, $LB(p) \le cost(p^+)$, where p^+ represents any placement reachable from p in a single hop of migration, and $cost(\cdot)$ represents user defined cost to execute a certain application, or run a VM. In this paper, the cost function balances the total power saving and performance of the system. The cost function is given as below.

$$cost = C(Migration) + C(\#PM) + C(Utilization)$$

where $C(Migration)$ is the cost incurred by live migration. We take the number of migrations from the start placement to current placement as the cost. $C(\#PM)$ is the energy consumed by physical machine, we take the number of PM used as such cost. $C(Utilization)$ is the measurement on how busy the servers are. Generally, if more servers are very busy, the performance of the system is poorer under surge service demand. We take the number of servers with more than 90% of CPU Utilization as the cost. In our algorithm, we denote bp as the best placement found so far, and cp means the current placement in the search, and sp represents the initial placement, where the algorithm starts the search. Therefore, if $LB(p) > cost(bp)$, it will not search the placements reachable from p for the optimal placement.

When triggered to work, the algorithm uses two tables in the search, the open table and the close table, where the open table records the initial mapping between VMs to PMs (or the placement where the search algorithm begins) and the temporary placements which are possible to reach the optimal placement during the search procedure. So the placements in open table are those which need to be explored. The close table keeps the already explored placements. When the open table becomes empty or some criteria are met (such as cost, search timeout) during search, one or more than one near optimal placements may be detected. Then one of them will be chosen based on some standards or policies and related actions list will be generated for further live migration and power on/off actions.

The heuristic algorithm we presented here is used to find a VM placement and related action list that minimizes total cost, in terms of both the Physical Machine (PM) cost and the VM live migration cost in each optimization cycle. Figure 2 gives a instance in searching the placement space, where a node represents a placement and an arrow indicates a live migration, which leads to another placement. In Figure 2, the black placements are those which have been explored but ruled out from the possibilities in the future search, because it is not possible for black placements and their "neighboring" placements to compete with the current best solution. Hence, black nodes are called *cut placements,* and the live migration to the *cut placements* are crossed out. The nodes in *explored space* except black nodes are

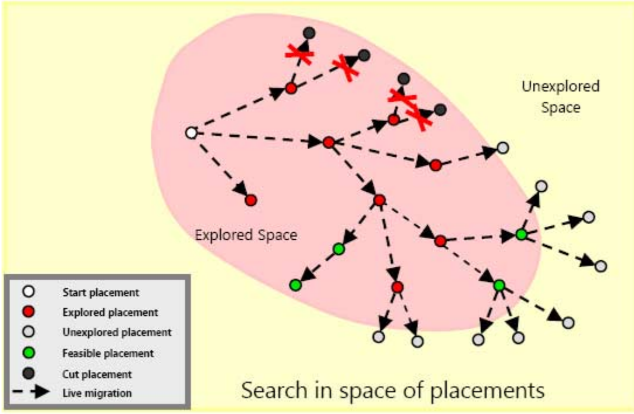

Figure 2. Heuristic search for optimal replacement

placed in close table, and the placements in the *unexplored space* (i.e., the space to be explored) are in open table.

4.2 Implementation

In our *GreenCloud* architecture, Xen has been used to support the modified SUSE Linux Enterprise Server. The ported OSes are running on isolated VMs that are nominated "Domain U" in Xen terminology. In Xen version 3.0 [28], the live migration is supported and we will achieve better energy efficiency of *GreenCloud* computing architecture through the VM live migration feature provided by Xen. Xen's live migration capability supports efficient transfer of the memory and thread of control between two domains (which means the execution context that hosts a running VM). It performs an iterative process that reduces the amount of time the virtual machine is unavailable to an almost unnoticeable level [30]. However, Xen does not support the migration of the root file system image. Xen assumes that the root file systems available on both the source and destination hosts. In our *GreenCloud* solution, we implement the migration of the file system through NFS.

It requires virtualized sharing storage to enable VM remote booting for supporting live migration of running virtual machines between physical hosts. In our environment, standard network storage sharing protocols NFS is used to provide storage to virtual machines. Firstly a root file system in a directory on the NFS server machine is populated. Then we configure the NFS server to mount the image disk of a VM and export this file system over the network by adding a line to /etc/exports, for instance:

```
/share/xen/images/tremulous01/mount
*(rw,sync,no_root_squash)
```

Finally, some values should be added into the domain configuration file of a VM to support NFS root. The following is a typical domain configuration file sample to use NFS root in addition to the normal variables:

```
# VM Kernel for NFS booting
kernel="/boot/vmk/vmlinux"
name="VM05"          # VM name
uuid="e1ccdf9d-44b5-1f84-d6d8-6536ae4fbf7a"      # VM UUID

# VM resource allocation
memory=2048
vcpus=4

# VM run level
extra="5"

#VM network and MAC
vif=[ 'mac=00:16:3e:60:b5:49', ]
vfb=['type=vnc,vncunused=1']
ip='9.186.63.125'
netmask='255.255.255.0'
broadcast='9.186.63.255'
gateway='9.186.63.1'

# VM NFS boot
root="/dev/nfs"
nfs_server="9.186.63.112"
nfs_root="/share/xen/images/tremulous05/mount"
```

Because the guest domain will need network access at boot time, the domain booting kernel is compiled to support necessary advanced features (such as NFS booting and DHCP and so on) besides compatible with Xen kernel.

5. GREENCLOUD EVALUATION

In this section, we present the experiment set up and evaluation of our GreenCloud architecture.

5.1 Experiment Setup

We implemented the GreenCloud architecture prototype in the IBM China Research Lab (CRL), and carried out extensive experiments. In this prototype, 5 IBM x-series servers are deployed: three IBM X346 with 4 cores of 3.0G CPU, one IBM X3950 with 16 cores of 3.0 CPU and one IBM X336 with 2 cores of 3.0 CPU. The X3950 machine is deployed with 16 GB memory, while the rest of the machines are all equipped with 3.0 GB memory. Each machine has 2 NIC with 1 GB bandwidth, and Wake-On-LAN (WOL) support is enabled for all machines. The configuration of all the physical machines is shown in Table 1 below:

Table 1. Physical Machine in GreenCloud

Physical Machine	Type	CPU	Memory	Network	WOL Support	NFS Server
Green01	X336	3.0G (2 Core)	3.0G	2 *1 Gigabyte	Y	N
Green02	X3950	3.0G (16 Core)	16.0G	2 *1 Gigabyte	Y	Y
Green03	X346	3.0G (4 Core)	3.0G	2 *1 Gigabyte	Y	N
Green04	X346	3.0G (4 Core)	3.0G	2 *1 Gigabyte	Y	N
Green05	X346	3.0G (4 Core)	3.0G	2 *1 Gigabyte	Y	N

Table 2. VM Configuration in GreenCloud

Virtual Machine	Type	CPU	Memory	NFS Booting Support	Gaming application
VM01	ParaVirt	2 Core	2.0G	Y	Tremulous
VM02	ParaVirt	4 Core	2.0G	Y	Tremulous
VM03	ParaVirt	4 Core	2.0G	Y	Tremulous
VM04	ParaVirt	4 Core	2.0G	Y	Tremulous
VM05	ParaVirt	4 Core	2.0G	Y	Tremulous

As we stated before, Xen is utilized as the standard VMM in our experiment. To verify the energy saving performance of the *GreenCloud* prototype, we take an online real-time gaming service Tremulous **Error! Reference source not found.**, as VM application on each VM. Tremulous is a response-time-sensitive online game, which serves well for our performance evaluation purposes. On our *GreenCloud* prototype, it is worth noting that among all experiments during VM migrations, the players who played the game could not feel the small delay caused by server migration. In fact, all participants could not notice the difference. In the experiments, all the VMs are configured as the description of Table 2.

5.2 Data Collection

An iPDU (Intelligent Power Distribution Unit) power meter is adopted to monitor the real-time power consumption of physical machines. Power-related parameters monitored by the power meter for a machine include Current, Voltage, Power and Kilowatt-hour. To inspect energy consumption details of IT equipment and facilities in the system, the power meters are deployed as the following topology design as shown in Figure 3.

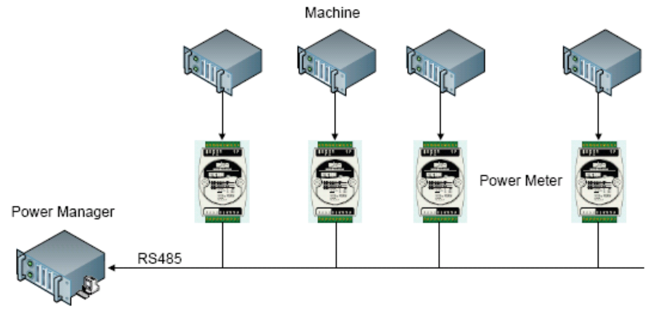

Figure 3. Power Meters Deployment

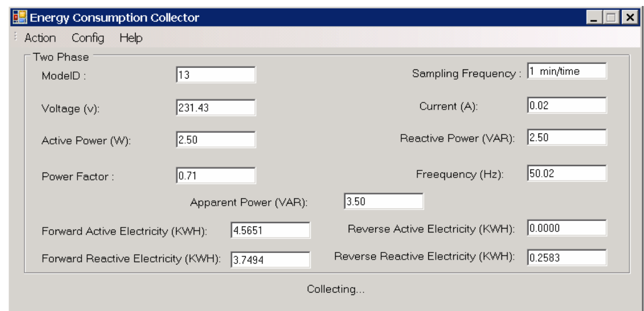

Figure 4. Energy Consumption Collector

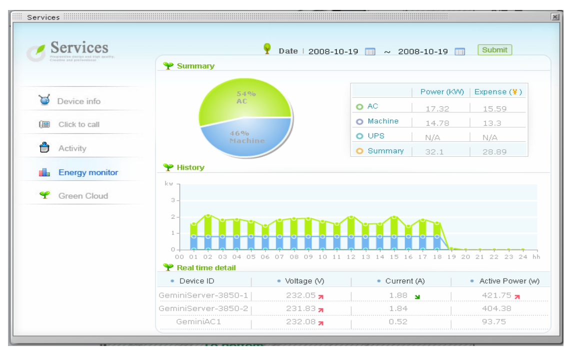

Figure 5. Energy Consumption Visual Interface

Power Manager is implemented to collect monitored information from power meters or other monitoring devices, consolidate data into a database, analyze/mine historical measurements, and provide customized query services and reports so that server, storage, and facilities measurements can be brought together into integrated views with visualization that provides a clear understanding of data center energy consumption and temperature behavior. Figure 4 and figure 5 present the Energy Consumption Collector and Visualization interface of the *GreenCloud* prototype, respectively.

5.3 VM & Physical Host Management

The Xen Management API is utilized in *Migration Monitor* as the interface to remotely configure and control VMs running on a Xen-enabled host. The Xen API is built on top of XML- RPC and can be used by the user space components of Xen, such as the *xm*

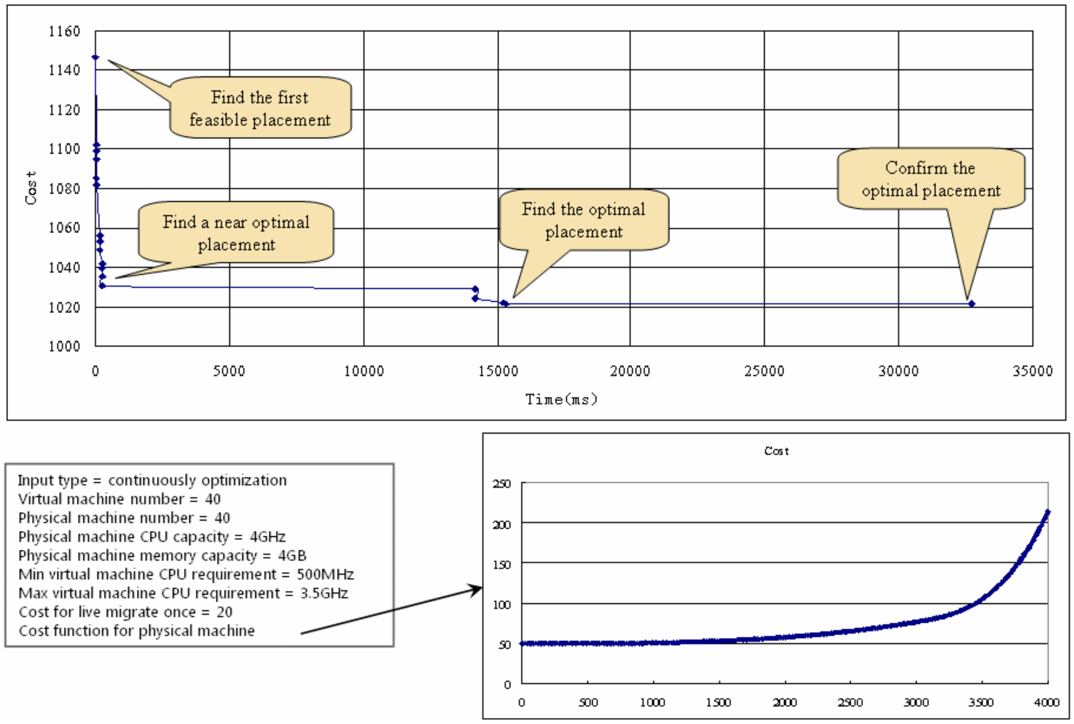

Figure 6. Performance of Heuristic Search

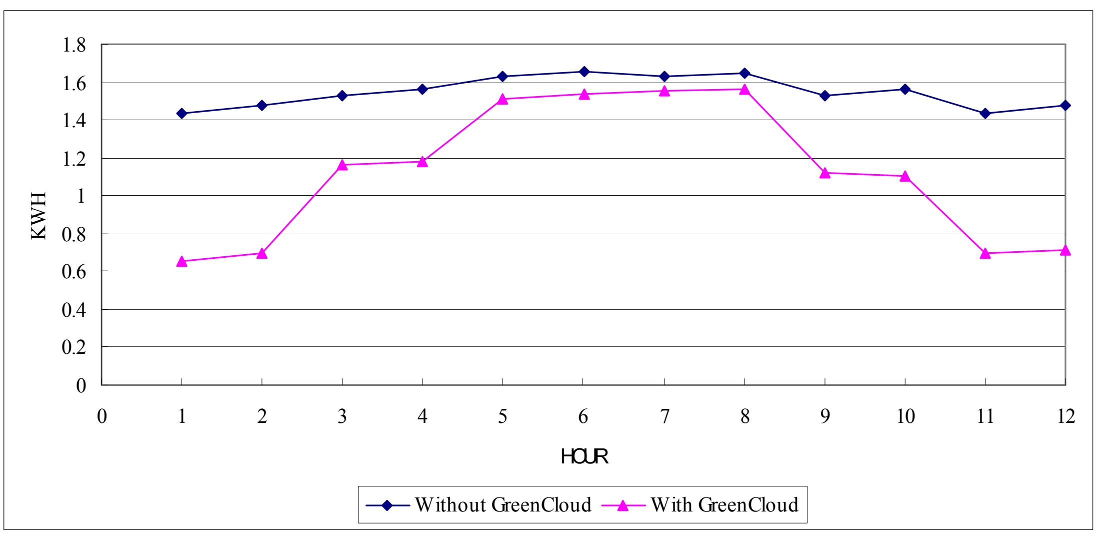

Figure 7. Comparison of Energy Consumption

command-line tool to control the system. The *xend* daemon listens for XML-RPC connections and then performs a number of administrative functions as required, such as dynamically migrate VMs from a host to another.

On the other hand, in order to switch on a remote computer from the network, the Migration Monitor will send a magic packet (using the Wake-On-LAN technology) through the network using the IP address and the MAC number of the computer intended to Wake Up. Most of host management functions including resource allocation, monitoring and provisioning VMs can be achieved by Xen Management API. For instance, to shutdown a host remotely, users can call the following Xen API:

void shutdown (session_id s, host ref host).

5.4 Evaluations

First, we did a verification work to confirm the effectiveness of our heuristic search algorithm (Section IV). A running sample is shown in Figure 6 . The x-axis shows the algorithm running time, and y-axis shows the performance cost measure. The performance cost measure is in terms a normalized cost totaling the total server energy consumption cost and server migration costs. The goal of the heuristic algorithm is to find the optimal placement solution to minimize the total performance cost. As we can see from Figure 6, our algorithm can obtain a near-optimal solution very fast in less than 300 ms in such a test environment.

In order to evaluate the energy saving brought by *GreenCloud,* we setup a typical application scenario for hosting real-time online gaming services. We developed a workload emulator to facilitate generating variable workload to the gaming servers easily. It consists of two components: a front proxy agent and emulated user agents. The proxy agent can remotely control and manage emulated user agents distributed on different servers. It also provides interfaces for testers to set an expected CPU utilization (such as low, middle or high) on a designated server. When an emulated user agent receives such commands to adjust the server workload, it will start or stop user threads to occupy CPU time using workloads with different characteristics designed. The dynamically changed workload on servers will trigger the Migration Manager to take corresponding actions to balance the performance and power consumption. The testing workload is presented in Table 3. As we can see from Table 3, the testing workload is ramping up from 30% to 55% to 75%, then going down back to 30% again during the total 12-hours experiment period.

The total energy consumption results are reported in Figure 7. We compared the case when there is no *GreenCloud* is deployed ("without *GreenCloud"*) to that of the case when *GreenCloud* is deployed ("with *GreenCloud"*). The x-axis shows the experiment time, and y-axis shows the total energy consumptions of the servers in the unit of KWH. Each point corresponding to time T, corresponds the total energy used in the previous hour (i.e. in the time interval [T-1, T)). We can see at the beginning and the end of the experiment, when server workload is low, using *GreenCloud,* we can get significant energy reduction. This is because in this case, VMs are consolidated using live migration, hence only 1 of the physical servers is running while the other servers are turned off by *GreenCloud,* thus gives us significant energy saving. When the workload reaches the highest (from hours [5-8] corresponding to the workload in Table 3), *GreenCloud* use 4 of the servers, so the energy consumption is still less than that when no *GreenCloud* is used. It is worth noting that during the adjustment using *GreenCloud*, there is no interruption to the services running on the cloud computing platform: our prototype testbed shows that the response-time is below 750 *ms*, successfully meeting the SLA (Service Level Agreement) of this response time sensitive gaming application.

As a conclusion, *GreenCloud* effectively saves energy by dynamically adapts to workload leveraging live VM migrations, at the same time meeting system SLAs.

Table 3 Workload Simulation

CPU Utilization	30%	55%	75%	55%	30%
Time (Hour)	2	2	4	2	2

6. FUTURE WORK & CONCLUSION

Cloud computing is emerging as a significant shift as today's organizations which are facing extreme data overload and skyrocketing energy costs. In this paper, we propose *GreenCloud* architecture, which can help consolidate workload and achieve significant energy saving for cloud computing environment, at the same time, guarantees the real-time performance for many performance-sensitive applications. The *GreenCloud* leverages the state-of-the-art live virtual machine migration technology to achieve these goals. Through evaluation, we show that *GreenCloud* achieved our design goals effectively in the Cloud Computing environment.

In the future, there are still a number of research activities that we plan to carry out, which could improve the performance of *GreenCloud* and bring solid value to users to achieve their business goals and their social responsibility in Green IT. First, further studies should be given to explore whether utility-based methodology can be used to communicate performance-power tradeoffs between the OS and the application/middleware. Second, we need to adjust *GreenCloud* to meet the requirement of real business services, such as the web service, Online Transaction Processing (OLTP), and the human resource management etc. Our future work also includes the VM live migration through Wide Area Network (WAN) with energy efficient scheme.

7. REFERENCES

[1] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, X. Zhu. No Power Struggles: Coordinated multi-level power management for the data center. In Thirteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '08), Mar. 2008.

[2] A. Weiss. Computing in the Clouds. netWorker, 11(4):16-25, Dec. 2007.

[3] R. Buyya, C. S. Yeo, S. Venugopa. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08, IEEE CS Press, Los Alamitos, CA, USA) 2008.

[4] Ensim. Ensim Virtual Private Servers, http://www.ensim.com/products/materials/ datasheet_vps_051003.pdf, 2003.

[5] A. Whitaker, M. Shaw, S. D. Gribble, "Lightweight Virtual Machines for Distributed and Networked Applications". Technical Report 02-02-01, University of Washington, 2002.

[6] J. Heo, D. Henriksson, X. Liu, T. Abdelzaher, "Integrating Adaptive Components: An Emerging Challenge in Performance-Adaptive Systems and a Server Farm Case-Study," in Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS'07), Tucson, Arizona, 2007.

[7] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, K. Salem, "Adaptive control of virtualized resources in utility computing environments", Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, 2007

[8] EPA Report on Server and Data Center Energy Efficiency. U.S. Environmental. Protection Agency, ENERGY STAR Program, 2007.

[9] P. Bohrer et al. The case for power management in web servers. In Power Aware Computing (PACS), 2002.

[10] D. Brooks, M. Martonosi. Dynamic thermal management for high-performance microprocessors. In 7th International Symposium on High-Performance Computer Architecture, 2001.

[11] E. V. Carrera, E. Pinheiro, R. Bianchini. Conserving disk energy in network servers. In 17th International Conference on Supercomputing, 2003.

[12] J. Chase et al. Managing energy and server resources in hosting centers. In 18th Symposium on Operating Systems Principles (SOSP), 2001.

[13] J. Chase and R. Doyle. Balance of power: Energy management for server clusters. In 8th Workshop on Hot Topics in Operating Systems, May 2001.

[14] Y. Chen et al. Managing server energy and operational costs in hosting centers. In ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, June 2005.

[15] M. Elnozahy, M. Kistler, R. Rajamony. Energy-efficient server clusters. In Power Aware Computing Systems (PACS), February 2002.

[16] X. Fan et al. Power provisioning for a warehouse-sized computer, In 34th ACM International Symposium on Computer Architecture, CA, June 2007.

[17] W. Felter et al., A performance-conserving approach for reducing peak power consumption in server systems. In 19th International Conference on Supercomputing, 2005.

[18] M. Femal, V. Freeh. Safe over-provisioning: Using power limits to increase aggregate throughput. In Power-Aware Computing Systems (PACS), December 2004.

[19] Twenty Experts Define Cloud Computing, http://cloudcomputing.syscon.com/read/612375_p.htm, July 2008.

[20] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/, July 2008.

[21] Microsoft Live Mesh, http://www.mesh.com

[22] Google App Engine, http://appengine.google.com

[23] Sun network.com (Sun Grid), http://www.network.com

[24] R.Nathuji, K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems", ACM Symposium on Operating Systems Principles, Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, 2007.

[25] L. A.Barroso, U. Hölzle, The Case for Energy-Proportional Computing, IEEE Computer, vol 40, no. 12 (2007): 33-37.

[26] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing Network Energy Consumption via Sleeping and Rate-Adaptation, In Proceedings of the 5th USENIX Symposium on Networked Systems Design & Implementations (NSDI'08), San Francisco, CA, April 2008.

[27] T. Horvath, T.Abdelzaher, K. Skadron, X. Liu, Dynamic Voltage Scaling in Multi-tier Web Servers with End-to-end Delay Control, in IEEE Transactions on Computers (ToC), vol. 56, pp. 444-458, 2007

[28] Xen User Manual, http:// bits.xensource.com/Xen/docs/user.pdf

[29] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam, M.Rosenblum. Optimizing the migration of virtual computers. In Proc. of the 5th Symposium on Operating Systems Design and Implementation (OSDI-02), December 2002.

[30] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of Virtual Machines. In USENIX NSDI, 2005.

[31] S. Osman, D. Subhraveti, G. Su, J. Nieh. The design and implementation of zap: A system for migrating computing environments. In Proc. 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI-02), pages 361–376, December 2002.

[32] VMWare, VMWare Inc. http://www.vmware.com

[33] Microsoft Virtual Server, Microsoft Coroporation, http:// www.microsoft.com/windowsserversystem/virtualserver/

[34] Microsoft 2008 Hyper-V, Microsoft Corporation, http://www.microsoft.com/canada/windowsserver2008/serverunleashed/html/hyper-v.aspx?wt.srch=1

[35] Azure Service Platoform, Wikipedia, http://en.wikipedia.org/wiki/Microsoft_Azure

[36] Azure Service Platform, Micrsoft Corportation, http://www.microsoft.com/azure/services.mspx

[37] J. G. Hansen, A. K. Henriksen. Nomadic operating systems. Master's thesis, Dept. of Computer Science, University of Copenhagen, Denmark, 2002.

[38] H. H¨artig, M. Hohmuth, J. Liedtke, S. Sch¨onberg. The performance of microkernel-based systems. In Proceedings of the sixteenth ACM Symposium on Operating System Principles, pages 66–77. ACM Press, 1997.

[39] X. Jin, Q. Wang etc, "A Framework for Virtualized Service Hosting", in preparation for publication.

[40] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao. "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services", in Proceedgins of the 5th USENIX Symposium on Networked Systems Design & Implementation (NSDI'08), San Francisco, CA, April 2008.

[41] Tremulous official website, http://tremulous.net