Power Consumption of Virtual Machines in Cloud Computing: Measurement and Enhancement

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yan Bai

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Haiyang Wang

July 2016

Acknowledgements

I would like to take this opportunity to thank Dr. Haiyang Wang for the continuous support of my Master study and research. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Yang Li, Prof. Ted Pedersen for their encouragement, insightful comments, and hard questions. My sincere thanks also goes to Jim Luttinen, Lori Lucia and Clare Ford for their timely help.

Dedication

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, Jinbo Bai and Fengying Gao whose words of encouragement and push for tenacity ring in my ears.

I also dedicate this dissertation to my many friends and church family who have supported me throughout the process. I will always appreciate all they have done.

Abstract

Virtulization is one of the cornerstone technologies that makes utility computing platforms such as cloud computing a reality. With the accelerating adoption of cloud computing, the virtualizaion-based cloud platforms are consuming a significant amount of energy. However, the design of a green and efficient virtualization technology remains an open issue to both industry and academia.

In this thesis, we for the first time investigate the virtual machine's (VM's) power consumption while supporting different services and applications (e.g., web, database and streaming). In particular, we establish a cloud computing platform in the The University of Minnesota Duluth. This platform consist of both Xen and KVM nodes and the VMs can be easily accessed from the Internet. Our real-world measurement indicates that the existing virtulization technologies add considerable energy overhead to the data centers. For example, a busy virtualized database server can consume 30% more energy than its non-virtualized counterparts. To address such a problem, we propose a shared-memory-based enhancement to reduces the extra interrupts and memory copies for cloud virtualization. The evaluation indicates that our approach can reduce VM's energy consumption by 11% without noticeable loss of its running performance.

Contents

List of Tables

List of Figures

1 Introduction

Cloud computing has dominated information technology industry in recent years. Because of the on-demand, flexible and scalable service it is able to provide, a lot of enterprises which previously deployed locally have migrated their businesses to cloud. Cloud computing also benefits personal users as it provides convenient services which replace desktop applications. Cloud computing provides services over the Internet and is supported by large, distributed, virtualized data center. Each data center contains countless servers. Each physical server is divided into several, independent, isolated virtual machines. This strategy has many benefits such as increased resource utilize, migration, high flexibility and scalability. However, the increasing popularity of could computing also introduces problems, especially the issue of energy efficiency. According to the Natural Resources Defense Council (NRDC), nationwide data centers used a total of 91 billion kilowatt-hours (kWh) of electrical energy in 2013, and they will use 139 billion kWh by 2020. Currently, data centers consume up to 3 percent of all global electricity production while producing 200 million metric tons of carbon dioxide [8]. So, how to control the power consumption of servers in data centers has become a crucial topic. In this paper, we discuss the power consumption of virtualized servers in data centers. Since each physical machine contains several virtual machines, so if there are one million physical servers in a data center, there can five or six million virtualized servers running. If we could reduce the power consumption of individual virtual machine and then apply the optimization to every virtual machine, the final saving will be significant. We measured the power consumption of virtual machine in a cloud computing platform. We measured the power consumption of virtual machines in three scenarios. Our research reveals that virtual machines consume up to 40 percent more power than physical machines. We then propose an approach that adopts shared memory between host and guest to improve energy efficiency of virtual machines, and therefore reduce the overall power consumption of each physical server. The evaluation shows that our approach could reduce the power consumption by up to 12 percent. The remainder of the thesis is structured as follows: In chapter 2, we present the background and related works. Chapter 3 introduces CloudStack [11] cloud computing platform in which we conduct our research and experiment. Chapter 4 presents our power consumption measurement approach and results. In chapter 5, we propose an approach that uses shared memory to improve energy efficiency of virtual machines. We conclude our research and future work in chapter 6.

2 Background

This thesis is about energy efficiency issue of virtual machine in cloud computing. In this section, we will introduce background and related work about cloud computing, virtualization technology and energy efficiency issue.

2.1 Cloud Computing

Cloud computing is one of the most popular topic and fast-growing technique in the field of information technology. Traditionally, computer user operated almost everything locally before the popularity of cloud computing. Personal users stored their files on local drive and installed a lot of program on their computers. Enterprises purchased a lot of expensive hardware and maintained them internally to run their information system. Nowadays, the first application a personal user installed on a new operating system must be a browser. Then many services can be accessed via browser instead of installing on the computer. For example, Google Docs [16] is an excellent alternative to Microsoft Office [22]. DropBox [9] provides file storage service which could replace flash-drive. These services are inspired and supported by cloud computing and they have many advantages compare with traditional desktop applications. For enterprises, there is a trend that more and more enterprises are moving their services to third-party clouding computing service provider such as Amazon Web Services [2]. Take advantage of cloud computing, enterprises could provide flexible and scalable service and cut off cost.

Cloud computing offers on-demand computing resources - everything from applications to data centers - over the Internet on a pay-for-user basis. Cloud computing provides a various of computing resources from applications to data centers. The services could be categorized into three buckets according to their service model.

The first one is infrastructure as a service(Iaas) which is probably the most popular cloud computing service model. In the case of Iaas, the service provider offers virtualized hardware to users. Virtualized hardware includes a virtual machine, storage space, virtual network. Iaas service providers maintain massive physical resources across the world. These physical resources are virtualized by splitting individual resource into several virtualized, isolated resources. This approach maximums utilization of individual resource and offers flexibility. For example, Iaas service providers split a physical machine into several virtual machine and rent them to different users. All tenants use their own virtual machine on same physical machine but they are isolated. Even a user shutting down his/her virtual machine won't affect other users. In this model, users have full control of the virtual resource and they do not need to worry about maintenance. This is a huge benefit for both enterprise customers and personal customers. It enables enterprise customers to build cost effective and easily scalable IT solutions where the complexity and expense of managing underlying hardware are outsourced to service providers so they just need to focus on their own business. If the scale of their business increase, they could purchase more virtual resource and release them when the scale decrease. Famous Iaas service provides includes Amazon Web Service [2], Windows Azure [21] and Google Cloud [15].

The second one is platform as a service(Paas). In this model, service providers offer both hardware and software to users. Compare with Iaas, Paas provides software environment in addition to underlying software. For example, deploying a website requires developers to buy and install hardware, operating system, development environment, database, web server, then develop the website and deploy it. After deploying it, developers need to maintain and monitor it. It is also common to develop a analysis system to display statistics of the website. Paas service providers simplify this process by offering a configured environment so developers only need to log in and start programming the website which is their core work. This model is good choice for individual developer and small enterprise which does not have a big IT team. These customers usually have specific purpose such as develop a website or a CRM system and they do not have enough time and energy to take care their application and underlying environment and hardware. If they want to develop a website, they could use Google App Engine [14]. And choose SalesForce [26] if they want a CRM system.

The last one is software as a service(Saas). Iaas services delivery infrastructure, Paas services delivery platform so Saas services delivery a software. This software does not refer to traditional software needed to be installed locally. It usually refers to a central hosted software and user accesses it via a thin client or a web browser. Iaas and Paas services basically serve developers and IT teams but Saas usually serve personal users and enterprise customers. Typical Saas service provides include Dropbox [9] which provides file storage service, Google Docs [16] which provides document editing service and Netflix [23] which provides video streaming service.

Cloud computing shows a very important idea that help people focus on their core work without worrying about other relevant work. For example, what is its main concern when a food company decides to develop an online store. It is not what server and environment this online store running on but is how this online store looks like. The IT team of this food company needs to design and develop this online store. Without cloud computing, the team needs to build underlying infrastructure and software environment. Cloud computing take care of those for customers by offering a flexible and robust solution. Users could focus on their major business without worrying about many technical details.

The main enabling technology for cloud computing is virtualization and broad network. Next sections we will explain virtualization technology and how it supports cloud computing.

2.2 Virtualization

Virtualization technology is the cornerstone of cloud computing. Virtualization technology separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. Virtualization including but not limited to a virtual computer hardware platform, operating system, storage device, or computer network resources. The ability to run multiple virtual machines(VMs) on single server hardware platform provides cost, system management, and flexibility advantage in IT infrastructure today. Hosting multiple VMs on single hardware platforms reduces hardware expenses and helps minimize infrastructure costs such as power consumption and cooling.

The story of virtualization started in the early 1960's. Virtualization is first pioneered by compaines like General Electric, Bell labs and IBM. IBM developed CP-67 system which is the first commercial Main Frame to supoort Virtualization. In 1993 , VMware [29] figured out how to virtualize the x86 platform. The x86 architecture offers four levels of privilege known as Ring 0, 1, 2, 3. The lower the ring number, the higher the privilege of instruction being executed. The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3. Most instructions from guest OS could be executed on the level where guest OS located. However, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0. The difficulty in trapping and translating these sensitive and privileged instruction requests at runtime was the challenge that originally made x86 architecture virtualization look impossible [28].

VMware introduced full virtualization to solve this issue. The approach is a combination of binary translation and direct execution on the processor. The guest OS is placed on Ring 1 and hypervisor is placed on Ring 0. The hypervisor scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions. When these instructions are identified, they're trapped into the VMM, which emulates the behavior of these instructions. The method used in this emulation is called binary translation. Non-privileged instructions run directly on the hardware. This approach does not require guest OS modification since hypervisor could trap critical instructions and emulate them. The guest OS is fully abstracted from the underlying hardware by the virtualization layer. The guest OS is not aware it is being virtualized and requires no modification. However, the performance of full virtualization is not very ideal since it involves binary translation which is time-consuming [18].

In paravirtualization model, the guest OS is modified to export privileged instructions to hypervisor. Hypervisor provides a bunch of APIs and the guest OS calls these APIs to transfer privileged instructions to hypervisor which is called "hypercall". Paravirtualization also is able to replace multiple privileged instructions with single hypercall which reduce context switch between privileged and nonprivileged instructions. This model reduce the amount of work hypervisor needs to do compare to full virtualization. This results in a lower virtualization overheads and close to native performance. The problem of paravirtualization is its compatibility and portability may be in doubt, because it must support the unmodified OS as well.

Hardware assisted virtualization is a type of full virtualization where the microprocessor architecture has special instructions to aid the virtualization of hardware. Intel Virtualization Technology (VT-x) and AMD's AMD-V which both target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0. The privileged and sensitive calls are set to automatically trap to the hypervisor, removing the need for either binary translation or paravirtualization. The guest state is stored in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Blocks (AMD-V). Processors with Intel VT and AMD-V became available in 2006, so only newer systems contain these hardware assist features.

The technology behind virtualization is known as virtual machine monitor(VMM) or virtual manager, which separates compute environment from the actual physical infrastructure. KVM [12] and XEN [13] are the most widespread virtual machine monitor. Both of them are open source project right now. KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It was merged into Linux kernel mainline in kernel version 2.6.20, which was released on February 5, 2007. KVM requires a processor with hardware virtualization extension.

Xen is well known for its use of paravirtualization and near-native performance. The Xen hypervisor is managed by a specific privileged guest running on the hypervisor knows as Domain-0 or Dom0. Dom0 is a specially modified Linux kernel which is started by the Xen hypervisor during initial system start-up. It is responsible for managing all the aspects of the unprivileged virtual machine or Domain-US (DomU)that are also running on the hypervisor. DomU guests do not have direct access to the physical hardware but instead are required to make CPU, I/O and disk requests to the Xen hypervisor (Dom 0). DomU guests required a modified kernel to run on the Xen hypervisor.

2.3 Energy Efficiency

The widespread use of cloud computing services is expected to increase the power consumed by equipment in cloud computing environments rapidly. Major cloud computing service providers maintain a distributed computing resources network across the world. The AWS Cloud operates 32 Availability Zones within 12 geographic Regions around the world. Each availability zones has at least one data center which usually contains 50,000 to 100,000 servers. There are several cloud computing service providers have similar scale of resources to resources of Amazon and also there are much more medium and small enterprises have their own data center running their own cloud. In 2013, U.S. data centers consumed an estimated 91 billion kilowatt-hours of electricity, equivalent to the annual output of 34 large (500-megawatt) coal-fired power plants. Data center electricity consumption is projected to increase to roughly 140 billion kilowatthours annually by 2020, the equivalent annual output of 50 power plants, costing American businesses $13 billion annually in electricity bills and emitting nearly 100 million metric tons of carbon pollution per year [8].

Many researches regarding power consumption of cloud computing have been done. Some researchers estimated and improved power consumption of individual virtual machine. Marcu et al. [20] explores how virtualization influence the power consumption of physical systems they are implemented on with VMWare virtualization solution. It shows that virtualization solution doesn't significantly increase power consumption. Ricardo et al. [19] evaluates the performance and energy efficiency of virtual machine with different power model. Ryan et al. [25] measured the power consumption of network transaction with different VM hypervisor. It shows that virtual machine consume more power than physical machine for doing same task. Qiang et al. [24] studies the effect of live migration on power consumption of virtualization system. Satoshi et al. [27] proposed two VM packing algorithm to lower power consumption of virtual machine.

In contrast, some researchers studied and optimized power consumption of cloud computing on architecture level. Anubha Jain et al. [3] proposed several high level ideas regarding cloud computing energy efficiency. Bharti et al. [6] reviews the efforts made by various researchers to make Cloud Computing more energy efficient, to reduce the carbon footprint rate by various approaches and also discusses the concept of virtualization and various approaches which use virtual machines scheduling and migration to show how these can help to make the system more energy efficient. Jayant et al. [17] present an analysis of energy consumption in cloud computing. It shows that energy consumption in transport and switching can be a significant percentage of total energy consumption in cloud computing. Awada et al. [5] provide generic energy consumption models for server idle and server active states. The result can be used for developing potential energy legislation and management mechanisms to minimize energy consumption.

There are a few works about how to measure and model cloud power consumption. Feifei et al. [10] presents a new energy consumption model and associated analysis tool for Cloud computing environments. It measures energy consumption in Cloud environments based on different runtime tasks. Ata et al. [4] present a novel power modelling technique, VMeter, based on online monitoring of system-resources having high correlation with the total power consumption. The monitored system subcomponents include: CPU, cache, disk, and DRAM. Chongya et al. [7] proposes methods to meter the power consumption of VM, and design scheduling strategies based on power consumption to migrate VM between hosts in order to achieve power capping. Aman et al. [1] presents a solution for VM power metering, named Joulemeter. It infers power consumption from resource usage at runtime.

3 CloudStack Platform

The topic of my research is power consumption of virtual machine in cloud computing platform. We first measure the power consumption of virtual machine and then propose an approach to reduce the power consumption. We create virtual machine with different hypervisors(KVM and XEN) and measure power consumption in different scenarios(HTTP, Database, video converting). We conducted all the measurement in cloud computing platform - CloudStack. CloudStack is an open-source project belongs to Apache foundation. It is designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform. We conduct our research in CloudStack for several reasons. The first reason is that CloudStack provides a real cloud computing platform so we are able to measure the energy efficiency of virtual machine in it. We could measure the power consumption on a standalone virtual machine but measure the power consumption in CloudStack reflects the condition in real cloud computing platform. In addition, there is no cloud computing platform in University of Minnesota Duluth campus so it is helpful to establish one. Students or faculties who are interested in cloud computing could use it as study platform. Future students could conduct their research on this platform. In the future, if the university or computer science department needs a cloud computing platform, the CloudStack platform we established could be a prototype.

Figure 3.1 shows a small scale architecture of CloudStack. It is consisted of management server, computing node, nfs server, switch and firewall. We will explain management server, computing node and nfs server below.

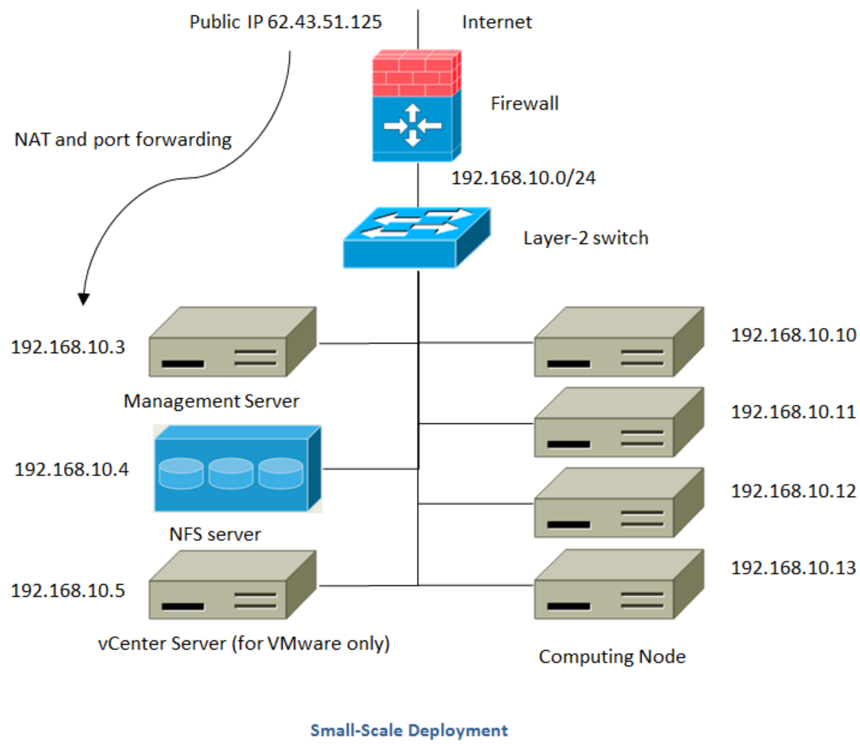

Figure 3.1: CloudStack Architecture.

Management server is the core of CloudStack platform. It is responsible for managing all the physical and virtual computing resources in platform including computing node, virtual machine, virtual network, virtual router, internet address resource. In addition, it provides a web console which make the management easily. Figure 3.2 and 3.3 show the web console of CloudStack.

Computing node, also know as host, is at where virtual machine locate. Host machine are connected with management server and are managed by management server. Cloudstack agent has to be installed on host so the host could be managed by management server. Also, the host machine must has virtual machine manager such as KVM or Xen.

NFS server is the main storage of the platform. NFS stands for network file system which is a distributed file system protocol which allows user on a client access file on remote machine. In CloudStack, there are two storage - primary storage and secondary storage.

Primary storage persists virtual machine system images. Saving virtual machine images on NFS server decouples system image and computing resource. So a particular virtual machine could be deployed on any host rather than a fix host. This enables virtual machine migration and increase system flexibility. Secondary storage persists system template or system iso. User could create system template or import system iso and they could be used to install operating system on a fresh new virtual machine.

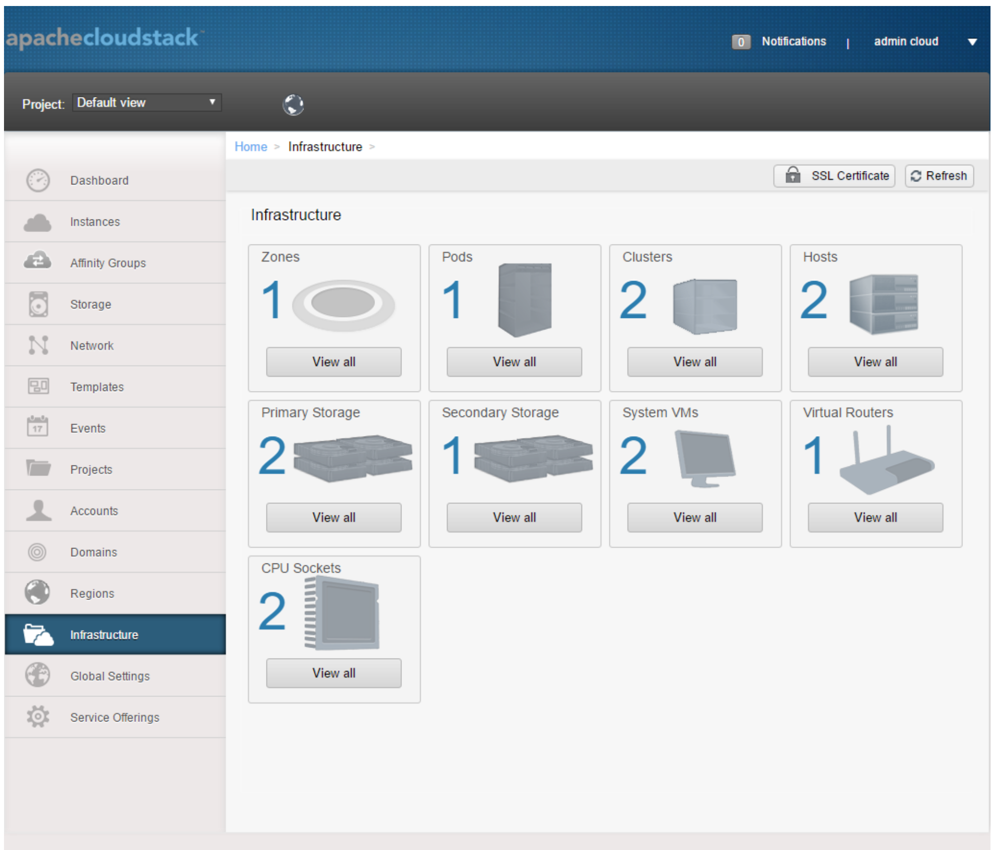

Figure 3.2: CloudStack Web Console - infrastructure.

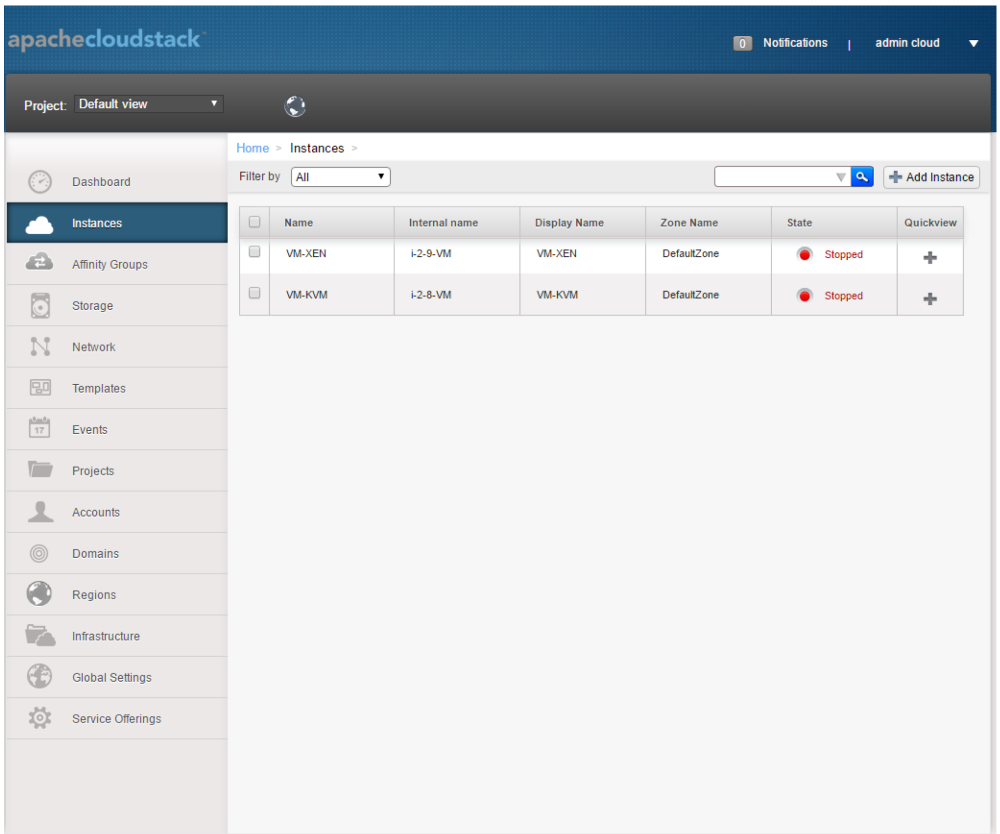

Figure 3.3: CloudStack Web Console - instance.

4 Power Consumption measurement

In this chapter, we present our approach of measurement and result. We measured power consumption of virtual machine in CloudStack platform. We added different workload on virtual machine and observer theirs' power consumption. Section 4.1 explains how we read the power consumption of virtual machine and section 4.2 shows measurement result.

4.1 Power Reading

To determine the power consumption of virtual machine, we wired a digital multimeter into the power cord of physical computer. The current will pass through the multimeter so the multimeter could measure the current. We connect the multimeter with a PC so that the multimeter could send the reading data to the PC. The multimeter read the current data every 2 seconds and we calculate the average value of current. Finally, multiply the current value with resistance to get watt value. Following images shows how we cut the power cord and wired a digital multimeter into the power cord. Figure 4.1 and 4.2 shows how we connect the multimeter with power line.

4.2 Measurement Result

We measured power consumption of virtual machine in three scenarios. The first scenario is HTTP service power consumption. HTTP is abbreviation of Hypertext Transfer Protocol which is an application protocol for distributed, collaborative, hypermedia infor-

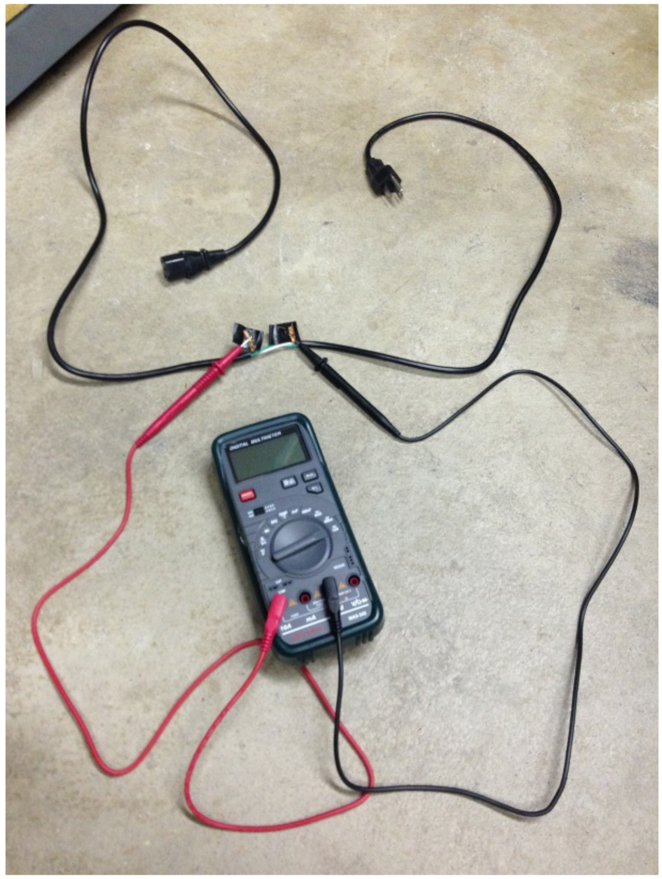

Figure 4.1: Power line and multimeter.

mation system. HTTP is the foundation of data communication for the World Wide Web. It functions as a request-response protocol in the client-server computing model. The client submit a HTTP request message to the server. The server, which provides resources such as HTML files or other content, returns a response message to the client. Cloud computing users access service via internet. Specifically, they use their browser or client application to access cloud computing server and the browser and client rely on HTTP connection to communicate with server. So HTTP is one of the most important service in cloud computing environment. There are many HTTP web service application such as Apache web service and Nginx. We use Apache web service in our experiment since it is the most popular HTTP web service.

Figure 4.2: Multimeter probe.

The second scenario is DBMS service power consumption. DBMS is abbreviation of database management system. It is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. Almost all internet service providers rely on DBMS to store and retrieve data. It also functions as a request-response protocol in the client-server computing model. The client submit a database request message to the server. The server, which provides data resources , returns a response message to the client. It is as important as HTTP service. Almost every user operation incur one or more HTTP requests and one or more DBMD request. There are many popular DBMS such as Oracle, MySQL, IBM DB2, Microsoft SQL Server. We use MySQL in our experiment.

The last scenario is video converting operation. Recent studies show that internet traffic generated by video streaming is the main source of all internet traffic. Video streaming service provider such as Youtube, Netflix and Hulu are getting more and more popular. And that incur more and more video processing work on the server side. In general, video processing work such as video converting job takes a lot of computing resources. We choose FFmpeg as video processing application in our experiment.

4.2.1 HTTP Service

We first created virtual machine on KVM and XenServer host respectively. Then we installed Apache web service 2.2 on virtual machine. We have a client machine on the lab and we generate HTTP request from the client machine using ab. Ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving. Following example shows how to generate HTTP request with ab:

```
$ ab -n 100 -c 10 http://192.168.1.11/test.html
```

This command will generate 100 HTTP request to the target URL with 10 threads. We generated traffic with 1 thread, 10 threads and 100 threads. We want to see are there any difference among different threads of request.

Above three figures show the power consumption of virtual machine and physical machine with HTTP workload. Figure 4.3 shows the result of 10 concurrent http clients. The bare metal machine consumes 35.9 watt power, virtual machine with KVM hypervisor consumes 37.8 watt power and virtual machine with Xen hypervisor consume 41.5 watt power. Virtual machine with KVM hypervisor consume 5% more power than bare-metal machine and virtual machine with Xen hypervisor consume 15% more power than bare-metal ma-

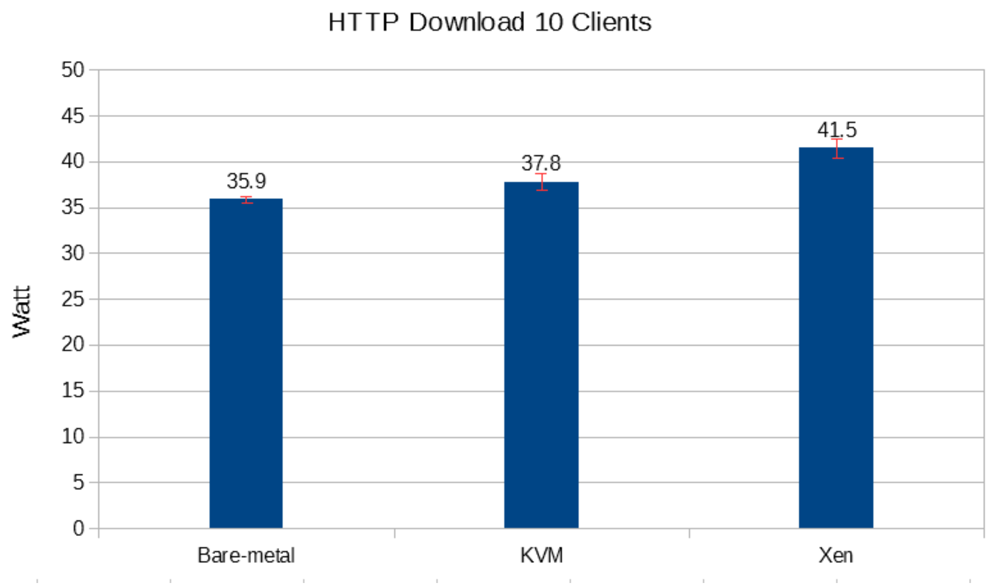

Figure 4.3: HTTP 10 Client.

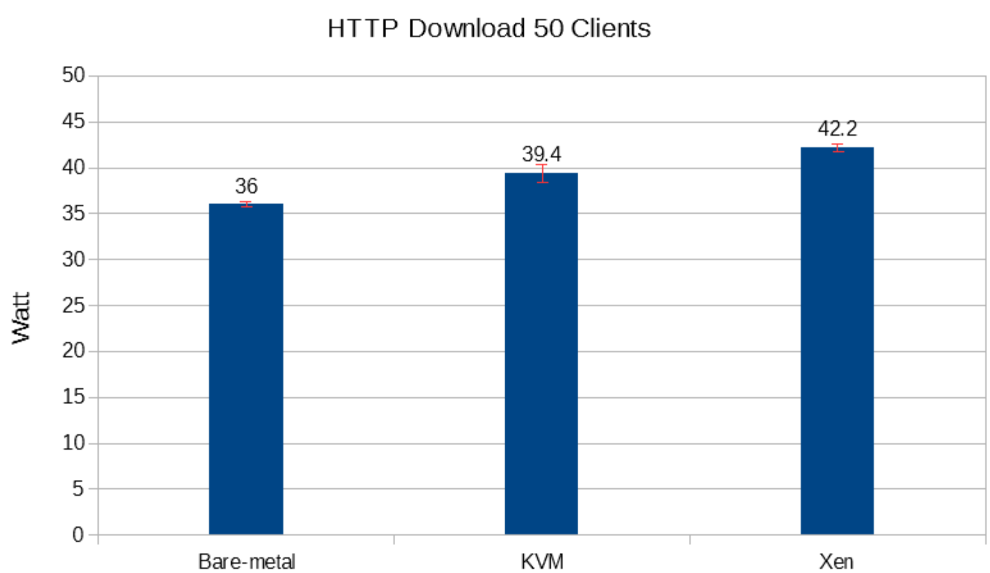

Figure 4.4: HTTP 50 Client.

chine. We find this difference also in http 50 concurrent clients and http 100 concurrent clients. Another finding is the number of concurrent http client does not impact on the power consumption. For example, for virtual machine with KVM hypervisor, it consume

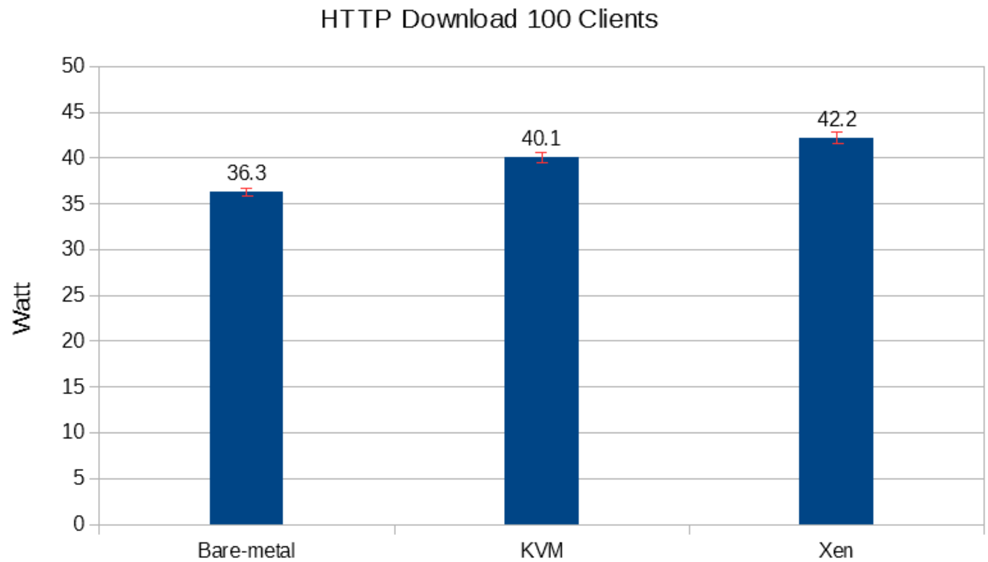

Figure 4.5: HTTP 100 Client.

37.8 watt with 10 http clients, consume 39.4 watt with 50 clients and 40.1 watt with 100 clients.

4.2.2 RDBMS Service

We first created virtual machine on KVM and XenServer host respectively. Then we installed MySQL server on virtual machine. We have a client machine on the lab and we generate SQL request from the client machine using mysqlslap. Mysqlslap is a diagnostic program designed to emulate client load for a MySQL server and to report the timing of each stage. It works as if multiple clients are accessing the server. Following command shows how to generate SQL request:

```
$ mysqlslap --user=sysadmin --password --host=
    server_address  --concurrency=50
  --iterations=10 --create-schema=employees --query="
    SELECT * FROM dept_emp";
```

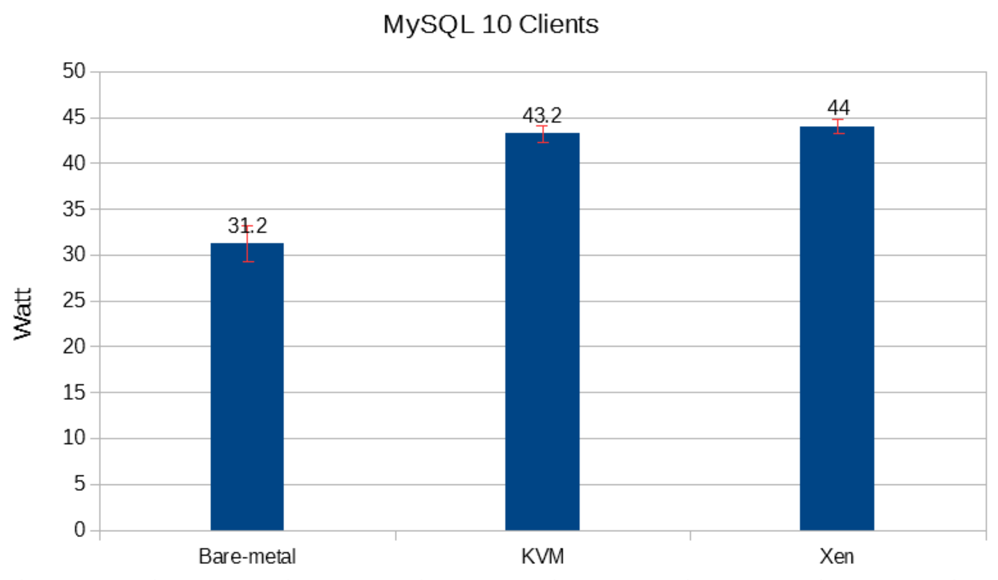

Figure 4.6: MySQL 10 Client.

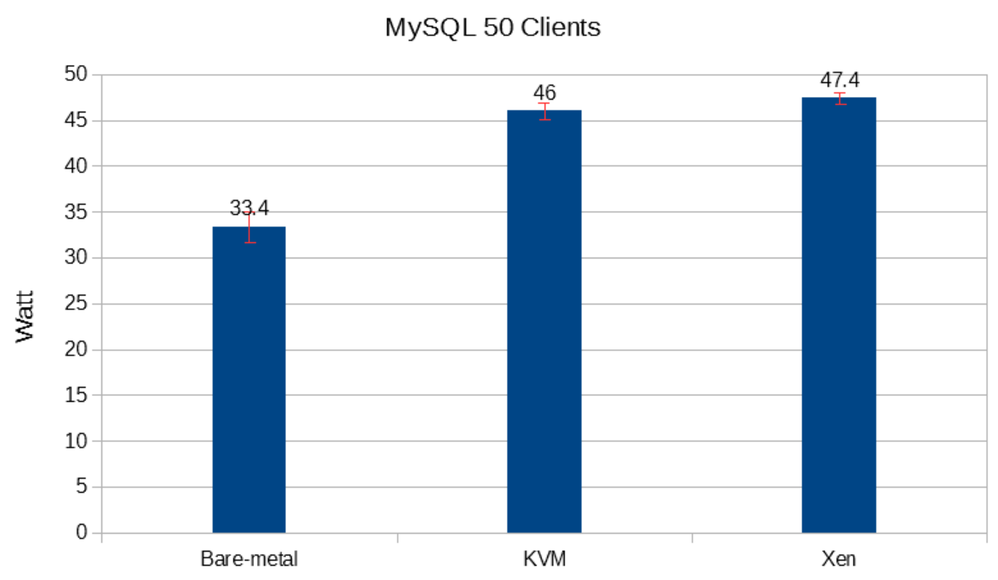

Figure 4.7: MySQL 50 Client.

Above figures show the power consumption of virtual machine and physical machine with MySQL workload. Figure 4.6 shows the result of 10 concurrent MySQL clients. The bare metal machine consumes 31.2 watt power, virtual machine with KVM hypervisor con-

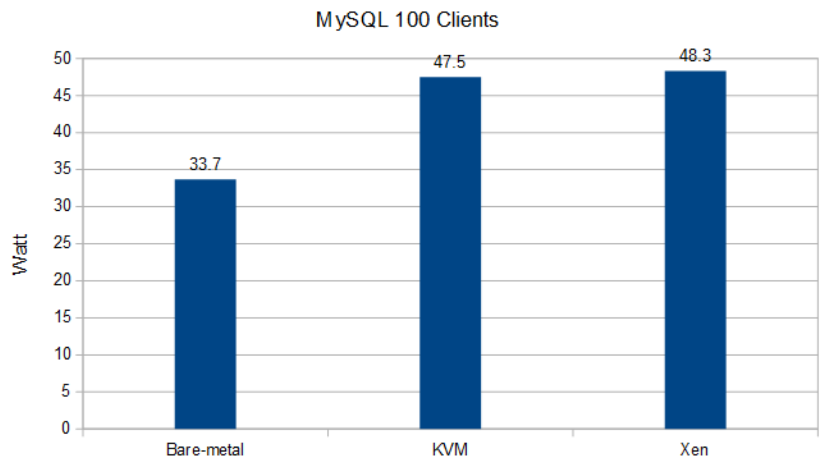

Figure 4.8: MySQL 100 Client.

sumes 43.2 watt power and virtual machine with Xen hypervisor consume 44 watt power. Virtual machine with Xen hypervisor and KVM consume same amount of power. And they consume almost 40% power more than bare-metal machine. We find this difference also in database 50 concurrent clients and database 100 concurrent clients.

4.2.3 Video Processing

In this scenario, we present the result of power consumption measurement when processing video converting task. In our experiment, we converted a MP4 video to AVI video with FFmpeg. The video file we use is a 4GB video file in MP4 format and we convert it into AVI format. We keep all configuration of the video file expect the format. The convert approach is pretty simple, just use ffmpeg coomand line and give the input file and output file path:

```
$ ffmpeg -i input.mp4 output.avi
```

The figure above shows that virtual machine and physical machine consume almost same power with video processing transaction. The bare-metal machine consume 45.8 watt

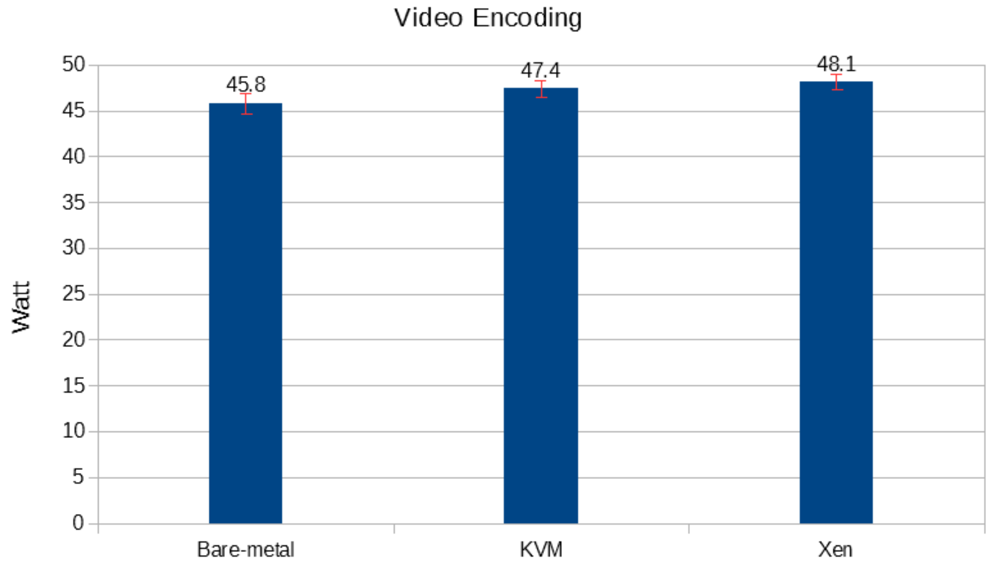

Figure 4.9: Video Encoding.

in average when converting MP4 file to AVI file. The virtual machine with KVM hypervisor consume about 2 more watt when processing the converting task. The virtual machine with Xen hypervisor consume almost same power with virtual machine with KVM hypervisor when processing the converting task. The result of video converting is different from the results in the scenario of HTTP download and RDBMS system. The reason could be that in HTTP download and RDBMS scenario, the virtual machine needs to send response to client machine. The virtual machine needs to send the response to host machine then the response is sent to client machine. The process of transferring the response from virtual machine to host machine could be the main reason of additional power consumption. Since that requires copy data in memory space of virtual machine into the memory space of host. The copy will generate additional interrupts. However, in the case of video converting, all the operations are in the virtual machine and that does not require communication between host and virtual machine.

4.2.4 Large and Small Virtual Machine

Besides measuring power consumption of virtual machine with different workload, we also measure same workload with different virtual machine configuration. We created virtual machine with two configurations. The configurations are presented in tables below:

Table 4.1: Small and Large VM

Small Instance	1 CPU core and 1 GB RAM
Large Instance	4 CPU cores and 4 GB RAM

We add same workload to both virtual machine and observer the different between in terms of energy efficiency. The first measurement we made is 100 HTTP downloading. The result 4.10 shows that small instance and large instance almost consume same power. The KVM large instance consumes 40.8 watt while the small instance consumes 39.4 watt. There is a difference between them but the different can be ignored. The Xen large instance consumes less power than small instance and also the different can be ignored. The result of database test verifies the result of HTTP download test. To process 100 database query requests, KVM small instance consumes 47.9 watt while the large instance consumes 46.7 watt. Xen small instance consumes 47.1 watt while the large instance consumes 47.8 watt. The difference between the large and small instance could be ignored. We find same trend in video encoding scenario. KVM small instance consumes 47.4 watt while large instance consume 48.6 watt. KVM small instance consumes 48.1 watt while large instance consumes 48.4 watt.

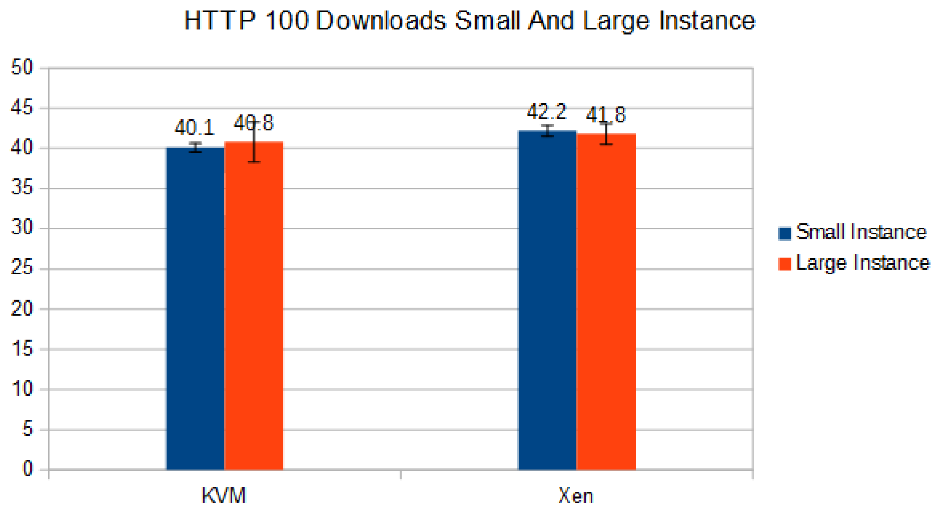

Figure 4.10: Video Encoding.

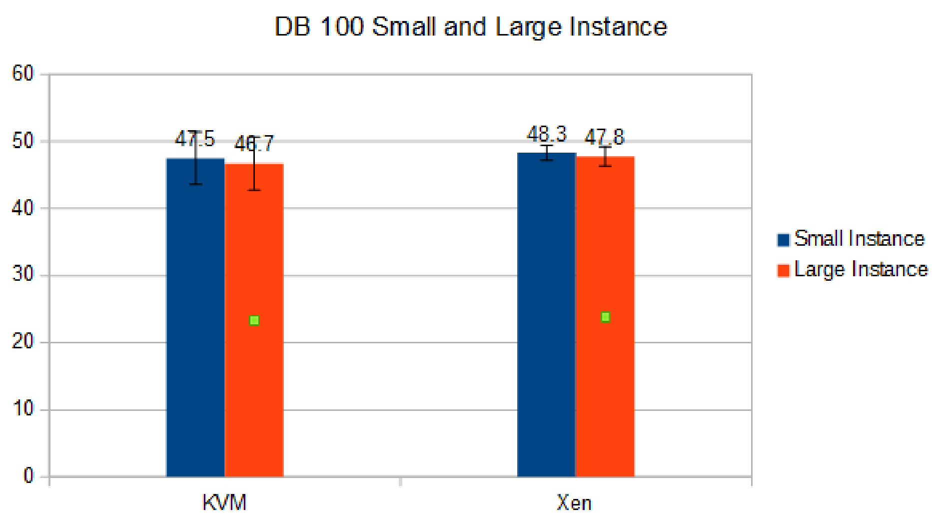

Figure 4.11: Video Encoding.

4.2.5 Power Consumption Analysis

Last section shows that virtual machine consume additional power and this section takes a close look at how virtualization technology works to determine why there is additional energy consumption. We use example of how KVM hypervisor process HTTP download task. When client machine sending a HTTP request to virtual machine, the packet is first

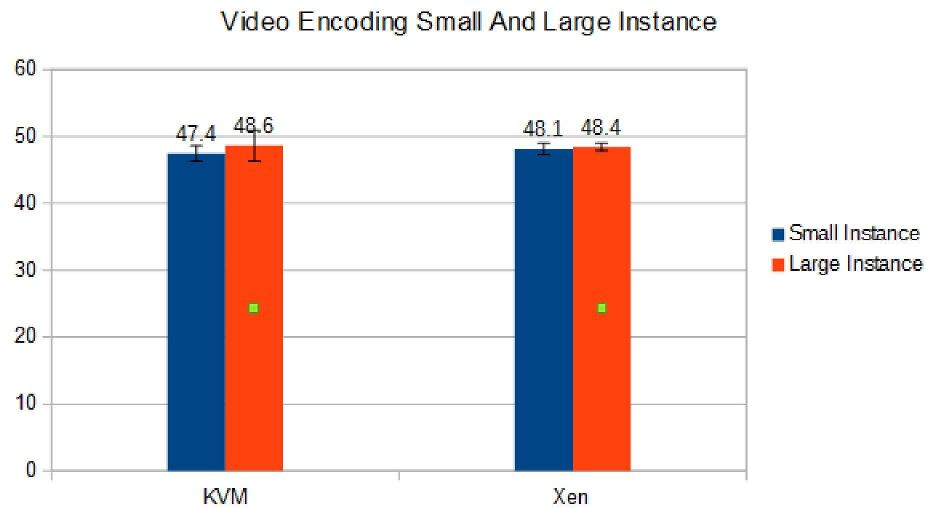

Figure 4.12: Video Encoding.

handled by the physical NIC. Then the physical NIC generates a hardware interrupt to alert the operating system and CPU of the incoming packet. After that, the operating system sends the packet to the virtual NIC through a network tap device. The operating system then notifies KVM hypervisor of the incoming packets through a software interrupt. Then the KVM hypervisor copies the packet fro the host's memory space into the virtual machine's memory space, and sends an interrupt to the virtual machine. Finally, the operating system of virtual machine gets the packet from the virtual NIC and passes the packet to application. When virtual machine sending a reply to client machine, the virtual machine simply pushes a packet along in the reverse direction through these steps. Compare with bare-metal machine, it requires more steps when processing in virtual machine. All of these additional steps cause additional power consumption.

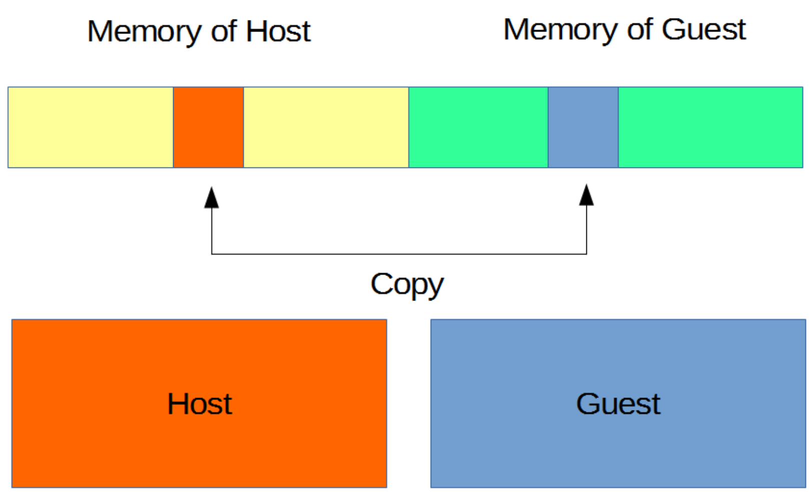

Figure 4.13: Host and guest memory copy.

5 Reducing Power Consumption

In this section, we propose an approach of reducing power consumption of virtual machine with KVM hypervisor and present evaluation of our approach. The results of our experiment shows that virtual machine consume additional power when processing task compare to physical machine. Based on our previous description of how packets are processed in virtual system, we find that virtual system requires a couple of additional steps when processing task. These additional steps generate hardware and software interrupts. These interrupt are the major source of additional power consumed by virtual machine. So the power consumption of virtual machine will be reduced if we are able to reduce the interrupts generate by virtual system. There is a previous approach introduced which uses cache mechanism to reduce the interrupt, therefor reduce the overall power consumption. In that approach, the packet is not be handled once it be received by the server. Instead, that approach establishes a cache to store incoming packets. Then it releases all packets in cache once the cache is full and the virtualized system processes all packets at same time. This approach significantly reduces the number of interrupts generated. However, this approach introduces unnecessary latency since the packet is not be handled on time. This approach is not acceptable if the application requires low latency. Based on our observation, we proposed that shared memory between host and guest machine could help reduce the overhead caused by the packet copy, and therefore reduce the power consumption. In our previous analysis of how packets are process in virtual system, we find that the KVM hypervisor needs to copy data in host's memory space into guest's memory space in order to transfer the data. This strategy generates a lot of interrupts and takes a lot of CPU cycles. If shared memory is set between host and guest, the data is not needed to be copied between host and guest. This will reduce the interrupts generated by the copies and therefore reduce the power consumption. Shared memory will achieve zero-copy between host and guest system since the data is stored in a shared memory by host and guest so no data copy in memory is required.

5.1 Shared Memory Implementation

In this section, we present our approach of establishing shared machine. KVM has a tool called 9p-virtio(Plan 9 folder sharing over VirtIO). It uses a paravirtual file system driver, which avoids converting the guest application file system operations into block device operations, and then again into host file system operations. The QEMU server on the host elects to export a portion of its file system hierarchy, and the client on the guest mounts this using 9P2000.L protocol. Guest see the mount point just like any of the local file systems, while the reads and writes are actually happening on the host file system. 9p-virtio uses Plan-9 network protocol for communication between the guest and the host. It provides a direct memory sharing on top of the native host and guest I/O transport. The shared RAM space is ported as a local file system on the guest VM. This allows the memory space shared in such a way that both guest and host I/O operations can be zero-copy. Figure 5.1 shows the architecture of shared memory between host and guest.

Creating shared memory with KVM 9p-virtio is naturally supported on Linux operating system. Use qemu-kvm command will create the shared memory.

```
# /usr/bin/qemu-kvm -m 1024 -name centos6 -fsdev local,
    security model=passthrough, id=fsdev0,path=/tmp/share
    -device virtio-9p-pci, id=fs0, fsdev=fsdev0,
    mounttag=hostshare
```

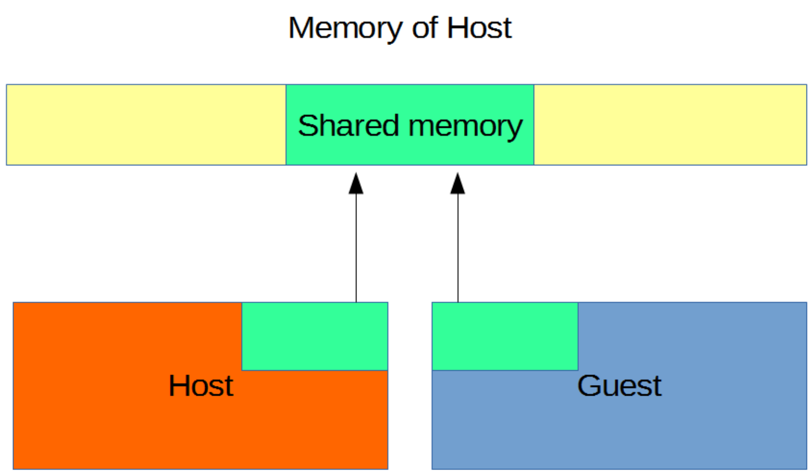

Figure 5.1: Shared Memory Architecture.

This tells qemu on the host machine to create a 9p virtio device exposing the mounttag hostshare. That device is coupled to an fsdev named fsdev0, which specifies which portion of the host filesystem we are sharing, and in which mode. Now, in the guest we need to mount the 9p filesystem from the host using the virtio transport. The mounttag is used to identify the host's share.

```
# mkdir /tmp/host_files
# mount -t 9p -o trans=virtio,version=9p2000.L hostshare
    /tmp/hostfiles
```

Now the shared memory has been created and it represents as a shared folder. In the guest machine, all the I/O operations on /tmp/hostfiles directory is actually operated on host machine. This achieve memory zero-copy and could reduce power consumption.

5.2 Evaluation

In previous section, we propose an approach which use 9p virtio shared memory to improve energy efficiency of virtual machine I/O operations. In the section, we present our evaluation of our proposed approach. We use HTTP download, RDBMS and video encoding scenario to verify the effect of our approach. In HTTP scenario, we change the directory of HTTP server to the mount point so the target download file will be stored in the shared memory. This could eliminate the process of copy the target download file from guest's memory space to host's memory space. In RDBMS scenario, we install MySQL database in the shared memory directory. So all database data is stored in the shared memory space. In video encoding scenario, we put the source video file in shared memory space and set the destination video file also in shared memory space.

In HTTP download experiment, the virtual machine with shared memory consumes 36.1 watt. It shows 11% energy efficiency improvement compare with virtual machine without shared memory which consumes 40.8 watt power. In MySQL database experiment, we find shared memory cut 5% power consumption by virtual machine. The virtual machine with shared memory consumes 45.2 watt power while virtual machine without shared memory consumes 47.9 watt. In video encoding experiment, virtual machine with and without shared memory consume almost identical power. Virtual machine with shared memory consumes 46.6 watt power while virtual machine with shared memory consumes 47.4 watt.

Shared memory works better in HTTP download scenario compare with MySQL and video encoding scenario. The reason is that in HTTP download task, the guest needs to copy every byte of downloaded file from guest's memory space into host's memory space. However, there is no data switch between host and guest and all the I/O operations happen in guest's space so the shared memory does not help. In MySQL experiment, the host needs to copy SQL query packet from it memory space into guest's memory space. And guest

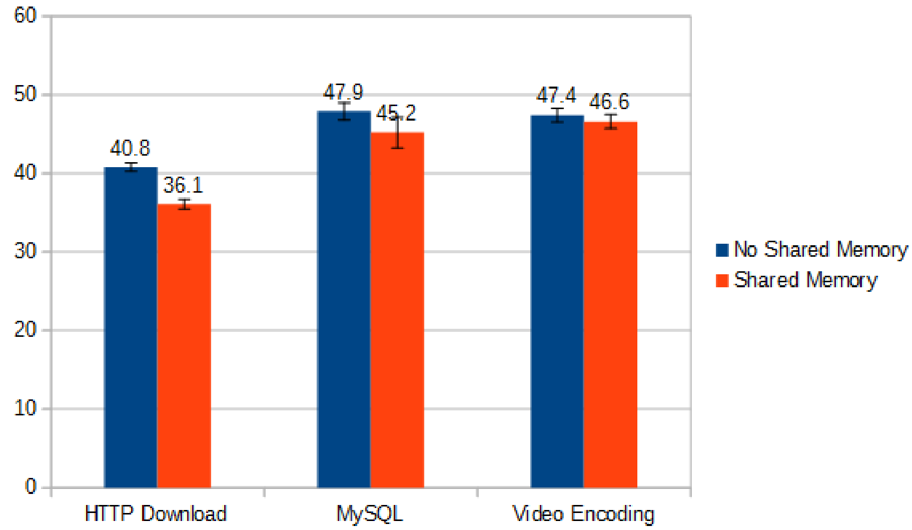

Figure 5.2: Measurement of shared memory.

needs to copy query result from its memory space into host's memory space. However, the total data to be copied in MySQL experiment is much less than the data in HTTP Download experiment.

6 Conclusions

In my research, we measured power consumption of virtual machine in cloud computing platform with typical real world application. We established a local cloud computing platform using CloudStack and conducted experiment on the platform. We created virtual machine with KVM and XenServer respectively and compare energy efficiency with bare metal machine. We add three typical workload to virtual machine and bare metal machine and read the power consumption data with multimeter. The workload we added are HTTP file downloading, MySQL database query and FFmpeg video encoding. The result of experiment demonstrates that virtual machine consume more power in all three scenarios. Based on our analysis of the process of virtualized system, we identified that data copy between host's memory space and guest's memory space is one of source of additional power. We propose an approach of reducing power consumption of virtual machine using shared memory between host and guest machine. We implemented a shared memory space with 9p-virtio filesystem. 9p-virtio creates a space on host and guest system mount the space so the guest machine could read and write data on the shared space. All I/O operations happen in the shared space are operated on host system. It eliminates the need of data copy between host's memory and guest's memory. Our evaluation of the proposed approach shows that shared memory could reduce up to 11% power consumption. The shared memory we used in our approach is not flexible and dynamic. However, it is very easy to implement and we consider it as a verification of our theory. A dedicated approach of creating shared memory could be a big improvement:

- Our approach leverage 9p-virtio to create a shared memory space between host and guest. However, 9p-virtio is not designed for shared memory so we have to put all data in the mounted space. Also, the capacity of shared memory can not be change dynamically. It is more efficient and convenient to create a flexible and dynamic shared memory.

- Also, the shared memory we created can only be shared between guest and host machine. Another idea is creating a shared memory space which could be shared by all guests machine running on same host machine. Since in real world cloud computing platform, one single physical server contains several virtualized system so a memory space which could be shared by all guests could improve performance and efficiency of virtualized system.

Bibliography

[1] K. Aman, Z. Feng, L. Jie, and K. Nupur. ``Virtual machine power metering and provisioning''. In: *SoCC '10 Proceedings of the 1st ACM symposium on Cloud computing*. New York, U.S.A: ACM, 2010, pp. 39–50. DOI: 10.1145/1807128.1807136 (cit. on p. 10).

[2] Amazon. *Amazon Web Service*. URL: https://aws.amazon.com/about-aws/ (cit. on pp. 3, 4).

[3] J. Anubha, M. Manoj, and P. Sateesh Kumar. ``Energy efficient computing - Green cloud computing''. In: *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference*. Indian Institute of Technology, Roorkee, India: IEEE, 2013, pp. 978–982. DOI: 10.1109/ICEETS.2013.6533519 (cit. on p. 9).

[4] B. Ata E Husain and C. Vipin. ``VMeter: Power modelling for virtualized clouds''. In: *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium*. University of Buffalo, New York, U.S.A: IEEE, 2010, pp. 1–8. DOI: 10.1109/IPDPSW.2010.5470907 (cit. on p. 10).

[5] U. Awada, L. Keqiu, and S. Yanming. ``Improving cloud computing energy efficiency''. In: *Cloud Computing Congress (APCloudCC), 2012 IEEE Asia Pacific*. Dalian University of Technology, Dalian, China: IEEE, 2012, pp. 53–58. DOI: 10.1109/APCloudCC.2012.6486511 (cit. on p. 10).

[6] W. Bharti and V. Amandeep. ``Energy saving approaches for Green Cloud Computing: A review''. In: *Engineering and Computational Sciences (RAECS), 2014 Recent Advances*. University Institute of Engineering and Technology, Chandigarh, India: IEEE, 2014, pp. 1–6. DOI: 10.1109/RAECS.2014.6799608 (cit. on p. 9).

[7] M. Chongya, J. Zhiying, Z. Ke, and Z. Guangfei. ``Virtual machine power metering and its applications''. In: *Global High Tech Congress on Electronics (GHTCE), 2013 IEEE*. Chinese Academy of Sciences, Beijing, China: IEEE, 2013, pp. 153–156. DOI: 10.1109/GHTCE.2013.6767262 (cit. on p. 10).

[8] N. R. D. Council. *Data Center Efficiency Assessment*. Tech. rep. Aug. 2014. URL: https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IP.pdf (cit. on pp. 1, 9).

[9] Dropbox. *Dropbox*. URL: https://www.dropbox.com/about (cit. on pp. 3, 5).

[10] C. FeiFei, S. Jean-Guy, Y. Yun, G. John, and H. Qiang. ``An energy consumption model and analysis tool for Cloud computing environments''. In: *Green and Sustainable Software (GREENS), 2012 First International Workshop*. Swinburne University of Technology, Melbourne, Australia: IEEE, 2012, pp. 45–50. DOI: 10.1109/GREENS.2012.6224255 (cit. on p. 10).

[11] A. Foundation. *CloudStack Open Source Project*. URL: https://cloudstack.apache.org/about.html (cit. on p. 2).

[12] L. Foundation. *Kernel Virtual Machine*. URL: http://www.linux-kvm.org/page/Main_Page (cit. on p. 8).

[13] L. Foundation. *Xen Project*. URL: http://www.xenproject.org/ (cit. on p. 8).

[14] Google. *Google App Engine*. URL: https://cloud.google.com/appengine/ (cit. on p. 5).

[15] Google. *Google Cloud Platform*. URL: https://cloud.google.com/ (cit. on p. 4).

[16] Google. *Google Docs*. URL: https://www.google.com/docs/about/ (cit. on pp. 3, 5).

[17] B. Jayant, A. Robert, and K. Hinton. ``Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport". In: *Proceedings of the IEEE*. IEEE, 2010, pp. 149–167. DOI: 10.1109/JPROC.2010.2060451 (cit. on p. 10).

[18] G. F. Kai Hwang Jack Dongarra. *Cloud Computing: Virtualization Classes*. Tech. rep. Feb. 2012. URL: https://technet.microsoft.com/en-us/magazine/hh802393.aspx (cit. on p. 7).

[19] R. Lent. ``Evaluating the Performance and Power Consumption of Systems with Virtual Machines". In: *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference*. London, Britain: IEEE, 2011, pp. 778–783. ISBN: 978-1-4673-0090-2. DOI: 10.1109/CloudCom.2011.120 (cit. on p. 9).

[20] M. M and T. D. ``Power Consumption Measurements of Virtual Machines". In: *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium*. Timisoara, Romania: IEEE, 2011, pp. 445–449. ISBN: 978-1-4244-9108-7. DOI: 10.1109/SACI.2011.5873044 (cit. on p. 9).

[21] Microsoft. *Microsoft Azure*. URL: https://azure.microsoft.com/en-us/overview/what-is-azure/ (cit. on p. 4).

[22] Microsoft. *Microsoft Word*. URL: https://products.office.com/en-us/word (cit. on p. 3).

[23] Netflix. *Netflix*. URL: https://media.netflix.com/en/about-netflix (cit. on p. 5).

[24] H. Qiang, G. Fengqian, W. Rui, and Q. Zhengwei. ``Power Consumption of Virtual Machine Live Migration in Clouds''. In: *Communications and Mobile Computing (CMC), 2011 Third International Conference*. Shanghai Jiaotong University, Shanghai, China: IEEE, 2011, pp. 122–125. DOI: 10.1109/CMC.2011.62 (cit. on p. 9).

[25] S. Ryan, W. Haiyang, and L. Jiangchuan. ``Power consumption of virtual machines with network transactions: Measurement and improvements''. In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. Simon Fraser UniversityBritish Columbia, Canada: IEEE, 2014, pp. 1015–1059. DOI: 10.1109/INFOCOM.2014.6848035 (cit. on p. 9).

[26] Salesforce. *Salesforce CRM*. URL: http://www.salesforce.com/what-is-salesforce/ (cit. on p. 5).

[27] T. Satoshi, T. Atsuko, S. Maiko, N. Hidemoto, K. Tomohiro, and Y. Akiko. ``Virtual Machine packing algorithms for lower power consumption''. In: *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference*. Tsukuba, Ibaraki, Japan: IEEE, 2012, pp. 161–168. DOI: 10.1109/CloudCom.2012.6427493 (cit. on p. 9).

[28] VMWare. *Understanding Full Virtualization, Paravirtualization, and Hardware Assist*. Tech. rep. Nov. 2007. URL: http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf (cit. on p. 6).

[29] VMWare. *VMWare*. URL: https://www.vmware.com/ (cit. on p. 6).

A Appendix A

A.1 CloudStack Setup

In this section, we present how we establish the CloudStack cloud computing platform in our lab. CloudStack is an open-source project belongs to Apache foundation. It is designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform.

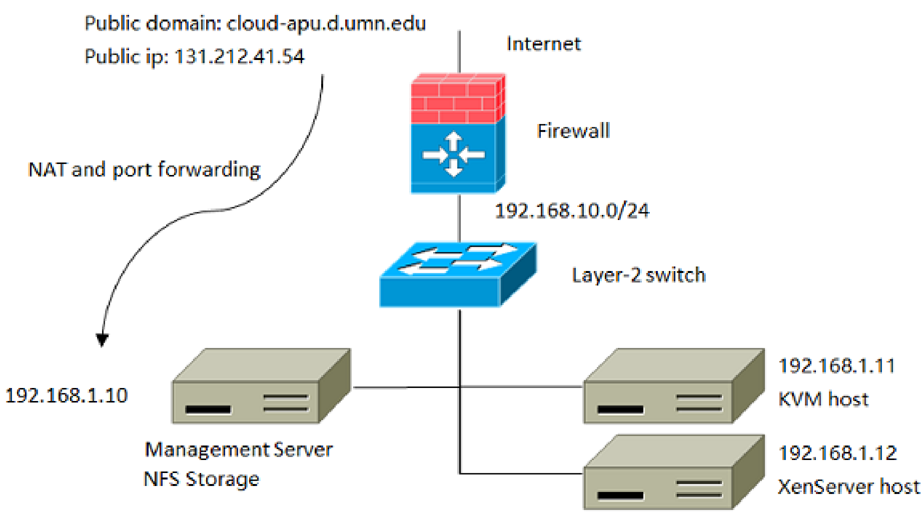

Figure A.1: CloudStack Architecture in Lab.

Above figure A.1 shows the architecture of our CloudStack platform. Following table shows the hardware configuration:

Table A.1: Hardware Configuration

Management Server	AMD Athlon(tm) II X2 B24 Processor 4GB DDR3 RAM Seagate 160GB HardDisk Broadcom NetXtreme BCM5761
Host Machine	Intel 4th generation Core i5 Quad Core 4GB DDR3 RAM Seagate 160GB HardDisk RTL8111
Router	NETGEAR N300

A.1.1 Management Server Setup

We install management server on CentOS 6.5 minimal operating system. So we first installed CentOS 6.5 minimal on the management server machine.

Configuring the network

The management server must has a static IP configuration and we assigned 192.168.1.10/24 to it. Then network configuration information in included in file /etc/sysconfig/network-script/ifcfg-eth0. We show the network configuration information of the management server below:

```
DEVICE=eth0
HWADDR=64:31:50:40:93:CA
TYPE=Ethernet
UUID=a7562c45-db04-4001-8203-68ae6654bfec
```

```
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=none
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.1
DNS2=192.168.1.1
```

At the moment, for CloudStack to work properly SELinux must be set to permissive. Selinux configuration is /etc/selinux/config.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing – SELinux security policy is enforced.
# permissive – SELinux prints warnings instead of enforcing.
# disabled – No SELinux policy is loaded. SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# targeted – Targeted processes are protected,
# mls – Multi Level Security protection.
SELINUXTYPE=targeted
```

NTP

NTP configuration is a necessity for keeping all of the clocks in your cloud servers in sync. NTP could be installed with Yum:

```
# yum –y install ntp
```

Configuring the CloudStack Package Repository

We need to configure the machine to use a CloudStack package repository. To add the CloudStack repository, create /etc/yum.repos.d/cloudstack.repo and insert the following information.

```
[cloudstack]
name=cloudstack
baseurl=http://cloudstack.apt-get.eu/centos/6/4.6/
enabled=1
gpgcheck=0
```

NFS

Our configuration is going to use NFS for both primary and secondary storage. We are going to go ahead and setup two NFS shares for those purposes. We'll start out by installing nfs-utils.

```
# yum -y install nfs-utils
```

We now need to configure NFS to serve up two different shares. This is handled in the /etc/exports file.

```
/secondary  *(rw,async,no_root_squash,no_subtree_check)
/primary  *(rw,async,no_root_squash,no_subtree_check)
```

Then create the two folders.

```
/secondary  *(rw,async,no_root_squash,no_subtree_check)
/primary  *(rw,async,no_root_squash,no_subtree_check)
```

Now you'll need uncomment the configuration values in the file /etc/sysconfig/nfs.

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Now we need to configure the firewall to permit incoming NFS connections. Edit the file /etc/sysconfig/iptables.

```
–A INPUT –s
172.16.10.0/24 –m state ––state NEW –p udp ––dport 111 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p tcp ––dport 111 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p tcp ––dport 2049 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p tcp ––dport 32803 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p udp ––dport 32769 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p tcp ––dport 892 –j
  ACCEPT –A INPUT –s
172.16.10.0/24 –m state ––state NEW –p udp ––dport 892 –j
  ACCEPT –A INPUT –s
```

```
172.16.10.0/24 -m state --state NEW -p tcp --dport 875 -j
    ACCEPT -A INPUT -s
172.16.10.0/24 -m state --state NEW -p udp --dport 875 -j
    ACCEPT -A INPUT -s
172.16.10.0/24 -m state --state NEW -p tcp --dport 662 -j
    ACCEPT -A INPUT -s
172.16.10.0/24 -m state --state NEW -p udp --dport 662 -j
    ACCEPT
```

A.1.2 CloudStack Host Setup

We show how to setup a KVM and XenServer host in this section. XenServer is naturally supported by CloudStack and the only configuration is static IP address so we do not discuss XenServer here. We just discuss how to setup a KVM host here.

We use CentOS 6.5 minimal as the operating system of host machine and CentOS has integrated KVM hypervisor. After installed CentOS, we did the same thing as we did in Management server setup:

Configuring the network

Hostname

Selinux

NTP

Configuring the CloudStack Package Repository

After that, we installed CloudStack agent on the host machine:

```
# yum -y install cloudstack-agent
```

That concludes our setup of CloudStack host.