2011

Virtualized guest live migration profiling and detection

Joey Nirschl
Iowa State University

Recommended Citation

Nirschl, Joey, "Virtualized guest live migration profiling and detection" (2011). *Graduate Theses and Dissertations*. 12055.
https://lib.dr.iastate.edu/etd/12055

Virtualized guest live migration profiling and detection

by

Joey John Nirschl

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Computer Engineering;

Information Assurance

Program of Study Committee:

Doug Jacobson, Major Professor

Tom Daniels

Tien Nguyen

Iowa State University

Ames, Iowa

2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Cloud Computing is quickly becoming a mainstream commodity. Several industry players are behind the push and hype of their cloud services. They provide the services to customers for a cost. Most information about their infrastructure is proprietary and incompatible with other provider's solutions. This is a concern for open standards and system security. Virtualization of operating systems is a component necessary to operate in the cloud. Live migration is the capability to migrate an executing virtual operating system while incurring no noticeable downtime in system operation and connectivity. Virtualization solutions have a capability to live migrate virtual operating systems between different physical systems. The capabilities of live migration will be of considerable importance to cloud operations in the future. Live migration has the opportunity to be misused by cloud providers creating additional security concerns for the customer. This research looks at the live migration process of the Xen hypervisor to determine system anomalies that occur during the transfer process. Anomalies with the network and CPU were discovered that are detectable by the virtualized operating system. Based on these findings, the anomalies are statistically measured to create a profile. Detection functions are created and analyzed for effectiveness.

CHAPTER 1. Introduction

Every decade, there is a paradigm shift in the computing industry. Starting with the use of mainframes in the 1960's, the cost of computing components drastically decreased in the 1980's giving rise to the personal computer. Networking, ARPANET, the Internet, and the World Wide Web followed. Media became portable and allowed users to take it anywhere. Today, people can connect and communicate to the Internet almost anywhere. Cloud computing leverages this accessibility to give users access to unprecedented amounts of data storage and execution of resources for their needs without requiring more than a terminal. Cloud Computing is the future according to Amazon, Microsoft, and other cloud providers. In the wake of the recession, organizations are looking to minimize their resource costs and cloud computing is being touted as the solution to the problem.

Cloud computing is a collection of joined technologies, primarily achieved through virtualization. Most large organizations have implemented virtualization technologies for years to reduce physical resource costs. Cloud computing is seen as an extension to this cost reduction. As organizations weigh the risks of joining the cloud, they recognize that they limit their control of any data put on the cloud while they remain fully responsible for its protection. Organizations have a hard time discerning how a cloud provider's underlying technologies are constructed. Most often, cloud providers are so busy trying to accrue customers and evolve their infrastructure that little thought is put into understanding the original assumptions of the individual technologies, which are violated when combined with the other systems. These violations increase the likelihood that threats will materialize [1].

Virtualization technologies have long offered solutions to migrate data between different physical hosts and customers have taken advantage of this capability in their datacenters. As cloud computing continues to evolve, eventually, customers will insist on having the live migration capability in the cloud, or providers will find the feature to useful to ignore. By creating a profile of the live migration process from the view of a virtualized operating system, ad hoc detection algorithms are derived to monitor system activity and determine when live migration occurs.

CHAPTER 2. Background and Related Work

In this chapter a formal definition of cloud computing is presented and the main technology behind it, virtualization, is also addressed. Several cloud computing frameworks will be investigated as well as the advantages and disadvantages of cloud computing.

2.1 Cloud Computing

Cloud computing is the paradigm shift of the next decade. NIST's definition is the most prevailing and dominate.

> "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three services models, and four deployment models [2]."

The essential characteristics are: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The service models include: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). The deployment models are: private cloud, community cloud, public cloud, and hybrid cloud.

2.1.1 Advantages

Most often, what makes organizations consider the cloud is the reduced cost. As customers are charged per execution-hour or gigabyte of storage, they do not need to worry about hardware maintenance and upgrade costs or the additional cost that comes with underutilized physical systems. The use of virtualization allows for easy scalability, whether by duplicating instances or by changing the amount of CPU and memory available on a virtual machine. Mobility has several advantages. The location and placement of resources in the cloud is not a factor in accessing the information. A benefit is that the execution environment and data can be placed closer to the location of highest demand. The cloud computing environment moves the administration of the physical systems to the cloud provider, creating a central administration of cloud services. This allows customer's IT departments to focus on their organizations solutions. Most cloud providers have several locations where they host customer data. This distributed approach to resources creates system redundancy. If portions of the resources go down, it will have minimal affect on the other resources.

2.1.2 Disadvantages

The biggest operational disadvantage is the lack of interoperability between providers. This has occurred mainly as organizations have built their cloud and are keeping the structure, architecture, and framework private. Even though many cloud providers' market 99% or more service availability, many applications are not well suited for use in the cloud. Two application types include those of high-availability and real-time environments. When data is stored on the cloud, there is an expectation that it is frequently backed up to alternate locations. This is not always the case. Organizations that do not have separate and distinct data back-up locations from the cloud provider alternate locations for back-up data are at risk of losing their business data and potentially customers if things go wrong. Many Service Level Agreements (SLAs) protect the cloud provider from fault, lost revenue and business, and legal action [3]. The biggest concerns with the cloud are security and privacy.

Cloud providers advertise a high level of security within their environments. There is not much a customer can do except have a contract dictating actions of the provider. Another aspect of security is customers' data is only as secure as the weakest cloud customer [4]. The provider can manage the minimum security level of the environment, but system security is also dependent on how secure the other customers are with their virtualized environments. When one of those environments is compromised or malicious, it is easier for intruders to get access to other customers' environments. There are several reasons why privacy is at risk. First, the provider or provider's employees have the opportunity to view customer's stored data. Secondly, the cloud provider has their own interests and is more likely to give data to authorities. This was the case when the FBI seized servers from a data center affecting 50 organizations [5]. Finally, a cloud provider could potentially disappear or be sold, leaving the customer in a situation where they may not be able retrieve or recover their data.

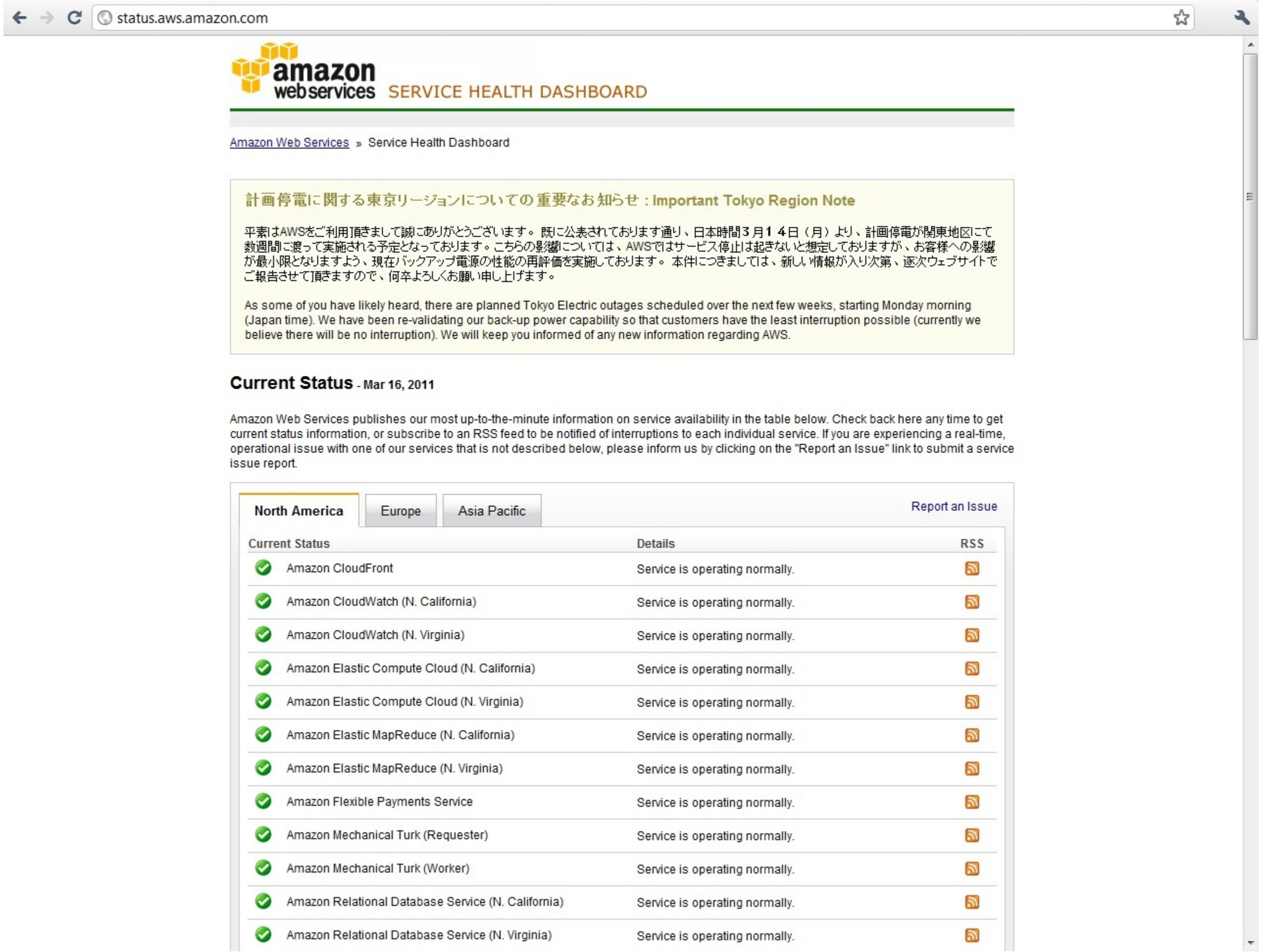
2.1.3 Cloud Computing Incidents

Two major issues illustrate the risk of moving data to the cloud, affecting the cloud's public appearance: hacking and data loss. Recently, Google's web-based email service, Gmail, had a glitch [6] that treated a small portion of existing users as new users, removing all their mail, and showing them the interface seen by new users. In this case, Google was able to restore the data to the accounts. Similarly, a Flickr user's account, which was wrongfully deleted by an employee, was not as lucky. Flickr does not have the tools to restore deleted data or accounts [7]. A hacker was able to access Twitter's confidential information, stored on Google's cloud with Google Apps, when a Twitter employee's email password was obtained [8]. These incidents show the fragility of the cloud and what happens when things go wrong.

2.1.4 Amazon EC2

The Amazon Elastic Compute Cloud (EC2) is one of eight services of the Amazon Web Services (AWS) infrastructure, their cloud computing platform [9]. EC2 provides a virtual computing environment using the Xen hypervisor. Amazon charges customers on a pay-per-use pricing model. Amazon allows customers to scale the number of running instances to meet their needs in a timely manner. With the capability to bridge with existing IT infrastructures, it is proving to be effective for many businesses with its guarantee of 99.95% uptime [10]. Currently, live migration is not possible on Amazon's platform [4]. Amazon allows monitoring of services through its Service Health Dashboard [11], see Figure 2.1.

Netflix is one of the larger consumers of Amazon's cloud platform. Amazon is a public cloud and there may be multiple tenants on the servers being utilized by Netflix. Netflix does not



SERVICE HEALTH DASHBOARD

Amazon Web Services » Service Health Dashboard

計画停電に関する東京リージョンについての重要なお知らせ：Important Tokyo Region Note

平素はAWSをご利用頂きまして誠にありがとうございます。既に公表されております通り、日本時間３月１４日（月）より、計画停電が関東地区にて数週間に渡って実施される予定となっております。こちらの影響については、AWSではサービス停止は起きないと想定しておりますが、お客様への影響が最小限となりますよう、現在バックアップ電源の性能の再評価を実施しております。本件につきましては、新しい情報が入り次第、逐次ウェブサイトでご報告させて頂きますので、何卒よろしくお願い申し上げます。

As some of you have likely heard, there are planned Tokyo Electric outages scheduled over the next few weeks, starting Monday morning (Japan time). We have been re-validating our back-up power capability so that customers have the least interruption possible (currently we believe there will be no interruption). We will keep you informed of any new information regarding AWS.

Current Status - Mar 16, 2011

Amazon Web Services publishes our most up-to-the-minute information on service availability in the table below. Check back here any time to get current status information, or subscribe to an RSS feed to be notified of interruptions to each individual service. If you are experiencing a real-time, operational issue with one of our services that is not described below, please inform us by clicking on the "Report an Issue" link to submit a service issue report.

Current Status	Details	RSS
✓ Amazon CloudFront	Service is operating normally.	▨
✓ Amazon CloudWatch (N. California)	Service is operating normally.	▨
✓ Amazon CloudWatch (N. Virginia)	Service is operating normally.	▨
✓ Amazon Elastic Compute Cloud (N. California)	Service is operating normally.	▨
✓ Amazon Elastic Compute Cloud (N. Virginia)	Service is operating normally.	▨
✓ Amazon Elastic MapReduce (N. California)	Service is operating normally.	▨
✓ Amazon Elastic MapReduce (N. Virginia)	Service is operating normally.	▨
✓ Amazon Flexible Payments Service	Service is operating normally.	▨
✓ Amazon Mechanical Turk (Requester)	Service is operating normally.	▨
✓ Amazon Mechanical Turk (Worker)	Service is operating normally.	▨
✓ Amazon Relational Database Service (N. California)	Service is operating normally.	▨
✓ Amazon Relational Database Service (N. Virginia)	Service is operating normally.	▨

Figure 2.1 Amazon Web Services: Service Health Dashboard

know what tasks the other tenants are performing, but has been able to determine that certain workloads of other tenants cause Netflix to be penalized, leading to a loss in performance. Their methods of measuring performance are not public but could include characteristics such as response time, throughput, or resource utilization. When performance reduction has reached a certain threshold, Netflix is no longer willing to remain on those servers. However, in order to get on a new server, it is necessary to destroy the current virtual machine and start a new instance, which will most often result in placement on a different physical server [12]. This is one instance where live migration would be extremely effective and useful.

2.1.5 Open Source Cloud Projects

Several open source projects have been created in an effort to expand the use of the cloud. The EUCALYPTUS Cloud Computing Platform was created by researchers at the University of California Santa Barbara to bring cloud computing to research and academia. Built to be similar to Amazon's infrastructure, it is capable of interacting with it. EUCALYPTUS has since been turned into a corporation and is no longer entirely open source [13]. OpenStack is another cloud computing project [14]. Many large industry players support the project. A big contributor to open source cloud products is NASA with their NEBULA Cloud Computing Platform [15].

2.1.6 Gartner Hype Cycle

The research firm Gartner has placed cloud computing on its Hyper Cycle list of emerging technologies for three years, since 2008. Figures 2.2 [16], 2.3 [17], and 2.4 [18] show the placement of different emerging technologies. Based on the tends shown, cloud computing will enter the "Trough of Disillusionment" in the next two years. In this phase, publicity will be less effective and the uptake by consumers will be less. Several cloud providers may fail, but it is unlikely for few of the major providers who are capable of meeting the needs of the early adopter customers.

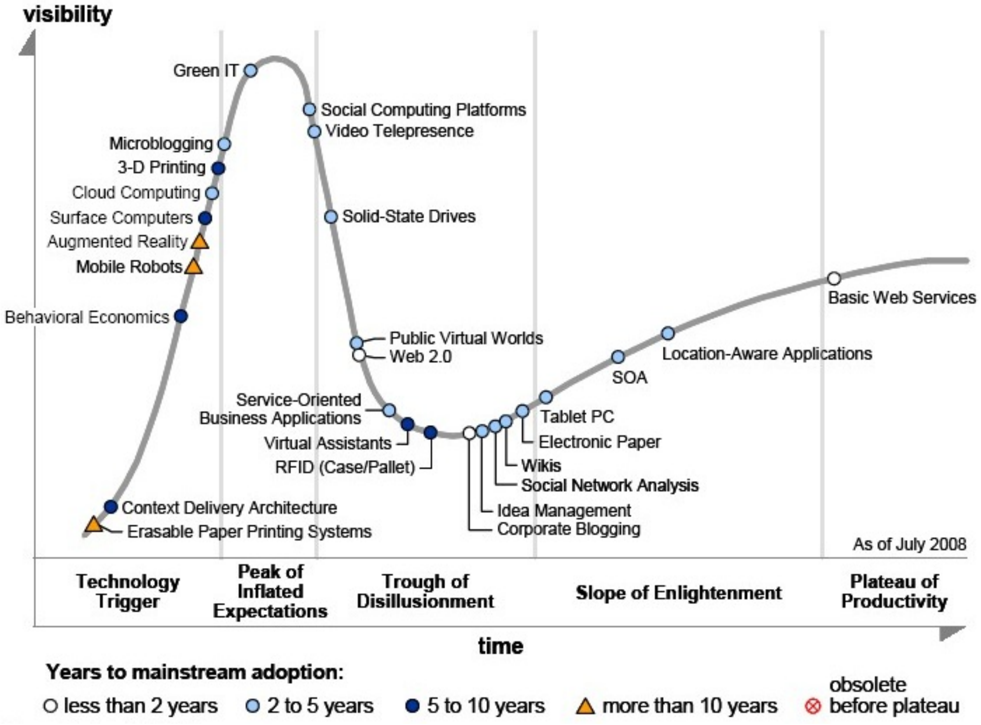


Figure 2.2 Gartner Hype Report 2008

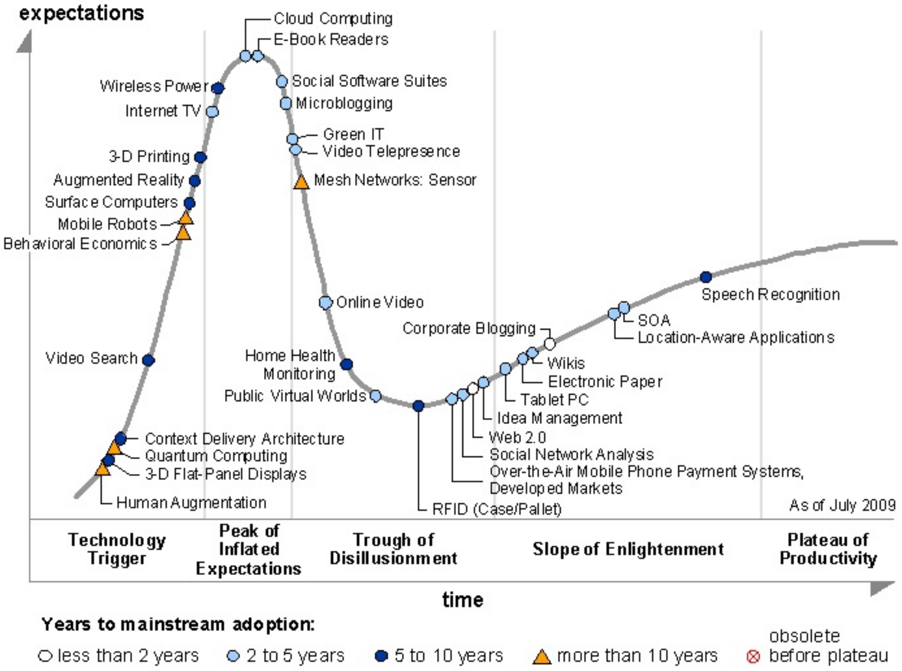


Figure 2.3 Gartner Hype Report 2009

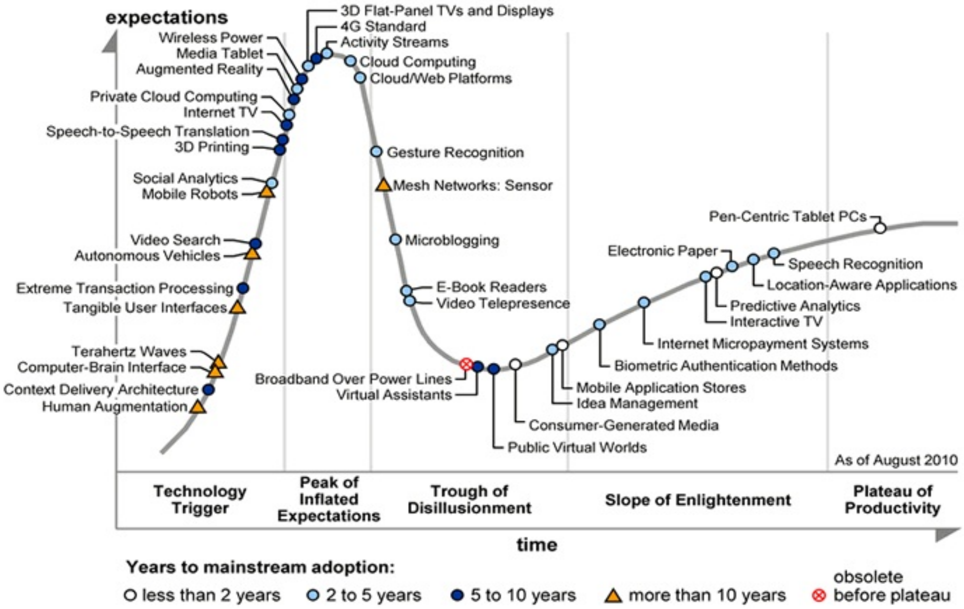


Figure 2.4 Gartner Hype Report 2010

2.2 Virtualization

Since the introduction of commercial virtualization products, virtualization has moved beyond its humble mainframe origins [19]. Virtualization is playing an increasingly dominate role in enterprise and personal computing. It provides several benefits including increased efficiency and a reduction in physical resources. This thesis uses the virtualization definition of [20]:

> “Virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.”

This emulation of physical computer attributes creates an abstraction, hiding the underlying physical implementation and functionality from the virtual environment. One advantage of this emulation is the ability to easily migrate between different physical hosts. Microsoft Virtual PC, VMware, and Xen are the dominate commercial virtualization products capable of emulating x86-based computers. VirtualBox and KVM are other available solutions most commonly associated with linux environments. Virtualization products differentiate themselves by how they virtualize the environment. VMware and Xen provide the capability to live migrate virtual operating systems through the tools VMotion and XenMotion respectively. There are several different ways environments can be virtualized. These virtualized environments are being monitored and executed by a software component that manages all virtual machine executions known as the virtual machine monitor or hypervisor. The hypervisor is also commonly referred to as a *host* (or *dom0* of Xen nomenclature). The virtual machines are referred to as *guests* (or domU's for Xen, where U is an integer larger than 0, each referring to a different virtualized *guest*). Xen and VMware have integrated into their hypervisors the capability to monitor and manage the migrations for the cloud provider. The technology was never intended to inform the *guest* due to the virtualization abstraction.

2.2.1 Full Virtualization

Full virtualization is the ability to run unmodified operating systems in a virtual machine [21][22]. Another type of virtualization is hardware-assisted virtualization. This uses a CPU execution mode feature added in recent hardware technology to simplify the work of the virtual machine manager. Although this is a distinct virtualization technique, nearly all virtualization solutions require these hardware modes to be active to make full virtualization possible. AMD and Intel provide hardware-assisted solutions through their AMD-V [23] and VT series [24] products respectively.

2.2.2 Paravirtualization

Paravirtualization provides greater performance over full virtualization [22]. Better performance can be characterized short response time, high throughput, or low resource utilization. Environments that run in this configuration require modification to kernel calls. This type of operating system is aware it is being virtualized. Typical modifications include device drivers, memory management and non-virtualizable instructions. These instructions are replaced with calls to the hypervisor, which will take care of the requests.

2.2.3 Migration Techniques

A virtual machine and its state are stored as a file and can be manipulated as such by being created, duplicated, shared, and copied [25]. When a virtual machine is being copied between different physical hosts, it is said to be migrating. The migration techniques can be classified as either static or dynamic. There are two methods for static migration and several for dynamic migration. Static and dynamic techniques differ in their process of migrating data. A virtual machine is completely inoperable while being migrated using static techniques while dynamic techniques attempt to minimize the total downtime by allowing execution to continue while the migration is occurring. The static techniques include static migration and cold migration. In static migration, the virtual machine is completely shutdown through the operating system. In cold migration the virtual machine is paused, suspended, or frozen in its current execution state. Dynamic migration, also known as live migration, transmits memory pages between the source and target hosts as the virtual machine continues executing. Live migration can benefit both the cloud provider and cloud customer. If cloud providers' have live migration enabled for them but not cloud customers, transparency of the provider may become a customer concern. The essential characteristic of on-demand self-service also comes into question, as technology already exists to perform the migration and why not let the customer have some control. A customer has a better understanding of how it will be useful for their systems. In the right environment, it would allow a customer to live migrate a machine from their data center to the cloud, something that at this point is only a dream. This does not mean that cloud providers and cloud customers have to have the same live migration rights or capabilities. It does however give customers more choices and allows for ease of transfer of the execution environment. The two dynamic migration techniques are pre- and post-copy. The difference in these techniques comes from which system (the *source* or *target*) is the primary unit of execution.

The *source* machine is the physical host containing the execution state of a virtual machine prior to the migration process starting. The *target* machine is a different physical host where the execution state of a virtual machine is being migrated too. The *target* machine will contain the entire execution state after migration is completed. The data accessed by a virtual machine does not have to be on the same host of the execution state. Both VMotion and XenMotion use the live migration pre-copy approach. See Figure 2.5 or Figure 2.6 to see how pre-copy and post-copy live migration work respectively. The live migration process is composed of three phases: setup, halt, and recover phase [26].

Live Migration Phases

Setup Phase: During this period, a target node and resources necessary for the virtual machine on the target node are reserved.

Pre-Copy Method: Transfer rounds of dirtied memory pages

Post-Copy Method: No Addition actions

Halt Phase: At this time, the source node is halted and any required components to have the target node become the primary node are transferred at this time.

Pre-Copy Method: Enters this phase when iterative transfer round limit is reached or reduced it the paged set to the writable working set.

Post-Copy Method: Transfer of minimum execution state and target host become primary machine.

Recover Phase: At this point, remaining components are migrated to the target node. The target node is started and the source node is destroyed.

Pre-Copy Method: All components now exist on the target node.

Post-Copy Method: The target node begins the execution as pages are transferred from the source.

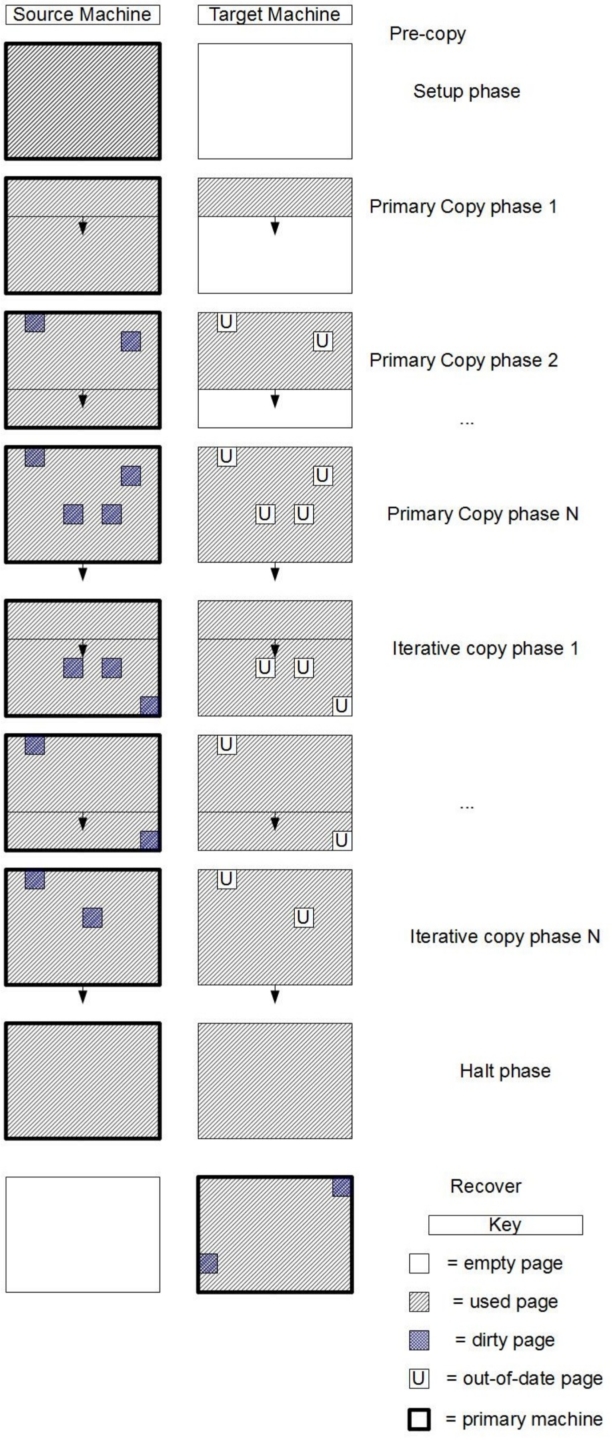


Figure 2.5 Pre-copy Live Migration Approach

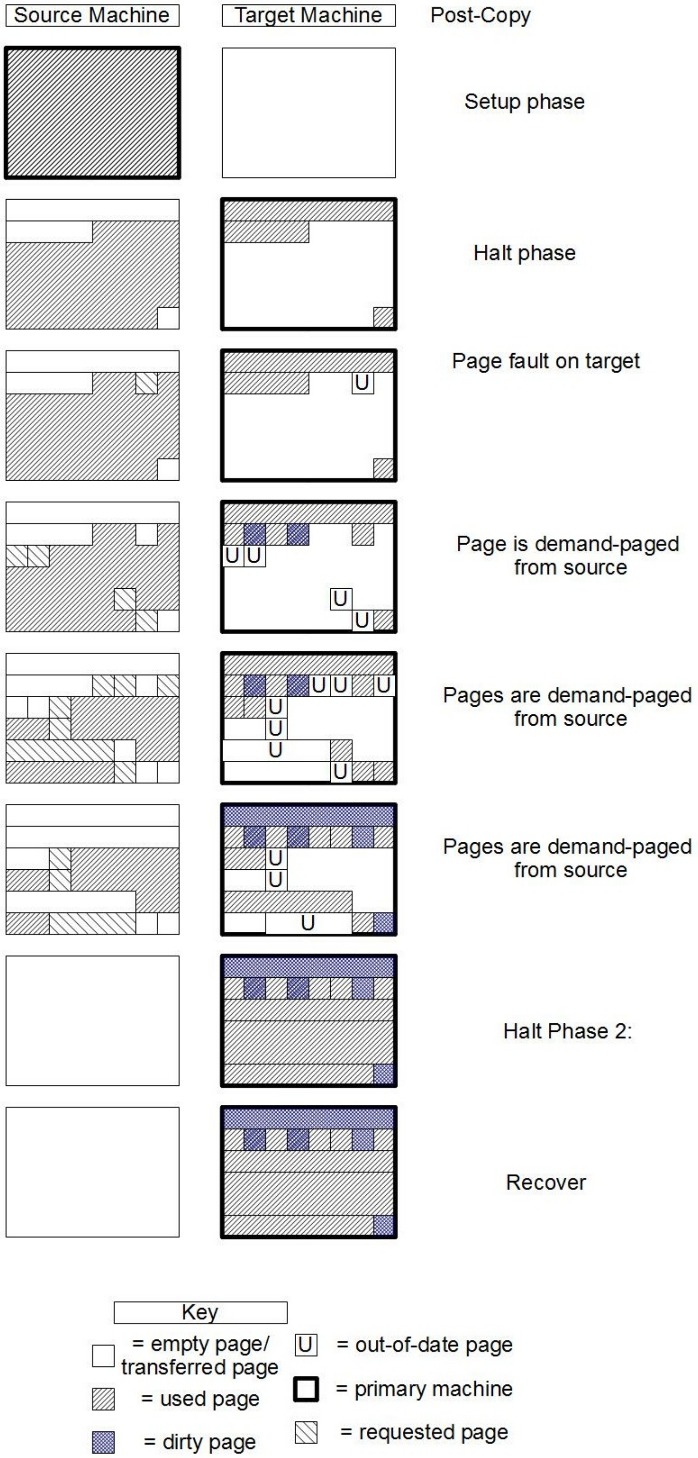


Figure 2.6 Post-copy Live Migration Approach

2.3 Related Work

This section discusses several different applications of the related technologies. These projects had significant influence in the cause and direction of this research.

2.3.1 Xen worlds: leveraging virtualization in distance education

The Xen Worlds project at Iowa State University uses Xen virtualization to create a virtual lab environment for computer security courses [27]. The use of virtualized environments allows students to practice security concepts on isolated systems. Previous research with this project has created several security assignments and their associated environments. Due to a shift in virtualization technology, research is expected to change from Xen to KVM.

2.3.2 PHANTOM

PHANTOM is a research initiative from IBM to assist organizations in managing their risk by enhancing the security of virtualized systems [28]. The technology coverage is over five domains:

- Information Security
- Threat and Vulnerability
- Application Security
- Identity and Access Management
- Physical Security

IBM's technology integrates with the hypervisor to support compliance initiatives. Integration at this level allows for comprehensive security monitoring below the virtual machines to ensure they remain secure, in addition to increasing the security of the hypervisor.

2.3.3 Trusted Virtual Environment Module (TVEM)

Much of the trust in the cloud is in the cloud provider and their software. The TVEM research project extends the trust model to allow the client to verify the trustworthiness of the host platform and virtual environment [29]. The trustworthiness is determined by *inferred* trust (the trust of other associates) and *inherited* trust (technical trust based on services and configurations). Based on the calculated trustworthiness result, a client can determine whether or not to use a service.

2.3.4 Security-as-a-Service

The cloud has created many "as-a-service" technologies, including Security-as-a-Service. Security-as-a-service is defined as an "outsourcing model for security management [that] typically. . . involves applications such as anti-virus software delivered over the Internet [30]." AEP Networks, Isheriff, McAfee, Panda, Symantec, Trend Micro, and Zscaler are several organizations offering Security-as-a-Service capabilities.

2.3.5 Empirical Exploitation of Live Virtual Machine Migration

Researchers at the University of Michigan have worked on taking advantage of weaknesses in virtualization technologies to enable virtual systems to be migrated to a malicious host. They define three different threat classes capable of exploiting the migration process; control plane, data plane, and the migration module [31]. The control plane concerns the communication mechanism that enables migration to occur. The data plane refers to the virtual machine state as it is transferred across the wire. The migration module is the component that performs the migration between the different physical hosts. This research was put to the test when it was demonstrated at the 2009 Black Hat conference by security researchers Jon Oberheide and Johanna Rutkowska. They discussed how attackers could use live migration to intercept the virtualized environment during the transfer stages [32]. Although, they do not believe the technique has been used by cyber-criminals, their demonstration showed that systems are capable of being exploited.

CHAPTER 3. Research Environment

This section will discuss the computing components involved in cloud architectures, the chosen architecture, and applicable threat models are presented. This architecture is used for Chapter 4 and Chapter 5.

3.1 Cloud Architectures

There are two main configurations for a cloud architecture, as seen in Figure 3.1 and Figure 3.2. Several distinctions can be made between the industry cloud model and the research model. The industry cloud model is much larger in scale. For each service provided there are potentially a large number of dedicated machines dispersed wherever the cloud provider desires. These services and the data the use are connected by either Fibre channel or iSCSI communication channels to Network Attached Storage or Storage Area Networks. The cloud provider has several front end applications that allow customers the ability to manage components of their virtual machines including starting, stopping, amount of memory and CPU's per instance, number of instances, and payment options for their usage of the services. Behind this front end service is automated services that allocate, deallocate, and reserve system resource, and monitor and charge for for system usage. Currently the research model is less scalable with a focus on virtual machine execution services. The research test bed demonstrates the capabilities of the cloud using two systems sharing data via a Network File System share and communicating over Ethernet.

3.2 Test Environment Architecture

In order to have the most visibility and control of the the environment, the NFS cloud method was used (see Figure 3.2). Red Hat Enterprise Linux has good documentation on how to set up this environment [33]. There are two systems, *host1* and *host2*. *Host1* is the host containing the NFS share. The systems have the following configuration:

- Red Hat Enterprise Server release 5.6
- 2.6.18-238.5.1.el5xen kernel
- Two dual-core Intel Xeon processors at 1.6GHz
- 15 GB Memory
- 1 TB Hard Drive
- Libvirt Version: 0.8.2
- Xen hypervisor: 3.1.0
- Intel CPU virtualization extension (vmx) enabled

The migrated *guest* used to gather data had the following configuration:

- Red Hat Enterprise Server release 5.5
- 2.6.18-194.el5 kernel
- 1 virtual CPU
- 512 MB Memory
- 10 GB Hard Drive

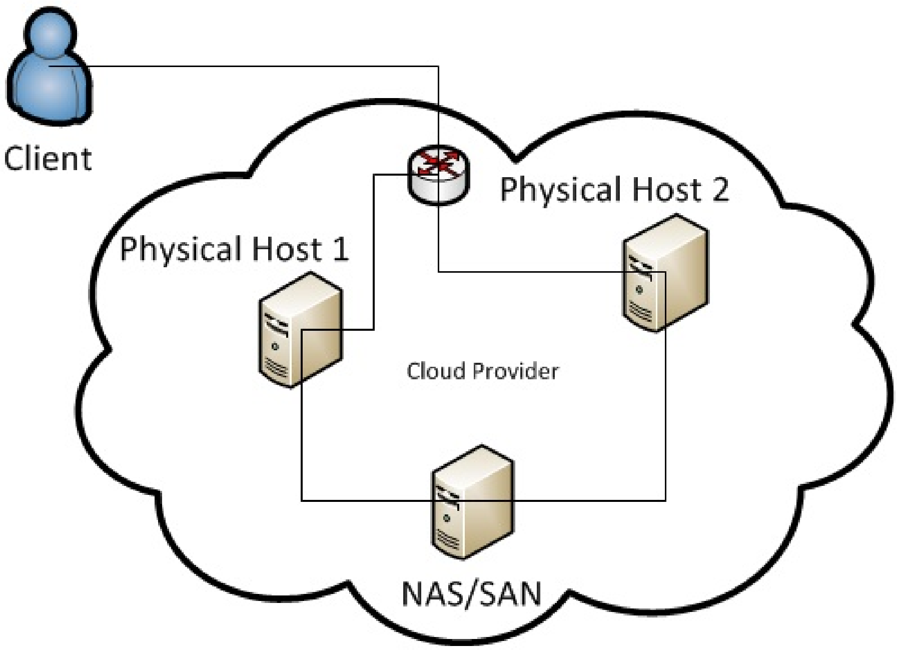


Figure 3.1 Industry Cloud Setup

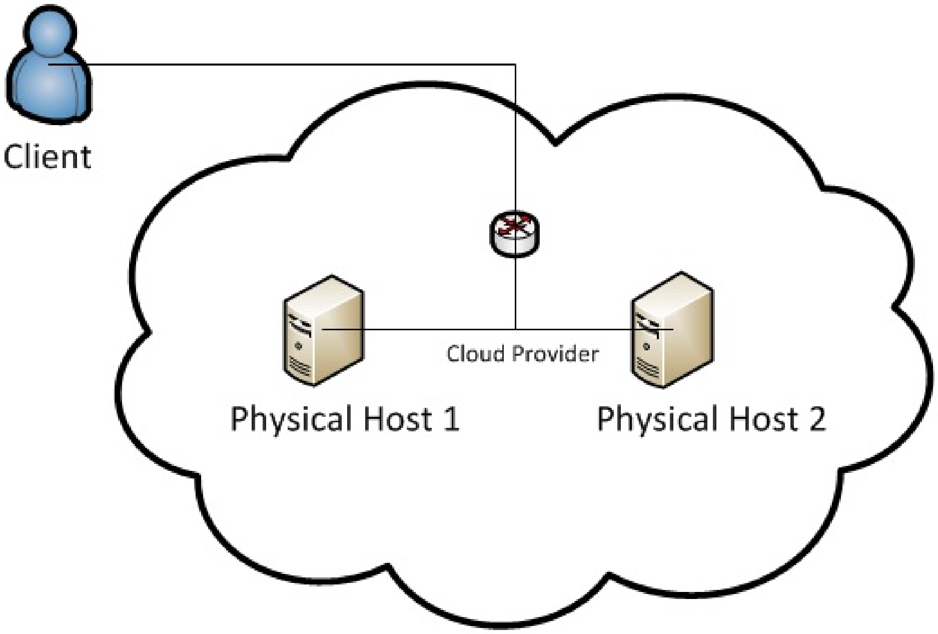


Figure 3.2 Cloud Research Setup

3.3 Threat Models

Cloud providers have control over their infrastructure and customers data. They own and administer the infrastructure, how it evolves, and specifics of where a customer's data resides (they may be given a general area or region). It is necessary to watch the watchers. This research assumes the cloud provider is a threat. From this perspective, there are two applicable threat models:

> Threat Model 1: The cloud provider is live migrating a customer's virtual machine between two physical machines. The cloud provider is not trying to hide the action besides not informing the user.

> Threat Model 2: The cloud provider is live migrating a customer's virtual machine between two physical machines and is deliberately trying to hide the migration process from the customer.

This research refrains from addressing cloud provider motives for the live migration and focuses instead on the act of live migration and whether or not the customer is informed as the matters of importance. The first threat model can then be described as an uninformed live migration, while the second model is a deceitful live migration. Both threat models can use the same technique to detect live migration.

3.4 Repercussions

Whether or not live migration is occurring can reveal if the virtual machine is being introduced to additional vectors of attack by other methods external of the cloud provider's action. These external threats can be classified into control plane, data plane, or the migration module. Refer to Section 2.3.5 for more information on these threats. Live migration may also indicate execution resources are being separated from data resources. This added separation can decrease the response time and throughput. Data or system accessibility may also be reduced or fail. These performance reductions increase the cost of utilizing the service while reducing the benefits.

CHAPTER 4. Investigation

This section investigates the different properties of a virtual operating system and how migrations or the cloud provider's influence of the system may change the properties in a noticeable way. These properties are investigated to determine reliable trust anchors. A trust anchor is a reliable characteristic within a system that maintains its characteristics unless disturbed by external influences causing a deviation from standard behavior. The trust anchors will be based on the inherited trust of technology, inferred trust will not be applied. Differences in full and paravirtualization occur mainly in the execution methods. The data received via these methods are the same. There is no indication of one virtualization method providing a better platform for detection.

4.1 Host Properties

The virtualization abstraction creates difficultly in gathering specific information regarding the different *hosts* where a *guest* may reside. However, there are events within the virtual environment that may indicate a change in system capabilities. When looking at host properties there are several data properties of interest found in various files throughout the system.

4.1.1 CPU Information Profiling

File '/proc/cpuinfo' contains CPU related information including number of processors, vendor, clock speed, and CPU flags taken from kernel data structures. A change in the kernel would be reflected in this file. It would be practical to give the *guest* capabilities of the current execution platform.

Signature: The migrations caused no change to this file. The properties were maintained throughout the current execution lifecycle. In order to change the properties a system has to be rebooted. In order for a *guest* to execute on a system, it has to have system properties that do not exceed those of the *host*. By giving the *guest* the properties of the least capable system, it is capable of migrating to a larger set of *hosts*. This property is unreliable as a trust anchor.

Cloud Provider Action: The characteristics maintained in this file are important and active changes seen throughout an execution lifecycle indicate worse problems. As many utilities depend on the state of the data during system boot, a change during execution would not properly be resolved throughout the system leading to new vulnerabilities and potential program failure.

Cloud Customer Resolution: Encrypting the virtual operating system and kernel is the best precaution. This would make it difficult for the provider to determine if the customer is accessing the file. It also reduces the attack vectors of the provider, requiring them to do an action at the hypervisor level instead of on the guest.

4.1.2 Memory Information Profiling

Another component of interest is file '/proc/meminfo' containing system memory utilization. The total amount of memory available is not as important a factor as the amount used. A technique called ballooning simplifies the transfer of unused memory. Considering memory is the driving factor in the migration process, it would make sense that there would be a distinct change.

Signature: The data in this file is constantly varying. No correlation was found between the migrations and the consumption of memory. The dynamic nature of this file made it an unreliable source to consider for detecting migration. This property is unreliable as a trust anchor.

Cloud Provider Action: The cloud provider can increase the amount of memory on the physical systems without the *guest* being aware. If the cloud provider were to reduce the amount below that which the *guest* expects, it would not be able to migrate to that system. If memory was removed from the machine it was operating on, which is unlikely as damage to the physical system could result, the *guest* would see a portion of memory fail.

Cloud Customer Resolution: There is not much that the customer can do. Encrypting the virtual operating system and kernel is a start, as no other solutions are available at this time.

4.1.3 System Clock

The phrase “time is relative” is especially true in virtual environments. However, accurate time is an important factor in applications and protocols, even those operating on the cloud. On physical systems, the system clock depends on the environment where the machine is operating; including temperature, altitude, pulse rate, and quality of material used in the device. Where accurate time is important, Network Time Protocol or similar protocol is typically used. Since system time is measured with the system clock, between periods of updating, there is a possibility to shift the *guests* view of time. This view may affect other system operations in unforeseen ways.

Signature: Clock Speed was maintained throughout experiments. Because cloud environments require the use of similar technologies for proper operation, a change that falls outside of false-positive categories would indicate a move of long distances. This property is unreliable as a trust anchor.

Cloud Provider Action: Although there is likelihood to change the clock rate in minute intervals, doing so could affect multiple customers depending on their interactions.

Cloud Customer Resolution: It is essential that provider's provide quality service for continued use of the services by a customer. Several legal factors (even with an SLA) may affect the provider if this sort of action is taken.

4.2 Network Properties

As discovered in the previous section, *guest* configurations were unchanged throughout the migration experiments. In order to keep the guests operating after migration, other characteristics must have changed. Unlike the *guests* characteristics, the *guest* has limited control over what is happening in the network. There is opportunity for diverse interaction with devices, which reveal much about the entities location, either through characteristics maintained by the *guest* or retrieved network data. The network must remain stable and operational or cloud applications begin to fail. This presents a challenge to providers desiring to hide their network configuration while maintaining connectivity.

4.2.1 IP and MAC Address

Both the IP and MAC address are important to remain connected to all necessary systems. Even when the *guest* is being virtualized, it makes sense for the MAC address to remain the same. It uniquely identifies that specific device. This is not necessarily true for IP addresses. Consider for example a wireless campus (such as a university, business, or community) interconnected with multiple access points. It is possible to "migrate" or move, between the access points without changing the IP address. However, occasionally an access point may require the device to release the IP address and reassociate to the new access point. A similar situation is likely to happen in the cloud.

Signature: The properties obtained during start-up were maintained throughout the current execution lifecycle. If one were to correlate a change in IP address to a migration, false positive results would occur. Limitations of the test environment may have led to these results. If more machines and additional subnetworks were in place, the added complexity may cause the hypervisors to react differently. This is not a reliable trust anchor.

Cloud Provider Action: A provider is likely to use Dynamic Host Configuration Protocol (DHCP) for IP address distribution considering the variability of machines in the cloud at any instant. A lease time of a specific duration could be set that hides the migration. The use of a NAT and private IP addresses within the cloud infrastructure would lead to the customer always seeing the public IP while the private IP is also hidden from the *guest* by the *host*. The use of a VLAN would add to the difficultly of the customer's detection.

Cloud Customer Resolution: A curious cloud customer could attempt a series of DHCP release commands to determine if the *guest* receives a new IP address. This could help in identifying a *host* change as that *host* may be configured to a different server or subnet by revealing different network configuration information. If a NAT is used, the *guest* could communicate the private IP through a secure channel to an external source. This source would track public and private IP addresses over time and signal when there is a concern.

4.2.2 ARP Table Profiling

The ARP table reveals a lot about the local network configuration. Specifically, what other machines are talking on the local area network. ARP works by sending a protocol packet on the wire to let other systems know a devices IP to MAC address mapping. This information is accessible via the ARP program or stored in the file "/proc/net/arp" (on RHEL 5).

Signature: In the default Xen network setup, the *guest* only sees one entry, that of the *host* it resides on. For both machines a private IP address network is used. The *host* acts a router for the *guest* through use of a virtual bridge. Even with multiple *guests* on a *host*, there was difficulty communicating between them. During a migration, the MAC address of the *host* would change. Migration revealed that the MAC address was random and multiple *guests* could have a different MAC while residing on the same *host*. The environment setup limited the the number of *hosts* and *guests* on the network, so full discovery of the extent usefulness has not been studied. However, given the setup of the cloud and other network systems (a user would expect these to be static), the characteristics found were unusual enough to be used to detect the migration. This is a reliable trust anchor.

Cloud Provider Action: A possible action to defeat the effectiveness of this approach is to create false positives, changing the MAC address as viewed by the *guest*. This adds complexity as a gratuitous ARP is required potentially affecting other systems.

Cloud Customer Resolution: It is in the best interest of the provider to keep network connections alive, the customer is at an advantage as a provider's actions could affect communication between *guest* and *host* if done excessively.

4.2.3 Network Activity

The *tcpdump* program was used to gather data about the surrounding environment. On several occasions, the tcpdump failed during live migration giving as an error the network was down. This anomaly became of particular interest, however, it was hard to reproduce and it did not happen during all migrations (it did not occur at other times) and the cause of the issue is still unknown.

Signature: When the tool detected that the network went down, the *guest* appears to be being prepared for the halt phase. Although there is a high probability of false negatives, for purposes of detecting migration, this technique is considered a reliable trust anchor.

Cloud Provider Action: Use of a migration method that does not cause a system to detect a fake network outage is required. The use of the pre-copy migration method is a possible candidate for revealing halt phase. The provider benefits because the technique is not 100% effective.

Cloud Customer Resolution: This technique has false negatives, however when it does happen its cause is either migration or network failure. A cloud provider would not want to cause undue network failure, so migration is most likely what is happening when it is detected.

CHAPTER 5. Implementation

This research is a side project of the Iowa State University (ISU) Xen Worlds project. Recent technological advancements are causing the ISU Department of Electrical and Computer Engineering to play catch up in teaching virtualization and cloud computing. The Xen Worlds project is a perfect platform to build these solutions. This chapter builds on the previous chapter by experimenting with one host property and one network property. The host property looks that the CPU throughput while the network property characterizes the network based on packet response delay. Their statistical properties are measured to create ad hoc detection algorithms. These algorithms are then scrutinized to determine their effectiveness.

5.1 Migration Monitoring

The Xen dom0 stores its log information in '/var/log/xen/xend.log,' containing migration as one of many characteristics. Both *source* and *target* machines log distinct messages containing start and ending times. This information was collected to determine the true migrations in addition to their durations. Other notable information recorded include the number of iterations of transferred memory and when a *guest* is suspended and restarted during the transfer phase. Baseline measurements were taken of the migration process to determine duration and transfer characteristics. The measurements were performed when the dom0 and domU were idle and there was no unnecessary network activity. Table 5.1 shows data collected over a period of 20 migrations. The results show a consistent migration period where the time recorded by the *source* machine (migration average of 48.15 seconds) was greater than the *target* machine (migration average of 47.36 seconds). The overall migration average was 47.75 seconds. This discrepancy comes from the migration process, where it starts sooner on the source and only ends on the *source* machine once the migration is a success and it is notified by the *target* machine. *Note: All systems used NTP to ensure time accuracy.*

Migration Number	Source Machine	Target Machine	Source Duration (seconds)	Target Duration (seconds)
1	host1	host2	48.38	47.47
2	host2	host1	48.46	47.58
3	host1	host2	48.10	47.26
4	host2	host1	48.01	47.23
5	host1	host2	47.84	47.25
6	host2	host1	48.43	47.56
7	host1	host2	47.96	47.09
8	host2	host1	48.13	47.37
9	host1	host2	47.98	47.16
10	host2	host1	47.90	47.14
11	host1	host2	48.12	47.69
12	host2	host1	48.31	47.45
13	host1	host2	48.04	47.22
14	host2	host1	47.99	47.17
15	host1	host2	47.97	47.12
16	host2	host1	47.95	47.16
17	host1	host2	49.07	48.43
18	host2	host1	48.11	47.26
19	host1	host2	48.26	47.43
20	host2	host1	47.98	47.18

Table 5.1 Base Migration Measurements

5.2 CPU Throughput Profiling

An application type that exists in the cloud focuses on data computation where throughput (number of instructions executed in a time interval such as a million instructions per second [MIPS]) of the CPU is important. Memory is affected during the transfer process but no research has been done to determine how the CPU is affected. Memory and disk may be comprised of many hierarchies of different size and speed. They are built to support cache misses (where it takes additional time to retrieve data from a different hierarchy layer). The CPU is affected by the response times of memory and disk accesses. It is unknown how it behaves during the live migration process. In the simplest data computation case, memory and disk access are minimized. This research looks at this simple case.

5.2.1 Data Collection

The method to collect data from this experiment was to record the duration it took to perform a series of simple CPU events. The execution of these events is meant to be predictable and reliably complete in the same time duration. For example, consider the MIPS ISA addi instruction (syntax is “add \$d, \$s, \$t”). The instruction is capable of being executed repeatedly and consistently complete in the same duration. The *guest* used a C program compiled in GCC. The program contained an infinite busy-loop and counter, capturing and printing the change in system time required to execute $4x10^8$ instructions. Use of memory is minimized to remove any influence it may have on CPU execution.

5.2.2 Baseline Characteristics

The baseline data was collected from the *guest* for a period of 15 minutes. The results can be seen in Table 5.2. The normalized baseline data reflects a normal distribution. There are several measured data points above the execution duration of 1.64 seconds. These times reflect independent execution cycles affected by external influences that are still undetermined. However, this may include context switching for the execution of higher priority processes. In an effort to simulate a user level process, no attempt was made to alter the programs default priority.

Measure	Baseline	Non-migration	Migration
	(seconds)	Interval (seconds)	Interval (seconds)
Minimum	1.48	1.42	1.45
Maximum	1.74	2.13	5.23
Mean	1.55	1.55	1.63
Median	1.55	1.55	1.58
Mode	1.48	1.55	1.58
Standard Deviation	0.04	0.03	0.27

Table 5.2 Throughput Characteristics

5.2.3 Migration Characteristics

Information regarding the migration characteristics is seen in Table 5.2 or Figure 5.1 (outliers removed). Figure 5.1 shows the relationship between the non-migration and migration phases. The migration and non-migration intervals best approximate a normal distribution. There were several outliers such as a maximum of 5.23 seconds to complete execution. Ignoring those, there is considerable amount of overlap between the two distributions. Performing a two-sample Kolmogorov-Smirnov test reveals that the null hypothesis (both samples are drawn from the same distribution) cannot be rejected. The asymptotic p-value for the data set was 0.3874.

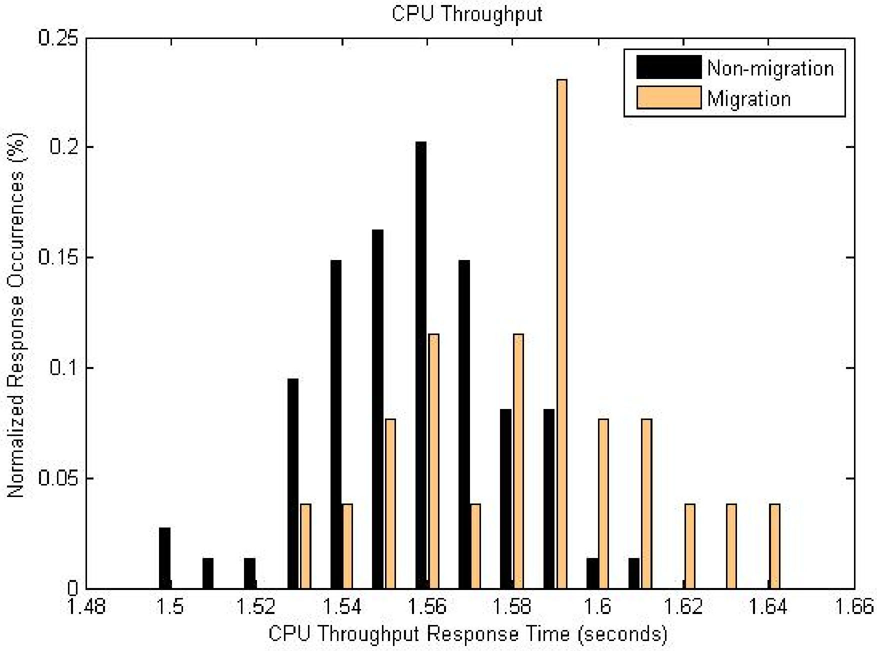


Figure 5.1 Normalized Throughput Response Times

5.2.4 Detection Algorithm

This algorithm will keep a history of 3 minutes of collected CPU through data. The history will contain the duration it took to execute the known set of instructions. This length is needed to get a the most accurate measure of mean and standard deviation. It also reduces the chance that migration data will fill the entire data set. A measured data set (10 additional data points) are collected and measured against the history. The mean and standard deviation of the history and measured data set are calculated. If the measured data sets mean is outside the range of the histories standard deviation from its mean, a migration is occurring and the system is alerted. The next two measured data sets are ignored. If the data set following the ignored data sets has a mean within the standard deviation of the initial data set, all the captured data sets are moved into the history. Otherwise, if the initial data set was within the range calculated by the histories, it is incorporated into the history. The next gathered data set will be the initial data set in the algorithm and the process repeats itself.

5.2.5 Results

Initially ignoring data sets overcomes the issue of poisoning the history. However, this data is incorporated after it has been maintained for several data sets, once it has determined that a migration did not occur but the activity is changing from other external forces. The solution appears to be effective and migrations are capable of being detected.

Cloud Provider Action: To ensure inconsistent results, it is best to slow the CPU's execution speed during the non-migration phase. This will reduce or eliminate the difference in the mean between migration and non-migration intervals to an undetectable level.

Cloud Customer Resolution: In the virtual environment the customer has a lot of control. When solutions are being created, it is necessary to build them to reliably and consistently complete on the same time. If this is possible on a users personal machine, there should be consistency in the cloud.

5.3 Network Profiling

As discovered in Section 4.2, the network can be very revealing. In addition to monitoring the view of the network, measuring a networks operation is also possible. Cloud provider's claim little or no down-time but as discussed in Section 2.2.3, the live migration methods may reveal that a migration is occurring.

5.3.1 Data Collection

A simple program reads input from *stdin*, records the system time the input was received before writing the time and input data to a file. For this experiment, the ping program is used as a means of determining variability in the network. The cloud provider is still the threat. To reduce the influence of the provider, computing resources outside the cloud environment are accessed with ping. Ping is useful as it records the time it takes to hear a response from the destination device. Ping uses Internet Control Message Protocol (ICMP) echo request packets and measures the time from transmission to when a response is received. Sources were unaware of their assistance in the detection scheme. This method reduces the memory imprint to a level that does not affect the results.

At no time during the data collection process did ping encounter unresponsiveness (Destination Host Unreachable) on behalf of the destination device. If it were to have happened, a single result could be ignored, as each echo request/reply is independent. However, a large number of unreturned requests could indicate a migration.

5.3.2 Baseline Characteristics

Baseline data, as viewed by the *guest*, was collected for a period of 15 minutes. The website www.google.com (74.125.225.19) was used as the destination for the metric. The traceroute program revealed there are 10 hops between the *guest* and the source. The first hop was recognized as the dom0. Characteristics of a normal distribution can be seen in the output. Data was recorded on intervals of a second. This is a side effect of the ping reports its findings.

Measure	Baseline (ms)	Non-migration Interval (ms)	Migration Interval (ms)
Minimum	26.30	26.70	26.90
Maximum	29.60	72.70	331.00
Mean	26.46	28.04	276.19
Median	26.40	27.00	286.00
Mode	26.40	27.00	286.00
Standard Deviation	0.15	2.83	11.73

Table 5.3 Ping Response Characteristics

5.3.3 Migration Characteristics

There were several interesting characteristics introduced when migration occurred. Figure 5.3 baseline data plotted with migration data. Timing is not reflected in the Figure. First, the migration process increased the transmission duration by at least a factor of 10. Secondly, there was a duration of 20 to 40 seconds (depending on the migration) where no ping activity occurred. This was located at the end of the live migration (indicated by the large time intervals) and the return to normal baseline levels. This is unusual for the ping utility as outside of that time period it recorded on a second interval. This is revealing as it indicates the halt phase of the live migration process. Finally, the ping process stabilized quickly after migration was complete. Migration characteristics are shown in Table 5.3 and Figure 5.2. Figure 5.3 shows a ping response time of a complete live migration compared to baseline measurements over a similar period of time. Performing a two-sample Kolmogorov-Smirnov test reveals that the null hypothesis (both samples are drawn from the same distribution) cannot be rejected. The asymptotic p-value for the data set was 0.2928.

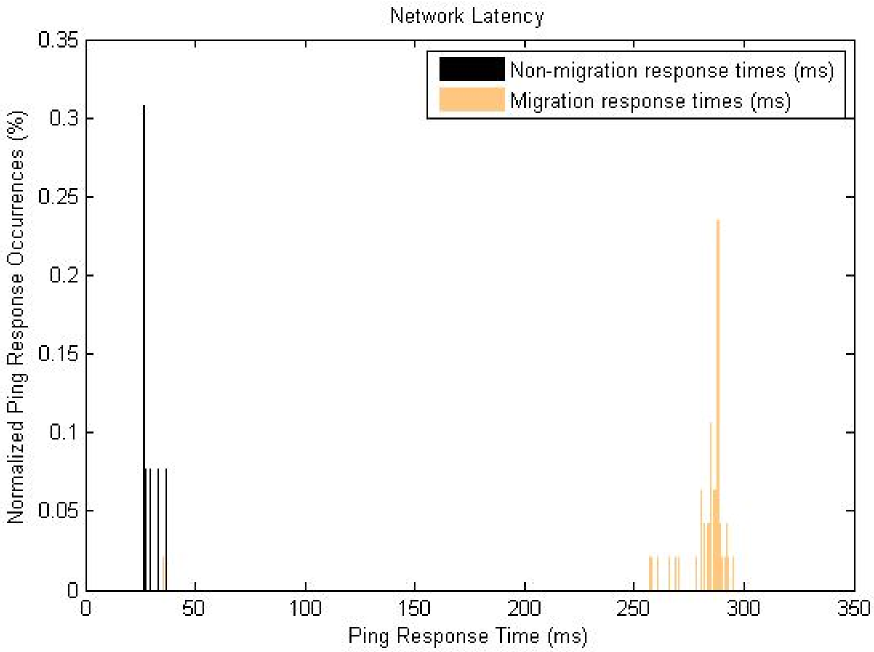


Figure 5.2 Normalized Ping Response Times

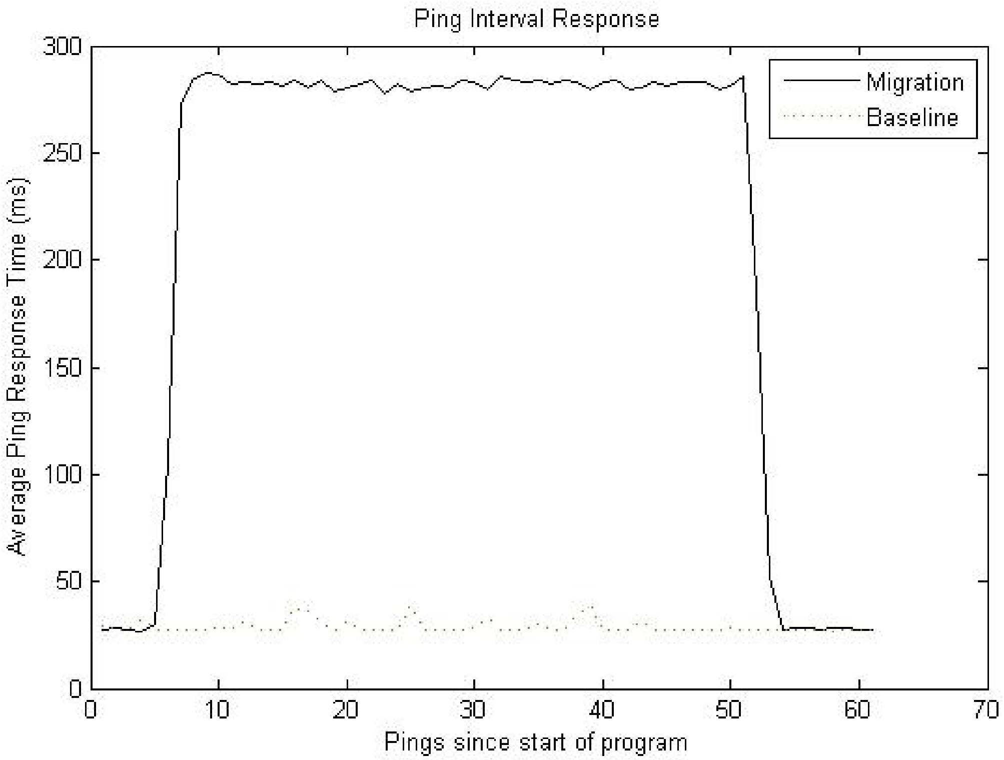
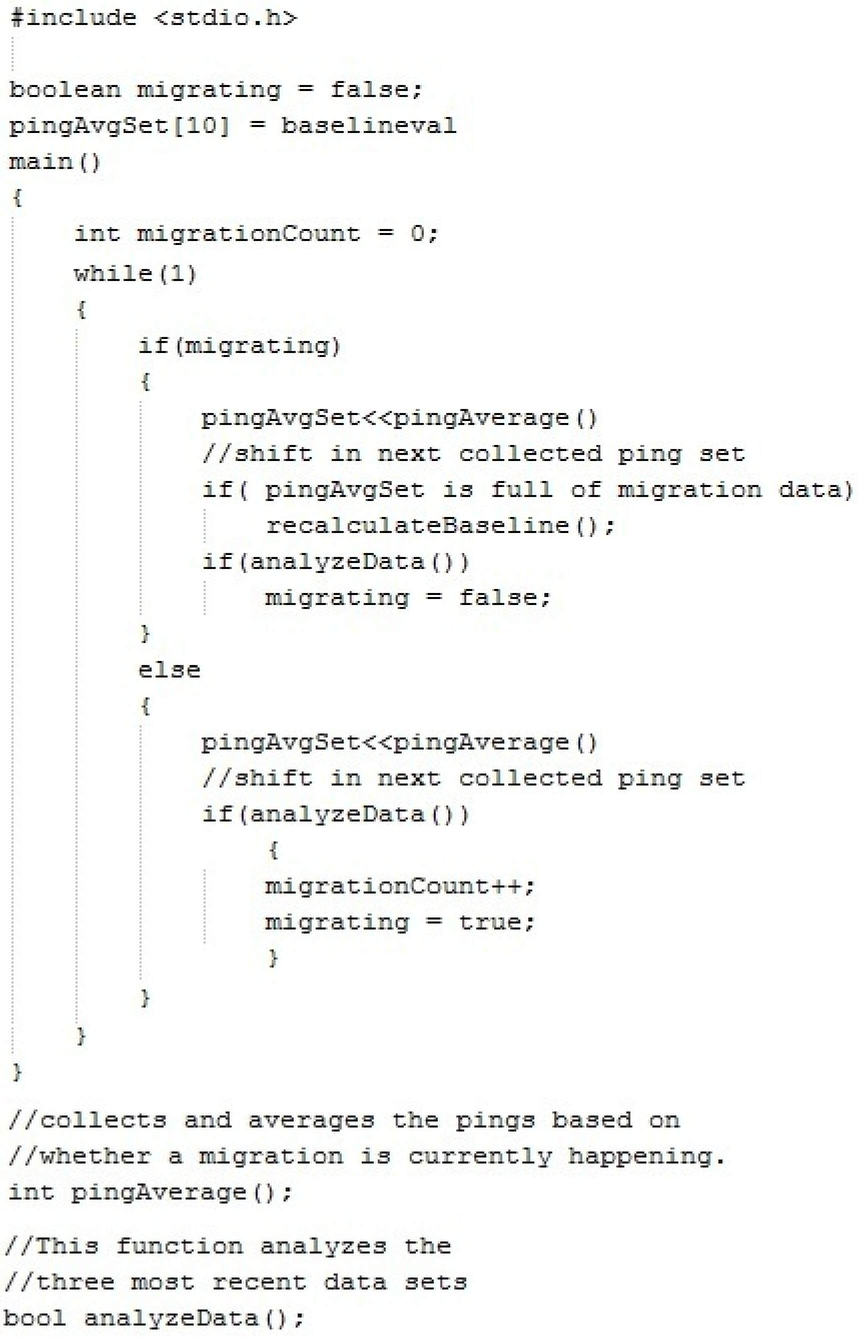


Figure 5.3 Ping Interval Response

5.3.4 Detection Algorithm

The components used for this algorithm are mean, variance, and down-time. Down-time is time when an action should have been occurring, however it was unable to occur, mimicking what happens to ping at the end of migration). A history of the 20 most resent data actions is kept at all times. These are used to calculate the mean and variance, the down-time is ignored when calculating these values. The down-time is an obvious factor that migration is occurring. A down-time of more than 7 seconds is flagged as a migration. The algorithm is described as follows: Set *migrationCount* to zero and *migrating* watch boolean to false. A history would be kept by initializing an array of size 10 with baseline measured data. Until the program is executed the following would happen. If a migration had not been detected (*migrating* is false), every 2 pings would be averaged before being put into the history. The three most recent history elements are compared and if the difference is larger than twice the standard deviation, *migrating* is set to true. If migration is occurring, every 7 pings would be averaged before being put into the history. The three most recent history elements are compared and if the difference is larger than twice the standard deviation and the most recent value is smaller, *migrating* is set to false. See Figure 5.4 for algorithm pseudo code.



```
#include <stdio.h>

boolean migrating = false;
pingAvgSet[10] = baselineval
main()
{
    int migrationCount = 0;
    while(1)
    {
        if(migrating)
        {
            pingAvgSet<<pingAverage()
            //shift in next collected ping set
            if( pingAvgSet is full of migration data)
                recalculateBaseline();
            if(analyzeData())
                migrating = false;
        }
        else
        {
            pingAvgSet<<pingAverage()
            //shift in next collected ping set
            if(analyzeData())
                {
                migrationCount++;
                migrating = true;
                }
        }
    }
}
//collects and averages the pings based on
//whether a migration is currently happening.
int pingAverage();

//This function analyzes the
//three most recent data sets
bool analyzeData();
```

Figure 5.4   Algorithm Pseudo code

5.3.5 Results

Data was collected from a variety of labs around the Iowa State University campus and multiple sites were used as destinations. The effectiveness of the detection algorithm was worse when the sample set had a small standard deviation. Added variability in the network, even a random second or two would cause a migration alert to be triggered. Most sample sets that had a standard deviation of between 1 and 3 seconds did not experiences any difficulty. Another issue with the algorithm is switching destination devices. This could cause the transmission duration to change outside the currently measured bounds inaccurately triggering a migration. This could be fixed by reinitializing the history with baseline data associated with the new entity before measurements are taken.

Cloud Provider Action: The provider could keep the network delay at or above the migration delay. From the view of a customer it would create many false negatives.

Cloud Customer Resolution: With no control in how the network is operated, the best a customer can do with network properties is monitor them. The provider will do what they can to minimize cost. It would take additional resources and time to delay many devices on the network (it is unlikely that a provider would only perform the act of hiding migration from one customer).

5.3.6 Potential Implementation

It would be ineffective to constantly use the ping utility. It is necessary to build around a mechanism that is required to be used. The *guest* needs to communicate with the distant data storage device when reading or writing data to its disk. A monitoring program can wrap around this process gathering the statistical data, reporting observed violations to the *guest* and/or an alternate source.

5.3.7 Alternate Scenario: Client Response

Although this component was not studied due to time constraints and technical difficulty, it brings up a complementary view of the recent the network profiling discovery measured through the perspective of the *guest*. There are times a customer may be interacting with their cloud services and a potential migration may occur during that interaction. Based on the previous results it would follow that the client would encounter some anomalies. It would be interesting to see the clients perspective of a migration 0 correlates to the *guest*.

5.4 Summary

The algorithms described in this section are ad hoc. More rigorous approaches are needed. These approached would not ignore history data. The environment these algorithms were constructed and tested under was idea. There was minimal noise introduced on the test systems. If noise was introduced it could affect the algorithms detection capability leading to different false positive and false negative rates. These algorithms also took into account that migration durations would last between 45 to 65 seconds. This time duration was true for this environment but may not be true for a different environment.

CHAPTER 6. Conclusion

As discussed in Chapter 1, the cloud is Pandora's box and there is a need for research of the security of available technologies. Based on the investigated characteristics, detection is rather difficult with many characteristics experiencing false positives or false negatives. The affect of which was studied in the characteristics of CPU throughput and network latency. The network characteristics were the most successful in identifying the live migration process. This thesis looked at security elements of virtualization and how changes may indicate a live migration has occurred. This research provided the following contributions: building a test bed to simulate a cloud computing environment, addressing the need for novel security techniques in the cloud by investigating virtual machine and network properties for abnormal behavior, creating ad hoc algorithms to detect live migration. In the future, additional elements worth investigating include migrating between different time zones, under variable bandwidth conditions, over longer distances, how other programs are affected by the migration process, and the live migration process of other virtualizations solutions compares the Xen. As well as actual implementation and testing of these characteristics on a third-party cloud computing providers environment.

As the Xen Worlds project continues to evolve, so will the need for understanding virtualization, cloud computing, and the security implications of connecting new technologies. Future research would look into the live migration capabilities of VMware, KVM, and other virtualization solutions to add to the number of system profiles. As the ISU Department of Electrical and Computer Engineering expands its course into these new realms, new labs and environments will have to be created to support course objectives.

BIBLIOGRAPHY

[1] Shawn Hernan, et al. Threat Modeling: Uncover Security Design Flaws Using The STRIDE Approach, Nov. 2006. Accessed: Aug. 14, 2010. Available: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx.

[2] Tim Grance and Peter Mell. TNIST. NIST Definition of Cloud Computing. Oct. 7, 2009. Accessed: Dec 14, 2010. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc.

[3] Amazon. AWS Customer Agreement, Feb. 8, 2011. Accessed: March 30, 2011. Available: http://aws.amazon.com/agreement/

[4] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. 2009. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security* (CCS '09). ACM, New York, NY, USA, 199-212. DOI=10.1145/1653662.1653687 http://doi.acm.org/10.1145/1653662.1653687

[5] Rich Miller. Data Center Knowledge, Apr. 3, 2009. Accessed: Nov. 10, 2010. Available: http://www.datacenterknowledge.com/archives/2009/04/03/fbi-seizes-servers-at-dallas-data-center/

[6] Laurie Segall. CNN Money. Google nukes thousands of Gmail accounts, Feb. 28, 2011. Accessed: Mar. 4, 2011. Available: http://money.cnn.com/2011/02/28/technology/gmail_outage/index.htm?source=cnn_bin &hpt=Sbin.

[7] Laurie Segall. CNN Money. Google nukes thousands of Gmail accounts, Feb. 2, 2011. Accessed: Mar. 4, 2011. Available: http://money.cnn.com/2011/02/02/technology/flickr_deletes_account/index.htm?iid=EL.

[8] NetSentry. Decrease the Risks of Cloud Computing: Develop A Data Capture Strategy, Aug. 23, 2010. Accessed: Sept. 17, 2010. Available: http://netsentry.blogspot.com/2010/08/decrease-risks-of-cloud.html.

[9] Amazon Web Services LLC. What is AWS?, 2010. Accessed: Dec. 14, 2010. Available: http://aws.amazon.com/what-is-aws/.

[10] Amazon Web Services LLC. Amazon Elastic Compute Cloud (Amazon EC2), 2011. Accessed: Jan. 9, 2011. Available: http://aws.amazon.com/ec2/

[11] Amazon Web Services LLC. Amazon Web Services Service Health Dashboard, 2010. Accessed: Dec. 14, 2010. Available: http://status.aws.amazon.com/.

[12] Charles Babcock. InformationWeek. Cloud Connect: Netflix Finds Home In Amazon EC2, Mar. 8, 2011. Accessed: Mar. 14, 2011. Available: http://www.informationweek.com/news/cloud-computing/infrastructure/showArticle.jhtml;jsessionid=JHX2XRCUCJ0RZQE1GHPCK HWATMY32JVN?articleID=229300547&pgno=1&queryText=&isPrev=.

[13] Timothy Morgan. NASA and Rackspace open source cloud fluffer: OpenStack targets one million machine Nebula, July 19, 2010. Accessed: December 20, 2010. Available: http://www.theregister.co.uk/2010/07/19/nasa_rackspace_openstack/.

[14] OpenStack Cloud Software. Rackspace Cloud Computing. Accessed: Mar. 1, 2011. Available: http://www.openstack.org/.

[15] Raymond O'Brien. NEBULA Cloud Computin Platform, Oct. 18, 2010. Accessed: Oct. 31, 2010. Available: http://nebula.nasa.gov/.

[16] Gartner. Gartner Highlights 27 Technologies in the 2008 Hype Cycle for Emerging Technologies, Aug. 11, 2008. Accessed: Aug. 14, 2010. Available: http://www.gartner.com/it/page.jsp?id=739613

[17] Gartner. Gartner's 2009 Hype Cycle Special Report Evaluates Maturity of 1,650 Technologies, Aug. 11, 2009. Accessed: Aug. 15, 2010. Available: http://www.gartner.com/it/page.jsp?id=1124212

[18] Gartner. Gartner's 2010 Hype Cycle Special Report Evaluates Maturity of 1,800 Technologies, Oct. 7, 2010. Accessed: Oct. 14, 2010. Available: http://www.gartner.com/it/page.jsp?id=1447613

[19] VMInnovative. History of Virtualization, 2010. Accessed: Sept. 1, 2010. Available: http://www.vmtech.com.au/virtualization/history.html

[20] Amit Singh. An Introduction to Virtualization, Jan. 2004. Accessed: Jan. 11. 2011. Available: http://www.kernelthread.com/publications/virtualization/.

[21] Red Hat, Inc. Types of Virtualization: Full Virtualization. Accessed: Feb. 1, 2011. Available: http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/pr01s05.html.

[22] VMware. Understanding full virtualization, paravirtualization, and hardware assist. Technical report, VMware, Inc., 2007. Accessed: Jan 23, 2011. Available: http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf.

[23] AMD. AMD Virtualization (AMD-V) Technology, 2011. Accessed: Nov. 13, 2010. Available: http://sites.amd.com/us/business/it-solutions/virtualization/Pages/amd-v.aspx

[24] Intel. Intel Virtualization Technology (Intel VT). Accessed: Nov. 13, 2010. Available: http://www.intel.com/technology/virtualization/technology.htm

[25] Tal Garfinkel and Mendel Rosenblum. 2005. When virtual is harder than real: security challenges in virtual machine based computing environments. In *Proceedings of the 10th conference on Hot Topics in Operating Systems - Volume 10* (HOTOS'05), Vol. 10. USENIX Association, Berkeley, CA, USA, 20-20.

[26] Michael R. Hines, Umesh Deshpande, and Kartik Gopalan. 2009. Post-copy live migration of virtual machines. *SIGOPS Oper. Syst. Rev.* 43, 3 *July*2009, 14-26. DOI=10.1145/1618525.1618528 http://doi.acm.org/10.1145/1618525.1618528

[27] Benjamin R. Anderson, Amy K. Joines, and Thomas E. Daniels. 2009. Xen worlds: leveraging virtualization in distance education. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (ITiCSE '09). ACM, New York, NY, USA, 293-297. DOI=10.1145/1562877.1562967 http://doi.acm.org/10.1145/1562877.1562967

[28] IBM. IBM Bolsters Clients' Security Arsenal, Apr. 8, 2008. Accessed: March 8, 2011. Available: http://www-03.ibm.com/press/us/en/pressrelease/23833.wss.

[29] F. John Krautheim, Dhananjay S. Phatak, and Alan T. Sherman. 2010. Introducing the trusted virtual environment module: a new mechanism for rooting trust in cloud computing. In *Proceedings of the 3rd international conference on Trust and trustworthy computing TRUST*′10, Alessandro Acquisti, Sean W. Smith, and Ahmad-Reza Sadeghi *Eds.*. Springer-Verlag, Berlin, Heidelberg, 211-227.

[30] SearchSecurity.com. Definition - What is security-as-a-service (SaaS)?, Feb. 15, 2010. Accessed: Feb. 27, 2011. Available: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1381571,00.html.

[31] J. Oberheide, E. Cooke, F. Jahanian, Empirical Exploitation of Live Virtual Machine Migration, Black Hat Security Conference, Washington, DC, February 2008.

[32] Andy Greenberg. Forbes. Virtualization's Dark Side, Apr. 9, 2008. Accessed: Jan. 20, 2011. Available: http://www.forbes.com/2008/04/09/virtualization-rsa-malware-tech-virtualization08-cx_ag_0409virtual.html.

[33] Red Hat, Inc. Red Hat Enterprise Linux 5 Virtualization Guide, 2008. Accessed: Nov 1, 2010. Available: http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html-single/Virtualization/index.html#sect-Virtualization-Security_for_virtualization-SELinux_and_virtualization