The Pennsylvania State University

The Graduate School

Department of Computer Science & Engineering

THE USE OF CLOUD COMPUTING IN HEALTH CARE

A Dissertation in

Computer Science and Engineering

by

Richard L. H. Rauscher

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2015

The dissertation of Richard L. H. Rauscher was reviewed and approved* by the following:

Raj Acharya
Department Head and Professor, Department of Computer Science & Engineering
Dissertation Advisor
Chair of Committee

Padma Raghavan
Associate Vice President for Research and Director of Strategic Initiatives
Director, Institute for CyberScience
Distinguised Professor, Computer Science and Engineering
Affiliate Professor, Information Sciences and Technology

Eugene Lengerich
Professor, Public Health Science

Wang-Chien Lee
Associate Professor, Department of Computer Science and Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Attaining computational efficiency in health care is becoming increasingly important. Currently, 17.9% of the GDP of the United States is spent on health care. It is an information intense industry, and, through private and governmental incentives, is increasingly using digital information to manage care. Cloud computing and computer virtualization have complicated the decision process around how to architect solutions for health care. Health care decision makers must seek to minimize their spending while ensuring sufficient computational capacity, regulatory compliance and security. The intensity of the use of digital information in health care is expected to increase rapidly.

This dissertation makes several contributions to the fields of health care informatics and computer science. We first established motivation for studying this area by examining the attitudes of health care technology leaders throughout the United States. This research led us to believe that many health care organizations will continue to make use of private clouds for some time. Given this assumption, we examined how the predictability of health care workloads could be used to improve the use of private cloud computing infrastructures. Using historic resource requirements as a guide, we constructed an algorithm that could be used to by systems architects to make better informed hardware allocation decisions. Continuing to use the concept of predictability, we created a system that dynamically migrated virtual machines within a private cloud in expectation of increased workload. This preemptive migration based on history-generating foresight reduced the overhead associated with other trigger-based migration schemes of up to 99.6%.

The original survey also indicated that health care technology leaders were concerned about security and the changing regulations associated with data leakages. Our research focus shifted to reducing the risk of data leakage in hybrid computing clouds. We devised a mechanism that prohibited layer three network communications between different network zones to reduce

the risk of data leakage while ensuring, to the extent practical, that traditional multilevel security rules could be implemented. Finally, we proposed mechanisms required to successfully deploy health care data in a dynamic Intercloud such that data could move freely between private and various public cloud providers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

Richard Rauscher was primary author and Raj Acharya was second author on all of the papers listed below which contributed to this dissertation. Richard Rauscher performed the research that was scrutinized and reviewed by Raj Acharya.

 Chapter 5 is an extended version of the paper presented in Chapter 5.

 The project described was supported in part by the National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant UL1 TR000127. The content is solely the responsibility of the authors and does not necessarily represent the

official views of the NIH. The user reaction quantifier program described in section III was developed by Autumn Rauscher.

ACKNOWLEDGEMENTS

Completing a doctorate while employed full-time and a father of two growing daughters is not possible without the support of a large group of people. Absolutely first and foremost, I must acknowledge the contributions of my wife, Laura Carey Rauscher. She has sacrificed countless evenings and weekends to reduce my responsibilities at home. She has been my tireless editor, proofreader, and motivator. I must also acknowledge the direct and indirect contributions of my children, Zoe and Autumn. They barely remember a time when Dad wasn't working on something for school on many weekends. My parents, Richard and Patricia Rauscher, always encouraged me to pursue education and made significant sacrifices so that they could buy me my first computer at age 12 in 1981. My sister, Erica Rauscher-Johnson, who completed her dissertation before me, showed me that it was important to write something of not just volume but impact.

Of course, this journey would not have been possible without the support of my employer, Penn State Hershey. Thomas Loughran supported my initial efforts and was on my committee for several years. Thomas Abendroth provided time and, more importantly, encouragement. Rodney Dykehouse supported and motivated me to finish. Daniel Notterman encouraged and financially supported my presentations at various academic meetings. Madhu Reddy often gave me informal feedback early in my research processes which helped me to avoid several dead-ends.

My fellow doctoral students, especially Kristi Campoe and Tricia Ruma Spence, were an important source of emotional support and camaraderie.

Other members of the faculty at Penn State– too numerous to mention – provided me with advice both scientific and personal. I will also mention two close friends who traveled this road before me – Sophia Hassiotis of Stevens Institute of Technology and Richard Martin of

Rutgers were always helpful and inspirational. My professors from my previous degrees, particularly Ken Christensen and Nagarajan "Ranga" Ranganathan at the University of South Florida and Eric Allender, Barbara Ryder and the late Irving Rabinowitz from Rutgers contributed significantly to my fascination with computer science.

I would like to acknowledge the support of my doctoral committee. Wang-Chien Lee and Eugene Lengerich each contributed to my thesis with helpful suggestions and academic wisdom. Padma Raghavan, who is remarkably busy with the many administrative posts that she holds at the University, was never too busy to meet with me to discuss the direction of my research.

Finally, I can't thank my thesis advisor Raj Acharya enough. He has provided extraordinary guidance throughout this entire process. He was the first to suggest that I embark on this journey and has, on multiple occasions, been a major source of motivation and perseverance.

Chapter 1

Introduction

The term “cloud computing” has been embraced by both academia and industry. Although the term has been used to refer to everything from off-site storage to wireless networking, its primary connotation is the use of computational resources at some unknown or variable location. There are several varieties of cloud computing: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and software-as-a-service (SaaS). This compendium focuses strictly on IaaS cloud computing but some of the concepts may be applicable to PaaS and SaaS. There are dozens of technologies that have contributed to what we now call cloud computing. Cloud computing represents the convergence of several important technologies: ubiquitous availability of multimegabit Internet connectivity, mature software and hardware support for machine virtualization, rapid improvements in the performance of hardware which have outpaced the demands of most software, and “live migration” technologies. These advances have led to commercially successful cloud computing providers and have had a profound impact on reducing the costs of deploying applications. Despite the apparent advantages, healthcare providers have been slow to adopt cloud technologies for several reasons. Regulations and the risks associated with security are dynamic and are perceived to apply differently to cloud computing then to traditional enterprise computing. Health care providers often make use of infrequently updated and entrenched software that was designed well-before cloud computing was regular practice. These applications may benefit from deployment into the cloud even though they were not written with cloud deployment considered.

Throughout this dissertation, we refer to cloud computing, private clouds, hybrid clouds and the Intercloud. Unless otherwise noted, the following definitions apply.

A computing cloud is a group of physical computers and associated storage upon which virtual machines execute. We assume that the virtual machines operate as if they were running directly on physical hardware. In practice, the VMs may move from one physical machine to another without affecting the operation of the VM. A master cloud controller called the hypervisor manages the placement execution of the VMs.

A public cloud is a computing cloud that is operated by a cloud provider who sells access to cloud resources to other entities. In some ways, the public cloud providers are analogous to public utilities like telephone or electricity companies. they tend to have a minimalistic relationship with their clients in that they sell a product for a price but don't typically understand how that product is being utilized. A private cloud is an enterprise-based computing cloud that is under the ownership and operation of the user organization. Although there is no well-defined difference, in practice, private clouds are typically much smaller than public clouds. The effect of this reduced size is that private cloud owners can't behave as if they have infinite capacity as public cloud providers often do.

The Intercloud is a relatively new concept which was introduced in 2009 [1]. In today's public cloud computing, VMs can move freely and services are easily assigned within one providers' framework. Although it's possible for a cloud user to migrate their VMs from one cloud provider to another, there is no well-defined and implemented standard for doing so. However, many researchers in this area have and continue to publish papers advocating for this direction [1-6]. It has been argued that the Intercloud is to what cloud computing as the Internet is to what was once a collection of disconnected networks.

Although health care computing shares many of the characteristics of general computing, our researched focused on that which is different. Health care is remarkably predictable. Although the same can be said for other industries, we considered how this predictability could be used to more efficiently allocate hardware and manage the locations of VMs. Health care is also

noteworthy in that, as an industry, it is slow to change. We considered this when considering how older health care applications may use cloud computing. Finally, health care is heavily regulated. The regulations of health care restrict what can and can't be done and are not necessarily updated at the same pace as changes in technology.

The research in this dissertation followed a somewhat unexpected path (Figure 1-1). The original three aims were to focus on the sizing of private health care computing clouds, the placement of virtual machines within private health care computing clouds and the level of additional capacity required in a private health care cloud.

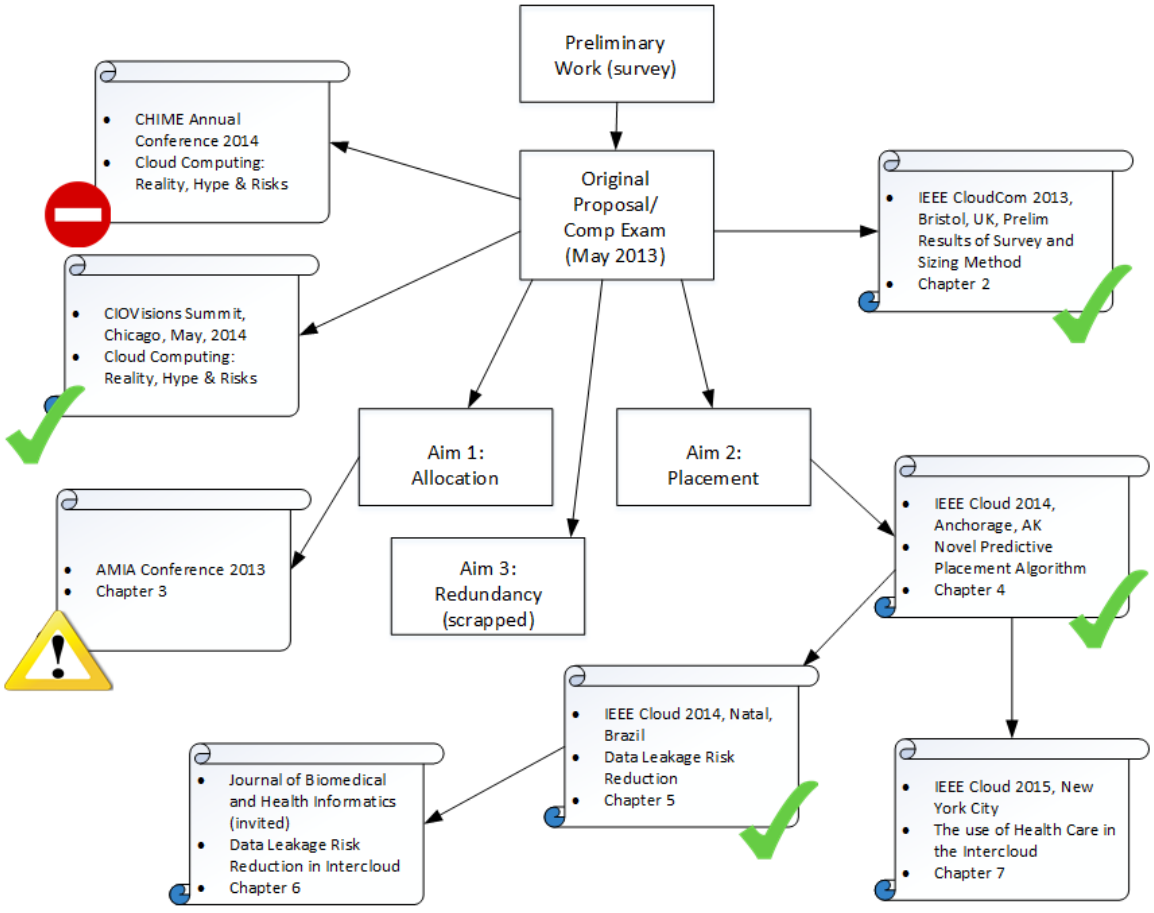

Figure 1-1. Research and Presentations Path

Based on the advice of the dissertation committee, in addition to presenting academic papers, opportunities were also sought to present to practicing technology leadership. The topics were presented in lay terms at the CIOVisions conference in Chicago in 2014. The original topic was motivated by preliminary research that surveyed members of the College of Healthcare Informatics Management Executives (CHIME). CHIME is a vetted group of primarily CIOs who lead the hospitals and other health care operations. As an outcome of the survey, I initially focused on sizing private health care clouds. The survey and the novel sizing algorithm were briefly presented at IEEE Cloud in Bristol, UK. The paper from that conference is included as chapter 2. A more detailed paper about the sizing algorithm was submitted to the annual conference of the American Medical Informatics Association where it was accepted as a paper. We chose not to present as we determined it was not a good use of finite travel resources. It is included as chapter 3. Aim 2 is the subject of the paper presented at CloudCom in Anchorage, Alaska. It is included as chapter 4. Aim 3 was abandoned as it was felt that it was not sufficient interesting for current cloud conferences. While attending CloudCom, we decided that we would begin to focus on how the Intercloud could be used to support health care. Thus, we first focused on how to reduce the risk of data leakage in health care using classic multilevel security models. This work was presented at Healthcom in Natal, Brazil and was well-received. It is included here as chapter 5. The conference organizers invited us to expand the presented paper for inclusion in the IEEE Journal of Biomedical and Health Informatics. This is included as chapter 6. Finally, we considered the architectural requirements for the use of Intercloud by health care organizations given current US regulations. We proposed the architectural elements required and closely examined the performance-based decision making regarding inter-cloud-provider migration. We proposed a method to anticipate how much time would be required to migrate from one provider to another. That work has been submitted to the IEEE 2015 Cloud Conference and is included as chapter 7.

Chapter 2 Background

Attaining computational efficiency in health care is becoming increasingly important. Currently, 17.9% of the GDP of the United States is spent on health care. It is an information intense industry, and, through private and governmental incentives, is increasingly using digital information to manage care. This chapter describes research that will examine the specific requirements for utilizing cloud computing in health care.

Cloud computing and computer virtualization have complicated the decision process around how to architect solutions. Health care decision makers must seek to minimize their spending while ensuring sufficient computational capacity, regulatory compliance and security. The intensity of the use of digital information in health care is expected to increase rapidly. One expected area of significant data growth is genomics. The cost of whole genome sequences for individuals is dropping precipitously; if current trends are sustained, by 2018, a full genome sequence will cost less than a magnetic resonance imaging (MRI) scan. Data collection will also increase when the use of wearable and implantable devices become widespread. Early research is currently being conducted that collect data from asymptomatic individuals for the purpose of detecting insipient disease states before clinical presentation [7]. These devices, with high sampling rates, may lead to longer lives but will also require vastly greater data and analytic capabilities.

Efficiently computing upon and storing massive amounts of health care data is vital. The expected exponential growth of health care data creation and the advent of cloud computing requires that we consider both when contemplating future states of health care computing infrastructures. As there remain many uncertainties regarding the legal complexities of public clouds, some health care organizations are turning towards private computing clouds. A private computing cloud is defined as a cloud where the entity is in possession of the computational

resources, administers the computational resources and the resources are under control of that entity [8, 9]. These private computing clouds must be efficiently provisioned so as to meet the needs of the health care provider while minimizing waste. Unlike public clouds, where resources are set aside for just-in-time allocation, organizations that invest in their own private clouds are less likely to tolerate large amounts of idle resources.

This research focuses on three distinct areas of private cloud computing with specific health care considerations: hardware provisioning, application organization within the private cloud and levels of redundancy. In preparation for researching these areas, the authors have considered the performance characteristics of health care information systems as well as the attitudes of health care CIOs towards cloud computing models.

Background

History

At its most basic level, this research focuses on minimizing the hardware necessary to reliably address a defined level of demand. The earliest disciplined mathematical examination of finite resource management where demand exceeded supply was performed by the Danish mathematician Agner Krarup Erlang at the dawn of public telephony [10]. Erlang's and subsequent contributions by Kleinrock [11] and scores of others to queuing theory and loss models are relevant today and used to model contentious queuing systems. Early work in performance and resource management was related to telephone and data networking but it has been repurposed for everything from operating systems to optimizing customer service in emergency rooms. Haverkort et al have examined performance issues with respect to degradable systems. They worked in the area of research now known as performability modeling [12].

Machine virtualization (and therefore, cloud technology) is not new. Although not called virtual machines, Supnik and others were using simulated computers to test new architectures in the late 1960s and early 1970s [13]. Early work in commercial virtualization took place at IBM within the VM/370 team that constructed the first hypervisor [14]. Work on virtualization and hypervisors was limited in the 1980s and 1990s and mostly confined to mainframes and executing non-native operating systems (e.g. running Microsoft Windows on a PowerPC-based Macintosh). A query of the Association for Computing Machinery's database shows that there were no titles of papers with the word "hypervisor" between 1976 and 1995. Interest in virtual machines resurfaced in the late 1990s when several events converged: operating systems, originally constructed for consumers, began to be used in enterprises; the utilization of the computational machinery of those operating systems was relatively low; and operating environments and applications became more interdependent and required custom operating system configurations for each application.

Health Care and Cloud Computing

There is no evidence of health care organizations using private computing clouds prior to 2003. We published the first description of a hospital utilizing a private computing cloud in 2004 [15]. In the United States, the set of regulations known as the Health Insurance Portability and Accountability Act (HIPAA) define how health data are managed and secured. Due to HIPAA, the use of public cloud computing for health care has been fraught with legal uncertainty. Schweitzer performed an analysis of HIPAA requirements and how they may be met to make use of cloud computing [16]. Schweitzer identified and enumerated the elements that should be part of a contract with a cloud provider and to which the cloud provider would have a legal obligation to comply. Negotiating rigorous contracts with shared liability (particularly when the liability is

not easily calculable) will indirectly increase the cost of using a public cloud service. Armbrust et al conducted an early analysis regarding the economics of using public cloud computing [17]. This report indicated that public cloud computing, if all costs were considered, was marginally more expensive than private cloud computing. Cloud providers have been unwilling with sign HIPAA-defined business associate agreements (BAAs) due to the liability associated with doing so. The January 2013 HIPAA Omnibus rules clarified the need for a BAA and the liabilities for public cloud providers who host protected health information [18]. Where there was previously a level of ambiguity regarding the need for public cloud computing providers to sign a BAA, none now exists: public cloud computing providers who knowingly host protected health information (PHI) for covered entities must sign a BAA and accept the associated liabilities. As stated previously, they may not be willing to sign these agreements and thus may not legally accept responsibility for hosting PHI. Although the number of cloud providers seemingly willing to sign BAAs is increasing [19], the individual cloud provider may not be willing to sign the particular health care entity's BAA depending upon the level of liability that is extended to the provider.

To measure the sentiment of health care IT leadership, the authors conducted an IRB approved (PSU IRB #41384) survey among members of the College of Health Care Information Management Executives (CHIME). The survey sought to understand the attitudes regarding the use of public and private cloud computing. The ultimate findings of the survey were that 63% of the CIOs surveyed were not pursuing public cloud solutions primarily due to security and legal concerns. By contrast, more than 61% were considering private cloud solutions.

Cloud Computing Performance

Cloud computing performance has been an active area of research since the earlier virtualization. Gambi and Toffetti considered models of the dynamic growth and shrinkage of

computing clouds. They indicated that traditional linear and simple queuing systems are not sufficient for modeling complex and dynamic cloud systems [20]. They proposed using Kriging (Gaussian Process Regressions) as a model of cloud dynamism. Smith considered the use of standard industry benchmarks to measure a cloud's performance characteristics [21]. Yigitbasi et al built a system for creating load and measuring resources from inside the cloud and examined the performance of several cloud models [22]. Like Gambi and Toffetti, Brebner examined the elasticity of clouds and the performance implications and characteristics of growth and shrinkage in computing clouds [23]. Duong et al considered an approach for public cloud providers to optimize their revenue while maintaining or minimally sacrificing their quality-of-service agreement with their clients [24]. They specifically focused on applications with small latency tolerances such as on-line gaming. Cecchet et al proposed a system called Dolly to manage the stateful nature of databases in consideration of dynamic growth and shrinkage in cloud systems [25]. Liu and Wee determined that no single optimal configuration for cloud services supported all types of user behavior. They therefore proposed a dynamic switching architecture which optimizes an applications use of public cloud resources depending upon resource needs. Varadarajan et al discovered a novel attack vector for stealing resources from neighboring virtual machine for the benefit of the attacker [26]. Tan examined the performance of highly parallel scientific computing problems and their performance within public computing clouds [27]. Similarly, Ostermann, Iosup et al developed a cloud performance measurement methodology and found disappointing performance characteristics of Amazon's EC2 public cloud for solving scientific computing problems [28]. Zhao examined the use of cloud computing for coordinating multi-player games. Their work suggests that gaming may achieve higher performance if cloud based resources are utilized instead of client-based resources [29]. Deng et al considered another dimension of performance. They examined the environmental impact of data centers and methods to balance carbon emissions by shifting virtual machines between data centers in

different parts of the world [30]. The VOLARE system provides a context awareness for mobile devices connecting to cloud services to offer differentiated services for varying capabilities of mobile devices [31].

Cloud Computing Provisioning

Provisioning resources in cloud computing to match computational requirements has also been an active area of research. Although provisioning goes hand-in-hand with performance analysis, performance analysis subject matter has focused on how well cloud models perform under certain application loads while the provisioning literature has focused on how resources are allocated to a cloud. Much of the research examines how clouds grow and shrink dynamically. Tan et al examined how I/O paths limit dynamic provisioning of cloud infrastructure and modeled the system using a traditional Jackson network [32]. Chapman created a new language construct and tested it in the RESERVOIR system to manage cloud elasticity [33]. From a service provider's perspective, Rao et al built a system where the cloud organization was determined by a distributed learning algorithm [34]. Rao claimed to have used such a learning algorithm to efficiently provision a cloud configuration with low overhead and few required samples. Tan et al modeled resource provisioning in a cloud using a traditional telephony/queueing system and assumed discrete Erlang admin/no-admit rules for provisioning cloud services [27]. The CloudNet system considers the resources required to migrate "live" virtual machines over wide area networks (WANs) [35]. CloudNet improves the migration of virtual machines over WANs by 65%. Bayllocator is a system that allocates RAM between different virtual machines running in a cloud based on a Bayesian analysis and prediction of memory utilization [36].

History as a Predictor of the Future

Significantly, Stokely et al examined the predictability of resource usage in a cloud environment. They found that individual behavior was difficult to forecast. However, they found that *in aggregate*, they were able to successfully forecast the future using historical information with a margin of error of 12% [37]. This research contributes to our algorithm for computing an efficient level of hardware provisioning.

Prepatory Research

Survey Results

As indicated above, we have conducted a survey of Chief Information Officers of health care entities in January 2013. The ultimate finding of the survey was that health care CIOs were much more likely to utilize private than public cloud computing technologies.

EMR Simulation and Resource Utilization

To gain insight into the performance and resource requirements, we examined the characteristics of a OpenEMR, an open-source electronic medical record system. We choose OpenEMR because the source code is available and there are no commercial constraints. Furthermore, many of the commercial database vendors prohibit or restrain the publication of benchmarking data without their consent [38].

Experimental Design

The architecture of OpenEMR is similar to the model represented in Figure 1-1 [39]. The following elements were measured with varying simulated user loads: input/output operations, CPU loads, RAM loads (excluding file caching), RAM loads (including file caching). Each server within the OpenEMR system was configured as a virtual computer on a Ubuntu 10.4 workstation with 12 processing cores and 12 GB of RAM. The hypervisor was Oracle VirtualBox 4.1.10. Each virtual machine was allocated one virtual processor. The virtual appliance (pre-configured virtual machine) was downloaded from the main OpenEMR website. It was tested and shown to be functional and subsequently cloned and configured such that the original virtual appliance represented the web and application server and the second virtual machine managed only the database. The two parts of the application were split to facilitate finer measurement of application and database computational requirements.

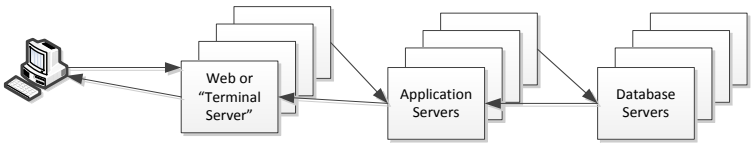

Figure 2-1. Typical EMR Application Architecture.

OpenEMR is a web-based application, thus a web load inducer was used to simulate the load of multiple users on the application. The web load inducer selected was Apache Jmeter [40] which executed manually generated traces. Jmeter was run from the host operating system. To avoid extraneous traffic, the guest operating systems which housed the OpenEMR framework were technically limited such that they could only communicate to each other and the host machine. Wherever possible, the traces used to drive the simulations were created by recording "normal" EMR activity. This was done by recording HTTP transactions through proxy server

that was built into Jmeter. The OpenEMR software was adjusted to use plain-text transfer methods (as opposed to the default of SSL) so that the transactions could be observed by the proxy server such that they could be repeated. The recorded activities were replayed to simulate human behavior. The data collected were CPU load average, memory utilization, page fault rate (to ensure other statistics weren't confounded by page faults), I/O operations per second and network bandwidth utilization. The experiment was run with 1, 5, 10, 15, 25, 50 and 100 simulated users.

EMR Simulation Results

As the number of users increased, the CPU load average of the database server increased linearly. When 100 users were simulated, the load average on the database server was sustained at about 90. The CPU load average of the application server was minimal and always less than one. The network utilization of the database server and the application server were similar with the application server using slightly more overall bandwidth. This is to be expected since the application server was communicating with the simulated clients and the database server. Both increased linearly with increases in simulated users and peaked at about 12 Mbits/sec. I/O operations per second were nearly zero on the application server and varying linearly with the number of simulated users on the database server. Page faults were absent as the physical RAM allocated to each virtual machine was not exhausted. Due to space restrictions, graphs have been omitted.

Aim 1: efficient provisioning of private cloud

Problem Statement

Given that health care systems are likely to use private clouds, are predictable and have well-defined application behaviors, determine a system to efficiently provision hardware to the private health care cloud.

Motivation

The first aim of this research was motivated by a practical problem at the Penn State Milton S. Hershey Medical Center: what is the appropriate level of resources to allocate to our private cloud? The need to execute this research has been further motivated by a survey of US-based health care chief information officers and their preference towards the use of private computing clouds.

Background

This research examines private (infrastructure-as-a-service) cloud computing for health care. Private cloud computing is an alternative to public cloud computing in that it provides many of the benefits of public cloud computing without the onus of executing a HIPAA-compliant business associate agreement with a vendor. To optimize efficiencies, health care organizations must ensure that their private cloud is properly sized given a set of parameters (operational characteristics, room for growth and redundancy). This section discusses a novel tool for evaluating sizing options for efficiently creating a private cloud given the unique nature of health care computer operations.

Cloud Simulators

There have been several systems constructed to provide estimates of potential cloud utilization (eg. [41] [42] [43]). These systems have focused on simulating public cloud use, network issues, virtual machine instantiation and destruction or other aspects primarily focused on public clouds. These general purpose cloud simulators do not address the unique nature of health care computing. Specifically, they do not take into account the predictability of private health care clouds.

Unique Aspects of Health Care Computing

Hospital-based health care is an around-the-clock operation with well-defined patterns of care. Nursing staff, who are the primary deliverers of care in in-patient settings, typically work either three eight hour shifts or two twelve hour shifts in twenty-four hour periods. Ambulatory, low-acuity clinics operate with standard office hours. Time-of-day and day-of-year patterns in care provisioning cause reasonably predictable patterns of computer usage. Ambrust et al, indicated that public cloud computing is best suited for usages where the resource consumption was unpredictable due to its elasticity. By contrast, health care operations and its computer use are remarkably predictable.

Health care and the practice of medicine is remarkably conservative and resistant to change [44]. This culture of change resistance permeates health care information technology departments and application vendors. Additionally, many hospitals use closed-source applications that cannot be easily modified for a specific environment.

Methods

A novel algorithm for determining the most efficient provisioning of resources was developed by the authors. This algorithm was presented in [45]. The algorithm takes into consideration the unique cyclical nature of health care computer operations. Unlike the previous cloud simulators, this system relies heavily on a trace-driven analysis using historic resource consumption behavior as the main predictor of future utilization. Historic resource utilization has been shown to be an accurate predictor of future performance [37]. This new method uses a minimally-invasive measurement system based on the host resources (RFC 2790) simple network management protocol (SNMP) interface [46]. The SNMP interface is reasonably ubiquitous, supported on most operating systems and avoids the pragmatic issue of potential conflicts with vendor-provided application software. The function of the algorithm is described below. At a basic level, the algorithm works as follows:

1. Collect data.
2. Divide and discretize the resource utilization data.
3. Sum the use of homogenous resources across all applications for each short time period.
4. Combine the use of each time of day for each resource into a "bucket".
5. Perform statistics on the data in the bucket and compute a point that is 99.5% greater than the sample set.
6. Find the maximum for each homogenous resource across all buckets.

We illustrate the use of this algorithm by example. Let's assume we're examining a single homogenous resource (e.g. RAM). Let's also assume a twenty-four hour period and a time quantum of one minute. Let's assume ten operating environments and a data collection period of two weeks. For each of the ten computers, there should be utilization data for each minute of the fourteen day period. Due to data collection errors, this is not always true (e.g. a machine is down or data are not recorded properly). Thus, a normalization step is required to correct for missing or extraneous data. Then, for each of the time quanta across all of the time periods, sum the utilization. This step characterizes the actual resource utilization over the entire sampling period.

For example, if each of the ten computers had an average of 2GB of utilization at 8:03am on day 3, the sum would be 20GB of usage for that minute. This is repeated for every minute (quantum) over the entire collection period. Then for each minute of each day, place the value of the previous step into a "bucket" such that the data from minute one of each day are grouped and minute two of each day are grouped, etc. Calculate statistics upon each bucket and determine a utilization amount that is greater than 99.5% of the samples. The maximum of these calculations across all buckets is a point that is statistically greater than 99.5% of all probable utilization scenarios.

Explanation

This example is a specific description of a generalizable algorithm. This method will produce the level of resources that should, based on history, satisfy at least 99.5% (or any arbitrary level) of all resource demands. Unlike other methods which simply sum the maximum resources potentially demanded by each application, this method takes into account staggered use of resources over time. In the near future, we plan to compare this method of hardware provisioning with other approaches.

Expected Contributions

In addition to the contribution of the algorithm above, we expect to create a tool that data center managers and systems administrators can either download or use via a web interface. This tool will assist them in collecting data in order to appropriately size their private cloud. We believe that this will improve the efficiency of computing in private clouds in health care and other organizations with similar requirements.

Aim 2: Virtual Machine organization

Problem Statement

Given a fixed set of hardware and applications with well-defined and predictable behavior, determine the efficient static or dynamic mapping of virtual machines to physical machines. This problem has been studied in generalized terms. This research will examine the specific aspects and optimizations possible in a highly predictable environment.

Motivation

Given the assumptions that health care systems will tend towards the use of private clouds and that the utilization of health care information systems is predictable, this research will examine how virtual machines should be efficiently mapped to physical machines within a private cloud. It will examine not just the relative location of virtual machines to each other but also internal cloud network optimizations (e.g. Xenloop [47]). It will also contemplate the use of a hybrid (private/public) cloud but continue to assume that most health care providers must be able to operate "off the grid".

Research Approach

We will create discrete event simulations that will characterize performance under varying circumstances. This will include varying machine placement, hardware architectures and specific private hypervisors. We will also implement novel experimental network stacks within specific open source health care information systems and note the different user-perceived performance characteristics given varying loads and varying architectures.

Expected Contributions

We expect that the contributions from this research will help to determine if health care is a special case and worthy of further research or specific approaches. We will also determine if special purpose protocols are useful for improving the user-perceived performance of health care applications in a private cloud.

Aim 3: Determining appropriate redundancy

Problem Statement

Given that health care providers are likely to use self-administered private clouds to host and manage their data, build a methodology for determining the appropriate level of redundancy.

Motivation

The level of redundancy required to support a given system is a complex decision and there is little guidance for data center and network managers to use to affect a reasonable balance. Additionally, increased redundancy typically increases the level of complexity. The increased complexity can paradoxically have an overall negative effect on reliability.

Research Approach

This research, which is admittedly in its early stages, will draw heavily on the performability work of Trivedi, Havenhort and others. Ultimately, we would like to construct a calculus that will provide guidance to system architects given a set of reliability constraints. We

expect to consider the Trivedi continuous Markov models that represent the “degree of failure” of the system, the transition probabilities within the state models and the economics of increased hardware redundancy.

Conclusion

Attaining computational efficiency in health care is vital as the amount of data collected, stored, and analyzed continues to grow at an increasing rate. The use of public clouds comes with some risk; on the fore-front are the issues of security and liability. Because of this, health care IT leadership is considering a move towards the use of private clouds instead. With this future need in mind, the authors set forth to develop methods to ensure computational efficiency with the use of private clouds in health care settings (as well as other organizations with predictable patterns of computer usage). Our aims are to develop novel methods to: (1) determine appropriate levels of hardware provisioning, (2) develop algorithms for virtual machine to physical machine mapping, and (3) determine appropriate levels of redundancy. Most of the research performed to date is related to Aim 1. As we move forward, we are grateful for the opportunity to present our ideas and to receive feedback/guidance from the reviewers and attendees.

Chapter 3 Appropriately Sizing Private Computing Clouds for Health Care

Introduction

Attaining computational efficiency in health care is important. Health care is an information-intense industry that represents a significant and growing portion of the national gross domestic product of the United States and is predicted to significantly increase in information intensity. Health care information technology leaders are challenged by increasing demands for services while trying to reduce the cost of infrastructure. Cloud computing has received much attention but health care leaders remain skeptical and cautious due to an ever-changing regulatory environment. This chapter discusses a novel approach to optimizing the so-called "private cloud" given the needs and characteristics of computing in health care.

Background

Public infrastructure-as-a-service (IaaS) cloud computing is a relatively new concept where consumers of raw computational capacity lease time and space on computing hardware from "public" providers. The consumers pay the providers based on actual usage and are relieved from the need to purchase hardware. The providers profit from this arrangement because they can over-subscribe the systems and, essentially sell the same hardware several times. Furthermore, cloud computing consumers can typically increase or decrease their allocated resources with little effort or expense to meet the needs of unpredictable workloads. Health care provides interesting challenges and opportunities for the use of cloud computing.

The regulatory environment surrounding the management of health information continues to change frequently. The potential liability associated with data breaches and the insecure

management of health information, as a function of historically leveraged fines, is also changing. IT leaders of HIPAA-designated covered entities are weary of new technologies that lack a history of legal scrutiny. Similarly, since the enactment of the Health Information Technology for Economic and Clinical Health (HITECH) regulations, vendors who manage HIPAA-designated protected health information (PHI) are liable for breaches of confidentiality [48]. Many popular vendors refuse to sign business associate agreements (BAAs) with covered entities as they are not willing to assume the HIPAA-related liability.

To measure the sentiment of health care IT leadership, an IRB approved (PSU IRB #41384) survey was conducted among members of the College of Health Care Information Management Executives (CHIME). The survey sought to understand the attitudes regarding the use of public and private cloud computing. Of the forty-one respondents, 81% indicated that they will need to increase their computational capacity in the next five years; 37% indicated that they were considering the use of public cloud solutions. When they identified a reason for avoiding public cloud computing solutions (in a free-form response), 70% indicated that security and regulatory concerns drove their decision. By contrast, 61% indicated that they were investigating the use of private cloud computing. It is noteworthy that this survey was conducted several weeks prior to the publication of the HIPAA Omnibus final rule [18]. In the final rule, the responsibilities of third-party hosting companies, which include public cloud computing providers, have become better defined. Where there was previously a level of ambiguity regarding the need for public cloud computing providers to sign a BAA, none now exists: public cloud computing providers who chose to host PHI for covered entities must sign a BAA and must accept the associated liabilities. As stated previously, they may not be willing to sign these agreements and thus may not legally accept responsibility for hosting PHI.

Technology can currently assist with reducing the risk of a breach of confidentiality and could contribute to reducing some of the concerns of health care IT leaders; however, there is no

perfect solution. The perfect solution would require a practical computational method to resolve the problem of fully homomorphic encryption (FHE). FHE would allow for storage and computation to be performed on encrypted data and would return encrypted results, keeping confidential the contents of the data even to the computer on which the computations were taking place. Unfortunately, there is no known computationally practical algorithm for implementing FHE [49]. It is an active area of computer science research.

Given the complexities of the regulatory environment and the apprehension of health care IT leaders to make use of public cloud technology, we created a system to improve the computational utilization of private computing clouds.

The system described in this chapter relies on two basic assumptions of health care systems: 1) that the practice of health care is temporally cyclical (e.g. nurse shift changes occur at generally the same time daily; physicians round at similar times each day; clinics open and close at particular times, etc.) and thus its use of computational resources is also temporally cyclical and 2) that past performance is a strong predictor of future activity. We validated these assumptions by observing data from Penn State Hershey Medical Center (PSHMC). The PSHMC, like many other health systems, consists of inpatient rooms and outpatient clinics. The outpatient clinics are open for essentially the same times every weekday and the rhythm of the activities in the inpatient clinics are fairly repetitive over a period of time, which, in this case, is typically twenty-four hours. Anecdotally, we've also been able to observe that the number of simultaneous users in various systems are excellent predictors of the number of simultaneous users in the same systems for subsequent normal week days ("normal" means it was not a holiday and there was no significant weather event). Granted, this is not proof that all health systems operate in predictable temporal patterns and it would be trivial to contrive a counter-example. However, we assert that most health care systems operate within some temporal periodicity.

Methods

The system described herein expands on an algorithm we first described previously [45]. The system examines the various offered loads across many computer systems and calculates the appropriate computational requirements for a composite computer or set of computers. Said another way, the output from this system is the set of requirements of the private computing cloud that would host the observed systems. The algorithm works as follows: data are collected from every machine at an operating system level (this could be virtual or physical computers); the sum of the homogenous resources are calculated for each time slice; each sum is then put into a bucket that corresponds to the identical time slice across all periods. That is to say, if C=24 hours and q = 1 minute, all minutes from every day are compared to each other (see Figure 3-1).

Gathering Phase:

```
foreach machine m
      foreach homogenous resource r
            measure and record util(r)
      end
end
```

Analysis Phase:

```
define C /*the temporal cycle in terms of q*/
define q /*the time quantum */
foreach q over all samples
      foreach homogenous resource r
            foreach machine m
                  sum_{r,q}=sum_{r,q} + r_m /* simultaneous usage for quantum*/
            end
      end
end
foreach homogenous resource r
```

```
      foreach q over all samples
            insert_into_bucket(sum_r,q,q mod C);
      end
      calculate average, stdev, single_ended_cnf_int
end
calculate max(single_ended_cnf_int) over all q mod C buckets
```

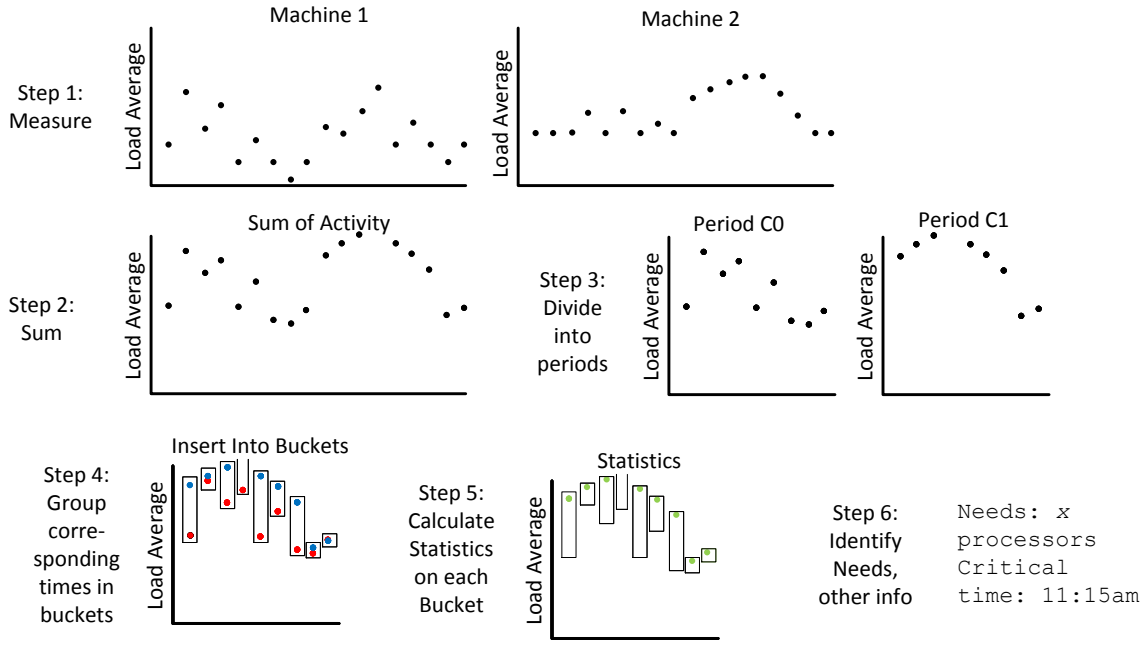

Figure 3-1: Graphical Illustration of Algorithm

Functionally, the system must be able to interrogate each operating system for specific statistics. To minimize the invasiveness of such interrogation, we chose to use Simple Network Management Protocol (SNMP) as the method by which these statistics are collected [46]. SNMP is native to most operating systems (e.g. Unix, Linux and Microsoft Windows) and provides a reasonably consistent interface for querying about the utilization of various operating system components.

In our system, we queried for utilization data about the components that we perceived were most likely to contribute to hardware constraints (processor, RAM, disk I/O and network I/O). There are idiosyncrasies associated with each resource that require some consideration.

There are two ways to measure CPU load within the SNMP resources. Actual load per core (which reflects how busy the CPU has been over a recent period of time) and offered load which considers how many threads of execution were ready for the CPU core but not necessary executed. We chose to consider offered load (load average) which is more reflective of the actual needs of the ultimate cloud system rather than the historical usage of limited CPU resources. Similarly, for RAM, there are two mechanisms that potentially confound the measurements from SNMP: the use of virtual memory (where offered memory load exceeds physical capacity) and file caching. We examined the total memory used (virtual plus physical) and subtracted the amount of memory used for file caching.

It is worth noting that the system cannot automatically compensate for maintenance activities that tend to drive resources to extraordinary levels of utilization (e.g. backups, antivirus scanning). Such activities tend to consume any available resources and do not represent the true needs of the system. At the moment, we manually remove readings during system maintenance from the calculation.

Example Exercise

To illustrate how this algorithm works, demonstrate below how it was executed on a small set of servers at the Penn State Milton S Hershey Medical Center. The servers contributed to the operation of REDCap, a database system used to support biomedical research. The system is architected as a traditional three-tier system (web server, application server and database server). For simplicity, in our example we examime only the CPU resource demands. Practically, all operating system hardware resources are interdependent and considering the utilization of one resource wouldn’t be sensible.

Step One

In the first step, we gathered historic CPU load information. In this example, we considered the use of the CPUs over approximately a three month period (see Figure 3-2). Although it may not be clear from the densely pictured data in the figure, most of the data points are close to zero – meaning most of the time, most of the CPUs are idle or nearly idle.

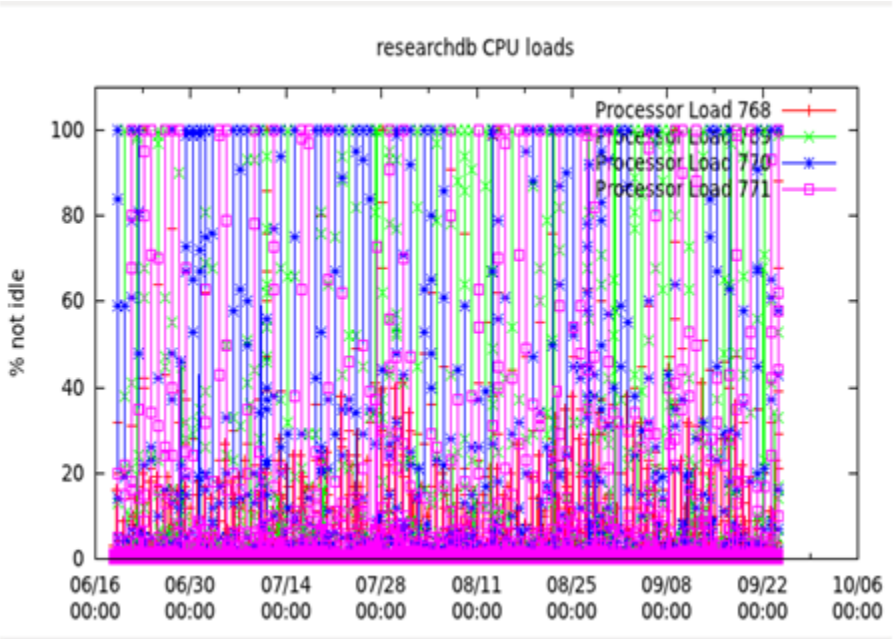

Figure 3-2: Migration Based on Behavior.

Step Two

In step two, we sum the usage of the homogenous resource (in this case, CPU utilization) together. For example, if at 7/28 at 14:22, four CPUs had load averages of 25%, 0%, 0% and 100%, the sum would have been 125%. See figure 3-3.

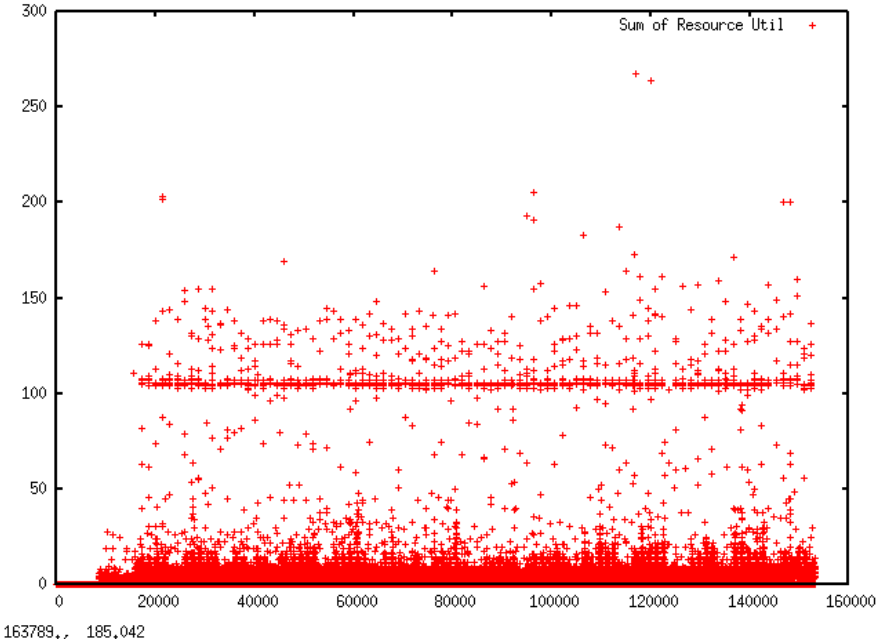

Figure 3-3: Summing the Resource Utilization

Step Three

In this step, we group all of the samples over all time into buckets within the time cycle. In our example, each bucket represents a minute-of-the-day (thus, there are 1440 buckets). See Figure 3-4.

Step Four

In this step, we calculate the amount of resources that would satisfying 99.5% of the resources demands for each bucket. This is illustrated in Figure 3-5 as the blue line. The green line represents the average of each bucket. Note that the surge in activity around minute 600 is related to the daily backup activity and would need to be manually removed from consideration.

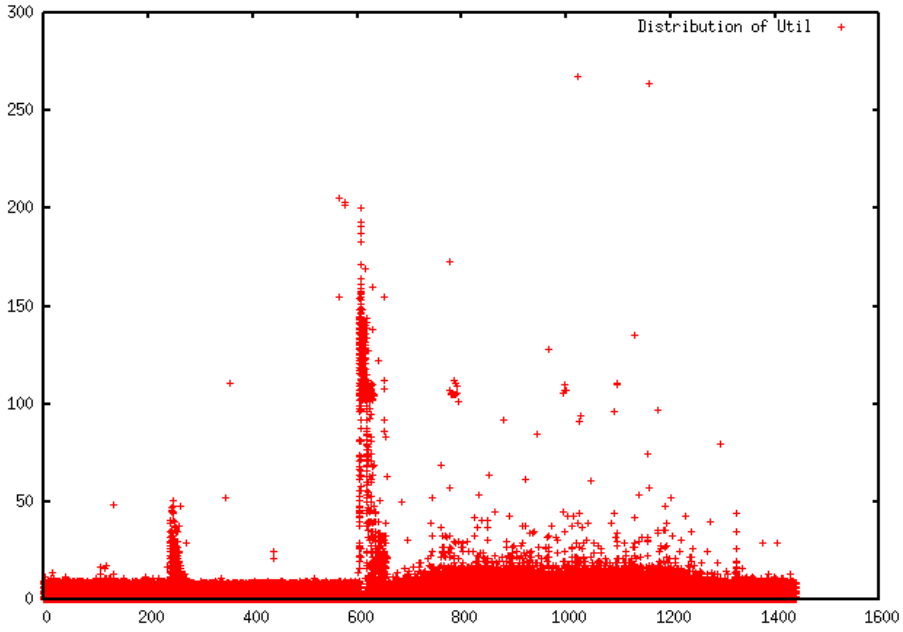

Figure 3-4: Bucketing the Samples

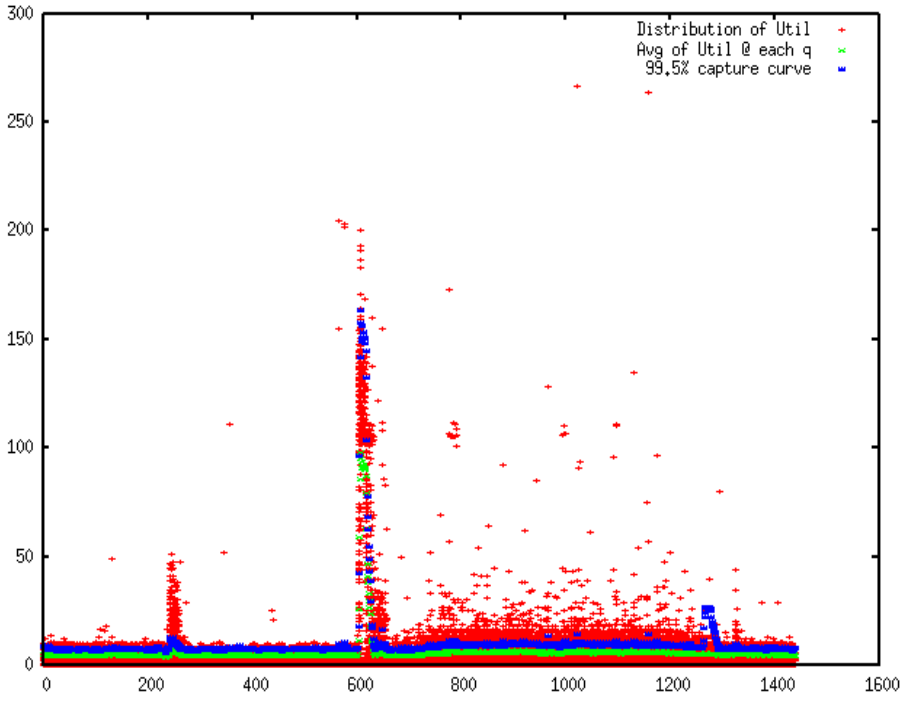

Figure 3-5: Calculating the 99.5%Requirement

Results

We have constructed a non-invasive system and algorithm that assists operators of health care clinical information systems to efficiently size their private computing cloud. The system relies on the cyclical nature of health care operations and, as our survey revealed, the tendency of health care IT groups to use private computing clouds instead of public computing clouds. We have used this system to analyze several internal applications at the PSHMC. Based on the information from this system, we de-allocated or allocated resources depending upon historic need.

Future Work

Ultimately, we envision producing a downloadable package whereby users could monitor their systems, analyze the results and construct appropriately sized private computing clouds. We would like to introduce a human-mediated method to isolate system maintenance activities which skew the results as described above. We would like to automatically detect trends in the data to increase the assurance of the validity of the proposed private cloud computing system over time. We would also like to include statistics about the overhead of various virtualization technologies to ensure adequate resources are suggested in the proposed cloud configuration.

Conclusion

This chapter offered two contributions: the survey results of health care CIO's attitudes towards cloud computing and a novel system for sizing private computing clouds. IaaS cloud computing has the potential to contribute to reducing costs of health care computing, however, the regulatory concerns and unwillingness of many vendors to assume the liabilities associated

with hosting confidential health information weakens the case for the use of public cloud computing. Private cloud computing has many of the benefits of public cloud computing. An important difference is that private computing clouds are limited to the resources provided by the entity. Thus, health care entities must intelligently design their private computer clouds to meet their computing needs while not over-buying capacity. The method discussed in this chapter provides a way for system engineers in health care to efficiently size their private computing clouds.

Chapter 4 Virtual Machine Placement in Predictable Computing Clouds

Introduction

The virtual machine mapping problem (VMMP) is reducible to the bin packing problem which is known to be NP-hard. As such, several heuristically-based solutions have been proposed to introduce a tractable yet useful system to ensure efficient resource utilization and improve user-perceived performance.

Much of the literature about cloud computing assumes that resource demands are not near-term predictable and that cloud computing is best suited for unpredictable resource demands [17]. However, it has been shown that predictability is a characteristic of some types of workloads [50]. We specifically found that this is the case in small-scale systems designed to support private clouds in health care. Hospital-based health care is an around-the-clock operation with well-defined patterns of care. Ambulatory, low-acuity clinics operate with standard office hours. Time-of-day and day-of-year patterns in care provisioning cause reasonably predictable patterns of computer usage.

At its core, the VMMP is a multivariable optimization problem that has historically been attacked using different approaches. There are many algorithms that seek to optimize different aspects of the computing cloud. We have, however, found no algorithms that consider the long-term predictability of the future resource demands as a means of determining virtual-to-physical machine mapping. Consider the trivial example of electronic medical record system that is divided into an application logic server and a database server. Let's assume that we have a virtual machine mapping algorithm (VMMA) that considers inter-VM communications as the primary determinant of virtual machine mapping (VMM). For most of the day, that database server is located proximate to its corresponding application server (see Figure 3-1, t_o). Each day, at nearly

the same time of the day, the content of the database is extracted into a central data warehouse. When the extraction process begins, communications between the database virtual server and the data warehouse virtual server increase significantly. The VMMA then relocates the database virtual server to the physical machine that hosts the data warehouse virtual server (see Figure 3-1, t_1). Subsequent to the data extraction process, the VMMA relocates the database virtual server back to the physical server that houses its corresponding application virtual server (see Figure 3-1, t_2). Given foresight, a VMMP that can conceptualize past history may select to migrate the VM that is about to be extracted before any real-time indicators suggest it should do so.

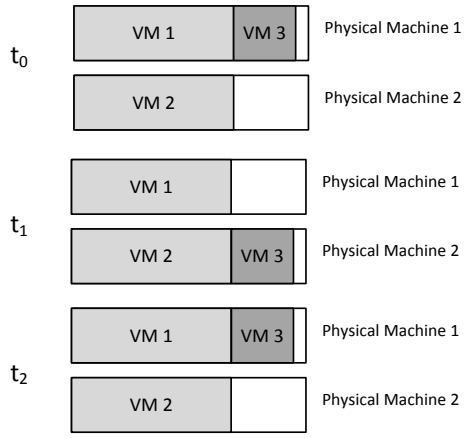

Figure 4-1. Migration Based on Behavior.

Background

Load Balancing Approaches

There are several approaches to distributing virtual machines among a set of physical machines. Due to space restrictions, we refer to [51] which provides an excellent survey on the

subject of diverse approaches to virtual machine mapping algorithms. To our knowledge, none consider long-term patterns of resource utilization when mapping virtual to physical machines.

Novel Algorithm

Most modern VMMAs consider recent activity when making mapping decisions [51]. These algorithms are excellent for unpredictable resource demands but ignore useful information when machine behavior is predictable. Thus, we have created a new algorithm that looks ahead based on past history. Rather than waiting for the hypervisor to react to new resource demands, the algorithm introduces a "look ahead window" which evaluates the current state of the resource demands compared to historical norms and reacts based on the probable resource demands of the entire system. We've called this algorithm "Look Ahead and React Based on History" or "LARBOH".

Computational Overhead of LARBOH

LARBOH requires additional computation by the hypervisor as well as historic state information. The hypervisor must be able to access a record of previous performance characteristics of each virtual machine running within the cloud. Depending upon the level of temporal resolution and the cycle duration, this will result in varying sizes of datasets. In our implementation, the computational overhead of looking ahead was of the same order of complexity as current VMMAs. The storage required to manage past history was also not significant as we focused on a period of one day with a sample frequency of one minute. Furthermore, we implemented LARBOH in our simulations such that it only had seven days of history.

Characterization of Predictability

A basic assumption of this research is that there exist systems that have predictable resource utilization behaviors. Stokely et al examined the predictability of resource usage in a cloud environment [37]. They found that aggregate behavior among many users was predictable with reasonable accuracy (+/- 12%). We have had access to resource utilization data at diverse academic health centers throughout the eastern United States (Tampa, Detroit, Boston and Hershey, PA). At these centers, past resource utilization was consistently a good predictor of future utilization. The following data are anecdotal but we contend that it is characteristic of health care resource utilization. During December, 2013, the same-day (week-to-week) differences in the number of users on the electronic medical record system varied by less than 1% (0.463%) for non-holiday weeks for a population of providers of more than 2000.

Experiment

Trace Creation

We constructed a simulation that used traces of actual performance of different component virtual machines of an electronic medical record (EMR) system. We used OpenEMR as the sample EMR. The traces were generated using the same configuration as described in [50]. We varied the number of simulated client machines using Apache Jmeter. We used SNMP to query each virtual machine regarding load average, RAM usage, RAM allocated to file caching (so that file caching was not a confounder of memory utilization), network bandwidth utilization and disk I/O operations per second. The results from the SNMP queries were written to text files and were used as the source of trace data that was fed into the subsequent simulation. RAM was

ultimately the most contentious resource during our simulations. The demand for RAM scaled linearly with the number of simultaneous users.

Simulation

We constructed a simulation written in C to examine how the timing of the migration of virtual machines between different physical machines impacted the utilization of the physical machines and the responsiveness of the virtual machines. We also examined how the timing of the migration affected user-perceived performance as well as the complexity of the migration. As we indicated previously, the use of RAM in our model systems scaled linearly with more users. The amount of RAM allocated is also a prime determinant of the computational cost of migration. To simplify the experiment, we relied on a single threshold variable: RAM utilization. Although others have successfully integrated multiple variables into the migration decision [52], it was sufficient to use one key resource as a trigger for migration to demonstrate that there is value in preemptive migration of virtual machines. We measured the impact on users as the product of the number of users and the amount of time required to migrate the virtual machine.

Results

Our simulation demonstrated that preemptive migration of virtual machines in anticipation of changes in resource requirements can improve the operation of the cloud system while introducing only a slight increase in computational overhead for the hypervisor. Depending upon the specific states of the machines and in highly predictable systems, LARBOH was able to reduce the user impact of virtual machine migration by up to 99.7%. This was based on the fact that the migrations occurred in our simulated environment during a period of time when users

were not on the system. This is an admittedly contrived and optimal circumstance. In real examples (such as our actual experiences with using an electronic medical record system at the Penn State Milton S Hershey Medical Center), there are often some users who continue to use the system during "low" periods who would be affected by a virtual machine migration.

Future Work

There were several arbitrary constants used in the initial algorithm that may be opportunities for optimization. Specifically, the look-ahead time range and the duration of history considered should be examined. Furthermore, it would be useful to quantify the definition of "predictable" and specify the level of predictability that makes LARBOH useful.

Conclusion

Preemptive positioning of virtual machines within a cloud, given a reasonably predictable environment, results in improved user-perceived performance. The LABOH algorithm, which considers historic virtual machine migrations, could be incorporated into future hypervisors' VMMAs to reduce the performance impact on users of migrated virtual machines.

Chapter 5 A Network Security Architecture to Reduce the Risk of Data Leakage for Health

Introduction

Health care is a highly regulated industry in which much value is placed upon privacy and confidentiality. The business of health care, particularly in certain academic environments, requires the use of data of varying sensitivities, including information from the public Internet. This chapter proposes a VLAN-based architecture for segregating data of varying sensitivities, a list of components that facilitate access to and distillation of data, and a method for one-way promotion of individual nodes from areas of lower security to areas of higher security. The inspiration for this work was the authors' experience at several large academic health centers (AHCs) where the need to restrict access to confidential patient information was often challenged by the need to ensure the free flow of information required to cultivate a rich and collaborative research and educational environment. This research is supportive of future work which seeks to minimize the risk of data leakage while making use of hybrid computing clouds.

It is noteworthy that no system can prevent all *intentional* forms of data leakage. The proposed architecture does nothing to prevent egregious behavior by authorized individuals who are committed to acting unethically or illegally. For example, technical security won't prevent a bad actor from capturing a screen image of confidential data using a camera or smart phone.

This chapter will detail: i) an implementable approach for managing data with varying degrees of sensitivity, and ii) a new method for dynamically changing VLAN assignments by specific nodes. *A note about wording:* we often refer to sensitive data metaphorically as a pollutant to be contained. This is apropos as sensitive data have many of the same characteristics as dangerous chemicals: they are useful if managed well but dangerous if control is lost.

Background

The need to restrict the flow of confidential information is a fundamental component of information security. Data leakage is the unintentional flow of data from trusted systems and networks to less trusted systems and networks. There are daily accounts in the popular press in the United States about unintentional data leakage [53]. There are many examples of where patient data were inappropriately stored on unencrypted laptop computers, written to portable storage devices or displayed on public web sites. The Bell-LaPadula (BLP) model remains the authoritative standard reference model for multilevel security. (A more thorough discussion of the BLP model is included as the appendix.) BLP is a purist approach. It has been well-recognized that there are pragmatic needs that cannot be addressed in an environment that stringently adheres to BLP [54]. For example, a strict implementation of BLP in health care would prevent patients from accessing their own health information from their (presumed to be insecure) personal computers and devices. This may reduce patient engagement and would be contrary to efforts promoted by health providers and governmental agencies. Some data leakage management schemes have sought to classify every datum of every system and facilitate the management of leakage avoidance through novel programming language constructs and appropriate technical controls. The abundance of legacy applications and the slow rate of change of applications in health care settings [44] makes these largely academic efforts impractical to apply.

In the United States, the Health Insurance Portability and Accountability Act (HIPAA) defines broadly how health care data are to be managed and secured [55]. There are similar laws in many jurisdictions worldwide. Although HIPAA was created in 1995 and went into full effect in 2003, varying degrees of enforcement and penalties have impaired the effectiveness of its adoption. In 2009, as part of rapidly-enacted legislation designed to avoid economic disaster, the penalties associated with non-compliance of HIPAA were strengthened [48]. A 2011 study found

significant variability among AHCs regarding compliance with HIPAA. Information security continued to be described as an afterthought [56]. Most AHCs surveyed lacked sufficient management support, culture and technical measures to ensure compliance. Patient data, though well-publicized due to HIPAA, is not the only data classification of concern for AHCs. They must also be concerned with the privacy of student data (in the United States, this is codified in the FERPA rules [57]), payment card data (as codified in the PCI-DSS contractual requirements [58]) and other rules depending upon the regulatory or contractual framework that governs the data. Additionally, AHCs have a moral and ethical obligation to ensure the privacy of patients and research subjects against emerging threats. During the 1990s and early 2000s, genetic/genomic information that lacked specific identifiers was considered to be "de-identified". However, in 2013, researchers were able to successfully re-identify genomic data that was thought to be de-identified through the application of several external databases [59]. Individual data use agreements, formal and informal agreements to collaborate between institutions, individual scientists and physicians add further dynamism and thus complexity. Regulations, threats and relationships are changing rapidly. Additionally, users at AHCs often require access to multiple classifications of data as part of their workflow. A well-understood framework to reduce the risk of data leakage would be useful to the health care industry and specifically at AHCs.

Proprietary methods for virtually segregating local area network (LAN) traffic over switched link layer networks were introduced in the mid-1990s. The IEEE amended the 802.3 protocol in 1998 to officially establish a standard for VLAN traffic [60]. Early in the history of VLANs, there were discussions about using VLANs to segregate traffic based on policy [61]. There have been many examples of research and practical implementations that have focused on using VLANs to segregate data [62-64]. We have considered this previous research in our architecture.

Architecture of System

The architecture of this system is meant to be implementable using existing protocols with minimal modifications and existing applications. There are a number of important considerations that drive this solution:

1) pragmatism is key -- this solution must be implementable using current technology and current (or old) applications;

2) therefore, it is not practical to classify every datum in a system; systems will be classified based on the most sensitive data they store (this assumes that an ordering exists upon which the data in systems can be compared);

3) highly sensitive data must be viewable with restrictions from low security areas; realistic needs of clinicians such as remote access to sensitive data must be satisfied; and

4) “multiple use” devices must be able to transition from being classified as low sensitivity to high sensitivity dynamically; a method should exist to “reset” the device to low sensitivity.

The overall architecture functions using the constructs described in the following sections.

Network Zones

Each network zone has a specific characteristics: a security designation which describes which data may transit and be stored within it; membership requirements which must be met by any node connecting to the network and enforced through administrative or technical mechanisms; a set of privileges associated with the security designation; and a set of prohibitions.

A practical example of this would be a network that permitted the storage and transit of regulated health data as described below in Table 4-1.

Table 5-1. Example of Zone Characteristics

Characteristic	**Example Restriction**
Security designation	PHI Zone
Membership requirements	Antivirus software; host-based IDS; 802.1x authentication
Privileges	Create, access, modify PHI
Prohibition	No Internet access; no access to email system

In this scheme, we assume that the network can securely segregate and maintain the separation of packets with different designations (tags). Modern layer two (switched Ethernet) networks perform this through the use of IEEE 802.1Q or other similar mechanisms. It is noteworthy that in our proposal there is no direct connectivity between zones of differing security classes. A layer three packet emanating from one zone *cannot* enter another zone. All information is conveyed through various filtering and proxying gateways at the application level of the network stack.

Sources

Static Sources

Packets emanate from information sources. At any time *t,* every information source *S* has a classification designation C(*S*). The packets originating from these sources are tagged with the same designation. Static sources have well-defined designations and may not change during their lifetime. An example of a static source may be a hospital registration system. It stores and computes upon protected health information (PHI) which has certain legal requirements which are enforced through technical measures.

Dynamic Sources

There are also dynamic sources of information. The security designation associated with dynamic sources may grow higher during operation but not lower. Formally, C(S) = i → C(S) = j, where j > i. However, the source may not make the reverse transition without executing the "decontamination" process (see below). An example of a dynamic source system would be a general purpose workstation which, by default, is set to the lowest level of access S_0. A general purpose secure workstation may access insecure systems such as the Internet. It may also access secure systems such as the registration system in the previous example. It may not, however, access the insecure system *subsequent* to accessing the secure system. Permitting it to do so would create a path where data may have left the registration system, been recorded on the insecure workstation and then transmitted to a lower security system (and violate the "no writes down" rule of BLP). Thus, before the system may receive a packet from of high security, its own security must be changed (in this case, $S_0 \rightarrow S_P$). Once the workstations designation is set to S_P, it may no longer send packets to targets with lower security designations.

Porthole

Portholes (as opposed to the over-used term "portal") are secure gateways that permit the access of higher security zones from lower or differing security zones. The portholes are designed to consolidate connectivity and minimize the risk footprint. Like their namesakes on ships, the porthole is/should be designed to provide an opening but not facilitate egress. Data may be viewed through a porthole but not copied (meaning that the risk of "copying and pasting" is removed). Practically and technically, however, the existence of the porthole increases the risk of

data leakage (over having no access whatsoever) as data may still be intentionally leaked through screen capture software or simply by taking a picture of the screen.

Declassifiers

Declassifiers are secure data processing mechanisms that accept as input information with security designation S_i (within a zone capable of supporting data with characteristic S_i) and output data with security designation S_j where $S_i > S_j$. For example, a declassifier may take as input several identified patient records and output statistical information.

Sanitizers

Sanitizers are mechanisms and procedures that cleanse dynamic source nodes so that their security level and associated network zone may be "reset". The sanitizer "decontaminates" the node of any data that should not leave the high security zone. Practically, this typically translates to erasing the long-term storage, resetting the RAM and reinstalling the operating system and application software.

Example

Figure 4-1 below depicts an example design for protecting health data with different zones and different interface mechanisms in a static environment. In our example, we depict a user of a personal computer accessing data with a high security classification from a zone of lower security classification through portholes. We also depict the use of declassifiers and how

they would make increasingly abstracted patient data available through i2b2 [65] within different zones.

Security Benefits and Risks

The proposed architecture reduces the possibility of data leakage through accidental disclosures such as copy-and-pasting, emailing and posting data into systems on the public Internet. Furthermore, the total lack of connectivity also protects against botnet-like leakages or other types of malware. Assuming that the network is and remains secure, a malware infection could corrupt data but not expose it outside of the organization (given the assumption that Internet connectivity is in the list of prohibitions for a secure zone).

The risk considerations and assumptions associated with the architecture are described below.

Porthole Image Capture

As described, the portholes access data through a secure mechanism of "screen scraping" which facilitates viewing of regulated data but not transmission. These protections can always be defeated through mechanisms that capture screen images (which could be something as simple as a camera) and the data could be re-constituted using optical character recognition techniques.

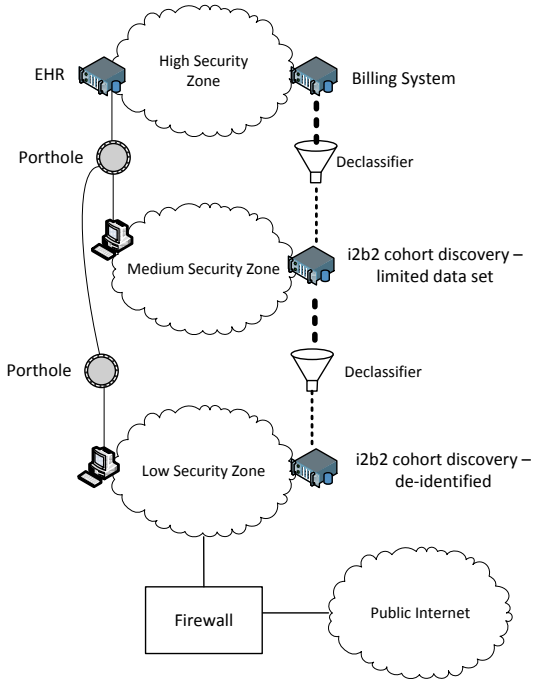

Figure 5-1. Data Leakage Protection Architecture.

Declassifiers

The declassifiers which are used to aggregate statistics about data or otherwise reduce the security classification of certain data must be formally evaluated to ensure that sensitive information cannot be leaked. Organizational policies, mature data governance and rigorous testing routines are required to ensure that the declassifiers don't become a path for data leakage.

Network Isolation

A foundational assumption of this work is that network isolation is feasible and that VLAN hopping or other kinds of VLAN or network manipulation are improbable.

Performance Considerations

Performance concerns are the motivation behind the second contribution of this chapter: a method to dynamically promote a node from a low security zone to a high security zone. We are concerned about the user-perceived performance of access to applications (also known as "quality of experience" or QoE). QoE is defined differently in several papers[66-68], so we will define it here: the user perceived experience associated with usage activities. We will concentrate on objective measurable events and leave user satisfaction for future work.

Based on previous experiences measuring the performance of EHRs [45] and work by Casas et al [68], we know that most user-triggered network transmissions involve keystrokes or mouse clicks while most server-triggered transmissions update screens. End-to-end delay between two nodes on a network is a function of the sum of the delays related to network queuing, transmission, propagation and processing time [69]. In the simplified mathematical model below, the QoE, is a function of the network-induced delays plus any delays associated with application responsiveness, thus,

$$QoE_{client\text{-}server} = f(d_{app}, d_{queue} + d_{prop} + d_{proc} + d_{trans})$$

Typical end-to-end processing of user-generated events must be processed first by the intermediate (porthole) server and then (typically) cause a transmission from the porthole server to a back-end server. Thus, the QoE associated with a porthole-based session is a function of more delay contributors:

$$QoE_{porthole} = f(d_{app}, (d_{queue} + d_{prop} + d_{proc} + d_{trans})_{client\text{-}to\text{-}porthole}, (d_{queue} + d_{prop} + d_{proc} + d_{trans})_{porthole\text{-}to\text{-}app}, d_{porthole})$$

or, more generally:

$$QoE_{porthole} = f(\sum_{k=1}^{n} (d_{queue} + d_{prop} + d_{proc} + d_{trans} + d_{porthole})_k + d_{app})$$

where there are *n* portholes involved in access high security zones from low security zones.

These additional network delays and contention for the porthole service itself can cause significant decreases in the QoE. The delay increases multiplicatively as the user is forced to traverse more "porthole hops". In modeling the system, we found that transmission and propagation delays contributed negligibly to the overall performance of the network while queuing delays at the porthole host and application delays were potentially significant. This culminated in the finding that, at times, it may be preferable for a node in one area of security to be "promoted" to a higher level of security to increase performance.

Security Zone Promotion

In this section, we propose a system to facilitate moving from one security zone to another (the zones are depicted in Figure 4-1). The rationale for this is the performance degradation associated with accessing applications through portholes. The porthole based access is inherently slower than direct client-server access as the porthole-based access adds another layer which introduces non-zero delays. In describing the re-zoning of a node in the network, we'll make several realistic assumptions: 1) there is some triggering event that causes the node to be re-zoned; 2) the node originally obtained its IP address through DHCP; 3) the loss and re-gain of link assertion at layer two (e.g. Ethernet) will trigger a DHCP lease request; and 4) upon re-zoning, all previously established TCP/IP network connections will be terminated. This triggering event (discussed in more detail later) could be a threshold violation or a user-initiated event. One could envision an icon on a computer workstation where the "glass is broken" is escalate the user to the next security level.

Security Discussion

The goal of the network architecture is to ensure preservation of the BLP security model. This proposed process ensures that data from the high security zone does not enter the low security zone. By changing the designation of the node from "low" to "high" security, we are essentially taking low security classification data and placing it into higher security classification zone which is permissible under BLP. The risk associated with doing this is the potential for malware or other undesirable data or code to enter into the high security zone. Although this risk must be managed, it does not violate BLP. We also assume that it is not feasible for an unauthorized individual or node to promote another node. We have not yet developed the details of the promotion mechanism.

Procedure

The procedure of the system is the following: i) communications with central service to promote node to new VLAN, ii) central services communication to network infrastructure services to change VLAN or machine, iii) network infrastructure link assertion removal from node, iv) delay associated with link assertion that is sufficient to cause node to remove network stack scaffolding; v) abnormal closure of existing network connections; vi) re-assertion of link; vii) failure of DHCP renewal and request for new DHCP lease; viii) connection to application without the need for porthole use.

Performance

The performance of the steps listed in the procedure is highly dependent upon implementation. We experimented with components of the process on two different operating

systems (Microsoft Windows 7, Linux/Ubuntu 10.04 LTS). The entire process, under ideal circumstances, took at least 2 seconds. Minimally, the following communications must occur (Table 1). These activities are all implementation specific and thus, cannot easily be quantified generally.

Table 5-2. Security Zone Transition Delay Contributors

Activity	**Description**
Promotion Request	Packet from client to promotion manager
Promotion Approval	Packet from promotion server to client
VLAN Re-assignment	Promotion server to network infrastructure
Link Removal	Switch de-asserts link
Link Removal delay	Sufficient to cause network on client to deactivate
Link Assert	Switch re-asserts link, negotiates speed and duplex
DHCP Renewal	Depending upon implementation, client may attempt to renew previous IP address, which will fail
DHCP Renew Failure	Depending upon implementation, client may attempt to renew previous IP address, which will fail
DHCP Lease Request	Client will request new DHCP address
DHCP Lease Response	Server will allocate new DHCP address
DHCP Lease Accept	Client will accept new DHCP address
Initiation of Applications	Client will launch applications (application/implementation dependant)

Thus, the performance of the promotion scheme must consider the one-time performance costs of the promotion activity and the on-going performance of the node's interactions with the application post-promotion, thus:

$$\mathrm{QoE}_{\mathrm{promotion}} = f(\mathrm{d}_{\mathrm{promotion}} + \sum_{k=1}^{n}(\mathrm{d}_{\mathrm{queue}} + \mathrm{d}_{\mathrm{prop}} + \mathrm{d}_{\mathrm{proc}} + \mathrm{d}_{\mathrm{trans}})_{\mathrm{k}} + \mathrm{d}_{\mathrm{app}})$$

Comparison

We used transactional throughput as a measure of productivity and a proxy measure for QoE. The transactional throughput is defined as the number of user interactive events that could practically occur over a specific time period. As performance declines, we expect the number of possible user transactions to decline. User events are defined as key presses and mouse clicks. This is, of course, a simplification. A single user-driven event could cause a video to start playing or a screen to render several times which result in significant traffic. However, for GUI applications that largely consist of text (such as an EHR), this is typically not the case. We measured the possible transactions over a given time.

The cumulative possible transactions at time *t* is the sum of all of the possible transactions that could have occurred in each of the proceeding discrete time units for this sessions, less the opportunity costs associated with delays, thus:

$$\mathrm{trans_c} = \sum_{k=1}^{t} \mathrm{trans_{k_t}} - \mathrm{trans_{k_t}} * \mathrm{delay}, \mathrm{trans_c} \geq 0$$

The promotion method will be superior from the perspective of more possible transaction could have occurred when,

$$\mathrm{d_{promotion}} < \sum_{k=0}^{t} \left(\mathrm{d_{porthole_k}}\right).$$

This measure does not take into account user preferences or usability issues that might otherwise sway users' decision making process.

Experiment

We considered several scenarios to compare the performance of the porthole based access versus the performance of the promotion mechanism. Our hypothesis is "for brief forays into zones of higher security, the porthole method would be preferable." For sustained use of systems in higher security zones, the zone promotion method would be preferable. Through our experiment, we sought to verify the hypothesis and determine the value of "brief."

First, we sought to understand the network traffic associated with user-generated events. We used Wireshark 1.4.0 to capture packets related to specific porthole technology. Since "porthole technology" is not well-defined, we connected to two technologies that may be candidates for remote application portholes: a multiuser version of Microsoft Windows with Citrix and XWindows running on Redhat Linux. To test simple activities (keystrokes and mouse-clicks), we opened "notepad.exe" on the Windows machine and initiated a remotely displayed "xterm" which was tunneled through SSH to display through XWindows. We then counted how many packets were generated by each activity. Individual keystrokes (which included transmitting the keypress event and the subsequent echoing back to the remote terminal) caused, on average three packets to be transmitted. Mouse-click events tended to result in an average of twenty-two packet transmissions.

We then created a simulation in C running under Cygwin. We measured the impact of various components of delay and only considered the elements which were likely to contribute significantly and vary between the different the porthole solution and the zone promotion solution. In order to keep the simulation manageable, we made several assumptions, specifically: the inter-node distances were kept constant at 1000m, a layer three network diameter of 5, no queuing delays in the network (but varying queuing delays on the portholes and server nodes),

and the networks operate at a constant 100 Mbits/sec. We assume that the TCP sessions are already established and thus no handshake is required.

Results

The performance reduction for the porthole based users was highly dependent upon the number of simultaneous nodes contending for access to the portal (and thus, the network queue on the server) and the amount of delay induced by the porthole system itself (see Figure 4-2). We varied promotion delay, and node processing delay.

Discussion

The performance of connecting to protected systems is quantifiably and intuitively improved for the nodes that are members of the protected zones. However, there are drawbacks that may contribute to users' decisions not to self-promote their nodes. Once part of the protected zone, access

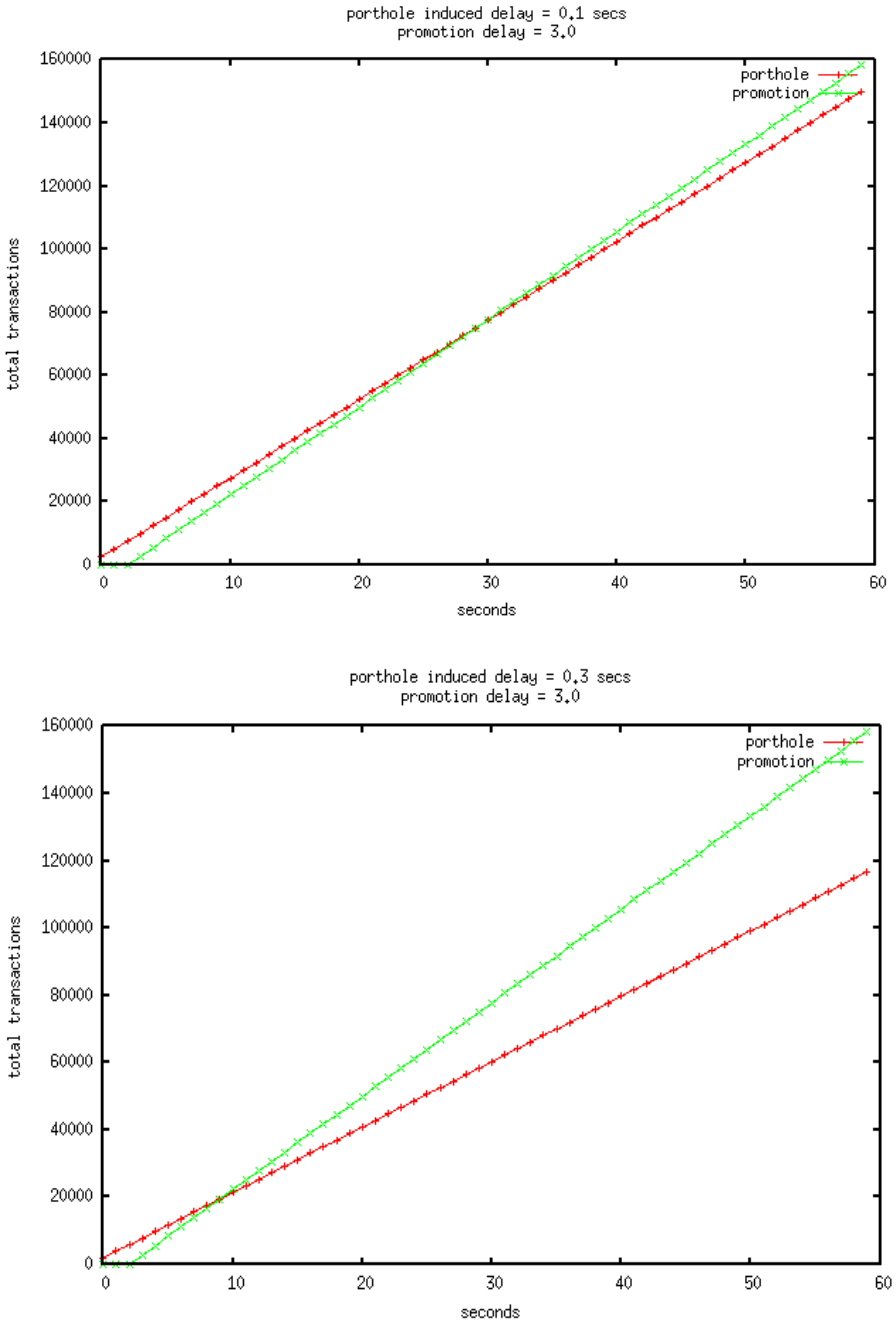

Figure 5-2. Varying Break Even Points for Differing Assumptions.

to information assets in lower zones may be restricted or lost completely depending upon the prohibitions associated with that zone. Similarly, there are also advantages for promoting into the protected zone – for example, a user may be restricted from "copying & pasting" data – even

between applications that are accessed through the porthole into the protected zone. One could also envision policy decisions that prohibit access to high security information through the porthole method. Another longer-term consideration for the user is the need to "decontaminate" the promoted node subsequent to its use in the promoted zone. There may be overall QoE costs associated with the decontamination process (time, effort) that increases their interest in the porthole based access method.

Future Work

In future work, we will consider the development of the secure mechanism to facilitate the node promotion process. Furthermore, we may also contemplate an automated method for zone promotion with specifically defined promotion triggers. We will also demonstrate how this security/tiered method could be utilized to ensure that hybrid computing clouds don't inappropriately offload virtual machines to inappropriate public cloud providers.

Conclusion

This chapter introduces two concepts: i) a VLAN based security architecture to facilitate compliance with the rules of BLP while facilitating pragmatic needs of organizations with varying classifications of data and ii) a system for promoting designated nodes to higher security zones to ensure critical access. The architecture facilitates increased protection to prevent accidental data leakage. The porthole and promotion methods both facilitate escalated access to sensitive data with different performance characteristics and access benefits and costs.

Chapter 6 Reducing the Risk of Health Care Data Leakage with Consideration for the Intercloud

Introduction

Health care is an information intense industry that is often paradoxically slow to adopt new information services. Regulations, entrenched software and safety concerns contribute to this conservative approach of the adoption of newer technologies. As an information-intense industry that is heavily regulated, information protection must be balanced with information availability. The balance of security and access is further complicated in certain academic environments. Academic Health Centers (AHCs) value the ability to broadly share information to facilitate research while simultaneously protecting that information from accidental leakages. This work was motivated by the authors' experience at several large academic medical centers in the United States.

This chapter describes: i) a pragmatic system for isolating different classifications of data, ii) a novel method for promoting user nodes from lower security zones to high security zones while conforming to classical multilevel security rules and iii) a discussion of how this system may be implemented in the universal hybrid cloud known as the Intercloud. A preliminary version of this work has been reported [70]. Note: within this chapter, we often refer to sensitive data metaphorically as "toxic." Sensitive data and dangerous chemicals have common characteristics; both are useful when used correctly but dangerous if control is lost.

The project described was supported in part by the National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant UL1 TR000127. The user reaction quantifier program described in section III was developed by Autumn Rauscher. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

\

Background

Foundational to information security is ensuring that information flows in known ways and adheres to prescribed restrictions. Data leakage occurs when data flow to unauthorized networks, devices and people. Unfortunately, data leakage occurs often either due to the intentional activities of bad actors or accidentally through lost or incorrectly configured devices [53]. The early history of multilevel information security was pioneered by research conducted for the benefit of the military. In 1973, Bell and LaPadula proposed a set of principles which became the de facto standard for preserving multilevel information security and thus preventing data leakage [54]. The Bell-LaPadula model (BLP) continues to be relevant today. However, it is a purist approach. A strict interpretation and adherence to the BLP model would make certain tasks such as remote access of information over the Internet impossible. Therefore, in our work, we aim to achieve the spirit of BLP while realizing that there are pragmatic but essential compromises that may be desirable. Continuing with the theme of pragmatism, we also recognize that it is often infeasible to classify every datum in an information system. Given the slow rate of change of technologies in health care [44], systems that rely on the classification of every datum are not pragmatic in modern health care enterprises.

The Health Insurance Portability and Accountability Act (HIPAA) is a law that was passed in the United States in 1996 [55]. One of the many implications of the act is that it defines the legal requirements to protect health care information from intentional and accidental leakage. The law was passed in 1996 and the information security component of the law took full effect in 2005. Many jurisdictions around the world have similar laws. HIPAA has been amended and was significantly strengthened through broader penalties and enforcement powers in [48]. A 2011 study found that compliance with HIPAA was highly variable among AHCs [56]. AHCs reported that security was often a secondary consideration to functionality and that support from

management to invest in security was limited. HIPAA defines how one type of data (health care) must be managed. There are other types of data managed daily at AHCs including student data, credit card data, and clinical research data. Depending upon the jurisdiction, there are many different laws that may require protections to be applied to different types of data.

During the mid-1990s, several technologies such as Ethernet, token-ring and asynchronous transfer mode (ATM) vied for the position of the dominant local area network (LAN) technology. Switched Ethernet, a derivative of Ethernet that borrowed from the switching technology of ATM, became popular. With switched Ethernet, it became possible to logically isolate groups of Ethernet nodes from other groups of Ethernet nodes that connected to the same set of Ethernet switches. This formed the early basis for virtual LANs (VLANs) that ultimately became supported by IEEE standard 802.3 in 1998 [60]. The idea of using VLANs to segregate different classifications of data from each other was an early concept [61]. We have considered the historical use of VLANs in data segregation in this work.

Proposed Architecture

Our goal is to create an architecture that can be used by modern health care organizations without substantial changes to their applications or networks. Thus, we were motivated by the following requirements: a) the proposed architecture must function with current technology and older technology; b) classifying every datum in a system and managing the flow of every datum is not practical; c) systems will be classified based the most sensitive data that they contain or process; d) the system must be able to facilitate the remote viewing of sensitive data on unsecured devices (e.g. home computers, smart phones) based on the policies of the individual organization; and e) designated devices must be able to logically move from low security zones to higher

security zones. The means by which we propose to accomplish these goals are described in the follow sections.

Zone Based Network Insulation

In our architecture, we propose a system of "zones" which are isolated networks which are not directly connected to other zones through any kind of network layer connectivity (though they are connected through application layer gateways). Each zone has its own set of characteristics, membership requirements, privileges and restrictions. See Table 5-1. We assume that zones may coexist on a layer two network and may be isolated by VLAN protocols (layer two tagging). The frames/packets emanating from one zone are technically prohibited from being conveyed onto another zone.

Table 6-1. Example of Zone Characteristics

Characteristic	**Example Restriction**
Security designation	PHI Zone
Membership requirements	Antivirus software; host-based IDS; 802.1x authentication
Privileges	Create, access, modify PHI
Prohibition	No Internet access; no access to email system

Sources

"Sources" are network nodes that have a designation of either "static" or "dynamic". Recapping what was reported in [1], at any time *t*, every source *S* has a particular security zone classification *C(S)*. The zone classification defines the network zone to which it is attached.

Through technical and/or administrative means, the network will only permit sources to attach to the network which meet the minimum requirements like those listed in Table I.

Static Sources

From [70], packets emanate from information sources. At time *t,* every information source *S* has a classification designation *C(S).* The packets originating from these sources are tagged with the same designation. Static sources have well-defined designations and may not change during their lifetime. An example of a static source may be a hospital electronic medical record system.

Dynamic Sources

There are also dynamic sources of information. The security designation associated with dynamic sources may grow higher during operation but not lower. Formally, C(S) = i → C(S) = j, where j > i. However, the source may not make the reverse transition without executing the "decontamination" process (explained later). An example of a dynamic source system would be a general purpose workstation which, by default, is set to the lowest level of access S_0. A general purpose secure workstation may access insecure systems such as the Internet. It may also access secure systems such as the registration system in the previous example. It may not, however, access the insecure system *subsequent* to accessing the secure system. Permitting it to do so would create a path where data may have left the registration system, been recorded on the insecure workstation and then transmitted to a lower security system (and violate the "no writes down" rule of BLP). Thus, before the system may receive a packet from of high security, its own

security must be changed (in this case, $S_0 \rightarrow S_P$). Once the workstations designation is set to S_P, it may no longer send packets to targets with lower security designations.

Porthole

Portholes (as opposed to the over-used term "portal") are secure gateways that permit the access of higher security zones from lower or differing security zones. The portholes are designed to consolidate connectivity and minimize the risk footprint. Like their namesakes on ships, the porthole is/should be designed to provide an opening but not facilitate egress. Data may be viewed through a porthole but not copied (meaning that the risk of "copying and pasting" is removed). Practically and technically, however, the existence of the porthole increases the risk of data leakage (over having no access whatsoever) as data may still be intentionally leaked through screen capture software or simply by taking a picture of the screen.

Declassifiers

As described in [70], declassifiers are secure data processing mechanisms that accept as input information with security designation S_i (within a zone capable of supporting data with characteristic S_i) and output data with security designation S_j where $S_i > S_j$. For example, a declassifier may take as input several identified patient records and output statistical information.

Sanitizers

As described in [70], sanitizers are mechanisms and procedures that cleanse dynamic source nodes so that their security level and associated network zone may be "reset". The

sanitizer “decontaminates” the node of any data that should not leave the high security zone. Practically, this typically translates to erasing the long-term storage, resetting the RAM and reinstalling the operating system and application software.

Illustration

Figure 1 depicts the essence of the architecture. There are different security zones connected by portholes. In the figure, we depict a user node (a personal computing) accessing data of a high security classification from a zone of a lower security classification via portholes. The use of declassifiers is also depicted. In this example, we show how declassifiers could be used to distill abstracted patient data for querying in an i2b2 [65] system.

Security Benefits and Risks

The proposed architecture reduces the possibility of data leakage given the assumption that copying and pasting are somehow administratively prevented through porthole sessions. The total lack of layer three connectivity reduces the risk of botnet-like infections that make use of the Internet as a transport. Assuming that the zone is and remains secure, it is notable that this architecture does not prevent malware from infecting high security zones as nodes of lower security could be promoted into zones of higher security. It simply reduces the risk that malware could cause data leakage.

VLANs and Network Isolation

A fundamental assumption of this proposal is that the systems managing VLAN membership are secure and that nodes cannot directly affect the VLANs upon which their frames are transmitted.

Screen Capture Risk

All of these methods are vulnerable to bad actors who use screen capture software or cameras to capture images of information on computer screens. Even using these methods of intentional data leakage, this architecture may contribute to reducing the rate of leakage. We assume that attacks such as VLAN hopping are technically mitigated.

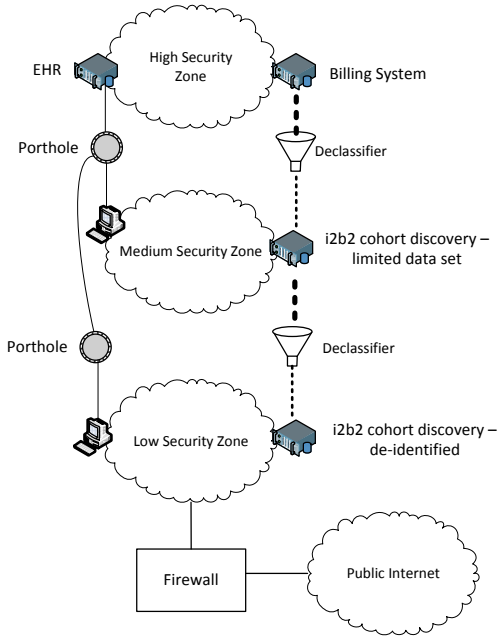

Figure 6-1. Data Leakage Protection Architecture

Declassifiers

A declassifier is an application-level interface system that takes data of a high classification and modifies it in such a way that it is useful but of a lower classification. A practical example is the transformation of transactional data from a patient billing system into a "limited data set" from which most patient identifiers have been removed but the value of the data for secondary analytics remains. Well-formed and understood data with proper data governance are essential to ensure that the declassifiers don't contribute to accidental data leakage.

Performance Considerations

Concerns for performance and the effect on the user experience motivated our proposed method for dynamically promoting a node from a lower security zone to a higher security zone. After anecdotal experimentation with using a system that was connected via multiple porthole "hops", performance degradation was perceivable. We sought to quantify the quality of experience of the user (QoE). QoE is defined differently in several papers [66-68]. We defined it as the user perceived experience associated with usage activities. We have performed other research around interactions with EHRs in [45]. Based our work and the work of others [68], we understand that user-initiated network activities involve keystrokes and mouse usage while server-initiated transmissions are updates to screen images. The end-to-end delay between two nodes on a network is a function of the sum of the delays related to network queuing, transmission, propagation and processing time [69]. In the simplified mathematical model below, the QoE, is a function of the network-induced delays plus any delays associated with application responsiveness, thus,

$$QoE_{client\text{-}server} = f(d_{app}, d_{queue} + d_{prop} + d_{proc} + d_{trans})$$

Typical end-to-end processing of user-generated events must be processed first by the intermediate (porthole) server and then (typically) cause a transmission from the porthole server to a back-end server. Thus, the QoE associated with a porthole-based session is a function of more delay contributors:

$$QoE_{porthole} = f(d_{app}, (d_{queue} + d_{prop} + d_{proc} + d_{trans})_{client\text{-}to\text{-}porthole}, (d_{queue} + d_{prop} + d_{proc} + d_{trans})_{porthole\text{-}to\text{-}app}, d_{porthole})$$

or, more generally:

$$QoE_{porthole} = f(\sum_{k=1}^{n}(d_{queue} + d_{prop} + d_{proc} + d_{trans} + d_{porthole})_k + d_{app})$$

where there are *n* portholes involved in access high security zones from low security zones.

These additional network delays and contention for the porthole service itself can cause significant decreases in the QoE. The delay increases linearly as the user is forced to traverse more "porthole hops". In modeling the system, we found that transmission and propagation delays contributed negligibly to the overall performance of the network while queuing and application processing delays (the entire network stack must be traversed twice) at the porthole host and application delays were potentially significant. This culminated in the finding that, at

times, it may be preferable for a node in one area of security to be "promoted" to a higher level of security to increase performance.

User Perceptions of Performance Impact

In the previous section, we considered the components that would contribute to end-to-end delay. We sought a way to more directly measure the quality of experience of the user. To this end, we created a simple game-like application that displayed one of five colors in a box. The user played the "game" by clicking the button that corresponded to the color in the box as quickly as possible (Fig. 2). After successfully clicking the button that corresponded to the color displayed, a random amount of time between 0.1 and 2.1 seconds would elapse before a new color was displayed. The application measured the amount of time required for the user to react to the change of color in the box. Each "game" consisted of thirty iterations. To ensure that the users' ability to memorize the location of the corresponding buttons did not affect the outcome, the local of color-corresponding buttons on the application were randomly shuffled after each color appeared. We then had the same user (the first author) use this application in varying settings to quantify the changes in user responsiveness. We executed the application under four different scenarios each with increasing "porthole" distances from the application. Each configuration was run five times to reduce the probability of an outlying test affecting the findings. The application was written in VB.Net Studio Express 2013 and was run on Windows 7. Each "hop" remotely controlled the same computer through a Windows Terminal Server (remote desktop) connection. The connectivity between the computers was not contentious and minimally over 100 Mbits/sec switched Ethernet networks.

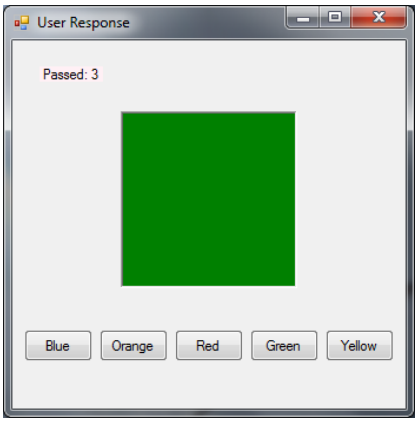

Figure 6-2. User Responsiveness Quantifying Application.

Results of User Responsiveness Testing

In our testing, each porthole hop introduced an average of 69ms of delay in terms of the user's ability to respond to application activities. The delay became perceivable when the second hop was introduced (total delay was 152ms). It is noteworthy that in telephony, delays greater than 125ms are typically considered non-real-time and perceivable. After the third hop was introduced, the delay was 209ms. At this point, mouse pointer control became "sloppy" and frustrating as reliably moving the mouse pointer from one part of the screen required more effort.

Limitations

We recognize that there are limitations to this experiment but we do not believe that they significantly reduce the validity. First, the only person to attempt the test was first author of this chapter. A more thorough test may have included more people who were blinded to what was being measured and under what configuration they were being tested. Second, the test was

performed under nearly ideal circumstances and other factors such as network contention, user contention, etc. would increase the negative effects of the porthole method.

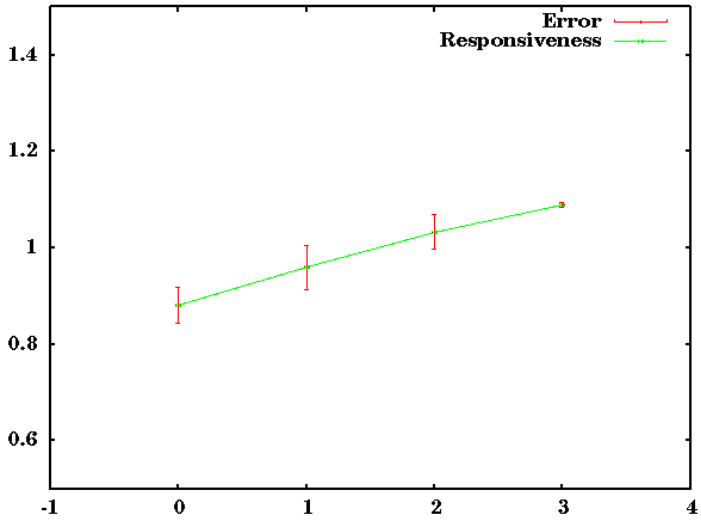

Figure 6-3. Results from User Testing.

Security Zone promotion

In this section, we propose a system to facilitate moving from one security zone to another (the zones are depicted in Fig. 1). The rationale for this is the performance degradation associated with accessing applications through portholes. The porthole based access is inherently slower than direct client-server access as the porthole-based access adds another layer which introduces non-zero delays. In describing the re-zoning of a node in the network, we will make several realistic assumptions: 1) there is some triggering event that causes the node to be re-zoned; 2) the node originally obtained its IP address through DHCP; 3) the loss and re-gain of link assertion at layer two (e.g. Ethernet) will trigger a DHCP lease request; and 4) upon re-zoning, all previously established TCP/IP network connections will be terminated. This triggering event (discussed in more detail later) could be a threshold violation or a user-initiated event. One

could envision an icon on a computer workstation where the “glass is broken” in order to escalate the user to the next security level.

Security Discussion

The goal of the network architecture is to ensure preservation of the BLP security model. This proposed process ensures that data from the high security zone does not enter the low security zone. By changing the designation of the node from “low” to “high” security, we are essentially taking low security classification data and placing it into higher security classification zone which is permissible under BLP. The risk associated with doing this is the potential for malware or other undesirable data or code to enter into the high security zone. Although this risk must be managed, it does not violate BLP. We also assume that it is not feasible for an unauthorized individual or node to promote another node. We have not yet developed the details of the promotion mechanism.

Promotion System Process

We propose the process below for promoting a node (see Figs. 4, 5, 6 and Table II). We will begin by assuming there is some triggering process that causes a “promoter” to evaluate if a node is eligible to be promoted to a zone of higher security. We make a few further assumptions with this model. First, we assume that the promoter can perfectly discriminate and determine the authenticity of a promotion request. Second, we assume that the user of the node is authorized to make the request or that the promoter can discriminate between authorized and unauthorized users. Although we have not described the mechanism to provide such assurances, many such mechanisms exist and describing them here does not enhance the proposed model.

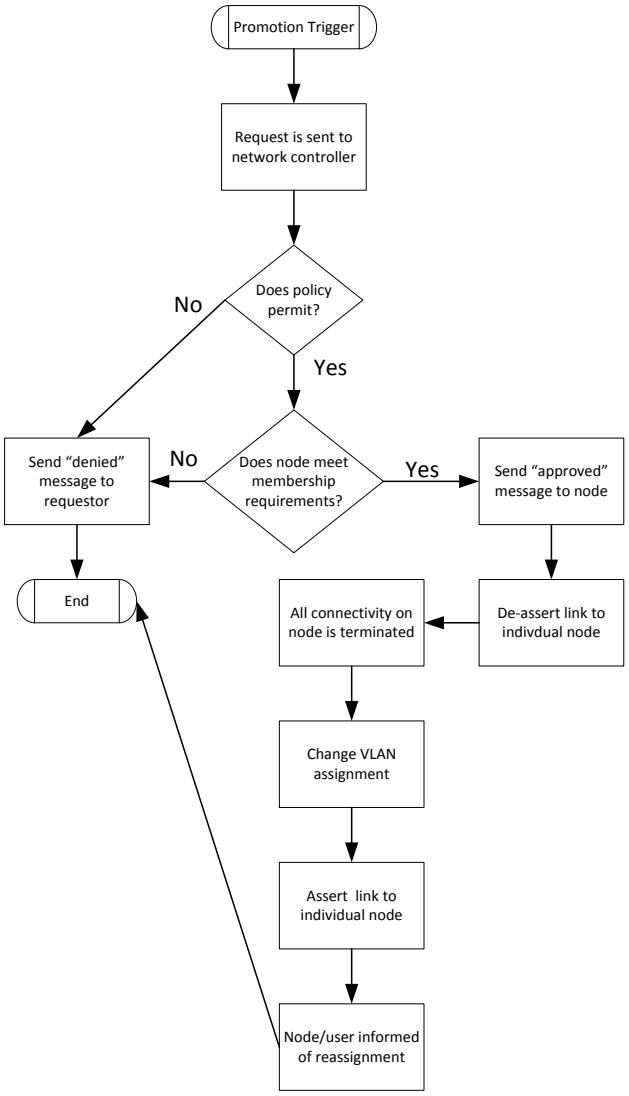

Figure 6-4. Flowchart of Promotion Method.

Requirements of the Promotion System

The node must be on the network that supports dynamic assignment of node-to-network mapping such as a switched Ethernet network with dynamic node-to-VLAN assignments. The nodes must tolerate temporarily losing their network connectivity and reassignment of IP addresses without significant interruption to the operating system; for example, we expect that the node's network changes will terminate existing TCP/IP connections; we do not anticipate that

users would need to reboot their computers or re-authenticate; such interruptions would diminish the value of this method.

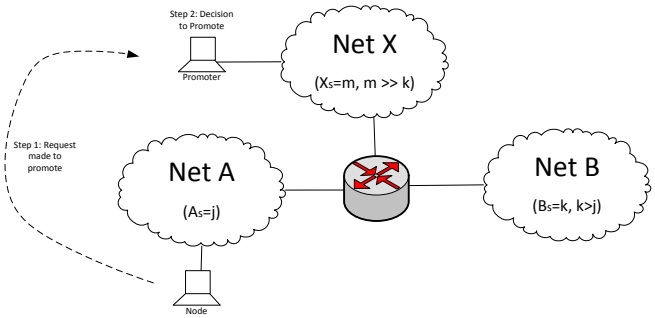

Figure 6-5. First two steps of node promotion process.

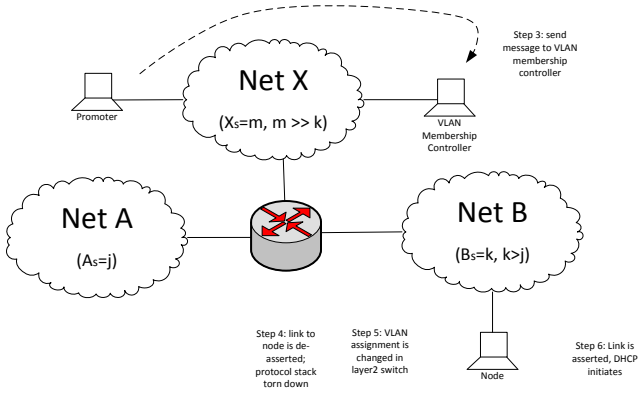

Figure 6-6. Last four steps of node promotion process.

Performance

The performance of the steps listed in the procedure is highly dependent upon implementation. We experimented with components of the process on two different operating systems (Microsoft Windows 7 and Linux/Ubuntu 10.04 LTS). The entire process, under ideal circumstances, took at least 2 seconds. Minimally, the following communications must occur

(Table II). These activities are all implementation specific and thus, cannot easily be quantified generally.

Table 6-2. Security Zone Transition Delay Contributors.

Activity	**Description**
Promotion Request	Packet from client to promotion manager
Promotion Approval	Packet from promotion server to client
VLAN Re-assignment	Promotion server to network infrastructure
Link Removal	Switch de-asserts link
Link Removal delay	Sufficient to cause network on client to deactivate
Link Assert	Switch re-asserts link, negotiates speed and duplex
DHCP Renewal	Depending upon implementation, client may attempt to renew previous IP address, which will fail
DHCP Renew Failure	Depending upon implementation, client may attempt to renew previous IP address, which will fail
DHCP Lease Request	Client will request new DHCP address
DHCP Lease Response	Server will allocate new DHCP address
DHCP Lease Accept	Client will accept new DHCP address
Initiation of Applications	Client will launch applications (application/implementation dependant)

Thus, the performance of the promotion scheme must consider the one-time performance costs of the promotion activity and the on-going performance of the node's interactions with the application post-promotion, thus:

$$\mathrm{QoE}_{\mathrm{promotion}} = f(\mathrm{d}_{\mathrm{promotion}} + \sum_{k=1}^{n}(\mathrm{d}_{\mathrm{queue}} + \mathrm{d}_{\mathrm{prop}} + \mathrm{d}_{\mathrm{proc}} + \mathrm{d}_{\mathrm{trans}})_{\mathrm{k}} + \mathrm{d}_{\mathrm{app}})$$

Comparison

We used transactional throughput as a measure of productivity and a proxy measure for QoE. The transactional throughput is defined as the number of user interactive events that could practically occur over a specific time period. As performance declines, we expect the number of possible user transactions to decline. User events are defined as key presses and mouse clicks. This is, of course, a simplification. A single user-driven event could cause a video to start playing or a screen to render several times which result in significant traffic. However, for GUI applications that largely consist of text (such as an EHR), this is typically not the case. We measured the possible transactions over a given time.

The cumulative possible transactions at time *t* is the sum of all of the possible transactions that could have occurred in each of the proceeding discrete time units for this session, less the opportunity costs associated with delays, thus:

$$\mathrm{trans_c} = \sum_{k=1}^{t} \mathrm{trans_{k_t}} - \mathrm{trans_{k_t}} * \mathrm{delay}, \mathrm{trans_c} \geq 0$$

The promotion method will be superior from the perspective that more possible transactions could have occurred when,

$$\mathrm{d_{promotion}} < \sum_{k=0}^{t} \left(\mathrm{d_{porthole_k}}\right).$$

This measure does not take into account user preferences or usability issues that might otherwise sway users' decision making processes.

Experiment

We considered several scenarios to compare the performance of the porthole based access versus the performance of the promotion mechanism. Our hypothesis is "for brief forays into zones of higher security, the porthole method would be preferable." For sustained use of systems in higher security zones, the zone promotion method would be preferable. Through our experiment, we sought to test the hypothesis and determine the value of "brief."

First, we sought to understand the network traffic associated with user-generated events. We used Wireshark 1.4.0 to capture packets related to specific porthole technology. Since "porthole technology" is not well-defined, we connected to two technologies that may be candidates for remote application portholes: a multiuser version of Microsoft Windows with Citrix and XWindows running on Redhat Linux. To test simple activities (keystrokes and mouse-clicks), we opened "notepad.exe" on the Windows machine and initiated a remotely displayed "xterm" which was tunneled through SSH to display through XWindows. We then counted how many packets were generated by each activity. Individual keystrokes (which included transmitting the keypress event and the subsequent echoing back to the remote terminal) caused, on average three packets to be transmitted. Mouse-click events tended to result in an average of twenty-two packet transmissions.

We then created a simulation in C running under Cygwin. We measured the impact of various components of delay and only considered the elements which were likely to contribute significantly and vary differently between the porthole solution and the zone promotion solution. In order to keep the simulation manageable, we made several assumptions, specifically: the inter-node distances were kept constant at 1000m, a layer three network diameter of 5, no queuing delays in the network (but varying queuing delays on the portholes and server nodes), and the

networks operate at a constant 100 Mbits/sec. We assume that the TCP sessions are already established and thus no handshake is required.

Results

Our results were previously published in [70], were heavily dependent on the local environment and assumptions and were inconclusive. Given the new information provided by the user test, we can state that based on our test bed, the decision, if based on performance alone, should be to execute the promotion process if the cumulative porthole-induced delay is expected to exceed approximately 2.0 seconds.

Discussion

The performance of connecting to protected systems is quantifiably and intuitively improved for the nodes that are members of the protected zones. However, there are drawbacks that may contribute to users' decisions not to self-promote their nodes. Once part of the protected zone, access to information assets in lower zones may be restricted or lost completely depending upon the prohibitions associated with that zone. Similarly, there are also advantages for promoting into the protected zone – for example, a user may be restricted from "copying & pasting" data – even between applications that are accessed through the porthole into the protected zone. One could also envision policy decisions that prohibit access to high security information through the porthole method. Another longer-term consideration for the user is the need to "decontaminate" the promoted node subsequent to its use in the promoted zone. There may be overall QoE costs associated with the decontamination process (time, effort) that increases their interest in the porthole based access method.

Use of this Concept in the Intercloud

Given the popularity of cloud computing and the expectation that the next generation of cloud computing will shift towards the "Intercloud," we considered how these techniques could be used in a setting where users were based within an enterprise and the information resources (e.g. servers) were based in the Intercloud.

Changes for Non-Continuous Networks

An implicit assumption in our architecture is that all nodes are members of a common set of virtual local area networks such that any node could be moved from one VLAN to another with minimal effort. This assumption is not valid in a cloud environment. With regards to cloud connectivity, we will make the assumption that there is a mechanism that permits extending connectivity to networks in the cloud over virtual private networks (VPNs) without reducing the security level of those networks. For example, there are existing services (e.g. Amazon's Virtual Private Cloud) that permit the creation of a virtual network within the public cloud provider's cloudspace such that the virtual nodes are network-logically isolated from all other machines and the connectivity of the virtual network is defined (and potentially severely limited) by the cloud customer's preferences (see Fig 7.). To reduce the risk of violating BLP, we can further assume that these routers are used only for this connectivity.

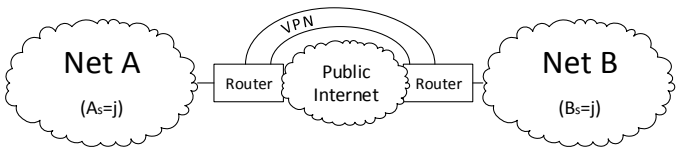

Figure 6-7. The customer network (Net A) and the cloud-based network (Net B) are connected via VPN.

Concept Revisited

If we assume that there are networks with the same security classification at both the enterprise and hosted in the Intercloud and that the two networks are connected through highly scrutinized VPN routers, the model does not change significantly (see Fig. 8). Access occurs in the same way that it occurred in the purely enterprise model presented in this chapter. Performance would be impacted by the introduction of the overhead associated with the VPN (e.g. encryption, decryption, reduced transmission unit size, etc.) and network-induced delays due to the introduction of the public Internet. Conceptually, however, the model is essentially unchanged.

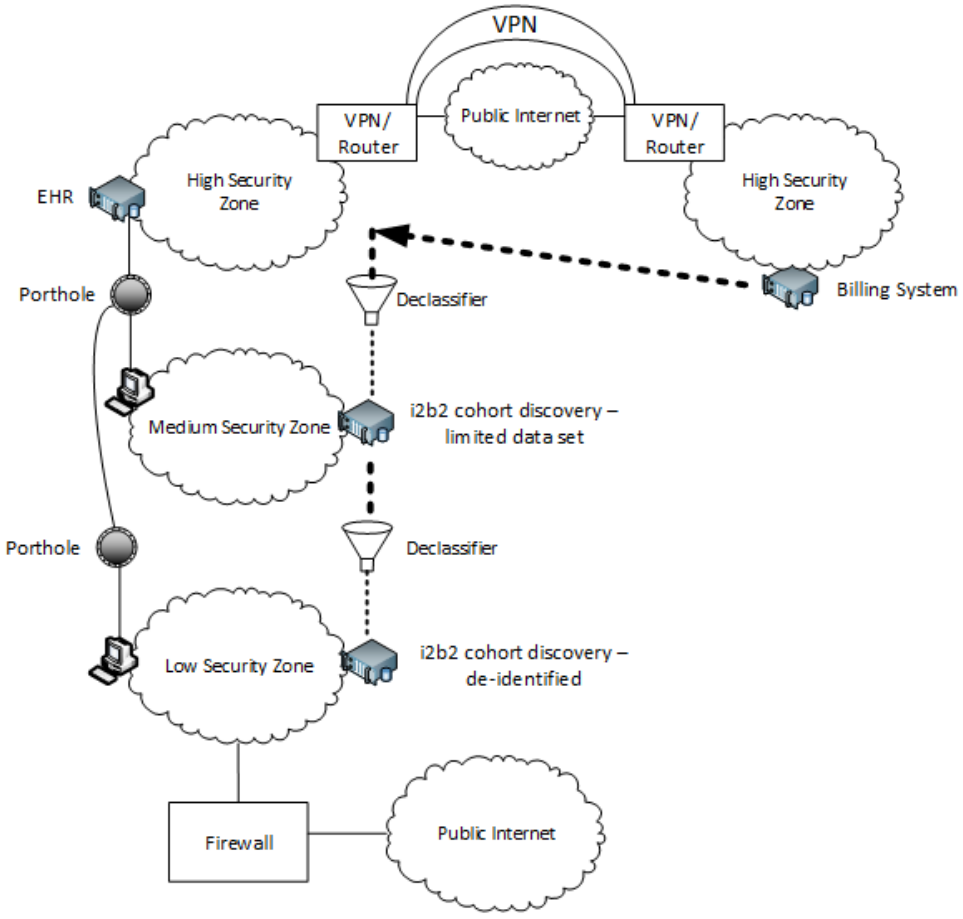

Figure 6-8: The "high security" zone consists of two subnets (one enterprise based, one cloud based) joined by a VPN router

Risk and Trust

In this concept, the enterprise must establish the normal and customary relationship with the cloud provider. The enterprise must be able to trust that the cloud provider will act as it expects and provide appropriate security. In this model, we assume that the VPN and the VPN routers are secure. In our model, we advocate that the VPN routers be of single purpose. Although one could conceptually use a single device to convey different classes of traffic, doing so reduces the model's adherence to BLP and introduces complexity which could potentially be exploited.

Future work

Fully implementing this model would require new software. The mechanism associated with node promotion would need to be fully constructed. Since each "zone" is insulated from other layer three networks, network services that are often assumed (e.g. domain name services, authentication services, etc.) would need to be duplicated within each zone. We have not yet considered how migration in the Intercloud would be managed. A cloud user must be able to migrate resources from one cloud provider to another to take advantage of the flexibility of the Intercloud.

Conclusion

This chapter introduces two concepts: i) a VLAN based security architecture to facilitate compliance with the rules of BLP while facilitating the pragmatic needs of organizations with varying classifications of data and ii) a system for promoting designated nodes to higher security zones to ensure critical access. The architecture facilitates increased protection to prevent

accidental data leakage. The porthole and promotion methods both facilitate escalated access to sensitive data with different performance characteristics, access benefits and costs. The system, as described, could potentially be extended to the Intercloud.

Chapter 7 Addressing the Barriers to the use of the Intercloud for Health Care

Introduction

The Intercloud

The Intercloud is the envisioned unification of various cloud computing services into a "cloud of clouds". In the Intercloud, a customer may easily migrate their computing usage from one cloud provider to another. Ideally, the customer would be able to migrate their computing usage between providers based solely on the suitability of the "cloud computing" product for their needs. They would be able to migrate their usage without interrupting services to their customers. Such a dynamic and commoditized market could increase the efficiency of the cloud computing market and ultimately result in downward price pressures and standardization which could result in further efficiencies and costs savings.

Health Care and Cloud Computing

Health care organizations are notoriously slow to adopt new technologies [44] and the adoption of cloud computing has been no different [45]. In the United States, the Health Insurance Portability and Accountability Act and its successor laws, amendments and regulations (collectively known as HIPAA) define the administrative and technical safeguards required to manage protected health data. HIPAA requires that health care providers establish contractual relationships with organizations that own computer systems upon which the health care providers' data is stored. These contracts are called "Business Associate Agreements" (BAAs) and minimally impart certain responsibilities and potential liabilities onto the hosting provider.

The legal necessity for a BAA between health care provider and computing was clarified with the HIPAA omnibus rule issued early in January of 2013. It is noteworthy that HIPAA has been amended several times and enforcement activities both in terms of volume and individual penalties for lack of compliance have increased.

Regulatory Contractual Requirements

The requirement that a BAA exist between the health care provider and the computing provider represents a substantial barrier to the use of cloud computing. Many cloud computing providers either refuse to sign a BAA or will only sign a BAA authored by their own attorneys that shifts most liability to the cloud computing customer. This is at least in part attributed to the unknown risk being assumed by the cloud computing provider. Constructing an automated method to ensure the existence of a legally durable BAA between the cloud customer (the health care provider) and the cloud provider (some entity in the Intercloud) will be a critical perquisite for the use of Intercloud by health care providers.

Security Requirements

The leakage of health care data can cause substantial regulatory or reputational cost. Although not a health care entity, the security breach at Target Corporation, a US-based retailer, is said to have cost the company 4.7% of its holiday season sales in 2013 and the CEO and CIO their jobs [71]. Although no unbiased published assessments quantifying the damage to a health care entity's reputation and business were found when researching this chapter, several penalties associated with security breaches have been made publicly available. New York Presbyterian

Hospital and Columbia University were jointly fined $4.8M for accidentally leaking health data associated with 6800 patients [72]. Security must be an imperative for health care organizations.

Performance Requirements

Given the modern pressures of resource-constrained health care, clinicians have little time to spend with patients. Underperforming computer systems increase provider frustration and reduce the amount of time providers can spend with patients which may contribute to lowering the quality of care. System outages require that clinicians make clinical decisions without all available information and without the benefit of automated alerts – which may also decrease the quality of care. Therefore, performance and overall reliability are important considerations when using the Intercloud to host health care applications.

Focus

Within this chapter, we will examine the issues that impede the adoption of the Intercloud in health care. Specifically, we will propose essential components to support health care in the Intercloud and describe a method by which a cloud customer may make automated decisions about transitioning from one cloud provider to another.

Previous Work

There has been previous work seeking to define the architecture for the Intercloud. Demchenko with other authors wrote several papers that focused on a proposed cloud architecture consisting of a set of services from a generalized perspective based on the work of various

standards bodies [4, 73, 74]. Their initial architecture focused on a general purpose set of services that could be used for any type of cloud provisioning (IaaS, PaaS, and SaaS). Our approach differs in that we examined the needs on a "bottom-up" basis specifically with health care applications and associated regulations in mind. Wlodarczyk et al have also considered the potential benefits and social implications as well as the security concerns with the use of cloud computing for health care data [6]. The computational cost of virtual machine migration has also been previously studied. Dargie considered how to estimate the temporal costs of live virtual machine migrations using shared disks in [75]. He and others [76, 77] also considered the energy costs associated with live migration of virtual machines. Katsipoulakis et al proposed a novel file system which transports and commits changes to migrating virtual machines [78]. We have previously considered performance issues associated with managing health care data in a hybrid computing cloud [50].

Proposal

We constructed an architecture that includes the essential components required to use the Intercloud by health care entities. As the decision-making process about how and when to transition from one cloud provider to another is important, we chose to focus on this area.

Architectural Components

We contemplated the required components from the perspective of the cloud computing customer (Figure 6-1).

- Intercloud – the set of all cloud providers between which a cloud computing customer could transition virtual machines or sets of virtual machines

- Intercloud metahypervisor – analogous to a normal hypervisor, the metahypervisor acts as the agent of the cloud consumer (the organization that is paying for the use of the cloud resources); the goal of the intercloud hypervisor is to optimize its organization's use as defined by a cost/benefit function with constraints.
- Market scanner – this customer-controlled component would be used to determine the spot price of different cloud providers' services.
- Cloud marketplace – similar to a stock exchange or health insurance marketplace, the cloud marketplace would be an independently controlled entity with which all participating cloud computing providers agree to list and maintain their current prices. This would obviate the need for a sophisticated customer-based market scanner.
- Cloud provider – a single entity that provides public cloud services (e.g. Amazon EC2, Microsoft Azure, etc.)
- Private cloud – a cloud within the control of the cloud customer entity
- Performance Proxy – one or more systems that are hosted by each cloud provider that facilitate testing inter-provider (and customer-to-provider) performance prior to making a transition

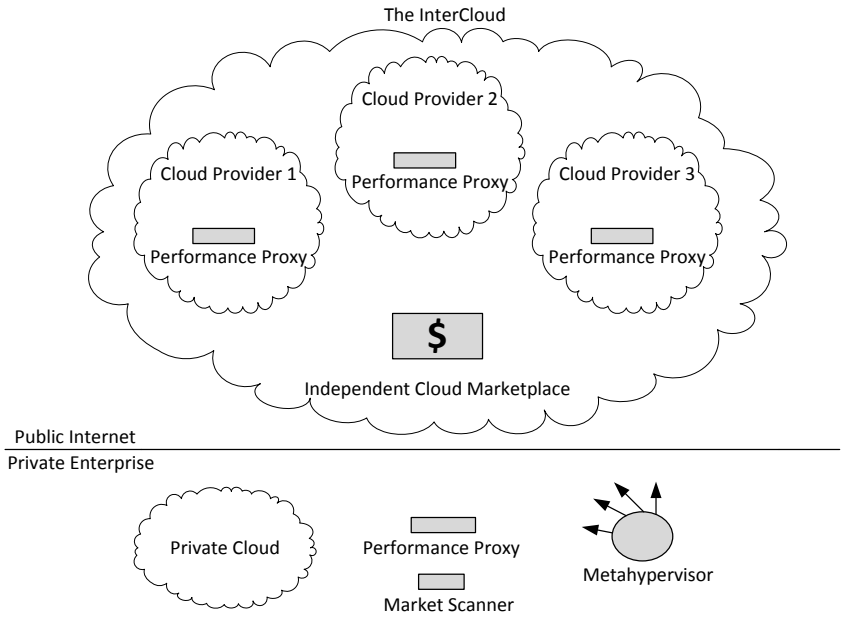

Figure 7-1. Proposed Intercloud Architectural Components.

Process

1. Trigger – a trigger occurs to compel the intercloud metahypervisor, who acts on the behalf of the "owner" of the virtual machine, to re-evaluate its current cloud organization. The trigger is based on policy; a policy may be based on any real-time statistic or a arbitrarily set timer (Figure 7-2).
2. Market scan – subsequent to the trigger process the intercloud metahypervisor evaluates the market of possible cloud computing platforms. Given the current state of technology, this may involve polling each and every possible sourcing option for pricing and performance information. In the future, one might envision a centralized automatic marketplace that would track cloud spot pricing given certain parameters (also Figure 7-2). Ultimately, one could envision an organized computing exchange market that is

similar to modern stock exchanges for the pricing, buying and selling of commoditize computing capacity.

3. Candidate selection – After completing the market scan, the Intercloud metahypervisor evaluates if it is worthwhile to continue the search for alternatives. It is possible that the metahypervisor will determine that any transition would not be cost effective; for example, the intercloud metahypervisor may determine that all possible alternatives are of higher operational cost (Figure 7-3).
4. Contract negotiation – for each of the possible candidates, the Intercloud metahypervisor may engage in contract negotiations; depending upon the sophistication of the cloud market place, certain pricing may only be knowable after a contract negotiation. For example, if a cloud provider is made aware that the customer is going to be storing a million patient records, they may need to purchase cyberinsurance for the perceived risk associated with housing records of this kind. Since the effort (computational or human) involved with negotiating a contract may be non-trivial, it is reasonable that a fee may be associated with the construction of a contract (Figure 7-3).

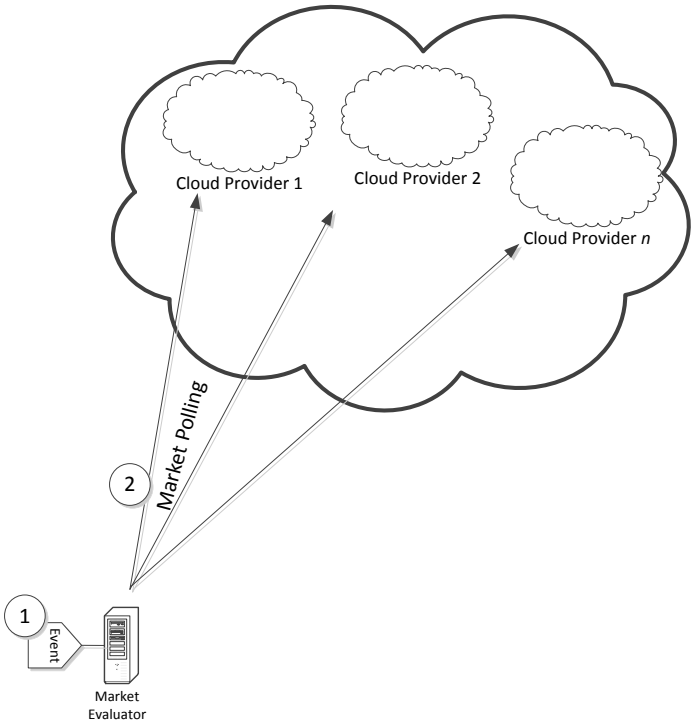

Figure 7-2. Trigger and Market Scan.

5. Transition decision – the metahypervisor, now given a list of contract proposals to consider, may chose to select all or none of the proposed contracts
6. Execute Contract – select one of the candidate proposals and accept terms and conditions
7. Environment setup – with the recently selected hosting service, establish environmental framework; this could include establishing security parameters (e.g. capabilities and restrictions) Network setup – establish VPNs or other required network parameters with new cloud provider.
8. Live migration – migrate "live" environments from old providers to new providers as much as possible.
9. Validate operation – perform operating system and application level validation of virtual machines

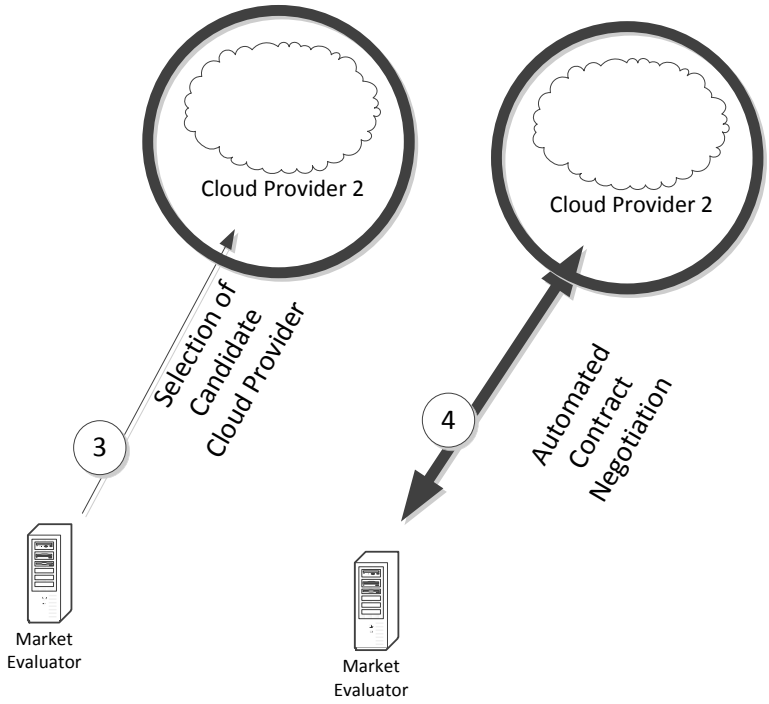

Figure 7-3. Market Evaluation and Contract Formation

10. Securely delete – securely delete old virtual machine instances from old cloud provider
11. Shutdown networking – discontinue network relationship with old provider
12. Teardown environment – delete old environment from old provider
13. Terminate contract – terminate the ephemeral components of old contract

Transition Decision Making

An important aspect of this process is the decision criteria for switching versus non-switching. The metahypervisor must consider the transition costs, the steady-state costs of the new configuration and the benefits of the new configuration over the life of configuration. There is a cost and benefit function with each state of the system. There is also a cost associated with transitioning between different states. The cost of the steady-state configuration is a function the

price paid to the provider for access, any incremental costs paid to the provider for use above and beyond the baseline. These costs are reasonably straight-forward to calculate. The benefits (which could be described and calculated as a negative cost) are a function of the performance and potentially other factors (such as a reliability, ease of administration, etc.). Benefit functions may be complicated and consider non-obvious factors such as reputation, security, geographic diversity, geographic bounds, etc. Although complex, they are easily known and a value can be placed upon them. Attributing a value to these different benefits is an exercise that must be executed by the cloud customer. The time during the transition is costly. During the live migration, both the old and new service provider must be active (Figure 7-4).

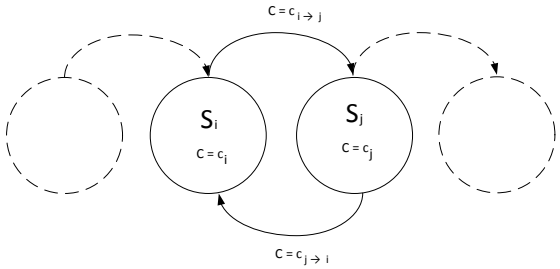

Figure 7-4. Transition Costs.

The performance of the machine being migrated is also likely to be impacted. Also, active connections to clients and other virtual machines may be impacted. Therefore, it is important to consider the transition costs prior to transitioning the machines. A key component of the cost of migration is the effective inter-provider bandwidth. According to [75], diminishing bandwidth exponentially impacts the amount of time required to perform a live migration of a virtual machine. Therefore, calculating the true cost of migration would require the ability to measure the effective bandwidth between the providers. Obviously, due to contention, changing routes and (albeit more slowly) change policies, the effective bandwidth between two sites changes frequently. However, to make this problem tractable, we will assume that the bandwidth

available immediately after the test is the bandwidth that will remain available during the duration of the possible transfer.

Transition Decision Making Prediction

Based on the work of Dargie [75] and our own experimentation, we constructed a model that estimates the amount of time required to perform a live migration of a running virtual machine. The time estimated to migrate a machine ($t_{migrate}$) from one cloud (s1) to another (s2) is the sum of the size of the disk image and the amount of RAM currently occupied by the VM over the effective bandwidth between s1 and s2, thus:

$$t_{migrate} = (S_{memory} + S_{disk}) / B_{s1 \rightarrow s2.}$$

We tested the accuracy of the predictions of this formula using the configuration and methods below.

Configuration

We created a small cloud consisting of two identical physical computers (Dell PowerEdge 620) with two Intel Xeon E5-2620 2GHz processors and 64 GBytes of RAM). Both physical computers were running Redhat Enterprise Server version 6.6. We used the libvirt version 0.10.2 as the virtualization API with Qemu/KVM as the hypervisor stack.

Estimations

We used iPerf version 2.0.5 using default settings for both the client and server to estimate the effective bandwidth between the host machines. This served as the "performance

proxy" described in the architecture above. To reduce the risk of incorrect readings from transient network traffic, we used the average of three iPerf tests to calculate the bandwidth used by the prediction formula ($B_{s1 \rightarrow s2}$). We determined the host memory consumed by the VM by examining the resident set size for the process that corresponded to the migrating virtual machine (S_{memory}). To determine the size of the VM's disk we simply examined the number of bytes occupied on the host systems filesystem as reported by the filesystem. We specifically chose to use a non-variable disk format ("RAW") to simplify this measurement. Unlike the experiments in [75], we assumed that the disk image must also be copied as that is a more realistic assumption for migrations in the Intercloud as opposed to a local cloud where shared storage is more likely.

Variables

We varied the following characteristics: bandwidth between physical servers, the level of activity of the migrating virtual machines, host system memory consumption of the virtual machines, and disk activity of the virtual machine. We used different distributions of Ubuntu Linux for the different virtual machine operating systems. All of the virtual hosts were configured to be bridged to the host computers network and both host computers were on the same subnet as to remove any issues associated with the migration of non-local IP addresses. To ensure that the "live migration" occurred without perceivable performance impacts or interruptions, we used the objective determinant of "no lost ICMP packets" to indicate that the migration was, in fact, "live". Subjectively, we also tested the migration of the virtual machine while it displayed streaming video content from Youtube. We judged the migrations, in all test cases to be "live".

Results

Varying the bandwidth available between the physical hosts did not affect the accuracy of the prediction. Consistently, the predicted amount of time for a VM migration tracked well with the actual migration time. We varied the available bandwidth by introducing link contention through various traffic generators as well as hard-coding the Ethernet interface to varying (10, 100, 1000 Mbits/sec) speeds. See Figure 7-5 (log scales) and Figure 7-6 (linear scales). This would seem to contradict [75] which indicated that migration times varied exponentially with the available bandwidth.

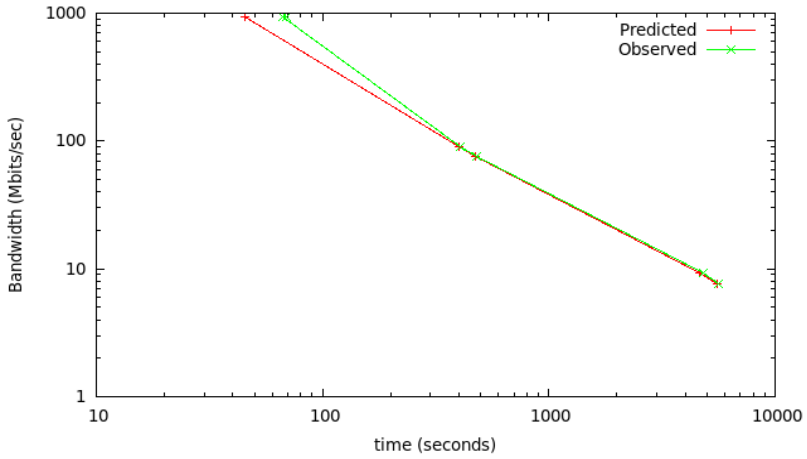

Figure 7-5. Predicted and Observed for migration (log scales both axes).

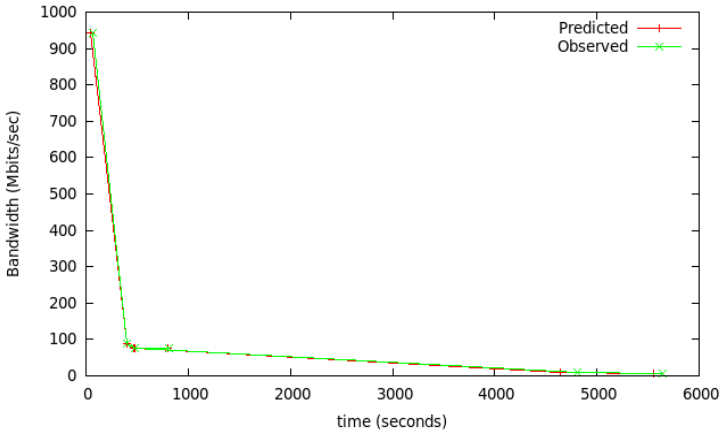

Figure 7-6. Predicted and Observed for migration (linear scales both axes).

We then varied the amount of RAM consumed by the virtual machines. We created a VM that was limited to 16 GB. The qemu/KVM hypervisor does not allocate RAM to a VM until it is actually accessed by the guest operating system. Therefore, we could vary the actual amount of RAM consumed simply by accessing a varying amount of RAM within the VM. This was done with a simple C program that incremented through an array of varying sizes. The size of the RAM also tracked linearly (see Figure 7-7).

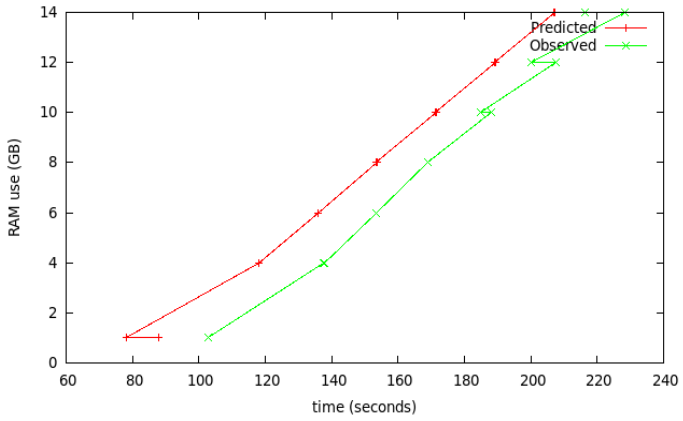

Figure 7-7. Varying the RAM utilized of the VM.

We also considered the impact of "memory intensity" and locality and predictability of RAM references. We did this by using the same array-traversing C program mentioned above. We created arrays of varying size (0, 4, 6, 8, 10, 12 and 14 GB) and accessed the RAM in predictable loops. To observe the difference between "local" and "distant" reference, we incremented the arrays differently (i.e. moving by one byte increments or 1024 byte increments). We also randomly referenced and updated elements within the large arrays. There were two significant results: 1) there were no appreciable differences between local, distant and random memory access in terms of the impact on VM migration and 2) we were able to construct an "adversary" memory referencing program that prevented the VM from successfully migrating. The size of referenceable memory region determined the success of the adversary process. Specifically, an adversary that constantly paged through memory of less than 81 MB of address space was migrated as predicted while a process that consumed 82 MB could prevent the VM from migrating. Determining the reason for this specific boundary was not in the scope of this research.

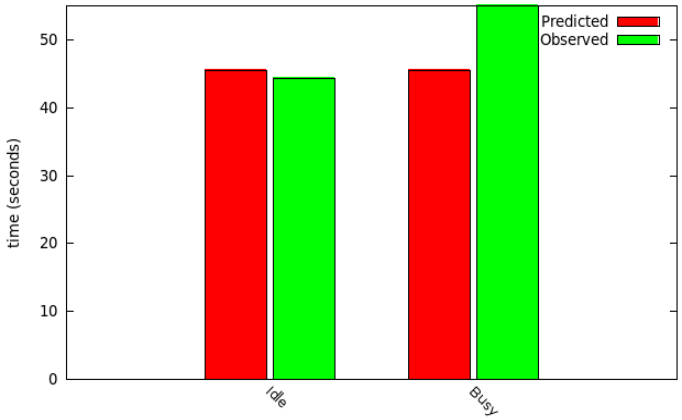

Figure 7-8. Idle and Busy Disks.

Finally, we observed the impact of "busy disks" within the VM. To create a high degree of disk I/O, we used the disk benchmarking program Bonnie++. The "idle" VMs were simply the

base operating system running with no user or maintenance processes executing. Although our testing shouldn't be considered exhaustive, we determined that increasing the disk load impacted the accuracy of the predicted migration time.

Discussion

Under a variety of circumstances, our prediction system was minimally 85% accurate except in the case where the migration failed completely due to the presence of an adversary program on the VM. Given more quantified information about other statistics available to physical host (such as disk references), the accuracy of the prediction could be improved. The predictions could also be negatively impacted by transient network contention or network policy-based traffic regulation (e.g. "leaky bucket" schemes). The determination of the minimal memory allocation required for an adversary to successfully thwart a VM from migrating may also be important to the future development of metahypervisors. Although the prediction method tracked well with actual transfer times, we do not know if these findings are broadly applicable to all or most cloud infrastructures or are specific the libvirt/qemu/KVM configuration on which we experimented. Finally, the "performance proxy" itself introduces potentials for abuse. Such a mechanism should require a minimal but obligatory financial obligation before access is granted to help reduce the risk of a denial-of-service attack.

Conclusion

We have introduced the essential elements of an architecture to facilitate the use of health care data in the Intercloud; we have tested one component of that architecture – the component that predicts the amount of time required to migrate a VM; incidentally, we have also identified

specific circumstances under which a VM can house an adversary process that thwarts VM migration.

Future Work

Although it is not clear how an adversary would benefit from preventing a VM from migrating, it is clear that -- at least in the system we built – it is possible for an adversary to do so. This is particularly easy if that adversary has command-interpreter (shell) access to the guest VM. Furthermore, it would not be difficult to imagine legitimate algorithms that cycle repeatedly through large regions of memory that would accidentally thwart VM migration. A thwarted VM migration, in the absence of some sort of watchdog process, essentially doubles the cost of cloud computing to the cloud computing customer as the VM lies in limbo between two physical servers. From a security perspective, work should ensure that "migration limbo" can be avoided. It is possible that the innovative disk migration mechanism described by Katsipoulakis in [78] would address the "migration limbo" problem.

Although we mentioned the use of automated contract formation, we did not explore the details deeply. To fulfill regulatory requirements, durable contracts between cloud user and cloud provider *without human intervention* will become essential if the vision of the Intercloud is to be useful in certain important industries such as health care.

Chapter 8 Conclusions and Future Work

Contributions

This dissertation has made several contributions to computer science and health care informatics. Chapter two provided insights into the current thinking of technology leaders in health care and their attitudes and propensities towards the use of cloud computing. Chapter three introduced a novel algorithm to apportion sufficient hardware resources to a private cloud. Leveraging the predictability of health care computing, chapter four introduced LABOH, an algorithm for rebalancing the VM load across different physical computers in anticipation of changing utilization. Chapter 5 proposed a VLAN-based method for ensuring practical adherence to the Bell-LaPadula multilevel security model and a node-promotion method to protect users from performance bottlenecks. Chapter 6 expanded this concept to the Intercloud and improved the assessment of the user-perceived performance degradation. Chapter 7 introduced a novel architecture for deploying health care information to the Intercloud and specifically proposed a method to anticipate how much time is required to migrate a full-disk VM from one cloud provider to another. While executing experiments, we incidentally discovered that a user-process could prevent the live migration of a virtual machine.

Future Work

Based on the reception of the various contributions, we propose the future work first concentrate on implementing the data leakage protection mechanisms described in chapters 5 and 6. Health care's use of the Intercloud must be built upon a solid foundation of security. Chapter 7

describes a locus of work to follow. Specifically, the future use of the Intercloud for health care will require that there be a method by which legally durable contracts can be formed between health care providers and cloud providers. This too is foundational. In the absence of legislative changes, the contracts (business associate agreements) must exist. Therefore, without the technology to rapidly create and terminate these contracts, the United States health care industry will not be able to take advantage of the flexibility and dynamism that is meant to be provided by the Intercloud. Undoubtably, there will continue to be health care enterprises that choose to continue to, at least partially, to self-host their infrastructure. For those organizations, determining the appropriate amount of computational resources to place into their private cloud is important. The algorithm presented in chapter 3 can assist those organizations but a user-friendly way to perform this analysis would be more helpful. Similarly, an application program that could quantify the level of predictability of an infrastructure and then react to it – perhaps built into a hypervisor – would be more useful than just the algorithm which was presented in chapter 4.

Closing Thoughts

Certain aspects of the health care industry are not unique. Health care is not the only industry that is information-intense, security-conscious and reasonably predictable. Many of the contributions of this dissertation may be reused for problems in other industries. Health care may be unique in that, as an industry, it is relatively slow to adopt new technologies due to safety and regulatory concerns. The use of the Intercloud may help in reducing the overall cost of health care but concerns continue to persist. Health care leaders continue to wrestle with often ambiguous rules and regulations, concerns about reliability and data ownership. The technical solutions proposed in this dissertation will only be useful in the context of a clear and well- promulgated administrative frameworks.

Much of the concerns about information security become less important if it were possible to compute upon data in the public cloud without disclosing those data. An efficient solution to fully homomorphic encryption would presumably address all security concerns of technology leaders, speed to the adoption of the Intercloud and, as economies of scale were achieved, lower the cost of computing generally.

I will take this opportunity to speculate about the future of health care and its use of technology.

Security as the Impetus

Information security within health care enterprises has been shown to be immature. In just over a decade, the use of digital systems in health care with all their benefits and flaws has skyrocketed and become heavily regulated. At times, health information technology has been promoted as the panacea that will cure the United States' growing health care cost problems. Security and the damages associated with failures in security will encourage more rigorous data security management practices within health care organizations. As security methodologies improve throughout the industry, health care organizations will become better at classifying data and systems to ensure mature data leakage prevention mechanisms.

Data Architecture, Governance and Research

Improved data classification, data architecture, data management and data governance will lead to high quality and more useful information. The information derived from health care data will be used to quickly identify incipient trends and reduce health care errors by increasing health care discipline. New information sources such as whole genome sequences from healthy

and diseased tissue combined with our ever-increase corpus of knowledge will facilitate better treatments.

Towards Cures

Well-described data from billions of patients combined with novel and compliant and ethical research protocols will help create computationally-generated hypotheses creating novel cause-effect-cure triplets.

In the long run, computational capacity will limit the rate at which patients are diagnosed, treated and new knowledge is elucidated. The safe and efficient use of cloud computing is one path towards increasing the availability of computational power for health care. This dissertation contributes towards that path.

Appendix

For completeness, a brief description of the Bell-LaPadula model (BLP) is included herein. The Bell-LaPadula security model, published in 1973 as a MITRE technical report in [79] is one of the earliest efforts to construct a formal model to ensure the confidentiality of information that exists in an environment where data of different levels of security are used. Motivated by the need to ensure the confidentiality of data in the military, it is one of the most widely referenced formal security models [80]. The BLP model was designed to be a mathematically-based, formal system that could be technically implemented. The BLP model can be used to ensure that actors ("subjects" in the BLP vernacular) could neither access information ("objects") outside of their authority nor cause information to move from higher security "areas" to lower security areas. This appendix will draw on examples related to the types of data typically stored at an academic health center.

Ordering and Categories

The BLP model contains several critical assumptions about the metadata associated with data (objects) and actors (subjects). All objects have a well-defined set of classifications and categories within the classifications. An ordering exists upon the classifications (e.g. c_0, c_1, c_2,…, c_n, such that $c_i < c_j$, meaning that the sensitivity of data in category *i* is strictly more sensitive than data with a sensitivity of category *j*). For example, in an academic health organization, patient data may be the highest classification (most secure), followed by a HIPAA-defined limited datasets (patient data limited to dates and five-digit zip codes), followed by high-level patient statistics. Furthermore, different categories may coexist at different classification levels. For example, student grade data (regulated by FERPA) and patient data (regulated by HIPAA) could

both be classified at the same security level but fall into different categories because different subjects (users) would have differing access to each type of data and intermingling the two different data is prohibited. The security model can be represented as a rooted directed tree (see Figure A-1).

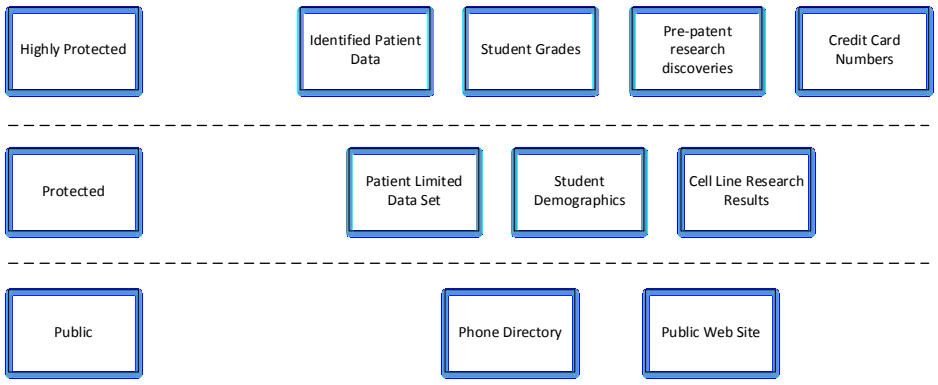

Figure A-1. Classification Levels and Varying Categories

Properties

There are three primary properties of a system that is compliant with the BLP security model. The first two properties constitute the mandatory access control definitions of BLP. The first property is called the ss-property or "simple security" property. The simple security property indicates that a subject (an actor such as a user or computer program) cannot access information at a classification above his, her or its own. In the context of an academic health center, a biostatistician with "limited data set only" privileges would not be able to access identified health data. The second property is called the "*-property" (said "star property") and indicates that a subject may not write to an area (i.e. zone, file system, etc.) of a lower classification. This prevents subjects from causing data to be leaked from a higher security classification zone to a lower security classification zone. The third property, the "ds-property", indicates that a subject may grant access to an object to another subject so long as the previous two access policies are

not violated. This rule ensures that BLP supports discretionary access policies. It is noteworthy that a subject may operate with different privileges at different times but some sets of privileges, if used at the same time, may conflict with the first or second properties.

Operation

The BLP model was envisioned to operate through a set of well-defined processes and data structures within a single operating system. The discussion of these functions and data structures is beyond the scope of this reference material. The reader is directed to Stallings and Brown's coverage of this material [80] for a thorough explanation.

REFERENCES

[1] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," in *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on*, 2009, pp. 328-336.

[2] J. Abawajy, "Determining Service Trustworthiness in Intercloud Computing Environments," in *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, 2009, pp. 784-788.

[3] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Security and Cloud Computing: InterCloud Identity Management Infrastructure," in *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on*, pp. 263-265.

[4] Y. Demchenko, M. X. Makkes, R. Strijkers, and C. de Laat, "Intercloud Architecture for interoperability and integration," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pp. 666-674.

[5] M. Gall, A. Schneider, and N. Fallenbeck, "An Architecture for Community Clouds Using Concepts of the Intercloud," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pp. 74-81.

[6] T. W. Wlodarczyk, R. Chunming, and D. Waage, "Challenges in healthcare and welfare intercloud," in *Digital Content, Multimedia Technology and its Applications (IDCTA), 2011 7th International Conference on*, 2011, pp. 45-48.

[7] J. Cohen. (2012, The Patient of the Future. *MIT Technology Review*. Available: http://www.technologyreview.com/featuredstory/426968/the-patient-of-the-future/

[8] Z. Shuai, Z. Shufen, C. Xuebin, and H. Xiuzhen, "Cloud Computing Research and Development Trend," in *Future Networks, 2010. ICFN '10. Second International Conference on*, pp. 93-97.

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski*, et al.*, "A view of cloud computing," *Commun. ACM,* vol. 53, pp. 50-58, 2010.

[10] (2013, 4/28/2013). *Agner Krarup Erlang*. Available: en.wikipedia.org/wiki/Agner_Krarup_Erlang

[11] (2013, 4/28/2013). *Leonard Kleinrock*. Available: http://en.wikipedia.org/wiki/Leonard_Kleinrock

[12] B. R. Haverkort, *Performability Modelling : Techniques and Tools*. Chichester, UK ; New York: Wiley, 2001.

[13] R. Supnik, "Debugging Under Simulation," in *Debugging Tecniques in Large Systems*, R. Rustin, Ed., ed: Prentice-Hall, 1971.

[14] C. J. Young, "Extended architecture and Hypervisor performance," presented at the Proceedings of the workshop on virtual computer systems, Cambridge, Massachusetts, USA, 1973.

[15] R. Rauscher, "Server virtualization. There's a way around supporting multiple servers and operating systems," *Healthc Inform,* vol. 21, pp. 66, 68, Oct 2004.

[16] E. J. Schweitzer, "Reconciliation of the cloud computing model with US federal electronic health record regulations," *J Am Med Inform Assoc,* vol. 19, pp. 161-5, Mar-Apr.

[17] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski*, et al.*, "Above the Clouds: A Berkeley View of Cloud Computing," EECS Department, University of California, Berkeley UCB/EECS-2009-28, February 10 2009.

[18] "Modifications to the HIPAA Privacy,Security, Enforcement, and Breach Notification Rules Under the Health Information Technology for Economic and Clinical Health Act and the Genetic Information Nondiscrimination Act; Other Modifications to the HIPAA Rules " in *45 CFR Parts 160 and 164*, ed, 2013.

[19] (2013, 4/28/2013). *HIPAA BAA Agreement, Omnibus rules new as of Jan 2013*. Available: https://forums.aws.amazon.com/thread.jspa?messageID=434601

[20] A. Gambi and G. Toffetti, "Modeling cloud performance with kriging," presented at the Proceedings of the 2012 International Conference on Software Engineering, Zurich, Switzerland.

[21] W. D. Smith, "Characterizing Cloud Performance with TPC Benchmarks," in *4th TPC Technology Conference, TPCTC 2012*, Berlin, Germany, pp. 189-96.

[22] N. Yigitbasi, A. Iosup, D. Epema, and S. Ostermann, "C-meter: a framework for performance analysis of computing clouds," Piscataway, NJ, USA, 2009, pp. 472-7.

[23] P. C. Brebner, "Is your cloud elastic enough?: performance modelling the elasticity of infrastructure as a service (IaaS) cloud applications," presented at the Proceedings of the third joint WOSP/SIPEW international conference on Performance Engineering, Boston, Massachusetts, USA.

[24] T. N. B. Duong, X. Li, R. S. M. Goh, X. Tang, and W. Cai, "QoS-Aware Revenue-Cost Optimization for Latency-Sensitive Services in IaaS Clouds," presented at the Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, Dublin, Ireland.

[25] E. Cecchet, R. Singh, U. Sharma, and P. Shenoy, "Dolly: virtualization-driven database provisioning for the cloud," *SIGPLAN Not.,* vol. 46, pp. 51-62.

[26] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, and M. M. Swift, "Resource-freeing attacks: improve your cloud performance (at your neighbor's expense)," presented at the Proceedings of the 2012 ACM conference on Computer and communications security, Raleigh, North Carolina, USA.

[27] Y. Tan, Y. Lu, and C. H. Xia, "Provisioning for large scale cloud computing services," presented at the Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems, London, England, UK.

[28] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," Munich, Germany, 2011, pp. 115-131.

[29] Z. Zhao, K. Hwang, and J. Villeta, "Game cloud design with virtualized CPU/GPU servers and initial performance results," presented at the Proceedings of the 3rd workshop on Scientific Cloud Computing Date, Delft, The Netherlands.

[30] N. Deng, C. Stewart, D. Gmach, M. Arlitt, and J. Kelley, "Adaptive green hosting," presented at the Proceedings of the 9th international conference on Autonomic computing, San Jose, California, USA.

[31] P. Papakos, L. Capra, and D. S. Rosenblum, "VOLARE: context-aware adaptive cloud service discovery for mobile systems," presented at the Proceedings of the 9th International Workshop on Adaptive and Reflective Middleware, Bangalore, India.

[32] J. Tan, H. Feng, X. Meng, and L. Zhang, "Heavy-traffic analysis of cloud provisioning," presented at the Proceedings of the 24th International Teletraffic Congress, Krakow, Poland.

[33] C. Chapman, W. Emmerich, F. G. Marquez, S. Clayman, and A. Galis, "Software architecture definition for on-demand cloud provisioning," presented at the Proceedings

of the 19th ACM International Symposium on High Performance Distributed Computing, Chicago, Illinois.

[34] J. Rao, X. Bu, K. Wang, and C.-Z. Xu, "Self-adaptive provisioning of virtualized resources in cloud computing," *SIGMETRICS Perform. Eval. Rev.,* vol. 39, pp. 321-322.

[35] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. v. d. Merwe, "CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines," presented at the Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Newport Beach, California, USA.

[36] E. Tasoulas, H. Haugerund, and K. Begnum, "Bayllocator: a proactive system to predict server utilization and dynamically allocate memory resources using Bayesian networks and ballooning," presented at the Proceedings of the 26th international conference on Large Installation System Administration: strategies, tools, and techniques, San Diego, CA, 2012.

[37] M. Stokely, A. Mehrabian, C. Albrecht, F. Labelle, and A. Merchant, "Projecting disk usage based on historical trends in a cloud environment," presented at the Proceedings of the 3rd workshop on Scientific Cloud Computing Date, Delft, The Netherlands.

[38] O. Corp. (2012, 12/4/2012). *Oracle Technology Network Developer License Terms*. Available: http://www.oracle.com/technetwork/licenses/standard-license-152015.html

[39] (2012, December 4, 2012). *OpenEMR*. Available: http://www.open-emr.org/

[40] A. Foundation. (2013, 1/6/2013). *Apache Jmeter*. Available: http://jmeter.apache.org/

[41] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience," ed.

[42] M. N. Rodrigo Calheiros, Cesar De Rose, Rajkumar Buyya, "EMUSIM: An Integrated Emulation and Simulation Environment for Modeling, Evaluation, and Validation of Performance of Cloud Computing Applications," *Software -- Practices and Experiences,* vol. 00, 2012.

[43] A. Nunez, J. L. Vazquez-Poletti, A. C. Caminero, J. Carretero, and I. M. Llorente, "Design of a new cloud computing simulation platform," presented at the Proceedings of the 2011 international conference on Computational science and its applications - Volume Part III, Santander, Spain.

[44] E. J. Topol, *The creative destruction of medicine : how the digital revolution will create better health care*. New York: Basic Books.

[45] R. Rauscher, "Cloud Computing Considerations for Biomedical Applications," in *Healthcare Informatics, Imaging and Systems Biology (HISB), 2012 IEEE Second International Conference on*, 2012, pp. 142-142.

[46] IETF, "RFC 2790: Host Resources MIB," vol. 2790, ed: Internet Engineering Taskforce, 2000.

[47] J. Wang, K.-L. Wright, and K. Gopalan, "XenLoop: a transparent high performance inter-vm network loopback," presented at the Proceedings of the 17th international symposium on High performance distributed computing, Boston, MA, USA, 2008.

[48] "American Recovery and Reinvestment Act of 2009," in *H. R. 1*, ed. United States of America, 2009.

[49] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," Tallinn, Estonia, pp. 129-148.

[50] R. Rauscher and R. Acharya, "Performance of Private Clouds in Health Care Organizations," presented at the CloudCom 2013, Bristol, UK, 2013.

[51] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," Los Alamitos, CA, USA, 2012, pp. 137-42.

[52] A. Ankit, J. Lakshmi, and S. K. Nandy, "Virtual Machine Placement Optimization Supporting Performance SLAs," in *CloudCom 2013*, Bristol, UK, 2013.

[53] J. Conn. (2014, 5/10/2014). *Record HIPAA settlement could portend tougher privacy enforcement*. Available: www.modernhealthcare.com/article/20140509/BLOG/305099995/record-hipaa-settlement-could-portend-tougher-privacy-enforcement

[54] J. McLean, "Reasoning about security models," Washington, DC, USA, 1987, pp. 123-31.

[55] "45 CFR Part 162: HIPAA Administration Simplification: Standard Unique Health Identifier for Health Care Providers; Final Rule," D. o. H. a. H. Services, Ed., ed. Washington, DC, 2005.

[56] J. W. Brady, "Securing Health Care: Assessing Factors That Affect HIPAA Security Compliance in Academic Medical Centers," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 2011, pp. 1-10.

[57] (2014, 5/10/2014). *Family Educational Rights and Privacy Act (FERPA)*. Available: http://www.ed.gov/policy/gen/guid/fpco/ferpa/index.html

[58] C. Blackwell, "The management of online credit card data using the Payment Card Industry Data Security Standard," in *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*, 2008, pp. 838-843.

[59] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science,* vol. 339, pp. 321-4, Jan 18 2013.

[60] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Supplement to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Frame Extensions for Virtual Bridged Local Area Network (VLAN) Tagging on 802.3 Networks," *IEEE Std 802.3ac-1998,* p. i, 1998.

[61] V. Rajaravivarma, "Virtual local area network technology and applications," in *System Theory, 1997., Proceedings of the Twenty-Ninth Southeastern Symposium on*, 1997, pp. 49-52.

[62] S. A. J. Alabady, "Design and implementation of a network security model using static VLAN and AAA server," Damascus, Syria, 2008.

[63] F. Weihong and L. Aixia, "VLAN Technology Application Research based on Campus Network Security," *Applied Mechanics and Materials,* vol. 220-223, pp. 2945-8.

[64] L. Jiajia and L. Wuwen, "Security analysis of VLAN-based Virtual Desktop Infrastructure," Piscataway, NJ, USA, pp. 301-4.

[65] V. Gainer, K. Hackett, M. Mendis, R. Kuttan, W. Pan, L. C. Phillips*, et al.*, "Using the i2b2 hive for clinical discovery: an example," *AMIA Annu Symp Proc,* p. 959, 2007.

[66] M. A. Siller and J. Woods, "QoE in multimedia services transmission," Orlando, FL, USA, 2003, pp. 74-6.

[67] A. Perkis, S. Munkeby, and O. I. Hillestad, "A model for measuring Quality of Experience," in *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, 2006, pp. 198-201.

[68] P. Casas, M. Seufert, S. Egger, and R. Schatz, "Quality of experience in remote virtual desktop services," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, 2013, pp. 1352-1357.

[69] J. F. Kurose and K. W. Ross, *Computer Networking : a Top-down Approach*, 4th ed. Boston: Pearson/Addison Wesley, 2008.

[70] R. Rauscher and R. Acharya, "A network security architecture to reduce the risk of data leakage for health care organizations," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 231-236.

[71] U. T. Hadley Malcolm, "Target CEO out as data breach fallout goes on," in *USA TODAY (Arlington, VA)*, ed, p. ARC.

[72] (2014, 11/30/2014). *Data breach results in $4.8 million HIPAA settlements*. Available: http://www.hhs.gov/news/press/2014pres/05/20140507b.html

[73] Y. Demchenko, C. de Laat, and A. Mavrin, "Defining generic architecture for cloud IaaS provisioning model," Setubal, Portugal, pp. 79-85.

[74] Y. Demchenko, C. Ngo, C. de Laat, J. Rodriguez, L. M. Contreras, J. A. Garcia-Espin*, et al.*, "Intercloud Architecture Framework for Heterogeneous Cloud Based Infrastructure Services Provisioning On-Demand," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pp. 777-784.

[75] W. Dargie, "Estimation of the cost of VM migration," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pp. 1-8.

[76] L. Wei-Chu, L. Chien-Hui, K. Kuan-Tsen, and C. H. P. Wen, "Flow-and-VM Migration for Optimizing Throughput and Energy in SDN-Based Cloud Datacenter," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, pp. 206-211.

[77] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pp. 671-678.

[78] N. R. Katsipoulakis, K. Tsakalozos, and A. Delis, "Adaptive Live VM Migration in Share-Nothing IaaS-Clouds with LiveFS," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, pp. 293-298.

[79] D. Bell and L. LaPadula, "Secure Computer systems: Mathematical Foundations," M. Corporation, Ed., ed, 1973.

[80] W. Stallings, *Computer Security : Principles and Practice*. Upper Saddle River, NJ: Prentice Hall, 2008.

VITA

Richard L. H. Rauscher

1115 Stone Creek Dr
Hummelstown, PA 17036
rauscher@psu.edu

EDUCATION/DEGREES AWARDED

University of South Florida, Tampa, Florida
M.S. Degree Computer Science, 12/1996
Improving the Performance of Time-Constrained Audio/Video over Ethernet

Rutgers University, New Brunswick, New Jersey
B.A. Degree Computer Science, 05/1991

PROFESSIONAL/TEACHING EXPERIENCE

Director, Research Informatics, Penn State Hershey, Hershey, PA 2005-2010, 2011-2015
Executive Director of IT, Boston University, Boston, MA 2010-2011
Chief Information Officer, Karmanos Cancer Institute, Detroit, MI 2003-2005
Manager of Technology Architecture, Moffitt Cancer Center, Tampa, FL 1995-2003
Assistant in Engineering Computing, University of South Florida, Tampa, FL 1993-1995
Senior Instructor, Penn State Capital College, Harrisburg, PA 2006-2007
Instructor, Wayne State University, Detroit, MI 2004-2005
Instructor, University of South Florida, Tampa, FL 1995, 1997

PUBLICATIONS

Rauscher, R., Acharya, R., 'A Network Security Architecture to Reduce the Risk of Data Leakage for Health Care Providers', Proceedings of the IEEE Healthcom 2014, Natal, Brazil, October 2014.

Rauscher, R., Acharya, R., 'Virtual Machine Placement in Predictable Computing Clouds', Proceedings of the 7th Annual Conference on Cloud Computing, Anchorage, Alaska, July 2014.

Rauscher, R., Acharya, R., 'Performance of Private Clouds in Health Care Organizations', Proceedings of the IEEE CloudCom 2013, Bristol, United Kingdom, December 2013.

Rauscher R., Proceedings of the 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology, 'Cloud Computing Considerations for Biomedical Applications.', 2012.

Rauscher R, Acharya, R. Proceedings of the First Annual Combined AMA and IEEE Meeting. 'A Protocol for Long-Term Preservation of Trust in Electronic Health Records Constructed from Heterogeneous Source.' 2010.

Sundstrom J, Sundstrom C, Sundstrom S, Fort P, Rauscher R, Gardner T, Antonetti D. 'Phosphorylation Site Mapping of Endogenous Proteins: a Combined MS and Bioinformatics Approach", *J Proteome Research*, 2009 Feb; 8(2):798-807