Ensuing Security in a Proposed Tertiary Institution Cloud Computing Environment: Introducing a NoHype Framework to the Private Cloud as a Way of Securing the IaaS Model

Shaquille Martey
Nanjing University of Science and technology
200 Xiaolingwei, Nanjing
Jiangsu, China

Gongxuan Zhang
Nanjing University of Science and technology
200 Xiaolingwei, Nanjing
Jiangsu, China

ABSTRACT

Cloud computing has reduced the large capital outlays for hardware storage and the human expense needed for its operation. As more institutions and organizations read and write (upload and download private data remotely on computer networks through an Internet connection, eliminating the need for local computer storage, data encryption from unauthorized access (intrusive hackers) becomes a priority. Tertiary institutions need for storage is a wide one, so the need for storing of data in the cloud arises simplifying data management and easing the workflow and pipeline of the staff, non-staff and students. The paper explores various research efforts, reviewed to introduce cloud to Tertiary institutions and employable methods for addressing the biggest fear of cloud adoption in educational organizations. This paper focuses on the security of data stored on the private cloud environment for tertiary institutions, proposing a security architecture for its infrastructure to employ for the satisfaction of such needs. For the purpose of this paper, the security of the private cloud is discussed with more focus on the Virtualization Infrastructure as a Service. This adopted security Architecture is integrated into the Proposed Tertiary Institution Cloud Computing Environment and ready for testing.

General Terms

IAAS, Cloud Security, Tertiary Institution Cloud Computing Environment (TICCE).

Keywords

Cloud Computing, IaaS Hypervisor, Architecture Framework, Virtualization

1. INTRODUCTION

According to the International Data Corporation (IDC), vendor revenue from sales of IT infrastructure products (server, enterprise storage, and Ethernet switch) for cloud environments, including public and private cloud, grew 47.2% year over in the third quarter nourishing some of 2018 (3Q18), reaching $16.8 billion today's internet- based services and software's with high revenue excesses [1]. Cloud computing has been noticed by end users as a low- cost technology trend that offers many efficient on-demand services, such as storage, hardware, and software. [2]. With cloud computing users can upload and download personal files with special read and write permissions remotely, end users can also use the processing speed of some remote hardware to facilitate, render a project or perform a task with other services that cloud computing has to offer. They can do without investing in infrastructure or paying for the cost of new software licensing.

Mell et al described Cloud computing as a pay-per-use model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction [8]. Cloud computing is often times confused with some important elements involved in its origination. Elements like utility computing, grid computing, Autonomic computing, platform visualization, etc. Cloud computing can usually incorporate some of these elements but is not synonymous with the list. One of the most important applications of Cloud Computing is the educational cloud. In the new globalized economy, educational institutions (e.g., universities) must provide high quality IT learning infrastructure and prepare students for the challenges of the 21st Century, with minimum budgets [19].

The cloud model is composed of three delivery models alongside four deployment models and five key characteristics. Fig. 1 below lists the components of the cloud model [8]. The Private deployment model is a model with an architecture designed solely for a specific organization and the infrastructure, services and software maintained over a non-public network. This reduces the security risks that cloud computing is faced with since only the members (staff, non-staff and students) of the organization (tertiary institution) are granted permission to the cloud services and resources.

Cloud computing when employed in Tertiary Institutions (TI) provides benefits like database management and access to research database from anywhere round the clock. This reduces the complexity being faced in management of large data since most tertiary institutions deal with a huge population eliminating the need for large computers and hard copy of files. Results, teaching staff, students management, exam record to name a few, are some of the services being rendered to tertiary institutions.

2. BACKGROUND AND RELATED WORK

A multi-stage methodology that utilizes case studies, internet articles, books and journals relating to the subject was employed. In relation to cloud services, "a survey of IT and library leads in UK education, carried out by Jisc, found that the most popular use of cloud technology currently relates to student email systems"[14]. Many leading universities have accepted the use of cloud computing to automate and share resources amongst students and staff including the e-learning plat- form run for some online universities. Here is a compiled list of some of the services:

- Email systems (E.g. Microsoft office365)
- Payroll/Fee Payment Portals
- Admin. Portal
- Journal/Research Library
- E-Learning
- Academic Portal
- Database Management

Despite the positive feedbacks Universities running a paid or open-source private cloud environment still face a few limitations. The following table list the problems and solutions identified by Sarvesh et al in their 2012 International Journal.

Table 1. Private cloud limitations.

Problems	Solutions
Low utilization of Servers	Virtualization, Multitenancy
Power, Space, and Constant Constraints	Utility billing and resource elasticity
Delays in launching new services	On-demand availability, Self-service
High overhead in provisioning services and users	Self-service
Unclear value contribution of center IT	Utility billing
Internal fracture	Internet delivery

2.1 Proposed Architecture For the TICCE Cloud Model

IT departments in Tertiary Institutions have to be running constant updates to meet up with the fast-changing cloud industry. Sarvesh et al identified both Deploying applications and delivering web-based student services at a rapidly accelerating rate and secondly, by drastically reducing CapEx and OpEx costs while maintaining the highest levels of security and privacy [16] as a way of catching up with the trend and updates. The Tertiary Institution Cloud Computing Environment (TICCE) model which is deployed on a private cloud, is delivered to the students, staff, and faculty using both the Infrastructure as a Service model (IaaS) in conjunction with the Software as a service (SaaS) model. The combination was adopted to produce an Architecture with a great level of scalability that can rapidly respond to demand. The end users (staff, students and faculty) of the TI, request for a service and after verification, the user is granted access to the private cloud. A filter runs on the private cloud which dedicates the request to either the IaaS or SaaS models depending on the resources and services needed for the execution of such request. Examples of IaaS and SaaS already existing today are listed in table 2.

Table 2. Tested IaaS and SaaS adoption for TICCE adoption.

Infrastructure as a service (IaaS)	Software as a Service (SaaS)
Education ERP	Education ERP
Campus Consortium/ Campus EAI	Campus Consortium/ Campus EAI
Rackspace	Google Apps
Amazon EC2	Microsoft office 365
Ensratuis	Jaspersoft

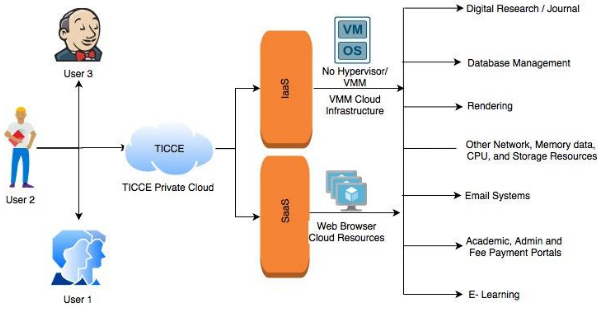

Fig. 1. A basic prototype of the architecture of a typical TICCE model.

IT departments in Tertiary Institutions have to be running constant updates to meet up with the fast-changing cloud industry. Sarvesh et al identified both Deploying applications and delivering web-based student services at a rapidly accelerating rate and secondly, by drastically reducing CapEx and OpEx costs while maintaining the highest levels of security and privacy [18] as a way of catching up with the trend and updates. The Tertiary Institution Cloud Computing Environment (TICCE) model which is deployed on a private cloud, is delivered to the students, staff and faculty using both the Infrastructure as a Service model (IaaS) in conjunction with the Software as a service (SaaS) model. The combination was adopted to produce an Architecture with a great level of scalability that can rapidly respond to demand. The end users (staff, students and faculty) of the TI, request for a service and after verification, the user is granted access to the private cloud. A filter runs on the private cloud which dedicates the request to either the IaaS or SaaS models depending on the resources and services needed for the execution of such request.

2.2 Virtualization

Here, the cloud vendor simply grants the TI Cloud administrator full read and write (INPUT/OUTPUT) permission to install and manage it's own operating systems and application systems for staff, students and faculty members providing full hardware platform and/or data centers (Backend Management). Virtualization, the intelligence from the network and a robust ecosystem as described by [21] offers the basis for obtaining operational efficiency, security, activity continuance, scalability, interoperability leading in the end to innovation [20]. Virtualization provides multi tenancy and scalability, and these are two significant characteristics of Cloud Computing as stated by [27]. The TICCE runs a virtualization software called the Virtual Machine Monitor (VMM). It comprises of a host machine and a guest machine.

The host machine runs one or many virtual machines (VM) where each VM is the guest machine.

2.3 Scalability

A TI having so many departments and groups observing holidays and accepting new students, [24] stated that in a cloud-based environment, security policies and framework must give room for scalability and future expansion. This makes scalability on - demand a critical component based on two key characteristics: multi-tenancy, where multiple tenants share the same service instance, and elasticity, where tenants can scale the amount of their allocated resources based on current demands [26].Scalability in the TICCE can be achieved since an exact number of enrolled students, staff and faculty members is recorded in the school database. Scalability limitations can be solved in various ways one of them being through various software approaches already existing.

2.4 Hypervisor

Hypervisor is an intermediate software layer running between the physical computer and the virtual machine operating system. It allows multiple virtual machine operating systems or applications to share the same set of basic physical hardware, so it can also be regarded as the "meta" operating system in the virtual environment. The hypervisor can operate directly on bare metal, called "bare metal architecture". As an operating system, it uses and manages the underlying hardware resources and provides the resource call interface to the virtual machine running on the upper layer. The representative products of this kind include VMware ESX server, Citrix XenServer and Microsoft Hyper-V, as well as open-source KVM under Linux. Also, as an application, which is called "host architecture". It uses the device drivers and underlying services provided by the host operating system to manage the memory, process scheduling and resource management of the virtual machine.

2.5 Integrity, Confidentiality and Availability Of Data

Security of data stored on the cloud must follow three critical concepts [8] which are data: Integrity, confidentiality and availability. Cloud service providers (CSP) scan and correct the user data keeping the data integrity intact giving users permission to access data stored on the cloud without any changes or corruption. Data confidentiality as explained by [22], is the prevention of intentional or unintentional unauthorized disclosure of information. Data confidentiality from untrusted servers/requests is ensured through disclosing encrypted data keys only to authorized servers/requests. Availability ensures the reliable and timely access to cloud data or cloud computing resources by the appropriate personnel [22]

2.6 Related Work

Security in the cloud is a popular topic in the IT world due to the fast-growing cloud services and its wide adoption. John R. Vacca proposed that "*Private cloud security has the advantage of fully controlling host equipment, firewall defensive systems, CPU and memory resource allocation, Web server management, direct database instance management, and many other aspects normally associated with on-prem data center facilities"[11]*. Before this can be achieved, End-point security management (EPSM)processes must be in place. Security points management is important in private clouds because cloud vulnerabilities carry risks that should be taken into consideration and can be done remotely through a private cloud service provider over the Internet. One of the advantages of EPSM is its ability to detect new and suspicious end user devices on connecting to a cloud.

John McCarthy also called the father of artificial intelligence predicted in 1961 MIT centennial celebration that “computing may someday be organized as a public entity just as the telephone system is a public utility” [3]. New technology initially creates fear and to a lot of end users, this consternation is about security since private and confidential data is shared with a decentralized remote cloud storage provider. Questions arise on whether or not to trust this cloud computing provider with complete access to personal data, do we trust this company with hope that it won’t manipulate, steal, sell or misuse their data knowing fully well it’s safer within their local firewall? The private cloud deployment model answers this question by providing cloud services within a tertiary institutions firewall.

Two surveys were carried out by the IDC in 2008 and 2009 analyzing the issues with the cloud computing model. Security was rated by 70% of the users as the major challenge faced in cloud computing, while performance and availability were rated second and third by more than 60% of the users.[9] (see figure 2a and 2b) on various security models used in the cloud environment and conference proceedings. Strength and flaws of the security models in relation to the private cloud was also researched on. Before diving into the security models, a research on the threats posed on cloud security was conducted.

3. PROPOSED NOHYPE SECURITY FRAMEWORK FOR THE TICCE MODEL

In this section, the NoHype framework is introduced into the TICCE or use in securing the Infrastructure. The next step may consist of the daily processing of the internal operations, addressing at the same time the components of public and private cloud in order to assure the security and protection policies [20]. Security controls must reply in accordance with environmental variables following data and workloads during upload and download, either as intrinsic workload segments like encryption and/or via a CMS (cloud management system). When done right, the possibility of corruption or loss of data in a cloud environment is eliminated.

The Proposed TICCE implements a robust security procedure designed by identifying the type of data, functions, applications and important procedures within the TI. The cloud Management Software (CMS) is the interface that students, staff and the faculty utilizes for the management, termination and requesting for VMs running on dedicated servers. The CMS security is assumed to be secure for the purpose of this research and the TI cloud users are obliged to protect software running inside their VMs.

3.1 IaaS Security Framework

The TICCE setup runs an Infrastructure security Architecture which eliminates the attack surface caused by the Hypervisor. The Hypervisor is the key component of virtualization sharing or emulating the resources among VMs while monitoring their activities for security and operational concerns [29]. The NoHYPE system proposed by [28] aims to protect the hypervisor against attacks which is usually from the guest VMs by eliminating the need for interaction between VMs and hypervisor. A malicious guest VM causes a VM shut down to occur allowing the injection of malicious code or triggering a bug. This occurrence leads to the potential violation of the confidentiality, integrity, or both of the guest and host VMs. This can also cause a DDoS attack violating the data availability policy by crashing or slowing down the hypervisor. The security of the guest Operating System (OS) is left for the cloud provider who can make available a set of slightly modified guest OS kernels needed for booting up a VM. Virtualization for a server in a generic cloud computing environment is illustrated in Figure 5 below.

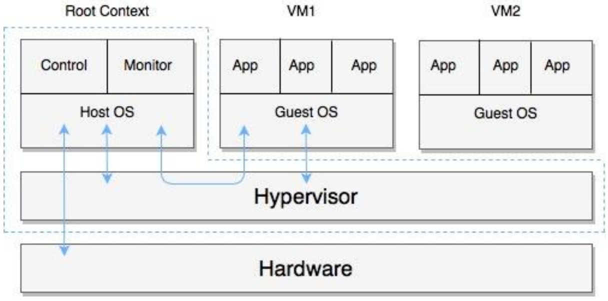

Fig. 2: Generic Virtualization of a single Server [33].

In the diagram above, points of interaction and components are highlighted, where the dotted lines shade the components that are trusted. Arrows signify interactions between the guest OS and hypervisor, host OS and hypervisor, guest OS and the host OS (via the hypervisor), and the host OS and the I/O devices. However, it is important to note that the host OS wields special administrative privileges like launching and shutting down VMs, leading to the direct interaction of the host OS with the hypervisor via hypercalls [31]. Virtualization layers are removed by the NoHype system while still retaining multi-tenant1 server settings, has some roles to play.

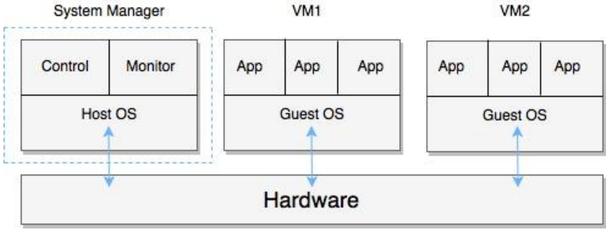

Fig. 3: A typical server with the hypervisor removed [33].

The direct interaction between VMs and cloud virtualization management software is eliminated in the server as illustrated in figure 3. The main point becomes that each individual guest VM should run directly on the hardware without the need for a hypervisor.

3.2 Key Ideas of the Nohype System

According to [28], the NoHype system embodies four key ideas:

i) Pre-allocation of processor cores and memory resources - This eliminates the need for the hypervisor to manage cloud resources dynamically by pre-allocating the processor cores and memory. The key to achieve the isolation of each VM is to ensure that each VM is restricted from accessing the physical memory of other VMs but granted access to it's own guest physical memory. The hardware paging mechanisms available in modern processors is utilized to enforce the memory isolation without an active hypervisor.

ii)

iii) Use of virtualized I/O devices - Input and Output (I/O) devices allows the computer to interact with the outside world by moving data into and out of a system [30]. I/O devices is dedicated to the guest VMs to eliminate the use of virtualization software to emulate devices. Dedicating physical I/O devices to each VM is not scalable and so, the NoHype virtualizes these devices. The NoHype takes advantage of the modern processors in assigning devices directly and virtualization extensions in modern commodity devices. The devices are controlled by the VMs through memory mapped I/O.

iv) Minor modifications to the guest OS to perform all system discovery during bootup - By slightly modifying the guest OS cache system configuration data for later use, the no hype architecture allows the normal boot up procedure of the guest OS, bringing changes to the guest OS to a minimal. A temporary hypervisor acts as support in order to overcome current limitations posed by commodity hardware. The infrastructure provider provides the modified OS kernel which is a requirement in the NoHype Architecture. This is to ensure that the end-user code doesn't attempt an attack on the temporary hypervisor by b3locking the execution of code in the presence of Virtualization software. It is important to know that the system does not restrict what applications and guest OS kernel modules the end user can run. The temporary hypervisor is disabled after the bootup sequence and the VM execution code is switched from the cloud provider to the end users code which allows the end-user to load any OS Kernel module desired or run any application.

v) IV. avoiding indirection by bringing the guest virtual machine in more direct contact with the underlying hardware: Indirections that map the virtual views to real hardware which is a requirement for hypervisors. The NoHype system avoids this indirection eliminating the need for the hyper visor to carry it out. A case of such indirection can be seen in the exchange between cores, where hypervisors create the illusion of running a dedicated system to each VM. Here, the hypervisor provides each VM with a unique processor ID starting from 0. The NoHype system in the process of dedicating cores to VMs, gives the guest VMs access to the real processor ID avoiding indirection.

vi) Furthermore, indirection is used to deliver interrupts to the correct VM and the hypervisor handles the interrupts routing them to the correct VM. The rerouting is entirely eliminated while dedicating cores to VMs because all interrupts are forwarded directly to the target.VM in the NoHype system.

4. IAAS NoHYPE FRAMEWORK METHODOLOGY

The proposed NoHype addresses all the major functions/roles of the virtualization layer listed below:

4.1.1 Scheduling Virtual Machines.

4.1.2 Memory Management.

4.1.3 Emulating I/O devices and arbitrating access to them.

4.1.4 Network packet processing (Switching, NAT and Access control.)

4.1.5 Starting/Stopping/Migrating Virtual Machines.

NoHype system Architecture is illustrated in figure 7. For brevity, the following paragraphs summarizes the NoHype Architecture on figure 7. Here, each core is allowed to run a single VM, eliminating any potential software cache-based side channel existing when an LI cache is being shared [32]. Because the TICCE cloud infrastructure is dynamic and the VMs needed by the university can be scaled on demand, idleness is handled by the tertiary institutions cloud administrator shutting down idle VMs instead of over subscribing.

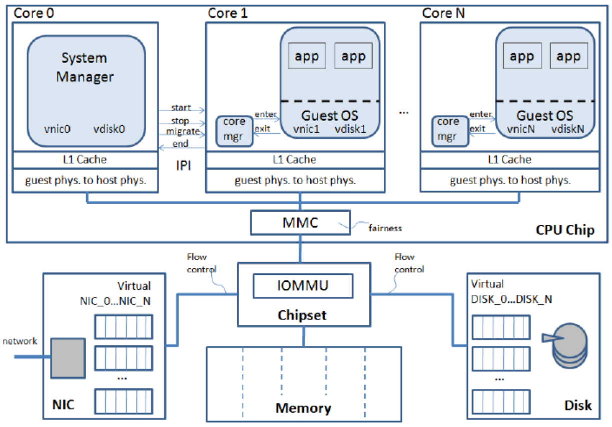

Fig. 4. Adopted NoHype Architecture

Partitioning of physical memory s proposed in the Architecture achieved by giving each guest OS a view of memory with the OS having a dedicated and guaranteed fraction f physical memory on a host system. Hardware support in the processor undertakes the mapping between the guest and host physical memory addresses restricting memory operations to the assigned ranges. This responsibility is the obligation of the multi-core memory controller (MMC) as can be seen in figure 7 and the hardware page table mechanisms with built-in support for carrying out the re-mappings.

Access to I/O devices in the physical system needs to be partitioned and each guest OS is assigned a physical device individually and direct access given to it. A virtual device can have more than one queue dedicated to it and each VM interacts with only the virtual device(s) its assigned to. This creates the interface that is detected by the linked VM.

Each device is mapped to a different range in memory for read /writes to/from the device initiated by the cores, giving permission only to memory ranges, enabling the direct interaction between the guest OS and it's assigned devices. Rate-limited access to every single I/O bus is attained through a flow-control mechanism which allows the I/O device control the rate of transmission solving the complication of the bandwidth sharing of I/O bus (e.g., PCIe) being limited.

Ethernet switches in the data center network are meant to perform the switching and security functions for networking and not a software switch in the virtualization layer. This gives VMs direct access to the network interfaces. Some of the benefits of these are:

1.1.1. Simplifying management by removing extra type of switch and layer in a switch hierarchy.
1.1.2. Freeing processors on the server
1.1.3. Permission to use all the feature s of the Ethernet switch.

Eliminating the need for the software switch is achieved by fusing support into hardware Ethernet switches enabling capabilities like allowing packet forwarding out of the same port as it was received.

When starting a VM, the CSP receives a command from the TICCE cloud administrator. The instructions to TICCE cloud administrator are issued by the staff/students/faculty specifying how many VMs and the OS (Linux/Windows in this case). The CSP then maps the memory and disk of the to be assigned into its space with the description of the VM and the location of the disk image supplied by the TICCE administrator. The CSP zeroes out the memory of the local disk after downloading and storing the disk image allowing the TICCE administrator to access the resources an initialize them. This procedure brings the guest OS image into the VM. The CSP then unmaps the memory and disk after initialization is complete so not to have access again, ensuing security. A '*start*' Inter-processor Interrupt (IPI) is issued by the CSP to the core. The core manager which is a code is executed to initialize the memory and I/O mapping and performs a VM exit to the guest OS. This starts the guest OS execution from the image now stored locally on the disk.

A guest OS being used by a staff/student/faculty exits when a '*stop*' command is issued by the CSP when being notified by the TICCE admin to stop a specific guest OS. The CSP directs a "stop" IPI code to the core running the VM which needs to be stopped. The core manager then saves the disk image of the VM depending on the SLA while clearing its disk and memory and also un-map the I/O and memory devices. It finally puts the current core to sleep mode and notifies the TICCE admin of completion.

When a guest OS performs an illegal operation, which could be trying to access memory not assigned to it, the core manager either sends an '*end*' IPI command to the CSP to inform the CSP of an abnormal exit or clears the disk and memory. This prevents data leaks and puts the core into an idle state while waiting for a '*start'* IPI command while the CSP ends a notification to the TICCE cloud admin of the VMs aborted status change.

For a LIVE migration operation to be executed, the TICCE admin, instructs the CSP on the source server to migrate a specific VM to a given target server. The CSP then sends a '*migrate'* IPI to the core being run by the VM. The core manager embodies an interrupt handler which stops the execution of the VM, securing the entire state of the VM by hashing and encrypting it and capturing its entire state. The CSP then sends this state to the target server, and sends an IPI to the core manager in return, which then checks the hash and decrypts the state restarting the VM and continuing execution. NoHype dedicates tracking page modification to the memory management unit, enabling the CSP to send IPIs periodically to obtain differences only, eliminating the hypervisor which was initially involved in this process.

5. FINAL NoHYPE FRAMEWORK

The NoHype architecture focuses on solving the concerns and limitations that should hinder the cloud adoption in a TICCE, by creating an architecture where the students, staff and faculty are giving an improved security on the virtualized infrastructure in addition to the protection against malware and physical security offered by the CSP. Attaining a comparable level of security which is the main aim of this project demands that the three critical concepts in ensuing cloud security must be observed by the NoHype architecture. Below is a summary of how the three the critical concepts of

Data Integrity, Data confidentiality and Data Availability is achieved in the TICCE framework.

5.1 Data Integrity and Confidentiality

Memory access violations are mitigated in the NoHype security architecture because of the following:

- Cores are not shared.
- Absence of hypervisors.

The only possible way a VM could gain access to the physical memory outside of its originally assigned range would the alteration of the mapping specifications table. Due to this mitigation, data confidentiality and data integrity is ensured since the CSP and TICCE admin is assumed to be trusted with changing the tables specifying the mapping of *guest physical addresses* to *host physical addresses* only. The CSP interacts only with the TICCE admin and the core managers and is completely isolated from the guest VMs.

5.2 Data Availability

Because no VM should have permission to affect the availability of another VM, any Infrastructure with VMs running Hypervisor based Architectures can be attacked in one of three ways [33]:

- An alteration in how the hypervisor scheduled the VMs.
- Core Interruption while the core is running a VM

By the attacker executing infinite I/O reads/writes or amounts of memory for an excessive amount of the *bus* which affects the VM performance of other end users.

The NoHype stops this first attack by disabling the hypervisor from making scheduling decisions and dedicating a core to a VM. The second is eliminated by the device interrupts and hardware masking of inter processors. While additions to the multi-core memory controller for providing fairness and through chipset to rate-limit access to I/O, the third attack is eliminated [34].

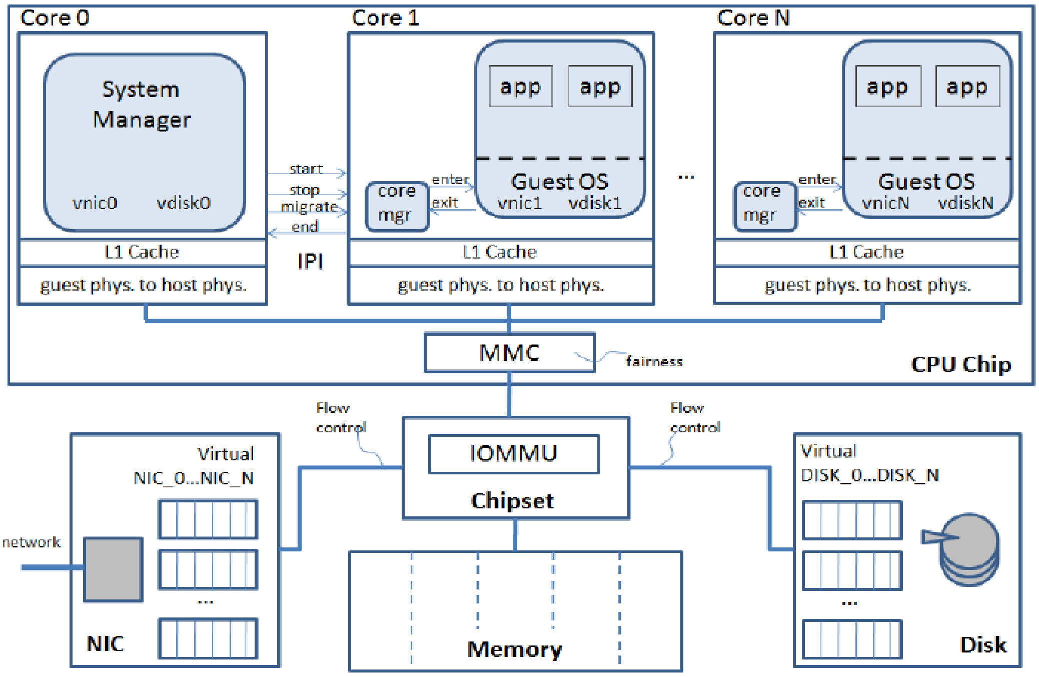

Figure 5: Final NoHype Architecture Framework

6. CONCLUSION

The spectrum of IaaS CSPs is quite wide, giving the TI an access to a higher level of technology solutions and its unique dynamic infrastructure scalability gives the TICCE the option to tailor the requirements of the TI at a coarse-grained level or fine-grained level. It also addresses the three critical security policies for cloud environments and shows how the NoHype enforces data integrity, availability and Confidentiality. Not to confuse Data privacy with data confidentiality, the latter refers to the ethical duty of the CSP on handling the data shared and agreeing not to disclose the data in question to any third party by law. The threat model is focused on the TI due to a limited number of students and staff is catered to and a limited number of services provided. The NoHype becomes a good fit for securing the TICCE. This is entirely an *open* design which to be tested by a myriad of cloud security professionals during the testing involving a number of activities, where each activity is based upon a formal methodology or standard that adds unique value to the overall security test.

This paper proposed an Architecture running on a Private Cloud for TIs and how to secure the infrastructure Service by removing the hypervisor. It introduces the idea of private clouds to the tertiary institution and its security More research has to be carried out on securing the SaaS since it is also a very popular model, for use amongst educational organizations.

7. ACKNOWLEDGMENTS

I will like to thank my supervisor Prof. Gongxuan Zhang for his assistance in this work.

8. REFERENCES

[1] Worldwide Public Cloud Services Spending Forecast to Reach $160 Billion This Year, According to IDC. (2018, September 27). Retrieved 28 May 2019.

[2] Aloraini, Afnan & Hammoudeh, Mohammad. (2017). A Survey on Data Confidentiality and Privacy in Cloud Computing. 1-7. 10.1145/3102304.3102314.

[3] Goertzel, Karen & Booz, Ciacssp & Hamilton, Allen. (2008). Enhancing the Development Life Cycle to Produce Secure Software. Retrieved 28 May 2019 from https://www.researchgate.net/publication/228704603_Enhancing_the_Development_Life_Cycle_to_Produce_Secure_Software

[4] McCarthy, J., "Centennial Keynote Address," MIT, 1961.

[5] Popovic, Kresimir & Hocenski, Zeljko. (2010). Cloud computing security issues and challenges. 344 - 349.

[6] Larry Dignan. (2019). Top cloud providers 2019: AWS, Microsoft Azure, Google Cloud; IBM makes hybrid move; Salesforce dominates SaaS. Retrieved 24th May 2019 from https://www.zdnet.com/ article/top-cloud-providers-2019-aws-microsoft-azure-google-cloud- ibm-makes-hybrid-movesalesforce-dominates-saas/

[7] Red Hat Inc. (n.d.). What is different about cloud security. Retrieved from https://www.redhat.com/en/topics/security/cloud-security

[8] Peter Mell & Tim Grance. Effectively and Securely Using the Cloud Computing Paradigm

[9] R. Balasubramanian & M. Aramudhan. (2012). Security Issues: Public vs Private vs Hybrid Cloud Computing. International Journal of Computer Applications (0975 – 8887). Volume 55– No.13.

[10] Securing the Internet of Things: A Proposed Framework (N.D).

[11] John R. Vacca. (2017). Computer and Information Security Handbook. 931 - 936.

[12] Upguard. 2018. What Are Cloud Leaks? Retrieved 28 May 2019.

[13] Taylor, R. W., Fritsch, E. J., & Liederbach, J. (2015). Digital crime and digital terrorism. (3rd ed.). Upper Saddle River, NJ:Pearson

[14] Craig Nelson & Tomer Teller. 2016. Cloud Attacks Illustrated:Insights from the cloud provider. RSA Conference 2016.

[15] B. Joshi, A. Vijayan, and B. Joshi, "Securing cloud computing environment against DDoS attacks," in Computer Communication and Informatics (ICCCI), 2012 International Conference on, Jan. 2012, pp. 1 –5

[16] QS. 2019. Cloud Technology in Higher Education. Retrieved 28 May 2019.

[17] SARVESH KUMAR, SAURABH SRIVASTAVA, VIJAY KUMAR, OPINDER KUMAR & ASHWANEE KUMAR SINGH. 2012. Private Cloud: A Paradigm of Cloud Computing with University Shared data Center (USDC)International Journal of Computer & Communication Technology ISSN (PRINT): 0975 - 7449, Volume-3, Issue-6, 7, 8, 2012121

[18] Anwar, Md & Masud, Anwar & Yong, Jianming & Huang, Xiaodi. (2012). Cloud Computing for Higher Education: A Roadmap. 10.1109/CSCWD.2012.6221872.

[19] Alzoubaidi, A. (2016). Private Database Cloud Deployment for Multi-Campus Universities.

[20] Mircea, Marinela & Andreescu, Anca. (2010). Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis. Communications of the IBIMA. 2011. 10.5171/2011.875547.

[21] Bozzelli, T. (2009). "Will the Public Sector Cloud Deliver Value? Powering the Cloud Infrastructure," CISCO. [Online], [Retrieved May 25,2019],http://www.cisco.com/web/strategy/ docsgov/2009_cloud_public_sector_tbozelli.

[22] Ronald L. Krutz & Russell Dean Vines. (2010). CLOUD SECURITY: A Comprehensive Guide to Secure Cloud Computing.

[23] Azeez, Nureni & van der Vyver, Charles. (2018). Security and privacy issues in e-health cloud-based system: A comprehensive content analysis. Egyptian Informatics Journal. 10.1016/ j.eij.2018.12.001.

[24] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. of CCS'06, 2006.

[25] Almorsy, Mohamed & Grundy, John & Ibrahim, Amani. (2012). TOSSMA: a tenant-oriented SaaS security management architecture. 10.1109/CLOUD.2012.146.

[26] Dawoud, Wesam & Takouna, Ibrahim & Meinel, C. (2010). Infrastructure as a service security: Challenges and solutions. 1 - 8.

[27] Szefer, Jakub & Keller, Eric & Lee, Ruby & Rexford, Jennifer. (2011). Eliminating the hypervisor attack surface for a more secure cloud. 401-412. 10.1145/2046707.2046754.

[28] Wensheng, Zhang & Cico, Betim & Meşecan, İbrahim. (2018). Survey on Hypervisor Security: Challenges and Solutions.

[29] http://www.cs.ucc.ie/~gavin/cs1001/Notes/chap01/ ch01_7.html

[30] Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2. http://ww.intel.com/ products/processor/manuals/.

[31] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," in 34th International Symposium on Computer Architecture (ISCA), pp. 494 – 505, 2007.

[32] Keller, Eric & Szefer, Jakub & Rexford, Jennifer & Lee, Ruby. (2010). NoHype: virtualized cloud infrastructure without the virtualization. ACM SIGARCH Computer Architecture News. 38. 350-361. 10.1145/1815961.1816010.

[33] D. J. Bernstein, "Cache-timing attacks on AES," in *University of Illinois at Chicago Tech Report*, 2005.