Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Why Businesses Still Remain Sceptical About Virtualization

Emmanuel C. Ogu[1], Oyerinde O.D.[2], Ogbonna A.C.[3], Michael I. Ogu[4], Omotunde A.A.[5]
[1, 3, 5] Department of Computer Science and Information Technology, School of Computing and Engineering Sciences, Babcock University, Ilishan Remo, Ogun State, Nigeria
[2]Department of Computer Science, Faculty of Natural Sciences, University of Jos, Jos, Plateau State, Nigeria
[4]Department of Political Science and Public Administration, Babcock Business School, Babcock University, Ilishan Remo, Ogun State, Nigeria

Abstract –Virtualization, the technology that allows multiple guests (clients) to reside on a single host (provider) machine and share the resources of the host machine, is rapidly gaining prominence in the corporate business world of the 21st century. Virtualization is a powerful technology for increasing the efficiency of computing services. However, besides its advantages, its flexibility also raises a number of questions regarding security, especially as it forms the bedrock of the revolutionary technology of cloud computing. Research has shown that Cloud Computing directly has the potential to tremendously impact positively the profit margins of leading 21st century businesses but the lack of satisfactory answers to some of these questions, especially by cloud service providers, has greatly limited the rate of adoption of full virtualization by many governments and organizations. This paper aims to elucidate and classify the questions and threats to virtualization, and suggest possible solutions to them.

Keywords –Virtualization, Multi-tenant Systems, Tenant Architectures, Cloud computing, Service Provider.

I. INTRODUCTION

It wasnot until the 1970s before mainframe users saw the first implementation of virtualization and symmetric multiprocessing. Prior to this time, the cost of computers and computing services were very high. Coupled with only sparsely intermittent computational needs, organizations, researchers and academicians, who were the first to welcome computing equipment and services. They found it difficult owning computers because they couldnot provide justification for investing at high costs in a computer that would sit idle for most of the time. To assuage this situation entrepreneurs came up with the idea of "renting" time; thus making it possible for organizations and users to either own or subscribe to computing resources at much lower costs [1]. It therefore became possible for users to find access to large scale mainframe computer systems using thin clients/ terminal machines which were often referred to as "static terminals / machines" because they were used for input and output communications only and had no internal processing capabilities. This technology made the use of expensive mainframe systems more efficient because it allowed multiple users to share both the physical access to the computer from multiple terminals as well as to share the processing and computing resources (CPU time, disk time and space, etc.). This practice was able to curb periods of "no activity" on the mainframes and allowed for greater returns on investment for companies that practiced such around the 1950s [2]. It is this technology that grew through various transformations and related nomenclature over the following time periods: Remote Job Entry in the 1950s [3], Shared and Dedicated Web Hosting (which are forms of Virtual Web Hosting) around 1995 to 1997 [4], [5]; Virtual Private Server (VPS) Hosting around 1998 [6], Grid/Utility Computing [7], [1]to become Cloud Computing (CC) about three decades later.

According to [8], Cloud Computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction; having characteristics of on-demand self-service, broad network access, resource pooling, rapid elasticity and payment per usage of various business models."

Virtualization is the technology that allows multiple virtual machines (also called guest machines) to run on a single physical machine (host machine) and share the resources of the physical machine [9]. This therefore makes it possible for a single physical server to host many guest virtual machines (VMs), operating systems, and applications without the additional cost and complexity that result from running multiple physical machines [10].This development became one of the forces that was to revolutionize information technology and cloud computing in the coming years [11].

Cloud computing services are delivered through three standardized service models: the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and the Software as a Service (SaaS) Models.

The Software as a Service (SaaS) model is the first and topmost model in which clients/consumers are provided access to the applications of a provider which are deployed on a cloud infrastructure. These applications are made accessible to various client devices either through a thin client interface, such as a web browser (e.g., web-based email), or through a program interface. Consumers do not manage or control the underlying cloud infrastructure including network, servers,

operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

The second model is the Platform as a Service (PaaS) in which clients/consumers are allowed to deploy onto cloud infrastructure, consumer-created or acquired applications that are created using programming languages, libraries, services and tools supported by a provider. Consumers do not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but reserve control over deployed applications and possibly the configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS) is the third and most foundational model. In this model, consumers are provided with processing, storage, networks, and other fundamental computing resources, which enable the consumer is able to deploy and run arbitrary system and/or application software. Consumers do not control the underlying cloud infrastructure (basically hardware), but reserve control over operating systems, storage, and deployed application; possibly with limited control of some networking components (such as host firewalls).

These service models directlydefine the three layers comprised in the core of most modern cloud computing infrastructure. Each of these layers provide varying services to a tailored consumer market segment while also subscribing to services offered by the supporting layer underneath it (except the IaaS layer) [12]. This is shown in figure 1:

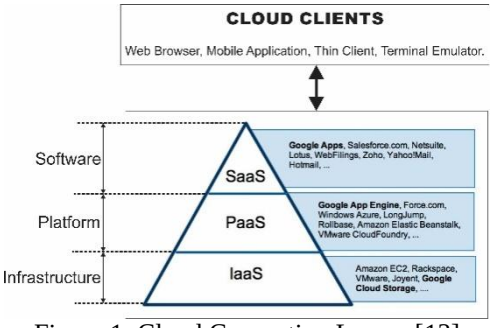

Figure 1: Cloud Computing Layers [13]

The first / foundational layer of Cloud Computing is the IaaS layer. The products here relate to hardware and associated services such as: general processing, servers, storage devices, database management, and all other hardware related services that are offered as a service to the end user. The next layer is the PaaS layer upon which developers can build and test applications that run on the IaaS, either for the IaaS layer itself or for the SaaS layer above it. The topmost layer is the SaaS, and this deals exclusively with applications for end users [12].

Virtualization is very important to cloud computing. As a matter of fact, it provides the abstraction from hardware state that cloud computing enjoys by taking a physical resource such as a server (computer) and dividing it into virtual resources called virtual machines (multiple computers that can be released to subscribers) to which users (businesses) could subscribe. "This abstraction from the hardware state allows not only multiple operating systems to coexist on the same hardware, but for one VMM to run on multiple different networked physical systems concurrently. By utilizing a VMM to mediate between the OS and the hardware, virtualization changes the one-to-one mapping of OSs to hardware to many-to-many" [14].

II. VIRTUALIZATION: THE TECHNOLOGY

Virtualization is basically a technology that makes it possible for an operating system to run within another operating system as an application [15]. It is a technology that provides for an environment that is equivalent to the operating environment of the host operating system, in which other operating systems can run as guests separated from the physical hardware [14]. Although the technology of virtualization is not an entirely new paradigm, it has begun to gain rapid prominence in recent times because most modern system architectures now come with built-in computer functionalities that make it possible for other operating systems to run on top of one major (host) operating system.

Virtualization could be commonly server-based [11] or system-based [14] depending on the level of abstraction from which it is viewed. Usually, in server-based virtualization, a physical resource (such as a server) is used to host other operating systems that are provisioned and delivered over a network environment, while in system-based virtualization, a host operating system within a particular computer is able to provision its resources to other operating systems that are able to run within it as guests that are decoupled from the hardware state. Other types of virtualization are network virtualization [16], storage virtualization [17], and desktop virtualization [18]. In this study, however, the term "virtualization"would be used to refer to the broad, generic, underlying concepts that power the technology of virtualization and not necessarily any of the specific types.

It should be noted that although various computer organizations and architectures implement virtualization using different implementation arrangements, certain generic components basically make virtualization possible. These include the:

Guest Operating System (OS),
Virtual Hardware,
Virtual Machine Monitor (VMM) or Hypervisor,
Host Hardware System,

Host Operating System,
VMM Drivers, and
Virtual Machine (VM) Applications [14].
These components are organized into a generic structure that is as illustrated figure 2:

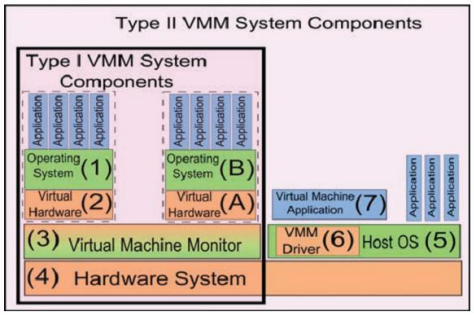

Figure 2: Core Components in Virtualized Architectures[14]

In virtualized environments, "operating systems operate on the hardware as privileged software, and are generally able to perform any operation the hardware supports, whereas programs running inside an operating system are less privileged, and generally cannot perform operations except those that the operating system permits. These privilege levels are often called *rings*, with the lower numbered rings (i.e., *ring 0* or *dom 0*) having higher privileges than those with higher designations" [19]. "Operating system kernels generally run in the lowest ring, and thus have control over everything running in the lower privilege (higher numbered) rings" [14].
Current virtualization solutions can be classified into three main categories. These three categories fall within two main classes: the *hypervisor-based virtualization* and the *non-hypervisor-based virtualization* (also known as operating system or container-based virtualization).
In the hypervisor-based virtualization, virtualization is made possible through a **virtual machine monitor** (VMM), otherwise known as a h**ypervisor,** which would be discussed shortly, while in the non-hypervisor-based categories, virtualization is made possible via operating system containers. The three main categories of virtualization solutions are:
Paravirtualization (PVM). This was one of the first adopted models of virtualization and is still widely used today. This virtualization solution needs no special hardware to achieve virtualization but instead relies on special kernels and drivers. This kernel(s) sends privileged system calls and hardware access requests directly to a hypervisor or virtual machine monitor [VMM]. The VMM then decides how to handle the request. The use of special kernels and drivers limits the window of flexibility in terms of choosing what operating systems to run. PVM must hence use an operating system (OS) that can be modified to work with the hypervisor. This virtualization solution reduces the overhead needed to virtualize privileged operating system calls since no special hardware is needed to intercept them. Examples of paravirtualized solutions include Xen and User Mode Linux. [9]
Hardware Virtual Machine (HVM). This solution stands at the lowest level of virtualization. HVM requires special hardware capabilities to be able to interpret privileged system calls from guest machines. It makes the full virtualization of a machine possible without the need for any special operating systems or drivers on the guest system. Guest machines actually interact through an emulated interface that appears as though they are communicating directly with the host hardware. Most modern processors have HVM capabilities which are often called *virtualization extensions*. These extensions detect when a guest machine tries to make a privileged system call like sending data on a Network Interface Card (NIC). This call is intercepted by the hardware and passed on to the hypervisor to decide how it should be handled. Great flexibility in terms of what OS to run is present. However, HVMs are known to have the highest overheads of all virtualization solutions [20], and usually donot come as the first choice in most situations. Examples of HVM virtualization solutions include the VMware Server, KVM, and Virtual-Box. [9]
Container Virtualization (CV). This non-hypervisor-based solution, also known as OS-level virtualization, creates multiple secure containers on a single operating system which runs different applications. It is based on the fact that a server administrator may want to isolate different applications for security or performance reasons, while maintaining the same OS across each container. It allows a user to share a single OS kernel between multiple containers and have them securely use computer resources with very minimal interference between containers. It is shown to have the lowest overhead of all virtualization solutions [20], but completely lacks flexibility in terms of which OS to run. Examples of CV solutions include OpenVZ, Linux-VServer and Solaris Zones. [9]
Table 1 gives a brief comparative summary of the categories of virtualization solutions:

Table 1: Comparative Summary of the categories of Virtualization Solutions [Source: authors]

S/N	Virtualization Solution	Performance Overhead	OS Flexibility	Performance Isolation	Hypervisor-Based
1	Paravirtualization (PVM)	Moderate	Minimal	High	Yes
2	Hardware Virtual Machine (HVM)	High	High	Least	
3	Container Virtualization (CV)	Low	None	Highest	No

The **hypervisor** or **virtual machine monitor** (VMM) is a piece of software with superior privileges that runs beside or beneath the operating system,and must be designed in such a manner that reflects an "*efficient, isolated duplicate of the real (or physical) machine*". It is a control program that functions as a *dispatcher* (for assigning tasks to other resident control modules), an *allocator* (for taking final decisions on the allocation of system resources) and an *interpreter* (for interpreting instructions and system calls for all resident guests) [21], [14]. In essence, the VMM is responsible for all tasks relating to the interfacing of guests with host hardware. The exact responsibilities (resource allocation and management, virtual disk space management, processing of request and system calls, etc.) that are borne by operating systems in standalone systems are the same responsibilities borne by the virtual machine monitor / hypervisor in multi-tenant architecture (or systems), where a single server is able to host several virtual machines (subscribers) [21].

There are basically two types of hypervisors. These types are distinguished relative to their position with respect to the host operating system and the host hardware. These types are as follows:

The Classical Type-I Hypervisor. This is also known as the bare-metal hypervisor. This type of hypervisor comes installed as the default boot system on the hardware and as such runs at the most superior privilege level (*ring 0* or *dom 0*). They have full control over all virtual machines that run on the hardware [22], [14]. The position of a Type-I hypervisor in a typical virtualized environment is illustrated in figure 3.

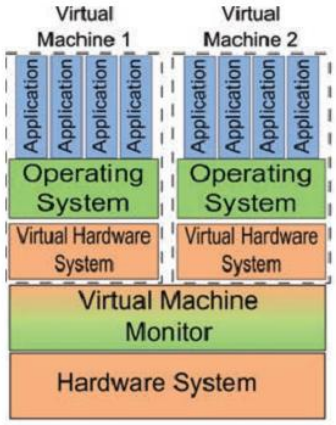

Figure 3: Generalized Architecture for Type-I Virtual Machine Monitors[22]

The Type-II Hypervisor. This is also known as the hosted hypervisor. This type of hypervisor runs alongside or above a host operating system which sits on an underlying hardware, possibly utilizing drivers of the host operating system when handling input / output (I/O) calls. This cooperation results in a VMM system that has not need of hardware-unique drivers for I/O operations, thus allowingVMs to cohabit the environment of an existing operating system. As a result, restrictions on the types of VM become minimal because there is no need to migrate or overwrite the existing OS to a multiple boot system before virtualization can occur [14], [22]. The position of a Type-II hypervisor in a typical virtualized environment is illustrated in figure 4.

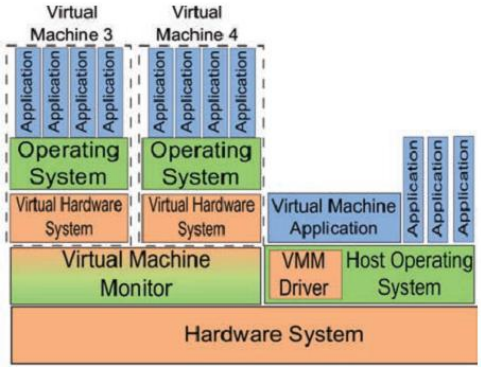

Figure 4: Generalized Architecture for Type-II Virtual Machine Monitors[22]

Table 2 shows a comparison of some other features of the Types I and II Hypervisors.

Table 2: Comparison of the Type 1 and Type 2 Hypervisors [23]

SI.No.	Feature	Type 1	Type 2
1.	**Definition**	Hypervisors run directly on the system hardware.	Hypervisors run on a host operating system.
2.	**Support**	Hardware virtualization.	Operating system virtualization.
3.	**Examples**	VMware ESXi and Citrix XEN Server.	KVM, Virtual Box, VMware Server and Microsoft Virtual PC.
4.	**Efficiency,**	Comparatively better than Type 2.	Though inferior, it is used mainly on

	Availability and Security		systems where support for a broad range of I/O devices is important.
5.	**Performance**	Very high. Resources are not being consumed by a bloated parent operating system.	Steep resource-overhead penalties reduce performance.
6.	**Ease of use**	Fairly easy to install but complicated to configure.	Easy to install, use and maintain.
7.	**High availability**	Yes	No
8.	**Reliability**	Yes	Moderate
9.	**Virtualization hypervisor management**	More options for management and automation. Centralized consoles to manage large number of hosts and VMs.	Fewer options for management and automation as well as limited VMs can be managed.
10.	**Cost**	Very Costly	Moderate
11.	**Scalability**	Very high (easily run hundreds of VMs on a single host).	Very limited scalability (in the size of the VMs and the number of VMs that can run on a single host).
12.	**Resource control**	It offers the least amount of resource overhead and advanced resource controls that allow you to guarantee, prioritize and limit VM resource usage.	It has no or limited resource controls, so VMs have to fight each other for resources.
13.	**Size OR Complexity**	Smaller	Bigger and more complex.

[21], proposed formal requirements for full virtualization of Virtualizable third generation architectures which have formed the foundation and yardsticks against which all hypervisors and hypervisor-based solutions have been measured overtime. Many other succeeding researches have either agreed with or slightly improved on the propositions of these requirements: [24], [25], [26], [23], [14] and [27]. Three properties for virtualized architectures include sameness, equality, and control.[21]

In sameness, "the VMM provides an environment for programs which is essentially identical with the original machine" (i.e. innocuous instructions must be run directly on the processor with interventions only where necessary).

In the case of equality, "programs run in this environment show at worst only minor decreases in speed"

Control denotes the situation where "the VMM is in complete control of system resources"

Two types (or properties) of virtualized instructions are also usually specified namely privilege level and sensitivity. [21]

Privilege level addresses the question of "does this instruction require a process to be highly privileged to call it directly? If the CPU traps and switches control to supervisory software (running in low rings) when the instruction is called from a process running in user mode (high rings) then the instruction is *privileged*, as it requires privilege to be executed [else, it is *non-privileged*]." [14]

Sensitivity on its part tries to answer the question of whether an instruction has the capacity to interfere with something the VMM should have complete control of."A *sensitive* instruction has the capacity to interfere with VMM operation, whereas an *innocuous* one does not possess such capacity. A simple example is that reading VMM program memory will not interfere with VMM behaviour, but writing to it could." [14], [21].

III. RATIONALE FOR VIRTUALIZATION

Despite the relatively cheap cost of commodity hardware and availability of networks, virtualization still has benefits that put it in the fore for most organizations. "Physical systems are associated with other costs which could be operational, physical and/or technical, all in addition to the initial purchasing cost. In addition, every physical machine must contend with requirements of physical space, cabling, energy, cooling, and software administration" [14]. These requirements inadvertently leave virtualization as a preferred alternative.

Furthermore, "modern commodity operating systems such Windows and Linux are very complex, usually comprising of tens of millions of lines of code (LOC) in the latest desktop versions. This situation results in a much larger vulnerability surface than can be easily or provably secured according to [25] and [24]". The fewer lines of codes in virtual machine monitors thus provide a preferred alternative in the above regard.

Similarly, it remains a fact that operating systems add "a single point of failure for everything (data, processes and information) that runs on them. The difficulty in securing this single, complicated point of failure poses a security risk for the system's data and processes, hence accounting for the choice of virtualization" [14].

Virtualization makes it possible to test new computer and IT aided business solutions in an environment that is less vulnerable to external vices. Using methods of sandboxing and container operating systems, new, untrusted and potentially vulnerable IT business solutions can actually be tested in a more secure environment where failures and breaches of the new systems or solutions would not hamper the current smooth running of existing deployed solutions.

Legacy business solutions that are already in use and properly functioning can easily be retained and run alongside new solutions deployed on virtualized platforms until these new solutions have been confirmed and smooth migration / transition has been achieved.

An added advantage is that the simulation of networks and independent business units in order to study and observe the cooperation and interactions between them is made easier by virtualization. In the same vein, independent business units can be simulated to run various independent processes that may either be interrelated or not, and the interactions between these can be observed more critically within a closed environment.

Virtualization further makes the previously dreaded tasks of migration / transition, backup and recovery of business systems a "walk-in-the-park", and the administration and management of these tasks can be more centrally done [28].

The implication of removing the dependency of operating systems on a system's physical state through system virtualization, according to [14], is that it allows multiple operating systems to be installed on a VMM, and thus multiple operating system VMs (called guest operating systems) can be installed on each physical system thereby allowing multiple VMs on the same hardware with its many advantages. Near-complete isolation between guest operating systems on the same hardware protects against OSs being a single point of failure; further allowing OS consolidation from different machines as is necessary to reduce system underutilization and maintain efficiency of operation." [14]. The abstraction from the hardware state consequently "allows not only multiple operating systems to coexist on the same hardware, but for one VMM to run on multiple different networked physical systems concurrently. By utilizing a VMM to mediate between the OS and the hardware, virtualization changes the one-to-one mapping of OSs to hardware to many-to-many" as reported by [14].

However, it must also be noted that the extent of virtualization could also impose extra performance overheads to computing infrastructure as already pointed out in tables 1 and 2.

IV. VIRTUALIZATION: BUSINESS AND INDUSTRY PERSPECTIVE

Although virtualization and cloud computing provides ICT executives with high hopes of lasting solutions to business problems and needs, therebyenhancing business growth and performance indices through advancements in virtual collaboration, green IT, enterprise mobility, business intelligence, amongst others [29], it has become necessaryto attempt to deal with the many apprehensionsslowing down full virtualization of IT infrastructure. These apprehensions, classified as straight-forward questions, would ensure that the journey to finding solutions may involvea more direct, precise and decisive approach to dealing with these issues.

The essential concept behind virtualization is the concept of multi-tenancy. As is the case in other areas of human endeavour, getting two different sets or groups (businesses or enterprises in this case) of people to co-habit the same habitat (be it economic, environmental or technological) generates some concerns which are further aggravated in the case of virtualization by advancements in information and communication technology.

It needs to be noted at this juncture businesses, organizations and governments have various reasons hindering full virtualization of their IT infrastructure, as reported in [11]. A careful examination, however, reveals that the reasons are rooted in pertinent concerns to which cloud service providers usually do not provide satisfactory answers when they are posed as the questions which will be dealt with subsequently.

V. DRIVERS OF VIRTUALIZATION

The influence of technology on government processes and procedures cannot be overemphasized. In turn, the procedures and processes of government – policy (formation and implementation) and infrastructure – can also influence the efficiency of several technological innovations within the state. Virtualization has been identified as one of the most promising technologies for the enhancement of business performances, leading to sustainability of the business, as well as productivity of both the individual employee and the organization as a whole. It is also important to discover if the environment within which virtualization occurs, the action and inactions – policies – of either the business organization or government and other efforts at training and development, have any direct or indirect implications for the efficacy of virtualization as a technological procedure.

Policy Implications: Policies are, more or less, the major instruments for running businesses or governments. The actions and inactions of government impact greatly on the level of progress experienced by the state, likewise those of business executives influence business outcomes to a large extent. [30], [31], among other authors, has asserted the link between public policy and technology. Policy formulation and implementation must be supportive of technological advancement and use, if such technologies will be effectual within the territories they are intended. Policies formulation without properly implementation, or vice versa, the merits of technology may not be fully accomplished.

Infrastructural Implications: Also as important as policy formulation and implementation is the provision of necessary and adequate infrastructure to ensure the deployment and use of such technologies for either business or government purposes. Several states in Africa face some very peculiar problems of infrastructure, which has made it difficult for the continent to compete favourably, globally. In addition to electricity, educational institutions are also very vital infrastructures that are necessary for ensuring that maximum potentials of technology are exploited to the benefit of businesses and governments alike.

Training and Development: Training and development is another major challenge of technology in many African states. In most cases, technologies exist, even at the very minimum levels, but they are not maximized owing to an inadequate knowledge of operating such technologies. Virtualization in businesses may not provide optimum benefits if employees are not properly trained on the use of such technologies and the benefits that maximum utility could provide for these business.

VI. SECURITY CONCERNS IN VIRTUALIZATION

The concerns that lie very foundational to why most businesses, organizations and governments remain apprehensive of full virtualization of their IT infrastructure can be categorised into six parameters namely: co-tenancy, sharing, trust and privacy, hacks and breaches, boundary enforcement and restrictions, and integration and compatibility.

CO-TENANCY is the concern that is related to the type and calibre of individuals who own and operate the other virtual machines that share the same platform as well as the applications that run on them – who the co-tenants are. Usually, competitors within the same business endeavour or industry are not keen on getting on the same virtual platforms with each other for fears of breaches in their trade secrets by the competitors who are on the same platform. In reality, virtualization involves a lot of sharing. There are shared folders, hardware and resources, and these are the primary target during introspective attacks. Also, successful subversion and escalation of privileges could give competitors snapshots or entire clones of virtual machines with the content in them [32].

SHARING expresses the concern related to the amount / extent (type and depth) of information that virtual machines share with co-tenants and with the host. Considering that virtualization involves a lot of sharing of folders, hardware and resources, clients would want to know what kind of information from their virtual machines can be seen in the shared folder as well as the extent / depth of information. This concern is pertinent because spoofing the data contained in the shared folders or the virtual drive tables could offer very useful information about virtual machines and their content [14].

TRUST& PRIVACY (provider's level of interference with private virtual machines).

Trust issues express a concern over the extent of manipulation or interference that virtual service or cloud providers could have with virtual machines running on their infrastructure. Clients want to be convinced that service providers cannot directly access and manipulate their virtual machines as well as their content. This is primarily due to the fear that service providers may be able to sell unencrypted clones / snapshots of the content of their virtual machines to competitors.

Privacy concerns are most often used synonymously and interchangeably with trust. Privacy concerns however are more on the part of the clients, denoting the extent to which clients are able to keep the activities, data and operations of their virtual machines away from the prying eyes of cloud service providers / administrators. [33], pointed out the fact that some activities of Cloud Service Providers (CSPs) may, in effect, go a long way to further complicate data privacy issues due to the extent of virtualization and cloud storage that are used to implement cloud services.

HACKS / BREACHES (security of virtual machines from intrusion and invasion)

Hacks / breachesconstitute a concern over the level of security that is present to protect virtual machines and their operations running on a provider host from external sources (people who are neither providers nor co-tenants). This concern remains one of the top and most recurrent identified by various researchers and primarily arising from denial of service (DoS), distributed denial of service (DDoS),as well as infiltration attacks. An instance of this concern occurredin September 2014, when the iCloud server (a cloud storage facility owned and operated by Apple Inc.) experienced a major hack / breach that led to the leakage of private photographs belonging to top, popular Hollywood celebrities by the hackers, thereby putting the computing giant in bad light[34]. Also, a recent survey by the International Data Corporation (IDC) revealed that 87.5% of people surveyed at various levels (from IT executives to CEOs) opine that security remains the top-most challenge in every cloud service infrastructure [35] and [23] further pointed out security as the highest risk issue in virtualization and cloud computing with some IT executives holding the view that not all applications or data should be virtualized and/or put in the cloud.

BOUNDARY ENFORCEMENT / RESTRICTION. Every virtual machine residing on a host is entitled to a virtual space which is confined within virtual boundaries. Boundary Enforcement / Restriction expresses concern over the extent to which tenants' boundaries are carefully marked off and how much they are able to breach or push their virtual boundaries / limits as are clearly and technically mapped out in service level (SL) agreements. Clients would also raise concerns relating to what degree other tenants would be able to push or extend their virtual spaces and boundaries and begin to breach the space of other guests residing on the host. SL breaches are becoming a major concern to cloud providers as more research and technologies have gone into the task of trying to enforce this to the uttermost. The boundary issue becomes a bigger one when it is expanded in relation to property rights, contracts, and tenancy laws which in reality have industry and geo-political intricacies.

INTEGRATION & COMPATIBILITY. Integration is the concern about the ability of the host environment to receive the virtual guest (client) along with its already existing service and application requirements and programs. Compatibility is the concern over how able the host environment can cope with, and satisfy client requirements and program needs throughout the period of the tenancy (service level) agreement. Research by [11] revealed that some organizations are uncertain as to which virtualized applications, servers and environments would be able to satisfy completely the needs that were assuaged by their current IT framework which other raised the issue of compatibility with their already existing hardware resources.

Furthermore, and because virtualization is the core that powers cloud computing, it is no gainsaying that the security issues that bother on virtualization, to a large extent, also bother on cloud computing especially when viewed as a symbiotic relationship.

Some of the most extensive studies represented by such sources as [14], [23], [27], and [32], and highlighted in table 3, attempts asummary of the known common / major threats to virtualization, as well the solutions to these; grouped under the various concerns highlighted in the above section.

Table 3: Summary of Security Concerns, Known Threats and Solutions [Source: authors]

SECURITY CONCERNS	THREATS	SUGGESTED SOLUTIONS
CO-TENANCY	VM Introspection;Meta-data Copying; VM Information Leakage; Hijacking	Hypervisor Integrity and Hardening; Signature Verification; Monitoring, Auditing and Administration; Strong Passwords Enforcement for VMs; Encryption
TRUST & PRIVACY	VMM Subversion / Insertion;VM Cloning; Illegal VM Snapshots; Shared Folder Copying;Privilege Escalation; VM Information Leakage; VM Escape; Hijacking	VM Isolation; Hypervisor Integrity and Hardening; Shared Folder Encryption; Virtual Roles Separation
HACKS / BREACHES	VM Cloning; DoS and DDoS; VM Sniffing and Spoofing; Arbitrary Code Executions; Virtual Network Compromise; Hijacking	Hypervisor Security and Hardening; Monitors and filters on networks;Patches and Security Fixes; Intrusion Detection and Prevention Systems (IDSs and IPSs)
BOUNDARY ENFORCEMENT / RESTRICTION	VM Scaling; SLA Breaches;VM Masking;Malware corruption from other VMs and/or Host;VM Escape; VLAN Hopping; VM Promiscuity.	VM Isolation; SLA Enforcement; Resource Allocation; Virtual Boundary Separation; Anti-malware installations
INTEGRATION & COMPATIBILITY	Runtime Errors; Hardware and Software Incompatibilities; Corrupt Virtual Images	Sandboxing; Virtual Hardware and Software Verification; Disabling incompatible virtual hardware and software; Proper Backups and Regular Continuity Protection Measures;

VII. CONCLUSION

Virtualization has proven time and again to be one of the most promising technologies for business performance enhancement, sustainability and productivity. However, frequently recurring security concerns as well as the potential threats that are brought up due to the nature of flexibility which virtualization provides have kept begging for answers. A new course has been pointed out in this paper as clear questions posed by clients and subscribers regarding virtualization and cloud computing; cloud service providers are now better informed on how to directly address these concerns by tapping from the solutions suggested herein. It must however be mentioned at this point that the technology of virtualization has undergone rapid, commendable improvements in various forms in recent times to provide answers and solutions to some of the challenges that previously plagued future prospects of this technology; and in line with these, this research has gone on to suggest a couple more.

ACKNOWLEDGEMENT

This work was originally presented at the 1st International Conference on Advanced Computerized Systems and Emerging Technologies (ICACSET 2014). Babcock University Guest House, Ilishan-Remo, Ogun State. Nigeria. Conference attendance was sponsored by *Professor Olawale Jacob Omotosho*, a Chartered Engineer of the Federal Republic of Nigeria.

REFERENCES

[1] Davies, A. (2004, June). Computational intermediation and the evolution of computation as a commodity. *Applied Economics, 36*(11: 1131). doi:10.1080/0003684042000247334

[2] Christopher, S. (1959). Time Sharing in Large Fast Computers. *Proceedings of the International Conference on Information processing, UNESCO.2.19*, pp. 336-341. UNESCO. Retrieved February 1, 2014

[3] IBM. (1970). *IBM System/360 Operating System: Conversational Remote Job Entry Concepts and Facilities.* International Business Machine (IBM). North Carolina, USA.: IBM Systems Reference Library. Retrieved February 1, 2014, from http://bitsavers.informatik.uni-stuttgart.de/pdf/ibm/360/rje/GC30-2012-0_CRJE_Concepts_and_Facilities_Jun70.pdf

[4] Nikiforakis, N., Joosen, W., & Johns, M. (2011). Abusing Locality in Shared Web Hosting. *Proceedings of the Fourth European Workshop on System Security: Article No. 2.* Salzburg, Austria: Association for Computing Machinery. doi:10.1145/1972551.1972553

[5] Urgaonkar , B., Shenoy, P., & Roscoe, T. (2009, February). Resource overbooking and application profiling in a shared Internet hosting platform. *ACM Transactions on Internet Technology (TOIT): Article No. 1 , 9*(1).

[6] Bhattiprolu, S., Biederman, E. W., Hallyn, S., & Lezcano, D. (2008, July). Virtual Servers and Checkpoint/Restart in Mainstream. *ACM SIGOPS Operating Systems Review - Research and developments in the Linux kernel, 42*(5), 104-113. doi:10.1145/1400097.1400109

[7] Buyya, R., & Bubendorfer, K. (2009). *Market-Oriented Grid and Utility Computing*. Wiley Publishing.

[8] Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing.* Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, United States Department of Commerce. Gaithersburg, MD 20899-8930: National Institute of Standards and Technology. Retrieved January 28, 2014, from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[9] Ryan, S., & Jiangchuan, L. (2012). Understanding the Impact of Denial of Service Attacks on Virtual Machines. *Journal of the IEEE*.

[10] Tupakula, U., & Varadharajan, V. (2011). TVDSEC: Trusted Virtual Domain Security. *Institute of Electrical and Electronic Engineers (IEEE)*, 57-63.

[11] CDW Corporation. (2010, January 11). *CDW Server Virtualization Life Cycle Report (Medium and Large Businesses).* Retrieved from CDW Newsroom: http://webobjects.cdw.com/webobjects/media/pdf/Newsroom/CDW-Server-Virtualization-Life-Cycle-Report.pdf

[12] Strømmen-Bakhtiar, A., & Razavi, A. R. (2011). Cloud Computing Business Models. *Springer Computer Communications and Networks*, 43-60.

[13] Gartner AADI Summit. (2009). *Cloud Computing as Gartner Sees it.* Gartner's Application Architecture, Development & Integration Summit.

[14] Pearce, M., Zeadally, S., & Hunt, R. (2013, February). Virtualization: Issues, Security Threats, and Solutions. *Association for Computing Machinery (ACM) Computing Surveys*, Article 17: 1-39.

[15] Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). *Operating System Concepts (Ninth Edition).* U.S.A.: Wiley Publishers.

[16] Chowdhury, K. N., & Boutaba, R. (2009, July). Network virtualization: state of the art and research challenges. *IEEE Communications Magazine, 47*(7), 20-26.

[17] Soundararajan, V., & Anderson, J. M. (2010). The impact of management operations on the virtualized datacenter. *Proceedings of the 37th annual international symposium on Computer architecture* (pp. 326-337`). New York, NY, USA: ACM.

[18] Miller, K., & Pegah, M. (2007). Virtualization: virtually at the desktop. *Proceedings of the 35th annual ACM SIGUCCS fall conference* (pp. 255-260). New York, NY, USA: ACM.

[19] Bratus, S., Johnson, P. C., Ramaswamy, A., Smith, S. W., & Locasto, M. E. (2009). The cake is a lie: privilege rings as a policy resource. *Proceedings of the 1st ACM workshop on Virtual machine security* (pp. 33-38). Illinois, USA.: Association for Computing Machinery.

[20] Padala, P., Zhu, X., Wang, Z., Singhal, S., & Shin, K. G. (2007). *Performance Evaluation of Virtualization Technologies for Server Consolidation.* Hewlett Packard, Enterprise Systems and Software Laboratory. Palo Alto: Hewlett Packard. Retrieved February 2, 2014, from http://137.204.107.78/tirocinio/site/tirocini/Tirocinio-Zuluaga/Documents/virtualizzazione/Technologies%20for%20Server.pdf

[21] Popek, G. J., & Goldberg, R. P. (1974). Formal Requirements for Virtualizale Third Generation Architectures. *Proceedings of the fourth ACM symposium on Operating system principles*, 412-421.

[22] Sugerman, J., Venkitachalam, G., & Lim, B. -H. (2001). Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. *Proceedings of the General Track: 2002 USENIX Annual Technical Conference* (pp. 1-14). Berkeley, CA, USA.: USENIX Association.

[23] Nagaraju, K., & Sridaran, R. (2012, September). A Survey on Security Threats for Cloud Computing. *International Journal of Engineering Research & Technology (IJERT), Volume 1*(Issue 7), 1-10.

[24] Seshadri, A., Luk, M., Qu, N., & Perrig, A. (2007). SecVisor: A Tiny Hypervisor to provide lifetime Kernel code integrity for commodity OSes. *Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles* (pp. 335-350). Association for Computing Machinery.

[25] Franklin, J., Seshadri, A., Qu, N., Chaki, S., & Datta, A. (2008). *Attacking, Repairing, and Verifying SecVisor: A Retrospective on the Security of a Hypervisor.* Cylab Technical Report. CMU-Cylab-08-008.

[26] Hirano, M., Shinagawa, T., Eiraku, H., Hasegawa, S., Omote, K., Okuda, T., . . . Yamaguchi, S. (2009). A Two-step Execution Mechanism for Thin Secure Hypervisors. *Third International Conference on Emerging Security Information Systems and Technologies, IEEE.* (pp. 129-134). Institute of Electrical and Electronic Engineers.

[27] Pék, G., Butty´an, L., & Bencsáth, B. (2013, June). A survey of security issues in hardware virtualization. *ACM Computing Surveys (CSUR): Article No. 40, 45*(3). doi:10.1145/2480741.2480757

[28] Texas Reliability Entity. (2013). *Virtualization and Cloud Computing. "Security is a Process, not a Product".* Texas, USA.: Texas Reliability Entity.

[29] Kretschmer, T. (2010). *"WORK – Working Connected in Business and Society".* Bonn, Germany.: Deutsche Telekom AG.

[30] Griffith, R. (2007). Technology, Productivity and Public Policy. *Fiscal Studies, 28*(3), 273–291.

[31] Teece, D. J. (1986). *Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy.* Berkeley, CA 94720, U.S.A.: School of Business Administration, University ofCalifornia.

[32] Garfinkel, T., & Rosenblum, M. (2005). When virtual is harder than real: security challenges in virtual machine based computing environments. *Proceedings of the 10th conference on Hot Topics in Operating Systems* (pp. 20-20). Berkeley, CA, USA: USENIX Association.

[33] Winkler, V. J. (2011). *Securing the Cloud: Cloud Computer Security Techniques and Tactics.* Syngress Publishing.

[34] Duke, A. (2014, October 12). *CNN Entertainment (5 Things to know about the celebrity nude photo hacking scandal)*. Retrieved November 1, 2014, from CNN International: http://www.cnn.com/2014/09/02/showbiz/hacked-nude-photos-five-things/

[35] Lv, H., & Hu, Y. (2011). Analysis and Research about Cloud Computing Security Protect Policy. *Institute of Electrical and Electronic Engineers (IEEE)*, 214-216.

