Sindh Univ. Res. Jour. (Sci. Ser.) Vol. 48 (3) 579-584 (2016)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Application Service Delivery in the Modern Virtualized Data Center-Improving Reliability and Scalability

A. RAZA, M. Y. KOONDHAR++, SINDHU*, M. HYDER*, G. D. MENGHWAR*, B. BALOCH*, A. SHAH

Kulliyyah of Information Communication and Technology, International Islamic University Malaysia

Received 17thMarch 2015 and Revised 6thSeptember2015

Abstract: Application Container virtualization has recently been re-introduced as a potential solution for rapid deployment of applications for developers within the ever changing infrastructure landscape. Although, there appears to be a 'gap' of when the Application Container, Infrastructure Hypervisor or both technologies should be leveraged to provide for scalability, reliability while improving on overall deployment time. This study is an attempt to review the two technologies and understand their specific/shared use-cases in order to develop a framework to assist with the decision making model that Information Technology (IT) professionals within both the Application and Infrastructure disciplines can use to determine the path they should follow as it pertains to their selection of one or both of these evolutionary technologies. This study incorporates the three Information System theories, technology-organization-environment, diffusion of innovation and process-virtualization.

Keywords: Container, Hypervisor, Virtualization and Technology.

1. INTRODUCTION

The modern data center has evolved from large multiple floor buildings to 'rooms' with minimal footprint. This transition has been assisted by several technology advancements within infrastructure services such as the advancement of processors with multiple cores, the 'flash' based vice platter based hard drive and the leveraging of a hypervisor for virtualization. With all these advancements, not much was being done for applications rather the server counts within data centers continued to increase, sprawled, as more and more applications demanded specific software dependencies such as programming languages and application deployment frameworks. This increase in servers caused IT professionals to re-evaluate the usage of servers and determined that the average usage of an application server was only 10 – 25% of the available cores and 15 – 40% off available RAM. This information caused a re-evaluation of how to 'stack' applications on servers to reduce the number of servers in the data center and also reduce the capital and reoccurring maintenance expenses of those servers.

2. BACKGROUND

Throughout the Information Systems (IS) and Application lifecycles, new technology has contributed the most to the ever changing processes, procedures and enhancements IT. These technology advancements have caused for rapid programming adaption and critical system variations to meet the requirements or needs of the technology. The Data Center of today is no stranger to change. These 'rooms' have been modified from large warehouse spaces with air conditioning capacities 5–10 time an office building to smaller footprint rooms leverage virtualized components (Network, Servers and Storage).

Indeed, with the introduction of cloud computing, the notion that the hardware is 'hosted' outside of the 'local' IT control, has brought back some of the bulkier data centers, however, this change is more for the commoditization of infrastructure than any other reasons.

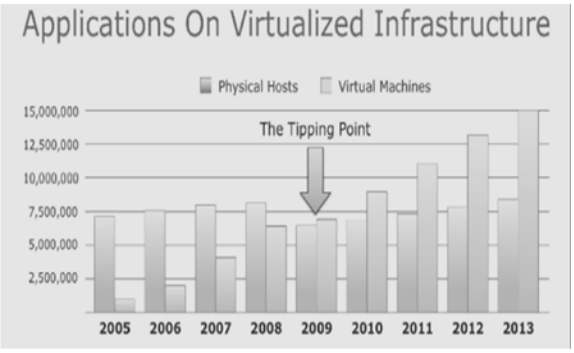

Fig.1. Applications on Virtualized Infrastructure

With virtual machine growth within the past 4 years at over 50% (Sharma, 2011), the usage of the hypervisor and leveraging of the single hardware server (Host) for multiple operating system servers (Guest) will continue to increase as Enterprises and others adopt either the Private, Public or Hybrid cloud computing model.

This growth in cloud computing also has a direct impact on the spending from IT shops around the world.

++ Corresponding Author: : yaqoobkoondhar@gmail.com
*Sindh Agriculture University, Tandojam

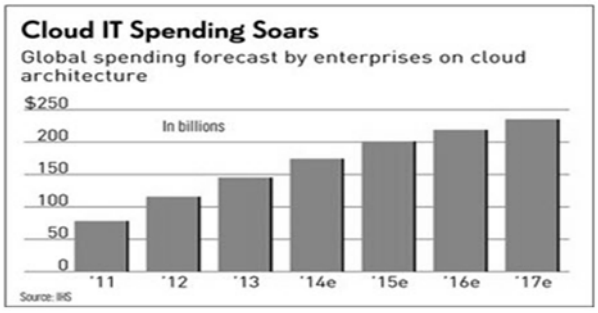

Fig.2 – Cloud IT Spending

With off premise hosting of either a Server (Infrastructure as a Service – IaaS), Application (Software as a Service – SaaS) or programming platform (Platform as a Service – PaaS), IT shops are spending less and less on capital – reducing overall IT cost and since the commoditization of cloud based services this has introduced even lower cost and higher benefit to IT shops.

3. HYPERVISOR VIRTUALIZATION

The focus of cloud computing virtualization technology has been focused on the hardware hypervisor abstraction which has been specifically for improvements within the infrastructure stack.

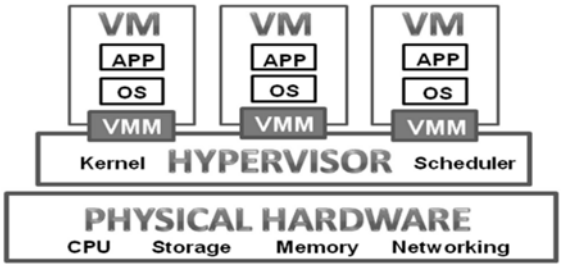

Fig. 3 – Hypervisor Stack

The benefit of this virtualization advancement is the leveraging bare metal systems to 'host' several operating system 'guest' thus allowing a consolidation from multiple physical to multiple virtual machines that reside on several physical servers and provide similar performance.

The reduced management overhead of the Hypervisor technology has also introduced several other benefits to overall IT investment and operations specifically Green IT and Development Operations (DevOPs).

Green IT is an operational model that reduces the electrical and other facility type needs of IT systems to reduce the overall cost and emissions to the environment from those systems. Green IT in concept is a means to quantify the usage of materials and resources that are typically not environmentally friendly in a way that shows reduction and conservation.

DevOps is a concept driven by the emergence of Application Developers who have enough knowledge and understanding of an infrastructure to manage and perhaps enhance it to their application needs. Typically, the infrastructure team will provide a provisioned machine with oversight, but the the Developer has the responsibility to manage and sustain the asset through its lifecycle.

The synergy between the Developer and the infrastructure elements grows exponentially with the virtualization platform. In public computing clouds such as Amazon and Azure, the Developer has the rights to "spin-up/spin-down" their virtual instance with no assistance from the infrastructure team. This means to allow for control of an asset with only oversight is appealing to all enterprise IT shops and this is shown tobe growing at an ecstatic rate (Eden,).

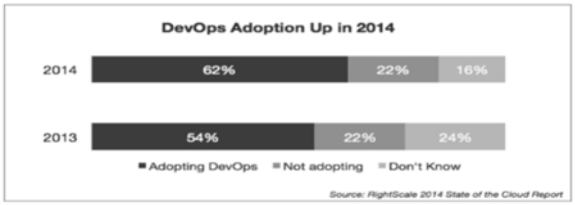

Fig. 4 – DevOps Adoption in 2014

As seen from the above graph, the adoption continues to grow and will eventually lead to environments such as Sandboxes and Development being fully owned by the application owner/developer.

One additional factor influencing an adoption of the DevOps operating model is the creation of automation software such as Chef and Puppet that allows for 'automatic' configuration of multiple machines to be accomplished with no direct interaction with the developer – a Golden Image as it would be considered. This is evident in the reported factors that are important for DevOps (State of the Cloud Report, 2014).

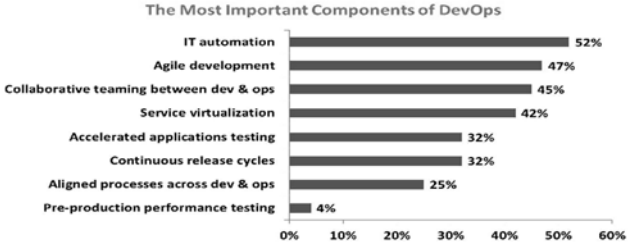

Fig. 6 – Most Important Components of DevOps

From this chart, IT Automation is the major contribution for the adoption of a DevOps model within various IT shops across the world.

4. CONTAINER VIRTUALIZATION

Although IT shops are experiencing the flexibility and agility of the cloud computing model, specifically the ability to meet Green IT goals, reduce overall capital and expense costs and leverage quicker time to test and validate applications with DevOps, the virtualization phenomenon is solely focused on the infrastructure as a service layer with little to no relief to the application developer themselves as it pertains to platforms, isolation, and security for their applications that may be consolidated on a virtual machine.

The concept of the 'container' is from the 1970's where UNIX and BSD operating systems would be able to isolate the user space of the application from the kernel space of the operating system making the application independent of the operating system itself. This feature was called 'change root (Vietnam Ubuntu Forum, 2011).

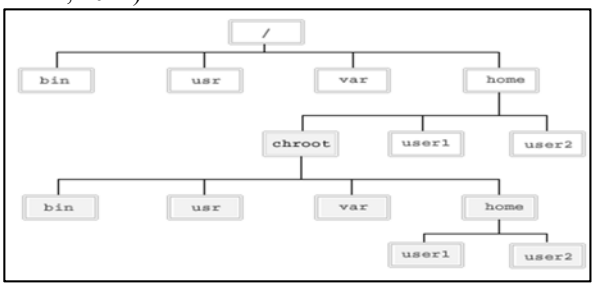

Fig. 7 – CHRoot Diagram

Although the Linux container is similar it has been improved since with advanced capabilities and easier management as compared to the legacy implementation. This technology addresses the application specific concern of isolation, portability, scalability and security just as Hypervisor has done for infrastructure services.

With the container concept, the operating system is still core to its operations; however, the various user space dependencies are maintained within the container therefore allowing the portability from one system to another as long as the same container 'engine' operating system is used. The scalability is increased since now more than one application can be 'installed' within an operating system without the concerns of conflicts. While the application is isolated from other processes of the 'shared' application containers due to its context being within the container only (Bottomley, 2014):

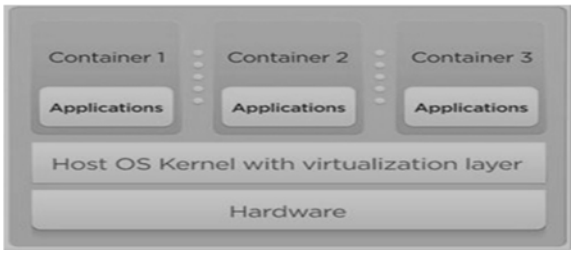

Fig. 8 – Container Architecture

The engine of the container solution, known as the Host Operating system Kernel with Virtualization layer, is installed on an operating system. This engine interacts with the Kernel space thus with the various hardware components that may be present on the server. Options do exist as to where the BINS/LIB files are stored – either in the container or shared amongst various container within the engine(Bottomley, 2014).

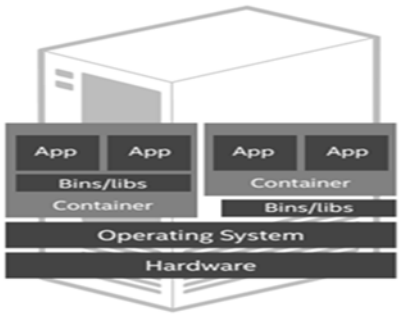

Fig.9 – Container Deployment Options

5. COMPARISON

Both the Container and Hypervisor technologies have advantages and disadvantages as it pertains to the performance, scalability and operations of workloads. These factors must be considered in order to ensure that the infrastructure, application or business needs are being met.

From an operational perspective, we have already discussed the various methods in which the container or hypervisor technology can be leveraged. For both 'systems,' the amount of operational support is the same – the shift is in which department is responsible for the maintenance of the 'system' – the Infrastructure (Hypervisor) or Application (Container) team. Although in both scenarios, the Infrastructure team does take on a bit more overhead since the Host servers themselves are offered as an Infrastructure as a Service model.

Scalability or the means by which more can be added or removed – elasticity, depending on demands and resource usage is slightly different between the two. They are similar to the sense that they both have a physical server limitation – meaning that if resources are fully consumed by the workloads each of these solutions would require the build-out of new infrastructure servers to allow for capacity constraints.

The container operations for scalability are simpler in such that the Application owner or programmer only needs to allocate a new container and then the application can leverage those resources, whereas with a hypervisor new storage, cores and memory needs to be allocated then an operating system is installed.

Fig. 10 – Scale of Hypervisor Guests

The hypervisor expansion is accomplished by adding more guests to the current host. These guests require resources in order to expand the density of its footprint. Those resources include the hardware, operating system and then also the applications that are provisioned for those instances.

Fig.11 – Scale of Containers

The container density is increased by adding more container id's which in turn add additional capacity within isolation from one another application with no additional operating system requirements.

Lastly, the performance of both the container and hypervisor solutions are different depending on the measurement being measured. For core computing power – Linpack: a measurement of the algorithmic computations performed by the CPU for a given algebraic input (Falter, *et, al* 2014).

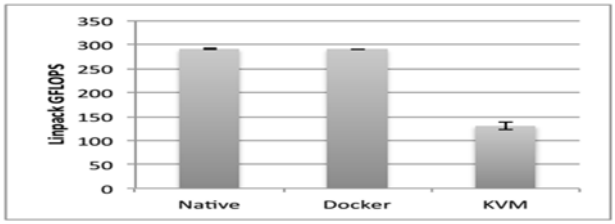

Fig.12 – Linpack Comparison [10]

The capability to execute mathematical complex computations is reliant on the libraries within the User Space with limited need for Kernel Space processing. Interpreting this graph, we see that the native and container systems process in about the same manner while the virtualized system is only half the capability. This would be expected since the processing of the algorithm is dependent on the systems interaction with the CPU through execution steps within either direct (Container) or abstracted (Virtualized) processing layer (Okai, *et., al* 2014), (Uddin *et, al* 2014).

Other performance factors exist including Disk I/O, Random Memory Access, and Network Latency – similar results were collected where the native and container systems performed in a similar manner while the virtualized system took some performance degradation(Uddin *et, al* 2014).

6. DECISION CRITERIA

Several factors must be considered when determining how to approach an architecture for deployment of IT services. Some of these factors are crucial to meet business needs while others would meet operational readiness factors such as reliability, scalability, security, and cost. Each of these factors have specific aspects which cause a weighting of the factors to be considered.

Reliability, the ability of the system to be both fault-tolerant and highly available with little to no down time due to system or application failures, is a key business objective. Murphy's Law states that when something can go wrong it will, in this case any IS system is susceptible to failure from hardware or software 'glitch.' The failures cause the system to become degraded or unusable therefore interrupting the business aspect of the system.

Scalability, the ability of the system to grow either organically or by demand, is both a business and IT objective. From a business perspective, the service must be able to address the needs and demand of the end users who interact with the system. While for IT, the scalability provides the flexibility to be distributed in such a manner that meets the IT requirements or needs at a point in time.

Security, the ability to provide protection for the confidentiality, integrity and availability (CIA – tenants of Information Security), is both a business and IT objective. Depending on the application, there could be sensitive data such as Personally Identifiable Information (PII), Company Intellectual Property or other regulated data which require specific information security controls in order to ensure the data is protected. Also, security includes the means by which to ensure that the operations of the IS application or system is being performed as designed rather than being used in a manner not designed for (Uddin *et, al* 2014).

With these factors in consideration and noting that the various specific considerations for each of the factors have a potential weighting associated with them, we can now begin to build a decision model that can be guide the business and IT staff to choose a specific deployment for any given application or system.

7. DECISION MODEL

The decision making process at the surface would appear to be one that can be made with little effort and with ease, however, when we consider the factors mentioned previously, the process is little more

complicated than initially perceived. When taking into consideration not only the cost which is typically the first and quite honestly the only factor most IT organizations consider when making a technology choice there is a more to consider and factor into the decision.

We can take the factors we reviewed previously which included reliability, scalability, security and cost. The simplest equation for calculating the 'score' would be represented by:

$$Score = R + S + S - C$$

We have found that there are various sub-factors that also need to be included and weighted in this equation such that all contributing factor are considered. For reliability, we must consider High Availability (HA), Fault Tolerance (FT) and Processing Time (PT). However, with these considerations, HA and FT are ½ the value of PT due to the nature of the PT being related to time vice an additional system parameter to maintain availability. The following equation would represent this:

$$\text{Re } liabilty = \frac{.5(HA + FT)}{PT}$$

This mathematical equation is interpreted to show that as HA or FT increase then Reliability would also increase, however as PT increases we see Reliability decrease.

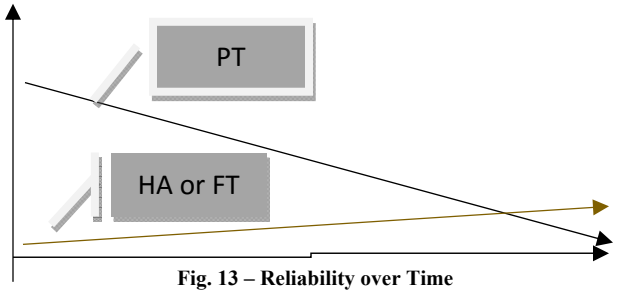

Fig. 13 – Reliability over Time

For scalability, we must also consider the following parameters, Processing (PR), Memory Usage (MEM) and Disk Input/output (IO). In this case the inverse of the addition of these factors would determine the scalability factor like:

$$Scalabilty = PR + MEM + IO$$

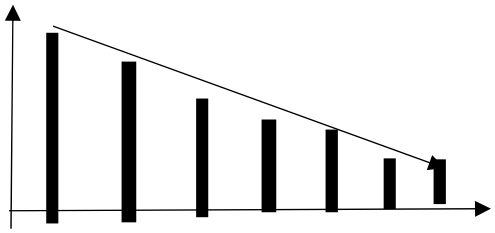

Fig. 14 – Scalability Component Not-Used over Time

This mathematical equation, over time, shows a linear relationship that when any of the resources of the system decrease thatthe scalability decreases – meaning we would need to add more resources to raise the scalability or handle the increased workload

For security, consideration must be made to the Regulations (REG), Security Controls (SEC) and Usability (USE). In this manner, the mathematical equation that represents this relationship is:

$$Security = \frac{SEC}{REG + USE^2}$$

This mathematical representation shows that as Regulations or Usability increase that Security will decrease while increasing Security Controls increases the Security of the system.

With these factors now identified, the simple scoring equation is more complex and becomes:

$$S = \frac{.5(HA + FT)}{PT} + (PR + MEM + IO) + \frac{SEC}{REG + USE^2}$$

Whilethe parameters of this equation were explained and the understanding of why these factors were identified, we must also consider the reasoning for a mathematical simplification of a complex decision we must understand that some of these factors can be empirically determined while others are more subjective in nature and must be thought through before making a representation of that specific variable.

The scoring (S), is based on the various factors in the initial simple equation which was the addition of Reliability (Mostly Virtualization), Scalability (Mostly Container) and Security (Both Contribute) – meaning there is a weighting towards which solution would meet the requirements of the system to be designed or optimized. In fact, we can simplify this equation to identify whether we need Virtualization or Containerization within our solution by this equation:

$$Score = \frac{.5(HA + FT)}{PT} + (\frac{1}{3}(PR + MEM + IO))$$

The reasoning behind removing security from the complex equation is threefold:

- Security Regulations affect Data not system
- Security Controls would be needed in either
- Usage would be similar across either

8. SCORING GRAPH

The scoring graph is a quantitative number identifying the solution that should be greatest considered and is represented by the following:

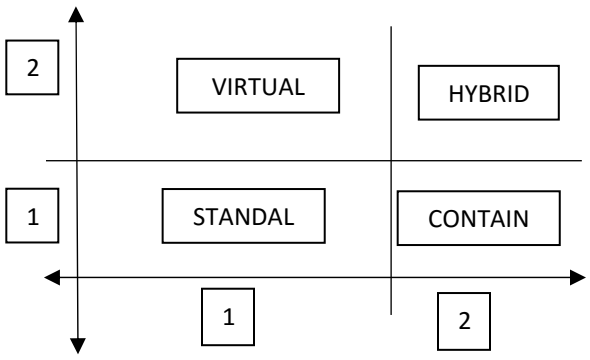

Fig.15 – Quadrant of Scoring

The scoring in this model would align to the criteria below for a probable choice to meet the requirements.

Example Usage

Consider a system such as Google search, a service that is constantly being used by millions of people and returning numerous pieces of information for End Users to consume. Google search would need HA and FT, however, let's assume that the Processing Time increased greatly from milliseconds to seconds, a factor of 1000, therefore the reliability equation would look like:

$$\text{Re } liabilty = \frac{.5(1+1)}{3} = .33$$

While the CPU, MEM, and IO are not fully utilized meaning that low percentages, scalability looks like:

$$Scalabilty = \frac{1}{3}(10\ \% + 10\ \% + 10\ \%) = .1$$

So, the score would be:

$$Score = .33 + .1 = .4$$

On our previous discussed scoring graph, this would indicate a standalone machine to be added to the system with Network Load Balancing (NLB).

9. CONCLUSION

The approach for this research has been to identify the various technologies that can be used to deploy a system or application, understand those technologies capabilities to include weaknesses and strengths, review the various factors in deciding which technology could or should be used and consider the effects on the business and IT requirements as it pertains to the suggested architecture.

This understanding was then represented in a mathematical equation and related to a graph to assist the IT Infrastructure or Application's teams with making a decision to leverage a particular technology. This guidance does make some assumptions such as the system or application is understood enough to perform the necessary measurements to determine the input values, the system or application is looked on as a single entity although a server or service may have multiple tenants and that Security is neutral with respect to the service delivery model chosen.

More information and research needs to be conducted as performance of Virtualization, Containerization and Standalone systems improve with time and research through breakthroughs like increased capacity, optimal resource usage and improved overall processes.

REFERENCES:

Eden, M.,(2014) A Survey of Performance Modeling and Analysis Issues in Resource Management across x86-based Hypervisors in Enterprise Cloud Computing Deployments

Okai, S., M. Uddin, A. Arshad, R. Alsaqour, A. Shah, (2014) "Cloud Computing Adoption Model for Universities to Increase ICT Proficiency," in SAGE Open, Vol. 4, No. 3, 2158244014546461.

Russell, B., (2014) IBM, Cloud Expo, Linux Containers – NextGen Virtualization for Cloud (Intro Overview)

Seitz, P., (2014) Cloud Computing Sales to triple by 2017, New Forecast Says,

Sharma, A., EMC Forum (2011), Flash 1st – A Powerful Date Centre Optimization and Consolidation Approach,

State of the Cloud Report, 2014

Uddin, M., A. Shah, A. Abubakar, I. Adeleke, (2014) "Implementation of Server virtualization to Build Energy Efficient Data Centers," J. Power Technol., Vol. 94, No. 2, 85–94.

Uddin, M., A. Shah, A. Rehman, (2014) "Metrics for Computing Performance of Data Center for Instigating Energy Efficient Data Centers," J. Sci. Ind. Res., Vol. 73, 11–15.

Uddin, M., R. Alsaqour, A. Shah, T. Saba, (2014) "Power usage effectiveness metrics to measure efficiency and performance of data centers," Appl. Math. Inf. Sci., Vol. 8, No. 5, 2207–2216.