**ECPI University**

**Cloud Solution:**

**“Effect of Virtualization on Scalability”**

**CIS 142**

**Rafat Khandaker**

**10/21/18**

**Introduction**

In the technological world, virtualization techniques are revolutionizing the way we think and use computer technology. Virtualization has improved the ability of on-premise infrastructure to handle scaling demands. The technology started from developing the ability for computers to run multiple threads to process simultaneously on the same system **(Sean, 2018)**. The ability to create new instances of a system within the same physical resource became to be known as virtualization **(Jamsa, 2013)**. Virtualization has many advantages in both vertical and horizontal scaling. In vertical scaling, virtualization offers the ability to load an image copy on multiple servers; essentially being able to move the service like a container **(Jamsa, 2013)**. Also, virtualization offers the ability to increase resource demands on the same system by increasing resource allocation blocks. In horizontal scaling, virtualization offers the ability to create failsafe deployments & ensuring disaster recovery improvements **(Jamsa, 2013)**. Also, virtualization allows applications to run in separate instances within the same domain, improving CPU utilization and segmented resources.

**Scaling Problems & Solutions**

Prior to virtualization, computers services were deployed on a single server, operating under a single processor. This single service, would usually be deployed with a database and other resources required within the same server **(Jignesh, 2018)**. When scaling is required to handle larger loads, organizations would have to invest into building a new server and deploying the same resources, handling requests through load balancing and etc. This process would consume a lot of time and money to re build infrastructure to fit an organizations’ need. This architectural model was known as the **Monolithic Model.** In order to make improvements to this model, organizational infrastructure migrated into a more **Microservice Architecture** approach**.** Chris Richardson has pointed out the advantages of scalability within the Microservice Architecture, a design technique to decouple services within a business system **(Richardson, 2017)**. The problem in Monolithic vs Microservice approach had to do with an organization's available resources. If a company had limited servers then they are forced to deploy their services in a more monolithic system, increasing the complexity to scaling new systems. Virtualization is able to solve the challenges, deploying microservice driven development and deployment by scaling a system based on its needs. Virtualization is able to segment and deploy multiple instances of a service within the same hardware resource, saving money and maximize resource utilization.

**Different Service Layer**s

Virtualization technology offers advantages in both vertical and horizontal scaling. The Java enterprise system architect framework, describes the different layers in both infrastructure and application services that is available for scaling. The infrastructure level platform services, or vertical scaling components are composed of Operating system platform & network transport system. The middle ware service is composed of the security policy, runtime environment, messaging & persistence of the service; this the entry or container to our application services. The application service composes of the integration and user collaboration **(Sun Java Enterprise, 2005)**.

**Horizontal & Vertical Scaling Advantages**

Virtualization is able to increase advantages of vertical scaling by containerizing the platform services (Network transport & Operating System) & deploying it in multiple physical resources or even increasing the utilization of the virtual instance, itself. This increase the ability for the service to run in multiple environments & ability for service to recover from disaster because the system itself is virtualized. If our physical system crash, we can simply deploy our virtual solution to another physical asset. If our virtual solution crash, we can simply re-image a new instance within the same physical asset.

The horizontal scaling benefits of virtualized technology includes the ability to add more virtualized machines into existing resources **(Jignesh, 2018)**. Developers are able to develop decoupled applications in multiple instances without disturbing the existing environment. Those decoupled instances can be scaled individually by deploying into separate instances. Fail save development patterns emerge from virtualized technologies. Images of virtualized services can be stored and backed up from previous versions.

**Conclusion**

In conclusion, virtualization has improved the capabilities of a company to scale its resources. Vertical scaling of resources is made easier through virtualization by containerizing the service and deploy into different server environments, including operating systems and improved network environments. Horizontal scaling is made easier because multiple instances of the same service can be deployed within the same hardware asset. Developers are able to decouple their services and inherit a microservice architectural environment, where services can be split into segments and scaled individually according to its needs. Virtualization also allows failsafe deployments and disaster recovery because system services can be stored and backed up as images.

**References**

**(1)** Jamsa, K. (2013). *Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security, and More*. Burlington, MA.: Jones & Bartlett Learning.

**(2)** *Chapter 2: Java Enterpise Systems Solution Architectures*. (2005). Sun Java Enterprise Technical Overview.Retrieved From: <https://docs.oracle.com/cd/E19396-01/819-0061/architecture.html>

**(3)** Solanki, Jignesh. (2018). *Vertical Scaling & Horizontal Scaling in AWS*. Cloud Zone. Retrieved From: <https://dzone.com/articles/vertical-scaling-and-horizontal-scaling-in-aws>

**(4)** Conroy, Sean. (2018). *History of Virtualization.* Retrieved From: [https://www.idkrtm.com/history-of-virtualization/ (Links to an external site.)Links to an external site.](https://www.idkrtm.com/history-of-virtualization/)

**(5)** Richardson, Chris. (2017). *Microservices Pattern*. Retrieved From: <https://microservices.io/patterns/microservices.html>