**Fredrick Onduso Ondieki**

**CST-221 Operating Systems Concepts**

**CST-221 Assessing Virtualization Software Assignment**

**John Zupan**

**April 01, 2019**

**Git url: https://github.com/FREDDYSMALLZ/Operating-Systems-Concepts-CST-221.git**

**CST – 221 Assessing Virtualization Software**

Activity Directions:

In this project assignment we were required to conduct a mini research task to identify current trends in software virtualization. Using the textbook as a guide on concepts and core technologies, search the GCU digital library (library.gcu.edu) and the Internet to identify the current leading virtualization software manufactures, as well as their products.

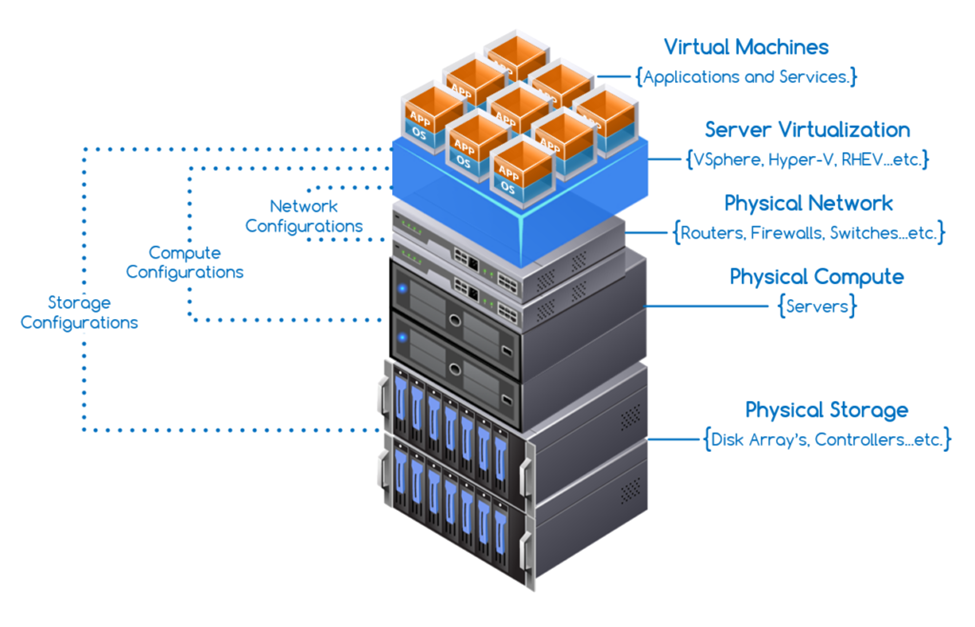
**Application Description relating to computing need for virtualization**

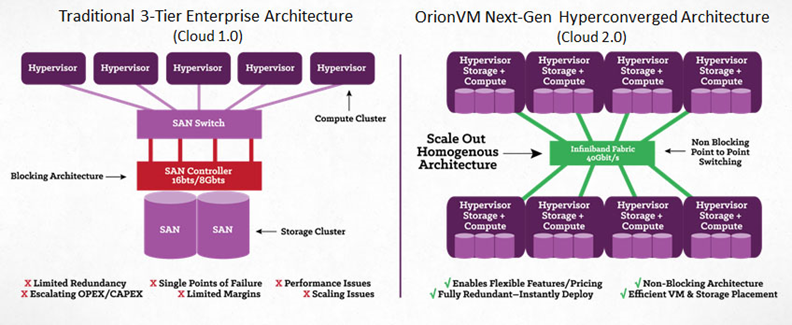
One of the topics that I found fascinating to study is the Hyperconverged Infrastructure (HCI). Hyperconverged Infrastructure is an example of Information technology framework that combines application resources and merges them into a single system. HCI was built to assist in streamlining the management and scaling of datacenter resources. Also, it allows for an organization to provide great mobility in workloads, applications, scale in/out resources and restore data.

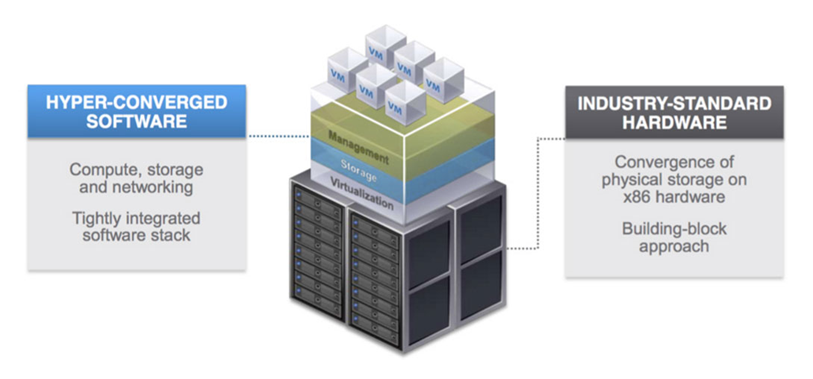
HCI is important to small enterprises as it allows them to start using their resources as they will expand over the time. Most of the vendors can realize great potentials as they are able to save in terms of cost in various areas such as IT, data centers and lastly on space. Vendors also see a drastic potential in cost savings in areas such as IT labor, data centers, power and space, and possibly not use licensed software.

Lastly, with HCI all the components that we have discussed above are constructed virtually and managed by the same set of tools and devices. The benefit of this is that it allows all aspects of each partition to distribute the very many resources associated with it as needed. As one of its benefit, it scales very easily; if you need more space for your enterprise, then it scales up to meet your needs. The user will simply install a new set of nodes and improve the overall performance of the system.

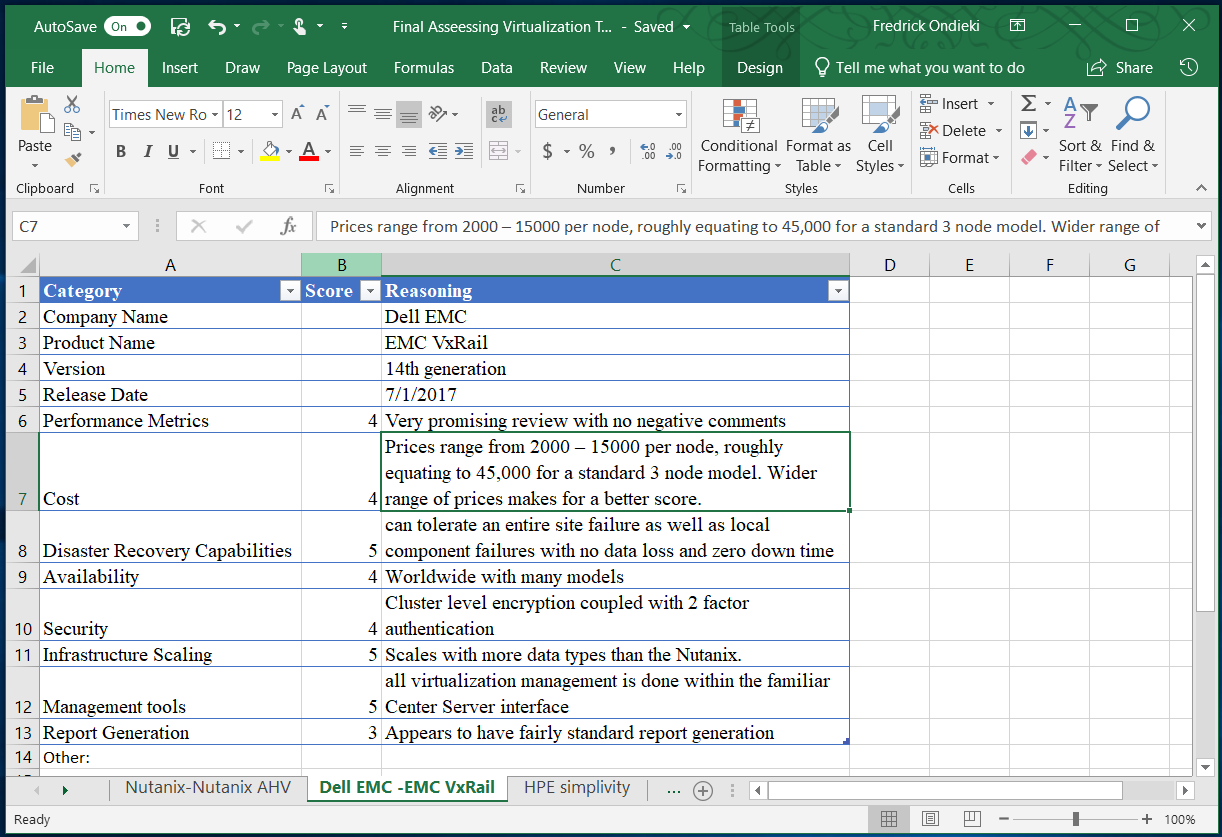
**Diagram of architecture of the virtualization environment**

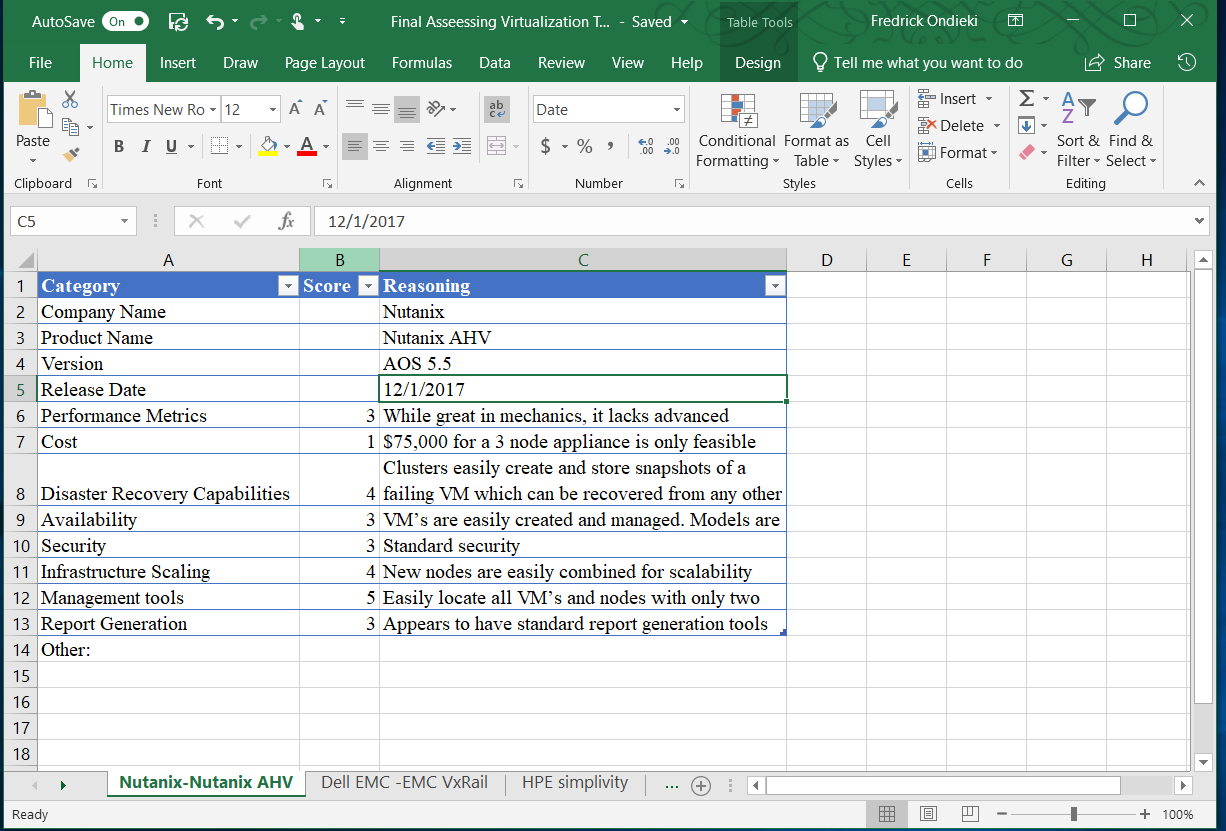
****

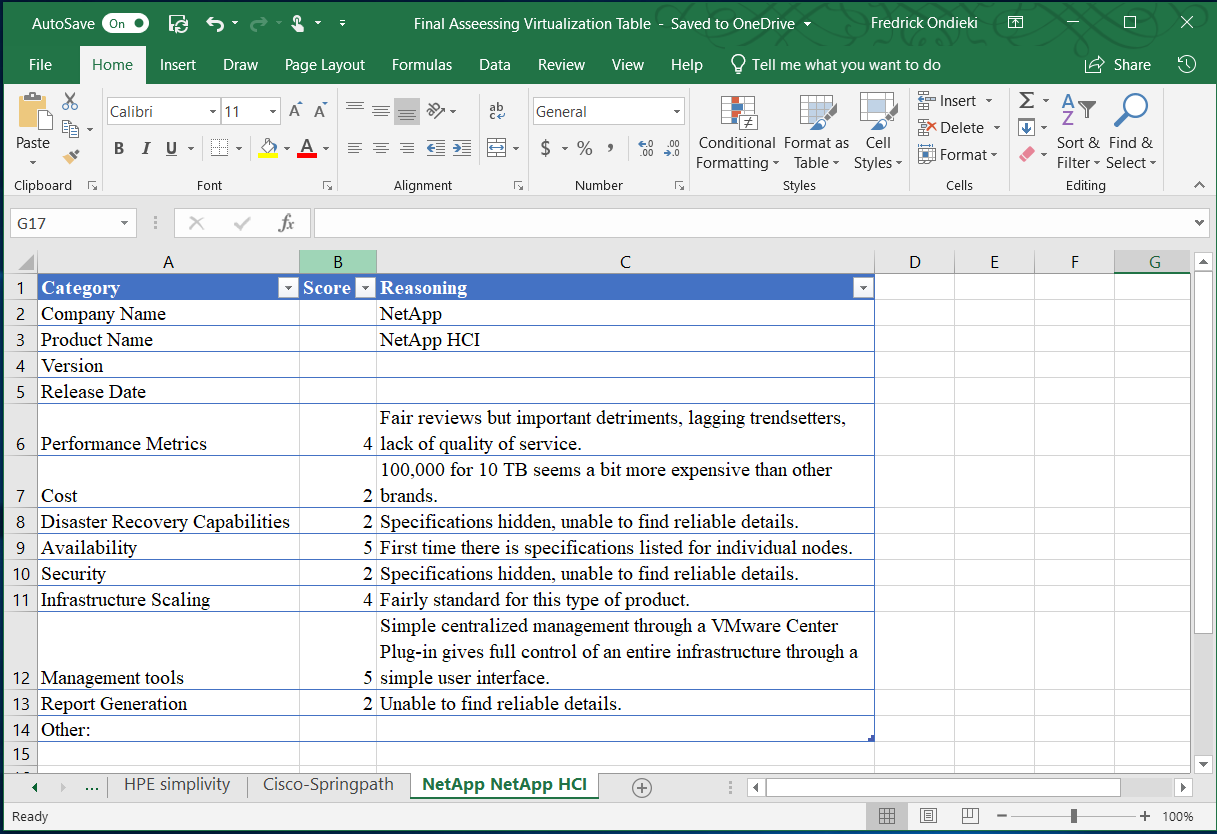
****

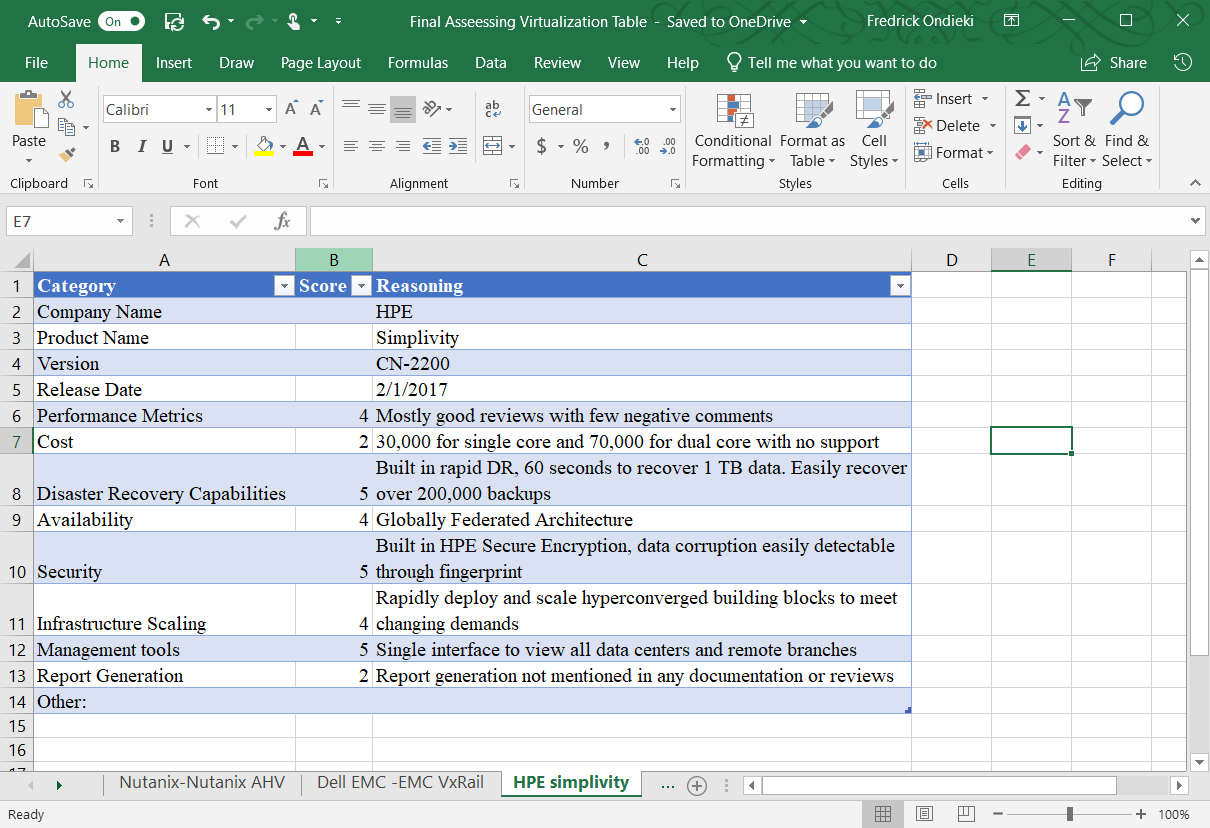
****

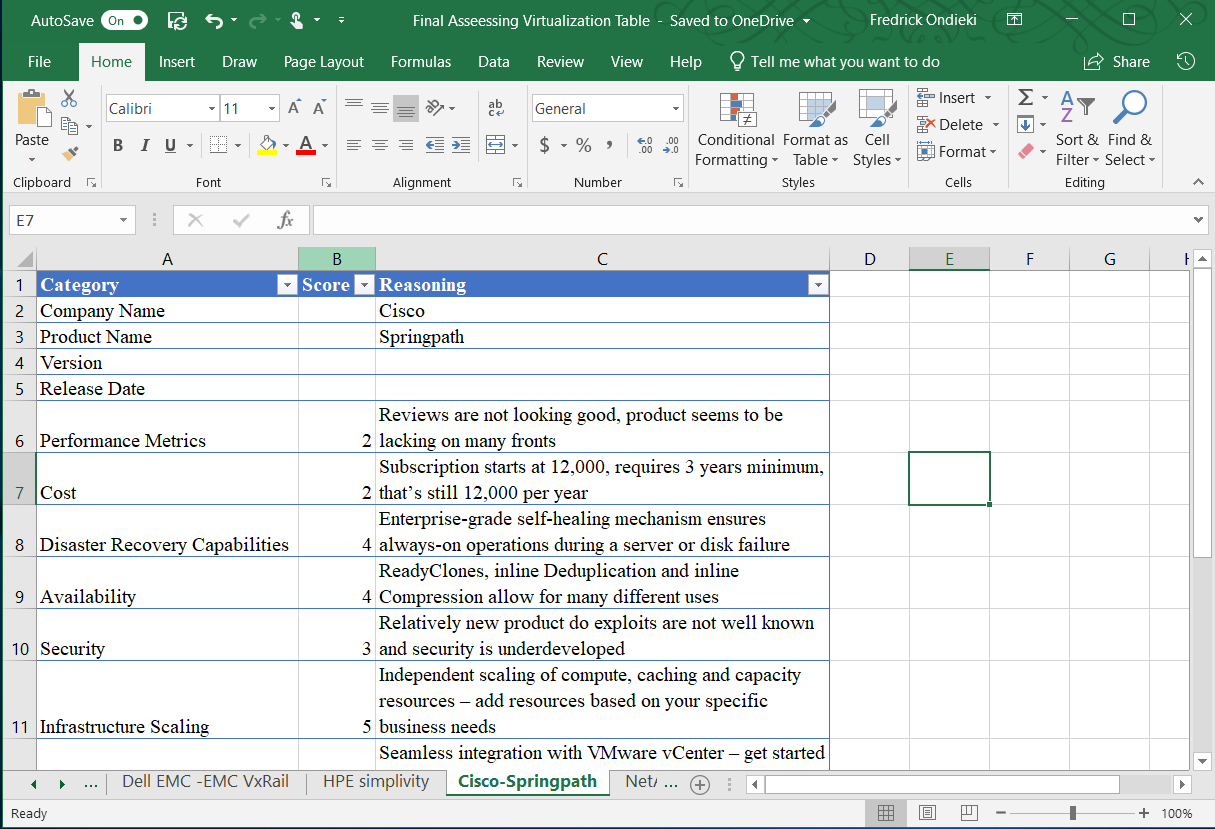
**Assessing Virtualization Software Tables**

The screenshots bellow shows the software tables used for the research in identifying the five current technologies and companies. (The data is included as an attachment for this assignment on a spreadsheet). In addition, the tables show the scores and justification behind the scores. 





****

****

**Top Choice Analysis**

After careful analysis of the available virtualization software available, Dell EMC *VxRail* is my top choice. The reason behind is that it has non-negative ratings and its reviews are high. Secondly, *VxRail* version does not need much of the end user ratings as it can tolerate site failures with minimal data loses. Compared to the other models, *VxRail* scales much comfortable and better than *springpath* and NetApp.

Lastly, it can also accommodate a wide range of datatypes compared to the other brands that we have seen above. Most importantly, it has a greater ability to virtualize all problems that may be associated with the operating system.

**Applications of Research Findings**

**Why a hardware-based approach using VT-enabled CPUs can perform poorly when compared to translation-based software approaches?**

Hardware based approaches using VT-enabled CPUs can perform poorly in relation to translation-based software approaches. While the hardware-based approach will run faster than the software counterparts, it suffers performance issues in that the guest and host must use the same platform. This is not true in the case of the translation-based software approach, because the software emulates the guest’s platform. The downside of the software approach is that the system will run more slowly.

**Give one case where a translated code can be faster than the original code, in a system using binary translation. What is the main cause of the difference in performance?**

Binary translation allows code written for an instruction set to run on another destination architecture, without access to the original source code. In addition, Binary translations are a means of low-level re-engineering focus on sending machine code from one system to another. Procedurally the source system translates the binary machine code into a higher-level abstract code before sending it to another machine which then interprets the rough calls and then places then through a binary machine code translator to generate new machine code.

**Example**

An example of this would be when a user on a network builds a program before sending it thorough the network to another user on a different device. Because the other user might be working with a different version of the operating system, the translation is done to allow the new device to run the code. A situation where the translated code could be faster than the original code is if the target system has an updated version of the source system. In the case of HCI systems, the binary file would not even have to be put through translation as it would simply be access from the other VM. This is because each VM is using the same set of nodes that is available to the whole system, so there are no differences between the virtual machines.

References

Dell EMC. (2018). Dell EMC VxRail. Retrieved from https://www.dellemc.com/en-us/converged-infrastructure/vxrail/index.htm

ItCentralStation.com. (2018). Cisco HyperFlex HX-Series vs Springpath. Retrieved from https://www.itcentralstation.com/products/comparisons/cisco-hyperflex-hx-series\_vs\_springpath

Rouse, M. (2018). Hyperconvergence. Retrieved April 21, 2019, from https://searchconvergedinfrastructure.techtarget.com/definition/hyper-convergence

StoragePricing.org. (2018). Nutanix Pricing. Retrieved from http://www.storagepricing.org/category/nutanix/

Springpathinc.com. (2018). Springpath. Retrieved from http://www.springpathinc.com/

Gelas, J. D. (2008, March 17). Hardware Virtualization: The nuts and bolts. Retrieved April 21, 2019, from https://www.anandtech.com/show/2480/4

HPE.com. (2018). HPE Simplivity. Retrieved from https://www.hpe.com/us/en/integrated-systems/simplivity.html

Mice. What is the difference between hardware and software virtualization? (2010, March 09). Retrieved April 21, 2019, from https://superuser.com/questions/117774/what-is-the-difference-between-hardware-and-software-virtualization

NetApp. (2018). NetApp HCI. Retrieved from https://www.netapp.com/us/products/converged-systems/hyper-converged-infrastructure.aspx

Ung, David. Cifuentes, Christina. (2005). Dynamic binary translation using run-time feedbacks. Elsevier B.V.

Nutanix. (2018). What is hyperconverged infrastructure? Retrieved from https://www.nutanix.com/hyperconverged-infrastructure/

Wyman, E. (2016). Nutanix Acropolis Review: "Making the jump to Acropolis is worth it". Retrieved from https://www.trustradius.com/reviews/nutanix-acropolis-2016-08-30-12-01-00