# Evaluation of VM Migration within a Cloud to Determine Transfer Latency and its Potential to Provide Optimization and Fault Tolerance Capabilities

Dennis Guster
Saint Cloud State University
dcguster@stcloudstate.edu

Laura Lebentritt
University of Maryland, University College
lauraleigh4097@att.net

Ashok Jonnalagadda
Saint Cloud State University
joas1301@stcloudstate.edu

## Abstract

Cloud computing offers many advantages, one of which is the ability to migrate or copy VMs within the cloud. This capability allows the basic design to be modified at will and allows for end-less logic configurations of resources within the underlying physical infrastructure. However, the migration of resources must be done in a timely manner particularly if real-time applications such as e-commerce are being supported. This paper used the authors' home cloud to measure the speed with which VM and their associated data stores can be migrated with the cloud. The results indicated that the VMs could be migrated in the 4 to 21 second range at the data store could be migrated in 10 to 86 seconds. It was found that generally the larger the block of transfer data the longer it took. However, exceptions occurred typically related to the characteristics of the underlying hardware layer. It was found that the migration values observed were a little high to truly support real time applications. However it was concluded that, if proactive planning is used the migration process can save time and resources.

# 1 Introduction

There are many advantages to cloud computing, but it also adds layers of complexity due to its virtualized architecture. Further, cloud computing compliments the security strategy through the isolation of applications [2]. Performance advantages on the processing, disk and network level can also be achieved [13] [14] [6]. For the most part these advantages stem from the fact that the basic hardware configuration can be enhanced through the flexibility that virtualization provides. In the traditional model a host was configured and applications were often added as long as adequate performance was achieved. In contrast, within a cloud based virtual world a needed application is identified and the necessary resources are allocated within the cloud. Further, the resource provisioning strategy is often dynamic so that if performance is not adequate resources can be shifted to that application from other parts of the cloud [2]. Or conversely, the virtual machine (VM) hosting a given application can be moved to a part of the cloud that is less busy.

The key to an effective cloud design is to maximize resource utilization while reducing the number of physical machines [17]. This strategy often requires a degree of finesse and the ability to be creative in the virtual cloud design is imperative. The resulting virtual design is often geared to providing improved security by using a layering approach [9] as well as improved network reliability and performance through the use of virtual networks/file systems [7]. The concepts of isolation and performance enhancement can be illustrated in the following example.

```
buster@hosta:~$ df
Filesystem              1K-blocks    Used    Available Use% Mounted on
/dev/sdc2               15480816    4529828  10164608  31%  /
nfshost.cloud9.local:   41283968    1591424  37595520   5%  /rhome
```

In this example, the file systems are displayed by a user named buster for the host “hosta”. There are two file systems displayed, one local and one remote. The first file system is local and mounted as the root on device sdc2 (sata physical drive c partition 2). The remote drive is mounted locally as /rhome and resides on the host nfshost in the domain cloud9.local (a domain isolated within the cloud).

```
buster@nfshost:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/sdc2       15480816 8106752   6587684  56% /
/dev/sdd3       41283904 1590768  37596036   5% /core
```

A look at the file systems contained on the host “nfshost” reveals that it appears to contain two local file systems, sdc2 which was also the primary local file system on the host “hosta” (note that sdc2 is being shared across hosts within the cloud) and the other local file system is sdd3 and is mounted on /core. A look at the directory structure on “nfshost” reveals the following.

```
buster@nfshost:~$ pwd
/core/rhome/buster
```

Therefore, one could conclude that /rhome on “hosta” is really stored on “nfshost” as /core/rhome and is stored on device sdd3 and results in the following benefits.

1. Encapsulation: the actual location of the data is not directly defined on “hosta” only the next host to search.
2. A single protection point: all user data in the cloud could be placed in /rhome and a multi-level security plan invoked.
3. A global file system that will follow the user no matter what host he/she is logged into.
4. A single physical disk system that can be optimized to provide the best performance, i.e. higher speed drive within a SAN architecture.
5. Automatic replication of user data with multiple copies.
6. Data transfer within the friendly confines of the cloud’s internal domain: cloud9.local.
7. The data transfer could be configured to take place on virtual networks that runs on the main-bus of a physical host in the cloud which provides both isolation and performance capabilities.

From the example above it is clear that there are numerous benefits to tuning the virtual configuration within a cloud. However, it cannot be accomplished by just one virtualization instance. Rather, a comprehensive plan is needed that applies virtualization across all resources in the cloud including: memory, processors, file systems and networks. Therefore, the purpose of this paper is to evaluate the flexible logic of virtualization and apply it to optimizing a virtual machine in the cloud by moving it within the cloud to optimize performance. This strategy will be applied to a University based cloud designed to support instruction and research. Once configured the cloud design will be analyzed for potential benefits in performance as well as security. However, before this process can be actively pursued, the limitations of that architecture, such as transfer delay need to be assessed. Further by evaluating this process, the goal is expanded to gain practical experience, which could then be used to provide students with meaningful hands-on activities that will further illustrate the basic principles of virtualization particularly as related to migration.

## 2 Review of Literature

The literature related to cloud computing makes it clear that cloud computing offers many advantages, however it involves many risks that need to be addressed [2]. To many the virtualized world is a mystery and dealing with unknown security challenges is often the biggest obstacle to the adoption of cloud services [3]. There are many misconceptions about cloud security such as a lack of isolation of data due to the transient nature of VMs [15]. Actually, a properly configured cloud featuring transient VMs can offer an improved level of trust and performance when compared to a classical hosting solution [9].

A useful tool within cloud computing is the ability to migrate VMs across the supporting physical architecture. This mechanism allows for a clean separation between hardware and software, facilitates fault management, load balancing and lessens the degree of system maintenance [4]. One of the most attractive features of this tool is the ability to move VMs within the cloud to optimize performance. This is done dynamically based on workload and the goal is to achieve the movement with a short down time. However if the move is not done correctly the service level of the application involved could be negatively affected [16]. The transfer methods employed could be classified as either adaptive or non-adaptive. The adaptive method utilizes the workload characteristics of the VM to determine when memory pages are transferred. Whereas the non-adaptive method simply transfers the memory pages at the maximum rate the network infrastructure will allow [11].

One means of enhancing the performance and high available during VM migration is to implement a fine grain block identification mechanism. Reducing the granularity of data transferred has proven to be successful from both a downtime and performance perspective [10]. A second method, allows the copying of pages to take place while the VM is still running instead of shutting it down before copying. This method has exhibited some success and the developers of this method have devised a non-linear optimization model to guide the deployment of this methodology [1].

## 3 Methodology

The vSphere software suite will be used to create, replicate and move the virtual machines (VMs). Replication is a particularly important concept within cloud computing because of the desire for fault tolerance, disaster recovery and load balancing. The goal then is to replicate a VMware vsphere to an alternate area. Preferably to a different physical hardware host and then make that duplicate accessible for reclamation through the VMware vSphere Web Client or through the coordination of a full fiasco recuperation agent such as VMware vCenter Site Recovery Manager [8]. One of the concerns of creating replicas is how quickly can they be created and how practical will it be to maintain concurrency? Under ideal conditions the VM would be replicated in multiple places. A common scenario would be to have the VM stored twice in its primary cloud (at different hardware locations) and once in a remote cloud. Updating the replica in the primary cloud is fairly straightforward assuming the two hardware assets are connected by a high speed LAN and only the changes are passed. However, the replica on a remote cloud is more problematic because the connectivity would be provided by a WAN with less speed and less reliability. However, the replica on a remote cloud is critical from a disaster recovery perspective.

Therefore, it is crucial to be proactive concerning VM recovery. If an existing VM fails primary access can be shifted to an existing VM and a new VM should be created immediately. Another concept involving replication deals with moving a VM from one part of a cloud to another. For example VM hosta could be performing poorly because it is competing for hardware resources in zone1 of the cloud, by moving it to zone2 where there is nothing going on would give it more resources and hence better performance. In both case how long it take to copy or move the replica is critical. If the VM is supporting web traffic the typical delay tolerated by an end user is around 3 seconds [5]. Therefore, the move/copy needs to be done most expediently is the applications it support are to function in a timely manner.

To ascertain how practical a move/copy strategy would be in supporting real time application within a cloud a series of experimental trials have been devised. These moves we designed to provide the expected latency for moving a VM image within the zones of the authors' equipment room and campus backup located in another building about 2 blocks away. So therefore all transfers within or between clouds could be considered a transfer across a local area network (LAN) with at least 1Gbps of bandwidth. The results from the first experimental trial appear below in Table 1. In this set of trials the goal was to determine the delay that would result in migrating live VMs with different RAM and virtual CPU levels (vCPU). These results were obtained configuring a VM with varying vCPU values and 12 samples were recorded. Generally, if the vCPUs are increased, then the migration time decreases. So the vMotion software (Migration form one host to other) that migrates the vCPU and RAM resources from one host to other is influenced by the volume of data to be transferred. So therefore, increasing the RAM (and volume of data to be transferred) causes a degree of delay. So performance tuning needs to be done on a case by case basis so that the optimum level of resource allocation results for a particular VM.

Further, one needs to look at the transfer in regard to the underlying architecture. For example, increasing the number of of vCPUs doesn't necessarily increase the performance all the time. In fact it can negatively affect the VM performance. This would be true in a situation in which a physical host is configured with only 10 vCPUs, but has two VM's to migrate, one with 8 vCPUs and one with 4 vCPUs. In cases in which the second VM is active and using 4 vCPUs that process limits the resources available to the first VM. Specifically, a performance test on 8vCPU VM would reveal that it only gets 6 slots for every process cycle and other two slots would have to wait their turn until the next processing cycle. So increasing the number of vCPU not always recommended. The optimum number of vCPUs needs to be coordinated with the underlying architecture. This can be further complicated by the concept of cores versus CPUs. A 4 core CPU while having 4 cores to do processing only has on physical socket and hence all 4 of those cores must share the same physical path to the bus which constrains transfer of IO.

Table 1

Transfer time at different RAM (GB) levels for an 80GB data transfer with different memory and CPU levels.

| Ram | 1 CPU | 2 CPUs | 4 CPUs | 8 CPUs |
|---|---|---|---|---|
| 4 | 5 | 4 | 6 | 4 |
| 8 | 5 | 5 | 6 | 4 |
| 12 | 5 | 7 | 8 | 5 |
| 16 | 6 | 7 | 5 | 21 |

Table 1: Transfer time at different RAM (GB) levels for an 80GB data transfer with different memory and CPU levels

The figures below graphically depict the performance at various levels of RAM at different vCPU levels. With only one vCPU the migration time is relatively stable until the 16GB RAM configuration is reached. With 2 vCPUs there is a speedup at the 4GB level and is the same at the 8GB level. However, 2 vCPUs result in a decreasing performance at both 8GB and 12GB levels. With 4 vCPUs performance was worse in every case when compared to the 1 vCPU level except at the 16 GB level. Interestingly, with 8 vCPUs performance improved across the board with the exception of the 16 GB level which took a dramatic nose dive.

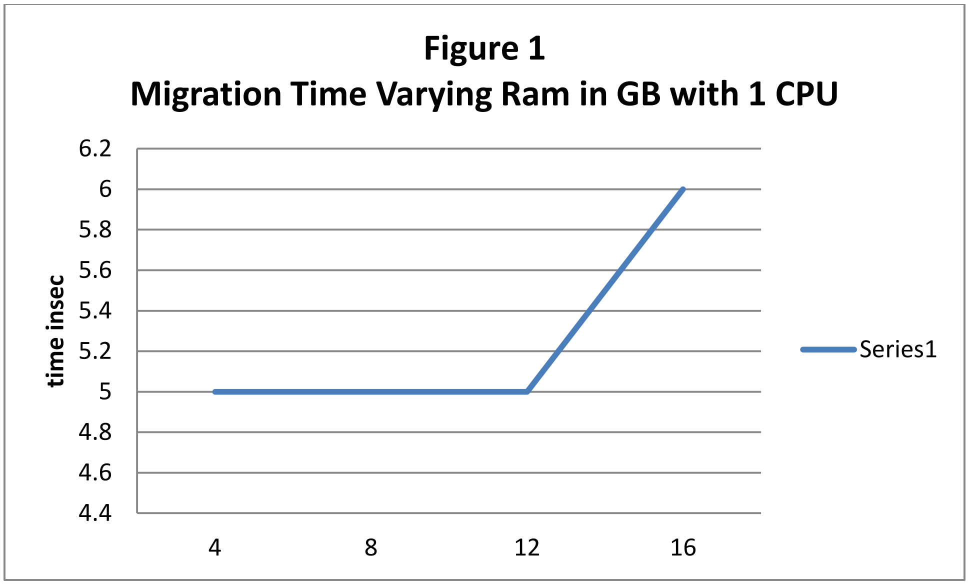


Figure 1: Migration Time Varying Ram in GB with 1 CPU

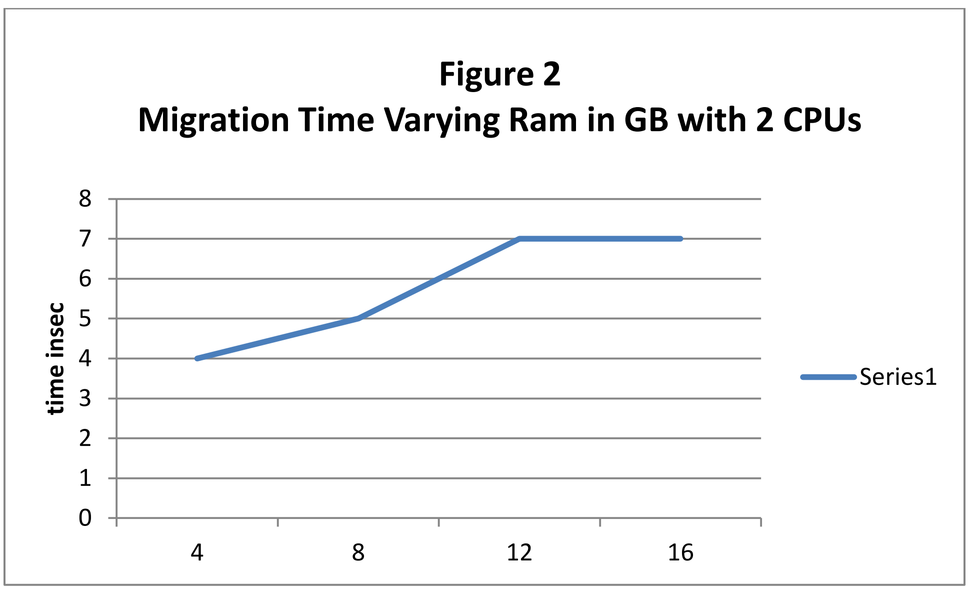


Figure 2: Migration Time Varying Ram in GB with 2 CPU's

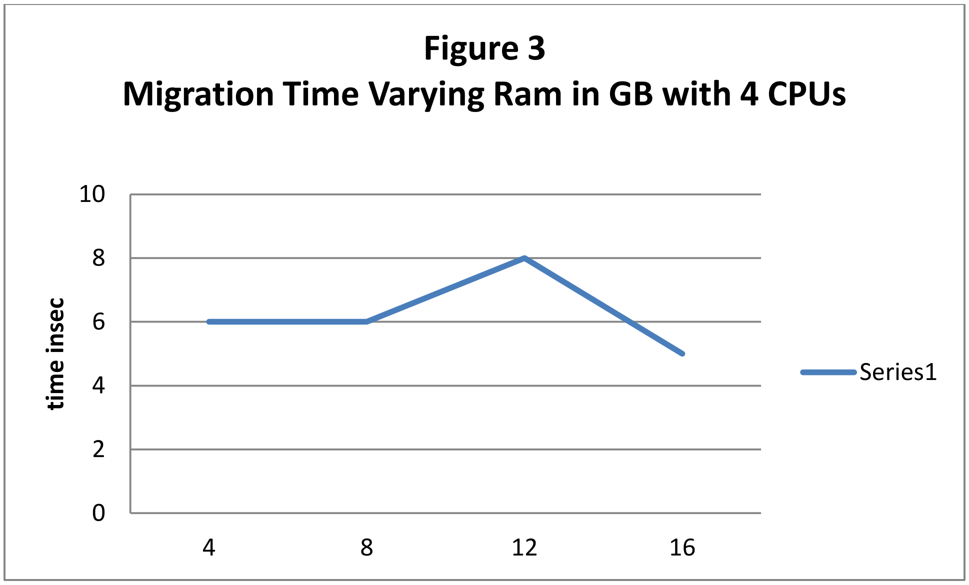


Figure 3: Migration Time Varying Ram in GB with 4 CPU's

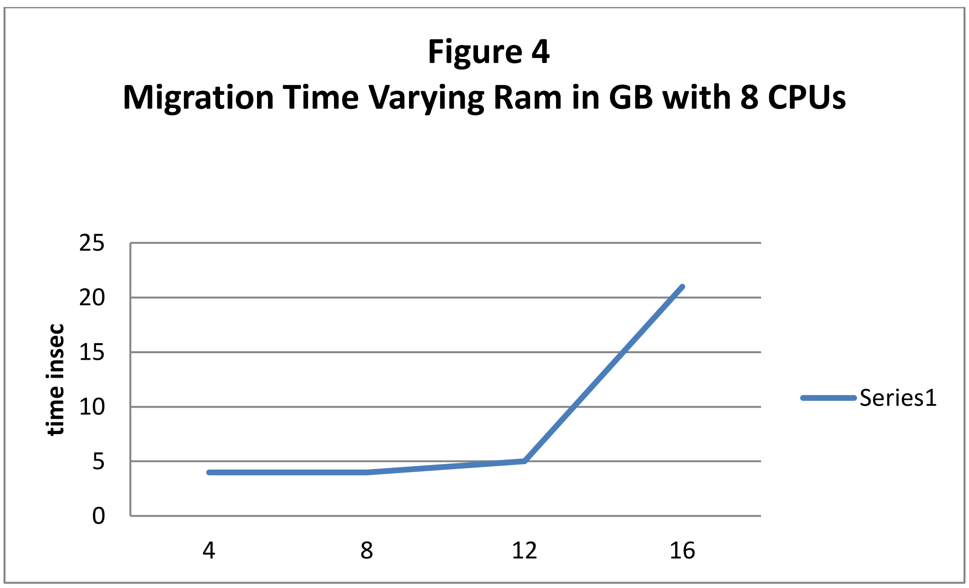


Figure 4: Migration Time Varying Ram in GB with 8 CPU's

In addition to looking at transfer times for live VMs it was decided that it would be useful to look at the transfer time of the storage image of a particular VM. Table 2 below shows the migration times recorded when varying the size of the stored image. Generally, there is a slight linear increase until reaching the 600GB level where there is a spike upward. Once again these values depend on what type of back end storage is being used. For example one would expect that a fiber channel (FC) base SAN (storage area network) protocol would yield superior performance to a network attached (NAS) protocol. Of course the type/speed of the storage processor, storage Type (Flash, SAS, NL-SAS), and RAID (redundant array of inexpensive disks) technology used would all figure in to the performance equation. These experimental trial were undertaken on SAS (serial attached SCSI) disks using RAID 5 technology linked together by the EMC fiber channel based protocol. This setup is pretty much enterprise based hardware and should provide decent performance.

Table 2
Transfer time at different Storage (GB) levels.

| Storage in GB | Time in Sec. |
|---|---|
| 64 | 10 |
| 80 | 12 |
| 99 | 14 |
| 250 | 20 |
| 600 | 86 |

Table 2: Transfer time at different Storage (GB) levels

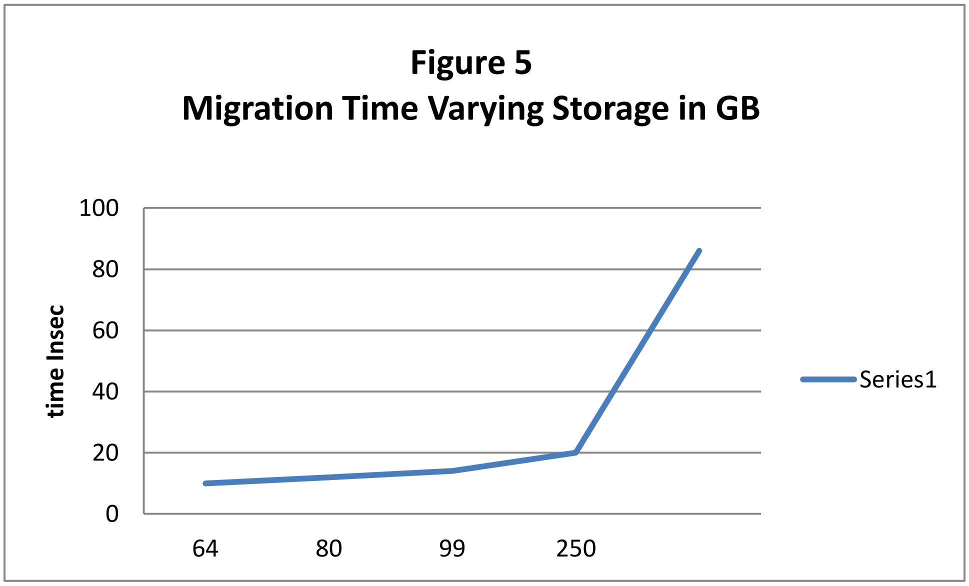

Figure 5: Migration Time Varying Storage in GB

# 4 Discussion/Conclusions

The results generally indicate that the as the amount of data to be transferred increases so does length of time to complete that transfer. However, the expected increase wasn't always linear and the underlying hardware infrastructure could have a major influence on how quickly data could be migrated. So therefore matching the number of vCPUs to the physical architecture as well as being aware of the IO issues inherent in multi-cored processors can help in streamlining the tuning process.

The actual migration values on the surface seem reasonable, but should be evaluated in regard to the purpose of the VM that is being migrated. A simple way of putting it into perspective would be to consider the acceptable client delay in an e-commerce application. It appears that a target response time of about 3 seconds is the maximum that can be safely allowed in standard e-commerce applications. Given the values observed herein the shortest transfer was 4 seconds so if the VM is being migrated to another part of the cloud to gain in performance the transfer time is slightly greater than the largest amount of acceptable. Perhaps if the clientele is limited to North America then there could be scheduled migrations in the middle of the night, but is the clientele come from all over the world there maybe now slack period. The 4 second threshold observed in this paper mean that live migration should only be considered as a last resort. However, it is certainly possible to use this same transfer logic to create replicas of a given VM and use software such as heartbeat to manage the failover when required which should take considerably less time to invoke than 4 seconds. Also, having multiple replicas would make it easy to configure and use a load balancing strategy particularly if each replica used the same data store. It was also shown that the length of time to transfer a data store varied between 10 and 86 seconds way beyond the acceptable minimum for a live system, which indicates that a proactive migration strategy would be required as well.

However, the flexibility and ease of migration provided by cloud software is still a useful system management tool. For example if new hardware is acquired it can be introduced into the cloud and the system can be migrated over the weekend. In case where the application must run 24/7 the systems can be copied and when in place the switchover to the newly configured hardware can take place. In this scenario the amount of personnel time would be greatly reduced. This is also true of any future management of the systems. So in some cases this migration logic will pay for itself in personnel savings over time. In a way these migration capabilities can be viewed in a OOPs framework in that it take the ability to reuse code to a higher level.

# References

[1] Aldhalaan, A., Menasce, D. (2013). Analytic performance modeling and optimization of live VM migration. *Lecture Notes in Computer Science, vol. 8168, pp. 28-42.*

[2] Anthes, G. (Nov. 2010). Security in the cloud. *Communications of the ACM , vol. 53 Issue 11, pp.16-18.*

[3] Bohli, J., Gruschka, N., Jensen, M., et.al. (July-Aug 2013). Security and Privacy-Enhancing Multicloud Architectures. *IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 212-224.* Retrieved from: http://www.computer.org.ezproxy.umuc.edu/csdl/trans/tq/2013/04/ttq2013040212.html

[4] Clark, C. (et.al) (2005). Live migration of virtual machines. *Proceedings of the 2nd Conference of Networked Systems Design and Implementation, vol.2, pp. 273-286*

[5] Guster, D. C., Lee, O. F., & Rogers, D. C. (2011). Pitfalls of devising a security policy in virtualized hosts. *Journal of Information Security Research, 2*(2), 75-83.

[6] Heisswolf, J. (July 2013). Virtual networks—distributed communication resource management. *ACM Transactions on Reconfigurable Technology and Systems, vol.6 Issue 2, article no. 8.*

[7] Houidi, I., et.al. (2010). Adaptive virtual network provisioning. *Proceedings of the ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures,New Delhi, India, pp 41–48.*

[8] Hunter, J. (2015, Feb.). VMware vSphere® Replication™ 6.0. Retrieved from: http://www.vmware.com/files/pdf/vsphere/VMware-vSphere-Replication-Overview.pdf

[9] Jules, A., Oprea, A. (Feb 2013). New approaches to security and availability for cloud data. *Communications of the ACM, vol. 56, issue. 2, pp. 64-73.*

[10] Lu, P., Ravindran, B., Kim, C. (2011). Enhancing the performance of high availability lightweight live migration. *Lecture Notes in Computer Science, vol. 7109, pp.50-64.*

[11] Nathan, S., et.al. (2013). Resource availability based performance benchingmarking of virtual machine migrations. *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, pp. 387-398.*

[12] O'Byrne, A. (2013). A 1 second delay on your page load can cause a 7% reduction in conversions. How do you solve it? Retrieved from: http://wemakewebsites.com/blog/a-1-second-delay-on-your-page-load-can-cause-a-7-reduction-in-conversions-how-do-you-solve-it

[13] Padala, P, et.al. (2007). Adaptive control of virtualized resources in utility computing environments. *Proceedings of the 2nd ACM SIPOPS European Conference on Computing Systems, vol.41 Issue 3, pp.289-302.*

[14] Tak, B., Tang, C., Chang, R. (2011). Designing a Storage Infrastructure for Scalable Cloud Services. *Technical Report*, The Pennsylvania State University.

[15] Viega, J. (July-Aug. 2012). Cloud Security: Not a Problem. *IEEE Security & Privacy, vol. 10, no. 4, pg.3.* Retrieved from: http://doi.ieeecomputersociety.org.ezproxy.umuc.edu/10.1109/MSP.2012.93

[16] Voorsluys, W., et.al. (2009). Cost of virtual machine live migration in clouds: A performance evaluation. *Proceedings of the 1st International Conference on Cloud Computing, pp. 254-265.*

[17] Yan, Y., Shao, L. (2011). Management of the migration in virtual resource. *Communications in Information Science and Management Engineering, vol. 1, issue. 4, pp. 6-11*.