J. Electrical Systems 20-11s (2024): 850-858

[1]Harikrishna Bommala,
[2]Manideep Karumanchi,
[3]Dr. A. Naresh,
[3]Jajjara Bhargav,
[3]Rokesh Kumar Yarava,
[4]Patttipati Naveen Kumar,
[4]Lavanya Rodda,
[4]Mohd Abdul Aleem,
[4]Gajula Sushmitha,
[4]Bangale Nagaraja Rao

Improving Data Transfer Performance to Unlock Potential of Cloud Service Providers

Abstract: - Efficient data transfer among CSPs is essential for the performance of cloud computing's basic operations, such as migration and disaster recovery. In this research, a novel approach to enhance data transmission performance using DTNs is presented. Using DTNs wisely, the proposed method enhances the speed and reliability of sending huge volumes of data across several CSPs. Coordinating local and remote copy operations is also a part of the process as well as a mechanism for DTN-to-DTN transfers. Optimized network settings with NFS connections maximized data transfer speed. In order to optimize the network, we carried out a battery of experiments to determine the best setting by varying network buffer size, synchronous and asynchronous modes, among others. With asynchronous modes and the best possible NFS settings, results showed a dramatic improvement in data transmission speed. The method is scalable and reliable when looking to address the challenge of inter-cloud improvements with regard to data transfer within a multi-cloud environment.

Keywords: Network File System, Data Transfer Nodes, Cloud Environment, Composed Image Cloning, Security

1. INTRODUCTION

As research becomes more reliant on cooperation with other institutions, effective storage, transmission, and exchange of vast datasets become critical components of successful collaborative research [1]. The

---

[1] Associate Professor, Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Moinabad, Hyderabad, Ranga Reddy, India, Mail: haribommala@gmail.com.

[2] Associate Professor, Department of Computer Science and Engineering, Bapatla Engineering College, India. Mail: manideep.karumanchi@gmail.com

[3] Associate Professor, Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Kadapa, Andhra Pradesh, India, Mail: pandu188@gmail.com

[3] Assistant Professor, Department of Computer Science and Information Technology, Chalapathi Institute of Engineering and Technology, India. Mail: bhargavchalapathi@gmail.com

[3] Associate Professor, Department of Computer Science and Engineering, Chalapathi Institute of Engineering and Technology, India. Rokeshy12@gmail.com.

[4] Assistant Professor, Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, India. Mail: naveenpattipati@gmail.com

[4] Assistant Professor, Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, India. Mail: lavanya.rvr517@gmail.com,

[4] Assistant Professor, Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, India. Mail: aleem1234@gmail.com

[4] Assistant Professor, Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, India. Mail: gajulasushmitha5009@gmail.com,

[4] Assistant Professor, Department of Computer Science and Engineering, Joginpally B.R. Engineering College, Hyderabad, India. Mail: nagarajarao521@gmail.com.

OptIPuter is a term that is an abbreviation for Optical networking, Internet Protocol, and computer technologies for scientific visualization, which is an advanced distributed cyber infrastructure meant to handle data-intensive applications and improve cooperation through dedicated light channels. The DYNES (Dynamic Network Systems) project is aimed at providing advanced cyber infrastructure for data-intensive scientific communities, such as high-energy physics and astronomy, through dynamic circuit provisioning services [2]. The requirement of fast and reliable transmission of significant scientific datasets makes it mandatory that worldwide scientific cooperation employ high-bandwidth networks rather than the conventional internet services used for corporate purposes. To meet these needs, there are several R&E Network providers around the world that have developed high-performance networks for scientific applications. Among them are Internet2, ESnet (Energy Sciences Network), GEANT (pan-European data network), and KREONET (Korea Research Environment Open NETwork). In 2010, ESnet designed the Science DMZ, a network architecture specially built to support high-performance research applications. This conceptual design includes system optimization, specialized transmission servers, and software methodologies for the efficient transmission, sharing, and storage of huge scientific information. The Science DMZ's essential element is the Data Transfer Node (DTN), a high-performance Linux server customized with a specific system kernel and an enhanced Transmission Control Protocol (TCP) stack for managing high-volume data transfer services. DTNs improve the transmission performance by using efficient parallel data transfer protocols such as GridFTP and BCP. Similar to scientific applications, cloud applications require fast data transmission under urgent conditions. Cloud VM images have to be transferred promptly to other cloud domains during disaster recovery or mitigation. Acknowledging the capability of DTNs to expedite VM migration in cloud settings, we offer a technique for swiftly moving cloud VM images across cloud service providers via dedicated DTNs. This solution encompasses local and distant copy procedures, as well as a DTN-to-DTN transfer process, all automatically coordinated via the fork system call, thereby obviating the need for further administrative involvement. We focus on improving the process of creating a local copy between a cloud controller and DTNs by optimizing read/write operations through improved mount methods inside the NFS protocol. From our experimental measurements, the highest throughput in write operations is achieved when both the NFS server and client are configured in asynchronous mode, with the cloud controller running the process of the NFS client and the DTN running the process of the server.

2. RELATED WORK

The Science DMZ concept became the foundation for a number of innovative initiatives to upgrade the quality of the research environment. For instance, the USP (University of Sao Paulo, Brazil) Science DMZ, developed by Pho et al., [7] has been aimed at sharing genomic data securely and efficiently over GridFTP [10]. They evaluated the performance, on average, of the different types of clouds and the supercomputing sites relative to one another with and without SDN enabled. On the same lines, the medical science DMZ [8] emphasizes data security with facilitation of large transfers of data over R&E networks for affiliated research entities [11]. Science Pass is another; it enhances data transfer performance for high-volume flows by safely transferring data over a 100 Gbps SDN environment and avoids traditional firewalls [9]. To address the security issues in the previous Science DMZ paradigm, CoordiNetZ was developed [12]. This system includes SciMon, SciFlow, Controller, and Coordinator; this all relies on router ACLs or Access Control Lists, thus giving it a basis of coarse-grained flow access control [13]. Host DTN context -inflected granular flow control along with a security policy framework is another enhanced security technique that the technology features. For instance, an advanced networking system, using an dedicated optical network to accumulate cyber infrastructure resources, therefore enabling its members to have a smooth share of virtualized resources, is OFFN: One Oklahoma Friction Free Network [14]. The authors successfully addressed the complexity in administering conventional VLAN in conventional networks through developing an application for VLAN in SDN by Nguyen and Kim [15]. This application utilizes important SDN capabilities that include programmable flow management and centralized controller architecture to support dynamic VLAN setup as well as simple configuration troubleshooting [17]. Our method puts more emphasis on using DTNs as performance accelerators to improve data transfer during virtual machine migration in cloud environments, in contrast to the previous approaches that center on software-defined networking, secure transfer, and institutional reference architectures within the Science DMZ framework [18]. Live VM migration allows for the transfer of operating virtual machines without turning them off, enabling smooth transfers between cloud service providers, as opposed to cold VM migration, which involves turning VMs down before relocation [19]. Voorsluys et al. [20] analyzed the performance impact of live VM migration by using Xen virtual machines and Web 2.0 apps in their study, focusing on SLA (Service Level Agreement) violations. A knowledge-based adaptive scheduler [21] was proposed as a solution to meet customer requirements and maximize provider profit in SaaS (Software as a Service) providers' SLA endeavors. This scheduler, which includes admission control, knowledge processing, and scheduling, is demonstrated to enhance total profit and average reaction time in simulations. Rathod and Reddy [22] proposed a decentralized virtual migration system with a dynamic virtual machine selection algorithm that optimizes VM allocation based on user workload. By splitting a virtual machine image into a user data block and a composable block with modular components, the CIC (Composed Image Cloning) approach attempts to reduce network bandwidth usage during virtual machine migration in federated cloud settings [24]. While current research focuses on optimizing the performance of live VM migration and SLA trade-offs, our approach optimizes the transfer throughput of cold VM migration using DTNs [25]. Following in the footsteps of Phoebus [20] and I-TCP (Indirect-TCP) [21], we use a store-and-forward mechanism to transmit huge datasets effectively. The WAN accelerator Phoebus quickly forwards data by dividing end-to-end connections to improve transmission speed after intercepting it, storing it in intermediate nodes, and then sending it on its way. [26]I-TCP partitions a TCP connection between wireless APs and mobile devices in order to increase the throughput of wireless networks. The major difference between our technology and those listed above is that we make use of dedicated DTNs in order to improve data transfer performance at the application protocol level, namely with the FTP and NFS protocols. As a result, TCP connection management, traffic monitoring, and interception are no longer required.

3. METHODOLOGY/ PROPOSED SOLUTION

A DTN offers data transfer services with large internal storages managed by a controller, like RAID (Redundant Array of Independent Disks) controller. On the other hand, it could connect to external storages using high-speed file systems like Lustre, GPFS (General Parallel File System), or NFS (Network File System) [27]. In many cases, DTNs connect to external file systems using high-speed networking technologies such as Gigabit Ethernet, Fiber Channel, or InfiniBand. Decentralized and loosely integrated DTNs and external storage systems have improved the flexibility of service design. Under the above two circumstances, we recommend using DTNs in cloud settings to enhance the efficiency of data transmission between cloud controllers and DTNs.

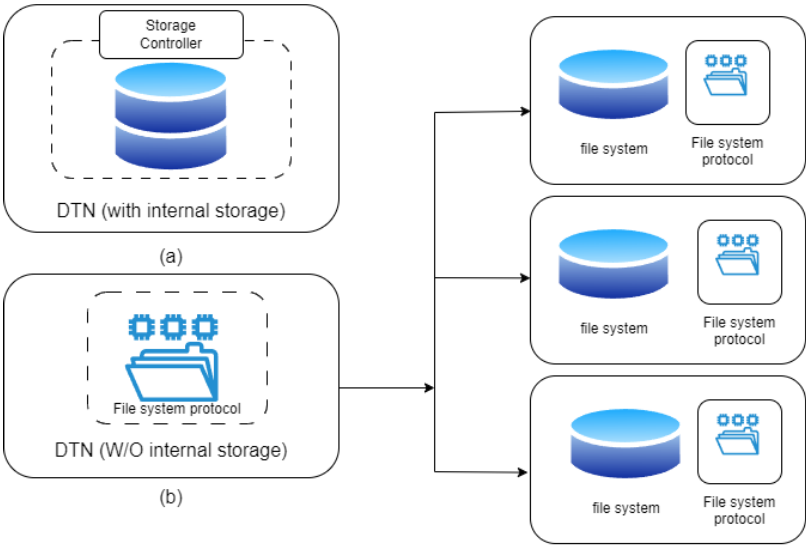

Figure 1. (a) A Data Transfer Node (DTN) with internal storage; (b) A DTN with external storage.

DTNs are mainly used in scientific applications for the quick transfer of large data quantities through high-performance networks [28]. This approach can be utilized to connect cloud service providers. For instance, when two cloud providers need to exchange VM images, they could create connections between their respective cloud controllers using DTNs as in Figure 2. From the two standard DTN deployment scenarios, we narrow down to the external storage use case and propose a means for connecting a DTN to a cloud controller over NFS. In this configuration, if a DTN is attached to an external storage server, then the storage system tends to act as the mount server for the file systems. From the perspective of the external storage, a cloud controller might be considered a sort of external storage system. A cloud controller typically manages a VM image repository, and hence is similar to Glance service. We would implement a process for NFS on DTN and then an NFS client process to run on the cloud controller. There are many benefits in this method. [30] Primarily, it eases the pressure of the cloud controller inside the cloud environment. While the services within the cloud could consist of a variety of microservices that are spread out across different physical nodes, all of them run on the cloud controller. Moreover, when the additional controllers intend to join the cloud, the existence of the NFS server process on a DTN makes it easy for the majority of the nodes that act as cloud controllers to configure and run the NFS client functions rather than server functions. The fundamental methods of our proposal are introduced. A cloud service provider compresses virtual machine images into a tarball file. An NFS service is created between cloud controllers and DTNs, which defines the role of the NFS server and client. NFS performance has to be tweaked and optimized to enhance overall efficiency. Then, a local copy procedure sends the tarred VM images from the cloud controller to a DTN through the network service provider. When the local copy procedure is completed, the massive file is transferred between DTNs through a long-distance network.

The target cloud service provider does a remote copy and restores the tarball file into uncompressed virtual machine files in reverse sequence. To improve the performance between the NFS server and client, several NFS tuning settings may be evaluated. Given that communication between a cloud controller and a DTN has to be smooth, we might focus on specific critical performance tuning criteria. To achieve performance improvement, we review mount options like network buffer sizes and synchronous/asynchronous settings in both the NFS client and the server. Optimizing read/write buffer size parameters rsize, wsize according to Linux kernels and hardware requirements may improve the performance of a cloud controller for data transfer in DTNs. Besides, the read/write strategies for a DTN need to balance dependability with throughput while choosing between synchronous and asynchronous options. In addition to the local copy between a cloud controller and DTNs, a DTN-to-DTN transfer mechanism using a data transfer protocol such as GridFTP and a remote copy are necessary for ensuring flawless end-to-end transmission between cloud controllers across multiple domains. The local copy process must be finalized before to the commencement of the DTN-to-DTN transfer, and the remote copy process should start subsequent to the completion of the DTN-to-DTN transfer. In light of this need, we suggest a way to synchronize the three processes—a local copy process, a DTN-to-DTN transfer process, and a distant copy process—through the use of a fork system call. Figure 1: proposed procedure used consists of a primary child process to control the local copy with NFS protocol, the second one executing the transfer using data transfer protocol of the DTN-to-DTN transfer, and finally the parent process that manages the remote copy. With such procedures, it can assure uniqueness among the three operations conducted to achieve complete transfer of tarred VM image files.

4. EXPERIMENTS AND ANALYSIS

We discuss here the results of our NFS mount and tuning experiments aimed at optimizing the transfer of a local copy to a DTN. To this end, we conducted a number of studies of optimal read/write buffer sizes and the best combination of synchronous and asynchronous modes for both NFS server and client. For the experiment setup, we used the OpenStack Mitaka version published in April 2016 and NFS version 4. The OpenStack controller server was equipped with four Intel Xeon E5620 processors operating at 2.40 GHz, 54 GB of RAM, and a 500 GB hard drive. The DTN version with FIONA, Flash I/O Network Appliance, had eight Intel Xeon E5-2630 cores running at 2.40 GHz, 32 GB of RAM, and a storage configuration with a 4TB SSD and a 32TB SATA disk drive. Its interconnection speed was 1 Gbps. We implemented our measurements of the read/write bandwidth over an NFS connection with the dd command, the Linux program that copies and transforms files based on given operands [33]. A useful feature of this program is that it can promptly create binary image files with predefined sizes. To benchmark write buffer size performance, we added the operands "if=/dev/zero" and "of=/home/nfsClient/testfile". For reads, "of=/home/nfsClient/testfile" and "if=/dev/null". The dd command allows many configuration options, including bs (block size) and count (the number of input blocks), thus making it easy to create diverse experimental situations. A trade-off often exists between dependability and performance when using synchronous vs asynchronous modes in NFS servers and clients [34]. In our experiments, the DTN operated the NFS server while the cloud controller node executed the NFS client processes to provide a local copy function via a high-performance NFS mount. We expected to experience performance variability depending upon the operational mode combinations in the NFS server and client operations. Consequently, we performed several tests across three scenarios to check the impact of various synchronous and asynchronous modes in the NFS server/client, along with different read/write buffer sizes. We considered three scenarios in Table 1: (i) NFS server, synchronous/client, asynchronous, (ii) NFS server, asynchronous/client, asynchronous, and (iii) NFS server, synchronous/client, synchronous. We employed a file produced by the dd command, which is about 256 MB in size, rather than an actual compressed VM image file, to support reading and writing on a destination node[36]. All experiments were performed using read/write buffer sizes from 1 Kb to 1024 Kb, and all phases for each buffer size were repeated three times. We then calculated the average throughput for each dimension of read/write buffer.

Table 1. Experiment cases.

Case	Controller (NFS Client)	DTN (NFS Server)
Case I	asynchronous	synchronous
Case II	asynchronous	asynchronous
Case III	synchronous	synchronous

The average of 118 Mbps was constant throughout all scenarios, with mean throughput of read operations. This means that in our specific configuration, the NFS server's and client's synchronous and asynchronous modes interaction along with the change in read buffer sizes has had little or no impact. Thus, the respective graph is not provided here [37]. Figures 6 and 7 illustrate the fact that average throughput of the write operations differs under various conditions significantly. Initially, it becomes stable at approximately 87 MB/s for an 8 Kb write buffer when both the NFS server is configured synchronous and the client is set asynchronous. Thereafter with the same size of buffer it hits about 106 MB/s when both NFS server as well as the client operates asynchronously. Finally, with a 64 MB write buffer, the throughput peaked at about 85 MB/s in an S-shaped curve when the NFS server and client were synchronous.

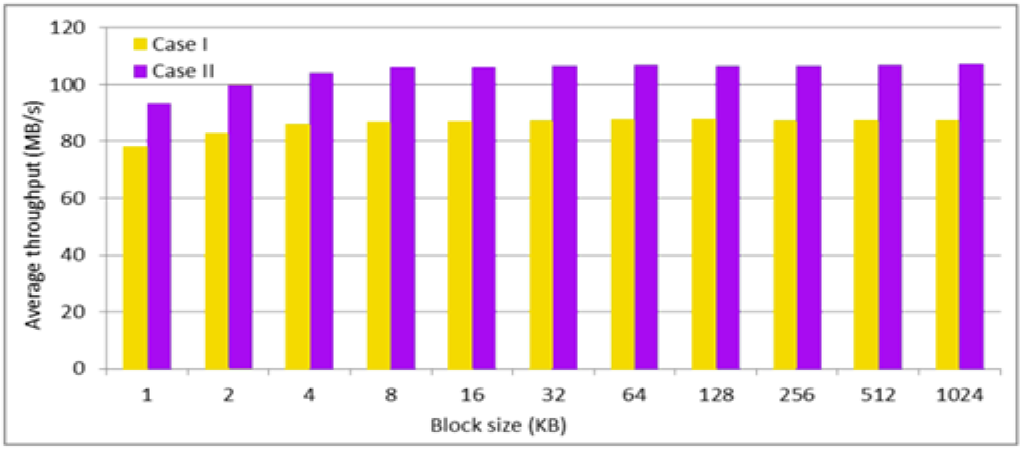

Figure 2. Throughput of the write operation in Cases I and II.

Finally, in this case, the throughput for a 1024 Kb write buffer was about 48 MB/s, which has not yet stabilized. We thus ran some more tests with a write buffer size of about 64 Mb. Our testbed environment determined when the NFS server and client were asynchronous, that the best performance was provided by an 8 Kb write buffer. It improved the performance of the write operation by about 22% compared to the asynchronous client and synchronous server configurations.

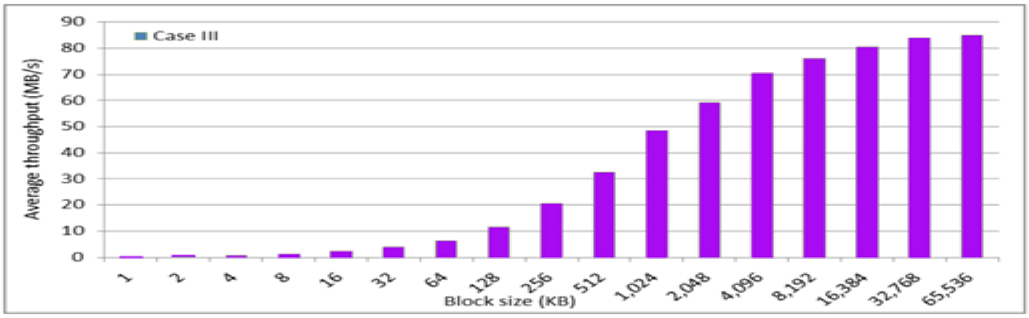

Figure 3. Throughput of the write operation in Case III.

CONCLUSIONS

This research reveals a procedure to accelerate image transfer of cloud virtual machine images over multiple DTNs among numerous cloud providers. The conceived process consists of the replication either locally or remotely, along with transmission over DTNs. To ensure all various processes involved in transferring photographs to their destination are kept separately, we used an innovative algorithm. We increased the transfer rate of pictures between the cloud controller and DTNs by using NFS mount and implementing several tweaks to the operational processes. We ran an experiment to test the performance of different configurations for the NFS server and client. Using asynchronous methods for both resulted in optimal performance, with a throughput of approximately 106 MB/s during write operations. This was an improvement of 22% over using synchronization for the server. Our technique enables data of utmost importance to move rapidly in cloud systems having high connectivity between cloud providers and network providers. Going forward, we would like to carry out further experiments involving different virtual machine images to test the effectiveness of our technology in real-time.

REFERENCE:

[1] HN Sultan (2018) Cloud computing for education: A new dawn? International Journal of Information Management 38: 61-72.

[2] T Ercan (2020) Effective use of cloud computing in educational institutions: Comparison of private and public cloud computing. Procedia Computer Science 158: 72-76.

[3] PK Senyo, E Addae, R Boateng (2018) Cloud computing research: A review of research themes, frameworks, methods, and future research directions. International Journal of Information Management 38: 128-139.

[4] RF El-Gazzar (2018) A literature review on cloud computing adoption issues in Creating Value for All Through IT. Proceedings of the 27th International Conference on Information Systems (ECIS) Portsmouth, UK 219-230.

[5] V Chang, RJ Walters, GB Wills (2019) Organisational sustainability modelling: An emerging service and analytics model for evaluating cloud computing adoption with two case International Journal of Information Management 35: 332-341.

[6] Khajeh-Hosseini, D Greenwood, I Sommerville (2019) Cloud migration: A case study of migrating an enterprise IT system to Proceedings of the IEEE 3rd International Conference on Cloud Computing Miami, USA 450-457.

[7] S Marston, Z Li, S Bandyopadhyay, J Zhang, A Ghalsasi (2019) Cloud computing—The business Decision Support Systems 51: 176-189.

[8] Sriram, A Khajeh-Hosseini (2019) Research agenda in cloud International Journal of Cloud Computing and Services Science 2: 1-18.

[9] N Sultan (2020) Reflective thoughts on the potential and challenges of wearable technology for healthcare provision and medical International Journal of Information Management 35: 627-631.

[10] M Armbrust, A Fox, R Griffith, AD Joseph, RH Katz, et (2018) A view of cloud computing. Communications of the ACM 53: 50-58.

[11] T Dillon, C Wu, E Chang (2019) Cloud computing: Issues and Journal of Internet Services and Applications 1: 7-18.

[12] DC Chou (2020) Cloud computing: A value creation Computer Standards & Interfaces 38: 72-77.

[13] E Brynjolfsson, A McAfee (2019) The business of artificial intelligence. Harvard Business Review 97: 56-64.

[14] TH Davenport, R Ronanki (2018) Artificial intelligence for the real world. Harvard Business Review 96: 108-116.

[15] Smarr, L.; Brown, M.; Laat, C. Special section: OptIPlanet—The OptIPuter global collaboratory. Future Gener.Comput. Syst. 2009, 25, 109–113.

[16] Zurawski, J.; Boyd, E.; Lehman, T.; McKee, S.; Mughal, A.; Newman, H.; Sheldon, P.; Wolff, S.; Yang, X. Scientific data movement enabled by the DYNES instrument. In Proceedings of the First International Workshop on Network-Aware Data Management, Seattle, WA, USA, 14 November 2011.

[17] Dart, E.; Rotman, L.; Tierney, B.; Hester, M.; Zurawski, J. The Science DMZ: A Network Design Pattern for Data-Intensive Science. In Proceedings of the SC'13 International Conference on High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 17–21 November 2013.

[18] Mansouri, Y.; Toosi, A.; Buyya, R. Data Storage Management in Cloud Environments: Taxonomy, Survey, and Future Directions. ACM Comput. Surv. 2017, 50, 91.

[19] Kokkinos, P.; Kalogeras, D.; Levin, A.; Varvarigos, E. Survey: Live Migration and Disaster Recovery over Long-Distance Networks. ACM Comput. Surv. 2016, 49, 26.

[20] Pho, N.; Magri, D.; Redigolo, F.; Kim, B.; Feeney, T.; Morgan, H.; Patel, C.; Botka, C.; Carvalho, T. Data Transfer in a Science DMZ using SDN with Applications for Precision Medicine in Cloud and High-performance Computing. In Proceedings of the SC'15 International Conference on High Performance Computing, Networking, Storage and Analysis, Austin, TX, USA, 15–20 November 2015.

[21] Peisert, S.; Bamett, W.; Dart, E.; Cuff, J.; Phil, D.; Grossman, R.; Balas, E.; Berman, A.; Shankar, A.; Tiemey, B. The Medical Science DMZ. J. Am. Med. Inform. Assoc. 2016, 23, 1199–1201.

[22] Balas, E.; Ragusa, A. SciPass: A 100 Gbps capable secure Science DMZ using OpenFlow and Bro. In Proceedings of the SC'14 International Conference on High Performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA, 16–21 November 2014.

[23] Nagendra, V.; Yegneswaran, V.; Porras, P. Securing Ultra-High-Bandwidth Science DMZ Networks with Coordinated Situational Awareness. In Proceedings of the 16th ACM Workshop on Hot Topics in Networks, Palo Alto, CA, USA, 30 November–1 December 2017.

[24] Neeman, H.; Akin, D.; Alexander, J.; Brunson, D.; Calhoun, S.; Deaton, J.; Fotou, F.; George, B.; Gentis, D.; Gray, Z.; et al. The OneOklahoma Friction Free Network: Towards a Multi-Institutional Science DMZ in an EPSCoR State. In Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, Atlanta, GA, USA, 13–18 July 2014.

[25] Nguyen, V.; Kim, Y. SDN-based Enterprise and Campus Networks: A Case of VLAN Management. J. Inf. Process. Syst. 2016, 12, 511–524.

[26] Kaur, P.; Rani, A. Virtual Machine Migration in Cloud Computing. J. Grid Distr. Comput. 2015, 8, 337–342.

[27] Voorsluys, W.; Broberg, J.; Venugopal, S.; Buyya, R. Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. In Proceedings of the 1st International Conference on Cloud Computing, Beijing, China, 1–4 December 2009.

[28] Motavaselalhagh, F.; Esfahani, F.; Arabnia, H. Knowledge-based adaptable scheduler for SaaS providers in cloud computing. Hum.-Centric Comput. Inf. Sci. 2015, 5, 16.

[29] CloudSim. Available online: https://github.com/Cloudslab/cloudsim (accessed on 27 March 2018).

[30] Calheiros, R.; Ranjan, R.; Beloglazov, A.; Rose, C.; Buyya, R. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. Softw.Pract. Exper. 2011, 41, 23–50.

[31] Rathod, S.; Reddy, V. NDynamic Framework for Secure VM Migration over Cloud Computing. J. Inf. Process. Syst. 2017, 13, 476–490.

[32] Celesti, A.; Tusa, F.; Villari, M.; Puliafito, A. Improving virtual machine migration in federated cloud environments. In Proceedings of the 2010 Second International Conference on Evolving Internet, Valencia, Spain, 20–25 September 2010.

[33] Kissel, E.; Swany, M.; Brown, A. Phoebus: A system for high throughput data movement. J. Parallel Distrib. Comput. 2011, 71, 266–279.

[34] Bakre, A.; Badrinath, B. I-TCP: Indirect TCP for mobile hosts. In Proceedings of the 15th International Conference on Distributed Computing Systems, Vancouver, BC, Canada, 30 May–2 June 1995.

[35] NFS. Available online: http://nfs.sourceforge.net/ (accessed on 27 March 2018).

[36] OpenStack. Available online: http://www.openstack.org/ (accessed on 27 March 2018).

[37] Dd Tool. Available online: https://en.wikipedia.org/wiki/Dd(Unix) (accessed on 27 March 2018).