Proceedings of the International Conference on Computer and Information Science and Technology
Ottawa, Ontario, Canada, May 7-8 2015
Paper No. 125

Dynamic Software Defined Network Provisioning for Resilient Cloud Service Provider Optical Networks

Casimer DeCusatis
Marist College, Department of Computer Science and Mathematics
3993 North Road, Poughkeepsie, NY, USA
casimer.decusatis@marist.edu

Aparicio Carranza and Carolyn Sher-DeCusatis
New York City College of Technology
300 Jay Street, Brooklyn, NY, USA
acarranza@citytech.cuny.edu , decusatis@optonline.net

Abstract - Cloud computing service providers (CSPs) face many challenges related to rapid, elastic provisioning and capacity planning for high resilience networks within and between data centers. The design space for cloud network services may be partitioned according to the required data semantics and trade-offs between consistency, availability, and partition tolerance. We investigate high availability solutions for resilient CSP networks. In particular, live migration of virtual servers may require dynamic provisioning of optical networks over 100 km in length. We demonstrate an experimental proof of concept automated test bed that reduces re-provisioning times from several days to under a minute. A simple analytic model for application migration time as a function of bandwidth is proposed and validated. We also demonstrate the application of fast dynamic optical network re-provisioning to mitigating the effects of traffic bursts in quasi-synchronous storage applications.

Keywords**:** Cloud, OpenFlow, Open Daylight, SDN, Optical, Network

1. Introduction

With the growing importance of data networks as critical infrastructure for cloud computing service providers (CSPs), the need for higher availability and network resilience has become increasingly apparent. Resilience has traditionally been defined as the data network's ability to provide and maintain an acceptable level of service in the presence of both natural and man-made disasters. This includes the network's ability to dynamically reconfigure in response to changing workloads, for example facilitating the mobility of virtual machines (VMs) and workload containers or accessing storage in the cloud as if it were attached within a local data center. These capabilities have been significantly expanded with the introduction of software defined networking (SDN). Some of the distinguishing features of an SDN network include separation of the data and management/control planes; a shift towards centralized management; and abstraction of key network attributes through an application programming interface (API), which enables the creation of virtual, automated network flows (DeCusatis et.al. 2013). Thus, network resilience now includes concepts such as dynamic, application aware topologies and programmable, automated network reconfiguration.

There have been prior efforts to control network traffic using software (particularly for telecommunication networks using multi-protocol label switching (MPLS)), but these approaches have not been broadly adopted for IP data networks. Recent industry trends such as dynamic workloads, secure multi-tenant cloud computing, wireless mobile systems, and big data analytics have led to a need for higher resilience in data networks. The abstractions provided by SDN can potentially reduce management complexity and standardize traffic routing mechanisms, which contribute to better network reliability, availability, and serviceability. Further, introducing a centralized SDN network controller enables end-to-

end network visibility and faster reconfiguration in response to disasters. For example, a fully automated system could be programmed to invoke disaster recovery protocols without the presence of a human administrator. SDN can reduce network recovery times, and make it easier to schedule and automate periodic testing of a business continuity plan. Prior to the introduction of SDN, re-provisioning a data center network could take days or weeks, while the network between multiple data centers might require weeks or months (Manville 2013). SDN makes it possible to build the physical network infrastructure once, then reconfigure it using only software. The resulting in location agnostic networks reduce reconfiguration time to a few minutes, the same order of magnitude as provisioning new virtual machines (VMs) on a server.

In this paper, we examine the impact of SDN on data network interconnecting cloud data centers for applications requiring high availability. Experimental results are presented from a test bed implementing live application migration over an extended distance SDN network, including a simple model to predict VM migration times. The use of SDN to implement elastic bandwidth for cloud bursting will be discussed.

2. SDN and the BASE-ACID Methodology

The design space for cloud network services may be partitioned according to the data semantics required by each service (Gray and Reuter, 1993). The trade-offs between consistency, availability, and partition tolerance for networks may be described by the CAP theorem (Panda et.al 2013). High availability CSP data center designs are focused on data consistency; the more lenient an application's data consistency (i.e. different end users can be returned different versions of the same data), the easier it is to implement a network with the expected availability. At one extreme, representing the minimal acceptable requirements for data consistency, is an approach called BASE (Basically Available, Soft State, Eventual Consistency) (Gray and Reuter, 1993). At the other extreme is a much stricter set of requirements for reliable processing of data transactions, which has become known by the acronym ACID (Atomicity, Consistency, Isolation, Durability) (DeCusatis 2013a). Many practical systems employ a hybrid approach; for example, some cloud service directories (Bohrer 2013) maintain a database using BASE semantics, but keep user customization profiles in an ACID database. SDN networks can provide features which are considered valuable regardless of whether an ACID or BASE taxonomy is employed.

For example, cloud services using the ACID methodology require continuous network availability; it is preferable for an ACID service to be unavailable than to function in a way that relaxes the ACID constraints. To guarantee this, a network supporting ACID taxonomies might include redundant multi-pathing with fast failover. Rather than relying on conventional TCP/IP and Ethernet principles such as LACP and dual homing, an SDN network controller can be programmed in advance to switch between redundant paths based on application requirements. For some large networks, SDN failover can be as much as 10 times faster than conventional Ethernet (DeCusatis 2013), which is a significant benefit for continuously available ACID applications. Further, the ACID methodology also requires continuous application availability in the event of system failures, server over-utilization, or network congestion. This can be provided via live VM mobility across a dynamically provisioned SDN network. We have developed an approach for rapid recovery and liver VM migration across optical metropolitan and wide area networks (MAN/WAN) between multiple cloud data centers. This implementation includes enabling optical wavelength division multiplexing (WDM) equipment to use SDN flow control, and the capability to slice a physical CSP network into multiple virtual sub-networks, as we will discuss further in section 3.

For other cloud services, however, the primary value to the end user is not necessarily atomic operations or strong data consistency, but rather resilience and high availability of data. SDN is also useful for these so-called BASE applications. For example, many enterprise data centers are finding it more cost effective to place storage in the cloud, rather than investing in storage devices for their own data centers. Conventional storage networks accessing cloud environments are statically provisioned, based on estimates of peak bandwidth requirements rather than actual bandwidth consumption. Traditional networks must be over-provisioned in this way because, as previously noted, it takes days or weeks to manually re-provision these connections, making real-time response impractical. This is highly inefficient for a CSP, since most

of the allocated network bandwidth is not used most of the time. SDN networks can potentially avoid this issue by automating optical network provisioning. For many such applications, BASE taxonomies are sufficient (i.e., stateful SDN controller failover with eventual consistency of the network state). SDN makes different trade-offs between consistency, availability, and partition tolerance than traditional Ethernet, though both are still compliant with Brewer's Theorem as described in Gilbert and Lynch (2002).

Further, SDN optical networks may offer new functionality which is not available with conventional optical networks between cloud data centers. Many BASE applications, such as quasi-synchronous storage backup and recovery to the cloud, can experience traffic bursts which exceed the pre-provisioned static network bandwidth. Under these conditions, known as cloud bursting, the storage system can repeatedly fail to complete successive access requests, causing degraded performance and eventually halting the application altogether. A typical example of an enterprise data center accessing a public cloud storage service is shown in figure 1, which depicts synchronous storage traffic (Mbits/) monitored every 15 minutes over a seven day period (names of the enterprise and cloud provider have been removed at their request). Multiple traffic bursts are clearly visible, occurring several times per day, the largest of which exceeds typical bandwidth usage by over six times and lasts between 15-30 minutes. For comparison, static bandwidth provisioning levels are shown; even if these levels increase by 45% year to year, it is still not possible to accommodate the largest traffic bursts. In addition, higher over-provisioning becomes increasingly inefficient in terms of both cost and network resources. Existing networks require days or even weeks to re-provision bandwidth, far too long for management of these traffic bursts (Manville, 2013). While this problem cannot be addressed using static network provisioning, we have developed an SDN optical network capable of monitoring application performance and dynamically provisioning incremental bandwidth for short traffic bursts.

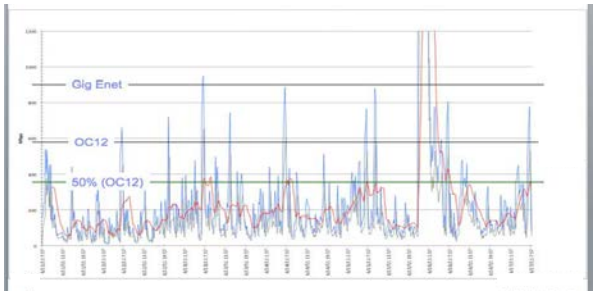


Figure 1 - Network bandwidth vs time, cloud synchronous storage

3.0 Experimental Optical SDN Test Bed

Our SDN optical network test bed is shown in figure 2, consisting of a 125 km single-mode fiber ring interconnecting three metropolitan area data centers. One data center represents the CSP, while the others are enterprise data centers sharing multi-tenancy in the cloud. The sites are interconnected with a dense wavelength division multiplexing (WDM) platform (Adva FSP3000), including discretionary wavelength pools which can be applied to the enterprise data center connections. Each site also contains inexpensive demarcation point hardware integrated with the WDM platform (Adva XG210) which serves as a traffic monitor or traffic injection source for test purposes. Data center iSCSI storage resources are connected via 1/10 Gbit/s Ethernet switches (Lenovo G8264 or Plexxi). Servers at each location host virtual machines (either KVM with OpenStack Icehouse release or VMWare/VSphere 5.0), which contain software defined network (SDN) controllers for the Ethernet switches and a network hypervisor for the WDM optical equipment. Optionally, application level orchestration is provided by IBM Cloud Orchestrator 2.4.

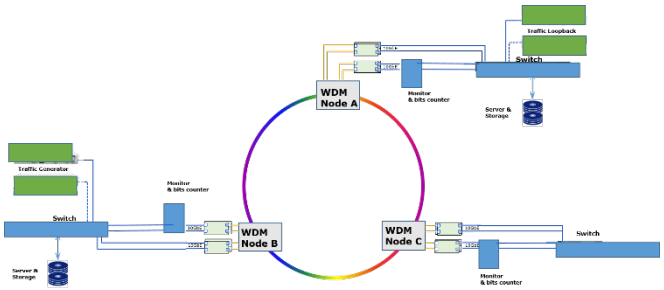


Figure 2 – Optical WDM test bed using SDN and network hypervisor control

We have created open source software which performs dynamic optical bandwidth provisioning (using OpenFlow 1.3.1) by interfacing with the WDM network hypervisor. We have also created software which allows the optical equipment to be managed and automated from an Open Daylight SDN controller. The CSP uses our code and the WDM network hypervisor to slice their physical network into two or more virtual network segments, each of which can be assigned to a different tenant data center. Each tenant is then able to use an open standards-based SDN controller of their choice to optimize bandwidth allocation within their slice of the network; both controllers support the BASE taxonomy. The XG210 hardware used for real time traffic monitoring is managed from the WDM platform network hypervisor using SNMP 3.0. When the traffic monitor exceeds pre-programmed thresholds, it triggers automated re-provisioning of Layer 0-3 optical connections to temporarily provide increased bandwidth for the traffic burst. We have experimentally achieved end-to-end re-provisioning in under a minute, more than fast enough for responding to measured traffic bursts in the synchronous storage application of figure 1. Once traffic returns to nominal levels, both the networks within and between data centers are restored to their previous configurations automatically.

A similar approach can be used to trigger automated VM or virtual container migration. The trigger event can either be generated by the XG210 or by monitoring other properties in the cloud network. For example, we have demonstrated using the open source application Ganglia to monitor events on the server, which hosts a VM running either an ACID or BASE application. Ganglia can monitor events such as server utilization or available memory. Server utilization may increase, for example, when too many virtual appliances are running on one physical server; this can affect performance of all the applications. When a pre-set threshold on server utilization is exceeded, Ganglia triggers an action such as migrating a VM to another server with available capacity. We can optionally decide where to migrate the application based on network analytics, such as models for predicting the VM migration time. If there is available discretionary bandwidth on the WAN, we may choose to dynamically re-provision additional wavelengths to speed up VM migration. A central SDN controller is then used to provision network traffic flows within the source and destination data centers, as well as between these data centers across the optical WAN. Once the end-to-end network path is provisioned, a migration algorithm such as VMware vSphere is used to live migrate the video streaming application across the WAN. After the migration successfully completes, the network may be programmed to automatically return to its original configuration.

We have demonstrated new functions such as automatically triggering live VM migration across an optical WAN in response to capacity monitoring on the attached servers, and orchestrating the end-to-end path for VM migration from a single network controller. In this case, so-called “live” VM migration (i.e. transferring active memory and execution state from a source to a destination with uninterrupted operation of the application) is triggered automatically when server utilization exceeds 75%. We have verified experimentally that it was possible to continuously “ping” the VM throughout the migration process, which has previously been established as an acceptable method for verifying live migration (Fox et.al. 2013). By automating this process and creating new software to provision the optical transport equipment from a centralized SDN controller, we have experimentally demonstrated that arbitrary reconfiguration of the WAN can be implemented in under 60 seconds. Dynamic provisioning on this timescale enables cloud bursting applications, which require large amounts of bandwidth for short periods of time. Although we

have automated the provisioning process, our management code also allows a system administrator to initiate the process with so-called "single click" provisioning from a mobile device such as a smart phone.

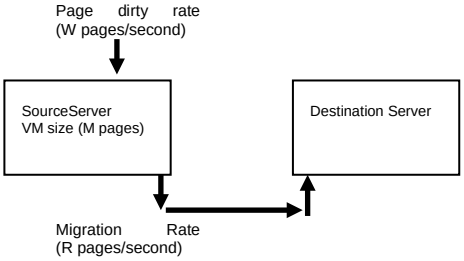
Similar to network resources within a data center, fixed wavelengths in most optical MANs are currently under-utilized and high cost, since they must be statically provisioned for estimates of peak network capacity. Our combination of virtual slicing, automated live VM migration, and automated cloud burst re-provisioning has the potential to significantly decrease solution costs. Our approach is compatible with existing colorless, directionless, agile core optical networks that have been enabled by component technologies such as wavelength tuneable optics, gridless reconfigurable optical add/drop multiplexers (ROADMs), and hybrid optical amplifiers (Manzalini 2013). In practice, reconfiguration of the physical WAN may require traffic re-engineering for these devices as well. While this is beyond the scope of our current work, a recently published approach (Birand 2013) proposed using distributed sensor arrays in the optical network to dynamically control optical power per wavelength.

4.0 Analytic Model for Virtual Machine Migration

High resiliency SDN optical networks can make use of simple analytic models to estimate the time and resources required to migrate a VM with a given amount of available bandwidth, and to determine whether adding more bandwidth would significantly affect migration time. There have been various proposals for modelling VM migration time (Akoush 2010, Liu 2013); for our purposes, we have developed a simple model as illustrated in figure 3. The source server has an initial cache of M memory pages, and the workload is writing to this memory (dirtying pages) at a rate of W pages per second (for many applications, it is reasonable to assume a constant page dirty rate (Akoush 2010)). At the same time, the VM migration process (for example, VMware vSphere) is migrating R pages per second to a destination VM. The time T, in seconds, required to migrate all M pages of memory is given by

$$T = M / (R-W) \quad (1)$$

As expected, we can improve migration time by using a faster network or an improved migration algorithm (affecting R in each case).



.

Figure 3 – Simplified VM migration model

By plotting eq. (1) as shown in figure 4, we see that for the trivial case where W=0, migration time is M/R. At the other extreme when W is much greater than R, the migration time is theoretically infinite (commercial migration algorithms enforce an upper bound on migration time, so this is never seen in practice).

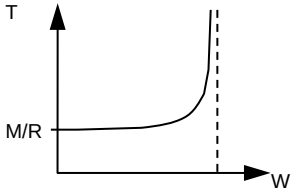


Figure 4 – simple model of VM migration time from eq. (1)

This model has implications for the comparison of two VM migration methods with different throughputs, as shown in figure 5, where method A has a throughput 3 times as large as method B. For small values of W (light workloads), our model predicts that the migration time for method A is 3 times larger than method B, as expected. However for a heavier workload (larger value of W), the migration time for method A can be much larger than 3 times the migration time of method B. For even heavier workloads, the migration may only be possible using method B, while for still heavier workloads neither method can successfully migrate the VM without stopping or throttling the workload.

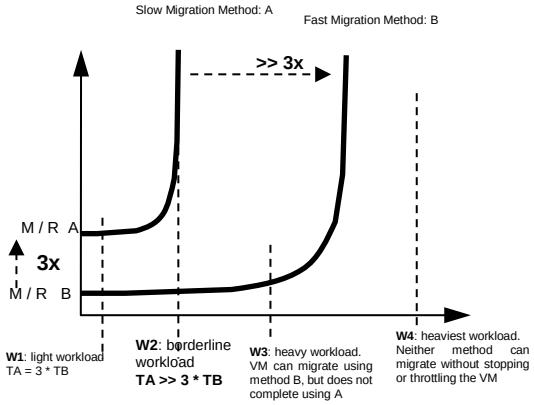


Figure 5 – Comparison of VM migration times for two different migration methods (A and B)

Since we are primarily interested in re-provisioning network bandwidth to improve migration time, we can also plot equation (1) with T and a function of R, as shown in figure 6 (this approach is only valid for R > W). As before, we observe a region of this curve where a relatively small change in R results in a significant reduction in T, for constant M and W.

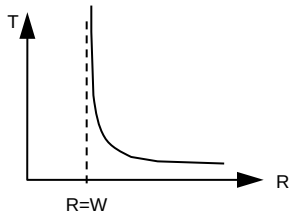


Figure 6 – Migration time as a function of network bandwidth

Many VM migration techniques pause the VM briefly during the migration process in order to improve performance. The simplest approach is *pure stop and copy*, which involves suspending the VM, then transferring the entire memory contents and architectural state (such as register contents) to another host, where the VM is re-instantiated. This approach suffers from poor performance and service disruptions (Liu 2013) due to the relatively long periods of downtime when the VM is stopped. Better performance can be achieved through *live migration* (transferring active memory and execution state of a VM from a source to a destination), which strives to minimize both VM downtime and total migration time. In a high availability environment, this allows seamless movement of applications and services without service disruption to the clients. One example of a live migration technique is an iterative convergence process called *precopy-based migration*, which transfers memory contents to the destination host while the VM continues to execute on the source host. This approach has been implemented in recent versions of VMware, KVM, and other hypervisors. Multiple copying steps are performed, and during each step the VM memory pages that were modified during the previous step are resent to the destination. While the

active VM memory is being transferred to its destination, the copy of the VM still executing will rewrite some of the transferred memory pages. These dirtied pages are tracked by the hypervisor memory management, and are resent to the destination in subsequent migration iterations. The iterative process continues until either a small working set size is reached, or until a predetermined iteration count is reached. At this point the VM is usually stopped briefly (the *stop-and-copy phase*) while the final portion of active memory and architectural state is transferred. We can re-write equation (1) to include the VM stop time, S (in seconds), for a precopy-based migration technique. The total migration time can now be expressed in the form

$$T = M / (R – W(T-S)/T) \quad (2)$$

Note that this solution is only valid for cases were W < R and M > W * S. The stop time moderates W because the workload cannot modify its pages while the VM is stopped; the effect is less if S << T.

For nonzero values of S, total migration time can be highly workload dependent. For small page dirty rates, migration time will still be close to the lower bound of M/R. However for large values of W, the number of pages transferred during the iterative copy phase may not keep up with the rate at which new pages are being dirtied, and the total migration time will be dominated by the stop time. Migration time can also be affected by other design choices; for example, using a network communication protocol such as InfiniBand which has lower overhead (i.e. reduces the number of acknowledgments associated with each packet transfer) will tend to increase R, and thus reduce migration times compared with using a communication protocol with higher overhead. There are conditions under which total migration time is a highly nonlinear function of R and W; operating in these regions means that the exact migration time is not straightforward to predict. We can use models like this to estimate total migration time for a given set of conditions, and determine whether it would make a significant difference to request additional network resources (for example, increasing R by adding more bandwidth on demand across the WAN). If we are moving bandwidth from another discretionary workload with a lower quality of service to improve VM migration time on a workload with a higher quality of service, we can estimate the impact on both workloads using this approach. We note that in some cases live migration degrades performance of all other VMs running on the same host (a condition sometimes called *brownout*). This is yet another reason to minimize total VM migration time. Various trade-offs are possible; for example, the VM migration application may attempt to minimize VM stop time at the expense of slightly longer total migration times. If the VM stop time is too low, the migration may not complete; while if the stop time is too large, network and storage connections may time out or other system interruptions may occur. At some point, the benefits gained from agile resource management may diminish as the resource management overhead becomes larger. While the details of the migration methods used in some commercial products are not always readily available, in our test bed, we have evaluated workloads with M = 1028 MB with 4 KB per memory page, S = 10-100 s, and R = 100 – 1000 Mbps; total live migration times on the order of 1-2 minutes are achievable under these conditions. These results are consistent with previous models (Akoush et.al. 2010, Lui et.al. 2013).

We have simulated a model for precopy-based migration, incorporating the VM stop time, as shown in figure 7, for different values of S and R based on equation (2) and following the methodology of Isci et.al (2011). We note the significant nonlinear regions in this result, where a small change in the inputs can cause a relative large change in the output. For some workloads, service disruptions may occur if the number of pages transferred during the iterative copy phase doesn't keep up with the rate at which new pages are being dirtied. In this case, total VM migration time will be dominated by the stop time, especially when there is a slow network connection. We can improve migration time in several ways, for example by using a faster network or an improved migration algorithm (affecting R in each case).

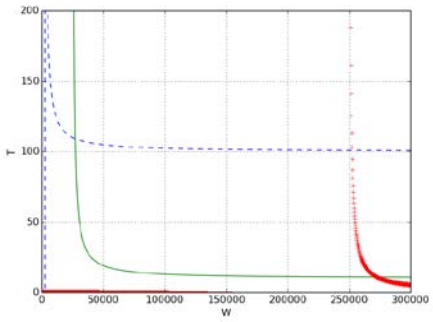


Figure 7 – Analytic model of migration time, T, vs page dirty rate, W, from equation (2); assuming M = 1028 MB with 4 KB per memory page; dashed line shows S = 100 s, R = 100 Mbps; solid line shows S=10 s and R=10 Mbps; and plus signs show S=1s, R=10 Gbps.

4. Conclusion

We have demonstrated automated, dynamic provisioning of high resiliency CSP networks using SDN. There are benefits for both ACID and BASE taxonomies. We have significantly improved optical WAN provisioning time, enabling the optical network to respond in under a minute to traffic bursts such as those experienced in quasi-synchronous storage backup applications. Using dynamic bandwidth allocation, we can also improve VM live migration times, thereby enhancing application availability. We have also developed a simple model to help predict VM migration time as a function of workload characteristics, and validated the simulation results. Our approach is based on the OpenStack Neutron interface for network orchestration. There are several proposed extensions to this interface currently under consideration which would improve our management capabilities. For example, industry standard certification programs for OpenStack are still under development (a recent proposal from the Ubuntu OpenStack Interoperability Lab may be a first step towards addressing this concern).

References

Akoush S, Sohan T, Rice A, Moore A.W., and Hopper A, "Predicting the performance of virtual machine migration",_Proc. 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, p. 37-46, Miami, FL (Aug 17-19, 2010)

Birand B, Wang H, Bergman K, Kilper D, Nandagopal T, Zussman G, "Real-time power control for dynamic optical networks - Algorithms and experimentation," Proc. 21st IEEE International Conference on Network Protocols (ICNP'13) (Oct 2013)

Bohrer E, "Amazon Web Services DynamoDB", Proc. AWS Re-Invent, San Francisco, CA (Oct. 2013)

DeCusatis C., Marty I,, Cannistra R, Bundy T, Sher-DeCusatis C.J., "Software defined networking test bed for dynamic telco environments", Proc. SDN & OpenFlow World Congress, Frankfurt, Germany (October 22-24, 2013).

DeCusatis C, editor, *Handbook of Fiber Optic Data Communication*, 4th Edition, Academic Press/Elsevier, New York (2013)

Fox A, Gribble S, Chawathe Y, Brewster E, Guatheier P. "Cluster Based Scalable Network Services." Proceedings sixteenth ACM Symposium on Operating Systems Principles (SOSP-16). October 1997

Gilbert S, Lynch N, "Brewer's conjecture and the feasibility of conssistent, available, partition-tolerant web services", ACM SIGACT News vol 33 no 2 p 51-59 (June 2002)

Gray J, Reuter A, *Transaction Processing*, Morgan Kauffman, New York (1993)

Isci C, Liu J,Abal B, Kephard J.O., Kouloheris J, "Improving server utilization using fast virtual machine migration", IBM Journal of Research and Development vol 55 no 6 paper 4 (Nov/Dec 2011)

Liu H, Jin H, Xu C.Z., Liao X, "Performance and energy modeling for live migration of virtual machines," Cluster Computing, Springer, Vol. 16, No. 2, pp. 249-264, June 2013

Manville J, "The power of a programmable cloud", OFC 2012 annual meeting, Anaheim, CA, paper OM2D.2 (March 18-22, 2013)

Manzalini A, et al., "Clouds of Virtual Machines in Edge Networks," IEEE Communications Magazine, Vol. 51, No. 7, pp. 63-70 , July 2013.

Panda A, Scott C, Ghodsi A, Koponen T, and Shenker S, "CAP for networks", Proc. HotSDN, Hong Kong, China (August 16, 2013)