A "Network-Paging" Based Method for Wide-Area Live-Migration of VMs

Yasusi Kanada, Toshiaki Tarui

Central Research Laboratory, Hitachi, Ltd.
Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185-8601, Japan
{Yasusi.Kanada.yq, Toshiaki.Tarui.my}@hitachi.com

***Abstract* – In cloud-computing environments, migration of virtual machines (VMs) between data centers can solve many problems such as load balancing and power saving. One of the difficulties in wide-area migration, however, is the "address-warping" problem, in which the address of the VM warps from the source server to the destination server. This confuses or complicates the status of the WAN, and the LANs connected to the WAN. We propose two solutions to this problem. One is to switch an address-translation rule, and the other is to switch multiple virtual networks. The former is analogous to paging in memory virtualization, and the latter is analogous to segmentation. The "network-paging" based method is described and our evaluation results are shown. It took less than 100 ms in average to switch from the source to the destination server using this method.**

I. INTRODUCTION

In cloud-computing environments, migration of virtual machines (VMs) between data centers is a very important operation. Wide-area VM migration can solve many problems such as load-balancing, disaster avoidance and recovery, and power saving. However, many problems must be solved to enable migration between distant locations.

One problem is "address-warping". When a VM is moved from one location to a more distant location, the addresses (i.e., the IP and MAC addresses in the current local and global network architecture) "warp" from the source to the destination servers. This confuses or complicates the status of both the WAN and LANs. If the WAN connecting these locations is an IP network, these LANs usually use different subnets. Therefore, the warped IP address must use a special mechanism such as Mobile IP. The same subnet may be used in the two locations by applying a special mechanism, such as L2 tunneling, but this makes the WAN routing far from optimal.

Therefore, it is difficult to move a VM that has real-time applications such as conferencing or online games. It is important to reduce the downtime of VMs in order to move VMs that contain real-time applications. To reduce the downtime, the confusion and complexity must be avoided. A type of virtualization technique can be introduced for this purpose.

We present a solution to this problem using address translation (i.e., a type of NAT [Sri 01]). In our method, the source and/or destination data-center subnets are mapped into different subnets in the WAN, and a VM motion causes a dynamic change of the user's address mapping that is handled by switching an address-translation rule. This method translates the original and moved addresses of the VM, which have the identical addresses but exist in different locations, into different addresses in the WAN. It is a type of virtualization similar to *paging* in memory virtualization [Kan 11].

In the rest of this paper, related work is shown in Section II. The two conventional methods of memory virtualization are described and the analogy of network and memory virtualizations is explained in Section III and two wide-area migration methods are proposed in Section IV. The method of "network-paging" based migration is proposed in Section V and evaluate it in Section VI. Section VII is the conclusion.

II. RELATED WORK

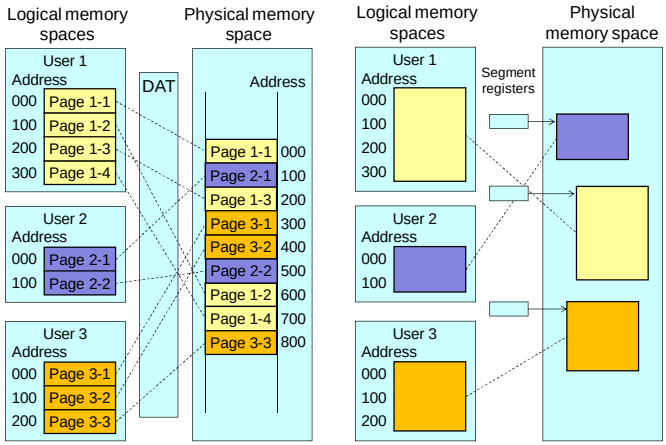
Live migration techniques were developed in Xen [Bar 03] [Cla 05], and independently developed in VMware [Wal 02] [Nel 05]. Xen enabled live migration of a session of Quake 3, which is an online game, in 60 ms [Cla 05]. However, both Xen and VMware only support migration within a LAN segment.

Many researchers [Li 08][Tra 06][Bra 07][Liu 09][Ram 07] [Sil 09][Voo 09][Hir 09] have worked on the development of wide-area live migration. Qin Li, et al. [Li 08] tried a Mobile-IP-based approach; i.e., they gave Mobile IP addresses to VMs, and handled proxy ARP (Address Resolution Protocol) messages. The downtime was about 30 sec. Some other researchers also used Mobile IP for VM migration. Liu [Liu 09] achieved downtime as short as several tens of milliseconds in many applications. Travostino, et al. [Tra 06] described a live-migration demo using Xen at iGRID 2005. They used a special-purpose light-path between Amsterdam, Chicago, and San Diego and IP tunnels, and the downtime was 0.8 to 1.6 sec. Bradford, et al. [Bra 07] experimented on migrating Web servers through a LAN and a WAN. They used dynamic DNS and tunnels in the WAN. The downtime was 3 sec in the LAN, and 68 sec in the WAN.

A simpler method for wide-area migration is to use a layer 2 VPN, such as VPLS (Virtual Private LAN Service), to connect two data centers by Ethernet protocol. In both Mobile IP and L2VPN based methods, packets from clients (VM users) to a VM are redirected by the source data center to the destination center. So the WAN path between the clients and the VM is not optimized.

III. PAGING, SEGMENTATION, AND MIGRATION

Historically, virtualization techniques were first developed in main-memory virtualization for computers. Two types of memory virtualization architectures [Tan 08] were developed.



(a) Paging (b) Segmentation

Fig. 1. Two types of memory virtualization architecture

- **Paging**: The memory space is divided into fixed-sized pages, and pages of all the users of a computer are mapped into a single large address-space (see **Fig. 1**(a)). Logical and physical memories are mapped to each other by using dynamic address-translation (DAT).
- **Segmentation**: The memory space is divided into logically separated and variable-sized segments, and each user uses a segment (see Fig. 1(b)). Logical and physical memories are mapped to each other by using segment registers that point to the head of the physical-memory segments.

Architecture similar to segmentation is widely used in network virtualization. VPN identifiers or VLAN identifiers, which correspond to segment identifiers or segment-register numbers in segmentation, are used in VPNs or VLANs. In contrast, architecture similar to paging seems to have been seldom used in network virtualization. However, several methods for using such architecture has been developed [Kan 11].

Both segmentation-based and paging-based methods can be used for wide-area migration. We developed a paging-based wide-area migration method.

IV. Two Wide-Area Migration Methods

To avoid the confusion and complexity caused by address warping, we should use a method that allows two or more of the identical addresses in different locations. Because it is not allowed in one network to have multiple identical addresses, the only available solutions are as follows.

- **Using multiple networks**: If there are multiple separate networks, multiple identical addresses may exist. The networks can be identified by identifiers or numbers (VPN IDs, VLAN IDs, etc.). This method is similar to *segmentation* in memory virtualization.
- **Using address translation**: If the identical addresses are translated into different addresses, they can coexist in a network. When a VM is moved, users can continually access the VM using the identical address through the address translation, which translates the address inversely. This method is similar to *paging* in memory virtualization.

V. Paging-Based Wide-Area Migration Method

In this section, we explain the method of live migration using address translation, which can be referred to as a "network-paging" [Kan 11] based method, and techniques for suppressing confusion in the data center LAN.

A. Assumed physical network structure

To simplify and clarify the description of the migration method, the network structure shown in **Fig. 2** is assumed. The Peak-time Data Center (PDC) and the All-day Data Center (ADC) are connected by a WAN. If we can move all VMs from PDC to ADC, we can shut down PDC and some routers in the WAN (such as R1 in the figure). The internet protocol (IP) is assumed to be used in the WAN, and a routing protocol such as Open Shortest Path First (OSPF) is used there. There are one or more user sites (such as Gateway C and Client U).

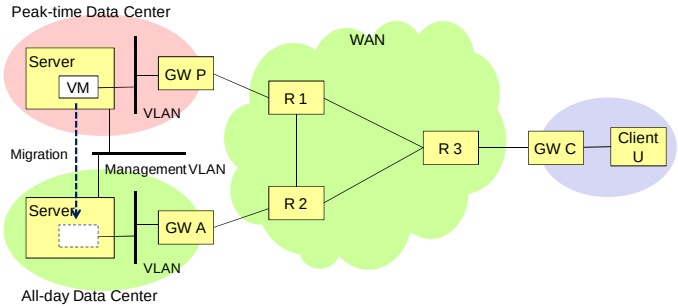


Fig. 2. Physical structure of network.

The WAN has server-side and client-side edge routers. In Fig. 2, R1 and R2 are server-side, and R3 is client-side; namely, R1 is connected to PDC, R2 is connected to ADC, and R3 is connected to the user site. There may be more routers (e.g., core routers) in the WAN. Client U uses the VM in PDC, but this VM will be moved to ADC before PDC is shut down.

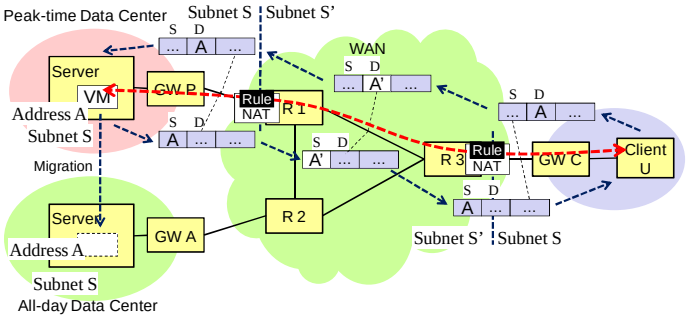
PDC and ADC can usually be managed separately using a local VLAN within each data center. These VLANs are separated from each other and may contain the identical IP and MAC addresses. However, there is also a management VLAN (which is an L2 network that may be an L2VPN through the IP-based WAN) between PDC and ADC to manage the migration. The management software can distinguish the servers in PDC and ADC even if they contain VMs with the identical addresses. The memory content and storage content are moved through the management VLAN when a VM is migrated.

B. Method of communication

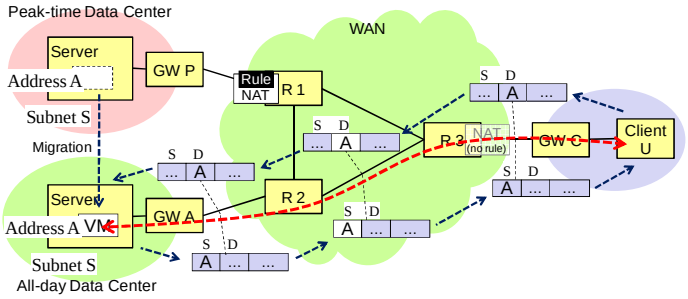
Before describing the VM motion technique itself, we explain the method used to switch packet streams from the source to the destination data center using **Fig. 3**. This switching process must be synchronized to the VM motion.

Both PDC and ADC are assumed to use subnet S and address A ($A \in S$) for the VM. There are two translators (NATs) on the path between PDC and the client, but no translators between ADC and the client. In Fig. 3(a), all the addresses in PDC are translated by the translator in R1 into subnet S' ($S' \neq S$), which is not used by any other subnets. This means the subnet addresses are translated, but the host addresses within

the subnet are invariant. For example, if S is **172.16**.*.* (an IPv4 subnet) and S' is **172.15**.*.* and the host address A is 172.16.**10.2**, then the translated address A' is 172.15.**10.2**.



(a) Communication before VM motion



(b) Communication after VM motion

Fig. 3. Logical structure of the network.

The translator in R3 translates addresses of VMs in PDC from subnet S' to S, but it does not translate addresses of VMs in ADC. Because the VM is in PDC now, address A' in the WAN is translated into A. In the above example, the address of the VM is translated into 172.16.10.2.

The translation rule (or translation table entry) in R1 can be described as follows.

$$\text{LAN : S} \underset{\text{destination}}{\overset{\text{source}}{\rightleftarrows}} \text{S' : WAN}$$

This rule means that the source address (subnet) of packets coming from PDC (LAN) is translated from address A in subnet S to address A' in subnet S', and the destination address (subnet) of packets coming from the WAN is translated from address A' in subnet S' to address A in subnet S (See Fig. 3(a)).

The translation rule in R3 can be described as follows.

$$\text{WAN : S'} \underset{\text{destination}}{\overset{\text{source}}{\rightleftarrows}} \text{S : LAN}$$

This rule means that the *source* address (subnet) of packets coming from the WAN is translated from A' in S' to A in S, and the *destination* address (subnet) of packets coming from the user site (LAN) is translated from A in S to A' in S', when the address is that of a VM in PDC. This rule is not applied to the addresses of VMs in ADC. This usage of translator is different from normal usage (i.e., reversed).

With this method, there is no need for application-level address translation; namely, IP addresses that occur in the IP payload do not need to be rewritten because all the application programs see the original addresses as the IP source and destination addresses. However, the content of the Internet Control Message Protocol (ICMP) and multicast protocols must be handled carefully [Sri 09], because WAN routers see and process the content. In addition, TCP/UDP checksums should probably be rewritten at the translators [Sri 01].

The communication between the client and the VM after the VM motion is depicted in Fig. 3(b). If client U is not assumed to use other VMs in PDC, the addresses in packets are never translated. After the VM is moved to ADC, address A (172.16.10.2) is used as is in the WAN. This address is unique in the WAN, so no confusion occurs. The translator in R3 does not translate the addresses in subnet S. Thus, in the above example, the address is kept at 172.16.10.2 so the user can continue to communicate with the VM.

C. Method of switching data centers

With this method, the switching caused by the VM motion does not change the configuration, nor the routing of the WAN, so it never confuses the WAN. If the configuration of ADC is properly prepared before the motion, the LAN in ADC is never confused either. The VM can access everything using the identical IP and MAC addresses as before.

The NAT in the client-side edge router (i.e., R3) in Fig. 2, must know which data center has the VM because it must translate the destination address of data packets from the client from A to A' when the VM is in PDC. So, when a VM is moved, the router must receive a message that contains the destination of the VM. This message sequence can be generated in many ways. One method is to install a program that captures a packet from the destination VM or server and that generates a message to the user-side edge routers.

Specific types of packets, such as ARP or RARP (Reverse Address Resolution Protocol) packets, can be captured for this purpose. Just after a VM motion, ARP or RARP packets are usually generated. In Xen, ARP packets are generated [Cla 05]. In VMware, RARP packets are generated [VMw 08]. The program can catch these packets. To do so, the program must be put somewhere in the same LAN segment that the VM uses for global communication. It may exist in a server, a separate box, or a LAN switch.

D. Method of routing

If dynamic routing is used in the networks shown in Fig. 2, addresses in route advertisements have to be translated. The WAN router connected to PDC, R1, must translate not only addresses in data packets but also routing information; namely, the routes in PDC must be advertised not as routes in subnet S but as routes in subnet S'. Routes in subnet S are advertised by the router connected to ADC, R2. The client-side router, R3, may also have to translate routing information.

E. Alternative methods for data center switching

In the above method, translators are inserted between the source data center and the client. However, we can use a reverse method, namely, we can insert translators between the destination data center and the client. In Fig. 3, if the VM is moved from ADC to PDC (i.e., in the reverse direction), we can use this method. In the initial state (Fig. 3(b)), no translator

works for the VM. However, two translators are between the VM and the client after the VM motion (Fig. 3(a)).

A more generalized method can also be used. Translators can be placed at both server-side edge routers (i.e., R1 and R2) in Fig. 2. In one of the edge routers (e.g., R1), address A in subnet S is translated into address A' in subnet S'. In the other edge router (i.e., R2), address A in subnet S is translated into address A" in subnet S". In the client-side edge router (i.e., R3), A' in S' is translated into A in S if A' is in PDC, and A" in S" is translated into A in S if A" is in ADC. This translation is symmetric between addresses A' and A" (subnets S' and S"). This method can be extended for VM motion among three or more data centers.

For simplification, the network in Fig. 2 contains only one client site, but there may be two or more client-sites that are connected to the same edge router. If multiple client sites exist and they use different edge routers of the WAN, the same method as above can be applied, but all the edge routers must translate addresses in the same manner.

F. Address-translation-based method and "network-paging"

The address mapping used in this method is shown in **Fig. 4**. The logical address spaces in the data centers are mapped into a single address space in the WAN with no address overlapping. There are two VM address spaces in this figure, but there could be three or more; VMs can be moved between three or more data centers. The addresses are mapped into the users' address space again. The address space of the WAN should be large, so IPv4 may not be sufficient. IPv6 is better suited for this purpose. The address space of VMs and that of users may be IPv4, IPv6, or any other type of space. All of them can be mapped into the single address space of the WAN.

In this figure, each address space contains only one page; namely, only one translation rule exists. However, many pages (i.e., many rules) with the same or different sizes may exist. Thus, the relationship between the left two address spaces is very similar to Fig. 1(a).

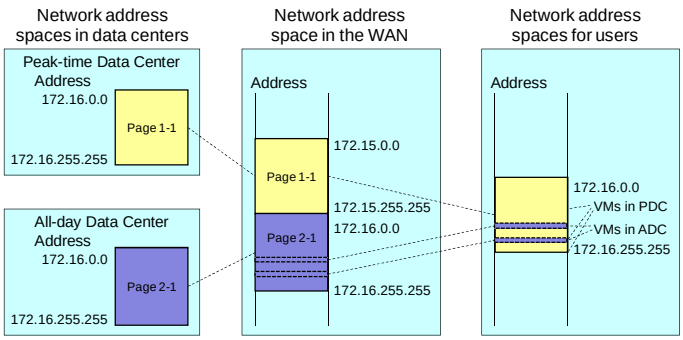


Fig. 4. "Network-paging" for migration.

G. Method of Live Migration

Essentially the same method can be used for live migration as that used in Xen or VMware. A management VLAN for moving memory and storage data can be used.

Identical addresses can coexist in the source and destination data centers. Therefore, the address of the moving VM in ADC can be created; namely, the addresses in ADC can be advertised, before activating the moved VM. This will make the communication of the VM smoother. However, it may be difficult to do so when using conventional server virtualization software such as Xen or VMware. If so, we may wait until the virtualization software creates the addresses in the moved VM.

The restart of the moved VM must be detected and the user-side router of this must be notified. As stated above, an ARP or RARP packet can be captured to do this.

When the VM is moved, the default gateway of the VM in PDC (i.e., GWA) is replaced by that in ADC (i.e., GWB). If the MAC addresses of these gateways are different, the VM will probably try to use the old gateway, GWA, first and will fail to communicate through it. We observed this type of failure using VMware. It took about 30 sec to recover the WAN connection by the moved VM. To avoid this type of failure, not only the identical local IP address but also the identical MAC addresses in GWA and GWB should be used. Then the communication through the gateway will never fail.

VI. EVALUATION

We evaluated the proposed method by using the network shown in **Fig. 5**. Three layer-3 switches (Alaxala AX6608S) connected by 10-Gbps links, two sets of blade servers (Hitachi BS1000) with VMware ESX Servers [Wal 02], two address translators (Linux PCs with two NIFs), and a client PC are used.[1] The subnets of the server LANs had a fixed size (16 bits). The address translators contained a translation program using promiscuous mode; i.e., it receives all the packets on the cable. The servers were managed using VMware VCenter through a management VLAN (a normal VLAN).

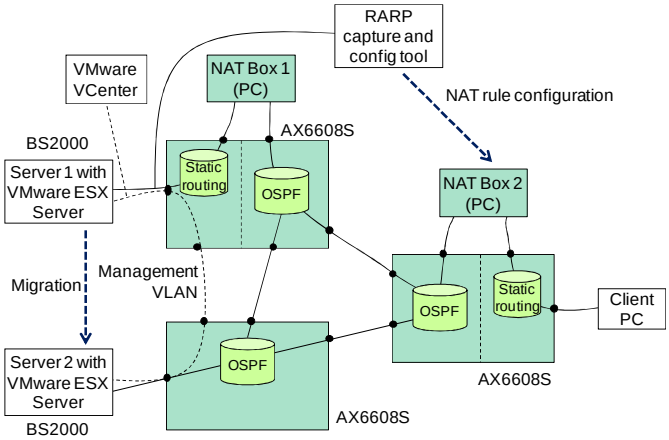


Fig. 5. Experimental network.

The simulated WAN was dynamically routed by OSPF, but static routing was used between the server-side translator and Server 1 and between the client-side translator and the client PC. The server was not directly connected to the translator because in order to use the identical MAC address in the two gateways, it was easier to give the identical MAC address to two switches than to give it to a switch and the translator. The switches were used both for OSPF routing and static routing,

[1] Alaxala, Hitachi, and VMware are company names, and AX6608S, BS1000, and ESX Server are products of these companies.

but a switch can have multiple routing tables (VRFs), so they were completely separated.

The destination server of a moved VM generates RARPs. It is captured by a "RARP capture and configuration tool". This tool sends a message to the client-side translator through the WAN, and this message updates the configuration of the translator. We measured the switching time, i.e., the time between the RARP packet generation and the VM restart at the destination server. The result was 80 ms in average (the standard deviation σ was 70 ms). We also measured the VM downtime, i.e., the time between the VM stop at the source server and the first RARP packet generation at the destination server using a UDP-packet generator and receiver programs. The generator generates a packet every 10 ms. The result was 790 ms in average (σ = 260 ms). The downtime was much larger than the switching time, but it may be shorter if the VM resources and the migration mechanism are optimized.

Our methods can be compared to other L3-based methods. If we move a subnet that contains the moved VM between the simulated data centers and we depend on update of dynamic routing, we must wait for 30 seconds or more until the network is stabilized again. Even if dynamic routing updates quickly, if the MAC addresses of the switches that are connected to the servers are different, our experiments showed that it took several seconds after moving the VM to recover the connection to the clients. In addition, Most of mobile IP based methods takes same order of time. Therefore, our method is much quicker than conventional methods.

VII. Conclusion

Two solutions to the "address-warping" problem were proposed. One is to switch a translation rule, and the other is to switch multiple virtual networks. We focused on the former in this paper, i.e., the "network-paging" based method. Methods for translating addresses between LANs and a WAN and a method for VM motion between distant data centers were proposed. The results of an evaluation using Linux-PC-based translator were shown. It took less than 100 ms in average to switch from the source to the destination server using this method. It was much shorter than the downtime of the VM. It is sufficiently small for most applications on VMs. However, the downtime caused by VM motion was too large for real-time applications, so it must be improved.

In the future, we will work on improving the performance of address translation. Because the performance of a PC-based translator is poor, application of large-scale (carrier-grade) NAT [Nis 09] should be considered. In addition, because many edge routers may have to be configured simultaneously in the proposed method, a scalable translator-configuration method (configuration protocol) should be developed. We also plan to do research on segmentation-based methods and to compare paging- and segmentation-based methods.

VIII. Acknowledgment

Part of the research results described in this paper is an outcome of the Eco-Internet Project (R & D on Power-Saving Communication Technology − Realization of Eco-Internet −) in fiscal year 2009 and the successor project in fiscal year 2010. Both projects were funded by the Ministry of Internal Affairs and Communications of the Japanese Government.

References

[Bar 03] Barham, P., Dragovic, B., Fraser, K., Hand, S., and Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., "Xen and the Art of Virtualization", 19th ACM Symposium on Operating Systems Principles (SOSP '03), pp. 164–177, 2003.

[Bra 07] Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H., "Live Wide-Area Migration of Virtual Machines Including Local Persistent State", *3rd ACM/Usenix International Conference On Virtual Execution Environments* (*VEE '07*), pp. 169–179, 2007.

[Cla 05] Clark, C., Fraser, K., Hand, S., Gorm Hansen, J., Jul, E., Limpach, C., Pratt, I., and Warfield, A., "Live Migration of Virtual Machines", *2nd Symposium on Networked Systems Design and Implementation*, pp. 273–286, 2005.

[Hir 09] Hirofuchi, T., Ogawa, H., Nakada, H., Itoh, S., and Sekiguchi, S., "A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds", *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 460–465, 2009.

[Kan 11] Kanada, Y. and Tarui, T., "Address-Translation-Based Network Virtualization", *10th International Conference on Networks* (*ICN 2011*), January 2011.

[Li 08] Qin Li, Jinpeng Huai, Jianxin Li, Tianyu Wo, and Minxiong Wen, "HyperMIP: Hypervisor Controlled Mobile IP for Virtual Machine Live Migration across Networks", *11th IEEE High Assurance Systems Engineering Symposium*, pp. 80–88, 2008.

[Liu 09] Haikun Liu, Hai Jin, Xiaofei Liao, Liting Hu, and Chen Yu, "Live Migration of Virtual Machine Based on Full System Trace and Replay", *18th ACM Int'l Symposium on High Performance Distributed Computing* (*HPDC '09*), pp. 101–110, 2009.

[Nel 05] Nelson, M., Lim, B.-H., and Hutchins, G., "Fast Transparent Migration for Virtual Machines", *2005 USENIX Annual Technical Conference*, pp. 25, 2005.

[Nis 09] Nishitani, T., Yamagata, I., Miyakawa, S., Nakagawa, A., and Ashida, H., "Common Functions of Large Scale NAT (LSN)", draft-nishitani-cgn-03, Internet Draft, IETF, November 2009.

[Ram 07] Ramakrishnan, K. K., Shenoy, P., and Van der Merwe, J., "Live Data Center Migration Across WANs: A Robust Cooperative Context Aware Approach", *2007 SIGCOMM Workshop on Internet Network Management* (*INM '07*), pp. 262–267, 2007.

[Ros 05] Rosenblum, M. and Garfinkel, T., "Virtual Machine Monitors: Current Technology and Future Trends", *IEEE Computer*, May 2005, pp. 39–47, 2005.

[Sil 09] Silvera, E., Sharaby, G., Lorenz, D., and Shapira, I., "IP Mobility to Support Live Migration of Virtual Machines Across Subnets", *The Israeli Experimental Systems Conference* (*SYSTOR 2009*), Article 13, 2009.

[Sri 01] Srisuresh, P. and Egevang, K., "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, IETF, 2001.

[Sri 09] Srisuresh, P., Ford, B., Sivakumar, S., and Guha, S., "NAT Behavioral Requirements for ICMP", RFC 5508, IETF, 2009.

[Tan 08] Tanenbaum, A. S., "Modern Operating Systems", Third Edition, Pearson Prentice Hall, 2008.

[Tra 06] Travostino, F., Daspit, P., Gommans, L., Jog, C., De Laat, C., Mambretti, J., Monga, I., Van Oudenaarde, B., Raghunath, S., and Wang, P. Y., "Seamless Live Migration of Virtual Machines over the MAN/WAN", *Future Generation Computer Systems*, Vol. 22, No. 8, pp. 901–907, October 2006.

[VMw 08] "Implementing Microsoft Network Load Balancing in a Virtualized Environment", VMware Infrastructure 3, Technical Note, VMware, 2008.

[Voo 09] Voorsluys, W., Broberg, J., Venugopal, S., and Buyya, R., "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation", *Cloud Computing 2009*, Lecture Notes in Computer Science, Volume 5931, Springer Verlag, 2009.

[Wal 02] Waldspurger, C. A., "Memory Resource Management in VMware ESX Server", *5th Symposium on Operating Systems Design and Implementation* (*OSDI '02*), December 2002.