Communication Infrastructures for Cloud Computing

Hussein T. Mouftah
University of Ottawa, Canada

Burak Kantarci
University of Ottawa, Canada

A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Managing Director: Lindsay Johnston
Production Manager: Jennifer Yoder
Publishing Systems Analyst: Adrienne Freeland
Development Editor: Christine Smith
Acquisitions Editor: Kayla Wolfe
Typesetter: John Crodian
Cover Design: Jason Mull

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Library of Congress Cataloging-in-Publication Data

Communication infrastructures for cloud computing / Hussein T. Mouftah and Burak Kantarci, editors.
pages cm
Includes bibliographical references and index.
ISBN 978-1-4666-4522-6 (hardcover) -- ISBN (invalid) 978-1-4666-4523-3 (ebook) -- ISBN 978-1-4666-4524-0 (print & perpetual access) 1. Cloud computing. 2. Internetworking (Telecommunication) I. Mouftah, Hussein T., editor. II. Kantarci, Burak, 1981- editor.
QA76.585C66 2013
004.67'82--dc23
2013023985

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 17
Virtual Machine Migration in Cloud Computing Environments:
Benefits, Challenges, and Approaches

Raouf Boutaba
University of Waterloo, Canada

Qi Zhang
University of Waterloo, Canada

Mohamed Faten Zhani
University of Waterloo, Canada

ABSTRACT

Recent developments in virtualization and communication technologies have transformed the way data centers are designed and operated by providing new tools for better sharing and control of data center resources. In particular, Virtual Machine (VM) migration is a powerful management technique that gives data center operators the ability to adapt the placement of VMs in order to better satisfy performance objectives, improve resource utilization and communication locality, mitigate performance hotspots, achieve fault tolerance, reduce energy consumption, and facilitate system maintenance activities. Despite these potential benefits, VM migration also poses new requirements on the design of the underlying communication infrastructure, such as addressing and bandwidth requirements to support VM mobility. Furthermore, devising efficient VM migration schemes is also a challenging problem, as it not only requires weighing the benefits of VM migration, but also considering migration costs, including communication cost, service disruption, and management overhead. This chapter provides an overview of VM migration benefits and techniques and discusses its related research challenges in data center environments.

DOI: 10.4018/978-1-4666-4522-6.ch017

Specifically, the authors first provide an overview of VM migration technologies used in production environments as well as the necessary virtualization and communication technologies designed to support VM migration. Second, they describe usage scenarios of VM migration, highlighting its benefits as well as incurred costs. Next, the authors provide a literature survey of representative migration-based resource management schemes. Finally, they outline some of the key research directions pertaining to VM migration and draw conclusions.

INTRODUCTION

With rapid expansion of Information Technology (IT) infrastructures in recently years, managing computing resources in enterprise environments has become increasingly complex. In this context, virtualization technologies have been widely adopted by the industry as a means to enable efficient resource allocation and management, in order to reduce operational costs while improving application performance and reliability. Generally speaking, virtualization aims at partitioning physical resources into logical resources that can be allocated to applications in a flexible manner. For instance, server virtualization is a technology that partitions the physical machine into multiple Virtual Machines (VMs), each capable of running applications just like a physical machine. By separating logical resources from the underlying physical resources, server virtualization enables flexible assignment of workloads to physical machines. This not only allows workload running on multiple virtual machines to be consolidated on a single physical machine, but also enables a technique called *VM migration*, which is the process of dynamically moving a virtual machine from one physical machine to another.

VM migration shares many similarities with its precursor called process migration, which aims at migrating a running process from one machine to another. Similar to VM migration, process migration moves the state of a running application process from one physical machine to another. However, its objective is to migrate running processes rather than VMs. Process migration has been extensively studied during the 1980s; however, it has been rarely used in practice due to the difficulty in handling the dependencies between various operating system modules. VM migration, on the other hand, does not suffer from these limitations. As VM migration moves the entire operating system along with the running processes, the migration problem is simplified and can be handled efficiently. Over the past decade, VM migration has proven to be a powerful technique for achieving a number of objectives, including workload consolidation, load balancing, reducing energy consumption, facilitating maintenance activities as well as supporting mobile applications. Consequently, it has received wide adoption in the industry in recent years. However, VM migration also has inherent challenges related to service disruption, bandwidth consumption, management overhead, and increased security risks. As such, devising applications that make effective use of VM migration has become a research question that gained considerable interest in the research community.

This chapter provides a comprehensive study of VM migration, highlighting its benefits, costs and underlying research challenges. First, it provides an overview of VM migration technologies found in the literature, and discusses the benefits and costs pertaining to VM migration. Then, it surveys various schemes that leverage VM migration for resource management in virtualized environments, and discusses key research directions related to VM migration. The ultimate goal is to provide an in-depth understanding of the state-of-the-art developments in the area of VM migration and to foster further research on this topic.

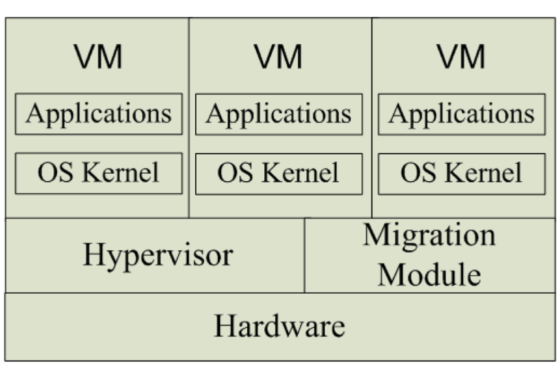
OVERVIEW OF VM MIGRATION TECHNOLOGIES

This section overviews existing virtual machine migration technologies, detailing their execution procedures and implementation approaches. The typical architecture of a migration-enabled virtualization platform is depicted in Figure 1 (Rosenblum & Garfinkel, 2005; Bobroff, Kochut, & Beaty, 2007). In addition to the hypervisor which is responsible for runtime resource allocation, a component called migration module is used to perform dynamic VM migration. As a VM primarily consumes four types of resources, namely CPU, memory, disk and network resources, the migration module is responsible for migrating the state of each type of resource from the source to the destination machine. In the following subsections, we discuss the way the state of each type of resource is transferred along with the description of the various migration techniques proposed in the literature.

Non-Live Migration

The simplest and most naive migration technique is non-live migration, which requires suspending and resuming the execution of VMs before and after the migration, respectively. As the VM execution is paused during the migration process, the migration problem simplifies to transferring the state of each type of resource to the destination machine. For instance, Zap (Osman, Subhraveti, Su, & Nieh, 2002) uses partial OS virtualization to group processes into process domains with isolated namespace for file handles and sockets, such that they can be easily migrated to a given target machine. Sapuntzakis et al. (2002) describe a migration scheme where the state of CPU registers, memory, disk and I/O devices are captured in data structures called capsules. The authors have proposed several techniques to reduce the data transfer time for copying capsules, such as using "ballooning" (Waldspurger, 2002) to avoid copying unmodified pages, on-demand page transfer and hash-based filtering for redundancy elimination. Similarly, Internet Suspend/Resume (Kozuch & Satyanarayanan, 2002) is a framework for supporting migration of VMs in mobile environments. To reduce the transfer time, the authors proposed techniques to exploit spatial and temporal locality of disk images as well as prefetching techniques to proactively transfer VM state to the destination machine in the background. However, despite its simplicity, non-live migration has not been widely used in the industry except for special circumstances, mainly due to the long and undesirable VM downtime during the migration process.

Figure 1. Typical architecture of a migration-enabled virtualization platform



Partial Migration

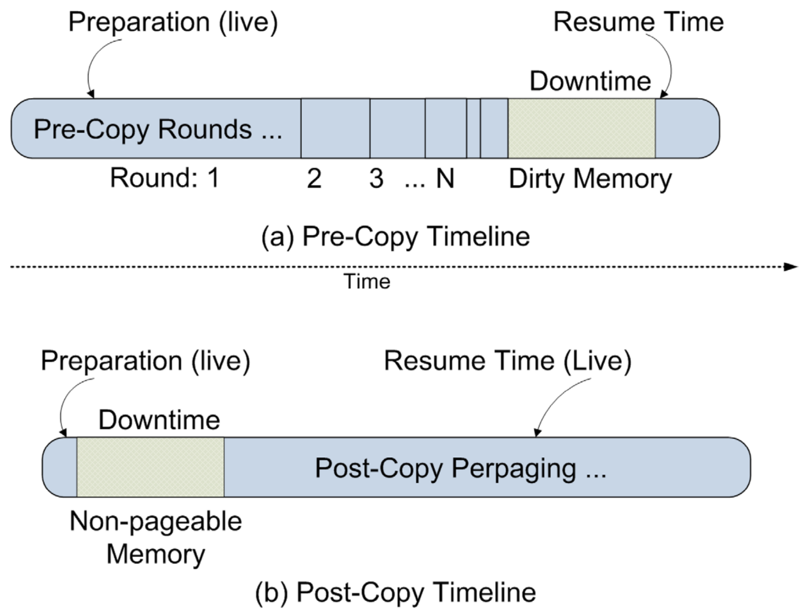
Partial live migration is a type of migration where only part of the VM image is copied to the destination machine. Partial live migration is useful when the VM will be migrated back to the source machine in the near future. In particular, Jettison (Bila et. al. 2012) is a framework that uses partial live migration to consolidate idle desktops in enterprise environments, in order to reduce energy consumption. Particularly, partial VM migration only transfers the VM descriptor (which contains VM configuration information), the CPU register state and the page table. When the VM needs to access memory content that resides on the source machine, a memory fault is generated, causing the page to be fetched from the source machine. In order to reduce the number of page faults, the migration module uses a prefetching technique which proactively transfers memory content to the destination machine in the background, based on the observed memory access patterns. Since the size of copied content is usually small, the migrated VMs are usually easy to consolidate. Generally speaking, partial live migration is only useful when the working set (i.e., the frequently accessed pages) is small, which is the case for idle desktop machines. It is still an open question whether partial VM migration can be applied to other scenarios as well.

Local Area Live VM migration

The most common type of VM migration is the local area live migration. In contrast with non-live VM migration, the goal of live VM migration is to maintain high availability of the running VM during the migration process, while reducing as much as possible the total transfer time. Generally speaking, there are two main approaches for live migration of VM process and memory states, as described below.

- **Pre-Copy Migration:** In Pre-copy migration, memory contents are copied to the target machine in the background while the VM is still running (Figure 2). As memory content can be changed during the transfer processes, the changed contents (called dirty pages) are iteratively copied to the target machine. The process continues until either the number of remaining pages is small, or a fixed threshold is reached, whichever happens first. VM is then suspended, allowing the remaining pages to be copied over. The VM will then resume its execution in the destination machine, and the source VM is then destroyed. The main benefit of pre-copy migration is low VM downtime (required for copying the remaining dirty pages). On the other hand, the total migration time can be long due to repeated copying of dirty pages.
- **Post-Copy Migration:** Post-copy migration refers to transferring memory content after the process state has been transferred (Figure 2). Specifically, in post-copy migration, the process states are first copied to the destination machine, allowing the VM to resume quickly. VM's memory contents are then actively fetched from source to target. All access to memory contents that have yet to be migrated are trapped by memory faults, causing the missing content to be fetched from source machine. As frequent memory faults can cause significant service disruption, additional techniques, such as memory prepaging, are often used to reduce the number of memory faults. Memory prepaging assumes that VM memory access exhibits special and temporal locality, therefore the subsequent memory access can be predicted with high accuracy. Therefore, by proactively transferring the related memory pages with high access probability, the number of memory faults can be significantly reduced. Finally, as the source VM no longer maintains the up-to-date memory contents, a failure during the migration process can potentially lead to unrecoverable VM states. One possible way to address this limitation is to checkpoint the VM state from the destination VM back to the source VM. Overall, the main benefit of post-copy migration is to reduce migration time, as memory contents are copied at most once during the entire process. However, it can cause more service disruption due to the occurrence of memory faults.

Figure 2. Timeline for pre-copy vs. post-copy (Hines, Deshpande, & Gopalan, 2009)



It is also possible to combine both pre-copy and post-copy migration into a hybrid migration technique. In particular, memory contents are copied proactively to the destination machine as in the pre-copy migration. Once complete, the VM state is suspended, allowing the dirty pages to be copied over to the destination machine. The VM execution is then resumed, with the new dirty pages pushed to the destination machine as in the post-copy migration. It is easy to see that this method achieves a trade-off between the advantages and disadvantages of both approaches.

Live Storage Migration

So far most of the discussion on live migration has been focusing on migrating memory and process states, and has ignored the issues pertaining to storage and network connection migration. Of course, when a networked storage such as a Storage Area Network (SAN) is present, there is no need to perform live storage migration. However, in environments where such networked storage is not available, live storage migration can be a challenging issue when performing VM migration. Mashtizadeh, Celebi, Garfinkel, & Cai (2011) described several techniques for live storage migration:

1. Snapshotting relies on periodically capturing snapshots of the file system. When live migration occurs, the most recent snapshot is copied to the destination machine and consolidated in the destination file system. At the same time, a new snapshot is created at the source machine that contains the change (i.e., deltas) in file system since the creation of the previous snapshot. Similar to pre-copy migration, snapshots are copied iteratively until the amount of data in the snapshot becomes small, at which time the VM is suspended to allow the final snapshot to be copied over.
2. Dirty block tracking uses a bitmap to keep track of the modified (i.e., dirty) blocks on the source disk, and iteratively copy the dirty blocks to the destination machine. To reduce the complexity of finding dirty

blocks, a hash-based filtering technique is used to quickly reduce the search space in the bitmap. Finally, as certain "hot" blocks can be written repeatedly, their migration is performed last to avoid repeated copying of the same blocks during the migration.

3. IO mirroring traps the IO write access at the source VM and mirrors the operation at the destination VM. During the migration process, the VM write access to a region that has been copied to the destination is mirrored in the target machine. On the other hand, writes to a region that is being copied to the destination are queued until the copy process finishes.

The main limitation of snapshotting is that it does not provide atomicity. As a result, canceling a migration can lead to different snapshots on the source and destination machines. On the other hand, dirty block tracking does not provide guaranteed convergence. If the I/O speed of the destination machine is slower than that of the source machine, dirty block tracker can lead to scenarios where the disk content is never synchronized between the source and destination machines. In contrast, IO mirroring does not have these limitations. Experiments using VMware ESX products showed that IO mirroring achieves best performance in terms of VM migration time and downtime (Mashtizadeh, Celebi, Garfinkel, & Cai, 2011), making it an attractive solution for live storage migration.

Network Connection Migration

Network connection migration is also an important aspect of live VM migration. As a VM may engage in multiple network connections simultaneously, it is necessary to ensure the liveness of the network connections during the migration. This is usually not an issue if both the source and destination machines are located in the same broadcast domain. However, in large data center networks or wide-area networks, maintaining active network connections during live VM migration requires support from the underlying network architecture. For instance, the forwarding and addressing schemes may be designed to support VM migration by allowing migrated VMs to maintain the same IP address in order to avoid service disruption. Indeed, many virtualization technologies such as VMware and Xen provide live migration capabilities with almost zero downtime (WMware; Xen). However, it is not always possible for a VM to maintain the same IP address after the migration in many circumstances. Specifically, many operators organize their data centers into different subnets/broadcast domains; this means that migrating a VM from one subnet to another requires a change of IP address. Unavoidably, this results in interrupting established TCP connections, which, in turn, leads to a service disruption.

We now discuss technologies that can be deployed to support network connection migration either for an intra-data center migration or wide-area migration.

VM Mobility within a Data Center

A trivial solution to overcome VM mobility challenges within a data center is to limit the migration to the same subnet/broadcast domain, however, this restricts the number of possible locations where the VM can be moved to. Ideally, selecting the physical machine able to host the migrated VM should not be limited by the underlying network configuration (e.g., broadcast domains, subnetworks), and the placement/migration of VMs should be driven by the availability of resources in the data center (i.e., CPU, memory, disk, bandwidth), application's performance or provider's management concerns.

In order to allow a migrated VM to retain its IP address, and thereby avoid service disruption, recent proposals have advocated decoupling the IP address of the VM from its location within the data center. As a consequence, packets can be delivered to the VM regardless of its current location. This decoupling between the IP address and the machine location can be achieved through various techniques (Bari et al., 2012): (1) layer 2 packet forwarding: (2) additional encapsulation (tunneling); (3) centralized address/location management. For instance, the Portland architecture (Mysore et al., 2009) uses a flat layer 2 topology (i.e., using only layer 2 equipment) with a special forwarding scheme that uses only MAC addresses to route packets inside the data center (even when packets do not belong to the same IP subnetwork). As a result, when a VM is migrated, it maintains its IP address and only its MAC address has to be updated according to its new location within the data center.

Another common technique for supporting IP mobility is to use an additional packet encapsulation. In particular, NetLord (Mudigonda, Yalagandula, Stiekes, & Pouffary, 2011) and VL2 (Greenberg et al., 2009) architectures use Ethernet-in-IP and IP-in-IP encapsulation, respectively. Packet forwarding is not performed using IP and MAC addresses of VMs, but rather using the ones of the physical machines (in NetLord) or the switches (in VL2). Consequently, the MAC and IP addresses are maintained when a migration is performed. Following the same idea of additional encapsulation, recently, some major networking and virtualization companies like Cisco and VMware have proposed Virtual eXtensible LAN – VXLAN, a mechanism that allows to extend Layer 2 broadcast domains within a data center. Basically, VXLAN encapsulates layer 2 frames into layer 3 packets (using an Ethernet-in-UDP encapsulation). Hence, migrated VMs retain their IP address and can be placed anywhere within the data center. The VXLAN technology ensures that VMs can communicate as if they were in the same broadcast domain, and hence the IP address of the migrated VM can be maintained regardless of its location. Of course, the major drawback of introducing an additional encapsulation is the high packet overhead (e.g., the additional overheads for NetLord, VL2 and VXLAN are 38, 20, and 50 bytes, respectively) (Mahalingam et al., 2012).

Finally, other architectures like SEC2 (Hao, Lakshman, Mukherjee, & Song, 2010b) and VICTOR (Hao, Lakshman, Mukherjee, & Song, 2010a) use a centralized management server to dynamically update forwarding tables of switches with the mapping of IP address to location. The traffic destined to the migrated VM is then rerouted according to the new mapping. Table 1 summarizes the technologies supporting IP mobility in data center environments and highlights their advantages and disadvantages.

VM Mobility Across Wide Area Networks

VM mobility across wide area networks can be achieved through techniques such as Mobile IPv4 and IPv6 described in IETF RFC 3344 and IETF RFC 3775, respectively. Mobile IPv4 defines two addresses for the migrated VM, namely its

Table 1. Network technologies supporting VM migration

Technique	Technology	Advantages	Disadvantages
Layer 2 packet forwarding scheme	Portland	- No encapsulation overhead	- Requires additional functionalities in switches.
Additional encapsulation	NetLord, VXLAN, VL2	- Standard Ethernet packet forwarding	- Encapsulation overhead - Requires a change either to the hypervisor or to the switches.
A centralized address/location management server	SEC2, VICTOR	- No additional encapsulation	- Single point of failure

original IP address called home address and a second address called care-of-address, which reflects the current point of attachment to its new location. A "Home" and "Foreign" agents are installed at the home network (i.e., the original data center) and the foreign network (i.e., the new hosting data center), respectively. When the migrated VM exchanges data, packets issued from a correspondent node are received by the home agent and then redirected through an IP tunnel towards the foreign agent, which in turn delivers them to the migrated VM. On the other hand, packets originating from the migrated VM are sent directly to the correspondent host. The disadvantage of IPv4 is that the communication between the migrated VM, the home agent and the correspondent host forms a triangular route. IPv6 provides some enhancement in this regard by allowing optimal data paths between the migrated VM and its correspondent node. Specifically, the care-of address is communicated to the correspondent host so that it can use it to forward packets directly to the migrated VM rather than sending them through the home agent. This provides an optimal communication path between the migrated VM and its correspondent host. One limitation of Mobile IPv6 is that it requires end-points to install IPv6 agents and intermediate nodes along the path to support IPv6.

Wide-Area Live Migration

Even though most of existing technologies for VM migration focus on local area networks (LANs), in many practical scenarios, such as geographical load balancing and cloud bursting (Wood, Ramakrishnan, Shenoy, & Merwe, 2011; Breitgand, Kutiel, & Raz, 2010; Harney, Goasguen, Martin, Murphy, & Westall, 2007; Kantarci, Foschini, Corradi, & Mouftah, 2012), it is also important to consider migration across multiple geographical domains (e.g., data centers). Unlike local-area migration, wide-area migration often requires transferring the disk image in addition to CPU and memory states. Furthermore, as network connections are less stable in wide-area networks, it is necessary to ensure reliability during the migration, while minimizing the total bandwidth usage. Bradford et al. (Bradford, Kotsovinos, Feldmann, & Schiberg, 2007) proposed a wide-area live VM migration scheme that simultaneously performs pre-copy migration and live disk migration. The changes to the memory and disk contents during the copy phase are recorded and enacted in destination machine once the copy phase is complete. However, as highly write-intensive workloads can cause significant increase in network traffic for synchronizing disk contents, the authors use write throttling to delay the write operations in the source machine when the rate of disk writes exceeds a fixed threshold. Similarly, CloudNet is a framework supporting wide-area VM migration which ensures the liveness of network connections of the VM being migrated (Wood et al., 2011). To achieve this objective, CloudNet uses Virtual Private LAN Service (VPLS) protocol to extend the broadcast domain and to ensure packets are delivered to the right host. CloudNet also provides several techniques to improve the efficiency of wide-area migration, such as adaptive thresholding for iterative copying of memory pages, in order to find an optimal trade-off between VM downtime and bandwidth consumption. Overall, wide-area VM migration is a relatively new technology that brings interesting challenges to the design of VM migration schemes. Nevertheless, with the rapid growth of online applications that provide services to multiple geographical regions, wide-area VM migration is expected to gain importance in the near future.

Summary

This section discussed implementation techniques for VM migration. Non-live migration is the easiest technique to implement but may result in high performance penalty in terms of service downtime. In turn, partial migration only migrates part of

VM memory state to the destination machine, anticipating that VM will be migrated back to the source machine in the near future. On the other hand, live-migration techniques aim at migrating VMs without causing significant disruption to VM execution. This is achieved at the expense of additional complexity. This section also discussed the migration of different types of resources and elaborated on the wide-area migration. As a summary, Figure 3 illustrates the procedures of each of the migration techniques described in this section. In the following, we will focus on the use of VM migration in data centers and describe the associated benefits and costs.

MIGRATION BENEFITS

This section discusses the benefits of VM migration and how it can be used to achieve various performance objectives.

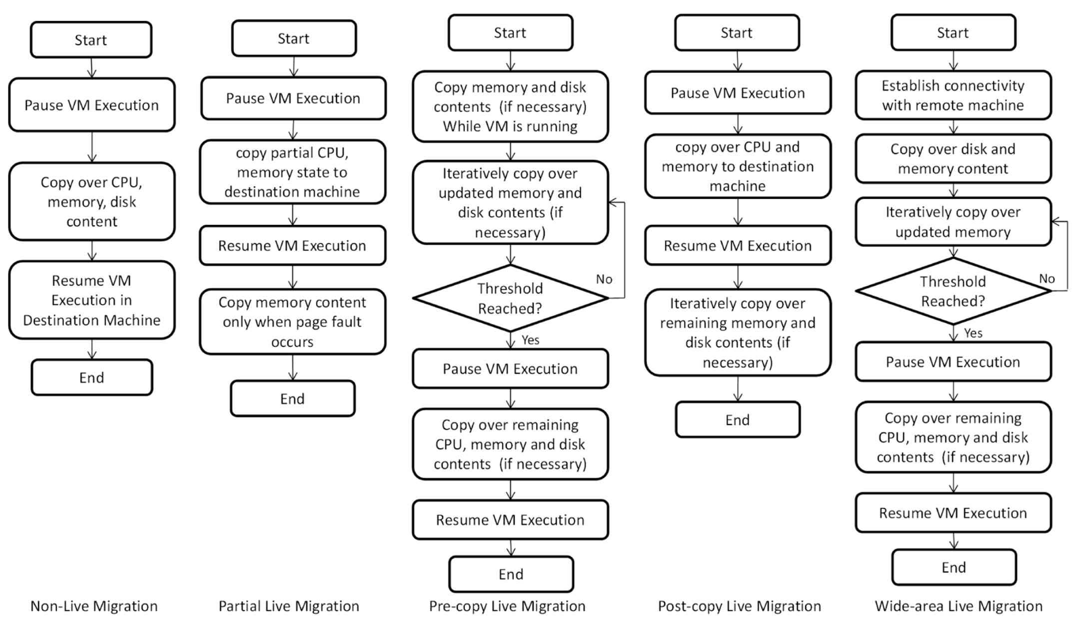
Server Consolidation

One of the main benefits of server virtualization is the ability to consolidate multiple VMs and pack them into a smaller number of physical machines, so the physical machines that are not hosting any running VM can be hibernated or turned off in order to reduce power consumption. This way, VM migration improves the flexibility of server consolidation by allowing VMs to be consolidated dynamically. In a production environment where workload fluctuates over time, it is often the case where the total workload is much less than the total capacity of the provisioned machines. In such a case, VM migration can facilitate workload consolidation on a few machines, so as to allow more machines to be set in a power-saving state and hence save energy.

Load Balancing

Although server consolidation brings many benefits in terms of energy saving and operational cost, it may also lead to performance degradation

Figure 3. Migration procedure for different live VM migration techniques



if it is not performed properly. Packing many VMs in a small number of physical machines may lead to machine overloading conditions (i.e., hot-spots), where the total resource usage of VMs exceeds machine capacity. As a consequence, the performance of the applications will be affected, resulting in long server response time and potentially low service availability. Another scenario that can cause overloading conditions is when a VM changes its usage pattern, but there is insufficient capacity on the allocated physical machine to continue supporting the VM execution. Using VM Migration, data center operators can balance the load on different machines so as to avoid overloading conditions. It should be pointed out that load balancing and server consolidation can conflict each other: On the one hand, server consolidation tries to consolidate VMs on a few servers, resulting in higher chances for machine overload to occur. On the other hand, load balancing tries to spread workload evenly on all machines, which often leads to resource under-utilization. Thus, finding an optimal trade-off between server consolidation and load balancing is essential for achieving efficient resource utilization in virtualized data centers.

Improving Data and Network Locality

VM migration can also be used to move a VM to a better location so as to improve data locality. In many data center environments where the file system is distributed across physical machines (e.g., Google File System and Hadoop file system), VM migration can be used to move a VM closer to the data it needs to access. This not only leads to significant improvement in I/O performance, but also reduces the traffic that needs to be carried in the data center network.

VM migration can be applied to improve network communication locality. For instance, if a pair of VMs exchange a large volume of traffic, it would be better to place them close to each other (e.g., in the same physical machine or the same rack). This not only reduces bandwidth consumption, but also reduces network access latency, resulting in better VM performance.

The benefit of using VM migration to improve network locality becomes more apparent in wide-area networks, where VMs and applications running on multiple virtualized infrastructures (data centers, personal computers and mobile devices) need to communicate with each other. Using VM migration can significantly reduce communication latency and network usage between these VMs. There are also other scenarios where wide-area migration is beneficial. For instance, VMs can be moved to data centers in close proximity to end-users in order to reduce access latency.

Reducing Energy Costs and Carbon Footprint

Wide-area migration can also be driven by other constraints, such as energy efficiency, availability of renewable resources and electricity price. The workload can be moved in response to electricity price fluctuations between different regions in order to ensure efficient and cost-effective execution of the applications. Furthermore, since many data centers are supported by renewable sources of energy that are available only in certain circumstances (e.g., daytime for solar power, only part of the time for wind power), migrating VMs based on the availability of such energy can maximize the use of green energy, minimize the carbon footprint of their infrastructure and eventually help to cut down energy costs. Therefore, VM migration provides management flexibility for cloud providers to improve the utilization of green energy and environmental friendliness of their data centers.

Reducing Hosting Costs

In today's cloud computing environments, resource price can differ significantly from one cloud provider to another and from one location to another. Furthermore, many cloud providers also introduce advanced dynamic resource pricing schemes (e.g., Amazon EC2 Spot Instance Service), where resource prices fluctuate over time. In this context, service providers can resort to wide-area VM migration techniques to move services between data centers in order to take advantage of the different prices among available offerings and their fluctuation over time. Such a strategy can achieve significant cost savings in terms of hosting service fees.

Facilitating Maintenance

VM Migration is also important in the context of maintenance as it provides more flexibility to data center operators such as the ability to migrate services before performing routine maintenance operations, including data center cleaning, device replacement, power and data cables physical inspection, and equipment reconfiguration. This is particularly useful since maintenance tasks usually require error-prone human interventions. Live migration mitigates such risks by allowing administrators to migrate VMs between different clusters within the same data center or between different data centers without disrupting ongoing services. Furthermore, live migration can be leveraged for ensuring business continuity in events of natural disasters by proactively moving critical services from the affected or soon to be affected areas.

MIGRATION COSTS

Despite the significant benefits that can be achieved using VM migration, there are also inherent costs that are introduced by existing VM migration technologies. This section discusses these costs.

Resource Consumption

Migrating VMs from one location to another can consume various types of resources such as CPU, disk as well as the bandwidth along the path from the source to the destination machine. Various studies in the literature have reported the resource overhead associated with VM migration. In particular, Wood et al. (Wood, Shenoy, Venkataramani, & Yousif, 2009) proposed an automated VM migration scheme for mitigating resource contention. They have found CPU overhead during VM migration can be up to 20% of machine utilization, and thus cannot be neglected. Similarly, Nelson, Lim, and Hutchins (2005) reported that it may require up to 30% CPU utilization to achieve maximum network throughput for VM migration over a gigabit Ethernet link. On the other hand, the disk and network overhead of VM migration is dependent on how much data need to be transferred (i.e., memory and disk image) as well as the duration of the VM migration process. In particular, there is a non-trivial trade-off between minimizing resource overhead and minimizing total migration time. If more resources such as CPU and bandwidth are allocated for VM migration, the migration process will finish faster. However, the additional CPU and bandwidth will have negative impact on the performance of VMs running on both the source and destination machines, as well as the network flows along the migration path (Takouna, Dawoud, & Meinel, 2012).

Various techniques have been proposed to mitigate the impact of resource overhead of VM migration. For example, Wood et al. use a CPU threshold for triggering VM migration, in order to allow sufficient free CPU capacity to absorb VM migration overhead. Another solution is to use rate limiters to control the bandwidth allocated for VM migration (Clark et al., 2005). This will not only reduce the bandwidth overhead, but CPU overhead as well. Memory compression (Jin, Deng, Wu, Shi, & Pan, 2009) is another technique for reducing bandwidth consumption during the migration process, as it reduces the total amount of data to be transferred to the destination machine. More advanced technologies also provide features for reducing VM migration overhead. For example, high speed interconnects such as InfiniBand can support Remote Direct Memory Access (RDMA), allowing VM migration time to be reduced significantly (Huang, Gao, Liu, & Panda, 2007). However, despite these techniques, finding effective policies for applying them is still a challenging problem, as it requires a careful understanding of the application performance objectives and finding a balance between resource overhead and total migration time.

Service Discontinuity

Despite current advances in live-migration technologies, service unavailability or short service downtime caused by migration is still unavoidable. Furthermore, during the migration process the applications running on source and destination machines may experience downgraded performance due to the additional incurred resource overhead. For example, Voorsluys, Broberg, Venugopal, and Buyya (2009), experimentally analyzed the performance penalty on a typical Web application, and found the application experienced 3 seconds of downtime and more than 44 seconds of downgraded performance. They also discovered that VM migration can also cause SLA violations in terms of 90th percentile and 99th percentile service response time for up to 300 seconds. Nelson et al. (2005) evaluated the impact of the VM migration using several industry benchmarks and found that the throughput for a database application can suffer up to 20% penalty during VM migration, even though the typical service downtime is less than 1 second. Thus, it is evident that performance penalty of VM live migration cannot be neglected, especially for applications with high performance objectives. In practice, a service disruption or long service response time translate into a profit loss (e.g., for commercial Web servers) or penalties (e.g., monetary penalties incurred by cloud providers such as Amazon EC2 for violating Service Level Agreements in terms of VM availability). Service disruption such as unavailability or degraded performance also has a hidden cost usually overlooked and consisting in customer dissatisfaction and eventually customer churn. All the above costs must be factored in when considering VM migration.

Management Overhead

A VM should be migrated to a physical machine that satisfies its new requirements. Many physical machines may satisfy the VM's requirements in terms of capacity. However, choosing the "best" placement of the VM should take into account multiple parameters such as the physical topology, the communication pattern between VMs (volume of data exchanged between VMs), the migration overhead (consumed bandwidth, migration duration), the service continuity, the energy consumption, security, management complexity and price, as shown in Table 2. For example, security compliance may require that the VM should only be scheduled on particular machines having special security features (e.g., firewall, intrusion prevention system) and trustworthiness. Therefore, finding an optimal placement of VMs on physical machines that balances all or some combination of the above objectives over time is a challenging problem. Furthermore, this problem must be solved in a scalable manner as the number of physical servers hosted in a single data center can go from ten to tens of thousands of servers running a larger number of VMs. This typically requires a non-negligible computational cost and significant management overhead.

Security Vulnerabilities

Another issue introduced by VM migration is the new security vulnerabilities that could be exploited by attackers. This is especially the case in public

Table 2. Physical machine (PM) selection criteria and corresponding benefits

PM selection criteria	Benefit
PM capacity	- Improve VM performance
Physical topology	- Reduce Latency between VMs - Reduce overall traffic in the Data Center - Increase bandwidth availability - Load balancing - Server consolidation
Patterns of communications between VMs	- Reduce Latency between VMs - Reduce overall traffic in the Data Center
Same broadcast domain as the original location	- Ensure service continuity
Least-loaded PM	- Load balancing - Avoid server overheat
Closest PM	- Reduce Latency between VMs - Reduce overall traffic in the DC
Hosting price	- Reduce costs
Machine utilization	- Reduce energy
Security compliance	- Ensuring VM security

cloud environments, where VMs belonging to multiple tenants with potentially conflicting interests can be collocated in the same data center. Oberheide, Cooke, and Jahanian (2008) provide an empirical study of vulnerabilities exposed by the current live VM migration in Xen. Specifically, the security vulnerabilities can occur at 3 different levels:

1. **Control plane:** Malicious users can issue false migration commands that cause victim VMs to be migrated to undesired locations. For example, a spoofing or replay attack can cause a VM to be migrated to an overloaded machine, resulting in service disruptions.
2. **Data plane:** As VM migration requires transferring memory and disk content across multiple machines and networks, a malicious user can eavesdrop or actively manipulate the content being transmitted causing the VM to malfunction.
3. **Migration module:** The migration module must be protected to avoid users from gaining full access to the virtual machine being migrated.

In order to secure VM live migration, it is necessary to develop techniques that prevent unauthorized access to and control over the virtualization infrastructure, as well as eavesdropping and manipulation of VM content during migration.

SURVEY OF MIGRATION SCHEMES

A large and growing body of research works has explored methods leveraging VM migration to improve data center resource management in terms of efficiency, performance and flexibility. This section surveys representative works.

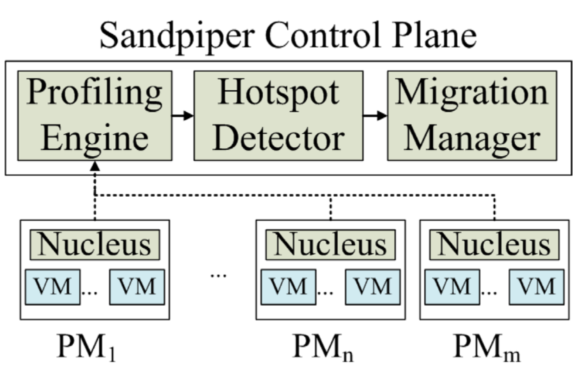
Sandpiper

Sandpiper (Wood, Shenoy, Venkataramani, & Yousif, 2009) is a VM migration scheme for data center environments designed to avoid machine overloading conditions (i.e., performance hotspots). Figure 4 shows sandpiper architecture. It consists of a control plane and a set of nuclei, which are daemons running on a Physical Machine (PM) in order to collect statistics about the hosted VMs. Sandpiper provides two monitoring strategies to collect statistics. The first one is the black-box strategy where statistics are collected without knowing the performance criteria of the application running on the VM. The second is the gray-box strategy where statistics not only contain resource consumption information of VMs but also the operating system statistics and application performance metrics.

As shown in Figure 4, the Sandpiper control plane consists of a profiling engine, a hotspot detector and a migration manager. The profiling engine receives the statistics from the nuclei and builds usage profiles for every PM and every VM. The hotspot detector monitors VMs' usage profiles and detects hotspots. A hotspot occurs when the usage of any resource (CPU, memory, disk, bandwidth) exceeds a threshold for a sustained period of time. When a hotspot is detected, the migration manager decides which VMs have to be relocated. The authors introduce a new metric called volume that captures the load of a virtual or physical machine in terms of cpu, memory, disk and bandwidth utilization. Whenever a hotspot is detected in a physical machine, the migration manager sorts its hosted VMs based on their volume and footprint memory and then try to move them one by one to the least loaded physical machine able to satisfy the requirement of the migrated VM.

The main limitation of Sandpiper is that it only considers application performance objective and ignores other important objectives such as energy consumption. It also fails to consider the communication pattern and the volume of data exchanged between VMs. Thus, there is a risk that the new location of the VM results in an increase of the latency between VMs and the higher amounts of traffic within the data center.

Figure 4. Sandpiper architecture



pMapper

pMapper (Verma, Ahuja, & Neogi, 2008) is a framework designed for dynamic, migration-aware workload placement in data centers that aims at finding a tradeoff between application performance and power consumption. The authors first studied the energy consumption characteristics of virtualized servers, and found energy cost is usually a concave function of resource utilization. As the workload placement problem generalizes the bin-packing problem, several heuristics are then proposed. The first greedy algorithm uses First-Fit Decrease for placing VMs in servers that are arranged in decreasing order of energy efficiency. However, this algorithm is only suitable for placing new workloads, as it does not consider existing workload placement. The second algorithm uses a local-search technique that attempts to balance the load on each machine using migration. The third algorithm is a hybrid approach, where a desired placement configuration is first computed, and then local search heuristics are used to incrementally convert current workload placement to the desired workload placement. Through simulations using realistic workload traces, it has been shown that pMapper is capable of achieving all the aforementioned objectives while achieving significant energy savings.

The main limitation of pMapper is that it makes simplistic assumptions; such as single resource type (e.g., CPU) while in reality each VM consumes multiple types of resources. It also fails to consider communication patterns between VMs.

AppAware

Optimizing bandwidth usage is a primary goal of data center resource management frameworks (Biran et al., 2012; Shrivastava et al., 2011). In this context, AppAware is a technique aiming at finding the optimal destination for a migrating VM in order to minimize the data center network load (Shrivastava et al., 2011). The main idea behind AppAware is to place VMs that communicate close to each other in order to reduce the latency and the overall traffic inside the physical network.

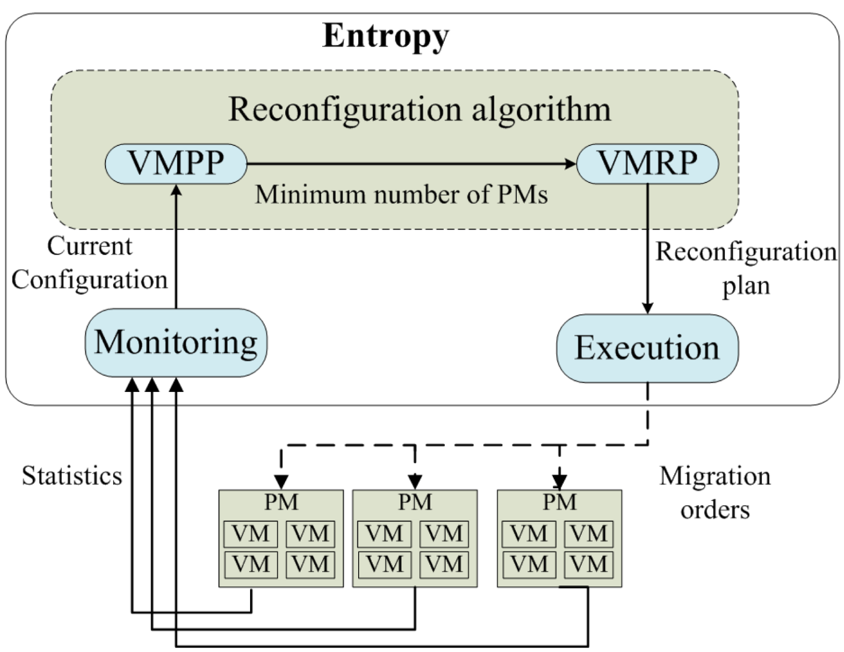
AppAware migrates an overloaded VM to a physical machine based on its resource requirements and a migration impact factor that is determined by (1) the network distance (latency or number of hops) between the VM to be migrated and other VMs, and (2) the dependency between VMs. Two VMs are dependent if they are running applications that communicate with each other. This dependency is proportional to the volume of traffic exchanged between them. The physical machine that satisfies the VM resource requirements while having the lowest migration impact factor is selected to be the destination of the VM.

The main benefit of AppAware is that it minimizes exchanged traffic within the data center. However, it does not take into consideration the available bandwidth between physical nodes. Thus, there is a risk that two communicating VMs require more bandwidth than what is available between their hosting PMs. Furthermore, AppAware triggers a migration only if a VM is overloaded, which means that it does not take into account other objectives such as load balancing and server consolidation.

Entropy

Entropy is a resource manager that relies on VM migration to dynamically achieve server consolidation while meeting VM capacity requirements for resources including CPU and memory (Hermenier, Lorca, Menaud, Muller, & Lawall,

Figure 5. Architecture of entropy



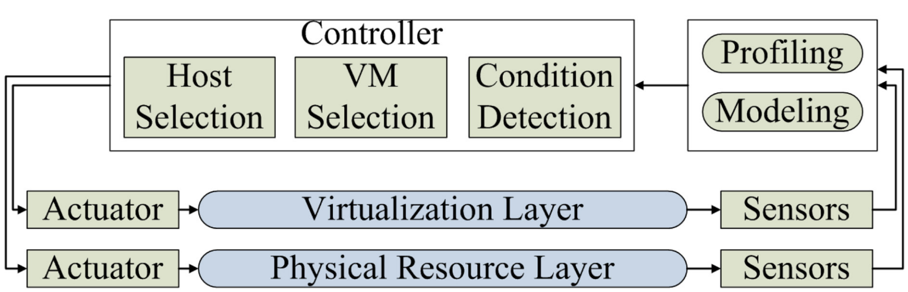
2009). As shown in Figure 5, Entropy achieves its objective in two phases. The first phase resolves the *Virtual Machine Packing Problem* (VMPP), which aims at finding the minimum number of physical machines that can host all VMs and satisfy their requirements. Once complete, the next challenge is to find a feasible migration strategy that converts the current VM placement configuration to the desired configuration. This is handled in the second phase. Specifically, the second phase solves *Virtual Machine Replacement Problem* (VMRP), which aims at finding a migration plan that minimizes the migration cost. The migration cost is measured as the number of required migrations and the amount of CPU, memory and bandwidth consumed by each migration.

Both VMPP and VMRP are solved using *Constraint Satisfaction Programming* (CSP). The CSP models a problem as a set of variables defined with their domains (i.e., possible values for each variable) and a set of constraints that should be satisfied by those variables. CSP is then solved by searching a combination of values that satisfies all constraints. The main drawback of Entropy is that it does not take into consideration the communications between VMs. In addition, it is only applicable for homogeneous clusters (i.e., all servers are identical in terms of capacity).

Multi-Objective Approach

Motivated by the fact that heat imbalance within a data center can lead to higher cooling costs (Moore, Chase, Ranganathan, & Sharma, 2005), Xu and Fortes (2011) derived a multi-objective approach to virtual machine management in data centers that aims not only to improve the performance of VMs (in terms of CPU, memory, and disk) but also to take into account power consumption and thermal properties (Xu & Fortes, 2011). The authors proposed a cross-layer framework that leverages monitoring data from different layers in order to manage VM migration. As depicted in Figure 6, sensors are placed in both virtualization layer and physical resource layer, in order to collect statistics including resource utilization, power consumption and server temperature. Based on the collected statistics, a profiler creates models for temperature and power consumption, which are then used to detect when migrations should take place. Specifically, there are three conditions under which migration should be triggered. The first condition is referred to as thermal emergency caused by an overheated server. In such a case, VMs having the highest CPU usage load should be migrated. The second condition happens when there is a high resource contention in the same physical server. In this case, all VMs with utilization higher than the average utilization are migrated. The third condition is when the utilization of one of the servers becomes lower than a threshold, which results in energy wastage. The controller decides then to migrate all VMs residing in the under-utilized server in order to turn it off and save energy. Once a VM is designated for migration, the controller selects physical ma-

Figure 6. Cross-layer control



chines on which it can be hosted using a utility function that ensures selecting the "coolest" host that has in the same time the highest utilization and enough resources to host the VM. Such utility function aims at avoiding heat imbalance within the data center.

However, the authors consider memory size of the VM as the only metric to evaluate migration cost since they assume that it has an impact on the migration time and overhead. This is indeed one of the weaknesses of this approach as migration cost depends also on the CPU and bandwidth usage as well as the thermal energy released during migration. One of the objectives of this work is to balance the temperature across the data center by placing VMs in the coolest hosts. However, authors do not take into consideration the data center topology as well as the physical placement of servers. Furthermore, this multi-objective approach relocates VMs without considering the communication patterns and the volume of data exchanged between VMs. In practice, these two factors certainly have a significant effect on application performance as well as server temperature.

Mistral

Mistral (Jung, Hiltunen, Joshi, Schlichting, & Pu, 2010) is a workload consolidation framework that dynamically adjusts VM placement in order to achieve the optimal tradeoff between power consumption, application performance, and adaptation costs (which includes the cost of adjusting VM capacity, live migrating VMs and shutting down/restarting a physical machine). Mistral relies on offline measurements to estimate run time adaptation costs, and a workload predictor to estimate the periods during which the workload for each application is stable. Given the workload and power conditions, Mistral first computes a desired solution using a simple optimization algorithm, then tries to find an optimal sequence of adaptation actions that brings the system to the desired optimal state. However, due to the large search space generated by the combinations of adaptation actions, finding the optimal adaptation sequence is intractable. Thus, Mistral proposes a self-aware A* search algorithm that keeps tracking a desired utility level and uses it to prune the search space. Experiments show that Mistral can significantly reduce the cost of energy and performance penalty compared to naive solutions as well as solutions that consider only a single objective (either energy or application performance).

However, the main limitation of Mistral is its high complexity. Given the fact that there can be thousands of physical machines and tens of thousands of virtual machines in large data centers, it is unclear whether Mistral can find a high quality solution in a reasonable time.

Cost-Aware Live Migration

Minimizing the impact of VM migration on application performance is a key challenge in cloud computing environments. As mentioned previously, pre-copy migration can be divided into two major phases: (1) iterative-copy phase, where memory pages of the VM are iteratively copied from the source machine to the destination machine without stopping the execution, and the (2) stop-and-copy phase, where VM is paused to allow any remaining pages to be copied to the destination host. Most of the existing work primarily focuses on minimizing downtime during the stop-and-copy phase and largely ignores the performance impact of the iterative-copy phase.

Recently, Breitgand, Kutiel, and Raz (2010) observed that there is a non-trivial trade-off between minimizing the duration of iterative-copy phase and maintaining acceptable QoS (Breitgand, Kutiel, & Raz, 2010). To address this issue, they proposed an analytical model to capture both migration time and the cost of meeting the QoS objective. Using the analytic model, it is possible to mathematically express the total cost of pre-copy phase as a function of the bandwidth allocation for the iterative-copy phase. Thus, it is possible to determine the optimal bandwidth allocation to minimize the total cost of the entire VM migration process. Finally, a practical online algorithm that dynamically adjusts the bandwidth allocation is proposed to minimize the total cost of the iterative-copy phase using the proposed analytical model.

While the work addresses an important problem in VM migration, it relies on a few assumptions to simplify the model: (1) dirty rate is uniform for all memory pages, and (2) both service rate and arrival rate follow Poisson distributions (i.e., the service model is a M/M/1 queuing model). We believe a more accurate VM migration model without these two assumptions will be more realistic for practical implementation.

Dynamic Migration-Aware Virtual Data Center Embedding for Clouds

Motivated by the fact that offering VMs without any network performance guarantee can hurt application performance and response time (Ballani, Costa, Karagiannis, & Rowstron, 2011), several recent works started offering virtualized resources in the form of Virtual Data Centers (VDCs). Specifically, a VDC consists of Virtual Machines (VMs) connected through virtual switches, virtual routers and virtual links with guaranteed bandwidth. However, most existing techniques for VDCs mapping in data center networks have not fully explored the possibility of using migration for improving the success rate of VDC mapping requests also known as VDC embedding. One exception is VDC Planner which provides a framework for migration-aware dynamic virtual data center embedding (Zhani, Zhang, Simon, & Boutaba, 2013). Migration is a key feature in VDC Planner used to dynamically adjust the allocation of physical resources in order support various usage scenarios, including VDC embedding, VDC scaling up and down as well as dynamic VDC consolidation.

VDC Planner uses migration to improve VDC embedding potential while minimizing total migration costs. Those costs are expressed in terms of penalties to be paid in case of service disruption caused by migration. It has been shown that efficient use of VM migration can significantly increase the number of embedded VDCs and consequently the revenue of the cloud provider. Unfortunately, this approach does not consider the potential migration impact on the performance of the data center network and other VDCs.

Network-Aware VM Management for Data Centers

As discussed earlier, VM migration can incur a significant overhead in terms network bandwidth, which may severely hurt the performance of other flows in the network. In this context, Mann et al., (2012) proposed a mathematical model to estimate the migration cost expressed as the amount of migration traffic by considering the number of required pre-copy iterations, the estimated page dirty rate, the VM memory size and the allocated bandwidth for the migration.

The authors have then proposed Remedy, a data center management framework that aims to balance the network load by relocating VMs. For each congested link, VMs to be migrated are chosen based on the migration cost, the number of communicating neighbors and also the total input/output traffic. The destination machines are then selected in order to satisfy the CPU, memory and bandwidth requirement of the migrating VMs. Finally, to select the migration path, Remedy uses collected link traffic statistics to ensure that no congestion could happen in any part of the network during the migration.

One limitation of this approach is that the derived migration cost does not factor in the number of hops separating the source and the destination of the moved VM. This can cause the migration traffic to cross multiple physical links and hence increase the load of the data center network.

Summary and Discussion

This section compares and contrasts the surveyed migration schemes. We identified four criteria for the comparison, namely the migration goal, the PM selection metrics, the migration cost, and the migration benefit. The migration goal determines the condition under which migration is triggered. The PM selection metrics determine the desired destination machine for migration. The migration cost summarizes how migration cost is computed and considered in the migration scheme. Finally, the migration benefit describes the benefits provided by each migration scheme.

Table 3 summarizes our comparison results. It can be seen from the table that none of the techniques addressed all the objectives at the same time. The multi-objective approach and Mistral tried to take into consideration multiple objectives including reducing energy costs and improving application performance. However, computational complexity becomes a key challenge as it is often difficult to compute near optimal solutions that simultaneously achieve multiple objectives in a scalable manner. The key result of the above comparative study indicates that it is still a challenge to devise comprehensive yet scalable solutions using VM migration for dynamic resource management in data centers.

KEY RESEARCH DIRECTIONS

This section discusses some of the research challenges related to VM migration that still require further investigations.

Network Architecture for Migration

In order to minimize service disruption, the underlying network architecture and communication protocols must ensure that it is possible to maintain the same IP address when migrating VMs.

For migration within the same data center, recent proposals still suffer from many drawbacks, as they require additional features that should be available at the switches (e.g., IP forwarding, MAC address rewriting) (Mysore et al., 2009; Hao et al., 2010b) or they incur high overhead (e.g., encapsulation overhead) (Mahalingam et al., 2012). An interesting direction that is worth investigating is to devise application or transport layer solutions for handling the service disruption due to migration. For example, one can devise a migration-aware transport layer protocol or a variant of TCP that can efficiently handle packet loss and delay incurred due to migration so as to minimize service disruption even when the IP address changes.

In wide area VM migration, achieving a short service downtime is still a challenging problem. Many solutions are already available such as mobile IPv4 and IPv6, however, Mobile IPv4 can lead to a non-optimal paths (triangular routing) for the exchanged traffic whereas Mobile IPv6 can incur significant management overhead especially when the number of migrated VMs becomes large. Alternatively, several proprietary solutions have been proposed recently such as Overlay Transport Virtualization (OTV) and Locator/ID Separation Protocol (LISP) proposed by Cisco, and Ethernet Virtual Interconnect (EVI) proposed by HP. Basically, these technologies aim at extending Layer 2 Ethernet networking across distant data centers to enable VM migration while maintaining the same IP address and reducing the risks of service disruption. However, they either require significant updates to switching and routing tables, or sending the traffic to the new destination through the original location. The former case results in

Table 3. Comparison of the migration schemes

Scheme	PM selection metrics	Migration cost	Migration goals
SandPiper	- VM resource requirement - Least-loaded PM	- Proportional to the amount of data transferred	- Avoid server overload
pMapper	-Adjust placement according to reference utilization	- Application performance	-Reduce power consumption -Avoid server overload
AppAware	- VM resource requirement - Latency between VMs - Reducing traffic between VMs	- Not considered	- Avoid server and VM overload - Improve data and network locality
Entropy	- VM resource requirement	- Number of triggered migrations - Resources consumed during the migration	- Avoid VM overload - Server consolidation
Multi-Objective approach	- VM resource requirement - Coolest physical machine - High utilization	- Proportional to the VM memory size	- Avoid server overload - Avoid server overheating - Reduce power consumption - Reduce cooling costs.
Mistral	- VM resource requirement - Energy cost - Migration cost	- Migration duration - Application response time - Change in power consumption	- Server consolidation - Maximize revenue while minimizing energy and adaptation costs
Cost-Aware Live Migration	N/A	- Application response time	- Minimize the service disruption during iterative-copy phase
VDC Planner	- Enough resources to accommodate a new VM. - Possibility of moving previously embedded VMs to make room for the new VM.	- Service disruption penalty	- Make room for incoming VDC requests - Improve network locality - Maximize revenue and number of embedded VDC
Remedy	- VM resource requirement	- The amount of traffic generated by the migration	- Improve data and network locality - Balance the network load

a significant management overhead whereas the latter case leads to higher bandwidth consumption. We believe that more practical solutions with less management overhead need to be developed to allow wide-area VM migration while maintaining the same IP address. For example, software defined networks such as OpenFlow technology can play a central role in reducing management overhead by automating the reconfiguration of the network element and simplifying the way traffic is routed to the new host of the migrated VM.

Automated Management

According to Gartner (Technology Research - Gartner Inc.), in a near future, 50 percent of all x86 data center server workloads will be running on virtualized hardware, and this number is expected to grow during the next couple of years. Consequently, the number of VMs hosted in data centers is also expected to rise significantly, which leads to an increase in management complexity. At the same time, many conflicting objectives and requirements related to performance and energy consumption should be taken into consideration in order to find acceptable tradeoffs between costs and benefits of VM migration. Previous work proposed gray-box and black-box monitoring strategies (Wood, et al., 2009) for making VM migration decisions, however, none has evaluated the ability of such centralized approaches to scale in large data centers. Finding a scalable solution for monitoring and defining effective policies to drive migration decisions is a challenge that requires further exploration.

Automated VM migration management techniques need to be defined for addressing the issues of complexity and scalability. In particular, sophisticated migration management strategies need to be devised for large scale VM deployment. For instance, a possible avenue for reducing the management overhead of migrating a large number of VMs is to assign migration priorities and define migration plans so as to mitigate the overhead and the impact on application performance.

Furthermore, the management schemes should also consider the reconfiguration of networking elements (e.g., firewalls, routing tables, access control lists) subsequent to VMs migration. Any solution that handles migration at scale must have the capability to perform such additional operations in a synchronized and effective manner. The challenge becomes more difficult for wide area migration, which requires also the configuration of WAN equipment.

Migration Technology

As VM migration becomes widely used in enterprise environments, improving the efficiency and effectiveness of VM migration is becoming an important problem. A number of recent proposals provided techniques to reduce migration time as well as service down time. For instance, memory and storage content can be compressed to reduce data transfer time during migration. There are also techniques that delay the copying of frequently modified content to reduce resource wastage during the migration process. However, we have found that most of the work to be application-oblivious. We believe that by exposing application characteristics and performance objectives, it may be possible to devise more efficient migration plans that minimize migration cost. Recent work by Breitgand et al. (2010) represents an initial effort towards this direction by finding a bandwidth allocation scheme that balances the tradeoff between total migration time and application SLA penalty for Web servers. However, much more work needs to be done for other types of resources and different workload types.

Wide-Area Migration

Wide-area migration is motivated by the desire to allow VMs to be migrated according to the distribution of service demand. For example, several recent proposals such as SAVI (Smart Applications on Virtual Infrastructure. http://www.savinetwork.ca), cloudlet (Verbelen, Simoens, De Turck, & Dhoedt, 2012), edgecloud (Islam & Gregoire, 2010), nano-data centers (Valancius, Laoutaris, Massoulié, Diot, & Rodriguez, 2009) rely on small-scale edge data centers at the access networks to improve performance especially in terms of latency and service response time. At the same time, remote cloud data centers can be used to host less latency-sensitive services as well as computing- and storage-intensive applications. Such architectures drive the need for a seamless and efficient live migration of services between edge data centers and remote data centers. Managing inter-data center migrations brings several new research challenges.

As discussed earlier, when it comes to live WAN migration, one fundamental problem is how to ensure continuous availability of the service offered by a VM. One important parameter is the migration time, which is highly dependent on the VM size and the performance of the WAN links. Recent studies reported that performance of WAN links is extremely variable which raises serious concerns about the possibility of ensuring fast and reliable WAN migrations (Wood et al., 2011). Dedicated links for supporting VM migration can be one solution, however, it is not always possible to guarantee such reserved links between distant sites. Ideally, WAN migration techniques should be adapted to the dynamicity of the WAN links.

At the same time, another issue that is worth investigation is how to reduce the amount of data to be migrated across distant sites.

Finally, there is also an issue regarding when WAN migration should be triggered. Many factors can drive such a decision. For instance, the VM can be migrated closer to third party services used by the hosted application in order to reduce response time. Another reason for migration is to move the VM closer to end-users so as to improve performance by reducing latency. VMs can be also migrated if more capacity in terms of processing, memory, and bandwidth is required. Furthermore, hosting costs can also play a role in such a decision. These costs depend not only on power consumption that can differ from one data center to another, but also on fluctuation of electricity price in each geographical region (Zhang, Zhu, Zhani, & Boutaba, 2012). Devising an inter-data center management framework that can take into account all these factors to perform inter-domain migration is a challenging problem that needs to be addressed in order to fully capitalize on the agility offered by virtualization technology at the inter-data center level.

Support for Mobile Cloud Computing

Mobile cloud computing is an emerging technology that tries to leverage the abundant resources in data centers to overcome the issues related to the scarcity of resources and limitation of energy on mobile devices. A key motivation for mobile cloud computing is to allow mobile users to run data and computation intensive tasks in the cloud, while using the mobile device as a thin client for user interaction. In this context, recent proposals advocated to use VMs hosted in the cloud as backends for mobile devices. However, this raises the question of how to ensure continuous connectivity and low latency between mobile devices and their corresponding VMs, especially when considering the mobility of users. Live VM migration can be a promising technique to solve such a problem as it can be used to ensure that each VM "is following" its corresponding mobile device. This requires the design of a management framework capable of controlling the migrations while addressing challenges related to scalability (e.g., the number of VMs to be managed), fault-tolerance (e.g., what happens if a migration fails) as well as request routing during migrations. Thus, we believe designing a management framework that leverages VM migration to facilitate the collaboration between the cloud and mobile devices is a key research problem in mobile cloud computing environments.

Security

VM migration has significant benefits and can facilitate resource management in cloud computing environments. However, it also raises new security challenges that need to be addressed by the virtualization infrastructure. As mentioned previously, even though Xen does not provide sufficient security counter-measures against spoofing and manipulation attacks, many techniques are available and can be used to mitigate these issues. For example, the Trust Platform Module (TPM) is an industry-standard trusted hardware that can be used as a trust root device. Recently, Berger et al. (2006) proposed a technique for virtualizing TPM (vTPM) modules and using vTPM to ensure secure VM live migration (Berger et al., 2006). Using vTPM, it is possible to check the manipulation of VM images through verification of image digests created using cryptographic techniques. Though vTPM seems capable of preventing manipulation attacks, it does not provide defense against other types of threats, such as eavesdropping and replay attacks. We believe more sophisticated and collaborative defenses are necessary to provide a more secure platform for supporting VM migration.

CONCLUSION

As virtualization technologies gain wider adoption in enterprise and cloud environments, devising effective schemes for managing virtualized resources has become a critical issue. In this context, VM migration can serve as a powerful tool for adjusting workload placement in a dynamic manner to achieve a variety of resource management objectives, including load balancing, server consolidation, improving data and communication locality, reducing energy consumption, as well as supporting mobile applications. However, despite these benefits, VM migration has inherent costs in terms of service disruption, resource consumption, management overhead, as well as security risks. This chapter provided a comprehensive study of VM migration, including an overview of existing VM migration technologies, a discussion of the advantages and disadvantages of VM migration under different contexts, and a survey of recent works applying VM migration for resource management. This chapter also identified several key research directions that require further exploration in order to bolster the expected benefits of VM migration and circumvent its potential costs. We believe that VM migration is a promising technology that will continue to improve and to stimulate new applications in the future.

REFERENCES

Ballani, H., Costa, P., Karagiannis, T., & Rowstron, A. (2011). Towards predictable datacenter networks. *SIGCOMM Computer Communication Review*, *41*(4), 242–253. doi:10.1145/2043164.2018465.

Bari, M., Boutaba, R., Esteves, R., Granville, L., Podlesny, M., Rabbani, M., et al. (2012). Data center network virtualization: A survey. *IEEE Communications Surveys Tutorials,* (99), 1-20.

Berger, S., Caceres, R., Goldman, K. A., Perez, R., Sailer, R., & van Doorn, L. (2006). vTPM: Virtualizing the trusted platform module. In *Proceedings of USENIX Security Symposium*. Berkeley, CA: USENIX Association.

Bila, N., de Lara, E., Joshi, K., Lagar-Cavilla, H., Hiltunen, M., & Satyanarayanan, M. (2012). Jettison: Efficient idle desktop consolidation with partial VM migration. In *Proceedings of the 7th ACM European Conference on Computer Systems* (pp. 211–224). ACM.

Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., et al. (2012). A stable network-aware VM placement for cloud systems. In *Proceedings of the IEEE/ACM international symposium on cluster, cloud and grid computing* (pp. 498–506). Washington, DC: IEEE Computer Society.

Bobroff, N., Kochut, A., & Beaty, K. (2007). Dynamic placement of virtual machines for managing SLA violations. In *Proceedings of the IFIP/IEEE international symposium on integrated network management* (pp. 119-128). IEEE.

Bradford, R., Kotsovinos, E., Feldmann, A., & Schiöberg, H. (2007). Live wide-area migration of virtual machines including local persistent state. In *Proceedings of the 3rd international conference on virtual execution environments* (pp. 169–179). IEEE.

Breitgand, D., Kutiel, G., & Raz, D. (2010). Cost-aware live migration of services in the cloud. In *Proceedings of the 3rd Annual Haifa Experimental Systems Conference*. New York: ACM.

Chowdhury, N. M. K., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, *54*, 862–876. doi:10.1016/j.comnet.2009.10.017.

Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., et al. (2005). Live migration of virtual machines. In *Proceedings of the symposium on networked systems design & implementation* (vol. 2, pp. 273–286). Berkeley, CA: USENIX Association.

Edwards, A., Fischer, A., & Lain, A. (2009). Diverter: A new approach to networking within virtualized infrastructures. In *Proceedings ACM WREN*. ACM.

Gartner. (n.d.). *Technology research*. Retrieved from http://www.gartner.com

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., & Lahiri, P. et al. (2009). VL2: A scalable and flexible data center network. *SIGCOMM Computer Communication Review*, *39*(4), 51–62. doi:10.1145/1594977.1592576.

Hao, F., Lakshman, T. V., Mukherjee, S., & Song, H. (2010a). Enhancing dynamic cloud-based services using network virtualization. *SIGCOMM Computer Communication Review*, *40*, 67–74. doi:10.1145/1672308.1672322.

Hao, F., Lakshman, T. V., Mukherjee, S., & Song, H. (2010b). Secure cloud computing with a virtualized network infrastructure. In *Proceedings of the USENIX conference on Hot topics in cloud computing* (pp.16–16). Berkeley, CA: USENIX Association.

Harney, E., Goasguen, S., Martin, J., Murphy, M., & Westall, M. (2007). The efficacy of live virtual machine migrations over the internet. In *Proceedings of the international workshop on virtualization technology in distributed computing* (pp. 8:1–8:7). New York, NY: ACM.

Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G., & Lawall, J. (2009). Entropy: A consolidation manager for clusters. In *Proceedings of the ACM SIGPLAN/SIGOPS international conference on Virtual execution environments* (pp. 41–50). ACM.

Hines, M. R., Deshpande, U., & Gopalan, K. (2009). Post-copy live migration of virtual machines. *SIGOPS Operating Systems Review*, *43*(3), 14–26. doi:10.1145/1618525.1618528.

Huang, W., Gao, Q., Liu, J., & Panda, D. K. (2007). High performance virtual machine migration with rdma over modern interconnects. In *Proceedings of the IEEE international conference on cluster computing* (pp. 11–20). Washington, DC: IEEE Computer Society.

Islam, S., & Gregoire, J.-C. (2010). Network edge intelligence for the emerging next-generation internet. *Future Internet*, *2*(4), 603–623. doi:10.3390/fi2040603.

Jin, H., Deng, L., Wu, S., Shi, X., & Pan, X. (2009). Live virtual machine migration with adaptive, memory compression. In *Proceedings of the IEEE International Conference on Cluster Computing and Workshops (CLUSTER)* (pp. 1 -10). IEEE.

Jung, G., Hiltunen, M. A., Joshi, K. R., Schlichting, R. D., & Pu, C. (2010). Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)* (pp.62–73). IEEE.

Kozuch, M., & Satyanarayanan, M. (2002). Internet suspend/resume. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications* (pp. 40–46). IEEE.

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., & Sridhar, T. et al. (2012). *VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks*. Internet Draft – IETF.

Mann, V., Gupta, A., Dutta, P., Vishnoi, A., Bhattacharya, P., Poddar, R., et al. (2012). Remedy: Network-aware steady state VM management for data centers. In *Proceedings of the international IFIP TC 6 conference on networking* (pp. 190–204). Berlin, Germany: Springer-Verlag.

Mashtizadeh, A., Celebi, E., Garfinkel, T., & Cai, M. (2011). The design and evolution of live storage migration in VMware ESX. In *Proceedings of the USENIX annual technical conference* (pp. 14–14). Berkeley, CA: USENIX Association.

Moore, J., Chase, J., Ranganathan, P., & Sharma, R. (2005). Making scheduling cool: temperature-aware workload placement in data centers. In *Proceedings of the USENIX Annual Technical Conference* (pp. 5–5). Berkeley, CA: USENIX Association.

Mudigonda, J., Yalagandula, P., Stiekes, B., & Pouffary, Y. (2011). NetLord: A scalable multi-tenant network architecture for virtualized datacenters. In *Proceedings of ACM SIGCOMM*. ACM.

Mysore, R., Pamboris, A., Farrington, N., Huang, N., Miri, P., Radhakrishnan, S., et al. (2009). PortLand: A scalable fault-tolerant layer 2 data center network fabric. In *Proceedings of ACM SIGCOMM*. ACM.

Nelson, M., Lim, B.-H., & Hutchins, G. (2005). Fast transparent migration for virtual machines. In *Proceedings of the USENIX annual technical conference* (pp. 25–25). Berkeley, CA: USENIX Association.

Oberheide, J., Cooke, E., & Jahanian, F. (2008). *Empirical exploiting live virtual machine migration*. Paper presented at Blackhat dc briefings. Washington, DC.

Osman, S., Subhraveti, D., Su, G., & Nieh, J. (2002). The design and implementation of zap: A system for migrating computing environments. *ACM SIGOPS Operating Systems Review, 36*(SI), 361–376.

Perkins, C. (Ed.). (2002). *IP mobility support for IPv4*. RFC Editor.

Rosenblum, M., & Garfinkel, T. (2005). Virtual machine monitors: Current technology and future trends. *IEEE Computer*, *38*(5), 39–47. doi:10.1109/MC.2005.176.

Sapuntzakis, C. P., Chandra, R., Pfaff, B., Chow, J., Lam, M. S., & Rosenblum, M. (2002). Optimizing the migration of virtual computers. *SIGOPS Operating Systems Review, 36*(SI), 377–390.

Shrivastava, V., Zerfos, P., & Lee, K. won, Jamjoom, H., Liu, Y.-H., & Banerjee, S. (2011). Application-aware virtual machine migration in data centers. In Proceedings of IEEE INFOCOM (pp. 66 -70). IEEE.

Takouna, I., Dawoud, W., & Meinel, C. (2012). Analysis and simulation of HPC applications in virtualized data centers. In *Proceedings of the IEEE international conference on green computing and communications (GreenCom)* (pp. 498-507). IEEE.

Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., & Rodriguez, P. (2009). Greening the internet with nano data centers. In *Proceedings of the International Conference on Emerging networking experiments and technologies* (pp. 37–48). New York: ACM.

Verbelen, T., Simoens, P., De Turck, F., & Dhoedt, B. (2012). Cloudlets: Bringing the cloud to the mobile user. In *Proceedings of the 3rd ACM workshop on Mobile cloud computing and services* (pp. 29–36). New York: ACM.

Verma, A., Ahuja, P., & Neogi, A. (2008). pMapper: Power and migration cost aware application placement in virtualized systems. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware* (pp. 243–264). ACM.

Virtual Machine Mobility with VMware vMotion and Cisco Data Center Interconnect Technologies. (n.d.). Retrieved from http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns836/white_paper_c11-557822.pdf

VMware. (n.d.). Retrieved from http://www.vmware.com

VMware vSphere. (n.d.). *vMotion for live migration*. Retrieved from http://www.vmware.com/products/vmotion/features.html

Voorsluys, W., Broberg, J., Venugopal, S., & Buyya, R. (2009). Cost of virtual machine live migration in clouds: A performance evaluation. In *Proceedings of the international conference on cloud computing* (pp.254–265). Berlin, Germany: Springer-Verlag.

Waldspurger, C. (2002). Memory resource management in VMware ESX server. *ACM SIGOPS Operating Systems Review, 36*(SI), 181–194.

Wood, T., Ramakrishnan, K. K., Shenoy, P., & van der Merwe, J. (2011). CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines. *SIGPLAN Notification*, *46*(7), 121–132. doi:10.1145/2007477.1952699.

Wood, T., Shenoy, P., Venkataramani, A., & Yousif, M. (2009). Sandpiper: Black-box and gray-box resource management for virtual machines. *Computer Networks*, *53*, 2923–2938. doi:10.1016/j.comnet.2009.04.014.

Xen. (n.d.). Retrieved from http://xen.org

Xu, J., & Fortes, J. (2011). A multi-objective approach to virtual machine management in datacenters. In *Proceedings of the ACM international conference on Autonomic computing (ICAC)* (pp. 225–234). ACM.

Zhang, Q., Zhu, Q., Zhani, M. F., & Boutaba, R. (2012). Dynamic service placement in geographically distributed clouds. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*. ICDCS.

Zhani, M. F., Zhang, Q., Simon, G., & Boutaba, R. (2013). VDC planner: Dynamic migration-aware virtual data center embedding for clouds. In *Proceedings of the 13th IFIP/IEEE Integrated Network Management Symposium*. Ghent, Belgium: IEEE.

KEY TERMS AND DEFINITIONS

Dynamic Consolidation: The ability to dynamically consolidate multiple VMs into a small number of physical machines.

Live Migration: A type of VM migration that allows the VM to remain operational during the migration period.

Non-Live Migration: A type of VM migration that suspends the execution of the virtual machine during the entire migration process.

Partial Live Migration: A type of live migration where only a part of the VM image is copied to the destination machine.

Virtual Data Center: A virtual infrastructure that consists of virtual machines connected through virtual switches, virtual routers and virtual links with guaranteed bandwidth.

Virtual Machine: A software implementation that emulates a physical machine environment where one can execute programs and applications just like a physical machine.

VM Migration: The process of moving a virtual machine, and more precisely the transfer of its storage, memory, and network connectivity from one physical machine to another.