

Dynamic Resource Management Using Virtual Machine Migrations

Mayank Mishra, Anwasha Das, Purushottam Kulkarni, and Anirudha Sahoo
Indian Institute of Technology Bombay

ABSTRACT

Virtualization is a key concept in enabling the “computing-as-a-service” vision of cloud-based solutions. Virtual machine related features such as flexible resource provisioning, and isolation and migration of machine state have improved efficiency of resource usage and dynamic resource provisioning capabilities. Live virtual machine migration transfers “state” of a virtual machine from one physical machine to another, and can mitigate overload conditions and enables uninterrupted maintenance activities. The focus of this article is to present the details of virtual machine migration techniques and their usage toward dynamic resource management in virtualized environments. We outline the components required to use virtual machine migration for dynamic resource management in the virtualized cloud environment. We present categorization and details of migration heuristics aimed at reducing server sprawl, minimizing power consumption, balancing load across physical machines, and so on. We conclude with a discussion of open research problems in the area.

INTRODUCTION

Cloud computing [1] provides a “computing-as-a-service” model in which compute resources are made available as a utility service — an illusion of availability of as much resources (e.g., CPU, memory, and I/O) as demanded by the user. Moreover, users of cloud services pay only for the amount of resources (a “pay-as-use” model) used by them. This model is quite different from earlier infrastructure models, where enterprises would invest huge amounts of money in building their own computing infrastructure. Typically, traditional data centers are provisioned to meet the peak demand, which results in wastage of resources during non-peak periods. To alleviate the above problem, modern-day data centers are shifting to the cloud. The important characteristics of cloud-based data centers are:

- Making resources available *on demand*. The operation and maintenance of the data center lies with the cloud provider. Thus, the cloud model enables the users to have a computing environment without investing a

huge amount of money to build a computing infrastructure.

- *Flexible resource provisioning*. This provides ability to dynamically scale or shrink the provisioned resources as per the dynamic requirements.
- *Fine-grained metering*. This enables the “pay-as-use” model, that is, users pay only for the services used and hence do not need to be locked into long-term commitments.

As a result, a cloud-based solution is an attractive provisioning alternative to exploit the “computing-as-service” model.

However, implementing cloud-based data centers requires a great deal of flexibility and agility. For example, the dynamic scaling and shrinking requirement needs *compute resources* to be made available at very short notice. When computing hardware is overloaded, it may be required to dynamically transfer some of its load to another machine with minimal interruption to the users. *Virtualization technology* can provide these kinds of flexibilities.

In a cloud environment, the service provider would like to operate the computing resources at optimum utilization levels to meet the service level agreements (SLA) of users. Overcommitment of resources can result in SLA violations, whereas underutilization of resources would mean loss of revenue for the provider. Thus, efficient *resource management* is a very critical component in cloud-based solutions.

Virtualization is a popular solution that acts as a backbone for provisioning requirements of a cloud-based solution. Virtualization provides a “virtualized” view of resources used to instantiate virtual machines (VMs). A VM monitor (VMM) or hypervisor manages and multiplexes access to the physical resources, maintaining isolation between VMs at all times. As the physical resources are virtualized, several VMs, each of which is self-contained with its own operating system, can execute on a physical machine (PM). The hypervisor, which arbitrates access to physical resources, can manipulate the extent of access to a resource (memory allocated or CPU allocated to a VM, etc.).

The decoupling between physical and virtual resources provided by the hypervisor enables flexibility of resource provisioning for VMs. This

decoupling also yields efficient state (more correctly memory state) capture of VMs, which enables migration and restoration of virtual machines across physical machines.

A VM, like a PM, has associated resource levels of CPU, memory, and input/output (I/O) devices. While instantiating VMs, each machine needs to be provisioned/allocated these resources. These resources can be overcommitted, and multiplexing them across VMs is the hypervisor's responsibility. Typically, a "sizing" process, based on resource-usage profiles of applications or estimations to meet load requirement and so on, is used to determine initial provisioning-levels of a VM. Changes in workload conditions of VMs can lead to "hot spots" — not enough resources provisioned to meet demand — or "cold spots" — provisioned resources are not utilized efficiently. In each case, resource allocations are varied dynamically to address the situation under consideration.

In this article, we discuss the use of VM migrations [2] for dynamic resource management in virtualization-based cloud systems. As mentioned earlier, migration is the process of transferring the state (all memory pages) of a VM from one physical machine to another. Different techniques for live migration exist — *suspend-and-copy*, *pre-copy* and *post-copy*. Suspend-and-copy, suspends a VM, copies all its pages and then resumes the VM on the target machine. The pre-copy approach [2] (as shown in Fig. 1) transfers pages iteratively to the target machine without suspending the VM (and hence is live). Once "sufficient" pages are transferred, the VM is suspended at the source and remaining state transferred to the target machine. While suspend-and-copy minimizes migration time, the downtime is proportional to the "size" of the VMs and network resources available for state transfer. Live migration techniques, aim to minimize downtime, by either copying pages (pre-copy) before a VM is suspended for final state transfer or copying minimal state (post-copy) to start the VM and using demand-paging over the network to fetch the remaining state. While both pre-copy and post-copy techniques differ in overheads and migration time trade-offs, they provide *live migration* semantics, so the VMs have minimal downtime and execute during the migration process.

DYNAMIC PROVISIONING USING VIRTUAL MACHINE MIGRATION

Virtual machine migration is a key enabler for dynamic resource management in cloud-based systems. Figure 2 depicts the important components of resource management from a cloud provider's perspective. Virtual machine sizing, the first step toward deploying a VM, is to determine the expected resources required on deployment. Once these levels are determined, a provisioning step determines where to "place" the VM and instantiate it. Once a VM is instantiated, a resource monitoring engine tracks the resource usage and performance indicators related to the (applications of the) VM. Under dynamic workload conditions, VMs can experi-

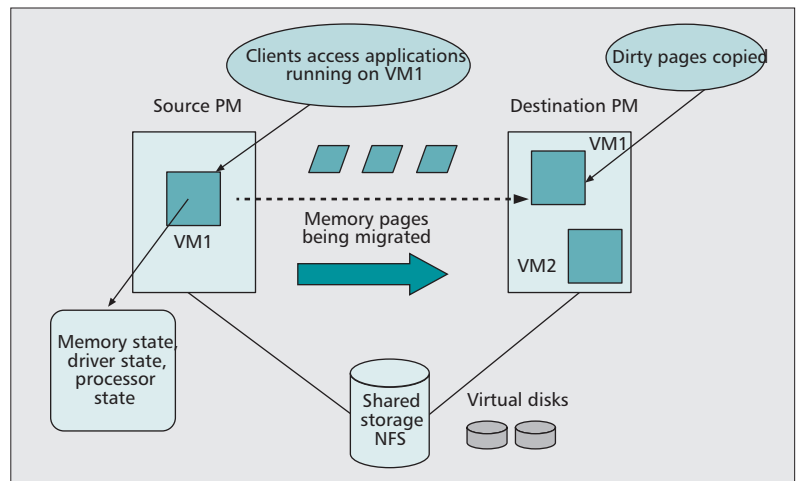


Figure 1. Live VM migration procedure.

ence "hot spots" (inadequate resources to meet performance demands) and "cold spots" (over-provisioned resources with low utilization). Moving VMs in order to allocate more resources (to alleviate hot spots) or consolidate VMs on fewer PMs to tackle cold spots is enabled through migration. From a cloud provider's perspective, alleviating hot spots is essential to meet SLAs with clients and tackling cold spots to use resources (including power consumption) efficiently.

Figure 3 shows two conditions: load balancing and consolidation of VMs based on migration. In the first case, either the goal is to distribute "load" evenly across PMs, or a VM needs more resources and hence is migrated to another PM. With consolidation, machines are migrated to fewer PMs to reduce *server sprawl*. In the next section, we discuss the goals, issues, and techniques related to such migration-enabled dynamic resource provisioning scenarios.

A cloud provider's resource management actions toward simultaneously minimizing resource usage and maximizing SLA adherence can be classified as follows:

Server consolidation: The goal of consolidation is to avoid server sprawl — many PMs host low-resource-usage VMs. As shown in Fig. 3, VMs on lightly loaded hosts can be "packed" onto fewer machines to meet resource requirements. The freed-up PMs can either be switched off (to save power) or represent higher-resource-availability bins for new VMs.

Load balancing: The goal of load balancing is to avoid a situation where there is a large discrepancy in resource utilization levels of the PMs (refer to Fig. 3). A desired scenario could be to have equal residual resource capacity across PMs (to help increase local resource allocations during increase demands). Virtual machine migrations can be employed to achieve this balance.

Hotspot mitigation: Active resource and application-level monitoring of VMs if required to identify hot spot conditions in which a VM has inadequate resources to meet its SLA requirements. Under such conditions, additional resources can be allocated either locally (on the

If the resource utilization shows an increase for some time window, then it is likely that it may result in a hotspot in the future. Such time-series analysis based techniques help in avoiding hotspots even before they occur.

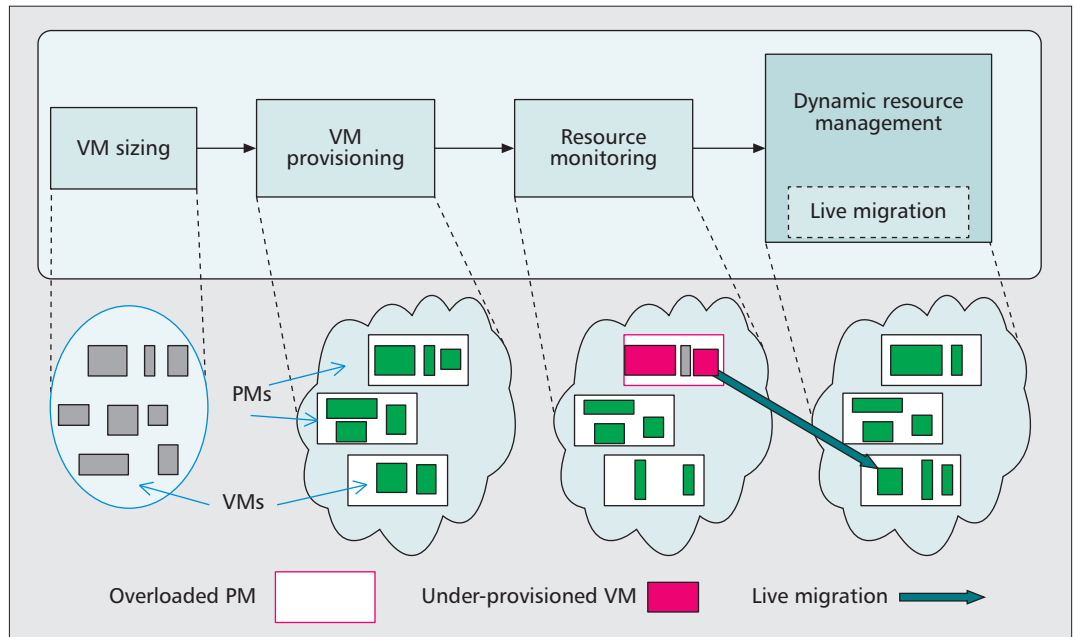


Figure 2. Different steps in resource management.

same PM) or within the set of PMs available for provisioning. When local resources are not sufficient to remove the hot spot, VMs can be migrated to another host to make the resources required available to mitigate the hot spot.

HEURISTICS FOR RESOURCE MANAGEMENT USING MIGRATION

For each of the three goals — consolidation, hotspot mitigation and load balancing — VM migration-based heuristics need to address three important questions:

- When to migrate
- Which VMs to migrate
- The set of destination host machines for migration

Table 1 summarizes the high-level goals while answering these questions for each of the desired goals. Migration heuristics address each of these questions based on several constraints: overhead of the migration process, impact on applications during migration, degree of improvement in intended goals of performance of resource utilization, and so on.

WHEN TO MIGRATE?

There are many situations when migration of VMs becomes necessary to maintain the overall efficiency of the data center. These situations or triggers are shown in Fig. 4 and discussed next.

Periodic — The migrations in a data center can be triggered periodically. For example, data centers in one part of world may be heavily used in daytime (9 a.m. to 9 p.m.), whereas they may be underloaded during the night. Such “time of day” based migration of VMs ensures that VMs are “near” clients, and the communication delays and overheads are minimized. Migrations can

also be done periodically to consolidate the reduced loads.

Due to Hot Spot — A hot spot is the overloaded condition of a PM. It can also be defined as the state when performance of a system falls below the minimum acceptance level. Detection of a hot spot can be done both proactively and reactively. Proactive hot spot detection techniques predict the occurrence of a hot spot by analyzing the trends in resource utilizations of the VM. If the resource utilization shows an increase for some time window, it is likely that it may result in a hot spot in the future. Such time-series analysis-based techniques help avoid hot spots even before they occur. One such technique to predict CPU utilization is discussed in [3]. More sophisticated proactive techniques analyze the request arrival rates. Increase in request arrivals suggests that the VM will require more resources to fulfill them, thus causing a potential hot spot. Reactive hot spot detection techniques use more direct techniques like observing the page thrashing rate, CPU and memory utilization levels, and so on. Hot spots can be locally mitigated if enough capacity is available at the host PM. Extra resources can be allocated to the VM showing signs of overload. When extra capacity is not available locally, migration is the only option available.

Excess Spare Capacity — Low utilization of PMs results in resource wastage. An optimum level of utilization is required to be maintained for the efficient working of a data center. Physical machines that have excess spare capacity (i.e., low resource utilization) cause overall inefficiency in the data center. At the level of a PM, the hypervisors have monitoring tools, similar to normal operating systems, which can provide the utilization information of different resources for that machine. Resource utilization levels of PMs

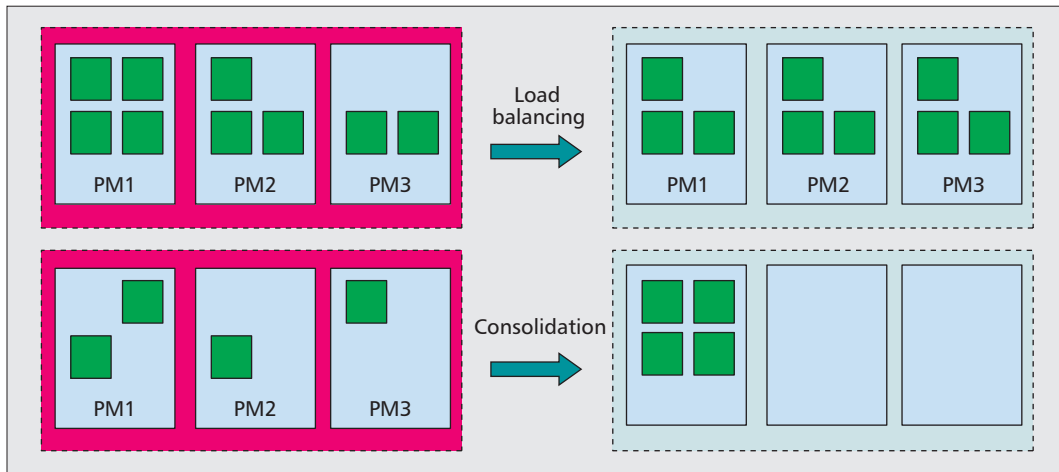


Figure 3. Load balancing and consolidation scenarios.

A new PM can be used to offset the load of an overloaded PM by migrating VMs from the latter to the former. Similarly, hosting new VMs may result in future overloads of some PMs, which again require migrations to be triggered.

across a data center are continuously monitored, and whenever the utilization levels fall below a certain threshold, migrations can be triggered. When a number of PMs are underutilized, VMs are migrated from such machines to make them completely free. Such “freed” PMs can then be shut down to save power, which results in consolidation.

Load Imbalance — Virtual machines change their resource requirements dynamically. This dynamism leads to imbalances in the resource utilization levels of different PMs. Some PMs can get heavily loaded while others may be lightly loaded. In a data center, resource utilization levels of PMs are monitored continuously. If there is large discrepancy in the utilization levels of different PMs, load balancing is triggered. Load balancing involves migration of VMs from highly loaded PMs to low loaded ones. An overloaded PM is undesirable as it causes delays in service of user requests. Similarly, the PMs that are lightly loaded cause inefficient resource utilization.

Addition/Removal of Virtual Machines and Physical Machines — Virtual machines and PMs can be added and removed in a virtualization-based data center. Addition/removal of VMs and PMs affects the availability of the resources and may require a change in the placement plan of VMs. A new PM can be used to offset the load of an overloaded PM by migrating VMs from the latter to the former. Similarly,

hosting new VMs may result in future overloads of some PMs, which again require migrations to be triggered.

WHICH VIRTUAL MACHINE TO MIGRATE

Selecting one or more VMs for migration is a crucial decision of the resource management heuristic. The migration process not only makes the VM unavailable for a certain amount of time but also consumes resources like network and CPU on source and destination PMs. Performance of other VMs that are hosted on source and destination PMs are also affected due to increased resource requirements during migration. Some VM selection approaches are straightforward and only consider the VM that is resource constrained (e.g., in a hot spot); other approaches employ a more holistic approach where all the VMs on a PM are considered before selecting the candidate VM. Generally, the aim of VM selection is to minimize the migration effort.

Resource Constrained Virtual Machine — This is the easiest way to select the candidate VM for migration. The VM whose resource requirements cannot be locally fulfilled is selected for migration. During hot spots, it is easy to find the most loaded VMs; hence, this simple selection can work. However, in operations like consolidation and load balancing, where the cause is not a single VM, the choice is not straightforward.

Goals	When to migrate	Which VM to migrate	Where to migrate
Server consolidation	Large number of underutilized PMs	VMs from the lightly loaded PMs	More loaded designated PMs
Load balancing	Imbalance in the load of PMs	VMs from the overloaded PMs	Lightly loaded PMs
Hotspot mitigation	Resource requirements of VMs not locally fulfilled	Set of VMs from the hotspot-host	PMs which can accommodate the incoming VMs without causing new hotspots

Table 1. Dimensions of dynamic resource management heuristics.

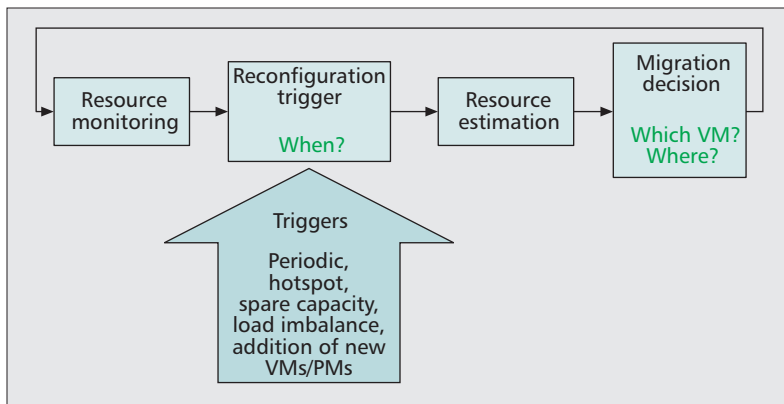


Figure 4. Steps for migration based dynamic resource management.

Holistic Approach — During hot spots, it may not always be efficient to select the overloaded VM for migration. Consider a case where the VM facing a resource crunch is utilizing a large amount of memory. In such a case the time and effort required for migration will be high. Instead, if a comparatively smaller memory VM is selected for migration, the time required will be less (assuming that the freed memory is sufficient to mitigate the hot spot). Also, the memory freed by the smaller VM can then be allocated to the larger VM. Such an approach of VM selection requires a holistic view of all the VMs and PMs present in the system in terms of their resource needs and availability, respectively. The VMs can be arranged in order of their resource utilizations, and a suitable sized VM can then be selected for migration. The decision also depends on the availability of a destination PM that has enough resources available. We can see that such a holistic approach requires quantification of resource requirement of VMs and PMs in order to compare them. In other words, they need to differentiate between two VMs (or PMs) on the basis of utilizations on multiple resource dimensions. As there are multiple resource types (e.g., CPU, MEM, I/O), it becomes difficult to directly compare the resource requirements of different machines. Generally, a function of different resource types is used for comparison. There are many such functions proposed in the literature. One such function is volume. It is the product of the individual resource utilizations. A variant of the function “volume” is used in the scheme mentioned in [3] and is calculated as

$$volume = \frac{1}{1-cpu} * \frac{1}{1-mem} * \frac{1}{1-net}$$

where *cpu*, *mem*, and *net* are normalized resource utilization values. Similarly, some schemes mentioned in [4, 5] use a function of different resource types, which is a vector. Some other functions of resources like *weighted sum* and *maximum among required resources* are also used. Different types of resource functions and their desired properties are discussed in [5].

Affinity based — These heuristics also incorporate other objectives instead of considering the resource requirement only. For example,

some *affinity-aware* migration techniques consider communication costs among VMs while performing migration. For instance, if two VMs are communicating with each other, it is better to host them on the same PM. This will reduce the overall communication cost among the VMs by reducing network usage. Similarly, memory sharing between VMs can also affect the VM selection for migration. Migrating a VM to a PM where it can share memory with other hosted VMs can result in effective memory usage [6]. Virtual machines, which share memory, can be migrated together with less effort as similar-content memory pages are required to be transferred only once. Such a scheme is known as *gang scheduling* of VMs. The approach is to proactively track the identical contents of collocated VMs and transfer those contents only once while migrating all those VMs simultaneously to another PM. This method optimizes both memory and network overhead of migration. Such mechanisms can be fruitful when an entire rack of servers have to be evacuated and all the collocated VMs running on them have to be shifted to a different location. A detailed description of gang migration is given in [7].

WHERE TO MIGRATE

During migration, the destination PM should have enough resources so that it can support the incoming migrating VM. Here we discuss factors for selecting a PM as a destination for a migrating VM.

Depending on Available Resource Capacity

— Only considering the availability of resources at the destination is not enough. Some other factors also need to be taken into consideration, such as whether the destination is a *best fit* (leaving minimum remaining resources) for the migrating VM, how will the performance of VMs that are already hosted on destination PM get affected. The destination selection to minimize waste of resources is a field of research in itself. The schemes proposed use heuristics of bin packing and vector packing problems ([5, 8] describe these problems) for destination selection since the optimal placement solution is intractable. For example, in vector-based destination selection, vector arithmetic like dot product is performed on resource vectors to find the best fit. A good discussion on such heuristics can be found in [3–5]. Virtual machines and PMs are sorted in some order based on their resource requirements, and then the First Fit or Best Fit scheme is applied to select the most suitable PM.

Depending on Affinity of Virtual Machines

— Apart from selecting PMs solely on the basis of resource availability, some schemes try to leverage the relations (or affinity) between the VMs to identify a suitable host PM. For example, a scheme mentioned in [6] tries to achieve consolidation by collocating VMs that have high memory sharing potential. Periodically, based on memory fingerprints of VMs, best matches of hosts for VMs can be found and migrations can be triggered. This scheme is called *memory-aware* migration. The VM can be remigrated if

some other VM on some other PM becomes a better memory sharing partner. The overhead of migration is taken into consideration. The rationale behind this method is that VMs that can share part of their memory will require less overall memory than VMs that do not share memory. Similarly, if two VMs, hosted on different PMs, communicate heavily, one of the VMs can be migrated to the PM where its communicating partner is hosted.

WIDE-AREA VIRTUAL MACHINE MIGRATION

The discussion regarding live migration so far has considered resource management within a single (local) virtualization-based hosting setup. In such a setting, the VMs (and PMs) are assumed to be located within the same local area network. In a cloud-based provisioning scenario this assumption may not always hold; VM hosting data centers can be spread around the world, or enterprises with offices worldwide may have private data centers at several locations. In such cases, migrating VMs over wide area networks poses a set of different and interesting challenges.

The main differences between VM migration over local and wide area networks are as follows.

Migrating storage: Machines within a local network connect to shared storage system using mechanisms such as NFS, NAS, and storage area networks (SANs). In such a setting, since access to disks is primarily via the network, migration of VM is accomplished by migrating the memory state. In a wide-area scenario, the shared storage assumption seldom holds. As a result, wide-area migration entails not only transferring memory state, but more important, the state of local disks. With the lack of high end-to-end network bandwidth over wide area networks and the potential transfer of large storage state, the downtime and migration time are expected to be high in such scenarios.

Network reconfiguration: Unlike migration within a local area network, crossing network boundaries results in network reconfiguration. Moving into a new subnet forces the machine to get a new IP address and, as a result, breaks existing network connections. Either the network addresses have to be preserved or applications need to be made aware of the network reconfiguration semantics.

Figure 5 depicts the local-area and wide-area migration scenarios. While VMs assume availability of networked storage systems in local networks, this condition is hard to recreate over wide area networks. The limited end-to-end bandwidth makes such an equivalent exercise futile. Furthermore, IP reconfiguration will not support state-full network protocols.

The issue of IP reconfiguration can be handled in a few ways. Bradford *et al.* [9] depend on DNS-resolutions to do the job. When VMs migrate, they maintain their canonical names, and the new IP address is registered with the name server. Lookups for the VM based on the canonical name, subsequent to migration, will resolve to the new (correct) IP address. This seamless change in original IP address and resolution of new IP address while the VM migrates across different networks is done through IP

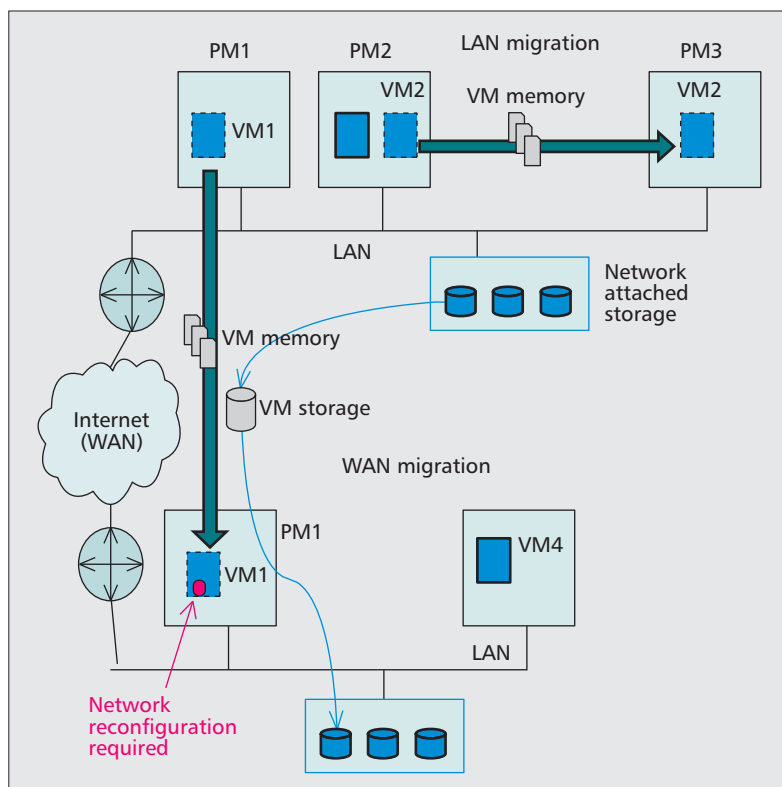


Figure 5. Virtual machine migrations over LAN and WAN.

tunneling. Tunneling is a mechanism of providing a path across networks/LANs of different IP configurations by taking help from the gateways encountered on the way to the destination network (where the designated host resides). Gateways provide tunnel endpoints, preventing any intermediate loss of connectivity. Note that this solution places the burden on managing endpoints on the applications (i.e., they need to be aware of the IP address change). An alternative solution to address the reconfiguration issue is to address the problem in the network protocol stack. Cloudnet [10] employs a combination of layer 3 virtual private networks (VPNs) and layer 2 virtual private LAN service (VPLS) to provide end-to-end routing across multiple networks and bridge LANs at different locations. The unified virtual network provides the view of a LAN to migrating VMs, with VMs maintaining single IP addresses.

The second component of migrating disks is more time consuming due to limited network bandwidth and high-latency wide-area links. Schemes based on proactive storage replication and use of content-based hashing (for redundancy elimination) [10] can be employed to speed up storage migrations. Nevertheless, since storage volumes are large, either the cost for WAN-migrations has to be proactively (with continuous storage replication) or reactively (on-demand transfer of disks state) borne.

In spite of the costs related to storage migrations and the overheads of network reconfiguration, WAN migrations are useful in several cases. Large data centers or enterprises having computing infrastructure around the world migrate VMs to follow the sun. Machines

With the increase in the popularity of cloud computing systems, virtual machine migrations across data centers and diverse resource pools will be greatly beneficial to data center administrators. Live virtual machine migration is an indispensable tool for dynamic resource management in modern-day data centers.

migrate to locations during certain times of the day to be “closer” to users in a geographic region. Another application is in use of maintenance and upgrades. Machines can be moved from one site to the other, keeping services on them alive, while maintenance operations process on other sites. Collaborative projects across research groups spanning different cities can derive benefit through WAN migration of the VM containing the project instead of maintaining consistent replicas. In cloud bursting scenarios, where the capacity of a data center is saturated, additional infrastructure, even if geographically far, can provide secure resources of VMs. Wide-area-based migrations can be useful in these conditions as well.

CONCLUSION

In this article, we discuss the important role of live virtual machine migration in dynamic resource management of virtualized cloud systems. Migration enables several resource management goals like consolidation, load balancing, and hot spot mitigation. Researchers have leveraged live virtual machine migration to come up with efficient resource management mechanisms. We discussed the components — when to migrate, which VM to migrate, and where to migrate — and approaches followed by different heuristics to apply migration techniques for goals of consolidation and hot spot mitigation. Challenges with wide-area migration — storage migration and network reconfiguration — and some techniques to enable it were also discussed. With the increase in the popularity of cloud computing systems, virtual machine migrations across data centers and diverse resource pools will be greatly beneficial to data center administrators. Live virtual machine migration is an indispensable tool for dynamic resource management in modern-day data centers.

REFERENCES

- [1] M. Armbrust et al., “A View of Cloud Computing,” *Commun. ACM*, vol. 53, no. 4, 2010, pp. 50–58.
- [2] C. Clark et al., “Live Migration of Virtual Machines,” *Proc. 2nd Conf. Symp. Networked Systems Design & Implementation*, vol. 2, USENIX Association, 2005, pp. 273–86.
- [3] T. Wood et al., “Black-Box and Gray-Box Strategies for Virtual Machine Migration,” *Proc. 4th Conf. Symp. Networked Sys. Design & Implementation*, 2007.
- [4] A. Singh et al., “Server-Storage Virtualization: Integration and Load Balancing in Data centers,” *Proc. 2008 ACM/IEEE Conf. Supercomputing*, IEEE Press, 2008, pp. 1–12.

- [5] M. Mishra and A. Sahoo, “On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach,” *Proc. 4th Int'l. Conf. Cloud Computing*, 2011, pp. 275–82.
- [6] T. Wood et al., “Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers,” *Proc. ACM SIGPLAN/SIGOPS Int'l. Conf. Virtual Execution Environments, VEE*, 2009, pp. 31–40.
- [7] U. Deshpande, X. Wang, and K. Gopalan, “Live Gang Migration of Virtual Machines,” *High-Performance Parallel and Distributed Computing*, June 2011.
- [8] R. M. Karp, M. Luby, and A. Marchetti-Spaccamela, “A Probabilistic Analysis of Multidimensional Bin Packing Problems,” *Proc. 16th Annual ACM Symp. Theory of Computing*, 1984, pp. 289–98.
- [9] R. Bradford et al., “Live Wide-Area Migration of Virtual Machines Including Local Persistent State,” *Int'l. Conf. Virtual Execution Environments (VEE)*, 2007.
- [10] T. Wood et al., “CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines,” *Int'l. Conf. Virtual Execution Environments*, 2011.

BIOGRAPHIES

MAYANK MISHRA [StM] (mayank@cse.iitb.ac.in) received a B.E. degree in information technology from RGPV Bhopal in 2004 and an M.Tech. degree from Indian Institute of Technology Bombay in 2006. He is currently pursuing a Ph.D. degree at the Indian Institute of Technology Bombay. He has worked as a software engineer in the Wireless Networking Business Unit of Cisco Systems, Bangalore from 2007 to 2010. His research interests include cloud computing, data center networking, and computer networks.

ANWESHA DAS (anwasha@cse.iitb.ac.in) has been a computer science M.Tech. student at the Indian Institute of Technology, Bombay since 2009. Prior to this she was an intern at the Institute of Mathematical Sciences, Chennai. She received her B.Tech. degree in computer science engineering from West Bengal University of Technology, Kolkata in 2008. Her research interests include networked systems and cloud computing with emphasis on leveraging the benefits of virtualization.

PURUSHOTTAM KULKARNI (puru@cse.iitb.ac.in) received a B.E. degree in computer science from the University of Pune, India, in 1997 and an M.S. degree in computer science from the University of Minnesota, Duluth, in 2000. He received his Ph.D. degree in computer science from the University of Massachusetts, Amherst in 2006. Currently, he is an assistant professor at the Department of Computer Science and Engineering, Indian Institute of Technology Bombay. His research interests are in the areas of virtualization-based systems, computer networks, distributed systems, and technology solutions for developing regions.

ANIRUDHA SAHOO [M] (sahoo@cse.iitb.ac.in) received a B.Sc. (Engg.) degree in electrical engineering from NIT, Rourkela, and M.S. and Ph.D. degrees in computer science from the University of Louisiana and Texas A&M University, respectively. He is currently an associate professor in the Department of Computer Science and Engineering, Indian Institute of Technology Bombay. Prior to joining IIT Bombay, he worked as a software engineer in Intergraph Corporation, Alabama, for five years, and then as a senior software engineer at Cisco Systems, California, for five years.