IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-6, Special Issue-2, March 2018 E-ISSN: 2347-2693

Resource Provisioning for Ensuring QoS in Virtualized Environments

L.Jayasimman[1] , B.Geetha Dhanalakshmi[2*]

[1]Department of Computer Science,Srimad Andavan Arts & Science College, Trichy -5
[2] Department of Computer Science,Srimad Andavan Arts & Science College, Trichy -5

Available online at: www.ijcseonline.org

Abstract- Live VM migration help attain both cloud-wide load balancing and operational consolidation while the migrating VMs remain accessible toward users. To avoid period of high-load for the complicated resources, IaaS-cloud operators assign specific time windows for such migrations to occur in an orderly manner. Moreover, provider normally relies on share-nothing architectures to get scalability. In this paper, we focus on the immediate scheduling of live VM migrations in large share-nothing IaaS clouds, such that migration are complete on time and without adversely affecting agreed-upon SLAs. We offer a scalable, distributed network of brokers that oversees the progress of all on-going migration operations within the context of a provider. Brokers make use of an fundamental exceptional purpose file system, termedMigrateFS, that is capable of both replicating and keeping in sync virtual disks while the hypervisor live-migrates VMs (i.e., RAM and CPU state). By restrictive the resources consumed during migration, brokers implement policies to reduce SLA violations while seeking to complete all migration tasks on time.

Keywords-Distributed Systems, Cloud Computing, IaaS Clouds, Virtual Machine Migration

INTRODUCTION

Large IaaS cloud providers offer high quality services by constantly adjusting resource usage and balancing the load in their infrastructure. Since IaaS providers predominantly offer virtual machines (VMs), load-balancing is usually achieved through VM migration, i.e., transferring a VM from one physical machine (PM) to another. VM movement helps offload congested physical nodes, can enhance the utilization of the underlying resources, and can ultimately achieve an improvement in the quality of provided services. In this paper, we focus on live VM migrations for IaaS cloud providers that rely on a share-nothing infrastructure. In a share-nothing infrastructure, PMs use their resources (e.g., memory, disk storage) independently, instead of accessing common resources via synchronization layers (e.g., common storage layer). The main advantage of share nothing infrastructures is scalability, as less synchronization means fewer bottlenecks (more in Section. A live VM migration is performed while the migrating VM remains on–line and involve a short downtime hardly noticeable by users interact with the VM. Practically, in a share-nothing infrastructure, a live VM migration require that a virtual disk on the order of multiple GBytes be transferred from the source to target PM (used for hosting the VM), while the VM remains fully accessible to its users. Furthermore, VM migrations should not coincide with high-load periods for the involved VMs and network resources. Window for each migration to take place. Failing to complete a migration within the time window will most likely degrade the QoS experienced by the affected users and may lead to a number of Service Level Agreement (SLA) violations.

The problem we study in this paper is the real–time scheduling of live VM migration tasks in share-nothing IaaS-clouds. Given a new PM host and a time window for each VM migration (decided by a cloud reallocation policy), a real–time scheduling mechanism must: control the resources allocated to each migration task based on the QoS degradation and the SLA violations that any affected VM may experience. For example, the network bandwidth consumed by a migration task can lead to an SLA violation for a VM hosted on the migration's target PM: an efficient scheduler must adjust the network resources allocated to the migration task, to avoid such SLA violations. limit the migration side-effects experienced by the users of a migrating VM. Ideally, there should be no restrictions on the usage of a migrating VM. However, letting a migrating VM constantly write on blocks that need to be re-transferred to the target PM, can extend the migration's duration beyond its assigned time window. To prevent a migration from extending beyond its time window, a real-time scheduler may have to limit the VM´s write rate. An efficient scheduler should take into account the tradeoff between side-effects and the migration's duration and be as non-interventional as possible while still preventing a migration from running beyond its window. • prioritize concurrent migrations and impose limitations that minimize the overall cost, taking into account potential SLA violations and implications on the QoS.Focused on synchronizing individual virtual disks. In contrast, we focus on the multiple simultaneous migrations setting, as discussed above, and we leverage results from prior work to form the foundation of our approach. Our approach consists of Migrates, a low-level special purpose file system, and higher-level resource allocation policies, designed to accommodate large numbers of simultaneous migration tasks.

The Migrates file system runs on every PM: instances of Migrates communicate over the network and jointly control the transfer of a virtual disk image between any two PMs. Migrates continuously adjusts the consumption of both VM and migration resources based on hints provided by performance monitoring tools. In particular, MigrateFS tunes two rates during VM–disk shipment: a) disk throughput available to the VM's internal processes that access the virtual disks during migration, and b) network throughput used for the purposes of migration. In this way, Migrates is able to accurately estimate the completion time of a migration. Estimates also allow Migrates to delay a migration (while still completing it before the end of the assigned time window) when the cloud experiences heavy workloads. Resource allocation policies are implemented by a coordinating Migrations Scheduler and a distributed network of Brokers. Essentially, resource allocation policies allow for prioritization of migration tasks while taking into account the network status so that "hot" physical network links are not further stressed by virtual disk shipments. Brokers apply such policies and drive the operation ofMigrateFS instances by indicating how the network and disk throughput must be restricted, in each case. Our evaluation, based on both a Migrates prototype and simulation of large infrastructures stress on saturated PMs during migration. The main cost in our approach comes from the need to keep track of I/O operations. In our evaluation, we show ofMigrateFS is outperformed by local file systems, yet it is more efficient than network storage solutions that enable VM migration. For large cloud providers, there are cases where the SLAs of different VMs vary widely: violating the SLA of one VM will impose a financial cost that may be orders of magnitude greater if compared to the SLA.

ISSUES AND CHALLENGES

The problem we study in this paper is the real–time scheduling of live VM migration tasks in share-nothing IaaS-clouds. To control the resources allocated to each migration task, based on the QoS degradation and the SLA violations that any affected VM. Limit the migration side-effects experienced by the users of a migrating VM. There should be no restrictions on the usage of a migrating VM. It can extend the migration's duration beyond its assigned time window.

MOTIVATION

We first discuss suspend-resume and live VM migration and then we offer an overview of how different cloud architectures support VM migration. In the end of this section, we describe how the proposed approach addresses the high-level challenges in supporting large-scale live VM migration.

RESEARCH OBJECTIVES

This research will provide new techniques to ensure security in Cloud Computing data. The objectives of proposed work are formulated as below:

- In this propose a simplified, yet general, model that quantifies the cost different SLA violations entail.
- This model inspires a cost-driven resource allocation policy that runs across the distributed network of Brokers.
- The baseline approach that is unaware of the differences among SLAs.
- It provides an overview of VM migration in IaaS clouds.
- It is empowered by ourMigrateFS; the latter offers the means to control resource consumption.

RELATED WORKS

In [1] C. Weng, M. Li, Z. Wang, and X. Lu et al, System virtualization can aggregate the functionality of multiple standalone computer systems into a single hardware computer. It is significant to virtualized the computing nodes with multi-core processors in the cluster system, in order to promote the usage of the hardware while decrease the cost of the power. In the virtualized cluster system, multiple virtual machines are running on a computing node. However, it is a challenging issue to automatically balance the workload in virtual machines on each physical computing node, which is different from the traditional cluster system's load balance. In this paper, we propose a management framework for the virtualized cluster system, and present an automatic performance tuning strategy to balance the workload in the virtualized cluster system load balance. In this paper, we propose a management framework for the virtualized cluster system, and present an automatic performance tuning strategy to balance the workload in the virtualized cluster system. We implement a working prototype of the management framework (VEMan) based on Xen, and test the performance of the tuning strategy on a virtualized heterogeneous cluster system. The experimental result indicates that the management framework and tuning strategy are feasible to improve the performance of the virtualized cluster system.

In [2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, And A. Warfield et al,A datacenter to balance the load, save energy or prepare production servers for maintenance. Although VM placement problems are carefully studied, the underlying migration schedulers rely on vague adhoc models. This leads to unnecessarily long and energy-intensive migrations. We present mVM, a new and extensible migration scheduler. To provide schedules with minimal completion times, mVM parallelizes and sequentializes the migrations with regards to the memory workload and the network topology. MVM is implemented as a plugin of BtrPlace and its current library allows administrators to address temporal and energy concerns. Experiments on a real testbed show mVM outperforms state-of-the-art migration schedulers. Compared to schedulers that cap the migration parallelism, mVM reduces the individual migration duration by 20.4% on average and the schedule completion time by 28.1%. In a maintenance operation involving 96 VMs migrated between 72 servers, mVM saves 21.5% Joules against BtrPlace. Compared to the migration model inside the cloud simulator Clouds, the prediction error of the migrations duration is about 5 times lower with mVM. By computing schedules involving thousands of migrations performed over various fat-tree network topologies, we observed that the mVM solving time accounts for about 1% of the schedule execution time.

In [3] Z. Liu, W. Qu, W. Liu, and K. Li, "Xen et al, The increasing numbers of technology areas using Virtual Machine (VM) platforms, challenges exist in Virtual Machine migrating from one physical host to another. However, the complexity of these virtualized environments presents additional management challenges. Unfortunately, many traditional approaches may be either not effective well for reducing downtime or migration time, or not suitable well for Xen VMs platforms. This paper presents the design and implementation of a novel Slowdown Scheduling Algorithm (SSA) for Xen live VM migration. In our SSA methodology, the CPU resources which have been assigned to migration domain are decrease properly. That is, the dirtying page rate is reduced according to the decrease of CPU activity. Experimental results illustrate that our SSA approach can shorten both the total migration time and downtime obviously under high dirty page rate environment. The Virtual machine (VM) technology has seen a tremendous growth in recent years, which mainly due to its capabilities of isolating, consolidating and migrating workload. These have led to a data center to serve multiple users in a secure, flexible and efficient way. These technology shifts not only provide efficient and secure computing resource containers, but also can be migrated smoothly among multiple physical machines. VM running services can be migrated to other hardware platforms without disrupting client access.

In [4] W. Voorsluys, J. Bromberg, S. Venugopal, and R. Buyya et al, Virtualization technology has become commonplace in modern data centers and cluster systems, often referred as computing clouds". In particular, the capability of virtual machine (VM) migration brings multiple benefits such as higher performance, improved manageability and fault tolerance. Moreover, live migration of VMs often allows work- load movement with a short service downtime. However, service levels of running applications are likely to be negatively acted during a live VM migration. For this reason, a better understanding of its effects on system performance is highly desirable. In this paper, we present a performance evaluation on the effects of live.

migration of virtual machines on the performance of applications running inside Xen VMs. Results show that in most cases, migration overhead is acceptable but cannot be disregarded, especially in systems where service availability and responsiveness are governed by strict Service Level Agreements (SLAs). Despite that, there is a high potential for live migration applicability in data centers serving enterprise-class Internet applications. Our results are based on a workload composed of a real application, covering the domain of multi-tier Web 2.0 applications.

In [5] W. Voorsluys, J. Bromberg, S. Venugopal, and R. Buyya et al, Virtualization technology has become commonplace in modern data centers and cluster systems, often referred as computing clouds". In particular, the capability of virtual machine (VM) migration brings multiple benefits such as higher performance, improved manageability and fault tolerance. Moreover, live migration of VMs often allows work- load movement with a short service downtime. However, service levels of running applications are likely to be negatively acted during a live VM migration. For this reason, a better understanding of its effects on system performance is highly desirable. In this paper, we present a performance evaluation on the effects of live migration of virtual machines on the performance of applications running inside Xen VMs. Results show that in most cases, migration overhead is acceptable but cannot be disregarded, especially in systems where service availability and responsiveness are governed by strict Service Level Agreements (SLAs). Despite that, there is a high potential for live migration applicability in data centers serving enterprise-class Internet applications. Our results are based on a work- load composed of a real application, covering the domain of multi-tier Web 2.0 applications.

EXISTING ALGORITHM

PRIORITYBASEDMIGRATIONSMANAGEMENT

Limiting the network bandwidth is not based only on the danger threshold. We also limit the network usage rate if we detect a network contention, as indicated by the back off flag. This flag is set to true under two conditions: a) the migration process causes a network SLA failure of a running VM, b) the network bandwidth consumed should be given to another migration task that is about to violate its time constraint. The shouldBackOff function detects saturated network links on the path between the source and target hosting PMs.The Migrations Scheduler provides the VM shipment path upon the Broker’s instantiation. The Broker registers for notifications on saturated switches to the corresponding cloud monitoring tools. As soon as the Broker projects that the designated migration time-constraint will be violated, it needs to request other Brokers to release (if possible) some of the network bandwidth they occupy. As the Migrations Scheduler is aware of all migrating VM disks sharing network paths any Broker can exploit this information to notify only those Brokers with which it shares saturated network links. Since Brokers exchange messages directly with each other in a peerto-peer fashion there is no single message exchange hub. The distributed nature of Broker communication allows our approach to scale to the size of large cloud installations.

SET NETWORK LIMIT

It equipment a policy that require no communication with the additional consumers of the network bandwidth. Priority is given to VMs failing their SLAs and to migration tasks in danger of violating their time constraints. In this background, we decide on for a low-cost communication policy among *Brokers* as we target large cloud infrastructures. *Brokers* want only to announce the threat of violating the time constraint of a VM migration to a glowing specified subset of other *Brokers* so that the *shouldBackOff* call yields valid back-off requests.

MINIMIZE TOTAL COST ALGORITHM

We believe the transportation problem of influential nonnegative shipments from a set of m warehouse with given availabilities to a set of n markets with given requirements. Three objectives are defined for each explanation: (i) whole cost TC (ii) bottleneck time BT (i.e., utmost transport occasion for a positive shipment) and (iii) bottleneck shipment SB (i.e., total shipment over routes with tailback time). An algorithm is given for determining all competent (pareto-optimal or nondominated) (TC, BT) answer pairs. The special case of this algorithm when all the component cost coefficients are zero is shown to be the same as the algorithms for minimize BT provided by Szwarc and Hammer. This algorithm for minimize BT is shown to be computationally superior (take only about a second on the customary for transport or task troubles.

COST–DRIVEN BROKER

In our cost-driven policy, the network of Brokers is enhanced to obtain into explanation the economic penalties inflict by the resource restraint decisions. Cost-driven Brokers aim at cloud providers that can largely benefit from fine-tuning the migrations’ disk/network-rate restrictions, based on the different SLAs. To this end, we use a easy model suitable to any type of SLA and cloud infrastructure.

VM MIGRATION SCHEDULING ALGORITHMS

It Consisting of interoperating IaaS-clouds and SDNs and examine how our approach can be functional to PaaS-clouds so that real-time service migration is realize. We also intend to extend the SLA monitoring apparatus of our advance to take corrective action not only when the network is strained but also when disk I/O spikes. Finally, we shall investigate how our approach fares in conjunction with a resource–reallocation algorithm competent of determining VMs require migration so that cloud efficiency charge are further better.

VM MIGRATION, CLUSTERING ALGORITHM

Begin with the disjoint clustering having level L(0) = 0 and progression number m = 0. 2. Find the least dissimilar couple of clusters in the existing clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)], where the minimum is over all pairs of clusters in the current clustering. 3. Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)] 4. Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and article equivalent to the newly formed cluster. The closeness between the new cluster, denoted (r,s) and old cluster (k) is distinct in this way.

A RESOURCE–REALLOCATION ALGORITHM

Resource allocation in computing systems deals with the allocation of available system resources to the various tasks equipped to be executed. This is a process that extensively affects the generally performance of the system. Typically, resource allocation algorithms take as input a list of tasks or processes that are prepared to be executed at several particular times as provide by a system scheduler. The scheduler considers a mission stream graph in order to resolve task dependencies. Traditionally, reserve allocation in off chip multiprocessor systems concentrate on the portion of software tasks to every of the processor nodes (usually individual processors with narrow cache and memory), such that the largely performance of the system is maximized. This is a well-known problem, with a large amount of research contributions towards efficient utilization of the massive hardware parallelism available in such systems using various exact and heuristic approaches. In the case of a lot of center systems, significant on-chip constraints such as limited buffer capacity for on-chip message, on-chip network congestion, power compactness, and limited I/O bandwidth command the development of existing algorithms or even the expansion of new algorithms, in an effort to put together the emerging challenges.

Table 1: A Comparative Performance Evaluation on Different Algorithms

S.NO	NAME OF THE ALGORITHM	MERITS	DEMERITS	FOCUSING AREA
1.	**Priority Based Migrations Management**	Inaccessible from one another.	accessible from one another	The Migrations Scheduler provides the VM shipment path upon the Broker's instantiation.
2.	**Set Network Limit**	The danger of violating the time constraint of a vm migration.	Brokers as we target large cloud infrastructures.	A VM migration to a well specified subset of other Brokers so that the shouldBackOff call yields valid back-off requests.
3.	**Minimize Total Cost Algorithm**	An online algorithm is proposed for tackling the electricity cost minimization problem.	The heavy reliance on accurate prediction. To handle uncertainty or noisy data.	This algorithm when all the unit cost coefficients are zero is shown to be the same as the algorithms for minimizing BT provided by Szwarc and Hammer
4.	**Cost–Driven Broker**	Cost-driven Brokers aim at cloud providers that can largely benefit from fine-tuning the migrations'	The financial penalty inflicted by the resource restriction decisions.	We use a simple model applicable to any type of SLA and cloud infrastructure.

5.	**VM Migration Scheduling Algorithms**	VM scheduling algorithms are used to schedule the VM requests to the Physical Machines (PM) of the particular Data Center (DC)	The network is stressed in vm scheduling.	It is used for optimization of different factors like Time, Cost, Energy and Security.
6.	**VM migration, Clustering algorithm**	Find the least dissimilar pair of clusters in the current clustering	Selection of optimal number of clusters is difficult.	Measurable and efficient in large data.
7.	**A Resource–Reallocation Algorithm**	1. User does not expand software and hardware. 2. No limitation of place and medium	Cloud providers are not fully trusted by users.	Algorithms take as input a list of tasks or processes that are ready to be executed at some particular time as provided by a system scheduler.

CONCLUSION

Migrating VMs in live fashion is of key importance to IaaSclouds as it helps accomplish major operational and administrative objectives including effective load-sharing and improved utilization of physical machinery. The movement of VMs over the network inevitably consumes significant cloud resources, thus such tasks should be scheduled during periods of low load. In this work, we focus on emerging highly-scalable share-nothing cloud installations and employ on-demand virtual disk synchronization across PMs to attain live migration under explicit time-constraints. Our approach is empowered by the combined use of a network of Brokers and the Migrates file system. Migrates effectively synchronizes disk images between physical computing systems, while the Brokers manage the resources of the share-nothing cloud elements. The joint objective of the two components is to offer a scheme that gracefully deals with time-constrained VM migration requests and at the same time, does not deplete cloud resources. The resource management policies we developed apply on both clouds with uniform SLAs across VMs and clouds with widely varying SLAs. Our lightweight priority-based policy adjusts the network and virtual disk bandwidth in time using a simple, yet effective, protocol. In cases where the cloud provider needs to differentiate among the operation of different VMs, our cost-driven policy offers a general model to capture different costs and intelligently adjust the network and disk rates. Our prototype experimentation demonstrates the I/O performance gains compared to network storage solutions, and the significantly reducing SLA violations due to heavy network traffic. Moreover, the extension of the cost-driven policy offers a 2x improvement in the cases where it applies. In the future, we plan to examine statistics-driven VM migration scheduling algorithms in settings consisting of interoperating IaaS-clouds and SDNs and investigate how our approach can be applied to PaaS-clouds so that real-time service migration is realized. We also intend to extend the SLA monitoring mechanism of our approach to take corrective action not only when the network is stressed but also when disk I/O spikes. Finally, we shall investigate how our approach fares in conjunction with a resource–reallocation algorithm capable of determining VMs requiring migration so that cloud efficiency rates are further improved.

REFERENCES

[1] C. Weng, M. Li, Z. Wang, and X. Lu, "Automatic Performance Tuning for the Virtualized Cluster System," in *Proc. of the 29th IEEEInt. Conf. on Distributed Computing Systems*,Montreal, Canada, June 2009.

[2] VMware, "VMware DRS - Dynamic Scheduling of System Resources," www.vmware.com/products/drs/overview.html, Oct. 2009.

[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *Proc. of the 2nd Symposium on Networked Systems Design & Implementation*, Boston, MA, May 2005.

[4] Z. Liu, W. Qu, W. Liu, and K. Li, "Xen Live Migration with Slowdown Scheduling Algorithm," in *Proc. of the 2010 Int. Conf. on Parallel and Distributed Computing, Applications and Technologies*, ser. PDCAT '10, Wuhan, China, Dec. 2010, pp. 215–221.

[5] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of Virtual Machine LiveMigration in Clouds: A Performance Evaluation," in *Proc. of the 1st Int. Conf. on Cloud Computing (CloudCom'09)*, Beijing, China, Dec. 2009.

[6] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and IncrementalWhole-System Migration of Virtual Machines Using Block-Bitmap," in *Proc. of IEEE Int. Conf. on Cluster Computing*, Tsukuba, Japan, September 2008.

[7] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schi ¨oberg, "Live Wide-Area Migration of Virtual Machines Including Local Persistent State," in *In VEE '07: Proc. of the 3rd Int. Conf. on Virtual Execution Environments*, San Diego, CA, June 2007.

[8] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. van der Merwe, "CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines," *SIGPLAN Not.*, pp. 121–132, March 2011.

[9] A. Mashtizadeh, E. Celebi, T. Garfinkel, and M. Cai, "The Design and Evolution of Live Storage Migration in VMware ESX," in *Proc. of the 2011 USENIX Annual Technical Conference*, Portland, OR, 2011.

[10] D. Josephsen, *Building a Monitoring Infrastructure with Nagios*. Upper Saddle River, NJ: Prentice Hall PTR, 2007.