International Journal of Pure and Applied Mathematics
Volume 118 No. 24 2018
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue
http://www.acadpubl.eu/hub/

CLOUD COMPUTING BASED OPTIMIZING OF MULTIPLE VIRTUAL MACHINE USING LIVE MIGRATION

Mr.Lochan.B[1],Mr.Venkatagiri J[2]
Dr.Lokesh A[3]
[1,2]Asst.Professor, Dept of CSE,
[3]Assoc.Professor, Dept of ISE,
SVCE, Bangalore
lochanchirag2@yahoo.com, venkatajgiri@yahoo.co.in
lokeshyadav.ka@gmail.com

April 29, 2018

Abstract

Cloud computing paradigm is enabling disruptive and innovative services by allowing enterprises to lease computing by using network, storage and host from physical infrastructure owners. This shift infrastructure management responsibility has brought new challenges and new revenue models to Cloud providers. One of those challenges is to competent the migrate multiple virtual machines (VMs) within the hosting infrastructure with less service interruptions. In this paper we first present a live-migration efficiency testing, captured on a production-level Linux-based virtualization platform, that motivates the need for a better Multi-Virtual machine migration strategy. We then propose a device whose goal is to optimize the bit rate allocation for the live-migration of multiple Virtual Machines and minimize the total migration time, defined as a tradeoff cost function between resource utilization time and user-perceived downtime. By solving our model we gained quantitative and qualitative insights on the design of more efficient solutions for multi-Virtual Machines live migrations. We found that merely few transferring rounds of information memory pages are enough to significantly lower the total migration time. We also signals that under realistic settings, the proposed method converges faster to an optimal bit rate assignment making our approach a viable solution for improving present live-migration implementations.

***Key Words*:**Cloud Computing, Virtualization, Live Migration, Bit rate allocation, Resource utilization.

1 Introduction:

Cloud computing is completely differs the way users approach to information technology (IT) services, both for leisure and business and purposes. The highly distributed features such as utility paradigm allows for cheap, fast, easy, on-demand, present, and transparent access to shared or personal data and computing services. Cloud computing enables users to store information, run suite, develop software tools, and customize entire virtual IT structure without worrying about the cost of the required hardware resources[1]. On the other hand, Cloud providers can take advantage of massive sharing of their data center infrastructure and provide "pay-per-use" IT services at fierce prices. This standard cycle offer and demand has caused the public Cloud computing market to increase in the last few years, with predict of continuing revenue growth. Resource virtualization is one of the key application for an efficient deployment of Cloud services. Decoupling service case from the underlying storage, processing and communication hardware allows to flexibly deploy any application on any server within any data center, independently of the specific platform and operating system being used. In particular, the use of virtual machines (VMs) to implement client service is now a well established practice.

A Virtual Machine can be duplicated, instantiated, migrated, rolled back to a previous state without expensive hardware interventions, enabling rigid workload management operation such as multi-tenant isolation and server consolidation. This is particularly useful in a distributed cloud system, where Virtual Machine can be easily moved to a distant location as long as hypervisor compatibility is guaranteed. Live migration is an important feature that allows to move VMs from one host to another, with minimal disruption to clients availability[2]. Most of the services hosted today within a Virtual Machine are based on multi layer applications. Typical examples include business logic, front-end and back-end tiers of e-commerce services, or MapReduce computing environments. The deployment of multiple correlated Virtual Machines is also envisioned by the emerging Network devices. Function Virtualization models, which is radically changing the way network operators plan to build and develop future network infrastructures, adopting a Cloud-oriented system approach . In general, a Client should be considered as a tenant running multiple VMs, that could be either loosely or strictly correlated, and exchange significant amount of signals among each other. Therefore, moving a given tenants workload often means migrating a group of Virtual Machines, as well as the networks used to interconnect them, to guarantee that the network between the Virtual Machines is best-effort, but has a dedicated channel with reserved bandwidth. Such scenario opens a number of challenges related to multiple VMs live migration, especially when considering transfers across various multiple Clouds, e.g., due to Cloud bursting, where communication resources play a important role. For example, in high-efficiency computing or delay-sensitive applications such as augmented reality, online gaming, video streaming or high-frequency trading, using a sub-optimal bit rate on the management channel needed for VM migrations may lead to Service Level Agreement (SLA) violations, less result in the experience or financial losses. How many Virtual Machines have be migrated? How much bandwidth it is required for each VM transfer and why? What is the amount of efficiency degradation, resource consumption and service interruption that a migration can cause? These are only few of the challenges that may arise in managing a scalable and elastic for the Cloud service. With some existing products do offer a range of solutions for joint management of multiple VMs running multi-tier platforms, they do not allow simultaneous VM live migration and best fit for our knowledge, there are no readily available product solutions that enable the strategy optimized allocation of migration bandwidth.

2 RELATED WORK

Although the idea of virtualized computing dates back to more than 30 years ago the widespread use of Virtual Machines to expense to run multiple commodity operating systems on top of conventional hardware has become a reality with last few decade. The highly flexible workload management enabled by server virtualization allows to duplicate, instantiate and migrate Virtual Machines without expensive hardware interventions[3]. Since the original idea of performing live-migration to transfer Virtual Machines with minimal interruption with several implementations and various research efforts have been carried out on the topic. Only a few solutions, however, deal with the issue of transferring multiple Virtual Machines. To the best of our knowledge first migration service specifically optimized for the cloud transfer of groups of correlated Virtual machine executing many applications. The migration focus on huge VM disk images by taking advantage of the similarities typically shown within Virtual Machines in the same “flock”, which can reach ratios as high as 90%. With the help of a high throughput distributed, de-duplication algorithm, it limits the volume of data migration over the network thus it reducing the migration time. In addition it adopts prioritized data transfers and early Virtual Machine boot techniques to accelerate application start-up at the destination host. However, differently from our work this approach is intended mainly for non-live Virtualization migrations and, in the best case, it requires to pause the Virtual Machine execution for as long as needed to migration the whole memory snap-shot along with the disk image. Furthermore, the best fit- optimal allocation of inter-Cloud link bandwidth will not be considered. Ye et al. performed an extensive experimental evaluation of multiple Virtual Machines live-migration, investigating the role of different resource techniques and migration procedure. In particular, the authors show that identification allocation of CPU and various memory resources on both source as well as destination hosts that improves the migration efficiency and avoiding bugs especially if parallel and workload-aware migration strategies are adopted. Kikuchi et al. investigate the performance of concurrent live-migrations in both dispersion and consolidation experiments and then propose and verify a queuing model. Differently from our request that considers an optimal bit rate allocation the cited solutions do not consider the relevant and do not provide a formal definition of optimal band-width allocation. Other implementation-based studies have been carried out under different pursuing and assumptions different objectives to evaluate simultaneous live-migration of VMs. Some of them apply dispense memory page de-duplication strategies to VMs that are co-located in the same rack or host while others aim at understanding the implications of live-migrating virtual groups or clusters of VMs together as well as the virtual network interconnecting them[4]. Although the cited work offer important insights on implementation aspects of the state-of-the-art techniques for simultaneous VM migrations, they do not provide a bandwidth optimization framework targeted to this purpose. Among the existing approaches to plan multiple Virtual Machine migrations within a data center, the heuristic algorithm proposed by Sarker et al. is aimed at minimizing the total downtime and migration time while taking into account computing, mutual bandwidth and memory requirements of the Virtual Machines.

3 VIRTUAL MACHINE LIVE MIGRATION: BACKGROUND AND CHALLENGES

The migration of VMs from one hosting server to another while they are still running is called live-migration, is a feature of high interest to Cloud managers . In fact, the VM does not need to be shut down at the source or restarted at the destination, allowing the maintenance of both the kernel states, and the states of all processes running within the Virtual Machine. The migration procedure should have less impact on the availability of the services provided by the Virtual Machine given that the whole VM environment is maintained regular across the transfer. In particular, the consistency of the following three important states must be guaranteed:

1. Network: the network states are to be maintained, in order to avoid deliver the ongoing TCP connections and process the current services;
2. Storage: once migrated, the VM has to perform its file system with that used at the source host;
3. Memory: all changes made by the guest operating system to the Virtual Machines system memory during the migration process must be detected and copy at the destination host.

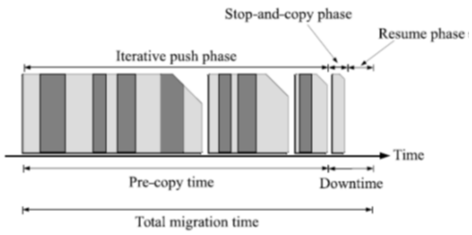

Fig1.Pre-copy live migration timings

Fig1.explain about the pre copy live migration timing where the network state regular methods are easily satisfied in a local Cloud environment, i.e., when the VM is connected to a bridge to the same physical LAN at both source and destination hosts. In this case, the VM can keep the same IP address and, when its execution is interruption at the destination, a ARP packet is sufficient to provide all switches and neighbors aware of the new Virtual Machine location. More complex is the case of a VM migration to a remote data center are connected within a different network/domain: in principle, the same IP address cannot be used again and all ongoing connections are dropped[5]. However, recent IP mobility solutions can be applied to control this issue such as those based on naming the application and not the interact, or those based on the so-called identifier including the Host Identity Protocol (HIP), the Identifier-Locator Network Protocol (ILNP), and the Locator/ID Separation Protocol (LISP). All these solutions can be applied within the VM mobility problem . Another possibility is to extend the Virtual Machines Local Area Network segment over multiple Cloud locations by taking advantage of the Virtual Private LAN Service (VPLS) technology provided by the MPLS protocol. Another prominent approach is to adopt the programmable network reconfiguration ability offered by Software Defined Networking, which allows migrating entire virtual networks from one data center to another, or change rerouting external traffic after a VM has been migrated remaining size of dirty memory is small enough to be transmitted in less than a target interval, or the hypervisor concludes that the memory migration process is not connect, e.g., because the page dirtying rate appears to be more than the transfer rate. At this point, the copy and stop phase starts: the VM is effect at the source host and the remaining dirty pages are copied to the destination. Finally, during the restart phase the VM is brought back on-line at the destination host with consistent storage, memory and network states.

4 CHARACTERIZING LIVE-MIGRATION IN LINUX QEMU-KVM ENVIRONMENTS

In order to define expected live migration model, we analyze in details how such task is evaluated in the Linux QEMU-KVM hypervisor. We also carry out an experimental characterization of the live-migration plan, to assess the role of the main system parameters, i.e., available memory dirtying rate, bit rate and maximum allowed down-time[6]. The VM live-migration procedure in a libvirt-managed QEMU-KVM environment can be brief in the following steps:

1. The maximum transfer rate are allowed downtime parameters, if specified are move to the QEMU-KVM hypervisor, and the migration timeout counter is started by libvirt.

2. A secure channel has been established between source and destination hosts.

3. If needed, the image disk replication begins, while dirty blocks are being tracked.

4. A paused case of the VM to be migrated is created at the destination.

5. Memory pages and/or disk blocks are continuously sent over the fixed channel until the maximum. transfer rate is reached the repeatedly memory push phase begins, while dirty pages are being monitored.

6. The size of the remaining blocks is used to estimate their total transfer time at the maximum possible channel bit rate, i.e., not considering the arrange rate limitation.

7. The estimated remaining transfer time will be larger than the maximum allowed downtime and the migration timeout has not cease.

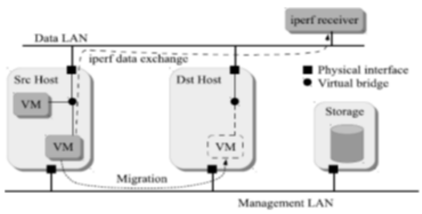

Fig.2 Experimental setup used for single Virtual machine live migration in libvirt environment

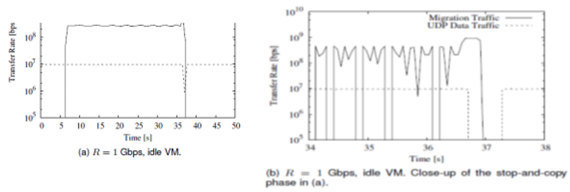

Fig3.Measured transfer rate of migration traffic and UDP data flow generated by the virtual machine

Fig3.explain about measured transfer rate of migration traffic and UDP data flow generated by the virtual machine for the two value of the bit rate available for migration and different virtual machine execution states. The full bandwidth is used in the stop and copy phase only with corresponding UDP flow interruption. Maximum transfer rate violations due to increased dirtying rate bandwidth available on the migration channel forced by the hypervisor to use stop and copy phase prolonging the total migration time about 30s.

5 PERFORMANCE EVALUATION:

In this section we have the various results obtained from our simulation confined. By solving our geometric program across a wide range of parameter field, we are able to answer measuring design questions such as live-migration should always be used when moving multiple Virtual Machines and quantitative questions such as few dirty page transferring rounds are often enough to lesser the total migration time[7]. Summarizing the details in three main tasks, we found that, under the simulation settings to minimize the total migration time: (1) the downtime should be as less as possible, (2) we should always have at least one moving round, but (3) prolonging the pre-copy time too much does not help upgrade the total migration time.

5.1 Simulation Environment:

To validate our system model to gain insights on the performance of multi-Virtual Machine in live-migration to conduct a simulation campaign using the geometric program solver. It uses the dual interior point method to solve the various version of the original geometric program. All our code is available in various parameter space but we present only a significant representative subset that brief our take-home messages. Even though it support optimization problems with larger inputs when properly configured our performance evaluation results are limited to a small set of Virtual Machine migrations[8]. Scalability is not a better concern in our settings since the majority of Cloud applications for enterprises need to performs migrate only small sets of Virtual Machines. With the intent of keeping a general approach, we evaluate different distributions on the Virtual Machine memory size, which is a critical parameter in live migrations: (i) a uniform distribution to be considered for complete randomness; (ii) a gaussian distribution to take into account unique from a large number of samples around a mean value (iii) a bimodal distribution to be consider for two kinds of Virtual Machines, small vs. large, a common case in Cloud computing In some cases, we assumed a constant Virtual Migration memory size.

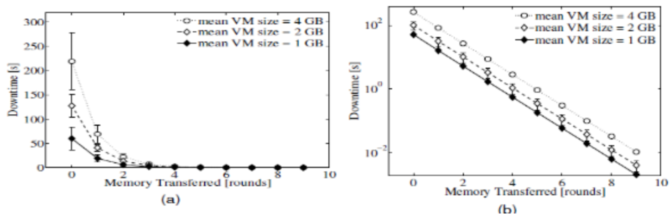

Fig.4 the downtime significantly decreases when increasing the number of rounds for impact of downtime only for multiple

5.2 Simulation Results

(1) The downtime can be smaller up to two order of magnitudes while maximum the number of transferring rounds. The impact of the downtime, when migrating repeatedly 3 VMs. The maximum rate was set to 1 Gbps, and the values of minimum live migration time are obtained by solving geometric Problem when Cpre = 0 and Cdown = 1. The size of the Virtual Machines is chosen from a distribution with mean and standard deviation value we plot the same data with logarithmic scale, to verify that the downtime may diminish of two orders of magnitude, as we increase the number of rounds[9]. We have generated the VM sizes sampling from a bimodal distribution where 20% of the time the VM has size 10 times bigger than what expressed in the legend, and same data is expressed with semi-logarithmic scale[10]. In Figure 4, we have instead generated the VM sizes sampling from a uniform distribution with average reported in the legend, and standard deviation of 300 MB.

(2) The total migration time improvement decline as we improve the number of transferring rounds. This diminishing effect has a outcome of the diminishing duration of each subsequent transferring round[11]. This result gives insights on the features for allocating enough bandwidth, to guarantee a given quality of service to Virtual Machines running memory-rigious applications or with high dirtying rate. The values of minimum migration time were obtained solving geometric problem when Cpre = 1 and Cdown = 0. In the impact of the pre-copy time only is computed for Virtual Machines at a maximum available rate of 0:5 Gbps[12]. The size

of the VMs is chosen from a gaussian distribution with mean and standard deviation 300 MB. We show that another evaluation with identical parameter set except with that we double the maximum available rate R the available capacity grows, the gain threshold point is reached maximum[13]. This result are await and it is a good sanity check of our simulator show a similar experiment of Virtual Machines of constant memory size as expressed in the legend and it introduce randomness by changing the number of dirty pages per second[14]. The various sample of the dirty rate D from a gaussian distribution with average and a standard deviation of 300 pps[15]. Each page is assumed to have various Linux standard size of 4096 Bytes. It has show the effect of doubling the number of VMs to be migrated[16].

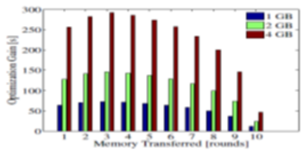

Fig 5. Optimization gain for geometric program for Non-Optimal, Fixed Migration Allocation

Fig5 explain about the geometric program for non-optimal, fixed migration allocation the optimization gain as we increase the number of transferring rounds to migrate three Virtual Machines with various different size[17]. In this situation it compute the total migration time difference with a fixed bit rate and non-optimal allocation equal to R=3, where R = 1 Gbps[18]. The optimization gain is increased for a few number of rounds, confirming for various the existing results[19]. The main reason is that, when the number of transfer becomes too large both the fixed and optimal bandwidth allocation reach a saturation point[20[21].

6 CONCLUSION

In this paper it mainly deals with the live-migration of multiple virtual machines problem that are networked in Cloud service management. Firstly it described the live-migration process and its features with some experimental results that help identifying the key parameters of the live-migration evaluation. We have used various insights to build a geometric programming and live migration model formulation that when solved the returns the minimum total migration time by favorable allocating the bit rates across the multiple Virtual Machines to be migrated. Our optimization problem aims at simultaneously limiting both the service interruption and the time of Virtual Machine pre-copy along with a proper control of the repetitive memory copying algorithm used in live migration. By solving the geometric program across a wide range of parameter settings, we are able to answer qualitative design questions such as live-migration should always be used while transferring various multiple Virtual Machines to quantitative questions such as few dirty page transferring rounds are often enough to minimize the total migration time. In realistic settings, the proposed geometric program converges sharply to an optimal bit rate assignment making it a usable and viable solution for improving the current live-migration implementations. As an interesting question it leave the various design and implementation based on widely used open-source virtualization environments such as an experimental testbed supporting concurrent live-migration with optimized bit rate allocation in the various phases. Another interesting question is the investigation of the migration of larger pools of Virtual Machines or even entire networks for various different migration request and time-scales.

References

[1] R. Buyya, J. Broberg, and A. M. Goscinski, Eds., Cloud Computing: Principles and Paradigms. New York: Wiley, 2011.

[2] A. Bartels, J. R. Rymer, and J. Staten, The Public Cloud Market is Now in Hypergrowth. Sizing the Public Cloud Market, 2014 to 2020. Forrester Research, 2014.

[3] R. P. Goldberg, Survey of virtual machine research, IEEE Computer, vol. 7, no. 6, pp. 3445, June 1974.

[4] V. Medina and J. M. Garca, A survey of migration mechanisms of virtual machines, ACM Computing Surveys, vol. 46, no. 3, pp. 30:130:33, January 2014.

[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, Live migration of virtual machines, in Proceedings of the 2nd USENIX Symposium on Networked Systems Design & Implementation (NSDI), Boston, MA, May 2005.

[6] X. Wang, Z. Du, Y. Chen, and S. Li, Virtualization-based autonomic resource management for multi-tier web applications in shared data center, Journal of Systems and Software, vol. 81, no. 9, pp. 15911608, September 2008.

[7] J. Dean and S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM, vol. 51, no. 1, pp. 107113, January 2008.

[8] Network Functions Virtualisation (NFV): Network operator perspectives on industry progress, ETSI White Paper, The European Telecommunications Standards Institute, 2013.

[9] F. Cuadrado, A. Navas, J. C. Duenas, and L. M. Vaquero, Research challenges for cross-cloud applications, in Proceedings of the IEEE INFOCOM 2014 Workshop on Cross-cloud Systems (CrossCloud 14), Toronto, Canada, April 2014.

[10] S. J. Bigelow. (2014, November) How can an enterprise benefit from using a VMware vApp? [Online]. Available: http://searchvmware.techtarget.com/answer/ How-can-an-enterprise-benefit-from-using-a-VMware-vApp

[11] [Online]. Available: http://www.linux-kvm.org

[12] [Online]. Available: http://libvirt.org

[13] R. J. Duffin, E. L. Peterson, and C. M. Zener, Geometric Programming: Theory and Application. New York: Wiley, 1967.

[14] S. Devine, E. Bugnion, and M. Rosenblum, Virtualization system including a virtual machine monitor for a computer with a segmented architecture, US Patent 6 397 242, 2002.

[15] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, Xen and the art

of virtualization, in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 03), Bolton Landing, NY, October 2003.

[16] M. Nelson, B.-H. Lim, and G. Hutchins, Fast transparent migration for virtual machines, in Proceedings of the 2005 USENIX Annual Technical Conference (ATEC05), Anaheim, CA, April 2005.

[17] D. Kapil, E. S. Pilli, and R. C. Joshi, Live virtual machine migration techniques: Survey and research challenges, in Proceedings of the 3rd IEEE International Advance Computing Conference (IACC 2013), Ghaziabad, India, February 2013.

[18] S. Acharya and D. A. DMello, A taxonomy of live virtual machine (VM) migration mechanisms in cloud computing environment, in Proceedings of the International Conference on Green Computing, Communication and Conservation of Energy (ICGCE 2013), Chennai, India, December 2013.

[19] R. Boutaba, Q. Zhang, and M. F. Zhani, Virtual machine migration: Benefits, challenges and approaches, in Communication Infrastructures for Cloud Computing: Design and Applications, H. Mouftah and B. Kantarci, Eds. IGI-Global, 2013.

[20] S. Al-Kiswany, D. Subhraveti, P. Sarkar, and M. Ripeanu, VMFlock: Virtual machine co-migration for the cloud, in Proceedings of the 20th International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC 11), San Jose, CA, June 2011.

[21] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, Live migration of multiple virtual machines with resource reservation in cloud computing environments, in Proceedings of the 4th IEEE International Conference on Cloud Computing (CLOUD 2011), Washington, DC, July 2011.