

Live Migration of Virtual Machines in Multiple Datacenters

Luyang Tang^{1,a}, Dongcheng Zhao^{1,b}, Zhi Tan^{3,c}, Gang Sun^{1,2,d} and Dan Liao^{1,2,e}

¹Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China

²Institute of Electronic and Information Engineering in Dongguan, UESTC, China

³GuangDong Poya Information & Technology Co.,Ltd , China

^a1403812008@qq.com, ^b1061097257@qq.com, ^c380003093@qq.com, ^dgangsun@uestc.edu.cn, ^e349440173@qq.com

Keywords: Virtual Machine, Migration, Virtualization, Cloud computing.

Abstract. Most research on virtual machine (VM) migration focus on the issue of single VM migration by using the post-copy or pre-copy migration strategy. In this paper we focus on multiple VM migration, which becomes important when we have several cooperating VMs providing services, and propose an improved serial migration strategy that is based on the serial migration strategy. We then present the m mixed migration strategy that is based on the improved serial migration strategy and the parallel migration strategy. We conduct detailed simulation experiments to evaluate the performance of the m mixed migration strategy in terms of the migration time and the downtime.

1. Introduction

Cloud computing uses virtualization technologies to enable cloud service providers to offer a variety of services to users. The increasing popularity of cloud computing has led to the deployment of multiple datacenters to satisfy service requirements and fault tolerance. Therefore, recent studies on cloud computing focus more on multiple datacenters [1, 2]. To improve and/or maintain the service level agreements (SLAs), researchers proposed virtual machine (VM) migration technologies [3]. The migration of VMs has been typically done in an offline manner. More recently, online migration has become available in [4], which enables the live migration of VMs.

Based on the system demands various performance indicators are optimized. The work in [5] proposed an analytical performance model for the live migration of a single VM. In [6], the researchers theoretically analyzed the necessary bandwidth resources to satisfy the total migration time and the downtime constraints in the single VM live migration process.

Most research considers the live migration strategy for migrating single VMs. However, when multiple cooperating VMs are necessary to provide certain services, we now also need to consider migration of groups of cooperating VMs [7]. In [8], the authors studied a strategy to migrate virtual clusters from a physical machine to another physical machine. In [9], the researchers utilized the live migration feature of virtual machine monitors to migrate a job from one sub-cluster to another. In [10] the researchers proposed a method for managing migration of a virtual machine cluster. The technique migrates the VMs to a new physical destination machine when the resources of the source physical machine hosting the VM are close to a minimum critical occupied resource weight ratio. The work in [7, 11] proposed the serial and parallel migration strategies for multiple VMs. However, there is room for improvements of the serial and the parallel migration strategies for multiple VMs, when the maximum tolerable downtime is given in the SLAs.

In this paper, we study the improved serial migration strategy for multiple VM migration and simulate and analyze its performance. Then we present the m mixed migration strategy. This m mixed migration strategy aims to minimize the total migration time while satisfying the maximum downtime constraint that users agreed with the service providers in the SLAs. In order to evaluate the blocking rate, we model the problem as an M/M/C/C queuing model. We use the M/M/C multi-server queuing model to evaluate the average waiting queue length and the average waiting time.

2. Existing migration strategies

In this section, we discuss the migration of multiple VMs. In the pre-copy migration mechanism [11], the VM memory is migrated iteratively. The VM memory contains the original memory and the dirty (i.e., generated or modified) memory. The original memory is the memory when starting the VM migration, whereas the dirty memory is generated or modified during the iterative transmission process. The iterative migration process of VM_{*i*} (*i*=1, 2, ..., *M*) is described in [11]. We can calculate the total migration time of the *i*-th VM (denoted as $T_{i,mig}$) as follows.

$$T_{i,mig} = \sum_{j=1}^{n_i+1} T_{i,j} = \frac{V_m}{R} \frac{1-r^{n_i+1}}{1-r}, \quad (1)$$

$$n_i = \min\{\lceil \log_r(V_{th}/V_m) \rceil, n_{max}\}. \quad (2)$$

Eq. 1 shows that the migration time is the sum of the time of each iteration. Eq. 2 gives the actual number of iterations which denoted as n_i . Where $T_{i,j}$ is the total time consumed in the *j*-th iteration for transmitting the *i*-th virtual machine's dirty memory. We assume that the amount of original memory of each VM is V_m . We use V_{th} to denote the iteration threshold for stopping the iteration and n_{max} to present the maximum number of iterations. R is the transmission rate and $r = PD/R$ is the ratio of the dirtying rate to the transmission rate, where D and P are memory page dirtying rate and memory page size, respectively. Here the memory page dirtying rate is the rate at which the dirty (or modified) memory page is generated. The number of VMs in the VDC request (i.e., multiple VMs) is M .

In this paper, we focus on the migration strategy for multiple VMs. In [7, 11], the authors proposed a serial migration strategy for migrating multiple VMs. In the strategy, the M VMs are migrated one by one, and the actual number of each VM's iterations is $n(s)$. Therefore, the serial migration time (denoted as T_{mig}^s) and the downtime (denoted as T_{down}^s) can be calculated as in Eq. 3 and 4, respectively.

$$T_{mig}^s = \sum_{i=1}^M T_{i,mig} = \frac{MV_m}{R} \frac{1-r^{n(s)+1}}{1-r}, \quad (3)$$

$$T_{down}^s = \frac{V_m}{R} r^{n(s)} + (M-1) \frac{V_m}{R} \frac{1-r^{n(s)+1}}{1-r} + T_{res}, \quad (4)$$

where $n(s) = \min\{\lceil \log_r(V_{th}/V_m) \rceil, n_{max}\}$ and $0 < r < 1$.

The authors in [7, 11] also proposed a strategy for the parallel migration of multiple VMs. In the parallel migration strategy, the M VMs are simultaneously migrated, and the actual number of each VM's iterations is $n(p)$. The migration time (denoted as T_{mig}^p) and the downtime (denoted as T_{down}^p) are formulated as follows.

$$T_{mig}^p = \sum_{i=1}^M T_{i,mig} = \frac{MV_m}{R} \frac{1-(Mr)^{n(p)+1}}{1-Mr}, \quad (5)$$

$$T_{down}^p = \frac{MV_m}{R} (Mr)^{n(p)} + T_{res}, \quad (6)$$

where $n(p) = \min\{\lceil \log_{Mr}(V_{th}/V_m) \rceil, n_{max}\}$ denotes the actual number of iterations of each VM, $0 < Mr < 1$ is the ratio of the dirtying rate of memory page to the transmission rate.

3. Strategies for multiple VMs migration

3.1 Improved serial migration of multiple VMs

Since, in serial migration strategy of multiple VMs, there is room for shortening the migration time and the downtime. Accordingly, we propose an improved serial migration strategy of multiple VMs.

In the improved serial migration strategy, we use the pre-copy scheme to migrate the first VM and use the post-copy scheme[12] to migrate the remaining VMs one by one. In the post-copy strategy, when the VM is restarted, the VM memory is empty. If the VM tries to access a memory page that has not yet been copied, this memory page needs to be brought from the source VM. Accordingly, we get the total migration time of the i -th VM in the worst case as follows.

$$T_{i,mig} = \frac{V_m}{R}, i = 2, \dots, M \quad (7)$$

Eq.7 gives the time for migrating all the memory of a VM. However, most of the time, many of the memory pages will not be used, hence we only need to copy the necessary VM memory. Therefore we introduce a factor α into Eq. 7, where α is the ratio of the actual migrated memory to all memory. So αV_m denotes the actual amount of memory that needs to be migrated. Since V_{th} is the threshold for stopping the iterations, so αV_m is larger than V_{th} , namely, $\alpha V_m > V_{th}$, thus $\alpha > V_{th} / V_m$. Therefore, we can get $V_{th} / V_m < \alpha \leq 1$.

$$T_{i,mig} = \frac{\alpha V_m}{R}, \frac{V_{th}}{V_m} < \alpha \leq 1, i = 2, \dots, M \quad (8)$$

In the improved serial migration strategy, the actual number of iterations of first VM is $n(s_1)$ and $n(s_1) = n(s)$, and the remaining $M-1$ VMs are migrated by using the post-copy migration scheme, so they do not need iterative migration. According to the improved serial migration strategy for multiple VMs and Eq. 1 and 8, we can get the migration time of the improved serial migration strategy (denoted as T_{mig}^{s1}) as follows.

$$T_{mig}^{s1} = \frac{V_m}{R} \frac{1-r^{n(s)+1}}{1-r} + (M-1) \frac{\alpha V_m}{R}, \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (9)$$

When the first VM stopped, the other VMs will also be stopped and the service will be unavailable. Thus, we can get the downtime of the improved serial migration strategy (denoted as T_{down}^{s1}) as follows.

$$T_{down}^{s1} = \frac{V_m}{R} r^{n(s)} + (M-1) \frac{\alpha V_m}{R} + T_{res}, \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (10)$$

3.2 Comparisons on time consumption

Similar to [7, 11], we assume that $n(s) = \log_r(V_{th} / V_m)$ and $n(p) = \log_{Mr}(V_{th} / V_m)$. We can compare our improved serial migration strategy with the serial migration strategy as follows.

$$T_{mig}^s - T_{mig}^{s1} = \frac{(M-1)V_m}{R} \frac{(1-\alpha) + (r - r^{n(s)+1})}{1-r} > 0 \quad (11)$$

where $V_{th} / V_m < \alpha \leq 1, 0 < r < 1$. We can also get:

$$T_{down}^s - T_{down}^{s1} = \frac{(M-1)V_m}{R} \frac{(1-\alpha) + (r - r^{n(s)+1})}{1-r} > 0 \quad (12)$$

where $V_{th} / V_m < \alpha \leq 1, 0 < r < 1$.

From Eq. 11 and 12, we can see that the downtime and the migration time of our improved serial migration strategy are shorter than that of the serial migration strategy.

In the following, we compare the migration time and downtime of improved serial migration strategy with that of the parallel migration strategy.

$$T_{mig}^p - T_{mig}^{s1} = \frac{1}{R} \frac{(Mr-r)(V_m - V_{th})}{(1-Mr)(1-r)} + \frac{(M-1)}{R} \left(\frac{V_m - MrV_{th}}{1-Mr} - \alpha V_m \right) > 0 \quad (13)$$

where, $\frac{1}{R} \frac{(Mr-r)(V_m - V_{th})}{(1-Mr)(1-r)} > 0$, $\frac{V_{th}}{V_m} < \alpha \leq 1$ and $0 < Mr < 1$ when $\alpha = 1$, $\frac{(M-1)}{R} \left(\frac{V_m - MrV_{th}}{1-Mr} - \alpha V_m \right)$ is minimum value.

Since $\frac{(M-1)}{R}(\frac{V_m - MrV_{th}}{1-Mr} - \alpha V_m) > \frac{(M-1)}{R} \frac{Mr(V_m - V_{th})}{1-Mr} > 0$, we have:

$$T_{down}^p - T_{down}^{s1} = \frac{(M-1)(V_{th} - \alpha V_m)}{R} < 0, \text{ when } \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (14)$$

Eq. 13 and 14 show that the migration time of our improved serial migration strategy is shorter than that of the parallel migration strategy; and when $V_{th}/V_m < \alpha \leq 1$, the downtime of our improved serial migration strategy is longer than that of the parallel migration strategy.

3.3 m mixed migration of multiple VMs

The migration time and downtime will affects the quality of experience (QoE) of users. In order to guarantee the QoE, the service providers have to negotiate the maximum tolerable downtime of each migration request with the users. While satisfying the constraint on maximum tolerable downtime, we should reduce the migration time as much as possible, since it has a substantial positive impact on the performance of the whole IT infrastructure.

For this purpose, we propose a mixed migration strategy for multiple VMs that is based on the improved serial migration and parallel migration strategies. We first use the pre-copy migration scheme to migrate m VMs in parallel. When the m VMs have been stopped running, we stop the remaining VMs, and then use the post-copy migration strategy to serially migrate the remaining VMs. When all VMs are migrated completely, we restart all VMs simultaneously. We call such mixed migration strategy as the m mixed migration strategy in this work.

Therefore, we can get the migration time T_{mig}^m and the downtime T_{down}^m of the m mixed migration strategy as follows, where the actual number of iterations of each VM is denoted as $n(m) = \log_{mr}(V_{th}/V_m)$, $1 \leq m \leq M$ and $0 < Mr < 1$. Note that, when $m=1$, the m mixed migration strategy turns into the improved serial migration strategy. Similarly when $m=M$, the m mixed migration strategy turns into the parallel migration strategy.

$$T_{mig}^m = \frac{mV_m}{R} \frac{1-(mr)^{n(m)+1}}{1-mr} + (M-m) \frac{\alpha V_m}{R}, \quad \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (15)$$

$$T_{down}^m = \frac{mV_m}{R} (mr)^{n(m)} + (M-m) \frac{\alpha V_m}{R} + T_{res}, \quad \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (16)$$

From Eq. 15 and 16, we can see that the migration time increases with the growth of m , and the downtime decreases with the growth of m . Assuming $m_1 < m_2$, we have:

$$T_{mig}^{m_1} - T_{mig}^{m_2} = \frac{(m_2 - m_1)((m_1 + m_2 - m_1 m_2 r)V_{th} - V_m)}{R(1 - m_1 r)(1 - m_2 r)} + \frac{\alpha(m_2 - m_1)(1 - m_1 r)(1 - m_2 r)V_m}{R(1 - m_1 r)(1 - m_2 r)},$$

Since $\frac{\alpha(m_2 - m_1)(1 - m_1 r)(1 - m_2 r)V_m}{R(1 - m_1 r)(1 - m_2 r)} > 0$, when $\alpha=1$, $T_{mig}^{m_1} - T_{mig}^{m_2}$ achieves the maximum value.

So, $T_{mig}^{m_1} - T_{mig}^{m_2} \leq \frac{(m_2 - m_1)((m_1 + m_2 - m_1 m_2 r)(rV_{th} - rV_m))}{R(1 - m_1 r)(1 - m_2 r)} < 0$.

Since $0 < Mr < 1$, $0 < m_1 r < 1$ and $0 < m_1 m_2 r < m_1$, hence: $m_1 + m_2 - m_1 m_2 r > m_1 + m_2 - m_1 > m_2 > 0$.

When $m=1$, the m mixed migration strategy turns into the improved serial migration strategy; whereas when $m=M$, the m mixed migration strategy turns into the parallel migration strategy. Therefore, when $1 \leq m_1 < m_2 \leq M$, we have:

$$T_{mig}^{s1} = T_{mig}^1 \leq T_{mig}^{m_1} < T_{mig}^{m_2} \leq T_{mig}^M = T_{mig}^p, \quad 1 \leq m_1 < m_2 \leq M. \quad (17)$$

Similarly, $T_{down}^{m_1} - T_{down}^{m_2} = \frac{(m_2 - m_1)(\alpha V_m - V_{th})}{R} > 0$.

Where αV_m is the VM memory requiring transmission in the post-copy migration process. So when $\alpha V_m > V_{th}$ and $1 \leq m_1 < m_2 \leq M$, we have:

$$T_{down}^p = T_{down}^M \leq T_{down}^{m_2} < T_{down}^{m_1} \leq T_{down}^1 = T_{down}^{s1}, 1 \leq m_1 < m_2 \leq M. \quad (18)$$

From Eq. 17 and 18 we can see that the migration time increases with the growth of m , and the downtime decreases with the growth of m .

For a given maximum downtime T_{down}^{\max} and $n(m) = \log_{mr}(V_{th} / V_m)$, we can obtain the inverse solution of m through Eq. 16, and m can be computed as in Eq. 19. Note that, because the downtime of the m mixed migration strategy decreases with the growth of m , so T_{down}^{\max} must fall into $[T_{down}^M, T_{down}^1]$.

$$m = \left\lceil \frac{RT_{down}^{\max} - RT_{res} - \alpha MV_m}{V_{th} - \alpha V_m} \right\rceil, \quad \frac{V_{th}}{V_m} < \alpha \leq 1 \quad (19)$$

4. Numerical results

In this section, we numerically compare and analyze the performance of the serial migration strategy, the improved serial migration strategy, the parallel migration strategy and the m mixed migration strategy for multiple VMs. Similar to [11], we set the values of parameters as follows: $\alpha=1$; $M=8$; $R=1\text{Gbps}$; $V_m=1\text{GB}$; $P=4\text{KB}$; $D=2500\text{pps}$; $r=0.08$; $V_{th}=0.1\text{GB}$; $n_{max}=8$; $T_{res}=0.1\text{s}$.

Fig.1 (a) shows the migration time and downtime of the serial migration, the improved serial migration and the parallel migration strategies as a function of the ratio of the dirtying rate to the transmission rate. From Fig.1, we can see that the numerical results and our theoretical analysis are consistent. The migration time of the serial migration strategy is shorter than the parallel migration strategy; and the downtime of the serial migration strategy is longer than the parallel migration strategy. The migration time and the downtime of the improved serial migration strategy are also shorter than the serial migration strategy. The migration time of the improved serial migration strategy is shorter than that of the parallel migration strategy, and the downtime of the improved serial migration is longer than that of the parallel migration. Therefore, different migration strategy may be applicable for different performance requirements. Based on the improved serial migration strategy and the parallel migration strategy, we propose the m mixed migration strategy, for further improving the performance.

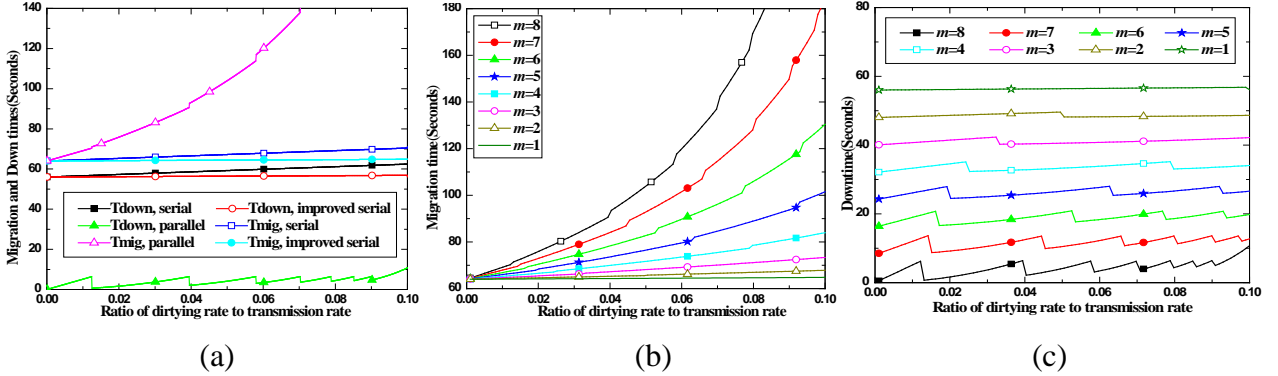


Fig.1. Comparisons on migration time and downtime.

Fig.1 (b) shows the migration time of the m mixed migration strategies. We can see that a larger m leads to a longer migration time. This is because that larger m means more VMs need to be parallel migrated and the migration time of the parallel migration strategies is longer than that of the improved serial migration strategy. Thus, the migration time of the m mixed migration strategy increases with the growth of the value of m . Thus, service providers who want to reduce the migration time, should try to lower the ratio of dirtying rate to transmission rate or choose the m mixed migration strategy with smaller m .

Fig.1 (c) demonstrates the downtime of the m mixed migration strategies. We can see that the downtime of m mixed migration strategy decreases with the growth of m . This is because larger m means more VMs need to be parallel migrated and the downtime of the parallel migration strategy is shorter than that of the improved serial migration strategy. Therefore, if service providers intend to reduce downtime, should choose the m mixed migration strategy with larger m .

5. Conclusions

In this paper, we study the problem of live migration for multiple VMs within multiple datacenters. We present the improved serial migration strategy and the m mixed migration strategy for efficient VM migration. In the m mixed migration strategy, when the number of parallel migrated VMs (i.e., m) varies from 1 to M (i.e., the total number of VMs), the migration time and the downtime of the m mixed migration strategy vary with the change of m . Therefore, the m mixed migration strategy can provide multiple “candidate” migration solutions, and for a given maximum tolerable downtime, the service providers can find an “optimal” migration solution with the shortest migration time.

Acknowledgements

This research was partially supported by the National Grand Fundamental Research 973 Program of China under Grant (2013CB329103), Natural Science Foundation of China grant (61271171, 61571098), China Postdoctoral Science Foundation (2015M570778), Guangdong Science and Technology Project (2012B090400031, 2012B090500003, 2012B091000163), and National Development and Reform Commission Project.

References

- [1] Hao Jin, Deng Pan, Jing Xu, Niki Pissinou. Efficient VM Placement with Multiple Deterministic and Stochastic Resources in Datacenters. IEEE GLOBECOM, 2505-2510, 2012.
- [2] Ahmed Amokrane, Mohamed Faten Zhani, Rami Langar, Raouf Boutaba, Guy Pujolle. Greenhead: Virtual Datacenter Embedding across Distributed Infrastructures. IEEE Transactions on Cloud Computing, 1(1), 2013.
- [3] Vivek Shrivastava, Petros Zerfos, Kang-Won Lee, Hani Jamjoom, Yew-Huey Liu, Suman Banerjee. Application-aware virtual machine migration in datacenters. IEEE INFOCOM, 66-70, 2011.
- [4] Microsoft, Inc. Windows Server 2012 R2. <http://www.microsoft.com/en-us/server-cloud/hyper-v-server>.
- [5] Hamzeh Khazaei, Jelena Masic, Vojislav B. Masic. Performance of an IaaS Cloud with Live Migration of Virtual Machines. IEEE GLOBECOM, 2289-2293, 2013.
- [6] Jiao Zhang, Fengyuan Ren, Chuang Lin. Delay Guaranteed Live Migration of Virtual Machines. IEEE INFOCOM, 574-582, 2014.
- [7] Walter Cerroni. Multiple Virtual Machine Live Migration in Federated Cloud Systems. IEEE INFOCOM Workshop on Cross-Cloud Systems, 25-30, 2014.
- [8] Kejiang Ye, Xiaohong Jiang, Ran Ma, Fengxi Yan. VC-Migration: Live Migration of Virtual Clusters in the Cloud. The 13th ACM/IEEE International Conference on Grid Computing (GRID 2012), 209-218, 2012.
- [9] Muhammad Atif, Peter Strazdins. Adaptive parallel application resource remapping through the live migration of virtual machines. Future Generation Computer Systems, 37, 148-161, 2014.
- [10] Chao-Tung Yang, Jung-Chun Liu, Kuan-Lung Huang. A method for managing green power of a virtual machine cluster in cloud. Future Generation Computer Systems, 37, 26-36, 2014.
- [11] Franco Callegati, Walter Cerroni. Live Migration of Virtualized Edge Networks: Analytical Modeling and Performance Evaluation. IEEE Workshop on Software Defined Networks for Future Networks and Services (SDN4FNS 2013), 1-6, 2013.
- [12] Michael R. Hines, Umesh Deshpande, Kartik Gopalan. Post-copy live migration of virtual machines. ACM Operating Systems Review, 43(3), 14-26, 2009.