Scheduling Cloud Platform Managed Live-Migration Operations to Minimize the Makespan

Xiaoyong Yuan, Ying Li, Yanqi Wang, Kewei Sun

Poster: Scheduling Cloud Platform Managed Live-Migration Operations to Minimize the Makespan

Yuan Xiaoyong[1], Li Ying[1 2], Wang Yanqi[3], and Sun Kewei[3]

[1] School of Software and Microelectronics, Peking University, Beijing, CHINA,
[2] National Engineering Center of Software Engineering, Peking University, Beijing, CHINA,
[3] IBM Research - China, Beijing, CHINA

Abstract. Live-migration of virtual machines (VMs) has become an indispensable management operation of cloud platforms. The cloud platforms need to migrate multiple co-located and live VMs from one physical node to another for power saving, load balancing and maintenance. Such live-migration operations are critical to the running services, and thus should be completed as fast as possible. State-of-the-art live-migration techniques optimize the migration performance of single or multiple VMs by concentrating on Virtual Machine Monitor (VMM), little attention has been given to the cloud platforms which control and schedule the multiple migration operations. In this paper, we consider the problem of scheduling migration operations to minimize the makespan.

1 Cloud platform managed migration operations

Live-migration of VMs has become an indispensable management operation of cloud platforms. Cloud platforms present users the ability to deploy VMs over a cluster of physical machines on demand from a centralized management node, thus building what is usually referred to as a VM-based cloud, which can then be used to provide IaaS. At the current stage, however, several management issues still deserve additional investigation, such as performance of management operations. The research[1] reveals that the burst of management operations such as VM live-migration is the rule rather than the exception, in the VM-based cloud, and planning and orchestrating management operations is essential for efficient cloud operations. State-of-the-art live VM migration techniques optimize migration performance of single or multiple VMs performed by VMM, whereas the optimization of scheduling live-migrations centralized managed by cloud platform is still missing, especially for new platform(eg. OpenStack). For example, it will take more than 2 minutes for OpenStack to migrate 30 idle VMs (KVM driver), for that the applications within VMs suffer from degraded performance.

In cloud platform like OpenStack and CloudStack, live-migration as a management operation, is viewed as a transactional interaction between a controller node and two compute nodes which provide computation capability. The migration operation has 4 phases: 1) *checking* that the scheduler on controller node finds a proper destination node; 2) *pre-migration* that the destination node builds a new idle instance for receiving contents; 3) *live-migration* that VMM (eg. Libvirt in OpenStack) is invoked to perform live-migration of VM from source to destination using pre-copy method[2]; 4) *post-migration* is a phase that the instance tears down network and updates its status on source node. In this paper, we consider optimizing migration operations of multiple co-located VMs from one physical node to another. Supposing we are given n VMs on the source node to be migrated, cloud platform will schedule and perform a set of n migration operations $\mathbf{M} = \{MO_1, MO_2, \ldots, MO_n\}$,

as depicted in Figure 1. For each migration operation MO_i, T_{ij} is its processing time in phase j. Besides phase $1-4$, there is a phase 0 to indicate its waiting time for performing. The objective is to find a feasible schedule of minimum completion time of n migration operations; that is, to minimize the makespan $\mathbf{C}$:

$$\min_{\mathbf{M}} \mathbf{C}(\mathbf{M}), \tag{1}$$

where makespan $\mathbf{C}$ is the maximum completion time of n migration operations:

$$\mathbf{C}(\mathbf{M}) = \max_{i} \Sigma_{j=0}^{4} T_{ij}, \quad i = 1, 2, \ldots, n. \tag{2}$$

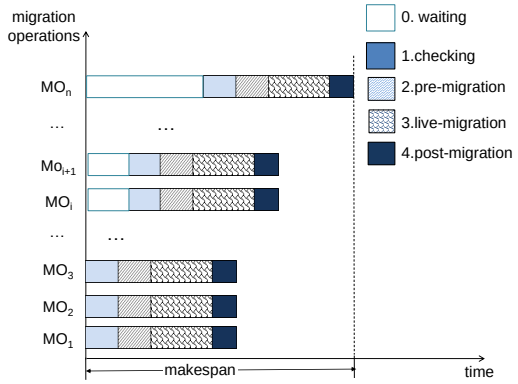

Fig. 1. cloud platform managed VM migration operations

2 Scheduling

2.1 Multiple Migration Operations of idle VM

In order to save physical resource and reduce operational cost, the cloud platform need to migrate idle or light workload VMs among servers for consolidation. When the cloud platform performs n migration operations of idle VMs between two nodes, the makespan varies with the number of migration operations m performed concurrently. If $m = 1$, n migration operations are performed sequentially, and with m increasing, more migration operations are executed simultaneously in one group. The migration operation in different phases is mainly performed on different node (i.e., phase 1 on controller node, phase 2 on destination node, phase 3 and 4 on source node). When $m = N_i$, the processing time of m migration operations in phase i will greatly exceed that of $m = N_i - 1$, because the nodes capacity can't afford that concurrency level. For example, in phase 1, controller node can't afford N_1 concurrent migration operations and the processing time would be extremely large compared with that of $N_1 - 1$ operations. The processing time of m concurrent migration operations in phase i is defined by

$$T_{ij}(m) = \begin{cases} (n - N_j)T_j^*, & n > N_j \\ T_j^*, & n \leq N_j \end{cases} \quad (j = 1, 2, 4)$$

T_i^* is a constant coefficient of each equations. Here we perform m migration operations simultaneously by group. To complete n migration operations, $\frac{n}{m}$ groups are going to be migrated in all. Because the duration of live-migration phase of migration operations of idle VM is short compared with other phases, T_{3i} can be assumed as constant, and let $T_{3i} = T_3^*$. For there's no different in T_{i1}, T_{i2}, T_{i4} among VMs, for these phases are cloud management related, not VM related. We let $T_{i1} = T_1(m), T_{i2} = T_2(m), T_{i4} = T_4(m), m = 1, 2, \ldots, n$ for convenience. Waiting time in each group should be the maximum time among $\{T_1(m), T_2(m), T_4(m)\}$ (Figure 2), so that there is no overlap between different groups and won't affect each other. Hence the makespan of n migration operations is:

$$\mathbf{C}(\mathbf{M}) = T_{complete}(m) + T_{wait}(m)(\frac{n}{m} - 1), \tag{3}$$

where $T_{complete}$ is the complete time in one group: $T_{complete}(m) = T_1(m) + T_2(m) + T_3^* + T_4(m)$, and T_{wait} is the waiting time between groups: $T_{wait}(m) = \max\{T_1(m), T_2(m), T_4(m)\}$. The optimal number m in one group should be the minimum point of $\mathbf{C}(\mathbf{M})$.

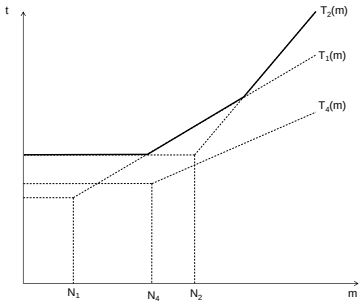

Fig. 2. waiting time function $T_{wait}(m)$

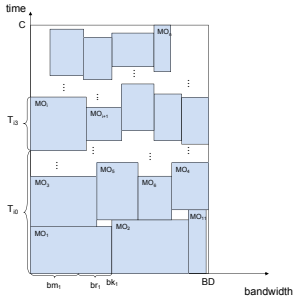

Fig. 3. schedule for migration operations

2.2 Multiple Migration Operations of busy VM

Sometimes, the cloud platform need to migrate busy VMs among servers for load balancing. For cloud platform managed migration operations of busy VM, the live-migration phase is the most influential one than others. We pay attention to migration operation in phase 3 this time. Supposing the migration operation MO_i with memory size VM_i, dirty page rate DR_i and bandwidth of each operation regulated as bm_i, we estimate processing time of live-migration phase T_{i3} by approximate algorithm[3]. As shown in Algorithm 1, once given input of MO_i: VM_i, bm_i, DR_i, the migration time T_{i3} can be estimated by simulating migration operation.

According to approximate algorithm, network bandwidth is the most influential parameter. As network bandwidth decreasing, migration performance starts to degrade rapidly especially for busy VM. We consider following network parameters: network bandwidth BD in cloud, bandwidth bm_i utilized in each operation, and bandwidth br_i reserved to maintain an acceptable quality of service in live-migration phase[5]. bd_i, sum of bm_i and br_i, is a necessity of network bandwidth for both migration performance and service quality during each migration. When migration operations are

performed simultaneously, the number of migration operations m is limited so that the sum of m migration operations' bandwidth bd_i can't exceed network bandwidth BD. To minimize the makespan, our work is to arrange the order and waiting time for migration operations properly.

Now it is kind of strip-packing problem: pack items with various width and height into a big strip which has fixed width and variable height, and the objective is to minimize the height of strip. In Figure 3, we illustrate n migration operations, each having live-migration time T_{i3} and waiting time T_{i0}. The width of strip is bandwidth of network. The height of strip shows the highest operation (the $(n-1)$th operation in Figure 3), the makespan in fact. After packing n migration operations into a 2-D strip composed by time and bandwidth, we will get an optimal schedule. Though strip packing problem is a NP problem, there are still some approximation algorithms such as Next-Fit Decreasing Height (NFDH), or metaheuristic algorithms like annealing and genetic algorithm[4]. Algorithm 2 uses the Bottom-Left (BL) algorithm to find an optimal schedule to minimize the makespan of cloud platform managed migration operations of busy VM.

Algorithm 1 Performance Model for Migration

INPUT: bm_i, VM_i, DR_i **OUTPUT**: T_{i3}
$v_o \leftarrow VM$, $v_{mig}; \leftarrow 0$, $t_{mig} \leftarrow 0$; //Given $iter_{th}$, v_{th} as default values
for $i = 0$ **to** $iter_{th}$ **do**
  $t_i \leftarrow \frac{v_i}{bm_i}$;
  $v_{i+1} \leftarrow t_i \cdot DR_i$;
  $t_{mig} \leftarrow t_{mig} + t_i$;
  $v_{mig} \leftarrow v_{mig} + v_i$;
  if $v_{i+1} \leq v_{th}$ **or** $v_{mig} \geq VM_i$ **then**
    break;
  end if
end for
$T_{i3} \leftarrow t_{mig} + \frac{v_{i+1}}{bm_i}$

Algorithm 2 Scheduling of migration operations

INPUT: $BD, bm_i, br_i, VM_i, DR_i$ **OUTPUT**: $T_{i0}, \mathbf{C}$
for $i = 0$ **to** n **do**
  $T_{i3} = f(VM_i, bm_i, DR_i)$; //$f$ denotes Algorithm 1
end for
$bd_i = bm_i + br_i$;
$[bd_{k_1}, bd_{k_2}, \ldots, bd_{k_n}] = sort([bd_1, bd_2, \ldots, bd_n])$; //sort by non-increasing sequence
for $i = 0$ **to** n **do**
  //find lowest possible position left justified for operation k_i
  $(h_{k_i}, w_{k_i}) = lowleft(bd_{k_i}, T_{i3})$;
  $T_{k_i 0} \leftarrow h_{k_i}$;
end for
$\mathbf{C} \leftarrow \max(h_{k_i} + T_{k_i 3})$;

References

1. Vijayaraghavan Soundararajanand Jennifer M. Anderson, The impact of management operations on the virtualized datacenter, ACM SIGARCH Computer Architecture News, June 2010, 19-23.
2. Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Live migration of virtual machines, Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, 2005, 273-286.
3. Anja Strunk, Costs of virtual machine live migration: A survey, IEEE Eighth World Congress, 2012, 323-329.
4. Andrea Lodi, Silvano Martello, Michele Monaci, Two-dimensional packing problems: A survey, European Journal of Operational Research, 2002, 241C252.
5. David Breitgand,Gilad Kutiel, Danny Raz, Cost-aware live migration of services in the cloud, SYSTOR, 2012