A Scheduling Method for Multiple Virtual Machines Migration in Cloud

Zhenzhong Zhang[1], Limin Xiao[1], Xianchu Chen[1], and Junjie Peng[2]

[1] State Key Laboratory of Software Development Environment,
School of Computer Science and Engineering
Beihang University, Beijing, China
[2] School of Computer Engineering and Science
Shanghai University, Shanghai, China
{zzzhang,chengxc}@cse.buaa.edu.cn,
xiaolm@buaa.edu.cn, jjie.peng@shu.edu.cn

Abstract. Infrastructure as a Service(IaaS) is important in Cloud Computing, which provides on-demand virtual machines(VMs) to users. The resource management plays an important role in IaaS cloud, which deploys and relocates virtual machine on available hosts for different targets, such as load balancing, power saving and resource utilization improving. The virtual machine placement problem can be considered as a bin packing problem. Many researchers use the heuristic algorithms based approach to solve this virtual machine placement problem. However, they all focus on how to find the optimization solution for the bin packing problem of virtual machine placement. These studies did not consider the scheduling of multiple virtual machine migration that involved in the transfer process from one V-P mapping to another. Because of the large overhead produced by virtual machine migration, the optimization of multiple virtual machines migration process could reduce the overhead of resource management in IaaS cloud, and accelerate the migration process. In this paper, we analyse and formal the multiple virtual machines migration problem, and propose a scheduling method to reduce the VM migration times and accelerate the migration process. Experiments show that our method can decrease the VM migration times, reduce the traffic and accelerate the process of multiple virtual machine migration.

Keywords: Cloud Computing, Virtual Machine Schedule, Multiple Virtual Machines Migration, IaaS.

1 Introduction

Cloud computing[1] is a popular trend in current computing which attempts to provide cheap and easy access to computational resources. IaaS(Infrastructure as a Service)[2] provides infrastructure or the actual hardware to customers who are responsible to install operating systems and necessary softwares. Based on virtualization technology[3], IaaS cloud is usually provided to users in the form of Virtual

C.-H. Hsu et al. (Eds.): NPC 2013, LNCS 8147, pp. 130–142, 2013.

Machines (VMs), such as Amazon EC2[4] and VMware vCloud[5]. On IaaS cloud platform, resources are provided by need as services, and it guarantees to the subscribers that it sticks to the Service Level Agreement (SLA). IaaS cloud platform needs to dynamically deploy and relocate virtual machine onto proper physical hosts in order to meet different needs, such as avoid hotspot, power saving and load balancing[5-7]. Therefore, how to dynamically and efficiently schedule virtual machines among physical hosts to meet the needs of different targets becomes a problem.

The traditional resource management methods usually schedule virtual machine or allocate resource in cloud when some certain conditions are triggered[8-10], such as threshold for load balancing. Due to the complexity and variability of a large number of virtual machines in IaaS cloud, traditional methods are difficult to carry out global resource optimization management. To address this issue, many optimization theory based virtual machine placement approaches are used to solve the resource management problem in IaaS cloud[11, 12]. In such scenario, the resource management problem is considered as a bin packing problem, which need to find the proper mapping of virtual machines to available physical machines.

Linear programming[13], genetic algorithm[14] and ant colony algorithm[15] have been used to solve the bin packing problem, and obtained good results. All these researches are focused on how to find the actual mapping of virtual machines to available physical machines(V-P mapping). However, these studies did not consider the scheduling of multiple virtual machine migration that involved in the transfer process from one V-P mapping to another. Because the virtual machine migration process needs to copy large amounts of data(memory data or even virtual disk) from the source host to the destination host, it will produce large CPU overhead and network traffic, and cost much resource. Therefore, the optimization of multiple virtual machines migration process could reduce the overhead of resource management in IaaS cloud and accelerate the migration process. The optimization includes reducing the number of virtual machine migration, and migrating small VM instead of big VM.

In this paper, we study the multiple virtual machines migration problem, and propose a scheduling method to optimize multi-virtual machine migration process. The main contributions of this paper are concluded as the followings: (1)Modeling and formalization of the multiple virtual machines migration problem; (2)A scheduling method optimizing the multiple virtual machines migration process.

2 Related Work

For resource management in cloud data center, previous work has focused on the problem of placing and replacing VMs in servers, in order to optimize resource management for different criteria, including performance, power and cost. There are some molded products and research projects on virtualized resource management, such as VMware DRS[8]. They dynamically allocate the CPU, memory and I/O resources to partitioned virtual machines according to customer's requirements, but they ignore the QoS[9]. The work in [16] minimized the number of physical machines using dynamic

adaptation technique based on off-line analysis of application performance, which is seen as a function of machine utilization. The load forecasting techniques are also widely used in the management of cloud resources, such as load-balancing and resource scheduling[17]. In [18], a model-predictive controller is proposed to minimize the total power consumption of the servers in an enclosure subject to a given set of QoS constraints.

Optimization and heuristic methods are wildly used by virtual machine scheduling and placement in IaaS cloud for different targets, such as load balancing or power saving. Integer Linear Programming is used to solve an interference-aware VM placement problem(IAWMP)[13]. They first formulate this problem by an Integer Linear Programming (ILP) model to solve it optimally. They also propose a polynomial-time heuristic algorithm to efficiently solve the IAWMP problem. In [14] a general model is proposed for resources allocation of virtual machines in multi-tier distributed environments. Their model describes each virtual machine and each physical host by a multi-dimensional resource vector, allowing the coexistence of both quantitative and qualitative resources, also handling different SLAs. [15] proposes a multi-objective ant colony system algorithm for the virtual machine placement problem. Their method could efficiently obtain a set of non-dominated solutions that simultaneously minimize total resource wastage and power consumption. [19] proposes a runtime virtual machine mapping framework(GreenMap), and designs a probabilistic, heuristic algorithm to mapping VMs onto a set of physical machines under the constraint of multi-dimensional resource consumptions.

However, all above optimization and heuristic researches are focus on how to find the actual mapping of virtual machines to available physical machines(V-P mapping). Although some virtual machine migration research[20, 21] can help to accelerate virtual machine migration process, however, the scheduling of multiple virtual machine migration that involved in the transfer process from one V-P mapping to another was rarely considered. Therefore, in this paper, we will study the multiple virtual machines migration problem, and propose a scheduling method to optimize multi-virtual machine migration process.

3 Problem Analysis and Formulation

3.1 Analysis of Multiple Virtual Machines Migration Problem

In this paper, the heuristics-based resource management methods are consist of two steps, which is shown in Figure 1. Firstly, the bin packing problem of virtual machine management is solved by a heuristic based algorithm, and the global approximate optimal virtual machine groups are obtained. The VMs mapping to same physical host are in same group, called a VM-cluster. Next, For the global optimal virtual machine grouping, we also need to convert the virtual machines to physical host mapping (V-P mapping) from current state(initial VM location) to the target state(final VM location).

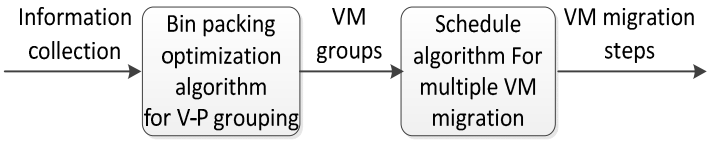

Fig. 1. The process of VM placement in IaaS cloud

The same VM group can locate on different hosts, therefore the target state is not unique. This process of V-P mapping convert involves the migration of multiple virtual machines, which could be scheduled to reduce the migration cost of VMs. Our study is focused on the optimization of the multiple virtual machines migration process. The purpose of this optimization is to minimize the overhead of CPU and network cost and avoid resource conflicts during the migration process(migration the number of virtual machines, choose the smaller virtual machine migration, adjust virtual machine migration steps to avoid conflicts). The optimization includes the choosing of VM for migration, the selecting of destination physical host, and accelerate the migration process of all VMs. For example, If several V-P mappings have same effect, we will choose the one that has minimum number of VM migration. If several V-P mapping have same VM migration steps, the one with minimal numbers of VM migration times will be chosen.

We need to develop a model to describe the scheduling problem of multiple virtual machines migration. In this model, the virtual machines which belong to same physical host are in the same group, and called a VM-cluster. At the initial moment, the VM-clusters of all virtual machines in IaaS cloud are the initial state of V-P mapping. Our schedule method is responsible for converting the V-P mapping from one type of VM-cluster to another VM-cluster. In this process, the proper physical host should be chosen for every VM-cluster, and the proper order of virtual machine migration determined, which is equivalent to create a new mapping of VM-clusters to physical hosts. Of course, this new mapping must meet the demand that the number of VM migration and the amount of traffic are as small as possible.

3.2 Formal Description of the VM Schedule Problem

We formalize the virtual machines to physical hosts mapping problem in this section. Firstly, we define some basic objects below:

- Host is denoted by H_k(k=1,2...,N), and the set of hosts is H={H_1,H_2,...,H_N},(N>=1). V_j denotes all the virtual machines that need to be migration. The set of virtual machine is V = {V_1,V_2,…,V_n}(n>=1). The virtual machines that locate on the same physical host form a virtual machine set. We call this set the VM-cluster which is denoted by C_i(i=1,2...,M). The set of VM-cluster is C={C_1,C_2,…,C_M}. In order to simplify this model, we require M always be no more than N, that is the number of VM-clusters should be less than or equal to the number of hosts.
- L(V_j) is the amount of network traffic that generated by the migration of virtual machine V_j. Cost is the total amount of traffic generated by multiple virtual machines migration. In the initial state, each virtual machine running on a particular station host to form a set of virtual machine clusters.

- At the initial state, each virtual machine running on a particular host to form a set of VM-cluster. And it will be converted to another set of VM-cluster by our schedule method. If the virtual machine V_j belongs to a cluster C_i, it is denoted as $V_j \in C_i$. The optimal mapping of virtual machines to physical hosts will be selected by our scheduling method.

3.3 The VM Scheduling Method

After management strategy at initial state, all the virtual machines in the system are re-divided into a set of VM-clusters(shown as $V_j \in C_i$, $C_i \in C$). The establishment of mapping of virtual machines to physical hosts is equivalent to choose the suitable of physical host H_k for each VM-cluster C_i. Our multiple virtual machine migration scheduling method need to create mapping of VM-cluster set(C) to physical host set(H) as f(M→N), and select the optimal one that meet our targets(minimum migrations, etc.). Each mapping establishes a relationship between VM-clusters to hosts, and determines the target hosts that each virtual machine will migrate to. Therefore migration path of the virtual machines is also determined. Because the mapping of VM-cluster(C, the number of C is M) to physical hosts(H, the number of H is N) will generate a large number of virtual machine migration paths(A_n^m). Our scheduling algorithm needs to search for the optimal solution in all migration paths, and we will use heuristic approaches to simplify the process. After the optimal migration path is obtained, our scheduling algorithm will further determine the optimal virtual machine migration steps based on the principle of minimum system cost.

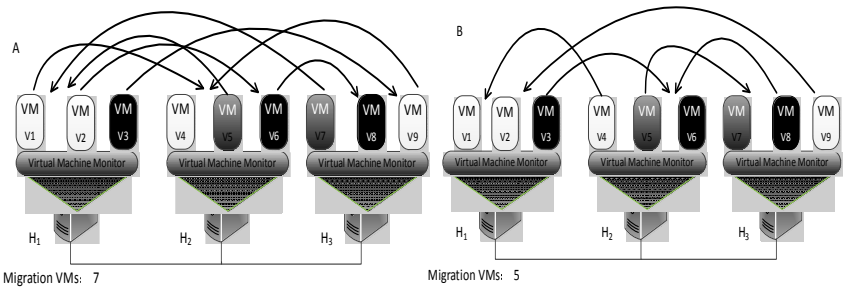

Fig. 2. The migration path of f1 and f2

We show a simple examples here, to describe our method in detail. As a case that virtual machines V1-V9 run on three hosts(H1, H2, H3). After bin packing optimization, these virtual machines are divided into three clusters(C1:V1,V2,V4,V9; C2:V3,V6,V8;C3:V5,V7). Two different mappings are available, the mapping of f1 is (C1→H2,C2→H3,C3→H1), and the mapping of f2 is (C1→H1,C2→H2,C3→H3). F1 and f2 are shown in Figure 2.A and 2.B. These two mappings correspond to two multiple virtual machine migration paths. These two migration paths choose different physical hosts for VM-clusters, resulting in a different number of VM migrations. Obviously, the path of f2 needs to migrate 5 VMs, which is less than 7 VMs of f1.

When the two mappings have the same VM migration times, the different size and load of virtual machines will also lead to a larger difference of network traffic and CPU load. For example, the migration of virtual machine with large memory and higher workload needs to copy more data and consume more CPU resources. $L(V_j)$ indicates the network traffic generated by virtual machine V_j during migration. The total transfer data of fx is Cost(fx), which is the sum of all $L(V_j)$. Like the first step, the cost of each migration paths are estimated and the one with minimum migration traffic would be chosen.

While the virtual machines starts migrate, the migrating virtual machines which have same source or destination host would migrate one by one. Because the node migration performance of host would decreased significantly due to simultaneous execution of multiple virtual machine migration. And of course, the virtual machines which their migration do not conflict with each other, could migrate at same time.

4 Design and Implementation

4.1 Host Selection Algorithm for Virtual Machine

The host selection algorithm for virtual machine is used to find the optimal mapping from all possible V-P mapping, and to obtain the proper destination hosts for the VMs. Specifically, it needs to search all A_n^m mappings of VM-cluster(C, the number is M) to physical hosts(H, the number is N), and selects the optimal mapping as the multiple virtual machine migration path. The optimal selection principle is minimum migration times of multiple migration process and minimum amount of migration network traffic(ie, the migration of VMs with smaller size will generate less network traffic). An improved genetic algorithm is be used to search the best mapping solution in this case. Due to limited space, the details of the genetic algorithm are not shown here.

```
1.  Input: fs ∈{fs| 0 < s <= MN}
2.  for each fs ∈{fs| 0 < s <= M^N} do
3.      for each j (1<= j <= n) do
4.          if (The current host of Vj is not the host of Vj in fs mapping)
5.              Yj' = The host ID of Vj belong to in fs mapping;
6.          Else
7.              Yj' = Yj;
8.      End for  /* We can get <Y1',Y2',…,Yn'>  */
9.      if ( ∑_{j=0}^{n} Yj ⊕ Yj' == min∑_{j=0}^{n} Yj ⊕ Yj' )
10.         insert f into collection F;
11. End for
12. for each f in F do
13.     Cost(f) <= Min(F);
14.     Min(F) = Cost(f);
15. End for
16. Output: Min(F)
```

Fig. 3. The algorithm for Optimal mapping selection

The pseudo-code of our based algorithm is shown in Figure 8. The algorithm traverses all A_n^m mappings (lines 1-11) to obtain the minimum value of VM migration times and generates the set F which contains all the mappings with the minimum migration times (lines 8-9). Finally, it calculates the network traffic(Cost(f)) for each mapping in F, and select the mapping with minimum network as the return result of the algorithm (lines 12-15). In the algorithm, V_j is the virtual machine with number j, Y_j is the host where the virtual machine V_j current locates, fs is a mapping of virtual machines to hosts, Y_j' is the host that V_j will migrate to with the fs mapping, F is a set of mapping with minimum VM migration times, Cost(f) is the network traffic generate by VMs migration of mapping f and Min(F) is the mapping with minimum Cost(f) in set F.

4.2 The Migration Order and Parallelization of Multiple VMs Migration

After the selection of target hosts for multiple virtual machine hosts migration, the schedule algorithm next needs to determine the migration order of these virtual machines to be migrated. Our scheduling algorithm will generate a trituple $<V_j, H_j^s, H_j^d>$ for each virtual machine that needs to be migrated. V_j is the virtual machine need to be migrated, H_j^s and H_j^d are the source and destination hosts for migration of V_j. The order of multiple virtual machine migration could be represented by this trituple sequence. Our algorithm is mainly based on the following two principles to arrange the order of multiple virtual machine migration, and execute the migration process.

Firstly, the scheduler give priority to migrates virtual machine on high-load host to the low-load host, and give priority to migrates virtual machines with higher workload and bigger size. The migration of virtual machine will cost much CPU resource both on source and destination hosts. While priority migrate virtual machine on high-load host to low-load host, we need only consider the impact of migration on the source host(high-load host). And on the other hand, if migrate VM to a high-load host, there may be no enough resource remained for this virtual machine. Meanwhile, the scheduler priority migrate virtual machine with higher workload on source host. This is mainly because the virtual machine with higher workload is more sensitive to resource competition. A bit more workload increase will cause the performance of the application decline. But for the application on low-load virtual machine, the workload increase is tolerable. And based on the queuing theory, the VM with bigger memory size has less priority to migrate. Therefore, according to the above aspects, the scheduler can minimize the impact of VM migration on application performance.

Secondly, the parallel migration can be used for the virtual machines whose migration do not interfere with each other. For performance and stability reasons, a physical host can only deal with one virtual machine migration, either as an source host or as a destination hosts. Therefore, we can refer to the realization of processor's instruction-level parallelism to parallel processing the multiple virtual machines migration. In this case, the trituple sequence of migration VM is handled as the sequence of instructions of processor. When a VM migration is in processed, the source and destination hosts are marked busy, and reset free after VM migration. Therefore, while the source and

destination hosts of the trituple being processed are not busy, the migration of this trituple could be executed immediately. Based on this strategy, our method can achieves parallel virtual machine migration, and avoid resource conflict.

5 Experiment and Evaluation

In this section, we validate and test the multiple virtual machine migration scheduling method, then analysis its performance. Using different test cases of multiple virtual machines migration to verify the effect of our schedule algorithm. The performance of our method is compared with default migration method, which migrate virtual machines one by one randomly.

For our experiment environment, we use a cluster composed by eight computer servers and one storage array. The configuration of server includes two AMD Opteron 2350 quad-core CPUs running at 2.0GHz and 12GB DDR RAM. They are all running XenServer 6.1[22]. The storage array connects four hosts through optical fiber, as a shared storage. All the servers are connected by a Gigabit LAN. The virtual machine templates are configured with one VCPU, 1GB,2GB,4GB RAM and one virtual network card. The load generator program will randomly call some of the popular applications to generate the CPU, network and disk I/O workload, such as kernel compilation, file compression, and FTP, etc.

5.1 Verify the Effectiveness of Our Algorithm

We select a test case to validate effectiveness of our scheduling algorithm. This test case uses 12 virtual machines and 8 physical hosts. The detail configuration of each virtual machine and physical host are shown in Table 1. These configurations of test case include input of algorithm, memory of VM and physical, the mapping of virtual machine to physical before and after migration, the workload of each virtual machine and physical host and the output of algorithm at stage 1 and 2.

Table 1. The configuration of test case

Input of VM-cluster	C1:v1,v2,v3,v10;C2:v4,v5,v6,v7;C3:v8,v9,v11,v12		
Information of VMs			
VM ID	Mem(MB)	Init reside host ID	Load(%)
V1	1024	H1	30%
V2	1024	H1	60%
V3	1024	H4	50%
V4	2048	H6	15%
V5	2048	H4	80%
V6	2048	H4	50%
V7	3072	H6	40%
V8	3072	H2	35%
V9	3072	H2	45%
V10	4096	H5	70%
V11	4096	H7	50%
V12	4096	H2	10%

Table 1. (*continued*)

Information of physical hosts before VMs migration			
Host ID	Mem(GB)	VMs	Load(%)
H1	12	V1,V2	15%
H2	12	V8,V9,V12	15%
H3	12	None	0%
H4	12	V3,V5,V6	30%
H5	16	V10	10%
H6	16	V4,V7	10%
H7	16	V11	10%
H8	16	None	0%
Information of physical hosts after VMs migration			
Host ID	Mem(GB)	VMs	Load(%)
H1	12	V1,V2,V3,V10	35%
H2	12	V8,V9,V11,V12	25%
H3	12	None	0%
H4	12	None	0%
H5	16	None	0%
H6	16	V4,V5,V6,V7	30%
H7	16	None	0%
H8	16	None	0%
Output of stage 1: V10-->H1, V3-->H1, V11-->H2, V5-->H6, V6-->H6			
Output of stage 2: V5,V3,V6,V10,V11			

The output of our algorithm is listed in Table 1. Because our algorithm is divided into two stages, our validation also has two stages.

Firstly, the output of the first stage of our algorithm is a set of V-P mapping which has the minimum virtual machine migration times. For this test case, the minimum migration times is 5, and the mapping in the set is (f1: V10-->H1, V3-->H1, V11-->H2, V4-->H4, V7-->H4) and (f2: V10-->H1, V3-->H1, V11-->H2, V5-->H6, V6-->H6). Then the algorithm compares the migration traffic of each mapping, and chooses the V-P mapping with minimum migration traffic as the optimal migration path. In this test case, the Cost(f1) = L(V10)+L(V3)+L(V11)+L(V4)+L(V7) = 14GB, and Cost(f2) = L(V10)+L(V3)+L(V11)+L(V5)+ L(V6) = 13GB. Because Cost(f1) is greater than Cost(f2), our algorithm will select the f2 as the optimal migration path for multiple virtual machine migration.

Table 2. The trituple Sequence for migration step

Id	Trituple	Load(VM/Host)	Mem(MB)
1	(V10,H5->H1)	70%/10%	4096
2	(V3,H4->H1)	50%/30%	1024
3	(V11,H7->H2)	50%/10%	4096
4	(V5,H4->H6)	80%/30%	2048
5	(V6,H4->H6)	50%/30%	2048
Migration step: V5,V3,V6,V10,V11			

Table 3. The parallel migration process for multiple virtual machine migration

Time	Migrating VMs/ Mem(G)	Waiting VMs	Busy Hosts
T1	V5(2G),V10(4G),V11(4G)	V3(1G),V6(2G)	H4,H6,H5,H1,H7,H2
T2	V5(2G),V10(4G),V11(4G)	V3(1G),V6(2G)	H4,H6,H5,H1,H7,H2
T3	V10(4G),V11(4G),V6(2G)	V3(1G)	H5,H1,H7,H2,H4,H6
T4	V10(4G),V11(4G),V6(2G)	V3(1G)	H5,H1,H7,H2,H4,H6
T5	V3(1G)	None	H4,H6

The second stage of the algorithm is to generate a virtual machine migration sequence based on the optimal mapping f2 which obtained by the first stage. The migration sequence is shown as a trituple sequence in Table 2. According to our optimization strategy, the migration order output by the second stage of out algorithm is V5, V3, V6, V10, V11. This is because the virtual machine V5 has a higher workload, and the physical host V5 reside on has the highest workload in the physical hosts. The virtual machine V10 and V11 migrate at last, due to the large memory(4098MB).

The Table 3 shows the parallel migration process of multiple virtual machines based on the migration order. We divide virtual machine migration process to several periods. During each period, 1GB data could be transferred. As Table shows, the migrating VMs, waiting VMs and physical hosts which be occupied for each period during migration process. In Table 3, at T1 period, the migration of virtual machine V5 is executed. However, VM V3 and V6 are waiting due to resource confliction, and VM V10 and V11 can migrate at T1 period without interference with each other. At the beginning of T3 period, VM V5 finishes the migration. The next VM V3 could not start migration because of the destination host(H1) of V3 is busy. Thus, the VM V6 starts migration at T3 period. The VM V3 finishes migration at end of T5 period. The total time of the parallel migration are 5 periods. Compared with the 13 periods of serial migration, the parallel virtual machine migration greatly reduces migration time(160%), and avoids resource conflicts in multiple virtual machine migration.

5.2 Verify the Versatility of Our Algorithm

We choose several test cases to validate the algorithm's versatility. The detailed configuration of these test cases are shown in Table 4. 30 virtual machines are running on three physical hosts, each VM has 1GB memory. The VM-cluster inputs of five test cases are shown in Table 5.

Table 4. The distribution state of virtual machines

VM/Host	Host1	Host2	Host3
VM ID	V1~V10	V11~V20	V21~V30

Table 5. Five test cases

ID/ VM-cluster	C1	C2	C3
Case 1	1,11,12,13,14,15,16,17, 18,19	2,3,4,5,6,7,8,9,10,20,30	21,22,23,24,25,26,27,28,29
Case 2	1,2,3,5,11,12,14,17,21,2 4,25	4,13,15,16,18,19,20,22,23 ,26,28	6,7,8,9,10,27,29,30
Case 3	1,2,3,5,24,25	4,13,15,16,18,19, 26,28,11,12,14,17,21	6,7,8,9,10,27,29,30,20,22, 23
Case 4	1,14,17,21,24,25,16,18, 19,20	4,13,15, 22,23,26,28,7,8,9	6,10,27,29,30,2,3,5,11,12
Case 5	1,12,14,17,21,24,19,20, 22, 9,10,27,25	4,13,15,2,3,5,11,16,18, 23,26,28	6,7,8, 29,30

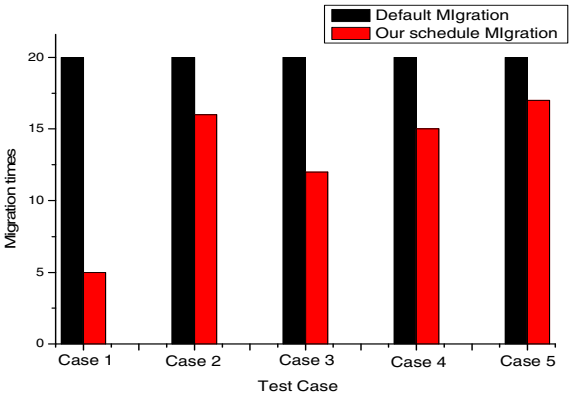

Fig. 4. Migration time of five time cases

The results of migration time for each test case are shown in Figure 4. We can see that, compared with default migration strategy, the average VM migration times of our algorithm is lower. The degree of optimization is different for each test case depending on the initial mapping state. This experiment shows that in most cases, our algorithm can optimize the process of multiple virtual machine migration.

6 Conclusion and Future Work

This paper presents a schedule method for multiple virtual machine migration. The contributions of this paper include:(1) We analyze and formal the problem of multiple virtual machine migration; (2) We propose a schedule algorithm for multiple virtual machine migration. The algorithm contains two parts. The first is finding the minimum cost VM migration path from all mapping of virtual machine to physical hosts, and the second an algorithm generating the optimization virtual machine migration sequence, and accelerating multiple virtual machine migration process based on parallelize techniques; (3) We use some experiments to verify the effectiveness and versatility of the algorithm.

In this paper, the algorithm proposed has high complexity, and it's performance is not good while dealing with large-scale multi-virtual machine migration case. Therefore, we need to improve the scalability of the our method.

Acknowledgments. This work is supported by the National Natural Science Foundation of China(61232009); the State Key Laboratory of Software Development Environment (SKLSDE-2012ZX-07); the Doctoral Fund of Ministry of Education of China (20101102110018); the Hi-Tech Research and Development Program (863) of China (2011AA01A205); the Beijing Natural Science Foundation(4122042); Shanghai Science and Technology Innovation Action Plan(11511500400)

References

1. Armbrust, M., et al.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
2. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1(1), 7–18 (2010)
3. Rosenblum, M., Garfinkel, T.: Virtual machine monitors: Current technology and future trends. Computer 38(5), 39–47 (2005)
4. Amazon Elastic Compute Cloud, `http://aws.amazon.com/en/ec2/` (accessed May 2013)
5. vCloud, `http://en.wikipedia.org/wiki/VCloud` (accessed May 2013)
6. Ren, X., Lin, R., Zou, H.: A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. In: International Conference on Cloud Computing and Intelligence Systems (2011)
7. Srikantaiah, S., Kansal, A., Zhao, F.: Energy aware consolidation for cloud computing. In: USENIX Conference on Power Aware Computing and Systems (2008)
8. Vmware. Resource management with VMware DRS. VMware Whitepaper (2006)
9. Song, Y., Li, Y., Wang, H., Zhang, Y., Feng, B., Zang, H., Sun, Y.: A service-oriented priority-based resource scheduling scheme for virtualized utility computing. In: Sadayappan, P., Parashar, M., Badrinath, R., Prasanna, V.K. (eds.) HiPC 2008. LNCS, vol. 5374, pp. 220–231. Springer, Heidelberg (2008)
10. Zhang, Z., et al.: A VM-based Resource Management Method Using Statistics. In: International Conference on Parallel and Distributed Systems (2012)
11. Bobroff, N., Kochut, A., Beaty, K.: Dynamic placement of virtual machines for managing sla violations. In: International Symposium on Integrated Network Management (2007)
12. Verma, A., Ahuja, P., Neogi, A.: pMapper: power and migration cost aware application placement in virtualized systems. In: Issarny, V., Schantz, R. (eds.) Middleware 2008. LNCS, vol. 5346, pp. 243–264. Springer, Heidelberg (2008)
13. Lin, J.W., Chen, C.H.: Interference-aware virtual machine placement in cloud computing systems. In: International Conference on Computer & Information Science 2012 (2012)
14. Campegiani, P., Presti, F.L.: A general model for virtual machines resources allocation in multi-tier distributed systems. In: International Conference on Autonomic and Autonomous Systems (2009)
15. Gao, Y., et al.: A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences (2013)

16. Khanna, G., et al.: Application performance management in virtualized server environments. In: Network Operations and Management Symposium (2006)
17. Prevost, J.J., et al.: Load prediction algorithm for multi-tenant virtual machine environments. In: World Automation Congress, WAC (2012)
18. Wang, X., Wang, Y.: Coordinating power control and performance management for virtualized server clusters. Transactions on Parallel and Distributed Systems 22(2), 245–259 (2011)
19. Liao, X., Jin, H., Liu, H.: Towards a green cluster through dynamic remapping of virtual machines. Future Generation Computer Systems 28(2), 469–477 (2012)
20. Liu, H., et al.: Performance and energy modeling for live migration of virtual machines. In: International Symposium on High Performance Distributed Computing (2011)
21. Jin, H., et al.: Live virtual machine migration with adaptive, memory compression. In: International Conference on Cluster Computing and Workshops (2009)
22. Xenserver, `http://www.citrix.com/products/xenserver/overview.html` (accessed May 2013)