2012 ACM/IEEE 13th International Conference on Grid Computing

VC-Migration: Live Migration of Virtual Clusters in the Cloud

Kejiang Ye, Xiaohong Jiang, Ran Ma, Fengxi Yan
College of Computer Science, Zhejiang University
Hangzhou 310027, China
{yekejiang,jiangxh,maran,yanfengxi}@zju.edu.cn

***Abstract*—Live migration of virtual machines (VM) has recently become a key ingredient behind the management activities of cloud computing system to achieve the goals of load balancing, energy saving, failure recovery, and system maintenance. However, to our knowledge, most of the previous live VM migration techniques concentrated on the migration of a single VM which means these techniques are insufficient when the whole virtual cluster or multiple virtual clusters need to be migrated. This paper investigates various live migration strategies for virtual clusters (VC). We first describe a framework *VC-Migration* to control the migration of virtual clusters. Then we perform a series of experiments to study the performance and overheads of different migration strategies for virtual clusters, including concurrent migration, mutual migration, homogeneous VC migration, and heterogeneous VC migration. After that, we present several optimization principles to improve the migration performance of virtual clusters. The HPCC benchmark is selected to represent the virtual cluster workloads, and the metrics such as downtime, total migration time, and workload performance are measured. Experimental results reveal some new discoveries which are useful to the future development of new migration mechanisms and algorithms to optimize the migration of virtual clusters.**

***Keywords*-virtual machine; virtual cluster; live migration; performance; cloud computing;**

I. INTRODUCTION

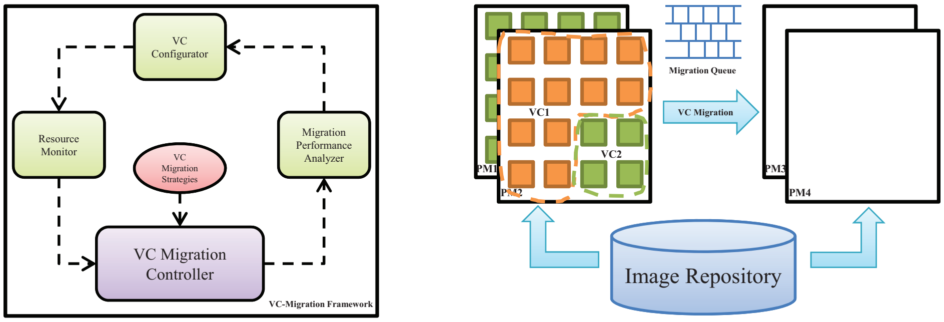
Cloud computing [1] has currently attracted considerable attention from both the industrial community and academic community. In this new computing paradigm, all the resources are delivered as the services (Infrastructure Service, Platform Service, and Software Service) to the end users via the Internet. Virtualization [2, 3] is a core technique to implement the cloud computing paradigm. Virtualization provides an abstraction of hardware resources enabling multiple instantiations of operating systems to run simultaneously on a single physical machine. Another prominent advantage of the virtualization is the live migration technique [4, 5] which refers to the act of migrating a virtual machine from one physical machine to another even as the virtual machine continues to execute. Currently, live migration has become a key ingredient behind the management activities of cloud computing system to achieve the goals of load balancing, energy saving, failure recovery, and system maintenance [6].

Virtual Cluster (VC) [7, 8] is a group of virtual machines configured for a common purpose, such as high performance computing (HPC) or parallel computing [9], with associated storage resource, operating system, software environment, communication protocol, and network configuration. Due to the benefits brought by the virtualization technology, it becomes more and more popular to run high performance computing workloads on virtual clusters. Two notable features of virtual cluster are *large scale* and *intensive communication* which are challenging for the live migration of virtual clusters.

Live migration of virtual clusters faces several new challenges: *(i) Huge Amount of Data.* The virtual cluster needs to transfer large volumes of memory data due to the large cluster size, e.g., a 16-node virtual cluster with 512MB DRAM for each virtual machine needs to transfer 16*512MB=8GB data across the network which is far more than the data volume transferred in the single virtual machine migration. *(ii) Limitation of Network Bandwidth.* When multiple virtual machines need to migrate concurrently to another physical machine, the network will become overloaded and the applications will suffer from the degraded performance. *(iii) Intensive Communication between VMs.* In a virtual cluster, all the virtual machines need to communicate with each other to solve a huge task via specific communication engine such as MPI or MapReduce [10]. *(iv) Synchronous Latency.* Most of the applications running on the virtual cluster are in parallel which need to be synchronized with each virtual machine at regular intervals to maintain a consistent state. Although the downtime of live migration is very short, it indeed can affect the application performance, especially for the parallel applications. *(v) Complex VC Migration Strategies*. When multiple virtual clusters need to migrate from one host to another, the migration order and the migration patterns will be different under different migration strategies, such as sequence migration or concurrent migration.

In this paper, we study the performance and overheads of live migration of virtual clusters from experimental perspective and investigate different VC migration strategies. We describe a framework *VC-Migration* to control the migration of virtual clusters. Based on this framework, we perform a series of experiments to understand the performance bottleneck and overheads of virtual cluster migration. We first study the performance characterization of virtual cluster, including the performance of cross-domain virtual cluster

1550-5510/12 $26.00 © 2012 IEEE
DOI 10.1109/Grid.2012.27



(a) VC-Migration Framework

(b) VC-Migration Scenario

Figure 1. VC-Migration Framework for the Live Migration of Virtual Cluster: (a) The Framework Modules; (b) An Example of Virtual Cluster Migration.

and the scalability of virtual cluster. Then we study the dynamic migration strategies of virtual clusters, including migration scalability, concurrent migration granularity, mutual migration. After that, we investigate the migration scenario of multiple virtual clusters, including homogeneous VC migration, and heterogeneous VC migration. We also compare the migration performance of master node and slave node. Experimental results reveal some new discoveries. Based on the experimental results, we propose several optimizations to improve the migration performance and reduce the migration overheads of virtual clusters.

The rest of the paper is structured as follows. In Section II, we present a framework for the live migration of virtual clusters and describe various virtual cluster migration strategies. In Section III, we study the performance characterization of virtual cluster. In Section IV, we perform a comprehensive evaluation and analysis on different virtual cluster migration strategies. In Section V, we focus on the multiple-VC migration scenarios and study the migration efficiency of multiple virtual clusters. Section VI presents the related work. Finally we give our conclusion and future work in Section VII

II. VC-MIGRATION FRAMEWORK

In this section, we propose a framework for the live migration of virtual clusters - *VC-Migration*, and design various VC migration strategies according to the different VC migration scenarios.

A. Framework Design

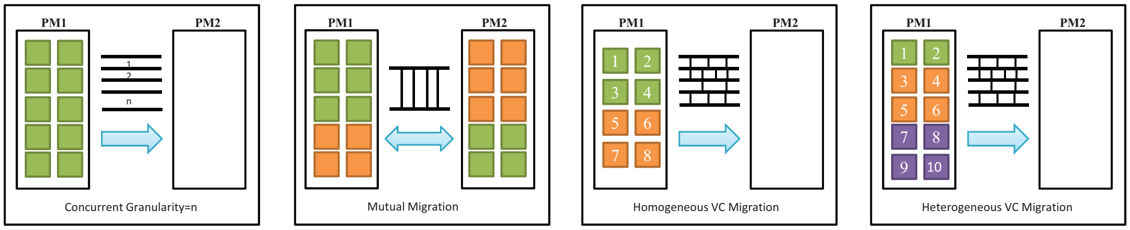
Figure 1 illustrates the live migration framework for the virtual clusters. It consists of five main modules: *VC Migration Controller*, *Resource Monitor*, *Migration Performance Analyzer*, *VC Configurator*, *VC Migration Strategies*. All the five modules corporate with each other to complete the live migration of virtual clusters. Figure 1(b) shows an example of live migration of virtual clusters. In this example, physical machine PM1 and PM2 host several virtual clusters, such as VC1 and VC2. When the virtual clusters need to be migrated to other physical machines, such as PM3 and PM4, all the virtual machines belonging to the migrated virtual clusters need to queue in the *Migration Queue*. There may exist several parallel migration queues. Each virtual machine in the same migration queue is migrated in sequence. All the virtual machines images are stored in a separate shared storage *Image Repository*.

VC Migration Controller: is the main module in the migration framework. It controls the entire migration process of virtual clusters, such as when and how to migrate the virtual machines belonging to different virtual clusters. The detailed migration strategies are made by the *VC Migration Strategies* module. *VC Migration Controller* executes the detailed migration process, such as concurrent migration. It is implemented by encapsulating the migration interface provided by the virtual machine monitor (VMM), such as Xen virtual machine monitor [2].

Resource Monitor: is responsible for monitoring the resource status of both the source machines and destination machines. The utilizations of CPU resource, memory resource, disk I/O resource, and network resource are monitored. The virtual cluster size and configuration information are also recorded to make efficient migration decisions. This module is implemented by using the *Monitor Module* of the *vTestkit* toolkit that is a performance benchmarking tool for the virtualization environments [11].

Migration Performance Analyzer: measures the migration performance of virtual clusters. The metrics such as downtime, total migration time, and application performance are measured to analyze the migration performance and overheads which can feedback useful information to the *VC Migration Strategies* to adjust the migration strategies of virtual clusters. This module is implemented by extended the *Virt-LM* benchmark [12] which is a research benchmark for the live migration of single virtual machine. We add additional functionalities such as multiple virtual machines migration and concurrent execution.

VC Configurator: is responsible for adjusting the configuration of virtual clusters. The adjustment will be done



(a) Concurrent Migration Granularity **(b) Mutual Migration** **(c) Homogeneous Multi-VC Migration** **(d) Heterogeneous Multi-VC Migration**

Figure 2. Virtual Cluster Migration Scenarios: (a) Concurrent Migration Granularity; (b) Mutual Migration; (c) Homogeneous Multi-VC Migration; (d) Heterogeneous Multi-VC Migration.

according to the results generated by the *Migration Performance Analyzer*. It can be implemented by re-configure the configuration parameters of virtual clusters and virtual machines themselves.

VC Migration Strategies: is responsible for making proper migration strategies according to specific conditions. The strategies might be affected by the virtual cluster size, the weight (or importance) of virtual clusters, workload characterization, and other requirements. The *VC Migration Strategies* solves the problems of migration order, concurrency granularity, etc. The detailed VC migration strategies will be provided in Section II-B.

B. VC Migration Strategies

In this section, we describe four typical virtual cluster migration strategies, they are *concurrent migration with various granularity*, *mutual migration*, *homogeneous multi-VC migration*, and *heterogeneous multi-VC migration* (see Figure 2). When multiple virtual machines in one virtual cluster or multiple virtual clusters need to be migrated, it is necessary to choose an efficient way to migrate all the virtual machines with the best performance and least overheads.

1) Concurrent Migration with Various Granularity: Concurrent migration (see Figure 2(a)) is common when the virtual cluster size scales. Is the concurrent migration performs better than sequence migration? Which is the best concurrency degree or granularities when migrating the whole virtual cluster? We will investigate different concurrent degrees to quantify the performance of virtual cluster migration.

2) Mutual Migration: Mutual migration may occur when both the two virtual clusters on two physical machines need to migrate to each other physical machine simultaneously. Figure 2(b) shows an example of mutual migration that 10VM on PM1 need to migrate to PM2 while the other 10VM on PM2 need to migrate to PM1 simultaneously.

3) Homogeneous Multi-VC Migration: When there are more than one virtual cluster in the cloud need to migrate, the situation becomes more complex. We divide the multi-VC migration scenario into homogeneous migration and heterogeneous that the virtual cluster itself is homogeneous and heterogeneous respectively. In Figure 2(c), two homogeneous 4-node virtual clusters on PM1 both need to migrate to PM2. Which migration order is the best: *node by node* (1, 2, 3, 4, 5, 6, 7, 8) or *cluster by cluster* (1, 5, 2, 6, 3, 7, 4, 8) or concurrent virtual cluster migration (take Granularity 2 for example, 1 2, 3 4, 5 6, 7 8 or 1 5, 2 6, 3 7, 4 8)?

4) Heterogeneous Multi-VC Migration: Differ from the the homogeneous multi-VC migration, the heterogeneous multi-VC migration faces new problems, such as small-size VC migrates first or using the same strategies as the homogeneous multi-VC migration?

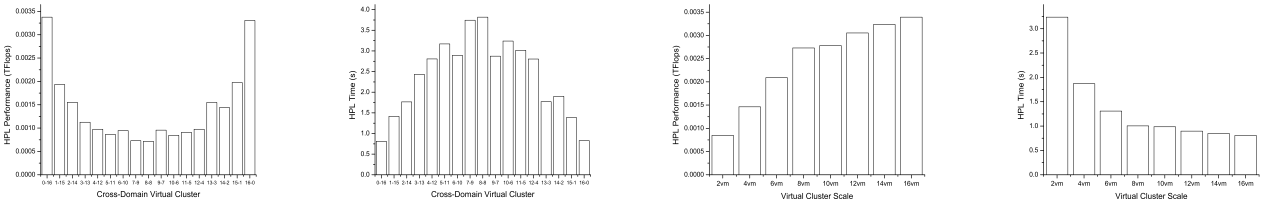
All the above migration strategies of virtual clusters will be investigated in the following sections.

C. Experimental Configuration

1) Virtual Cluster Configuration: All the experiments are performed on two Dell T710 server, with 2 Quad-core 64-bit Xeon processors E5620 at 2.40GHz and 32GB DRAM. We use CentOS 5.6 with kernel version 2.6.18-238.12.1.e15xen in Domain 0, and Xen 3.3.1 as the hypervisor. Each virtual machine is installed with Ubuntu 8.10 as the guest OS with the configuration of 1VCPU and 256MB DRAM. The MPI environment in the virtual cluster is MPICH 2.1.0.8. All the virtual machine images are stored on a separate NFS storage server.

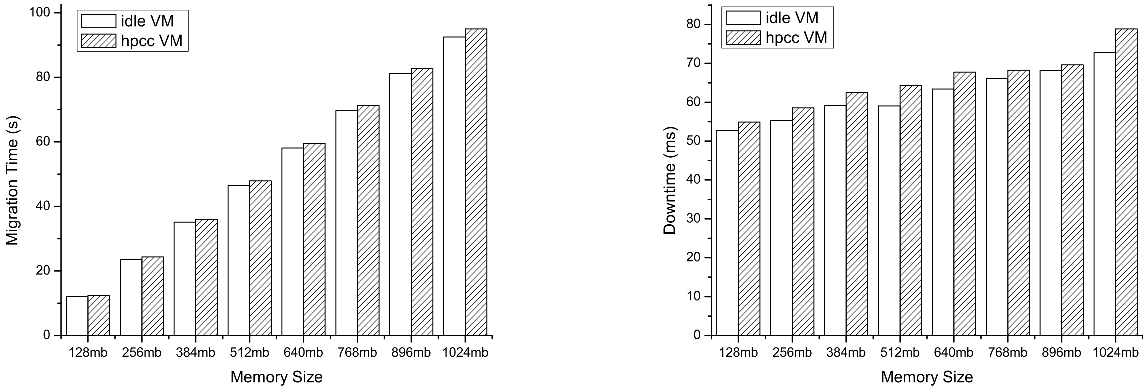
2) Benchmarks: We use the HPC Challenge Benchmark Suite (HPCC) [13] as the virtual cluster workloads which is a mainstream benchmark for the High Performance Computing and Parallel Computing. The HPCC benchmark suite includes 7 sub-benchmarks: *HPL, DGEMM, STREAM, PTRANS, RandomAccess, FFT, Communication bandwidth and latency* which can reflect the comprehensive performance of virtual cluster. In our experiments, three problem sizes are evaluated, they are 800, 1200, and 1600. The block sizes used are 128, 256, and the grid sizes used are 1*16, 2*8, 4*4 for the 16-node virtual cluster.

To measure the migration performance and overheads of virtual cluster, we extend the formal *Virt-LM* Benchmark [12]. We add the functionalities of concurrent control and concurrent testing which can record the migration time and downtime of each virtual machine and the whole virtual cluster.



(a) Cross-Domain VC: HPL (b) Cross-Domain VC: HPL Time (c) VC Scalability: HPL (d) VC Scalability: HPL Time

Figure 3. The Performance of Virtual Cluster Running HPCC Benchmark.



(a) The Migration Time of Virtual Cluster (b) The Downtime of Virtual Cluster

Figure 4. The Impact of Virtual Cluster Memory Configuration on VC Migration.

3) Experimental Precision: In order to ensure the data precision, each of the shown experimental results were obtained via running benchmarks three times with the same configuration and average the three values.

III. VIRTUAL CLUSTER PERFORMANCE

In this section, we first study the virtual cluster performance with no migration as the baseline performance. Two typical virtual cluster scenarios are measured.

A. Cross-Domain Virtual Cluster

Due to the large size of virtual cluster and the limited resources in physical machines, a virtual cluster may cross multiple domains (physical machines). We study the performance impact of cross-domain virtual cluster by creating a 16-node virtual cluster with different cross-domain situations when running HPCC benchmark. In order to save the space, we mainly analyze the HPL (High Performance Linpack) performance which is used to measure and rank the TOP500 Supercomputers [14].

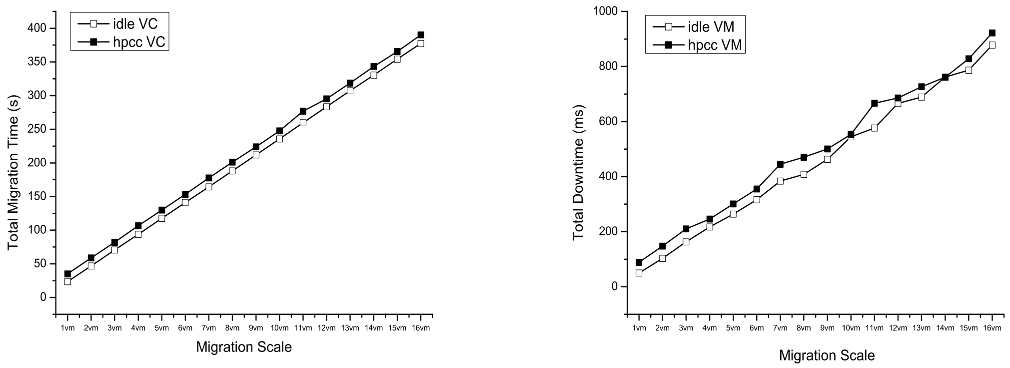
Figure 3(a)(b) show the HPL performance and its execution time under various cross-domain situations. From the figure, it is obvious that when all the 16 virtual machines are deployed on one physical machine (0-16 or 16-0), the HPL receives the best throughput with about 0.0034 TFlops; Conversely, when the 16 virtual machines are evenly deployed onto two physical machines (8-8), the HPL receives the worst throughput with only 21% of the best situation. It is because the communication overheads become a main bottleneck when deploying the virtual cluster across multiple physical machines.

B. Virtual Cluster Scalability

Figure 3(c)(d) show the result of virtual cluster scalability when running HPL. When the size of virtual cluster scales, the overall performance of virtual cluster (reflected by the TFlops) increases obviously. It means that the virtual cluster has a good scalability and is a suitable platform for the high performance computing and parallel computing.

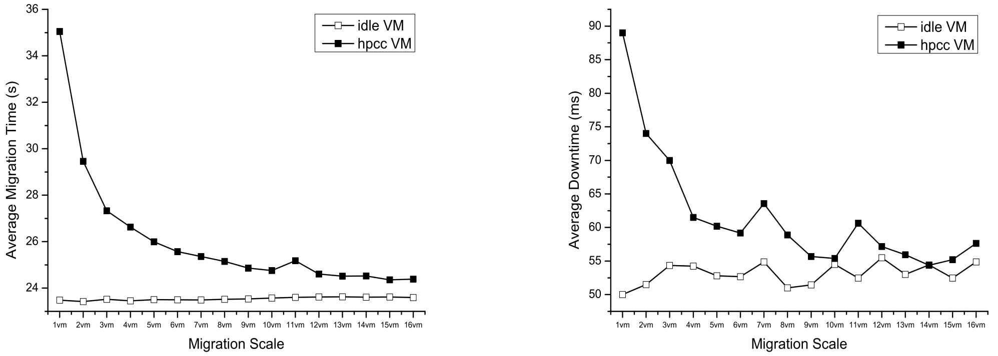
IV. DYNAMIC MIGRATION STRATEGIES OF VIRTUAL CLUSTER

In this section, we study the dynamic migration strategies of virtual cluster. We first investigate the impact of memory configuration of virtual cluster on the migration performance. Then, we investigate the concurrent migration granularity issue. After that, we study the mutual migration strategies.



(a) Total Migration Time

(b) Total Downtime



(c) Average Migration Time

(d) Average Downtime

Figure 5. The Migration Scalability of Virtual Cluster.

Table I
THE PINGPONGLATENCY UNDER THE MIGRATION SCALABILITY OF VIRTUAL CLUSTER.

Number of Migration Nodes	**1VM**	**2VM**	**3VM**	**4VM**	**5VM**	**6VM**	**7VM**	**8VM**
PingPongLatency (usec)	59.13	64.38	68.20	71.21	74.01	75.66	78.03	77.47
Number of Migration Nodes	**9VM**	**10VM**	**11VM**	**12VM**	**13VM**	**14VM**	**15VM**	**16VM**
PingPongLatency (usec)	78.90	77.81	76.81	74.13	72.22	69.21	65.44	60.79

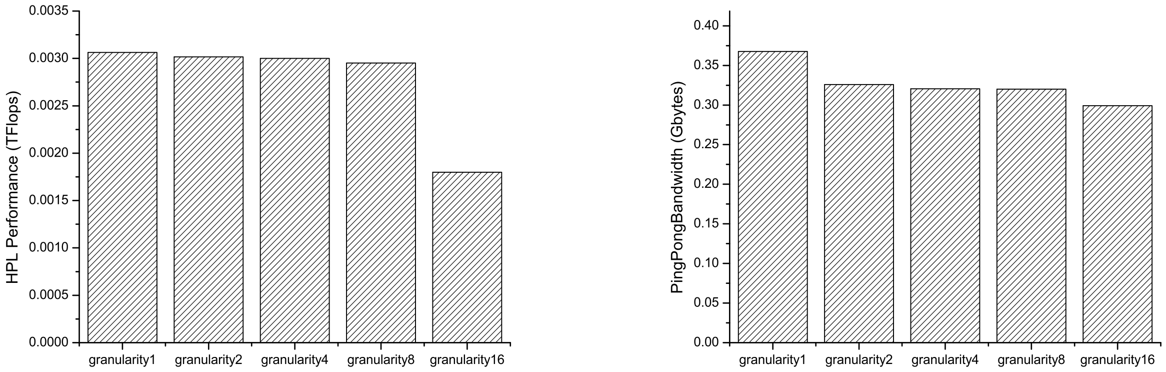
A. Effects of Memory Configuration

Figure 4 shows the changes of migration time and downtime of virtual cluster migration when the memory size of each virtual machine scales from 128MB to 1024MB. From the figure, we find that the average migration time is sensitive to the memory size. It is because more virtual cluster memory makes more data need to be transferred across the network which will increase the overall cluster migration time. While, the downtime of virtual cluster keeps between 50ms and 80ms for both the idle virtual cluster and HPCC virtual cluster. The reason is that the downtime is only affected by the dirty page rate and page transfer rate and has no direct relationship with the memory size.

B. Migration Scalability

In this section, we study the migration scalability of virtual cluster which means only part nodes of the virtual cluster migrate to the other machine while the remaining nodes still run on the source machine.

Figure 5 shows the migration time and downtime of virtual cluster when the number of migrated nodes increases. Virtual machine nodes in this experiment are migrated in sequence, so the total migration time is the sum of each migration node and increases approximate linearly as the number of migration nodes increases from 1 to 16 (see Figure 5(a)). And the downtime is also increases approximate linearly as the number of migration nodes increase (see Figure 5(b)). We find the downtime increases in relatively



(a) HPL Performance

(b) MaxPingPongBandwidth

Figure 6. The Workload Performance of Virtual Cluster under various Concurrent Granularities.

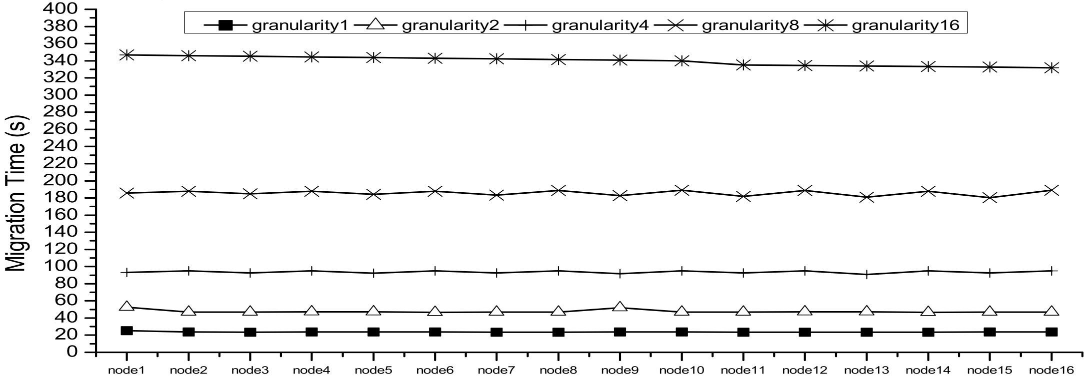


Figure 7. The Migration Time of Virtual Cluster under various Concurrent Granularities.

larger range from 50ms to 922ms since the total downtime is also the sum of downtime of each migration node.

Figure 5(c)(d) shows the average migration time and downtime of each virtual machine node. We find that both the average migration time and downtime decrease as the number of migration nodes increases, which means the virtual cluster has good migration scalability.

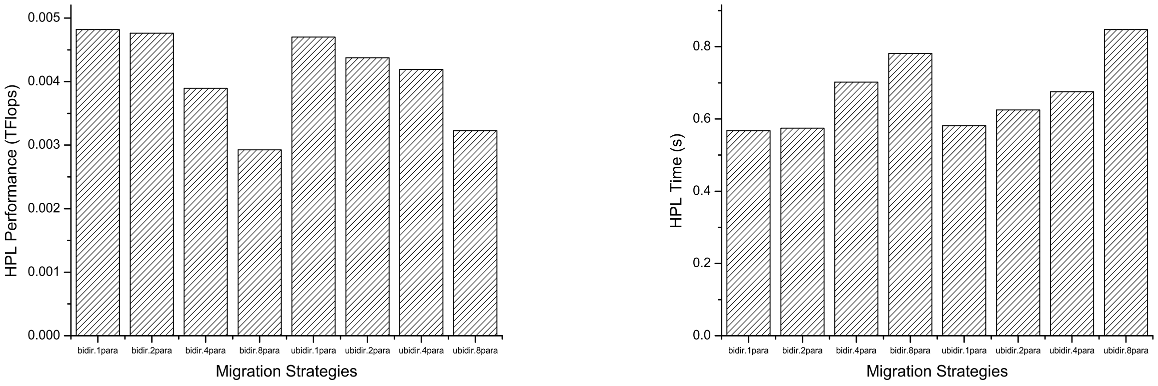
However, the migration efficiency cannot be identified only according to the migration time and downtime. We should also analyze the workload performance. Table I shows the network latency between virtual machine nodes. It is obvious that when the number of migration nodes increases to 8, the network latency is the largest due to the slow virtual cluster communication across two physical machines. So it needs to do the tradeoff between the migration overheads and the workload performance to determine the best migration number.

C. Concurrent Migration with Various Granularities

When the virtual cluster size scales, there are different choices to migrate the entire virtual cluster, such as concurrent migration. We present the concept of *Concurrent Migration Granularity* or *Concurrent Migration Degree* to represent multiple virtual machine nodes migrate simultaneously.

Figure 6 shows the HPL throughput and MaxPingPongBandwidth under different concurrent migration granularities. We find that the HPL throughput decreases dramatically when the concurrent migration granularity goes up to 16 which means all the 16 virtual machine nodes in the virtual cluster migrate simultaneously. Because the overloaded network bandwidth and the MPI synchronization delay between virtual machine nodes affect the HPL performance seriously. Figure 6(b) shows that the MaxPingPongBandwidth of the virtual cluster decreases from 0.37 Gbytes to 0.30 Gbytes when the concurrent granularity scales, which verifies the concurrent migration indeed causes some pressure on the network bandwidth.

Figure 7 shows the detailed migration time of each virtual machine node in the virtual cluster under different concurrent granularities. When the concurrent granularity increases, the migration time of each node increases dramatically since



(a) HPL Performance

(b) HPL Time

Figure 8. The Workload Performance of Virtual Cluster under Mutual Migration

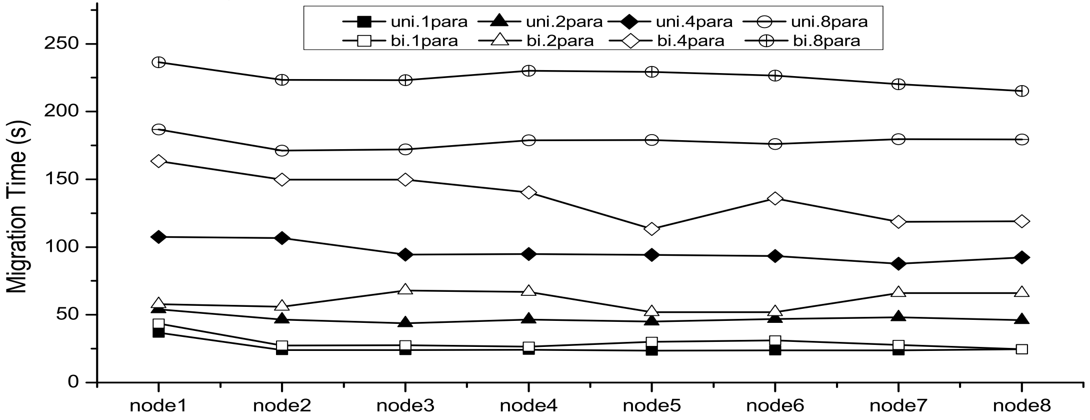


Figure 9. The Migration Time of Virtual Cluster under Mutual Migration.

the network bandwidth becomes the main bottleneck in the concurrent migration.

D. Mutual Migration

Mutual migration happens when two virtual clusters both need to migrate to the other host machine. Firgure 8 shows the HPL performance of virtual cluster under the mutual migration scenario. We compare the mutual migration efficiency (*bidirection migration* vs. *unidirection migration*) under different concurrent migration granularities. From the figure, we can find that the sequential migration (concurrent migration granularity = 1) is better than the concurrent migration, which is consistent with the previous mentioned conclusion. Further, the unidirection migration is slightly better than the bidirection migration due to limited network bandwidth.

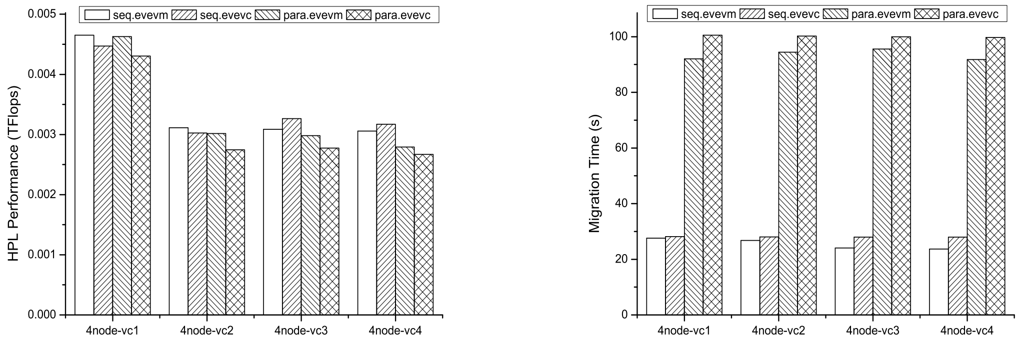
Figure 9 shows the migration time under mutual migration. We can draw the conclusion that when the concurrent granularity is small, the migration performance of bidirection migration is very close to the unidirection migration. However, when the concurrent granularity scales, the performance of bidirection migration decrease quickly due to the intensive competition of network resources.

V. MULTI-VC MIGRATION STRATEGIES

In this section, we study the scenario of multiple virtual clusters migration and investigate the multi-VC migration strategies such as migration order. Figure 10 and 11 show the migration performance of multiple virtual clusters under different migration strategies, including homogeneous VC migration and heterogeneous VC migration.

A. Live Migration of Homogeneous Virtual Clusters

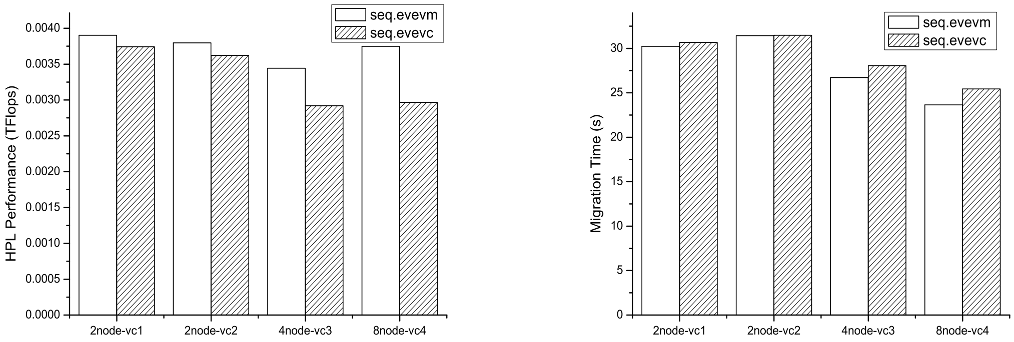
We create four 4-node virtual clusters with the same configuration, and assume the node numbers in each virtual cluster are: VC1(1,2,3,4), VC2(5,6,7,8), VC3(9,10,11,12), VC4(13,14,15,16). Similar to the example in Figure 2(c), in this section, we study four multi-VC migration strategies: (i) *seq.evevm* means sequently migrating 16 virtual machines *node by node* with the migration order: 1, 2, 3, 4, 5, 6, 7, 8,



(a) Homogeneous VC: HPL

(b) Homogeneous VC: Mig. Time

Figure 10. Migration Strategies of Multiple Homogeneous Virtual Clusters.



(a) Heterogeneous VC: HPL

(b) Heterogeneous VC: Mig. Time

Figure 11. Migration Strategies of Multiple Heterogeneous Virtual Clusters.

9, 10, 11, 12, 13, 14, 15, 16; (ii) *seq.evevc* means sequently migrating the 16 virtual machines *cluster by cluster* with the migration order: 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 4, 8, 12, 16; (iii) *para.evevm* means concurrently migrating within the virtual cluster itself with the migration order: 1 2 3 4, 5 6 7 8, 9 10 11 12, 13 14 15 16; (iv) *para.evevc* means concurrently migrating between the virtual clusters with the migration order: 1 5 9 13, 2 6 10 14, 3 7 11 15, 4 8 12 16.

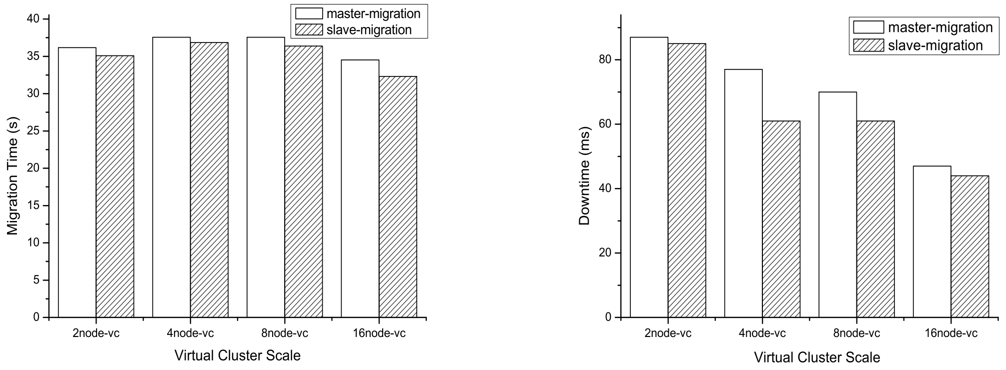
Figure 10 show the HPL performance and the migration time. From the figure, we find that sequential migration performs better than concurrent migration, especially on the migration time. *Node by node* migration is slightly better than *cluster by cluster* migration due to the less status synchronization overheads between each virtual machine in each virtual cluster. One interesting finding in Figure 10(a) is that VC1 achieves the best overall performance among all the four virtual clusters. It is because the VC1 is migrated first and benefits from the sufficient resources in the destination machine.

B. Live Migration of Heterogeneous Virtual Clusters

In this section, we study the migration strategy of multiple virtual clusters with different sizes. We create four virtual clusters: VC1(1,2), VC2(3,4), VC3(5,6,7,8), VC4(9,10,11,12,13,14,15,16). Due to the space limitation, we only study the sequential migration strategies: (i) *seq.evevm*: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16; (ii) *seq.evevc*: 1, 3, 5, 9, 2, 4, 6, 10, 7, 11, 8, 12, 13, 14, 15, 16. It is obvious in Figure 11 that sequential migration with *node by node* is better than *cluster by cluster*. What's more, the HPL performance with *seq.evevc* migration strategy is worse than that with *seq.evevm* strategy. In the *seq.evevc* strategy, the virtual clusters cross on the two physical machines, therefore increase the communication burden and affect the workload performance.

C. Master Node Migration vs. Slave Node Migration

Usually there is a master node in the cluster to control the whole running process, so it is interesting to compare the two migration strategies of migrating the master node and



(a) Migration Time

(b) Downtime

Figure 12. Performance Comparison between Migrating Master Node and Slave Node.

migrating the slave node. Figure 12 shows the migration time and downtime when migrating a master node and slave node respectively in the scenario of multiple heterogeneous virtual clusters. We can find that migrating the slave node is better than the master node. What's more, it will benefit more when the cluster size is larger.

VI. RELATED WORK

Virtual cluster is an important application scenario of virtualization technology. It can be used to do the high performance computing task [8, 9] or parallel computing task such as MapReduce task [15]. Due to the overheads brought by virtualization, Ye et al. evaluated the performance of virtual cluster and presented a performance model [7].

Live migration of single virtual machine has been widely studied. Pre-copy technique [4, 5] is the classic mechanism to implement the live migration in different hypervisors. Differ from the pre-copy technique, Hines et al. implemented a post-copy technique to avoid the duplication of data transmission [16] in which each page needs to be transferred only once to the destination host. In order to reduce the amount of data transmission, Jin et al. presented a method based on the memory compression technique [17]. However, all the above migration methods only transferred the memory and CPU status and didn't refer to the disk transmission. Luo et al. presented a new algorithm to transfer the whole virtual machine including the disk data [18].

Most recently, Ye et al. proposed a new method to migrate multiple virtual machines with a resource reservation mechanism in the cloud computing environments [19]. Huang et al. presented a benchmark for the live migration of virtual machine [12]. Deshpande et al. implemented a de-duplication based approach to perform concurrent live migration of virtual machines [20]. Al-Kiswany et al. used the similar method to co-migrate the virtual machines to avoid the data de-duplication within their VMFlock migration service [21]. However, all the above work didn't solve the problem of live migration of *virtual cluster* in which the frequent synchronization and communication operations among the virtual machines can affect the migration performance.

VII. CONCLUSION

In this paper, we have studied the live migration performance and overheads of *virtual clusters* from the experimental perspective and investigated different VC migration strategies. We describe a framework *VC-Migration* to control the migration of virtual clusters. Based on this framework, we perform a series of experiments of virtual clusters to understand the performance bottleneck and overheads. We first study the performance characterization of virtual cluster, including the performance characterization of cross-domain virtual cluster and the scalability of virtual cluster. Then we study the dynamic migration strategies for virtual clusters, including migration scalability, concurrent migration with various granularity, and mutual migration. After that, we investigate the migration scenario of multiple virtual clusters, including homogeneous VC migration and heterogeneous VC migration. We also compare the migration performance of master node and slave node.

Experimental results reveal some new discoveries, based on which we can propose several optimization principles to improve the migration performance of virtual clusters: (i) The main contradiction of VC migration is the large amount of image data and the limited network bandwidth. (ii) The virtual machines belonging to the same virtual cluster should be deployed together as far as possible to reduce the communication and synchronization latency across different physical machines. (iii) Virtual cluster has good scalability and is suitable for the high performance computing and parallel computing tasks. (iv) When a virtual cluster needs to be migrated, it is important to select a suitable concurrent migration granularity. Large concurrent granularity will decrease the VC performance dramatically. (v) Mutual migration should be avoided due to the long overall migration time. (vi) The migration of slave node incurs relatively less overhead compared to the master node. So it should give a priority to the slave migration. (vii) Migration order is important when multiple virtual clusters need to be migrated. The long-time cross-domain virtual cluster will decrease the overall performance of applications. (viii) There is a big optimization space in the live migration of virtual clusters.

Future work will include optimization of the migration mechanism in the hypervisor to improve the migration efficiency of virtual clusters and design efficient migration algorithms for the virtual clusters.

ACKNOWLEDGMENT

This work is supported by National High Technology Research 863 Major Program of China (No.2011AA01A207), National Natural Science Foundation of China (No.11071215), MOE-Intel Information Technology Foundation (No.MOE-INTEL-11-06).

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] 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 nineteenth ACM Symposium on Operating Systems Principles*, 2003, p. 177.

[3] C. Waldspurger, "Memory resource management in VMware ESX server," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, p. 194, 2002.

[4] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, 2005, p. 286.

[5] M. Nelson, B. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2005, p. 25.

[6] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, "Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective," in *Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications (GreenCom)*, 2010, pp. 171–178.

[7] K. Ye, X. Jiang, S. Chen, D. Huang, and B. Wang, "Analyzing and modeling the performance in Xen-based virtual cluster environment," in *2010 12th IEEE International Conference on High Performance Computing and Communications (HPCC)*, 2010, pp. 273–280.

[8] M. F. Mergen, V. Uhlig, O. Krieger, and J. Xenidis, "Virtualization for high-performance computing," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 2, pp. 8–11, 2006.

[9] W. Huang, J. Liu, B. Abali, and D. K. Panda, "A case for high performance computing with virtual machines," in *Proceedings of the 20th Annual International Conference on Supercomputing (ICS)*, 2006, pp. 125–134.

[10] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[11] K. Ye, J. Che, X. Jiang, J. Chen, and X. Li, "vTestkit: A performance benchmarking framework for virtualization environments," in *2010 Fifth Annual ChinaGrid Conference (ChinaGrid)*, 2010, pp. 130–136.

[12] D. Huang, D. Ye, Q. He, J. Chen, and K. Ye, "Virt-LM: a benchmark for live migration of virtual machine," in *Proceeding of the second ACM/SPEC International Conference on Performance Engineering (ICPE)*, 2011, pp. 307–316.

[13] "HPC Challenge Benchmark," *http://icl.cs.utk.edu/hpcc/*, 2012.

[14] "TOP 500 Supercomupter Sites," *http://www.top500.org/*, 2012.

[15] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi, "Evaluating mapreduce on virtual machines: The hadoop case," in *1st International Conference on Cloud Computing (CloudCom)*, 2009, pp. 519–528.

[16] M. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, 2009, pp. 51–60.

[17] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive memory compression," in *IEEE International Conference on Cluster Computing*, 2009, pp. 1–10.

[18] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in *Proceedings of the IEEE International Conference on Cluster Computing*, 2008, pp. 99–106.

[19] 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 *2011 IEEE International Conference on Cloud Computing (CLOUD)*, 2011, pp. 267–274.

[20] U. Deshpande, X. Wang, and K. Gopalan, "Live gang migration of virtual machines," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing (HPDC)*, 2011, pp. 135–146.

[21] S. Al-Kiswany, D. Subhraveti, P. Sarkar, and M. Ripeanu, "VMFlock: virtual machine co-migration for the cloud," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing (HPDC)*, 2011, pp. 159–170.