SLM: Synchronized Live Migration of Virtual Clusters across Data Centers

Tao Lu (Student), Morgan Stuart (Student), Xubin He
Virginia Commonwealth University
{lut2,stuartms,xhe2}@vcu.edu

1 Introduction

Cluster platforms, deployed in data centers worldwide, are the backbone of the popular cloud computing services. For scalability, manageability and resource utilization, one physical machine in the cloud platform can be virtualized into a bundle of virtual machines (VMs). Each VM works as an independent node. To solve a time consuming task, several VMs are grouped as a virtual cluster and collaborate on the task.

It's essential for a cloud computing platform to support live migration of virtual clusters. First, virtual cluster migration enables load-management in data centers [1, 5, 3]. Second, live migration of virtual cluster provides transparent infrastructure maintenance [3]. Third, virtual cluster migration can be used to support enterprise IT consolidation [5]. Finally, flexible deployment of virtual clusters across data centers is also a key enabler of federated clouds [1].

Co-migration of a group of VMs and migration of virtual clusters have attracted considerable interest for data center management [1, 5, 3] and HPC cluster computing [2]. VCT [2] focuses on devising mechanisms to manage a tightly coupled HPC virtual cluster as a single entity, making cluster level operation such as suspending, migrating or resuming a synchronous process across all nodes. However, VCT requires the cluster to be offline for as long as tens of minutes. Many time sensitive applications and services cannot afford this extended downtime. VMFlock, CloudNet and Shrinker [1, 5, 3] employ the same technique, data deduplication, to reduce the network traffic during migration. Besides eliminating redundant data, VMFlock also accelerates instantiation of the applications at the target data center through transferring the essential set of data blocks first. CloudNet employs dynamic VPN connectivity to migrate networks and "smart stop and copy"to intelligently pick when to halt the iterative transfer of dirty pages to decrease downtime and latency.

VMFlock, CloudNet and Shrinker have succeeded in eliminating the migration of redundant data blocks so as to reduce network traffic and total migration time. However, VMFlock adapts an offline migration mode. CloudNet and Shrinker, though they support live migration, both fail to consider that VMs in a cluster still need to collaborate on tasks during live migration.

2 Problem Statement

We claim that, during live migration of a virtual cluster, synchronizing the migration progress of every VM is critical for system performance. Suppose that the migrating virtual cluster is supporting a Hadoop platform running a data intensive task such as a MapReduce job. The VM which serves as the master node takes on the role of JobTracker and NameNode, while the other VMs serve as slave nodes performing as TaskTrackers and DataNodes. The JobTracker needs to communicate with each TaskTracker to schedule tasks, and the DataNodes need to push or receive input splits and intermediate results to or from each other. Normally, the communication and data flow occur via a high speed LAN, however this is not always the case during live migration.

If the VMs are migrated out of step, then some VMs may have completed transferring their disk and memory states, while others have not. At this point, the nodes, which have finished the transfer of their states, and are ready for suspending and resuming, have two choices. First, suspending briefly to transfer the final memory and processor states to the target host, then instantiating and running at the destination. Second, keep running at the source site and wait for the other VMs to finish transferring their memory and disk states. Then, suspend all VMs at the source, transmit the final memory and processor states, and follow by an instantiation of all VMs at the destination. If taking the first choice, some nodes in the cluster are running at the source site while the others are running at the destination site. The communication and data flow between these remote nodes are supported by a cross-datacenter WAN with a significantly lower bandwidth. As a cluster interconnection network, LAN has long been the performance bottleneck of cluster systems [4], connection through WAN will certainly make this bottleneck even worse. If taking the second choice, the nodes which have already finished transferring their disk and memory state may need to retransmit large sums of data, because many memory pages and disk blocks may have become dirty during the waiting period, especially in data intensive environment. This retransmission is a waste of expensive WAN bandwidth and will certainly increase the migration latency. Therefore, neither of the two choices is advisable.

We propose synchronizing the migration process of VMs in a cluster to avoid the above-mentioned dilemma. An ideal scenario is every VM completes migration and instantiates at the target site at the same time , this can prevent these nodes from communicating and pushing data through WAN as well as retransmitting the pages which become dirty during the waiting period.

3 Proposed Mechanism

To achieve synchronized live migration of a virtual cluster(SLM), we need to (1) monitor the VMs' status and available migration bandwidth, (2) simulate workloads to predict migration costs and latency and (3) make migration strategy and manage migration. The overview of the proposed SLM system is illustrated in Figure 1.

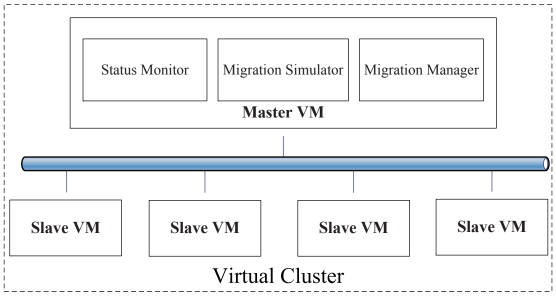

Figure 1: SLM Virtual Cluster System Overview

Three components, a status monitor, a migration simulator, and a migration manager are built into the master VM. The Status Monitor detects the available migration link bandwidth, collects VMs' states information such as memory size, disk size, and page dirty rate of each VM. The Migration Simulator exploits a model to predict the migration latency of each VM. The model uses the information collected by the Status Monitor as inputs and predicts the relative migration time of each VM. The Migration Manager uses the predicted relative migration time of each VM as a weight factor and allocates available migration bandwidth for each VM proportionably. This way, the migration of VMs can be kept in step.

Given the complexity of parallel computing and network environments, it's infeasible to create a single migration strategy for all scenarios. Therefore, the migration manager instructs the migration simulator to re-evaluate its prediction periodically so as to adjust the migration strategy accordingly.

4 Status

This report primarily focuses on why synchronizing the live migration of virtual cluster is important to system performance. We have had a preliminary design of the system architecture of SLM. Our final goal is to implement an efficient mechanism which supports synchronized live migration of virtual clusters. We will compare our mechanism with the free-running live migration, from the view point of system performance, for data intensive and HPC applications. We will also measure the costs of adding such a synchronization mechanism into virtual clusters. Moving forward, we'll detail and optimize the system design, implement the system and finally test and evaluate it.

Once we have completed the implementation, tests, and evaluation of SLM system, we expect to be able to present our work comprehensively.

References

[1] Al-Kiswany, S., Subhraveti, D., Sarkar, P., and Ripeanu, M. VMFlock: Virtual machine co-migration for the cloud. In *HPDC'11* (San Jose, USA, June 2011).

[2] Anedda, P., Leo, S., Manca, S., Gaggero, M., and Zanetti, G. Suspending, migrating and resuming hpc virtual clusters. *Future Generation Computer Systems 26*, 8 (2010), 1063–1072.

[3] Riteau, P., Morin, C., and Priol, T. Shrinker: Improving live migration of virtual clusters over wans with distributed data deduplication and content-based addressing. In *Euro-Par'11* (Bordeaux, France, August 2011).

[4] Sterling, T., Savarese, D., Becker, D. J., Dorband, J. E., Ranawake, U. A., and Packer, C. V. BEOWULF: A parallel workstation for scientific computation. In *ICPP'1995* (Urbana-Champain, USA, August 1995).

[5] Wood, T., Shenoy, P., Ramakrishnan, K., and der Merwe, J. V. CloudNet: Dynamic pooling of cloud resources by live wan migration of virtual machines. In *VEE'11* (Newport Beach, USA, March 2011).