An Analysis of Live Migration in OpenStack using High Speed Optical Network

Md Israfil Biswas, Gerard Parr, Sally McClean, Philip Morrow and Bryan Scotney

School of Computing and Information Engineering

University of Ulster

Coleraine, Northern Ireland

Email: {mi.biswas, gp.parr, si.mcclean, pj.morrow, bw.scotney}@ulster.ac.uk

*Abstract*— Virtualization technology has become very common trend in modern data centres as Virtual Machine (VM) migration brings several benefits like improved performance, high manageability, source consolidation and fault tolerance. Live Migration (LM) of VM is used for transferring a working VM from on host to another host of a different physical machine without interfering the VM. However, there are very few research has been done in real time considering the resource consumption and latency of live VM migration that reduce these benefits to much less than their potential. In this paper we have presented an analysis of LM in our high speed optical fibre network connecting Northern Ireland, Dublin and Halifax (Canada). We show that the total migration time as well as network data transfer for post-copy LM are dominated by specific VM memory patterns using loaded or unloaded VMs. We also found that the downtime is not extremely varied and no severe effect over our long distance network.

Keywords—Virtual Machine, Live Migration, Openstack.

# Introduction

LM is one of the important enabling technologies that perform transformation of VMs between distinct physical nodes without stopping of running tasks. Hence, LM is adopted by datacentre administrators for fault tolerance and load balancing [12]. LM is vital for Virtualization where resource scheduling and management frameworks are maintained continuously.

Although VM live migration technology has attracted considerable interest for datacentre management and cluster computing, there are still optimization and development required in this area for a variety of application cases. Few studies are available considering real time resource consumption and load balancing over long distance networks. Low-latency migration could reduce resource requirements up to 50% and service-level agreement violations by up to 20%, and they demonstrated the correlation between resource efficiency and migration latency [2] . In our previous work [3] we also showed that prioritizing tasks for the nearest servers or with low latency not only improve the quality of service (QoS) but also demonstrates better utilization of the resources. Post-copy live migration technique has several benefits compared to pre-copy live migration like lower downtime and effective utilization of the network bandwidth [9].

Hence, in this paper we have analysed post-copy live migration for specific VM memory patterns using loaded and unloaded VMs. The purpose of this research is to perform live migration on high speed 10G interface fibre optic networks considering real time long distance network.

Different latency may lead to significant differences in LM performance. Considering total migration time, downtime etc previous studies demonstrated that it could vary significantly between different workloads, ranging from milliseconds to tens of seconds in the case of 1G interface network. This is mostly due to the diversity of VM configurations and workload characteristics. For instance, the initial memory size of a VM and applications’ memory access pattern are critical factors that have a decisive effect on the migration latency, i.e. the total time a VM is undergoing performance penalty and high power state.

In this paper we have analysed the performance of post-copy live migration using 10G interfaces where our network infrastructure is based on Openstack nova development [4]. We also have used shared storage system ‘Network File System (NFS)’ for this work of virtualization.

The rest of the paper is organized as follows: Section II describes the Background in this area of research with discussion to related works in this area, Section III describes our cloud testbed under high speed fibre optic 10G network infrastructure, Section IV describes the experiment setup and configuration for this work, Section V presents the performance through experimentation of our Cloud testbed using 10G interfaces in different VM memory patterns using loaded and unloaded VMs. Finally, the paper concludes with the Conclusions with a view for future work.

# Background

Live migration instances contain its states, memory and emulated devices is transferred from one hypervisor to another with no downtime (if possible). In this paper we have analysed post-copy LM in our Wide Area Network (WAN) with 10G interfaces.

In this work we have implemented post-copy live migration technique in our Ubuntu KVM as it has lower downtime and effective utilization of network bandwidth.

## Pre-Copy Live Migration: In Pre-Copy LM, memory is transferred before VM allocation. However, the issue is how to copy memory while it is re-dirtied over and over again by the guest VM? This is solved by first copying all the memory followed by intervals of copying newly dirtied pages until the remaining state is small enough. Hence, the total migration time in this process is the Reservation time, Iterative Pre-Copy time (could be several rounds depending on the dirtied pages), the time required to ‘Stop and Copy’ and the Commit time (time that is running in the destination host). Pre-Copy LM is implemented by all most all hypervisors (e.g., Xen, Qemu, VMWare). Pre-Copy LM is often challenged fast memory dirtying applications.

## Post-Copy Live Migration: Post-Copy LM is the process where the Transfer-Memory is transferred only after VM relocation. It is important to ensure that VM performance is not degraded after the relocation for the network bound page faults. Hence, fast interconnects and improved page fault mechanisms are required to solve this issue which is challenged by fast memory reading applications. The main advantage of Post-Copy is lower downtime (because CPU and short VM stats will be migrated while the VM is stopped.

## Related Works

Hines et al [7] analysed post-copy LM to reduce the migration time to show promising results for commercial workloads. Post-copy techniques have not been thoroughly evaluated for scientific workloads particularly for remote machines that started and migrate without copying the memory pages but copied on demand. Moghaddam et al [8] analysed post-copy LM to reduce the downtime for copying changed memory pages that may significantly slowdown the migrated instance.

Aidan Shribman et al analysed pre-copy and post-copy migration in [9] where they have proposed a page reordering policy ‘Least Recently Used (LRU)’ that has lower chance of re-dirtied and migrated earlier for pre-copy LM. They also propose delta encoder, Xor Binary Zero Run Length Encoding (XBZRLE) to reduce the cost of the page re-send. In post-copy LM they have proposed Remote Direct Memory Access (RDMA) for low-latency resolution of network-bound page faults and pre-paging/pre-fetching to reduce the overall page faults integrating page faulty mechanism and hybrid live migration.

In contrast, we provide an analysis of post-copy live migrations in a high speed optical fibre network where hosts are remotely connected with 10G interfaces with different VM memory sizes of loaded and unloaded VMs. In this paper we investigate the VM migration techniques that deal with 10G interference and represents comparative analysis of various strategies dealing with the effect of QoS parameters. This work can be helpful to the service provider, cloud service developers and cloud service consumers to identify the interference affecting the application performance.

#  UlsterCloud

Ulster University Cloud testbed (UlsterCloud) is designed considering three intelligent controllers that can automate the management of resources in the provider network and in the cloud computing data centres respectively. UlsterCloud aims to provide IaaS, PaaS, SaaS to local and remote users while securely linked with enterprise sites. The test-bed is designed primarily to provide a platform for development of next generation Monitoring, intelligence and orchestration tools interface with next generation standard tools to provide seamless resource monitoring and orchestration with flexible network management.

The testbed incorporates a range of industry-standard servers, physical networking fabric and storage nodes to outperform existing virtualization technologies at the server, router, and network levels to create dynamic resource pools that can be transparently connected to enterprises.

Our contract with Hibernia Atlantic [10] has provided fibre optic connection from Hibernia Cable Landing Stations (CLSs) and provided multipoint circuits between Coleraine (Northern Ireland), Dublin and Halifax (Canada) as shown in Figure 3. The circuits from Hibernia Networks provide a direct 10G interface from Coleraine CLS to Ulster University campus. The other two hosts are connected to Hibernia Dublin CLS and Hibernia Halifax CLS. However, each can be incremented to the far-end CLS sites and able to cope with 10Gb/s and beyond 10Gb/s toward 100Gb for burst traffic at short intervals over our allocated wavelengths.

## The Testbed Buildout

Our contract with Hibernia Network is not providing any gateway to outside (i.e. no Internet access is provided). Moreover, our University will not allow us to connect with the JANAET with Hibernia network (i.e. not allowed to bridge the two networks for security reason).



**Figure3.** Planned Ulster Interconnected testbed



**Figure4.** Internet Access from ADSL connection (BT broadband) with HP switch 5800

As we need remote access to configure, monitor, conduct software updates and run experiments between the systems over the Hibernia links, we need secured internet access or outside connectivity. Hence, an ADSL connection for the Internet access is added with a network Switch at the Coleraine University (shown in the Figure 4). Therefore, the switch connects the Hibernia 10Gb/s interface with 100Mb/s ADSL link and any existing testbed at Ulster University for future experiments.

## File Storage and Sharing through Virtualization

Our Network File System (NFS) Server for UlsterCloud is a Virtual Private Cloud (VPC) that provides seamless and secure Virtualization with NFS File Storage Solution (Figure 5). The vision is efficient pooling of geographically secluded data center resources with optimized support for live migration of VMs.

By default, migration only transfers in-memory state of a running domain for example memory, CPU state etc. Disk images are not transferred during migration but they need to be accessible at the same path from both hosts. Therefore, some kind of shared storage needs to be setup and mounted at the same place on both hosts. The simplest solution is to use NFS.

We have configured an NFS server on a host serving as shared storage:

*# mkdir -p /exports/images*

 *Edit etc/exports*

*/exports/images \*(rw,no\_root\_squash)*

The exported directory needs to be mounted at a common place on all hosts running libvirt (the IP address of our NFS server is 192.168.1.3):

Edit *etc/fstab*

*192.168.1.3:/exports/images /vm\_images nfs auto 0 0*

*# mount /vm\_/images*



**Figure5.** UlsterCloud NFS Shared storage for VMs Live Migration

 We observed that naive solution of exporting a local directory from one host using NFS and mounting it at the same path on the other host would not work. The directory used for storing disk images has to be mounted from shared storage on both hosts. Otherwise, the domain may lose access to its disk images during migration because source libvirtd may change the owner, permissions, and SELinux labels on the disk images once it successfully migrates the domain to its destination. Libvirt avoids doing such things if it detects that the disk images are mounted from a shared storage.

# System Configuration and Setup

In this work, we have used three physical machines Dell PowerEdge R815 (AMD Opteron 6366HE@ 3,6GHz, 128GB RAM) as a modules of a Blade server providing 10Gb/s network interfaces through its backplane. We installed Openstack with QEMU / libvirt with post-copy support. Two servers were configured purely as a compute node running nova and nova-network services. The third one was configured as both a compute node and the controller node providing also all the other management services.

Following system configuration is used for these tests:

* 3 nodes: 1 control node (Coleraine), 2 compute nodes (Dublin, Halifax)
* Openstack Icehouse release
* Nova 2.18.1
* QEMU 1.2.1
* Libvirt 0.10.2

## System setup

The following configuration has been done for tests, some verification and check lists are omitted for the simplicity.

*1. Network configuration*

All hosts (hypervisors) are running in the same network/subnet.

*1.1. DNS configuration*

DNS configuration and consistency of */etc/hosts* file across all hosts are done.

*1.2. Firewall configuration*

The */etc/sysconfig/iptables* file is configured to allow libvirt listen on TCP port 16509 and need to add a record accepting KVM communication on TCP port within the range from 49152 to 49261.

-A INPUT -p tcp -m multiport --ports 16509 -m comment --comment "libvirt" -j ACCEPT

-A INPUT -p tcp -m multiport --ports 49152:49216 -m comment --comment "migration" -j ACCEPT

*2. Libvirt configuration*

We have enabled libvirt listen flag at */etc/sysconfig/libvirtd* file.

LIBVIRTD\_ARGS=”–listen”

The */etc/libvirt/libvirtd.conf* file is configured to make the hypervisor listen TCP communication with none authentication. SSH keys for authentication are strongly recommended as authentication is set to NONE.

listen\_tls = 0

listen\_tcp = 1

auth\_tcp = “none”

*3. Nova configuration*

To enable real live migration, we have set up *live\_migration* flag in */etc/nova/nova.conf* file as Openstack does not use real live migration mechanism as a default setting. This is because there is no guarantee that the migration is successful (e.g., faster dirtied pages than transferred to destination host).

live\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_LIVE

## Live Migration Execution

We first checked the list available for VMs, then checked VM details to determine which host an instance running on. After that we used commands to list the available compute hosts and to choose the host we want to migrate the instance to as it’s very much secured and efficient in nova.t

Then migration of the instance is done to a new host. For live migration using shared file system, we have used the following code:

$ nova live-migration <VM-ID> <DEST-HOST-NAME>

Note that, we have tested for block live migration and used the following command with block\_migrate flag enabled:

$ nova live-migration --block\_migrate <VM-ID> <DEST-HOST-NAME>

Finally, we checked VM details and also checked if it has been migrated successfully.

All tests were performed across 5 different VM flavors to examine the impact of the flavor. Another point we were curious about is how higher memory load of VMs can impact migration performance. Here, we present the results of our experiments which show how live migration works in these different scenarios.

**VM Flavors**

The Openstack installation supports the following standard flavors:

1. Description of used flavors

|  |  |  |  |
| --- | --- | --- | --- |
| **Flavor** | **VCPUs** | **RAM** | **Disk** |
| t | 1 | 512 | 1 |
| s | 1 | 2048 | 20 |
| m | 2 | 4096 | 40 |
| l | 4 | 8192 | 80 |
| xl | 8 | 16384 | 120 |

For the evaluation of live migration performance we focused on measuring the followings:

* Migration duration or, the total migration time
* Duration of VM unavailability or, the VM downtime
* Amount of data transferred through the migration interface using stress tool [11]

We measured the real throughput of 10-Gb/s interface in configuration using full 10 Gb/s fibre optical link from Hibernia using the iperf tool. We achieved throughput up to 7 Gb/s between Coleraine-Dublin with an average delay of 5.5ms and up to 3Gb/s between Coleraine-Halifax path with an average latency of 52.5ms.

The amount of time required to perform a live migration is dependent on the activity of the VM and if the VM is very active, it takes longer for the migration process to transfer the VM’s memory. Hence, this work analyses live migration with both unloaded and loaded VMs to determine the impact over the network.

**Unloaded VM:** usually VM was running Ubuntu 15.04 LTS but it was unloaded during the migration process.

**Loaded VM:** when VM with memory-intensive load or, VM running in Ubuntu 15.04 LTS. In this experiment stress tool is used to generate a stable memory load during migration and the size of stressed memory based is tuned with the VM flavour so that stress tool can consumed approximately 75% of the VM’s memory.

# Experiments

In this section, the paper analyses experiment results to measure the key metrics mentioned above.

****

1. Migration Durations for different flavours – loaded and unloaded

**Migration Duration**

We found noticeable differences for NFS live migration time as shown in Table II and Fig.1 of loaded and unloaded VMs. We found that the differences increases with size of the VMs (i.e., with different VM flavours). Only 0.1s and 0.2s of time difference found for Coleraine-Dublin and Coleraine-Halifax link respectively for loaded and unloaded tiny VM flavour (t). However, the difference is much higher for extra-large VM flavour (xl) 92.9s for Coleraine-Dublin and 130.6s for Coleraine-Halifax.

1. Migration Times for different flavours – loaded and unloaded

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link** | **VMs** | **t** | **s** | **m** | **l** | **xl** |
| Coleraine-Dublin | Unloaded | 11.2 | 28.8 | 30.1 | 42.2 | 55.3 |
| Loaded | 11.3 | 29.1 | 39.8 | 90.4 | 148.2 |
| Coleraine-Halifax | Unloaded | 11.8 | 30.6 | 47.5 | 54.6 | 66.8 |
| Loaded | 12.0 | 48.3 | 100.3 | 167.1 | 197.4 |

Our analysis show that there is not a significant increase of migration time for the unloaded VMs as the flavour size increases for both Coleraine-Dublin and Coleraine-Halifax link. Here the graph for the migration time of the unloaded VMs increases slowly even though there is a big difference in memory size of the different flavours. However, much higher increase for the loaded VMs (especially for Coleraine-Halifax link) that show the migration time increases significantly as memory size increases. Hence, our observations indicate that reducing VM memory load can lead to faster live migrations.

1. VM Downtimes for different flavours – loaded and unloaded

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link** | **VMs** | **t** | **s** | **m** | **l** | **xl** |
| Coleraine-Dublin | Unloaded | 0.3 | 0.4 | 0.4 | 0.5 | 0.5 |
| Loaded | 0.6 | 0.7 | 0.7 | 0.8 | 0.8 |
| Coleraine-Halifax | Unloaded | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 |
| Loaded | 0.7 | 0.7 | 0.8 | 0.8 | 0.9 |



1. Downtimes for different flavours – loaded and unloaded

Although we found significant difference in NFS live migration time for loaded extra large VMs, the downtime is not significantly changed in our experiments as shown in Table III and Fig.2.

We found 0.3s difference in downtime for almost all VM flavours of loaded and unloaded VMs using Coleraine-Dublin fibre optic link. Interestingly, we also found similar 0.2s-0.3s difference in downtime using Coleraine-Halifax link of loaded and unloaded VMs. Hence, our observation is changing the memory size of VMs does not severely affect the downtime of a high-speed link.

In the case of Block live migrations (without sharing using NFS), downtime for larger loaded VMs can be up to 75% of whole migration time. In that cases with memory load, downtime exceeds half of the migration time, which is not negligible, especially for time critical services. Hence, if live migration is to be used for such services, it needs to be used with care.

We found almost the total network data transfer is doubled of the VM memory size for different unloaded VM flovours using Coleraine-Dublin link as shown in Table IV and Fig. 3. Although less than 50% less data transferred comaped to the unloaded VMs, signiicant increase in data transfer was also observed for different flovours of loaded VMs. We noticed slow incraese of data transfer for different loaded flavours of VMs compared to the unloaded VMs using Coleraine-Halifax link (i.e., up to 40% more network data transfer for unloaded VMs compared to loaded VMs).

1. Total Data Transfer for DIfferent Flavours – loaded and unloaded

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link** | **VMs** | **t** | **s** | **m** | **l** | **xl** |
| Coleraine-Dublin | Unloaded | 1027 | 1810 | 8080 | 16600 | 31800 |
| Loaded | 430 | 1760 | 4820 | 8700 | 15100 |
| Coleraine-Halifax | Unloaded | 460 | 1810 | 4300 | 14300 | 16700 |
| Loaded | 380 | 1710 | 3890 | 8050 | 13500 |



1. Total Netwrk data transfers for different flavours – loaded and unloaded

The key observations in this work are:

* Significant increase of migration time for extra large loaded VMs whereas no significant change of NFS live migration time for tiny VMs.
* The downtime is not significantly changed in our experiments using NFS live migrations of different flavour VMs.
* Network Data transfer was doubled of the memory size for unloaded different flavours of VMs where as no significant change of total data transfer for loaded tiny VMs.

# Conclusions

Live migration is quite widely used technique for data canters in many scenarios. However, potential works still need to be done in Openstack live migration. In this work we have presented an analysis of post-copy NFS live migration in a high speed 10 Gb/s optical network infrastructure. Using the post-copy approach on 10 Gb/s infrastructure, the live migration has took significantly increased time for extremely large loaded VMs in both Coleraine-Dublin and Coleraine- Halifax Links. However, among the two links, the migration in Coleraine-Halifax took comparatively longer time because of high latency.

The change of the link latency has not severely affected the downtime. We observed as a stable and with a very little insignificant variation (all the cases 0.2s-0.3s change). Data transfer using the high speed optical network was highly increased with respect to the VM memory size and doubled of the memory size for unloaded VMs.

The results in this paper can help people understand the implications of using live migration in production Openstack environments in NFS. In future work, we will look at further development and optimization techniques for live migration.

Acknowledgements

This research is supported by the InvestNI Digital Infrastructure Project and is funded by Invest Northern Ireland.

##### References

1. B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster , “Resource Leasing and the Art of Suspending Virtual Machines”. The 11th IEEE International Conference on High Performance Computing and Communications (HPCC-09), June 25-27, 2009, Seoul, Korea.
2. N. Bobroff, A. Kochut, and K. Beaty, Dynamic placement of virtual machines for managing SLA violations,in Proc. IM, Munich, Germany, G2007, pp. 119–128.
3. M. I. Biswas, G. Parr, S. McClean, P. Morrow and B. Scotney, "SLA-Based Scheduling of Applications for Geographically Secluded Clouds", 1st workshop on Smart Cloud Networks & Systems (SCNS'14), December 3-5 Paris, France, 2014.
4. Welcome to Nova’s developer documentation! http://docs.openstack.org/developer/nova/
5. C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt,

and A. Warfield. Live Migration of Virtual Machines. The 2nd conference on Symposium on Networked Systems Design & Implementation -Volume 2, pages 273–286, 2005.

1. Ibrahim, K.Z., Hofmeyr, S., Iancu, C., Roman, E.: Optimized pre-copy live migration for memory intensive applications. In: Proc. 2011 Intl. Conf. High Performance Computing, Networking, Storage and Analysis, p. 40. ACM (2011)
2. M. R. Hines and K. Gopalan. Post-copy based Live Virtual Machine Migration using Adaptive Pre-paging and Dynamic Self-ballooning. The 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pages 51–60, 2009
3. F. Moghaddam and M. Cheriet. Decreasing live virtual machine migration down-time using a memory page selection based on memory change PDF. Networking, Sensing and Control (ICNSC), 2010 International Conference on, pages 355 –359, April 2010.
4. Aidan Shribman, Benoit Hudzia. “Pre-Copy and Post-Copy VM Live Migration for Memory Intensive Applications”, LNCS 7640, pp. 539-547, Springer-Verlag Berlin Heidlberg 2013.
5. Hibernia Networks: Security Through Diversity

http://www.hiberniaatlantic.com/documents/LATENCYApril2007-1.pdf

1. Stress(1): impose load on/stress test systems - Linux man page, <http://linux.die.net/man/1/stress>
2. W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya. Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. Proceedings of the 1st International Conference on Cloud Computing, pages 254–265, 2009.