SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

Enterprise Standards and Best Practices for IT Infrastructure

4th Year 2nd Semester 2014

Name : Meegama C.I

SLIIT ID : IT13011680

GitHub Link : https://github.com/chamath93/Baremetal-Virtual-Machine-Installation.git

Practical Session : WE Tuesday

Practical Number : vMotion

Date of Submission :

Date of Evaluation : _____

Evaluators Signature : _____

Contents

1. Introduction

VMware VMotion enables the live migration of running virtual machines from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. It is transparent to users.

VMotion is about migrating a virtual machine from one ESX host to another without incurring downtime.

VMotion lets you:

- Automatically optimize and allocate entire pools of resources for maximum hardware utilization and availability.
- Perform hardware maintenance without any scheduled downtime.
- Proactively migrate virtual machines away from failing or underperforming servers.

1. The first step is to ensure that the source VM can be operated on the chosen destination server.
2. Then a second VM process is started on the target system and the resources are reserved.
3. Next a system memory checkpoint is created. This means all changes to the source VM are written to an extra memory area.
4. The contents of the system memory recorded at the checkpoint are transferred to the target VM.
5. The checkpoint/checkpoint-restore process is repeated until only the smallest changesets remain in the target VM's memory.
6. The CPU of the source VM is stopped.
7. The last modifications to the main memory are transferred to the target VM in milliseconds.
8. The vMotion process is ended and a reverse ARP packet is sent to the physical switch (important: Notify Switches must be activated in the properties of the virtual switch). Hard disk access is taken over by the target ESX.
9. The source VM is shut down. This means the VM process on the source ESX is deleted.

vMotion checkpoints record:

- All devices and their status
- CPU registers
- Main memory contents
- A serialization of the status for transmission over the network

As you can see vMotion is concerned mostly with the transfer of the main memory contents from one ESX server to another, with a final notification telling the physical network about the new interface over which the VM is reachable sent once the process is finished. The guest system of course does not notice anything.

2. Requirements

vMotion is intervening in an active virtual machine without that virtual machine's knowledge, certain conditions must be fulfilled so that the process can run without problems or failures:

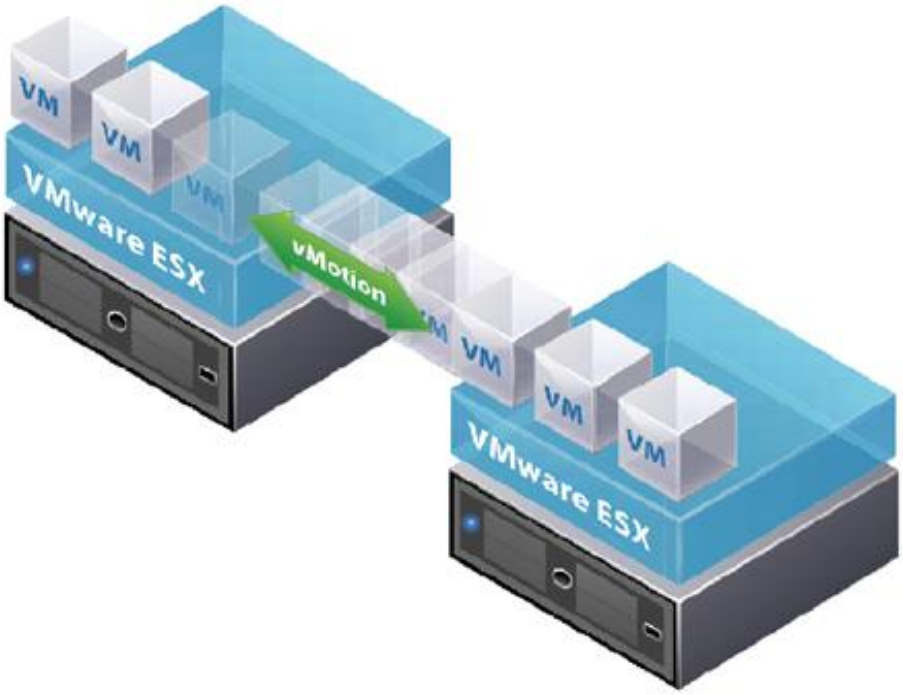
- CPU Compatibility
- vMotion interface (minimum 1GB adapter)
- Shared central mass storage
- Same naming for virtual port groups
- Sufficient resources on the target host
- At least one vSphere Essentials Plus license on the corresponding ESXi host.

For a VMotion event to be successful the following must be true:

- ✓ The VM cannot be connected to an internal vswitch.
- ✓ The VM cannot be connected to a CD-ROM or floppy drive that is using an ISO or floppy image stored on a drive that is local to the host server.
- ✓ The VM's affinity must not be set, i.e., binding it to physical CPU(s).
- ✓ The VM must not be clustered with another VM (using a cluster service like the Microsoft Cluster Service (MSCS)).
- ✓ The two ESX servers involved must both be using (the same!) shared storage.
- ✓ The two ESX servers involved must be connected via Gigabit Ethernet (or better).
- ✓ The two ESX servers involved must have access to the same physical networks.
- ✓ The two ESX servers involved must not have virtual switch port groups that are labeled the same.
- ✓ The two ESX servers involved must have compatible CPUs. (See support on Intel and AMD).

If any of the above conditions are not met, VMotion is not supported and will not start. The simplest way to test these conditions is to attempt a manual VMotion event. This is accomplished by right-clicking on VM in the VI3 client and clicking on "Migrate..." The VI3 client will ask to which host this VM should be migrated. When a host is selected, several validation checks are performed. If any of the above conditions are true then the VI3 client will halt the VMotion operation with an error.

3. vMotion Process



1. A request has been made that VM-A should be migrated (VMotioned) from ESX-A to ESX-B

2. VM-A's memory is pre-copied from ESX-A to ESX-B while ongoing changes are written to a memory bitmap on ESX-A.

3. VM-A is quiesced on ESX-A and VM-A's memory bitmap is copied to ESX-B.

4. VM-A is started on ESX-B and all access to VM-A is now directed to the copy running on ESX-B.

5. The rest of VM-A's memory is copied from ESX-A all the while memory is being read and written from VM-A on ESX-A when applications attempt to access that memory on VM-A on ESX-B.

6. If the migration is successful VM-A is unregistered on ESX-A.

4. vMotion Usage

Now that VMotion is enabled on two or more hosts, when should it be used? There are two primary reasons to use VMotion: to balance the load on the physical ESX servers and eliminate the need to take a service offline in order to perform maintenance on the server.

VI3 balances its load by using a new feature called DRS. DRS is included in the VI3 Enterprise edition along with VMotion. This is because DRS uses VMotion to balance the load of an ESX cluster in real time between all of the server involved in the cluster.

While DRS migrates VMs here and there with VMotion, it is also possible to migrate all of the VMs off of one host server (resources permitting) and onto another. This is accomplished by putting a server into "maintenance mode." When a server is put into maintenance mode, VMotion will be used to migrate all of the running VMs off it onto another server. This way it is possible to bring the first server offline to perform physical maintenance on it without impacting the services that it provides.