CMPE 283 Project 1

Building an Availability Manager to monitor the status of Virtual Machines
Individual project: This project is to be implemented by individual effort
Due date: March 17th 2014

1. Project Overview

The theme of the course project is disaster recovery. A disaster recovery system monitors the health of the virtual machines running at a site and if that site should fail, determines the site/host that is to be used to restart each failed virtual machine. Your project assignment is to prototype either the monitoring component of such a system.

1.1.Project Goals

The goal of this project is for you to

- Gain experience with several hypervisors and the corresponding management servers
- Explore the capabilities offered by these components and their APIs,
- Apply the concepts discussed in the course to a real-world problem, and
- Learn about the issues arising from interoperability

2. General Project Requirements

2.1.Requirements

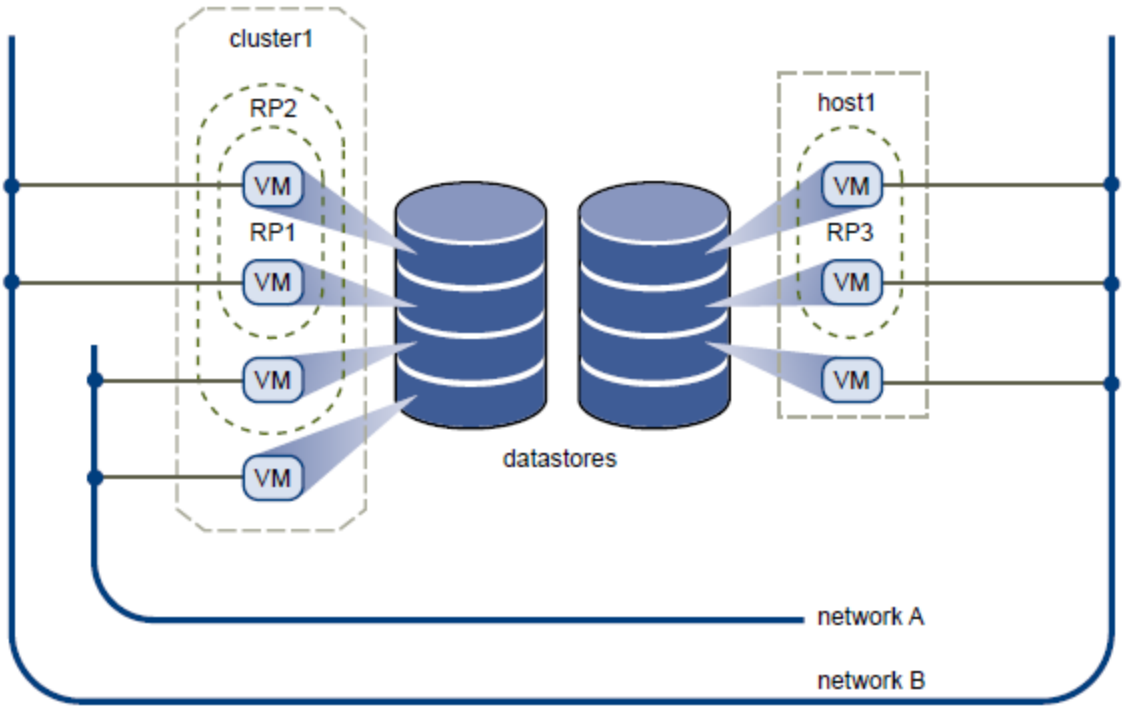


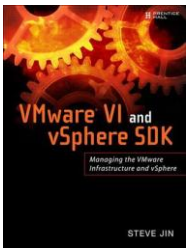
Figure 1 Virtual Datacenter Architecture

- Each will write an availability manager using ESXi's Java API (WS API) to provide the required functionality. It will interface to the virtual infrastructure using the APIs provide by the management servers and hypervisors.

- Your solution will be used to manage a "datacenter" comprising the following elements
  - A host running VMware ESXi that is managed by VMware vCenter Server
- You will be graded based on your solution working with the "datacenter" noted above.

2.2.What is VI API?

- The VMware Infrastructure API (VI API) provides language-neutral interfaces to the VMware infrastructure management framework.
- The VI API is implemented as industry-standard Web services, hosted on VirtualCenter Server and ESX Server systems.
- VMware Infrastructure (VI) SDK 2.5.0 Readme
- Java API: http://vijava.sourceforge.net//
- Tutorial at http://vijava.sourceforge.net/doc/getstarted/tutorial.htm

- Book: 

VMware VI and vSphere SDK: Managing the VMware Infrastructure and vSphere by Steve Jin

3. Availability Manager

3.1.Objective

Build an availability manager that monitors the liveness of the virtual machines (VM) running on any one of the hosts and restarts any virtual machines that fail on alternate, healthy hosts, using an earlier cached version of the VM. (In practice, such caching is done to enable recovering from complete site failures.)

3.2.Solution Requirements

For the purpose of this project, a VM can be considered to be live if it responds to pings, and dead if it stops responding for a configurable amount of time (say 1 minute ping heartbeat) .

Initially, there should only be one vHost added to the vCenter. When your solution detects that a VM has failed, it shall try to check if the vHost is active. If the vHost has failed, check if an alternative vHost is available. If a second live vHost is found, select that vHost for the VM and restart the VM on that vHost using a VM image format that is suitable for that vHost. In case if no other vHost is found, your solution should add another vHost to the vCenter, and restart the VM on that vHost The image format conversions must be done ahead of time and stored for later use. Your solution must provide a mechanism for refreshing this image cache from the current running instance of the virtual machine, for example every 10 minute update. The conversion and refreshing must be automated. That means that you need to write a program to automate this refreshing of the clone update.

When selecting a candidate host, your availability manager must take into account whether the host itself is live. (i.e., responds to pings).

Your solution can make use of unique capabilities of each of the hypervisors/management servers.

The solution must provide a mechanism for adding or removing a host from the set of hosts in the resource pool (collection of hosts (say 2 vHosts) being monitored), and for configuring the VMs that are to be monitored by it.

Your solution must provide following features:

1. Gather statistics (such as CPU, I/O, network etc) for a VM and display in a text format
2. Refresh the backup cache update every 10 minute.

3. When a VM fails with ping heartbeat, then failover to another VM host/resource pool using VMDK image format (Cold migration)
4. If the vHost is not alive, try to make it alive. If even after a fixed number of attempts, the vHost does not come up, remove the vHost from the list.
5. Add a vHost to the vCenter if there exist only one vHost which is not alive.
6. Setup alarm on VM power off. If a VM is powered off by a user, then it should be able to prevent a failover from occurring. (A VM is not failed in this case by powered off by a user)

3.3. Questions (answers to hardcopy submission)

1. In case of failure, what is a good approach during Disaster Management of Virtual Machines-
    - Check the Host first, then the Virtual Machine
    - Check the Virtual Machine, then the Host

   Justify your answer with sufficient reasons

2. How many threads do you think are required for the optimal functioning of your Availability Manager? Support your answer with reasons.

3. How did you set and use the Alarms to aid your Availability Manager? Where else can you use the alarms? Explain

3.4.Hints

- To simulate a VM failure, reconfigure the test VMs so that they no longer have network connectivity (disable network interface)

4. Project Submission

4.1.Report Requirements (Canvas Submission)

:

- Project report (Why? and Why not?)
    - Introduction (goals, objectives, needs, …)
    - Background (context of project)
    - Requirements (functional and nonfunctional)
    - Design (architecture, components, key workflows, …)
    - Implementation (environment, tools, approaches, scripting language and tools uses, screenshots(explain screenshots)
    - Answers to questions: Be sure to answer all three questions.
    - Discussion: Be sure to explain how the following four required features are implemented

        - The host add/remove mechanism
        - The approach used to configure the failure detection for each VM
        - How host failures were detected
        - The mechanism used to convert between the image formats used by the hypervisors
    - Conclusion (lessons learned, challenges, recommendation for future projects, …)
    - References

4.2.Submission

- **On-line submission**: Submissions shall be made via canvas. Submit your report in Project1_report.docx format. Source codes in projec1_source.zip in zipped format (**do not include executable codes, if so, there is automatic 5 points deduction**).
- **Only answers to questions in hardcopy**
- **Demo:** show demo if you are asked.