Krishna Chouhan
Mtech CSE,
IIIT Hyderabad

Virtualization Orchestration Layer

4th March 2015

Aim: Building a Virtualization Orchestration Layer

OVERVIEW

Introduction

In today's world, Hypervisor is the new OS and Virtual Machines are the new processes. Many system programmers are familiar with the low level APIs that are exposed by the operating systems like Linux and Microsoft Windows. These APIs can be used to take control of the OS programmatically and help in developing management tools. Similar to the OS, Hypervisors expose APIs that can be invoked to manage the virtualized environments. Typical APIs include provisioning, deprovisioning, changing the state of VMs, configuring the running VMs and so on. While it may be easy to deal with one Hypervisor running on a physical server, it is complex to concurrently deal with a set of Hypervisors running across the datacenter. In a dynamic environment, this is a critical requirement to manage the resources and optimize the infrastructure. This is the problem that we try to solve in this project.

Problem Definition

Build a fabric that can coordinate the provisioning of compute resources by negotiating with a set of Hypervisors running across physical servers in the datacenter.

Expected Outcome

1. Resource Discovery;
2. Resource Allocation;
3. A REST API Server;

Resource Discovery

Discover Resources available with the resources.

Resource Allocation

Decide what resources to allocate to fulfil the request. It should be loosely coupled in the sense that mechanism and implementation should be separate. What this means is it should be a possible to change the “Algorithm” for allocation with minimal code changes or still better with change in configuration only.

Rest API Server

A Rest API server that can be consumed by a variety of clients to deal with the virtual infrastructure.

SPECIFICATIONS

Expected APIs

VM APIs:

VM_Creation
VM_Query
VM_Destroy
VM_Type

Resource Service APIs:

List_PMs
List_VMs
PM_Query

Image Service APIs:

List_Images

Other:

Returns 0 for any other Queries.

MILESTONES

Phase 1

Goal in phase 1 is to achieve all the functionalities related to VM APIs. VM creation and destruction, query and type of VM, APIs are created considering that all the images of OS are located at a certain predefined location. Persistent Storage must be kept in mind.

Phase 2

Goal in phase 2 is to achieve all the functionalities in the project including VM and PM Queries along with final deployment of the server (Flask).