Project 7d
XCP and Open-Nebula Integration

Team Leader : SUBHASHREE S

Name	Roll No	Contact No	Email
LAXMIKANTH CH	MT2011032	9880480435	laxmikanth.chindam@iiitb.org
RAVIKIRAN S	MT2011123	8105096999	Ravikiran.S@iiitb.org
RAVI PARMAR	MT2011124	8088628992	Ravi.Parmar@iiitb.org
SHIKHA SINGHAL	MT2011146	8880230668	Shikha.Singhal@iiitb.org
SUBHASHREE S	MT2011155	7204382211	Subhashree.S@iiitb.org

Contents

1 Introduction

Background Information

Virtualization refers to technologies designed to provide a layer of abstraction between computer hardware systems and the software running on them [1]. OpenNebula is a virtualization tool which can be considered as an Infrastructure cloud manager that fulfils the role of management system in an IaaS Cloud [2]. OpenNebula aims to provide a open, flexible, extensible, and comprehensive management layer to automate and orchestrate the operation of virtualized data centers by leveraging and integrating existing deployed solutions for networking, storage, virtualization, monitoring or user management, while XCP addresses the needs of cloud providers, hosting services and data centers by combining the isolation and multi-tenancy capabilities of the Xen hypervisor with enhanced security, storage and network virtualization technologies to offer a rich set of virtual infrastructure cloud services [3].

Project Description

Our project work involves the integration of XCP and OpenNebula which shall be considered as a process of bringing together two highly sophisticated emerging technologies to enjoy the benefits of both. XCP as a platform, does not provide the overall cloud architecture, but rather focuses on configuration and maintenance of clouds. But it enables external tools such as OpenNebula, to better leverage the Xen hypervisor [4]. Hence, this integration will bring the rich capabilities of XenAPI to OpenNebula. XCP provides a complete cloud platform to OpenNebula, with enhanced security, storage and network virtualization. Meanwhile, OpenNebula provides cloud orchestration to XCP with adaptable, extensible, proven, and interoperable data center virtualization management [5]. The collaboration will add OpenNebula support to the list of Cloud Orchestration stacks that build on top of XCP.

2 Existing Systems

OpenNebula has been already integrated with some components, such as Claudia. Its modular design allows an easy integration with any components in the cloud ecosystem, this modularity stemming from its open architecture, interfaces and code [2]. In November 2011, the first integrated prototype of XCP and openNebula was released by C12G Labs. C12G Labs is the organization which provides commercial support and value-added solutions around

OpenNebula and the XCP project team and Xen.org community have provided the technical guidance and assistance to this open-source project. This prototype is said to be the first result of the collaboration between OpenNebula and Xen. The product has been developed and released in the form of XCP drivers addon (called as OpenNebula Toolkit for XCP) which enable the management of an OpenNebula cloud based on Xen Cloud Platform (XCP) hypervisors. It uses the `"xe"` command line tool to invoke the XAPI interface exposed by the XCP hypervisors, and features an installation process that will leverage the stability, performance and feature set of any existing XCP based OpenNebula cloud. These drivers enable OpenNebula to perform basic VM lifecycle (create, monitor, and shutdown) management [6]. This was considered as a huge step forward towards achieving a complete open-source stack for cloud infrastructure deployment. The XCP drivers have been hosted as freely available open source project on OpenNebula.org. and the implementated scripts are in Ruby.

3 Gap Analysis

There are two ways of going about doing this process of integration. As specified in the above section, components (XCP drivers built for OpenNebula) of the integrated system can make use of `"xe"` which is a command line client that speaks the XenAPI XML/RPC protocol. Another approach would be to apply the direct usage of the XML/RPC interface for OpenNebula integration rather than forking `"xe"` and parsing the results [7].

Between these two approaches, the first one is expected to produce feasible and efficient results as the other method tends to mean that building compatible components for integration of two widely varying sophisticated systems might turn out to be infeasible. Nevertheless, there are a few features that the first approach has not provided in the existing system [6]:

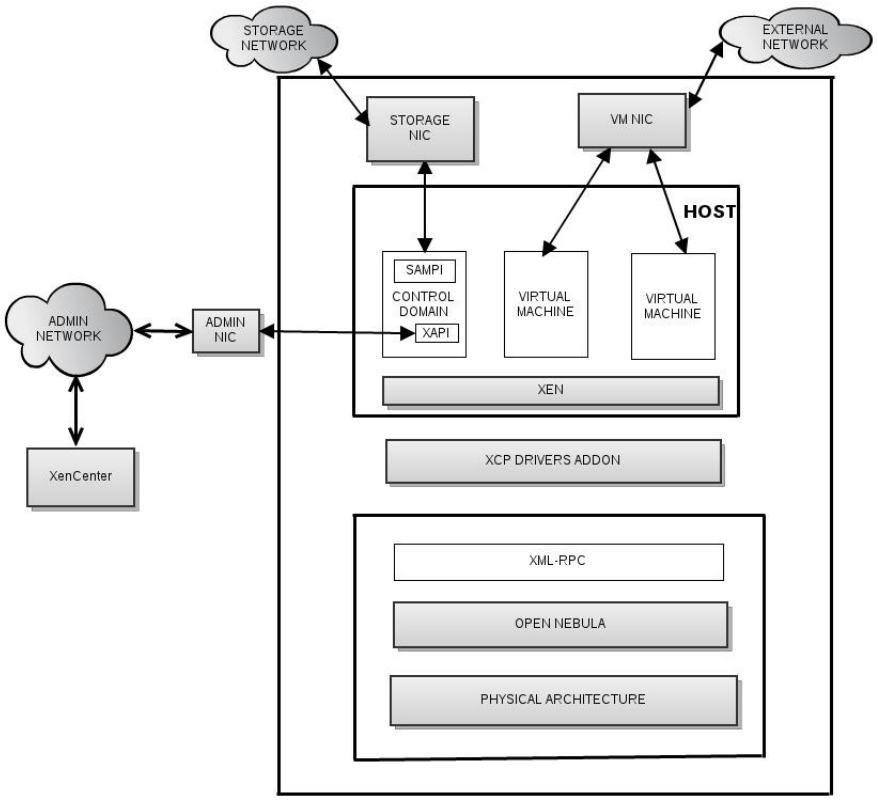
- Lacking support for live migration: Live migration is an extremelely powerful tool from the perspective of cluster administrators. Hardware and software considerations can be separately dealt with and consolidation of clustered hardware into a single coherent management domain can be performed through live migration [8]. If a physical machine needs to be removed from service, the facility for an administrator to migrate OS instances including the applications that they are running to alternative machine(s) is provided, freeing the original machine for maintenance.

- Sunstone cannot be used to add XCP hosts: Management of virtual

and physical resources by means of CLI is the norm in OpenNebula. Sunstone is a GUI intended for users and admins of OpenNebula, that will simplify the typical management operations in private and hybrid cloud infrastructures.

The proposed project will focus upon overcoming these limitations of the existing system too.

4 System Architecture



The architecture given above can be described as follows:

OpenNebula

The core of OpenNebula along with its drivers and physical infrastructure forms the bottom-most layer of the architecture. OpenNebula supports many kinds of interfaces among which XML/RPC is considered to be a predominantly important one. XML/RPC is a protocol which enables OpenNebula

to interact with different kinds of toolstacks using remote procedure calls to a system.

Need for XML/RPC in the Integration Process

Generally, using OpenNebula a text file which describes virtual machine can be created and this text file is processed using `"xm"` command. But the same text file cannot be processed in XCP because XCP comes with a separate set of options bundled in a peculiar form called `"xe"` which is a command line client. `"xe"` is the only way to talk to XAPI, the front end in XCP. XCP is found to lack the text file approach, but in turn has a more robust RPC based method. Hence the need to develop a separate set of drivers (XCP drivers addon) which uses `"xe"` to interact with XAPI arises.

XCP

XCP and its different components form the uppermost layer of the architecture. The components of XCP comprise the following [9]:

- Xen hypervisor which runs in 64-bit mode and virtualizes the CPUs, interrupts and host memory.
- The XAPI management stack runs inside the control domain and manages all resources required for running guest domains.
- The storage manager (or SMAPI) runs inside the control domain and provides a consistent interface to a variety of storage backends.
- Xen Cloud Platform can run multiple virtual machines on the same host, each of which provides entirely isolated computation.
- Finally, multiple Xen Cloud Platform hosts can be aggregated into a resource pool which acts as a single unit of administration across a cluster of machines.

Drivers for Integration

XCP drivers addon (also known as OpenNebula toolkit for XCP) finds its place in between the two layers in the architecture. It makes use of the XML/RPC mechanism to interface with the components of XCP. Our work would involve the study and analysis of these drivers in order to bring about the integration of XCP and OpenNebula in a successful manner and also to incorporate new features into this addon so as to offer the capabilities which the existing system is found to be lacking.

5 Development Plan

The implementation plan can be put into the following stages:

Stage 1

As both XCP and OpenNebula are highly sophisticated emerging technologies with very recent history, the first step in our plan would be setup the complete environments of OpenNebula and XCP and get a practical exposure to both. Looking up for and getting to know the important commands and techniques used in both would form a vital part of this stage of the plan. Some significant commands of XCP such as `"xe"` and `"xl"` are to be dealt with in detail as a part of this study.

Stage 2

The implementation of the open source XCP drivers addon needs to be understood completely in order to integrate XCP and OpenNebula successfully. This analysis would be performed in the second stage.

Stage 3

Using the OpenNebula Toolkit for XCP the actual integration of XCP and OpenNebula is to be performed in this stage. The toolkit invokes the XAPI interface exposed by the XCP hypervisors by making use of the `"xe"` command line tool provided by XCP.

Stage 4

Incorporating new features to the integrated system would be the primary goal of the fourth stage. The existing system is found to be devoid of a few powerful facilities like live migration and graphical interface. Provision of features to the integrated system such that the drawbacks of the existing system are met with, will be focussed upon here.

Stage 5

The last stage would emphasise on making the system completely compliant to develop OpenNebula clouds rich with XAPI features and testing of the entire system.

6 Timeline

Date	Task
January 17	Proposing the initial draft of the project and choosing of team leader.
January 27	Address review comments and finalize project goals document. Setting up of SVN repositories.
January 31	Installation of XCP and OpenNebula and getting hands on with them. This corresponds to Stage 1 of the development plan.
February 7	Brief presentation of the project architecture and plans. Analyze and understand the XCP drivers. This corresponds to Stage 2 of the development plan.
March 1	Mid term review of the project. Release of 0.1/alpha version (Integration of XCP and OpenNebula using XCP drivers addon, corresponding to Stage 3 of the development plan).
April 3	First draft of final documents. Second detailed review with release of Beta version (Integrated system with new features included). This corresponds to Stage 4 of the development plan.
April 19	Testing and review of the project, corresponding to Stage 5 of the development plan. Submission of final documents and reports.

References

[1] J. K. Waters, "Virtualization definition and solutions," http://www.cio.com/article/40701/Virtualization_Definition_and_Solutions.

[2] "OpenNebula: Infrastructure Management in Cloud Computing Architectures by DSA-Research Group, Universidad Complutense de Madrid." [Online]. Available: http://www.future-internet.eu/fileadmin/documents/architecture_workshop/documents/OpenNebula_As_A_Management_System_In_An_IaaS_Cloud_Computing_Architecture.pdf

[3] "Xen Cloud Platform Projects," http://www.xen.org/products/cloudxen.html.

[4] T. Cordeiro, D. Damalio, N. Pereira, P. Endo, A. Palhares, G. GonÃğandalves, D. Sadok, J. Kelner, B. Melander, V. Souza, and J.-E. MÃěandngs, "Open source cloud computing platforms," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, nov. 2010, pp. 366 –371.

[5] D. Marshall, "OpenNebula Releases Xen Cloud Platform Integration Prototype," http://vmblog.com/archive/2011/11/03/opennebula-releases-xen-cloud-platform-integration-prototype.aspx.

[6] "Xen Cloud Platform Drivers," http://www.opennebula.org/software:ecosystem:xcp.

[7] "About Xen OpenNebula Integration," http://old-list-archives.xen.org/archives/html/xen-api/2011-02/msg00005.html.

[8] C. Clark, K. Fraser, H. Steven, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2005, pp. 273–286.

[9] "Xen Cloud Platform 0.1 User Security," xen.org/files/XenCloud/user_security.pdf, October 2009, [Online; accessed 24-Jan-2012].