KVM & OpenVZ virtualization based IaaS Open Source Cloud Virtualization Platforms: OpenNode, Proxmox VE

A. Kovari, P. Dukan
College of Dunaújváros/Institute of Informatics, Dunaújváros, Hungary
kovari@mail.duf.hu, dukanp@mail.duf.hu

***Abstract* - The evolution of cloud computing over the past few years is one of the major advances in the history of computing. OpenNode Virtualization Platform is an open source CentOS based server virtualization and management solution, and Proxmox VE is a tweaked Debian distribution with a custom optimized kernel. The two platforms are highly identical, but they are based on distinct Linux distributions, and there are minor differences. In this paper these two open-source IaaS systems are analyzed as KVM & OpenVZ based virtualization solutions.**

I. Cloud, IaaS, Virtualization Platforms

Cloud computing is an aggregation of computing resources, networking and storage management solutions, and virtualization applications which are available on demand, and delivered economically. The cloud computing was coined in 2007 and this term came into popular use in 2008, currently emerges as a hot topic due to its abilities to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments, and configurable software services [1]. Cloud computing is a new field in Internet computing that provides novel perspectives in Internet Technologies and raises issues in the architecture, design, and implementation of existing networks and data centers. It is a new method of running application software and storing related data in central computer systems that providing access to customers or other users through the Internet [2]. So it is the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients over a network.

The evolution of cloud computing over the past few years is one of the major advances in the history of computing. Cloud computing is becoming an adoptable technology with its dynamic scalability and usage of virtualized resources as a service through the Internet (Fig. 1.). [8]

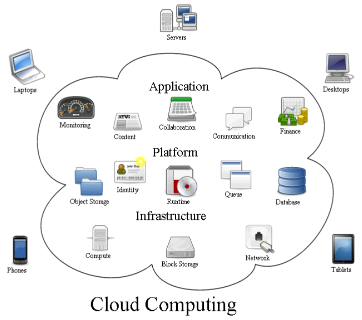

Figure 1. Cloud Computing (created by Sam Johnston)

The cloud platform is a system where software applications may be run in an environment composed of cloud services in a logically abstract environment. With the popularization of cloud computing, several enterprises and open-source communities have developed their own cloud solutions.

Cloud computing encompasses a number of different services and cloud computing providers offer their services according to three fundamental models (Fig. 2.). One set of services, called software as a service (SaaS), a software delivery model in which software and associated data are centrally hosted on the cloud and involves the supply of a discrete application to outside users. Platform as a service (PaaS) provides a computing platform and a solution stack as a service, involves the supply of remote computing capacity along with a set of software-development tools for use by outside software programmers. Infrastructure as a service (IaaS) involves the provision of computing resources and data storage to outside users, who are able to run their own applications and store their own data on the remote system. In this cloud service model, cloud providers offer physical or virtual machine (VM) based computers, raw (block addressing) storage, firewalls, load balancers, and networks. IaaS providers supply these resources on demand. [2] [3]

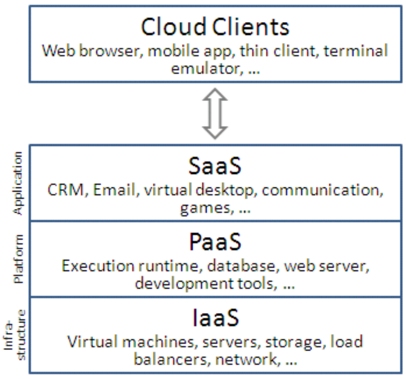
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
Application
SaaS
CRM, Email, virtual desktop, communication, games, ...
Platform
PaaS
Execution runtime, database, web server, development tools, ...
Infra-structure
IaaS
Virtual machines, servers, storage, load balancers, network, ...

Figure 2. Cloud Service Model (created by Bikeborg)

Virtualization is a framework or methodology of dividing computer resources into multiple isolated execution environments. There are six types of such virtualization: full-, hardware-, partial- and paravirtualization, operating system (OS) level virtualization, and multiserver (cluster) virtualization. Virtualization techniques create multiple isolated

978-1-4673-4750-1/12/$31.00 ©2012 IEEE

partitions – Virtual Machines (VM) or Virtual Environments (VEs) – on a single physical server. [4][5]

Full Virtualization: run any non-modified operating system which supports the platform being emulated. The main disadvantages of emulation are low performance and low density. Examples: QEMU, Microsoft Hyper-V, Oracle VirtualBox, VMware Workstation, VMware Server, Microsoft Virtual PC, KVM. [4][5]

Hardware Assisted Virtualization: hardware provides architectural support that facilitates building a virtual machine monitor and allows guest OSs to be run in isolation. [6] Examples: Microsoft Hyper-V, Oracle VirtualBox, VMware Workstation, Microsoft Virtual PC, KVM, Xen. [5]

Partial Virtualization: some, but not the whole target environment is simulated. The virtual machine simulates multiple instances of much of an underlying hardware environment, particularly address spaces and each virtual machine consists of an independent address space. Some guest programs, therefore, may need modifications to run in this virtual environment. Examples: IBM CP/CMS.

Paravirtualization – OS Assisted Virtualization: run multiple modified OSs on top of a thin layer called a hypervisor, or virtual machine monitor. Paravirtualization has better performance compared to emulation, but the disadvantage is that the “guest” OS needs to be modified. Examples: Xen, KVM, UML VM. [4][5]

Operating system-level virtualization: multiple isolated execution environments within a single operating system kernel. It has the best possible performance and density, and features dynamic resource management. On the other hand, this technology does not allow to run different kernels from different OSs at the same time. Examples: OpenVZ, Virtuozzo, FreeBSD Jail, Solaris Zones/Containers. [4]

Multi-server (cluster) virtualization: making very large groups of servers appear and act like a single system, as easy to manage as a single workstation. Cluster virtualization the most practical and cost-effective methodologies for reducing the complexity, cost and overall administrative burden of large scale computing enabling to get the most out of computing resources.

In this paper KVM & OpenVZ framework based virtualization platforms are investigated because the mentioned benefits of these virtualization environments: full-, hardware-, paravirtualization and operating system-level virtualization.

KVM (Kernel-based Virtual Machine) virtualization engine is a relatively new and simple unique powerful hypervisor, a Linux native kernel module, giving the Linux kernel native virtualization capabilities. Integrating the hypervisor capabilities into a host Linux kernel as a loadable module can simplify management and improve performance in virtualized environments. Because KVM uses hardware-based virtualization, it does not require modified guest operating systems. It can support any platform, given that it is deployed on a supported processor. KVM uses parts of the QEMU virtualization software to emulate actual hardware for devices. This approach has numerous advantages. Under this model, every virtual machine is a regular Linux process, scheduled by the standard Linux scheduler. Traditionally, a normal Linux process has two modes of execution: kernel and user. The user mode is the default mode for applications, and an application goes into kernel mode when it requires some service from the kernel. KVM adds a third mode, the guest mode. Guest mode processes are processes that are run from within the virtual machine. The guest mode, just like the normal mode (non-virtualized instance), has its own kernel and user-space variations. KVM makes use of hardware virtualization to virtualize processor states, and memory management for the virtual machine is handled from within the kernel. I/O in the current version is handled in user space, primarily through QEMU. A typical KVM virtualization engine consists of the following components: device driver for managing the virtualization hardware; user-space component for emulating PC hardware; and I/O model is directly derived from QEMU. [9]

Container Virtualization (OpenVZ): preferred technology for Linux servers as it is the fastest approach. OpenVZ is container-based virtualization for Linux. OpenVZ allows a physical server to run multiple isolated operating system instances, known as containers, Virtual Private Servers (VPSs), or Virtual Environments (VEs). OpenVZ technology creates multiple secure, isolated containers a single physical server enabling better server utilization and ensuring that applications do not conflict. Each container performs and executes exactly like a stand-alone server. OpenVZ is limited in that it requires both the host and guest OS to be Linux (Linux distributions can be different in different containers). A container can be rebooted independently and have root access, users and unique IP addresses, memory, processes, files, applications, system libraries and configuration files. [4]

The libvirt virtualization API: is a toolkit – developed by Red Hat – to manage most of the virtualization platforms though a common API. It can handle almost any hypervisors, container systems, or a paravirtualized kernel with the support of virtual networks and storage systems. There is a CLI for configuring these, named virsh.

Pre-built virtual appliances are available on virtualization platforms in many cases. These templates are pre-installed, pre-configured and ready to use applications and software solutions, coming prepacked with an operating system in form of a virtual machine.

II. OpenNode Virtualization Platform

OpenNode Virtualization Platform is an open source CentOS based server virtualization and management solution developed by Active Systems LLC and licensed under GPLv3. They provide paid support and development for OpenNode. Their plan is to bridge the gap that Fedora/Red Hat Enterprise Linux (RHEL) leaves because they aren’t supporting OpenVZ, but OpenNode is following CentOS as close as possible upstream apart from applying the OpenVZ patches.

The OpenNode platform is simple, based on "industry standard" components (RHEL/CentOS based distributions, libvirt, python, etc.). The platform combines strengths from different virtualization technologies: supports both OpenVZ container and KVM full virtualization with libvirt management interface.

OpenNode has automated partitioning, LVM (Logical Volume Manager), software RAID, host-node bridge networking setup, and consists of following components:

OpenNode bare-metal installer: providing easy to use ISO installer on a physical host. OpenNode install minimal CentOS server system with support of OpenVZ and KVM virtualization. It consist a local management OpenNode TUI (Terminal User Interface) Utility and well-known standard tools like virsh and vzctl are included. OpenNode provides a CentOS-to-OpenNode conversion script which can be used as an alternative to bare-metal ISO or networked PXE installations.

OpenNode Management Server (OMS): packaged and deployed simply as a VM instance onto OpenNode host. Easy installation and setup is provided via local OpenNode TUI Utility menu.

OpenNode WUI (Web-based User Interface, codenamed as FuncMAN): next-generation web-based management console on Qooxdoo AJAX application framework (does not require Java, ActiveX or Flash). WUI provides VM instances management on OpenNode hosts together with HTML5 based VNC and SSH VM consoles. OpenNode WUI is included in OpenNode Management Server appliance.

OpenNode CLI Management: OpenNode host provides and supports variety of well-known standard CLI (Command Line Interface) tools for a local VM management (no need for an OpenNode Management Server): vzctl and virsh for a OpenVZ containers management, virt-install for a KVM creation and virsh for a KVM management. Remote/central CLI management is possible with 'func shell' in OpenNode Management Server or using libvirt compatible tools.

VM templates can be downloaded by OpenNode TUI utility and WUI. These templates include prepackaged and supported IT solutions (Zimbra, OpenVPN, Samba-LDAP-PDC, LTSP, Apache, MySQL, etc.) together with solution/service management consoles that integrate directly into OpenNode WUI and OMS.

An OpenNode infrastructure contains three main segments: The heart of the cloud is the Management Server, which control the other endpoints (nodes). It is running the management services like the CLI, a webserver for the web-based Management Console (WUI), and an Application Programming Interface (API) for Remote Process Control (RPC). The next main segment composed by the nodes, which are given they resources into the cloud, and the last part is a client computer for using the WUI (Fig. 3.).

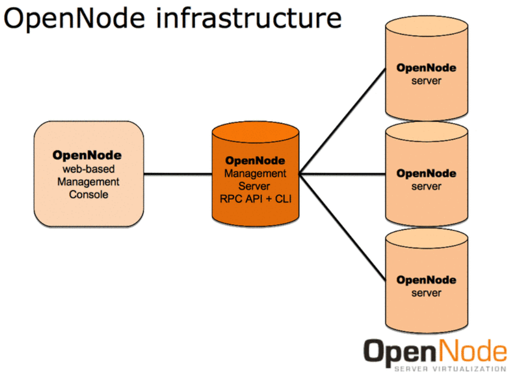
OpenNode infrastructure
OpenNode web-based Management Console
OpenNode Management Server RPC API + CLI
OpenNode server
OpenNode server
OpenNode server
OpenNode SERVER VIRTUALIZATION

Figure 3. OpenNode infrastructure

Another aspect is the two layer of the cloud. The upper layer is for control the nodes, and the lower layer contains the node-functions. That means separated modules like LVM, OpenVZ, KVM, CertMaster or the hardware for statistics (Fig. 4.).

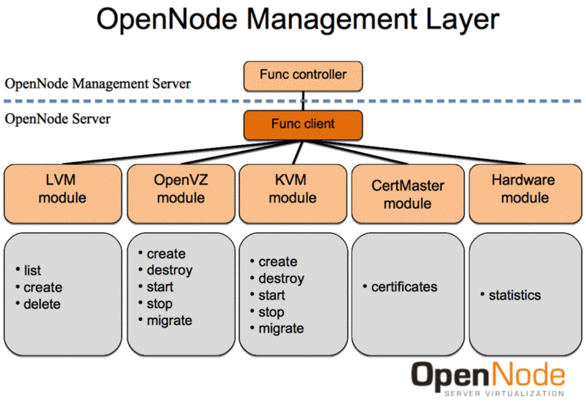
OpenNode Management Layer
OpenNode Management Server
Func controller
OpenNode Server
Func client
LVM module
OpenVZ module
KVM module
CertMaster module
Hardware module
• list • create • delete
• create • destroy • start • stop • migrate
• create • destroy • start • stop • migrate
• certificates
• statistics
OpenNode SERVER VIRTUALIZATION

Figure 4. OpenNode Management Layer

From the viewpoint of a system-administrator, the whole Private Cloud Infrastructure (PCI) can be reached from a client computer, by running the OpenNode Console (ONC). The ONC connect to the OMS which control the PCI nodes like Data Centers (Fig. 5.).

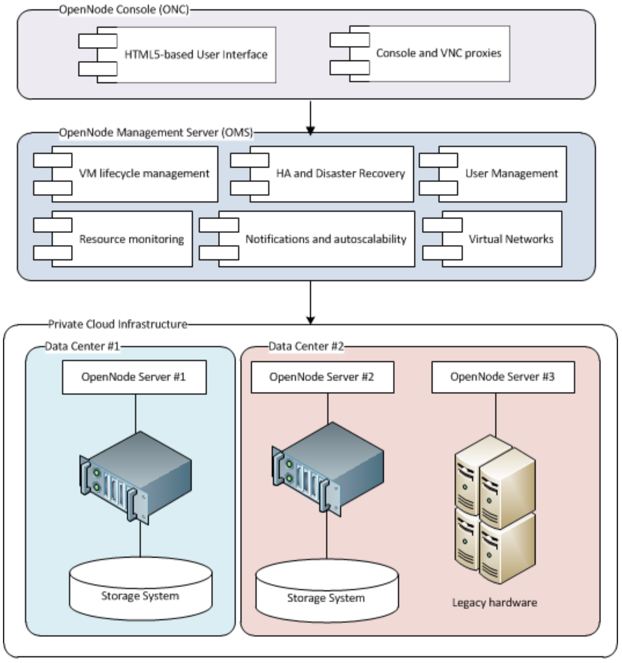

Figure 5. OpenNode Infrastructure

It is easy to see, that the key of the OpenNode distribution is the OMS. So let us take a look inside the working of that: If WUI is used, every command will be sent as a HTTP GET request. It will be handled by the RPC daemon, which can reply through HTTP JSON replies. Henceforward the daemon sends the command to the chosen service (depending on the selected function). This command will be translated to a GeneralService call (to preserve easier the compatibility with the previous versions), that performs the task by the "Func controller".

On the other side at the authentications and cache management the DB controller will be used, to reach the cache and session table (Fig. 6.).

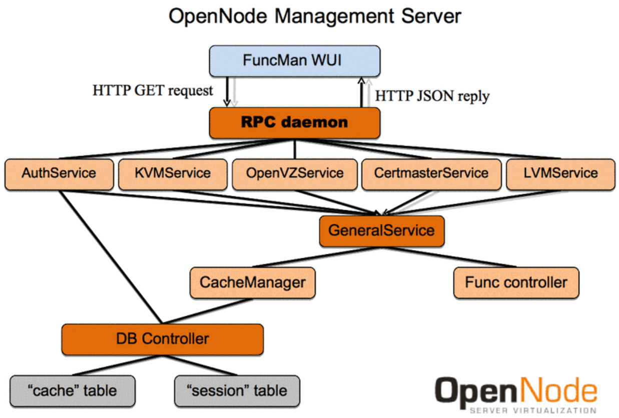

Figure 6. OpenNode Management Server

III. Proxmox VE Virtualization Platform

Proxmox VE is an open source project, developed and maintained by Proxmox Server Solutions GmbH. Proxmox VE 2.X is licensed under GPLv3. Proxmox VE is simple, easy-to-use turnkey solution. Everything can be done via a web browser: integrated console view to the Virtual Machines, integration and management of Proxmox VE Cluster, AJAX technologies for dynamic updates of resources, secure access to all Virtual Machines via SSL encryption. Proxmox VE is a tweaked Debian distribution that installs a web-based management interface, and has a custom optimized kernel. Proxmox VE offers two solutions for virtualization: full- or paravirtualization using KVM and OpenVZ container-based virtualization. Pre-built container based VM templates can be downloaded by web interface. Java-based VNC viewer can interact with the VM's console. Proxmox VE has role based user and permissions management for all objects such as VMs, storage pools, nodes, and so on, and user authentication can be done with Linux PAM or LDAP. Proxmox can migrate VMs to another Proxmox VE server and even cluster Proxmox VE to create high availability (HA) virtual machines. If a Proxmox server fails, all VMs running on it are automatically migrated to a working server node.

A Proxmox VE Cluster does not need a management node. All nodes have the same management capabilities, eliminates single point of failures. Proxmox Cluster File System (pmxcfs) is a database-driven file system for storing configuration files, replicated in real time on all nodes using Corosync Cluster Engine. It is used to store all Proxmox VE related configuration files. Although the file system stores all data inside a persistent database on disk, a copy of the data resides in RAM. That imposes restriction on the maximal size, which is currently 30MB. This is still enough to store the configuration of several thousand virtual machines. [7]

Proxmox VE uses a bridged networking. For connecting VMs to the outside world, bridges are attached to physical network cards assigned a TCP/IP configuration. For further flexibility, VLANs (IEEE 802.1q) and network bonding/aggregation are also possible to build complex, flexible virtual networks for the Proxmox VE hosts. [7]

Proxmox VE has flexible storage model. Virtual machine images can be stored on local or shared storage like NFS or SAN (e.g. using iSCSI or FC). Using DRBD for KVM guests is also possible. [7].

IV. Comparison of OpenNode and Proxmox VE Virtual Platforms

The two platforms are highly identical, but they are based on distinct Linux distributions, and there are major differences between the first release dates. The next table (Table I.) compares the main features, and user activities on the two platforms.

TABLE I.
Comparison of last release of OpenNode and Proxmox VE

	OpenNode	Proxmox VE
current release version	6.0	2.1
current kernel version	2.6.32	2.6.32
base Linux distribution	CentOS 6	Debian 6
Release date (first stable)	19.10.2010	29.10.2008
bare metal ISO-installer	yes	yes
support KVM	yes	yes
support OpenVZ	yes	yes
web based management	yes	yes
virtualization API	libvirt	Proxmox VE API
open-source	yes	yes
live migration	yes	yes
HA Clustering	no (future plan)	yes
SPICE protocol	no (future plan)	no (future plan)
VNC protocol	yes	yes
Intel VT/AMD-V	needed	needed
64bit	needed	needed
forum posts	ca. 450	ca. 55000
main release updates	year	half year
Google hits	76k	414k
web page updates in last month	539	8930

V. Conclusion

However the two platforms are similar in the main features, OpenNode cannot reach the popularity of Proxmox VE. Although OpenNode has achieved the level of services provided by Proxmox VE, does not gain the desired popularity yet.

Proxmox VE uses unique virtualization API but OpenNode based on libvirt which supports several types of virtualization solution. This feature would be benefit for OpenNode in the future.

At this time Proxmox VE is much more popular than OpenNode, it has more than ten times higher user activity than OpenNode. Proxmox VE is an easy-to-use turnkey solution, has role based user and permissions management for all objects such as VMs, storage pools, nodes, and so on, and user authentication can be done with Linux PAM or LDAP. Proxmox VE web interface can manage multiple computers as a cluster also but has some minor annoyances such as non-existing and outdated templates. If OpenNode enter some important services like PXE installs, HA clustering or Network management support it would be a reasonable choice instead of Proxmox VE.

References

[1] L. Wang, G. Laszewski, A. Younge, Xi HE, Marcel Kunze, J. Tao, C. Fu, "Cloud Computing: a Perspective Study", New Generation Computing, vol. 28, pp. 137–146, Ohmsha Ltd. and Springer, 2010.

[2] "Cloud Computing", Academic Room, Retrieved 2012-06-16. http://www.academicroom.com/topics/cloud-computing

[3] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology U.S. Department of Commerce, Special Publication 800-145, 2011

[4] K. Kolyshkin, "Virtualization Comes in More than One Flavor", Virtualization Magazine, Ulitzer, Inc., 2007

[5] VMware, "Understanding Full Virtualization, Paravirtualization, and Hardware Assis", White Paper WP-028-PRD-01-01, VMware, Inc., 2007

[6] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni; F.C.M. Martins, A.V. Anderson, S.M. Bennett, A. Kagi, F.H. Leung, L. Smith, "Intel virtualization technology", Computer, vol.38, no.5, pp. 48-56, May 2005

[7] Proxmox Server Solutions GmbH, "Proxmox VE Features", Retrieved 2012-06-29. http://proxmox.org/products/proxmox-ve/features

[8] T. Ercan, "Effective use of cloud computing in educational institutions", Procedia Social and Behavioral Sciences 2, pp. 938–942, 2010

[9] M.T. Jones, "Discover the Linux Kernel Virtual Machine", IBM developerWorks, May 2007