



My Humble Abode

Assemble() -> dump_brain() -> be_creative() -> program_new_world() .. but... dont_challenge_god()

Search Your Topic here.



[Home](#) > [KVM/ Qemu](#) > [Xen and Kvm](#)

Xen and Kvm

admin June 13, 2010 [KVM/ Qemu, Virtualization, Xen](#)

4

Hey Guys ,

Thought of sharing some bits on most popular open source virtualization technologies ..

Yeah .. it is “**Xen**” and “**Kvm**”.

Mainly I am focusing more on the hypervisor loading process and features.

Xen

Xen originated as a research project at the University of Cambridge, led by Ian Pratt, senior lecturer at Cambridge and founder of XenSource, Inc. This company supported the development of the open source project and also sells enterprise versions of the software. The first public release of Xen occurred in 2003. Citrix Systems acquired XenSource, Inc in October 2007 and the Xen project moved to www.xen.org.

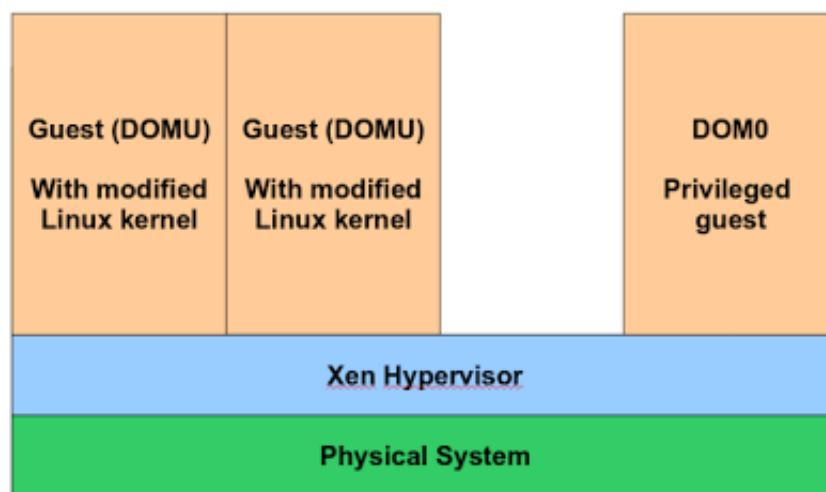
Loading Xen Hypervisor

Xen architecture consists of a Hypervisor, Domain 0 (in short Dom0, a privileged domain or guest) and Domains (DomUs or guests). The Xen hypervisor runs on the physical hardware directly and its functions includes scheduling, resource control, memory management and loading Dom0 kernel. Dom0 runs on top of hypervisor and acts as a privileged guest. It provides device drivers for host hardware devices like

network card, storage controllers and etc and it starts user space services like xend and management tools (xm, virsh, virt-manager etc). It also performs all other OS related functions.

The x86 architecture was notorious for its lack of virtualizability. Xen overcame these problems by modifying the guest OSes to communicate with the hypervisor. Whenever any instruction was to be executed which had to be virtualization-aware, a call is made to the hypervisor and the hypervisor performs the necessary task for the guest on its behalf. Due to this design, it's not possible to run OSes which cannot be modified on CPUs that do not have virtualization extensions.

When host starts, bootloader (ex: grub) boots the hypervisor first and then the hypervisor loads Dom0 kernel and initrd. DomUs or guests are later installed using the management tools provided inside dom0.



The above figure shows a general view of Xen architecture.

An example grub configuration for Xen

```

root (hd0,0)
kernel /xen.gz-2.6.18-xxx                               module /vmlinux-2.6.18-
xxx.xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet

module /initrd-2.6.18-xxx.img
  
```

The complex booting process of xen, makes hard to troubleshoot issues arises at the booting stage as it is difficult to isolate the issues in hypervisor and Dom0 kernel layer. Sometimes it is required to install and test unmodified kernel of host OS to isolate whether the issue is with Xen kernel or not. Another drawback is that, xen kernel does not support all the hardware and features that are available on linux kernel natively. Maintenance and development of xen code (without being part of upstream kernel) is an extra burden among developers. As mainly xen considered as a para virtualization instance, the guest operating system has to be modified. Unlike KVM, separate utilities (xm, xmtop ..etc) are developed

only for xen. When considering the security part, two privileged entities has to be secured, hypervisor and DOM0. But we respect Xen, for delivering near to native performance on the modified guest kernels. Also no special hardware support (virtualization Capability) is required to run guests.

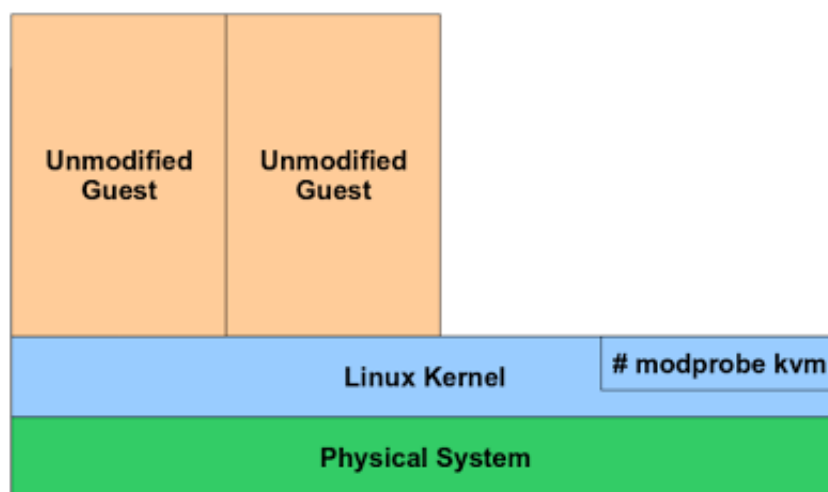
Kernel-based Virtual Machines (KVM)

KVM is a hardware assisted virtualization developed by Qumranet, Inc to offer a desktop virtualization platform using Linux servers using their proprietary (now open source) SPICE protocol. KVM make use of the advancement in the processor technologies Intel-VT_x and AMD-V to support virtualization. In 2007, KVM was merged with upstream 2.6.20 mainline kernel and in 2008, Qumranet was acquired by Red Hat, Inc.

Loading KVM Hypervisor and its components

KVM is a kernel module to the Linux kernel that actually makes Linux into a hypervisor when its a loaded. Device emulation is handle by a modified version of QEMU (qemu-kvm).

The guest running on KVM is actually executed in user space of the host. This makes each guest instance looks like a normal process to the underlying host kernel.



The above figure shows a general view of KVM architecture.

Executing the following commands as root will load KVM.

```
# egrep "svm/vmx" /proc/cpuinfo
```

If there's output from this command, it means the computer is capable of full virtualization and you can run guests under KVM.

```
# modprobe kvm
```

You also have to modprobe kvm-intel or kvm-amd depending on the type of host.

KVM enters into linux kernel by wearing “module” form, thus adopt all the features of a linux kernel module. It invites a talk that, “I dont want to reboot entire system to reflect my new patch”. KVM uses the following modules to make Linux kernel as a hypervisor.

kvm.ko → Core KVM module

kvm-intel → Used on Intel CPUs

kvm-amd → Used on AMD CPUs

A normal Linux process has two modes of execution: kernel mode and user mode. KVM adds a third one: guest mode. When a guest process is executing non-I/O guest code, it will run in guest mode. All the virtual guests running under KVM are just regular linux processes on the host. You will be able to see a process called “qemu-kvm” for each virtual guest and for each virtual CPU for a single guest. This brings another advantage that , these process’ priorities can be increased or decreased using normal priority adjusting (nice/renice) commands, can set affinity and use ps, top, kill, etc. Also, control groups can be created and used to limit resources that each guest can consume on a host.

Each and every virtual CPU of your KVM guests are implemented using a Linux thread . The Linux scheduler is responsible for scheduling a virtual CPU, as it is a normal thread. In KVM , Guest physical memory is just a chunk of host virtual memory, that said, the guest physical memory has emulated within virtual memory of the “qemu-kvm” process. So it can be Swapped, Shared, Backed by large pages, Backed by a disk file, COW’ed and is also NUMA aware.

By “**virtio**” drivers KVM is closely competing with XEN to deliver native performance. virtio provides a common front end for device (network, block etc) emulations to standardize the interface. The front-end drivers for virtio is implemented in the guest operating system and the back-end drivers is implemented in the hypervisor. Each front-end driver has a corresponding back-end driver in the hypervisor.

Share the knowledge!

4

You may also like:

HOW TO INCREASE/DECREASE KVM GUEST MEMORY DYNAMICALLY?

How to increase/decrease kvm guest memory dynamically?

Ärzte sind
SCHOCKIERT: Vater
nimmt mit dieser Pille
in 4 Wochen 12kg ab...
GesundheitsNachrichten.co

ENGINEERING UNDERSTANDING QCOW2 IMAGE, ITS HEADER AND VERIFYING CORRUPTION.

Engineering.

Understanding Qcow2 image , its header and verifying corruption..

COMPLETE GUIDE TO EXTEND GUEST DISK FROM

Xen and Kvm

Complete guide to extend guest disk from backend, LVM and Filesystem inside KVM guest.

SMOOTH UPGRADE FROM FEDORA 18 TO

Smooth upgrade from fedora 18 to fedora 19 !

VIRT-MANAGER IS NOT ABLE TO DETECT

virt-manager is not able to detect the ISO file ?

Like Share Be the first of your friends to like this.



About admin

A person who know something about Assembly, C programming, python..etc..
Interested in linux kernel && virtualization .

[View all posts by admin →](#)

🔖 xen kvm virtualization. 📌 Bookmark.

◀ What is virtualization ? and different types of virtualization

device mapper multipath ▶

2 Comments



Anonymous

May 10, 2012 at 6:24 pm

Nice and Simple Explanation..

[← Reply](#)



Harry

June 22, 2012 at 1:37 pm

This post is awesome.. It really helped me to understand these technologies...

[← Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name*

Email*

Website

Comment

You may use these HTML tags and attributes: ` <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike> <pre>`

Post Comment

Whats your category ?

Select Category ▾

Alexa Rank

Site Info

humblec.com

Rank: 26,629,262

Links in: 22

Powered by  Alexa

Follow me!

Follow Be the first of your friends to follow [humblec.com](#).

Find us on Facebook



humblec.com

Like



humblec.com

November 4, 2014

Please vote for Dockit

@<https://www.docker.com/community/globalhackday/>

Dockit & Docker Global Hackathon.

Dockit was one of the project in Docker Global Hackathon held on Nov 1 2014. As part of hackathon submission, I have prepared 2 mins video of Dockit and its here. I would request you to cast your vote

WEBSITE-HUMBLEC.RHCLLOUD.COM

152 people like [humblec.com](#).



Facebook social plugin

Select Ur topic..

Select Category



Recent Comments

admin on [How to build a virt-p2v ISO in fedora \(F18/F17\) using virt-p2v-image-builder ?](#)

Francois dagorn on [How to build a virt-p2v ISO in fedora \(F18/F17\) using virt-p2v-image-builder ?](#)

nayeem on [Docker -> fedora -> gluster .. Run gluster containers using docker ... \[part 1\]](#)

humblec on [Dropwatch to see where the packets are dropped in kernel stack ..](#)

johonson on [Dropwatch to see where the packets are dropped in kernel stack ..](#)

Forums

[Linux Command Tips and Tricks.](#)

[C-programming..](#)

[Git \(distributed revision control and source code management system\) Tips /Tricks..](#)

[Linux Questions & Answers..](#)

[Python Programming](#)

My Area to explore..

Powered by [humblec](#) & [WordPress](#).



0685396