About Me

• Fedora Documentation Project
• Fedora Ambassador
• General Open Source Evangelist
• And, I work for Red Hat
  - Technical Account Manager
  - Dedicated Support Contact for Enterprise Customers
Goals for Today

• Look at the state of open source virtualization today
• Take an initial look at the way Xen works
• Get a feel for the tools that Red Hat has developed for RHEL 5 and Fedora Core 6
• Look at some screenshots and do a demo
Virtualization Overview

• Brief Intro

• Virtualization Hardware Notes (for those interested)

• Comparison of virtualization platforms

• State of Virtualization in Fedora 6 and RHEL 5
  - libvirt, virsh, virt-manager, gnome-applet-vm

• How to...
  - Install, clone, change virtual hardware, live migration

• Questions? Come to the booth! samfw@redhat.com
Brief Overview

• Virtualization = not a new idea

• Why virtualize?
  − Manage Cost...
  − Manage Risk...
  − It's neat

• Full virtualization vs. Para Virtualization
  − Performance vs. OS modification
Some Term Definitions

- **Node** = physical machine
- **Hypervisor/VMM** = Software that runs on the node, which allows for virtualization
- **Domain** = Operating System instance booted by the hypervisor
  - Domain 0 (Dom0) = Host OS
  - Domain U (DomU) = Guest OS
Virtualization Hardware Notes

- Intel: Virtualization Technology (VT)
  - Hardware support for virtualization
- Intel: Trusted Execution Technology (TXT)
  - Secure start up and IO security
- AMD: AMD Virtualization (AMD-V)
  - Hardware support + secure start up and IO security
Virtualization Hardware Notes

• New operating modes
  - Intel: VMX Root, VMX non-root
  - AMD: Host Mode, Guest Mode
  - Privilege levels/Rings 1-3 available to guests
  - HV runs in root/host mode at PL/Ring 0
  - HV controls access to memory and CPU
  - Dom0 controls access to other hardware drivers
X86 Privilege Levels
Virt Comparison

- Hypervisor vs. Linux Based
  - Hypervisor based (Xen)
    - Hypervisor controls basic hardware access
    - Dom0 is privileged, DomU's use Dom0 for hardware bits
  - Linux based (KVM)
    - Host OS controls hardware access
    - VMs are like processes
Virt Comparison

• Advantages of Hypervisor Based
  – The system is controlled by a small piece of software, small enough to be audited for security issues
  – The hypervisor takes up little virtual address space, which matters on 32 bit systems.
  – The same kernel can be run as host OS (dom0) and guest OS (domU)
  – The same hypervisor can be used with multiple operating systems.
    • Eg. use Linux or FreeBSD as the host OS, with the same virtualization software.
Virt Comparison

• Disadvantages of Hypervisor Based
  - Because the hypervisor relies on Dom0, any compromise to Dom0 is effectively a compromise to the hypervisor. Thus, the software needed to be audited for security is not so small.
  - Because the hypervisor is separate, it needs to duplicate a lot of Linux functionality (such as NUMA and CPU frequency scaling)
Virt Comparison

• Which is better?
  - Right now, Xen is the only open source virtualization platform ready for the enterprise and production use
  - Ultimately, which approach is “better” depends on your view point
  - The Red Hat/Fedora tools are agnostic to the underlying virtualization platform
Virtrualization in RHEL 5 and Fedora 6

• Xen 3.0.x
• New API developed for accessing virtualization platform
• Tools that use the new API developed to manage the virtualization platform
• Other Red Hat/Fedora applications enhanced to work with virtualization (Cluster, GFS, RHN)
Fedora 6 and RHEL 5 Features

- **libvirt**
  - Developed as a long-term, stable API for Xen *and* other virtualization platforms as they mature
  - C-toolkit to interact with the virtualization capabilities of the OS – right now just Xen, but could be any virtualization platform
  - Goal: “to provide the lowest possible generic and stable layer to manage domains on a node.”
  - Libvirt just does the basics – more advanced things can be done by applications that use libvirt
Fedora 6 and RHEL 5 Features

- **virsh**
  - Virtual Shell to allow command line access to features of a virtualization platform
  - Scriptable batch operations
  - Exposes libvirt functionality
  - Similar functionality as “xm” but not limited to Xen
  - Commands run asynchronously
  - Functionality continuously expanding

  - `virsh <subcommand> <domain-id> [OPTIONS]`
Fedora 6 and RHEL 5 Features

• Basic Functions
  - Start: “Create” a domain from a stopped state
  - Pause: Freeze a domain. When un-paused it will immediately return its previous state. Frees CPU cycles but retains memory footprint.
  - Save: Shutdown a domain, saving its memory state to a file. Completely frees guest resources for use by other domains. Easily restore to previous state.
  - Shutdown – gracefully shutdown guest
  - Destroy – hard reset of guest
Fedora 6 and RHEL 5 Features

• Basic Options
  - Full Virt or Para Virt
  - Disk: file, partition, logical volume, can add multiple
  - Memory: Can only allocate physical RAM
  - CPU: Can allocate more than physical CPUs
  - Network: Up to 3 virtual NICs per domU (can have more aliases, limitation should be lifted soon)
Fedora 6 and RHEL 5 Features

- **virt-manager**
  - GUI!
  - Does basically the same thing as virsh but in a GUI environment
  - Monitor domain statistics graphically in real time
  - Create new domains, save, pause, modify virtual hardware
  - serial console, graphical framebuffer
Fedora 6 and RHEL 5 Features

- **gnome-applet-vm**
  - Uses libvirt API
  - Gnome panel applet that lists all domains, their attributes, and their current status
  - Provides pop-up notification of changes to domain states
  - Useful for workstation deployments
How to...

• Install a domain
  - Guest OS's native installer is launched
    • Via virt-manager GUI wizard
    • Or via the virt-install command line tool (interactive or with options passed on command line)
  - Para-virt: needs install tree via NFS, FTP or HTTP
  - Full-virt: Needs ISO or install CDs
  - Kickstart fully supported
How to...

• Start a guest domain
  - `virsh` and `virt-manager` currently do not have this ability
    • `xend` 3.0.3 had no support for inactive domains, but this is added to 3.0.4
    • Currently must use `xm` (hardcoded)
    • Feature will be added soon to `virt-manager`, `virsh`  
      - `xm create [-c] <domain name>`
How to...

• Clone a domain
  - No built-in feature
  - Copy disk and configuration file
    • LVM snapshot is easiest and most graceful
      - `lvcreate -L <size> -s -n <newname> <current LV>`
    • `dd` can also be used
      - `dd if=<dom1> of=<dom2> bs=1k seek=2048k count=1`
  - Change name, disk, MAC and UUID in config file
    • `/etc/xen/<domain name>`
How to...

- Change hardware settings
  - Use virsh or virt-manager
    - `virsh setvcpus --count <number> <domID>`
    - `virsh setmem --bytes <bytes> <domID>`
    - Domain--->Details--->Hardware
  - Memory and VCPUS can be decreased or increased to maximum set in config file at boot time
  - VCPUs can be increased beyond the number of physical CPUs (useful for multithreaded applications)
How to...

• Migrate a domain
  - `xm migrate --live <domID> <new hypervisor>`

• New hypervisor must have sufficient system resources

• Both hypervisors must be on same subnet – IP and MAC migrate with the domain

• Disk does not move with domain...
  - Shared storage required (SAN, NAS, GNBD, iSCSI).
FAQ

• Can I run 32 and 64 bit guests on 64 bit hardware?
  - Yes, with full virt guests. For para virt guests, the hypervisor does not [YET] have a 32-bit compatibility layer.

• Why does Xen require PAE support on i686 hardware?
  - The hypervisor can be built either for PAE or non-PAE, but not both. It makes more sense to build for PAE. In the future, it will be possible to run both PAE and non-PAE guests on a PAE hypervisor.

• Can I run something other than RHEL and Fedora as DomU?
  - Yes, with full virt many other OS’s can be installed and run.
  - For para virt, the OS must be modified to work (Xen kernel). Many Linux distributions have paravirt support.
  - Note the difference between “works” and “supported”
More Stuff to Learn

• Provision guest domains with Cobbler and Koan:
  − Set up a complete PXE provisioning tool for deploying virtual guests
    - http://cobbler.et.redhat.com/

• Set up GFS and/or Cluster Suite with guest domains:
  − Add an LV to both domains for shared storage
  − Could have 3 3-node clusters on 3 nodes
Demo: Guest Installation
Creating a new virtual system

This assistant will guide you through creating a new virtual system. You will be asked for some information about the virtual system you'd like to create, such as:

- A name for your new virtual system
- Whether the system will be fully virtualized or para-virtualized
- The location of the files necessary for installing an operating system on the virtual system
- Storage details - which disk partitions or files the system should use
- Memory and CPU allocation

Naming your virtual system

Please choose a name for your virtual system:

System Name: web-rarm3

Example: system1
Choosing a virtualization method

You will need to choose a virtualization method for your new system:

- **Paravirtualized:**
  Lightweight method of virtualizing machines. Limits operating system choices because the OS must be specially modified to support paravirtualization. Better performance than fully virtualized systems.

- **Fully Virtualized:**
  Involves hardware simulation, allowing for a greater range of operating systems (does not require OS modification). Slower than paravirtualized systems.

Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **fully virtualized** virtual system:

- **ISO Image Location:**
  ISO Location: `/root/boot-rhel4as-x86_64` (Browse...)

- **CD-ROM or DVD:**
  Please insert disc 1 of the installation media for the desired operating system now.

Install Media URL: **http://download.fedora.redhat.com/pub/fedora/core/linux/5/x86**

Example: nfs://servername.example.com/rhel-5.0/x86/ks

Kickstart URL: 

Example: `ftp://hostname.example.com/ks/rhel5.0-vs-x86.ks`
Assigning storage space

Please indicate how you’d like to assign space on this physical host system for your new virtual system. This space will be used to install the virtual system’s operating system.

- Normal Disk Partition:
  - Partition: 
  - Example: /dev/hdc2

Simple File:
- File Location: /root/test1
- File Size: 5000 MB

**Note:** File size parameter is only relevant for new files

**Warning:** If you do not allocate the entire disk at VM creation, space will be allocated as needed while the guest is running. If sufficient free space is not available on the host, this may result in data corruption on the guest.

**Tip:** You may add additional storage, including network-mounted storage, to your virtual system after it has been created using the same tools you would on a physical system.

Allocate memory and CPU

**Memory:**
- Please enter the memory configuration for this VM. You can specify the maximum amount of memory the VM should be able to use, and optionally a lower amount to grab on startup. Warning: setting VM memory too high will cause out-of-memory errors in your host domain!

  - Total memory on host machine: 1022.00 MB
  - VM Max Memory (MB): 500
  - VM Startup Memory (MB): 500

**CPUs:**
- Please enter the number of virtual CPUs this VM should start up with.

  - Logical host CPUs: 1
  - VCPUs: 1

**Tip:** For best performance, the number of virtual CPUs should be less than (or equal to) the number of logical CPUs on the host system.
Demo: virt-manager
## Virtual Machine Manager (Xen: woodie.lab.boston.redhat.com) (on woodie.lab.boston.redhat.com)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Status</th>
<th>CPU usage</th>
<th>VCPUs</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Domain-0</td>
<td>Running</td>
<td>3.91 %</td>
<td>8</td>
<td>1022.99 MB (6.24%)</td>
</tr>
<tr>
<td>29</td>
<td>rhel4u4vperf-x86_64</td>
<td>Running</td>
<td>0.87 %</td>
<td>1</td>
<td>1.01 GB (6.34%)</td>
</tr>
<tr>
<td>26</td>
<td>rhel5FVtoPV</td>
<td>Running</td>
<td>0.31 %</td>
<td>1</td>
<td>1.01 GB (6.34%)</td>
</tr>
<tr>
<td>27</td>
<td>rhel5FVtoPV2</td>
<td>Running</td>
<td>0.18 %</td>
<td>1</td>
<td>1.01 GB (6.34%)</td>
</tr>
<tr>
<td>15</td>
<td>rhel5rc1perf01</td>
<td>Running</td>
<td>0.06 %</td>
<td>8</td>
<td>1023.78 MB (6.25%)</td>
</tr>
<tr>
<td>28</td>
<td>rhel5rc1perf02</td>
<td>Running</td>
<td>0.06 %</td>
<td>8</td>
<td>1023.78 MB (6.25%)</td>
</tr>
<tr>
<td>30</td>
<td>windows2003sp1</td>
<td>Running</td>
<td>0.77 %</td>
<td>1</td>
<td>1.01 GB (6.34%)</td>
</tr>
</tbody>
</table>
Demo: virt-manager console

Fedora Core release 5 (Bordeaux)
Kernel 2.6.15-1.2054_FC5xenU on an x86_64

```
chcp-4-243 login: root
Password:
Last login: Tue Aug 22 16:01:49 on tty1
[root@chcp-4-243 ~]# uptime
16:01:47 up 3 days, 22:54, 1 user, load average: 0.00, 0.02, 0.03
[root@chcp-4-243 ~]# free
            total       used       free     shared   buffers/cache
            313548      295228      18320
            65456      248192        0
[root@chcp-4-243 ~]# uname -a
Linux chcp-4-243.virt.boston.redhat.com 2.6.15-1.2054_FC5xenU 16:18:03 EST 2006 x86_64 x86_64 x86_64 GNU/Linux
```

Red Hat Enterprise Linux AS release 3 (Taroon Update 1)
Kernel 2.4.21-9.EL on an i686

```
localhost login: _
```
Demo: pausing

Activating ISA DMA hang workarounds.
pci hotplug: PCI Hot Plug PCI Core version: 0.5
ACPI: Processor [CPU0] (supports C1)
Real Time Clock Driver v1.12
Linux agpgart interface v0.1.000 (c) Dave Jones
serio: i8042 AUX port at 0x60,0x64 irq 12
serio: i8042 KBD port at 0x60,0x64 irq 1
Serial: 16550a driver Revision: 1.99 A-2 ports, I/O sharing enabled
ttyS0 at /dev/ttyS0 
HMAIDSK driver initialized, 90678 blocks, block size 512 bytes
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
PIIX3: IDE controller at PCI slot 0000:00:01.1
PIIX3: chipset revision 0
PIIX3: not 100% native mode: will probe irqs later
   ide0: DM-DMA at 0xc000-0xc007, BIOS settings: hda:pio, hdb:pio
   ide1: DM-DMA at 0xc000-0xc00f, BIOS settings: hdc:pio, hdd:pio
hda: QEMU HARDDISK, ATA DISK drive
Using cfg io scheduler
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hdc: QEMU CD-ROM, ATAPI CD/DVD-ROM drive
ide1 at 0x170-0x177,0x376 on irq 15
hda: max request size: 1024KiB
hda: 4194304 sectors (2147 MB) w/256KiB Cache, CHS=416/255/63, (U)DMA
Sources and Further Reading

- Virtualization Wiki:
  - http://virt.kernelnewbies.org

- Fedora Documentation:
  - http://fedoraproject.org/wiki/CategoryVirtualization

- Xen Manual

- Mailing List:
  - http://www.redhat.com/mailman/listinfo/fedora-xen
Questions?