oVirt: An Open Management Framework for Virtualized Environments
Hugh Brock & Perry Myers
Red Hat
Agenda

- Introduction to oVirt
- oVirt Architecture
- Demo
- Features Roadmap
- Getting Involved
- Q & A
What is oVirt

- Represents next generation of open source virtualization
- **Project** from Red Hat's Emerging Technology Group

- Built on open source and open standards
- Built within the community

- Provides the foundation for Red Hat's next generation of virtualization **Products**
What is oVirt

- oVirt project delivers complete virtualization solution
  - Hypervisor
    - Built on Linux kernel
    - Managed using standards based tools
      - Based on libvirt, CIM and LDAP
      - From Red Hat, the community or ISVs
  - Virtualization Management Platform
    - Cross platform – multiple hypervisors
    - Built on open standards
      - LDAP, Kerberos, Libvirt, etc
    - Scalable
      - From 1 node to tens of thousands of nodes
      - From small host cluster to cloud computing infrastructure
  - Secure
    - Integrated policy and audit framework
Why do we need oVirt

- Current generation of solutions built on proprietary architectures and protocols
  - Lack of standards
  - Vendor lock in
    - Hypervisor and management platform intrinsically linked
      - Locked into management solution from hypervisor vendor
  - Lack of interoperability
    - Different management platform required for each hypervisor
  - Not integrated into enterprise
    - Separate tools for physical systems and each hypervisor
  - No centralized authentication, authorization or audit
    - Security concerns limiting deployment of virtualization solutions
Libvirt : Management based on open standards

- Provides a standard management interface
  - Hypervisor agnostic
    - Will work with multiple hypervisors
  - Stable API
    - Shield users from hypervisor changes
  - Consistent tools across hypervisors
    - eg. Same interface for Xen, QEMU, KVM, OpenVZ, LXC, LDoms, etc
  - Scriptable
    - Provides APIs for developers / tool vendors
- Secure
  - Encryption and authentication GSSAPI/SASL2
  - Allows vendors to build cross platform tools
Libvirt: Management based on open standards
What makes up oVirt and how does it all fit together?

- Two major components:
  - oVirt Managed Node
  - oVirt Server Suite
Managed Nodes

- Small footprint embeddable hypervisor
- Based on Linux kernel with KVM Hypervisor
- Requires processors with hardware virtualization
- Runs both Windows and Linux guests
Server Suite

- Administration Web Interface
- Authorization, Authentication and Audit
- Task Queuing
- Status Monitoring
- Performance Monitoring and Visualization
Storage Nodes

- Provides External Storage for Guests
  - NFS – File Based Storage
  - iSCSI
  - Fibre Channel
  - Local Disk
  - Logical Volume Management (LVM)
Integrated with well known Open Source projects:

- Linux kernel
- KVM – Hypervisor
- libvirt – Virtual Machine management
- FreeIPA – Authentication/Authorization
- Cobbler/Koan – Provisioning
- collectd – Performance Data Collection
oVirt Architecture Details

- oVirt Managed Node
- oVirt Server Suite
Managed Node Architecture
Managed Node – What is it?

- Lightweight, small-footprint Fedora 9 install
  - Today less than 64MB Image Size
    - Includes drivers for all hardware supported by Fedora
    - Smaller Images possible for specific hardware spins
      - Targeting 32-48MB

- Stateless

- KVM for Hypervisor

- libvirt for virtual machine management

- collectd for performance data collection
Managed Node – How is it made?

- Fedora 9 standard repositories
- oVirt specific RPMs hosted at http://ovirt.org
- Livecd-tools
  - Customized Kickstart
  - Image Creation/Packaging
  - Image minimization
- Will use the Appliance OS Toolkit to build the Managed Node
Managed Node – How is it deployed?

- Can be deployed in several different ways:
  - Embedded/External Flash
    - eg. SD cards and USB keys
  - Network Boot via PXE
  - Local CD-ROM
  - Installed on local hard drive

- Can be preloaded on OEM hardware
Managed Node – Boot Process

■ Stateless
■ Registers with oVirt Server
■ Authentication Credentials
  • Pulled dynamically from server
  • Locally attached storage
  • Trusted Platform Module (TPM)
■ Hardware Enumeration
■ oVirt Daemons
  • libvirtd
  • Status and Performance Monitoring
Managed Node Architecture
Server Suite Architecture
- oVirt preferred authorization and authentication package is FreeIPA Kerberos + LDAP
- Any standard Kerberos server will work
- Any standard LDAP server will work
Taskomatic pulls tasks off queue, makes libvirt calls, updates oVirt db with results

Host-browser waits for hosts to come online, updates oVirt db with host info

Host-status monitors hosts, updates oVirt db
All three daemons communicate with oVirt managed nodes via libvirt calls

Default oVirt transport is libvirt + SASL/GSSAPI

Larger scale installations will use MRG messaging instead
- AMQP Messaging
- Collectd + libvirt
- Round Robin Database (RRD) accumulates statistics
- oVirt Stats API provides statistics to UI
- Stats API extensible for SNMP, other performance data stores
Server Suite: UI

- Usability foremost
- Virtual infrastructure control to users
- Admins retain hardware control
- Rails allows rapid development
- Standard transactional RDBMS
Server Suite Architecture
Deployment Types

- Developer Appliance
- Bundled Appliance
- Production/Enterprise Installation
# oVirt: Management Interface

## Hosts

<table>
<thead>
<tr>
<th>Hostname</th>
<th>UUID</th>
<th>Hypervisor</th>
<th>CPUs</th>
<th>Speed (MHz)</th>
<th>Arch</th>
<th>RAM (MB)</th>
<th>Status</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>node138.priv.ovirt.org</td>
<td>node138.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node140.priv.ovirt.org</td>
<td>node140.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node141.priv.ovirt.org</td>
<td>node141.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node142.priv.ovirt.org</td>
<td>node142.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node143.priv.ovirt.org</td>
<td>node143.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node144.priv.ovirt.org</td>
<td>node144.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node145.priv.ovirt.org</td>
<td>node145.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node146.priv.ovirt.org</td>
<td>node146.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node147.priv.ovirt.org</td>
<td>node147.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node150.priv.ovirt.org</td>
<td>node150.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>unavailable (enabled)</td>
<td></td>
</tr>
<tr>
<td>node5.priv.ovirt.org</td>
<td>node5.priv.ovirt.org</td>
<td>QEMU</td>
<td>1</td>
<td>1995</td>
<td>x86_64</td>
<td>498</td>
<td>available (enabled)</td>
<td></td>
</tr>
</tbody>
</table>

**node5.priv.ovirt.org**

- **UUID**: node5.priv.ovirt.org
- **CPUs**: 1
- **Speed**: 1995 MHz
- **Memory**: 498 MB
- **Architecture**: x86_64
- **Hypervisor**: QEMU
- **Status**: available (enabled)

## Metrics

- **Overall Load**: 0
- **CPU**: 88%
- **Memory**: 34%
- **Network In**: 16 mb/s
- **Network Out**: 50 mb/s
Demo

- Show how the Management UI can be used to:
  - Create new Hardware/Virtual Machine Pools
  - Attach Storage
  - Create and Provision new Virtual Machines
- Show Managed Node Boot
Features Roadmap – Managed Node

- Standalone Node Operation
  - Local Management Console
  - No oVirt Server Needed
  - Local management with oVirt Server

- Additional Hypervisor Support
  - Xen
  - Other HVs as libvirt provides support

- xenner – Run Xen paravirtualized guests

- Clustering/High Availability Support

- Paravirtualized Drivers Support
Features Roadmap - Infrastructure

- Provisioning Enhancements
  - Tighter Integration with Cobbler
  - Support for Windows Provisioning
- Distributed Architecture
- Messaging Support
- Easy Integration with alternate Servers
  - Databases
  - Kerberos Servers
  - Directory Servers
Features Roadmap - Management

- Scriptable APIs for oVirt Server and Managed Node
- Flexible Rules Engine
- Finer Grained Permissions/Roles
- Guest Migration
  - offline/live
  - manual/automated
  - Intelligent migration
- Guest Performance Monitoring
- SLA Definitions
Getting Involved

- Active developer community at: http://www.ovirt.org
  - Downloads
  - Documentation
  - Features Roadmap
  - Wiki
- Mailing list: ovirt-devel@redhat.com
- IRC Channel: #ovirt on freenode.net
- Source Repository:
  
git clone git://git.et.redhat.com/ovirt
Download, Use, Contribute!

- Prebuilt developer appliance available
  - Host Hardware – Hardware virtualization support
  - Host OS – Fedora 9 x86_64
  - Download, extract and run

- Build your own appliance
  - Check out git repository
  - See Install Instructions
  - Build appliances for both i386 & x86_64 platforms
  - Build both bundled and developer style appliances

- Contribute patches to ovirt-devel mailing list
Questions & Answers
Contact Us!

- http://www.ovirt.org
- ovirt-devel@redhat.com
- #ovirt on freenode.net
- Hugh Brock - hbrock@redhat.com
- Perry Myers - pmyers@redhat.com