oVirt Multiplatform

KVM Forum

October 23rd, 2013

Edinburgh, UK

Speaker

Leonardo Bianconi

- Software engineer at Eldorado Research Institute.
- Eldorado is a not for profit organization located in Brazil, focused on technology development.

This presentation

Bringing multiplatform management capability to oVirt, initially x86 and PPC64.

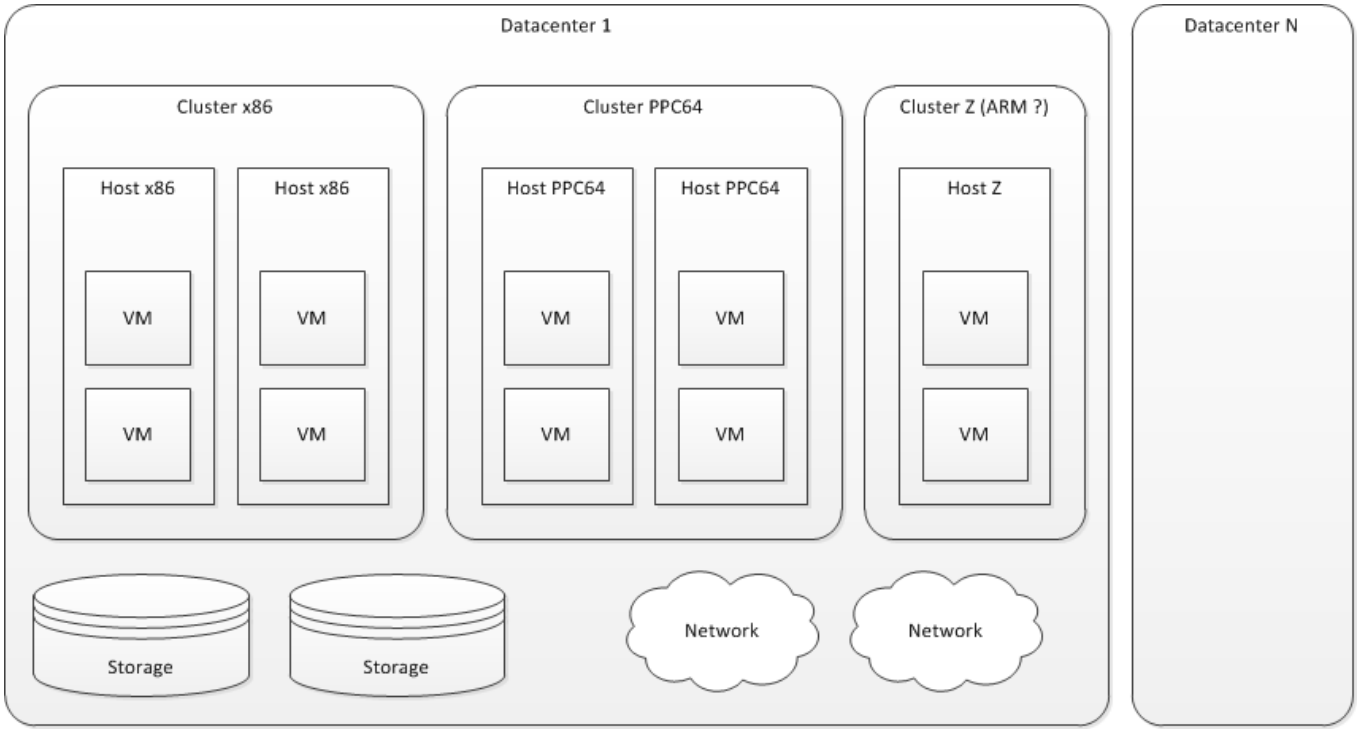
Why?

- KVM on POWER systems announcement during the Red Hat Summit in Boston (June 2013).
  http://www-03.ibm.com/press/us/en/pressrelease/41255.wss
- The OpenPOWER consortium announcement (August 2013).
  http://www-03.ibm.com/press/us/en/pressrelease/41684.wss

Agenda

- The idea.
- What has already been done?
- What is on the backlog?
- Where is the code?
- Conclusion.

The idea



Goal: Adding multiplatform awareness with minimal changes in UI, architecture and code.

The problem

1. The software was designed and developed for single platform.

Enumerators with no platform specification:

- Network types.
- Display devices.
- CD interfaces.
- Disk interfaces.

The problem

For example, the network interfaces:

```
package org.ovirt.engine.core.common.businessentities.network;
public enum VmInterfaceType {
   rtl8139_pv(0,"Dual mode rtl8139, Red Hat VirtIO"),
   rtl8139(1,"rtl8139"),
   e1000(2,"e1000"),
   pv(3,"Red Hat VirtIO");
}
```

The interface “e1000” is not supported by PPC64 architecture, so it need to be filtered by architecture.

The problem

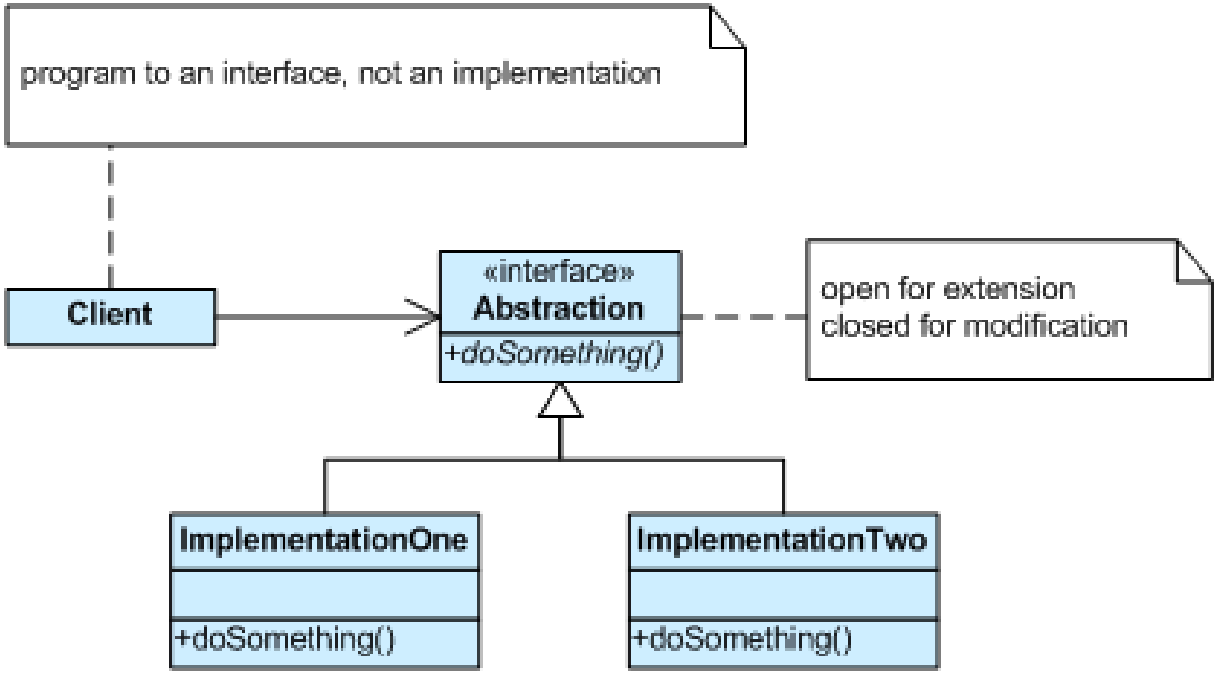
2. Assumptions based on specific architecture.

Example: Addressing CD is different for PPC64 and may be different for other architectures.

The code must switch behaviour on this case for each architecture.

Proposal

Refactoring oVirt using Strategy design pattern to be able to add support for other architectures.



Benefits

- Selects a behavior at runtime.
- Defines a family of algorithms - encapsulates each one.
- Avoids "if" to switch on architecture behavior.
- Easy identification of architecture specific code.
- Easy way to add another architecture and new architecture specific functionality.

Agenda

- The idea.
- What has already been done?
- What is on the backlog?
- Where is the code?
- Conclusion.

What has already been done?

- Moved x86_64 specific code.
- Application parameterizations.
- PPC64 code specific development.

Moving x86_64 specific code

- Architecture field.
- Initial development of Strategy Design Pattern.
- All the x86_64 specific code was encapsulated in a Strategy.

Parameterization w/ config files

OSInfo configuration file:

- Settings are defined per OS and per architecture.

Benefits: Flexibility

- Assignment of Lan/Video/Disk/CD for each OS.
- Filter items in the frontend.
- Compatibility check.
- Minimizes architecture specific code.

Parameterization w/ config files

```
# Other OS type to PPC64 Architecture
os.other_ppc64.id.value = 1001
os.other_ppc64.name.value = Other OS
os.other_ppc64.derivedFrom.value = other
os.other_ppc64.cpuArchitecture.value = ppc64
os.other_ppc64.bus.value = 64
os.other_ppc64.cdInterface.value = scsi
os.other_ppc64.devices.audio.value = ich6
os.other_ppc64.devices.network.value = rtl8139, pv
os.other_ppc64.devices.diskInterfaces.value = VirtIO, VirtIO_SCSI
os.other_ppc64.displayProtocols.value = qxl/qxl
os.other_ppc64.devices.network.hotplugSupport.value = true
os.other_ppc64.devices.network.hotplugSupport.value.3.0 = false
```

PPC64 code specific development

Engine:

- Addressing Disk.
- Addressing CD.

VDSM:

- Topology.
- Processor name.
- Hardware information.

Strategy design pattern - Before

```
protected void buildVmDrives() {...
  case VirtIO_SCSI:
    struct.put(VdsProperties.INTERFACE, VdsProperties.Scsi);
    if (disk.getDiskStorageType() == DiskStorageType.LUN) {
      struct.put(VdsProperties.Device, VmDeviceType.LUN.getName());
      struct.put(VdsProperties.Sgio, disk.getSgio().toString().toLowerCase());
    }...
}
```

File: VmInfoBuilder.java
Package: org.ovirt.engine.core.vdsbroker.vdsbroker

Strategy design pattern - After

```
protected void buildVmDrives() {...
  case VirtIO_SCSI:
    struct.put(VdsProperties.INTERFACE, VdsProperties.Scsi);
    if (disk.getDiskStorageType() == DiskStorageType.LUN) {
      struct.put(VdsProperties.Device, VmDeviceType.LUN.getName());
      struct.put(VdsProperties.Sgio, disk.getSgio().toString().toLowerCase());
    }
    if (StringUtils.isEmpty(vmDevice.getAddress())) {
      ArchStrategyFactory.getStrategy(vm.getArchitecture()).run(new AssignSCSIAddress(struct, maxUsedLunByController,
disk.getDiskInterface()));
  }
  break;
  case SPAPR_VSCSI:
    struct.put(VdsProperties.INTERFACE, VdsProperties.Scsi);
    if (StringUtils.isEmpty(vmDevice.getAddress())) {
      ArchStrategyFactory.getStrategy(vm.getArchitecture()).run(new AssignSCSIAddress(struct, maxUsedLunByController,
disk.getDiskInterface())); } break;...
    }
```

File: VmInfoBuilder.java
Package: org.ovirt.engine.core.vdsbroker.vdsbroker

Running code results

XML for x86_64:

```
<disk device="disk" snapshot="no" type="file">
  <address bus="0" controller="0" target="0" type="drive" unit="1"/>
  <source file="disk.img"/>
  <target bus="scsi" dev="sda"/>
  <serial>5a6b4589-6bef-49ac-b009-3a56a467eccf</serial>
  <driver cache="none" error_policy="stop" io="threads" name="qemu" type="raw"/>
</disk>
```

XML for PPC64:

```
<disk device="disk" snapshot="no" type="file">
  <address bus="0" controller="1" target="0" type="drive" unit="1"/>
  <source file="disk.img"/>
  <target bus="scsi" dev="sda"/>
  <serial>f9c146f0-9cdb-4830-94c6-fa0c19772229</serial>
  <driver cache="none" error_policy="stop" io="threads" name="qemu" type="raw"/>
</disk>
```

Visitor design pattern on subprojects

Strategy design pattern classes must be seen by all subprojects. Problem:

- Strategy needs subproject specific classes that cannot be seen by the common subproject.

Visitor design pattern:

- Strategy receives an object and runs the architecture specific code.
- Visitor class is located in the subproject.
- Easy to add new architecture specific code.

Visitor design pattern

Interface:

```
public interface ArchCommand {
  void runForX86_64();
  void runForPPC64();
}
```

File: ArchCommand.java

Package: org.ovirt.engine.core.common.archstrategy

Visitor design pattern

Implementation:

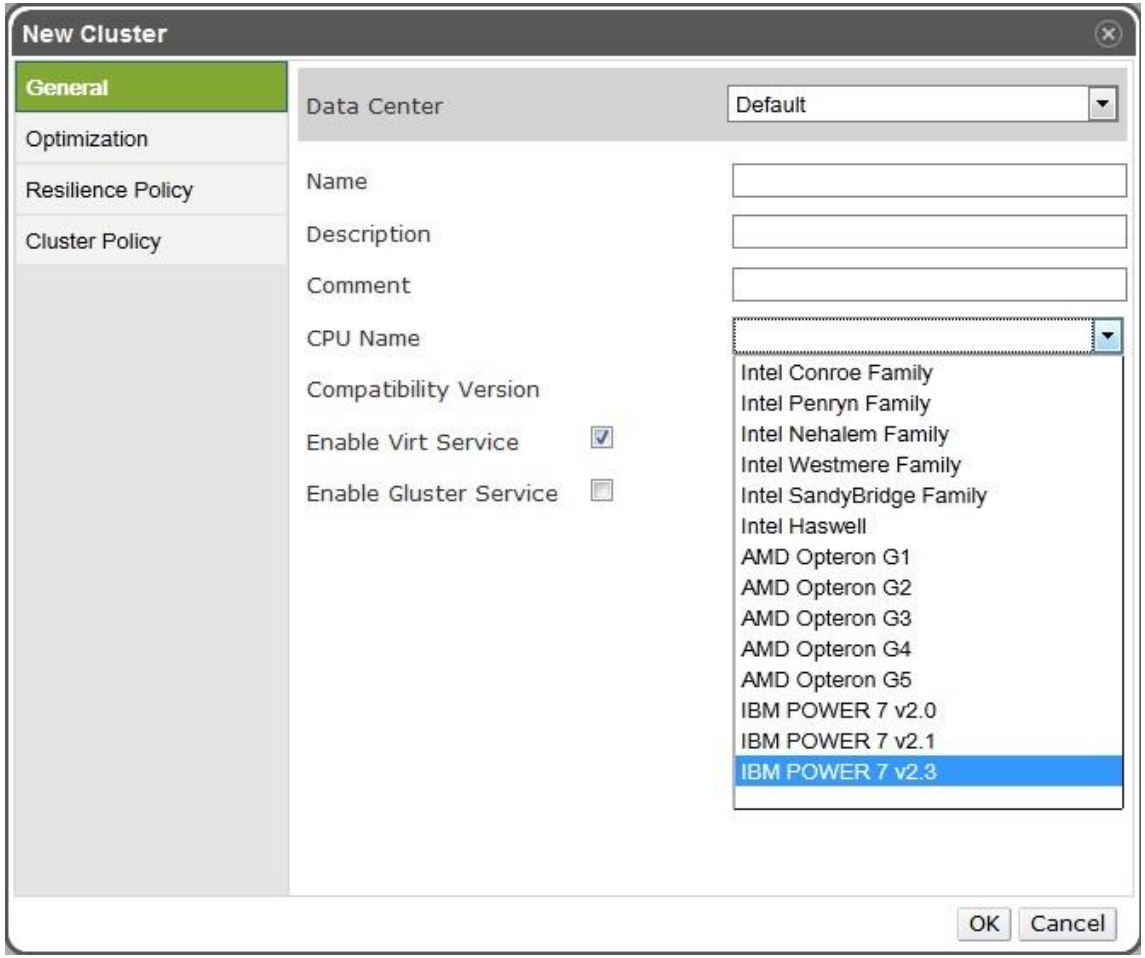
```
public class AssignSCSIAddress implements ArchCommand {...
  @Override
  public void runForX86_64() {
    // In the x86_64 there is only one VirtIO-SCSI controller present.
    // The default address given by libvirt works fine
  }
  @Override
  public void runForPPC64() {
    if (diskInterface == DiskInterface.VirtIO_SCSI) {
      SCSIAddressingUtils.dynamicAddressing(device, maxUsedLunByController, 1);
    } else if (diskInterface == DiskInterface.SPAPR_VSCSI) {
      SCSIAddressingUtils.dynamicAddressing(device, maxUsedLunByController, 0);
    }
  }
}
```

File: AssignSCSIAddress.java
Package: org.ovirt.engine.core.vdsbroker.architecture

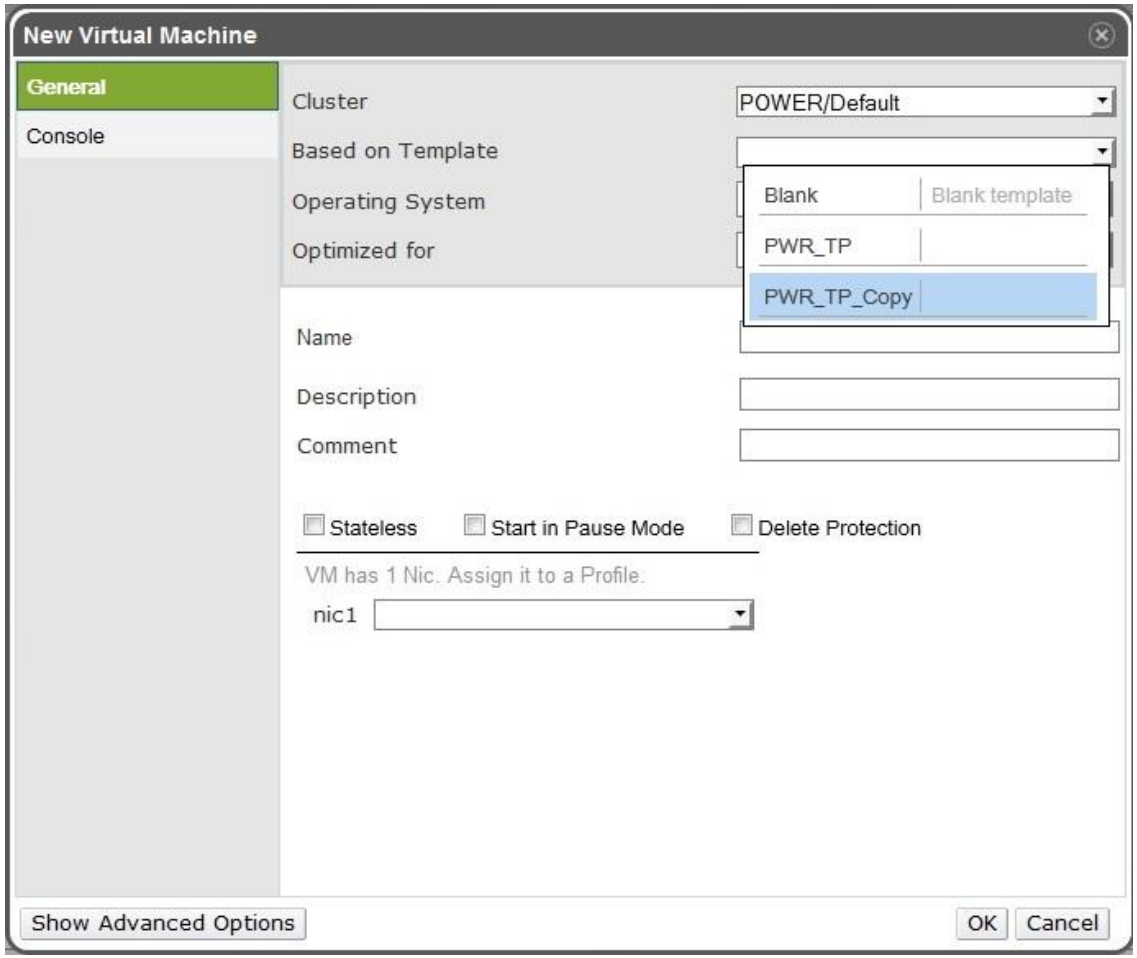
Which features are ready?

- Create Clusters, VMs, Templates and Pools.
- Import/Export VMs and Templates.
- Attach disks to VMs.
- Search VMs by architecture.
- Manage VMs.

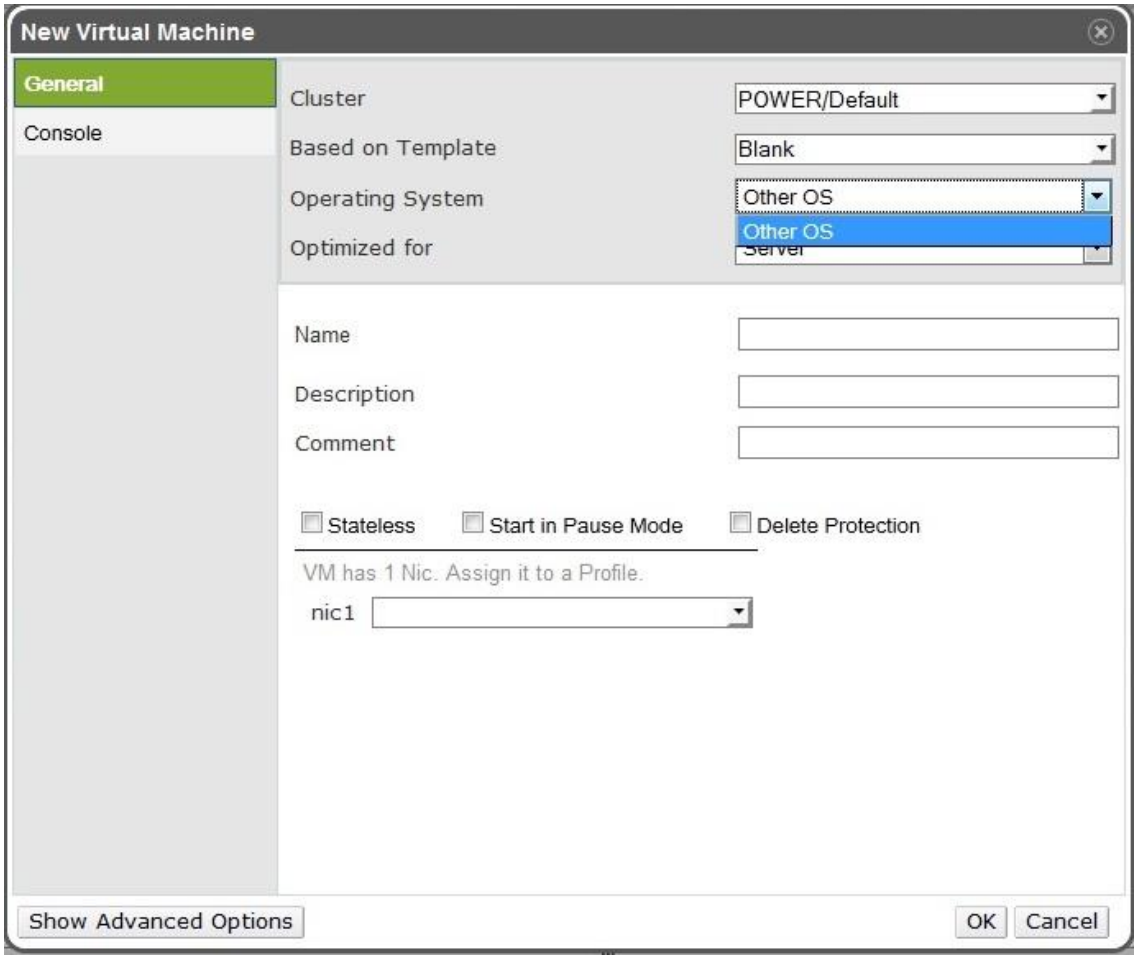
Screenshots



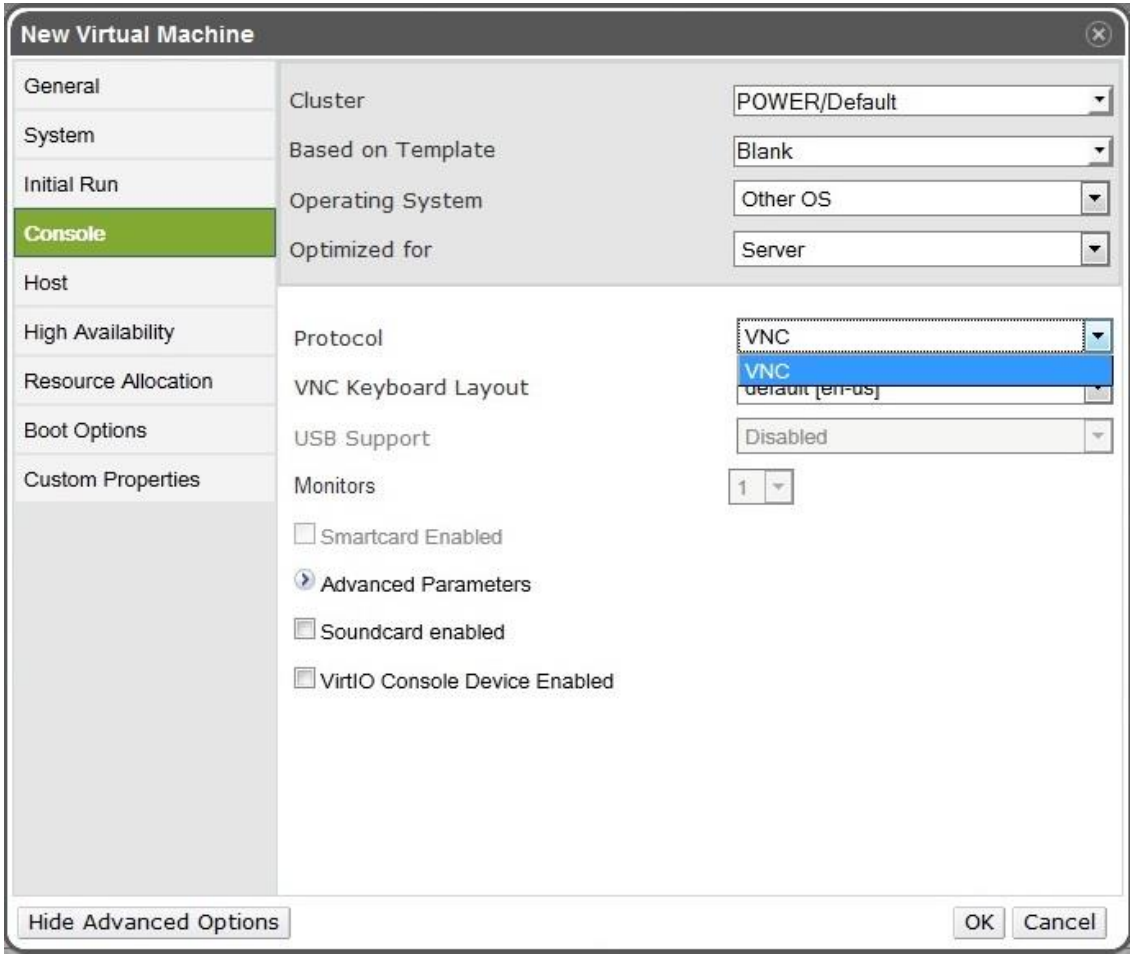
Screenshots



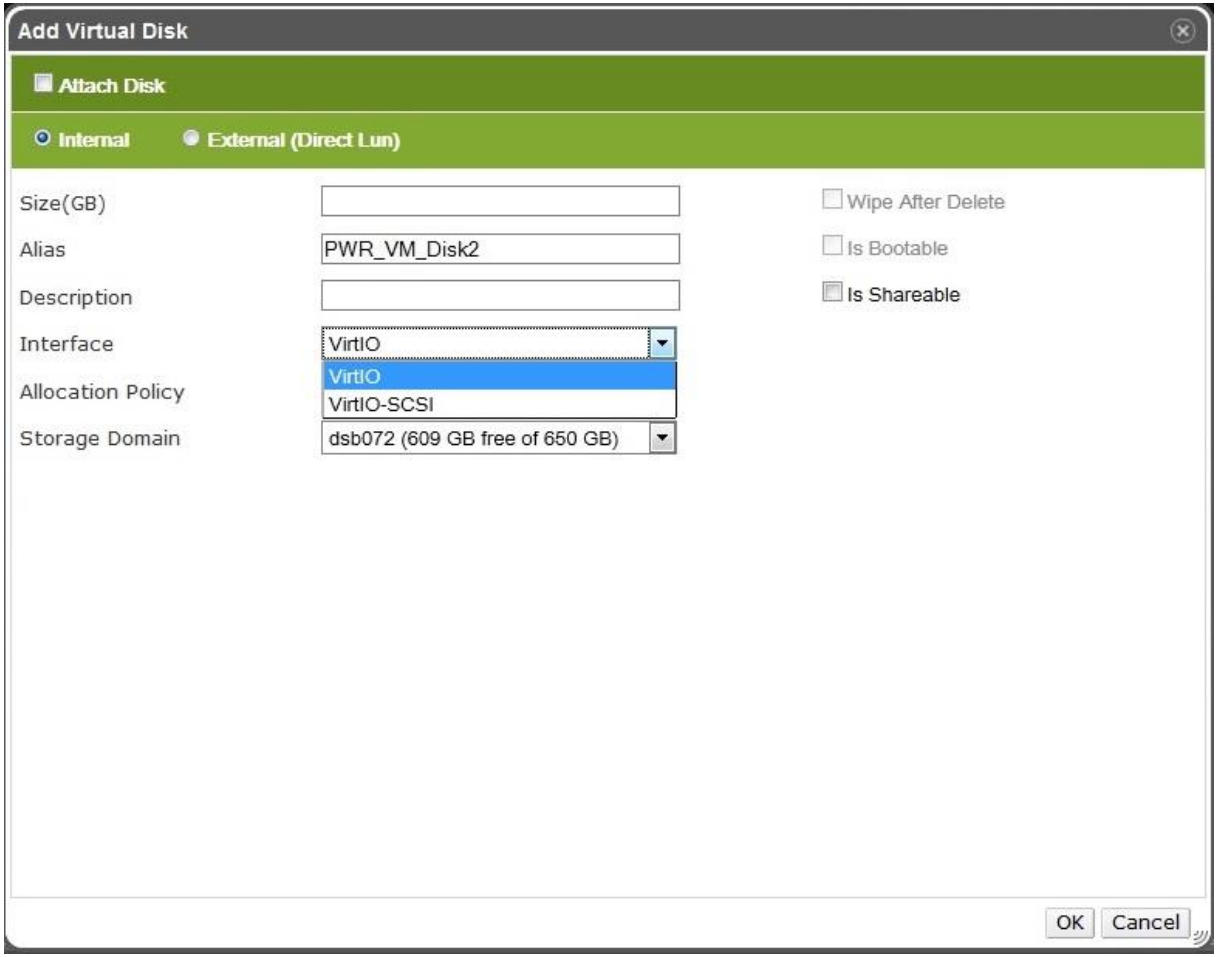
Screenshots



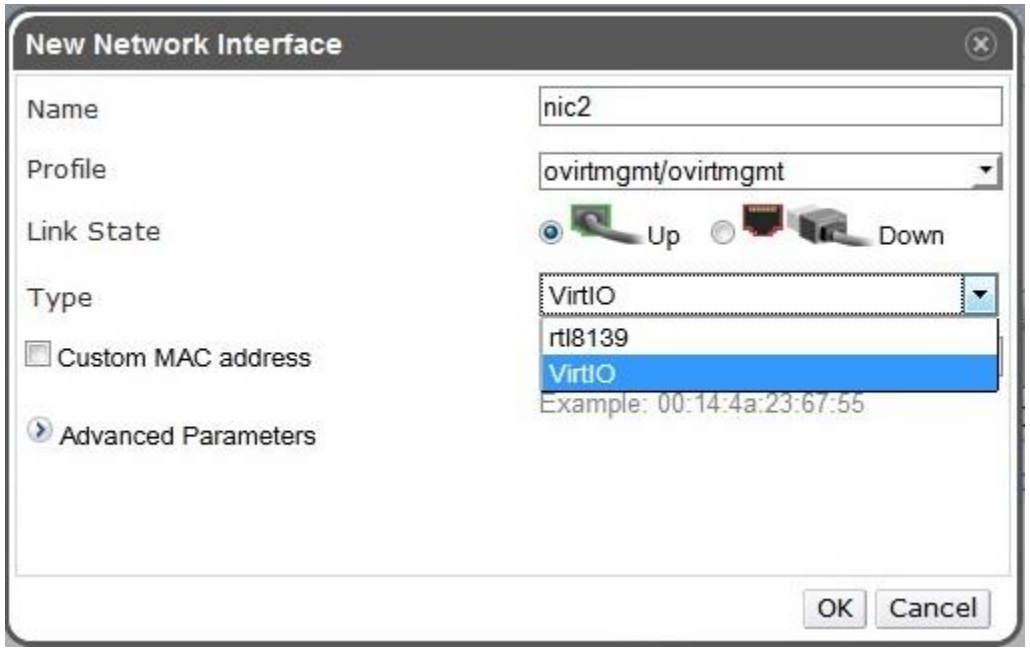
Screenshots



Screenshots



Screenshots



Agenda

- The idea.
- What has already been done?
- What is on the backlog?
- Where is the code?
- Conclusion.

Missing features

- Install guest OS.
- Network booting.
- Migration.
- Snapshotting.
- Hotplugging.

Development backlog

- Provide SPAPR VLAN and SPAPR VSCSI (PPC64 specific).
- Do not allow change CPU version of PPC64 clusters.
- Specify which disk interfaces can perform hotplugging.
- Provide network boot for PPC64.

Agenda

- The idea.
- What has already been done?
- What is on the backlog?
- Where is the code?
- Conclusion.

Where is the code?

Gerrit:

http://gerrit.ovirt.org

Git:

http://www.ovirt.org/Subprojects

git://gerrit.ovirt.org/ovirt-engine

git://gerrit.ovirt.org/vdsm

git://gerrit.ovirt.org/ovirt-host-deploy

Patches - Engine

17853 - core: Add POWER 7 to the CPU list

18938 - core, engine, webadmin: Initial support for alternative architectures

18220 - core: New OS for IBM POWER support

17972 - webadmin: Show only compatible OSes

18347 - engine: OS type validation

18702 - core: Fill and check arch when importing VM and Template

19012 - ui: OVF import in multiple architecture scenario

18226 - core, engine, webadmin: Cluster and architecture related changes

18227 - core, webadmin, engine: Added arch support for VM & Template

19487 - ui: Avoid the selection of incompatible templates

19132 - ui, core: Prevent architecture mismatches in the frontend

18622 - core, engine: SCSI CD-ROM on PPC64 VMs

19010 - core, engine: Architecture parameter on search backend

Patches - Engine

17964 - core, webadmin: Show only supported disk interfaces

- 18648 - engine: Disk interface validation

19188 - core: Vnic hotplug validation - Patch 1 of 2

- 19189 - webadmin: Vnic hotplug validation - Patch 2 of 2
  - 19601 - core: Disk hotplug validation - Patch 1 of 2
    - 19628 - webadmin: Disk hotplug validation - Patch 2 of 2
      - 19758 - core: Cleanup of Vnic and Disk hotplug

18677 - engine: VM Device Type for Display Type

- 17885 - webadmin: Show only supported displays
- 18150 - engine: Display type validation

18042 - engine: Vnic interface validation (Merged)

17423 - deployUtil: Remove null character from the id on IBM POWER (Merged)

19878 - core, engine: sPAPR VLAN support

Patches - VDSM

19395 - vdsm: Hardware information about POWER hosts

17437 – vdsm: Capabilities: List capabilities of the IBM POWER family

19875 - vdsm: Handling topology for ppc64

19396 - vdsm: Report fake capabilities

18718 - vdsm: Create VMs for the POWER architecture

17279 - vdsm: hardware: Remove null character from the id on IBM POWER (Merged)

All-in-one patch

We merged all those patches and published the code in one single DEMO branch:

Engine

https://bitbucket.org/gustavo_temple/ovirtenginemultiplatform

VDSM

https://bitbucket.org/gustavo_temple/ovirtvdsmmultiplatform

How to build and test?

Instructions in the wiki:

- http://www.ovirt.org/Features/Engine_support_for_PPC64#DEMO_version
- http://www.ovirt.org/Features/Vdsm_for_PPC64#DEMO_Version

Agenda

- The idea.
- What has already been done?
- What is on the backlog?
- Where is the code?
- Conclusion.

Conclusion

After the acceptance of all patches developed, the oVirt engine will became multiplatform, initially for x86_64 and PPC64, with all code structure to add other architectures.

oVirt for PPC64

Questions?

Get involved

Website

http://www.ovirt.org/Community

Wiki

http://www.ovirt.org/Features/Engine_support_for_PPC64

http://www.ovirt.org/Features/Vdsm_for_PPC64

My mail

leonardo.bianconi@eldorado.org.br

Thank you!