IO Performance Improvement on Virtualized Environment

Masroor V P (masroor.vpm@wipro.com)
Anil Gurumurthy (anil.murthy@wipro.com)
Introduction

- IOV is designed to remove the virtualization “tax” on QoS and performance, in order to start moving I/O intensive applications from dedicated servers to VMs.
  - SR IOV is supported in Linux and RHEL 5.4
  - VMDirectPath is supported in ESX 4.0
  - IOV compliant adapters from Neterion and other vendors are shipping
- “multi-core” IOV compliant devices provide I/O intensive VMs, protocols and applications with dedicated direct I/O
  - Performance is comparable to a native OS
  - QoS is comparable with I/O on dedicated servers
  - Migration and hypervisor control is preserved
  - Cost is comparable with non-differentiated NICs and HBAs
Agenda

- Server Virtualization and different IO models
- Typical network model of VM using bridge
- Direct hardware delegation to the guest aka pass-through
- Merits and challenges of Pass-through model
- PCI IO Virtualization
- Migration of VM with Pass-through IO
IO Virtualization models

- **Server Virtualization:**
  - Provides multiple instances of platform by means of hardware or software, so that multiple instance of different operating systems (VM) can run.

- **IO Virtualization model:**
  - Creates a virtual IO instance on the VM which shares the physical device with other virtual instances.

- **Some popular models:**
  - **Para-Virtualization**
    - Special hypervisor aware drivers
    - Front end drivers run on guest and back end driver runs on Driver domain.
  - **Device Emulation**
    - Processor emulates the entire platform
    - Unmodified operating systems can be run as guest.
Network bridge model

Guest 1
- Frontend driver
- Backend driver

Guest 2
- Frontend driver

Driver Domain
- Bridge
- Native Driver
- Backend driver

Hypervisor

NIC
Drawbacks

- They require special drivers
- Performance degradation (CPU utilization, throughput)
- Hypervisor intervention
- Hardware features not exposed to VM
Pass-through Model

- Dedicate real device to the virtual machine
- IO Memory directly mapped to guest memory, hence avoids the intermediate software layer for device operations
- Close to native performance
- Native driver can be loaded to VM
Performance profiling on different IO models using Neterion’s X3100 series 10Gb Ethernet Adapters.
Pass-through: Challenges

- Bound to the real device, and cannot be shared with other virtual machines

- DMA capable device can corrupt the memory, if enough protection (IOMMU) is not available

- As VM memory includes IO space, the VM cannot be migrated
PCIe IOV Capability

- PCI-SIG has introduced new standards for PCIe, single root IOV and multi root IOV.

- IOV architecture offers a single PCIe device to be act as multiple independent devices.

- The Device will have multiple physical functions (PF) and one or more virtual functions (VF) associated with each PF.

- The Alternative Routing Identifier (ARI) offers a maximum of 256 PF against the traditional 8 functions per device.
Non-IOV vs IOV

- **Traditional multi-function device**

  ![Traditional multi-function device diagram]

- **SRIOV Capable device**

  ![SRIOV Capable device diagram]
PCI Pass-through on IOV NIC

VM1

Native driver

VF1

IOV NIC

Hypervisor

VM2

Native driver

VF2

VMn

Native driver

VF3

VFn
Live Migration

- Relocation of VM from one physical host to another
- In live migration, relocation can be done without interrupting the VM tasks
- Save and copy VM states including CPU state, memory, virtual IO etc
Issues with a pass-through VM

- Writing internal states to device is nearly impossible
- Internal state of device may not be readable and unknown to hypervisor
- Devices have unique properties like MAC addresses which cannot be migrated
Solution

- Many hypervisors like Xen have well implemented PCI hot-plugging feature using ACPI.
- It helps to attach or detach the PCI devices at runtime of VM.
- During migration, NIC can be removed from VM using PCI Hot-plug. But this will cause traffic interrupt.
- A bonding module can be used to aggregate the pass-through channel and virtual interface in a active backup mode.
- When pass-through channel removed, traffic will be switched to virtual interface without interruption.
Migration with bonding module

Before live migration

During live migration
Conclusion

- Device pass-through is the proved IO model for VM and it provides the best IO performance.

- Current works like IOMMU and PCI IOV helps to overcome the limitations of pass-through model.

- VM Migration is one of the key feature of virtualization. The solution of bonding module makes possible the relocation of pass-through VM.
Thank You
Questions?

Q & A