Virtualizing I/O
Yiying Zhang
• Quiz next Mon: cover everything until (including) today’s lecture
  • Closed book
  • Quiz will be 15 minutes (T/F, MC, short answers)
• Project proposal due next Mon
Page Sharing: Scan Candidate PPN

VM 1
VM 2
VM 3

Machine Memory

Hash: ...06af
VM: 3
PPN: 43f8
MPN: 123b

hash page contents

...2bd806af

hint frame

hash table
Page Sharing: Successful Match
Question

- What is the benefit of keeping a "hint" entry for each scanned (but unshared) page (as compared to not maintaining anything for the page)
### Real-World Page Sharing Results

<table>
<thead>
<tr>
<th>Workload</th>
<th>Guest Types</th>
<th>Total MB</th>
<th>Saved MB</th>
<th>Saved %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate IT</td>
<td>Windows</td>
<td>2048</td>
<td>673</td>
<td>32.9</td>
</tr>
<tr>
<td>Nonprofit Org</td>
<td>Linux</td>
<td>1846</td>
<td>345</td>
<td>18.7</td>
</tr>
<tr>
<td>VMware</td>
<td>Linux</td>
<td>1658</td>
<td>120</td>
<td>7.2</td>
</tr>
</tbody>
</table>

- **Corporate IT** – database, web, development servers (Oracle, Websphere, IIS, Java, etc.)
- **Nonprofit Org** – web, mail, anti-virus, other servers (Apache, Majordomo, MailArmor, etc.)
- **VMware** – web proxy, mail, remote access (Squid, Postfix, RAV, ssh, etc.)
Conclusion

• Software and hardware solutions for memory virtualization both have pros and cons

• More things to take care of besides the basic mechanism of memory virtualization

  • Allocation, sharing, overcommitment and reclamation
Outline

• Device virtualization techniques
• Storage virtualization
• Network virtualization

Acknowledgment: some slides from Scott Devine’s lectures at Columbia, Teemu Koponen’s NSDI’14 presentation, and Ben Pfaff’s Network Virtualization lecture
I/O Virtualization

- Goal
  - Multiplexing device across guest VMs

- Challenges
  - Each guest OS has its own device driver
  - How can one device be controlled by multiple drivers?
  - What if one guest OS tries to format its disk?
Possible Solutions of I/O Virtualization

- Direct access: VM exclusively owns a device
- Device emulation: VMM emulates device in software
- Para-virtualization: split driver into guest part and host part
- Hardware assisted: hardware devices offer isolated “virtual interfaces”
Sol-1: Direct Access Device Virtualization

- VM Guest OS
- CPU
- MMU
- Memory
- Controller
- Interface
- NIC
- Controller
- Bridge
- Frame Buffer
- LAN
- CD-ROM
- USB
Direct Access Device Virtualization

• Positives
  • Fast, since the VM uses device just as native machine
  • Simplify monitor: limited device drivers needed

• Negatives
  • Hardware interface visible to guest (bad for migration)
  • Interposition is hard by definition (no way to trap & emulate)
  • Now you need much more devices! (imagine 100 VMs)
Sol-2: Emulating Devices

- Emulate a device
  - Implement device logic in pure software
  - Can even emulate non-existing devices
Example: Emulating Serial Port

Diagram showing the relationship between the Host OS, Device Emulator, Monitor, Guest, and Serial Chip layers.
Emulated Devices

- **Positives**
  - Platform stability (good for migration)
  - Allows interposition
  - No special hardware support is needed

- **Negatives**
  - Can be slow (it’s software emulated)
Sol-3: Para-Virtualized Devices

- VMM offers new types of device
- The guest OS runs a new driver (front-end driver)
- VMM runs a back-end driver for each front-end
- VMM finally runs the real device driver to drive the device
virtio: Linux’s paravirtualized I/O solution

- **Front-end Driver**
  - A kernel module in the guest OS
  - Accepts I/O requests from the user process
  - Transfer I/O requests to back-end driver
- **Back-end Driver**
  - Accepts I/O requests from front-end driver
  - Perform I/O operation via physical device
- **Virtqueue**
  - A memory region accessible from both guest and host OS
  - An interface implemented as vring
Sol-4: Hardware Support for I/O Virtualization

- A virtualization-enabled device can be configured to appear in the PCI configuration space as multiple virtual functions (VFs)

- The VMM assigns one or more VFs to a VM by mapping the actual configuration space of the VFs to the configuration space presented to the VM by the VMM
SR-IOV (Single Root I/O Virtualization)

- Hardware support for guest to access device without going through VMM
- Physical Function (PF)
  - A standard PCIe function
  - Can be associated with multiple VFs
- Virtual Function (VF)
  - A lightweight PCIe function (a unique PCIe xact source)
  - Each VF is isolated from other VFs
  - Has dedicated access to certain hardware resources
  - Share some other resources
- IOVM and PF driver: set up VFs and provide full features to each VF
# Virtualization Technologies Summary

<table>
<thead>
<tr>
<th>Virtualization</th>
<th>Software Solution</th>
<th>Hardware Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>• Trap &amp; Emulate&lt;br&gt;• Instruction interpretation&lt;br&gt;• Binary translation&lt;br&gt;• Para-virtualization</td>
<td>• VT-x&lt;br&gt;• Root / non-root mode&lt;br&gt;• VMCS</td>
</tr>
<tr>
<td>Memory</td>
<td>• Shadow page table&lt;br&gt;• Para-virtualization</td>
<td>• EPT</td>
</tr>
<tr>
<td>Device</td>
<td>• Direct I/O&lt;br&gt;• Device emulation&lt;br&gt;• Para-virtualization</td>
<td>• SR-IOV</td>
</tr>
</tbody>
</table>
Outline

- Device virtualization techniques
- Storage virtualization
- Network virtualization
Virtual Disk

- A virtual disk is just a file in host file system

- Hypervisor maps disk blocks to file offsets
  - Flat file (fix sized virtual disk)
  - Indexed file (virtual disk can grow on demand)
Outline

• Device virtualization techniques
• Storage virtualization
• Network virtualization
Data Center Network Design with VMs

- Core Switch
  - Aggregation Switch
    - "Top of Rack" Switch
      - Machine 1
        - VM VM VM
      - Machine 2
        - VM VM VM
      - ... (other machines)
      - Machine 40
        - VM VM VM

- virtual switch (= vswitch)
  - up to 128 VMs each

- One rack of machines

other agg switches
other ToRs
Problems?

- A physical topology is hard to support different virtual topologies
- Virtualized workloads stay in the physical network address space (L2)
- Slow provisioning
- Limited mobility
- Limited VM placement
- Hardware dependent
- Operationally intensive
Goals (Paraphrase with Machine Virtualization)
Network Virtualization

Packet Abstraction

Mgmt
Mgmt
Mgmt

Logical Network
Logical Network
Logical Network

Network Hypervisor

Control Abstraction

Packet Abstraction + Control Abstraction = Network Hypervisor
A Network Virtualization Distributed System

The Internet

Core Switch

Aggregation Switch

Machine 1

OVS

Machine 2

OVS

VM VM VM

VM VM VM

“Top of Rack” Switch

Core Switch

Aggregation Switch

Machine 3

OVS

Machine 4

OVS

VM VM VM

VM VM VM

“Top of Rack” Switch

Data Center 1

Data Center 2
Management, Control, and Data Planes

- **Management Plane**
- **Control Plane**
- **Data Plane**

**Network Virtualization Application**
- Desired State
- API (Used by Cloud Management System)

**SDN Controller**
- Discovered State
- Control Directives

**vSwitch**
Virtual Network Encapsulation (Data Plane Solution)
Distributed Network Services