Xen and the Art of Virtualization
- Barham et. al.

CSE 598c - Spring 2006
William Enck
Xen’s Goals

• Isolation
  ‣ Access Control
  ‣ Performance

• Heterogeneity
  ‣ Support a variety of Guest OSs

• Low Performance Overhead
Terms

- **Guest OS**: an operating system that Xen can host
- **Domain**: a running virtual machine within a guest OS executes
- **Hypervisor**: Xen, or the VMM. (operates at a higher privilege level than the supervisor of Guest OSs)
A Review

- Types of Virtual Machine Monitors (VMMs)

**Type I**

- App.
- Guest OS
- VMM
- Host Hardware

**Type II**

- App.
- Guest OS
- VMM
- Host OS
- Host Hardware
Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including Domain0 running control software in a XenoLinux environment.
Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including Domain0 running control software in a XenoLinux environment.
Xen Hypervisor

• Hypervisor provides only basic control operations
  ‣ CPU scheduling between domains
  ‣ filtering network packets before transmission
  ‣ access control for block reads

• No need to look deeper
  ‣ e.g. CPU sharing, type of transmitted data
  ‣ Potentially complex policy decisions are best performed by management software in a guest OS
Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including Domain0 running control software in a XenoLinux environment.
Domain0 (Dom0)

• The domain created at boot time is permitted to use the *control interface*
  ‣ Full access to the physical machine

• Responsible for hosting application level management software
  ‣ create and terminate other domains
  ‣ delegate access to machine resources (mem, disks, NICs)
### Domain Management

```
[enck@stout ~]% sudo xm list

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Mem(MiB)</th>
<th>VCPUs</th>
<th>State</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain-0</td>
<td>0</td>
<td>256</td>
<td>1</td>
<td>r-------</td>
<td>210.1</td>
</tr>
<tr>
<td>stout-0</td>
<td>7</td>
<td>64</td>
<td>1</td>
<td>-b------</td>
<td>15.6</td>
</tr>
<tr>
<td>stout-1</td>
<td>8</td>
<td>64</td>
<td>1</td>
<td>-b------</td>
<td>14.6</td>
</tr>
<tr>
<td>stout-2</td>
<td>9</td>
<td>64</td>
<td>1</td>
<td>-b------</td>
<td>14.5</td>
</tr>
<tr>
<td>stout-3</td>
<td>10</td>
<td>64</td>
<td>1</td>
<td>-b------</td>
<td>14.5</td>
</tr>
</tbody>
</table>
```


Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including Domain0 running control software in a XenoLinux environment.
User Domains (DomU)

• Use abstracted interfaces managed by Dom0
  ‣ Virtual network InterFaces (VIF)
    • Dom0 can prevent source address spoofing
  ‣ Virtual Block Devices (VBD)
    • Additional reordering and scheduling in Xen

• Receive a static amount of memory
  ‣ Strong Isolation
  ‣ Can be increased with the Balloon Driver
Device View

[enck@stout-0 ~]# /sbin/ifconfig eth0
eth0   Link encap:Ethernet  HWaddr 00:16:3E:31:33:07
       inet addr:10.0.0.50  Bcast:10.0.0.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:2029 errors:0 dropped:0 overruns:0 frame:0
       TX packets:27  errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:175431 (171.3 KiB)  TX bytes:1134 (1.1 KiB)

[enck@stout-0 ~]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/hda1      4.0G  311M   3.5G   9%   /
.tmpfs          35M   4.0K  35M    1%  /dev/shm

[enck@stout-0 ~]# free -mt
                total  used  free  shared  buffers  cached
Mem:            68   25   43    0      1       10
-/+ buffers/cache: 13   55
Swap:           511    0   511
Total:          580   25  555

[enck@stout-0 ~]#
A Review

• Challenges of Virtualization
  ‣ Processor Support
    • All sensitive instructions must be privileged (i.e. must trap)
    • Not the case for x86
  ‣ Problem Areas
    • Syscalls, Page Faults

• Work-arounds
  ‣ Full Virtualization (VMWare)
  ‣ Signal Handers (User-Mode Linux)
Paravirtualization

- **Drawbacks of Full Virtualization**
  - Efficiency
  - Need for real and virtual time (e.g. TCP timeouts)

- **Solution: Paravirtualization**
  - Present a virtual machine abstraction that is similar but not identical to the underlying hardware
  - Requires modification of the guest OS
    - does not require changes to the ABI (guest applications)
  - Xen does this differently than the Denali project
Control Transfer

• Domain to Xen
  ‣ Hypercall
    • synchronous call from a domain to Xen

• Xen to Domain
  ‣ Asynchronous Event Mechanism
    • Replaces device interrupts
      ‣ e.g., received data
    • Similar to traditional UNIX signals (must register callback)
Data Transfer

- **I/O Descriptor Rings**
  - For transferring data between a Domain and Xen
  - Indirectly reference Guest OS managed I/O buffers
  - Four pointers \{(Producer, Consumer) × (Xen, Domain)\}
    - Domains *produce* requests
    - Xen *consumes* requests
    - Xen *produces* responses
    - Domains *consume* responses

Figure 2: The structure of asynchronous I/O rings, which are used for data transfer between Xen and guest OSes.
Virtual Address Translation

• Hardware page tables cause problems
  ‣ No ASID tags on TLB = flush on address space switch

• VMWare: “shadow” page tables (hurts performance)

• Paravirtualization allows Xen to avoid this
  ‣ Xen registers Guest OS page tables directly with MMU
    • Restricts Guest OSes to read-only access
    • Guest OS manages page tables (hypercalls)
    • Xen need only be involved in page table updates
    • Xen in top 64MB
Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).
Applications of VMMs

- Server Consolidation
- Co-Located Hosting Facilities
- Distributed Web Services
- Secure Computing Platforms
- Application Mobility
Xen Since 1.0

• I/O changes
• Live Migration
• SMP for DomUs
• Support for Intel VT-x and AMD Pacifica
  ‣ No more paravirtualization!