Xen and the Art of Virtualization

Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield

Presented by Zhenkun Yang

Computer Science Department
Portland State University
• Virtual Machine Monitor (hypervisor)

• Initially developed by The Univ. of Cambridge Computer Lab
Outline

● History of Virtualization
● Xen
● Virtual Machine Interface
● Memory Management
● Control and Data Transfer
● Subsystem Virtualization
● Evaluation
● Conclusion & Current Status
History of Virtualization

- Full Virtualization
  - No need to modify guest OSes
  - Performance penalty
  - Problematic for certain privileged instructions

- Paravirtualization
  - Require modification to guest OSes
  - Near-native performance
Xen

• Paravirtualization
• High-performance
• Targeting to host 100 guest OSes
Xen
Virtual Machine Interface

- CPU
  - Privileged instructions must be validated by Xen
  - X86 exceptions -> Xen exception handler table
Virtual Machine Interface

• Device I/O
  • Clean and simple device abstraction
  • I/O data transfer between domain and Xen via shared memory and asynchronous buffer-descriptor rings
Virtual Machine Interface

- Memory Management
  - No software-managed TLB in x86
  - Guest OSes are responsible for hardware page tables
  - Xen exists in top section of each address space
  - Avoid TLB flush when guest OS enter/exist Xen
Memory Management

- Xen registers guest OS page table directly with MMU
- Guest OS only has read-only access
- Page table update validated by Xen
- Updates are batched to minimize # of Hypercalls
- **Xen**: globally readable *hardware-to-physical* table
- **Domain**: *physical-to-hardware* table
Control Transfer

- Hypercalls
  - Like system calls
  - From domains to Xen

- Asynchronous event
  - Like Unix signal
  - From Xen to domains
Data Transfer

- Circular queue of descriptors
  - Allocated by domain, accessible from Xen
  - Two pairs of producer-consumer
Subsystem Virtualization

- CPU scheduling
  - Borrowed Virtual Time scheduling algorithm
- Time and timer
  - Real time (advances regardless of execution domain)
  - Virtual time (advances within the context of the domain)
  - Wall-clock time (offset added to real time)
- Network & Disk
### Evaluation

<table>
<thead>
<tr>
<th></th>
<th>SPEC INT2000 (score)</th>
<th>Linux build time (s)</th>
<th>OSDB-IR (tup/s)</th>
<th>OSDB-OLTP (tup/s)</th>
<th>dbench (score)</th>
<th>SPEC WEB99 (score)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relative score to Linux</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>0.567</td>
<td>0.554</td>
<td>0.535</td>
<td>0.556</td>
<td>0.65</td>
<td>0.545</td>
</tr>
<tr>
<td>X</td>
<td>0.567</td>
<td>0.554</td>
<td>0.535</td>
<td>0.556</td>
<td>0.65</td>
<td>0.545</td>
</tr>
<tr>
<td>V</td>
<td>0.567</td>
<td>0.554</td>
<td>0.535</td>
<td>0.556</td>
<td>0.65</td>
<td>0.545</td>
</tr>
<tr>
<td>U</td>
<td>0.567</td>
<td>0.554</td>
<td>0.535</td>
<td>0.556</td>
<td>0.65</td>
<td>0.545</td>
</tr>
</tbody>
</table>

Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U)
## Evaluation

Table 3: lmbench: Processes - times in $\mu$s

<table>
<thead>
<tr>
<th>Config</th>
<th>null call</th>
<th>null I/O</th>
<th>stat</th>
<th>open close</th>
<th>slct TCP</th>
<th>sig inst</th>
<th>sig hndl</th>
<th>fork proc</th>
<th>exec proc</th>
<th>sh proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-SMP</td>
<td>0.53</td>
<td>0.81</td>
<td>2.10</td>
<td>3.51</td>
<td>23.2</td>
<td>0.83</td>
<td>2.94</td>
<td>143</td>
<td>601</td>
<td>4k2</td>
</tr>
<tr>
<td>L-UP</td>
<td>0.45</td>
<td>0.50</td>
<td>1.28</td>
<td>1.92</td>
<td>5.70</td>
<td>0.68</td>
<td>2.49</td>
<td>110</td>
<td>530</td>
<td>4k0</td>
</tr>
<tr>
<td>Xen</td>
<td>0.46</td>
<td>0.50</td>
<td>1.22</td>
<td>1.88</td>
<td>5.69</td>
<td>0.69</td>
<td>1.75</td>
<td>198</td>
<td>768</td>
<td>4k8</td>
</tr>
<tr>
<td>VMW</td>
<td>0.73</td>
<td>0.83</td>
<td>1.88</td>
<td>2.99</td>
<td>11.1</td>
<td>1.02</td>
<td>4.63</td>
<td>874</td>
<td>2k3</td>
<td>10k</td>
</tr>
<tr>
<td>UML</td>
<td>24.7</td>
<td>25.1</td>
<td>36.1</td>
<td>62.8</td>
<td>39.9</td>
<td>26.0</td>
<td>46.0</td>
<td>21k</td>
<td>33k</td>
<td>58k</td>
</tr>
</tbody>
</table>

Table 4: lmbench: Context switching times in $\mu$s

<table>
<thead>
<tr>
<th>Config</th>
<th>2p</th>
<th>2p</th>
<th>2p</th>
<th>8p</th>
<th>8p</th>
<th>16p</th>
<th>16p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0K</td>
<td>16K</td>
<td>64K</td>
<td>16K</td>
<td>64K</td>
<td>16K</td>
<td>64K</td>
</tr>
<tr>
<td>L-SMP</td>
<td>1.69</td>
<td>1.88</td>
<td>2.03</td>
<td>2.36</td>
<td>26.8</td>
<td>4.79</td>
<td>38.4</td>
</tr>
<tr>
<td>L-UP</td>
<td>0.77</td>
<td>0.91</td>
<td>1.06</td>
<td>1.03</td>
<td>24.3</td>
<td>3.61</td>
<td>37.6</td>
</tr>
<tr>
<td>Xen</td>
<td>1.97</td>
<td>2.22</td>
<td>2.67</td>
<td>3.07</td>
<td>28.7</td>
<td>7.08</td>
<td>39.4</td>
</tr>
<tr>
<td>VMW</td>
<td>18.1</td>
<td>17.6</td>
<td>21.3</td>
<td>22.4</td>
<td>51.6</td>
<td>41.7</td>
<td>72.2</td>
</tr>
<tr>
<td>UML</td>
<td>15.5</td>
<td>14.6</td>
<td>14.4</td>
<td>16.3</td>
<td>36.8</td>
<td>23.6</td>
<td>52.0</td>
</tr>
</tbody>
</table>
Evaluation

Table 5: lmbench: File & VM system latencies in $\mu$s

<table>
<thead>
<tr>
<th>Config</th>
<th>0K File create</th>
<th>0K File delete</th>
<th>10K File create</th>
<th>10K File delete</th>
<th>Mmap lat</th>
<th>Prot fault</th>
<th>Page fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-SMP</td>
<td>44.9</td>
<td>24.2</td>
<td>123</td>
<td>45.2</td>
<td>99.0</td>
<td>1.33</td>
<td>1.88</td>
</tr>
<tr>
<td>L-UP</td>
<td>32.1</td>
<td>6.08</td>
<td>66.0</td>
<td>12.5</td>
<td>68.0</td>
<td>1.06</td>
<td>1.42</td>
</tr>
<tr>
<td>Xen</td>
<td>32.5</td>
<td>5.86</td>
<td>68.2</td>
<td>13.6</td>
<td>139</td>
<td>1.40</td>
<td>2.73</td>
</tr>
<tr>
<td>VMW</td>
<td>35.3</td>
<td>9.3</td>
<td>85.6</td>
<td>21.4</td>
<td>620</td>
<td>7.53</td>
<td>12.4</td>
</tr>
<tr>
<td>UML</td>
<td>130</td>
<td>65.7</td>
<td>250</td>
<td>113</td>
<td>1k4</td>
<td>21.8</td>
<td>26.3</td>
</tr>
</tbody>
</table>

Table 6: ttcp: Bandwidth in Mb/s

<table>
<thead>
<tr>
<th></th>
<th>TCP MTU 1500</th>
<th>TCP MTU 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TX</td>
<td>RX</td>
</tr>
<tr>
<td>Linux</td>
<td>897</td>
<td>897</td>
</tr>
<tr>
<td>Xen</td>
<td>897 (-0%)</td>
<td>897 (-0%)</td>
</tr>
<tr>
<td>VMW</td>
<td>291 (-68%)</td>
<td>615 (-31%)</td>
</tr>
<tr>
<td>UML</td>
<td>165 (-82%)</td>
<td>203 (-77%)</td>
</tr>
</tbody>
</table>
Evaluation

SPEC WEB99 for 1, 2, 4, 8 and 16 concurrent Apache servers
higher values are better

Performance of multiple instances of PostgreSQL running OSDB in separate Xen domains.
8(diff) bars show performance variation with different scheduler weights
Scalability

Normalized aggregate performance of a subset of SPEC CINT2000 running concurrently on 1-128 domains
Conclusion

- Xen is a high performance VMM
- Xen uses paravirtualization
- Guest OSes need modification
- Performance near native Linux
Current Status

- Processors with Intel VT or AMD-V available
- Hardware Virtual Machine (HVM)  
  Allowing un-modified guest OS
- Xen 3.0 supports Intel VT and AMD-V (2005)

- Xen AB: Citrix, Fujitsu, HP, IBM, Intel, Novell, Oracle, and VA Linux Systems Japan
- Amazon EC2, Fujitsu Global Cloud Platform, Linode, and Rackspace Cloud use **Xen** as the hypervisor
Thank You!

Reference:

- http://xen.org/community/