Quantitative Comparison of Xen and KVM

Todd Deshane, Ph.D. Student, Clarkson University
Xen Summit, June 23–24, 2008, Boston, MA, USA.
Background

- Xen and the Art of Virtualization (2003)
  - Reported remarkable performance results
  - Validated performance results
  - Isolation Benchmark Suite
  - Performance isolation testing methodology
  - Lots of attention from virtualization developers and industry
Goals

- Understand architectural differences
  - Stand-alone versus integrated hypervisor

- Help developers realize areas of improvement
  - Difficult for developers to test all cases
  - Overall performance is important, but not the only factor

- Help users make informed decisions
  - Growing number of virtualization options to choose from
  - Different users have different virtualization needs
  - Hardware and software versions can make a big difference
Experimental Setup

- **Base machine**
  - Ubuntu Linux 8.04 AMD64

- **Software packages from Ubuntu repositories**
  - Linux kernel 2.6.24–18
  - Xen 3.2.1+2.6.24–18–xen
  - KVM 62

- **Guests**
  - Ubuntu Linux 8.04 AMD64
  - Automated debootstrap

- **Hardware**
  - 2.4 GHz Intel Core 2 CPU 6600, 4 GB of RAM, 250 GB disk
Virtualization Benchmarking Pillars

- Overall performance
  - Performance of the whole system and components
  - Focused on macro-benchmarks
  - Standard benchmarks and repeatable methods

- Performance isolation
  - Protection from resource consumption from other guests
  - Representative workload with and without stress tests

- Scalability
  - Ability to run more guests without loss of performance
  - Same workload on each guest, increase number of guests
Overall Performance: Methodology

- Macro–benchmarks
  - CPU
  - Kernel compile
  - Disk I/O
- Automated guest build
  - Benchvm virtualization benchmark suite
- Automated testing and reporting
  - Phoronix Test Suite
Overall Performance: Results

<table>
<thead>
<tr>
<th></th>
<th>Linux</th>
<th>Xen</th>
<th>KVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.000</td>
<td>0.999</td>
<td>0.993</td>
</tr>
<tr>
<td>Kernel Compile</td>
<td>1.000</td>
<td>0.487</td>
<td>0.384</td>
</tr>
<tr>
<td>IOzone Write</td>
<td>1.000</td>
<td>0.855</td>
<td>0.934</td>
</tr>
<tr>
<td>IOzone Read</td>
<td>1.000</td>
<td>0.852</td>
<td>0.994</td>
</tr>
</tbody>
</table>
Overall Performance: Discussion

- Xen and KVM had similar CPU performance
  - Xen: 0.999, KVM: 0.993
- Xen was better than KVM on kernel compile
  - Xen: 0.487, KVM: 0.384
- KVM was better on disk I/O
  - Write – Xen: 0.855, KVM: 0.934
  - Read – Xen: 0.852, KVM: 0.994
  - Disk caching effects?
Performance Isolation: Methodology

- Isolation Benchmark Suite
  - Memory stress test: calloc()
  - Fork stress test: fork()
  - CPU stress test
    - Mixed calculations in tight loop
  - Disk
    - Threaded IOzone read and write
  - Network receiver
    - Receive threaded UDP traffic from external host
  - Network sender
    - Send threaded UDP traffic to external host
Performance Isolation: Methodology

- VM1 (web server)
- VM2 (web server)
- VM3 (web server)
- VM4 (web server)

Virtualization System

- Machine 1 (specweb client)
- Machine 2 (specweb client)
- Machine 3 (specweb client)
- Machine 4 (specweb client)
### Performance Isolation: Results

<table>
<thead>
<tr>
<th></th>
<th>Xen Stressed VM</th>
<th>Xen Normal VM</th>
<th>KVM Stressed VM</th>
<th>KVM Normal VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>DNR</td>
<td>0</td>
<td>DNR</td>
<td>0</td>
</tr>
<tr>
<td>Fork</td>
<td>DNR</td>
<td>0</td>
<td>DNR</td>
<td>0</td>
</tr>
<tr>
<td>CPU</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disk</td>
<td>3</td>
<td>0.03</td>
<td>22.2</td>
<td>0.01</td>
</tr>
<tr>
<td>Network receiver</td>
<td>0.27</td>
<td>0.7</td>
<td>DNR</td>
<td>0.12</td>
</tr>
<tr>
<td>Network sender</td>
<td>2.53</td>
<td>2.08</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Performance Isolation: Discussion

- Xen was isolated on memory, fork, CPU, disk
- Xen was slightly isolated on network sender
- Xen showed no isolation on network receiver
  - Kernel bug(s) in Ubuntu?
- Xen had unexpectedly good disk performance
- KVM was well-isolated on all stress tests
- KVM had unexpectedly good network sender performance
- KVM had unexpectedly poor disk and network receiver performance
Scalability: Methodology

- Ran Apache compile in 1, 2, 4, 16, and 30 guests
- Measured compile time and number of guests that ran to completion
Scalability: Results

Xen Scalability

KVM Scalability
Scalability: Discussion

- Xen scaled linearly with respect to number of guests
- KVM had many guest crashes
  - 4 guests: 1 crashed guest
  - 8 guests: 4 crashed guests
  - 16 guests: 7 crashed guests
  - 30 guests: system crashed during compile
Lessons Learned

- Virtualization benchmarking is still difficult
- Testing on multiple categories is crucial
- Automated testing is important and useful
  - Transparency in testing methods
  - Repeatability is needed, yet challenging
  - Always more cases to test
  - Challenging to adequately benchmark rapidly evolving technologies
Future Work

- Extend testing to include Xen HVM, and KVM with paravirt I/O
- More complete automation of testing process with benchvm
- Port benchvm to Python
- Add support for more distros in benchvm
- Use Phoronix Test Suite–like functionality
  - Test profiles, test suites, batch benchmarking
  - Automated results parsing
  - Graphing/uploading of results
  - Automated system and test config collection and publishing
Acknowledgments

- Benchmarking co-researchers
  - Zachary Shepherd, Jeanna Neefe Matthews, Muli Ben-Yehuda, Amit Shah, and Balaji Rao

- Performance isolation and scalability researchers
  - Wenjin Hu and Madhujith Hapuarachchi

- Early developers and testers of benchvm
  - Cyrus Katrak and Martin McDermott

- Members of the Xen and KVM communities
  - Feedback and support
Questions / Comments / Suggestions?