Toward On-demand Nested Virtualization for Live-Refreshing Cloud Systems

Ryosuke Yasuoka (University of Tsukuba), Takaaki Fukai, Takahiro Shinagawa (The University of Tokyo)

**Background**

- VMM needs to be refreshed (=rebooted) for applying patches, upgrading, rejuvenation
- VMs live migrated avoids stopping VMs
- Problem: Heavy network load

**Our goal**

1. Keep the advantages of nested virtualization
2. Eliminate the nested virtualization overhead during normal runtime

**Proposal: On-demand Nested Virtualization with hardware partitioning**

**In refreshing the VMM**

1. Virtualize
2. Create partition & boot L1 VMM #1
3. Migrate VMs
4. Reboot L1 VMM #0

**During normal runtime**

1. Virtualize
2. Create partition & boot L1 VMM #1
3. Migrate VMs
4. Reboot L1 VMM #0

**Performance evaluation: KVS throughput**

- Workload: YCSB benchmark (Read : Update = 50%:50%)
- KVS: Redis

<table>
<thead>
<tr>
<th></th>
<th>Throughput [ops/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-nested KVM</td>
<td>8000</td>
</tr>
<tr>
<td>NestedKVM</td>
<td>4550 (↓36.2%)</td>
</tr>
<tr>
<td>Proposal (Nested)</td>
<td>4200 (↓4.1%)</td>
</tr>
<tr>
<td>Proposal (Devirt)</td>
<td>3980 (↓2.0%)</td>
</tr>
</tbody>
</table>

**Implementation**

- L0 VMM: Based on TinyVisor [1]
  (diff: +3318, -48)
- L1 VMM #0: Custom KVM (diff: +292, -0)
- L1 VMM #1 and L2 guest OS: no change

[1] https://osdn.net/projects/tinyvisor/

**Future work**

- Performance measurement on VMM refreshing
- Gradual resource reallocation
- Zero-copy migration