Painless Cache Allocation in Cloud

Lin Yang, lin.a.yang@intel.com
Sunku Ranganath, sunku.ranganath@intel.com
Malini Bhandaru, malini.k.bhandaru@intel.com
Last Level Cache Matters

- Cache resource contention introduces jitter and increased latency

NFV & RT workloads are time sensitive

Intel Resource Director Technology (RDT)

- Hardware based CPU cache control
  - Allocation per process (VM, application, etc.)
  - Shared and isolated usage models
- Monitoring and LLC partitioning mechanisms provide isolation and prioritization of VMs or apps
- Software Support: Resctrl fs, PQoS toolset, Resource Management Daemon

1: Source: UC Berkeley (UCB) Tests, 2016, see backup for details
Last Level Cache as a Resource – NFV use case

Scheduling Considerations:
- Node capacity of cache
- Workload need for DDIO
- Workload sensitivity and mix
- Isolated vs. Shared allocations

Node level cache manager:
- Crucial to have local agent on the host
- Dynamic cache control
  - Millisecond timescales for time sensitive workloads

Throughput Improvement of up to 46% in presence of noisy neighbor

Allocation with Resource Management Daemon (RMD)

RMD - A Linux daemon that:
- Runs on individual hosts
- REST API, accessible to orchestrator
- Accepts & enforces policy

Open Source: https://github.com/intel/rmd

Why use RMD:
- Ability to use LLC as a resource
- Satisfies multiple usecases with varying resource policies
## Design/Implementation Options

<table>
<thead>
<tr>
<th>Features</th>
<th>Via Libvirt</th>
<th>Via RMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache allocation &amp; updates</td>
<td><em>Static, One Time</em> Configuration</td>
<td><em>Dynamic, Run time</em> updates based on the <em>policy</em></td>
</tr>
<tr>
<td>Hardware Bitmasks</td>
<td><em>Operator</em> required to understand bitmasks for complex allocations</td>
<td><em>Hides</em> bitmasks constraints &amp; calculations</td>
</tr>
<tr>
<td>Cache Policies</td>
<td><em>No</em> concept of Isolated vs. Shared cache policies</td>
<td><em>Supports</em> Isolated &amp; Shared cache policies</td>
</tr>
<tr>
<td>Hyper-Convergence</td>
<td>May <em>not</em> be efficient</td>
<td><em>Out of the box</em> efficient configuration</td>
</tr>
<tr>
<td>Short timeframe resource control</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Low level resource control</td>
<td>Just at <em>VM Level</em></td>
<td><em>Process level</em> control</td>
</tr>
</tbody>
</table>

RMD is complimentary to Libvirt
OpenStack with RMD

Blueprint: Please provide feedback
https://review.openstack.org/#/c/568678/

Scale with RMD

• Enables secure cache allocation policies
• Manage multiple platform resources like memory bandwidth control using Intel RDT

Figure: Flavor Spec with max_cache:min_cache model
Thank you