Resource Allocation: Intel Resource Director Technology (RDT)

Fenghua Yu <fenghua.yu@intel.com>

July 14, 2016
Introduction

• Intel Resource Director Technology (RDT)
  • Monitoring: Cache Monitoring Technology (CMT), Memory Bandwidth Monitoring (MBM), and more.
    • Passively monitor resources usage to identify QoS and performance bottlenecks
  • Allocation: Cache Allocation Technology (CAT), Code and Data Prioritization (CDP), and more.
    • Actively allocate resources to achieve better QoS and performance
Problems of Cache Sharing

Sometimes sharing is bad…Noisy Neighbor

- Slow Down High Pri Process
- Long IRQ Latency in Real Time
- Low Throughput in Container
Solution?

No sharing.....Allocate separate cache for each app and no more noisy neighbor

Speed Up High Pri Process

Shorten IRQ Latency in Real Time

High Throughput in Containers
Cache Allocation Technology (CAT)

- Enables OS or Hypervisor or container to specify the amount of cache space an app can use
- Enables more cache space to be made available for high priority apps.
- CAT L3 was first introduced on Haswell server, then on Broadwell server and Skylake server
  - L3 is LLC (Last Level Cache) on the processors
- CAT L2 is released in Software Development Manual
Code and Data Prioritization (CDP)

• Extension of CAT
• Enables isolation and separate prioritization of code and data
  Code and Data Prioritization (CDP)
  • provides separate code and data masks per CLOSID.
• First implementation is on Broadwell server and then on Skylake server
CAT L3 Hardware Architecture

```
<table>
<thead>
<tr>
<th>CLOSID</th>
<th>Cap Bit Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CBM 0</td>
</tr>
<tr>
<td>1</td>
<td>CBM 1</td>
</tr>
<tr>
<td>2</td>
<td>CBM 2</td>
</tr>
</tbody>
</table>

Allocated for CPU1
```
Multi Resources Allocation: L2 and L3 CAT

<table>
<thead>
<tr>
<th>CLOSID</th>
<th>Cap Bit Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CBM 0</td>
</tr>
<tr>
<td>1</td>
<td>CBM 1</td>
</tr>
<tr>
<td>2</td>
<td>CBM 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CLOSID</th>
<th>Cap Bit Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CBM 0</td>
</tr>
<tr>
<td>1</td>
<td>CBM 1</td>
</tr>
<tr>
<td>2</td>
<td>CBM 2</td>
</tr>
</tbody>
</table>
Enable Features in Linux Kernel

Switch to a process

Resource allocation schema per CPU?

Y

Write closid in schema per task_struct to PQR register

N

Run the process with CLOSID in PQR and CBM in QoS_MASK

Feature enumeration and enabling from CPUID

User interface defines resource allocation schemas per task or per CPU
User Interface

- Kernel creates a file system “rscctrl” (standing for ReSourCe ConTRoL) to hold user interface
  - Mounted as /sys/fs/rscctrl
    - Upon mounted, the file system has the directory:
      - /sys/fs/rscctrl/tasks: contains all pids initially
      - /sys/fs/rscctrl/cpus: all zero’s initially
      - /sys/fs/rscctrl/schemas: all 1’s initially means all processes can use all resources for all tasks by default
  - User can create resource allocation schema for a group of tasks or cpus
    - mkdir /sys/fs/rscctrl/p1: Create a sub-dir p1 under the rscctrl file system.
      - /sys/fs/rscctrl/p1/tasks: user can add pids to the file to assign resources on the pids. Initial value is empty.
      - /sys/fs/rscctrl/p1/cpus: user can add cpu masks to the file to assign resources on the cpus. Initial value is 0.
      - /sys/fs/rscctrl/p1/schemas: user can write L3 and L2 cbms to this file. Initial value is all 1’s.
  - User modifies schemas, and assigns tasks/cpus to use the schemas.
Usage case 1 - CAT L3 for Open Stack

Open Stack specifies L3 allocation schema and starts Guest1 and Guest2

Kernel rscctrl interface

Kernel assigns guest a portion of L3

L3 Allocated for Guest1 Allocated for Guest2
Usage case 2 - CAT L3 for Real Time Apps

Sysadmin launches real time app p1

Sysadmin specifies L3 allocation schemas for p1 and p2

Kernel rscctrl interface

Kernel assigns app a portion of L3

Allocated for p1  L3  Allocated for p2

Sysadmin launches real time app p2
Performance Improvement by CAT (Case 1)

Intel® Resource Director Technology (Intel® RDT) - University of California, Berkeley

- Network functions are executing simultaneously on isolated core's, throughput of each Virtual Machines is measured
- Min packet size (64 bytes), 100K flows, uniformly distributed
- LLC contention causes up to 51% performance degradation in throughput

Max.% throughput degradation, normalized

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Configurations: see slide 31. For more complete information, visit http://www.intel.com/performance/datacenter
Performance Improvement by CAT (Case 1)(cont.)

Intel® Resource Director Technology (Intel® RDT) - University of California, Berkeley

- Network functions are executing simultaneously on isolated core's, throughput of each Virtual Machines is measured
- Min packet size (64 bytes), 100K flows, uniformly distributed
- VM under test is isolated utilizing CAT, 2 Ways of LLC are associated with the Network function. Isolation only causes ~2% variation

http://span.cs.berkeley.edu

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Configurations: see slide 28. For more complete information, visit http://www.intel.com/performance/datacenter.
Performance Improvement by CAT (Case 2)

No QoS: Thread Contention

With CAT applied: Reduced Thread Contention

Resource contention causes up to 4X slowdown in performance
(Need ability to monitor and enforce cache/memory resource usage)

Previous Contention Reduced Substantially!
Status

• Previous cgroup user interface Linux kernel patches were rejected by upstream because of cgroup and user interface design limitations.

• We proposed a new expandable and fine tuning user interface infrastructure design
  • Multi resources allocation: expandable to L3, L2, and so on.
  • Per resource domain allocation: fine control each resource allocation unit
  • Resource allocation for kernel thread:
    • Allocation per CPU:

• The new rscctrl user interface and kernel design patches were published on lkml on 7/12/2016 and are being reviewed by the community.
References

• x86 Software Developer Manual

• Latest patches for CAT and CDP: https://github.com/fyu1/linux/tree/cat16.1 and lkml

