White Paper

Data Center
Hyperscale Cloud Data Center

ByteDance: Performance Evaluation and Optimization for Workload Colocation

How Intel® Resource Director Technology and Intel® Platform Resource Manager improve cluster performance for ByteDance

ByteDance Authors

Sun Dianjun
Duan Xiongchun
Jiang Fan
Song Muchun
Yin Hongbo
Zhou Chenming
Xiang Wu

Intel Authors

Shen Huanxing
Shen Xiaochen
Li Cong
Huang Tai
Zhou Shen

ByteDance

Table of Contents

Background of Workload Co-Location

Co-locating workloads of different priorities on a single server is a new way to improve total server utilization. Traditionally, only one latency-critical workload runs on a single server, but usually it is not able to fully utilize all the CPU utilization all the time. This provides an opportunity to place low priority workloads to consume the unused CPU time. For example, an end-user-facing workload may enjoy a daily pattern of CPU utilization: high utilization at night and low utilization in the morning. In this case, low-priority tasks can run as much as they can in the morning and give back the CPU time in the evening. The bottom line is that the service level agreement (SLA) of the latency-critical workload cannot be violated.

Maintaining the SLA of a latency-critical workload turns out to be a great challenge for workload co-location. The entire computing stack, from high-level software to low-level hardware resource, needs to always be capable of meeting the demand from the high-priority workload. For example, the Linux kernel task scheduler decides which task to run. When a latency-critical workload receives bursting requests, the scheduler needs to reclaim the CPU time from co-located low priority tasks and let the critical task to run as soon as possible, otherwise, the performance of that latency-critical workload is at risk. Low-level hardware resources, for example, the last level cache and memory bandwidth, are even harder to be prioritized. Some workloads may run in a low CPU utilization, giving plenty of CPU time to other workloads to run. But its workload performance can be extremely sensitive to the last level cache miss when the workload keeps frequently visited data in the last level cache. If the co-located task aggressively takes last level cache, the latency-critical workload may not be able to maintain its SLA.

What makes the problem even more challenging is that the SLA definition becomes stricter. For better user experience, workload owners use the tail latency as the SLA for their critical workloads. It is a difficult goal, even in a non-co-located cluster where all the computing resources are dedicated to one workload.

When the workload SLA is violated in the cluster, it is difficult to identify the root cause. The cluster owner has to go through all the possible impact, from software code change to runtime configurations and from high level resource management to low level resource assignment. When running dozens of workloads in a cluster, it is a tedious job to analyze performance drop one by one. But without knowing the root cause, the cluster administrator is nowhere near confident to employ a mitigation plan.

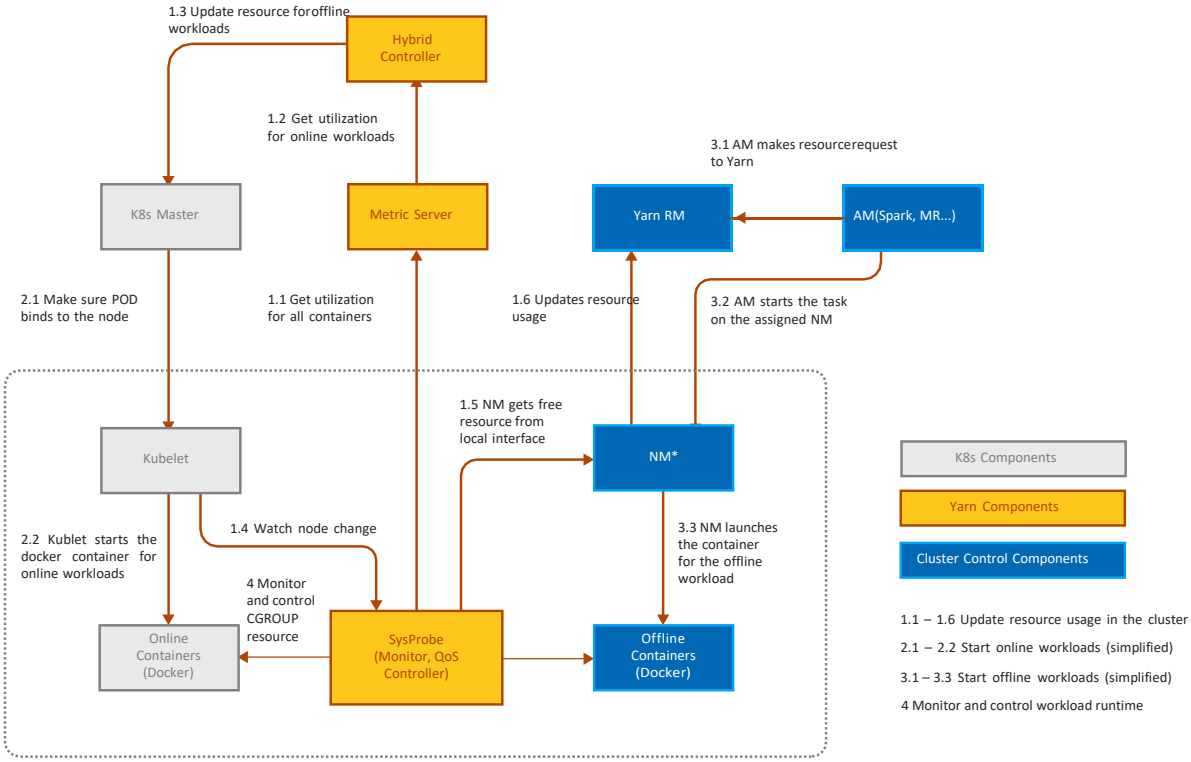

Figure 1. Overview of ByteDance Cluster Management

ByteDance Co-Location Cluster Overview

Based in Beijing, ByteDance is a Chinese internet technology company operating several machine learning-enabled content platforms. ByteDance has built up a co-located cluster to improve total server utilization. They have observed the CPU utilization daily pattern for some of the workloads and seen that as the opportunity to co-locate low priority jobs, for example, machine learning tasks, to consume the free CPU time. Two types of workloads have been co-located in the ByteDance co-location cluster: online workloads and offline workloads. Online workloads are Remote Procedure Call (RPC) services. They have stringent SLA requirements. Most offline workloads are throughput oriented, such as Hadoop tasks or video transcoding jobs. The goal of the co-location cluster is to improve the total server utilization and maintain the online workloads' SLA at the same time.

In order to maintain the online workload performance, CPU resources for offline workloads and online workloads are carefully maintained: offline workloads' CPU time is reclaimed as soon as online workloads ask for CPU time. It is implemented by cpuset. All the online workloads run within a cpuset and offline workloads in another cpuset. They do not share logical or physical CPU cores. The resource controller constantly adjusts the cpuset configurations according to the CPU load of all the online workloads. More CPUs are given to online workloads' cpuset when the CPU load of all the online workloads increases. When the CPU load decreases, the resource controller gives the CPU cores to the offline workloads. There could be dozens of different online workloads running on one server. They are allowed to run on any CPUs in the cpuset for the online workloads. The same logic is applied to offline workloads.

With workload co-location, the total server utilization is improved compared to the non-co-located clusters. However the fluctuation of the online workload tail latency is greater than that in the non-co-located clusters. The root cause is not clear. It could be impacted from the high-level resource competition between online and offline workloads. It could also be impacted from the low-level resource interferences, for example, last level cache and memory bandwidth. The only known fact is that performance worsens in co-location.

Shared Hardware Resource Interference Analysis

The goal of performance analysis for ByteDance's co-located cluster is to find out for a single online workload whether its performance is impacted by offline workloads and whether the performance loss is caused by low-level hardware interference, for example, last level cache. If the analysis result confirms the impact from last level cache or memory bandwidth, we have a great opportunity for it to be mitigated by Intel® Resource Director Technology (Intel® RDT).

A performance analysis method was designed to evaluate whether the online workloads are impacted by co-located workloads on the shared hardware resources. For one workload, a regression model was created based on the low-level performance counters collected at runtime. Results show that online workloads suffered from last level cache interferences, which in turn impacted workload performance. Based on the analysis, we believe that cache regulation can help to mitigate the cache interferences on the online workloads and secure their performance.

Low Level Performance Counters

The workload performance model is based on the low-level performance counters. Performance counters are a hardware feature provided by the platform to bookkeep certain hardware execution behavior. Three counters are chosen as the indicator of the workload performance from CPU perspectives: unhalted CPU cycles, retired instructions, and cache misses. Cycles per instruction (CPI) denotes how many CPU cycles are consumed by one instruction in the average. A higher CPI suggests more CPU cycles are consumed to complete the instructions. At the higher level, the workload performance might be impacted. Cache misses per kilo-instructions (MPKI) measures the number of last level cache misses per kilo instructions of the workload. It is used to pinpoint the root cause when CPI is higher than usual. If CPI and MPKI are simultaneously greater than usual, most likely the workload performance has been impacted by the cache misses. Closely monitoring these metrics for one workload at runtime, we will be able to find performance impacting interferences from shared low-level resource, if there are any.

The traditional performance indicator, e.g., tail latency, cannot be used in this case for performance evaluation. For ByteDance, an online workload makes function call to other services to complete one transaction, so its tail latency can be impacted not only by its own performance but also by how quickly other services complete the function call. As a result, we cannot use the tail latency of an online workload to study the interferences from co-located workloads.

In order to analyze the workload performance offline in the ByteDance co-location cluster, performance counters, cache occupancy metrics, and other supporting metrics, for example, CPU utilization and workload intensity, are collected every 30 seconds for each online workload. These metrics are either used to build the performance model or to validate the analysis findings.

Intel® Resource Director Technology

Intel® Resource Director Technology (Intel® RDT) brings new levels of visibility and control over how shared resources such as last-level cache (LLC) and memory bandwidth are used by applications, virtual machines (VMs), and containers. It's the next evolutionary leap in workload consolidation density, performance consistency, and dynamic service delivery, helping to drive efficiency and flexibility across the data center while reducing overall total cost of ownership (TCO). As software-defined infrastructure and advanced resource-aware orchestration technologies increasingly transform the industry, Intel® RDT is a key feature set to optimize application performance and enhance the capabilities of orchestration and virtualization management server systems using Intel® Xeon® processors.

Intel® RDT provides a framework with several component features for cache and memory monitoring and allocation capabilities, including CMT, CAT, MBM, and MBA (see Figure 2: Intel® RDT Features). These technologies enable tracking and control of shared resources, such as the Last Level Cache (LLC) and main memory (DRAM) bandwidth, in use by many applications, containers or VMs running on the platform concurrently. Intel® RDT may aid "noisy neighbor" detection and help to reduce performance interference, ensuring the performance of key workloads in complex environments.

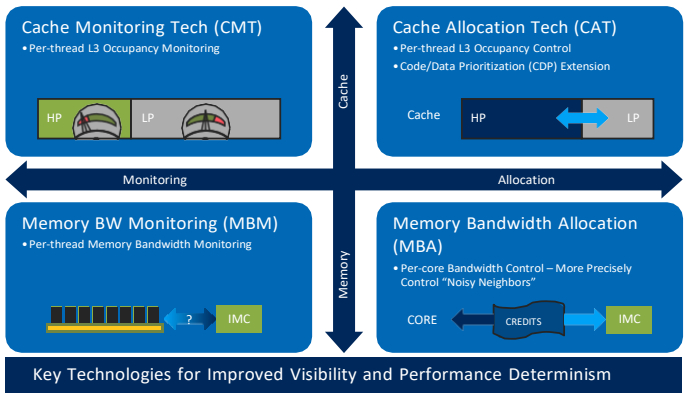

Figure 2. Intel® RDT Features

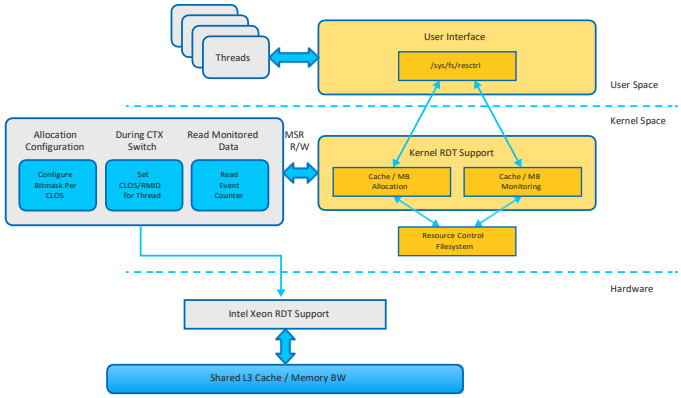

Figure 3. Intel® RDT Kernel Architecture

Figure 3: Intel® RDT Kernel Architecture depicts the Linux kernel framework and implementation for Intel® RDT features. The per-core and per-socket MSR register operations, such as capabilities enumeration, monitoring and allocation configuration, CLOS/RMID association with threads, reading monitoring counters, are wrapped into filesystem operations. From an end user's perspective, Intel® RDT monitoring and allocation features are enabled through resource control filesystem, which is mounted under /sys/fs/resctrl by default.

Intel® RDT hierarchy in resctrl filesystem is similar to control groups (cgroups). Comparing with cgroups, the resctrl filesystem interface has similar process management lifecycle and user interfaces. But unlike cgroups' hierarchy, it has single level filesystem layout.

Resource groups are represented as directories in the resctrl filesystem. The default group is the root directory which, immediately after mounting, owns all the tasks and cpus in the system and can make full use of all resources. The 'info' directory contains information about the enabled resources.

With RDT control enabled, user directories ("CG1", "CG2" in Figure 4: Intel® RDT hierarchy in resctrl filesystem) can be created in the root directory that specify different amounts of each resources. RDT control groups contain the following files: "tasks": Reading this file shows the list of all tasks that belong to this group. Writing a task id to the file will add a task to the group. "cpus": Reading this file shows a bitmask of the logical CPUs owned by this group. Writing a mask to this file will add and remove CPUs to/from this group. "schemata": A list of all the resources available to this group.

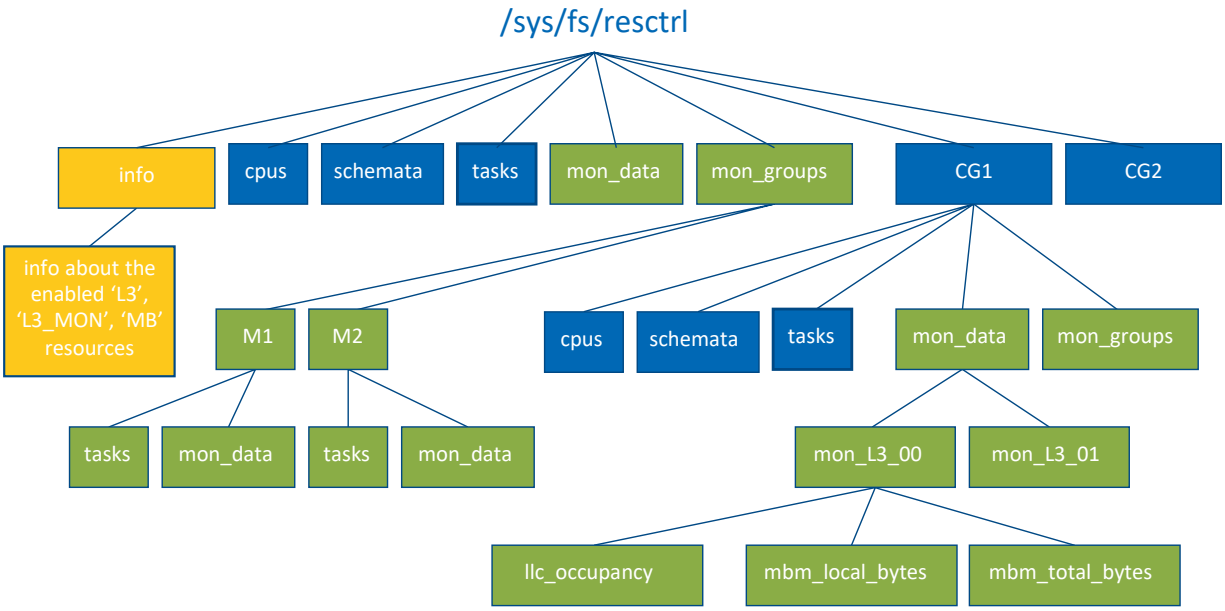

Figure 4. Intel® RDT monitoring and control in resctrl filesystem

With RDT monitoring enabled, the root directory and other top-level directories contain "mon_groups" directory in which user directories ("M1", "M2" in Figure 4: Intel® RDT hierarchy in resctrl filesystem) can be created to monitor group of tasks. "mon_data" directory contains a set of files organized by resource domain and RDT event. Each of these directories have one file per event ("llc_occupancy", "mbm_total_bytes", and "mbm_local_bytes"). These files provide a counter of the current value of the event for all tasks in the group.

Intel® Platform Resource Manager

Intel® Platform Resource Manager (Intel® PRM) is a suite of software packages to help you co-locate best-efforts jobs with latency-critical jobs on a node and in a cluster. The suite contains the following:

- Agent (eris agent) to monitor and control platform resources (CPU Cycle, Last Level Cache, Memory Bandwidth, etc.) on each node.
- Analysis tool (analyze tool) to build a model for platform resource contention detection.

Performance Modeling

A regression model is designed to model CPI and MPKI for an online workload. The model uses cycles and the total CPU utilization of the co-located offline workloads to build the models for CPI and MPKI.

CPI=f(CPU_cycles,Offline_workload_utilization)

MPKI=f(CPU_cycles,Offline_workload_utilization)

Here *f*(*,*) denotes a Gaussian Distribution.

The model is designed to explore the relationship between CPI/MPKI and influencing attributes. CPI and MPKI are related to workload intensity by nature. We add offline workload utilization as another attribute to the model in order to find any correlation among them. If under the same workload intensity, CPI is higher when offline workload utilization is higher, then most likely, the workload performance is impacted by offline workloads. If the MPKI has the same correlation, the impact is most likely from last-level-cache interferences.

A regression model is built for every online service from the same code base. The CPI and MPKI models are both built from seven-day runtime metrics. We split metrics twenty times to do model selection and randomly choose 500 samples to build the model. Two testing sets have been designed to check the correlation between offline workload utilization and CPI/MPKI. The first set is the combination of different CPU cycles with low offline workload utilization. The low offline workload utilization is sampled from the range lower than 10th percentile of the total offline workload utilization. Another testing set is sampled from different CPU cycles with higher offline workload utilization. The higher offline workload utilization samples are from the range greater than 90th percentile of the total offline utilization. Two testing sets are feed to the CPI/MPKI models to see whether the CPI or MPKI is higher when offline workloads utilization is higher.

We found for some online workloads, there was a correlation between CPI/MPKI and offline workload utilization. The result for one online workload is shown in Figure 5: Performance. modeling.result. It shows that with high offline workload utilization, the CPI and MPKI value is higher than that in the low utilization. As a result, we concluded that this workload suffered from cache interferences and its performance was impacted by the interference.

Intel® RDT metrics were also collected to validate the evaluation result. The cache occupancy for the same workload showed that when the intensity was high, the workload ran across the two NUMA domains. The offline workloads ran high in one of the NUMA domains, and competed got the last level cache occupancy in that domain against other online workloads. Given the analysis result and cache occupancy metrics, we can determine that cache regulation on offline workloads would be able to reduce the cache interferences of the co-located online workloads.

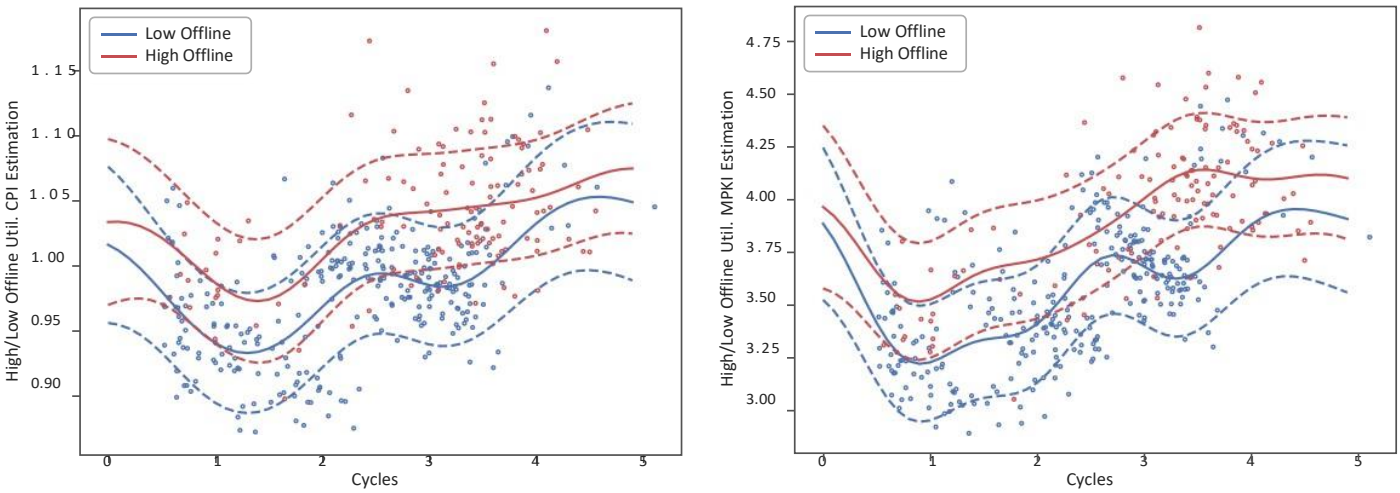

Figure 5. Performance.modeling.result of an online workload: Left: CPI Model with low/high offline workload CPU utilization. Right: MPKI Model with low/high offline workload CPU utilization.

Cluster Level Deployment and Result

According to the analysis result, it is believed that cache interference impacted the online workload performance in the co-located cluster. To mitigate the interference, ByteDance deployed the Intel® Resource Directory Technology to regulate the cache occupancy of offline workloads. Among two of the three workloads, the performance was improved.

RDT Configuration

For one server running both online and offline workloads, all the offline workloads were restricted to use two cache ways. Online workloads were able to use all the cache ways.

Evaluation Scope

A co-located cluster with more than 9,000 servers were deployed with the RDT configuration.

Evaluation Method

The fluctuation of the 99-percentile latency of an online workload was used to indicate the impact of cache regulation configuration. The fluctuation of the 99-percentile latency of a workload is defined as:

$$fluctuation_t = |99th\_latency_t - 99th\_latency_{t-1}| / 99th\_latency_t$$

It was compared in two scenarios:

1. When one workload ran in the co-located cluster without cache regulation configuration versus when it ran in the non-co-located cluster.
2. When the workload ran in the co-located cluster with cache regulation configuration versus when it ran in the non-co-located cluster.

The 99-percentile latency of the workloads were first collected from the co-located clusters without cache regulation and non-co-located clusters. After the cache regulation configuration was enabled in the co-located cluster, the metrics were collected again. The datasets of three online workloads were studied and each dataset covered 22 hours.

Evaluation Results

Two example workloads (workload A and workload B) showed positive results, compared to the non-co-located clusters, the fluctuation of the tail latency was greatly improved after cache regulation configuration was enforced.

Workload A

For Workload A, there were over 9000 instances in the co-located cluster and over 3000 instances in the non-co-located clusters. Without cache regulation, the fluctuation of tail latency was much higher in the co-located cluster than that in the non-co-located cluster (see Figure 6a: Workload A/Before). With cache regulation enabled in the co-located cluster, the fluctuation of tail latency became similar to that in the non-co-located cluster.

Workload B

The result of Workload B is also positive after cache regulation was enabled in the co-located cluster. This workload had more than 10,000 instances in the co-located cluster and more than 5,500 instances in the non-co-located cluster. Figure 7a: Workload B/Before shows the fluctuation of tail latency of this workload in the co-located cluster without cache regulation and that in the non-co-located cluster. It is obvious the fluctuation in the co-located cluster is much higher than that in the non-co-located cluster. Figure 7b: Workload B/After shows the same comparison after cache regulation was enabled in the co-located cluster. The difference of the tail latency fluctuation in the two clusters is no longer that dramatic.

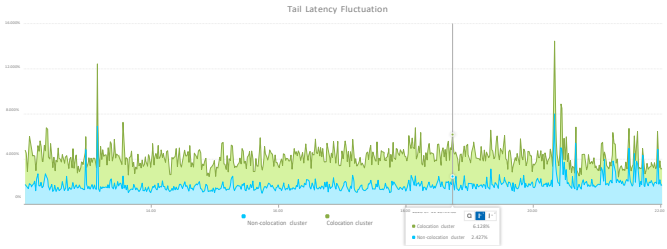

Figure 6a. Workload A/Before: The tail latency fluctuation of Workload A in the non-co-located cluster and the co-located cluster without cache regulation.

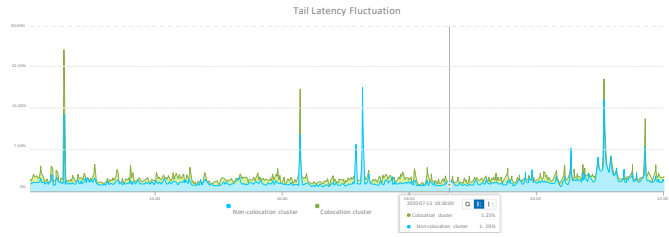

Figure 6b. Workload A/After: The tail latency fluctuation of workload A in the non-co-located cluster and the co-located cluster after cache regulation was enforced.

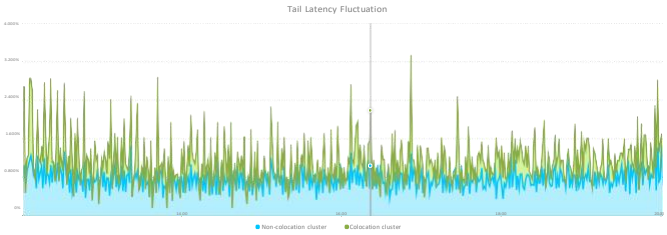

Figure 7a. Workload B/Before: The tail latency fluctuation of Workload B in the non-co-located cluster and in the co-located cluster without cache regulation.

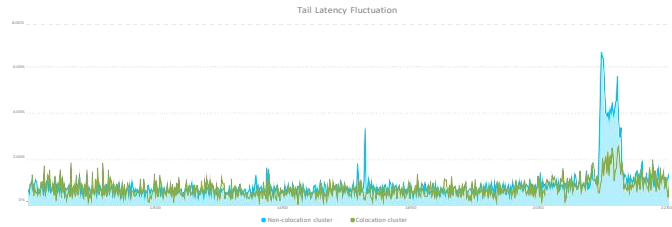

Figure 7b. Workload B/After: The tail latency fluctuation of Workload B in the non-co-located cluster and in the co-located cluster with cache regulation.

Conclusion

ByteDance used Intel® RDT and Intel® Platform Resource Manager to mitigate low-level hardware resource interferences for their co-location cluster. ByteDance had discovered that the performance of latency critical workloads degraded in the co-located cluster. A regression model was first designed to analyze whether the latency critical workloads were impacted by the co-located low priority jobs and how they were impacted. The analysis results showed that online workloads suffered from last-level-cache interferences. After cache regulation policy was enforced to the low priority jobs, the critical workload performance was recovered, comparable to the performance in the non-co- located cluster. This demonstrates how Intel® RDT and Intel® PRM can improve the latency critical workloads from low-level resource interferences for workload co-location, which in turn improves total server utilization in ByteDance co-located cluster on Intel® Xeon® platforms.

Where to Get More Information

For more information about Intel® RDT, refer to: https://www.intel.com/content/www/us/en/architecture-and-technology/resource-director-technology.html

For more information about Intel® RDT hierarchy in resctrl filesystem, refer to: https://www.kernel.org/doc/Documentation/x86/resctrl.rst

For more information on Intel® Platform Resource Manager, visit https://github.com/intel/platform-resource-manager or contact shen.zhou@intel.com.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
Your costs and results may vary.
Intel technologies may require enabled hardware, software or service activation.
Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
© 2021 Intel Corporation Printed in USA ♻ Please Recycle