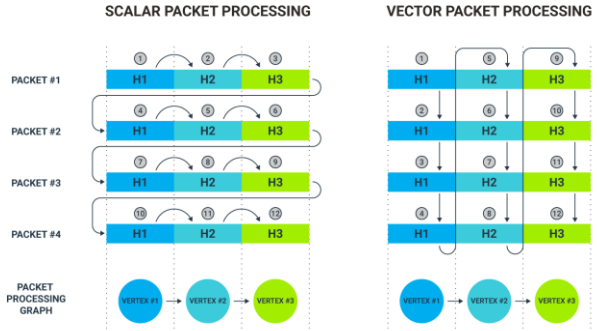
INTRODUCTION TO CLOUD SYSTEMS

Lecture 6 – Server telemetry collecting and processing – PMU, RDT, RAS, Collectors

Previous lecture

DPDK

DATA PLANE DEVELOPMENT KIT



Server telemetry

What is hardware telemetry?

Generally speaking, telemetry refers to a stream of data coming from any number of sources, including servers, software, and environmental equipment. For hardware telemetry, this data is generated by the components on your hardware platform, such as the CPU, memory, and PCIe interface. Registers on Intel® Xeon® Scalable processors monitor cache, CPU frequencies, memory bandwidth, and input/output (I/O) accesses. Hardware telemetry is generated from a robust set of model-specific registers (MSRs) and sensors on the Intel® platform.

Server telemetry

Why use telemetry?

Hardware telemetry can provide a detailed picture of exactly what is going on in all your servers and their components. This can help you find the root cause of a performance problem or degradation: whether the problem stems from the hardware (as in memory) or a workload imbalance on servers or components. These insights can speed and simplify problem resolution, freeing your time up for more interesting and more proactive work.

Telemetry monitoring can make data center management easier and more efficient. Because it shows you where the bottlenecks and overloads are occurring over time, you can fine-tune processes and orchestrate workload placement based on real-time workload profiling. This can help you protect your network performance and balance the load across your infrastructure.

Server telemetry

Telemetry can be used to check configurations and gain insights into resource utilization, power efficiency, and general system health. In a well-orchestrated data center, it can help you identify anomalies that might be early indicators of possible points of failure.

This can shift IT efforts from constantly reacting to problems to identifying—or even predicting—the first signs of a problem. Greater efficiency and fewer overloads can reduce the frequency of failures and improve key performance indicators (KPIs), including total cost of ownership (TCO), reliability, security, performance, and power consumption.

Server telemetry

What type of data should I collect with telemetry?

Keep in mind that when it comes to telemetry data, less can be more. The objective is not to collect the maximum amount of data, but to collect the right data in a useful amount. Collecting everything every few seconds can lead to enormous log files and, eventually, poor system performance. A best practice is to start with just your highest priority data and gradually add more streams. In addition, it's a good idea to scan data less frequently until an issue is identified, and then to increase the frequency until the issue is resolved. Table 1 provides guidance on the types of data to collect and how often to do so.

Server telemetry

What Data to Collect	Why to Collect It	How Often to Collect It
System configuration, kernel version, and log messages	Troubleshooting and failure-trend detection	On reboot
Power (CPU, memory, and system) and thermal (inlet, outlet, DIMM, and CPU airflow)	Manage power and thermal efficiency; identify spikes as early indicators of failure	30 seconds to 1 minute
Performance of the CPU, cache, and memory	Target machines for workload colocation	1–10 seconds
Memory errors	Identify issues causing device degradation and early indicators of failure	1–5 minutes Trigger more frequent scans on first uncorrectable memory error (UME)
DIMM performance and health	Identify issues causing device degradation and early indicators of failure	15–30 minutes and on startup
Throughput	Failure-trend detection	30 seconds to 1 minute

https://www.intel.com/content/www/us/en/cloud-computing/telemetry-monitoring-guide.html

PMU

The PMU is hardware built inside a processor to measure its performance parameters such as instruction cycles, cache hits, cache misses, branch misses, and many others. Performance monitoring events provide facilities to characterize the interaction between programmed sequences of instructions and microarchitectural sub-systems.

Performance monitoring events are actively used by performance profiling tools, e.g., the Intel® VTune™ Profiler, that provide event-based sampling microarchitecture analysis types to understand how effectively code uses hardware resources and recommend relevant optimization techniques.

The events listed are the performance monitoring events that can be monitored with the Intel® 64 or IA-32 processors. The ability to monitor performance events and the events that can be monitored in these processors are mostly model-specific, except for architectural performance events which are listed separately.

PMU

Intel processors already provide the capability to monitor performance events inside processors. In order to obtain a more precise picture of CPU resource utilization we rely on the dynamic data obtained from the so-called performance monitoring units (PMU) implemented in Intel's processors. We concentrate on the advanced feature set available in the current Intel® Xeon® 5500, 5600, 7500, E5, E7 and Core i7 processor series.

Implemented a basic set of routines with a high level interface that are callable from user C++ application and provide various CPU performance metrics in real-time. In contrast to other existing frameworks like PAPI and Linux, Intel support not only core but also uncore PMUs of Intel processors (including the recent Intel® Xeon® E7 processor series). The processor that contains the integrated memory controller and the Intel® QuickPath Interconnect to the other processors and the I/O hub. In total, the following metrics are supported:

- Core: instructions retired, elapsed core clock ticks, core frequency including Intel® Turbo boost technology, L2 cache hits and misses, L3 cache misses and hits (including or excluding snoops).
- Uncore: read bytes from memory controller(s), bytes written to memory controller(s), data traffic transferred by the Intel® QuickPath Interconnect links.

Intel® Resource Director Technology

Intel® RDT is an advanced resource monitoring and control feature set that is designed to improve visibility into, and control over, usage of shared platform resources such as memory bandwidth and LLC. It is featured on the Intel® Xeon® processor family as part of Intel's multi-year initiative to provide more insight into, and granular control over, shared resources. Future Intel® technologies will continue to advance noisy-neighbor detection and mitigation beyond what is currently available. Intel® RDT currently supports the following features:

- Cache Monitoring Technology (CMT) helps you find a noisy neighbor over-utilizing the last-level cache (LLC).
- Memory Bandwidth Monitoring (MBM) helps you find which noisy neighbor is using too much memory bandwidth.
- Cache Allocation Technology (CAT) allows you to quiet the noisy neighbor and allocate proper resources to apps or VMs that have higher priority.
- Code and Data Prioritization (CDP) enables you to give priority and protection to apps or VMs with large code or data footprints so that they don't have to contend for resources.

Intel® Resource Director Technology

Intel® Xeon® Processor E5 v4 Product Family Resource Director Technology (RDT)

Feature	Benefit	How Does it Work?
Cache Monitoring Technology (CMT)	Ability to monitor Last Level Cache occupancy for a set of threads	Each thread assigned a RMID (Resource Monitoring ID)
Cache Allocation Technology (CAT)	Ability to partition Last Level Cache, enforcement on a per thread basis Enables workload prioritization, consolidation, and resource partitioning Enables control over noisy neighbors	Each thread assigned a Class of Service Each Class of Service restricted to portion of LLC
Code and Data Prioritization (CDP)	A specialized extension of CAT which enables separate masks for code and data. This allows code to be protected at the L3 cache level for instance	Half of the masks are associated with code, the other half of the masks are associated with data
Memory Bandwidth Monitoring (MBM)	Monitors Memory Bandwidth utilization on an RMID basis. Identify memory bandwidth conflict issues and enable thread migration	RMIDs can be associated with one or a group of threads / applications

https://www.servethehome.com/intel-xeon-e5-2600-v4-broadwell-ep-launched/intel-xeon-e5-2600-v4-resource-director-technology/

RAS

Server reliability, availability, and serviceability (RAS) are crucial issues for modern enterprise IT shops that deliver mission-critical applications and services, and application delivery failures can be extremely costly per hour of system downtime.

RAS

RAS in relation to servers is defined as follows:

- Reliability – Reducing the mean time between hardware failures and ensuring data integrity. Data integrity is protected through error detection and correction — or, if not correctable, error containment
  - Error Detection and Self-Healing
  - Minimizes outage opportunities
  - Correct results continuously
- Availability – Refers to uninterrupted system and application operation even in the presence of uncorrectable errors
  - Reduce frequency and duration of outages
  - Self-diagnosing: work around faulty components or “self-heal”
  - Never stops or slows down
- Serviceability – Means a system can be maintained without disrupting operation. This capability requires both thoughtful platform design and innovative systems management.
  - Avoid repeat failures with accurate diagnostics
  - Concurrent repair on higher failure rate items
  - Easy to repair and upgrade

RAS

RAS in relation to servers is defined as follows:

- Reliability – Reducing the mean time between hardware failures and ensuring data integrity. Data integrity is protected through error detection and correction — or, if not correctable, error containment
  - Error Detection and Self-Healing
  - Minimizes outage opportunities
  - Correct results continuously
- Availability – Refers to uninterrupted system and application operation even in the presence of uncorrectable errors
  - Reduce frequency and duration of outages
  - Self-diagnosing: work around faulty components or “self-heal”
  - Never stops or slows down
- Serviceability – Means a system can be maintained without disrupting operation. This capability requires both thoughtful platform design and innovative systems management.
  - Avoid repeat failures with accurate diagnostics
  - Concurrent repair on higher failure rate items
  - Easy to repair and upgrade

RAS

RAS in relation to servers is defined as follows:

- Reliability – Reducing the mean time between hardware failures and ensuring data integrity. Data integrity is protected through error detection and correction — or, if not correctable, error containment
  - Error Detection and Self-Healing
  - Minimizes outage opportunities
  - Correct results continuously
- Availability – Refers to uninterrupted system and application operation even in the presence of uncorrectable errors
  - Reduce frequency and duration of outages
  - Self-diagnosing: work around faulty components or “self-heal”
  - Never stops or slows down
- Serviceability – Means a system can be maintained without disrupting operation. This capability requires both thoughtful platform design and innovative systems management.
  - Avoid repeat failures with accurate diagnostics
  - Concurrent repair on higher failure rate items
  - Easy to repair and upgrade

RAS

Intel® Xeon® Processor E7 Family: Reliability, Availability, and Serviceability

RAS

ADVANCED REDUNDANCY AND FAILOVER THROUGHOUT

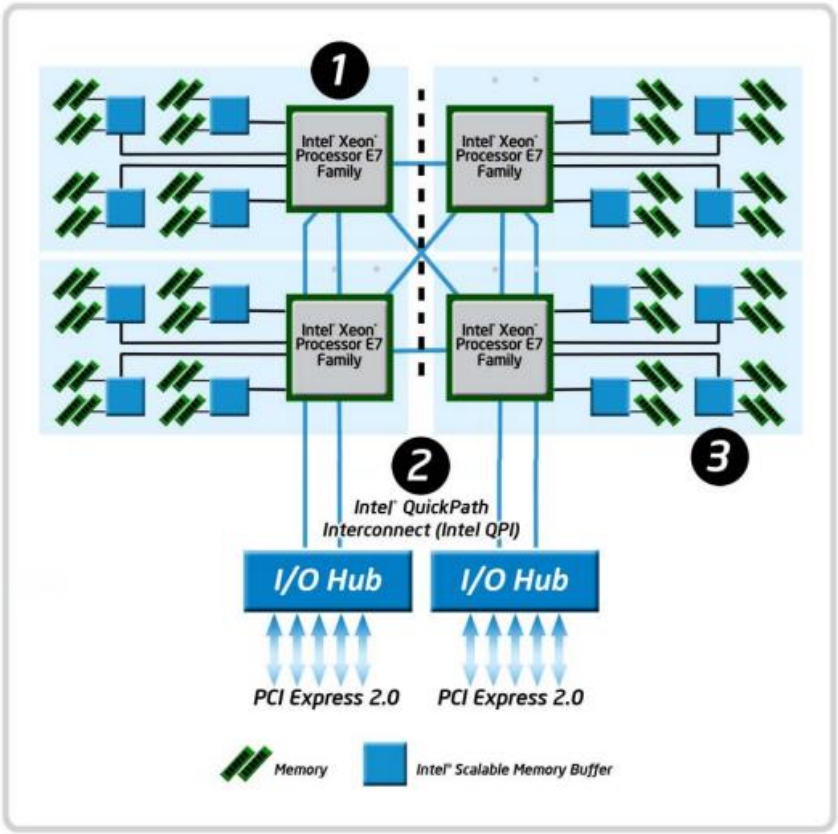
1 PROCESSOR/SYSTEM

- Corrupt Data Containment Mode
- Electronically Isolated Partitioning
- Processor Sparing and Migration*
- Core (Socket) Disable for Fault Resilient Boot
- Machine Check Architecture Recovery (MCA Recovery)*
- CPU Hot Add*
- PCIe Express Hot Plug
- Corrected Machine Check Interrupt (CMCI) for Preventive Failure Analysis*

*Requires operating system support.

2 INTEL® QPI

- Intel QPI Protocol Protection via CRC
- QPI Viral Mode
- Intel QPI Self-healing
- QPI Clock Failover
- QPI Packet Retry



3 MEMORY

- ECC
- Memory Address Parity Protection
- Memory Demand and Patrol Scrub
- Memory Thermal Throttling
- Enhanced DRAM Single Device Data Correction (SDDC)
- Enhanced DRAM Double Device Data Correction (DDDC+1)
- Fine Grained Memory Mirroring
- Memory Sparing
- Memory Migration
- Intel Scalable Memory Interconnect (SMI) Lane Failover
- Intel SMI Clock Failover
- Intel SMI Packet Retry
- Failed DIMM Identification
- Memory Hot Add*

*Requires operating system support.

RAS

Benefits for IT	RAS Silicon Features of the Intel Xeon Processor E7 Family
Protects Data	
• Reduces circuit-level errors • Detects data errors across the system • Limits the impact of errors	Error Correction Code (ECC)
	Memory Address Parity Protection
	Intel® QuickPath Interconnect (Intel® QPI) protocol protection via Cyclic Redundancy
	Memory Demand and Patrol Scrub
	QPI Viral Mode
	Corrupt Data Containment Mode
	Electronically Isolated Partitioning
Increases Availability	
• Heals failing connections • Supports redundancy and failover for key system components • Recovers from uncorrected data errors	Memory Thermal Throttling
	Single Device Data Correction and Enhanced DRAM Double Device Data Correction (DDDC)
	Fine Grained Memory Mirroring
	Memory Sparing
	Memory Migration
	Intel Scalable Memory Interconnect (SMI) Lane Failover
	Intel SMI Clock Failover
	Intel SMI Packet Retry
	Processor Sparing and Migration
	Socket Disable for Fault Resilient Boot
	Intel QPI Self-healing
	Intel QPI Clock Failover
	Intel QPI Packet Retry
	Machine Check Architecture (MCA) recovery
Minimizes Planned Downtime	
Helps IT • Predict failures before they happen • Maintain partitions instead of systems • Proactively replace failing components	Failed DIMM Identification
	CPU Hot Add
	Memory Hot Add
	PCIe Express Hot Plug
	Electronically Isolated Partitioning
	Corrected Machine Check Interrupt (CMCI) for Preventive Failure Analysis

Intel® Telemetry Collector

ITC is a reference collector that provides a quick introduction to the different metrics available, including power and thermal statistics, performance counters, process activities, threads, and operating-system-level disk, network, and memory statistics. With ITC, you can ingest and visualize data from various sources and multiple machines.