SoC Infra HC - 4/27/2022

RISC-V®

Disclosures

Agenda

- Disclosures
- RAS SIG charter and proposed task groups
- QoS SIG charter and gap analysis

Proposed RAS SIG Charter & Next Steps

“The objective of the RAS SIG is to identify both HW and SW high priority gaps within the RISC-V architecture relating to RAS (Reliability, Availability, and Serviceability) in all segments of the RISC-V ecosystem. In conducting that work, it is expected that the RAS SIG will spin up TGs to prescribe both HW and SW interfaces/conventions that solutions can be built upon. Likewise, a TG is expected to be formed to specify common terminology that the associated RAS specifications can leverage so that they are all using common language and concepts. It is anticipated that the RAS SIG, under the SoC infra HC, will work closely with other SIGs, TGs, and HCs to ensure the needs of the various groups are met. The work of the TGs will likely include both ISA and non-ISA architectural specifications and definitions depending on where the RISC-V RAS community would like to see improvements and commonality driven.” Link

Call for Chairs Forthcoming Imminently

TGs to be created based on community desires:

- RAS Reporting Register Interface TG
- RAS SW Abstraction API TG
- RAS Terminology TG

Proposed QoS SIG charter

Quality of Service (QoS) SIG objective is to identify high priority gaps in the RISC-V technology portfolio relating to QoS in all segments of the RISC-V ecosystem.

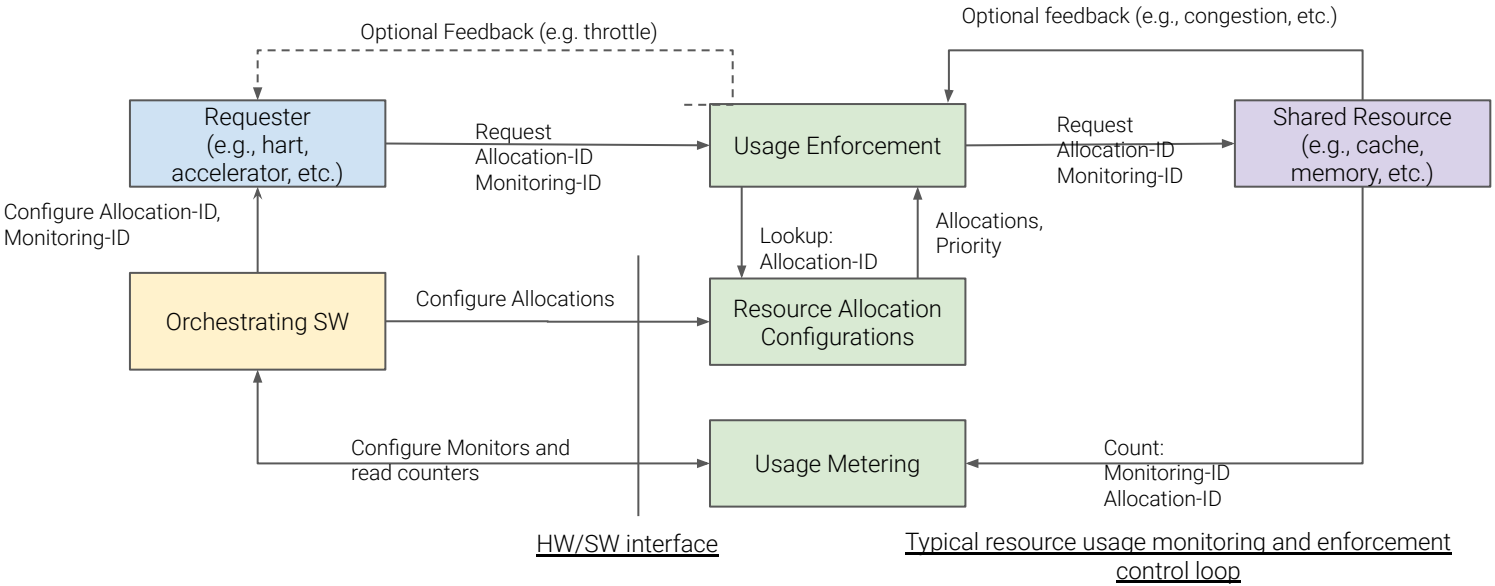
Quality of Service (QoS) is the minimal end-to-end performance that is guaranteed in advance by a service level agreement (SLA) to an application as measured through metrics such as instructions per cycle (IPC), latency of servicing work, etc. When multiple applications run concurrently on modern processors with large core counts, multiple cache hierarchies, and multiple memory controllers, the performance of an application may become less deterministic or even non-deterministic as the performance depends on the behavior of all the other applications in the machine that contend for the shared resources leading to unfair and/or unexpected performance impacts to the application.

The QoS SIG objective is to define the strategy to address the desire for achieving deterministic performance by minimizing the interference caused by contention for shared resources. The QoS SIG will work on a gap analysis to identify RISC-V technology gaps and define the strategy and priority to address the identified gaps. In conducting that work, it is expected that the QoS SIG will recommend formation of TGs to define the HW and SW interfaces in the form of non-ISA and/or ISA extensions that solutions can be built upon.

The QoS SIG, under the SoC infra HC, will work closely with other SIGs, TGs, and HCs to develop the necessary hardware and software technologies.

Agree to charter?

QoS technologies - first cut gap analysis



- Gap Analysis
- Key gaps
  - Cache capacity usage monitoring and allocation
  - Memory bandwidth usage monitoring and allocation

backup

Quality Of Service

QoS

Quality of Service (QoS)
- Minimal end-to-end performance that is guaranteed in advance by a service level agreement (SLA) to an application.
- Measured in the form of metrics such as instructions per cycle (IPC), latency of servicing work, etc.

Multiple applications are running concurrently on modern processors
- Large core counts, multiple cache hierarchies, and multiple memory controllers
- Performance becomes less deterministic or even non-deterministic

Factors affect performance
- Available cache capacity, memory bandwidth, interconnect bandwidth, CPU cycles, system memory, etc.
- Priority of arbitration required for shared resources

System software can control some of these resources available to the application
- E.g., number of hardware threads made available for execution, the amount of system memory allocated to the applications, the number of CPU cycles provided for execution, etc.

System software needs additional tools to control interference to an application due to sharing of
- Cache capacity usage, memory bandwidth usage, power, interconnect bandwidth usage, etc.

Sources of interference - cache and memory contention

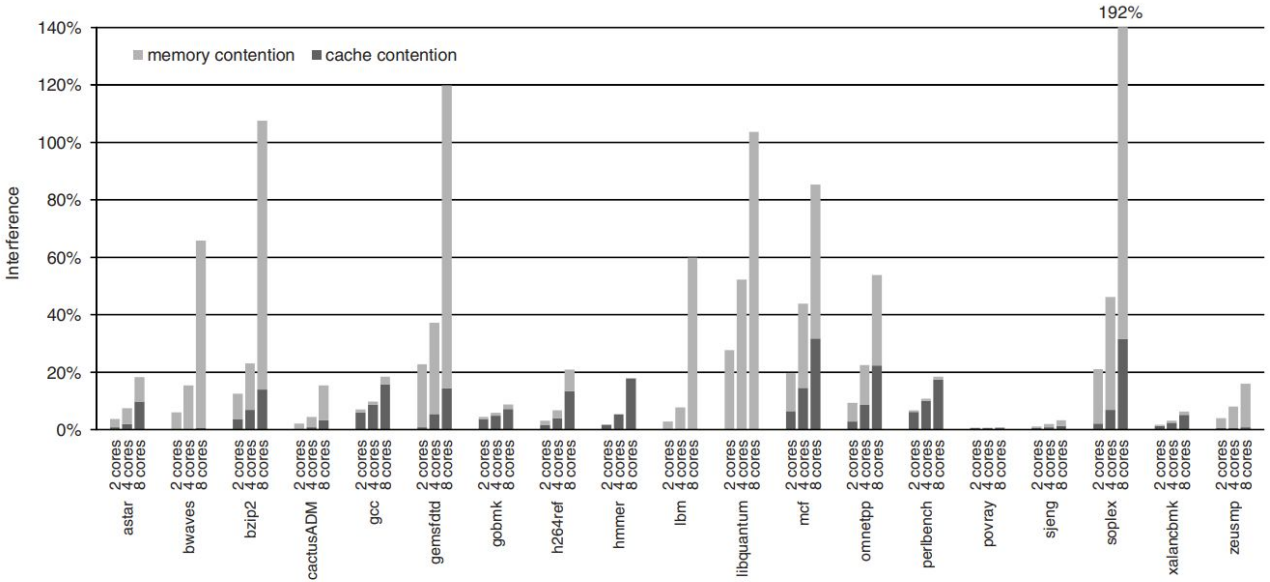


Fig. 1. Impact of inter-thread interference on per-thread performance for 2, 4, and 8 cores, breaking up interference in cache versus memory contention (average interference is reported across a set of job mixes per benchmark and assuming hardware prefetching).

- Per-Thread Cycle Accounting in Multicore Processors
  https://dl.acm.org/doi/pdf/10.1145/2400682.2400688

Cache, DRAM B/W, Power, SMT, network interference

websearch

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
LLC (small)	134%	103%	96%	96%	109%	102%	100%	96%	96%	104%	99%	100%	101%	100%	104%	103%	104%	103%	99%
LLC (med)	152%	106%	99%	99%	116%	111%	109%	103%	105%	116%	109%	108%	107%	110%	123%	125%	114%	111%	101%
LLC (big)	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	264%	222%	123%	102%
DRAM	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	270%	228%	122%	103%
HyperThread	81%	109%	106%	106%	104%	113%	106%	114%	113%	105%	114%	117%	118%	119%	122%	136%	>300%	>300%	>300%
CPU power	190%	124%	110%	107%	134%	115%	106%	108%	102%	114%	107%	105%	104%	101%	105%	100%	98%	99%	97%
Network	35%	35%	36%	36%	36%	36%	36%	37%	37%	38%	39%	41%	44%	48%	51%	55%	58%	64%	95%
brain	158%	165%	157%	173%	160%	168%	180%	230%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%

ml_cluster

	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
LLC (small)	101%	88%	99%	84%	91%	110%	96%	93%	100%	216%	117%	106%	119%	105%	182%	206%	109%	202%	203%
LLC (med)	98%	88%	102%	91%	112%	115%	105%	104%	111%	>300%	282%	212%	237%	220%	220%	212%	215%	205%	201%
LLC (big)	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	276%	250%	223%	214%	206%
DRAM	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	287%	230%	223%	211%
HyperThread	113%	109%	110%	111%	104%	100%	97%	107%	111%	112%	114%	114%	114%	119%	121%	130%	259%	262%	262%
CPU power	112%	101%	97%	89%	91%	86%	89%	90%	89%	92%	91%	90%	89%	89%	90%	92%	94%	97%	106%
Network	57%	56%	58%	60%	58%	58%	58%	58%	59%	59%	59%	59%	59%	63%	63%	67%	76%	89%	113%
brain	151%	149%	174%	189%	193%	202%	209%	217%	225%	239%	>300%	>300%	279%	>300%	>300%	>300%	>300%	>300%	>300%

memkeyval

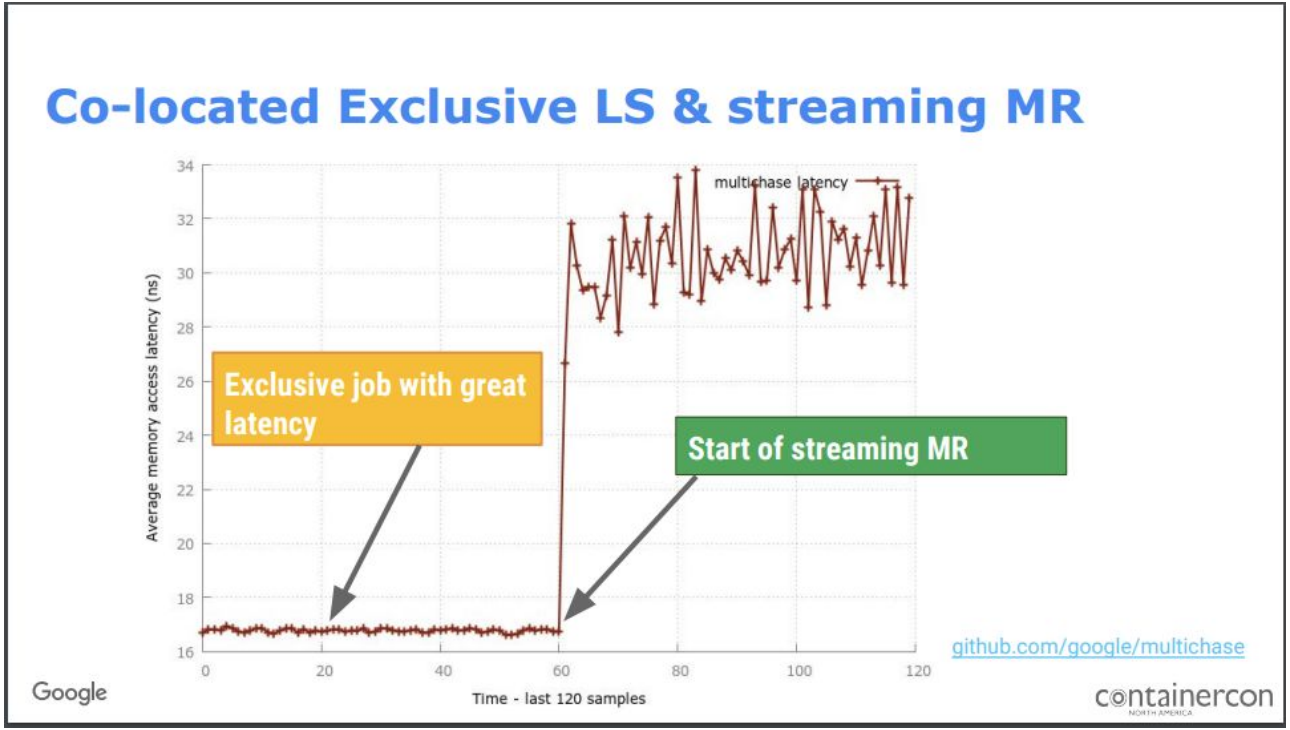
	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
LLC (small)	115%	88%	88%	91%	99%	101%	79%	91%	97%	101%	135%	138%	148%	140%	134%	150%	114%	78%	70%
LLC (med)	209%	148%	159%	107%	207%	119%	96%	108%	117%	138%	170%	230%	182%	181%	167%	162%	144%	100%	104%
LLC (big)	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	280%	225%	222%	170%	79%	85%
DRAM	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	252%	234%	199%	103%	100%
HyperThread	26%	31%	32%	32%	32%	32%	33%	35%	39%	43%	48%	51%	56%	62%	81%	119%	116%	153%	>300%
CPU power	192%	277%	237%	294%	>300%	>300%	219%	>300%	292%	224%	>300%	252%	227%	193%	163%	167%	122%	82%	123%
Network	27%	28%	28%	29%	29%	27%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%
brain	197%	232%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%	>300%

Each entry is color-coded as follows: 140% is ≥120%, 110% is between 100% and 120%, and 65% is ≤100%.

Figure 1. Impact of interference on shared resources on websearch, ml_cluster, and memkeyval. Each row is an antagonist and each column is a load point for the workload. The values are latencies, normalized to the SLO latency.

Heracles: Improving Resource Efficiency at Scale
http://csl.stanford.edu/~christos/publications/2015.heracles.isca.pdf

Cache interference - latency impact

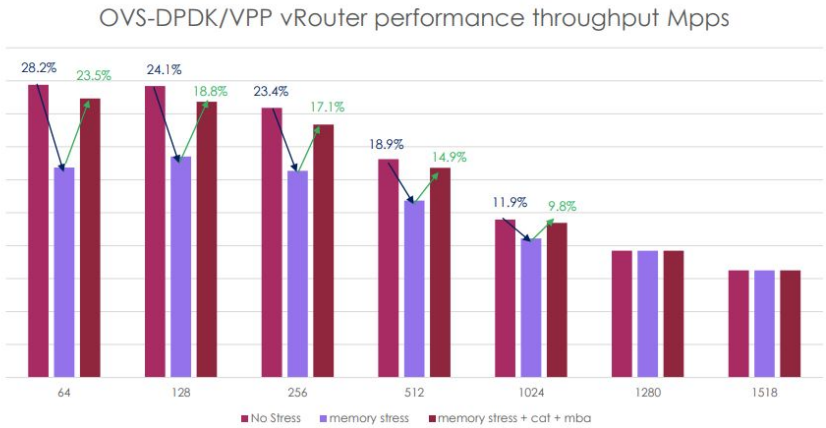


CAT @ Scale Deploying cache isolation in a mixed-workload environment at Google
https://static.sched.com/hosted_files/lcccna2016/a7/CAT%20%40%20Scale.pdf

Cache and Memory Interference

Performance data with RDT

DPDK
DATA PLANE DEVELOPMENT KIT



13

Achieve stable high performance DPDK Application on modern CPU
https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/01-DPDK_on_multicore_CPU_100.pdf

Further reading

- "Performance evaluation of cache allocation technology for NFV noisy neighbor mitigation": https://ieeexplore.ieee.org/document/8004214
- "An Integrated Instrumentation and Insights Framework for Holistic 5G Slice Assurance": https://ieeexplore.ieee.org/document/9165431
- “Resource Tuning for Energy Efficient Slicing”: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9385531
- “Resource Allocation: Intel Resource Director Technology (RDT) - LinuxCon 2016” https://static.sched.com/hosted_files/lcccna2016/59/cat.pdf

Next steps

- Multiple studies have shown that Cache Capacity and Memory Bandwidth lead to highest interference
- Enforcing QoS on RISC-V SoC needs
  - Associating a workload with an quality of service identifier
    - For capacity allocation, monitoring, and prioritization
  - Establishing capacity allocations for workloads in shared resources - e.g. cache controllers, memory controllers, etc.
  - Monitoring capacity usage by a workload
- Agree if we need an extension
  - If we agree then SIG or TG? Is the problem well understood?