

Sep 19, 2024 | 📅 RV Perf Analysis SIG

Attendees: Beeman Strong tech.meetings@riscv.org Snehasish Kumar Brian Coutinho
Zachary Jones Louis Feng

Notes

- **Attendees:** Beeman, Snehasish, Brian Coutinho (Meta), Zach Jones (Meta), PhillipR, MattT, Louis, MarkR, DanielB, ChunL, DanielP, RobertC, scrossley, Dmitriy, VictorL, Bruce, RajeevI
- **Slides/video** [here](#)
- **Dynolog talk** (Meta, Brian C & ZachJ)
- Lightweight monitoring daemon
- Example: FE_BOUND, try huge code pages, see drop in ITLB misses
- Always on performance monitor
 - CPU monitoring (events from Intel, AMD, ARM)
 - GPU monitoring for NVDA with DCGM
- Can optionally request deeper analysis, with other events, or higher sampling rate
- Is this push based?
 - Yes, always on pushes metrics to database
 - What are the always on metrics?
 - ~30 metrics, split between CPU and GPU. Things like CPU cycles, uptime, network BW, ...
 - Collected every 60s, configurable
 - Can add TLB/I\$ misses, etc
- Can output to various sinks: Prometheus, ODS, Scuba
- Started 10+ years ago, open-sourced in 2022
 - 30-40% of features open now, expect a few more
 - Netflix uses it too
- Is there automated analysis applied to this profiling data?
 - Lots of analytics built on top of this profiling data, in addition to human analysis
 - E.g., resource accounting, how efficiently are resources used by given users
- What is the overhead?
 - Daemon has ~5% overhead on a single core
 - Application/system overhead kept as low as possible, tune sample rate to keep it low (every 1s-10s)
 - Use BPF
- Only used in servers, or embedded too?
 - Smallest devices are 4-8 cores
 - No client/mobile
- CPU PMU events used to monitor resource utilization, identify bottlenecks
 - Hetero trend means code needs constant optimization
- More often issues are born of managing data as it moves through the system, not just in the CPU

- Rings to meshes to tiles, getting more complicated
 - Hoping RV will be less chaotic, but for high perf/AI hard to avoid
- Perf provides a standard ifc for HW events, SW events, tracepoints
 - But more and more other users are requesting perf events
- Added Heartbeat (hbt) to manage perf events, system-wide and per-cgroup
 - Counting mode only
 - Many sources of event info, not consistent. Some events are only in kernel json files, some only on github, some in both
- Per-cgroup monitoring is expensive, using BPF to share events per cgroup helps
 - Avoids context switch overhead, compute event deltas on the fly instead
- Hbt is newly open-sourced
- When x86 core changes, the uncore changes. On ARM, a core could be within multiple uncore configs.
 - RISC-V likely the same
 - A standard way to collect performance event data across SoC configurations/vendors would be helpful
 - How to discover uncore configuration? No CPUID equivalent
- AI monitoring
 - Only support NVDA, counters specific to them, through DCGM
 - Metrics include device util, SMs active, SM occupancy, ... collected every 10s
 - DCGM collects internally at a higher rate, but only output every 10s. Don't know how it is implemented.
 - At this rate overhead is negligible
 - Includes data on Tensor cores, NVLink, PCIe, ...
 - Always running on all GPUs
 - PyTorch provides CPU + GPU flamegraphs. Does have 3-5% overhead, so only collect this infrequently
 - How do you sync clocks between CPU & GPU?
 - Done by PyTorch
 - 10s of us granularity
 - Understanding network delays can be measured, but have to dig deeper
 - Inter-node is harder
 - Currently don't map kernels back to operators, not enough granularity
 - PyTorch collects call-stack info
- Under development
 - Looking at t-digests to allow higher sampling rates without having to export more data
 - Maybe up to 100ms sampling rate, to keep overhead low. Would like 10-25ms. This is for 1 metric.
 - Some metrics require several counters at once
 - Use QoS to limit noisy neighbors
 - Use monitoring to ensure groups are sized properly
 - Would be nice if there was a shared kernel resource control interface for the various different QoS architectures

- Right now x86-specific, ARM trying to generalize
- Use Dynolog also for process profiling/sampling?
 - Use other tools, StrobeLight is one (also open-source), focused on CPU
 - Includes BPF script to support some GPU profiling
 - Hoping NVDA will expose some events to BPF, no luck yet
 - Can operate on a single GPU, but not fleet-wide
 - GPU profiling depends on GPU (CUDA) drivers
- When did you decide to open source Dynolog, and how did you convince management?
 - In 2022
 - Their AI research clusters required open-source
 - Easier to use open-source in lab environment
- How do you use Intel PT?
 - Engineer left, not much progress since then
 - Hbt has PT setup
 - One use case: Time-travel Debug, could do it with LLDB
 - Another: fine-grained latency analysis

Action items

- ☐ Beeman Strong - Jul 28, 2022 - Reach out about proprietary performance analysis tools