Feb 8, 2024 | RISC-V Perf Analysis SIG Meeting

Attendees: Beeman Strong tech.meetings@riscv.org Marc Casas

Notes

- **Attendees**: MattT, Nick Romero, Ron Minnich, Ved, Beeman, Manisha, Chun, RobertC, BruceA, Snehasish, Atish, GregF, Dmitriy, QuanN
- **Slides/video** here
- Intros: Nick, Ron, and Manisha work on perf analysis at Samsung - Systems Arch Lab (SAL) Team
- **HW Counter Usage in HPC Application Analysis** talk by Matt Turner
    - SAL aims to break through memory/fabric walls to get more bytes/flop
    - Use counters heavily, need them to be accurate
    - Don't want traps to firmware to read, for perf reasons
    - Talk is from user perspective, not HW/implementation perspective
    - Roofline analysis: plots peak perf in Gflops/s and DRAM GB/s. Mem bound if capped by the latter, compute bound if capped by the former.
        - Can be some overlap, where mem and compute bound
        - Can further breakdown compute bound, e.g. by vector vs scalar vs FMA vs integer vs divides vs branch (incl mispreds), etc
            - FLOPS need to be broken down by precision (single, double, etc)
            - Should be non-spec (retired) counts
        - Or mem bound, e.g. looking at peak BW to L1, L2, DRAM (local vs remote), …
    - How to count RISC-V vector ops?
        - Preferable to get actual number of FP ops (unmasked, factor LMUL), but inst count is better than nothing. Allows seeing trends.
    - For branches, mostly looking at conditionals
    - Does HPC have much bad speculation? Intel paper on spec-cpu has HPC kernels, see some bad spec (10-40%) there
    - For integer, don't care about incrementing loop index, but may want calculation of array index
    - For mem & compute bound, often stalled by something else. Look at pipeline stalls, using Intel's top-down (TMA). Similar analysis capabilities in AMD & ARM cores.
        - Concerned that RISC-V may want some noise introduced to counters, for security reasons (fuzzing counters)
            - This was related to cycle and instret user space access discussion, was just an idea, not pursued. Instead provide access to those through perf.
        - Need events for each box. Would be nice to have dedicated counters for each, but not realistic. Can multiplex events, or rerun (depending on tool support for multiplexing).

        - Perf Events TG will be looking at some similar structured analysis for RISC-V
    - HPC systems use lots of power, 10s of MW, akin to the energy to run a small town
        - Would like counters that indicate energy used by an application, ideally broken down by memory access, network, CPU, etc
        - Then can optimize on perf/watt
        - Intel's RAPL framework gives energy consumption per run, which is good, but would prefer finer-grained data
    - Tools: TAU, HPCToolKit, PAPI, Likwid
        - The latter abstract events to make it easy for non-HW users to understand, use perf under the hood
        - Typically 100s of samples/sec, but sometimes need much deeper (<10 cycles)
            - Ideally want a very fast user mode counter read instr
                - May be able to tolerate speculative reads
            - Trace is another option for low-level latency info
    - Would like to be able to detect false sharing, don't see much of that in HPC yet
        - People tend to blame many perf issues on false sharing, without data
        - ARM has a paper on false sharing detection using SPE
        - Usually use indication of remote HITMs, indicates some issue with sharing, though may not be false
    - Supervisor Counter Delegation extension should avoid traps to firmware to write counters, will be ratified soon
        - Also there were issues on old HW with RDCYCLE that was very slow
    - Does HPC include AI/ML, with all the FP16/FP8 usage?
        - HPC is starting to use those more, does include AI/ML algorithms

Action items

- [ ] Beeman Strong - Jul 28, 2022 - Reach out about proprietary performance analysis tools
- [ ] Beeman Strong - Jul 28, 2022 - Reach out to VMware about PMU enabling