Mar 21, 2024 | RISC-V Perf Analysis SIG Meeting

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

Notes

- **Attendees**: Beeman, Bruce, Dmitriy, MattT, Snehasish, Ved, Atish
- **Slides/video** here
- Updates
    - CTR ARC-approved, working on freeze items
    - Smcdeleg/Ssccfg going to TSC for ratification vote this week
- Sampling proposal recap
    - Event sampling adds CSR with precise PC indication, interrupt still skids
        - Skid could be minimized if the implementation knows the counter is about to overflow
        - Could be per-event, such that some events indicate no skid. To be considered by Events TG
    - Instruction/Dispatch sampling selects inst in FE, collects sample metadata during execution, applies filters, stores sample data when inst is qualified
        - Can use these two together, e.g. for LLC miss, such that all samples have PC but some have data address, latency, etc.
        - How do we understand the sample population we got with instr sampling? How to tell what portion of retired insts were captured?
            - Could add events to count marked instrs, marked instrs cleared, marked instrs filtered out, marked instrs sampled
            - May sometimes want to capture speculative ops, to understand what's happening in the shadow (e.g., cache pollution)
        - Can increase sample rate if filtering for rare events, or maybe immediately re-arm when sample filtered out
        - Could have training in FE, to more frequently select PCs that hit event of interest
            - Feedback doesn't have to be immediate, even periodic
        - Could stream sample data out to trace. Will discuss as part of output topic (next)
        - Should there be a PC range? Is there a value to justify the HW cost? Seems not required
- Sample output
    - Could store sample data out to memory, but SW likely to still want LCOFI to get call-stack
    - In that case could instead record sample data to CSRs, and allow SW to collect it. But adds ISA state, needs context switch.
    - Could context switch just clear the sample data CSRs?
        - Would lose samples if switch occurs between sample writes and LCOFI. Should be rare.

        - Could leave it to kernel to either clear samples, or if sample is pending collect it then clear
            - On swap_out, if LCOFI pending (sip.LCOFI=1) then could collect the sample data and clear the pending interrupt. Then never need to restore sample state on swap_in.
            - CSR read has some impact on context switch
            - Might need an inst to clear sample state, like SCTRCLR
    - For memory option, have to deal with page faults
        - Sample data must reside in HW while fault is serviced. SW needs a way to flush it out once page is in place.
        - Also need some way to know when flush is complete, so it can be collected
        - Could avoid page faults by using phys memory or IOMMU, but not virtualization-friendly
- Out of time, will continue output discussion on mailing list, or next meeting

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