Mar 7, 2024 | RISC-V Perf Analysis SIG Meeting

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

Notes

- **Attendees**: Beeman, Snehasish, MattT, Kurush, Dmitriy, Bruce, Greg, Robert, Ved
- **Slides/video** here (started video after opens, sorry)
- Intro: Kurush - perf architect from MIPS
- Opens
    - Latest on CTR: went with sctrdepth register, where accesses from VS-mode trap. As long as accesses are rare these traps won't be costly.
    - CTR ARC review approved this week! Minutes coming soon
    - CTR has Qemu & Spike patches (thanks Rajnesh Kanwal !), will upstream once frozen
        - Patches, based on spec v0.8.1:
            - https://github.com/rivosinc/linux/tree/control_transfer_records_v1
            - https://github.com/rivosinc/qemu/tree/control_transfer_records_v1
        - Can send RFC now, to get feedback
        - Good to get stuff upstream sooner than ratification, though some reviewers have different preferences
        - Linux tends to be more restrictive (freeze), want changes to be very unlikely
        - POC is different from SW enablement, fewer requirements, just enough to prove the capability
        - Would be nice to have links shared publicly
            - The Jira might be a good option
- Sampling
    - Reviewed some takeaways from the industry talks
        - Call-stacks, precise attribution, and lack of bias important
    - (Re-)reviewed tradeoffs of instruction sampling vs precise event sampling
        - Inst sampling is easier to be precise, and easier to collect run-time metadata. Event sampling is better for identifying sources of rare events, predictable sample rate.
        - How does this relate to including event counts in CTR?
            - CTR records all "events" (control transfers), whereas this is statistically sampling on events
            - But similar in that both log some metadata about the instructions that are sampled
    - Proposed option to log precise sample PC on counter overflow, for select (non-spec) events
        - Then even with LCOFI skid have precise attribution
        - But other state (GPRs, etc) is impacted by skid

        - With CTR it might be possible to trace the execution path from the sample PC to the epc (the skid path)
        - Would be a simple extension to Sscofpmf capabilities
        - Believe this gets much but not all of the benefits of a true precise sampling mechanism (no skid)
            - If nothing else it's an improvement on what we have now
    - Proposed also to add an inst sampling extension
        - Akin to, but believe better than, what we see with SPE and IBS
        - E.g., filter at source (dispatch) and retire, to improve quality of sample population
        - If the sampled instruction is dropped, maybe should choose again quickly. Could recover the dispatch counter on clear?
            - This would address some issues with sample rate unpredictability
        - Could instr sampling have a predictor? When an inst gets an event of interest, bias pick towards that PC. Would need research.
    - Out of time, will continue next time and on email list

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