Sep 5, 2024 | RV Performance Event Sampling TG

Attendees: Beeman Strong, Snehasish Kumar, tech.meetings@riscv.org

Notes

- **Attendees**: Beeman, Snehasish, Atish, Chun, Dmitriy, RobertC
- **Slides/videos** here
- Reviewed output options discussed in meeting and online
    - What sample rate assumed in perf analysis? Every 100us
    - What is typical sample rate and data rate?
        - For datacenter usage, export data in perf.data format. Sample every 1M-10M retired branch. Perf.data has much more than just the sample data, probably 1MB-10MB per sample.
        - Duration of profiling also matters. 10s of seconds of profiling on a system at a time. Sample systems, don't land on every one.
        - Choose sampling rate to control overhead.
    - With memory buffer, could have threshold set to <= 1 record to get interrupt per sample
        - Agreed but think we may be better off with CSR method for this usage, reason is more implementation flexibility, on upcoming slides
- Proposed solution
    - CSR with interrupt per sample as base capability
        - Simplest implementation, supports all usage models, including emerging future usages.
        - But slow
    - Include optional extension to route sample data to memory buffer
        - For usages with >>1 sample per interrupt
        - Includes non-precise interrupt on buffer threshold met
        - Includes new fence instr, to ensure any buffered sample data is emitted
    - This memory buffer proposal allows for multiple implementations
        - Could use precise micro-trap, like PEBS, which interrupts SW execution
        - Could also implement more like trace, where trace data is emitted slowly over time
            - For this reason, threshold interrupt might skid, and need fence to ensure all data is flushed before processing
- Why new instr? Fence could take indefinite time.
    - This is akin to trace flush+poll
    - CTR originally proposed a CSR bit to clear, and ARC said no, add a new instr instead. Seems they don't like CSR bits with side-effects, prefer insts.
- This output to memory buffer is a lot like trace, should this optimized use case just use trace?
    - With CSR as base capability, it means that instr sampling would not be wholly dependent on trace, just optimized usage

    - Would everyone who implements optimized output also implement self-hosted trace?
        - In Chromebooks, the answer has been no. Some vendors seem to treat trace as a debug feature.
        - Also adds some security considerations, often not exposed in datacenter
        - Looking at industry, Intel has self-hosted trace but AMD does not. ARM technically does (recently), but do Ampere, Graviton, etc implement it? Unknown.
        - Agreed that don't want the optimized usage to depend on trace
- Why context switch the sample data CSRs?
    - With precise interrupt they should be consumed as soon as they are populated
    - Save/restore then only valuable if we got some higher priority interrupt pended in tight window between CSR update and precise interrupt
    - Depends on how SW uses it, SSE can't be pre-empted
    - Can probably discard the sample in such a rare corner case
    - Even clearing them has some cost. Do we need a new inst, like SCTRCLR? TBD.
- Reviewed order of instr sampling topics to discuss going forward
- Discussed plan to pivot to precise event sampling next, since it is simpler and we can close on it and start getting feedback while we work through the instr sampling details

Action items

- ☑ ~~Aug 1, 2024~~ ~~Beeman Strong~~ ~~Propose method for evaluating tradeoffs of CSR vs memory buffer~~
    - Sent email with perf overhead estimate methodology