Oct 20, 2022 | RISC-V Perf Analysis SIG Meeting

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

Notes

- **Attendees**: Beeman, Bruce, RobertC, Eric Lin, JohnS, StephaneE, Atish, Greg
- **Slides/video** here (slides will be added once I receive them)
- **Agenda**:
    - Branch Sampling (aka Control Transfer History) with guest speaker Stephane Eranian
- Stephane works in Google technical infrastructure, builds datacenter HW, OS, infra, including system SW. Works in Linux kernel team, responsible for performance monitoring in the kernel, and the low-level tools for collection (perf).
    - Influences HW vendors on PMU additions to meet Google's needs
- Google uses HW counters heavily, depend on them, used on the fly to guide run-time decisions, identify bottlenecks, feed into compiler, influence future purchase decisions
- Branch sampling required of any new HW platform
- Overall profiling overhead should be no more than 2%, branch sampling should have no slowdown during execution, only once sample is collected
- Useful for identifying where to insert SW prefetches (e.g., on path to L3 miss), direct inlining decisions, …
- FDO was a 2-pass profiling mechanism, autoFDO allows single-pass. ~10% speedup on avg for Google
    - No extra effort from developer, no benchmark required
    - Reduces insts per branch
- FDO requires hot basic block info (from branch sampling) + control flow graph
    - If have 16 entries, get ~15 basic blocks
    - BB definition is dynamic, bc don't know from code which branches are taken
- Must capture all control flow transfers, with src VA, dest VA, mispred bit, cycle latency
    - Optional: brtype, spec, priv mode
    - Mispred and latency used for prefetch placement, not autofdo
- Google collects samples system wide, SW tags them with process, and may filter entries after the fact
    - Look at all apps running on the fleet
- For PL filtering, generally don't see system call but do see return
    - Though this is asymmetric
- Can do brtype filtering in SW, but wastes the HW (end up dropping precious entries)
    - Should be aligned with PMU events, to count all branches captured
    - Autofdo doesn't use brtype filtering, but does use call-stack and freeze
- Freeze should capture events before the event, not after (including skid)
    - Otherwise skid branches could overwrite
    - Different from preferred behavior for counter freeze, which should only freeze at the time of interrupt

  - Important for precise sampling, where eventing PC is recorded. If not precise, then stopping on overflow could be problematic, because don't get path to sample pc (pc of inst before interrupt)
- Branch samples should not go to memory, too much memory traffic. Instead have internal circular buffer per hart.
  - Accesses should be low latency, done in sensitive code
- 16 entries gets most of autofdo benefits, 32 is a bit better. But more has costs in state management.
- Call-stack mode logs calls and pops them on rets
  - Helps work around limitations of stack unwinding, get partial call-stack 10% of time. And avoids memory accesses when doing so.
  - Useful for attributing time to functions/libraries, for calcing "datacenter tax"
  - But often doesn't go all the way back to main, e.g., if started after app started. So can't use it exclusively.
- If HW sampling is supported (logging overflows + state to memory rather than interrupting), should snapshot branch sampling buffer
- Perf_events supports branch sampling from other architectures, abstracts details from user
- Reads should be <15 cycles/reg. Optimization for context switch and clearing also helpful.
  - Context switch if buffer associated with a SW thread, but can just clear if associated with a hart
- Guest OS must be able to use it
- Must be standardized and discoverable, including num entries. Can't depend on knowing HW ID. Helps with virtualization, since capability may not be exposed to a guest.
- On sleep state entry, samples are often cleared
- Should be macrofusion-agnostic, should always point to branch, never op from fused op+branch
- AMD includes recording speculative branches, may be useful for evaluating cost of mispredicts but not sure if anyone uses it
- ARM BRBE has inst for instrumenting records
- No BRBE implementations yet, stuck using ETM for now
- Need branch metadata to be extensible, can continue to add goodness
- Autofdo isn't using all the goodness in existing implementations, focused just on each basic block rather than full path, haven't used latency data yet
- Propeller does post-link optimization, can use this data there
- Shadow-stack is better than branch sampling call-stack, ideal solution
- ARM suffering because don't have branch sampling, RISC-V needs it from the beginning

Action items

- ☐ Atish Kumar Patra - Aug 25, 2022 - check on how to read multiple counters in perf when taking an interrupt on one

- □ Beeman Strong - Jul 28, 2022 - Reach out about proprietary performance analysis tools
- □ Beeman Strong - Jul 28, 2022 - Reach out to VMware about PMU enabling
- □ Beeman Strong - Jul 28, 2022 - Talk to security HC about counter delegation