User-Space Enhancements for Linux Perf

Shay Gal-On, Laksono Adhianto, Nathan Tallent, William Cohen,

Rashawn Knapp, Emmanuel Oseret, Xingfu Wu, David Boehme

Session Notes

- Perf: Command-line performance analysis tool for Linux
- Often want to focus on specific program region (as in PAPI, Caliper, VTune, etc.) but perf doesn’t do that
  - Stack traces possible but not user defined regions in code
  - Software events are possible: https://developers.redhat.com/blog/2019/04/23/how-to-use-the-linux-perf-tool-to-count-software-events/
  - http://web.eece.maine.edu/~vweaver/projects/perf_events/
  - Need to differentiate different instances of a function etc.
- SystemTap markers may be a way insert traceable points: https://sourceware.org/systemtap/
- Oprofile as alternative? Same info as perf
- PEBS samples make context association inherently difficult

Wish List

- Add start/stop markers to limit perf stat event collection to the region
  - Multithreading? Difficult to propagate start/stop signals across threads
  - Main use case: Start/stop sampling for every thread / system-wide
  - Need to make sure multiple users don’t mess up start/stop handling
- Better correlation of different events (with different counts) in perf report
  - perf metrics introduced in Linux 5.0 may help with some of that
  - Run perf stat + perf record together?
- Report non-CPU info, for example process IO wait time
- More events / tracepoints available without root requirement
- Perf report: is organized by events, would be better to list all events in one table
- Read perf.data and export into hatchet or other tools