NSF/IUCRC CAC PROJECT

INTEGRATED VISUALIZING, MONITORING, AND MANAGING HPC SYSTEMS

Jie Li

Doctoral Student, TTU

06/19/2020

Advisors:

Mr. Jon Hass, SW Architect, Dell Inc.

Dr. Alan Sill, Managing Director, HPCC, TTU

Dr. Yong Chen, Associate Professor, CS Dept, TTU

Dr. Tommy Dang, Assistant Professor, CS Dept, TTU

OBTAIN PERFORMANCE COUNTER INFORMATION

Configuring the counters

Hundreds of events are available	A limited number of counters can be used simultaneously
Fixed counters: monitor fixed events	Programmable counters: the configuration is Performed by writing into model specific registers (MSRs)

Reading Counter Values

Pooling: the counters can be read at any instant. Counters are read using the MSRs

Event-based sampling: enabled through Performance Monitoring Interrupt (PMI)

Das, Sanjeev, Jan Werner, Manos Antonakakis, Michalis Polychronakis, and Fabian Monrose. "SoK: The challenges, pitfalls, and perils of using hardware performance counters for security." In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 20-38. IEEE, 2019.

REASONS LEAD TO INACCURATE MEASUREMENT

- **External sources**: the runtime environment may vary across runs
- **Non-determinism**: many sources of non-determinism can be hard to predict and mostly depend on OS behavior and the other applications running on the system
- **Over-counting**: performance counters may over-count certain events on some processors
- **Variations in tool implementations**: result from the techniques involved in acquiring the measurements. Polling or sampling, the measurement level (thread, process, core, multiple cores), the noise-filtering approach used etc.

Das, Sanjeev, Jan Werner, Manos Antonakakis, Michalis Polychronakis, and Fabian Monrose. "SoK: The challenges, pitfalls, and perils of using hardware performance counters for security." In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 20-38. IEEE, 2019.

ACQUIRING ACCURATE MEASUREMENTS

- **Context switch monitoring**: performance counter values must be saved during context switches to avoid any contamination due to events from other processes - non-trivial task
- **Interrupt handling:** PIMs can be handled in different ways, such as by writing a callback routine, API hooking, or hooking the PMI handler
- **Process filtering upon PMI**: filter performance counter data relevant solely to the process of interest
- **Minimizing the impact of non-deterministic events**: consider only deterministic events

Das, Sanjeev, Jan Werner, Manos Antonakakis, Michalis Polychronakis, and Fabian Monrose. "SoK: The challenges, pitfalls, and perils of using hardware performance counters for security." In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 20-38. IEEE, 2019.

MAIN STRATEGIES TO RECORD HPC MEASUREMENTS AT RUNTIME

- **Source code instrumentation:** instrument the source code with probing points before and after the main code. Requires source code modification, which may not always be possible, especially for closed-source software
- **Binary-compatible approach**: create the target process in suspended mode, and then enable the performance counters. Once the process terminates, performance counter values for the process are extracted. Monitoring starts immediately after the process is created, which is much earlier than the actual program begins.
- **Alternative of binary-compatible approach**: monitor a running process by attaching to and detaching from it after a particular condition is met.

Das, Sanjeev, Jan Werner, Manos Antonakakis, Michalis Polychronakis, and Fabian Monrose. "SoK: The challenges, pitfalls, and perils of using hardware performance counters for security." In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 20-38. IEEE, 2019.

HOW HARDWARE PERFORMANCE MONITORING COUNTER WORKS(HWPMC)

- Counters are kept on the chip
- The user selects what type of counter to use
- The kernel driver programs the chip to start counting events
- Chip generates an interrupt
- HWPMC driver reads out the counters into a memory buffer
- A user level library (PAPI) handles all access to the driver
- User level program reads information from the kernel

Ref: https://www.youtube.com/watch?v=lofmFRY-yjs. Figured by Jie Li.

PERFORMANCE COUNTERS

- A set of special-purpose processor registers
- Count hardware events such as L1 cache misses, number of FP operations etc.
- Limited in number
- Specific to processor type and vendor
- Tricky to use and understand
- Common problems: overflows, multiplexing, derived events

Ref: https://www.youtube.com/watch?v=2SGieT_ttTM

ADVANCED PAPI FEATURES - MULTIPLEXING

- Multiplexing
  - Allows more counters to be used than what is supported by the hardware
  - Allows a larger number of events to be counted simultaneously
  - Overcomes the limitation by subdividing the usage of the counter hardware over time(timesharing)
  - Unavoidably insures a small amount of overhead and can adversely affect the accuracy of reported counter values

Ref: S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, “A portable programming interface for performance evaluation on modern processors,” International Journal of High Performance Computing Applications, vol. 14, no. 3, pp. 189–204, 2000.

ADVANCED PAPI FEATURES - OVERFLOW

- Overflow
  - When a particular hardware event exceeds a specified threshold
  - Papi call user-defined handlers when a overflow occurs

Ref: S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, “A portable programming interface for performance evaluation on modern processors,” International Journal of High Performance Computing Applications, vol. 14, no. 3, pp. 189–204, 2000.

PLAN

- Explore other events and understand the meaning.
- Investigate the overhead of PAPI.
- Build metrics collector utilizing PAPI interface.

QUESTIONS?/COMMENTS?