Performance monitoring with Intel Architecture

CSCE 351: Operating System Kernels

Lecture 5.2

Why performance monitoring?

- Fine-tune software
  - Book-keeping
  - Locating bottlenecks
  - Explore potential problems or hidden problems
- Why do you need to fine tune your software
  - Performance, performance, performance
  - Cache misses, page faults …

Imagine …

- A JVM problem …
  - Three major components
    - Execution
    - Memory management
    - Synchronization
  - How would you measure the execution time of each component?

Or what about …

- Multithreaded application
  - Measure synchronization time
  - Measure amount of time in contention
  - Measure amount of time setting lock
  - Measure bus contention

How would you do it?

Time-stamp Counter

- Pentium defines a time-stamp counter mechanism
  - used to monitor and identify the relative time of occurrence of processor events
  - the architecture includes
    - instruction RDTSC is used to read the counter (a 64 bit register)
    - a feature bit (TCS flag) that can be read with the CPUID instruction,
    - a time-stamp counter disable bit (TSD flag) in control register CR4
    - a model-specific time-stamp counter

Time-stamp Counter (cont.)

- Following execution of the CPUID instruction
  - the TSC flag in register EDX (bit 4) indicates (when set) that the time-stamp counter is present in a particular Intel Architecture processor implementation.
- The time-stamp counter is a 64-bit counter that is set to 0 following the hardware reset of the processor

Time-stamp Counter (cont.)

- the time-stamp counter will increment ~6.3 x 10^{15} times per year at 200 MHz clock rate
  - it would take over 2000 years for the counter to wrap around
  - The RDTSC instruction loads the current count of the time-stamp counter into the EDX:EAX registers

Time-stamp Counter (cont.)

- the RDTSC instruction can be executed by programs and procedures running at any privilege level
- The TSD flag in control register CR4 (bit 2) allows use of this instruction to be restricted to only programs and procedures running at privilege level 0.

Time-stamp Counter (cont.)

- Since instructions in Pentium Pro and later can finish out of order
  - The RDTSC instruction is not serializing or ordered with other instructions.
    - it does not necessarily wait until all previous instructions have been executed before reading the counter.
    - subsequent instructions may begin execution before the RDTSC instruction operation is performed.

Performance Counters

- the P6 family also provides two 40-bit performance counters
  - count the number of events
  - or measure duration
- When counting event
  - a counter is incremented each time a specified event takes place

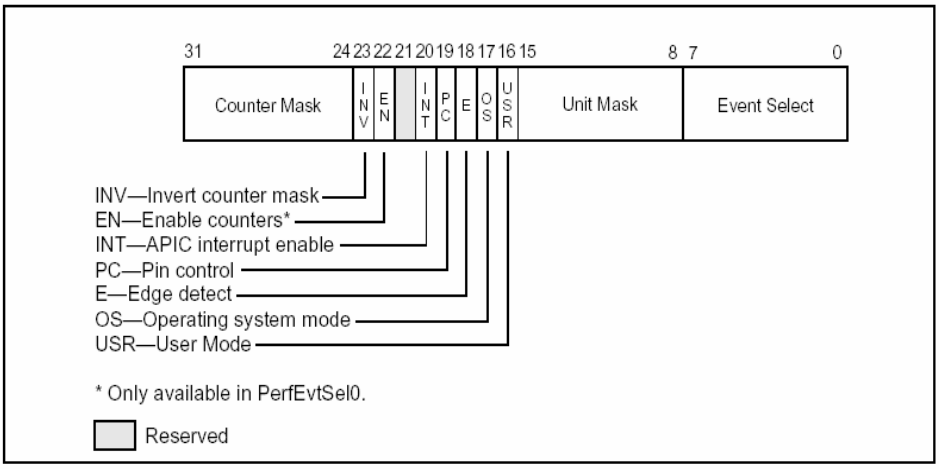
Performance counters (cont.)

- When measuring duration
  - a counter counts the number of processor clocks that occur while a specified condition is true
- The counters can count events or measure durations that occur at any privilege level.

Performance counters (cont.)

- There are four MSR (Model Specific Registers) that are used to support performance counters
  - PerfEvtSel0 and PerfEvtSel1 are used to select the events (32-bit)
  - PerfCtr0 and PerfCtr1 are used to count the number of occurrences (40-bit)

Performance counters (cont.)



Performance counters (cont.)

- RDMSR and WRMSR instructions are used to read and write to these MSR at privilege 0
  - RDMSR: MSR -> EDX:EAX registers
  - WRMSR: EDX:EAX -> MSR
- the PerfCtr0 and PerfCtr1 MSRs can be read from any privilege level using the RDPMC (read performance-monitoring counters) instruction.

Performance Counters (cont.)

- Starting and stopping the performance counters
  - start by writing valid setup information in the PerfEvtSel0 and/or PerfEvtSel1 and setting the enable flag in the PerfEvtSel0. The counter begins after the execution of WRMSR
  - Counter can be stop by clearing the enable counters flag or clearing all bit in PerfEvtSel0 and 1

Performance Counters (cont.)

```
if (ECX == 0)
{
        EDX:EAX = perfctr0;
}
else if (ECX == 1)
{
        EDX:EAX = perfctr1;
}
```

Performance Counters (cont.)

```
XOR ECX, ECX
RDPMC
MOV [MEM1], EAX
MOV [MEM2], EDX
ADD EDX, EAX
RDPMC
```

Monitoring Software

- to use the counters, the OS needs to provide an event-monitoring device driver that
  - provides feature checking (check CPUID return value)
  - initializes and starts counters
  - stops counters
  - reads the the counters or read the time-stamp counters

Sample Events

- clocks (79H)
- branch misprediction (C5H)
- data cache reference (43H)
- instruction fetch (80H)