An update on perfmon and the struggle to get into the Linux kernel

Andrzej Nowak

March 26th 2009

Generic needs for performance monitoring

- **Application developers and system tuners need performance monitoring tools to improve their software and hardware**
- **Generic requirements (at CERN)**
  - Linux compatible
  - Full x86 capability
  - Robust
  - Lightweight
  - Unintrusive
  - Reliability with HEP applications
  - No license issues, free and Open Source
- **The subsystem of choice: perfmon**
  - Not perfect, but very good

Perfmon profile (1)

- **Hardware-based performance monitoring in the operating system**
- **Features**
  - Linux support
  - System level monitoring
  - Program level monitoring, incl. following execution paths
  - Counting mode (direct PMU usage)
  - Flat profile mode; address to symbol translation when possible
  - Negligible performance impact (no cost if unused)

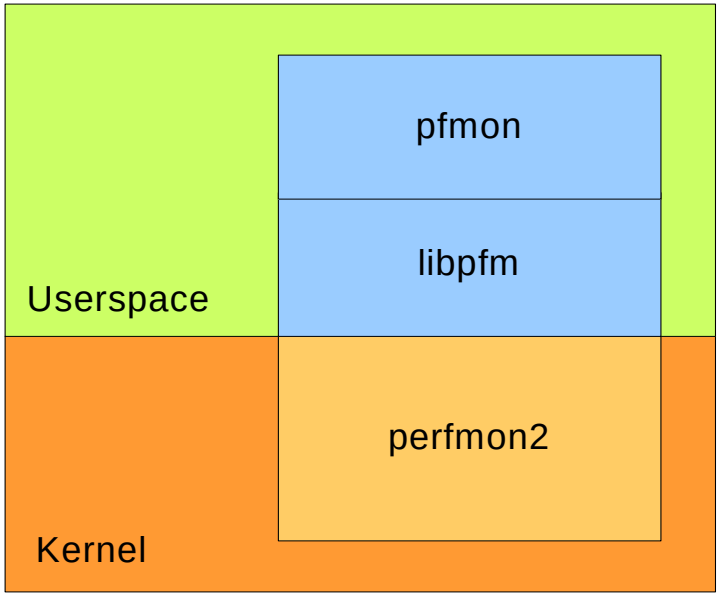
Perfmon profile (2)

- **Includes support for advanced x86 PMU features**
  - Masks, edges, latency (Core i7), PEBS, IBS
- **Manages to fully support and utilize diverse and complex PMU hardware**
  - Supported architectures: x86 (Intel/AMD), Itanium, SPARC, Cell, PowerPC, Cray, MIPS
- **Major contributors:**
  - HP Labs
  - Google
  - AMD
  - IBM
  - Intel
  - Sony
  - Toshiba
  - Cray
  - SiCortex
  - Broadcom
  - Cornell University

Perfmon architecture

> 3 components:

- Simple kernel-based PMU access facility and context switcher (as a kernel patch)
- Intermediate library
- Client applications (i.e. pfmon)



Usage scenarios

> Counting

- Example: How many instructions did my application execute?
- Example: How many times did my application have to stop and wait for data from the memory?

> Sampling

- Reporting results in “regular” intervals
- Example: every 100’000 cycles record the number of SSE operations since the last sample

> Profiling

- Example: how many cycles are spent in which function?
- Example: how many cache misses occur in which function?
- Example: which code address is the one most frequently visited? (looking for hotspots)

A brief history of perfmon

- Conceived in HP labs for the Itanium architecture (Stephane Eranian)
- Gradually expanded to support other architectures as well
- x86 support significantly improved in recent years; well maintained
- Expanded to be a standard performance monitoring framework for Linux (see Jan 24th 2008 presentation from S. Eranian)

perfmon at CERN (1)

- **Certain restrictions needed to be lifted**
  - Lack of robust symbol resolutions
  - Lack of compatibility with large frameworks and long execution chains
  - Lack of general interoperability
- **Significant effort on behalf of Stephane Eranian (then HP Labs), Ryszard Jurga (CERN) and Andrzej Nowak (CERN) to improve on those points in 2006 and 2007**

perfmon at CERN (2)

- Symbol resolution code rewritten 3 times
- Handles large software complexity and multiple monitoring modes well (per-thread, system-wide, kernel level etc)
- Used in many software and hardware related R&D projects at CERN openlab, in CERN IT and in other parts of CERN (Physics dept)
- Used in a test run on 60 batch nodes collecting CPU usage data in the background

Further developments

- **Other activities foreseen**
  - Continued and/or expanded monitoring of batch nodes (minor performance hit)
  - Easy deployment across standard CERN configurations
- **For that, perfmon should be included in Scientific Linux (SL)**
- **In order for that to happen, perfmon v2/v3 should be included in Red Hat Enterprise Linux which is the base for SL**
- **That means that perfmon should be available in the Linux kernel**
  - A goal which is in line with the Author's strategy

Kernel inclusion benefits

- **Recognition as a standard**
- **Adoption by major distributions**
  - Performance monitoring applications readily available to the user
- **ISVs regard favorably**
- **Hardware manufacturers regard favorably**
- **Assurance of stability, support and portability**

- **Thus, many influential people would like this, not just us**

Getting perfmon in the linux kernel (1)

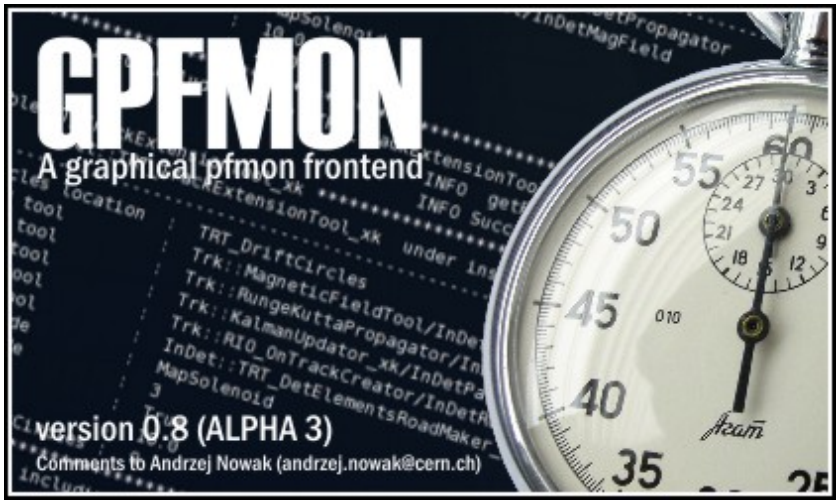
- **Working with the kernel community is difficult**
- **Years of relentless efforts of Stephane Eranian (HP Labs / Google) have yielded mixed results**
  - Git trees created
  - Several iterations of reviews on LKML over the years
  - Minimalistic and fragmented “perfmon v3” created specifically to satisfy LKML critiques
    - This version was very close to inclusion and already being tested
- **Rival patch posted in late 2008**
  - Different approach, redesign
- **Lots of politics involved**

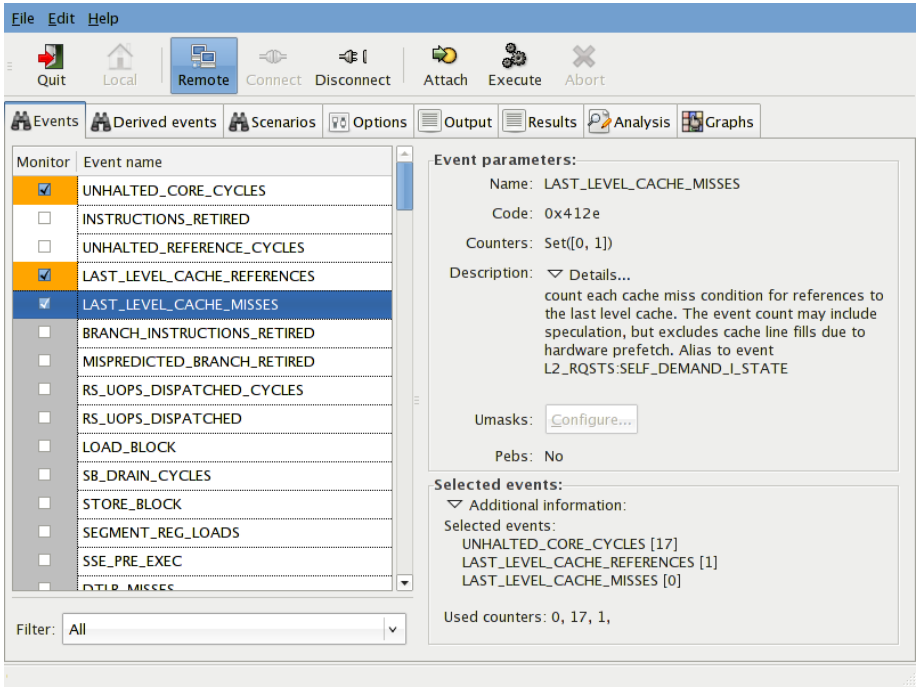
Getting perfmon in the linux kernel (2)

- **The future is unclear at this moment**
- **The good news**
  - There is some serious discussion about access to performance monitoring in “userspace”
  - Linux is finally nearing a solution to the performance monitoring problem
- **The bad news**
  - Years of experience and testing on multiple platforms are being ignored
  - The “new” solution seriously lacks robustness and expandability, dismisses the needs of expert users
  - Coders responsible for the patch seem to have remarkable kernel experience, but no significant performance monitoring experience
  - The “new” solution is not perfmon, which we know and successfully use on a wide variety of CPUs

> gpfmon – a graphical front-end to pfmon

- Other tools, such as HP Caliper and VTune feature robust GUIs





> Performance tuning workshops at CERN

Future directions

- **Continued perfmon support at CERN**
  - Usage
  - Development
  - Improvements, fixes
  - Testing (especially with new hardware platforms)
- **Closer integration with experiments**
- **Possible closer integration with batch processing**
- **Looking forward to a standard for performance monitoring in Linux**

Q & A

CERN openlab