Recent developments in Performance Monitoring

CERN openlab II quarterly review
31 January 2007

Ryszard Jurga

Outline

- Introduction to performance monitoring
  - Performance Monitoring Unit
  - Perfmon2 interface
- CERN user requirements
- Collaboration with HP
  - Meetings
  - CERN contribution
- Sample results
- Future plans
- Conclusions

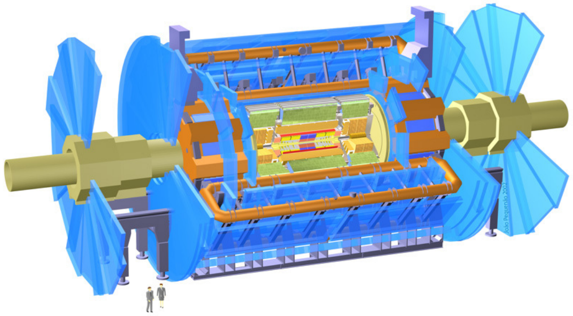
Introduction

- Performance Monitoring Unit (PMU)
  - a piece of CPU HW collecting micro-architectural events in all modern CPU: from pipeline, system bus ,caches…
  - diversity of PMU implementation
    - no-architected (e.g., P3/P4, Xeon)
      - large differences even inside a processor family
    - architected (e.g., IA-64, AMD64, Intel Core)
      - consistent across processor implementations
- Interfaces
  - perfctr, oprofile, VTUNE, perfmon2

perfmon2 interface

- portable across all PMU models
- with support for per-thread and for system-wide monitoring
- in user or kernel domain
- with support for counting and sampling
- with support for event multiplexing
- without special recompilation of a monitored application
- secure
- well documented

CERN User requirements



- CERN users
  - Atlas and LHCb experiments
  - simulation and reconstruction jobs
  - with 400+ dynamic libs per job
  - run by scripts (python)
  - on x86, x86_64 with Scientific Linux 3
- Experience from performance monitoring
  - Ryszard Jurga talk at Geant4 Collaboration Workshop, 14th Oct, Lisbon
    - results from profiling of different physics applications
    - existed tools do not meet CERN users requirements
    - symbol name resolution from dynamic libraries is a big challenge

Collaboration - Gelato ICE meeting in Singapore 2007

- HP and CERN presentations:
  - CERN experience from performance monitors
    - one scalable and portable tool across multiple platforms would be an ideal solution
    - perfmon2 and pfmon includes support for more and more processors and more useful features
  - HP update on the perfmon2 monitoring interface
    - support for more processors (i.e., Xeon, Core Duo 2, Montecito)
    - new features in pfmon (i.e., more mature sampling)
- common interest
  - HP TODO list vs. CERN list of requests
  - CERN contribution to pfmon
    - improving symbol resolutions (shared libs)
    - interface and tool testing on different processors with emphasis on x86 and x86_64

CERN contribution to pfmon

- improving symbol resolutions
  - support for shared libraries
    - linked against application
    - dynamically loaded during an execution (dlopen/dlclose)
    - resolving across multiple processes/threads
      - can follow fork, exec, pthread_create
    - new aggregation approach
  - support across multiple processors
    - one tool for all supported processors
    - Xeon, Woodcrest, Itanium
- patch with +2k lines of code submitted and pending verification by Stéphane Eranian, CVS repository changes

First results – geant4

one tool on all supported platforms

```
# counts   %self    %cum function name:file
Samples: 901644
 118736  13.17%  13.17% __ieee754_log:libm-2.3.4.so
  85733   9.51%  22.68% CLHEP::RanecuEngine::flat():libCLHEP-1.9.2.3.so
  50836   5.64%  28.32% __ieee754_exp:libm-2.3.4.so
  46250   5.13%  33.45% G4VProcess::SubtractNumberOfInteractionLengthLeft():ibG4procman.so
  31953   3.54%  36.99% G4SteppingManager::DefinePhysicalStepLength():libG4tracking.so
  26342   2.92%  39.91% G4UniversalFluctuation::SampleFluctuations():libG4emstandard.so
  20830   2.31%  42.22% G4Track::GetVelocity() const:libG4track.so
  16984   1.88%  44.10% cos:libm-2.3.4.so
  14004   1.55%  45.66% G4SteppingManager::InvokePSDIP():libG4tracking.so
```

Xeon

```
# counts   %self    %cum function name:file
Samples: 359161
 41046  11.43%  11.43% __ieee754_log:/lib64/tls/libm-2.3.4.so
 38217  10.64%  22.07% CLHEP::RanecuEngine::flat():libCLHEP-1.9.2.3.so
 24457   6.81%  28.88% __ieee754_exp:libm-2.3.4.so
 16188   4.51%  33.39% G4UniversalFluctuation::SampleFluctuations():libG4emstandard.so
 10620   2.96%  36.34% G4Track::GetVelocity() const:libG4track.so
 10155   2.83%  39.17% G4VProcess::SubtractNumberOfInteractionLengthLeft():libG4procman.so
  8337   2.32%  41.49% G4UrbanMscModel::ComputeGeomPathLength(double):libG4emstandard.so
  7979   2.22%  43.71% G4SteppingManager::DefinePhysicalStepLength():libG4tracking.so
  7558   2.10%  45.82% G4UrbanMscModel::SampleCosineTheta():libG4emstandard.so
  7206   2.01%  47.82% cos:libm-2.3.4.so
```

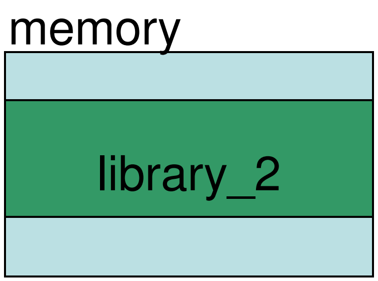
Core Duo 2

```
# counts   %self    %
Samples: 408514
 43914  10.75%  10.75% __divdi3:libgcc_s-3.4.6-20060404.so.1
 32918   8.06%  18.81% CLHEP::RanecuEngine::flat():libCLHEP-1.9.2.3.so
 24958   6.11%  24.92% __divdi3:libgcc_s-3.4.6-20060404.so.1
 16176   3.96%  28.88% G4SteppingManager::DefinePhysicalStepLength():libG4tracking.so
 10846   2.65%  31.53% exp:libm-2.3.4.so
 10776   2.64%  34.17% sqrt:libm-2.3.4.so
 10276   2.52%  36.69% G4UniversalFluctuation::SampleFluctuations():libG4emstandard.so
 10118   2.48%  39.16% G4SteppingManager::InvokePSDIP():libG4tracking.so
  9199   2.25%  41.41% G4SteppingManager::Stepping():libG4tracking.so
  8541   2.09%  43.50% log:/lib/tls/libm-2.3.4.so
```

Itanium

Results – dynamically loaded libs

```
main(){
  load(library1)
  function_hello1_from_library_1()
  unload(library1)
  load(library2)
  function_hello2_from_library_2()
  unload(library2)
}
```



```
% Total  Cumulat
  IP       % of            IP
Samples    Total         Samples  Function                        File
 100.00   100.00          472286  libhello1.so::hello_1_function_test
    …          …                                        …
```

- tested against different tools:
  - •q-tools, PerfSuite, oprofile, caliper, pfmon

pfmon, oprofile: all dynamic libs

```
# counts   %self    %cum function name:file
Samples: 145922
  78517   53.81%     53.81% hello_2_function_test:libhello2.so
  67390   46.18%     99.99% hello_1_function_test:libhello1.so
    …          …                                 …
```

Collaboration meeting at CERN

- Stéphane seminar: Overview of the perfmon2 interface
  - integrating into the mainline kernel source
    - resource sharing (i.e., NMI)
    - split into small pieces (~700k patch)
    - impact on CERN linux distribution
- discussion about CERN contribution
  - pfmon
  - unresolved symbols from ‘init’ section of dynamic libs: HP Caliper Team will be involved
  - impact of results on other HP tools: feedback to HP Caliper Team, will be solved in the next release (4.2)
- discussion about new features
  - output easy to parse by user scripts, programs
  - call graph (porting q-tools into x86_64)

Future plans

- Testing perfmon2 and pfmon at CERN
  - preparing a set of 20-50 nodes into production mode
    - Woodcrest
    - the SLC4 on board
    - kernel with perfmon2
    - afs, …
- improving the final data analysis, memory management
- stressing pfmon with physics applications and other complex programs
- adding new features in pfmon

Conclusions

- as soon as perfmon2 is in the mainline kernel source, we will get it in Scientific Linux at CERN
- with perfmon2 and pfmon we get one common interface to all supported processors and their performance units
- one common performance monitoring and profiling tool pfmon across all supported processors