CERN openlab

for DataGrid applications

Practical Experience with performance monitoring

Ryszard Jurga
CERN openlab

March 29, 2006

Agenda

- Introduction
  - perfctr
  - Pentium 4/Xeon
- Monitoring tool
  - sampling, multiplexing
- Sample measurements
  - Geant4 (test40), Atlas Simulation, make
  - lxbatch
- Applications
  - Profiling
- Conclusions

Introduction

- Special on-chip hardware of modern CPU
  - Direct access to CPU resources such as branch prediction, data and instruction caches, floating point instructions, memory operations
  - Event detectors, counters
    - Itanium2: 4 counters, 100+ monitorable events, two set of registers: PMC, PMD
    - **Pentrium4,Xeon**: 44 event detectors, 18 counters
  - Linux interfaces and libraries:
    - Part of kernel in order to per-thread and per-system measurements
    - Perfmon2
      - uniform across all hardware platforms
      - events multiplexing
      - the number of fully supported processors are very low except Itanium
      - kernel 2.6 (integrated for Itanium)
    - **perfctr**

perfctr

- version 2.6.19
  - per-thread and system-wide measurements,
  - user and kernel domain,
  - Support for a lot of CPU (P MMX/Pro/II/III/IV/Xeon/Celeron…), no support for Itanium
  - kernels 2.4 & 2.6,
  - No multiplexing,
  - Almost no documentation apart from comments in source files,
  - Require a deep understanding of performance monitoring features of every processors

Pentium 4 Performance Monitoring Features

- •44 event detectors, 9 pairs of counters
- •2 control registers (ESCR, CCCR)
- •2 classes of events:
  - •Non-retirement events – those that occur any time during execution (1 counter)
  - •At-retirement events – those that occurred on execution path and their results were committed in architectural state (1 or 2 counters)
- •multiplexing

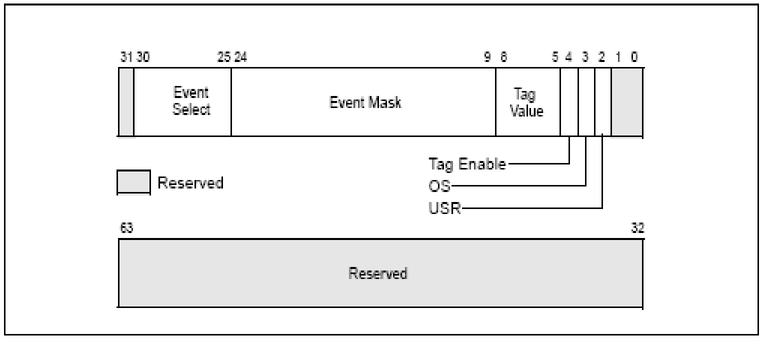


Figure 18-9. Event Selection Control Register (ESCR) for Pentium 4 and Intel Xeon Processors without HT Technology Support

from Intel documentation

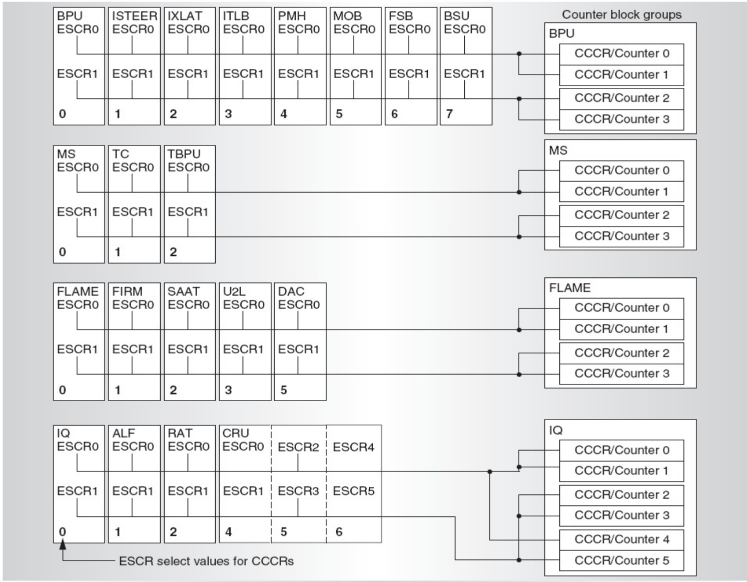


Figure 2. Interconnections among event detectors and event select control registers (ESCRs), and their associated counters and counter configuration control registers (CCCRs).

from B. Sprunt "Pentium 4 Performance-Monitoring Features"

CERN openlab for DataGrid applications

Monitoring tool - *gpfmon*

- uses perfctr,
- enables multiplexing,
- user and kernel domain,
- per single or total CPU,
- events:

CYC	TOT	BR_TP	BR_TM	L2LM	L2SM
CYC	TOT	FP	LD	L2LM	L2SM
CYC	TOT	SDS	ST	L2LM	L2SM
CYC	TOT	LDST	BR	L2LM	L2SM

CYC – CPU cycles

TOT – Instructions completed

BR_TP – Branch taken predicted

BR_TM – Branch taken mispredicted

L2LM – L2 load missed

L2SM – L2 store missed

FP – Floating point instructions

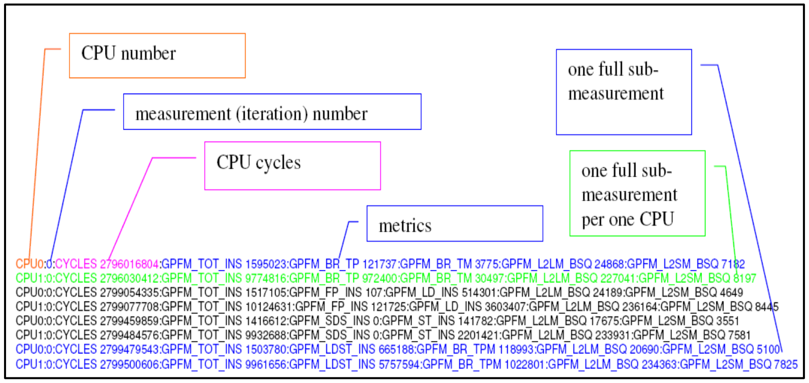
SDS – scalar instructions

LD – load intstructions

ST – store instructions

BR – BR_TP+BR_TM

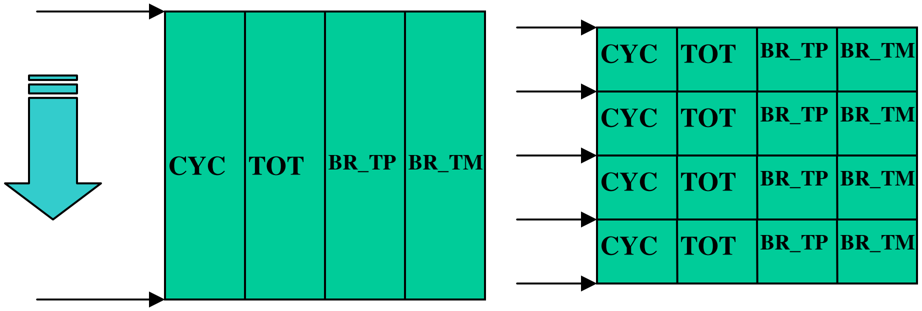
LDST - LD+ST


CPU number
measurement (iteration) number
CPU cycles
metrics
one full sub-measurement
one full sub-measurement per one CPU

```
CPU0:0:CYCLES 2796016804:GPFM_TOT_INS 1595023:GPFM_BR_TP 121737:GPFM_BR_TM 3775:GPFM_L2LM_BSQ 24868:GPFM_L2SM_BSQ 7182
CPU1:0:CYCLES 2796030412:GPFM_TOT_INS 9774816:GPFM_BR_TP 972400:GPFM_BR_TM 30497:GPFM_L2LM_BSQ 227041:GPFM_L2SM_BSQ 8197
CPU0:0:CYCLES 2799054335:GPFM_TOT_INS 1517105:GPFM_FP_INS 107:GPFM_LD_INS 514301:GPFM_L2LM_BSQ 24189:GPFM_L2SM_BSQ 4649
CPU1:0:CYCLES 2799077708:GPFM_TOT_INS 10124631:GPFM_FP_INS 121725:GPFM_LD_INS 3603407:GPFM_L2LM_BSQ 236164:GPFM_L2SM_BSQ 8445
CPU0:0:CYCLES 2799459859:GPFM_TOT_INS 1416612:GPFM_SDS_INS 0:GPFM_ST_INS 141782:GPFM_L2LM_BSQ 17675:GPFM_L2SM_BSQ 3551
CPU1:0:CYCLES 2799484576:GPFM_TOT_INS 9932688:GPFM_SDS_INS 0:GPFM_ST_INS 2201421:GPFM_L2LM_BSQ 233931:GPFM_L2SM_BSQ 7581
CPU0:0:CYCLES 2799479543:GPFM_TOT_INS 1503780:GPFM_LDST_INS 665188:GPFM_BR_TPM 118993:GPFM_L2LM_BSQ 20690:GPFM_L2SM_BSQ 5100
CPU1:0:CYCLES 2799500606:GPFM_TOT_INS 9961656:GPFM_LDST_INS 5757594:GPFM_BR_TPM 1022801:GPFM_L2LM_BSQ 234363:GPFM_L2SM_BSQ 7825
```

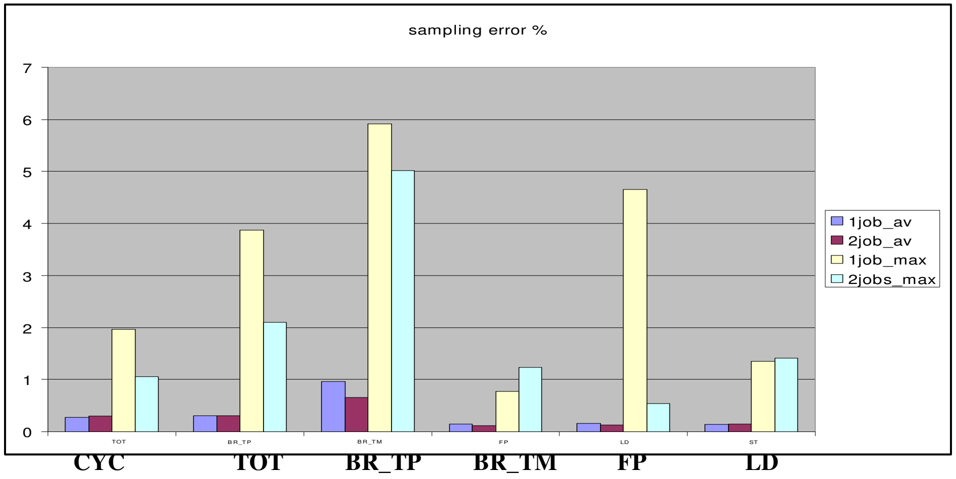
Sw sampling vs. *perfctr* sampling

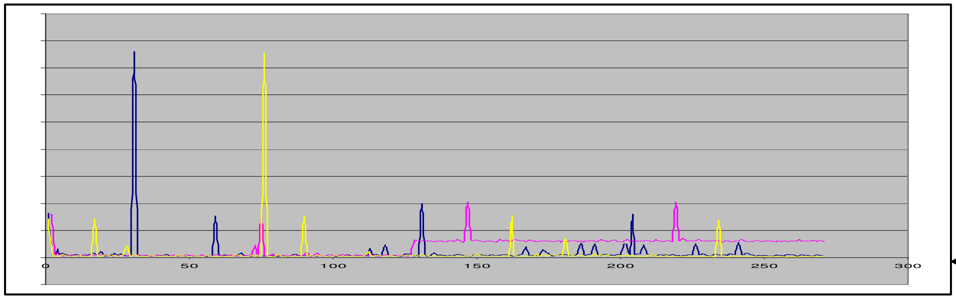
- test40
  - 4 sets, 3 times, sp 1s
    - 1,2 jobs
    - 3 jobs



$$\frac{\sum \frac{|X_{WS} - X_S|}{X_{WS}}}{n} * 100\%$$

X_{WS} - the value of counter without sw sampling
X_S - the value of counter with sw sampling
n – the number of collected samples

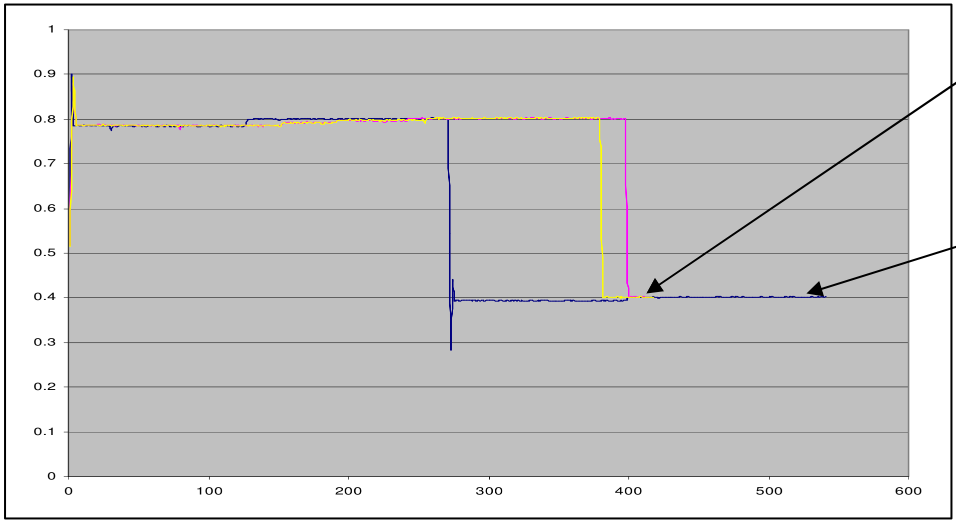




	Collected samples % 1job	Collected samples % 2jobs
CYC	98.88	98.52
TOT	99.00	98.9
BR_TP	99.06	99.09
BR_TM	97.05	94.31
FP	99.08	98.87
LD	99.03	98.84
ST	99.49	98.97
L2LM	99.71	97.45
L2SM	51.51	10

Sw sampling vs. *perfctr* sampling

- 3jobs



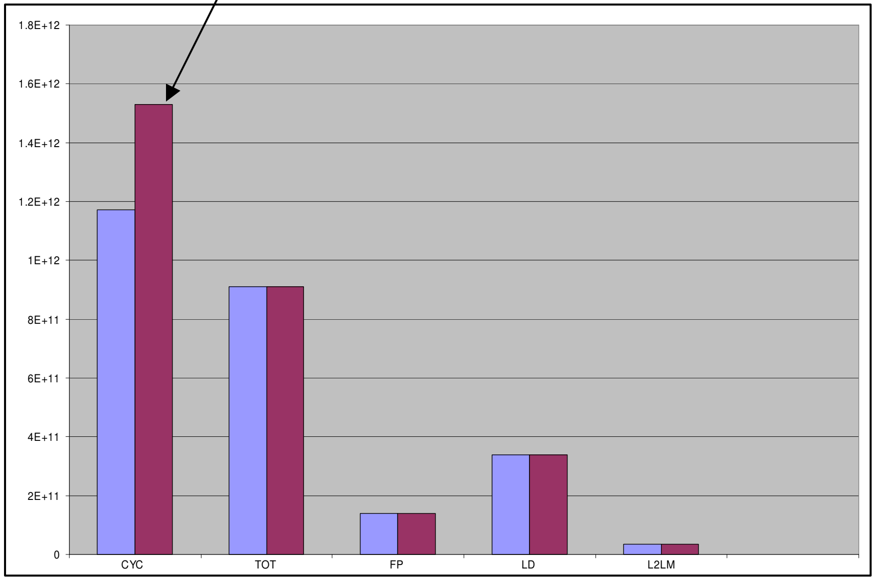
420s

540s

31%

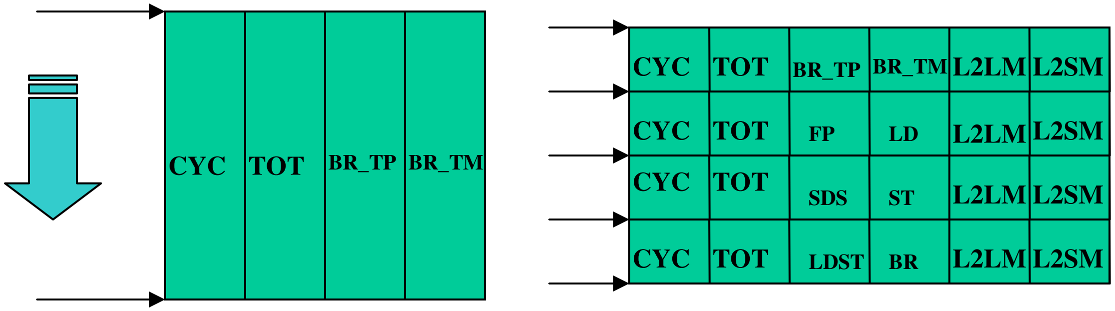
CYC	1170442842782	1528876572499
TOT	910449938595	910053742885
FP	141095332033	141023149439
LOAD	340282127317	340126643068
L2LM	36364751649	36374788916
L2LS	7802195	10010569

28%



Multiplexing

- test40
  - 4 sets, 3 times, sp1s
    - 1,2 jobs



CYC	TOT	BR_TP	BR_TM

CYC	TOT	BR_TP	BR_TM	L2LM	L2SM
CYC	TOT	FP	LD	L2LM	L2SM
CYC	TOT	SDS	ST	L2LM	L2SM
CYC	TOT	LDST	BR	L2LM	L2SM

$$\frac{\sum \frac{\left|X_{WS} - X_S\right|}{X_{WS}}}{n} * 100\%$$

X_{WS} - the value of counter without sw sampling
X_S - the value of counter with sw sampling
n – the number of collected samples

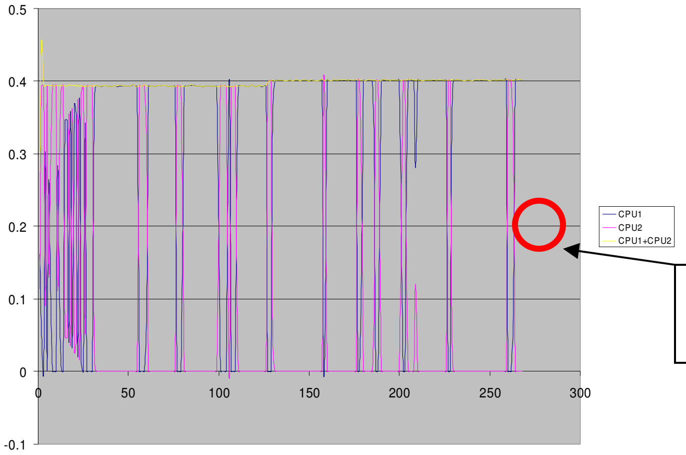
	1job		2jobs	
	average %	max %	average %	max %
TOT	0.12	1.38	0.19	16.65
BR_TP	0.07	5.64	0.12	5.48
BR_TM	0.08	11.85	0.13	11.49
FP	0.10	0.98	0.15	1.12
LD	0.10	3.14	0.16	3.52
ST	0.09	4.55	0.15	4.45

	samples % 1job	**samples % 2jobs**
CYC	98.75	98.09
TOT	98.86	86.01
BR_TP	99.69	99.51
BR_TM	98.73	96.84
FP	98.82	98.63
LD	99.07	98.89
ST	98.89	98.29
L2LM	99.43	86.05
L2LS	46.58	8.49

test40

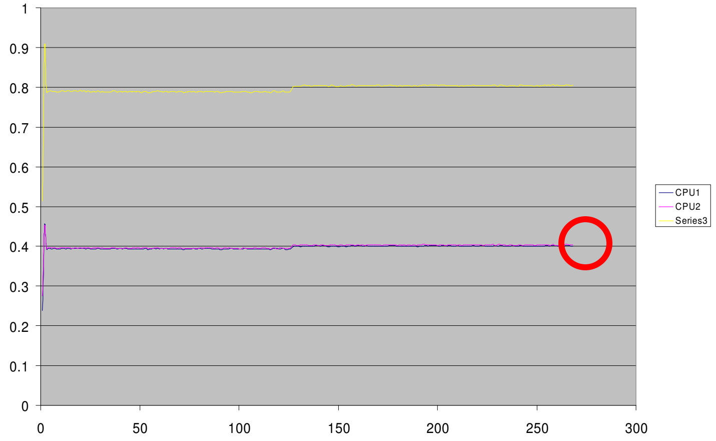
Total instructions

1job

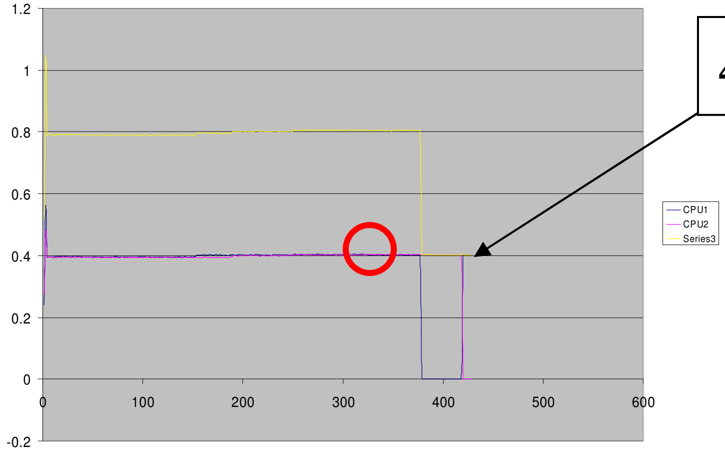


270s

2jobs

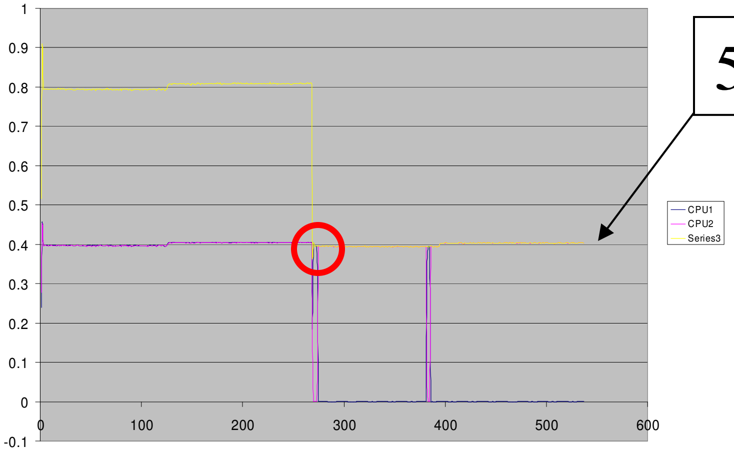


3jobs



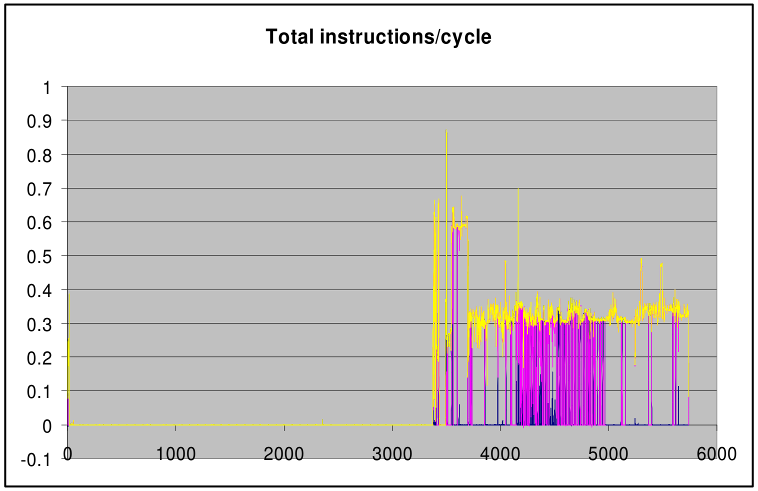
420s

3jobs



540s

Geant4 Atlas Simulations

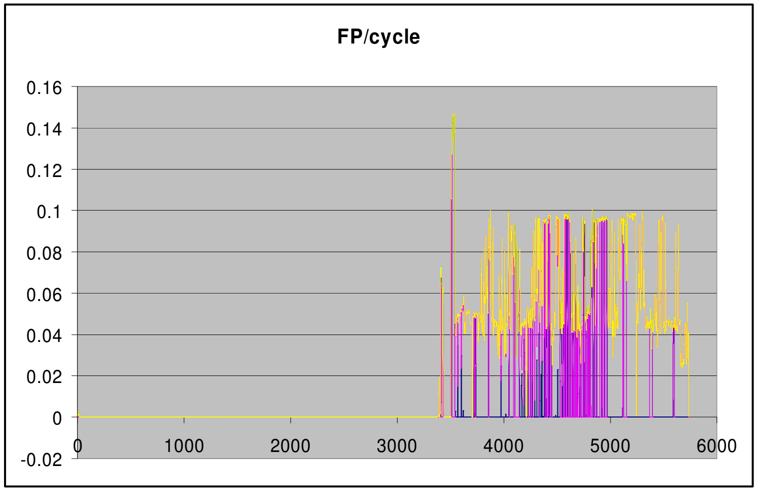


Total instructions

Cycles	16067552642403
Total inst	2216977123726
INS/CYC	0.138

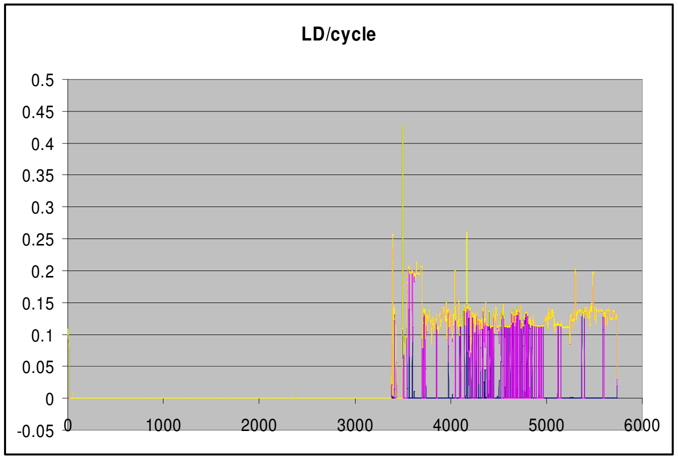
Floating-point instructions

FP	402251034688
FP/TOT	18.14%
FP/CYC	0.025



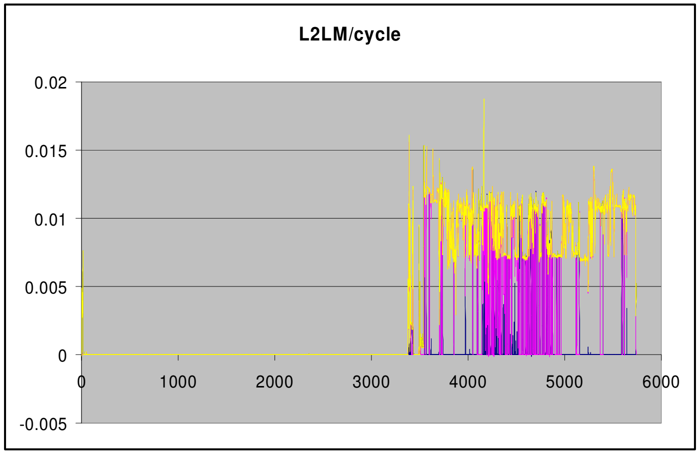
Geant4 Atlas Simulations

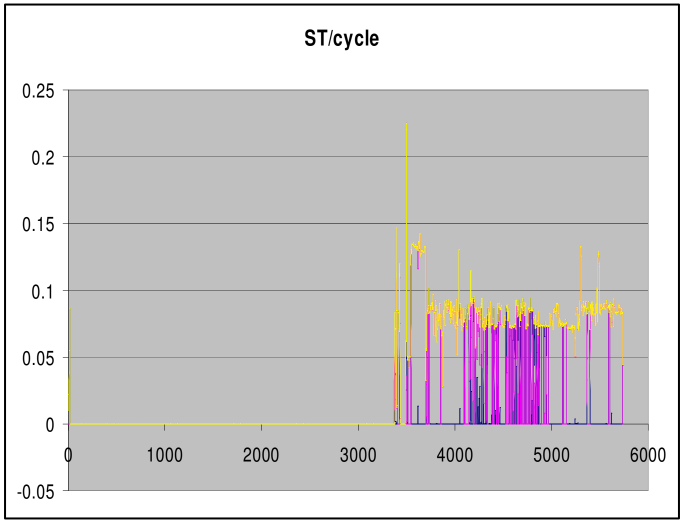
Memory



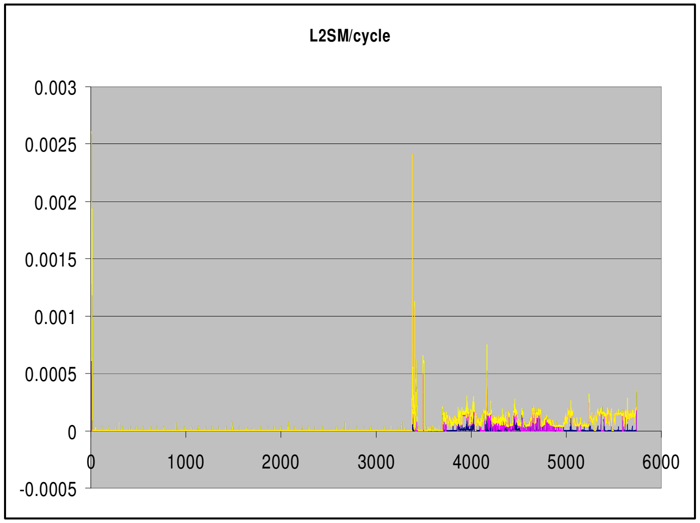
63%

LD	848049506780
LD/TOT	38.25%
LD/CYC	0.053
L2LM	61010720039
L2LM/LD	7.19%

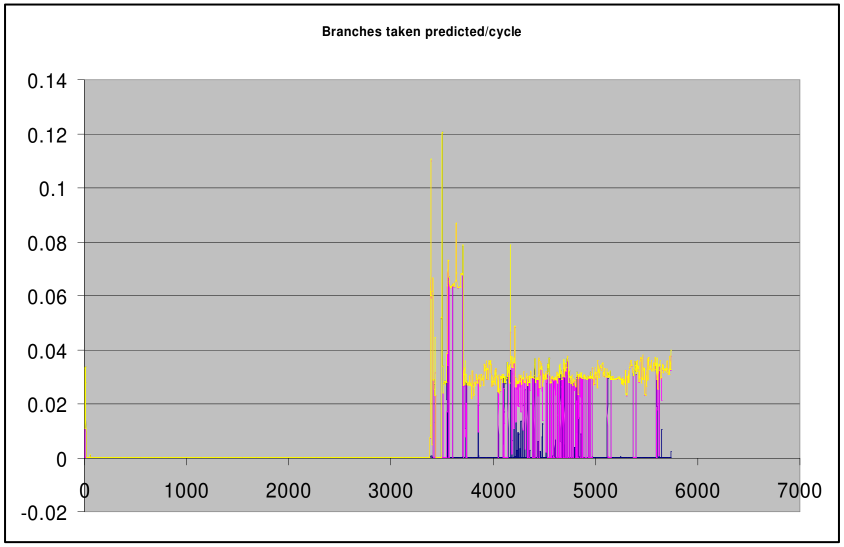


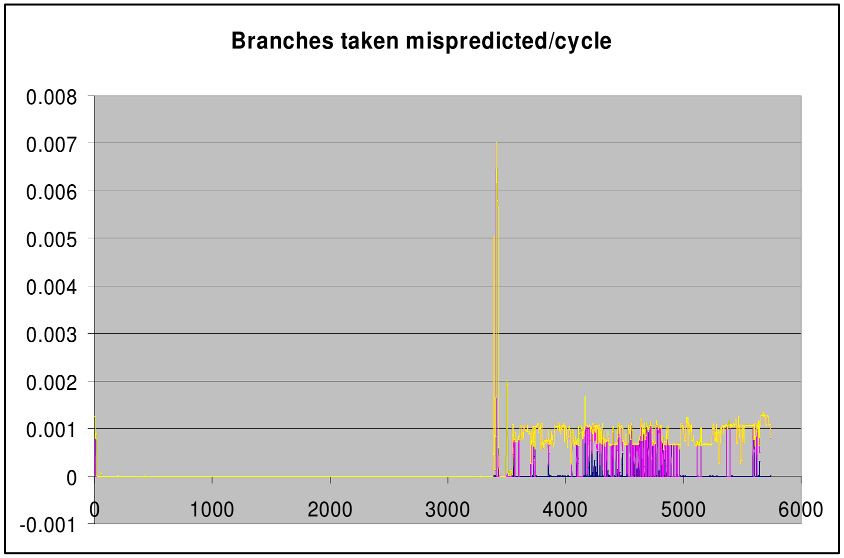


ST	548061694948
ST/TOT	24.72%
ST/CYC	0.034
L2SM	737751425
L2SM/ST	0.135%



Geant4 Atlas Simulations





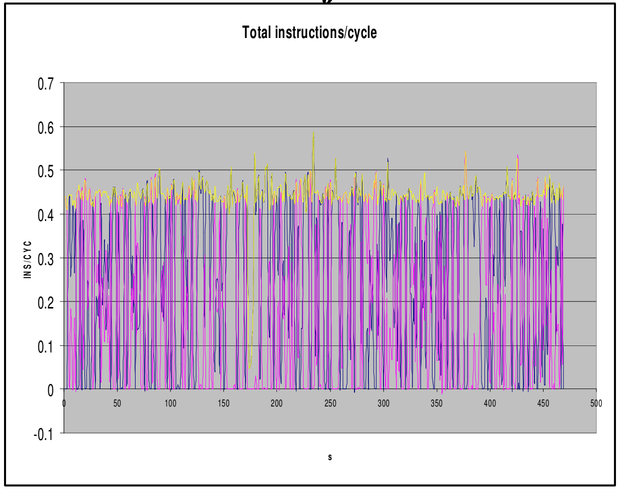
Branches

10%

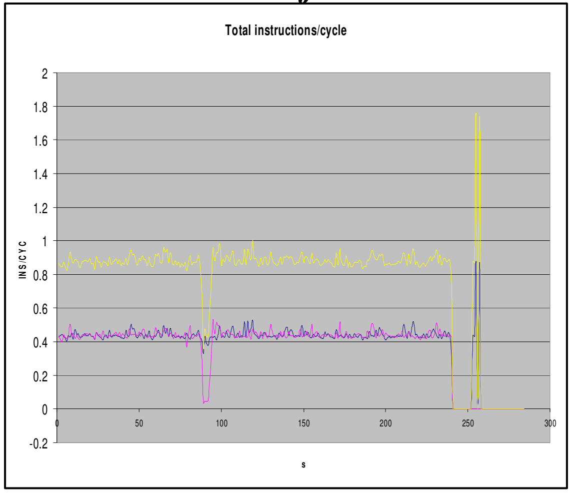
BR_TP	218342330220
BR_TM	5964007356
BR_TP/TOT	9.85%
BR_TM/TOT	0.269%

make

make –j1



make –j2



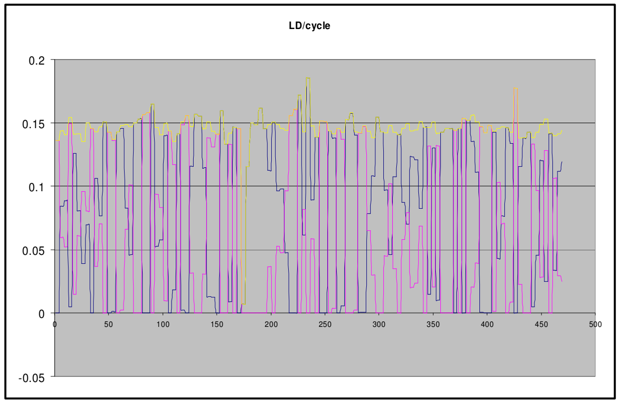
Total instructions

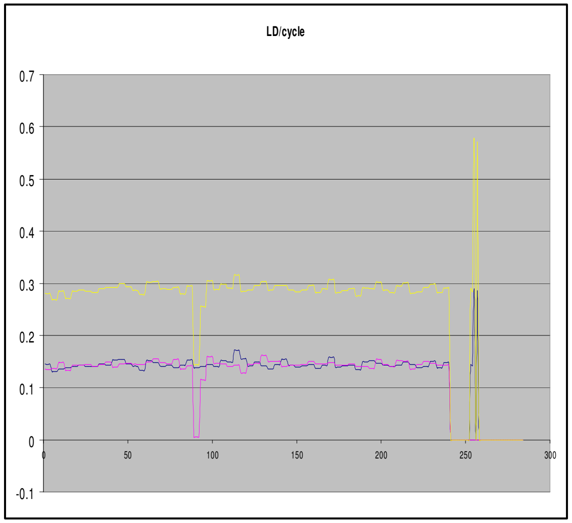
CYC	1328309944643	673216187945
TOT	586734515764	586734515764
INS/CYC	0.44	0.87

97%

Load instructions

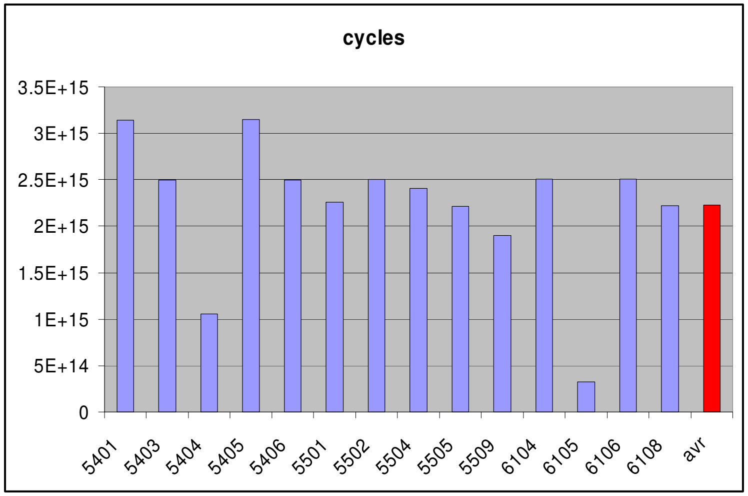
LD	193925962348	192317045567
LD/TOT	33.1%	32.7%
LD/CYC	0.146	0.286



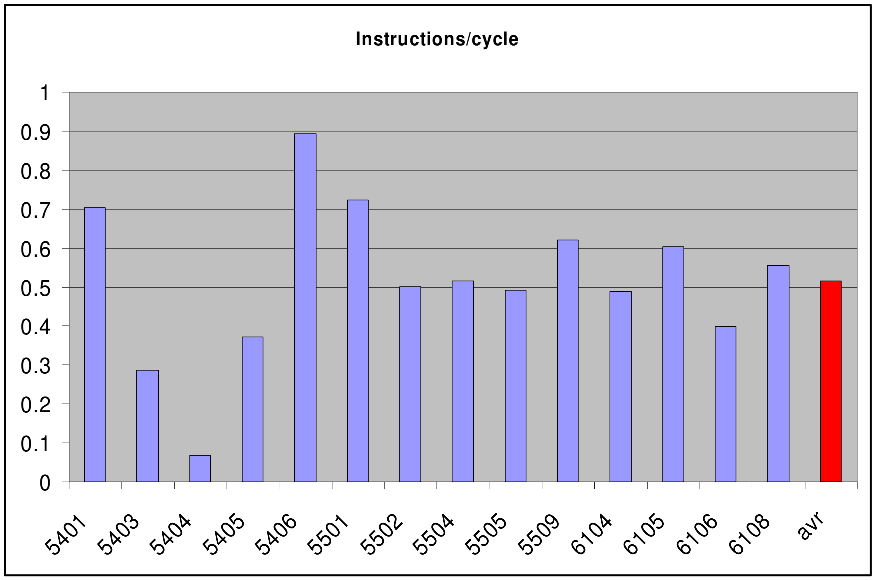


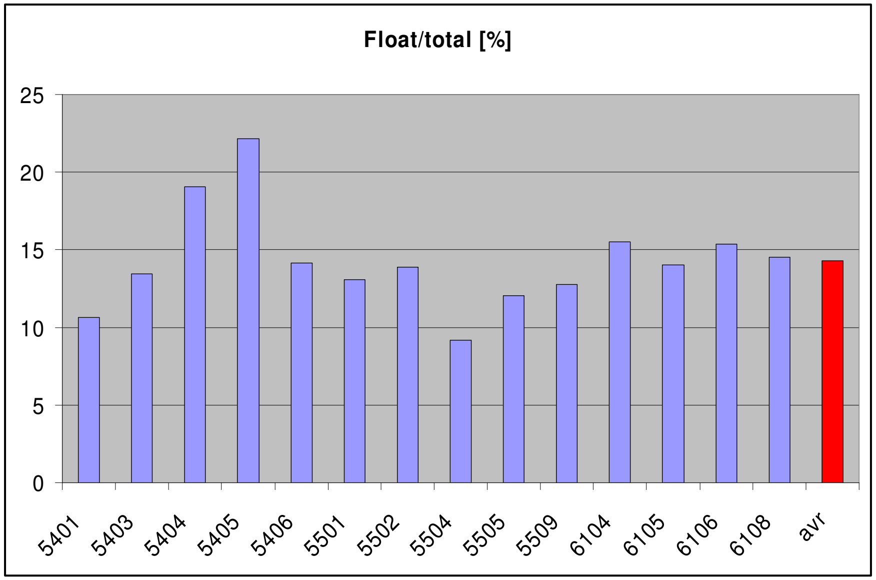
lxbatch monitoring

- 14 machines
- running from 2 day to 2 weeks
- Nocona(10), Irwindale (4)
- 2.8GHz
- 1MB L2(10) 2MB L2(4)
- SL3 (kernel 2.4)

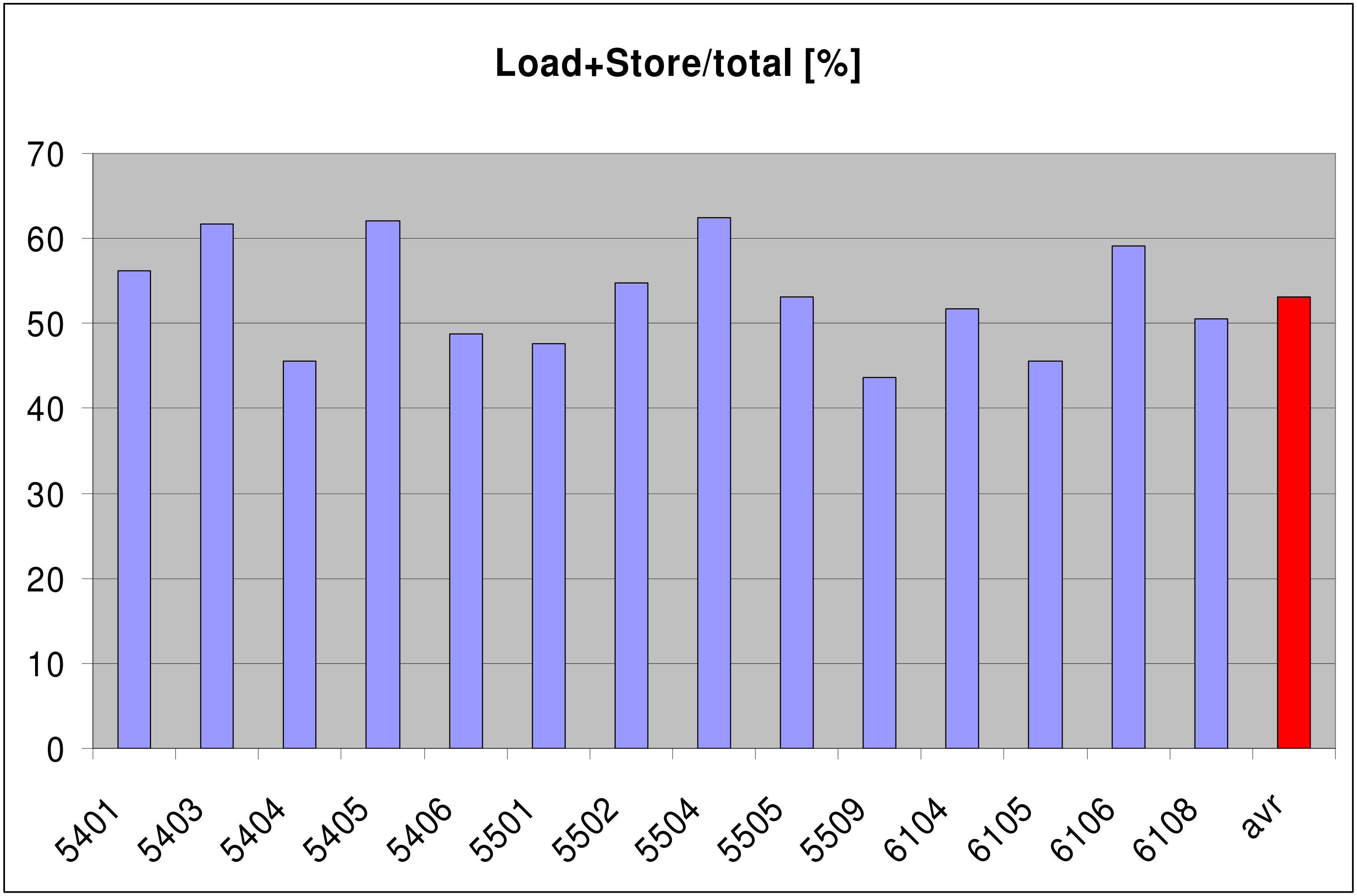


lxbatch


Instructions/cycle
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5401
5403
5404
5405
5406
5501
5502
5504
5505
5509
6104
6105
6106
6108
avr

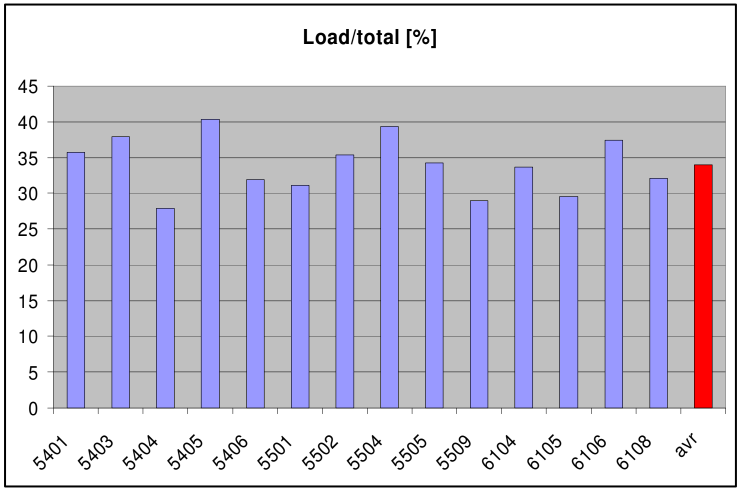

Float/total [%]
25
20
15
10
5
0
5401
5403
5404
5405
5406
5501
5502
5504
5505
5509
6104
6105
6106
6108
avr

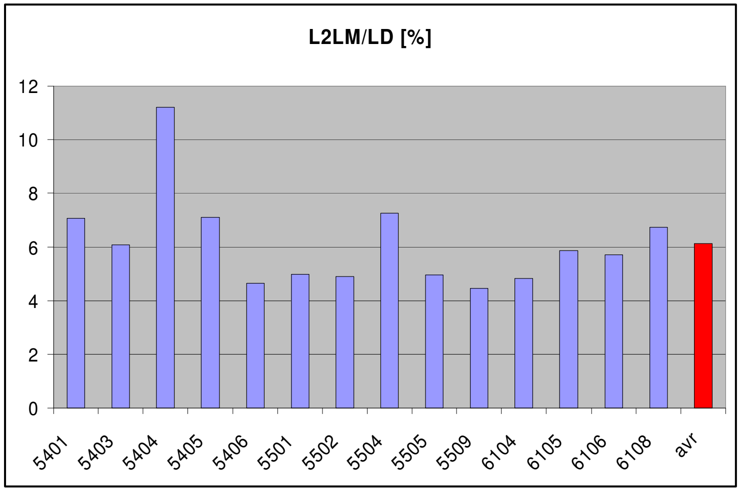
lxbatch - memory operations

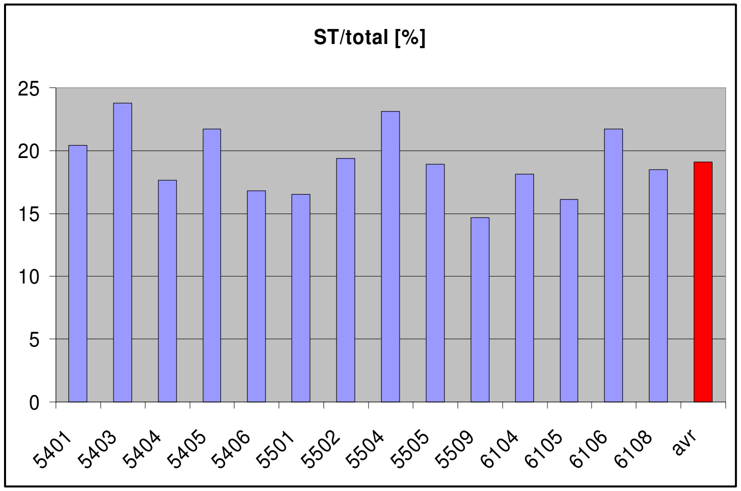


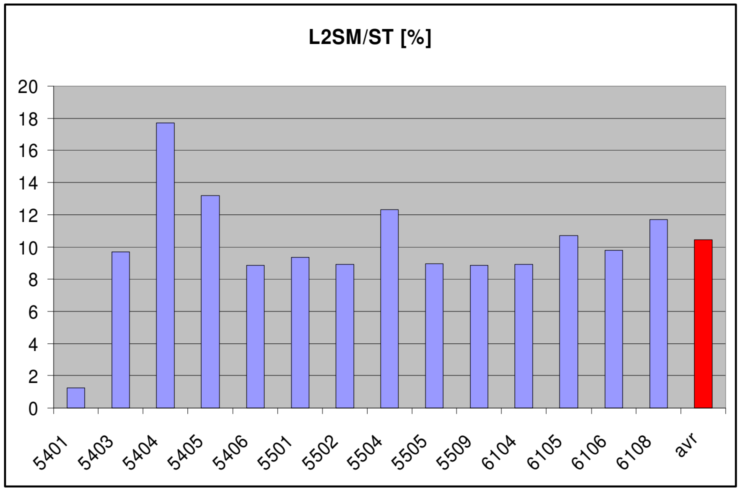
CERN openlab
for DataGrid applications

lxbatch - memory operations

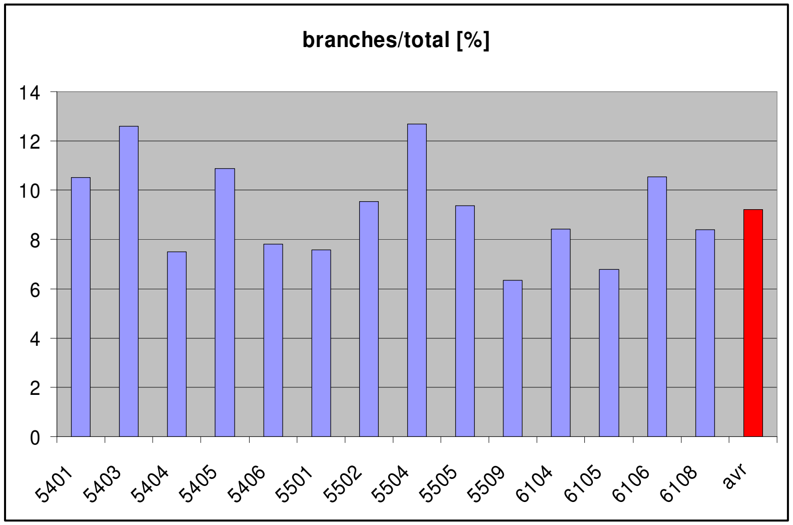


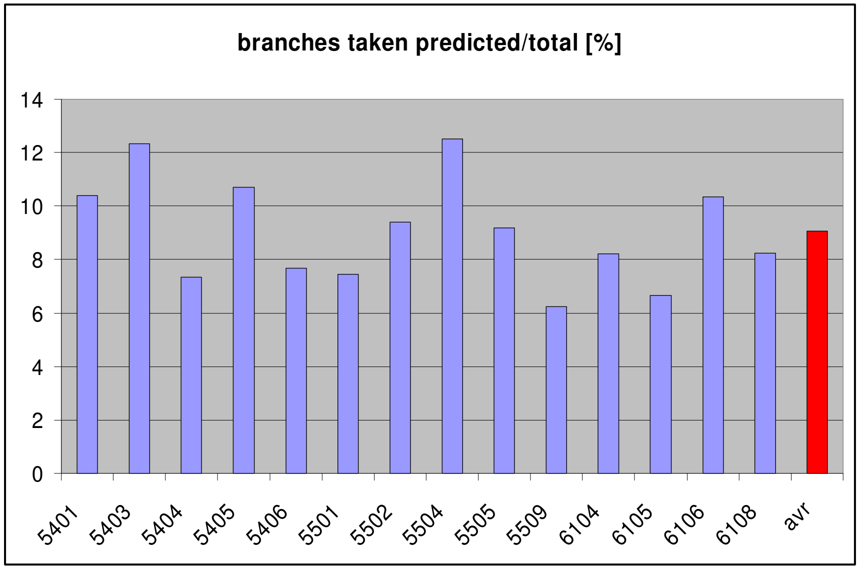


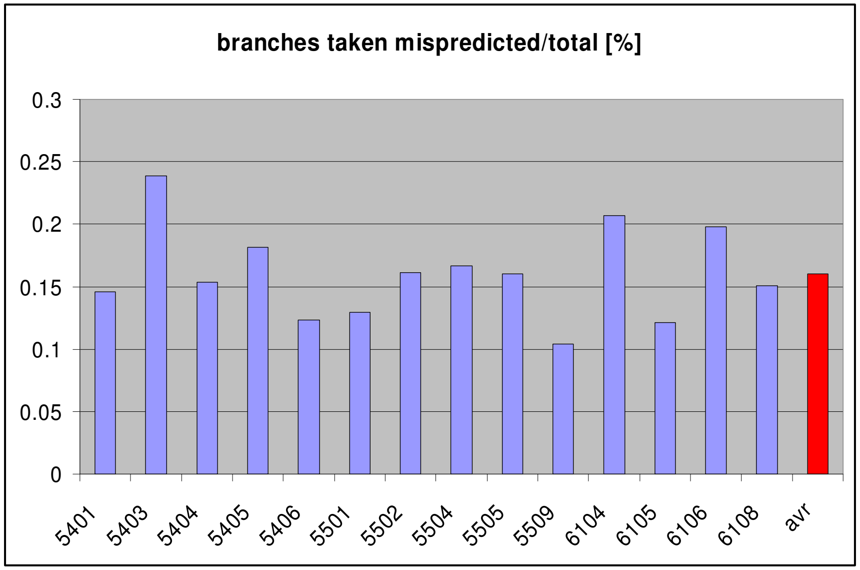




lxbatch - branches







Perfsuite - counters application

- Open source collection of tools, utilities and libraries for software performance analysis
- Hardware support is tightly integrated with PAPI
  - multiplexing
  - user metrics (xml)
  - platforms x86,x86-64, ia64
  - kernel 2.4 & 2.6
- psrun, psprocess
  - single and multi threads programs
  - counting and profiling mode

Perfsuite

- Profiling of Atlas Simulation applications
  - Written in C++, executed from python
  - Many libraries
    - Static
    - Dynamically linked (shared object) (ldd command)
    - Dynamic loaded (libdl – dlopen)
  - Perfsuite has a problem with dynamic loaded libraries
    - LD_PRELOAD – works with simple HelloWorld (dlopen) as a standalone application and with python, but does not work with the full simulation
    - Running the test40 from python (it works) and the profiling– work in progress

Perfsuit & LD_PRELOAD

```
Profile Information
==============================================
Class                    : PAPI
Event                    : PAPI_TOT_CYC (Total cycles)
Period                   : 50000
Samples                    : 719
Domain                    : user
Run Time                   : 17.52 (seconds)
Min Self %                 : (all)
Module Summary
--------------------------------------------------------------------------------
Samples   Self %  Total %  Module
   376   52.29%   52.29%  /usr/bin/python
   178   24.76%   77.05%  /lib/ld-2.3.2.so
   159   22.11%   99.17%  /lib/tls/libc-2.3.2.so
     4    0.56%   99.72%  /lib/tls/libpthread-0.60.so
     1    0.14%   99.86%  /lib/libdl-2.3.2.so
     1    0.14%  100.00%  /lib/libutil-2.3.2.so
Function Summary
--------------------------------------------------------------------------------
Samples   Self %  Total %  Function
   376   52.29%   52.29%  ??
   110   15.30%   67.59%  do_lookup_versioned
    40    5.56%   73.16%  _int_malloc
    31    4.31%   77.47%  strcmp
    22    3.06%   80.53%  _dl_lookup_versioned_symbol
    19    2.64%   83.17%  memcpy
    16    2.23%   85.40%  __libc_malloc
    11    1.53%   86.93%  free
     7    0.97%   87.90%  _int_free
     7    0.97%   88.87%  strlen
     6    0.83%   89.71%  memset
     6    0.83%   90.54%  do_lookup
     5    0.70%   91.24%  malloc_consolidate
     5    0.70%   91.93%  __mempcpy
     4    0.56%   92.49%  __i686.get_pc_thunk.bx
     3    0.42%   92.91%  strerror_r
     3    0.42%   93.32%  mremap_chunk
     3    0.42%   93.74%  _int_realloc
     2    0.28%   94.02%  .L969
     2    0.28%   94.30%  realloc
     2    0.28%   94.58%  mallopt
```

```
Profile Information
==============================================================
Class                    : PAPI
Event                    : PAPI_TOT_CYC (Total cycles)
Period                   : 50000
Samples                    : 721514
Domain                    : user
Run Time                   : 17.60 (seconds)
Min Self %                 : (all)
Module Summary
--------------------------------------------------------------------------------
 Samples   Self %  Total %  Module
  465515   64.52%   64.52%  /afs/cern.ch/user/o/oplaatl3/testdll/libhello2.so.1
  255433   35.40%   99.92%  /afs/cern.ch/user/o/oplaatl3/testdll/libhello1.so.1
     391    0.05%   99.98%  /usr/bin/python
     145    0.02%  100.00%  /lib/tls/libc-2.3.2.so
      26    0.00%  100.00%  /lib/ld-2.3.2.so
       4    0.00%  100.00%  /lib/tls/libpthread-0.60.so
Function Summary
--------------------------------------------------------------------------------
 Samples   Self %  Total %  Function
  255433   35.40%   35.40%  hello(int*)
  254920   35.33%   70.73%  sum(int*)
  210595   29.19%   99.92%  count(int*, int)
     392    0.05%   99.98%  ??
      36    0.00%   99.98%  _int_malloc
      22    0.00%   99.98%  memcpy
      13    0.00%   99.99%  __libc_malloc
      11    0.00%   99.99%  free
      10    0.00%   99.99%  do_lookup_versioned
       7    0.00%   99.99%  strcmp
       6    0.00%   99.99%  __open_nocancel
       5    0.00%   99.99%  _int_free
       4    0.00%   99.99%  memset
       4    0.00%   99.99%  malloc_consolidate
```

Conclusions and plans

- On-chip performance monitoring hardware can give a lot of detailed information and has a lot of applications, like the tuning and the profiling of applications. The big question is how to correctly understand the result and how to take advantage of it.
- One common interface is desirable in order to access the performance units
- gpfmon
  - accuracy of measurement must be investigated in more details in more scenarios,
  - the need for data processing script/application,
  - try to move to the perfmon interface,
  - looking into the counters on other CPU
- Profiling Atlas simulations

Questions

?