



PALPATINE : A profiling analyzer using Linux Perf

Matthieu CARRERE

Thesis - Optimization of HPC code for Gamma Ray experiments

<https://gite.lirmm.fr/cta-optimization-group/palpatine>

Directors : Luisa ARRABITO and Philippe LANGLOIS
Supervisor : David PARELLO

04/07/22

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

Motivations

- ▶ Multiple applications are on the market today : MAQAO, Valgrind, perf linux, gprof, Vtune,...
- ▶ Too intrusive, too much generic, proprietary or reports too complicated to understand without safety low level hardware concepts
- ▶ Uses call graph concept, roofline model or HPC¹
- ▶ Find a balance between **efficiency / precision / ease of use** :
PALPATINE²
 - ▶ **Based on the Linux Perf Tool using performance counters**
 - ▶ **Based on the concept of phase detection**

¹HPC: Hardware performance counters

²PALPATINE: Profiling Analyzer using Linux Perf Applications To Improve an application

Easily

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

Protocol : Material and software

- ▶ Laptop : Dell Latitude 5400 (Ubuntu 20.04 kernel 5.14)

CPU characteristic	Technical details of processor
CPU model	Intel i7-8665U
Cores	1 core used (until 4 cores / 8 threads)
Frequency	2.1Ghz used (until 4.8Ghz)
Cache (L1/L2/L3)	128Ko/1Mo/8Mo
Instructions sets	SSE-AVX-AVX2

- ▶ Python 3.8 + Numpy 1.21 : disabled Pthread for Numpy
- ▶ Gcc 9.4
- ▶ In the bios : disabled TurboBoost, SpeedStep and Hyperthreading

Context

Perf Stat - Matrix product examples

- ▶ Algorithms in C : Naive and Cache/Register Blocking
- ▶ Precision: Double; Size: 800x800 (14.7MB in memory)
- ▶ Commands : \$perf stat -e \$EVENTS ./naive-O3.out // blocking-O3.out

```
Performance counter stats for './naive-O3.out':
2560055536    cycles
4139788660  instructions 1,62 insn per cycle
984026446      mem_load_retired.l1_hit
518365089      mem_load_retired.l1_miss
467396259      mem_load_retired.l2_hit
50298243       mem_load_retired.l2_miss
48010870       mem_load_retired.l3_hit
2877750       mem_load_retired.l3_miss
 0,633742787 seconds time elapsed
 0,629755000 seconds user
 0,003985000 seconds sys
```

```
for './blocking-O3.out':
624347581    cycles
2166757751  instructions 3,47 insn per cycle
668691220      mem_load_retired.l1_hit
4097486      mem_load_retired.l1_miss
3113575      mem_load_retired.l2_hit
982984       mem_load_retired.l2_miss
946266       mem_load_retired.l3_hit
77766       mem_load_retired.l3_miss
 0,150127155 seconds time elapsed
 0,146067000 seconds user
 0,004057000 seconds sys
```

Version	Speedup	IPC	L1 ratio	L2 ratio	L3 ratio
Naive	1.00	1.62	35%	10%	6%
Blocking	4.22	3.47	1%	24%	8%

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

PALPATINE

Linux Perf

Phase detection

Backups

Summary

PALPATINE

Linux Perf

Phase detection

PALPATINE : A profiling analyzer using Linux Perf

Concepts

PALPATINE : A
profiling analyzer
using Linux Perf

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

- ▶ Developed in two steps :
 - ▶ Firstly : collect directly the counters with original perf modules
 - ▶ Secondly : collect the counters with the **phase detection** concept
- ▶ Collects & Reads performance counters on different scales

PALPATINE : A profiling analyzer using Linux Perf

Read in three scales

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Scale	Information
Overall application	Collect general counters Calculate different metrics (IPC, branch misses,...)
Overall functions	Sort functions by overhead for different events
Overall hotspots by function	Find some assembly code lines and compare with event's overheads

- ▶ Provide solutions (if possible) for an optimization
- ▶ Estimate a potential benefit
 - ▶ Ex: scalar instruction versus vectorial instruction

PALPATINE : A profiling analyzer using Linux Perf

Test Protocol : Conjugate Gradient

► Conjugate Gradient : $Ax = b$

- A : a symmetric square sparse matrix with the intensity of a monochromatic plane wave in 1D ¹
- b : a vector equal to $(f(x))$
- x : the vector to find equal to $(1\dots1)$ at the end
- Use an algorithm in python with the Numpy module (**1 matrix/vector product and 4 vector products by iteration**)
- PALPATINE / Phase detection tests :
 - Nb iterations $\approx 5200/3000$
 - Elapsed time $\approx 34/7$ seconds

¹ $f(x) = |Amp \cdot \cos(\frac{2\pi \cdot x}{\lambda})|^2$

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

PALPATINE : A profiling analyzer using Linux Perf

Conjugate gradient - Stat example

PALPATINE : A
profiling analyzer
using Linux Perf

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

counter	conjugate_gradient
IPC	1.68
Vectorization rate (double)	1.0
Arith instructions rate	0.8
Arith instructions by cycle	1.35
Arith operations by cycle	5.39
cache-misses-ratio	0.46
cycles	6.97e+10
instructions	1.17e+11
cache-misses	2.15e+09
cache-references	4.66e+09
fp_arith_inst_retired.scalar_double	9.43e+07
fp_arith_inst_retired.256b_packed_double	9.38e+10

PALPATINE : A profiling analyzer using Linux Perf

Conjugate gradient - Record Report example

- ▶ The most important function found : `mkl_bla...xdgemv_t1`

Counter	Usage	Events	functionSamples	library
cycles	94.69%	6.81e+10	138000	libmkl_avx2.so.1
instructions	95.48%	1.12e+11	139000	libmkl_avx2.so.1
cache-misses	98.37%	2.18e+09	139000	libmkl_avx2.so.1
cache-references	92.33%	4.41e+09	128000	libmkl_avx2.so.1
scalar	100.00%	9.2e+07	46	libmkl_avx2.so.1
vector	99.86%	9.37e+10	46800	libmkl_avx2.so.1

- ▶ scalar: `fp_arith_inst_retired.scalar_double`
- ▶ vector: `fp_arith_inst_retired.256b_packed_double`

¹mkl : Intel Math Kernel Library

Linux Perf

Hardware Performance Counters (HPC)

- ▶ Counters provided by CPU to a dedicated unit : Performance Monitoring unit (PMU)
 - ▶ Set of special registers build into wide variety of micro-architectures of processors (x86/x64, ARM, ...)
 - ▶ Limited number of registers
- ▶ Records **events** related to HPC in registers
 - ▶ Event examples : cycles, instructions, cache-misses
- ▶ A good way to profile and understand the behavior of a code while an execution

Linux Perf

Linux Perf Tool

- ▶ Perf Linux : Linux profiling tool that interfaces performance counters recovery with the operating system.
- ▶ Stat Module :
 - ▶ **Counts all selected events** for a processus
 - ▶ Multiplexing if : sum of events selected > number of material registers
 - ▶ Some options like topdown, interval, ...
- ▶ Record Module :
 - ▶ **Samples selected events** and records IP instructions to compare events with assembly/source code for a processus
 - ▶ Some options like call-graph, recovery frequency, ...
 - ▶ Overhead more important and output files can be huge

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Linux Perf

Perf Stat - Matrix product examples

- ▶ Algorithms in C : Naive and Cache/Register Blocking
- ▶ Precision: Double; Size: 800x800 (14.7MB in memory)
- ▶ Commands : \$perf stat -e \$EVENTS ./naive-O3.out // blocking-O3.out

```
Performance counter stats for './naive-O3.out':  
2560055536 cycles  
4139788660 instructions 1,62 insn per cycle  
984026446 mem_load_retired.l1_hit  
518365089 mem_load_retired.l1_miss  
467396259 mem_load_retired.l2_hit  
50298243 mem_load_retired.l2_miss  
48010870 mem_load_retired.l3_hit  
2877750 mem_load_retired.l3_miss  
0,633742787 seconds time elapsed  
0,629755000 seconds user  
0,003985000 seconds sys
```

```
for './blocking-O3.out':  
624347581 cycles  
2166757751 instructions 3,47 insn per cycle  
668691220 mem_load_retired.l1_hit  
4097486 mem_load_retired.l1_miss  
3113575 mem_load_retired.l2_hit  
982984 mem_load_retired.l2_miss  
946266 mem_load_retired.l3_hit  
77766 mem_load_retired.l3_miss  
0,150127155 seconds time elapsed  
0,146067000 seconds user  
0,004057000 seconds sys
```

Version	Speedup	IPC	L1 ratio	L2 ratio	L3 ratio
Naive	1.00	1.62	35%	10%	6%
Blocking	4.22	3.47	1%	24%	8%

Linux Perf

Perf Record - Perf Report (left) and Annotate (right) - Matrix product example - Naive

PALPATINE : A
profiling analyzer
using Linux Perf

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Samples: 5K of event 'cycles', Event count (approx.): 2113715930			
Children	Self	Command	Shared Object Symbol
+ 99,83%	0,00%	dg-select-03.out	dg-select-03.out [] _start
+ 99,83%	0,00%	dg-select-03.out	libc-2.31.so [] __libc_start_main
+ 99,83%	0,00%	dg-select-03.out	dg-select-03.out [] main
+ 99,83% 99,05%	dg-select-03.out	dg-select-03.out	dgemm
0,39%	0,04%	dg-select-03.out	[kernel.kallsyms] [k] asm_sysvec_apic_timer_interrupt
0,34%	0,00%	dg-select-03.out	[kernel.kallsyms] [k] sysvec_apic_timer_interrupt
0,25%	0,02%	dg-select-03.out	[kernel.kallsyms] [k] asm_sysvec_irq_work
0,25%	0,00%	dg-select-03.out	[kernel.kallsyms] [k] __sysvec_apic_timer_interrupt
0,23%	0,00%	dg-select-03.out	[kernel.kallsyms] [k] hrtimer_interrupt
0,23%	0,00%	dg-select-03.out	[kernel.kallsyms] [k] sysvec_irq_work
0,20%	0,00%	dg-select-03.out	[kernel.kallsyms] [k] __hrtimer_run_queues

Samples: 5K of event 'cycles', 4000 Hz, Event count (approx.): 2113715930

Percent

```
3 Déassembly de la section .text :
5 00000000000001a40 <dgemm>:
6 dgemm():
7     return 1;
8 }

9 // Matrix multiplication
10 void dgemm(double **c, double **a, double **b, size_t size)
11 {
12     endbr64
13     for (int i=0; i<size; i++)
14         test    %rcx,%rcx
15         jne    78
16         mov    %rdi,%r11
17         mov    %rcx,%rdi
18     {
19         push   %rbx
20         mov    %rdx,%r9
21         mov    %r11,%rbx
22         shl    $0x3,%rdl
23     for (int i=0; i<size; i++)
24         xor    %r10d,%r10d
25         nop
26     }
27     for (int j=0; j<size; j++)
28     {
29         for (int k=0; k<size; k++)
30         {
31             c[i][j]=c[i][j] + a[i][k] * b[k][j];
32             mov    (%rbx,%r10,i),%rb
33             mov    (%r11,%r10,i),%r11
34             xor    %rcx,%rcx
35             nop
36         for (int k=0; k<size; k++)
37             vmovsd (%rsi),%xmm1
38         }
39         xor    %eax,%eax
40         c[i][j]=c[i][j] + a[i][k] * b[k][j];
41         mov    (%r9,%rax,i),%rdx
42         vmovsd (%rdx,%rcx,i),%xmm0
43         vmlsd (%rb,%rax,i),%xmm0,%xmm0
44         add    $0x8,%rax
45         vaddsd %xmm0,%xmm1,%xmm1
46         vmove (%r11,%xmm1)
47         vmove (%xmm1,%r11)
48     for (int k=0; k<size; k++)
49         cmp    %rax,%rdi
50         jne    40
51 }
```

Linux Perf

Perf Overhead - Matrix product example - Time

PALPATINE : A profiling analyzer using Linux Perf

Matthieu CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

▶ Naive Algorithms - 8 events

Analysis	Output Size	Time(s)	Overhead factor
N/A	N/A	1.53	x1.0
Perf stat	1 KB	1.77	x1.2
Perf record	183 MB	4.47	x2.9

▶ Blocking Algorithms - 8 events

Analysis	Output Size	Time(s)	Overhead factor
N/A	N/A	0.32	x1.0
Perf stat	1 KB	0.49	x1.5
Perf record	21 MB	2.35	x7.3

Phase detection

A phase detection in Linux Perf

Matthieu
CARRERE

- ▶ **Idea** : include the phase detection in Linux Perf Tool
- ▶ **Goal** : profile **differently** and more **efficiently** with performance counters
 - ▶ Reduce the overhead created by the collect
 - ▶ Reduce the size of output data
 - ▶ Reduce the analysis time for the human
 - ▶ Improve the analysis precision
- ▶ **Principle** : Identify and sort while the runtime of an application different behaviors change

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

Definitions

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

- ▶ **Interval** : series of executed instructions delimited by a time interval
 - ▶ Events counted for HPC
 - ▶ 2 instruction positions
- ▶ **Phase** : series of executed instructions (intervals) with same properties
 - ▶ Create a phase : $|\text{variation rate}|_{\text{event}} \geq \text{limit}_{\text{event}}$
 - ▶ Group an interval in a phase : $|\text{variation rate}|_{\text{event}} < \text{limit}_{\text{event}}$

Context

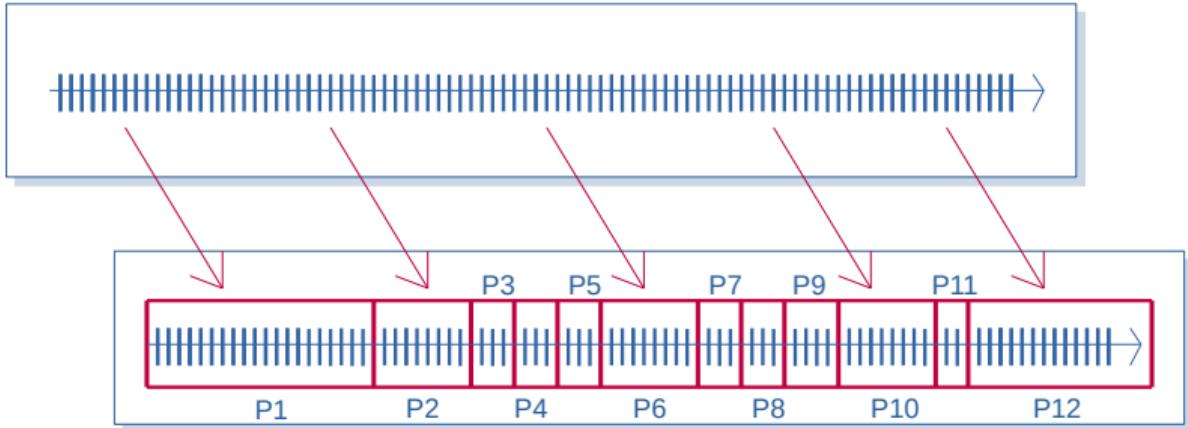
PALPATINE

Linux Perf

Phase detection

Backups

perf stat --phase-detection



Legend

| : interval

→ : timeline



: phase PX

Phase detection

Perf Stat - Method

Matthieu
CARRERE

- ▶ Recover events by time interval
- ▶ Events and metric used :
 - ▶ **IPC**¹ : instructions per cycle
 - ▶ **Branch-misses**² : bad prediction for a branch
- ▶ Different techniques allow to differentiate the phases

Context

PALPATINE

Linux Perf

Phase detection

Backups

¹Sherwood, Timothy, Suleyman Sair, and Brad Calder. "Phase tracking and prediction." ACM SIGARCH Computer Architecture News 31.2 (2003): 336-349.

²Balasubramonian, Rajeev, et al. "Memory hierarchy reconfiguration for energy and performance in general-purpose processor architectures." Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture. (2000).

Phase detection

Perf Stat - Differentiate phases

- ▶ 2 steps to differentiate phases :
 - ▶ Calculate the variation rate for an event or a metric at each time interval
 - ▶ Determine a limit for these variations to know if we are in a same phase or a new phase (**fixed limit**)
- ▶ 2 techniques to calculate the variation rate :

- ▶ **Naive** :

$$\tau_{br}(i) = \left| \frac{BM(i)}{BM(i-1)} - 1 \right| ; \quad \tau_{IPC}(i) = |IPC(i) - IPC(i-1)| \quad (1)$$

- ▶ **Median** :

$$\tau_{br}(i) = \left| \frac{BM(i)}{\overline{Median}_{br, j\&j-1}} - 1 \right| ; \quad \tau_{IPC}(i) = |IPC(i) - \overline{Median}_{IPC, j\&j-1}| \quad (2)$$

Matthieu CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

Test Protocol: Conjugate Gradient - Experiments

- ▶ An extensive series of comparisons has been carried out with the conjugate gradient calculation like first results :
 - ▶ Interval / Phases buffer sizes : [4;128] / [128;8096]
 - ▶ Methods : naive / median
 - ▶ Events/metric : IPC / BM / IPC&BM
 - ▶ Time interval : [1;8]ms
 - ▶ IPC / BM rate variation limits : [0.;1.] / [0.;1.]

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

First results: Comparison between multiple parameters

- ▶ 10 samples, time_{intervals}: 1ms

Versions	IPC limit	BM limit	Max(mean diff)	Nb Phases
Naive BM	N/A	[0.5;1.]	10%	[80;209]
Naive IPC	[0.2;1.]	N/A	14%	[27;474]
Naive IPCBM	[0.2;0.6]	[0.5;0.6]	9%	[83;169]
Median BM	N/A	[0.5;0.8]	12%	[76;144]
Median IPC	[0.6;0.9]	N/A	15%	[29;99]
Median IPCBM	[0.6;0.7]	[0.5;0.7]	10%	[55;85]

- ▶ Commune variation rate limits (IPC , BM) : (0.6 , 0.5)

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

PALPATINE

Linux Perf

Phase detection

Backups

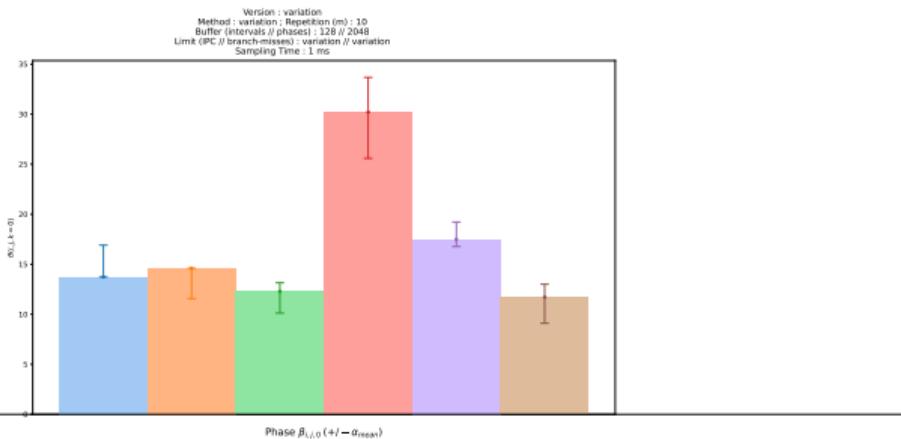
Phase detection

First results: version comparison for (IPC , BM) : (0.6 , 0.5) - Phases

$$\theta(i, j, k=0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k=0) = \frac{100 \cdot |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k=0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$



conjugate_gradient_median_bm_128_2048_0_0_1	9.500e+01 (+/-14.39%)
conjugate_gradient_median_ipc_128_2048_0.6_0_1	1.0100e+02 (+/-11.11%)
conjugate_gradient_median_ipcbm_128_2048_0.6_0_1	8.5000e+01 (+/-4.31%)
conjugate_gradient_naive_bm_128_2048_0_0_1	2.0900e+02 (+/-10.10%)
conjugate_gradient_naive_ipc_128_2048_0.6_0_1	1.2100e+02 (+/-5.88%)
conjugate_gradient_naive_ipcbm_128_2048_0.6_0_1	8.1000e+01 (+/-7.68%)

Context

PALPATINE

Linux Perf

Phase detection

Backups

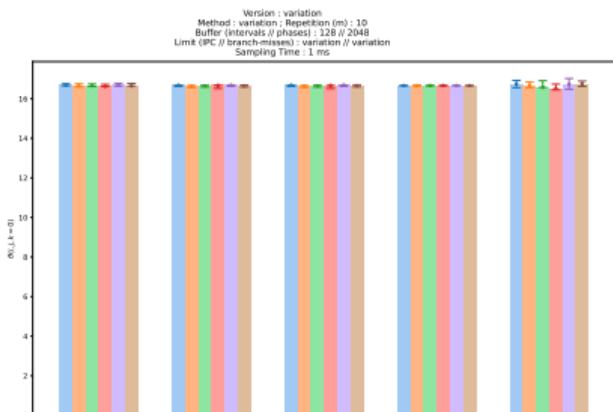
Phase detection

First results: version comparison for (IPC , BM) : (0.6 , 0.5) - Other parameters

$$\theta(i, j, k = 0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k = 0) = \frac{100 \cdot |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k = 0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$



	Intervals $\beta_{i,j,0} (+/- \sigma_{mean})$	Duration(s) $\beta_{i,j,0} (+/- \sigma_{mean})$	Cycles $\beta_{i,j,0} (+/- \sigma_{mean})$	Instructions $\beta_{i,j,0} (+/- \sigma_{mean})$	Branch-misses $\beta_{i,j,0} (+/- \sigma_{mean})$
conjugate_gradient_median_bm_128_2048_0_0_5_1	6.2960e+03 (+/-0.19%)	7.1169e+00 (+/-0.20%)	1.3449e+10 (+/-0.20%)	2.0927e+10 (+/-0.01%)	1.9847e+07 (+/-0.60%)
conjugate_gradient_median_ipc_128_2048_0_6_0_1	6.2880e+03 (+/-0.34%)	7.0979e+00 (+/-0.14%)	1.3406e+10 (+/-0.15%)	2.0929e+10 (+/-0.03%)	1.9762e+07 (+/-0.49%)
conjugate_gradient_median_ipcbm_128_2048_0_6_0_5_1	6.2840e+03 (+/-0.31%)	7.1038e+00 (+/-0.15%)	1.3426e+10 (+/-0.15%)	2.0932e+10 (+/-0.01%)	1.9642e+07 (+/-1.08%)
conjugate_gradient_naive_bm_128_2048_0_5_1	6.2720e+03 (+/-0.34%)	7.1144e+00 (+/-0.45%)	1.3441e+10 (+/-0.43%)	2.0934e+10 (+/-0.04%)	1.9583e+07 (+/-0.86%)
conjugate_gradient_naive_ipc_128_2048_0_6_0_1	6.3080e+03 (+/-0.23%)	7.1193e+00 (+/-0.21%)	1.3454e+10 (+/-0.21%)	2.0932e+10 (+/-0.01%)	1.9824e+07 (+/-0.52%)
conjugate_gradient_naive_ipcbm_128_2048_0_6_0_5_1	6.2790e+03 (+/-0.51%)	7.1036e+00 (+/-0.15%)	1.3425e+10 (+/-0.16%)	2.0928e+10 (+/-0.02%)	1.9852e+07 (+/-0.33%)

Context

PALPATINE

Linux Perf

Phase detection

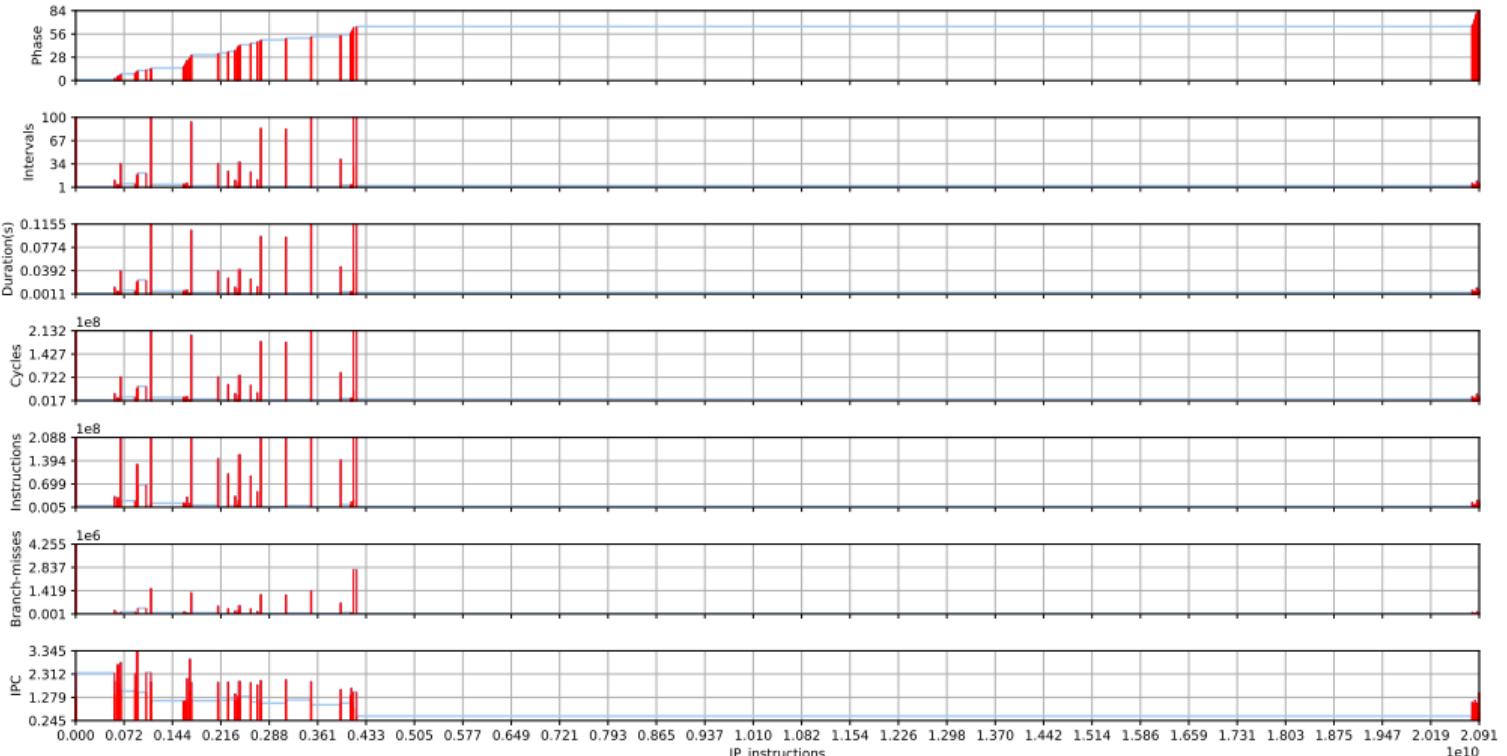
Backups

Phase detection

First results: Median IPCBM - 1 sample with phases for (IPC , BM) : (0.6 , 0.5)

Perf stat with detection phase option output for the conjugate gradient example

median_ipcbm_128_2048_0.6_0.5_1_data0



Context

PALPATINE

Linux Perf

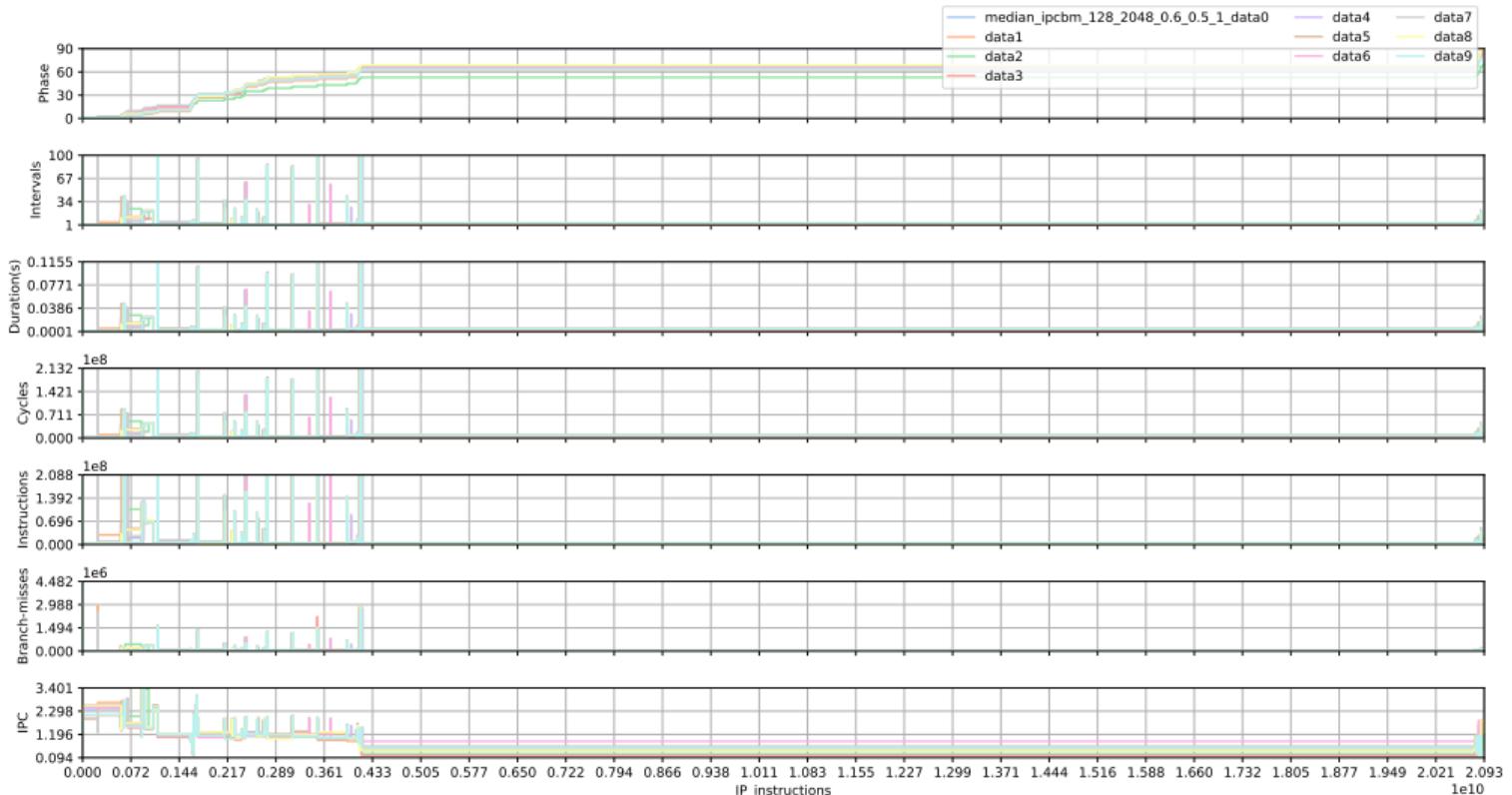
Phase detection

Backups

Phase detection

First results: Median IPCBM - 10 samples for (IPC , BM) : (0.6 , 0.5)

Perf stat with detection phase option output for the conjugate gradient example



Phase detection

Next steps

Matthieu
CARRERE

- ▶ **Multiply experiments** on different scientific codes (C/C++ codes)
 - ▶ Provide right limits for metrics/HPC (fixed or dynamic)
 - ▶ Estimate the best choice between : IPC, BM or IPC&BM
 - ▶ Provide the best technique to differentiate phases
- ▶ **Use instruction's address** with the phase detection option
- ▶ Merge this work with PALPATINE

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

PALPATINE

Linux Perf

Phase detection

Backups

Thank you for your attention

Context

PALPATINE

Linux Perf

Phase detection

Backups

Backups

Creation of metrics : Vectorization Ratio

- ▶ Possibility to combine HPC to create metrics like the vectorization ratio τ_{vec}
- ▶ Count floating-point instructions in single and double precision
- ▶ Differentiate scalar and vector instructions (avx,avx2,avx512)

$$O_{vec} = \sum(2 * I_{128,dbl} + 4 * I_{256,dbl} + 8 * I_{512,dbl}) \quad (3)$$

$$O_{scal} = \sum I_{scal,dbl} ; \quad \boxed{\tau_{vec} = \frac{O_{vec}}{O_{vec} + O_{scal}}} \quad (4)$$

Linux Perf

Metrics - CORSIKA7's Optimization Work¹

PALPATINE : A profiling analyzer using Linux Perf

Matthieu CARRÈRE

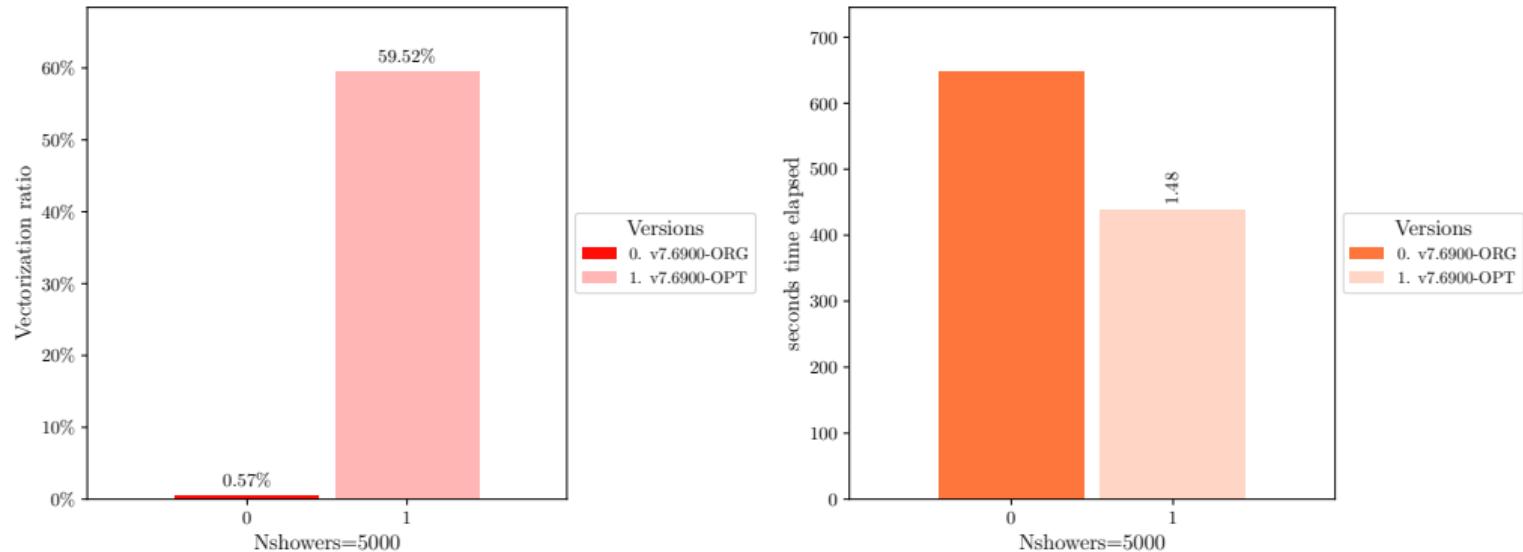
Context

PALPATINE

Linux Perf

Phase detection

Backups



¹Arrabito, L., Bernlöhr, K., Bregeon, J., Carrère, M., Khattabi, A., Langlois, P., ... Revy, G. (2020). Optimizing Cherenkov Photons Generation and Propagation in CORSIKA for CTA Monte–Carlo Simulations. Computing and Software for Big Science, 4(1), 1–14.

CORSIKA 7's experiment

Hardware and compile

- ▶ Hardware : Dali512 Server (2x Intel Xeon Gold 5122)

Machine	Technical details of processor
Dali512 Server	Cores : 4 (8 logical) Frequency : 3.6Ghz Cache (L1/L2/L3) : 64Ko/1Mo/16Mo Instructions sets : AVX-AVX2-AVX512

- ▶ Compiler : Gcc/Gfortran 8.3
- ▶ Flags AVX2 : -O3 -mavx2 -std=C99
- ▶ Flags AVX512 : -O3 -mavx512f -ffp-contract=off -std=C99

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

CORSIKA 7's experiment

Typical experiment

Matthieu
CARRERE

Parameter	Value
Incident particule	gamma
Interaction	electromagnetic
Showers	5000
Total energy	3 GeV to 330 TeV (pow law)
Incidence angles	20° (zenith) and 180° (azimuth)
Cut energy	20 MeV
Earth's magnetic field	21 μ T
Atmosphere heigh	112.8 km
Observator heigh	2.15 km
Telescopes	99
Telescope grid	10
Seed	fixed

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

Test protocol

PALPATINE : A
profiling analyzer
using Linux Perf

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

- ▶ Disabled Intel SpeedStep
- ▶ Disabled Intel Speed Shift Technology
- ▶ Disabled Intel TurboBoost
- ▶ Disabled HyperThread control

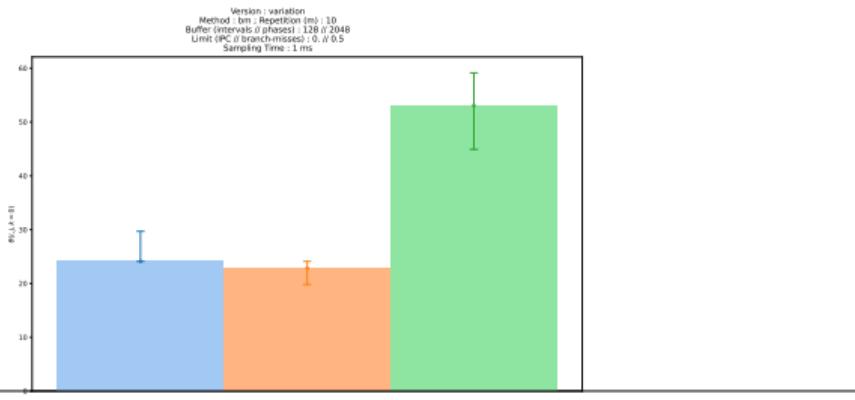
Phase detection

Histogram - version comparison for BM

$$\theta(i, j, k=0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k=0) = \frac{100 \cdot [min(max(\beta_{i,j,0:m-1}) - \beta_{i,j,k})]}{\sum_{l=0}^{n-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k=0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$



conjugate_gradient_median_bm_128_2048_0_0.5_1	9.5000e+01 (+/-14.39%)
conjugate_gradient_medianAdaptative_bm_128_2048_0_0.5_1	9.0000e+01 (+/-4.94%)
conjugate_gradient_naive_bm_128_2048_0_0.5_1	2.0900e+02 (+/-10.10%)

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

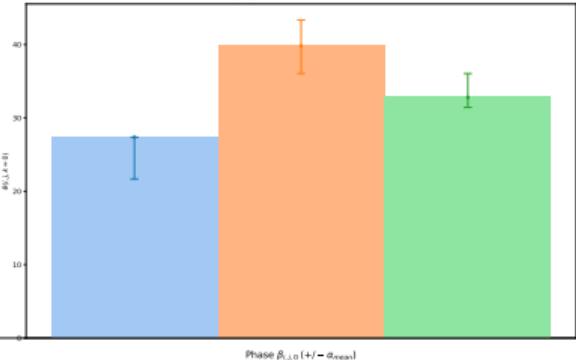
Histogram - version comparison for IPC

$$\theta(i, j, k=0) = \frac{100. * \beta_{i,j,k}}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k=0) = \frac{100. * |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k=0) = \frac{100. * \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$

Version : xenon
Method : ipc_overhead (ms) : 10
Buffer (intercept) phases : 128 / 2048
Limit (IPC // branch-misses) : 0.6 / 0.
Sampling Time : 1 ms



conjugate_gradient_median_ipc_128_2048_0_6_0_1	1.0100e+02 (+/-11.11%)
conjugate_gradient_medianAdaptive_ipc_128_2048_0_6_0_1	1.4700e+02 (+/-4.54%)
conjugate_gradient_naive_ipc_128_2048_0_6_0_1	1.2100e+02 (+/-5.88%)

Context

PALPATINE

Linux Perf

Phase detection

Backups

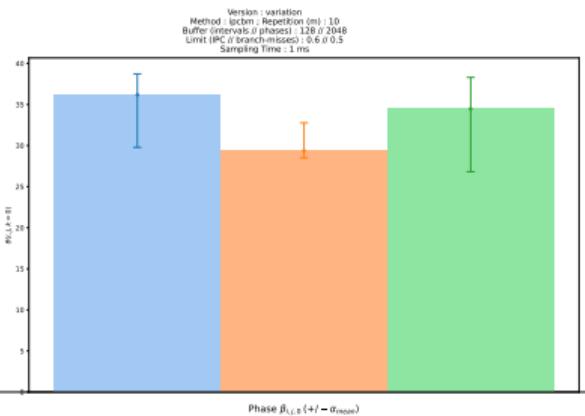
Phase detection

Histogram - version comparison for IPCBM

$$\theta(i, j, k = 0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k = 0) = \frac{100 \cdot |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k = 0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$



	Phase $\beta_{i,j,k} (+/- \sigma_{mean})$
conjugate_gradient_median_ipcbm_128_2048_0.6_0.5_1	8.5000e+01 (+/-4.31%)
conjugate_gradient_medianAdaptative_ipcbm_128_2048_0.6_0.5_1	6.9000e+01 (+/-3.86%)
conjugate_gradient_naive_ipcbm_128_2048_0.6_0.5_1	8.1000e+01 (+/-7.68%)

Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

PALPATINE

Linux Perf

Phase detection

Backups

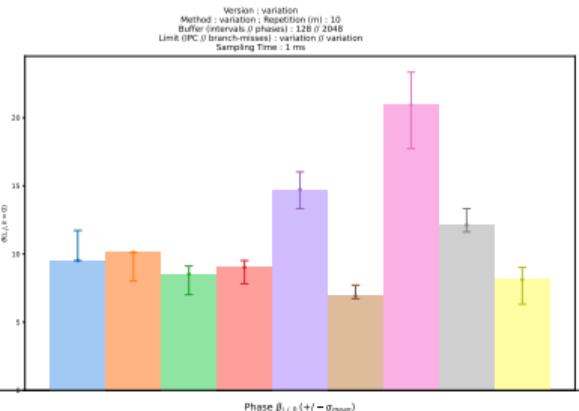
Phase detection

Histogram - version comparison for all

$$\theta(i, j, k=0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{min/max}(i, j, k=0) = \frac{100 \cdot |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{m-1} \beta_{i,l,k}}$$

$$\alpha_{mean}(i, j, k=0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,l} - \beta_{i,j,k}|}{m}$$



Method	Mean ($+/- \sigma_{mean}$)
conjugate_gradient_median_bm_128_2048_0_0_1	9.5000e+01 ($+/-14.39\%$)
conjugate_gradient_median_ipc_128_2048_0_6_0_1	1.0100e+02 ($+/-11.11\%$)
conjugate_gradient_median_ipcbm_128_2048_0_6_0_1	8.5000e+01 ($+/-4.31\%$)
conjugate_gradient_medianAdaptative_bm_128_2048_0_0_1	9.0000e+01 ($+/-4.94\%$)
conjugate_gradient_medianAdaptative_ipc_128_2048_0_6_0_1	1.4700e+02 ($+/-4.54\%$)
conjugate_gradient_medianAdaptative_ipcbm_128_2048_0_6_0_1	6.9000e+01 ($+/-3.86\%$)
conjugate_gradient_naive_bm_128_2048_0_0_1	2.0900e+02 ($+/-10.10\%$)
conjugate_gradient_naive_ipc_128_2048_0_6_0_1	1.2100e+02 ($+/-5.88\%$)
conjugate_gradient_naive_ipcbm_128_2048_0_6_0_1	8.1000e+01 ($+/-7.68\%$)

Context

PALPATINE

Linux Perf

Phase detection

Backups

Phase detection

Histogram - version comparison for all - other parameters

$$\theta(i, j, k=0) = \frac{100 \cdot \beta_{i,j,k}}{\sum_{l=0}^{m-1} \beta_{i,j,l}}$$

$$\alpha_{min/max}(i, j, k=0) = \frac{100 \cdot |min/max(\beta_{i,j,0:m-1}) - \beta_{i,j,k}|}{\sum_{l=0}^{m-1} \beta_{i,j,l}}$$

$$\alpha_{mean}(i, j, k=0) = \frac{100 \cdot \sum_{l=0}^{m-1} |\beta_{i,j,l} - \beta_{i,j,k}|}{m}$$



	Intervals $\beta_{i,j,0} (+/- \sigma_{mean})$	Duration(s) $\beta_{i,j,0} (+/- \sigma_{mean})$	Cycles $\beta_{i,j,0} (+/- \sigma_{mean})$	Instructions $\beta_{i,j,0} (+/- \sigma_{mean})$	Branch-misses $\beta_{i,j,0} (+/- \sigma_{mean})$
conjugate_gradient_median_bm_128_2048_0_0_5_1	6.2960e+03 (+/-0.19%)	7.1169e+00 (+/-0.20%)	1.3449e+10 (+/-0.20%)	2.0927e+10 (+/-0.01%)	1.9847e+07 (+/-0.60%)
conjugate_gradient_median_ipc_128_2048_0_6_0_1	6.2880e+03 (+/-0.34%)	7.0979e+00 (+/-0.14%)	1.3406e+10 (+/-0.15%)	2.0929e+10 (+/-0.03%)	1.9762e+07 (+/-0.49%)
conjugate_gradient_median_ipcbm_128_2048_0_6_0_5_1	6.2840e+03 (+/-0.31%)	7.1038e+00 (+/-0.15%)	1.3426e+10 (+/-0.15%)	2.0932e+10 (+/-0.01%)	1.9642e+07 (+/-1.08%)
conjugate_gradient_medianAdaptative_bm_128_2048_0_0_5_1	6.2640e+03 (+/-0.69%)	7.0664e+00 (+/-0.54%)	1.3356e+10 (+/-0.53%)	2.0931e+10 (+/-0.01%)	1.9876e+07 (+/-0.36%)
conjugate_gradient_medianAdaptative_ipc_128_2048_0_6_0_1	6.2930e+03 (+/-0.24%)	7.1231e+00 (+/-0.32%)	1.3435e+10 (+/-0.17%)	2.0926e+10 (+/-0.03%)	1.9823e+07 (+/-0.30%)
conjugate_gradient_medianAdaptative_ipcbm_128_2048_0_6_0_5_1	6.2910e+03 (+/-0.26%)	7.0960e+00 (+/-0.12%)	1.3413e+10 (+/-0.13%)	2.0929e+10 (+/-0.01%)	1.9822e+07 (+/-0.42%)
conjugate_gradient_naive_bm_128_2048_0_0_5_1	6.2720e+03 (+/-0.34%)	7.1144e+00 (+/-0.45%)	1.3441e+10 (+/-0.43%)	2.0934e+10 (+/-0.04%)	1.9583e+07 (+/-0.86%)
conjugate_gradient_naive_ipc_128_2048_0_6_0_1	6.3080e+03 (+/-0.23%)	7.1193e+00 (+/-0.21%)	1.3454e+10 (+/-0.21%)	2.0932e+10 (+/-0.01%)	1.9824e+07 (+/-0.52%)
conjugate_gradient_naive_ipcbm_128_2048_0_6_0_5_1	6.2790e+03 (+/-0.51%)	7.1036e+00 (+/-0.15%)	1.3425e+10 (+/-0.16%)	2.0928e+10 (+/-0.02%)	1.9852e+07 (+/-0.33%)

Context

PALPATINE

Linux Perf

Phase detection

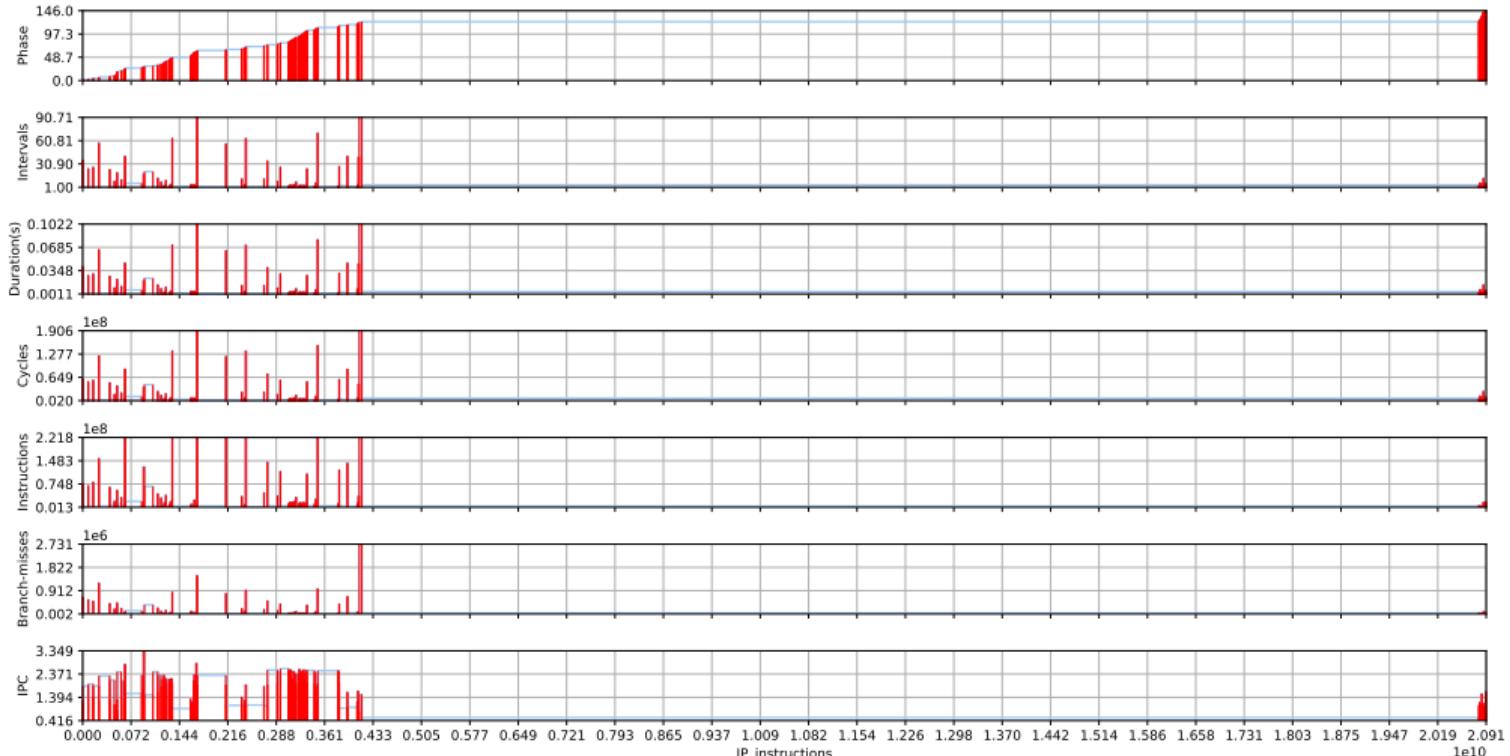
Backups

Phase detection

Graph - Median Adaptive IPC - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

medianAdaptive_ipc_128_2048_0.6_0_1_data0



Context

PALPATINE

Linux Perf

Phase detection

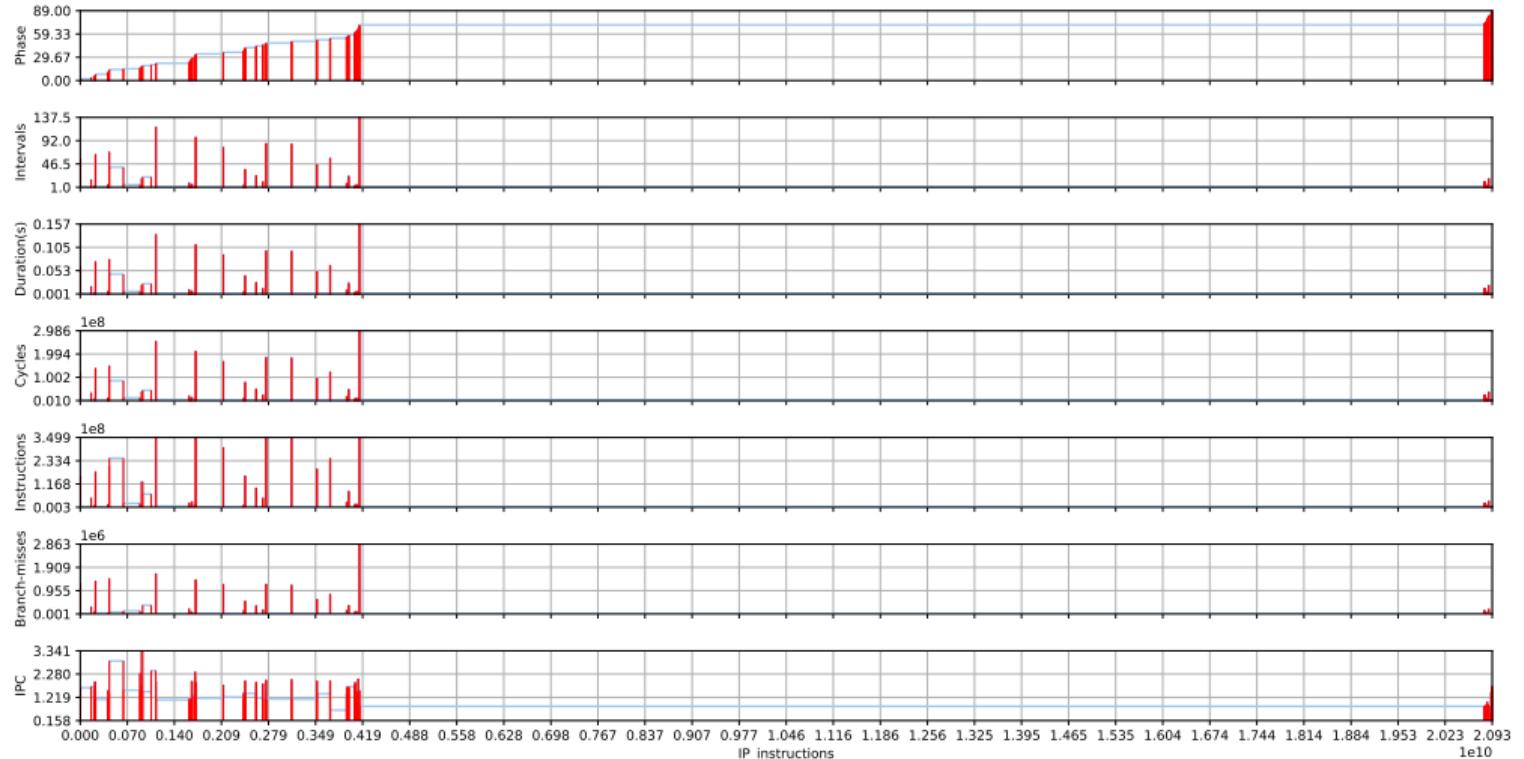
Backups

Phase detection

Graph - Median Adaptative BM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

medianAdaptative_bm_128_2048_0_0.5_1_data0

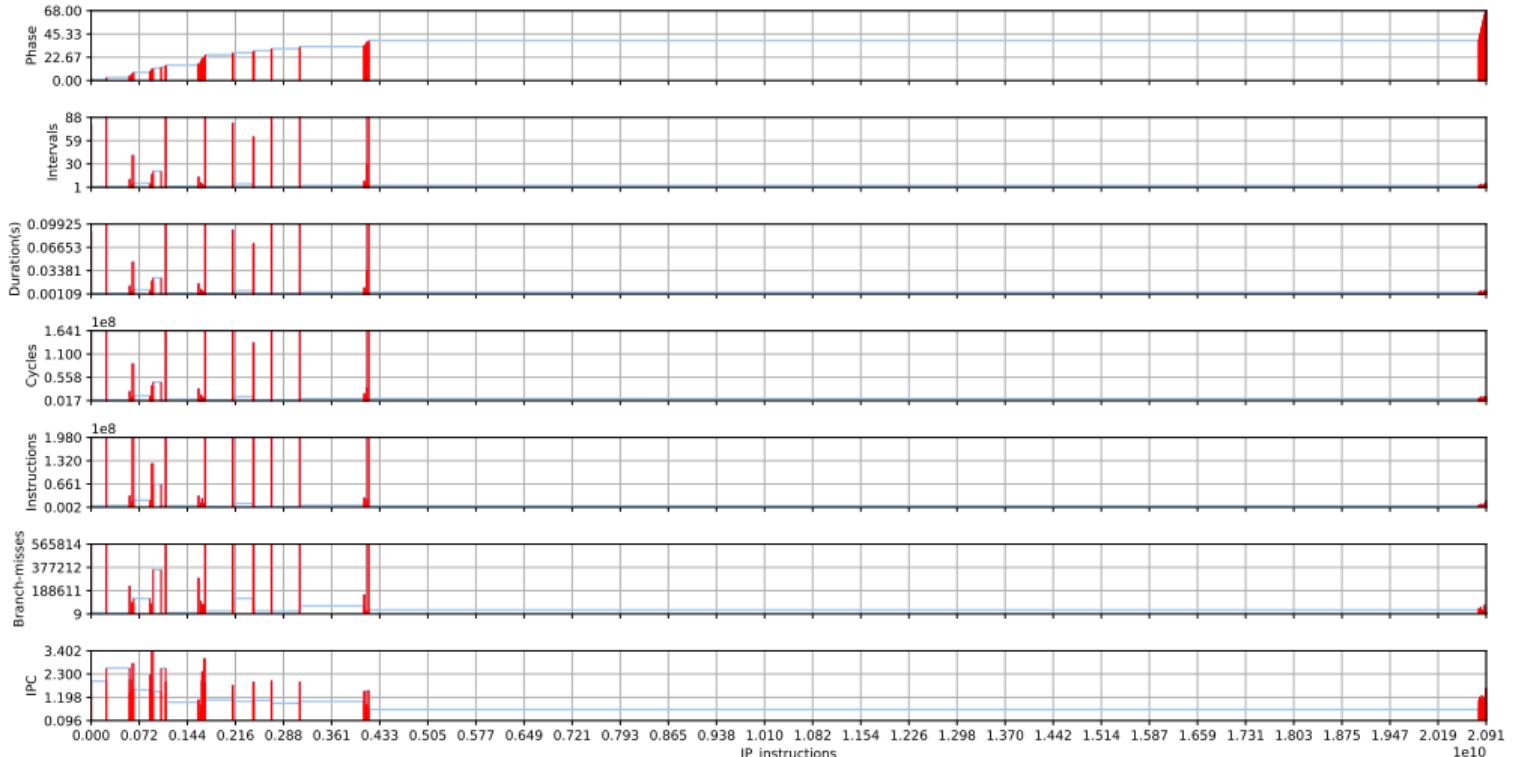


Phase detection

Graph - Median Adaptive IPCBM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

medianAdaptive_ipcbm_128_2048_0.6_0.5_1_data0



Context

PALPATINE

Linux Perf

Phase detection

Backups

Context

PALPATINE

Linux Perf

Phase detection

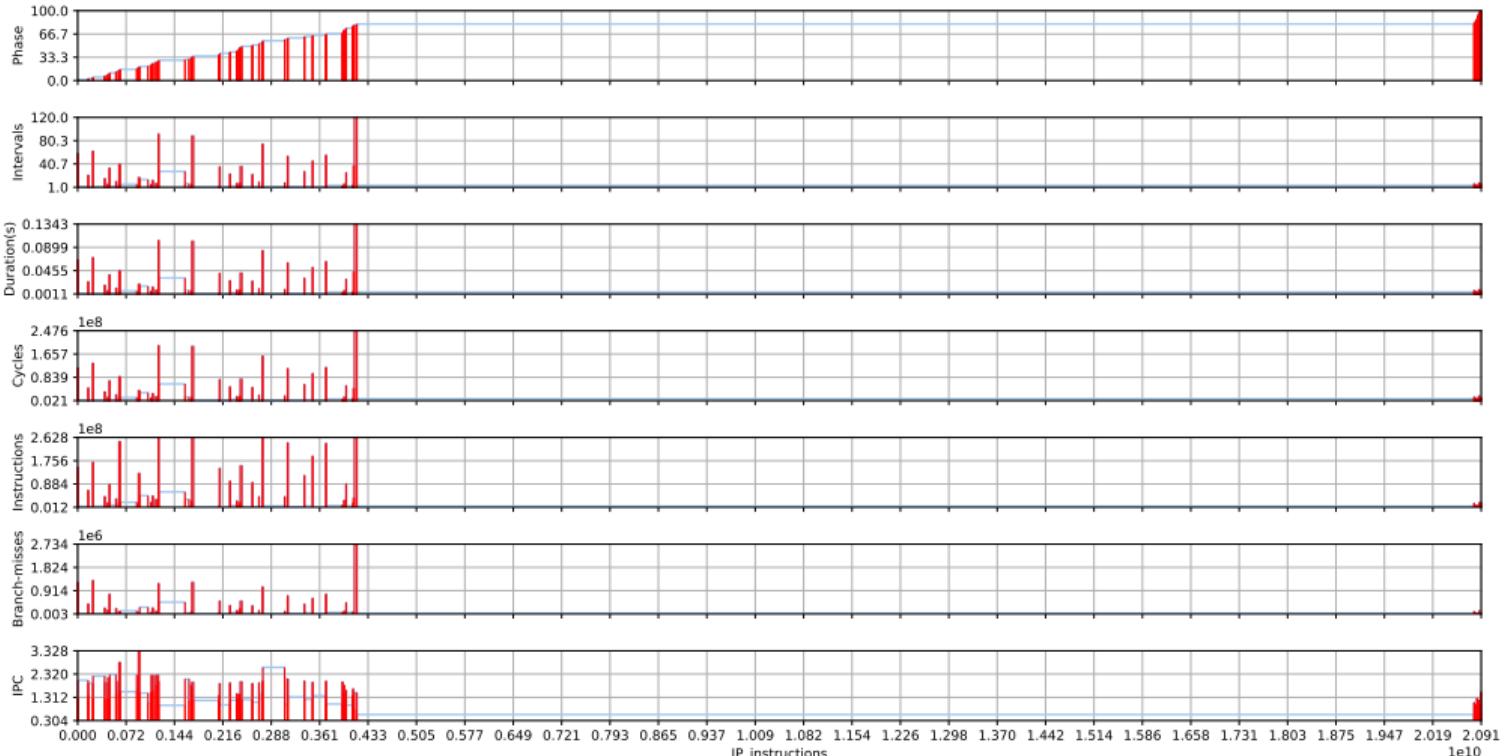
Backups

Phase detection

Graph - Median IPC - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

median_ipc_128_2048_0.6_0_1_data0



Context

PALPATINE

Linux Perf

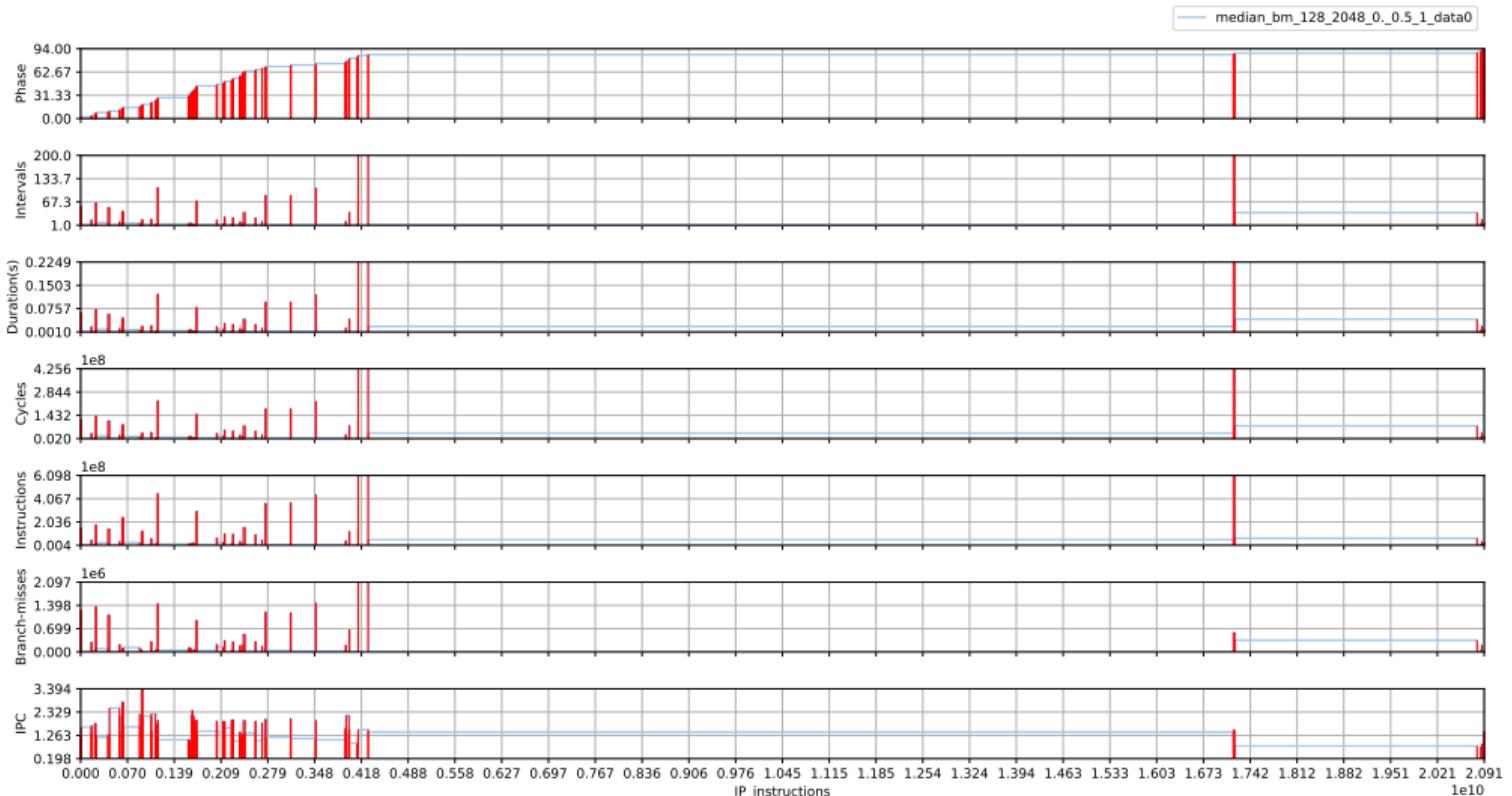
Phase detection

Backups

Phase detection

Graph - Median BM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example



Context

PALPATINE

Linux Perf

Phase detection

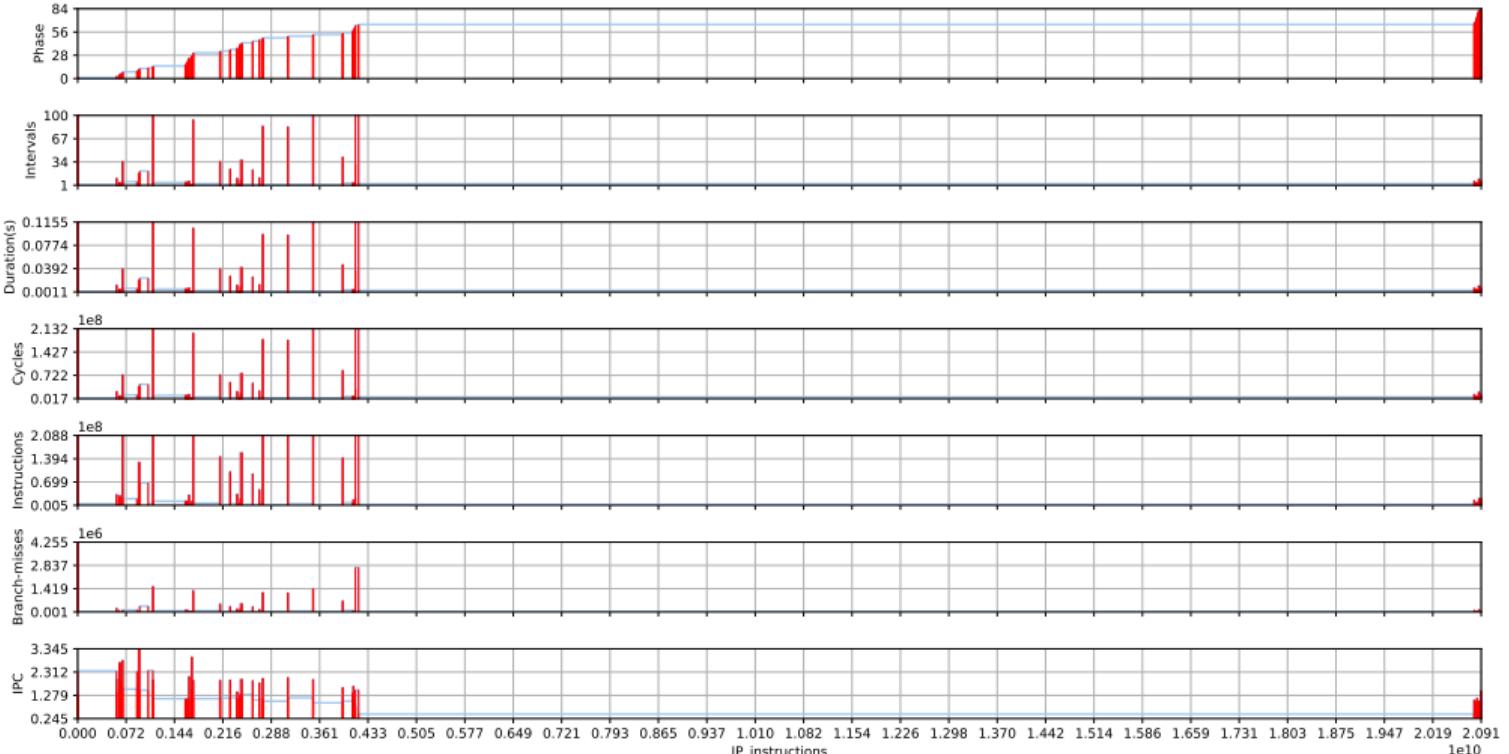
Backups

Phase detection

Graph - Median IPCBM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

median_ipcbm_128_2048_0.6_0.5_1_data0



Context

PALPATINE

Linux Perf

Phase detection

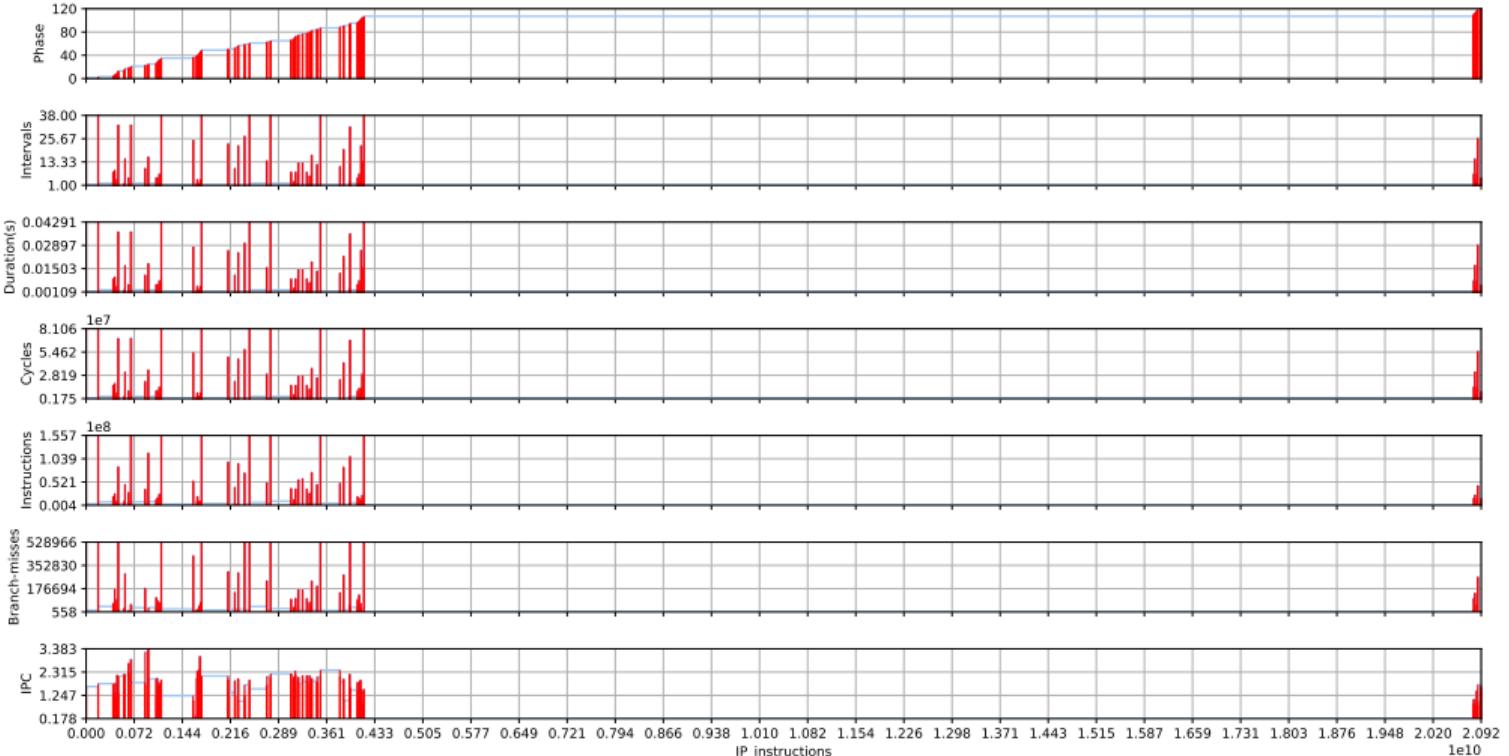
Backups

Phase detection

Graph - Naive IPC - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

naive_ipc_128_2048_0.6_0_1_data0

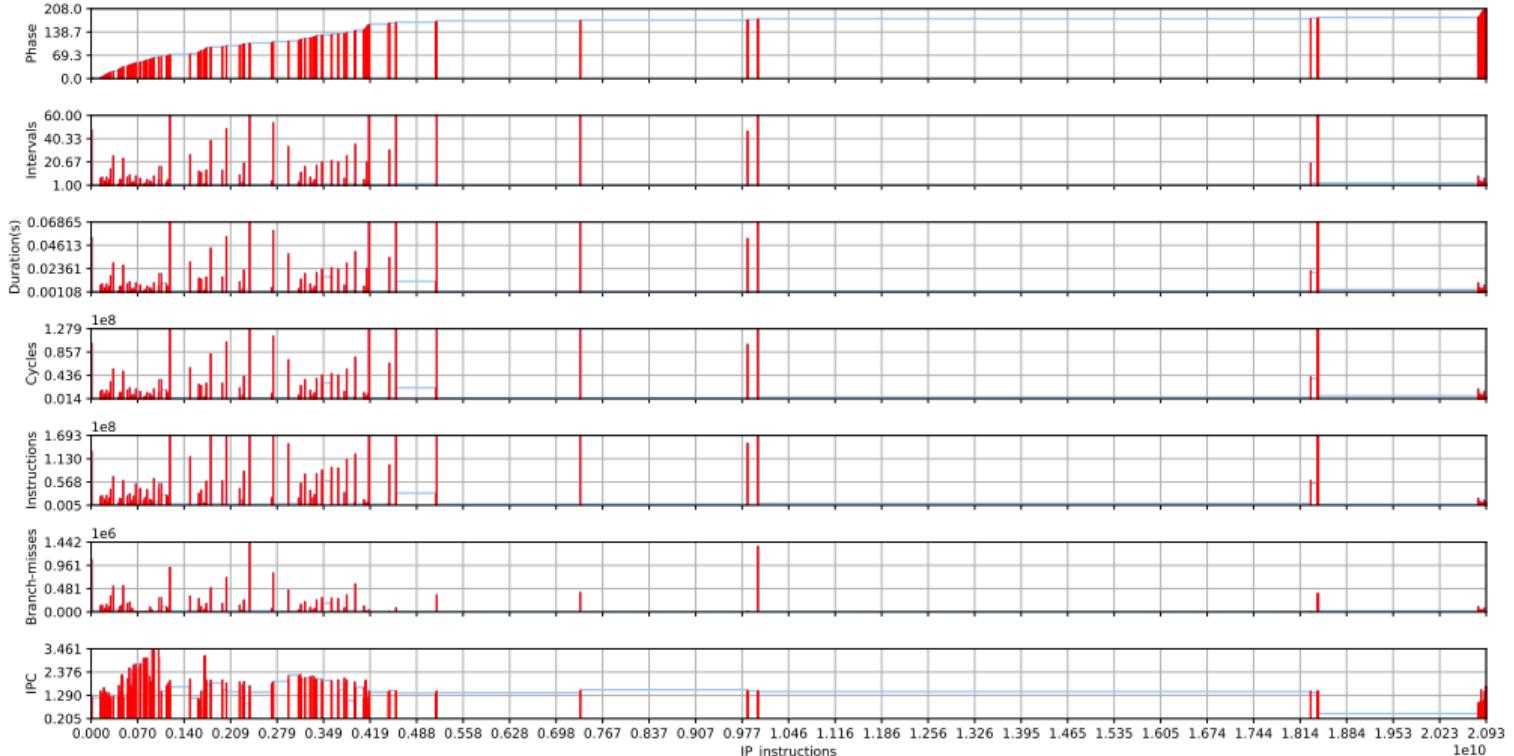


Phase detection

Graph - Naive BM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

naive_bm_128_2048_0_0.5_1_data0



Context

PALPATINE

Linux Perf

Phase detection

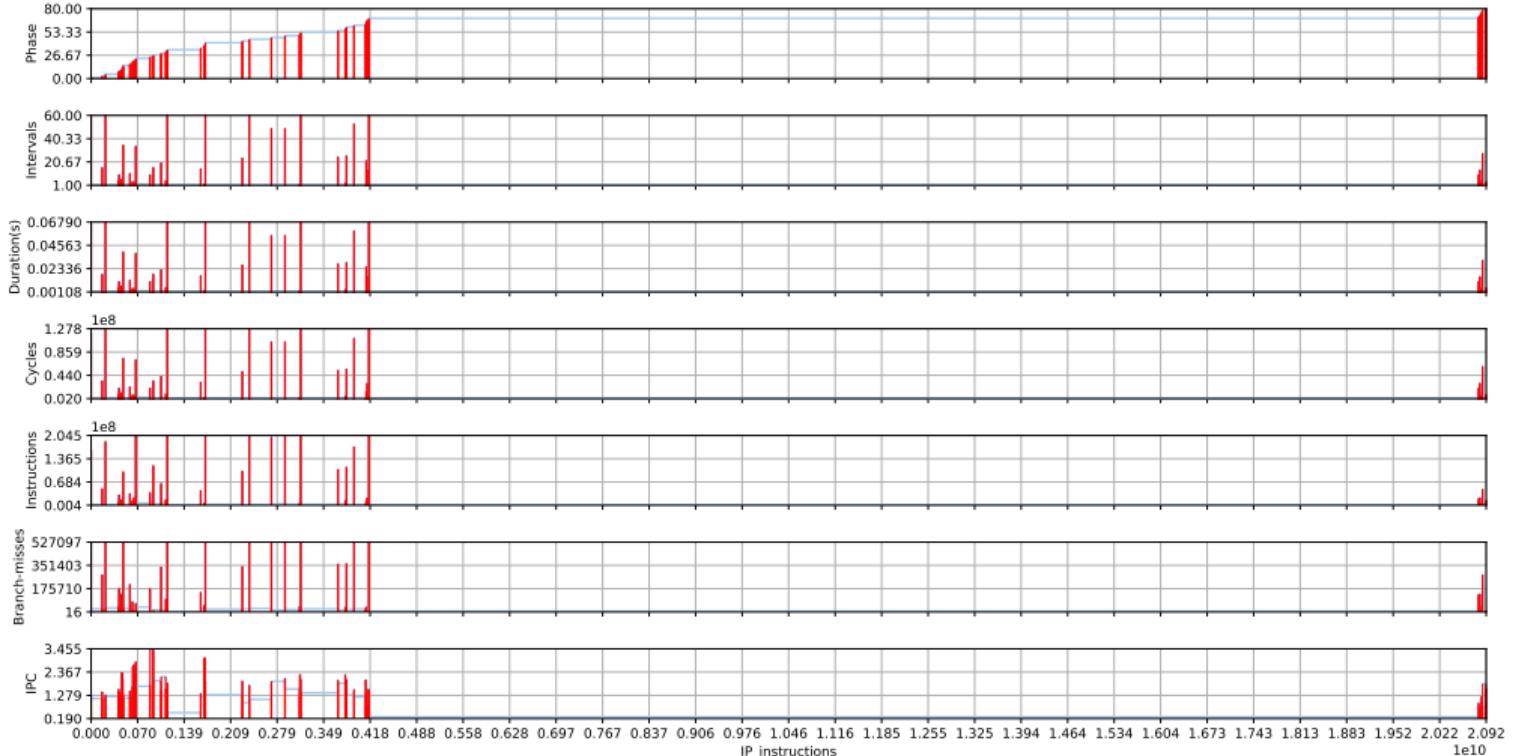
Backups

Phase detection

Graph - Naive IPCBM - 1 sample

Perf stat with detection phase option output for the conjugate gradient example

naive_ipcbm_128_2048_0.6_0.5_1_data0



Context

PALPATINE

Linux Perf

Phase detection

Backups

PALPATINE

PALPATINE : A
profiling analyzer
using Linux Perf

Counter	Hotspot	Function	Total_Usage	Assembly_Adress	Assembly_Events	Assembly_Line
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2e0	1.47e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2e6	2.44e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2ea	9.83e+08	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2ef	1.31e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2f4	6.66e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f2fb	8.91e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f302	8.36e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f308	8.73e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f30d	2.47e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f313	4.31e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f319	3.22e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f31f	4.09e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f323	3.24e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f327	3.11e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f32b	3.40e+09	vmovupd ymm10,YMMWOR[...]
cycles	??:0	mkl_bias_avx2_xdgemv_t	94.39	55f32f	3.05e+09	vmovupd ymm10,YMMWOR[...]

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

PALPATINE

PALPATINE : A
profiling analyzer
using Linux Perf

Counter	Hotspot	Function	Total_Usage	Assembly_Adress	Assembly_Events	Assembly_Line
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f2e6	2.00e+06	add rax,0x8
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f2ef	4.00e+06	add rax,0x8
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f2f4	4.00e+06	add rax,0x8
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f2fb	5.80e+07	add rax,0x8
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f302	1.20e+07	add rax,0x8
fp_arith_inst_retire[...]	??:0	mkl_blas_avx2_xdgemv_t	100.0	55f308	1.00e+07	add rax,0x8

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups

PALPATINE

PALPATINE : A
profiling analyzer
using Linux Perf

Counter	Hotspot	Function	Total_Usage	Assembly_Adress	Assembly_Events	Assembly_Line
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2e0	2.71e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2e6	4.50e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2ea	1.54e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2ef	1.99e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2f4	9.21e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f2fb	1.07e+10	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f302	1.01e+10	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f308	1.08e+10	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f30d	4.42e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f313	8.15e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f319	5.30e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f31f	4.81e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f323	3.90e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f327	3.83e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f32b	4.15e+09	vmovupd ymm10,YMMWOR[...]
fp_arith_inst_retire[...]	??:0	mkl blas_avx2_xdgemv_t	99.53	55f32f	4.15e+09	vmovupd ymm10,YMMWOR[...]

Matthieu
CARRERE

Context

PALPATINE

Linux Perf

Phase detection

Backups