Météo-France RAPS Benchmark

Ryad El Khatib, Philippe Marguinaud,
Louis-François Meunier*, Eric Sevault
(CNRM/GMAP/ALGO and DT/DSI/SC*)

- Contents of the benchmark
- Optimizations performed for RAPS
- First results on scalability
- Full MF benchmark contents

AROME

- Fine mesh regional model
- Belongs to the ARPEGE/IFS, ALADIN/ALARO family
- Non-hydrostatic dynamics
- Physics taken from MesoNH
- Operational resolution = 750x720 L60 over France = 2.5 km

Contents of MF RAPS benchmark

- Source code based on cy36t1 (quite recent)
- Gmkpack (compilation system) & auxiliary tools
- AROME forecast at various resolutions :
  - 64 x 64 L41 (runs on a PC),
  - 240 x 240 L41, 600x512L60
  - 750 x 720 L60 (current operational)
  - 1440 x 1350 L87 (4h30 with 4 NECSX9 nodes)

- The aim is :
  - Acquaint constructors with our code and our environment
  - Get some feedback on our code (portability, performance, possible optimisations)

- RAPSMF1006 available since 07/2010, updated 10/2010 ; BULL, IBM, SGI, CRAY, HP, NEC, FUJITSU, NVIDIA

Developments for RAPS

- Work on OpenMP & SURFEX
- Optimisation of OpenMP on the NEC
- Porting on IBM

SURFEX (sigh...)

Surface scheme used in AROME

- SURFEX V6 : not thread-safe, no OpenMP support

- Strategy to enable OpenMP defined with SURFEX development team: Global variables → « THREADPRIVATE » (about 2000)

- SURFEX V7 : not thread-safe, works with OpenMP

- BUT :
  - No support for other parallelization schemes
  - Hard to maintain
  - Probably incompatible with OOPS

SURFEX/MSE

Clean-up of the set-up of SURFEX:

Multiple reads of initial condition SURFEX file by **all** NPROMA blocks of **all** MPI tasks

↓

SURFEX fields are read once by MPI #1 and stored in a cache

+

Minimize the number of reads of namelists

Still to do : enable OpenMP in SURFEX set-up

Météo-France NEC SX9

- 2 clusters x 10 nodes x 16 procs
- 1TB of memory / node
- Vector processors
- Peak performance ≈ 100 Gflops / proc

OpenMP on the NEC

AROME/GARD(=1/4 FRANCE) – 16 procs – no RTTOV

16 MPI tasks x 1 thread

```
Real    Time (sec)           :     607.349

Memory size used (MB)        :     3264.000 [0,15]     7040.000 [0,0]      6708.000 x 16 = 107328 Mb

Instruction Cache miss (sec):        5.522 [0,15]       16.923 [0,0]        13.890

Operand     Cache miss (sec):       14.530 [0,15]       37.061 [0,12]       33.045
```

8 MPI tasks x 2 threads

```
Real    Time (sec)           :     676.188 (+69)

Memory size used (MB)        :     9472.000 [0,7]    10752.000 [0,0]    10512.000 x 8 = 84096 Mb

Instruction Cache miss (sec):       19.801 [0,7]       31.553 [0,0]       27.544 (+14)

Operand     Cache miss (sec):       75.109 [0,7]      129.730 [0,0]      113.708 (+70)
```

OpenMP on the NEC

AROME/GARD – 16 procs – no RTTOV

Dynamic allocations on the NEC (ALLOCATE) with OpenMP
→ “operand cache miss”

Before and after reducing dynamic allocations :

16MPIx1T

```
Real   Time (sec)           :      607.349
Real   Time (sec)           :      585.597
```

8MPIx2T

```
Real   Time (sec)           :      676.188
Real   Time (sec)           :      582.472
```

ARPEGE and ALADIN do not have this problem (automatic variables).

Profiling on the IBM

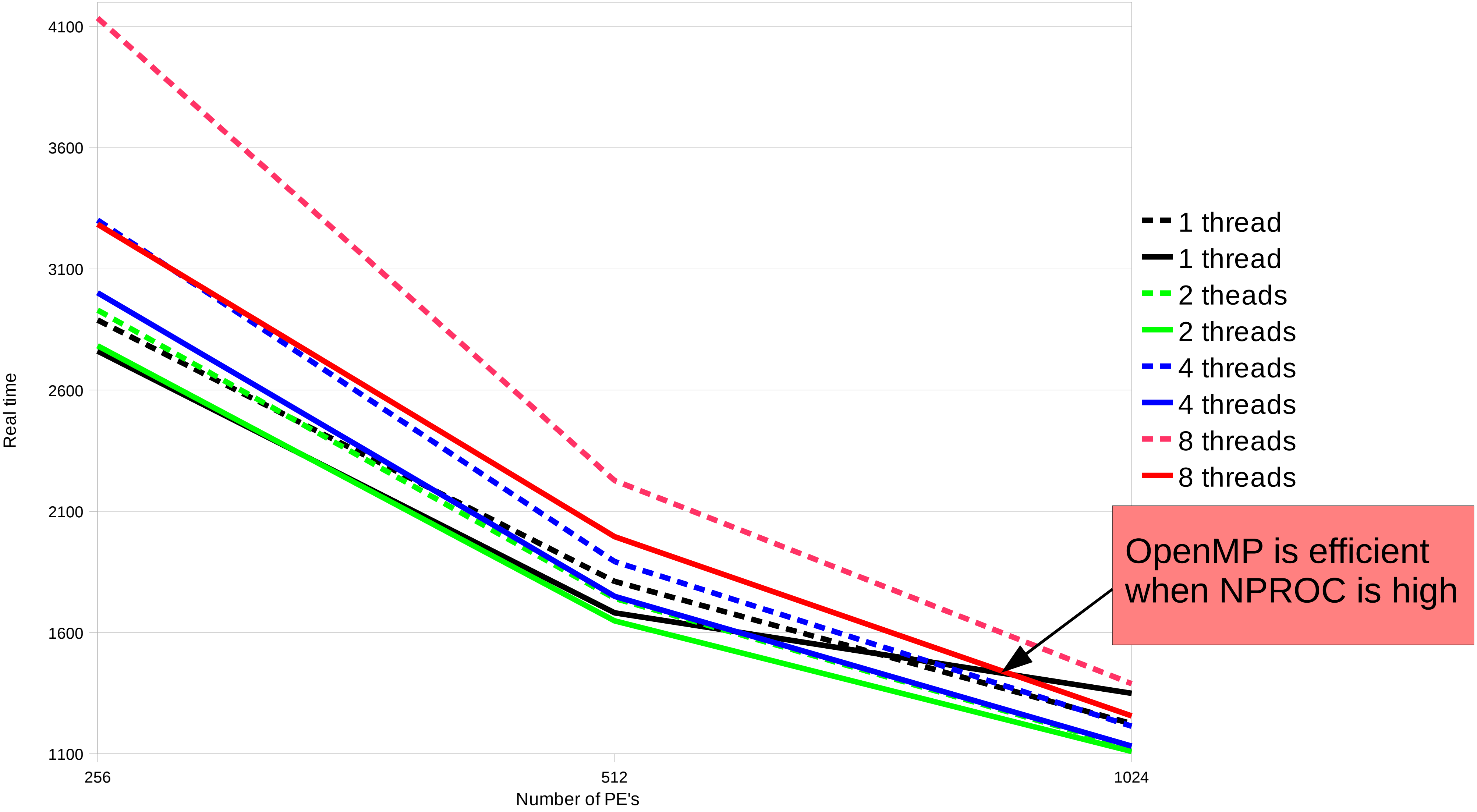
Machine : IBM cluster « C1A » (ECMWF)

- « Power 6 » processors
- 32 procs per node
- Tests with NPROC = 32, 64, 128, 256, 512, 1024

Namelists :

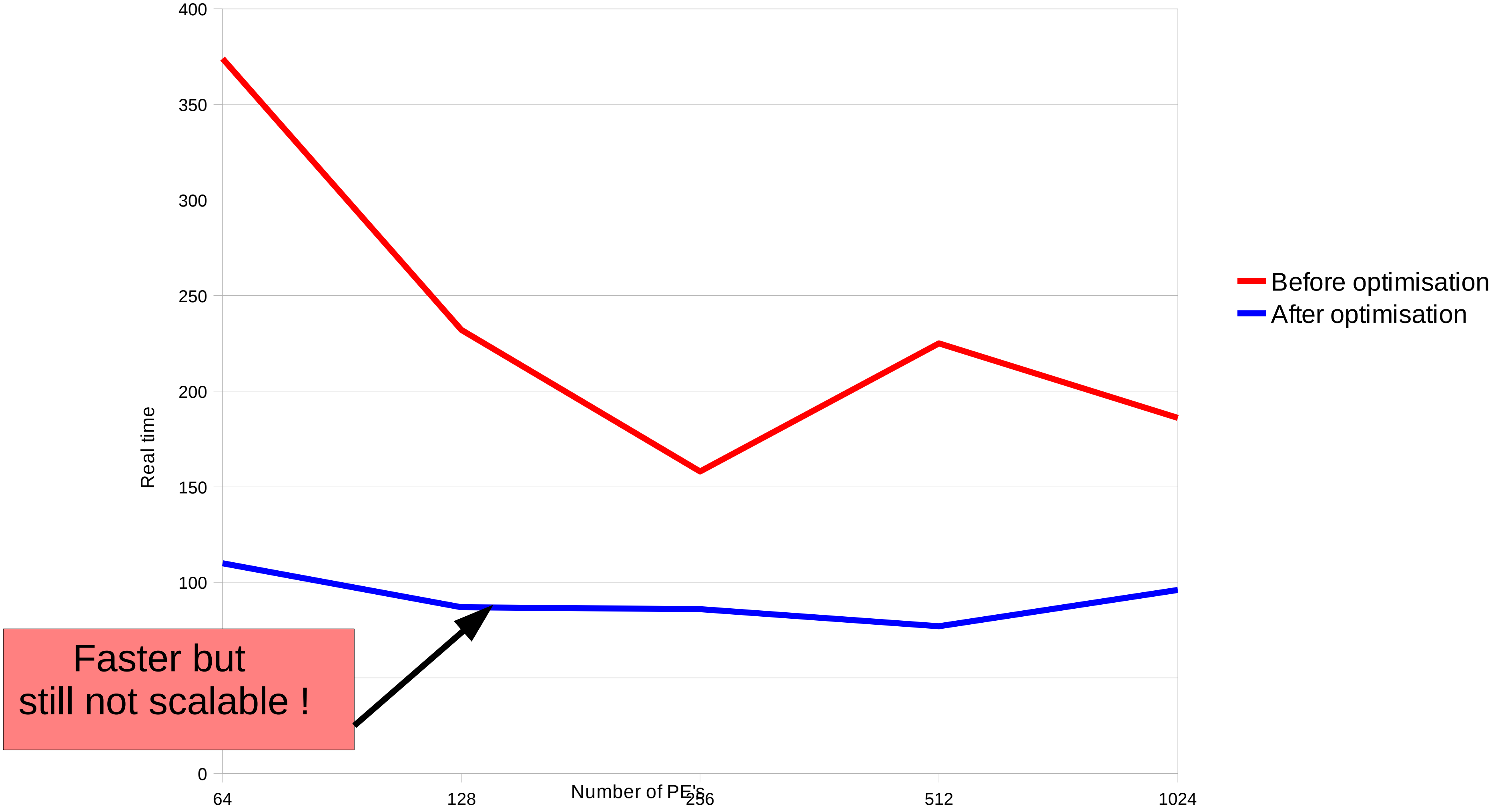
- No output
- No post-processing
- Forecast term = 30h

OpenMP on IBM
AROME 30h forecast



OpenMP is efficient when NPROC is high

OpenMP on IBM AROME set-up



Faster but
still not scalable !

OpenMP – Conclusion

- Reduce memory usage
- Reduce load imbalance
- Performances
  - IBM : good when the number of procs is high
  - NEC : no gain, even after optimization
  - PC : good improvement (20%)

Depends on the hardware + OS

and on the code

Scalability

(fixed initial conditions)

- High level structure

  (spectral transforms, semi-lagrangian)

  (from model dynamics)

- Amount of computation time per time-step

  (from dynamics & physics)

- IO sub-system design + amount of IO

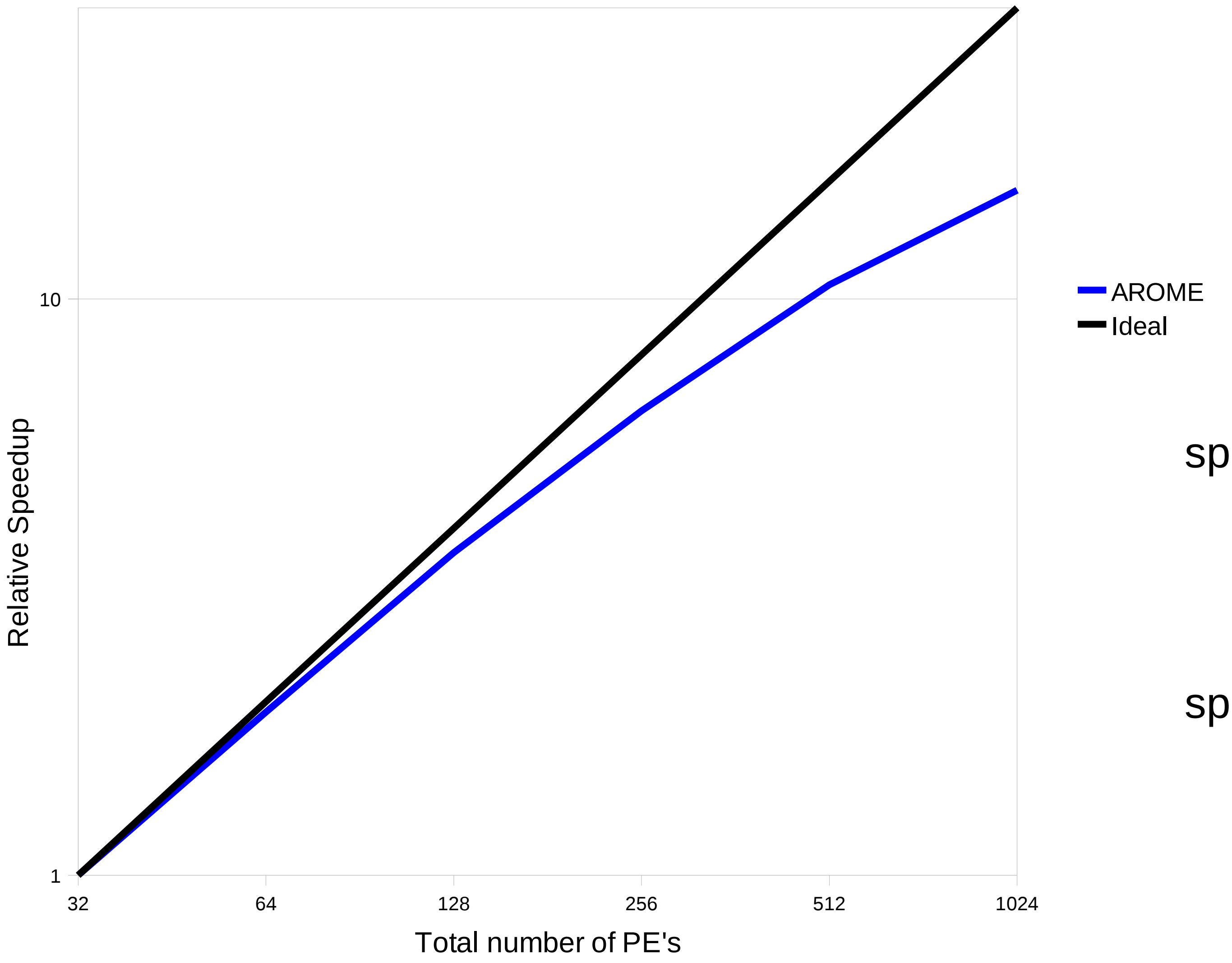
  (from the coupling frequency, the value of the time-step)

- Load imbalance

  (from the physics)

AROME/LACE

cycle 36T1 "V7gmap2" - IBM Power 6

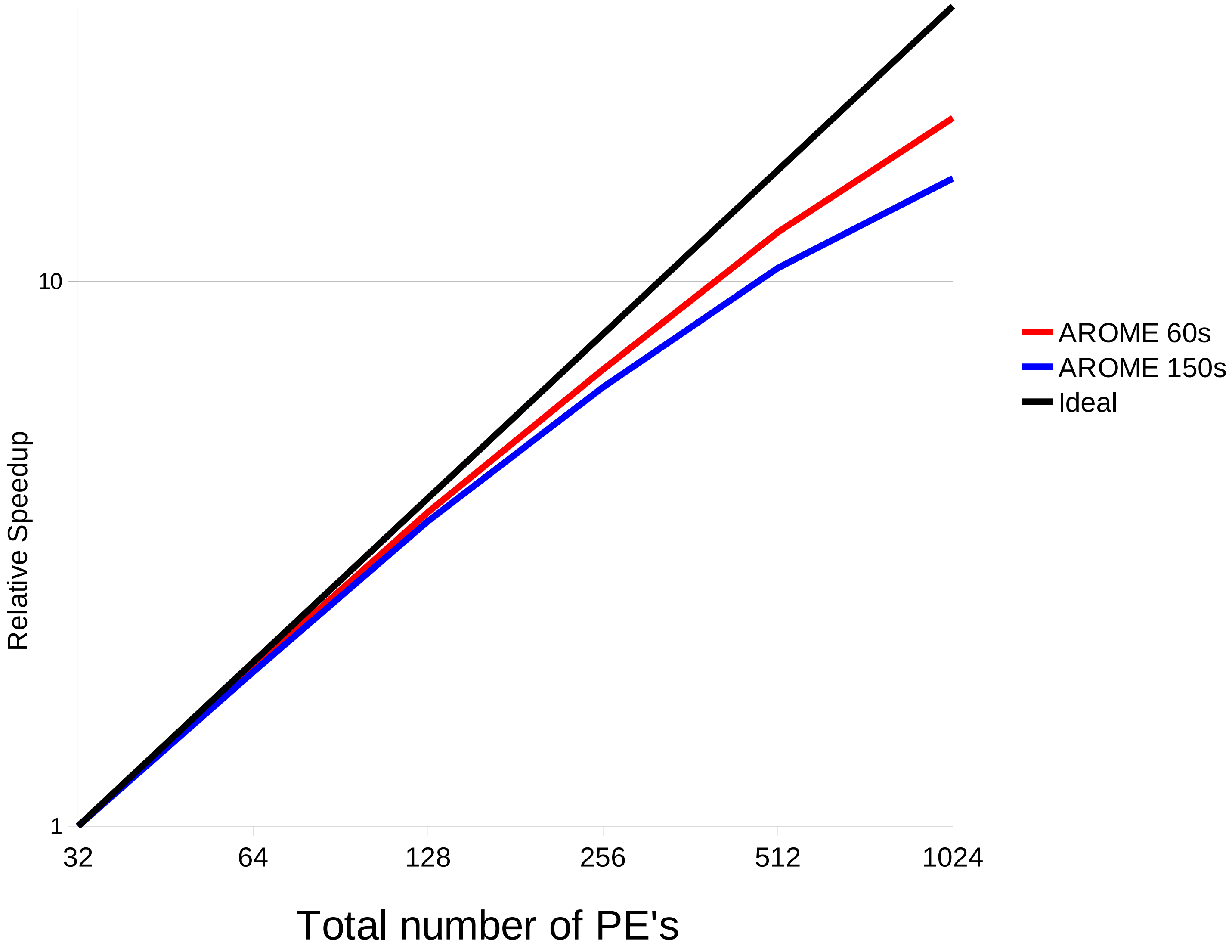


512 PE
speed-up = 10.6 vs 16
7.5 min

1024 PE
speed-up = 15.2 vs 32
5 mn

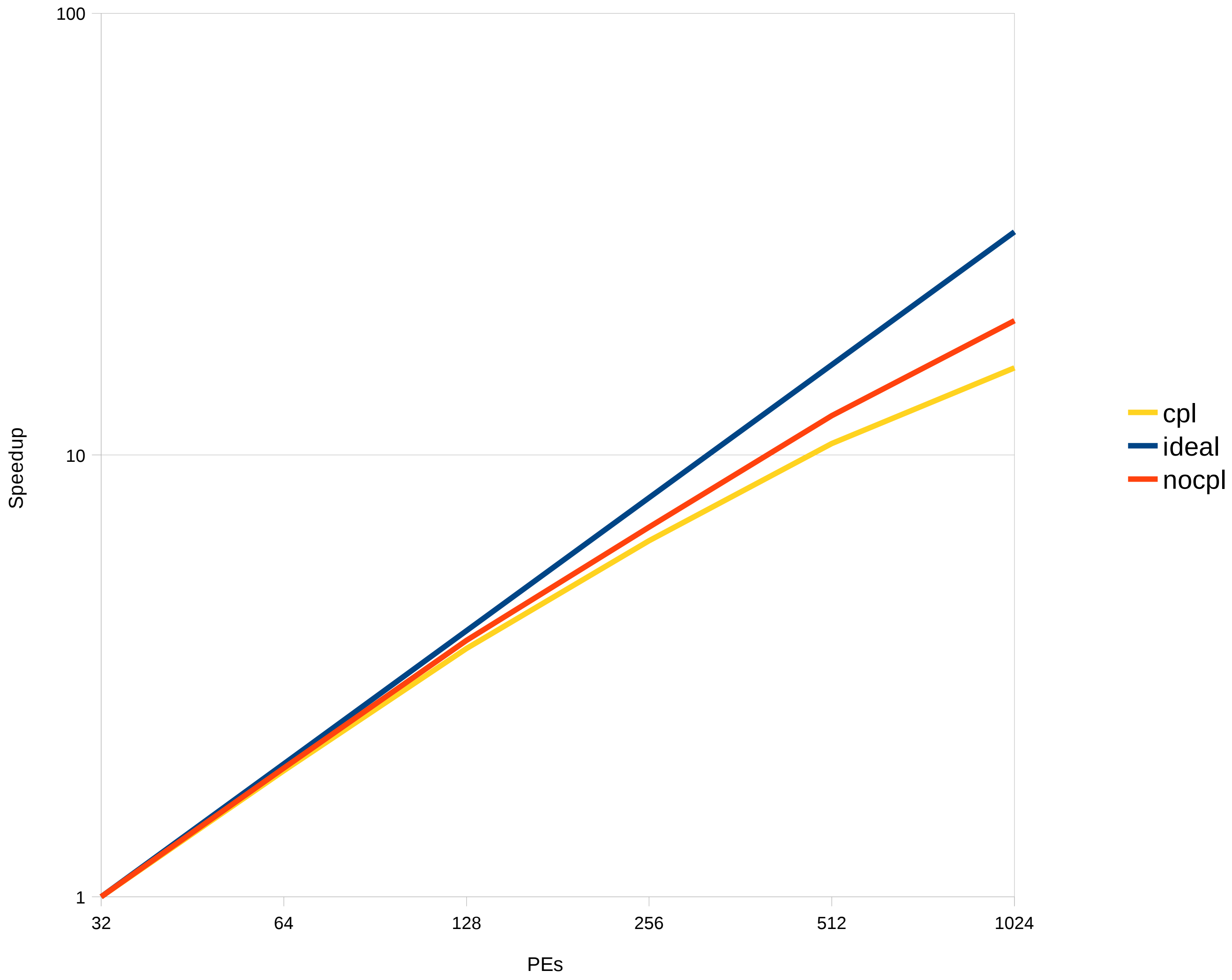
AROME/LACE

With different time-steps



Effect of reading coupling data

AROME/LACE, with and without coupling



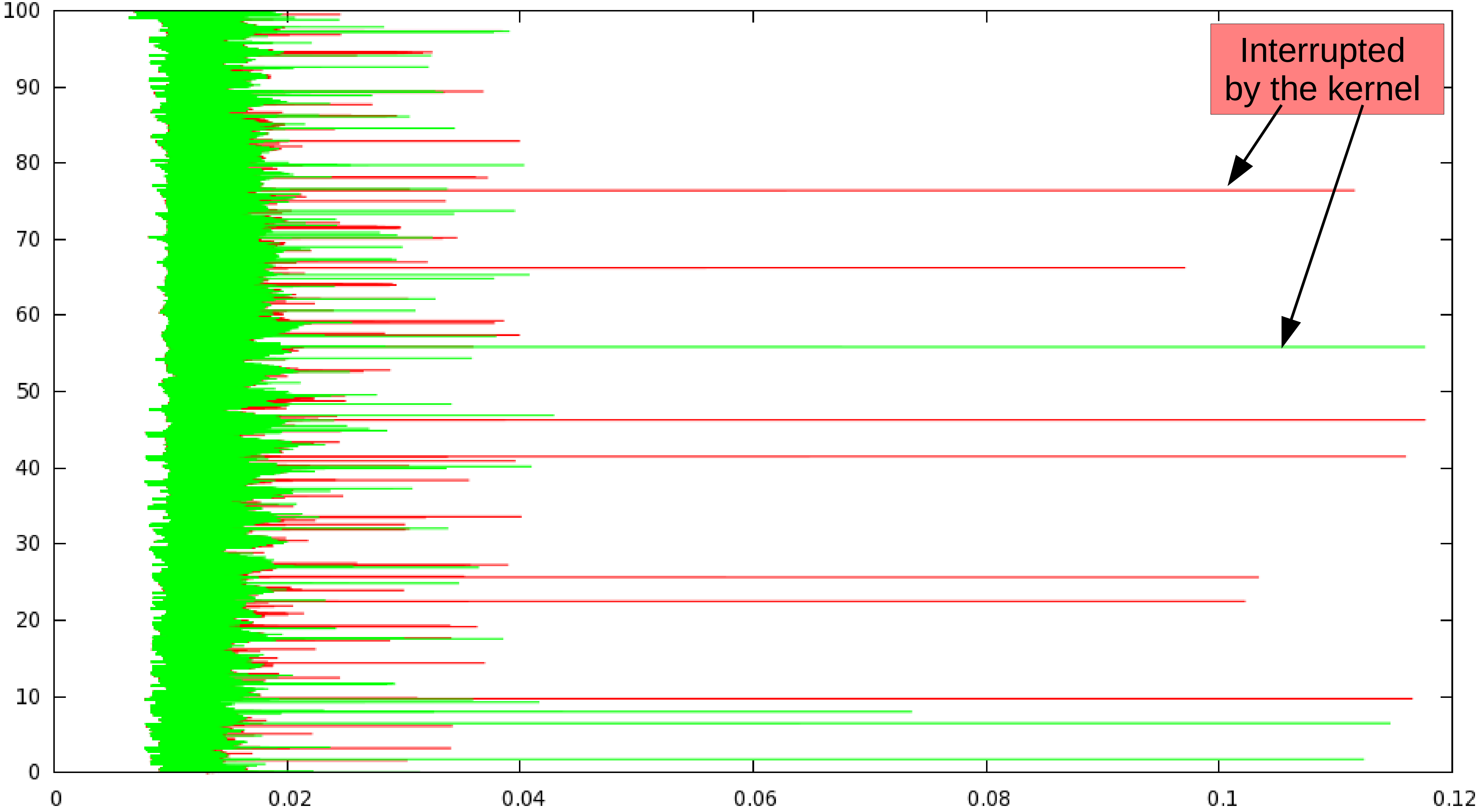
Scalability of the physics

Physics of the model :

- Independent columns of atmosphere
- Columns processed in batches of NPROMA (50 on the IBM, 3582 on the NEC)

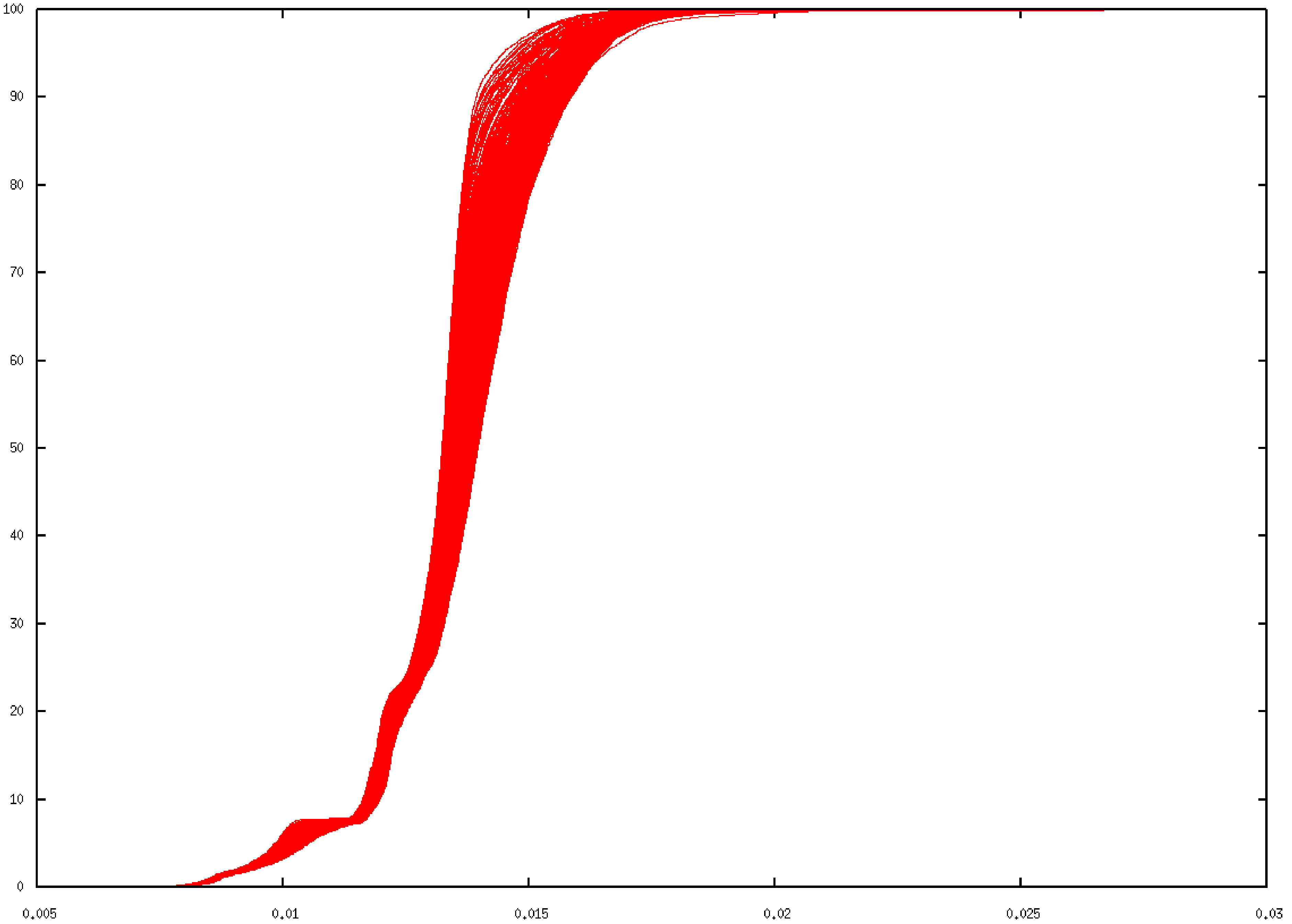
→ optimize use of cache or vector registers

Time (s) by NPROMA block

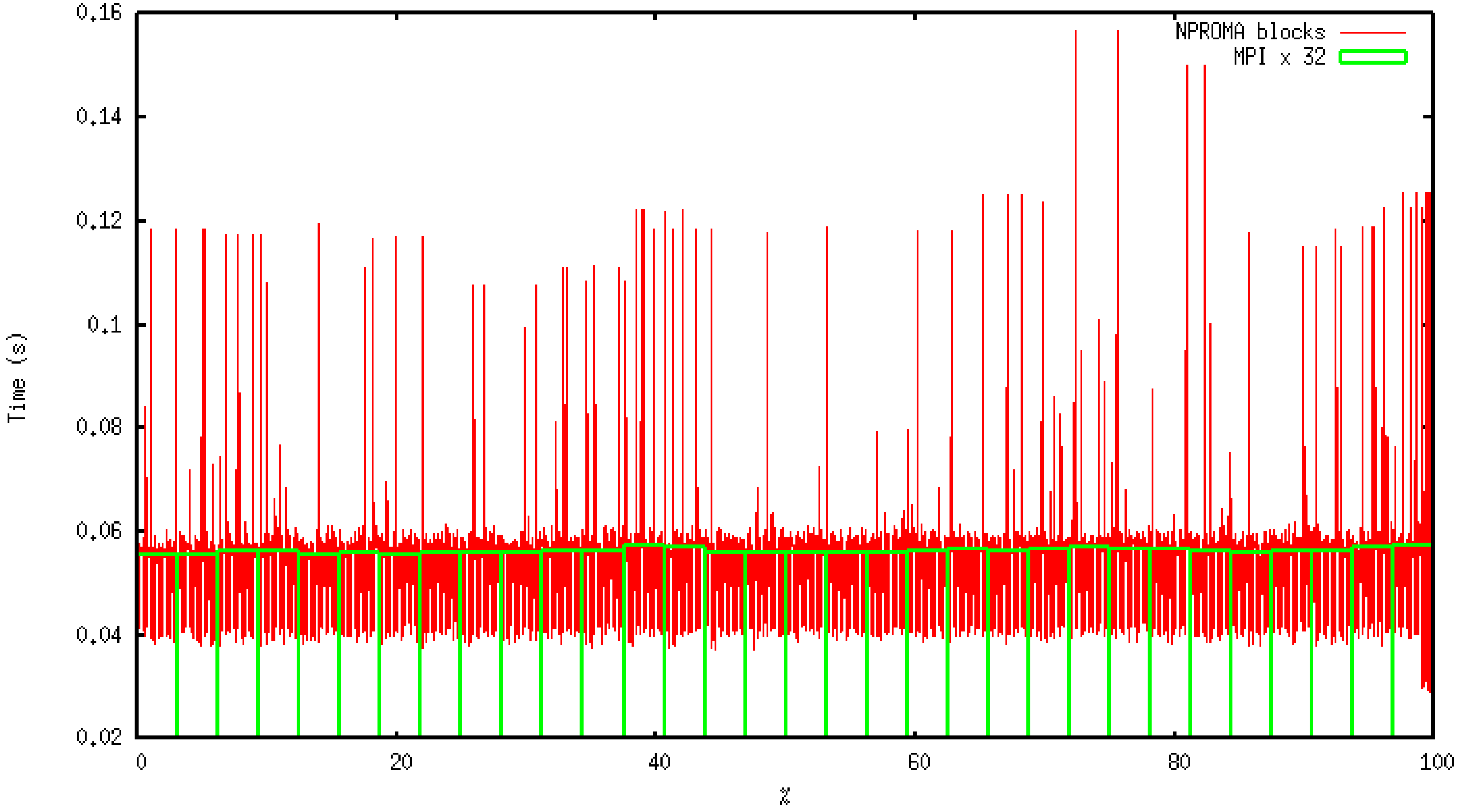


Interrupted by the kernel

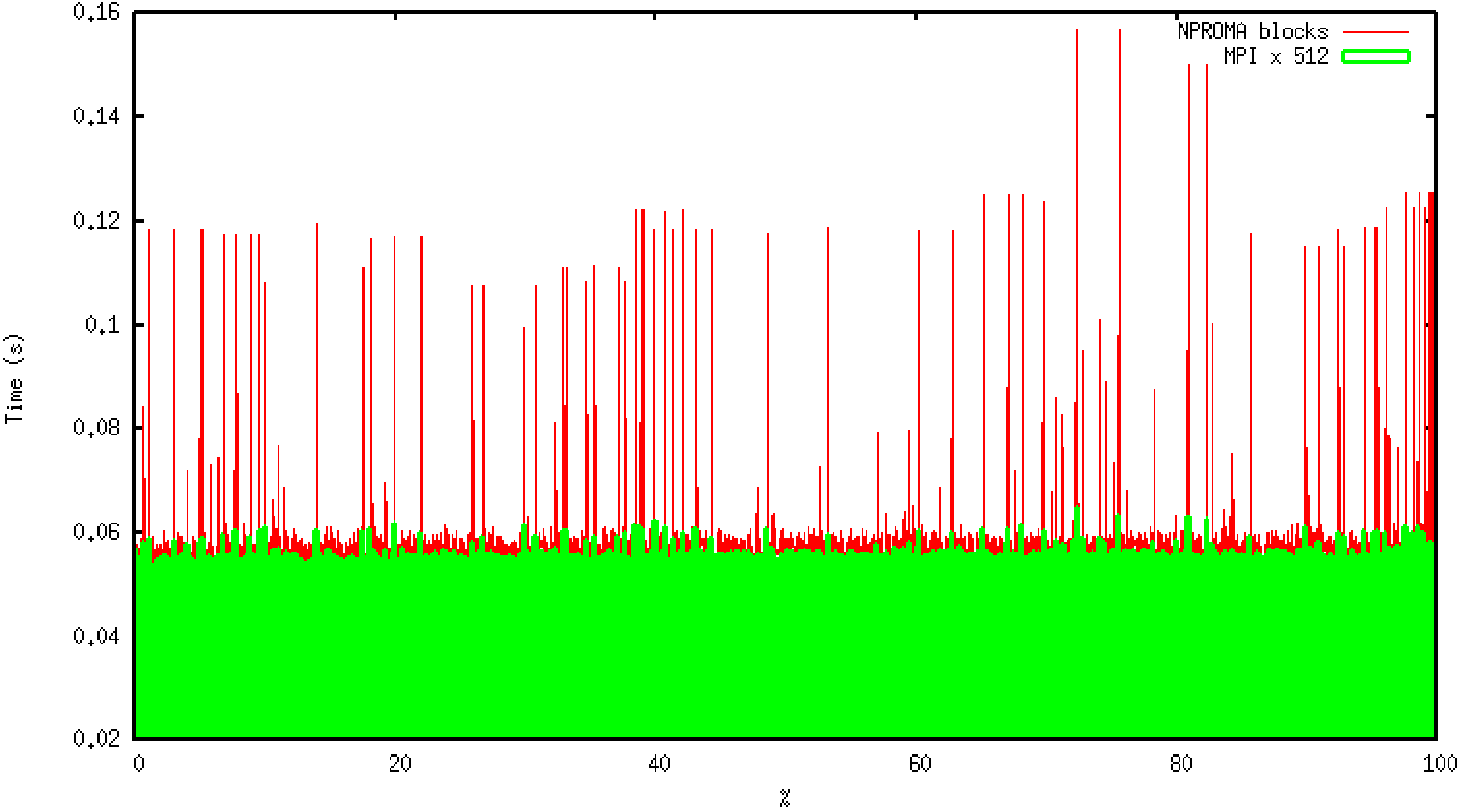
Sorted time of NPROMA blocks



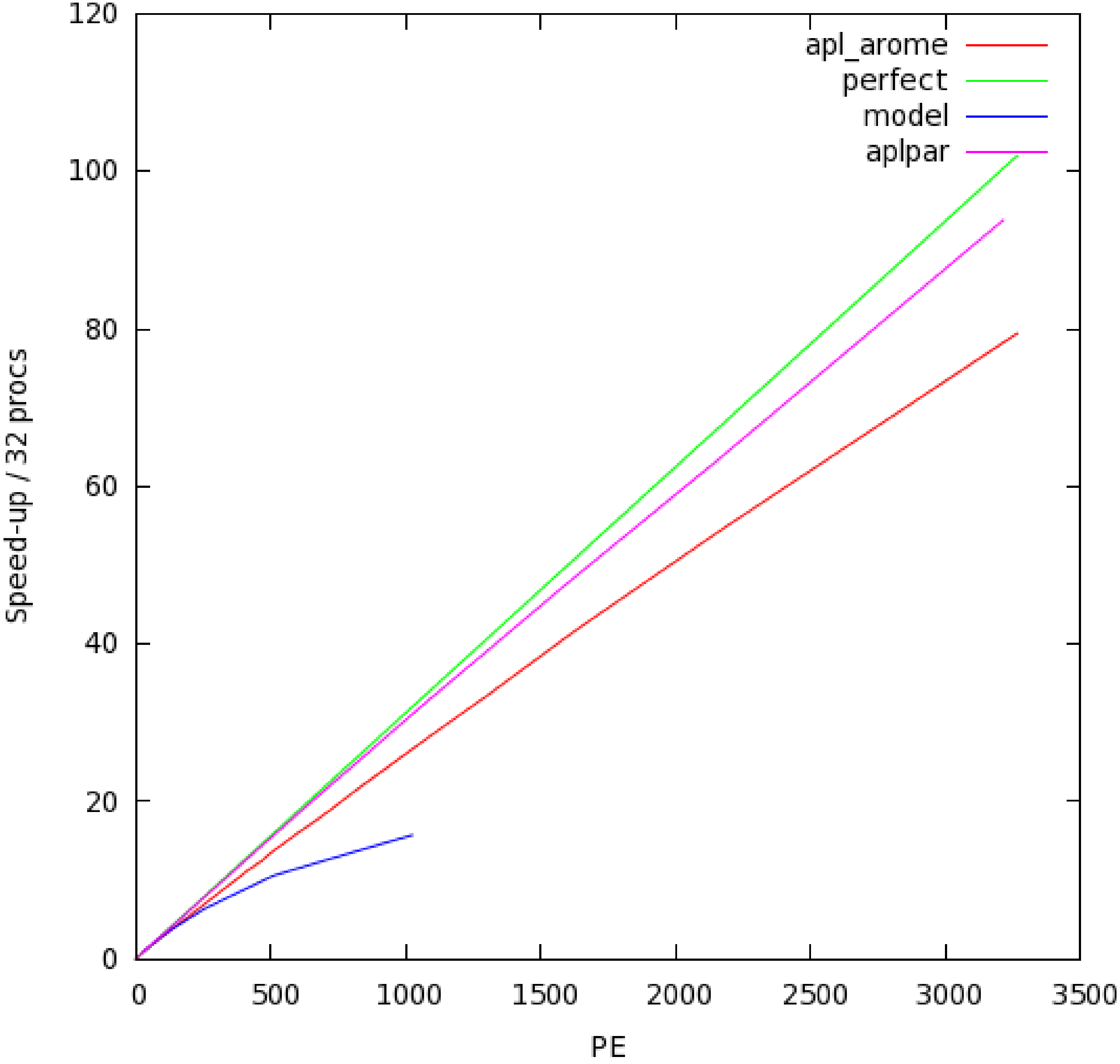
NPROMA blocks & MPI tasks



NPROMA blocks & MPI tasks



Inferred physics scalability



Our IO sub-system

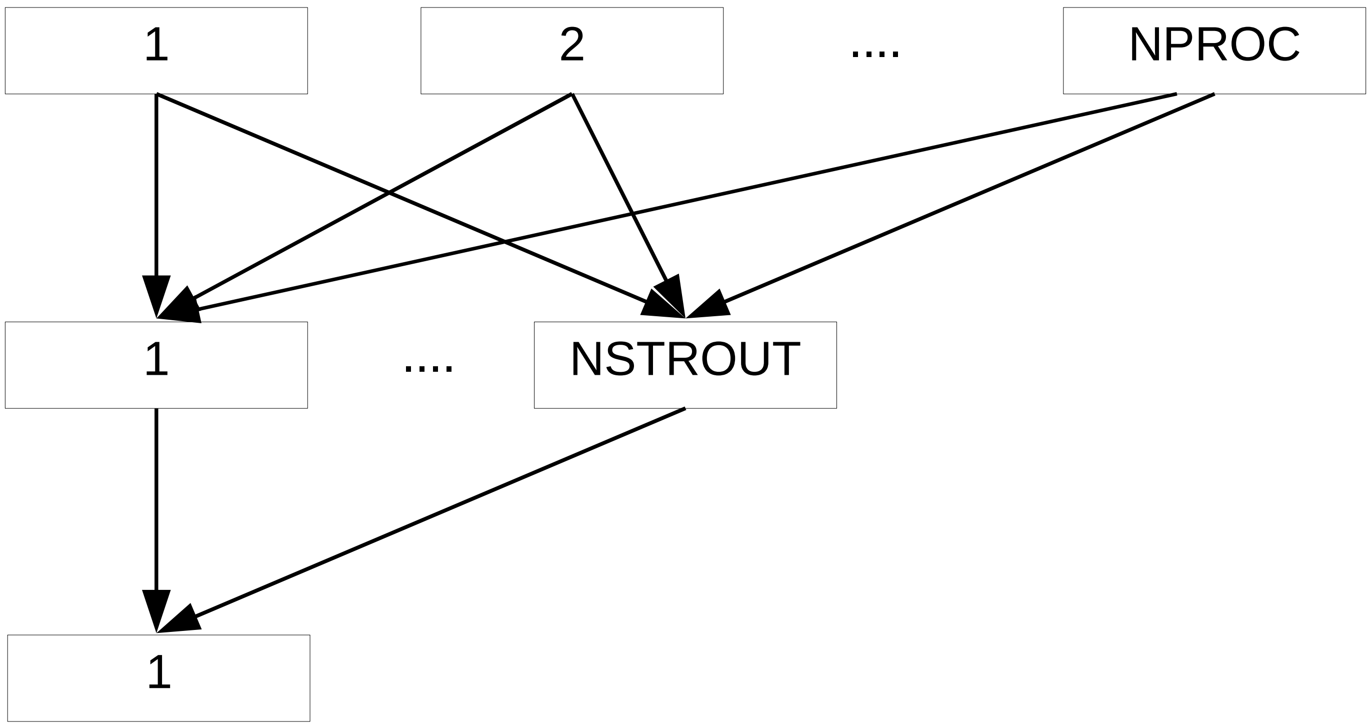
Step 1

Step 2 All to 1..NSTROUT
Gather whole fields

Step 3 Compress fields

Step 4 Gather on MPI #1

Step 5 Write to disk

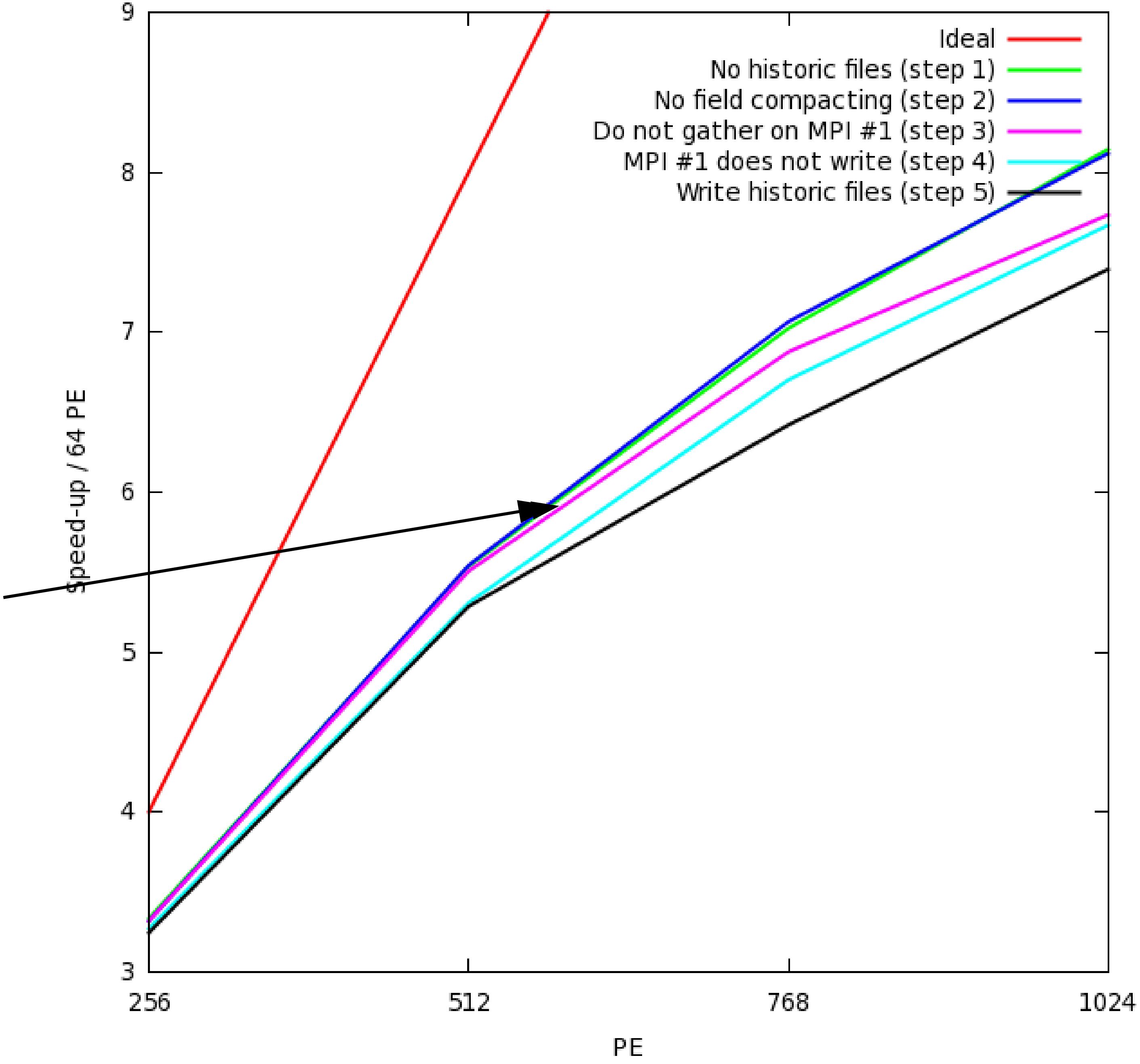


1 2 .... NPROC

1 .... NSTROUT

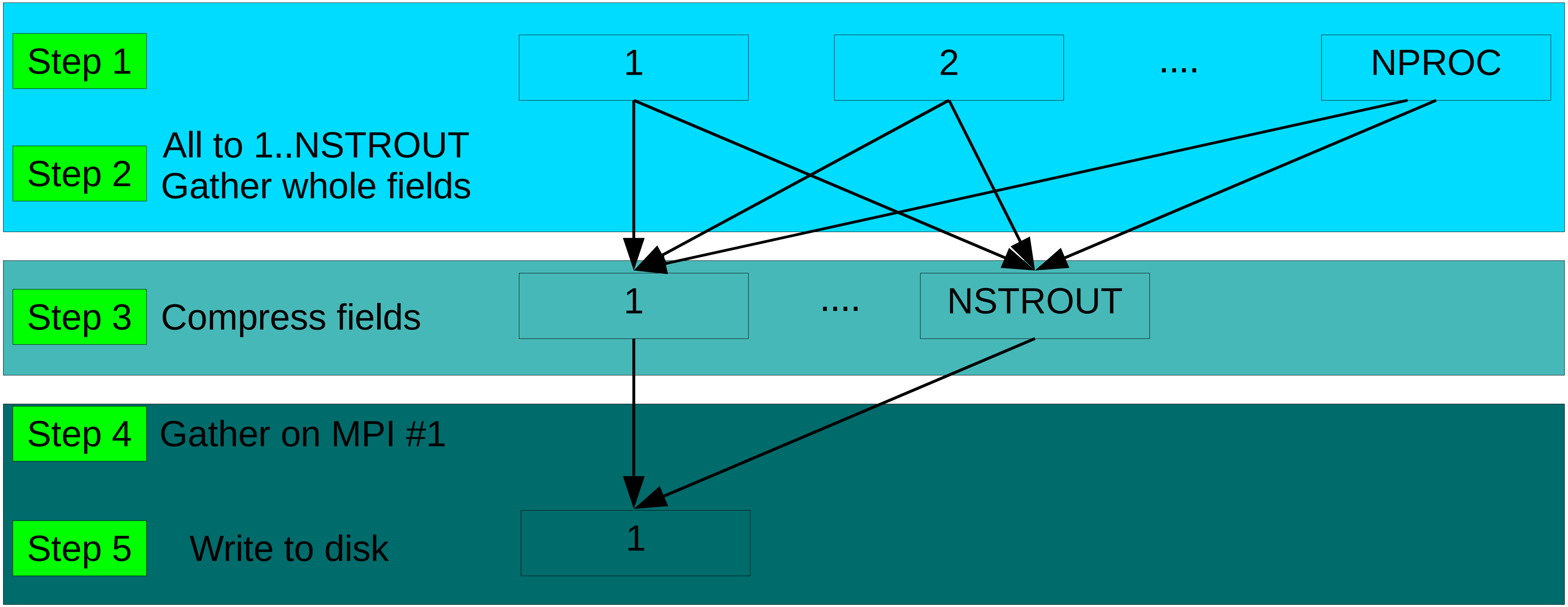
1

Scalability of IO



≈ 600 fields
Compacting
no more scalable

Our IO sub-system



Step 1

Step 2 All to 1..NSTROUT
Gather whole fields

Step 3 Compress fields

Step 4 Gather on MPI #1

Step 5 Write to disk

Météo-France NWP final benchmark

March 2011

Model	Forecast	Assimilation
ARPEGE	• T1198c2.2L105 • + post-processing • 10mn/24h	• 4DVAR T1198, T479, T107 • Observations = 5 x 2010 • 40mn
AROME	• 1440X1440 L87 • + post-processing • 30mn/24h	• 3DVAR • Observations = 6 x 2010 • 7mn (nowcasting) • 15mn (forecasting)