Science & Technology Facilities Council

Exploiting Extreme Processor Counts on the Cray XT4 with High-Resolution Seismic Wave Propagation Experiments

Mike Ashworth[1], Mario Chavez[2,3] and Eduardo Cabrera[4]

1 STFC Daresbury Laboratory, Warrington WA4 4AD, UK

2 Institute of Engineering, UNAM, C.U., 04510, Mexico DF, Mexico

3 Laboratoire de Géologie CNRS-ENS, 24 Rue Lhomond, Paris, France

4 DGSCA, UNAM, C.U., 04510, Mexico DF, Mexico

INSTITUTO DE INGENIERÍA UNAM | COMPUTO ACADEMICO UNAM | ENS | SCAT Scientific Computing Advanced Training

Science & Technology Facilities Council

Outline

- Introduction to seismic wave code
- Benchmark cases
- Optimization
- Performance profiling
- Benchmark results

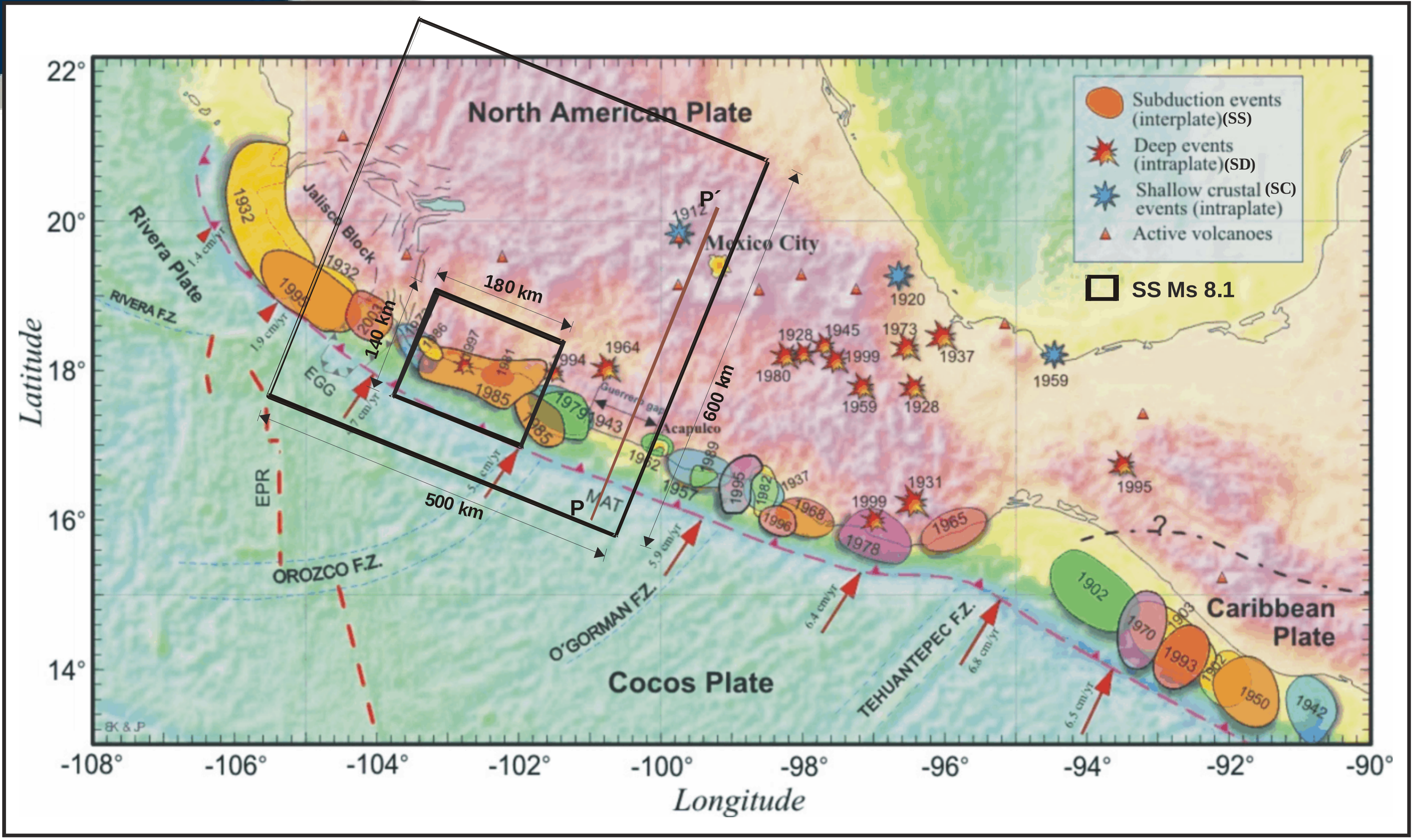
Science & Technology Facilities Council

Large subduction earthquakes

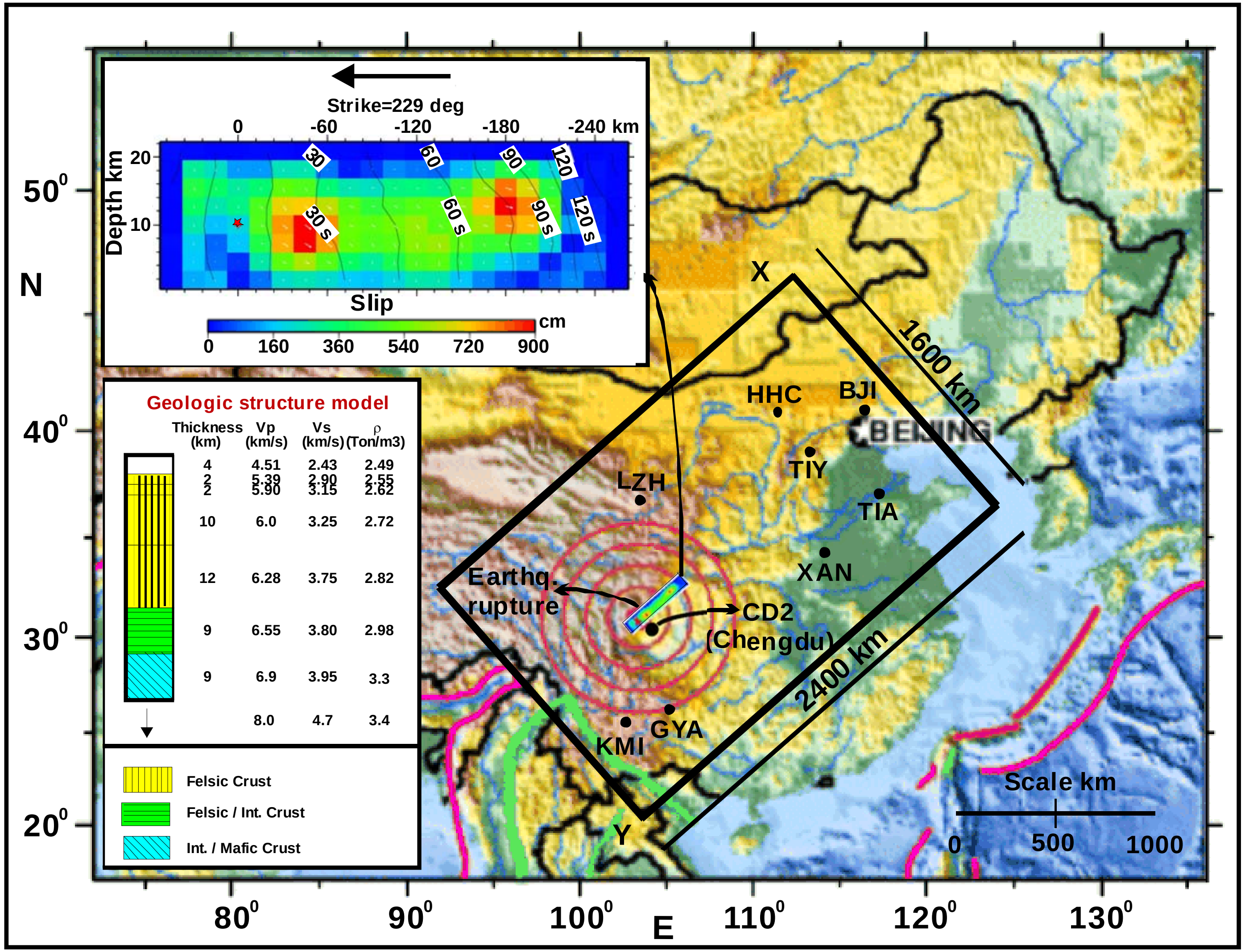
On 19th Sep 1985 a large Ms 8.1 subduction earthquake occurred on the Mexican Pacific coast with an epicentre at about 340 km from Mexico City. The losses were of about 30,000 deaths and 7 billion US dollars.

On 12th May 2008 the Ms 7.9 Sichuan, China, earthquake produced about 70,000 deaths and 80 US billion dollars loss.

Therefore, there is a seismological, engineering and socio economical interest to model these types of events, particularly, due to the scarcity of observational instrumental data for them.



Inner rectangle is the rupture area of the 19/09/1985 Ms 8.1 earthquake on the surface projection of the 500x600x124 km earth crust volume 3DFD discretization



Locations of: a) the epicenter (red dot) of the 12 05 2008 Sichuan Ms 7.9; b) its rupture area and its kinematic slip; c) 9 seismographic stations sites (black dots) of the China Seismographic Network; d) the surficial projection of the 2400 x 1600 x 300 km3 volume used to discretize the region of interest; f) the geologic structure adopted for the volume

Science & Technology Facilities Council

Sichuan earthquake 12th May 2008

Science & Technology Facilities Council

Seismic wave modelling

Realistic 3D modelling of the seismic wave propagation for these types of earthquakes, should include volumes of the earth crust of hundreds of kilometers

3D finite difference modeling of realistic-earth size seismic wave propagation problems has been successful, but very computationally demanding

fd3d earthquake simulation code

Seismic wave propagation

3D velocity-stress equations

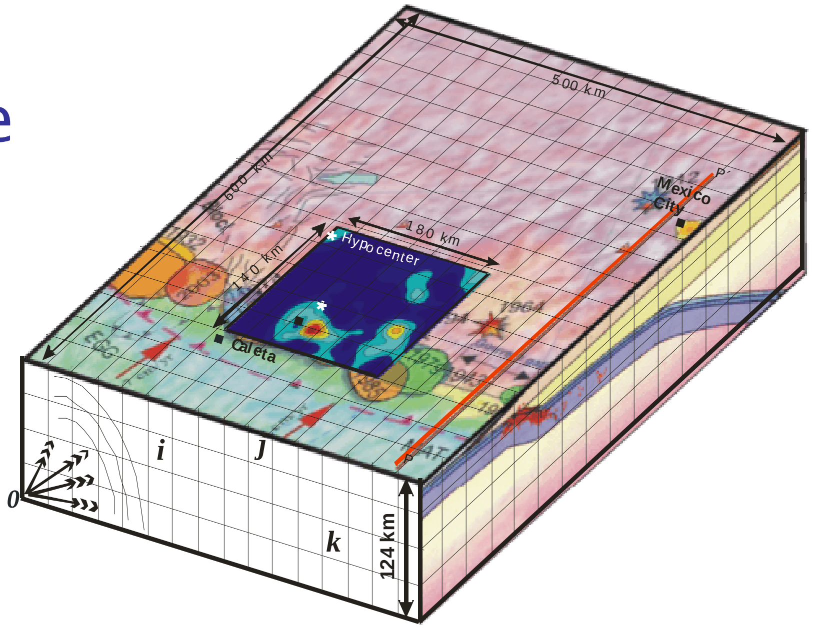
Structured grid

Explicit scheme

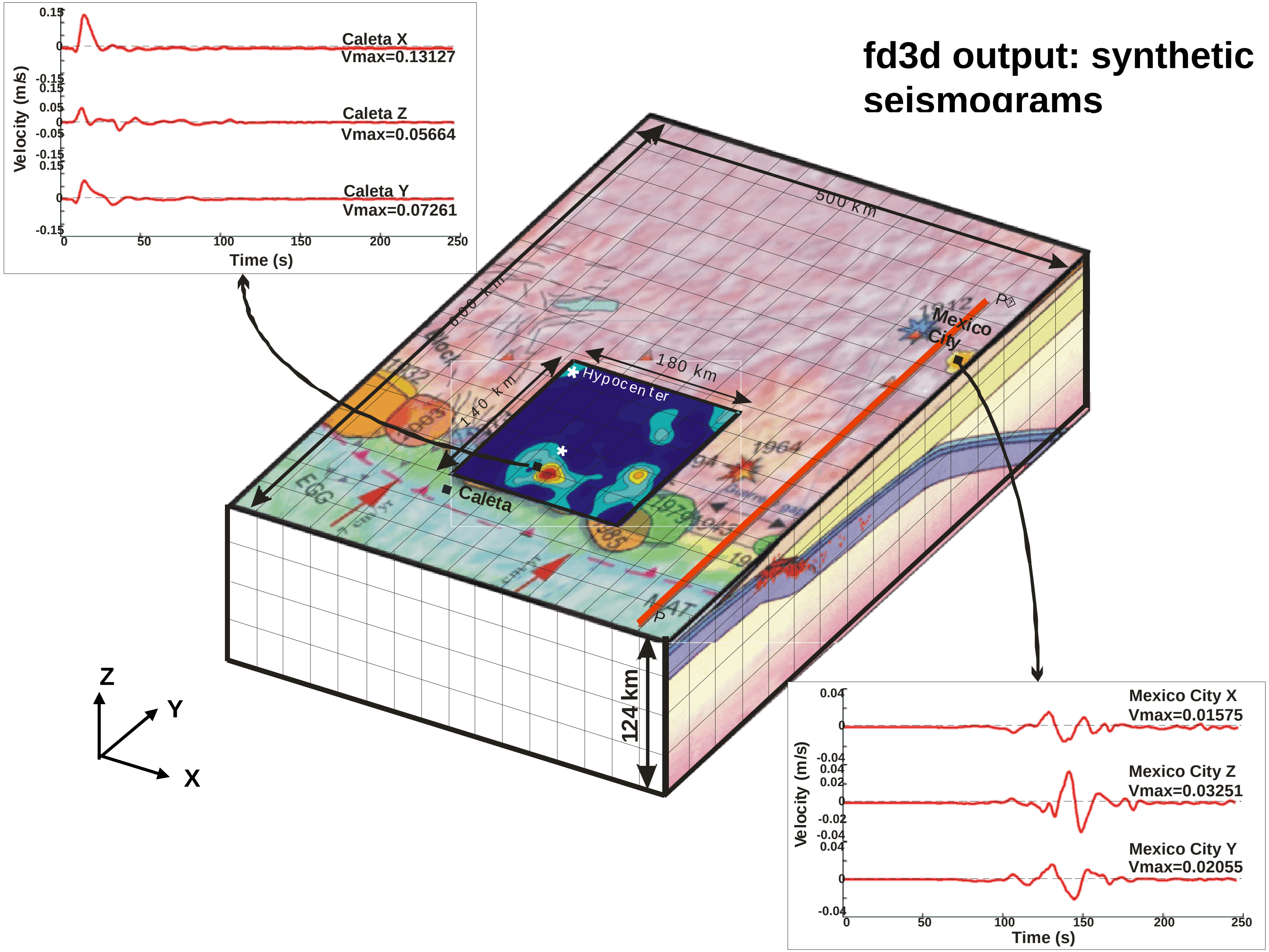
- 2nd order accurate in time
- 4th order accurate in space

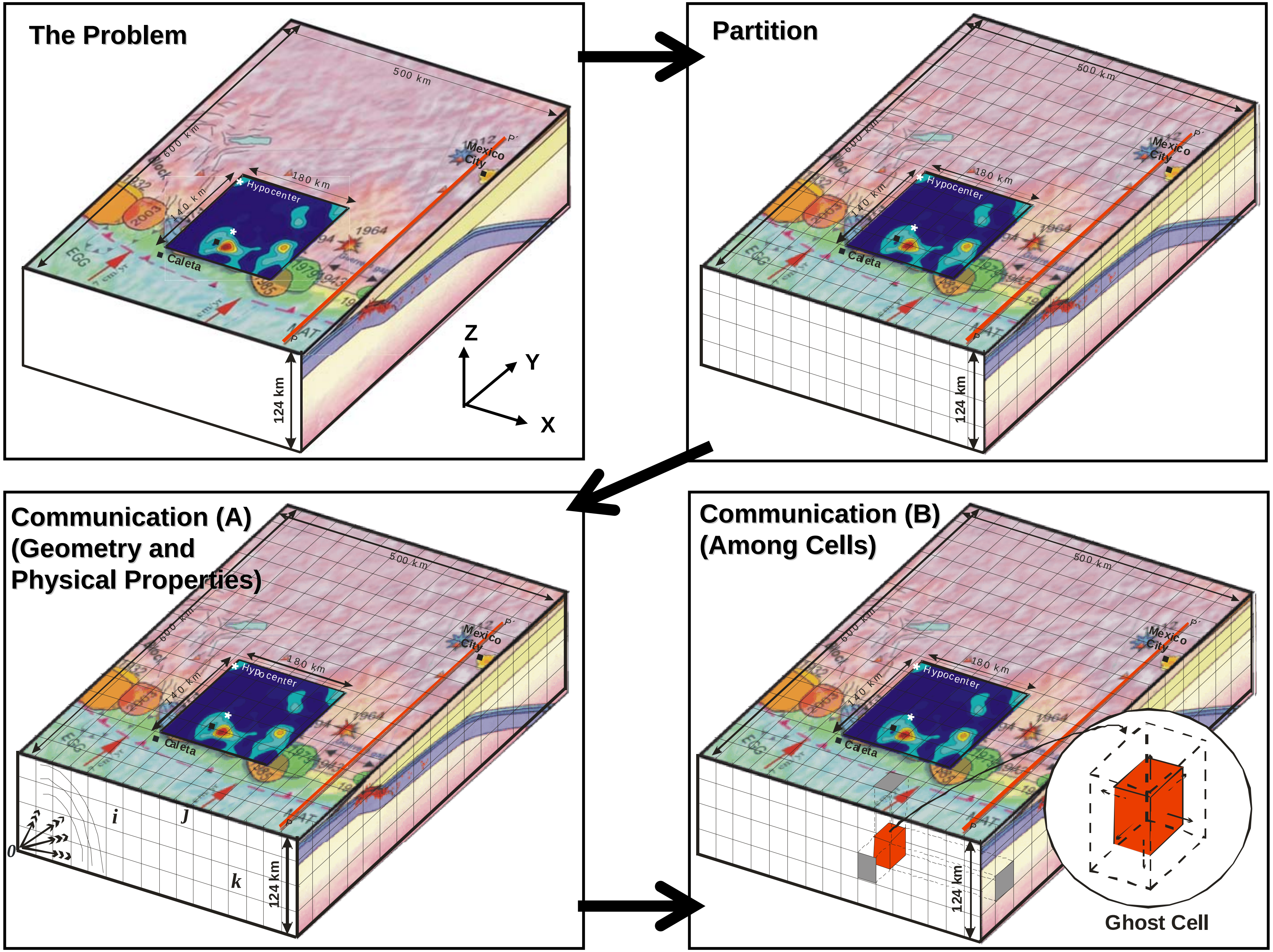
Regular grid partitioning

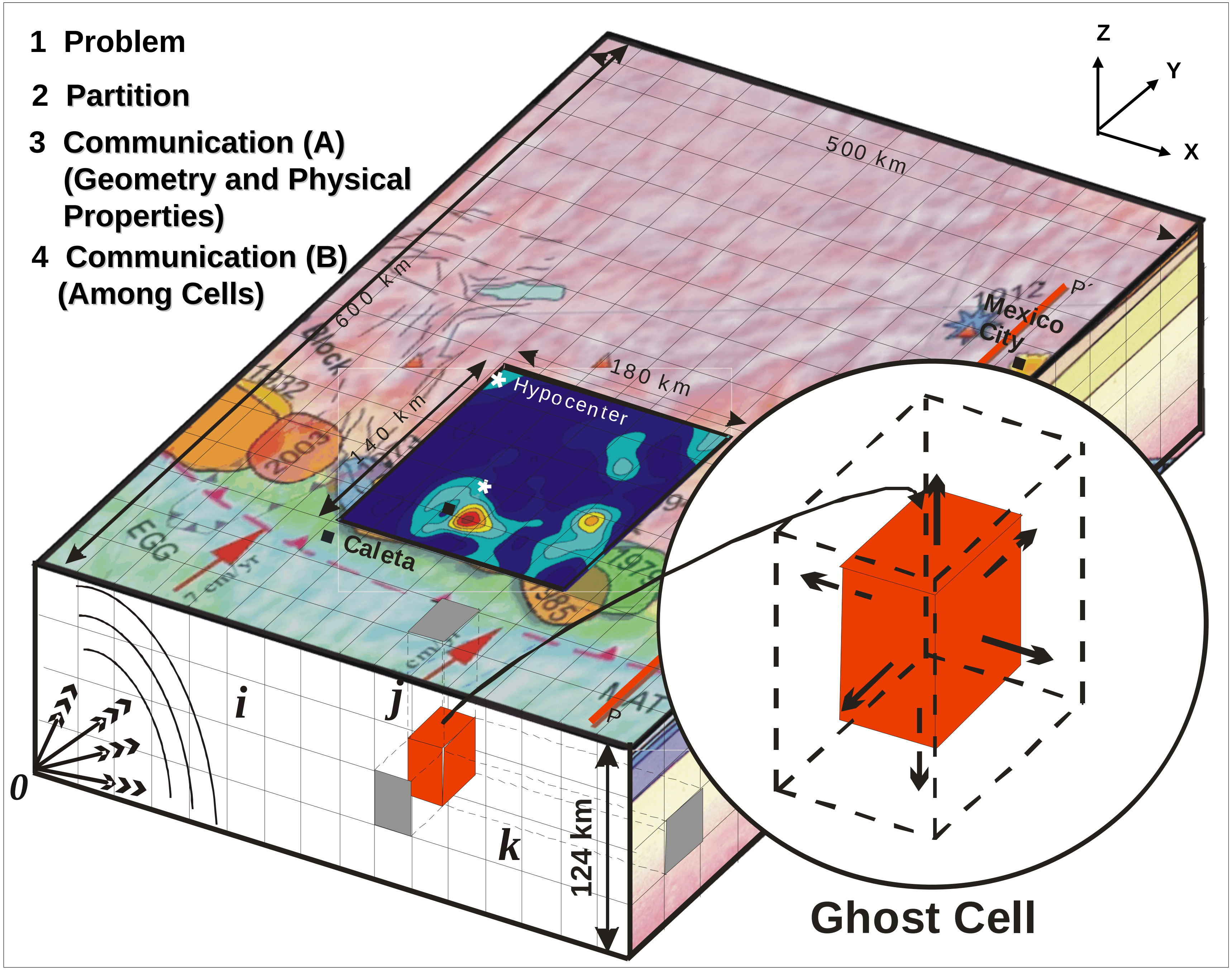
Halo exchange



fd3d output: synthetic seismograms




The Problem
Partition
Communication (A)
(Geometry and
Physical Properties)
Communication (B)
(Among Cells)
Ghost Cell
500 km
600 km
180 km
140 km
124 km
Hypocenter
Caleta
Mexico
City
1964
Z
Y
X
i
j
k
0


1 Problem
2 Partition
3 Communication (A)
(Geometry and Physical
Properties)
4 Communication (B)
(Among Cells)
Z
Y
X
500 km
600 km
180 km
140 km
Hypocenter
Caleta
Mexico
City
P'
P
i
j
k
0
124 km
Ghost Cell

Science & Technology Facilities Council

The benchmark cases

Size of domain is 500 x 260 x 124 km

Series of models:

• 500m resolution	1000 x 520 x 248 grid
• 250m resolution	…
• 125m resolution	…
• 62.5m resolution	…
• 31.25m resolution	16000 x 8320 x 3968

HECToR vs. Jaguar 'Pf'

HECToR dual-core

Core
- 2.8Ghz clock frequency
- SSE SIMD FPU (2flops/cycle = 5.6GF peak)

Cache Hierarchy
- L1 Dcache/Icache: 64k/core
- L2 D/I cache: 1M/core
- SW Prefetch and loads to L1
- Evictions and HW prefetch to L2

Memory
- 6 GB/node = 4 GB + 2 GB
- Dual Channel DDR2
- 10GB/s peak @ 667MHz

Jaguar 'Pf' quad-core

Core
- 2.3Ghz clock frequency
- SSE SIMD FPU (4flops/cycle = 9.2GF peak)

Cache Hierarchy
- L1 Dcache/Icache: 64k/core
- L2 D/I cache: 512 KB/core
- L3 Shared cache 2MB/Socket
- SW Prefetch and loads to L1,L2,L3
- Evictions and HW prefetch to L1,L2,L3

Memory
- 16 GB/node symmetric
- Dual Channel DDR2
- 12GB/s peak @ 800MHz

from Jason Beech-Brandt, Cray

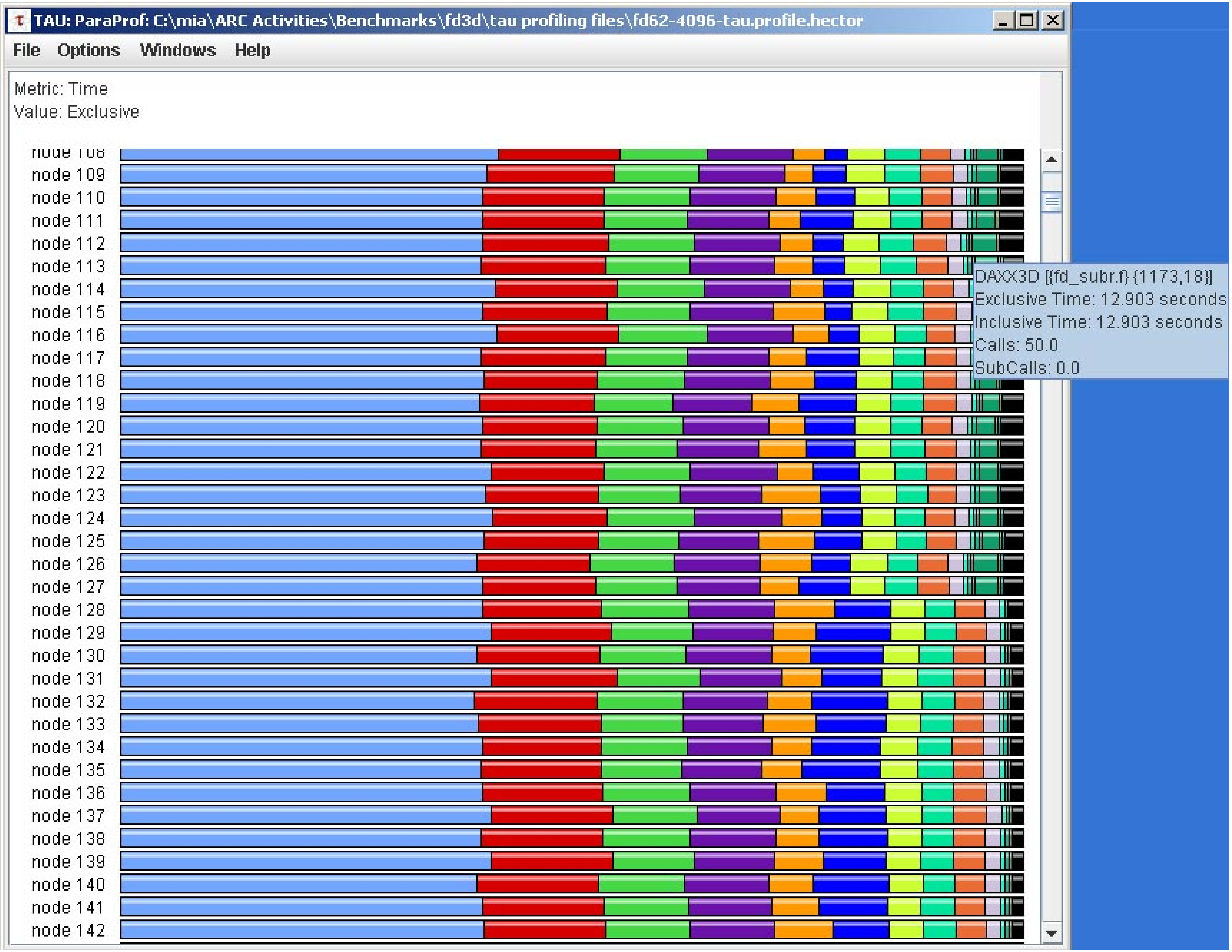
Science & Technology Facilities Council

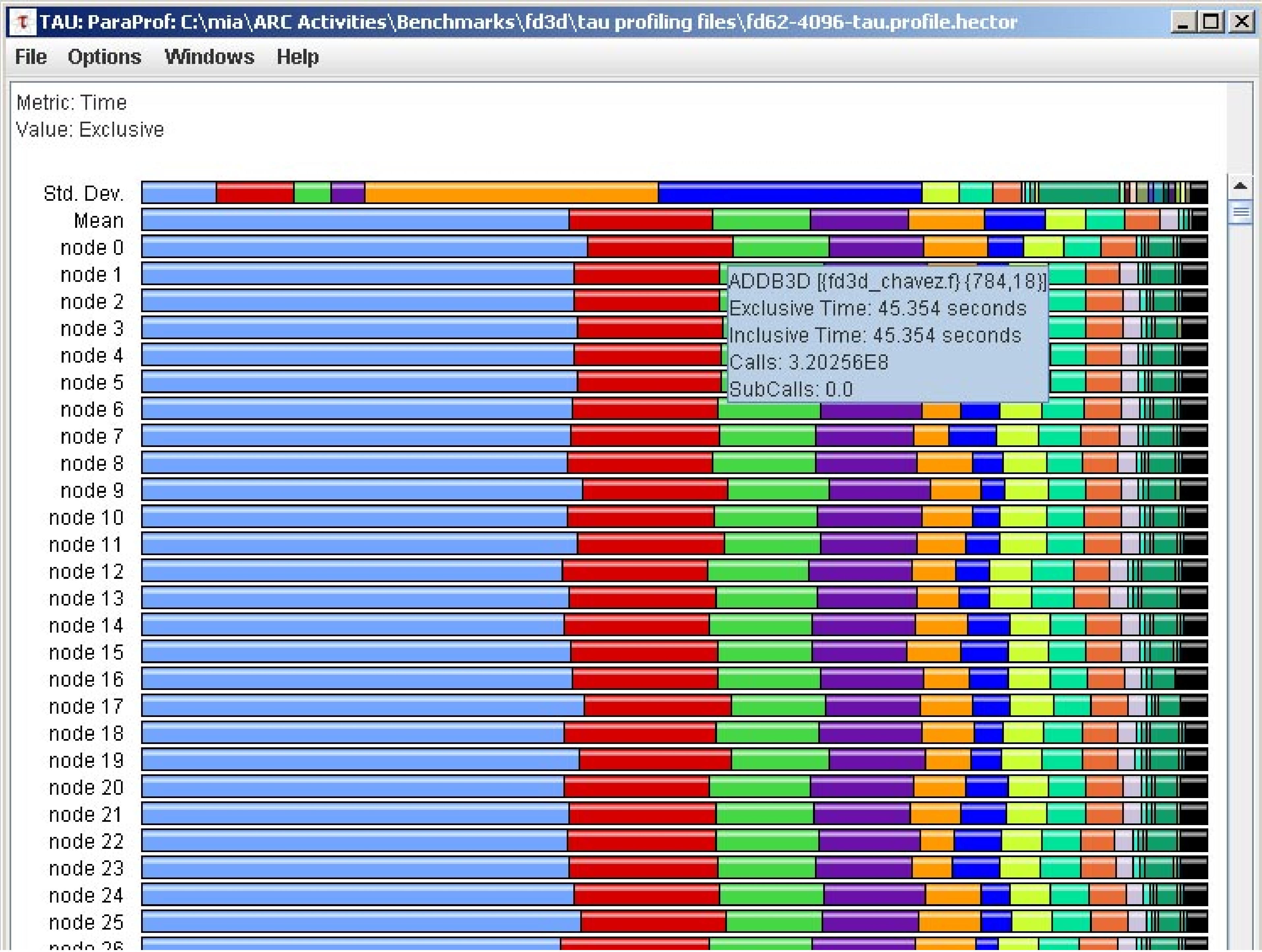
Optimizations

Opt 1: change MPI_sndrcv to MPI_IRecv, MPI_ISend preposting receives before buffer copies

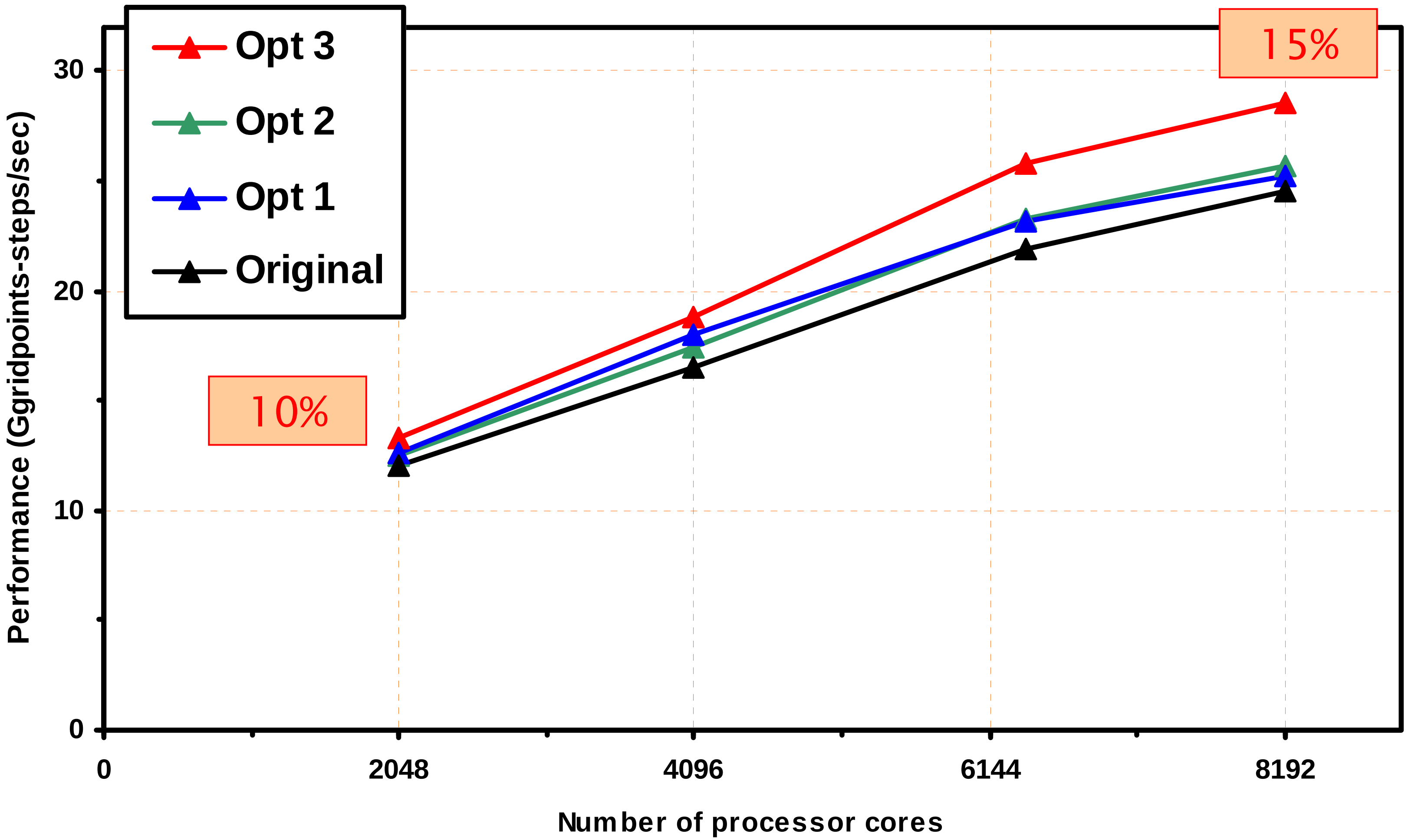
Opt 2: Opt 1 + BC code replace loops involving array syntax by a triply-nested loop so that the order of memory accesses is explicit

Opt 3: Opt 2 + two subroutines were being called 320 million times – push loop into subroutines


TAU: ParaProf: C:\mia\ARC Activities\Benchmarks\fd3d\tau profiling files\fd62-4096-tau.profile.hector
File Options Windows Help
Metric: Time
Value: Exclusive
node 109
node 110
node 111
node 112
node 113
node 114
node 115
node 116
node 117
node 118
node 119
node 120
node 121
node 122
node 123
node 124
node 125
node 126
node 127
node 128
node 129
node 130
node 131
node 132
node 133
node 134
node 135
node 136
node 137
node 138
node 139
node 140
node 141
node 142
DAXX3D [{fd_subr.f} {1173,18}]
Exclusive Time: 12.903 seconds
Inclusive Time: 12.903 seconds
Calls: 50.0
SubCalls: 0.0


TAU: ParaProf: C:\mia\ARC Activities\Benchmarks\fd3d\tau profiling files\fd62-4096-tau.profile.hector
File Options Windows Help
Metric: Time
Value: Exclusive
Std. Dev.
Mean
node 0
node 1
node 2
node 3
node 4
node 5
node 6
node 7
node 8
node 9
node 10
node 11
node 12
node 13
node 14
node 15
node 16
node 17
node 18
node 19
node 20
node 21
node 22
node 23
node 24
node 25
ADDB3D [{fd3d_chavez.f} {784,18}]
Exclusive Time: 45.354 seconds
Inclusive Time: 45.354 seconds
Calls: 3.20256E8
SubCalls: 0.0

Optimizations on HECToR



Science & Technology Facilities Council

Vectorization

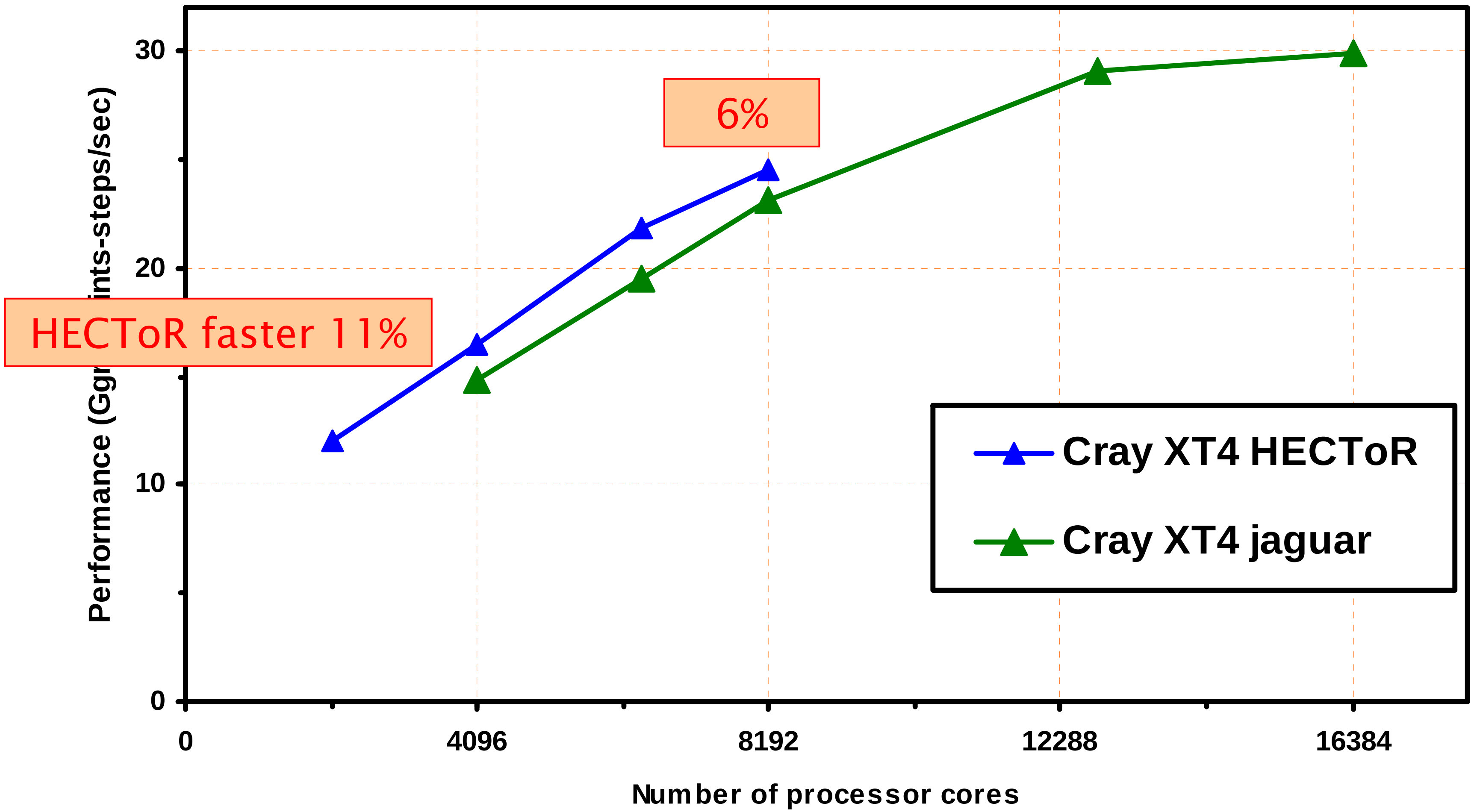
PGI compiler with –O3 –fastsse

836, Generated 3 alternate loops for the inner loop
Generated vector sse code for inner loop
Generated 8 prefetch instructions for this loop
Generated vector sse code for inner loop
Generated 8 prefetch instructions for this loop
Generated vector sse code for inner loop
Generated 8 prefetch instructions for this loop
Generated vector sse code for inner loop
Generated 8 prefetch instructions for this loop

Craypat HWPC

```
USER
------------------------------------------------------------------------
  Time%                                             100.0%
  Time                                         2442.711073 secs
  Imb.Time                                              -- secs
  Imb.Time%                                             --
  Calls                          0.0 /sec              2.0 calls
  DATA_CACHE_MISSES           26.767M/sec      64384203739 misses
  PAPI_TOT_INS              1273.576M/sec    3063435776248 instr
  PAPI_L1_DCA                724.935M/sec    1743745514390 refs
  PAPI_FP_OPS                557.064M/sec    1339951513747 ops
  User time (approx)        2405.380 secs    6735065200928 cycles  98.5%Time
  Average Time per Call                        1221.355536 sec
  CrayPat Overhead : Time     0.0%
  HW FP Ops / User time      557.064M/sec    1339951513747 ops   9.9%peak(DP)
  HW FP Ops / WCT            548.551M/sec
  HW FP Ops / Inst                                   43.7%
  Computational intensity     0.20 ops/cycle          0.77 ops/ref
  Instr per cycle                                     0.45 inst/cycle
  MIPS                    2608284.49M/sec
  MFLOPS (aggregate)      1140867.64M/sec
  Instructions per LD & ST   56.9% refs               1.76 inst/ref
  D1 cache hit,miss ratios   96.3% hits               3.7% misses
  D1 cache utilization (M)   27.08 refs/miss         3.385 avg uses
```

62.5m resolution



Dual-core vs Quad-core

Headline Linpack performance per core is faster

QC 7.0 DC 4.8 Gflop/s/core x1.45

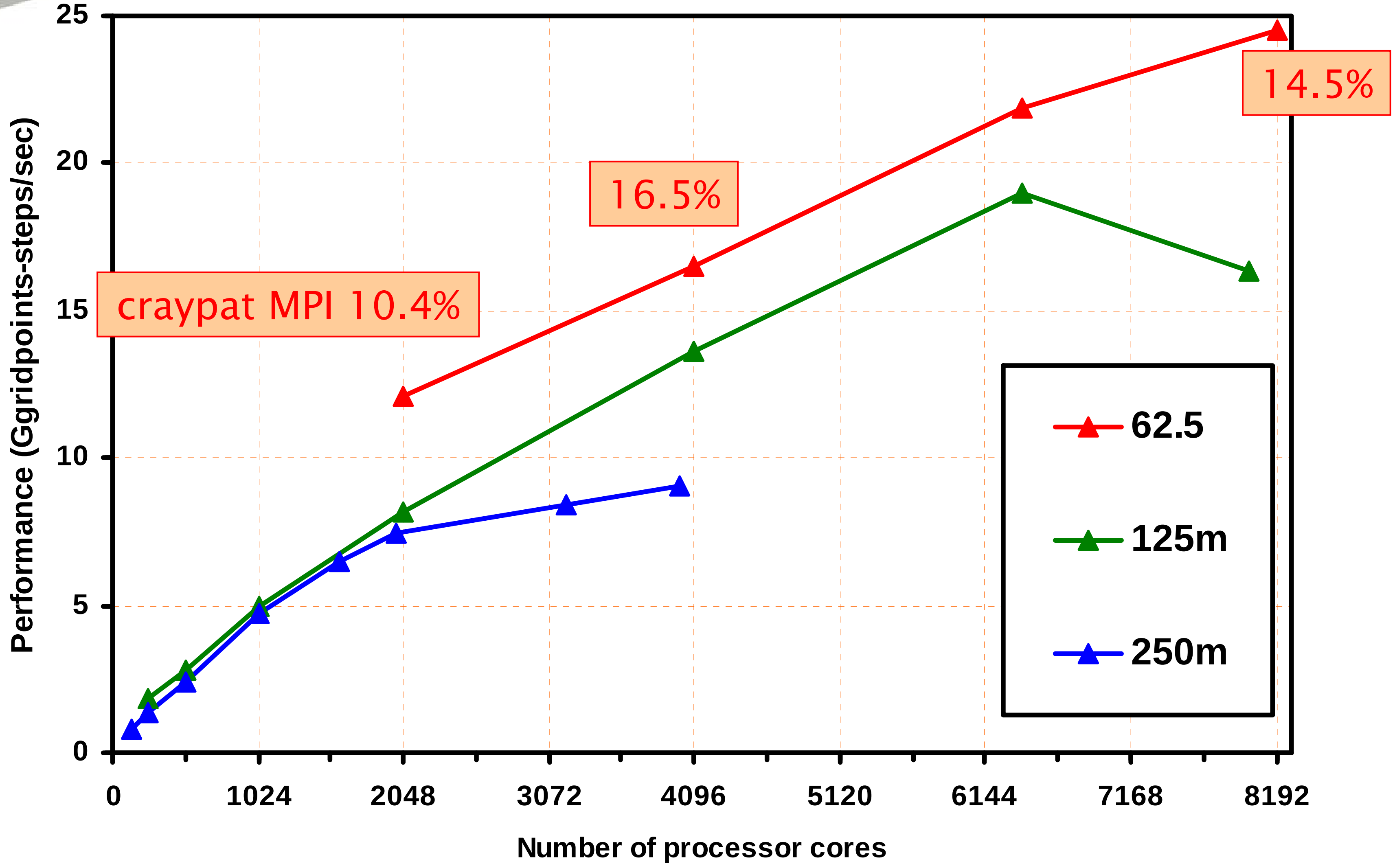
HECToR Allocation Unit is a notional processor running Linpack at 1Gflop/s for 1 hour

Gflop/s/core = AUs per core hour

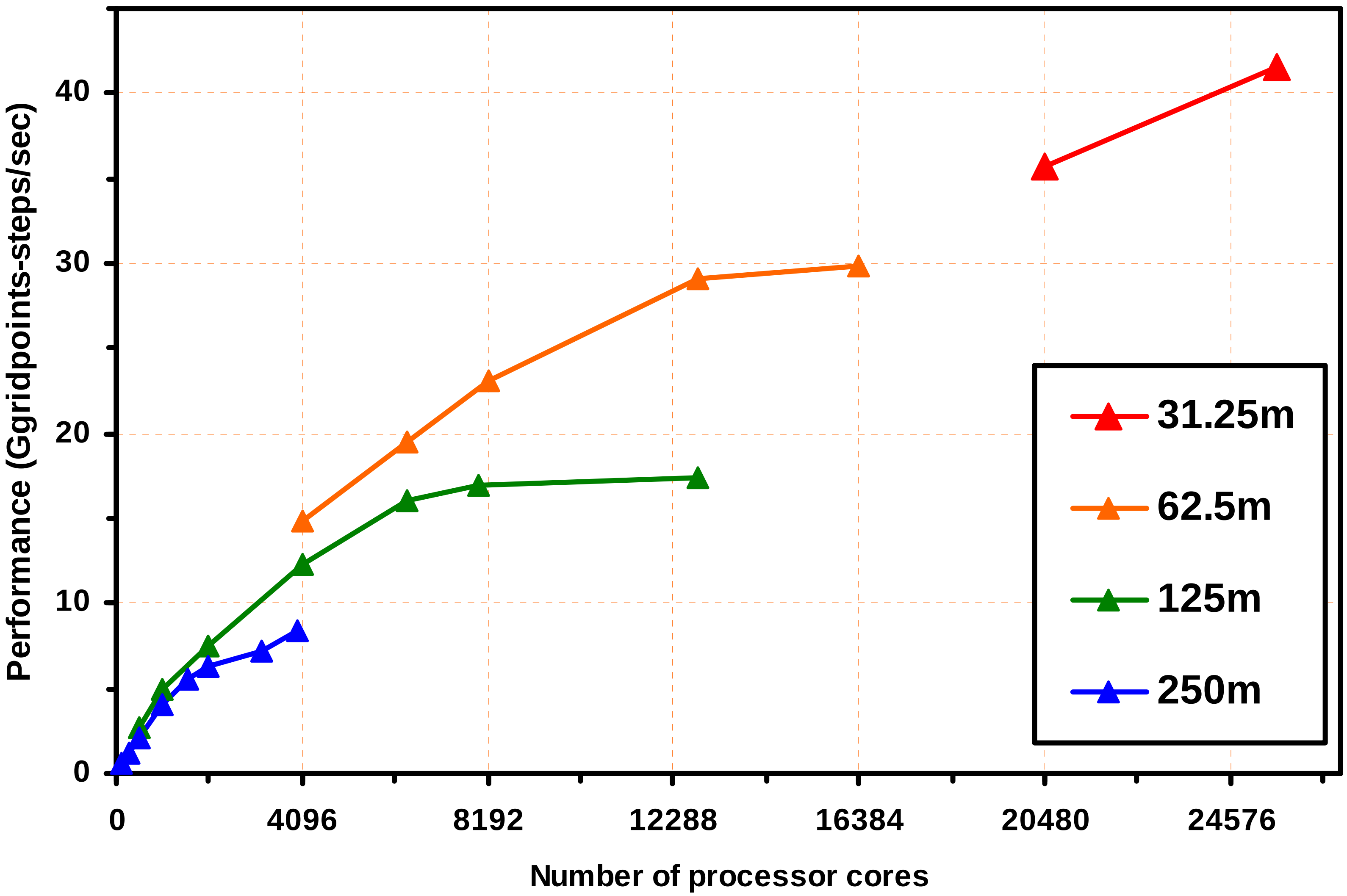
Unless your app scales as well as Linpack (x1.45) your Allocation Units will buy less app time

Science & Technology Facilities Council

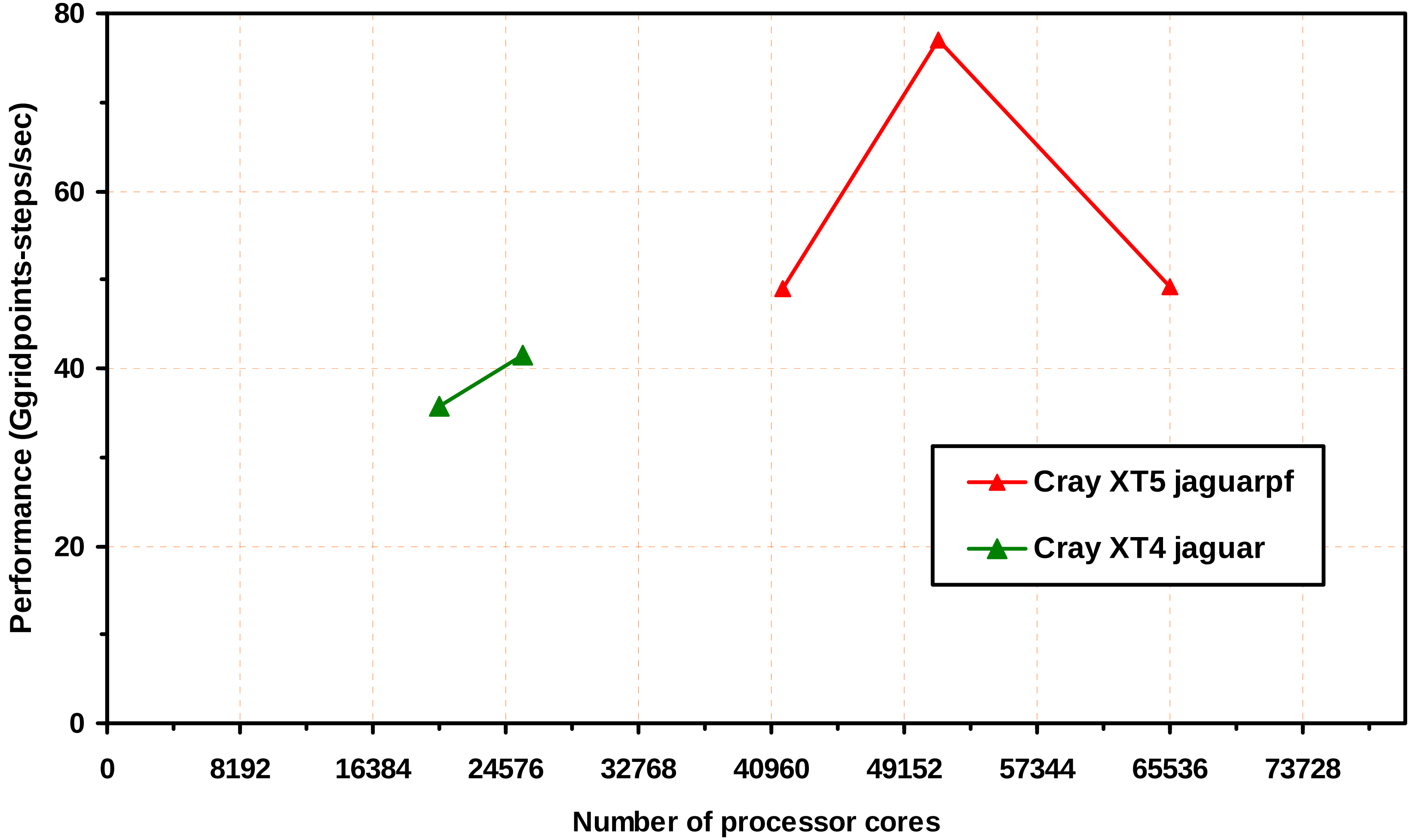
different resolutions on HECToR



different resolutions on jaguar



31.25m resolution from jaguar to jaguar 'Pf'



Science & Technology Facilities Council

Conclusions

We have carried out optimization and performance profiling of the seismic wave propagation code

We have run the code on dual-core and quad-core systems on up to 65536 cores

Performance continues to scale to around 65536 cores though there are some aspects which need further investigation

There are issues with the performance per core in moving from dual-core to quad-core with this code (and other codes of this type)

Science & Technology Facilities Council

Acknowledgements

This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-ASC05-00OR22725.

The authors also acknowledge support from the Scientific Computing Advanced Training (SCAT) project through Europe Aid contract II-0537-FC-FA. http://www.scat-alfa.eu

We are grateful to John Levesque of Cray Inc. for performing benchmark runs on the Jaguar Petaflop system.

If you have been … … thank you for listening

Mike Ashworth

http://www.cse.scitech.ac.uk/