

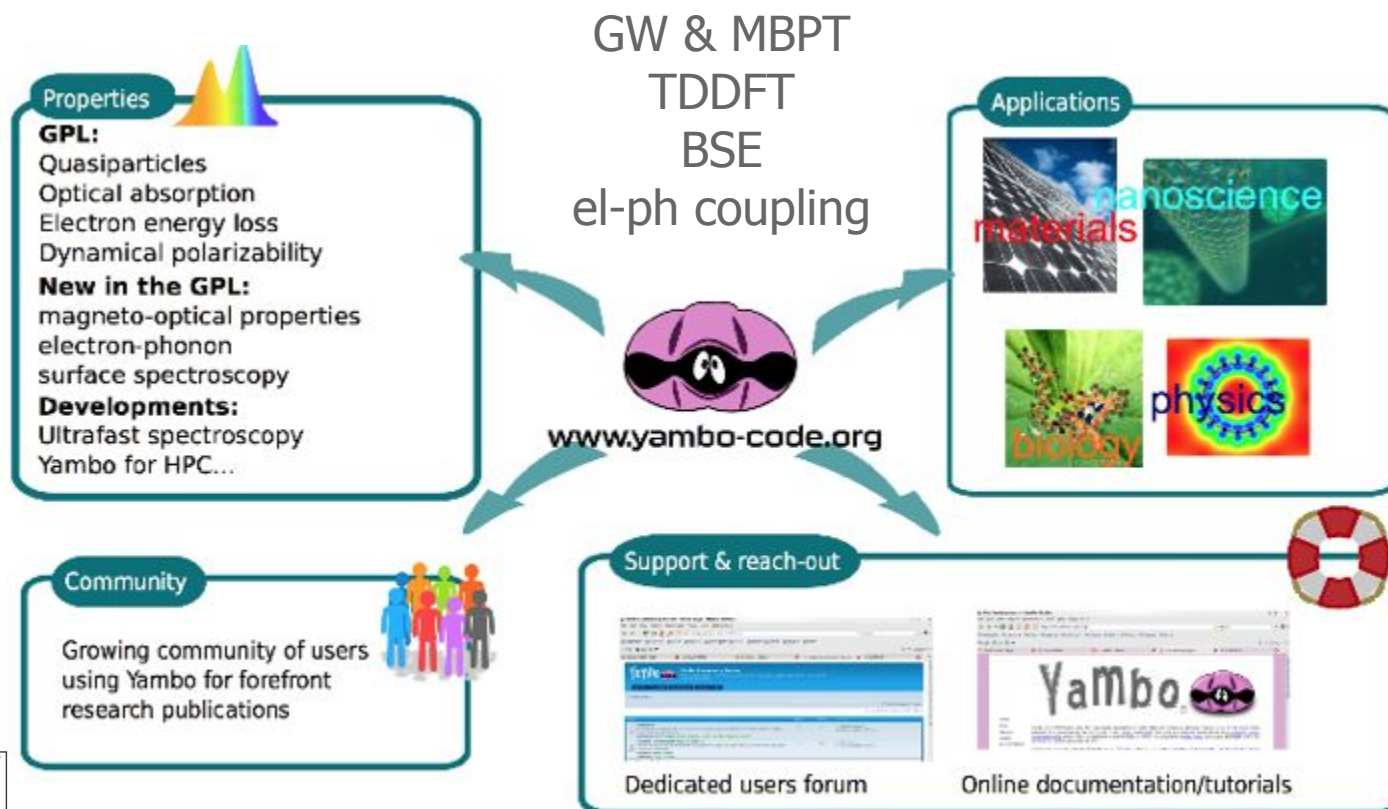


YOGA: Yambo on GPU Accelerators

Pietro Bonfà, Massimiliano Fatica, Andrea Ferretti,
Anton Kozhevnikov, Ivan Marri, Everett Phillips, Josh Romero

Eurohack 2018

Yambo Project



Yambo Parallelization

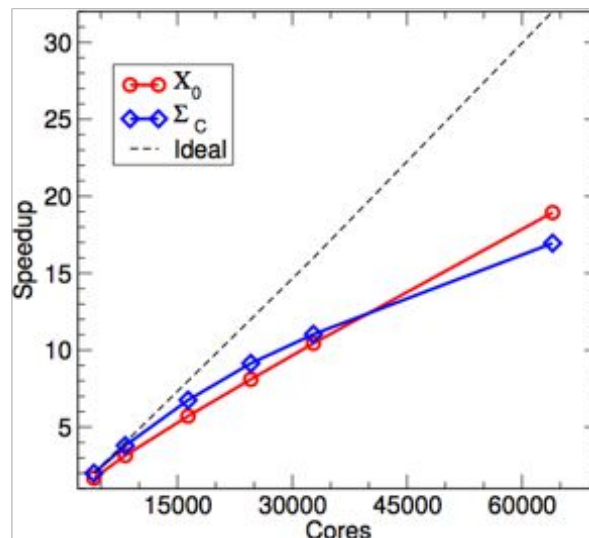
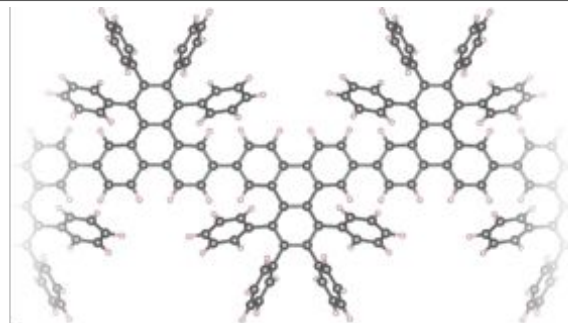


- yambo implements a **hybrid MPI+OpenMP** paradigm
- **MPI** works over several (3 to 5) **different levels**, according to the run level
- **OpenMP** works at a **lower level**, usually on space degrees of freedom (not always the case, though),
 - reaching very different levels of efficiency
- **parallel linear algebra** is supported (ScaLapack, SLEPC, PETSC)
- overall, yambo is quite **parallel oriented**



Yambo Performances

- Yambo **single GW calculation** scaling up to 1000 KNL nodes (~ 3 PFI/s)
- hybrid **MPI+OpenMP** +scaLapack
- Calculations relevant for an **active research field** (graphene nanoribbons)
- Performed on the recently deployed **KNL** partition of Marconi @ CINECA



Porting Strategy

Wave functions read
from QE not modified



moved to the
device once

MBPT numerical intense



well suited for OMP
and GPUs

CUDA FORTRAN
CUF KERNELS

WORK DONE:

identification of prototype use
cases



profiling



time consuming
kernels (hotspots)

Porting Steps

Trieste 2017: Cuda Fortran for Material Scientist, Feb 27- Mar 1

Barcelona 2018: MaX Hachathon, July 16 - 20

Lugano 2018: Eurohack, October 1-5



Events | Upcoming Events

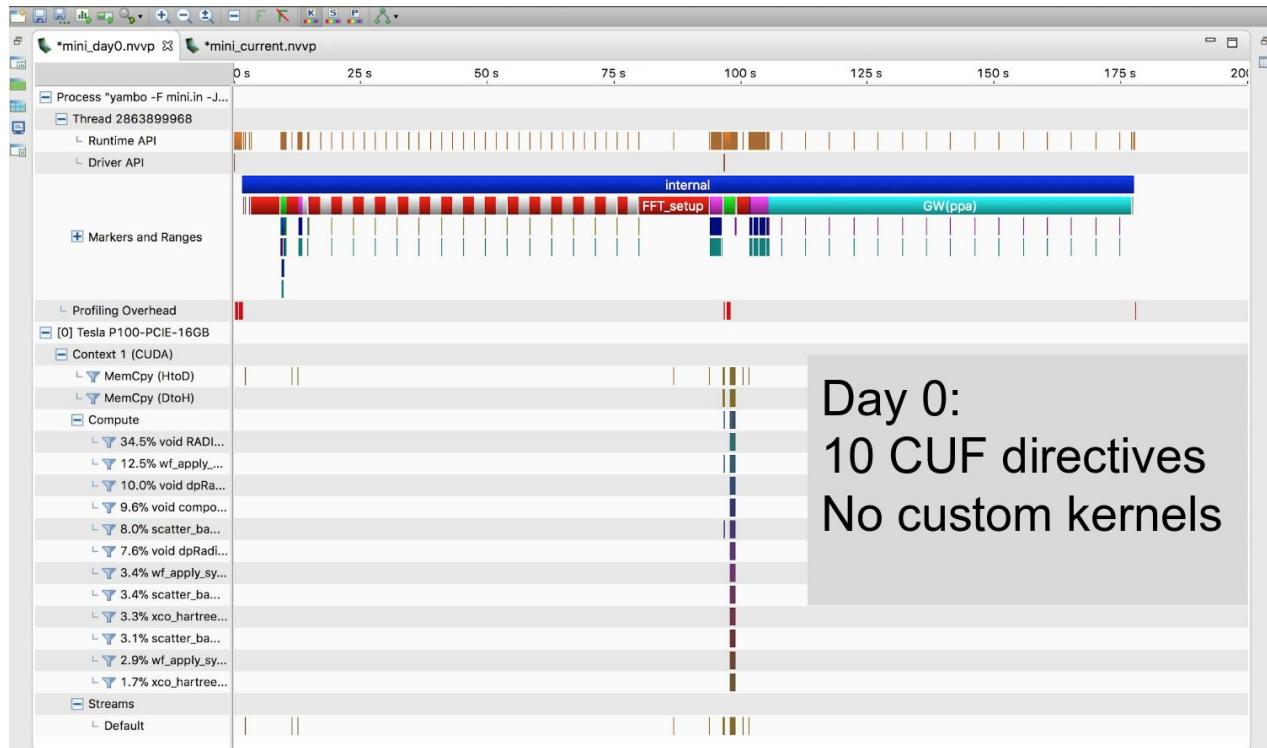
01.10.2018-05.10.2018

**EUROHACK18: GPU PROGRAMMING
HACKATHON**

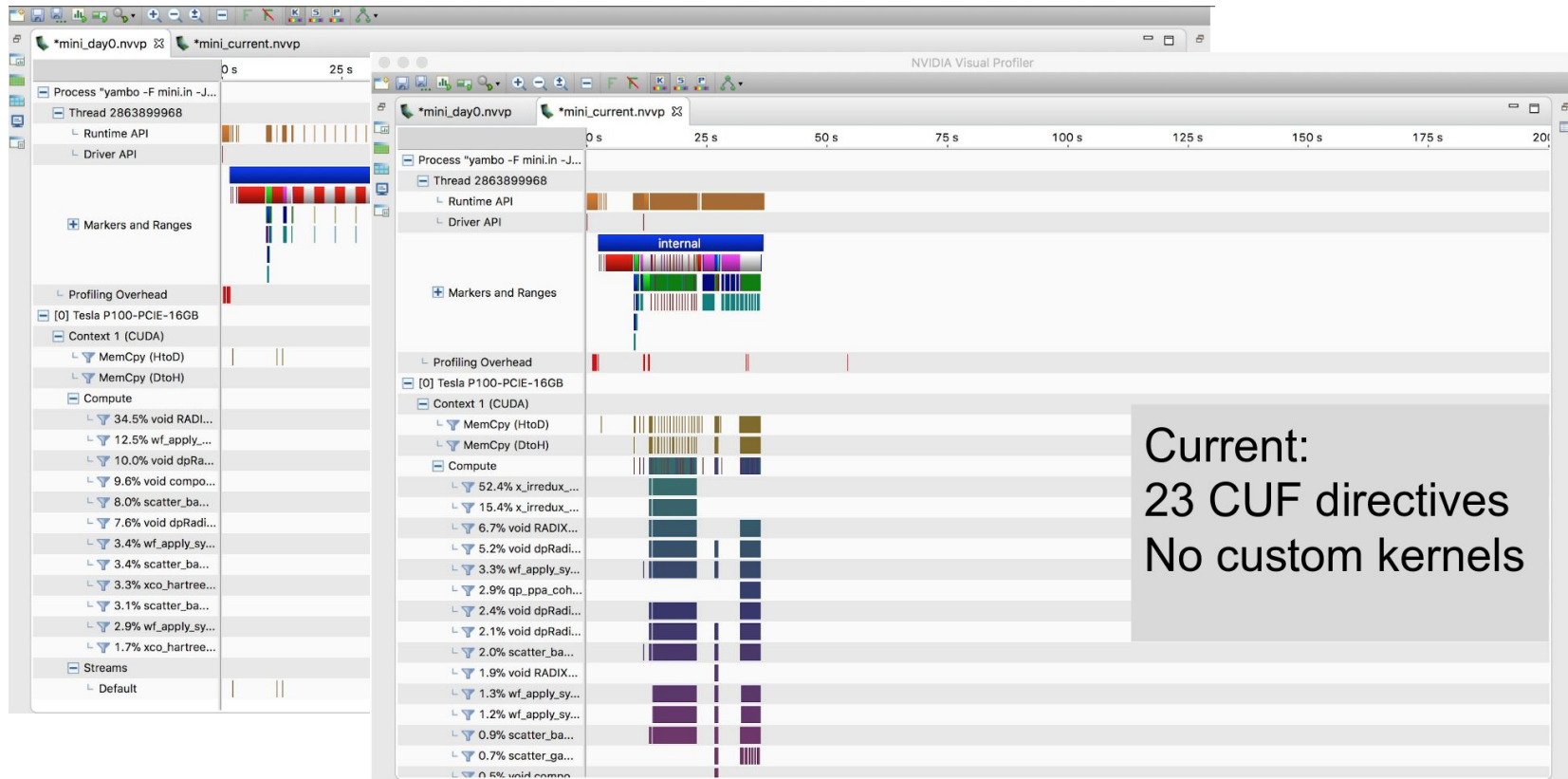
 Hotel De La Paix



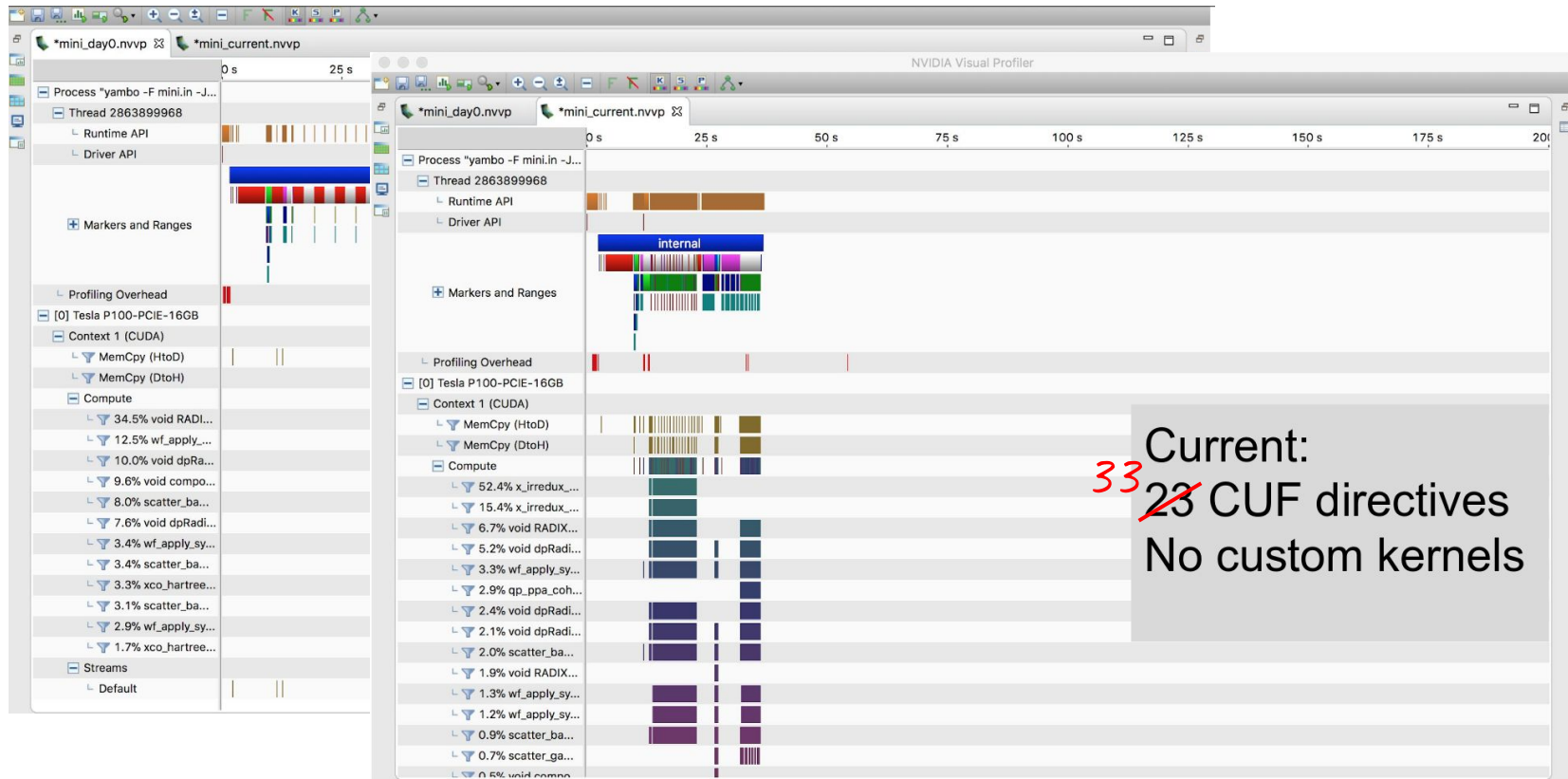
Profiling



Profiling

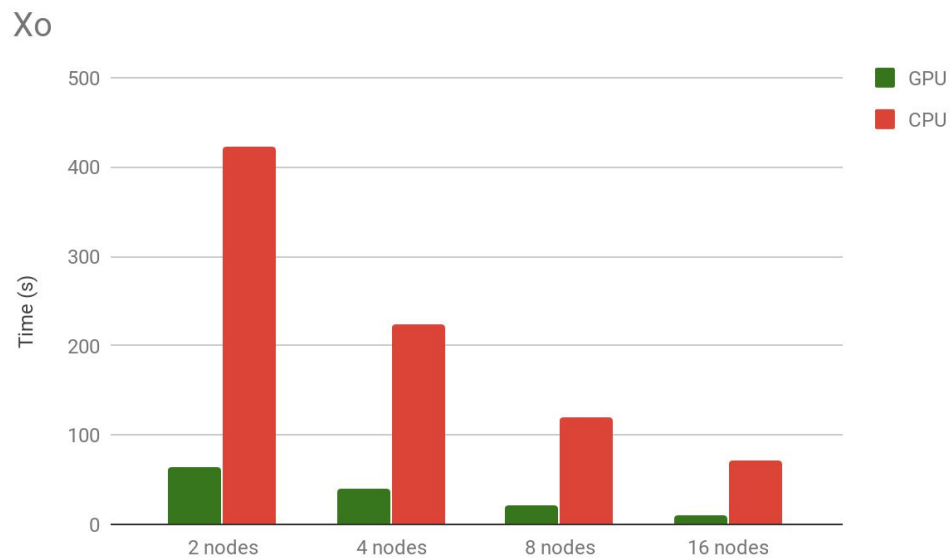


Profiling



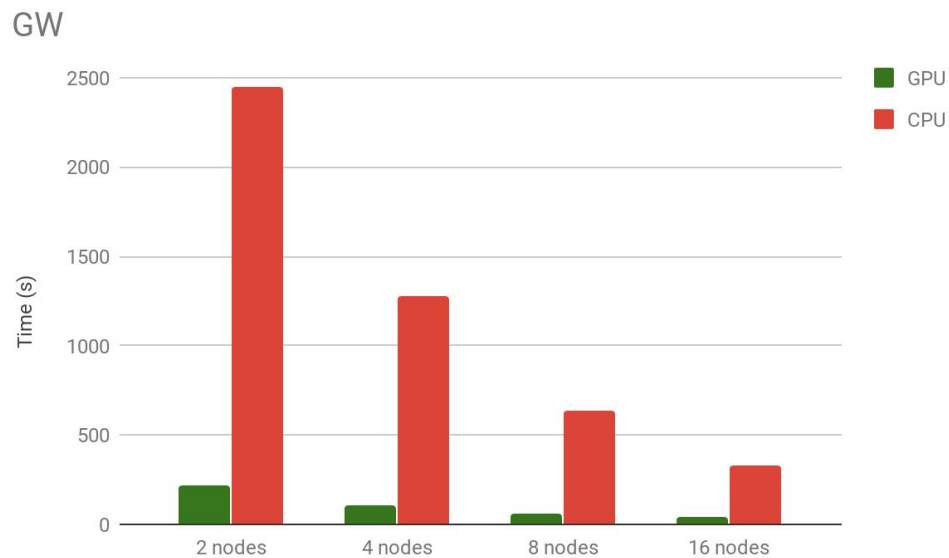
Performances

Full socket vs GPU on PizDaint XC50 partition.



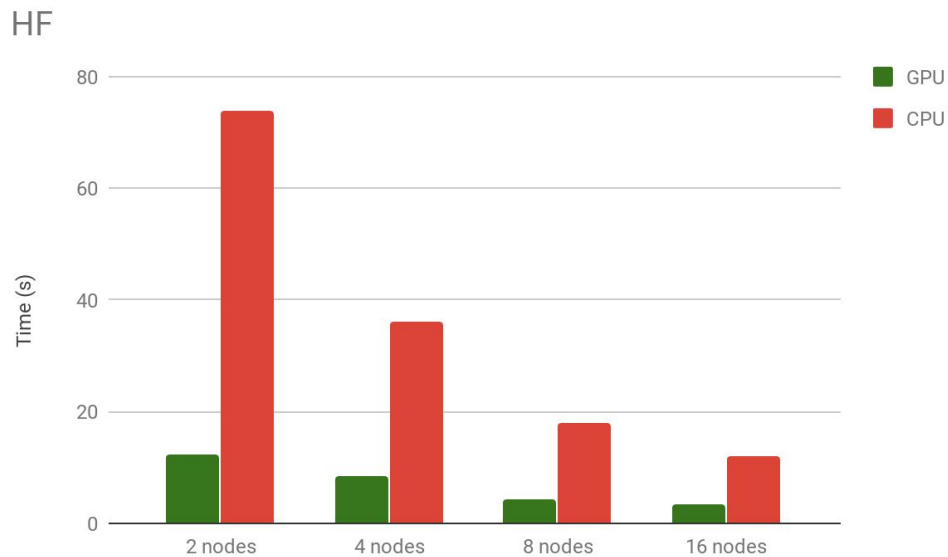
Performances

Full socket vs GPU on PizDaint XC50 partition.



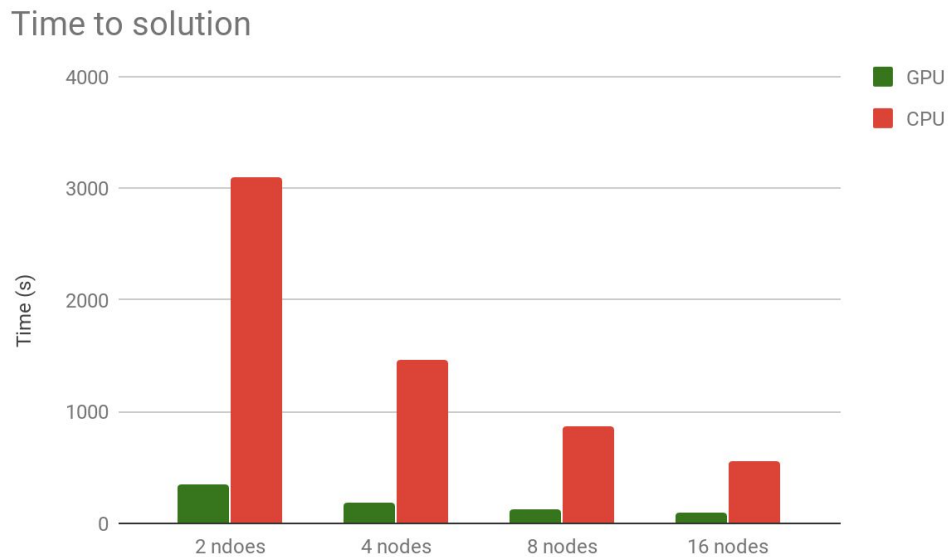
Performances

Full socket vs GPU on PizDaint XC50 partition.



Performances

Full socket vs GPU on PizDaint XC50 partition.



Conclusions

- CUDA Fortran based porting based on **cuf kernel directives** and cuda libraries (FFT)
- Profiling allowed to spot space also for some CPU optimizations.
- 5 to 10x speedup in optimized subroutines, **5 to 10x speedup** in time to solution.
- Small impact on the code, acceleration part localized in a few subroutines.

Conclusions

- CUDA Fortran based porting based on **cuf kernel directives** and cuda libraries (FFT)
- Profiling allowed to spot space also for some CPU optimizations.
- 5 to 10x speedup in optimized subroutines, **5 to 10x speedup** in time to solution.
- Small impact on the code, acceleration part localized in a few subroutines.

Thank you!