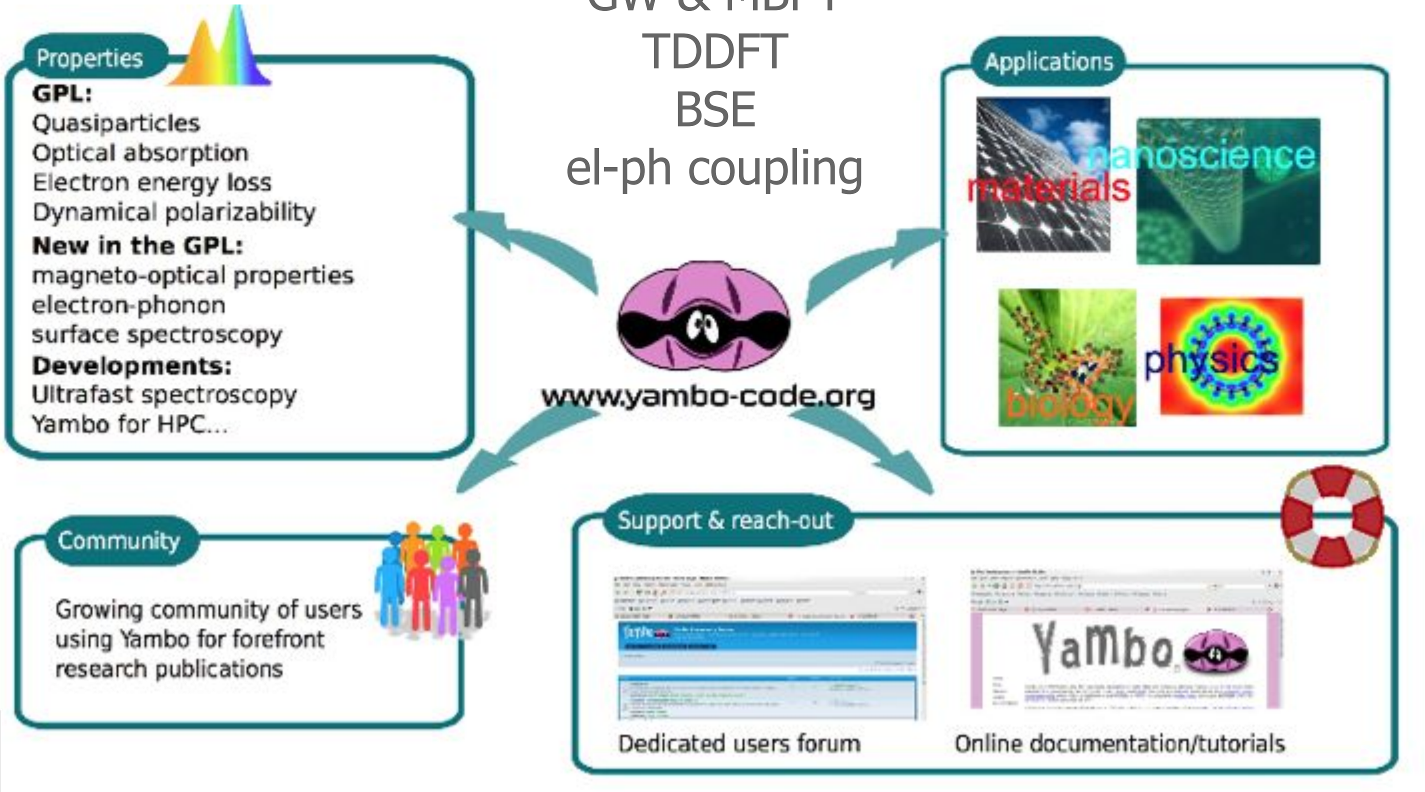
YOGA: Yambo on GPU Accelerators

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

Eurohack 2018

Yambo Project

GW & MBPT
TDDFT
BSE
el-ph coupling



Properties

GPL:
Quasiparticles
Optical absorption
Electron energy loss
Dynamical polarizability

New in the GPL:
magneto-optical properties
electron-phonon
surface spectroscopy

Developments:
Ultrafast spectroscopy
Yambo for HPC...

www.yambo-code.org

Applications

Community

Growing community of users using Yambo for forefront research publications

Support & reach-out

Dedicated users forum

Online documentation/tutorials

Yambo Parallelization

- yambo implements a **hybrid MPI+OpenMP** paradigm

- **MPI** works over several (3 to 5) **different levels**, according to the run level

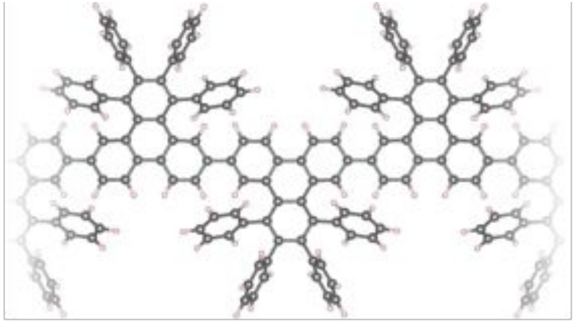
- **OpenMP** works at a **lower level**, usually on space degrees of freedom (not always the case, though),
- reaching very different levels of efficiency

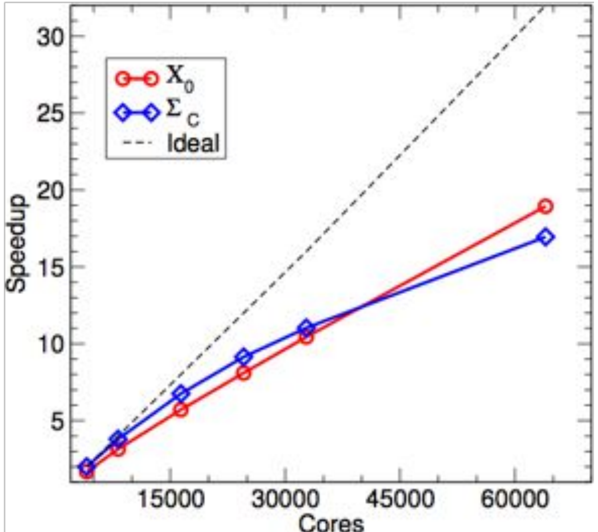
- **parallel linear algebra** is supported (ScaLapack, SLEPC, PETSC)

- overall, yambo is quite **parallel oriented**

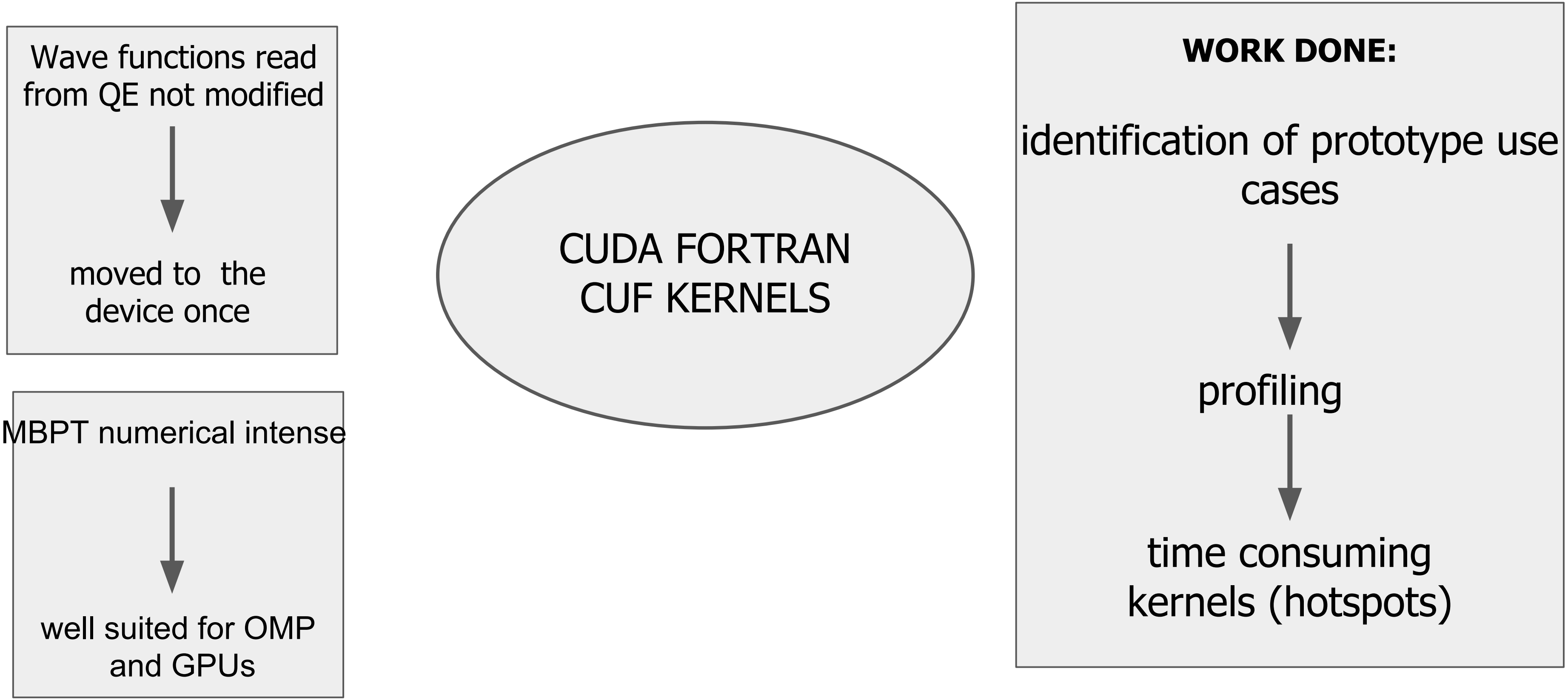
Yambo Performance

- Yambo **single GW calculation** scaling up to 1000 KNL nodes (~ 3 PFl/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

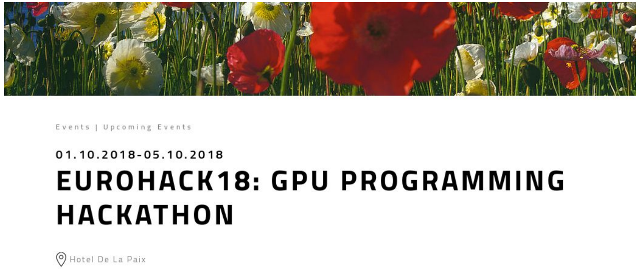
time consuming kernels (hotspots)

Porting Steps

Trieste 2017: Cuda Fortran for Materials Scientists, Feb 27- Mar 1

Barcelona 2018: MaX Hackathon, July 16 - 20

Lugano 2018: EuroHack, October 1-5



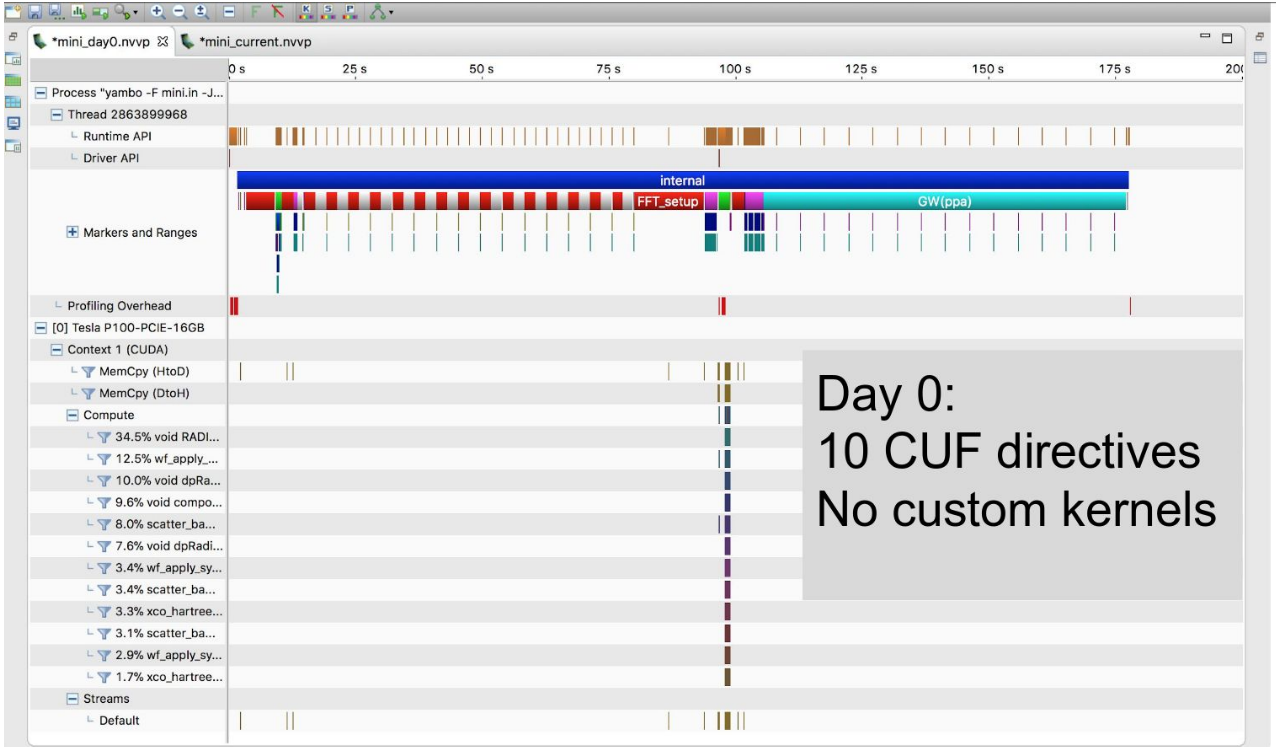
Events | Upcoming Events

01.10.2018-05.10.2018

EUROHACK18: GPU PROGRAMMING HACKATHON

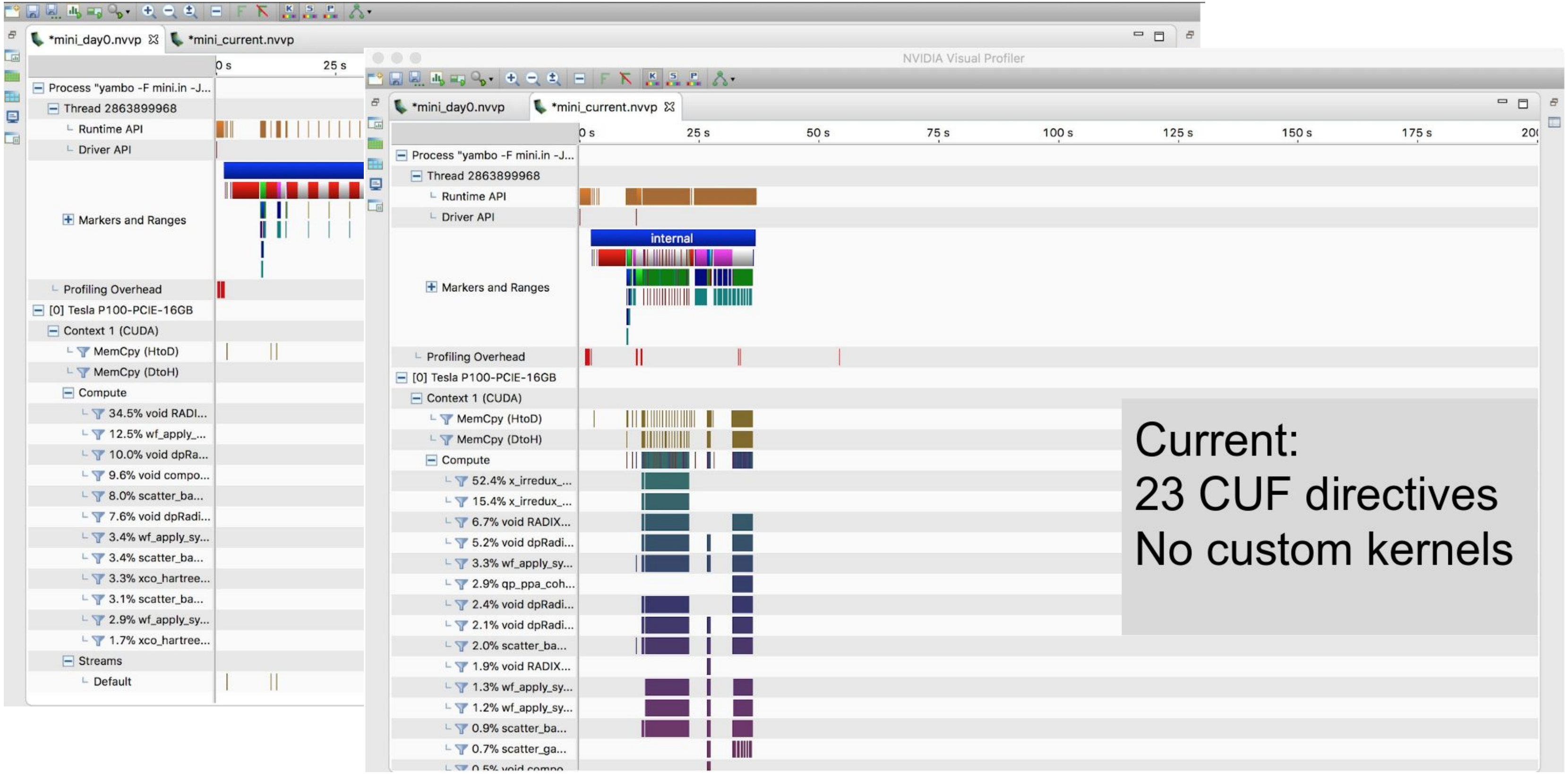
Hotel De La Paix

Profiling



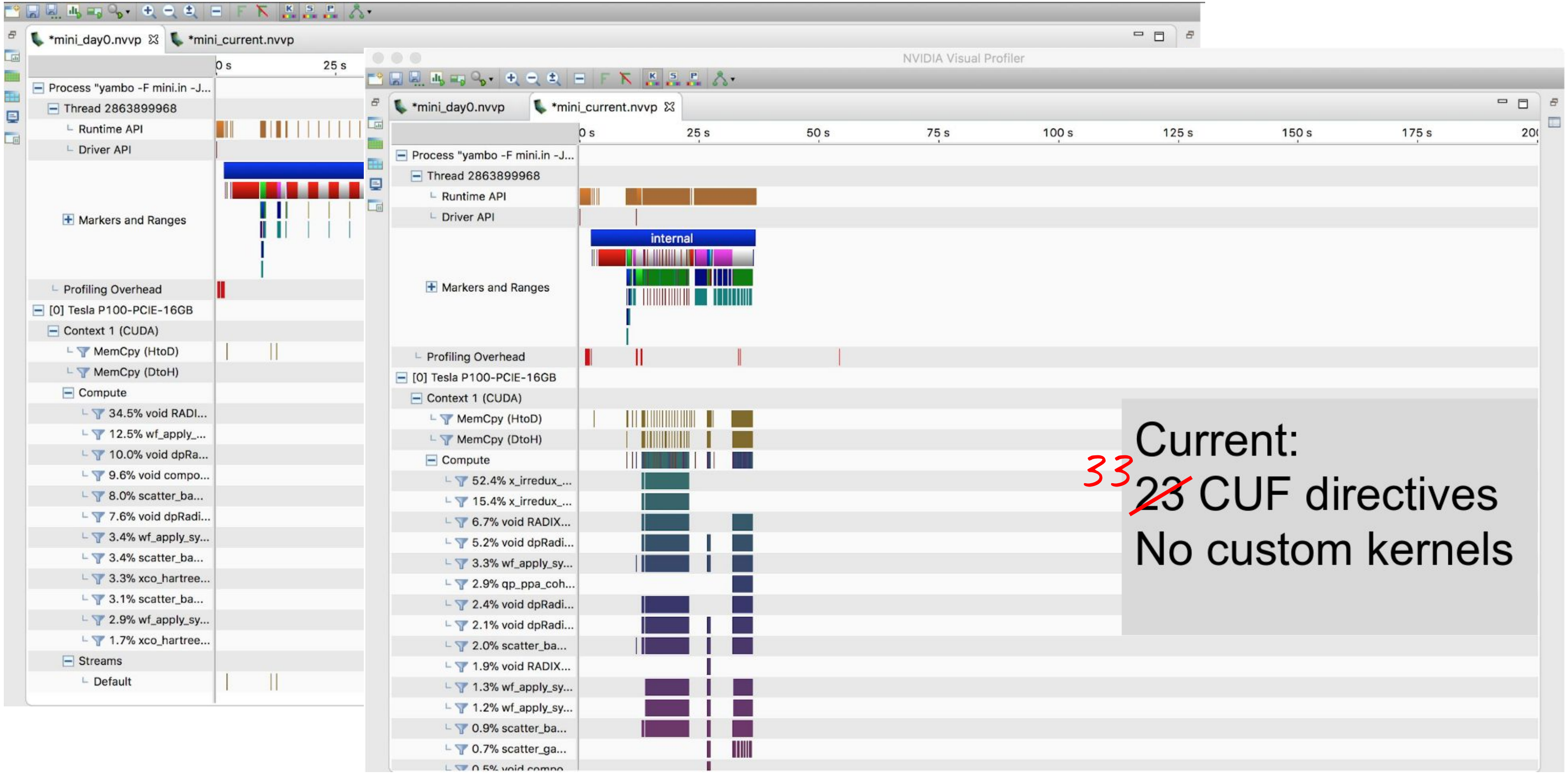
Day 0:
10 CUF directives
No custom kernels

Profiling



Current:
23 CUF directives
No custom kernels

Profiling



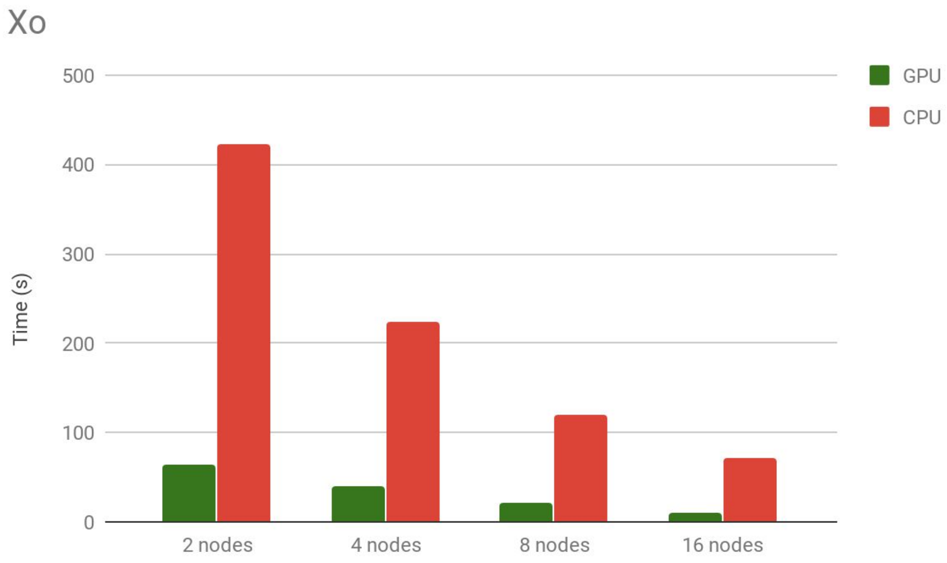
Current:

33 ~~23~~ CUF directives

No custom kernels

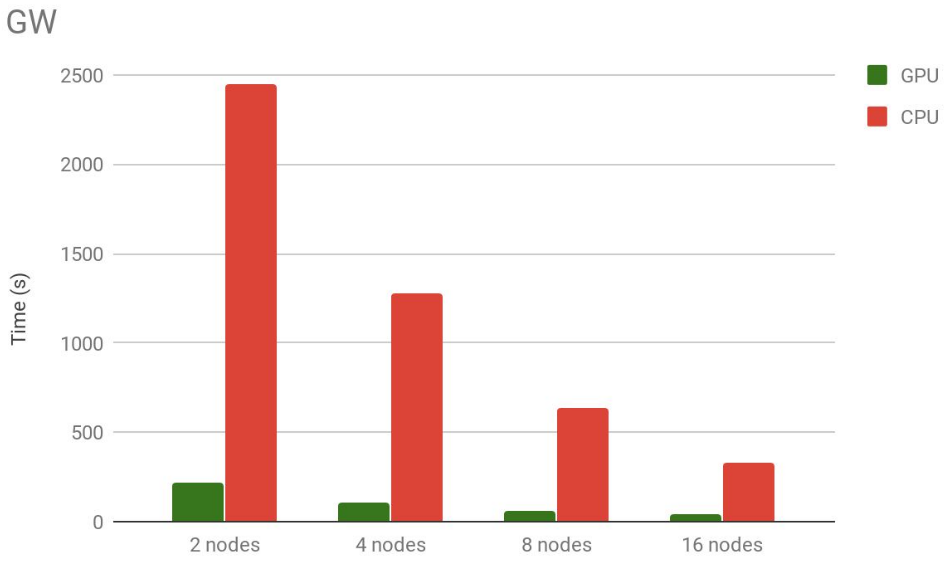
Performance

Full socket vs GPU on PizDaint XC50 partition.



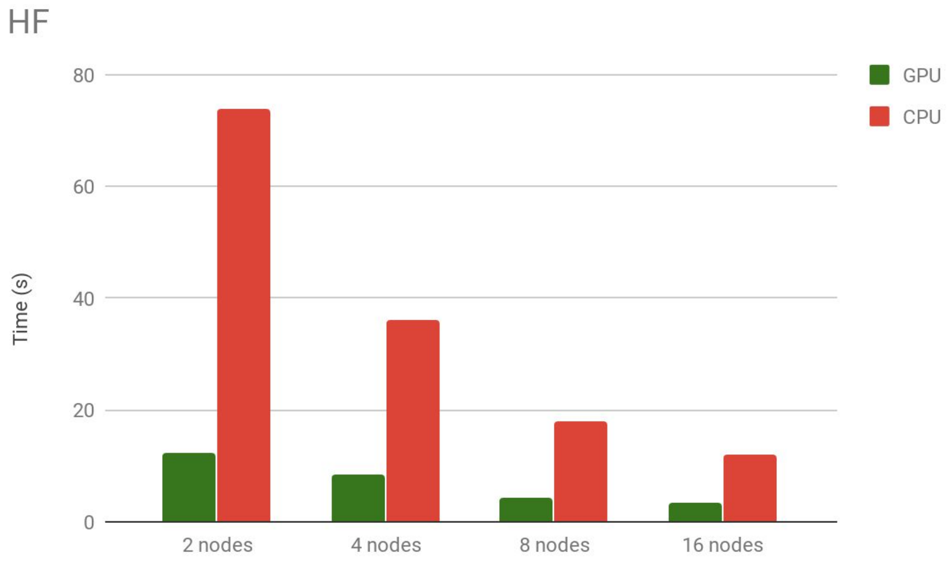
Performance

Full socket vs GPU on PizDaint XC50 partition.



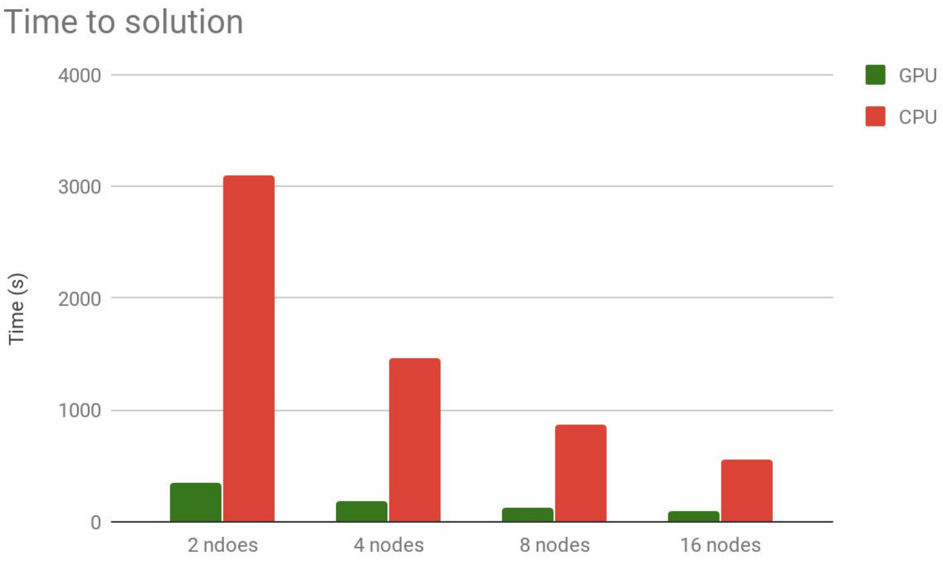
Performance

Full socket vs GPU on PizDaint XC50 partition.



Performance

Full socket vs GPU on PizDaint XC50 partition.



Conclusions

- CUDA Fortran based porting based on **cuf kernel directives** and cuda libraries (FFT)
- Profiling allowed us to spot room 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, accelerated part localized in a few subroutines.

Conclusions

- CUDA Fortran based porting based on **cuf kernel directives** and cuda libraries (FFT)
- Profiling allowed us to spot room 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, accelerated part localized in a few subroutines.

Thank you!