QE-GPU: between performance, correctness and sustainability.

Filippo SPIGA[1,2] <fs395@cam.ac.uk>

[1] High Performance Computing Service, University of Cambridge

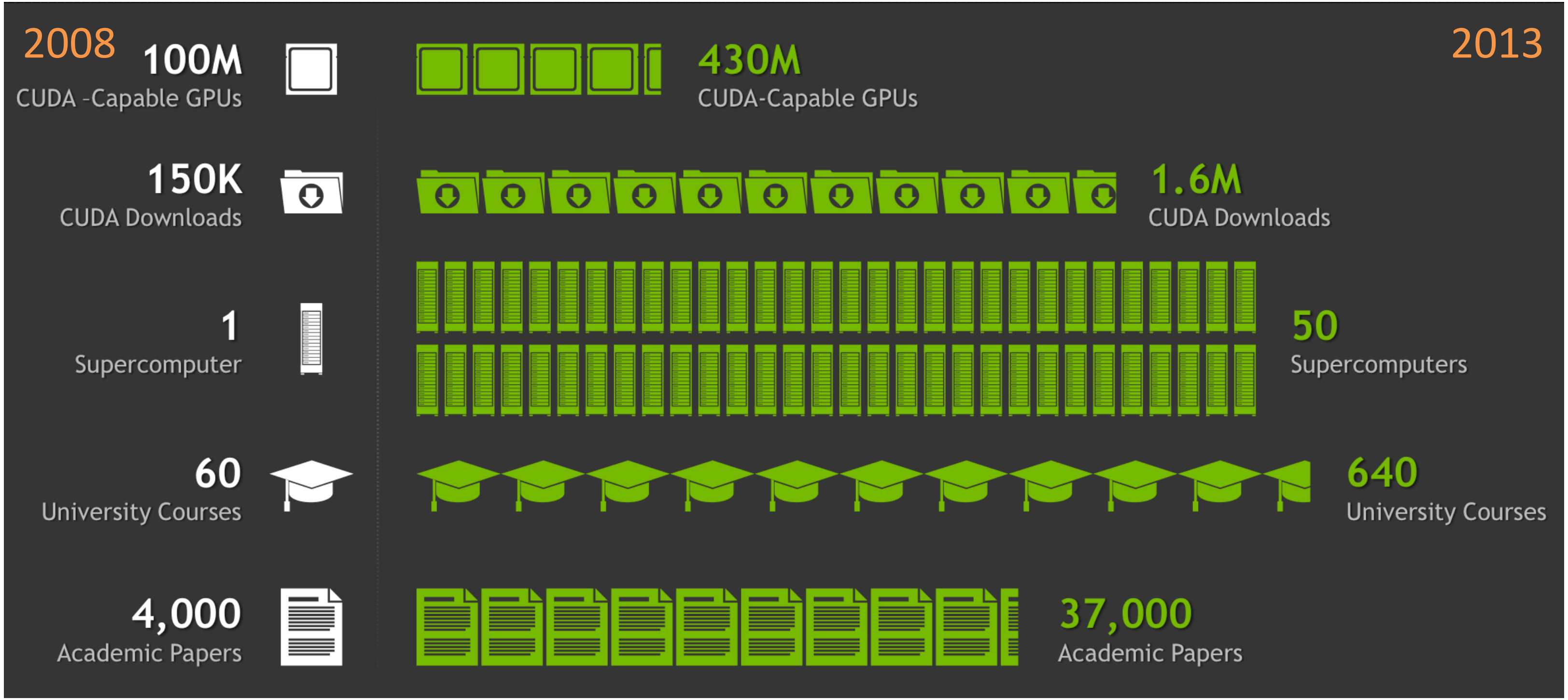
[2] Quantum ESPRESSO Foundation

«*What I cannot compute, I do not understand.*» (adapted from Richard P. Feynman)

Outline

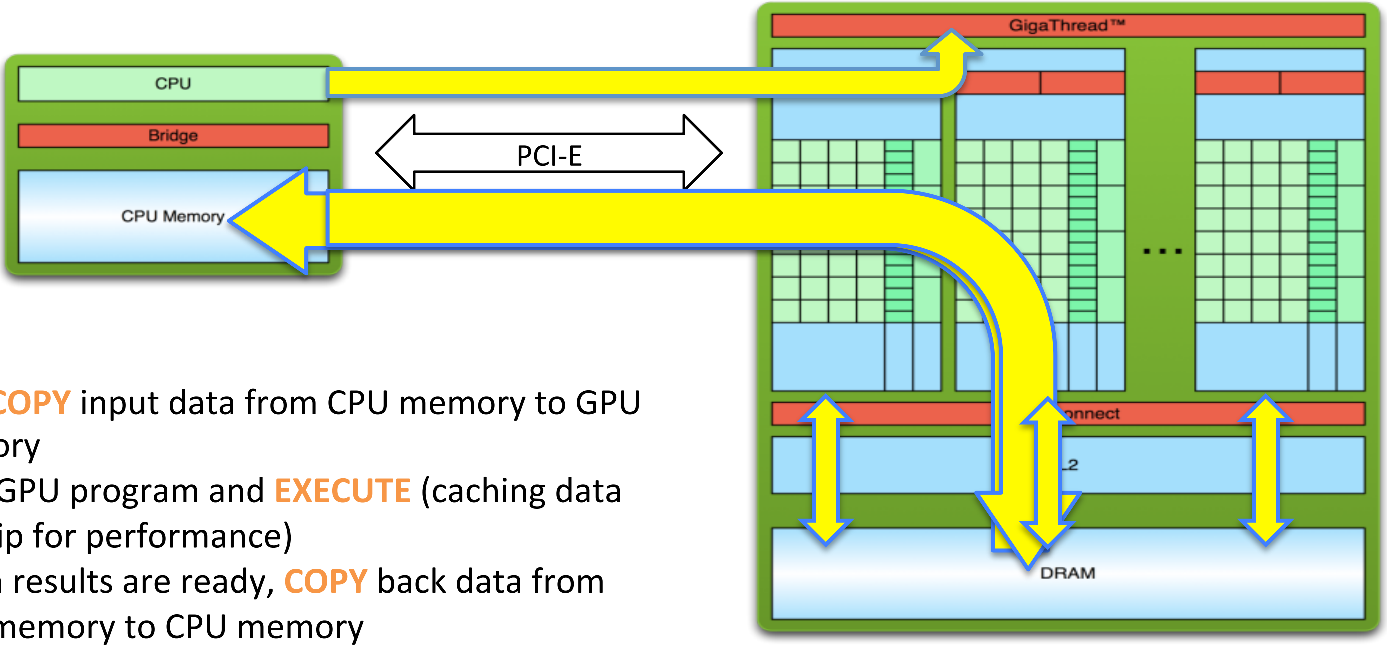
- Overview
- The Quantum ESPRESSO project
- PWscf GPU porting
- Performance evaluation
- Final considerations: performance, correctness and sustainability

GPU momentum continues to grow



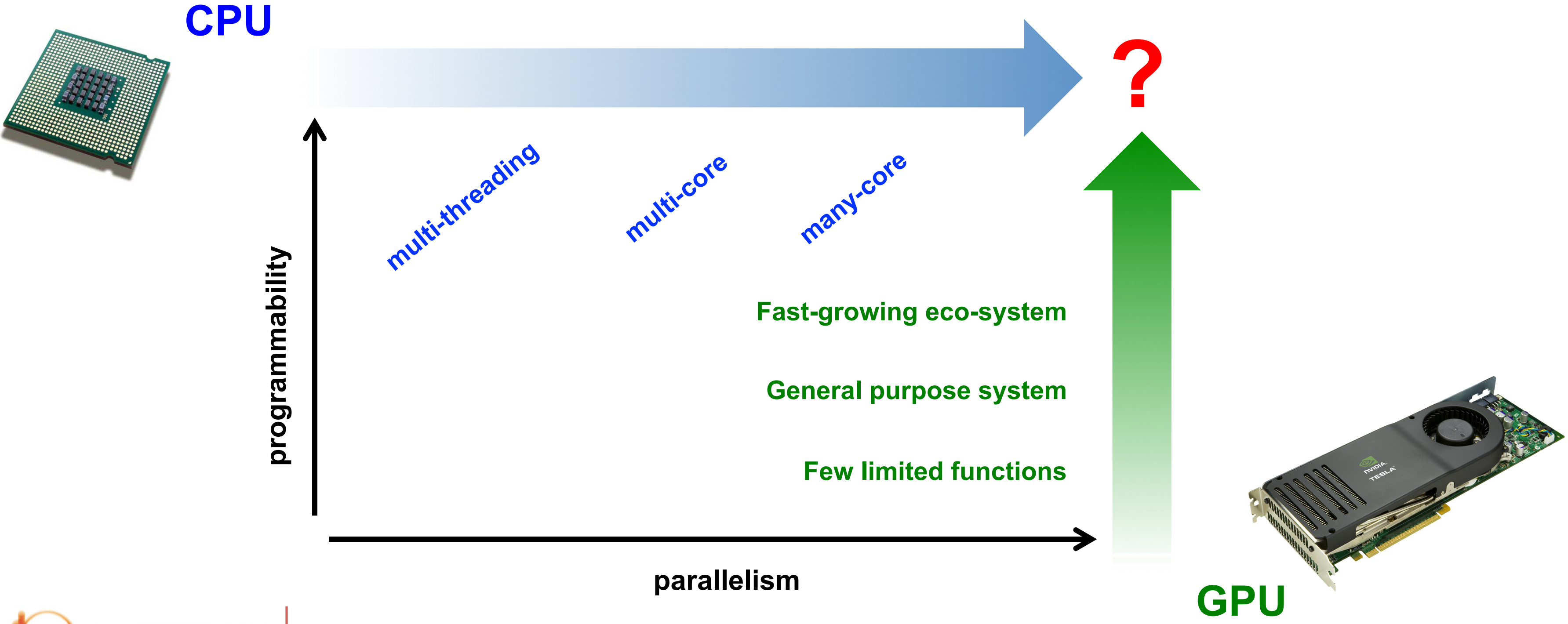
source: NVIDIA investors day 2013

The *Accelerator* model



1. First **COPY** input data from CPU memory to GPU memory
2. Load GPU program and **EXECUTE** (caching data on chip for performance)
3. When results are ready, **COPY** back data from GPU memory to CPU memory

Collision or Convergence?



CPU

?

multi-threading

multi-core

many-core

programmability

Fast-growing eco-system

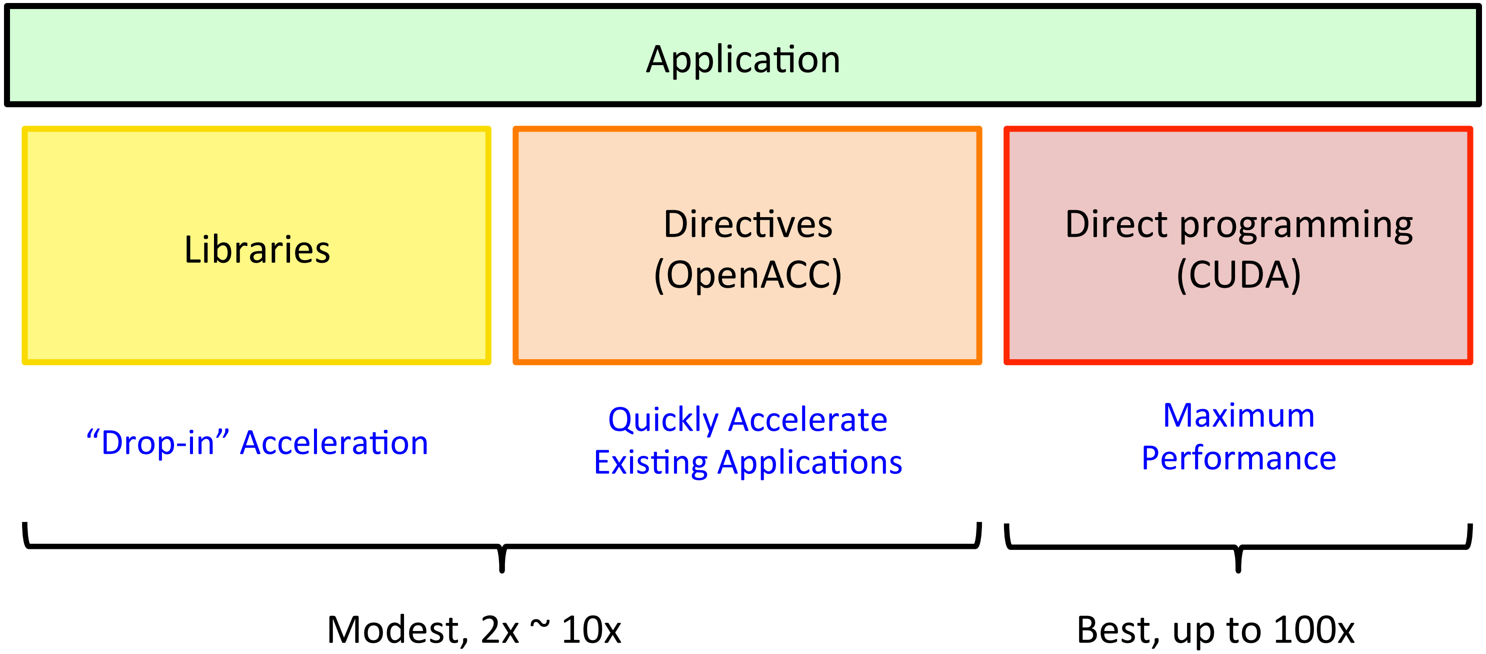
General purpose system

Few limited functions

parallelism

GPU

3 ways to accelerate codes using GPU



Application

Libraries

Directives
(OpenACC)

Direct programming
(CUDA)

“Drop-in” Acceleration

Quickly Accelerate
Existing Applications

Maximum
Performance

Modest, 2x ~ 10x

Best, up to 100x

QUANTUM ESPRESSO

What is Quantum ESPRESSO?

- Quantum ESPRESSO is an integrated suite of computer codes for atomistic simulations based on DFT, pseudo-potentials, and plane waves
- "ESPRESSO" stands for opEn Source Package for Research in Electronic Structure, Simulation, and Optimization
- Quantum ESPRESSO is an initiative of SISSA, EPFL, and ICTP, with many partners in Europe and worldwide
- Quantum ESPRESSO is free software that can be freely downloaded. Everybody is free to use it and welcome to contribute to its development

What can QUANTUM ESPRESSO do?

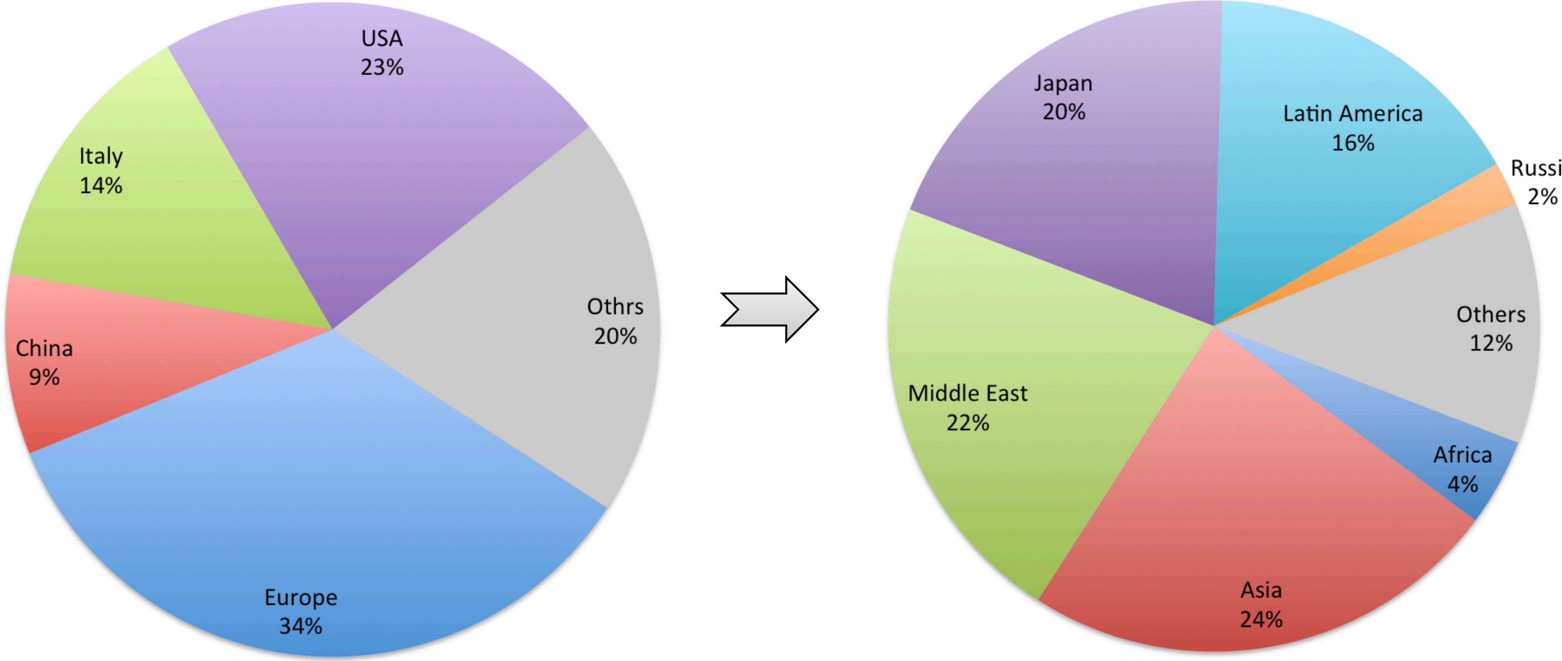
- norm-conserving as well as ultra-soft and PAW pseudo-potentials
- many different energy functionals, including meta-GGA, DFT+U, and hybrids (van der Waals soon to be available)
- scalar-relativistic as well as fully relativistic (spin-orbit) calculations
- magnetic systems, including non-collinear magnetism
- **ground-state calculations**
  - Kohn-Sham orbitals and energies, total energies and atomic forces
  - finite as well as infinite system
  - any crystal structure or supercell
  - insulators and metals (different schemes of BZ integration)
  - structural optimization (many minimization schemes available)
  - transition states and minimum-energy paths (via NEB or string dynamics) electronic polarization via Berry's phase
  - finite electric fields via saw-tooth potential or electric enthalpy
- **Wannier intepolations**
- **ab-initio molecular dynamics**
  - Car-Parrinello (many ensembles and flavors)
  - Born-Oppenheimer (many ensembles and flavors)
  - QM-MM (interface with LAMMPS)
- **linear response and vibrational dynamics**
  - phonon dispersions, real-space interatomic force constants
  - electron-phonon interactions and superconductivity effective charges and dielectric tensors
  - third-order an-harmonicities and phonon lifetimes
  - infrared and (off-resonance) Raman cross sections
  - thermal properties via the quasi-harmonic approximation
- **electronic excited states**
  - TDDFT for very large systems (both real-time and "turbo-Lanczos")
  - MBPT for very large systems (GW, BSE)

plus several post processing tools!

QUANTUM ESPRESSO in numbers

- 350,000+ lines of FORTRAN/C code
- 46 registered developers
- 1600+ registered users
- 5700+ downloads of the latest 5.x.x version
- 2 web-sites (QUANTUM-ESPRESSO.ORG & QE-FORGE.ORG)
- 1 user mailing-list, 1 developer mailing-list
- 24 international schools and training courses (1000+ participants)

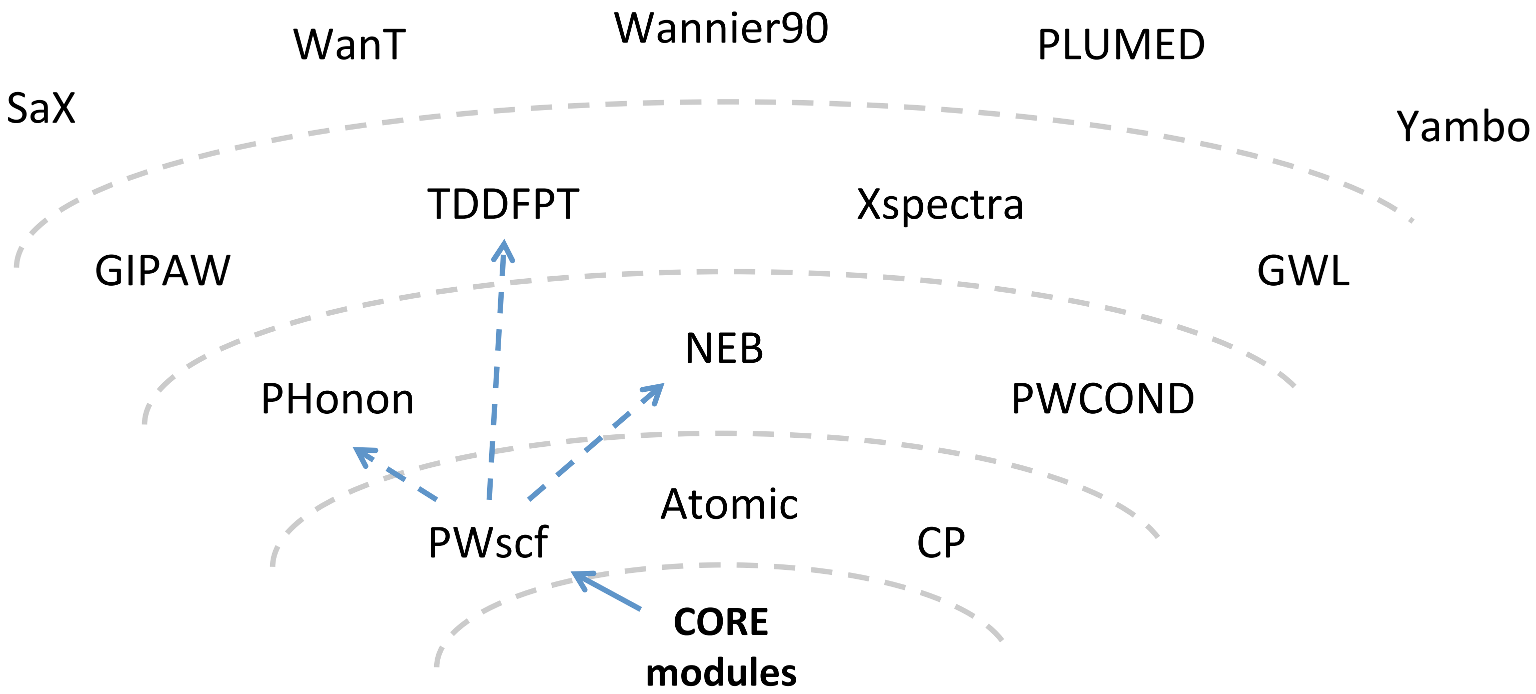
QUANTUM ESPRESSO: a global community



The development model

- QUANTUM ESPRESSO is **not a monolithic application**, but an integrated ecosystem thriving around a small number of core components developed and maintained by a small number of developers
- the ecosystem is designed so as to be alien-friendly: a number of **third-party QE-compatible applications and add-ons**, often designed to be code-agnostic, are distributed with QE (notable examples include wannier90, Yambo, EPW, WanT, XCrysDen, ...)
- the environment that allows the ecosystem to prosper is provided by the QE-FORGE.ORG platform, **freely available** to researchers and developers from all over the world

Quantum ESPRESSO package portfolio



WanT

Wannier90

PLUMED

SaX

Yambo

TDDFPT

Xspectra

GIPAW

GWL

NEB

PHonon

PWCOND

Atomic

PWscf

CP

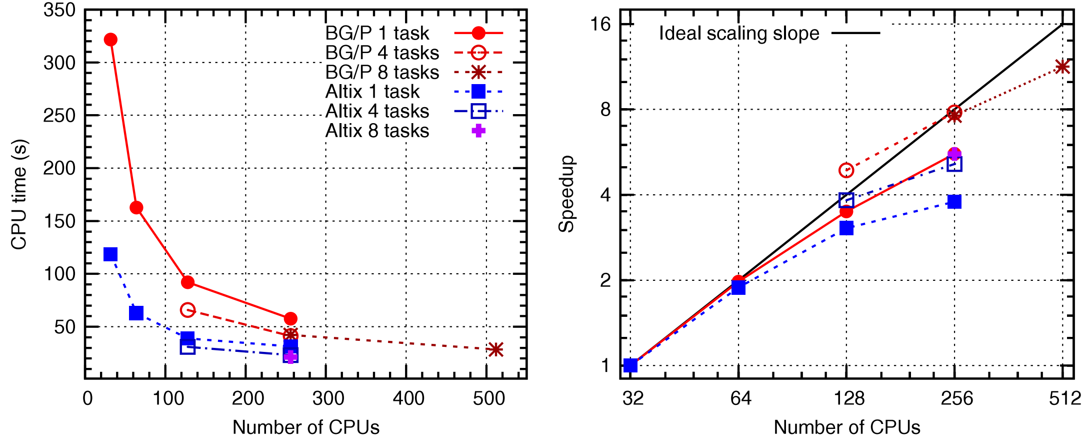
CORE modules

QE Scalability

(2010)

CP

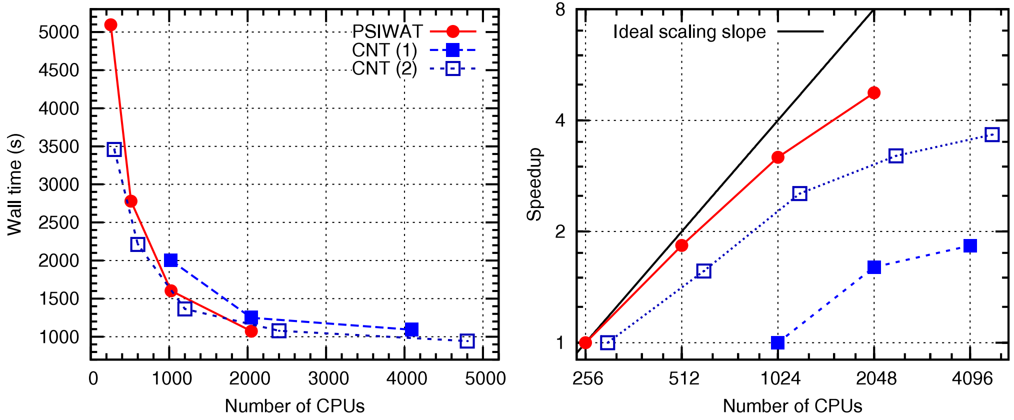
(*A*β-peptide in water: 838 atoms and 2311 electrons, gamma point)



scalability for < 1000 processors

PWscf

(PSIWAT: 587 atoms, 2552 electrons, 4 k-points)
(CNT: 1532 atoms, 5232 electrons, gamma point)



Scalability for > 1000 processors

QUANTUM ESPRESSO FOUNDATION

PWscf GPU PORTING

Plane Wave Self-Consistency Field (PWSCF)

The solution of the Kohn-Sham requires the diagonalization of the matrix H_{KS} whose matrix elements are

$$\langle \mathbf{k}+\mathbf{G}|\mathbf{T}|\mathbf{k}+\mathbf{G}'\rangle = \frac{\boldsymbol{\hbar}^{\mathbf{2}}}{\mathbf{2m}}(\mathbf{k}+\mathbf{G}')^{\mathbf{2}}\delta_{\mathbf{G},\mathbf{G}'}$$

Kinetic energy

$$\langle \mathbf{k}+\mathbf{G}|V_H|\mathbf{k}+\mathbf{G}'\rangle = V_H(\mathbf{G}-\mathbf{G}') = 4\pi e^2 \frac{n(\mathbf{G}-\mathbf{G}')}{|(\mathbf{G}-\mathbf{G}')|^2}$$

Hartree term

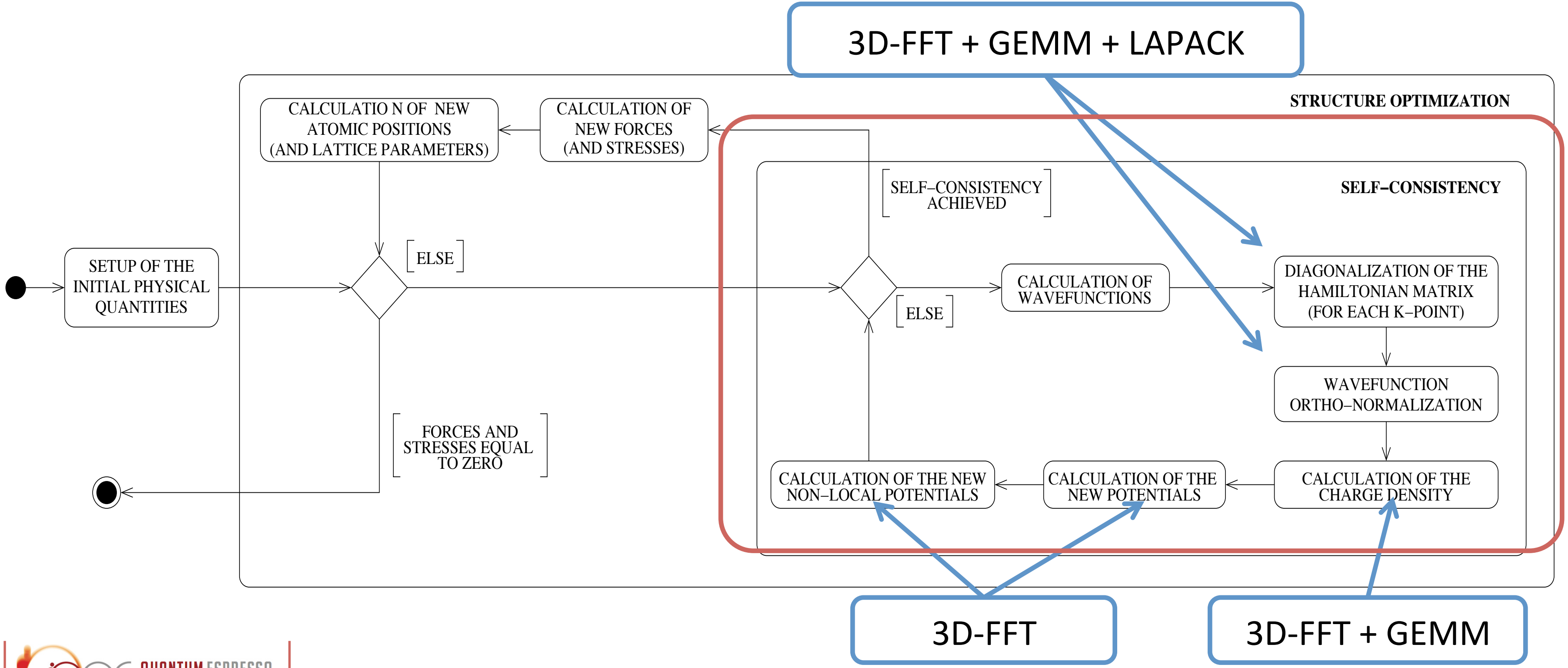
$$\langle \mathbf{k}+\mathbf{G}|V_{xc}|\mathbf{k}+\mathbf{G}'\rangle = FT[V_{xc}(r)]$$

Exchange correlation

PWscf arallelization hierarchy

Images	• Only for Nudged Elastic Band (NEB) calculations
K-points	• Distribution over k-points (if more than one) • Scalability over k-points (if more than one) • No memory scaling
BGRP_COMM	
Plane-waves	• Distribution of wave-function coefficients • Distribution of real-grid points • Good memory scale, good overall scalability, load-balancing
Linear algebra & task groups	• Iterative diagonalization (fully-parallel or serial) • Smart grouping of 3DFFTs to reduce *compulsory* MPI communications
Multi-threaded kernels	• OpenMP handled *explicitly* or *implicitly* • Extend the scaling on multi-core machines with “limited” memory

Simplified PWscf life-cycle


3D-FFT + GEMM + LAPACK
STRUCTURE OPTIMIZATION
CALCULATIO N OF NEW ATOMIC POSITIONS (AND LATTICE PARAMETERS)
CALCULATION OF NEW FORCES (AND STRESSES)
SELF–CONSISTENCY ACHIEVED
SELF–CONSISTENCY
ELSE
SETUP OF THE INITIAL PHYSICAL QUANTITIES
CALCULATION OF WAVEFUNCTIONS
ELSE
DIAGONALIZATION OF THE HAMILTONIAN MATRIX (FOR EACH K–POINT)
WAVEFUNCTION ORTHO–NORMALIZATION
FORCES AND STRESSES EQUAL TO ZERO
CALCULATION OF THE NEW NON–LOCAL POTENTIALS
CALCULATION OF THE NEW POTENTIALS
CALCULATION OF THE CHARGE DENSITY
3D-FFT
3D-FFT + GEMM

QE-GPU and PRACE

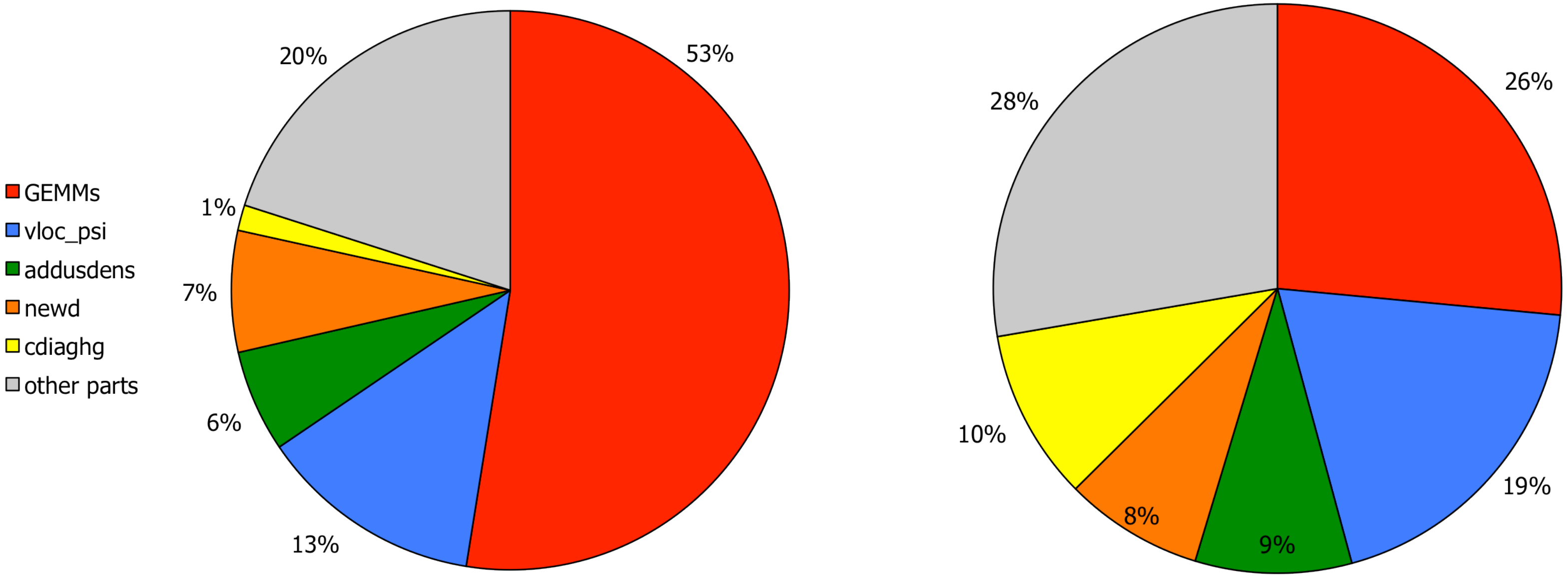
PRACE - *Partnership for Advanced Computing in Europe*

- PRACE Project 1-IP (2010 – 2012)
  - GPU development: PHIGEMM library and PWSCF
  - Key-rule to leverage Quantum ESPRESSO as an EU community code
  - (Better) Parallelization of the GIPAW code (over bands)

- PRACE Project 2-IP (2011 – 2013)
  - Extend the multi-threading support with OpenMP
  - Exploratory of the adoption of OpenACC (GPU with directives)
  - Improvement in the linear algebra and the diagonalization

A "lucky" starting point

AUSURF, 112 Au atoms, 2 k-point



1 SCF iteration

1 full SCF cycle

GPU developments

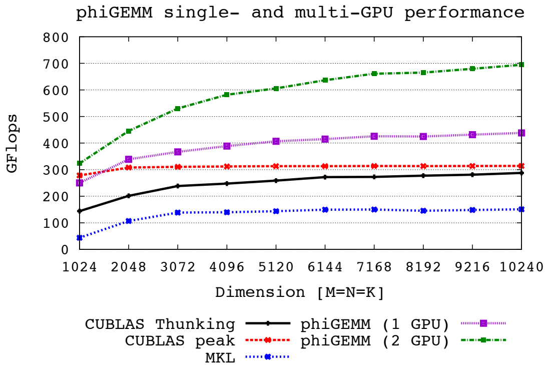
- MPI-GPU binding & GPU memory management
- NEWD → CUDA NEWD (multiple kernels combined)
- ADDUSDENS → CUDA ADDUSDENS
- VLOC_PSI → CUDA VLOC_PSI (CUDA kernels + CUFFT)
- BLAS3 *GEMM → PHIGEMM library (CUBLAS)
- (serial) LAPACK → MAGMA library

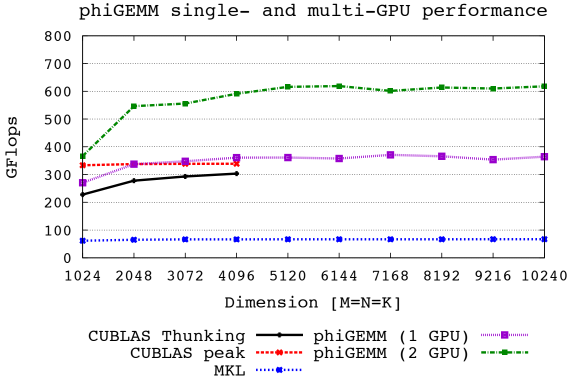
⚠ VLOC_PSI acts over <u>distributed</u> data
NEWD/ADDUSDENS act over <u>local</u> data

phiGEMM

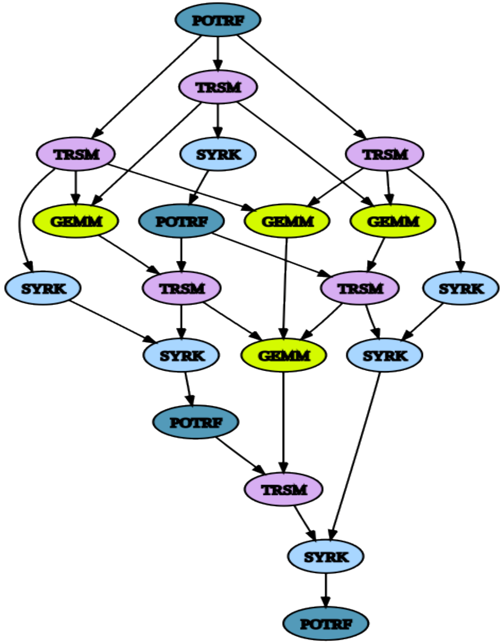
- Inspired by M. Fatica LINPACK work
- Independent open-source library, BSD license
- GPU+CPU BLAS 3 *GEMM routine
- Manual or “semi-automatic“ (SELF-TUNE) split
- Special-K for rectangular matrices
- GEMM→GEMV fallback
- Detailed call-by-call profiling
- Pinned/non-pinned, sync/async
- Support of multi-GPU

web: http://qe-forge.org/projects/phigemm/





MAGMA: LAPACK for GPU



MAGMA uses **HYBRIDIZATION** methodology based on

- Representing linear algebra algorithms as collections of TASKS and DATA DEPENDENCIES among them
- Properly SCHEDULING the tasks' execution over the multicore and the GPU hardware components

What does HYBRIDIZATION means?

- Panels (Level 2 BLAS) are factored on CPU using LAPACK
- Trailing matrix updates (Level 3 BLAS) are done on the GPU using "look-ahead"

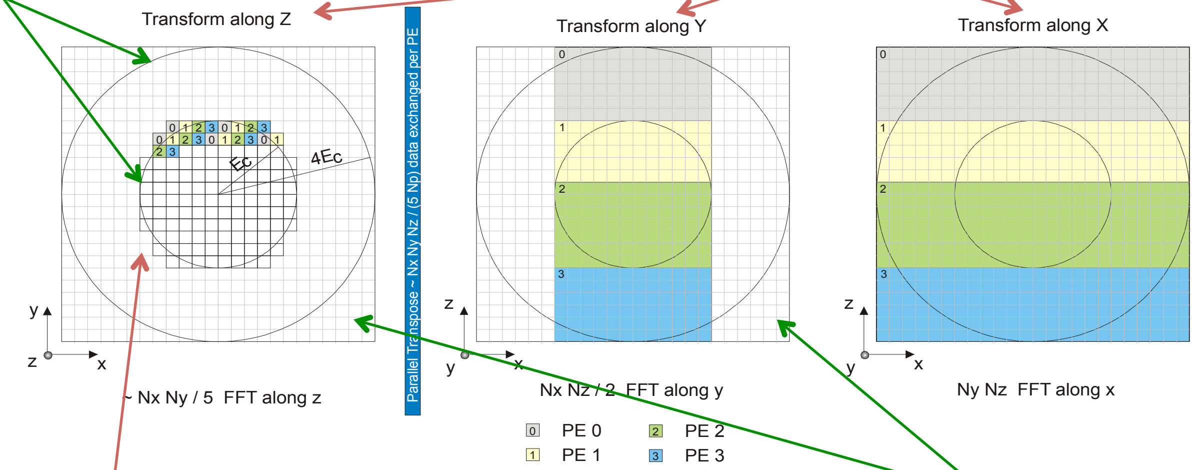
CUDA kernels

- ADDUSDENS
  - compute-bounded kernel
  - Best performance measured: 20x* (Realistic? 9x~10x)
- NEWD
  - compute-bounded kernels
  - Best performance measured: 7.2x* (Realistic? 3x~4x)
- VLOC_PSI
  - memory-bounded kernels
  - Best (serial) performance measured: 9x (Realistic? ...)
- All the data is moved to GPU memory only at once
- External loops over atomic species are kept on the CPU side

The parallel FFT "issue"

There are two "FFT grid" representation in Reciprocal Space: wave functions (E_{cut}) and charge density ($4E_{cut}$)

A single 3D-FFT is divided in independent 1D-FFTs



Data are not contiguous and not “trivially” distributed across processors

Zeros are not transformed. Different cut-offs preserve accuracy

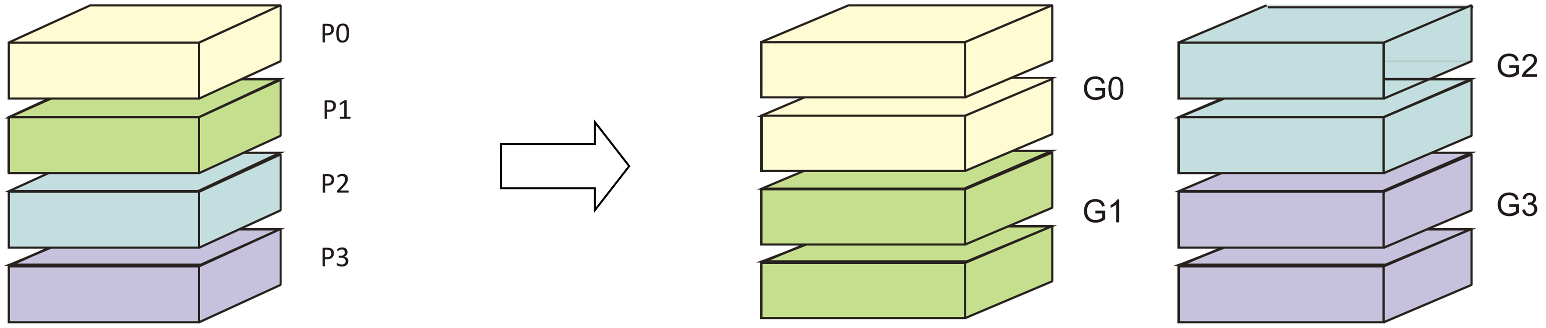
H * psi

```
compute/update H * psi:
    compute kinetic and non-local term (in G space)
        complexity : $N_i \times (N \times N_g + N_g \times N \times N_p)$
    Loop over (not converged) bands:
        FFT (psi) to R space
            complexity : $N_i \times N_b \times FFT(N_r)$
            compute V * psi
             complexity : $N_i \times N_b \times N_r$
            FFT (V * psi) back to G space
             complexity : $N_i \times N_b \times FFT(N_r)$
            compute Vexx:
             complexity : $N_i \times N_c \times N_q \times N_b \times (5 \times N_r + 2 \times FFT(N_r))$
```

$N = 2 \times N_b$ (where N_b = number of valence bands)
N_g = number of G vectors
N_i = number of Davidson iteration

N_p = number of PP projector
N_r = size of the 3D FFT grid
N_q = number of q-point (may be different from N_k)

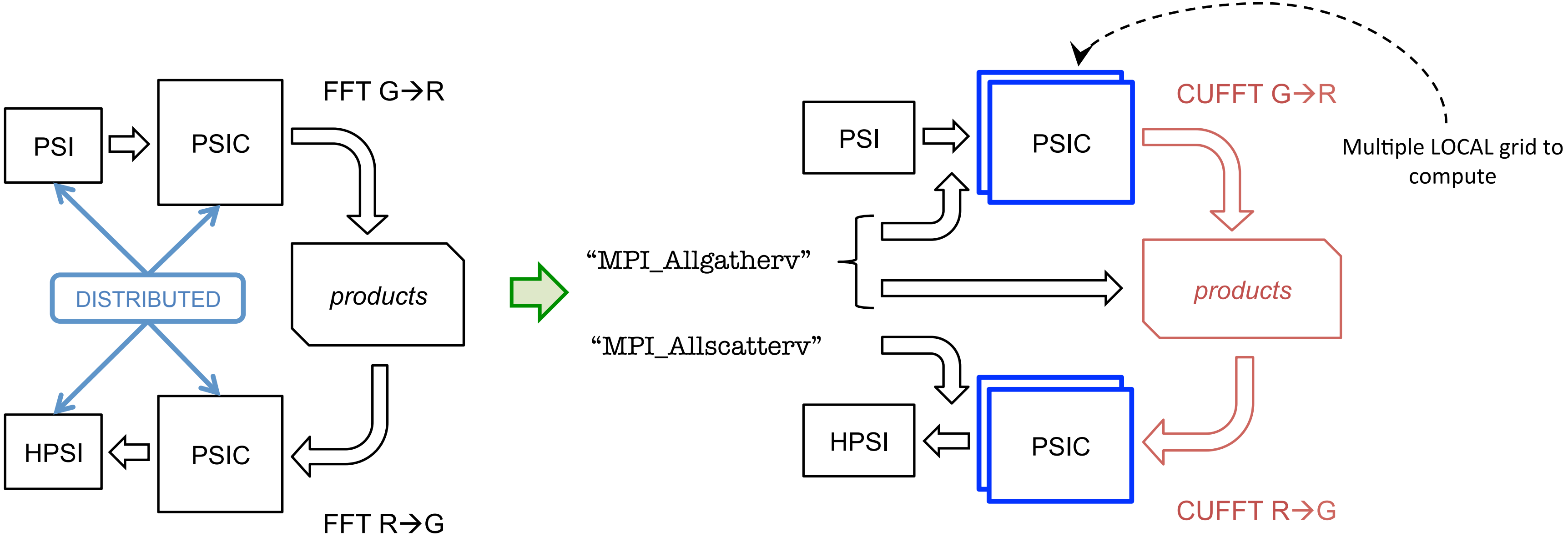
Task-group parallelization



```
do i = 1, n
   compute 3D FFT( psi(i) )
end do
```

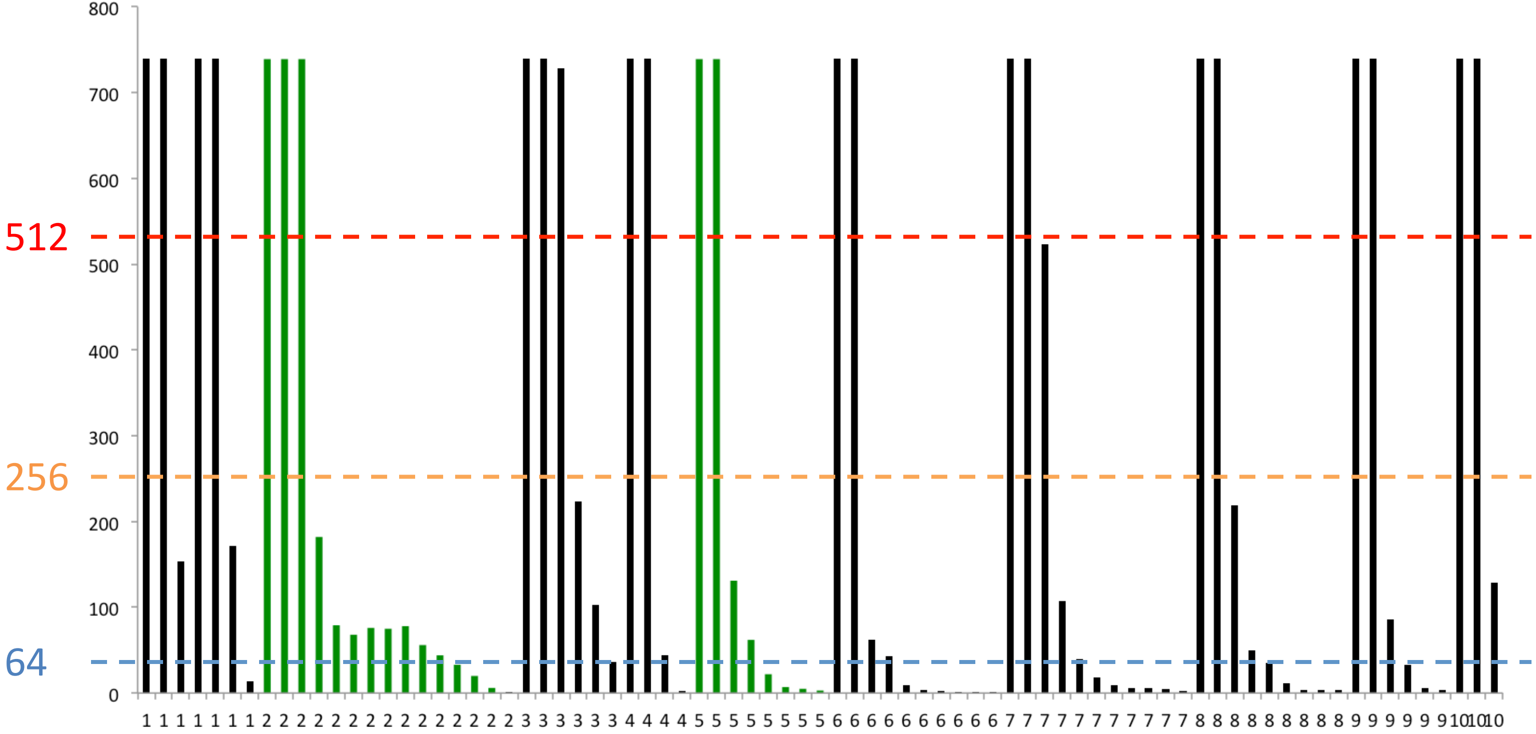
```
do i = 1, n/groupsize
  merge( psi( i ), psi( i + 1 ) )
  compute groupsize 3D FFT
   (at the same time)
end do
```

Parallel VLOC_PSI

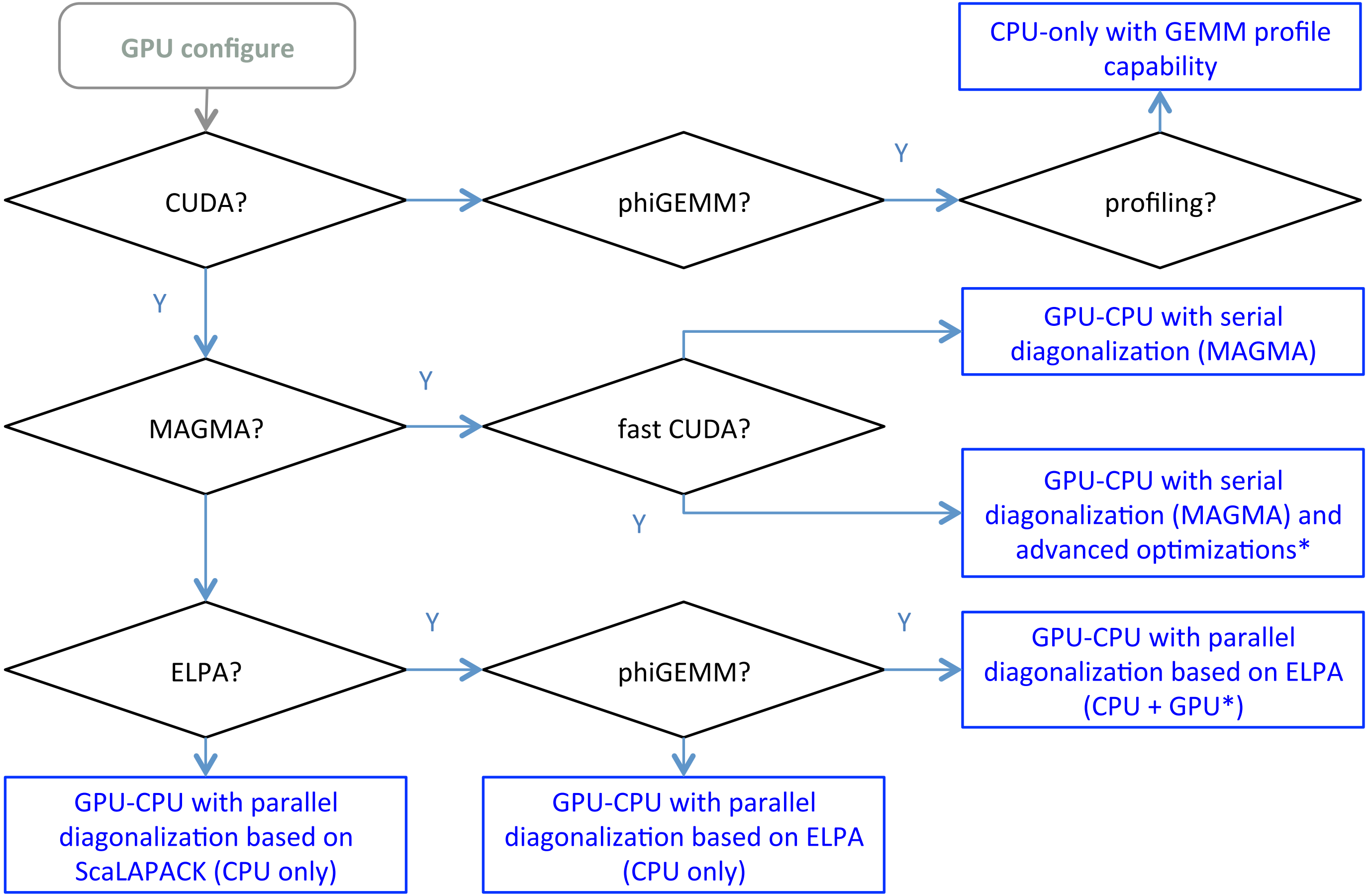


Overlapping is possible!!

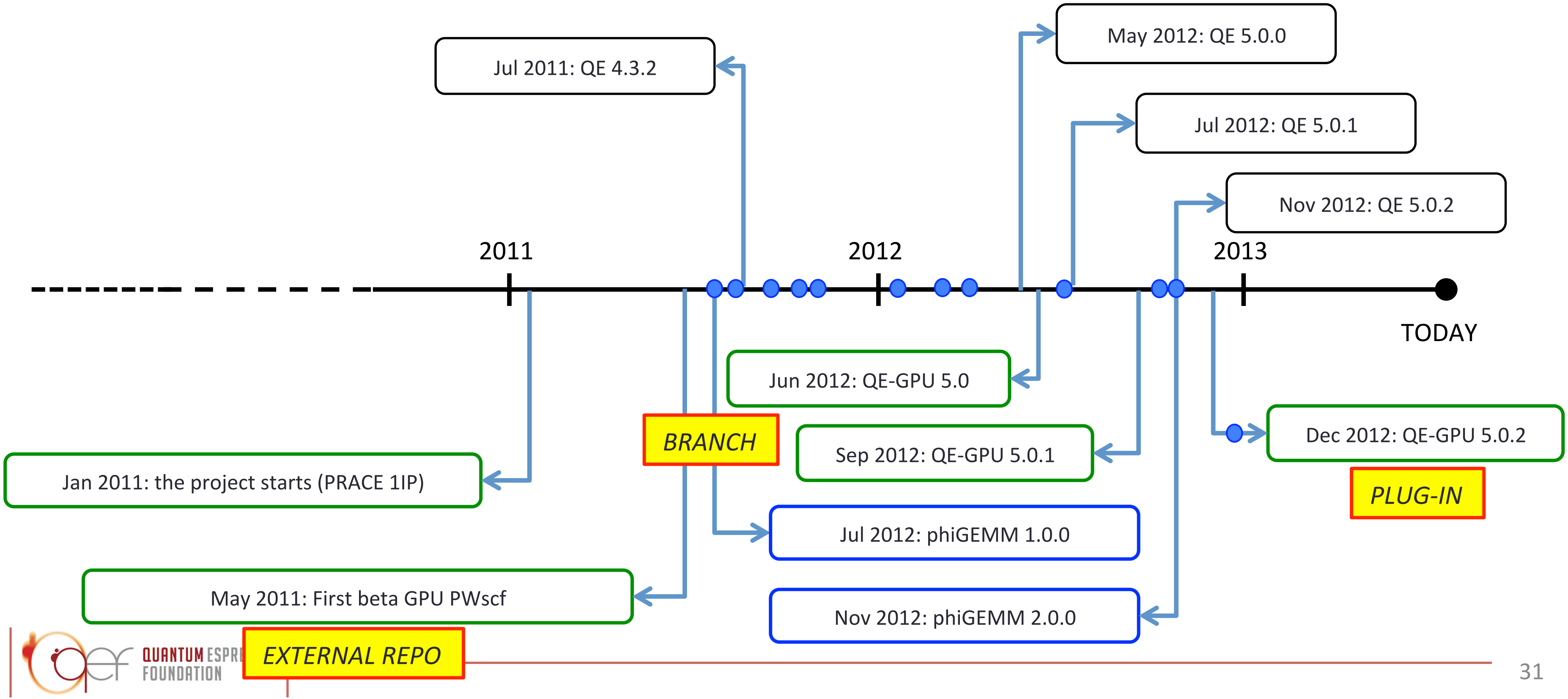
Parallel VLOC_PSI - Limitations



QE-GPU

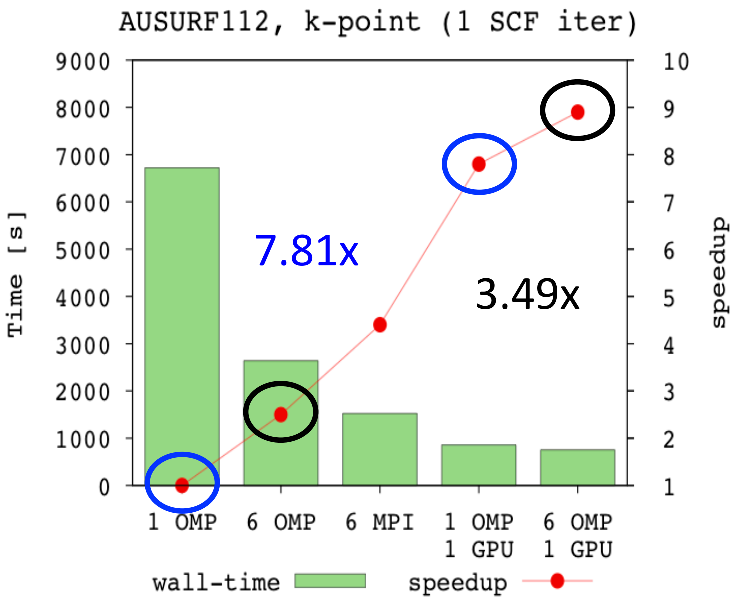

GPU configure
CUDA?
phiGEMM?
Y
profiling?
CPU-only with GEMM profile capability
Y
GPU-CPU with serial diagonalization (MAGMA)
Y
MAGMA?
fast CUDA?
Y
GPU-CPU with serial diagonalization (MAGMA) and advanced optimizations*
Y
ELPA?
phiGEMM?
Y
GPU-CPU with parallel diagonalization based on ELPA (CPU + GPU*)
GPU-CPU with parallel diagonalization based on ScaLAPACK (CPU only)
GPU-CPU with parallel diagonalization based on ELPA (CPU only)

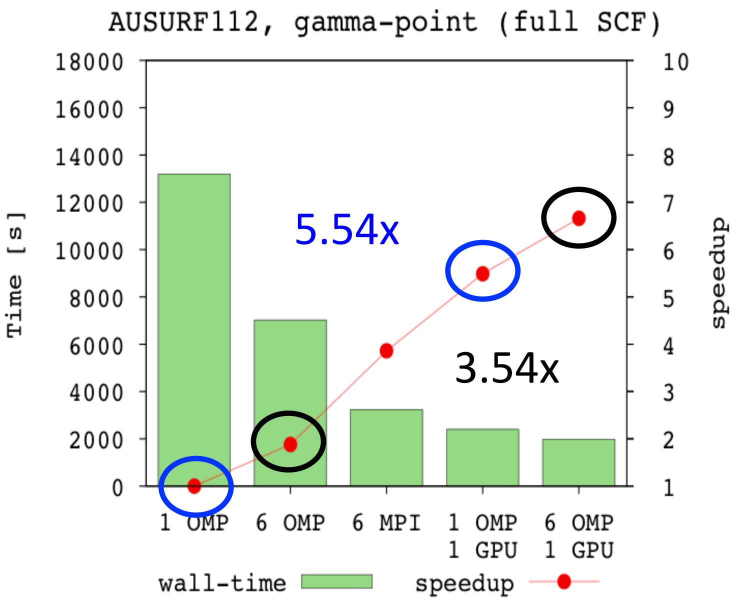
QE-GPU Timeline (~2010 – today)



PERFORMANCE EVALUATION

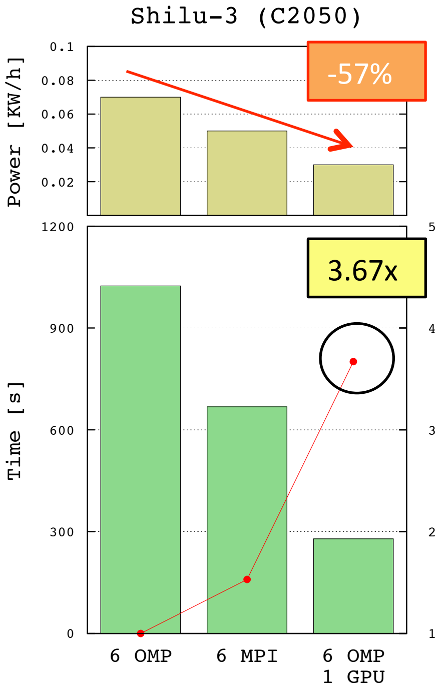
AUSURF112, serial

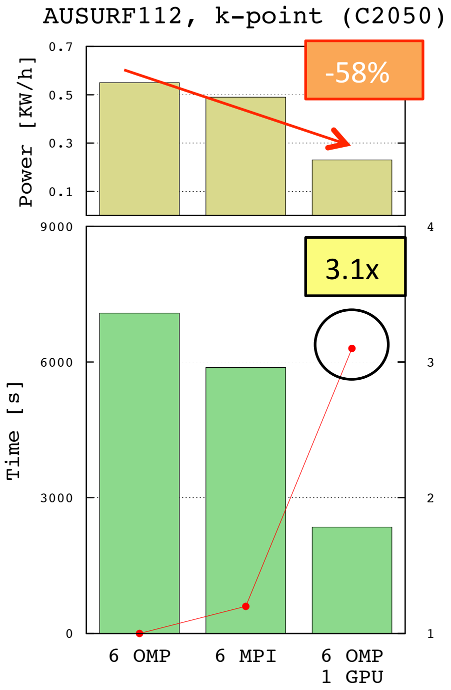


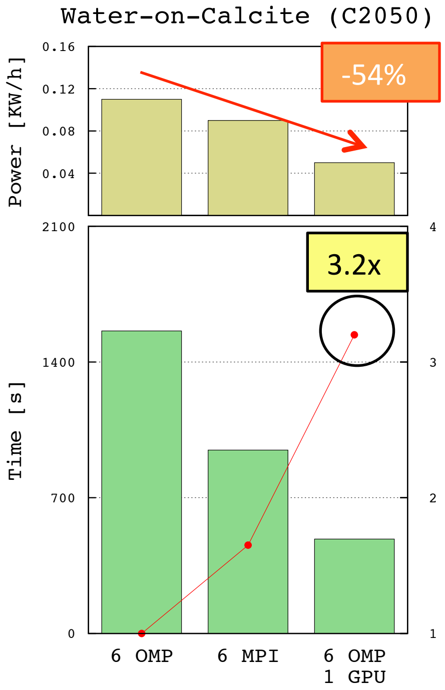


Tests run early 2012

Performance & Power consumption (serial)


Shilu-3 (C2050)

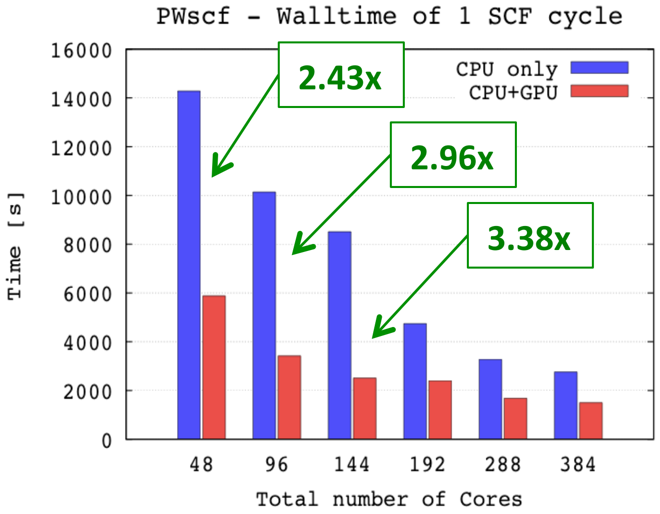

AUSURF112, k-point (C2050)

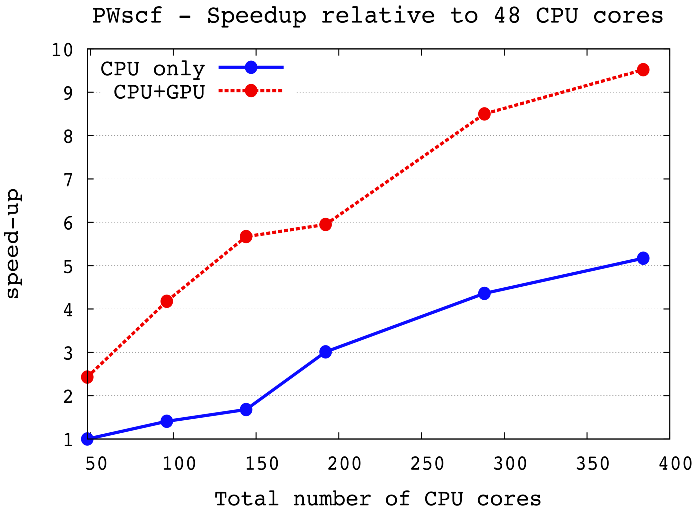

Water-on-Calcite (C2050)

Tests run early 2012

MGST-hex

216 atoms of {Ge, Mn, Te, Sb}, gamma-only (courtesy of *Zhang W.* – RWTH/AACHEN)

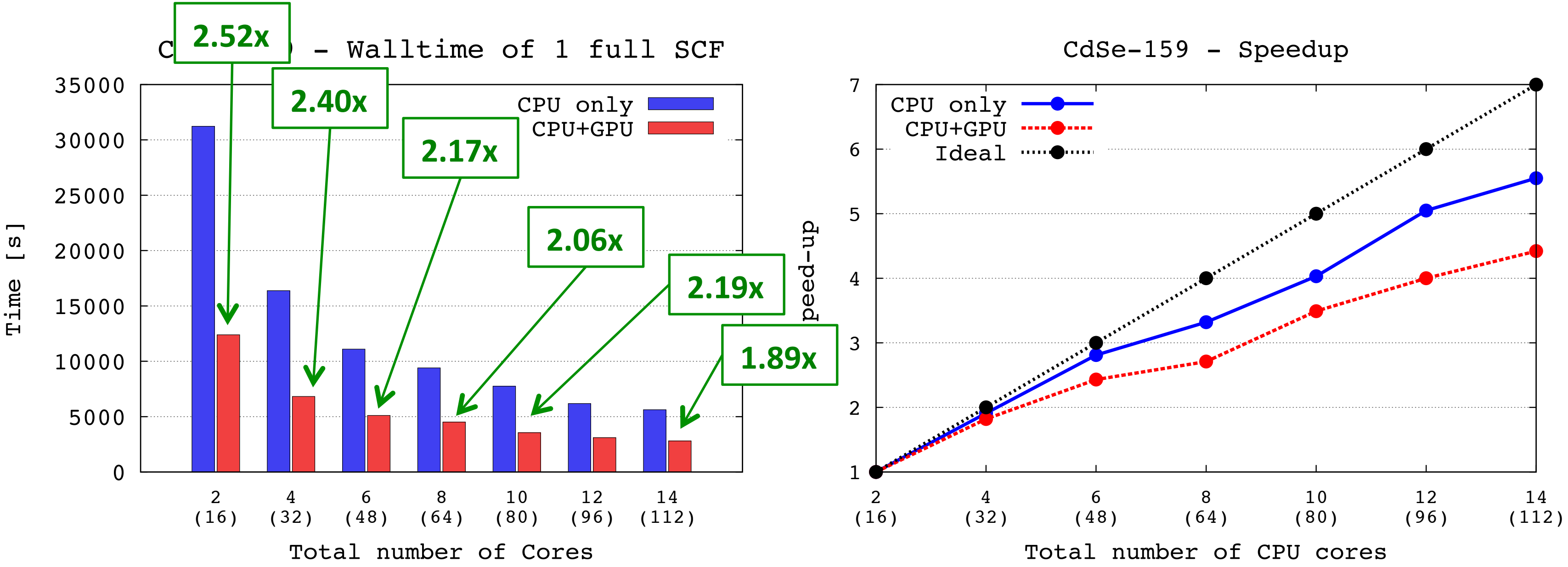




Tests run early 2012

CdSe159

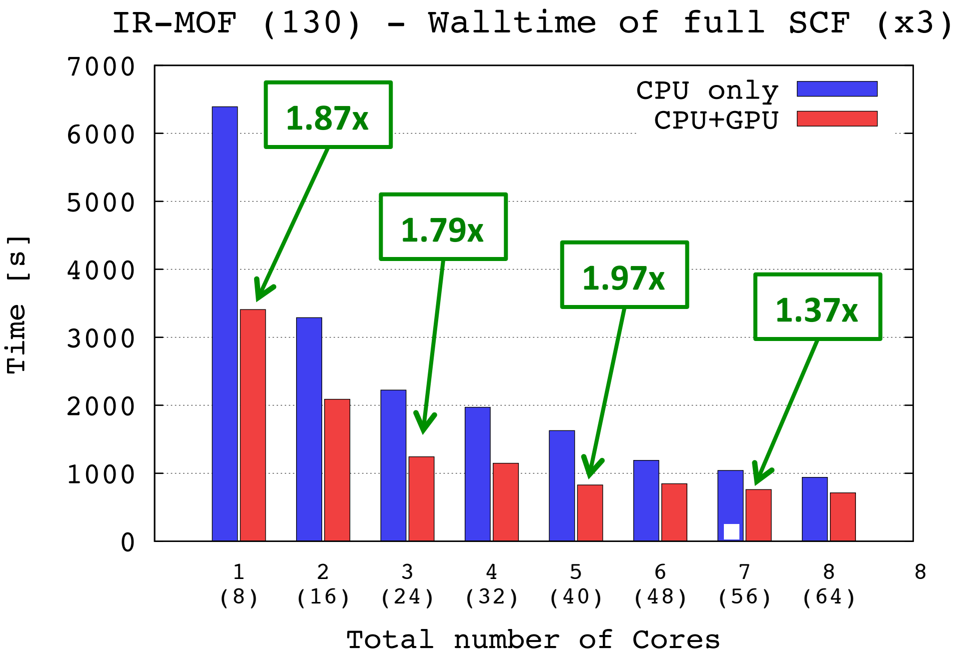
159 atoms of {Cd, Se}, gamma-only (courtesy of *Calzolari A.* – CNR/NANO)

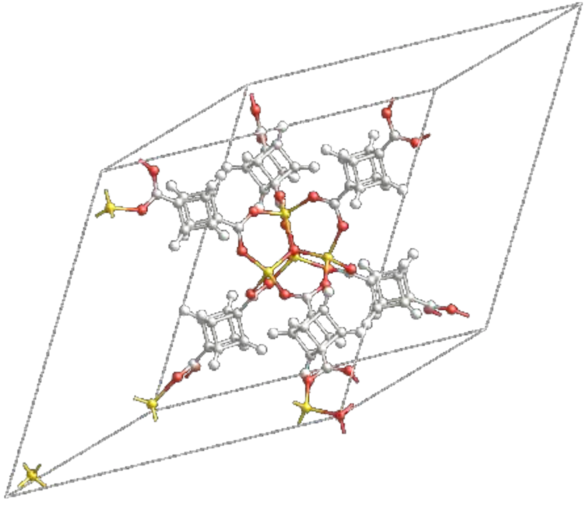


Tests run early 2012

IRMOF-M11

130 atoms of {O, Zn, C, H}, 1 K-point (courtesy of *Clima S.* - IMEC)

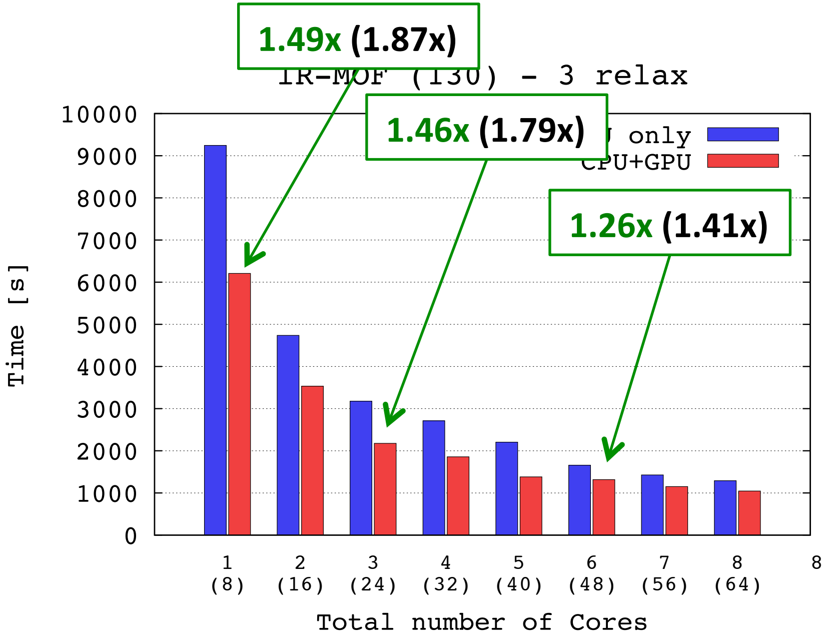




Tests run early 2012

Hitting the limit

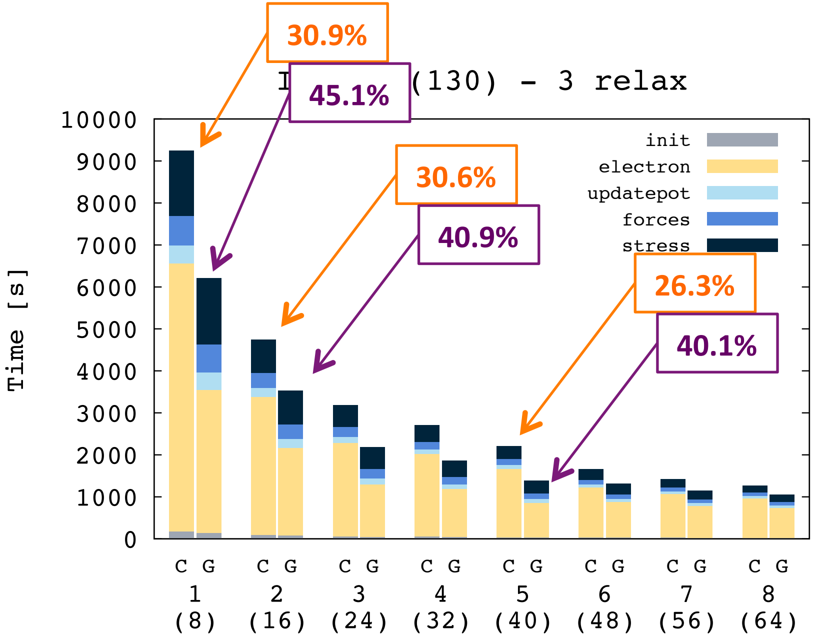
130 atoms of {O, Zn, C, H}, 1 K-point (courtesy of *Clima S.* - IMEC)



1.49x (1.87x)

1.46x (1.79x)

1.26x (1.41x)



30.9%

45.1%

30.6%

40.9%

26.3%

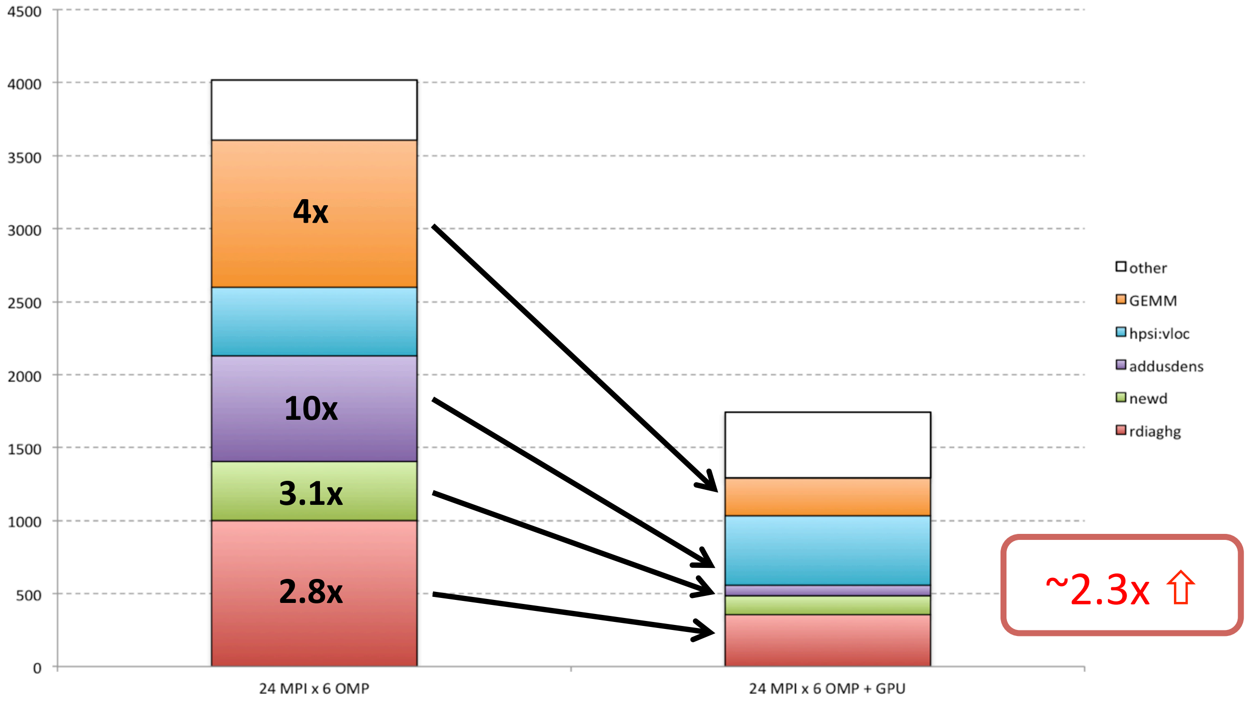
40.1%

Tests run early 2012

QUANTUM ESPRESSO FOUNDATION

CdSe489 - PRACE Preparatory Access

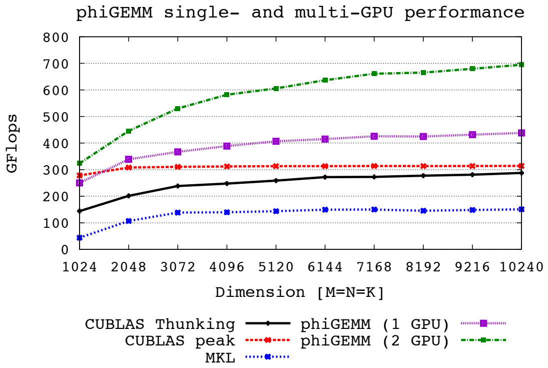
489 atoms of {Cd, Se} , 3 SCF steps, 24 MPI x 6 OMP (144 cores) @ PLX (CINECA)

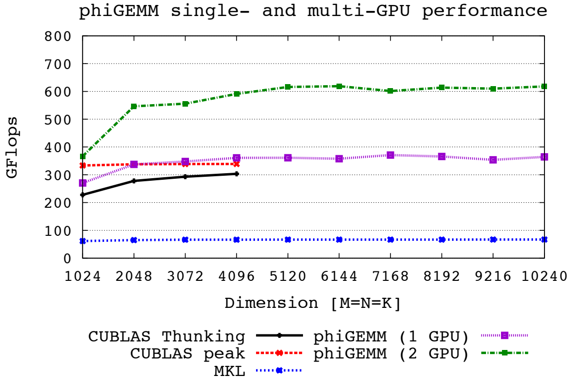


phiGEMM

- Inspired by M. Fatica LINPACK work
- Independent open-source library, BSD license
- GPU+CPU BLAS 3 *GEMM routine
- Manual or "semi-automatic" (SELF-TUNE) split
- Special-K for rectangular matrices
- GEMM→GEMV fallback
- Detailed call-by-call profiling
- Pinned/non-pinned, sync/async
- Support of multi-GPU

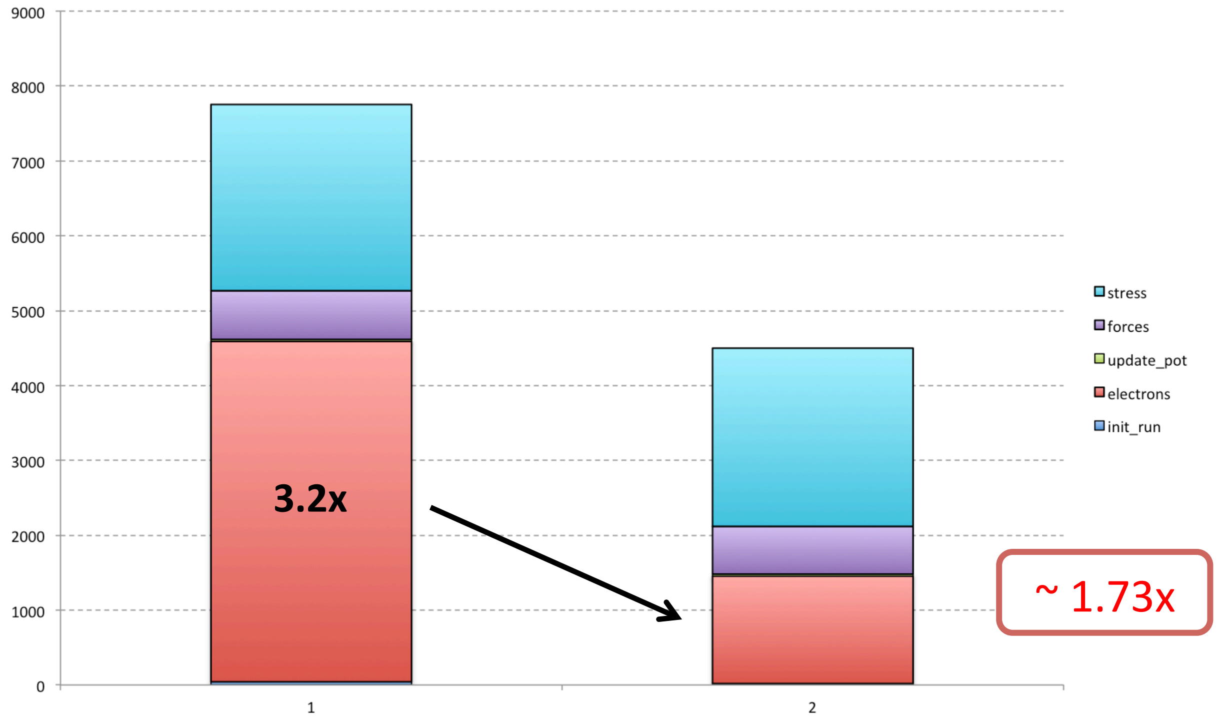
web: http://qe-forge.org/projects/phigemm/





Hitting the limit

489 atoms of {Cd, Se}, 3(+1/2) SCF steps & STRESS & FORCES, 24 MPI x 6 OMP (144 cores) @ PLX (CINECA)



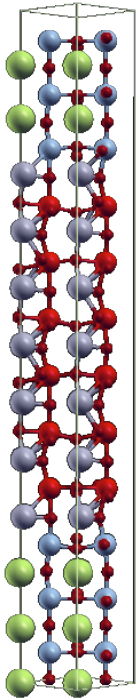
Tests run early 2012

Cheating using parallelism: USE_3D_FFT

120 atoms of {Bi, Fe, O, LS, Mn}, reduced to 8 k-point (courtesy of *Ferreira R.* – Rio de Janeiro Federal Univ.)

# pool = # MPI processes
(dimension each pool = 1, k-point distributed in "round-robin")

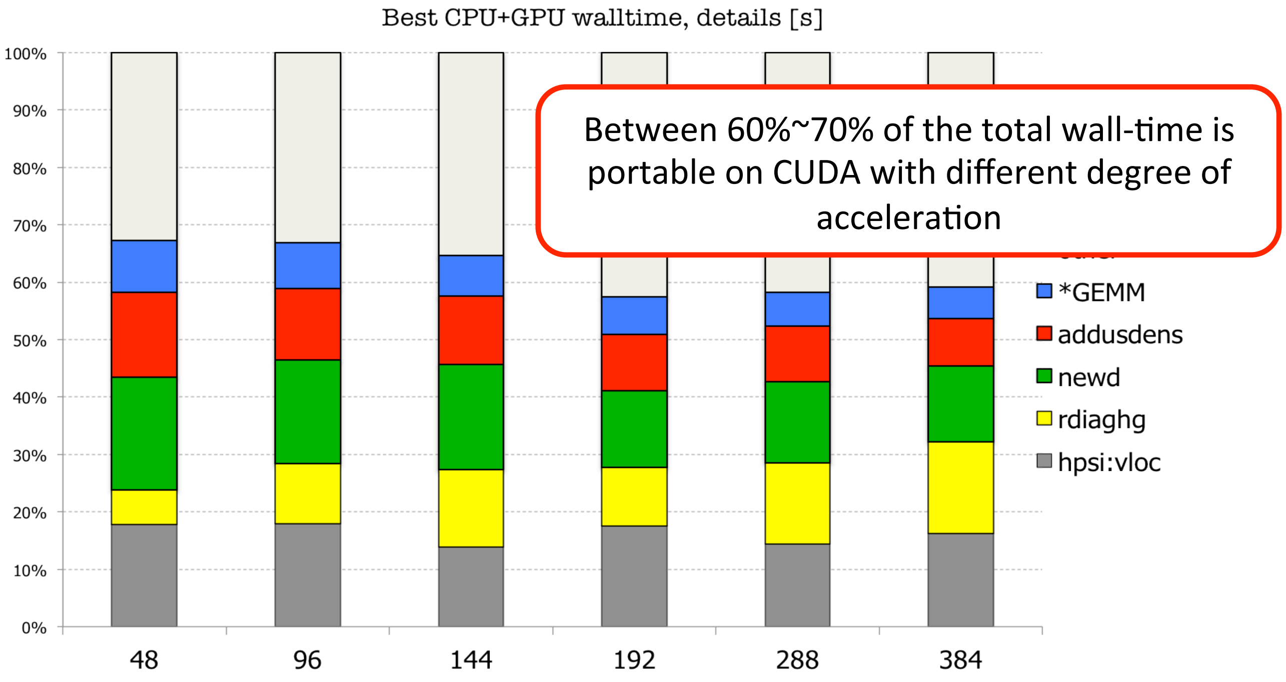
Computer Nodes	Execution Time [s] #10 self-consistency cycles	Speed-up
2 x iDataPlex DX360M3, dual Xeon E5645 6-cores 2.40 GHz (24 cores)	52057.22	
2 x iDataPlex DX360M3, dual Xeon E5645 6-cores 2.40 GHz (24 cores) + **4 NVIDIA 2070 (__USE_3D_FFT)**	**10029.1**	**5.2x**



Tests run early 2012

We are all limited by Amdahl

(parallel – MGST-hex)

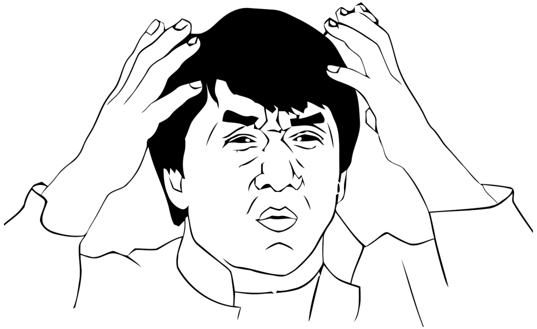

Best CPU+GPU walltime, details [s]

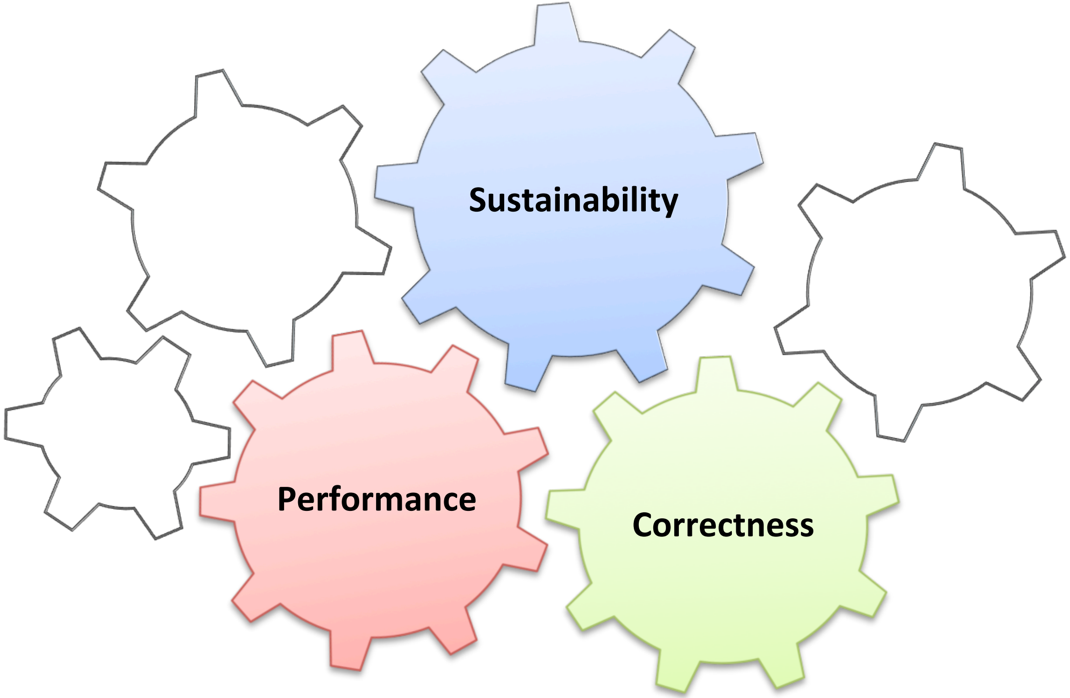
Between 60%~70% of the total wall-time is portable on CUDA with different degree of acceleration

FINAL CONSIDERATIONS

PERFORMANCE, CORRECTNESS AND SUSTAINABILITY

What drive (my) agenda?





Correctness vs Sustainability

- Bugs can be very tricky and not easy to detect
  - long runs → lot of output; big runs → lot of resources
  - time & reproducibility
- An embedded Unit Test suite is a solution but ...
- ... an Acceptance Testing Procedure is required too!
- Old-fashion debugging works but ...
  - dirty source code
  - plenty of preprocessor macro

Performance vs Correctness

- GPU are not CPU (*what a surprise!*)
- Third-part libraries hide complexity but also introduce "noise"
- Reduction operations (explicit managed or implicitly inherit) are the nightmare
- Innovative algorithms can improve performances but who does a proper validation?

Sustainability vs Performance

- CUDA or not-CUDA? (*Shakespearean dilemma*)
- Benchmarking requires time (*again, what a surprise!*)
- The need of certifying QE-GPU benchmarks*
- Guarantee performances is very difficult because...
  - users expect/pretend performance portability
  - too many HW combinations

OpenACC: yes, no, maybe?

PROs:

- easy learning curve
- fast first deployment
- acceptable performance compromises

CONs:

- Code has to be rewritten to suite the accelerator
- Lack of direct control of the generated GPU code
- memory sharing and competition

and now OpenMP 4.0 RC2 with *Directives for attached accelerators*

(pseudo-random) Personal thoughts

- Training users is challenging (we kept everything simple but...)
- User support is time consuming
- CUDA is easier than most people think
- Focusing on fine-grain performance ONLY at the final stage
- There is not only the parallel world
- No more old-fashion porting

The *Optimization Dilemma*

?

Is it worth to spend time/effort optimizing the code for the latest available architecture by using the latest cool set of features provided by the most updated version of CUDA toolkit/driver?

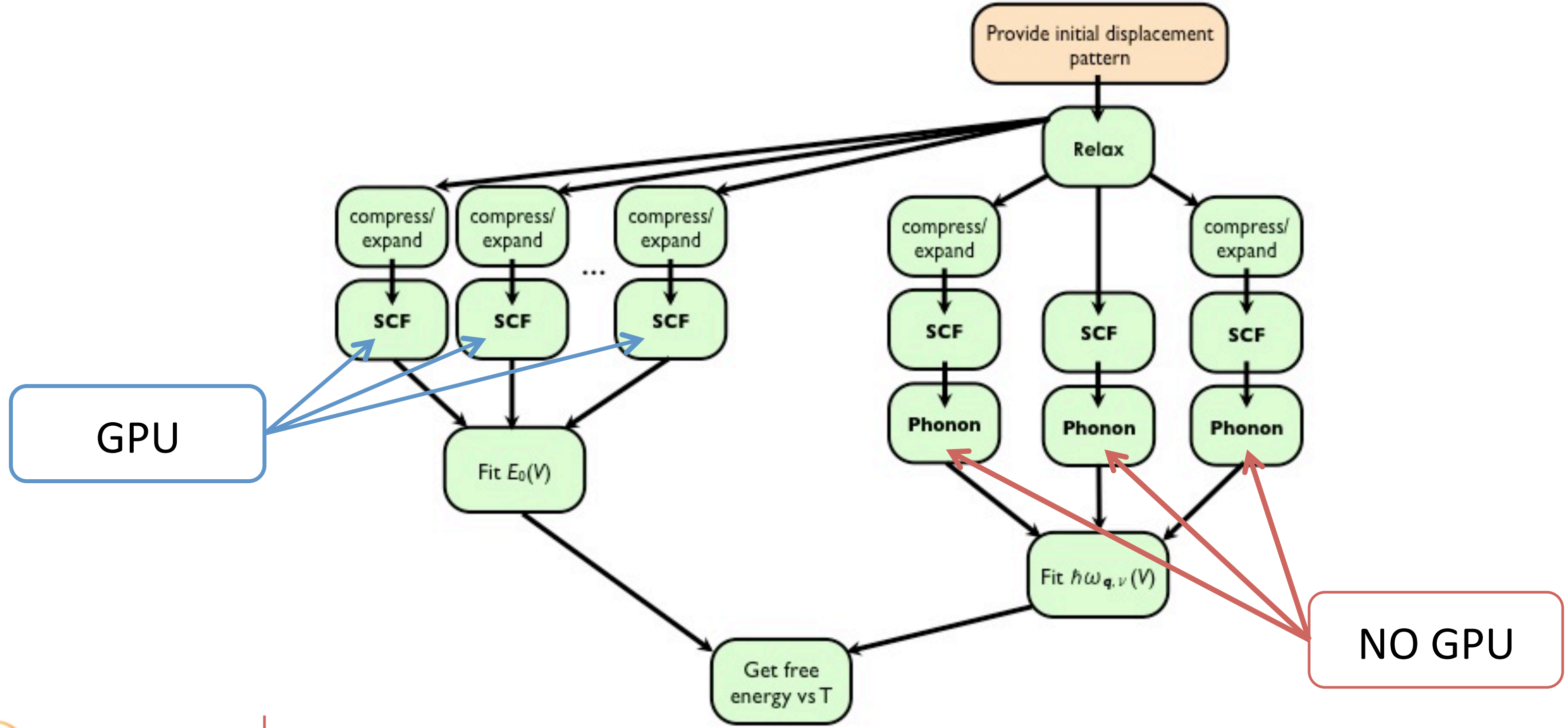
Who are the users? Are they users or developer? What hardware they have? What is their scientific workflow? What are the scientific challenges they have? What is their budget? What ROI they expect?

GPU-accelerated PWscf, when and why?

- Physical system criteria
  - "crazy" amount of atoms ($\mathcal{O}(1000)$) in gamma-only (or 1~2 k-point)
  - reasonable amount of atoms ($\mathcal{O}(100)$)
  - small amount of atoms ($\mathcal{O}(10)$) but many k-points
  - $\mathcal{O}(10)$~$\mathcal{O}(100)$ but with lot of atomic species
- Hardware criteria*
  - workstations equipped with commodity GTX and/or high-end TESLA GPUs
  - reasonable HPC clusters ($\mathcal{O}(100)$ nodes, 1:1 ration GPU versus CPU socket)
- Work-flow criteria
  - long SCF energy minimization
  - high throughput

What scientists do?

Courtesy of Prof. *Nicola Marzari* (THEOS, EPFL)



Next developments/priorities

TOP PRIORITY:

- Specific NVIDIA Kepler optimizations → NVIDIA Dev Tech
- non collinear & spin-orbital magnetization
- PHonon using OpenACC

"LESS" PRIORITY:

- first GPU-accelerated CP (special focus on DFORCE)
- Alternatives for current eigen-solvers (ELPA+GPU? Lancroz?)

and... DOCUMENTATION!

CASTEP and GPU

CASTEP and QE shares...

- similar data distribution (by plane waves, by k-point, by bands, ...)
- similar MPI communication patterns (MPI_alltoall, MPI_reduce)
- similar constrains in scalability

Where GPU...

- *Block Davidson solver with density mixing* → eigen-solvers, vloc_psi
- *Support of ultrasoft/norm-conserving pseudopotential* → newd & addusdens
- *Geometry optimization* → stress & forces calculations
- BLAS 3 operations → CUBLAS, phiGEMM

THANK YOU FOR YOUR ATTENTION!

Links:

- http://www.quantum-espresso.org/
- http://foundation.quantum-espresso.org/
- http://qe-forge.org/gf/project/q-e/
- http://qe-forge.org/gf/project/q-e-gpu/

UNIVERSITY OF CAMBRIDGE

hpcs Cambridge HPC service