ICHEC
Irish Centre for High-End Computing

Enabling faster material science modeling using the accelerated Quantum ESPRESSO

Filippo Spiga (ICHEC)
Ivan Girotto (ICHEC)
Carlo Cavazzoni (CINECA)

sfi Science Foundation Ireland

Ireland's EU Structural Funds Programmes 2007 - 2013
Co-funded by the Irish Government and the European Union

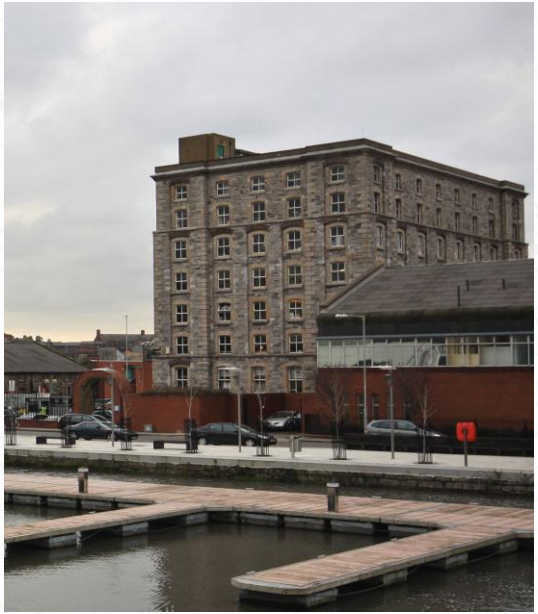
EUROPEAN REGIONAL DEVELOPMENT FUND

AN ROINN OIDEACHAIS AGUS EOLAÍOCHTA | DEPARTMENT OF EDUCATION AND SCIENCE

HEA
Higher Education Authority
An tÚdarás um Ard-Oideachas

Irish Centre for High-End Computing (ICHEC)

- Founded in mid-2005 by SFI and HEA
- ~25 staff member
- Two main systems:
  - SGI Altix ICE 8200EX (~4k cores)
  - Bull Novascale R422-E2, ~ 500 cores + 48 NVIDIA GPUs, National Service Production
- Objectives:
  - Provide computational resources
  - Provide education and training to third-level institutions
  - Tech transfer and consultancy services to develop Irish smart economy



Outline

- What is Quantum ESPRESSO
- Project Context, Objectives and Goals
- Implementation strategies
- Performance & Power Measurements
- Current Limitations & Best Practices
- Partnership & (Big) Challenges
- Future developments
- Q/A

What is QUANTUM ESPRESSO?

QUANTUM ESPRESSO is…

- an integrated suite of computer codes for electronic-structure calculations and materials modeling at nano-scale
- based on density-functional theory (DFT), plane waves (PW) basis set and pseudo-potentials (PP)
- a DEMOCRITOS initiative, later joined by ICTP, CINECA Bologna, EPF Lausanne, Princeton University, MIT, Paris VI, Oxford, IJS Ljubljana, **ICHEC**, et al.
- composed of many packages: **PWSCF**, CP, PHONON, ATOMIC, PWCOND, XSPECTRA, GIPAW, GWL, TDDFPT, WANT, …
- able to run in serial and parallel on several architectures (Linux clusters, IBM systems, CRAY, NEC)
- distributed under the GNU General Public License (GPL)
- supported by a worldwide community of developers and users

QUANTUM ESPRESSO in numbers…

- ~300,000 FORTRAN90/C lines of code (in total, including examples and docs, ~500,000)
- ~300 citations at 2010Q1 (>1,000 today)
- ~1,300 subscribers to the mailing-lists
- ~4,500 mails/year (2010/2011 traffic)
  - *new mailing list dedicated to the GPU PWscf, 56 member registered*
- 4,000~4,500 downloads from the website per version (SVN? who knows…)
- ~40 active projects on QE-forge (including PHIGEMM and QE-GPU!)
- 252 QE-forge users
- 115,633 QE-forge pages seen in 2010
- 20 international schools/events all over the world

(trends evolve very fast… the number always increasing!)

Project Context, Objectives and Goals

Who:

- the **Irish Centre for High-End Computing** within EC-funded *PRACE - Partnership For Advanced Computing in Europe, 1st implementation phase* project (FP7/2007-2013 under grant RI-261557) and *SFI - Science Foundation Ireland* (grant 08/HEC/I1450).
- CINECA & DEMOCRITOS, as technical and consultancy partners

What:

- accelerate the Plane Wave Self-Consistency Field (PWSCF) code exploiting the NVIDIA GPU capabilities
- target both serial and parallel version, assessing the numerical accuracy and the overall performance
- GPU code maintenance, support to the growing community and package dissemination

Why:

- PWSCF is one of the most used package of the Quantum ESPRESSO suite. SCF calculations represent the starting point of other type of calculations (PHONON, GIPAW, GWL,…)
- Quantum ESPRESSO is recognized at European level as *community code* and PWSCF is part of the PRACE Official Benchmark suite

Preliminaries: Kohn-Sham with plane waves

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

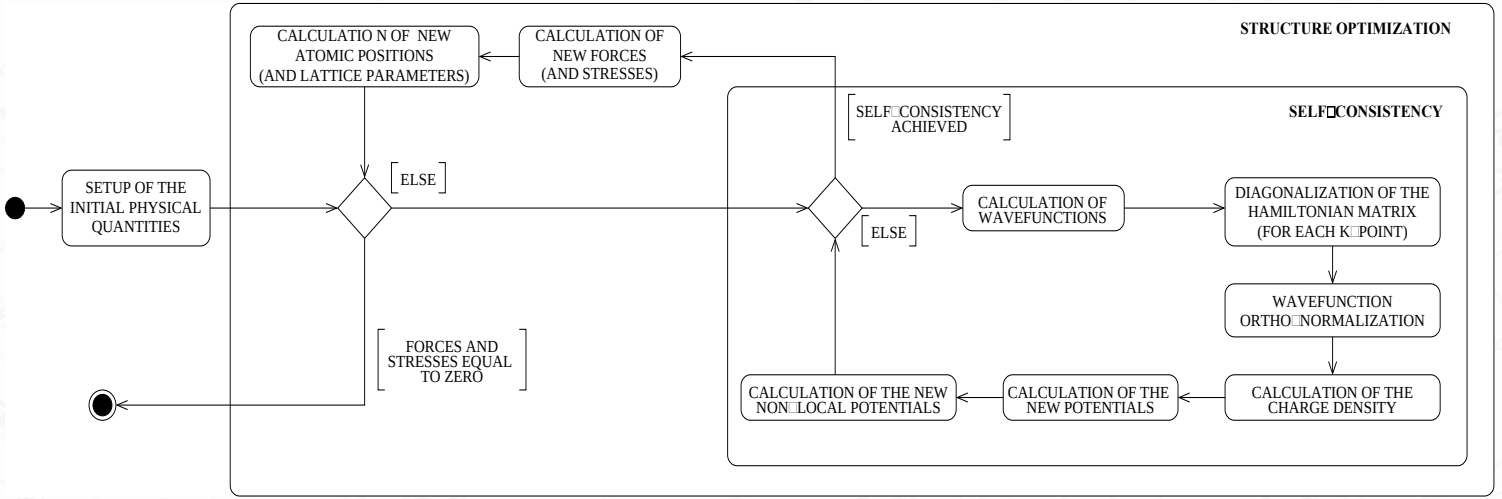
Kinetic energy $$\left\langle \mathbf{k}+\mathbf{G}\left|\mathrm{T}\right|\mathbf{k}+\mathbf{G}\text{¢}\right\rangle =\frac{\hbar^2}{2\mathrm{m}}\left(\mathbf{k}+\mathbf{G}\right)^2 d_{\mathbf{G},\mathbf{G}\text{¢}}$$

Hartree term $$\left\langle \mathbf{k}+\mathbf{G}\left|V_H\right|\mathbf{k}+\mathbf{G}\text{¢}\right\rangle =V_H\left(\mathbf{G}-\mathbf{G}\text{¢}\right)=4pe^2\frac{n\left(\mathbf{G}-\mathbf{G}\text{¢}\right)}{\left|\left(\mathbf{G}-\mathbf{G}\text{¢}\right)\right|^2}$$

Exchange correlation $$\left\langle \mathbf{k}+\mathbf{G}\left|V_{xc}\right|\mathbf{k}+\mathbf{G}\text{¢}\right\rangle =FT\left[V_{xc}\left(\mathbf{r}\right)\right]$$

FFT is used (or "abused") to move from real to reciprocal space (and vice-versa) in order to "simplify" some operations

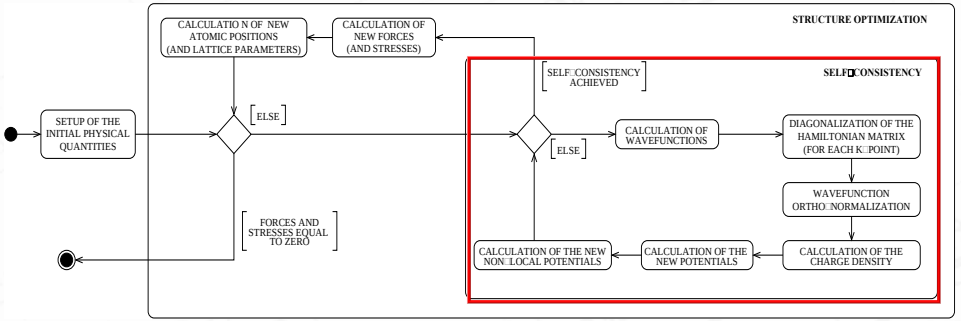
Simplified PWscf life-cycle



A REAL simulation for scientific purpose is usually composed by several SCF steps in a global structure optimization loop…

Time-consuming steps in PWscf

- Calculation of Charge Density
  - FFT (full 3D or set of 1D)
  - matrix-matrix multiplications
- Calculation of Potential
  - FFT (full 3D or set of 1D)
  - operations on real-space grid
- Davidson Iterative Diagonalization (SCF)
  - eigenvalues/eigenvectors
  - FFT (full 3D or set of 1D)
  - matrix-matrix multiplications

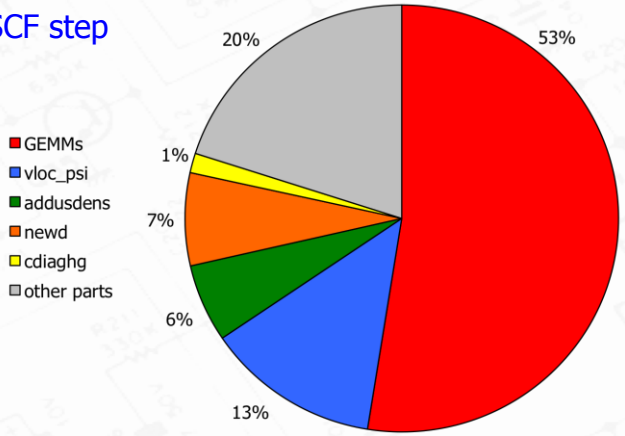


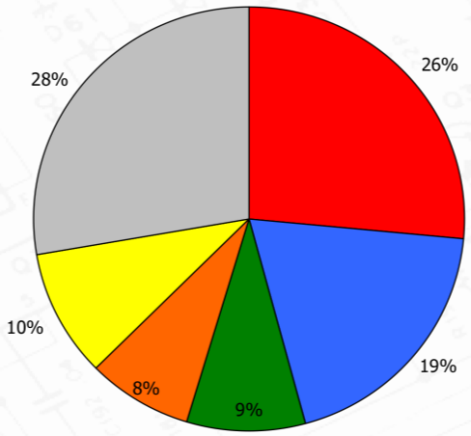
Basically most CPU time spent in linear-algebra operations, it is implemented among BLAS, LAPACK and FFT libraries!

Starting point: AUSURF112

Physical description: gold slab surface of 112 Au atoms
Technical description: both gamma- and k-point version, can run in serial and parallel. No high scalable.
Reference: subset of a big test case (PSIWAT), provided by Arrigo Calzolari (CNR-NANO)

k-point, 1 SCF step





k-point, FULL SCF loop

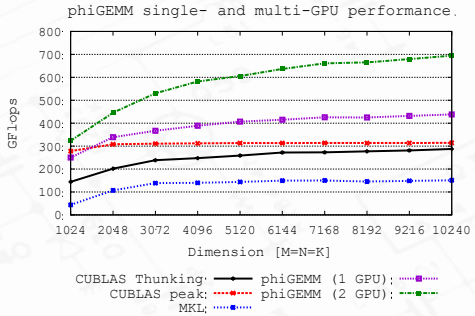
Development strategy: ADDUSDENS and NEWD

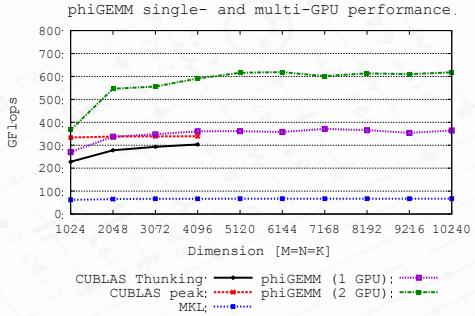
- Both routines are more compute-bounded than memory-bounded
  - we achieved up to 19~20-times* and 7~8-times acceleration
- All the data is moved to GPU memory only at once
  - some data structures do not change during the computation → preload
- External loops over atomic species are kept on the CPU side
- The QVAN2 (computation of the Fourier transformation of Q functions) is not yet CUDA
  - it is memory consuming and we are investigating how to split it
  - if the implementation is not efficient, better to keep it on the CPU (now we *overlap*)

Development strategy: PHIGEMM library

- Project started in 2010 (P. Yang)
- Inspired by M.Fatica LINPACK work
- Independent open-source library, BSD license
- GPU+CPU BLAS 3 *GEMM routine
- Manual or "semi-automatic" (SELFUNE) workload split
- Special-K for rectangular matrices
- C and FORTRAN interfaces
- Detailed profiling of each call (with a little "hack" in your code)
- CUDA 3.x* and CUDA 4.x compatible
- Pinned/non-pinned, sync/async
- Support of multi-GPU
- Current version 1.8.3

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





Development strategy: VLOC_PSI

It combines CUDA kernels and CUFFT/CUBLAS calls

- CUDA kernel (INIT_PSI) assembles the FFT grid
- CUFFT_INVERSE is performed (CUFFTEXECZ2Z)
- A CUDA kernel (VEC_PROD) performs multiplications over the vector
- CUFFT_FORWARD is performed (CUFFTEXECZ2Z)
- After transformation data need to be scaled (CUBLASZDSCAL)
- CUDA kernel (SAVE_HPSI) adds the new contribution to the final vector

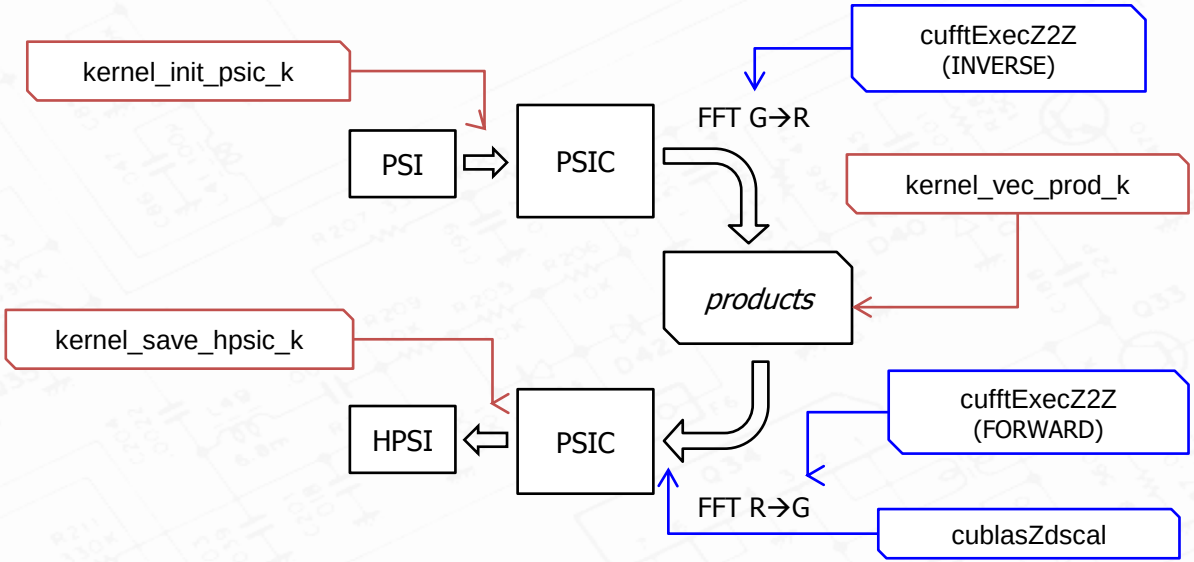
→ variants for gamma-point and k-point

→ assembling kernels are memory-bounded

Note:

- External loop over bands is kept on CPU side
- External loop over bands is not fixed (it decreases during the iterative SCF process!)

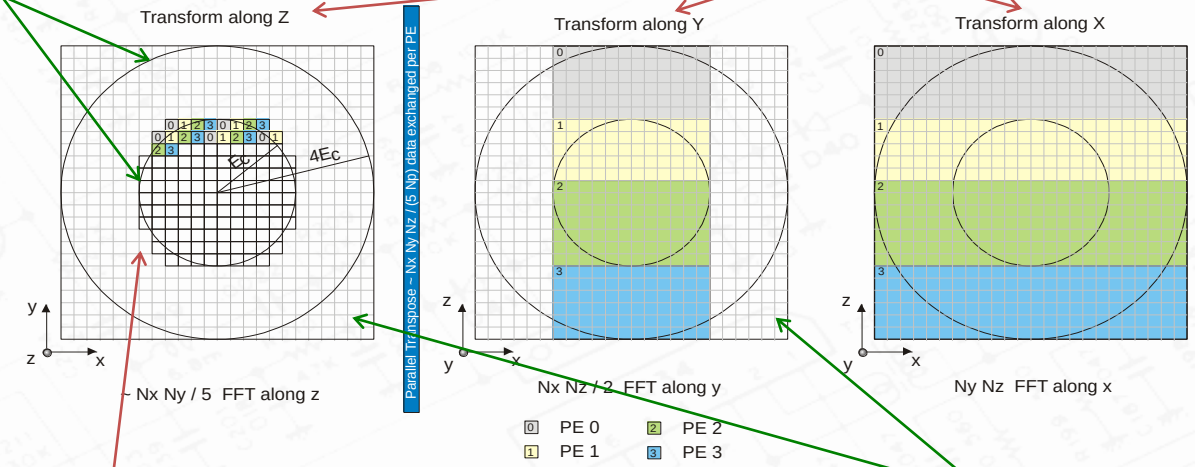
Development strategy: VLOC_PSI serial



The parallel "FFT issue"

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

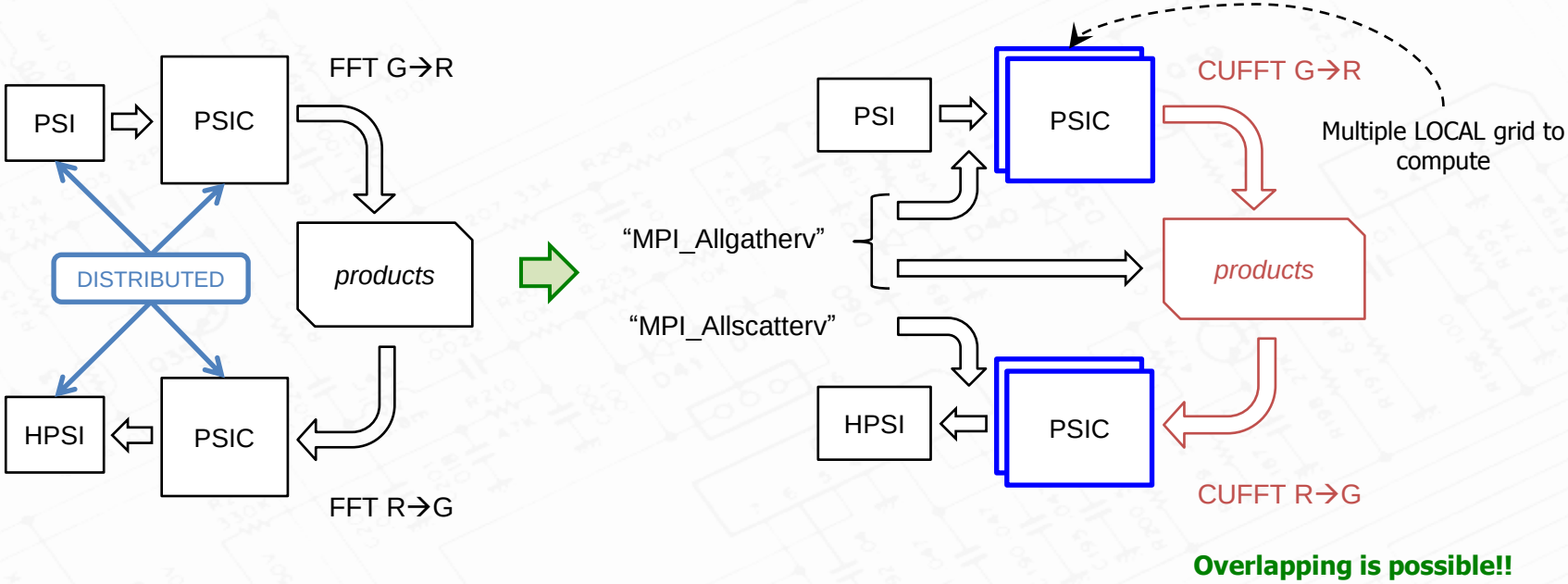
A single 3D-FFT is divided in independent 1D-FFTs



Data are not contiguous and not "trivially" distributed across processors

Zeros are not transformed. Different cut-offs preserve accuracy

Development strategy: VLOC_PSI parallel



Development strategy: memory allocation

On GPU side…

- Simple cudaMalloc of 85~90% of GPU memory
  - managed 1:1 MPI:GPU or N:1 MPI:GPU
  - if MAGMA is used than more space has to be left un-allocated
- Pointer to dev_scratch_QE is globally accessible
- phiGEMM (through phiGemmInit) makes use this buffer to perform GEMM operation

On CPU side…

- cudaMemCopy (few cudaMemSet)
- "shifts" are pre-computed considering memory alignment
- Specific data allocations can be either pinned or not
  - BUT pinned memory slow down all the application!
  - No pinned memory by default!

Benchmarking platforms

Wide set of workstations (and different GPU) & HPC clusters…

- **FERMI** (ICHEC): assembled workstation
  - CPU: 2 Intel Xeon X5650 (6-core), 24 GByte RAM
  - GPU: 2 {C2050, GTX480, C2075}
  - SW: CUDA 4.1, Intel compilers
- **GEMINI** (ICHEC): Dell Power Edge C6145 & C410x
  - CPU: 4 AMD 6136 (8-core), 64 GByte RAM
  - GPU: 4 M2090
  - SW: CUDA 4.x, Intel compilers, PGI (12.x)
- **STONEY** (ICHEC): Bull Novascale R422-E2, 24 GPU nodes
  - CPU: 2 Intel Xeon X5560 (4-core), 48 GByte RAM
  - GPU: 2 M2090
  - SW: CUDA 4.0, Intel compilers
- LONGHORN (TACC): Dell XD Cluster, 240 GPU nodes
  - CPU: 2 Intel Xeon 5355 (4-core), 48 GByte RAM
  - GPU: 2 Quadro FX 5800
  - SW: CUDA 4.0, Intel compilers
- CURIE (CEA): Bull GPUs B505 blades, 144 GPU nodes
  - CPU: 2 Intel Westmere (4-core), 24 GByte RAM
  - GPU: 2 M2090
  - SW: CUDA 4.1, Bull MPI stack, Intel compiler
- **PLX** (CINECA): IBM iDataPlex DX360M3, 264 GPU nodes
  - CPU: 2 Intel Westmere (6-core), 48 GByte RAM
  - GPU: 2 M2070
  - SW: CUDA 4.0, Intel compilers, PGI (11.x)

Power measurement

(cheap) Prodigit 200M Plug-in Main Power and Energy Monitor

- max voltage 250V
- max current 15A
- max active power 3750 Watts
- kWH accuracy: 30ppm
- kWh displayed accuracy: +/- 0.01

Plug to FERMI...

- measuring operational power absorbed
- the workstation is "equivalent" to 1 PLX node
  - IBM Dataplex nodes are higher energy efficient than a workstation, CPU are equivalent, GPUs absorb comparable amount of power
- *qualitative* considerations...

Benchmark philosophy

Impossible to represent a realistic scenario using only ad-hoc designed benchmarks...

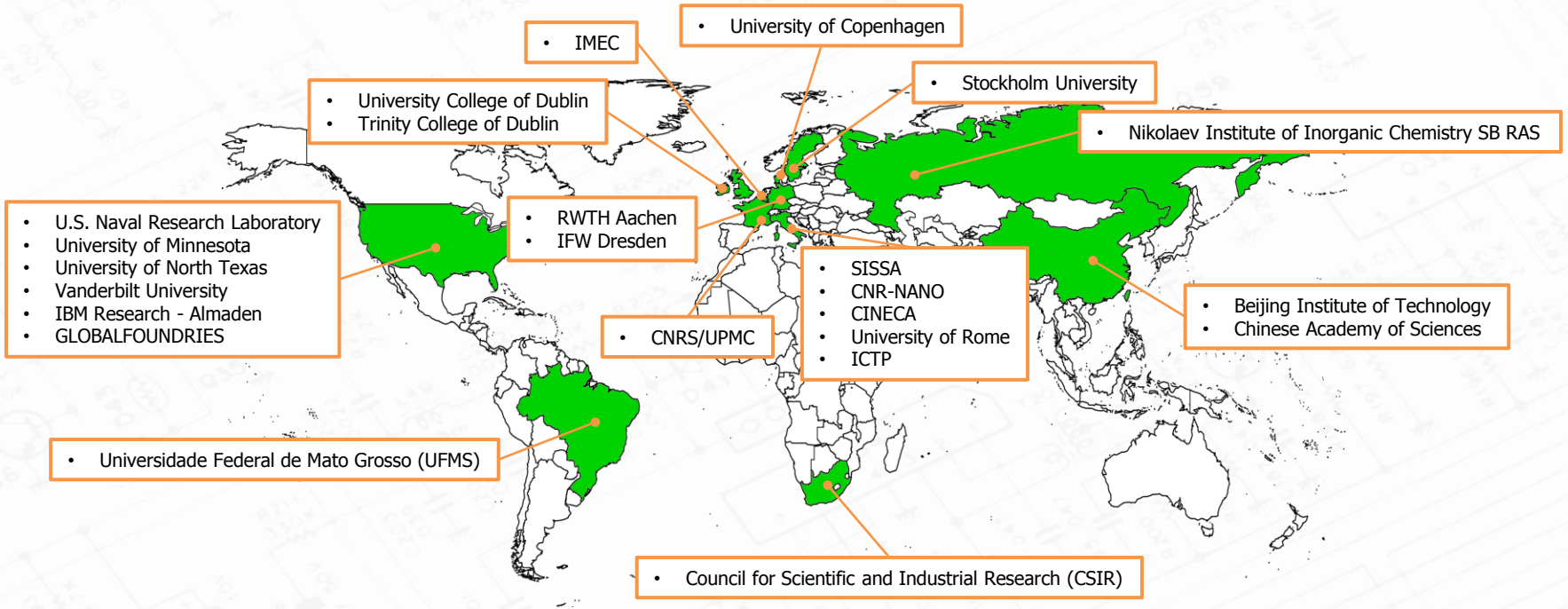
- user inputs that represent on-going investigations
- user inputs that represent challenge (too long to run here)
- user inputs that represent starting point to go through new "science"

Two-side goal was

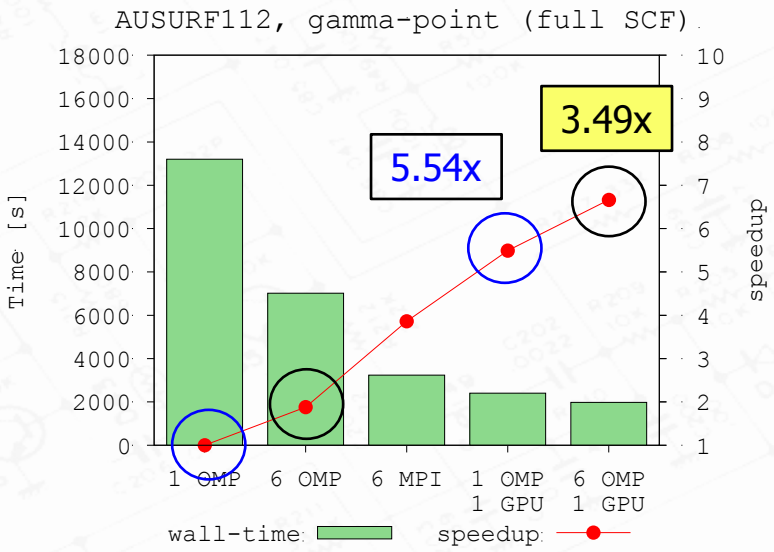
- provide a feedback to US, evaluating if what we implemented actually cover what a generic input might trigger
- provide a feedback to the USERS, encouraging them to try and support the GPU implementation

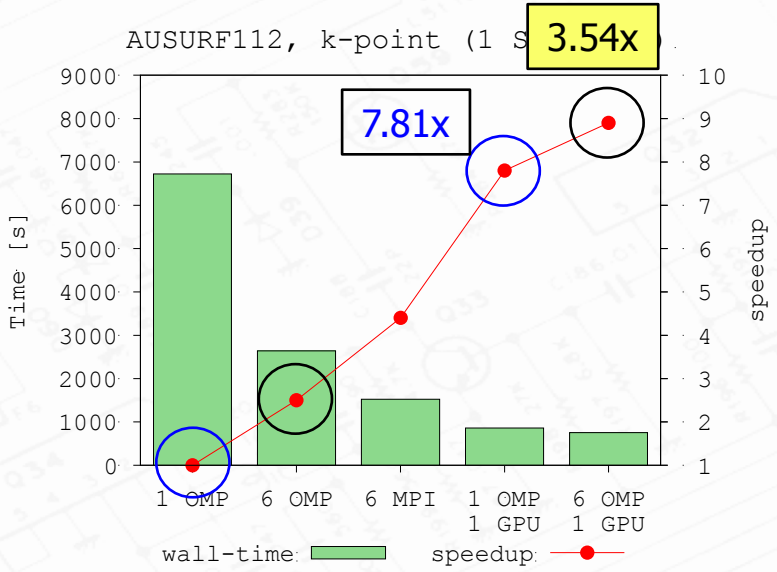
- We received almost 15 contributions after a "Call of Benchmarks"
- We selected 8~9 of them as benchmarks (both for serial and parallel
  - 3 are quite challenging (> 500 atoms, thousands of electrons)

Those who have contacted us directly...

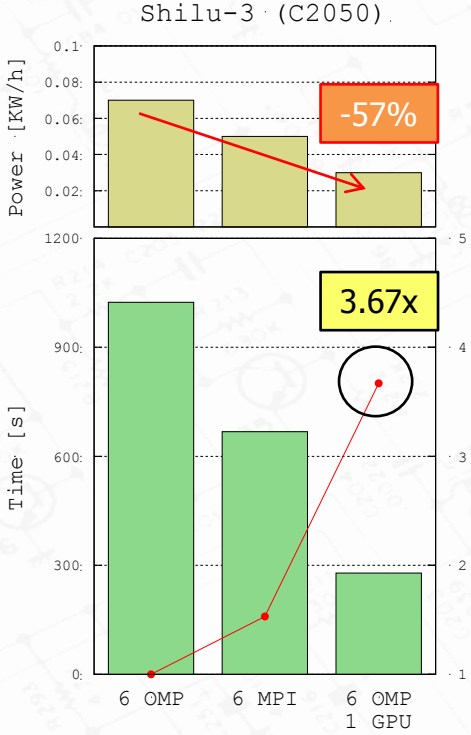


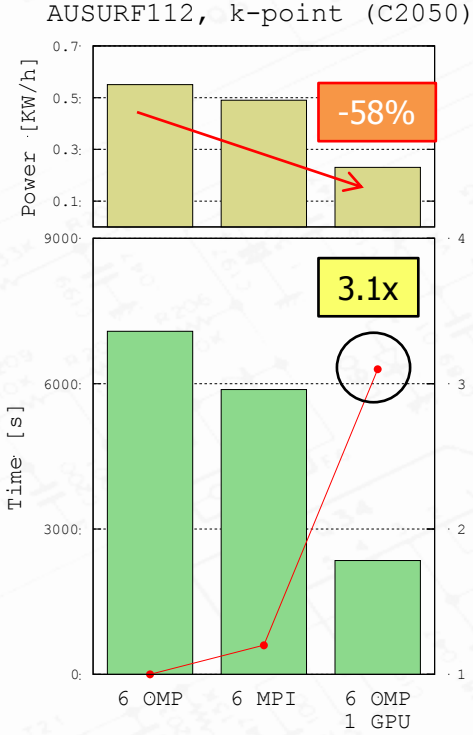
AUSURF112, serial (FERMI)

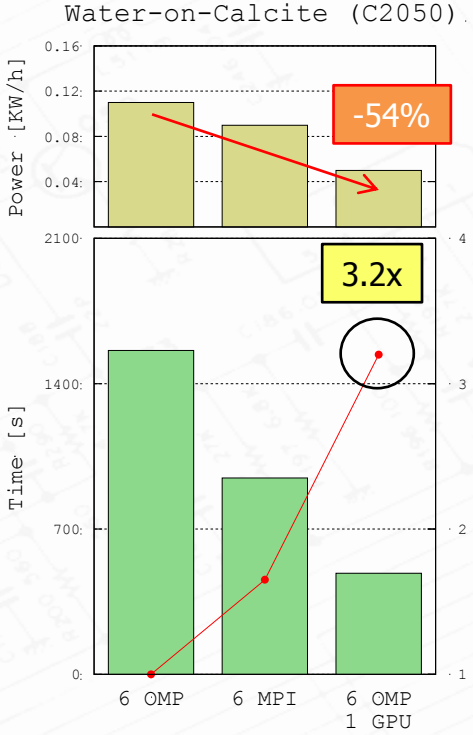




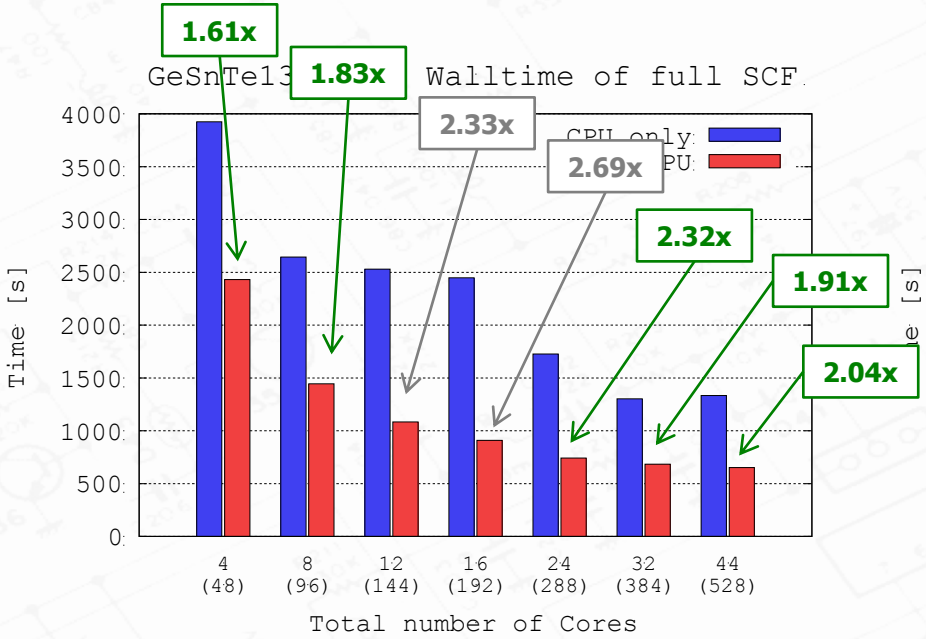
Performance & Power consumption (serial)



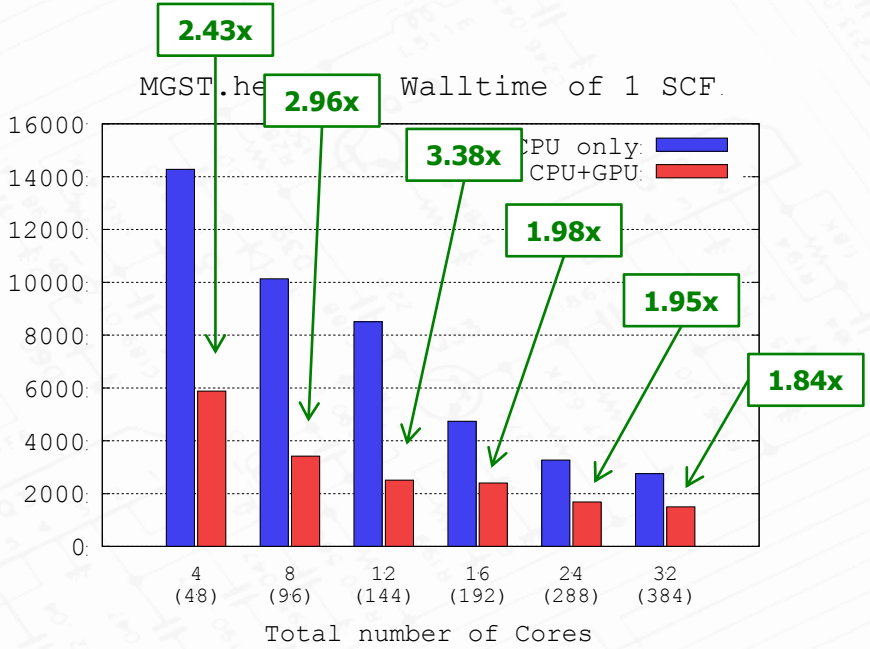




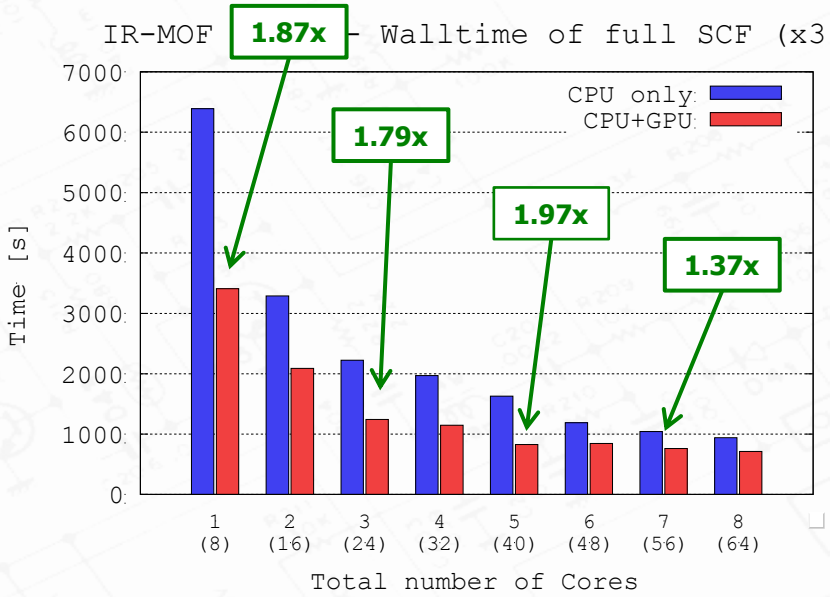
GeSnTe134.in, parallel (PLX)



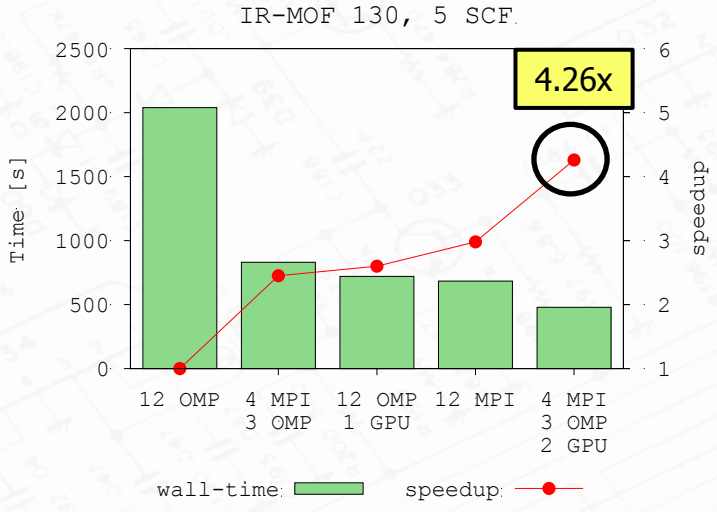
MGST.hex.rx, parallel (PLX)



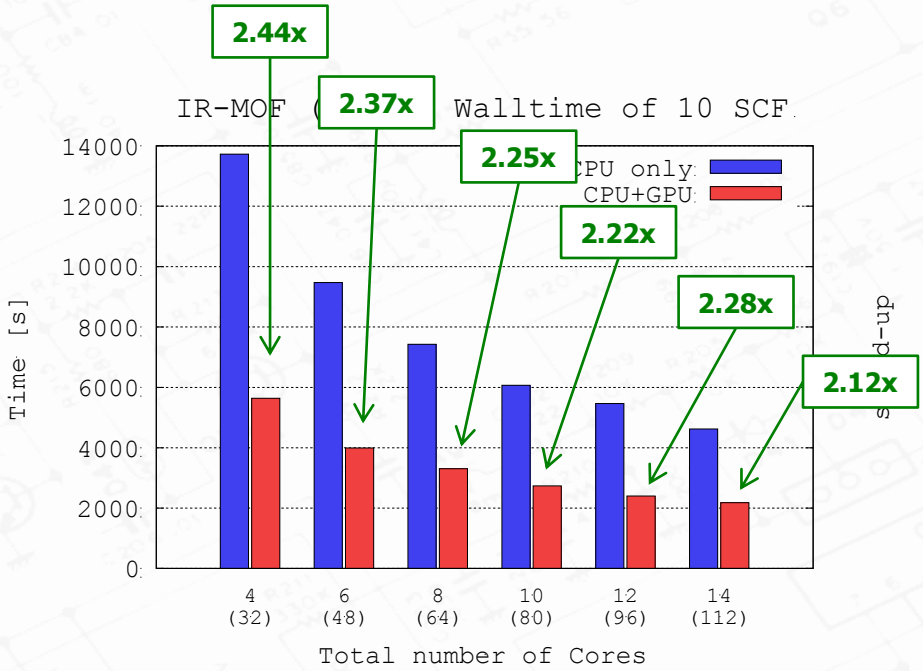
IRMOF-M11 (130), parallel (STONEY)

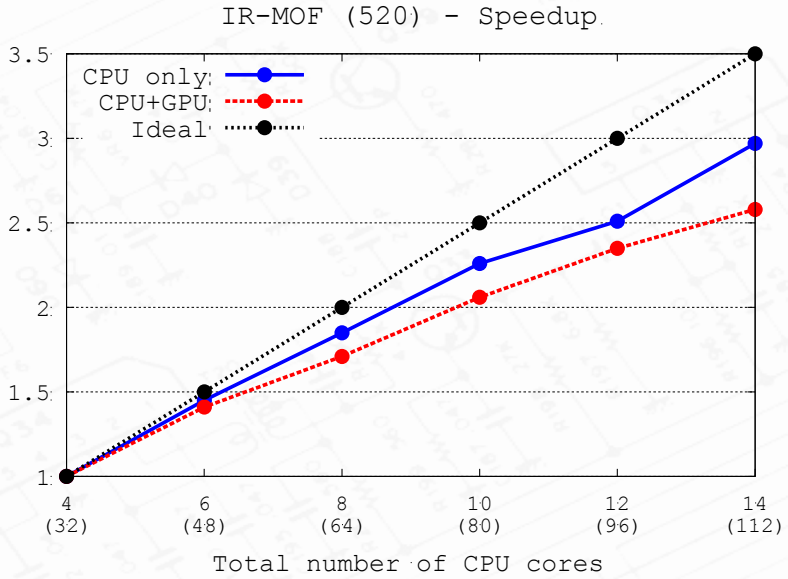


Very small system... it can even run in SERIAL!

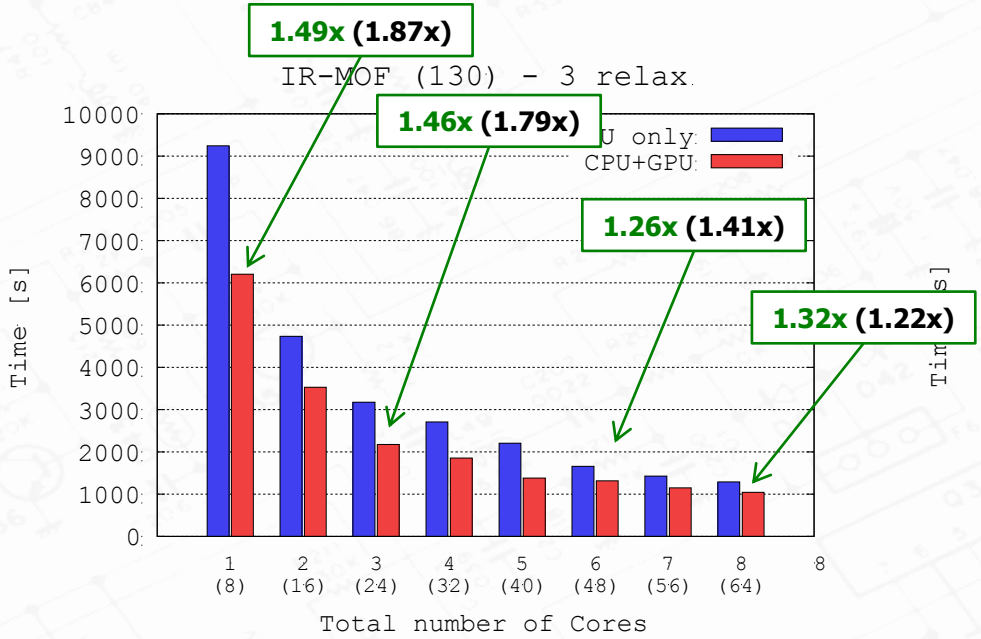


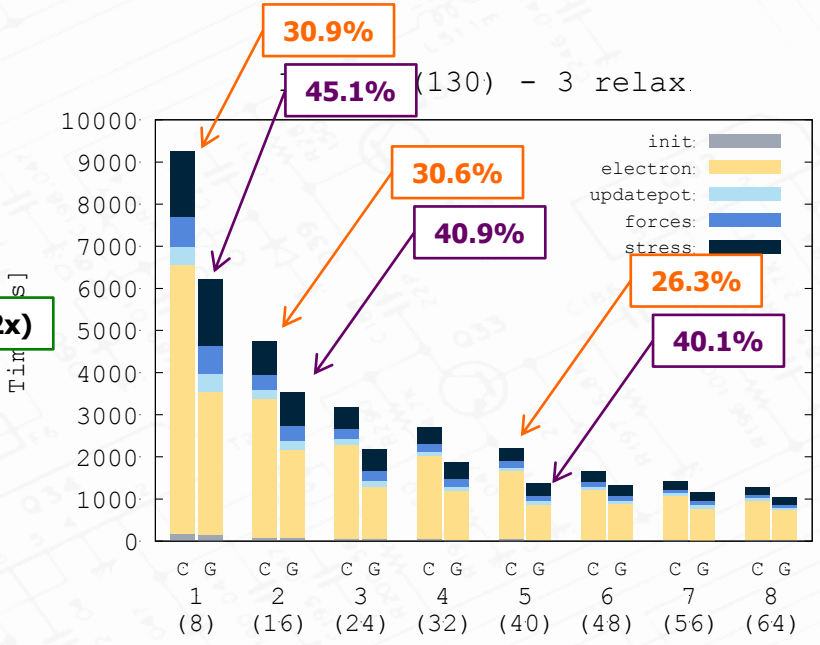
IRMOF-M11 (520), parallel (STONEY)





Hitting the limit

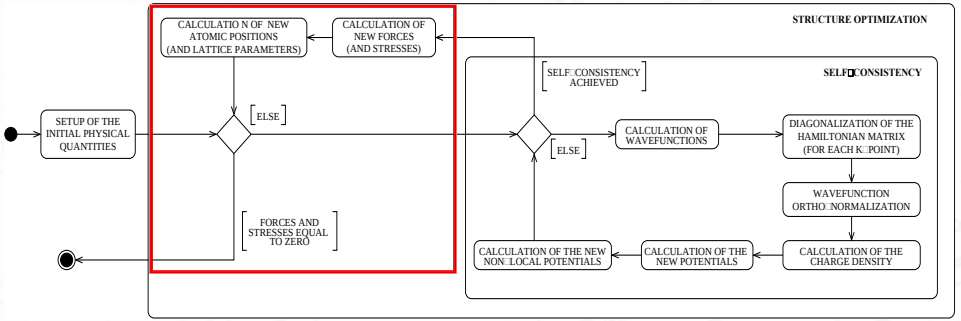




Closing the loop...

Accelerate FORCE and STRESS calculations

Open ACC



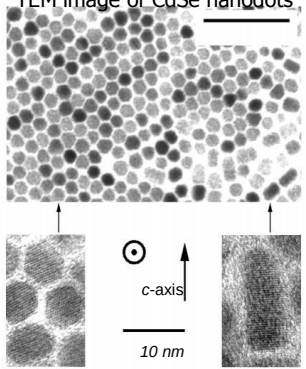
What we learn...

- easy (more or less) but *test-and-try* approach
- difficulties to compile the code using PGI
- (now) no data transfer overlapping
- **BUT** if not heavy computation → big loss of performance

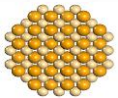
PRACE Preparatory Access

- Physical interests:
  - prototypical material for optoelectronic applications (e.g. light emitting diodes, solar cells)
  - easy-growth nanoparticles through and chemical processes (colloidal synthesis)
- Numerical challenge:
  - high electrons-to-atoms ratio in pseudo-potential
  - calculations due to the inclusion of Cd_{4d} electrons in valence shell
- GPU challenge:
  - accelerate stress/forces calculations using (first) OpenACC
- Collaborations: A. Calzolari (CNR-NANO), C. Cavazzoni (CINECA)
- Funding: project pa0699 (2012), CURIE cluster, 300K hours

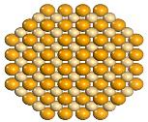
TEM image of CdSe nanodots



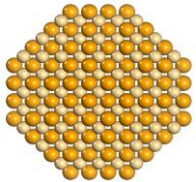
X. Peng et al. Nature **404**, 59 (2000).



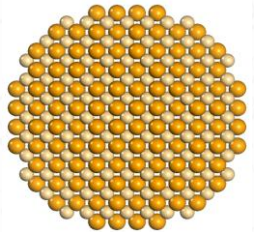
r=10 Å
#at =159, #el = 638



r=12 Å
#at =275, #el = 1110



r=15 Å
#at =489, #el = 1938



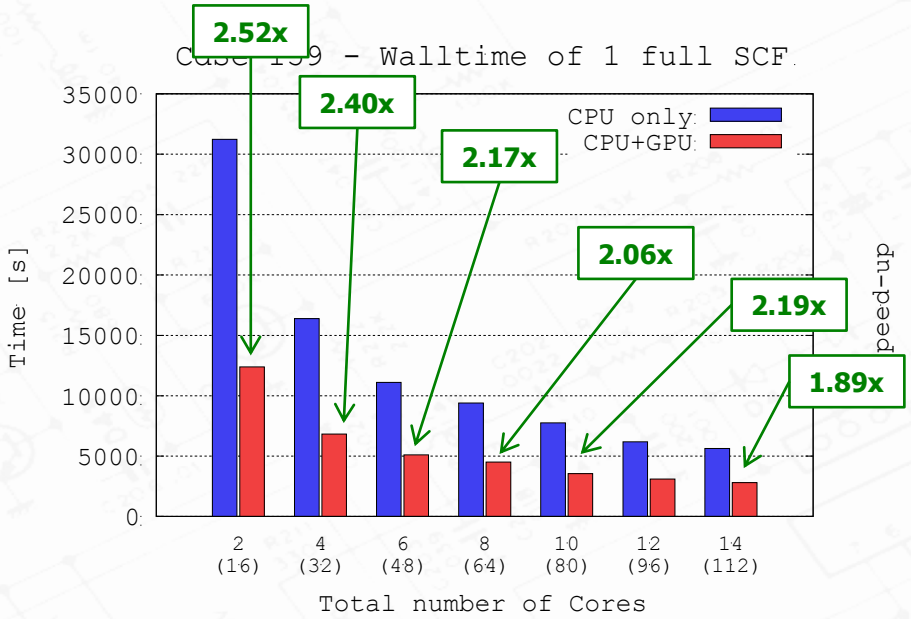
r=18 Å
#at =922, #el = 3668

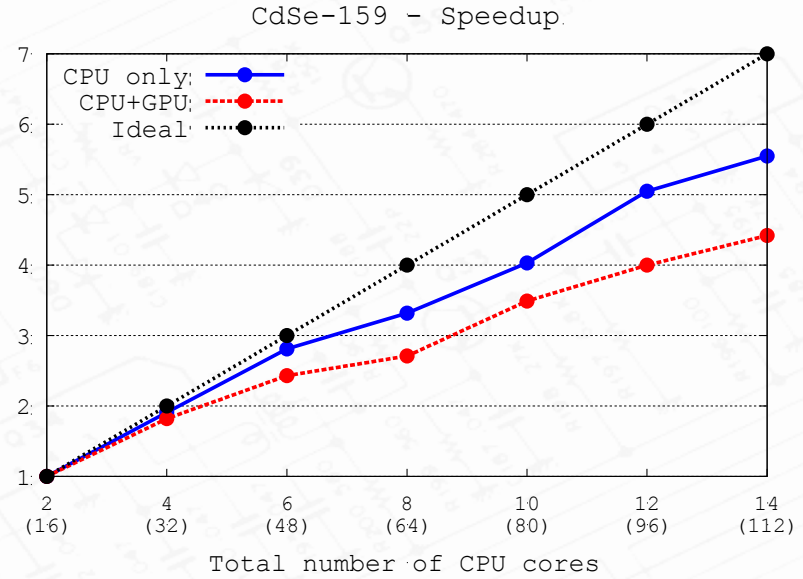
r=20 Å
#at =12149, #el = 4844

r=25 Å
#at =2365, #el = 9370

r=30 Å
#at =4109, #ela = 16282

CdSe-159, parallel (STONEY)





Achievements

Direct effects...

- PWSCF has been extended to use GPUs to accelerate both gamma- and k-points calculations
- Performance improvements for a **PRODUCTION** package
  - for serial, an average of 3-/3.5-times (full-socket vs full-socket + GPU)
  - for parallel, an average of >= 2-times (full-node vs full-node + GPUs)
  - no (visible) scalability improvement but **faster time-to-solution**
- 99% match of numerical consistency between CPU-only and CPU+GPU calculations
- tested on several platforms and several GPUs
- average of 150 downloads per released versions (6 different 0.X releases since July 2011)

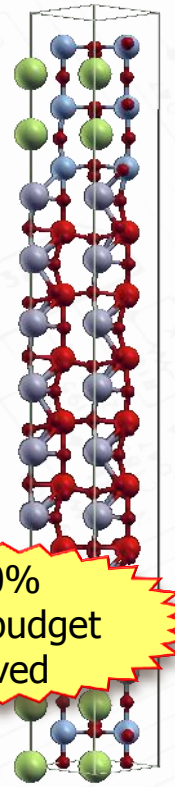
Side effects...

- new interesting benchmarks provided by users
- lots of profiling for both CPU and CPU+GPU (CPU load, GPU load, memory occupancy, internal clocks,...)
- few improvements on the CPU code (OpenMP)

Best Practices

- Scientific case: LSMO-BFO (120 atoms)
- PI: Rodrigo Neumann Barros Ferreira, PhD Student Solid State Physics Department – Physics Institute, Rio de Janeiro Federal University
- Description: 1024 electrons, 615 different quantum-mechanical states considered, **40 k-points** for the integration over the Brillouin zone.
- Goal: exploit QE *pool* parallelism and GPU → keep the FFT local by using npool=npocs

Computer Nodes	Execution Time [s] #10 self-consistency cycles	Speed-up
1 x IBM Power 575, P6 4.7 GHz (32 cores)	20314.59	
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 (2x)**



60% user budget saved

Current (known) limitations

- non-collinear calculations
  - special variant for VLOC_PSI not yet accelerated
- spin magnetization
  - additional (CPU) code that slow down the GPU performance
- Low number of atoms (<32)
  - if there is not "enough" work to keep CPU busy... GPU will not do better
- High number of k-points
  - systems with high number of k-points (> 8) suffer of performance degradation due to I/O
  - mitigable in principle by using in a smart way the MPI parallelism
- Periodic systems with low number of atoms (<32) + high number of k-points
  - better to physically create a large system with less k-point (if it has sense)
  - there might me exceptions... (i.e. when the cell is big)

Big Challenges & Collaborations

Petascale computations in mineral physics with the Quantum ESPRESSO

- P.I.: Prof Renata Wentzcovitch (Chemical Engineering & Materials Science, U. of Minnesota)
- Funded by: NSF
- Objectives: investigation of mineral properties must be investigated in a wide range of pressure, temperature, and chemical compositions
- Target machine (facility): Blue Waters (NCSA)

Center for software innovation: (R)Evolutionary Materials Development

- P.I.: Prof Marco Buongiorno Nardelli (Physics and Chemistry Departments, U. of North Texas)
- Funded by: DoE
- Objectives: mapping new materials enabled technologies (METs) genome across different length scales to enable accelerated discovery and technological transfer
- Target machine (facility): Jaguar/Titan (ORNL) + local resources

Future Plans

Future developments will focus on three independent directions...

1. Improve PWSCF by engage big scientific challenges with scientists (users drive the development priorities)
1. Extend GPU capabilities to other code of the suite (next: CP, PHONON)
1. Improve the support for multi-GPU in serial calculations

QE-GPU is an open collaboration:

- repository connected to the main one, QE-GPU is like "an extension" of the QE suite
- contributors can develop GPU code with no impact to the main suite (modularized structure)
- GPU technology exploration and evaluation

Thank you for your attention!

No CPUs or GPUs have been damaged during the preparation of this talk (-:

DOWNLOAD IT AT http://tinyurl.com/gpu-pwscf

Acknowledgments:

Ivan Girotto (ICHEC, now ICTP), Carlo Cavazzoni (CINECA), Paolo Giannozzi (U. of UDine/DEMOCRITOS), Layla Martin-Samos (DEMOCRITOS/U. of Nuova Goriza), Arrigo Calzolari (CNR-NANO), Wei Zhang (RWTH Aachen University), Clima Sergiu (IMEC), Koroteev Victor (NIIC SB RAS), Bhagawan Sahu (Globalfoundries) and many others...

Science Foundation Ireland

Ireland's EU Structural Funds Programmes 2007 - 2013

Co-funded by the Irish Government and the European Union

EUROPEAN REGIONAL DEVELOPMENT FUND

AN ROINN OIDEACHAIS AGUS EOLAÍOCHTA | DEPARTMENT OF EDUCATION AND SCIENCE

HEA
Higher Education Authority
An tÚdarás um Ard-Oideachas

What can QUANTUM ESPRESSO's PWSCF do?

- both gamma−point and **k**-point calculation
- both insulators and metals, with various flavors of broadening, or tetrahedra
- any crystal structure or supercell form
- ground-state energy and one-electron (Kohn-Sham) orbitals;
- atomic forces, stresses, and structural optimization;
- molecular dynamics on the ground-state Born-Oppenheimer surface, also with variable cell;
- Nudged Elastic Band (NEB) and Fourier String Method Dynamics (SMD) methods
- norm-conserving PP's in separable form, ultrasoft Vanderbilt PP's, PAW
- almost all flavours of LDA and of gradient-corrected exchange-correlation functionals (PW91, PBE, B88-P86, BLYP,...), DFT+U, exact exchange and a few hybrid functionals (PBE0, B3LYP), TPSS meta-GGA
- spin-polarized, magnetic systems (including non-collinear magnetism and spin-orbit interactions)

…

(see: http://www.quantum-espresso.org/whatcanqedo.php)

Coding numerical formulas …

```
SCF:
      compute potential
      solve KS eigen-problem
      Loop over k-points:
        Davidson iteration / CG iteration:
        compute/update H * psi:
          compute kinetic and non-local term (in G space)
          Loop over (not converged) bands:
                FFT psi to R space
                compute V * psi
                FFT V * psi back to G space

        project H in the reduced space (ZGEMM)
        diagonalize the reduced Hamiltonian:
                cholesky factorization
                call to LAPACK/SCALAPACK/MAGMA/PLASMA
                   diagonalization routine

      compute new density
        loop over k-points:
                loop over bands:
                        FFT psi to R space
                        accumulate psi
        charge density symmetrisation
```

$$\hat{H}_{KS}\left|y_{\vec{k},b}\right\rangle$$

$$\left\langle y_{\vec{k},a}\left|\hat{H}_{KS}\right|y_{\vec{k},b}\right\rangle$$

$$\hat{H}_{KS}\left|\psi_{\vec{k},b}\right\rangle=\varepsilon_{\vec{k},b}\left|\psi_{\vec{k},b}\right\rangle$$

$$n(\vec{r})=2\sum_{\vec{k}}\sum_{v}\left|y_{v,\vec{k}}(\vec{r})\right|^2$$

H * psi (VLOC_PSI)

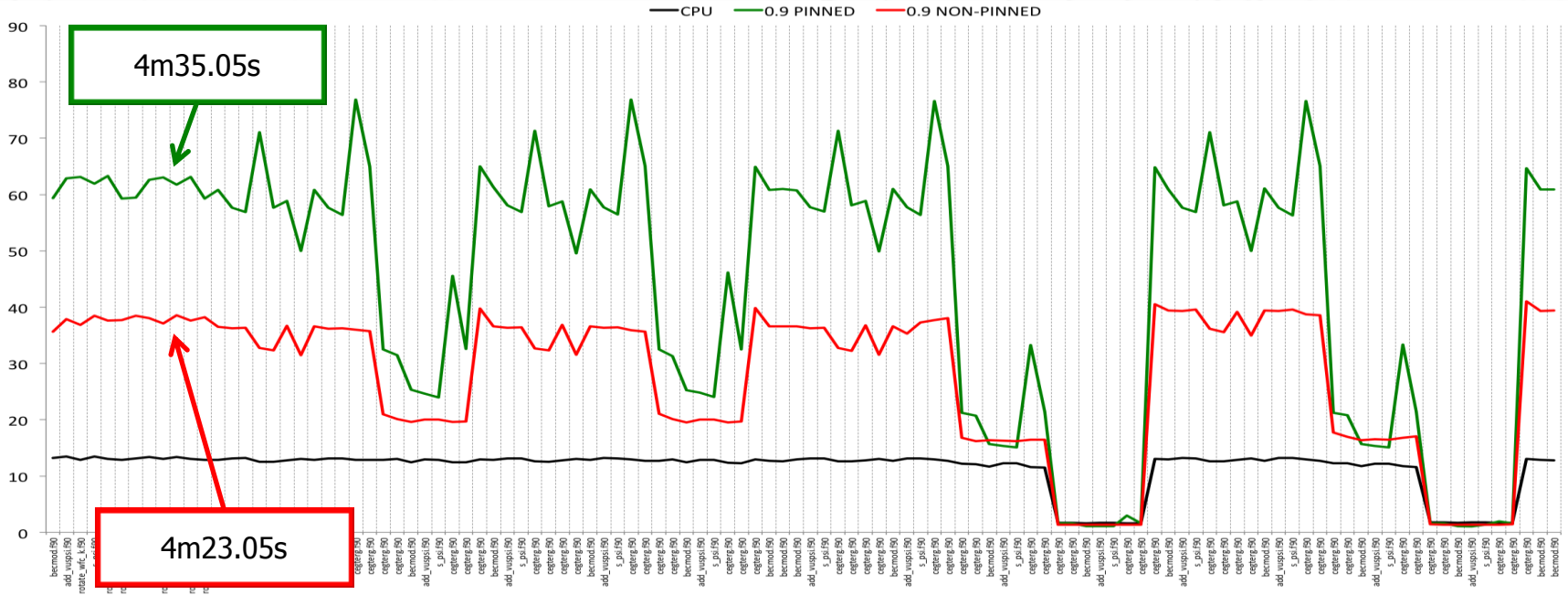
```
compute/update H * psi:
    compute kinetic and non-local term (in G space)
          complexity : N_i × (N × N_g + N_g × N × N_p)
    Loop over (not converged) bands:
          FFT (psi) to R space
               complexity : N_i × N_b × FFT(N_r)
            compute V * psi
               complexity : N_i × N_b × N_r
            FFT (V * psi) back to G space
               complexity : N_i × N_b × FFT(N_r)
            compute Vexx:
               complexity : N_i × N_c × N_q × N_b × (5 × N_r + 2 × 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 (usually about 10)

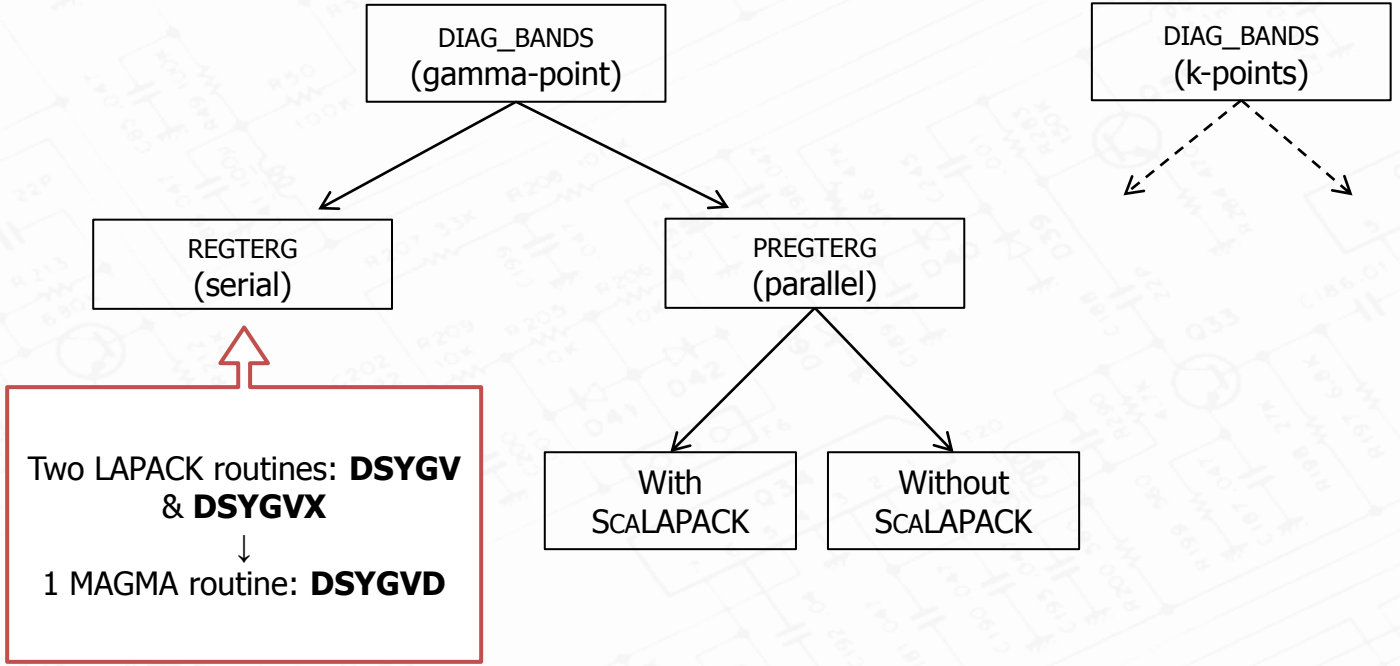
N_p = number of PP projector
N_r = size of the 3D FFT grid
N_q = number of q-point (may be different from N_k)

Pinned versus non-Pinned

AUSURF112, k point (ZGEMM), 4MPI x 3 OMP with 2 GPUs, few SCF iterations

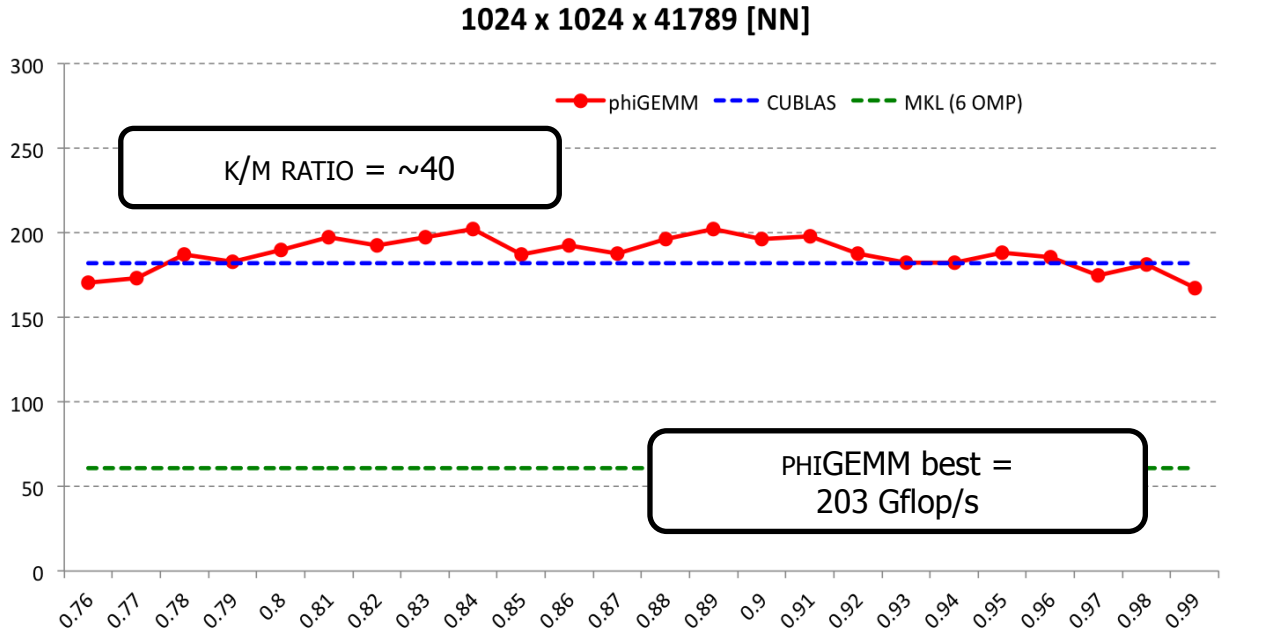


Development strategy: diagonalization



BACKUP SLIDE

PHIGEMM : special-K/1



PHIGEMM does not perform very well, overlapping COMP & COMM does not produce any increment in performance

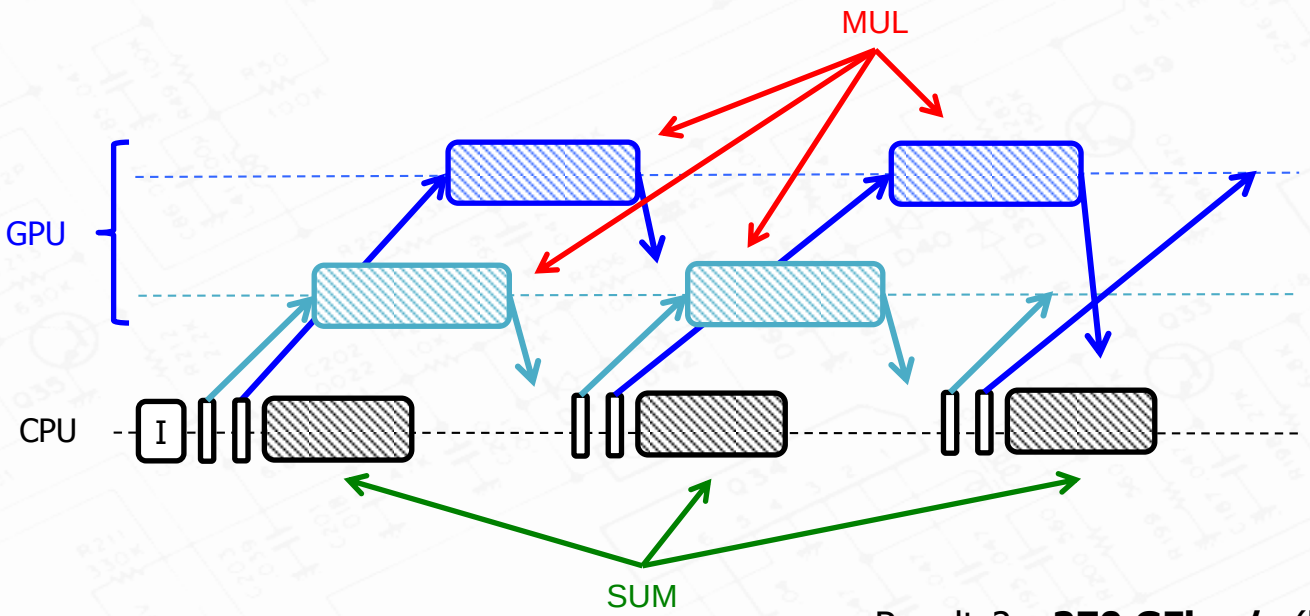
⇩

Split
C = alpha*op(A)*op(B) + beta*C
and perform the multiplication on GPU and the sum on CPU

⇩

small block "scheuduled" statically one after the other, overlapping in data movement (pinned buffer)

PHIGEMM : special-K/2



Results? **~270 GFlop/s** (before ~203)