CP2K: HIGH PERFORMANCE ATOMISTIC SIMULATION

Iain Bethune
ibethune@epcc.ed.ac.uk

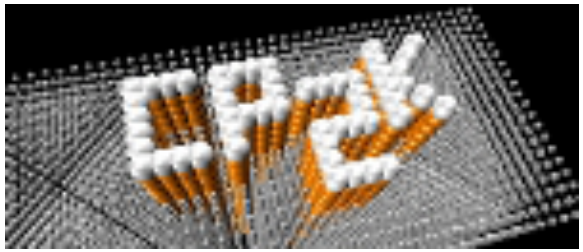
http://tinyurl.com/CP2K-Dublin-2016

epcc PRACE

CP2K Overview

"CP2K is a program to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as e.g., density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW) and classical pair and many-body potentials."

From www.cp2k.org (2004!)



epcc | PRACE

CP2K Overview

- Many force models:
  - Classical
  - DFT (GPW, GAPW + vDW)
  - Hybrid Hartree-Fock
  - LS-DFT
  - post-HF (MP2, RPA)
  - Combinations (QM/MM, mixed)
- Simulation tools
  - MD (various ensembles)
  - Monte Carlo
  - Minimisation (GEO/CELL_OPT)
  - Properties (Spectra, excitations …)
- Open Source
  - GPL, www.cp2k.org
  - 1m loc, ~2 commits per day
  - ~20 core developers

cp2k

epcc | PRACE

CP2K History

- 25th June 2001 – CP2K repository online at berliOS.de
  - Merger of Quickstep (DFT) + FIST (MD) codes
  - Jürg Hutter, Matthias Krack, Chris Mundy
- Oct 2011 – First 'official' release
  - CP2K 2.2
- 15 years on…
  - 1m lines of code, ~16k commits
  - 25 developers + many contributors
  - 1000s of users
  - Fully open-source (GPL)

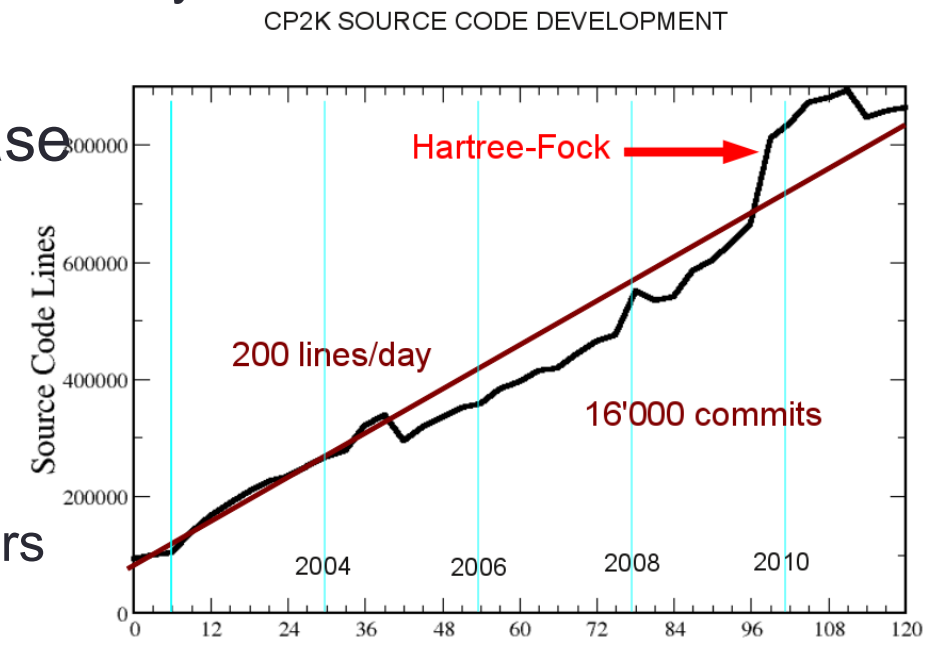
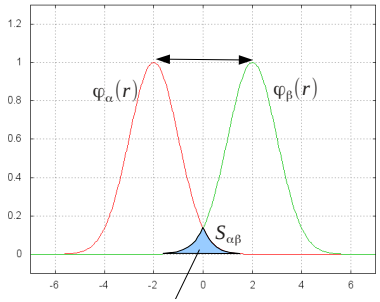


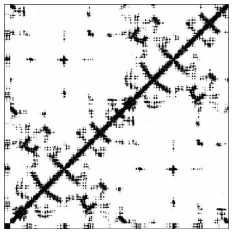
Image from Jürg Hutter

epcc PRACE

CP2K Features

- QUICKSTEP DFT: Gaussian and Plane Waves Method (VandeVondele *et al*, Comp. Phys. Comm., 2005)
  - Advantages of atom-centred basis (primary)
    - Density, KS matrices are sparse
  - Advantages of plane-wave basis (auxiliary)
    - Efficient computation of Hartree potential
  - Efficient mapping between basis sets
    - -> Construction of the KS Matrix is ~O(n)





- Orbital Transformation Method (VandeVondele & Hutter, J. Chem. Phys., 2003)
  - Replacement for traditional diagonalisation to optimise KS orbitals within SCF (non-metallic systems only)
    - Cubic scaling but ~10% cost

epcc PRACE

CP2K Features

$$
\begin{aligned}
E^{\mathrm{el}}[n] &= \sum_{\mu\nu} P_{\mu\nu} \left\langle \varphi_\mu \left| -\frac{1}{2}\nabla^2 + V_{\mathrm{loc}}^{\mathrm{SR}} + V_{\mathrm{nl}} \right| \varphi_\nu \right\rangle \\
&+ 2\pi\Omega \sum_{\mathbf{G}} \frac{\tilde{n}^*_{\mathrm{tot}}(\mathbf{G})\tilde{n}_{\mathrm{tot}}(\mathbf{G})}{\mathbf{G}^2} + \sum_{\mathbf{R}} \tilde{n}(\mathbf{R}) V^{\mathrm{XC}}(\mathbf{R}) \\
&= \sum_{\mu\nu} P_{\mu\nu} \left(\left\langle \varphi_\mu \left| -\frac{1}{2}\nabla^2 + V^{\mathrm{ext}} \right| \varphi_\nu \right\rangle + \sum_{\mathbf{R}} V_{\mu\nu}^{\mathrm{HXC}}(\mathbf{R}) \varphi'_{\mu\nu}(\mathbf{R}) \right)
\end{aligned}
$$

From Marcella Iannuzzi,
http://archer.ac.uk/training/course-material/2014/08/CP2K/Slides/gpw_gapw.pdf

epcc PRACE

CP2K Features

64 H_2O
32 CPUs IBM SP4

1 SCF iter	DZVP	TZVP	TZV2P	QZV2P	QZV3P
OT	0.50	0.60	0.77	0.87	1.06
Diagonalisation	6.02	8.40	13.80	17.34	24.59

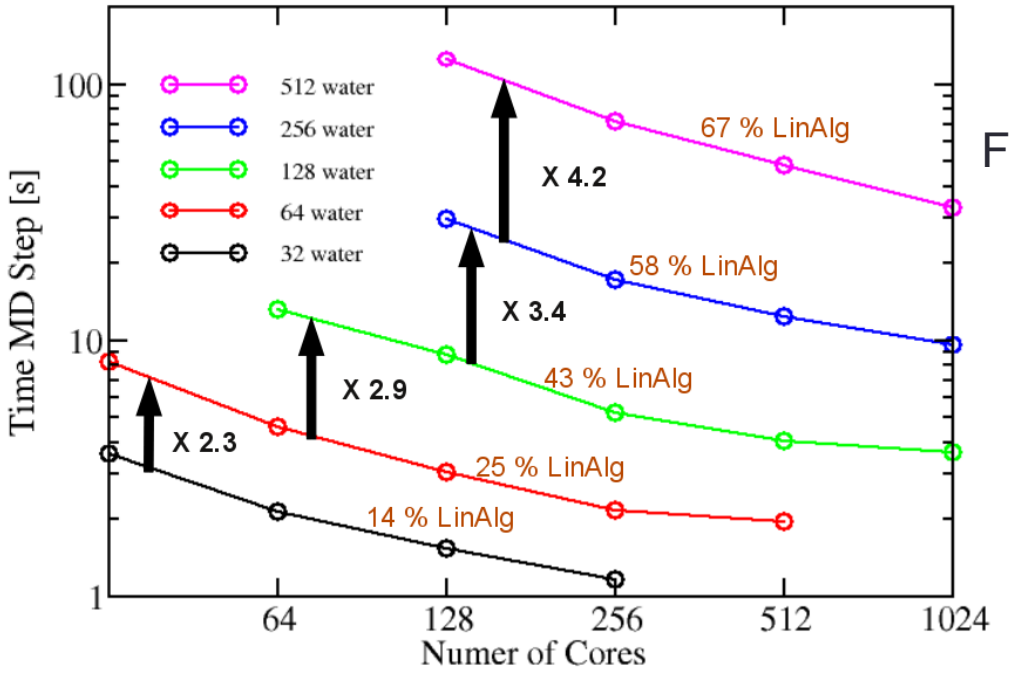


Figure from Jürg Hutter

CP2K Features

- QM/MM (Laino *et al*, JCTC, 2005, 2006)
  - Fully periodic, linear scaling electrostatic coupling
- Gaussian and Augmented Plane Waves (Iannuzzi *et al*, CHIMIA, 2005)
  - Partitioning the electronic density -> all-electron calculations
- Hartree-Fock Exchange (Guidon *et al*, JCP, 2008)
  - Beyond local DFT (also MP2, RPA…)
  - Auxiliary Density Matrix Method (Guidon *et al*, JCTC, 2010)
- Linear Scaling DFT (VandeVondele, Borstnik & Hutter, JCTC, 2012)
  - Fully linear scaling condensed-phase DFT, up to millions of atoms
- https://www.epcc.ed.ac.uk/sites/default/files/PDF/CP2K-UK-2015-Hutter.pdf

epcc PRACE

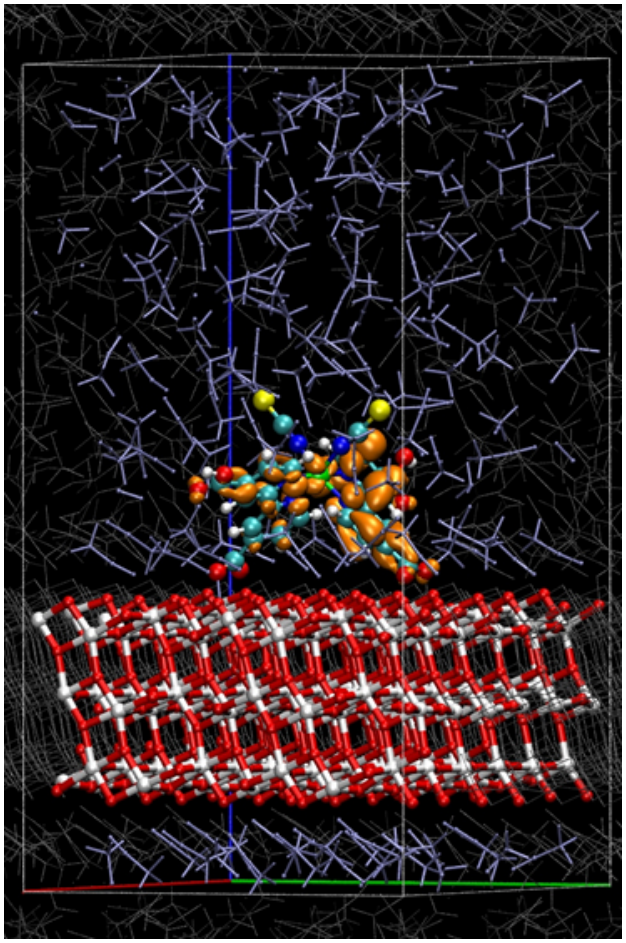
CP2K Features

- And LOTS more…
  - Recent review paper:

  Hutter *et al*, *WIREs Comput Mol Sci* 2014, 4:15–25
  http://dx.doi.org/10.1002/wcms.1159

- Some highlight applications:
  - http://www.cp2k.org/science

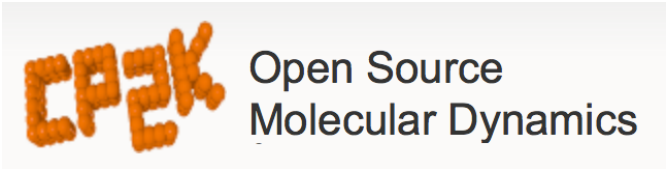
- All for free!
  - Please cite the references
  - Please give feedback / patches / feature requests
  - Please spread the word about CP2K!



DSSC: see Shiffmann *et al*, PNAS, 2010

CP2K Information

- CP2K Website (http://www.cp2k.org)
  - Everything else is linked from here
  - Now a wiki – so feel free to contribute!



- CP2K Sourceforge site (http://sf.net/p/cp2k)
  - Contains source code repository (SVN)
    - public read-only, read-write access to developers
  - Bug reporting
  - Source tarball / binary downloads

epcc PRACE

CP2K Information

- CP2K Discussion Group (http://groups.google.com/group/cp2k)
  - Email / web forum
  - Users and developers
  - Searchable history

- CP2K Input reference manual (http://manual.cp2k.org)
  - Documents *every* possible CP2K input keyword
  - Mostly with helpful descriptions
  - Beta CP2K online input editor
    - http://cp2k-www.epcc.ed.ac.uk/cp2k-input-editor

epcc PRACE

CP2K Information

- Which version?
  - Current release 3.0 (Dec 2015)

  + stable, major bug-fixes are back-ported
  + source and binaries available from http://www.cp2k.org/download
  - missing absolute latest features, minor bugs are not always fixed

  - Current release 2.6.2 (Sep 2015)

  + available for Ubuntu / Debian / Fedora via package managers

  - SVN trunk version 4.0

  + latest features, fixes, performance improvements
  + actively developed
  - bugs may exist (see http://dashboard.cp2k.org)
  - must be obtained from SVN and compiled from source

http://www.cp2k.org/version_history

epcc PRACE

Support for CP2K Users

- CP2K-UK: EPSRC Software for the Future
  - £500,000, 2013-2018
  - EPCC, UCL, KCL + 7 supporting groups

EPSRC
Pioneering research and skills

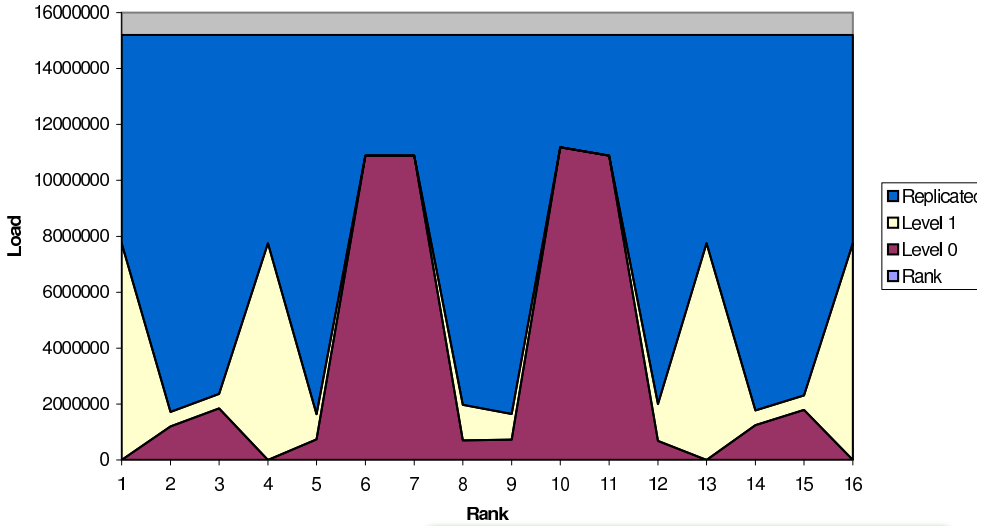
- Aims
  - Grow and develop existing CP2K community in UK
  - Lower barriers to *usage* and *development* of CP2K
  - Long-term sustainability of CP2K
  - Extend ability of CP2K to tackle challenging systems

- Annual user meetings & training
  - CP2K summer school 23-26th August, London
- Updates via mailing list

epcc | PRACE

CP2K Performance

- Distributed realspace grids
  - Overcome memory bottleneck
  - Reduce communication costs
  - Parallel load balancing
    - On a single grid level
    - Re-ordering multiple grid levels
    - Finely balance with replicated tasks



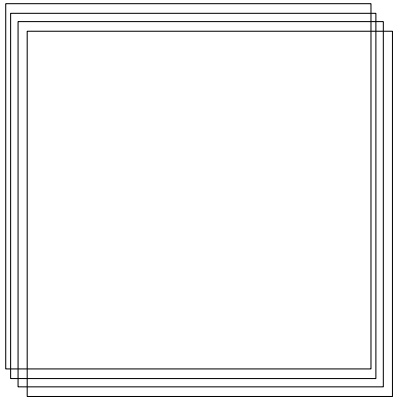
Level 1, fine grid, distributed

1	2	3
4	5	6
7	8	9

Level 2, medium grid, dist

5	6	8
3	1	7
9	4	2

Level 3, coarse grid, replicated



- `libgrid` for optimised collocate/integrate routines
- ~5-10% speedup typical

epcc | PRACE

CP2K Performance

- Fast Fourier Transforms
  - 1D or 2D decomposition
  - FFTW3 and CuFFT library interface
  - Cache and re-use data
    - FFTW plans, cartesian communicators
- DBCSR
  - Distributed MM based on Cannon's Algorithm
  - Local multiplication recursive, cache oblivious

- `GLOBAL%FFTW_PLAN_TYPE MEASURE | PATIENT`
- `Up to 5% Speedup possible`

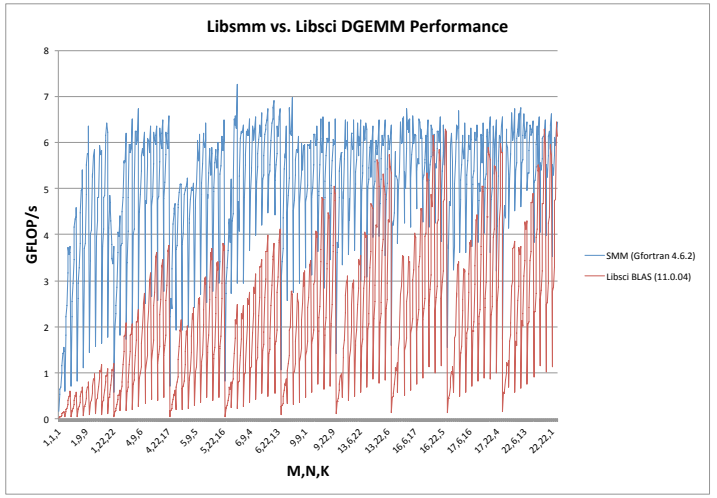
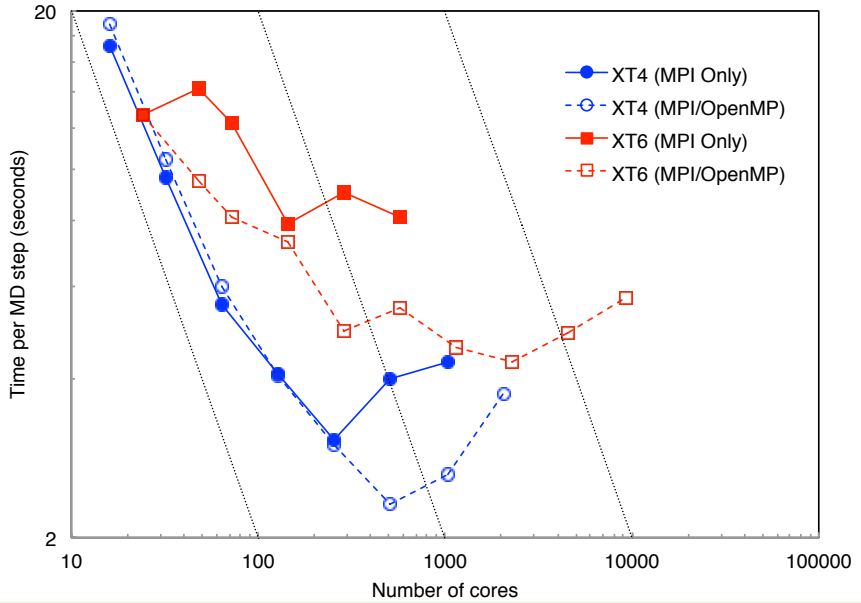


Figure 5: Comparing performance of SMM and Libsci BLAS for block sizes up to 22,22,22

- `libxsmm` for small block multiplications

CP2K Performance

- OpenMP
  - Now in all key areas of CP2K
  - FFT, DBCSR, Collocate/ Integrate, Buffer Packing
  - Incremental addition over time
    - Usually 2 or 4 threads per process
- Dense Linear Algebra
  - Matrix operations during SCF
  - GEMM - ScaLAPACK
  - SYEVD – ScaLAPACK / ELPA

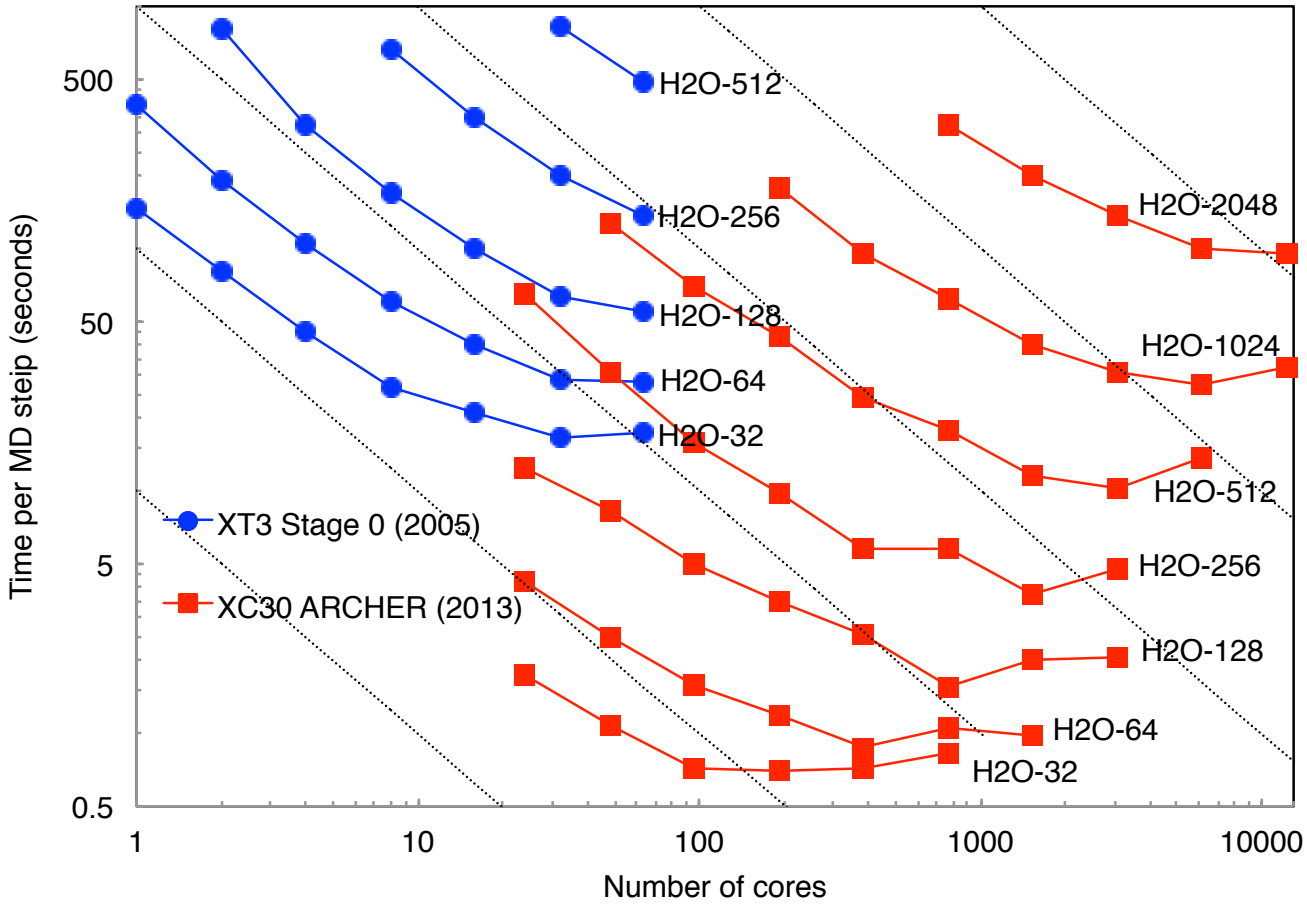


- `-D__ELPA2` and link library to enable
- `GLOBAL %PREFERRED_DIAG_LIBRARY ELPA`
- Up to ~5x Speedup for large, metallic systems

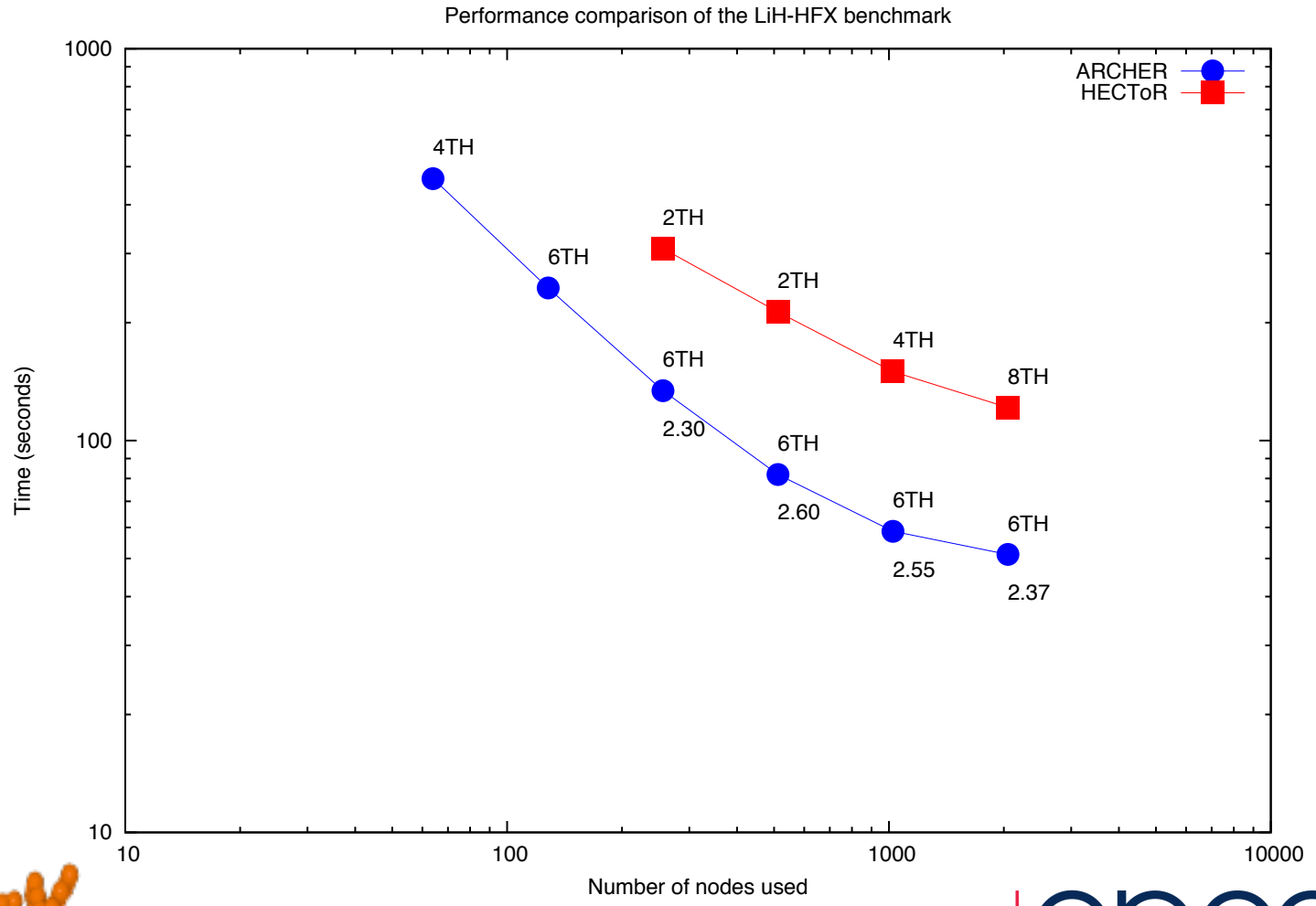
cp2k

epcc | PRACE

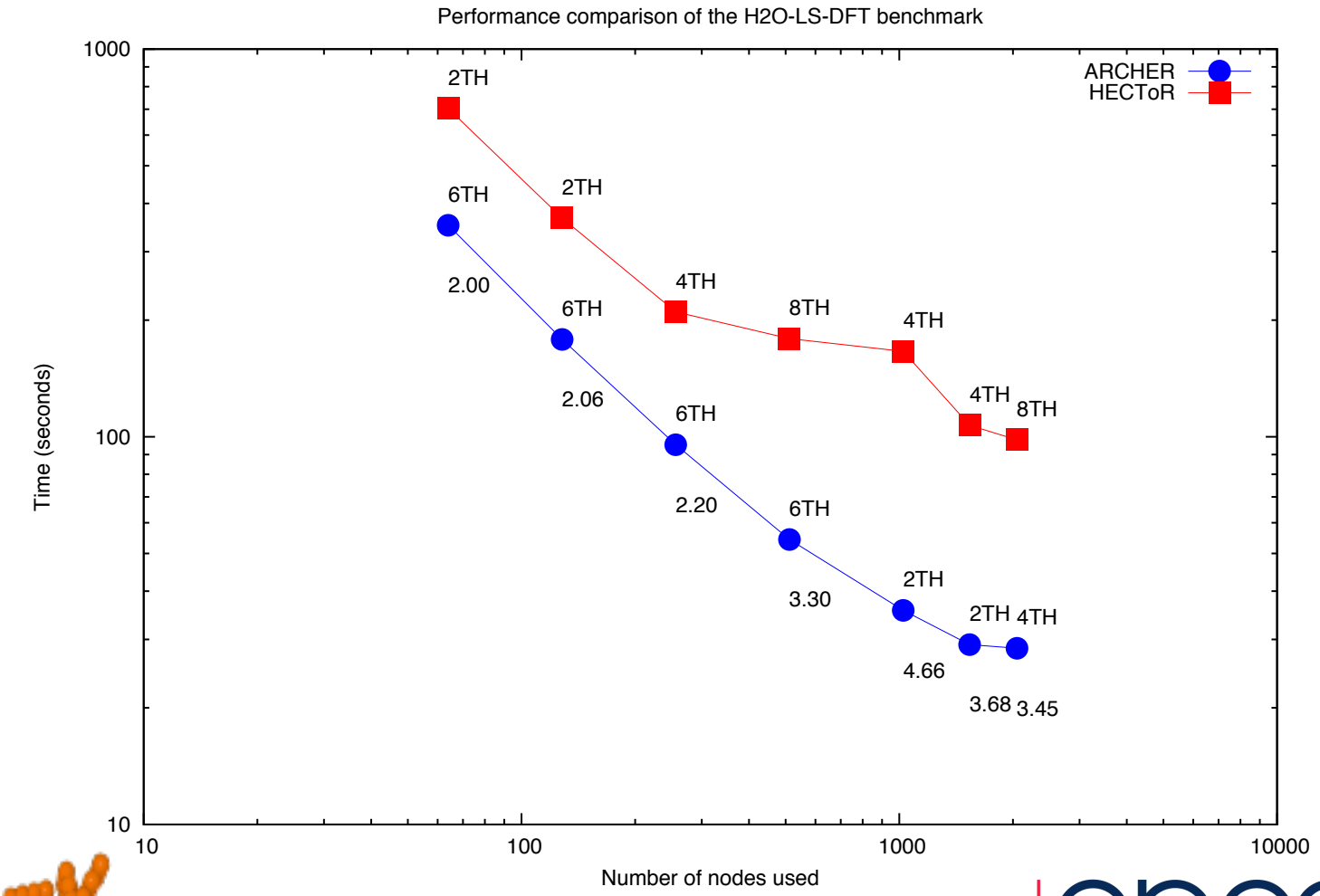
Parallel Performance : H2O-xx



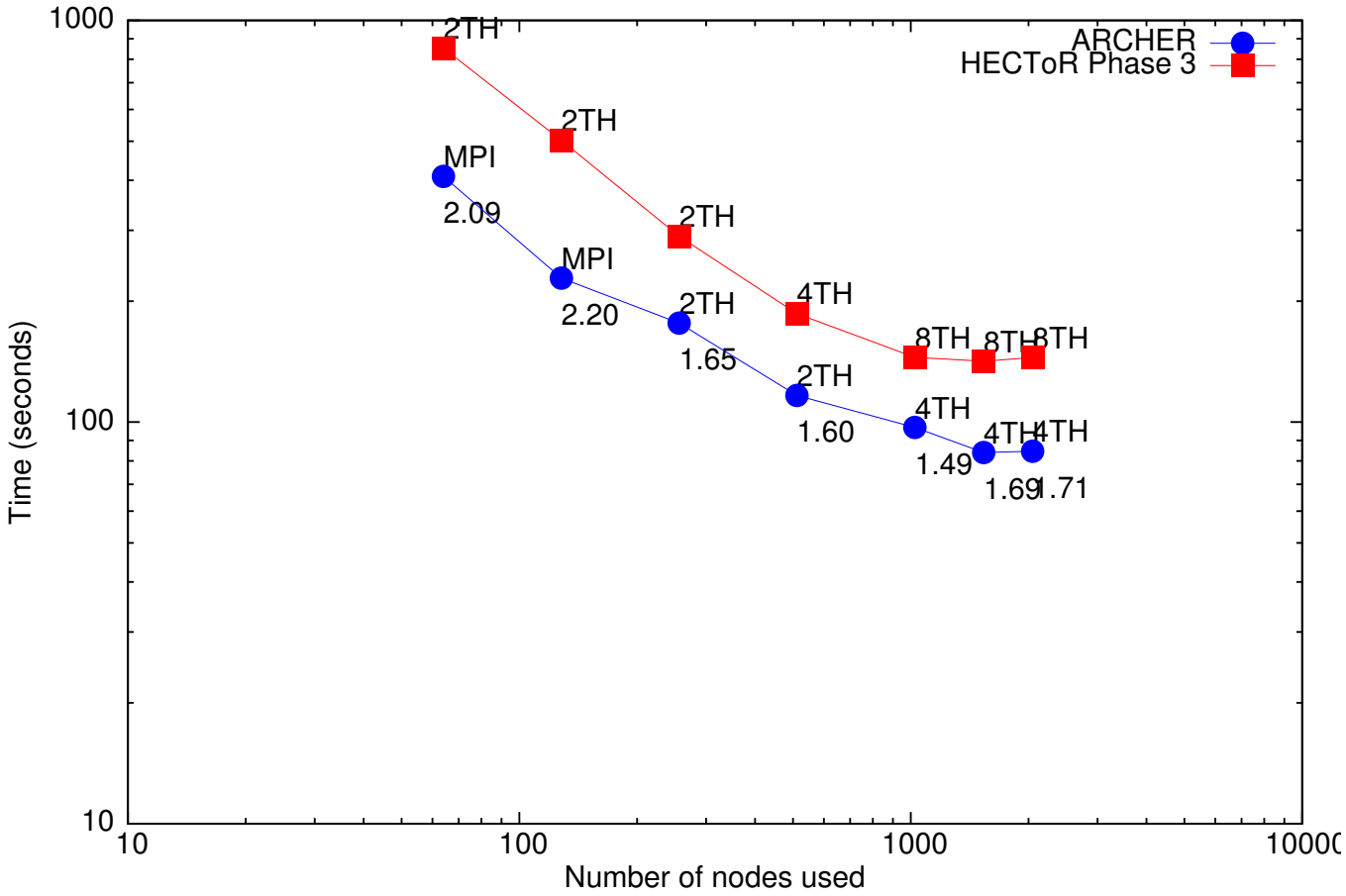
Parallel Performance: LiH-HFX

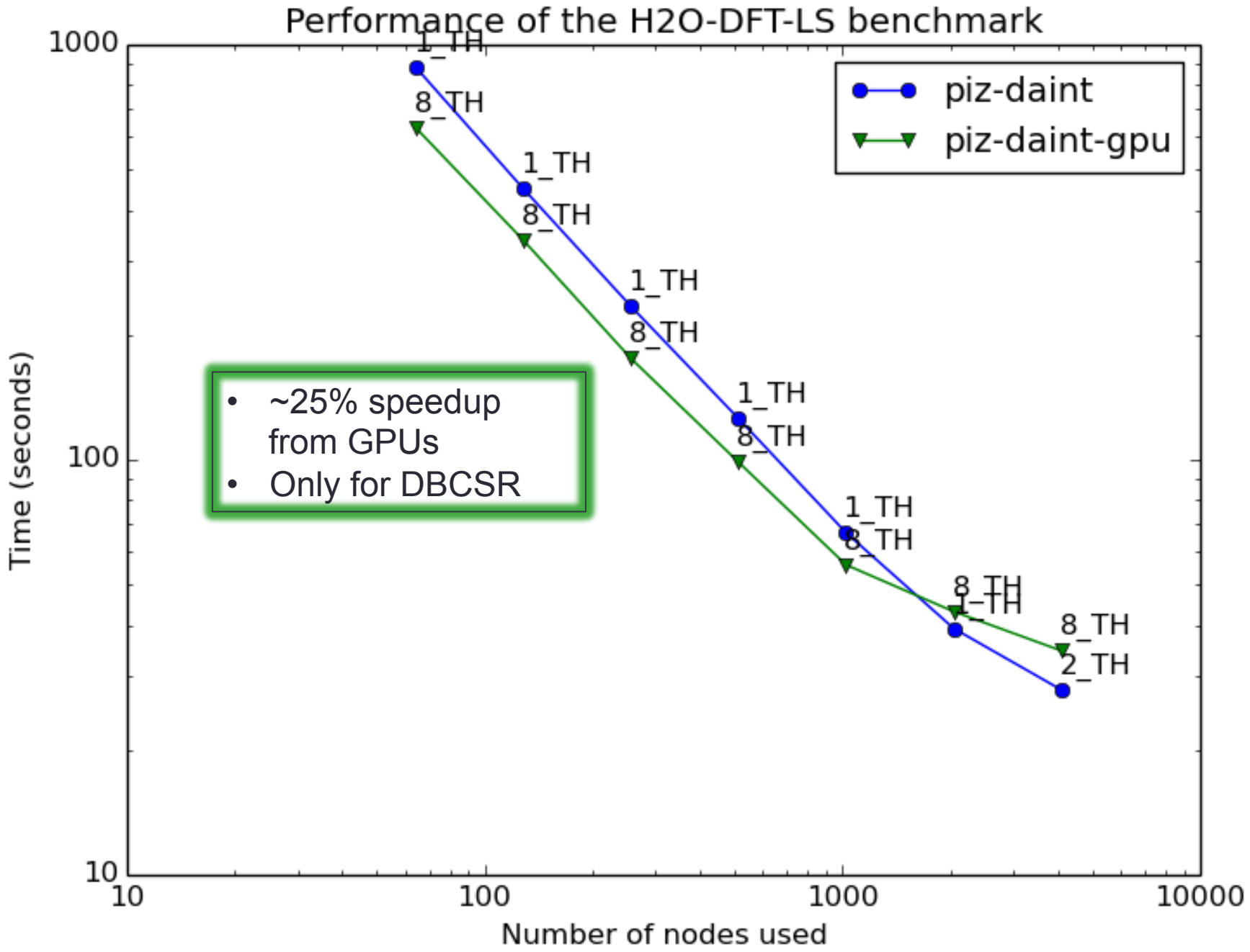


Parallel Performance: H2O-LS-DFT



Parallel Performance: H2O-64-RI-MP2




Performance of the H2O-DFT-LS benchmark
Time (seconds)
Number of nodes used
1000
100
10
10
100
1000
10000
piz-daint
piz-daint-gpu
1_TH
8_TH
2_TH
• ~25% speedup from GPUs
• Only for DBCSR

CP2K Timing Report

```
-------------------------------------------------------------------------------
-                                                                             -
-                                T I M I N G                                  -
-                                                                             -
-------------------------------------------------------------------------------
SUBROUTINE                       CALLS  ASD         SELF TIME        TOTAL TIME
                               MAXIMUM       AVERAGE  MAXIMUM  AVERAGE  MAXIMUM
CP2K                                 1  1.0    0.018    0.018   57.900   57.900
qs_mol_dyn_low                       1  2.0    0.007    0.008   57.725   57.737
qs_forces                           11  3.9    0.262    0.278   57.492   57.493
qs_energies_scf                     11  4.9    0.005    0.006   55.828   55.836
scf_env_do_scf                      11  5.9    0.000    0.001   51.007   51.019
scf_env_do_scf_inner_loop           99  6.5    0.003    0.007   43.388   43.389
velocity_verlet                     10  3.0    0.001    0.001   32.954   32.955
qs_scf_loop_do_ot                   99  7.5    0.000    0.000   29.807   29.918
ot_scf_mini                         99  8.5    0.003    0.004   28.538   28.627
cp_dbcsr_multiply_d               2338 11.6    0.005    0.006   25.588   25.936
dbcsr_mm_cannon_multiply          2338 13.6    2.794    3.975   25.458   25.809
cannon_multiply_low               2338 14.6    3.845    4.349   14.697   15.980
ot_mini                             99  9.5    0.003    0.004   15.701   15.942
-------------------------------------------------------------------------------
```

epcc

PRACE

CP2K Timing Report

- Not just for developers!
  - Check that communication is < 50% of total runtime
  - Check where most time is being spent:
    - Sparse matrix multiplication - `cp_dbcsr_multiply_d`
    - Dense matrix algebra – `cp_fm_syevd` (`&DIAGONALISATION`), `cp_fm_cholesky_*` (`&OT`), `cp_fm_gemm`
    - FFT – `fft3d_*`
    - Collocate / integrate – `calculate_rho_elec`, `integrate_v_rspace`
- Control level of granularity

```
&GLOBAL
  &TIMINGS
    THRESHOLD 0.00001
  &END TIMINGS
&END GLOBAL
```

Default is 0.02 (2%)

epcc PRACE

CP2K Performance Summary

- First look for algorithmic gains
  - Cell size, SCF settings, preconditioner, choice of basis set, QM/MM, ADMM…
- Check scaling of *your* system
  - Run a few MD steps / reduced `MAX_SCF`
- Almost all performance-critical code is in libraries
  - Compiler optimisation –O3 is good enough
  - Intel vs gfortran vs Cray – difference is close to zero
- Before spending 1,000s of CPU hours, build libxsmm, libgrid, ELPA, FFTW3…
  - Or ask your local HPC support team ☺

epcc | PRACE

Practical Exercises

- Exercises are all on the web
- Larger data files available on Fionn:
  - `/ichec/home/users/ibethune/exercises`
- Range of exercises depending on what you are interested in!

For CP2K beginners

- Short ‘HowTo’ exercises on various basic functions of CP2K:
- Single-point energy & force calculation using DFT
  - https://www.cp2k.org/howto:static_calculation
- How to converge the total energy w.r.t. the CUTOFF and REL_CUTOFF
  - https://www.cp2k.org/howto:converging_cutoff
- How to run geometry optimisation
  - https://www.cp2k.org/howto:geometry_optimisation

epcc PRACE

Intermediate Exercises

- ‘Surface Science’ using local DFT
  - https://www.cp2k.org/exercises:2015_pitt:gga
- Running *ab initio* MD of liquied water
  - https://www.cp2k.org/exercises:2015_pitt:aimd
- Hybrid functional calculations and dispersion corrections
  - https://www.cp2k.org/exercises:2015_pitt:hfx
- Linear Scaling DFT
  - https://www.cp2k.org/exercises:2015_pitt:ls
- Electron correlation: MP2 and RPA
  - https://www.cp2k.org/exercises:2015_pitt:mp2

epcc | PRACE

Extended Exercises

- Metadynamics calculations
  - https://www.cp2k.org/exercises:2015_cecam_tutorial:mtd1
- QM/MM of Urea in water
  - https://www.cp2k.org/exercises:2015_cecam_tutorial:urea
- Adsorption on metallic surfaces (Nudged Elastic Band)
  - https://www.cp2k.org/exercises:2015_cecam_tutorial:neb
- Force-field calculations on a protein
  - https://www.cp2k.org/exercises:2015_cecam_tutorial:forcefields
- Also VIBRATIONAL_ANALYSIS, NMR, X-Ray, DFT+U
  - In `/ichec/home/users/ibethune/exercises`

epcc | PRACE

Scaling Tests

- Several benchmark systems are provided at:
  - https://www.cp2k.org/performance

- Suggested experiments:
  - Explore the effects of simulation size, accuracy parameters etc. on performance

  - Try out performance ‘tweaks’

epcc PRACE

Bring-your-own system

- Convert a simulation from another code to CP2K
- Compare accuracy and performance
- Ask me for help!

epcc PRACE