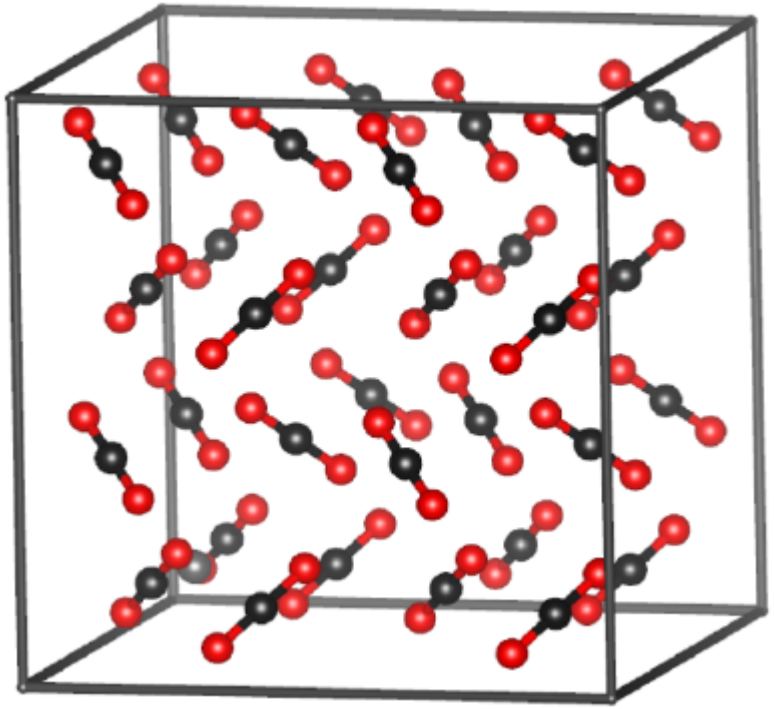
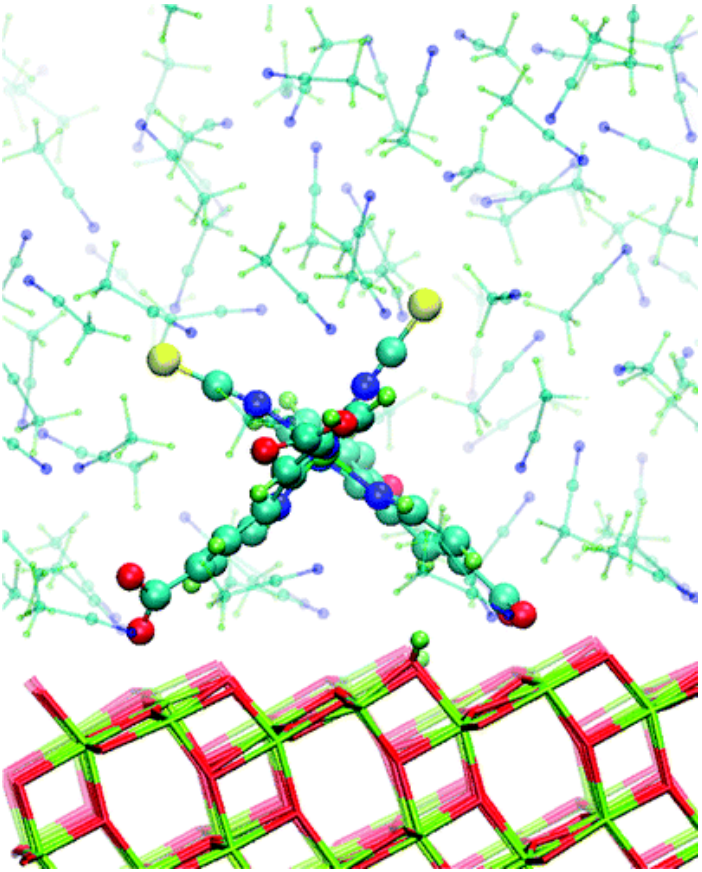
Large scale and hybrid computing with CP2K

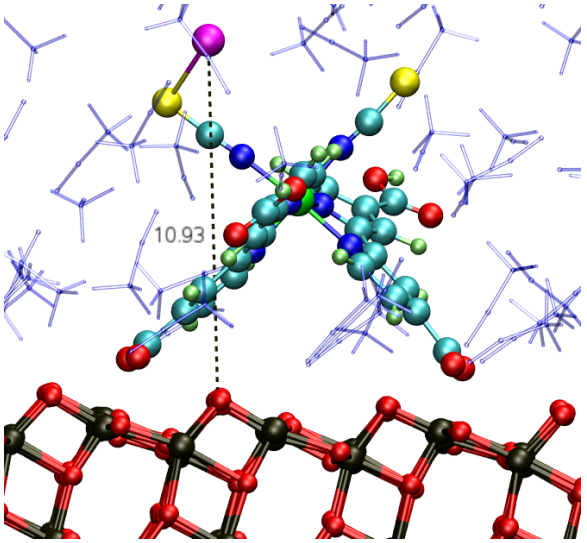


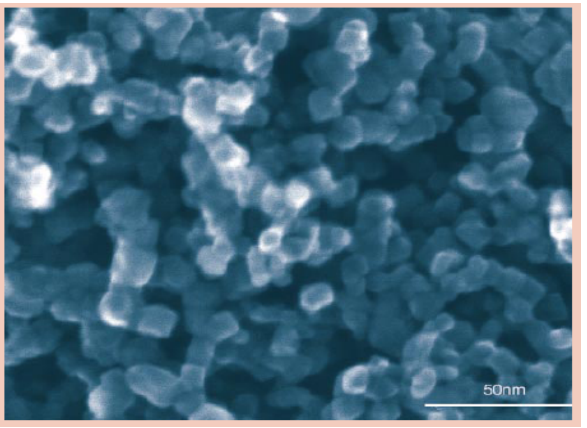


Joost VandeVondele
Nanoscale Simulations, ETH Zurich

Sustainable energy production

A grand challenge for society which will require novel materials engineered with atomistic precision

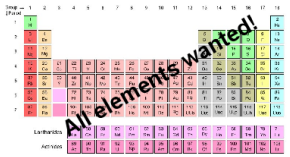


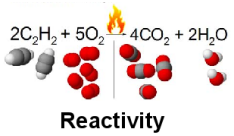


Grätzel, Nature (1991,2001)

Novel simulation tools can contribute towards rational design of complex systems

Modeling complex systems



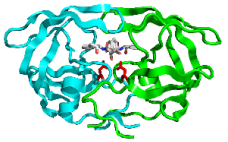


Reactivity

Needs a versatile theory...

Electronic properties

Extreme conditions



Weak interactions

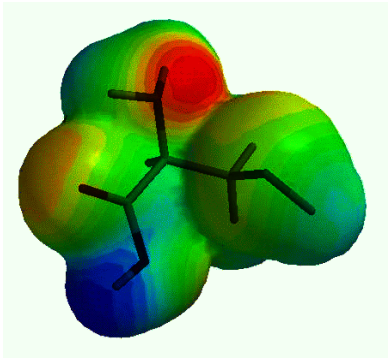
$$\left[-\frac{\hbar^2}{2m}\nabla^2 + V_s(\vec{r})\right]\phi_i(\vec{r}) = \epsilon_i\phi_i(\vec{r})$$

$$V_s(\vec{r}) = V(\vec{r}) + \int \frac{e^2 n_s(\vec{r}\,')}{|\vec{r} - \vec{r}\,'|}\mathrm{d}^3r' + V_{\mathrm{XC}}[n_s(\vec{r})]$$

Density functional theory (DFT) & computational methods (1998, Nobel Prize, Walter Kohn- John A. Pople)

A computationally tractable reference system of non-interacting particles can be used to obtain all properties (in particular the density and energy) of a true electronic system. The required external potential (Vxc) exists but is only known approximately.

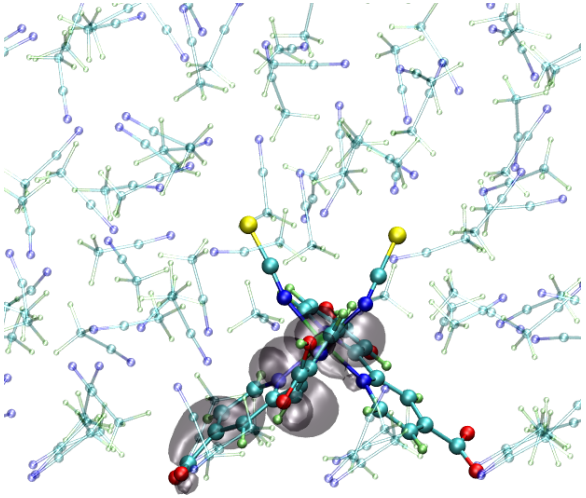
Modeling complex systems

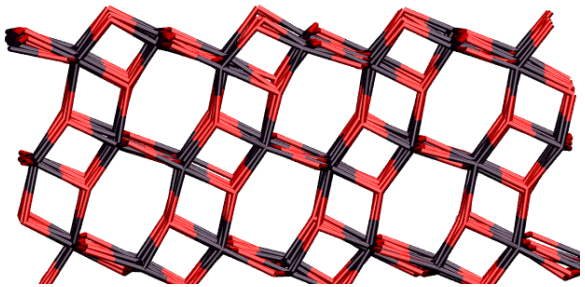


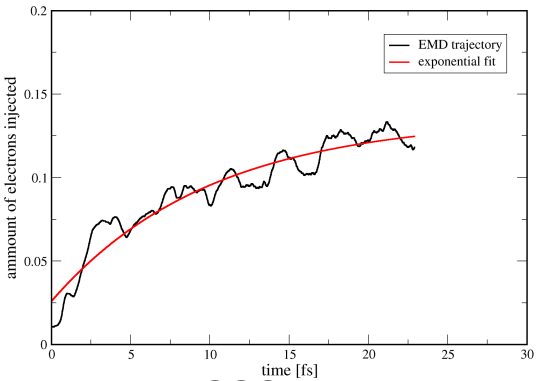
© nobelprize.org (1998)

~10 atoms
static

scale





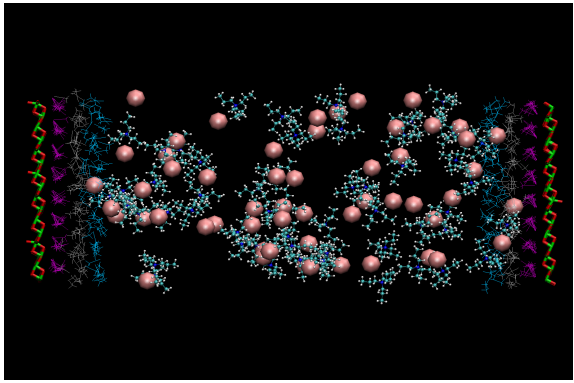


~1'000 atoms
~1-100 ps

Multi-scale

SIZE

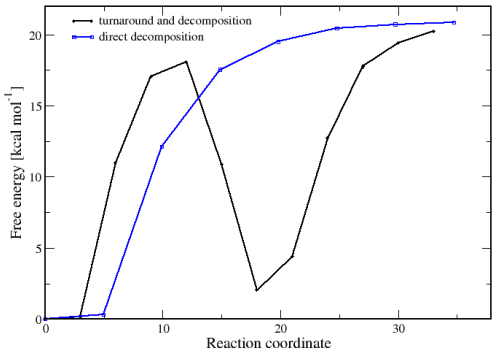
Empirical models



~100'000 atoms
~1-1000 ns

TIME

Enhanced sampling



Reduced dimensionality
All timescales

What is CP2K ?

CP2K is a freely available 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) [...]

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + V_s(\vec{r})\right]\phi_i(\vec{r}) = \epsilon_i\phi_i(\vec{r})$$

$$V_s(\vec{r}) = V(\vec{r}) + \int \frac{e^2 n_s(\vec{r}\,')}{|\vec{r} - \vec{r}\,'|}\mathrm{d}^3r' + V_{\mathrm{XC}}[n_s(\vec{r})]$$

On our web page since the initial version in 2004-10-16

Gaussian and plane waves: GPW in CP2K

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + V_s(\vec{r})\right]\phi_i(\vec{r}) = \epsilon_i\phi_i(\vec{r})$$

- Primary basis: Gaussians → Chemistry
  - compact
  - sparse H^{ks} (and P)
  - Many terms analytic
- Auxiliary basis: Plane waves → Physics
  - regular grid for e^- density
  - FFT for Poisson equation
  - No four center integrals needed

The GPW algorithm : compute the GGA Kohn-Sham matrix in O(N) time, PBC are natural.

J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing and J. Hutter, Comp. Phys. Comm. 167, 103 (2005).
Lippert, G; Hutter, J; Parrinello, M. Mol. Phys., 92 (3), 477-487 (1997).

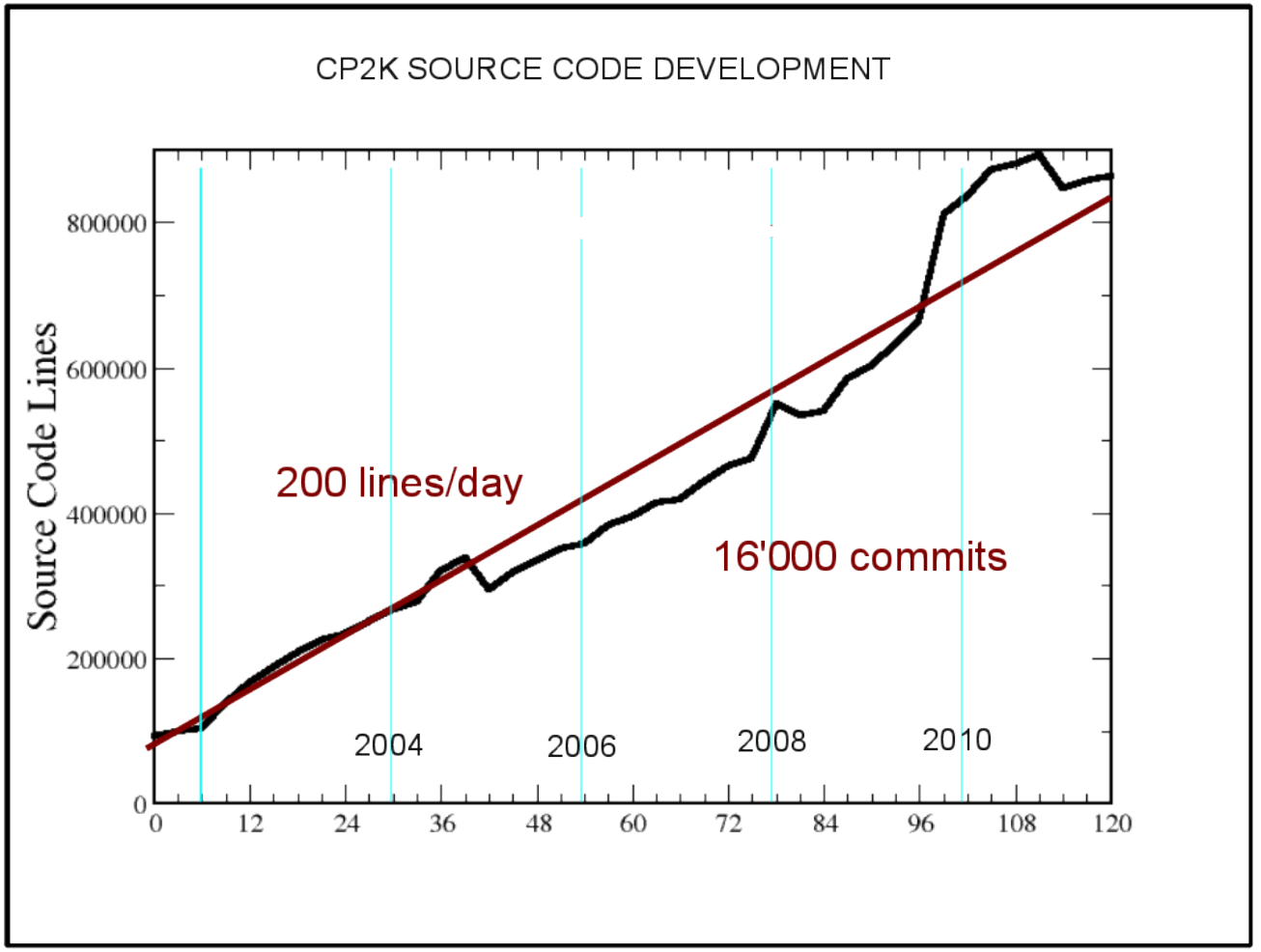
CP2K: the swiss army knife of molecular simulation

- A wide variety of models Hamiltonians
  - classical
  - semi-empirical
  - local and non-local DFT
  - Combinations (e.g. QM/MM)
- Various algorithms
  - Molecular dynamics & Monte Carlo
    - NVE, NVT, NPT
  - Free energy and PES tools
  - Ehrenfest MD
- Properties
  - Vibrational
  - NMR, EPR, XAS, TDDFT
- Open source & rapid development
  - 900.000 lines of code

CP2K: the swiss army knife of molecular simulation

- A wide variety of models Hamiltonians
  - classical
  - semi-empirical
  - local and non-local DFT
  - Combinations (e.g. QM/MM)
- Various algorithms
  - Molecular dynamics & Monte Carlo
    - NVE, NVT, NPT
  - Free energy and PES tools
  - Ehrenfest MD
- Properties
  - Vibrational
  - NMR, EPR, XAS, TDDFT
- Open source & rapid development
  - 700.000 lines of code

What is CP2K: team



With contributions from: Axel Kohlmeyer Barbara Kirchner Ben Slater Chris Mundy Fawzi Mohamed Florian Schiffmann Gerald Lippert Gloria Tabacchi Greg Schenter Harald Forbert Iain Bethune William Kuo Ken Bagchi Sundaram Balasubramanian Jochen Schmidt Jens Thar Jürg Hutter Matthias Krack Matt Watkins Marcella Iannuzzi Manuel Guidon Matthew McGrath Thomas Chassaing Thomas Heine Thomas Kuehne Teodoro Laino Urban Borstnik Joost VandeVondele Benjamin Levine Luca Bellucci Ari Seitsonen Louis Vanduyfhuys Mathieu Salanne Michele Ceriotti Lukasz Walewski Michael Steinlechner Rodolphe Vuilleumier Sebastiano Caravati Valery Weber Kevin Stratford Toon Verstraelen Marcel Baer Alessandro Laio Stefan Goedecker Luigi Genovese Thierry Deutsch Dieter Gleich Reinout Declerck Kurt Baarman Mauro DelBen Mandes Schönherr Yannik Misteli Fabio Sterpone Gerd Berghold Pietro Ballone Walter Silvestri Pekka Manninen Francois-Xavier Coudert Christiane Pousa Michele Parrinello Michiel Sprik Ilja Siepmann Michel Waroquier

Ten years of development at 200 lines / day = O(1M) SLOC, with a large base of contributors

How do we collaborate ?

- SVN
- Email
- Meet & Talk

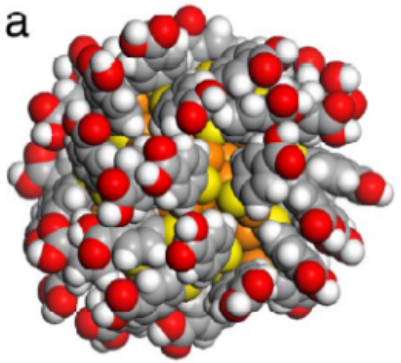
An animated view of SVN history

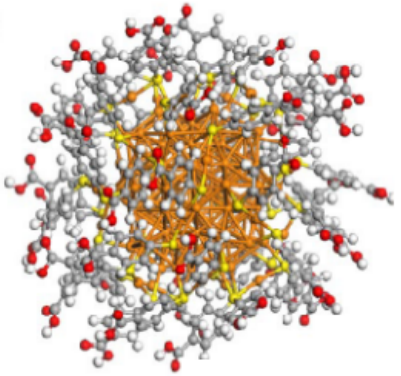
CP2K: science (I)

A unified view of ligand-protected gold clusters as superatom complexes

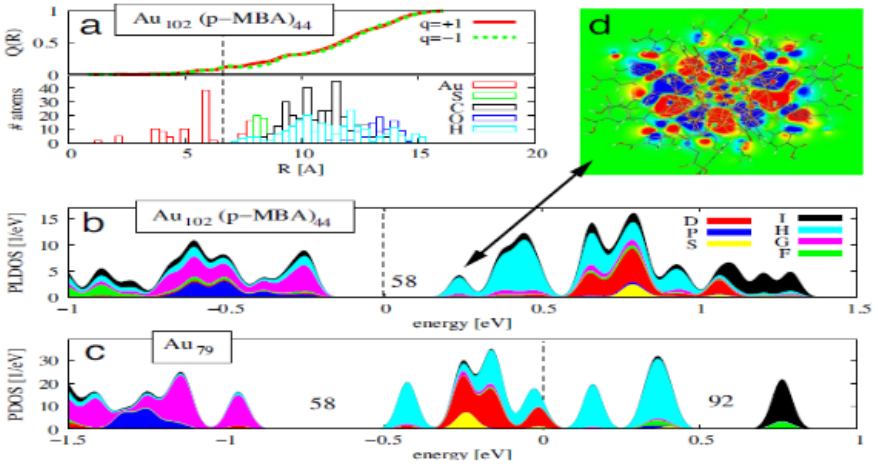
Michael Walter[†], Jaakko Akola[†‡], Olga Lopez-Acevedo[†], Pablo D. Jadzinsky[§¶], Guillermo Calero[§], Christopher J. Ackerson[§‖], Robert L. Whetten[††], Henrik Grönbeck[‡‡], and Hannu Häkkinen[†§§¶¶]

PNAS July 8, 2008 vol. 105 no. 27 9157–9162





System size: 762 Atoms
~3400 electrons



Electronic structure of nanoparticles

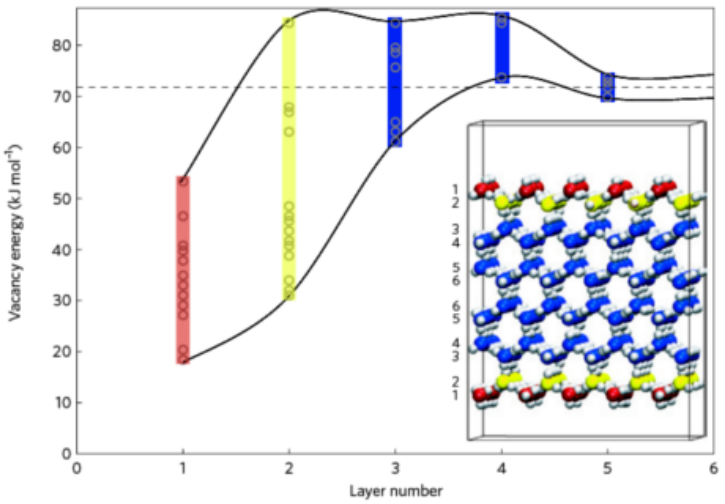
CP2K: science (II)

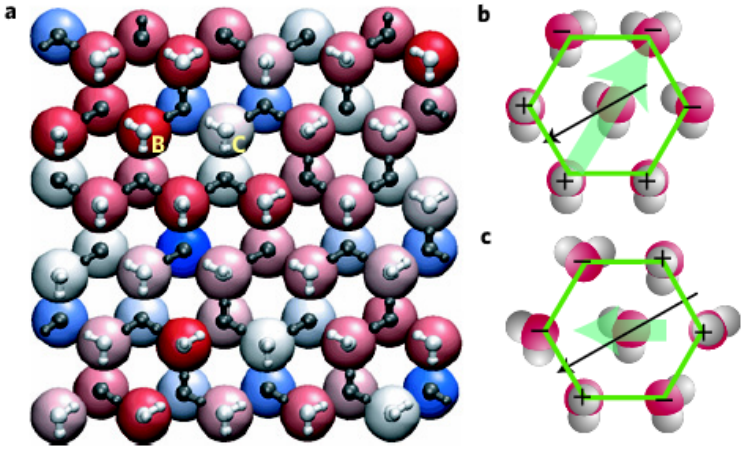
Large variation of vacancy formation energies in the surface of crystalline ice

M. Watkins[1,2,3], D. Pan[4], E. G. Wang[5], A. Michaelides[1,2,3], J. VandeVondele[6] and B. Slater[1,3]*

[1]Department of Chemistry, Christopher Ingold Building, 20 Gordon Street, University College London, London WC1H 0AJ, UK, [2]London Centre for Nanotechnology, University College London, London WC1H 0AJ, UK, [3]TYC@UCL, University College London, London WC1H 0AJ, UK, [4]Institute of Physics, Chinese Academy of Sciences, PO Box 603, Beijing 100190, China, [5]School of Physics, Peking University, Beijing 100871, China, [6]Institute of Physical Chemistry, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland. *e-mail: b.slater@ucl.ac.uk.

NATURE MATERIALS | VOL 10 | OCTOBER 2011



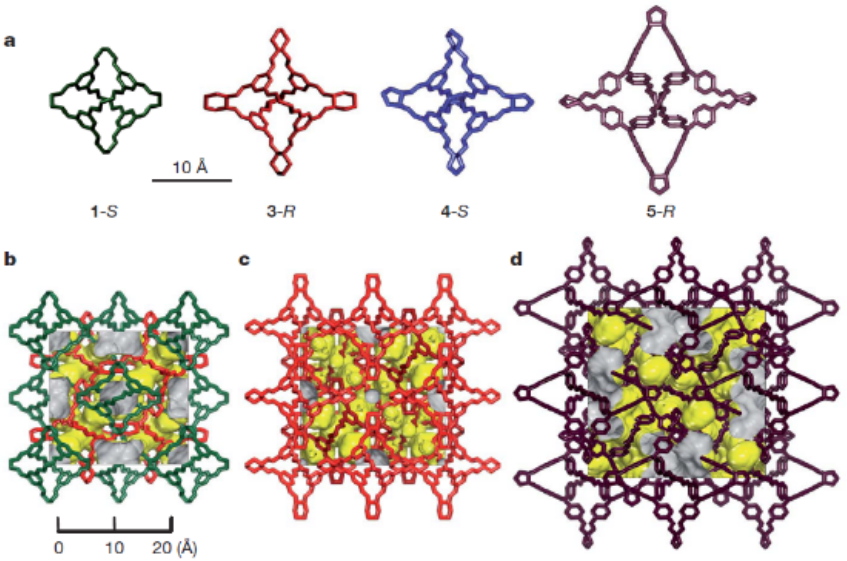


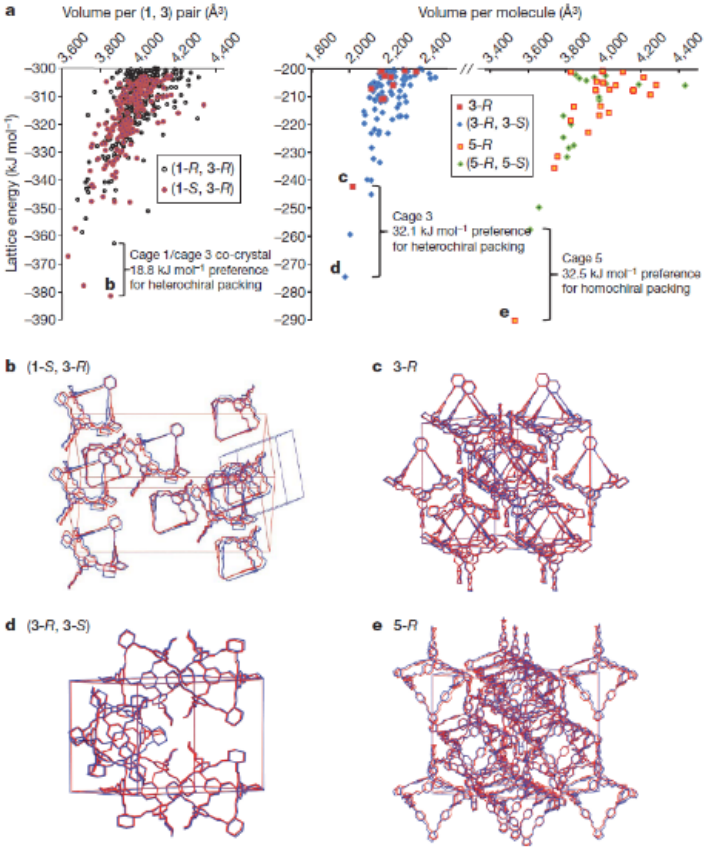
CP2K: science (III)

Modular and predictable assembly of porous organic molecular crystals

James T. A. Jones[1], Tom Hasell[1], Xiaofeng Wu[1], John Bacsa[1], Kim E. Jelfs[1], Marc Schmidtmann[1], Samantha Y. Chong[1], Dave J. Adams[1], Abbie Trewin[1], Florian Schiffman[2], Furio Cora[2], Ben Slater[2], Alexander Steiner[1], Graeme M. Day[3] & Andrew I. Cooper[1]

16 JUNE 2011 | VOL 474 | NATURE | 367





Structure prediction of metal organic frameworks

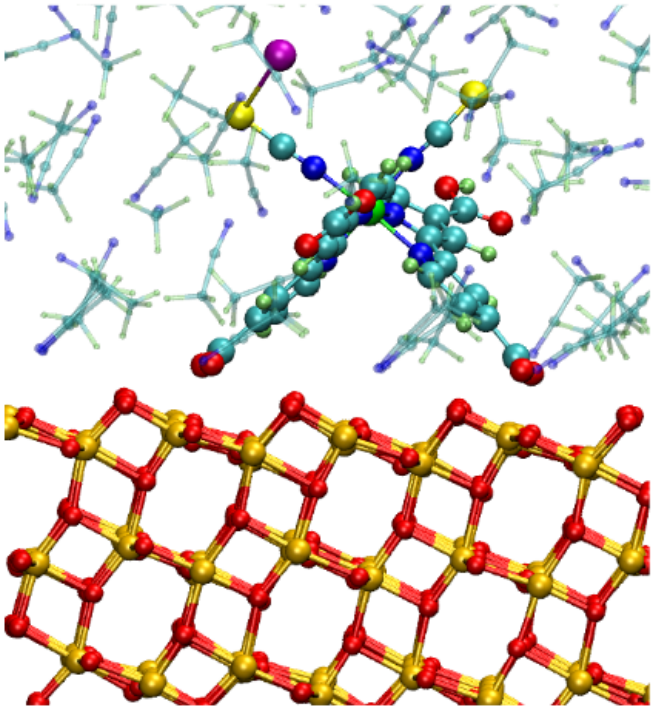
CP2K: science (IV)

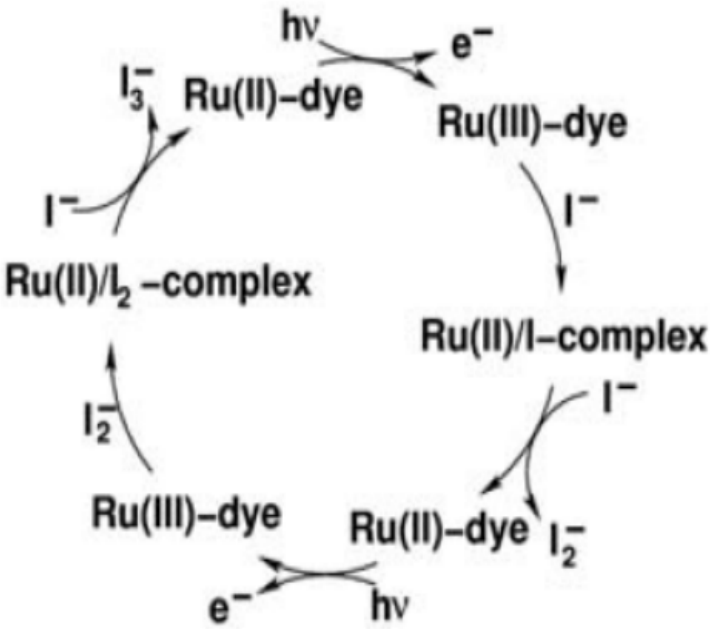
An atomistic picture of the regeneration process in dye sensitized solar cells

Florian Schiffmann[a], Joost VandeVondele[a,1], Jürg Hutter[a], Atsushi Urakawa[b], Ronny Wirz[b], and Alfons Baiker[b]

[a]Institute of Physical Chemistry, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland; and [b]Department of Chemistry and Applied Biosciences, Institute for Chemical and Bioengineering, ETH Zurich, Hönggerberg, HCI, 8093 Zurich, Switzerland

4830–4833 | PNAS | March 16, 2010 | vol. 107 | no. 11



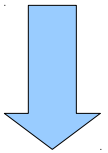


Functionalized solid/liquid interfaces

CP2K: algorithms

The power & challenge of CP2K:
a wide range of algorithms
with good scaling properties

- Regular grids: halo-exchange, 3D FFT, Poisson solver, multigrids
- Dense Linear Algebra: Multiply, Diagonalization, Cholesky,...
- Sparse Linear Algebra: Matrix Multiply
- Particles: time integration, Monte Carlo sampling
- Chemical: 4 center integrals, HFX, MP2, XC, ...



A single kernel rarely dominates
Scaling is O(N)...O(N**5)

a) Depending on method and system size various kernels will dominate
b) Depending on the dominating kernel, particular hardware might be suitable
c) If several kernels dominate optimization is more of a challenge

Modern Compute Resources for atomistic simulation

Perform calculations that

+/- complete faster
- AIMD: at the limit of strong scaling (network)
+ Ensemble simulations: connect to experiment

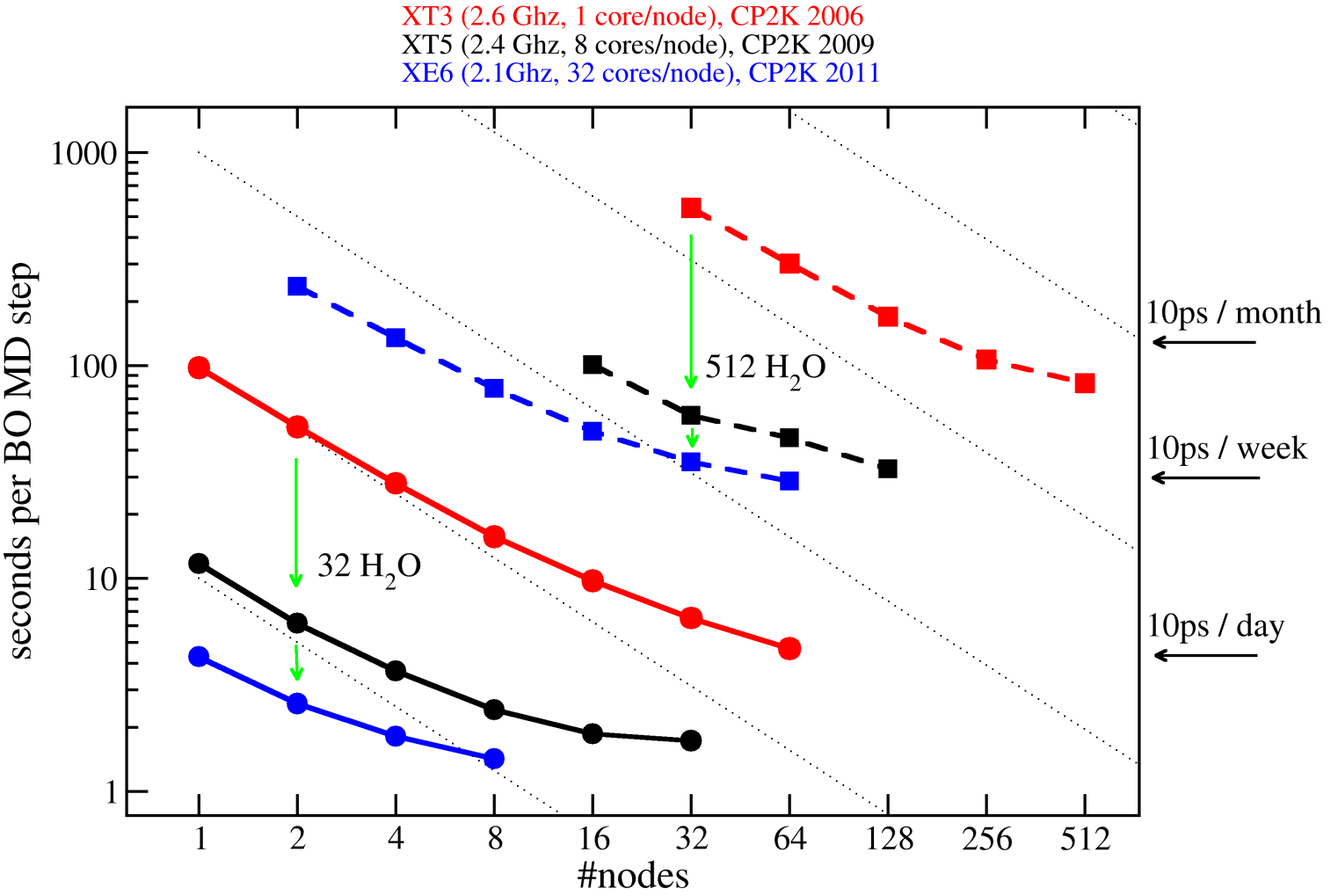
+ are cheaper

+ are technically better converged (grids, basis, sampling)

+ are based on more accurate theory

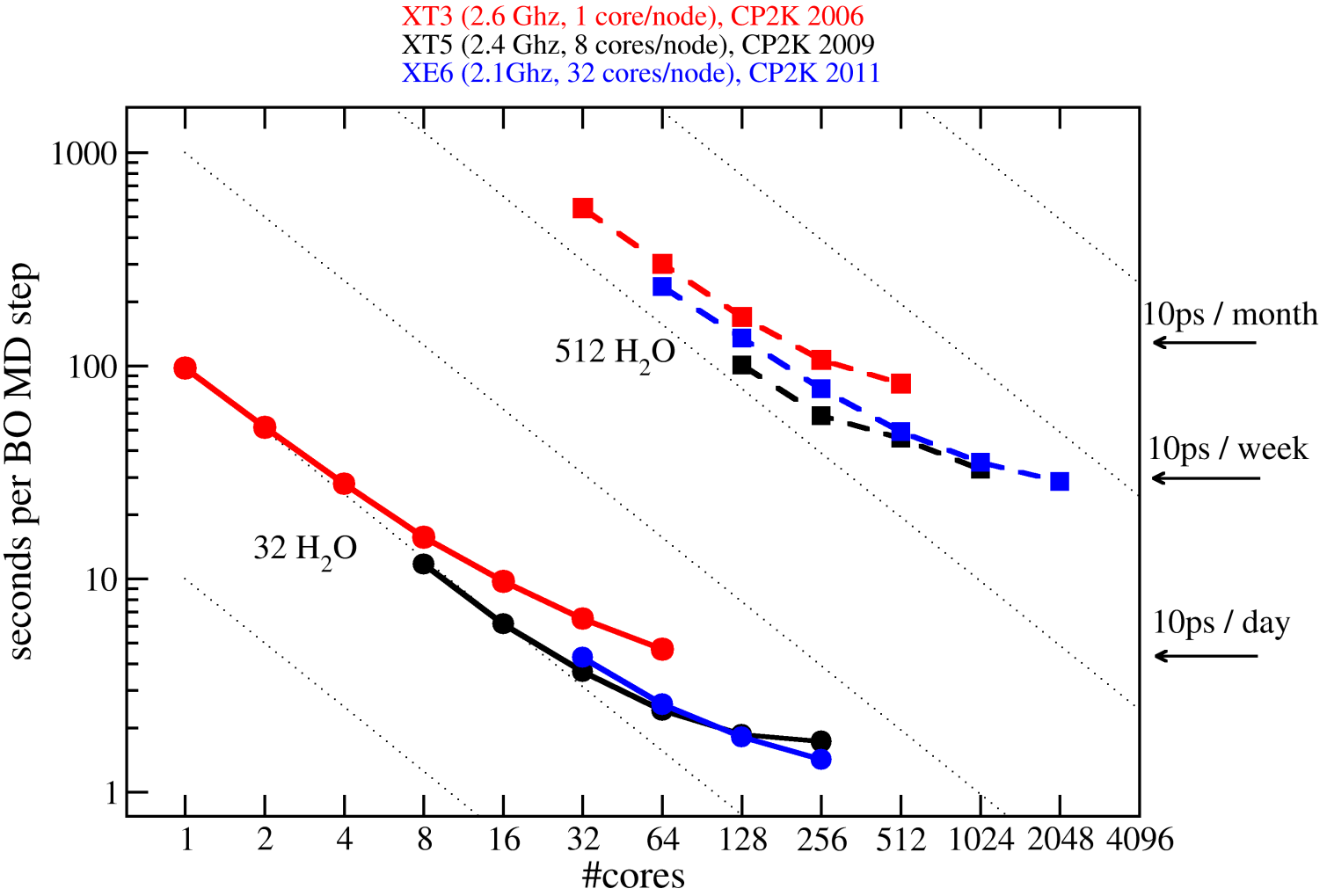
+ are based on larger models

CP2K on CSCS production hardware



For a typical user benchmark, the per node (~ per Watt) performance has improved significantly.

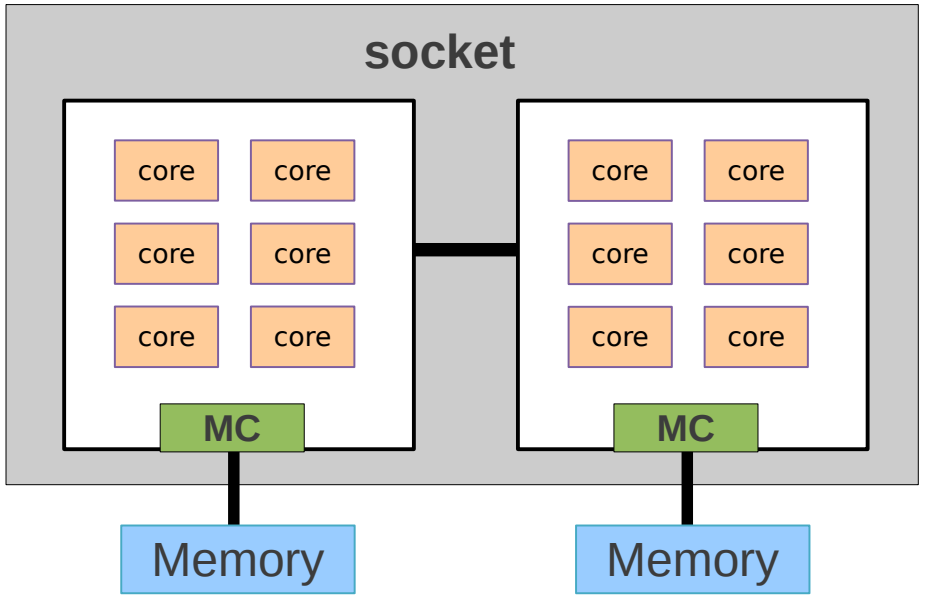
CP2K on CSCS production hardware



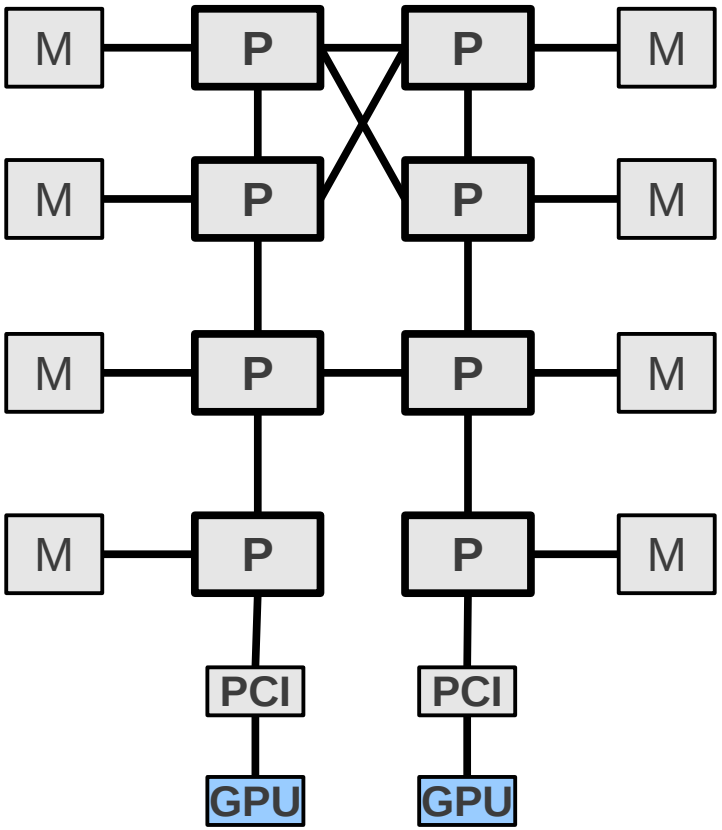
Per core, we can fight the clock-frequency trend....
XE6 in the above graph is 32 cores per node.

Complex hardware: NUMA nodes with PCI devices

NUMA Within Socket

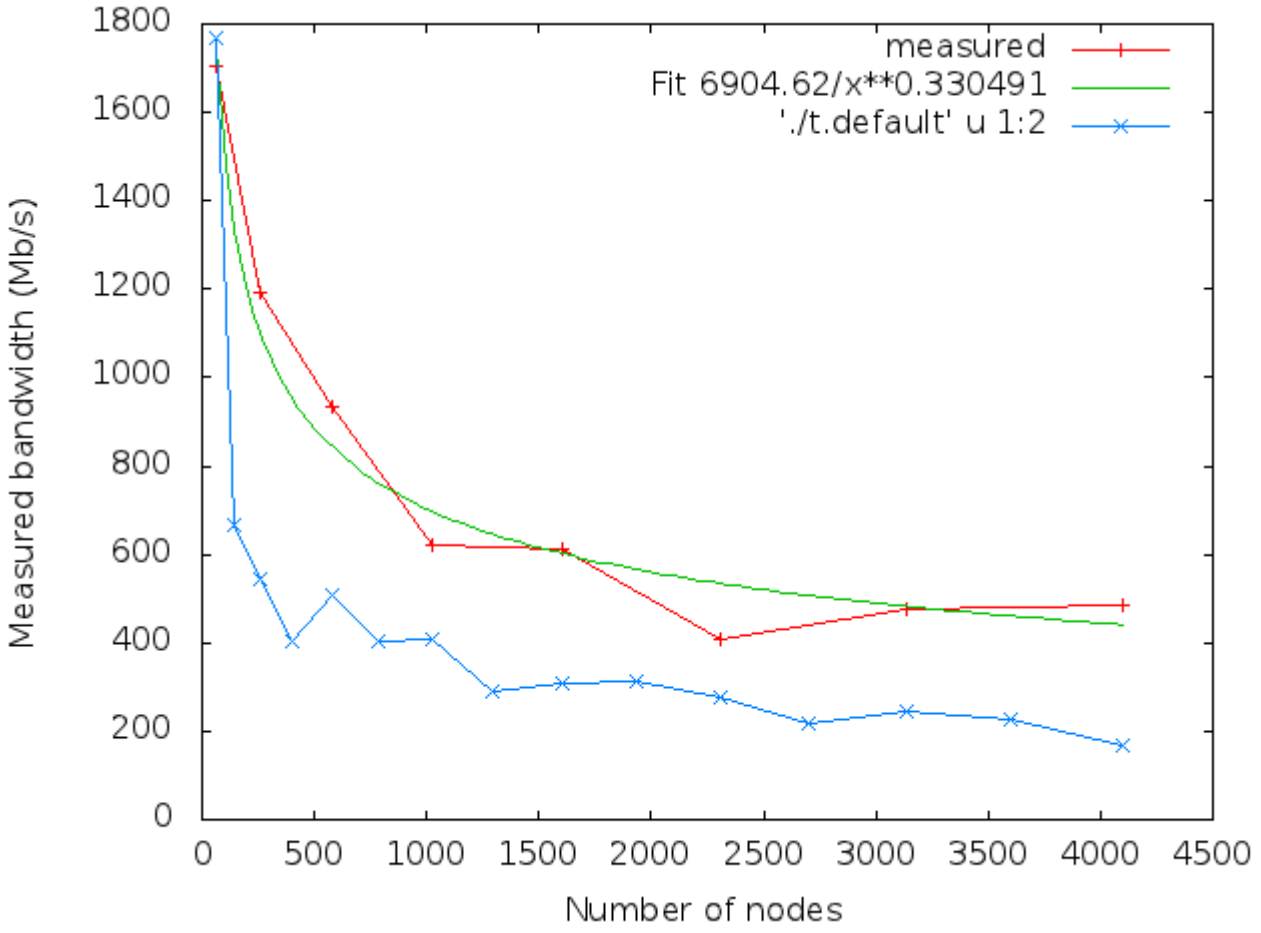


PCI devices (network / GPU)



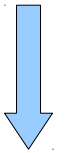
Should we leave thread placement,
Task placement and memory management to the system ?

Complex hardware: Network topology



3D Torus on the XK6

'Nearest neighbor comms on a 2D grid'



Need to think about process layout or we need better networks.

Entropy rules: sampling

Entropy and free energy are absolutely important properties, and required to make contact with experiment.

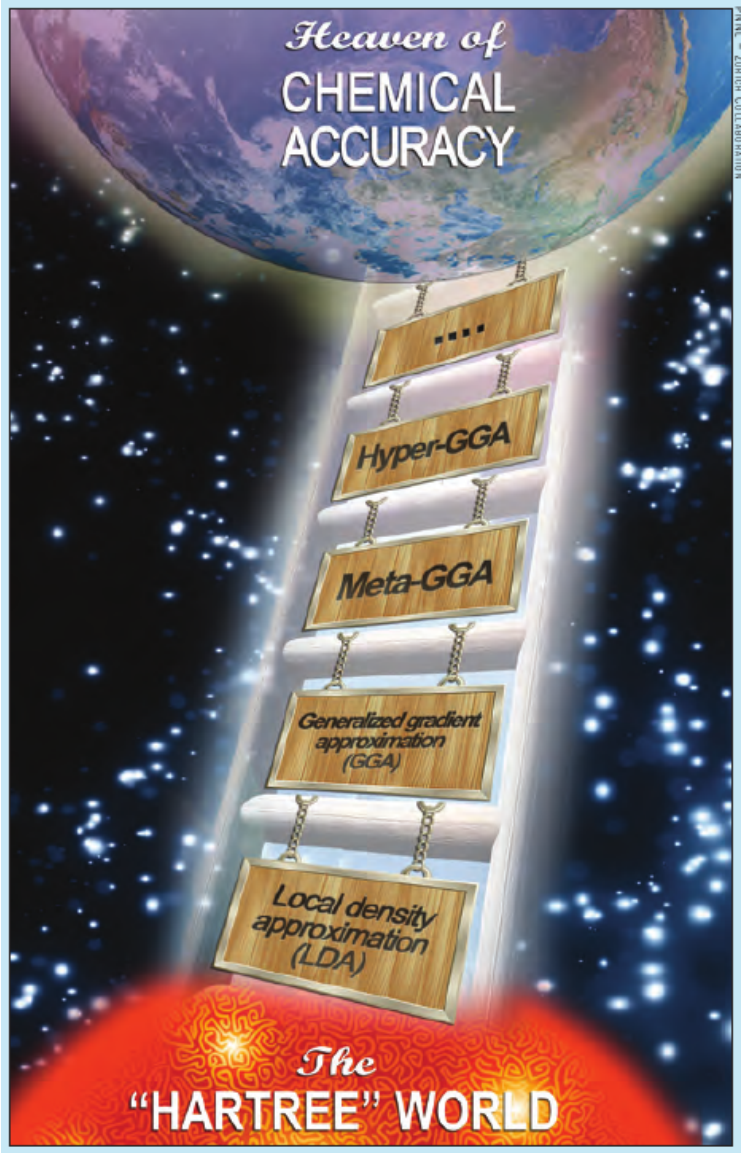
They can be computed by generating ensembles of configurations (dynamics / Monte Carlo)

High throughput computing is the best way to achieve good sampling and converged statistics.

More accurate theory:

From 'GGA' to 'HFX' to 'MP2'

Exchange & correlation functionals



$$V_s(\vec{r}) = V(\vec{r}) + \int \frac{e^2 n_s(\vec{r}\,')}{|\vec{r} - \vec{r}\,'|} \mathrm{d}^3 r' + V_{\mathrm{XC}}[n_s(\vec{r})]$$

Exchange and correlation functionals of improving can be constructed by adding new ingredients.

Each rung on the ladder improves accuracy, but also increases complexity

1) GGA: only relies on the electron density and its gradients (semi-local)

2) Hyper-GGA: hybrid functionals, includes density and the single particle orbitals directly in a non-local way through Hartree-Fock exchange (HFX)

3) Double hybrids: include some MP2-like terms

Mundy, Kathmann, Rousseau, Schenter, VandeVondele, Hutter, SCIDAC reviews (spring 2010).

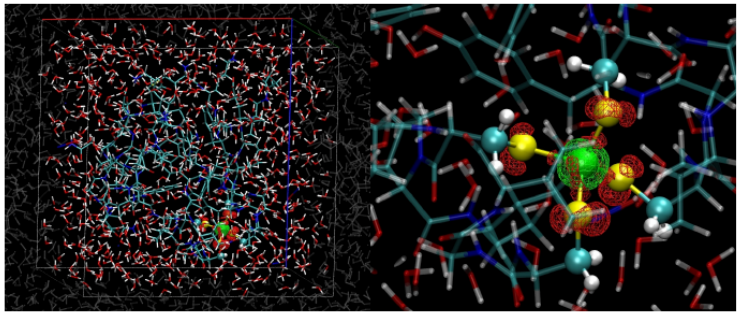
Hartree-Fock exchange

$$E_x^{\mathbf{HF}} = -\frac{1}{2} \sum_{\alpha\beta\gamma\delta} P_{\alpha\beta} P_{\gamma\delta} (\phi_\alpha \phi_\gamma | \phi_\beta \phi_\delta)$$

$$(\phi_\alpha \phi_\gamma | \phi_\beta \phi_\delta) = \int d\mathbf{r} d\mathbf{r}' \frac{\phi_\alpha(\mathbf{r}) \phi_\gamma(\mathbf{r}) \phi_\beta(\mathbf{r}') \phi_\delta(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}$$

An easy term in Gaussian basis sets, but brute force scaling as $O(N^4)$

2'825 atoms
→ 31'247 basis functions
→ 976'375'009 elements in P
→ 953'308'158'199'750'081 integrals
Exa-Pet-Ter-Gig-Meg-Kil



Do we need exa-scale computing, or can we be scientist ?

$O(N^4) \rightarrow O(N)$

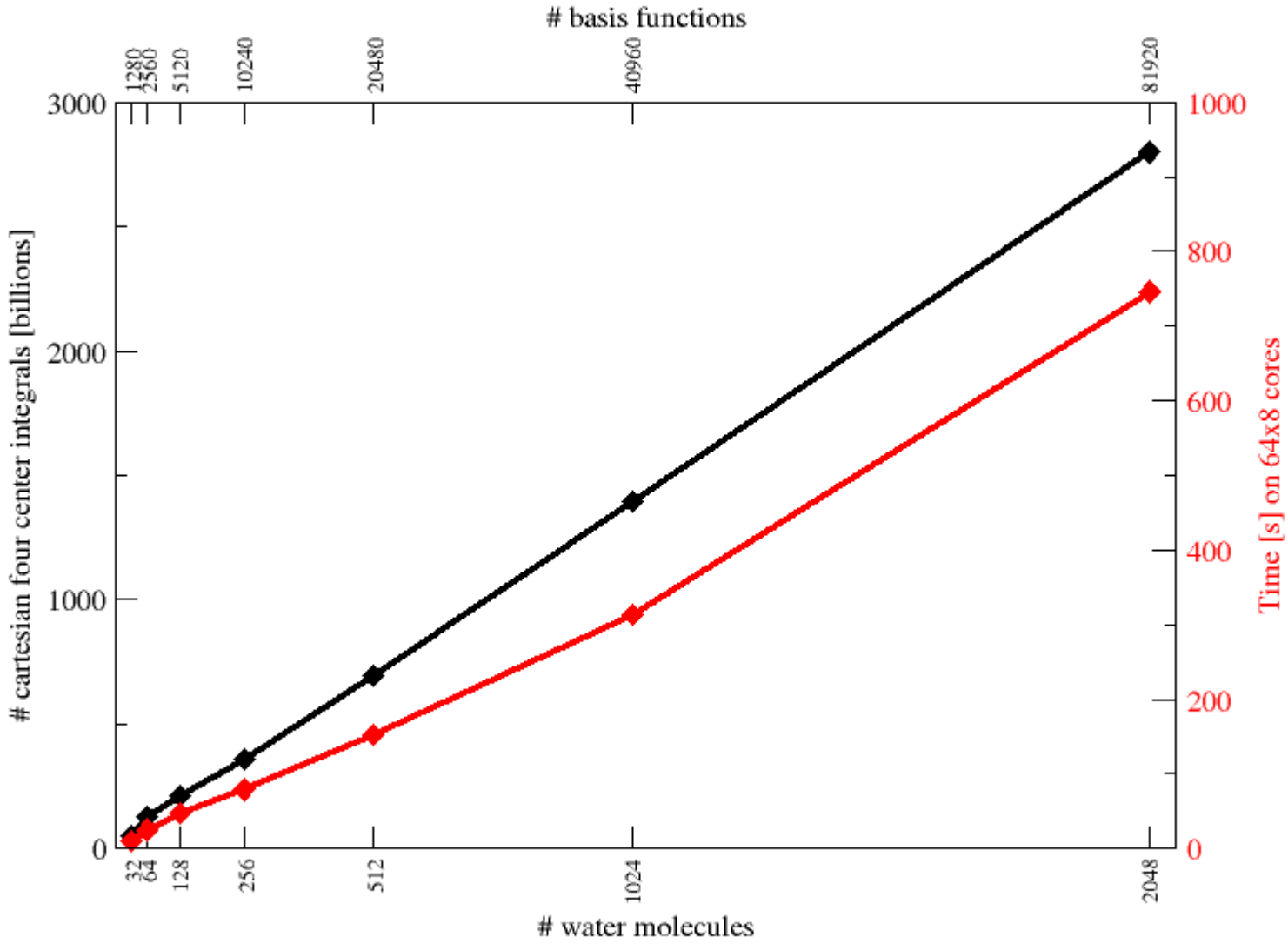
$$E_x^{\mathbf{HF}} = -\frac{1}{2} \sum_{\alpha\beta\gamma\delta} P_{\alpha\beta} P_{\gamma\delta} (\phi_\alpha \phi_\gamma | \phi_\beta \phi_\delta)$$

$$(\phi_\alpha \phi_\gamma | \phi_\beta \phi_\delta) = \int d\mathbf{r} d\mathbf{r}' \frac{\phi_\alpha(\mathbf{r}) \phi_\gamma(\mathbf{r}) \phi_\beta(\mathbf{r}') \phi_\delta(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}$$

Based on the fact that for large systems either the integrals
Or the density matrix become zero (to within a threshold eps)

Cauchy-Schwarz screening	$\lvert(ab\lvert cd)\rvert \leq \sqrt{(ab\lvert ab)\lvert(cd\lvert cd)}$	$O(N^2)$
Density matrix screening	$P_{\alpha\beta}$ decays exponentially	O(N)
Operator screening	Operators other than 1/r	O(N)

O(N) HFX: measurements



Linear scaling is key .... thousands of molecules possible
On 'standard' cluster hardware in minutes.

Parallel implementation: OMP/MPI

$E=v^T(Mv)$?

Distribute the matrix M (dim: 10^9 x 10^9), replicate the vector v

Simple communication pattern (v is distributed in GGA mode)
allows for exploiting the full symmetries (8x speedup)
$$(ab|cd) = (ba|cd) = (ba|dc) = (ab|dc) = (cd|ab) = (cd|ba) = (dc|ba) = (dc|ab)$$
Advanced load balancing model used

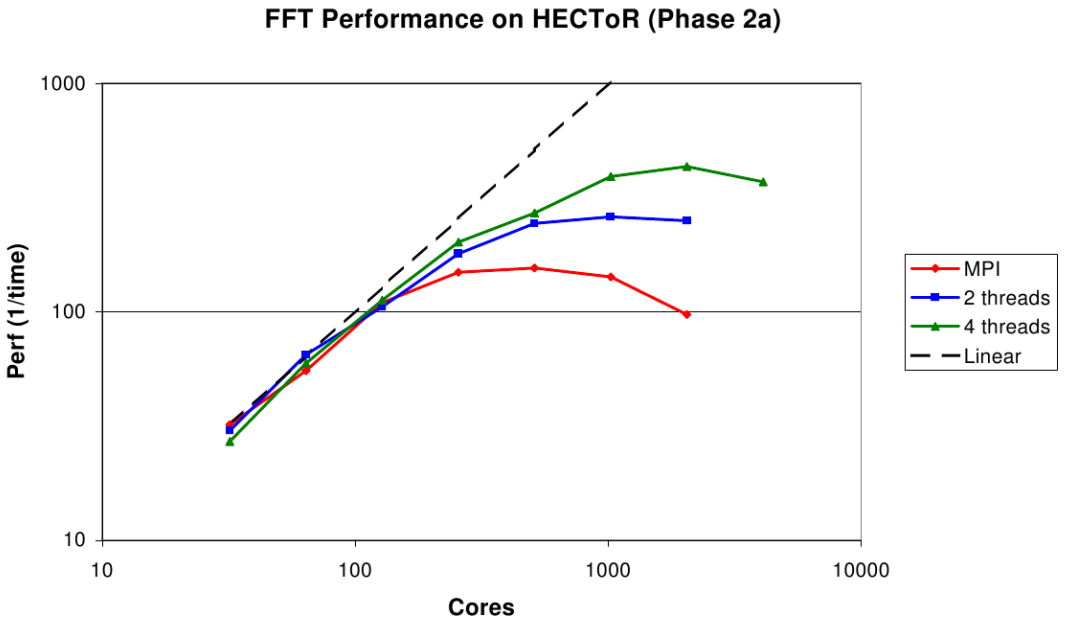
v and (Mv) can be rather large vectors (10^9 elements)

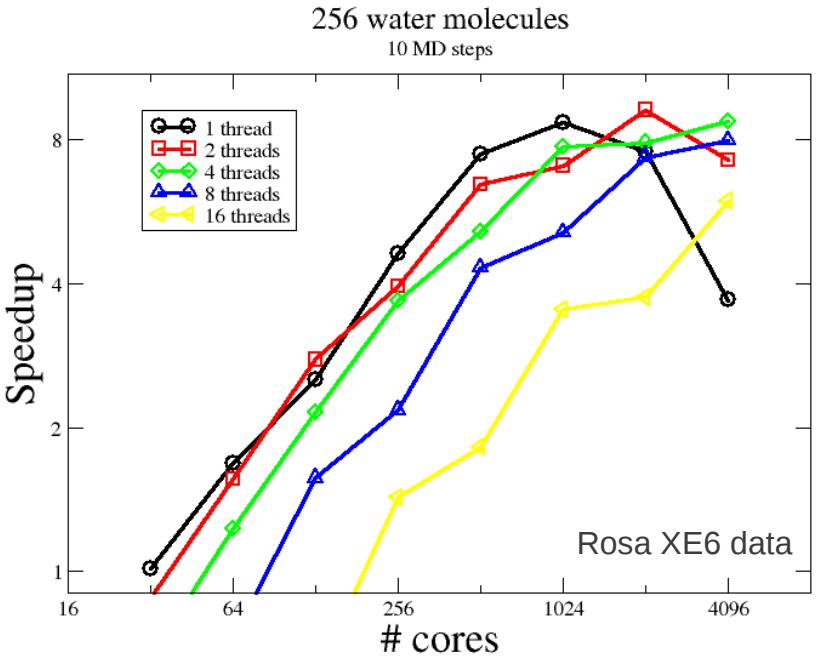
Exploit current architectures (e.g. 16 cores with 16Gb per node) → MPI/OpenMP
Shared v and (Mv), v is read-only, (Mv) is atomically updated
Exploit that only O(N) entries of v are non-zero → sparse storage for v
Remaining memory used for storing M.

Many-core era: OMP/MPI

Main benefits:
- Increase memory per MPI task
- Extend the scalability plateau
- Access >10000s of cores
- Interface to GPU (XK6)
- Speedups for some parts of the code

Main issues:
- Loop level OMP is not sufficient
- Analyzing OMP performance difficult
- Non-deterministic threading bugs
- Libraries (scalapack) poorly threaded
- Compilers & tools

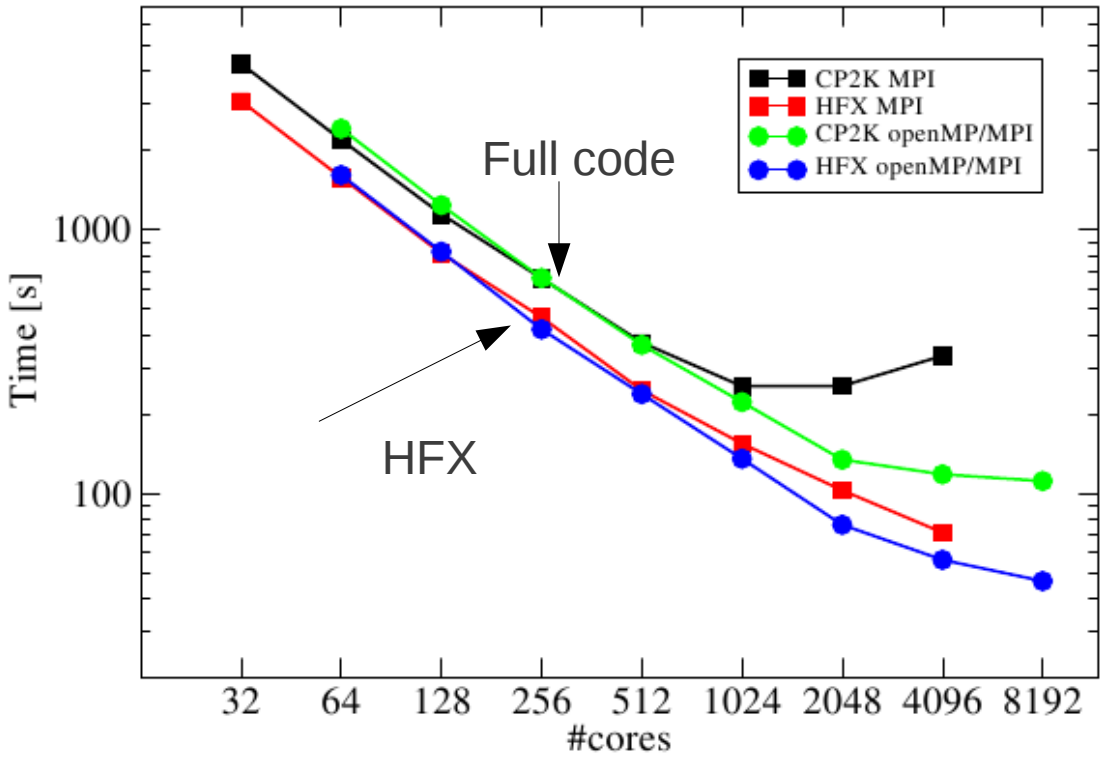




Here computer centers can help! EPCC (EPSRC/PRACE funded) did so.

Parallel efficiency

HFX remains computationally much more demanding than GGA DFT (10x?)
A good parallel implementation is mandatory

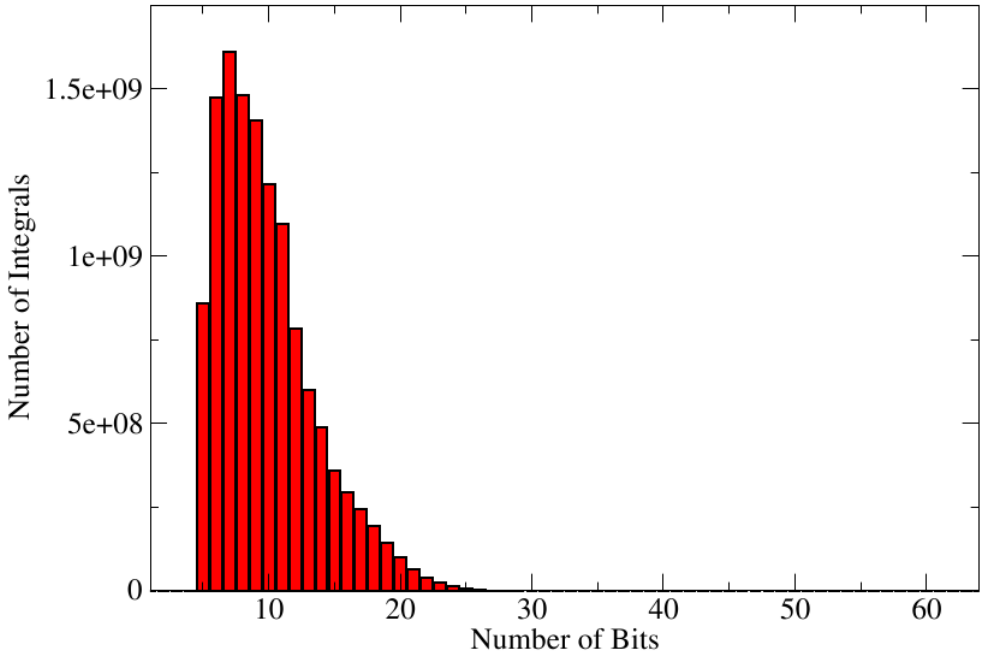


10 steps of MD, 64 H_2O, 2560 BF,
OpenMP: 8 threads/node

HFX code out-scales the
Remaining (GGA) part of CP2K

Provided enough compute power,
Hybrid simulations run essentially
as fast as GGA
(9s / BOMD step @ 4096 cores)

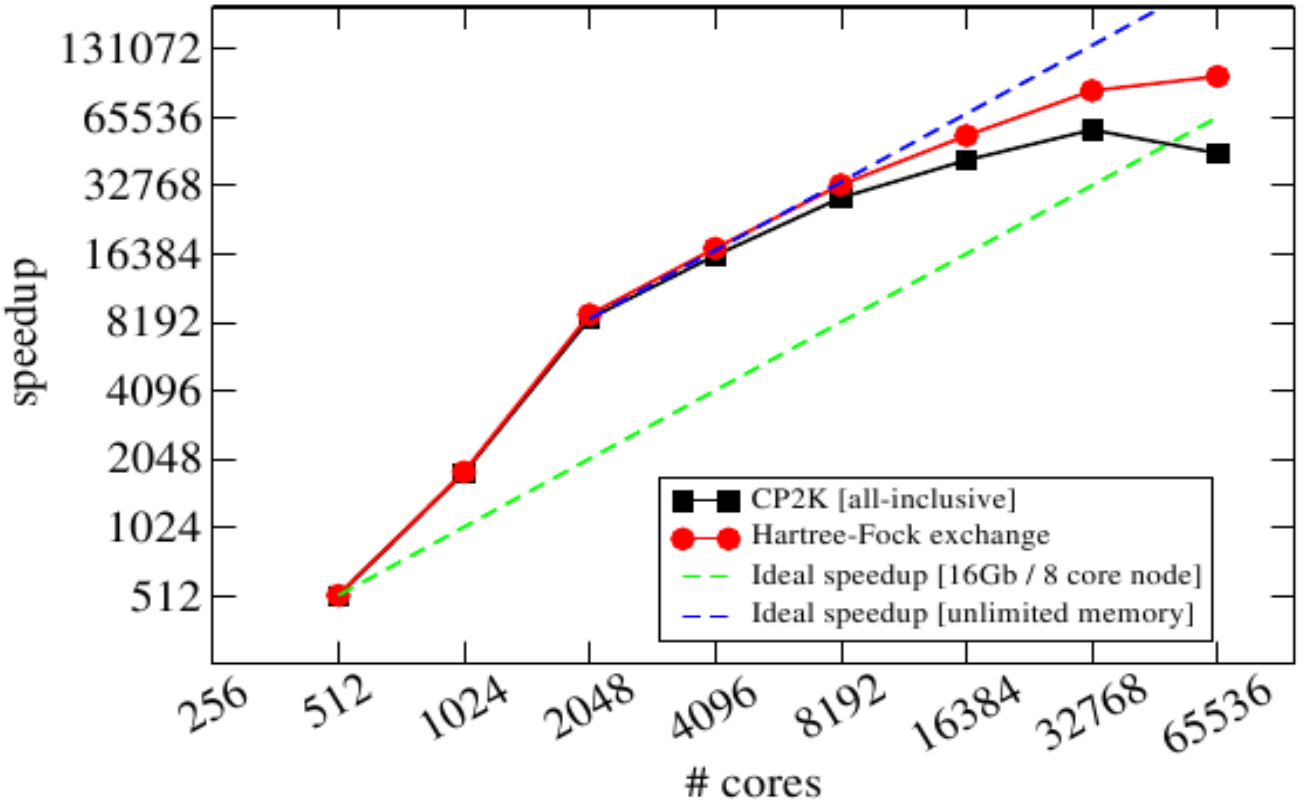
In-core integral compression



Almost all simulations are performed using an in-core algorithm
→ 10x speedup is observed.
Highly efficient scheme: index free and lossy compression

Guidon, M; Schiffmann, F; Hutter, J; VandeVondele, J. 2008 ; JCP 128(21): 214104

LiH: Parallel efficiency & in-core operation



Initial speed:
Using CPU efficiently

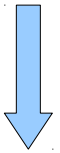
Superlinear speedups:
Using all memory

Good scale-out:
Using network

Paier J; Diaconu CV; Scuseria GE; Guidon M; VandeVondele J; Hutter J. 2009: PRB 80(17): 174114
Guidon M; Hutter J; VandeVondele J. 2009: JCTC 5(11): 3010-3021

Auxiliary Density Matrix Methods (ADMM)

For certain density matrices HFX can be computed very efficiently
(e.g. small basis sets or increased sparsity)



Transform an expensive matrix into a cheap one,
use a GGA for estimating the difference

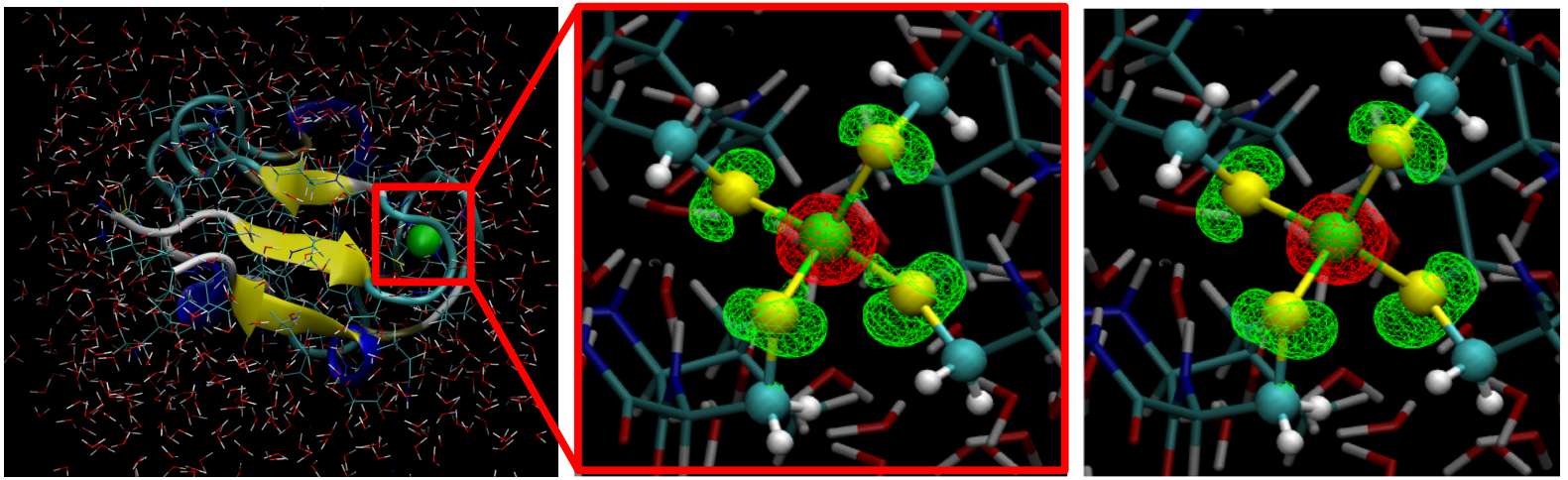
$$
\begin{aligned}
E_x^{\mathrm{HFX}}[P] &= E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{HFX}}[P] - E_x^{\mathrm{HFX}}[\hat{P}]\right) \\
&\approx E_x^{\mathrm{HFX}}[\hat{P}] + \left(E_x^{\mathrm{DFT}}[P] - E_x^{\mathrm{DFT}}[\hat{P}]\right)
\end{aligned}
$$

One example: wavefunction fitting, using an auxiliary basis

$$
\min_{\tilde{C}} \left[\sum_j \int \left(\psi_j(\mathbf{r}) - \tilde{\psi}_j(\mathbf{r})\right)^2 d\mathbf{r} + \sum_{k,l} \Lambda_{kl} \left(\int \tilde{\psi}_k(\mathbf{r}) \tilde{\psi}_l(\mathbf{r}) d\mathbf{r} - \delta_{kl} \right) \right]
$$

Guidon M; Hutter J; VandeVondele J; JCTC 6(8): 2348-2364

ADMM: performance



Standard hybrid

ADMM based hybrid

The DFT model: a solvated protein

- 48'000 cores
- 1. SCF step: 45 min

- 1'152 cores
- 1. SCF step: 75 s

A fully solvated protein computed within minutes using hybrid functionals

Computer science must be combined with domain knowledge!

Guidon M; Hutter J; VandeVondele J; JCTC 6(8): 2348-2364

Møller-Plesset Perturbation Theory

The energy:

$$E^{(2)} = -\sum_{ij,ab}^{occ,vir} \frac{(ia|jb)[2(ia|jb) - (ib|ja)]}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j}$$

Two electron integrals over canonical molecular orbitals (MO):

$$(ia|jb) = \int\int \psi_i(\vec{r}_1)\psi_a(\vec{r}_1)\frac{1}{\vec{r}_{12}}\psi_j(\vec{r}_2)\psi_b(\vec{r}_2)d\vec{r}_1 d\vec{r}_2$$

The four index transformation, going from AO to MO

$$(ia|jb) = \sum_{\mu\nu\lambda\sigma} (\mu\nu|\lambda\sigma)C_{\mu i}C_{\nu a}C_{\lambda j}C_{\sigma b}$$

MP2 is relatively expensive O(N^5), not easy to parallelize efficiently, and somewhat tricky in the condensed phase for an AO code.

GPW-MP2

A Gaussian and plane waves approach to MP2

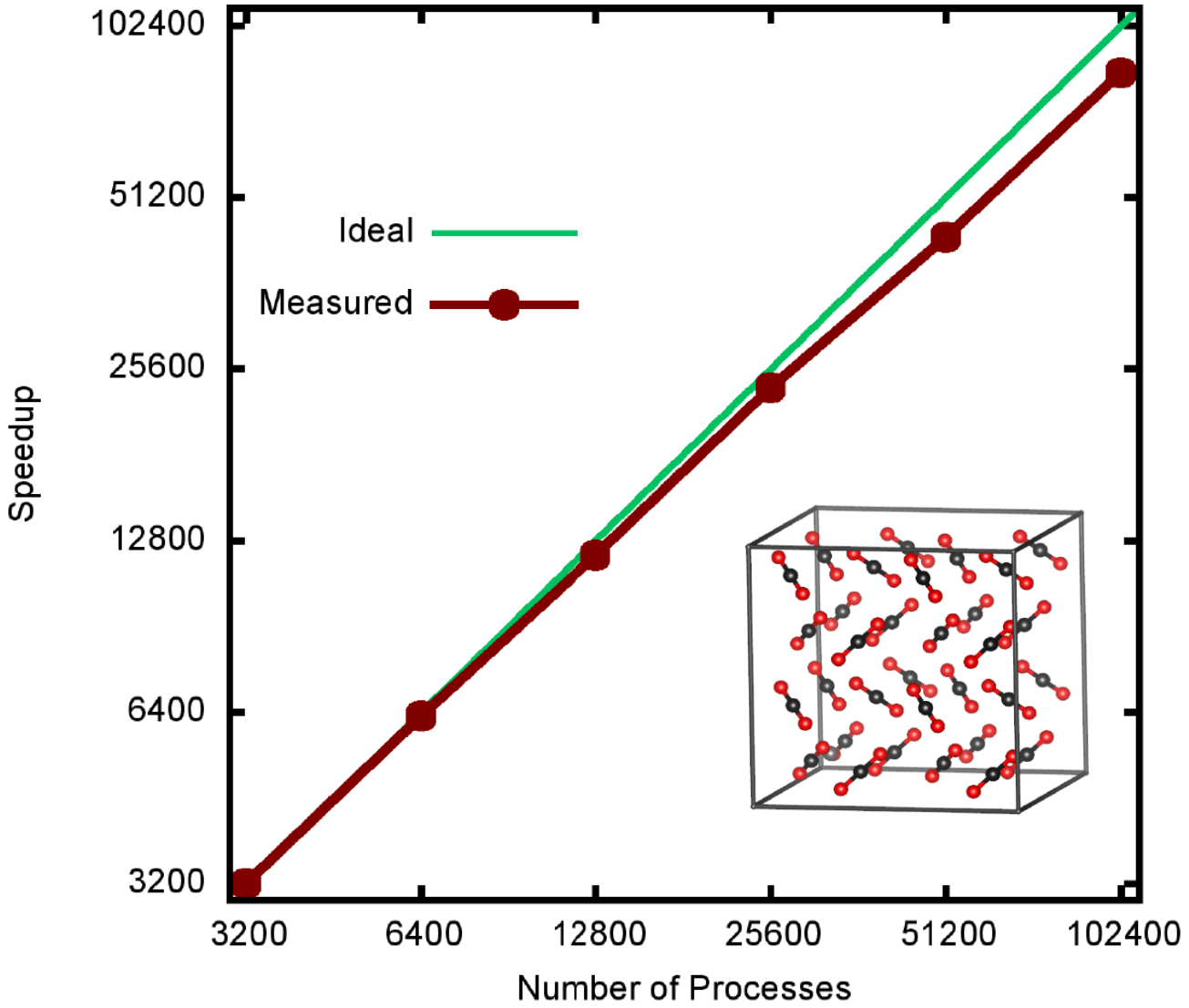
$$
\begin{aligned}
(ia|\lambda\sigma) &= \int\int \psi_i(\vec{r}_1)\psi_a(\vec{r}_1)\frac{1}{\vec{r}_{12}}\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_1 d\vec{r}_2 \\
&= \int\left[\int \frac{\psi_i(\vec{r}_1)\psi_a(\vec{r}_1)}{\vec{r}_{12}}d\vec{r}_1\right]\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_2 \\
&= \int\left[\int \frac{\rho^{ia}(\vec{r}_1)}{\vec{r}_{12}}d\vec{r}_1\right]\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_2 \\
&= \int v^{ia}(\vec{r}_2)\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_2
\end{aligned}
$$

Directly obtain half transformed integrals using the GPW approach:

Leads to a highly efficient parallel implementation.

Del Ben M, Hutter J, VandeVondele J: to be submitted

Parallel efficiency



CO_2 crystal (32 molecules)
cc-QZVP basis (5184 BF)
Canonical GPW-MP2 calculation

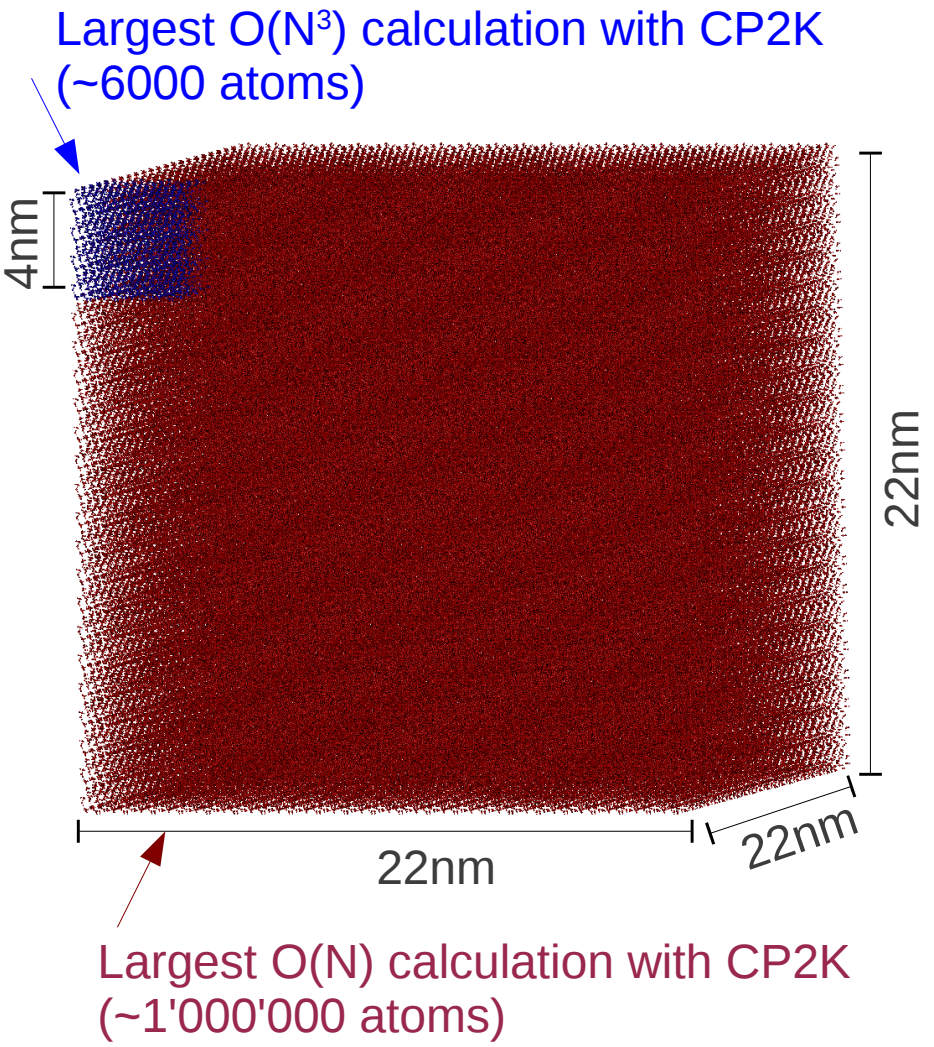
MP2 time: 9min.
parallel efficiency: 80%
On 102400 cores

Larger model systems

Linear Scaling SCF

Traditional approaches to solve the self-consistent field (SCF) equations are $O(N^3)$ limiting system size significantly.

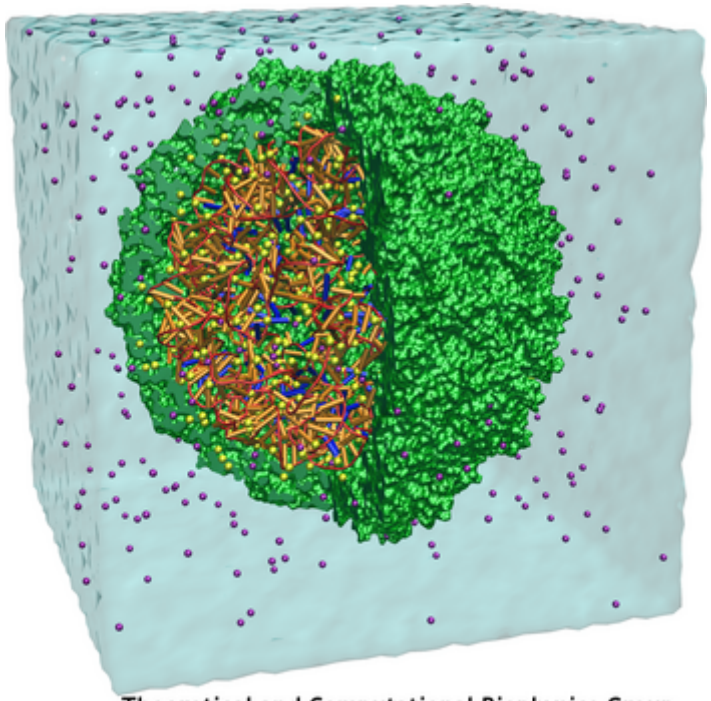
A newly implemented algorithm is O(N), allowing for far larger systems to be studied.



Linear Scaling SCF

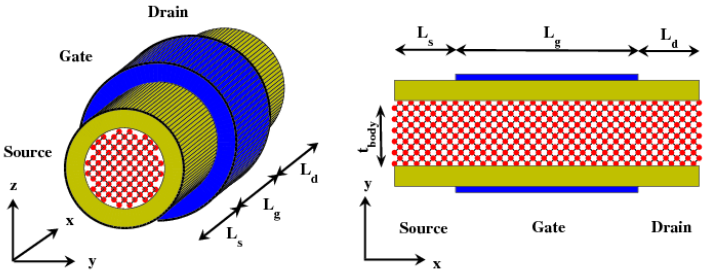
New regime: small devices, heterostructures, interfaces, nano-particles, a small virus.

Solvated STMV: 1M

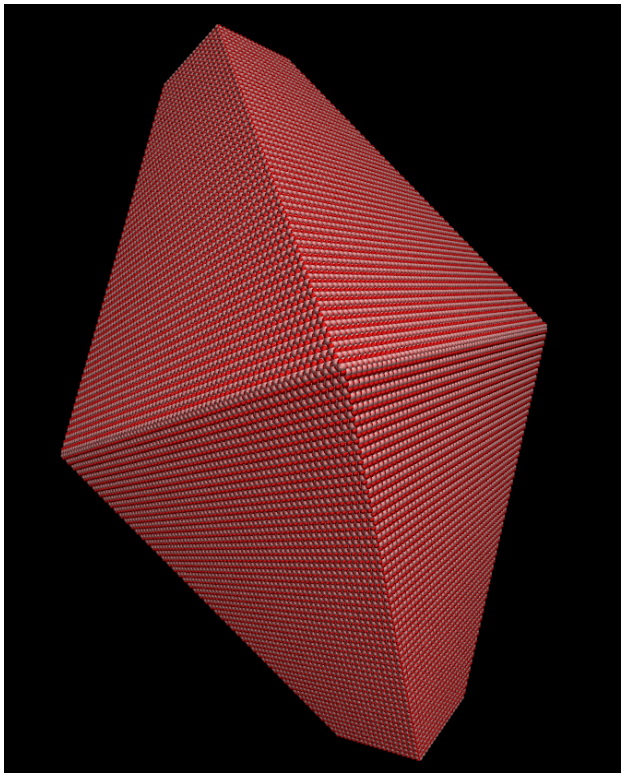


Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign

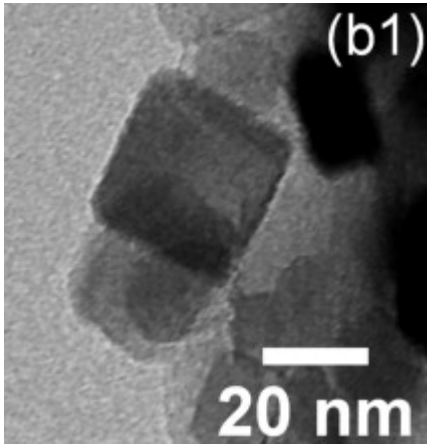
Gate-all-around FET



With Mathieu Luisier



1.5M atoms
Anatase nanocrystal



Caplovicova et al.
App. Cat. B, 224, 117

Sign matrix iterations

The density matrix (P) is function of H

$$P = \frac{1}{2}(I - \mathrm{sign}(S^{-1}H - \mu I))S^{-1}.$$

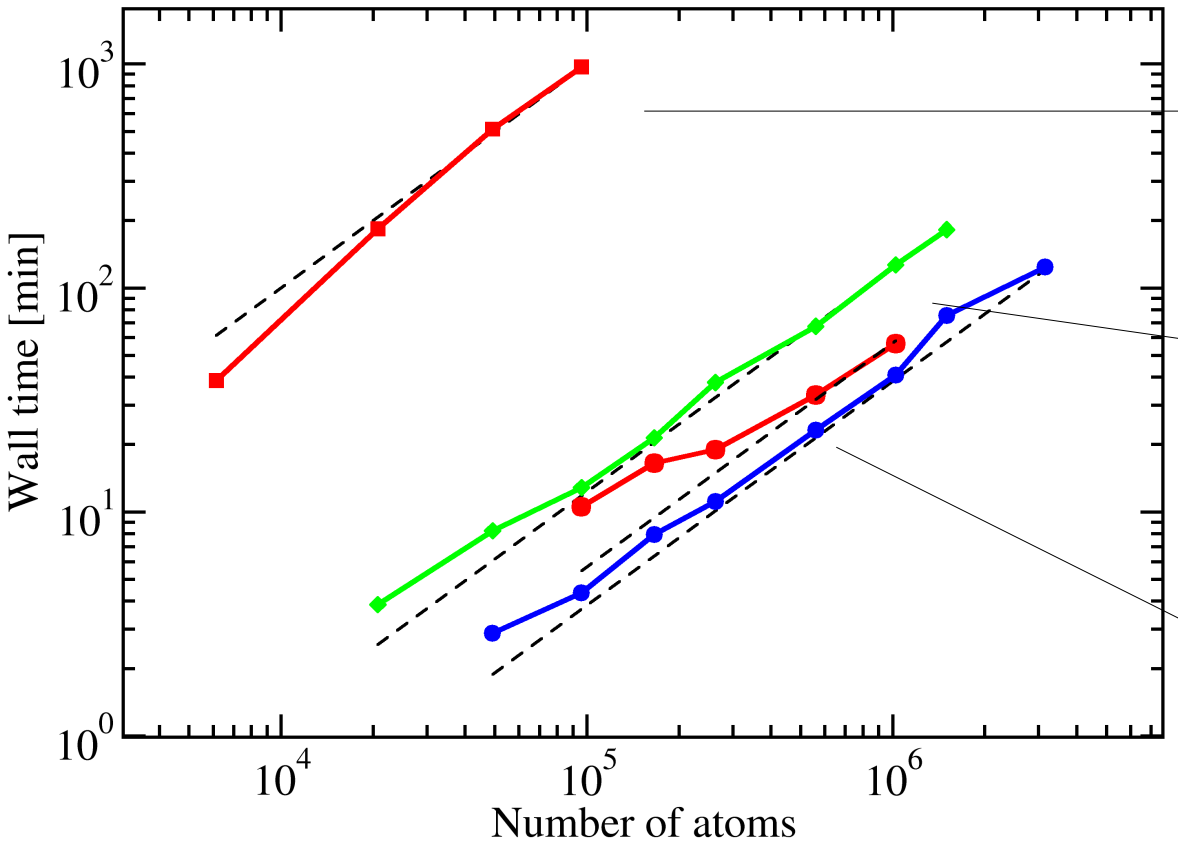
A simple iterative scheme (Newton-Schultz) gives sign(A):

$$X_{n+1} = \frac{1}{2}X_n(3I - X_n^2).$$

Using only sparse matrix matrix multiplies (not SPMV!)
linear scaling can be obtained

A dedicated sparse matrix multiply library is extremely important
This library is being ported to GPUs

Millions of atoms in the condensed phase



Accurate basis sets, DFT
46656 cores

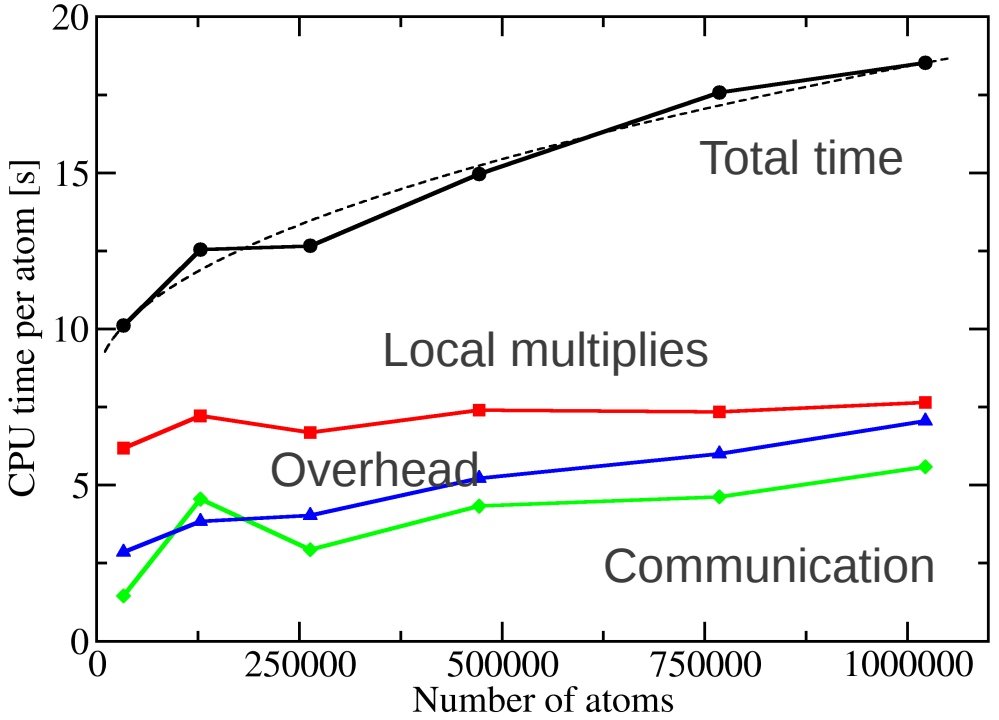
The electronic structure
$O(10^6)$ atoms in < 2 hours

Minimal basis sets:
DFT, NDDO, DFTB
9216 cores

Bulk liquid water. Dashed lines represent ideal linear scaling.

VandeVondele, Borstnik, Hutter, JCTC

Towards O(1) :
constant walltime with proportional resources



Stringent test:
- Small blocks, large overhead
- Very sparse matrices
- Running with 200 atoms / MPI task

Local multiplies constant (OK!).

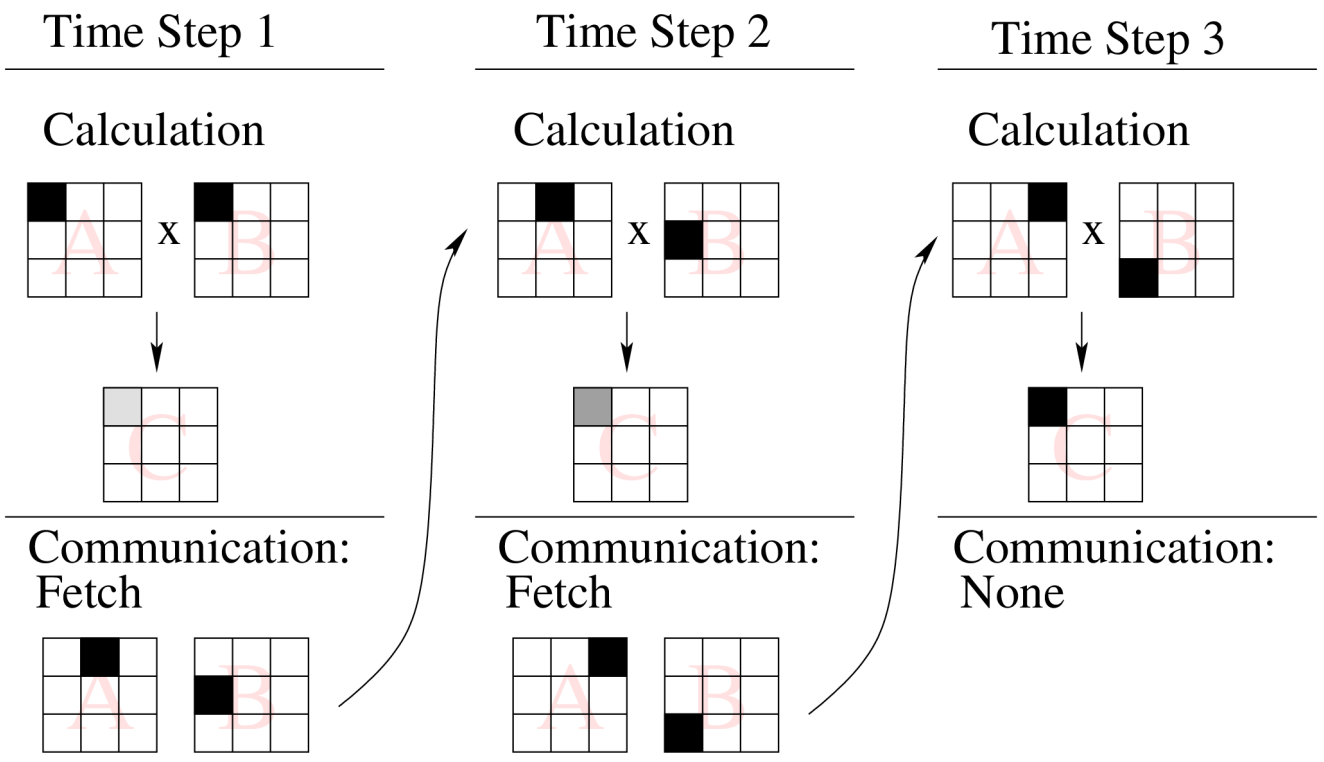
Overhead & Communication
- Grows with sqrt(N)
- Needs a replacement for Cannon

Work is underway to replace the Cannon algorithm with something new!
Retain the sqrt(N) max comm, yield constant comm in the limit.

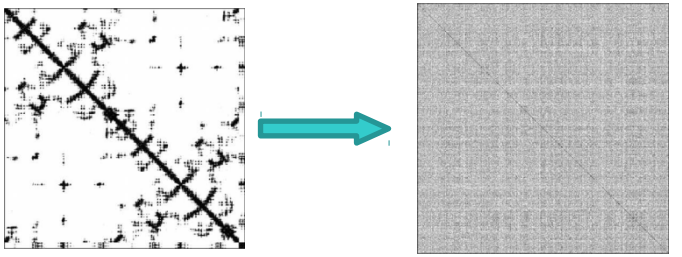
DBCSR: a sparse matrix library

Distributed Blocked Compressed Sparse Row
Distributed Blocked Cannon Sparse Recursive

Target the application: atoms → Blocks (e.g. 5x5, 13x13, 23x23)
Linear scaling → Sparse
Fully dense → Cannon
Large scale → Distributed
High Performance → Recursive



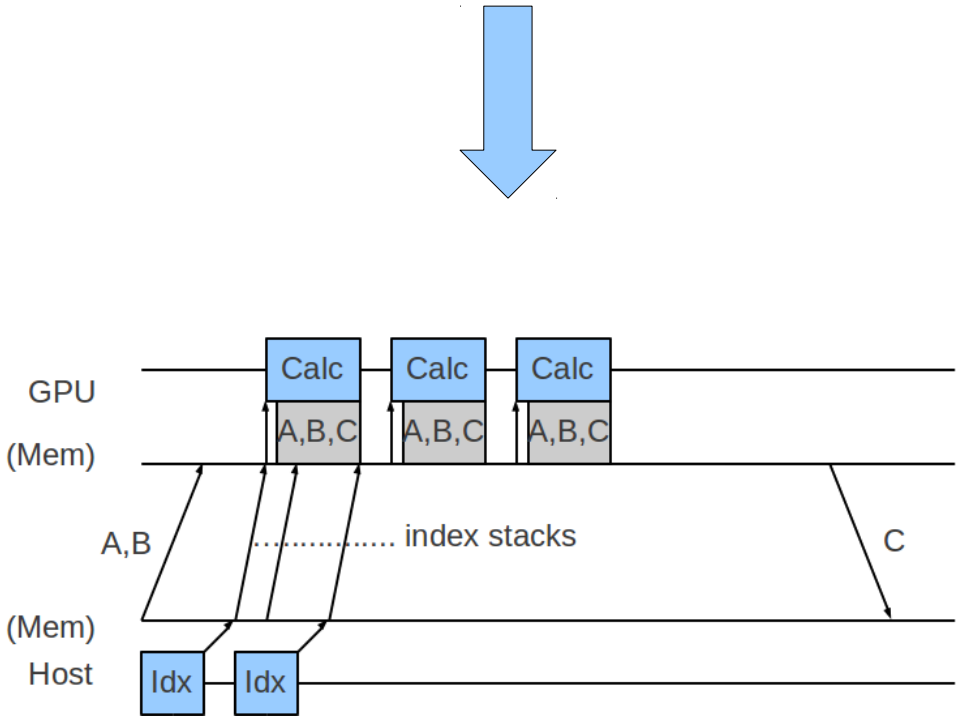
Cannon style communication on a homogenized matrix for strong scaling



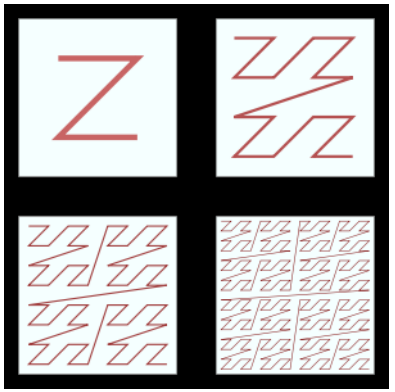
The local multiplication

A two stage process:

- Index building: figure out what to calculate
- Computation: Do the calculations



A cache-oblivious recursive approach



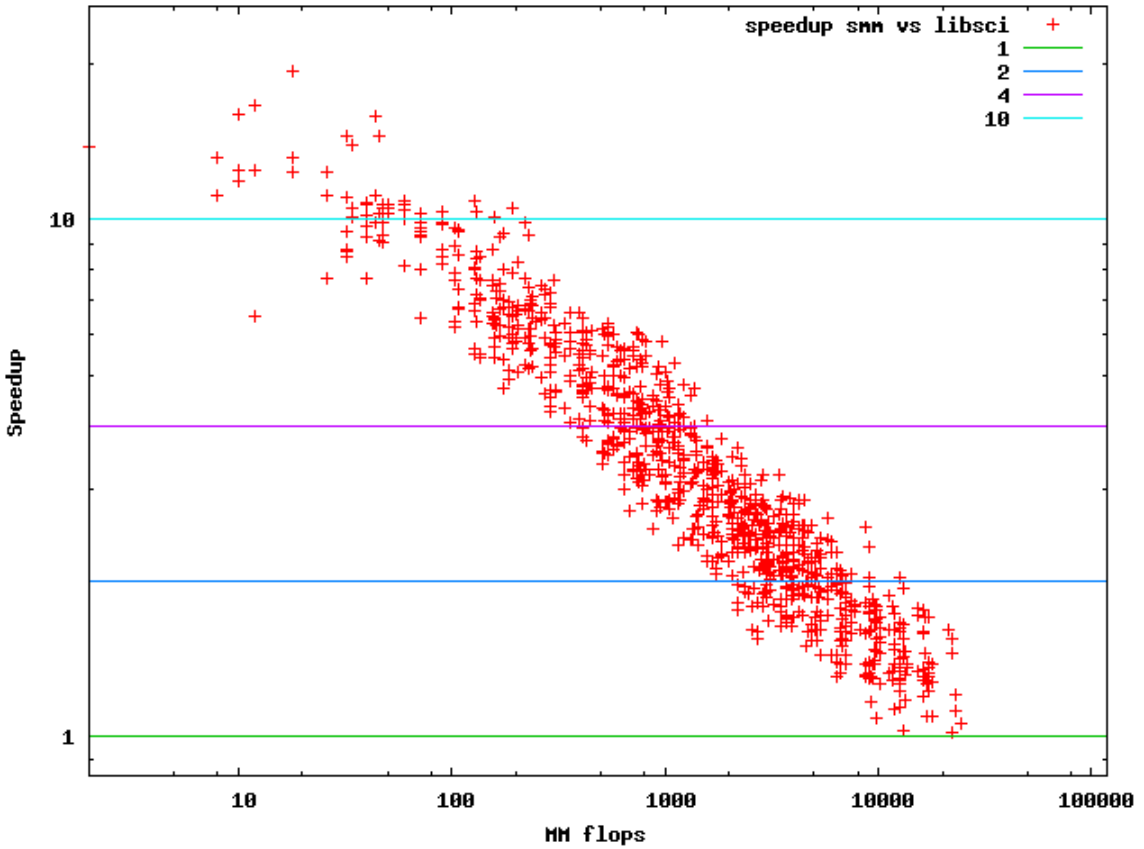
LIBSMM: an autotuned CPU library for small matrix multiplication: 2x faster than optimized blas for applications

An auto-tuned small matrix muliply library

C=C+A x B

- Various 'strange' sizes needed
  - 5x5, 5x13, 13x13 ...
- Hand-optimization
  - too difficult
  - too time-consuming
  - not-flexible enough (new chips)
- Auto-generate a separate library
  - Easy
  - Fast
  - Flexible
  - Now available

Libsmm outperforms optimized blas : 2x-10x



Autogeneration – Autotuning … a powerful approach to performance (Atlas, FFTW, ...)

GPU Kernel Description

Compute a batch Cij, Aik, Bkj of small matrix block products (5x5 .. 23x23)
Not too different from batched dgemm (Ci=Ci+Ai*Bi) available in cudablas 4.1

CUBLAS Batched dgemm reaches
Only ~50 Gflops for 23x23

Need to:
1) exploit dependencies
   (data reuse)
2) Better compute kernel

Sort batch on Cij so we have only 2 data read/writes instead of 4.
Hand-optimize a CUDA kernel for size 23x23 (playing with registers / occupation etc)

The GPU port of DBCSR (I)

Not so easy … ran into many issues with tools
… but received excellent support from Cray/NVIDIA/CP2K team

- Up to 3x performance slowdown in MPI for dynamically linked binaries
  - Fix: use a (not yet released) Cray patched libudreg.so.0 (?)
- ran into CCE miscompilation
  - Fix : compile selected file at -O0
- problems with (virtual) memory management
  - Fix: use hugepages
- A CP2K bug
  - Fix: add a memcpy.
- Up to 10X threading performance difference between compilers
  - Investigating (not crucial for final benchmark)
- Cuda-memcheck crashes/gave false positives
  - Ignore after checking carefully
- Usual tools (valgrind or CP2K internal timing/memory trace) do not work
  - accept

Not a particularly time efficient procedure...
Not for every developer …
Not without excellent support ….

The GPU port of DBCSR (II)

Open Challenges:

- Only one process per node can connect to GPU.
  Need a functional OMPed code as well

- Work sharing between GPU and CPU not yet optimal.
  Have the CPU also do stacks of multiplications ?

- Sending data panels to GPU not yet overlapping with any work
  Could be started as soon as MPI completes. Maybe double buffering

- GPU Kernel optimized for 23x23 matrices (water molecule). We need many sizes.
  Needs an auto-tuning and auto-generating framework for the GPU
  A strategy to deal with smaller blocks.

GPU multiplication benchmark

Benchmark matrix-matrix multiplication with random [0..1] matrix, quasi-realistic fill-in pattern

Hybrid MPI+OpenMP+CUDA Fortran/C code

NREP=6: Matrix dimension N=159’000;
50% occupation

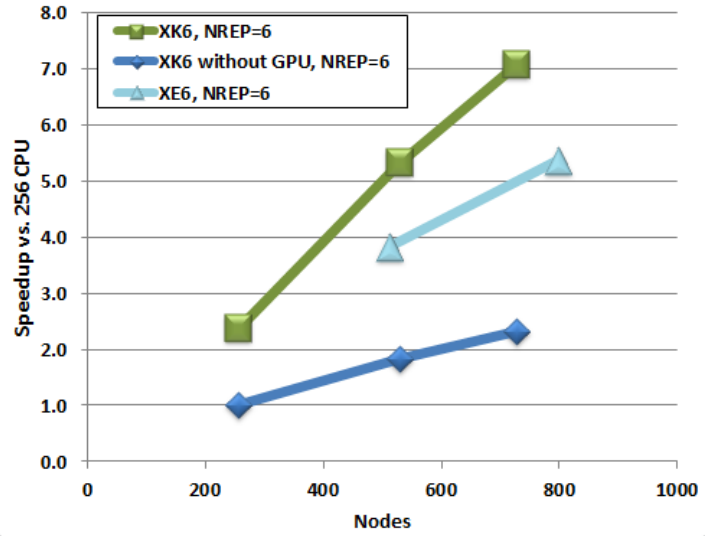
<370Mb of data per node on 256 nodes....

CPU (1xIL, XK6): 38 GFLOPS
CPU (2xIL, XE6): 77 GFLOPS (2.3-2.5 GFLOPs/core)
GPU : 114 GFLOPS (110 – 125 GFLOPs)

Current Kernel performance 170 Gflops.

Estimated Keppler performance: 1.6x speedup on kernel

Performance difference: time in MPI



(data Peter Messmer)

GPU application benchmark

- >400 multiplications for 1 run.
- Additional thresholding in multiplications (less flops for same data)
- This week's results.... subject to change

20736 atoms (6912 water molecules), matrix dim 159000, on 576 nodes XK6, ~60 matrix multiplications / iter.

XK6 without GPU : 1965s per iteration
XK6 with GPU : 924s per iteration

Speedup 2.12x

MPI performance (bandwidth) appears to be the bottleneck (e.g. 50% slowdown without custom rank reordering) :

- Still need to figure out MPI performance (incl. effectiveness of overlap).
- Is the dynamic linking still an issue ?
- Any interference between GPU+CPU ?
- One Communication thread per node enough ?

Further thoughts and conclusions

Computational atomistic modeling is a lot of fun!

A wide range of algorithms is important for DFT simulations

Implementation, algorithm and theory all have their role to play

Good theories and algorithms require knowledge of the scientific problem being solved

Understanding the hardware is important, but challenging given the rapidly changing field

Hierarchical parallelism will be needed to scale to 1000s of nodes.

GPU/MIC/etc. coding is still challenging, will improve as a wider range of software is ported

Large scientific codes will increasingly benefit from new development methodologies.

The Fortran/MPI combo is still the level you can expect in academia

Acknowledgements

Zürich
Juerg Hutter
Urban Borstnik
Christian Pousa
Mauro Del Ben
Manuel Guidon
Valery Weber

World-wide
CP2K Team

PNL&Minnesota
Chris Mundy

CSCS&EPCC&
CRAY&NVIDIA
Iain Bethune
Neil Stringfellow
Peter Messmer
John Levesque
Roberto Ansaloni

Flops&More
CSCS
ORNL
UZH
ETH
SNF
INCITE
EU-FP
DEISA
PRACE

You for your attention!