CP2K

New Frontiers

in

ab initio Molecular Dynamics

Jürg Hutter, Joost VandeVondele, Valery Weber

Physical-Chemistry Institute, University of Zurich

Ab Initio Molecular Dynamics

- Molecular Dynamics Sampling of classical statistical ensemble (alternatively Monte Carlo sampling). Introduces temperature and pressure into simulation, connection to experimental conditions

- Electronic Structure Theory Interaction potentials are calculated using quantum mechanics for electrons. Ab initio theory - no precomputed or empirically adjusted potential functions needed.

Ab Initio Molecular Dynamics

Applications

- Chemical reactions in solution
- Complex interfaces (liquid/vapor, solid/liquid)
- Structural phase transitions
- Proton and/or electron transfer
- etc.

Ab Initio Molecular Dynamics

Embarrassingly Parallel Sampling Approaches

Replica exchange algorithms

Multiple walker metadynamics

Free energy string methods

Time Series Generation (MD, MC)

Electronic Structure Calculation

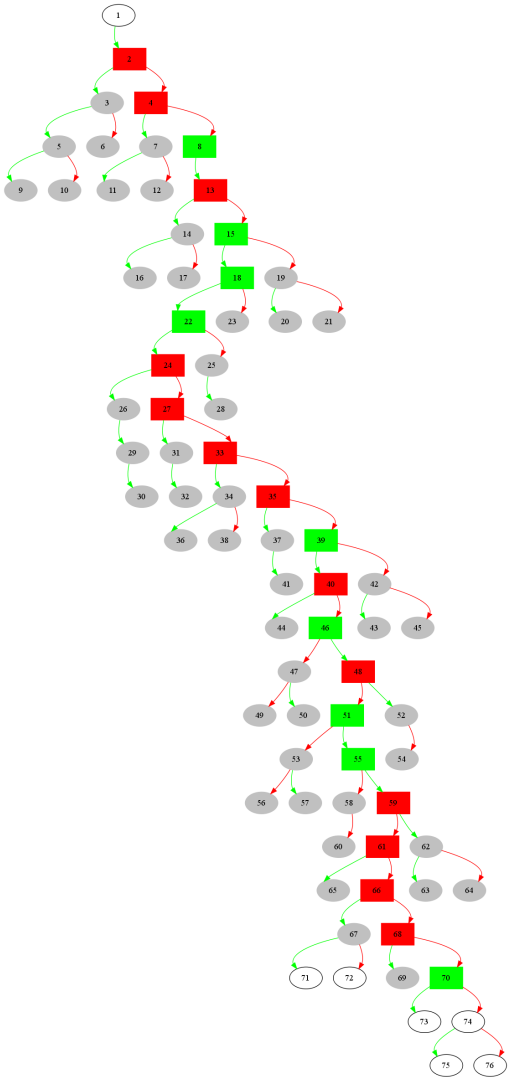
Parallel Time Series Generation

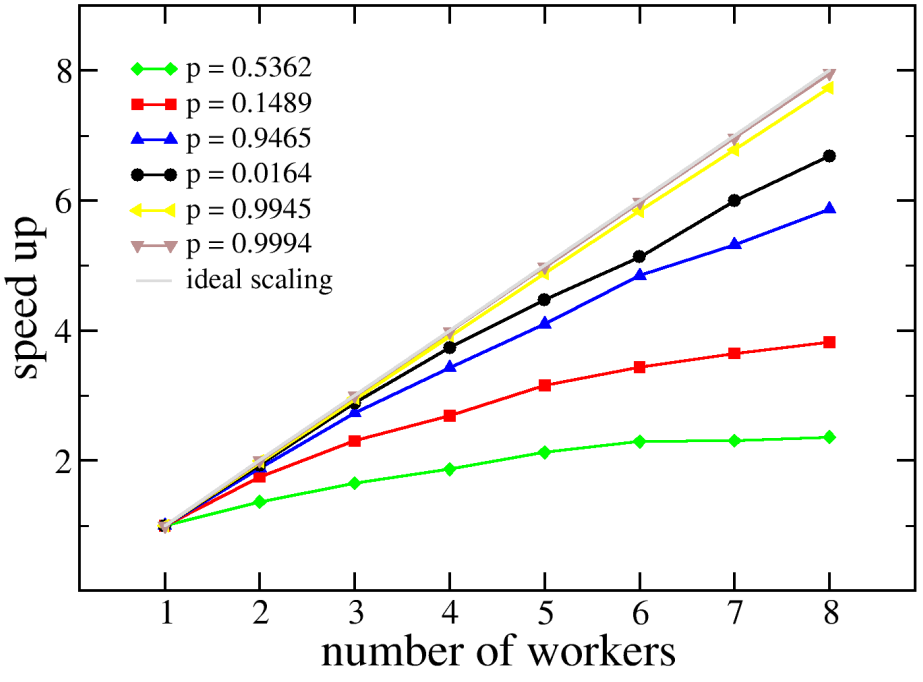
- Parallel Molecular Dynamics
  - Waveform relaxation;
    A. Bellen, M. Zennaro, J. Comput. Appl. Math. 25, 341 (1989)
  - Solar system dynamics;
    P. Saha, J. Stadel, S. Tremaine, Astron. J. 114, 409 (1997)

- Speculative Monte Carlo
  - Similar to branch prediction in CPUs
  - Effective if generation of states much cheaper than calculation of state energy.

Wasteful algorithms: Efficiency $\approx 0.2 - 0.6$

Speculative Monte Carlo





Electronic Structure Calculations

Kohn-Sham Density Functional Theory

- Nonlinear Schrödinger equation

$$\downarrow \qquad \text{basis set expansion}$$

Nonlinear generalized eigenvalue equation

$$\mathbf{H}[\mathbf{C}]\mathbf{C} = \mathbf{S}\mathbf{C}E$$

- Basis: Atom-centered Gaussian functions

$$\varphi(r) = Y_{lm}(\hat{r})\ \exp[-\alpha(r-A)^2]$$

Key Algorithms

- Gaussian to plane wave transform

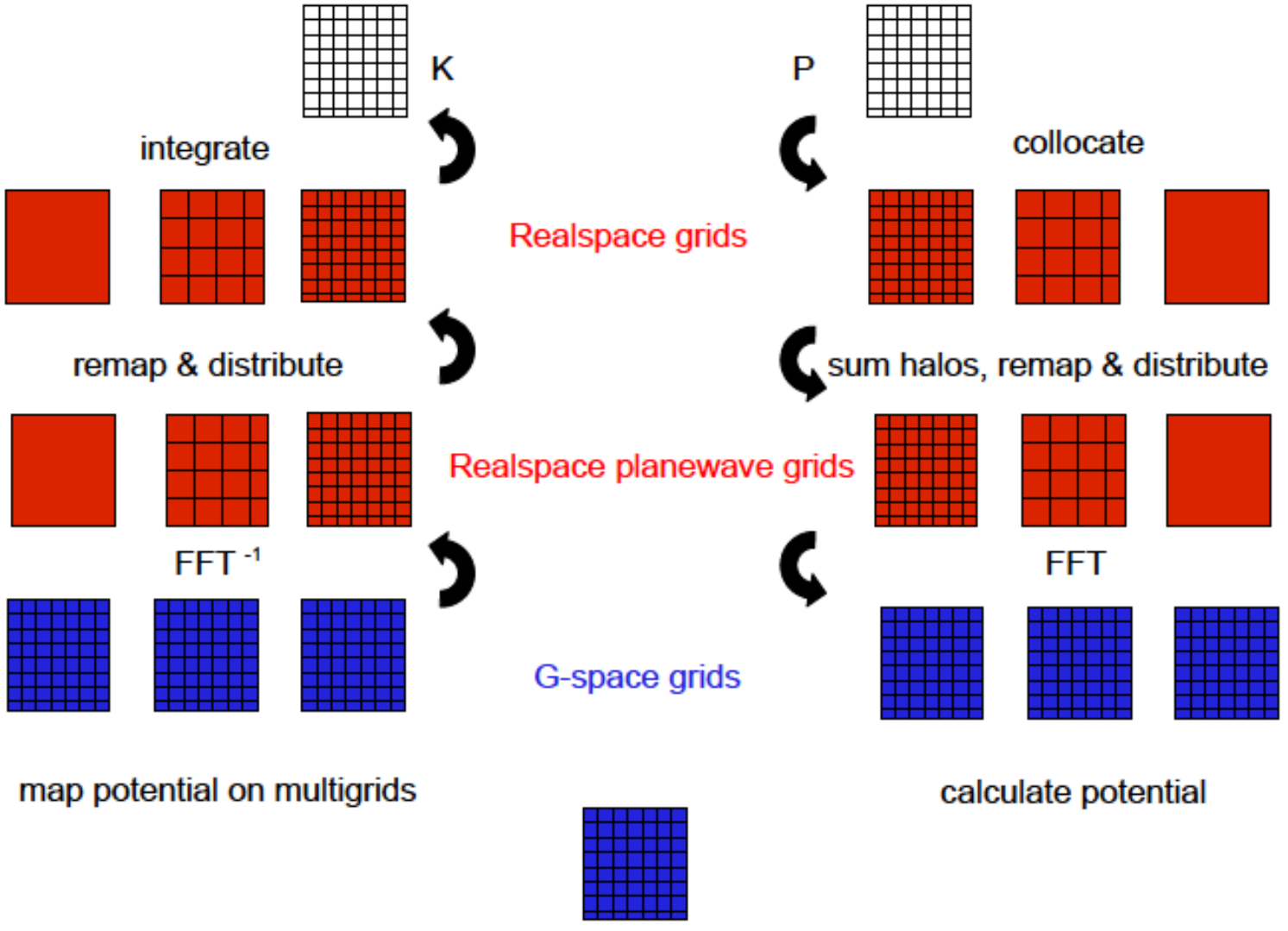
$$H^c_{\alpha\beta} = \sum_{\gamma\delta} P_{\gamma\delta} I_{\alpha\beta,\gamma\delta} = \sum_G (\alpha\beta, G) \frac{4\pi}{G^2} \left[\sum_{\gamma\delta} P_{\gamma\delta} (\gamma\delta, G) \right]$$

- Analytic Integrals for Hartree-Fock exchange

$$H^x_{\alpha\beta} = \sum_{\gamma\delta} P_{\gamma\delta} I_{\alpha\gamma,\beta\delta}$$

- Direct constraint subspace optimization
  Replace diagonalization by direct optimization with constraints
  Sparse matrix linear algebra

Gaussian to Plane Wave Transform



Gaussian to Plane Wave Transform

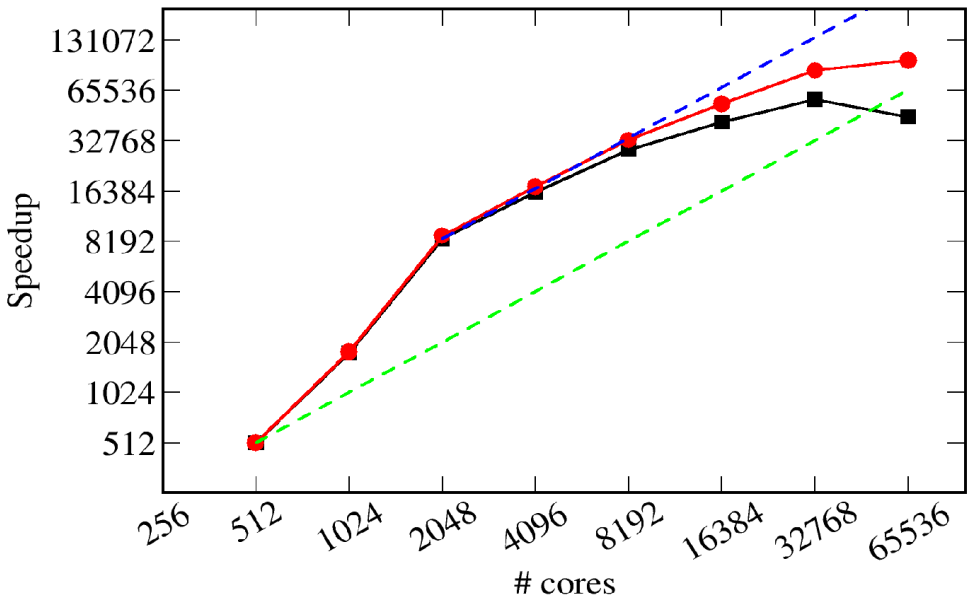
- Distributed sparse matrices and distributed grids
  Halo exchange
- 3d FFTs
- Complicated load-balancing problems
- Collaboration with EPCC Edinburgh
  MPI/OpenMP parallelization
  Target for co-array Fortran implementation

Hartree-Fock Exchange

- Large structured matrix vector multiply

$$H^x_{\alpha\beta} = \sum_{\gamma\delta} P_{\gamma\delta} I_{\alpha\gamma,\beta\delta}$$

- Current implementation:
  Distributed matrix ($I_{\alpha\gamma,\beta\delta}$), replicated vectors ($P_{\gamma\delta}$, $H^x_{\alpha\beta}$)



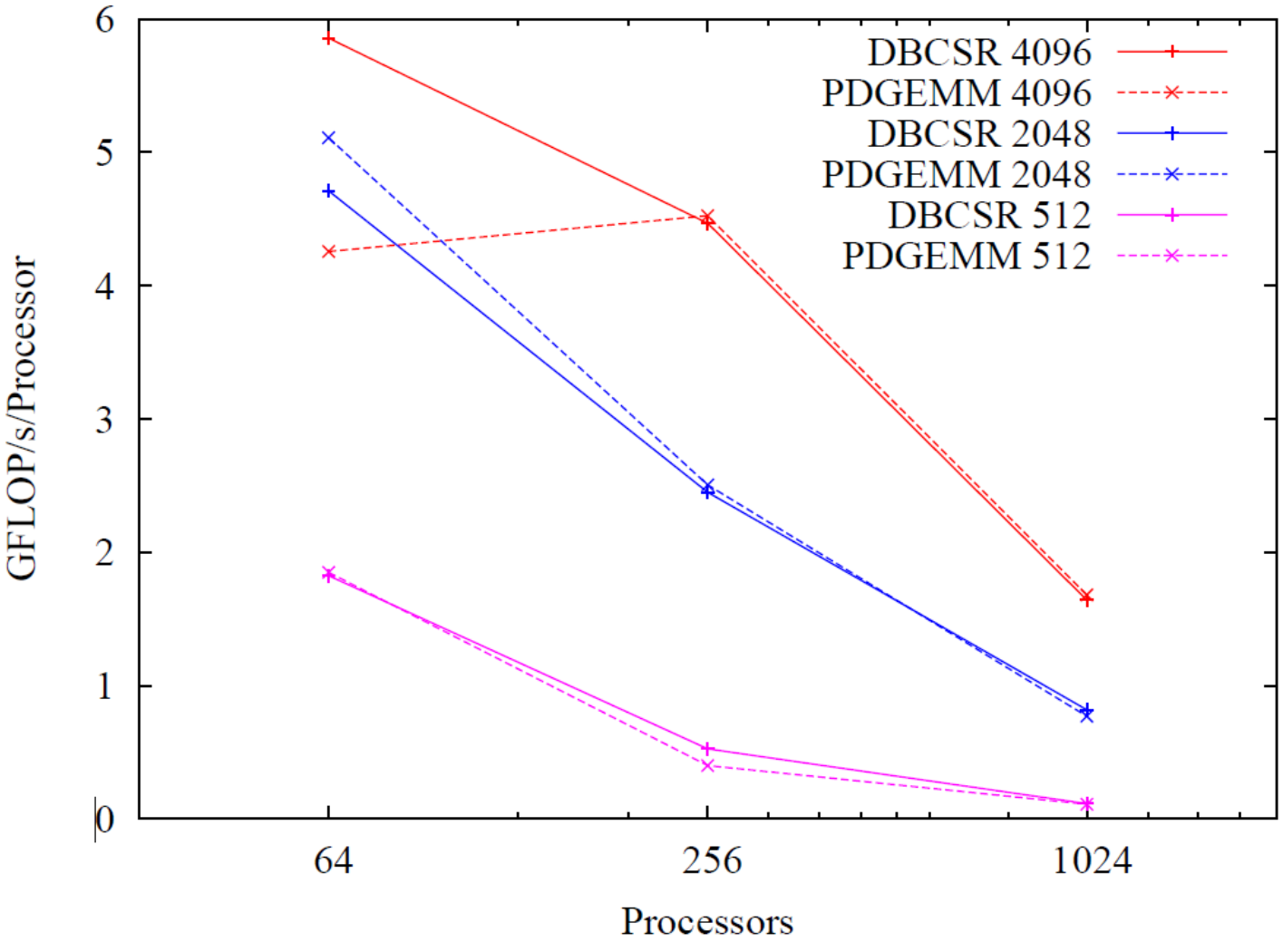
Sparse Linear Algebra

- Optimization (= solving the nonlinear eigenvalue equation) is mapped to matrix multiplications
- Old code: sparse matrices (operators) and dense matrices (solution vectors)
  Basic operation: sparse $\times$ dense matrix multiplication
- Linear scaling algorithms require
  Code refactoring: All matrices are of sparse type
- DBCSR library
  Distributed Block Compressed Sparse Row matrix format

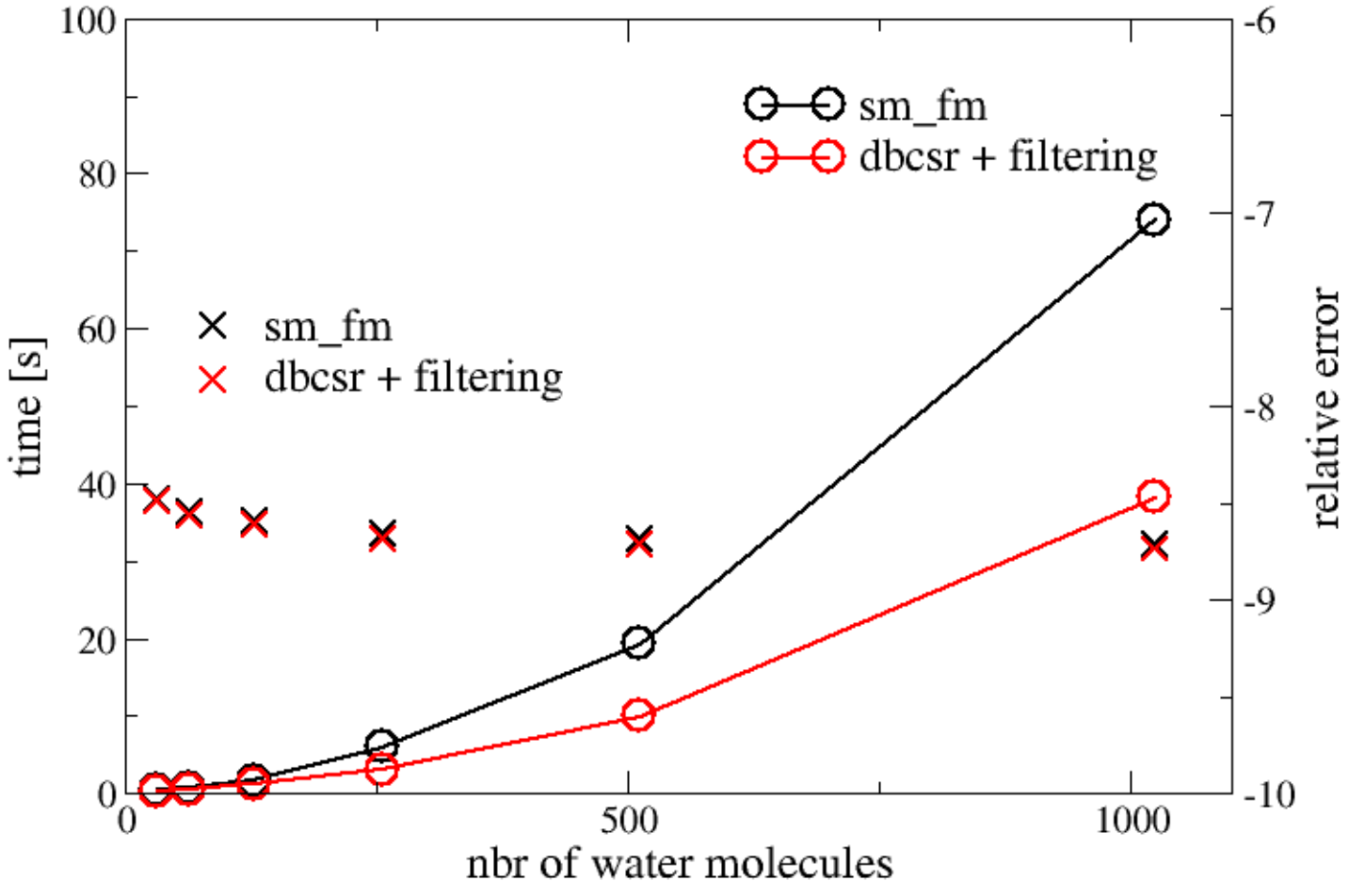
DBCSR Library

- ca. 50'000 lines Fortran 95
- Extensive functionality (single/double real/complex)
- Sparse-sparse matrix multiplication
  filtering, predetermined sparsity pattern
- Interface to ScaLapack
- MPI/OpenMP parallelization

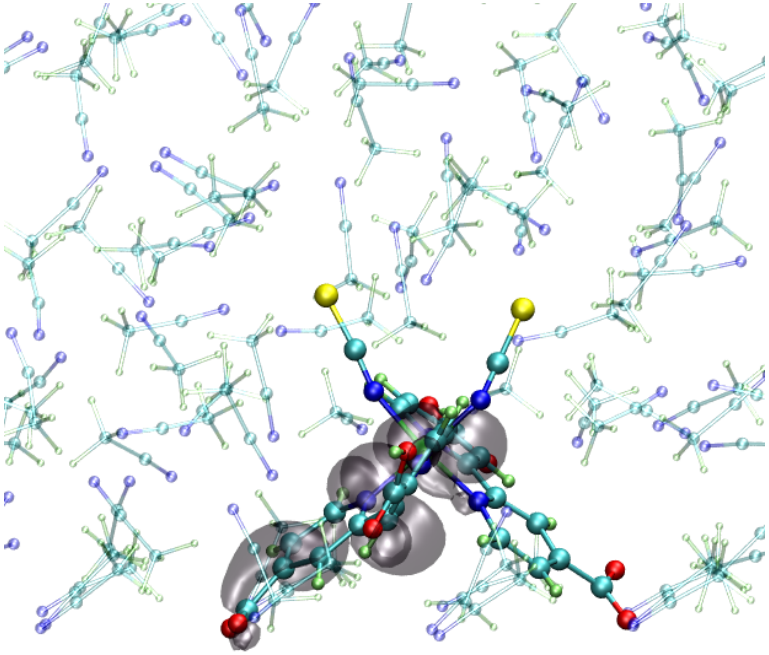
DBCSR vs. PDGEMM



DBCSR: Sparse-Dense Matrix Multiplication



State of the Art Application



- 1319 atoms, 5890 electrons, 15733 bsf
- Optimization: Function of 46 Mio. variables
- 4000 derivatives of the energy function
- 60s/MD step on 512 cores Cray XT-5
- 5 ps dynamics/week

Target Application Scenarios

- Small system - Very extensive sampling
  200-500 atoms – $10^7 - 10^9$ force evaluations
  2.1 Mio threads = 64 (replica exchange) $\times$ 16 (speculative MC) $\times$ 128 (MPI) $\times$ 16 (OpenMP)
- Medium system - Extensive sampling
  1000-5000 atoms – $10^5 - 10^6$ force evaluations
  524'288 threads = 32 (string method) $\times$ 1024 (MPI) $\times$ 16 (OpenMP)
- Large system - Minimal sampling
  10'000-100'000 atoms – $10^3 - 10^4$ force evaluations
  262'144 threads = 16384 (MPI) $\times$ 16 (OpenMP)

CP2K: Software Engineering

- $\approx$ 900'000 lines Fortran 95
  machine generated $\approx$ 200'000 lines
- Explicit subroutine interfaces- strong type checking
- Templates
- $\approx$ 1600 regression tests
  all commits are automatically tested
- External libraries: Lapack/BLAS, ScaLapack/BLACS, MPI, OpenMP, FFTW, libint

Refactoring

- Separate code into independent libraries (e.g. DBCSR)
- Use more unit testing (vs. regression testing)
- Extended OpenMP fine grain parallelization
- Identify code segments for heterogenous computing (GPGPU)
- Parallel I/O (MPI)
- Fault tolerance (MPI)

CP2K Universe

- GPL code
- Open CVS development (`cp2k.berlios.de`)
- User community (Google group, 250 members)
- Developers
  UZH, PNNL, LLNL, PSI, U Bochum, U Minnesota,
  IBM Research
- EPCC Edinburgh
  dCSE Grant: OpenMP parallelization
  Software Sustainability Institute
  Exascale Technology Centre

CP2K HP2C Team

Jürg Hutter	Prof.	UZH
Joost VandeVondele	senior postdoc	UZH
Valery Weber	senior postdoc	UZH
Urban Borstnik	postdoc	HP2C
N.N.	postdoc	HP2C
Mandes Schönherr	PhD student	HP2C/UZH
Manuel Guidon	PhD student	HP2C/UZH
Zelong Yi	PhD student	HP2C/UZH