ONETEP Parallelisation

Nicholas Hine

Condensed Matter Theory Group & Thomas Young Centre
Department of Physics & Department of Materials
Imperial College London

ONETEP Summer School, Wednesday 9th July 2008

Outline

Outline

Goals of the Architecture

Computational Resources for Total Energy of N atom system

- $O(N)$ scaling of Total Computational Effort
- $O(N)$ scaling of Total Memory Required

Parallelisation over N_p parallel processors

- $O(1/N_p)$ scaling of Total Time of Calculation
- $O(1/N_p)$ scaling of Memory per Processor
- Efficient Load Balancing
- Efficient Communications

Goals of the Architecture

Computational Resources for Total Energy of N atom system

- $O(N)$ scaling of Total Computational Effort
- $O(N)$ scaling of Total Memory Required

Parallelisation over N_p parallel processors

- $O(1/N_p)$ scaling of Total Time of Calculation
- $O(1/N_p)$ scaling of Memory per Processor
- Efficient Load Balancing
- Efficient Communications

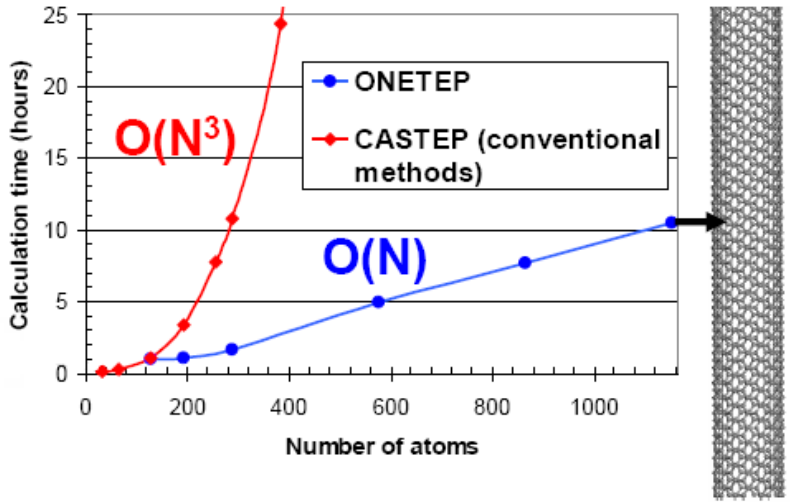
Parallel Implementation

$O(N)$ scaling with system size inherent to theoretical framework.

However, linear scaling crossover is at hundreds of atoms.

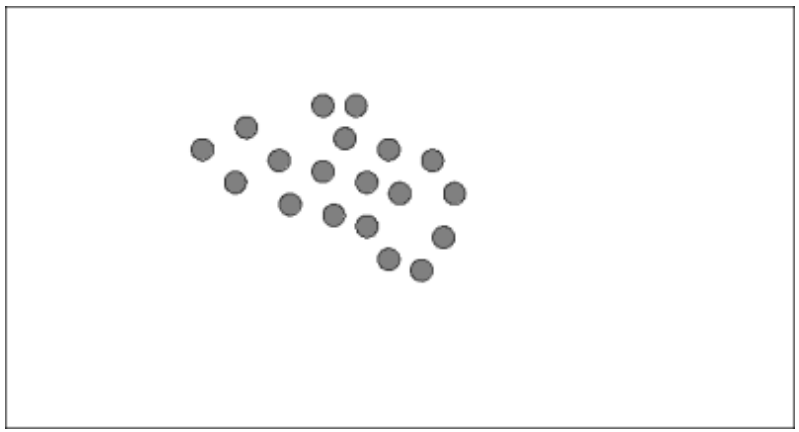
Efficient parallel implemention therefore required

Data and computational tasks must be distributable to arbitrary number of processors.

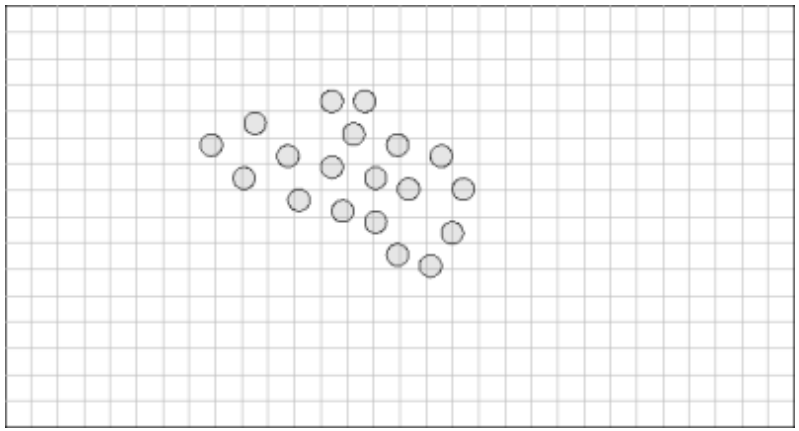


Outline

Data Parallelization



Atom Data

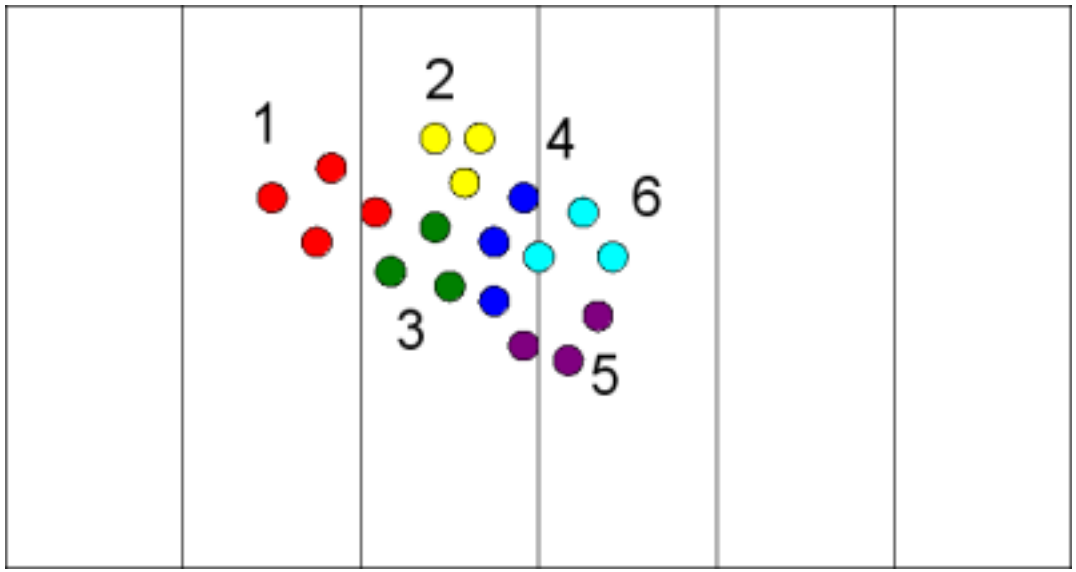


Cell Data

Skylaris, Haynes, Mostofi, Payne, Phys. Stat. Sol. (b) **243**, 973 (2006).

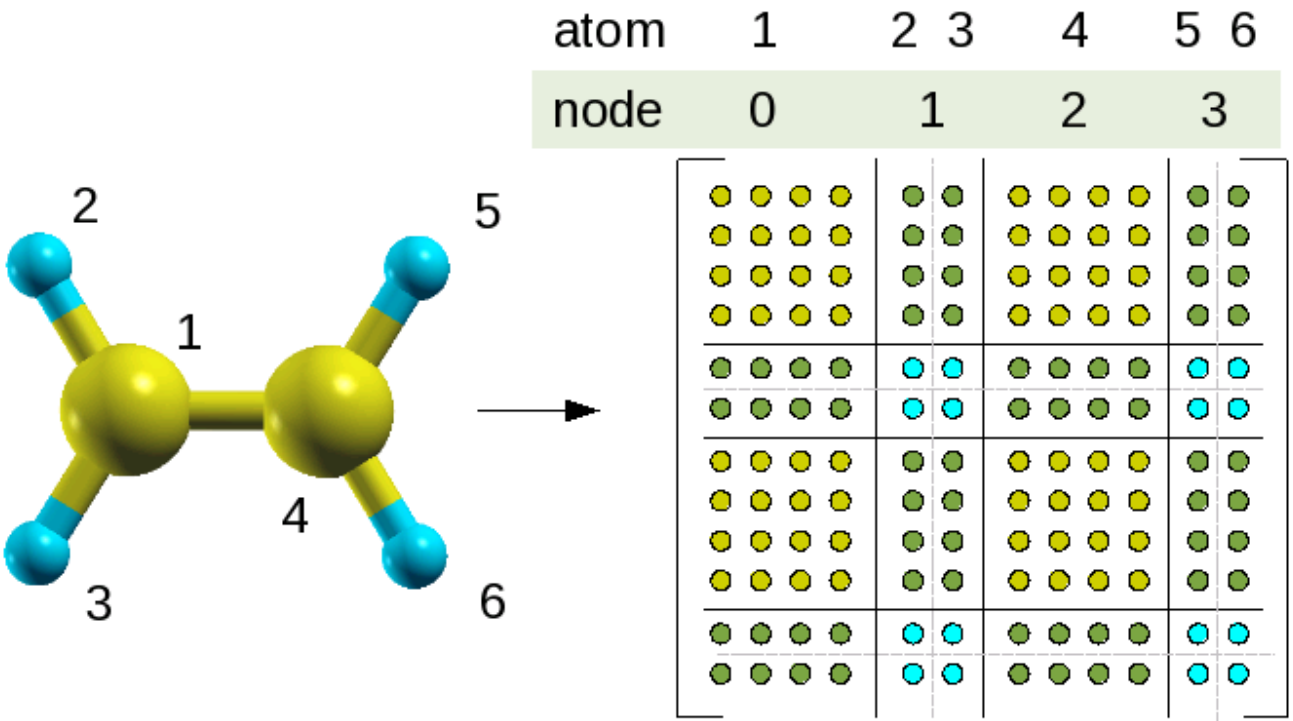
Atom, NGWF & Sparse Matrix Data

Distribute storage of atom-related data between processors
NGWFs follow distribution of atoms



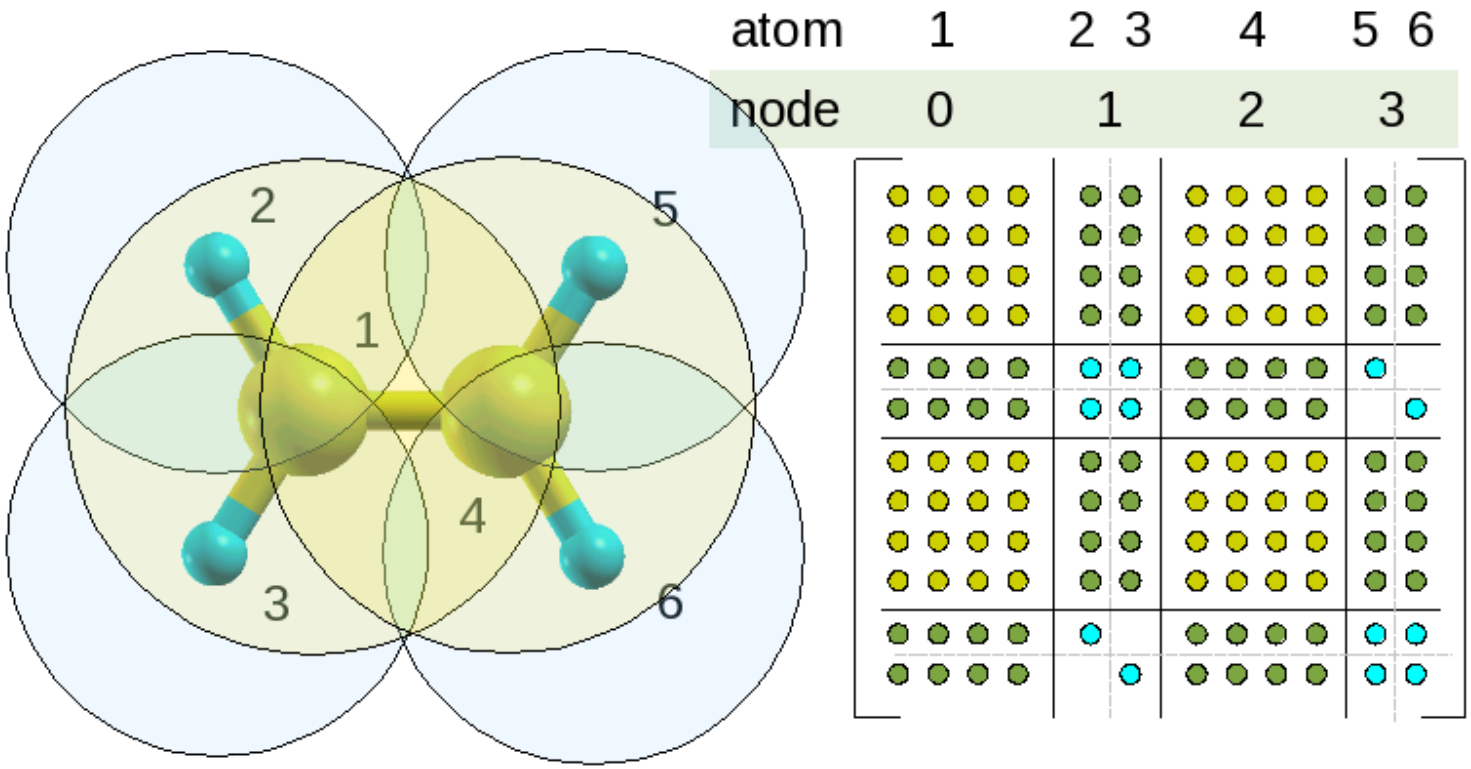
Matrix Data

Distribution of Matrix Data follows that of NGWFs



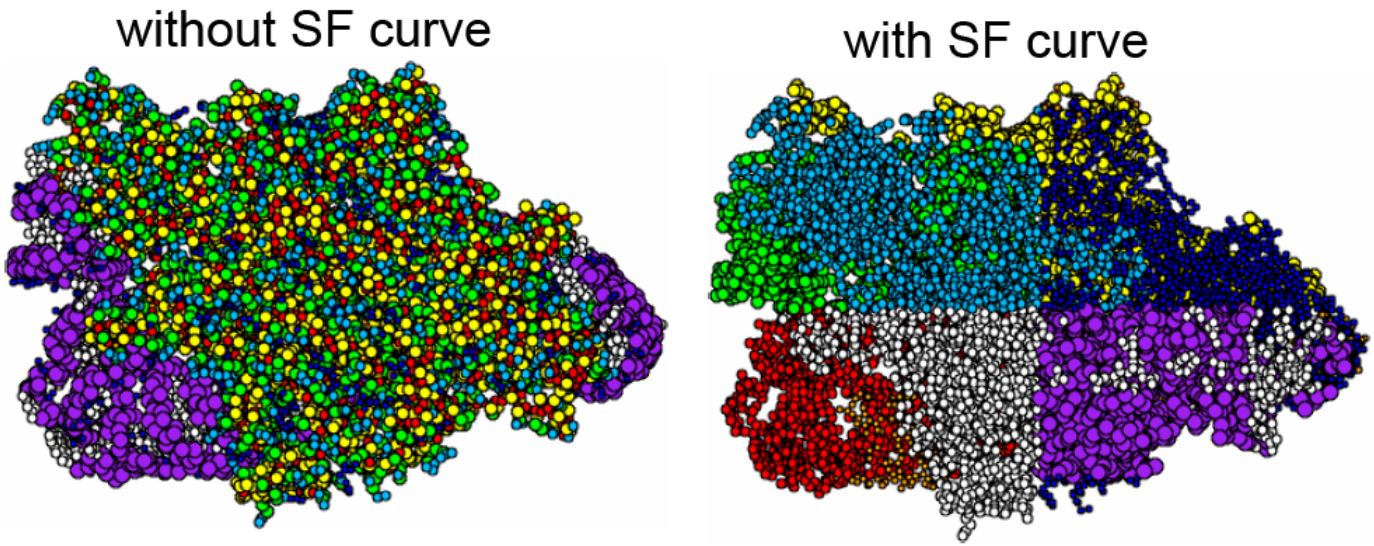
Sparsity

NGWF cutoffs determine sparsity pattern of **S**.
Kernel cutoff determines sparsity pattern of **K**.



Space Filling Curves

Space Filling Curve used to distribute atoms between processors.

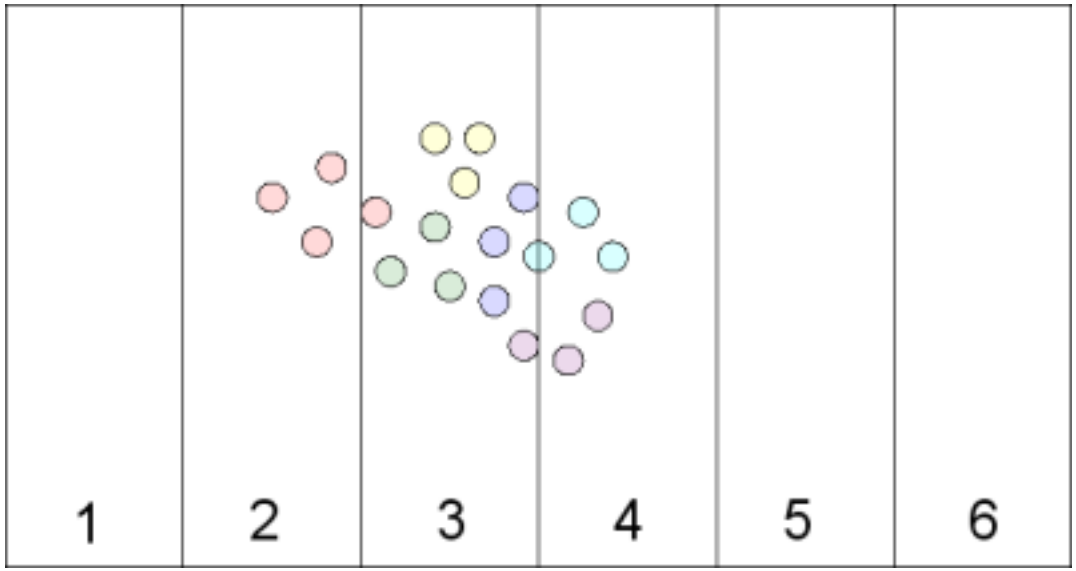


Aim: Minimise communication by maximising number of overlapping NGWFs close to the diagonal.

Whole-Cell Data

Calculations with huge supercells require whole-cell real- and reciprocal space data (densities, potentials, structure factors) to be distributed between processors.

Arrays of whole-cell data divided into parallel slabs:

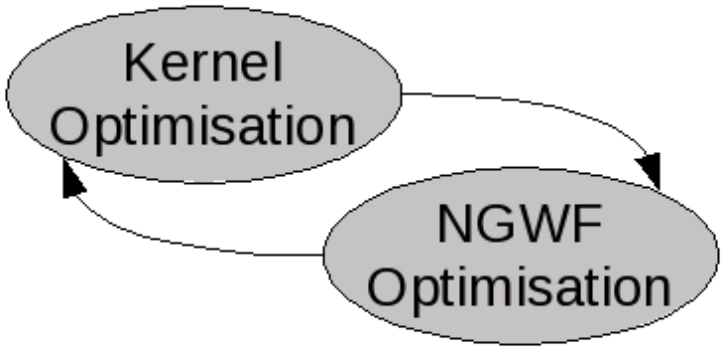


Outline

Tasks

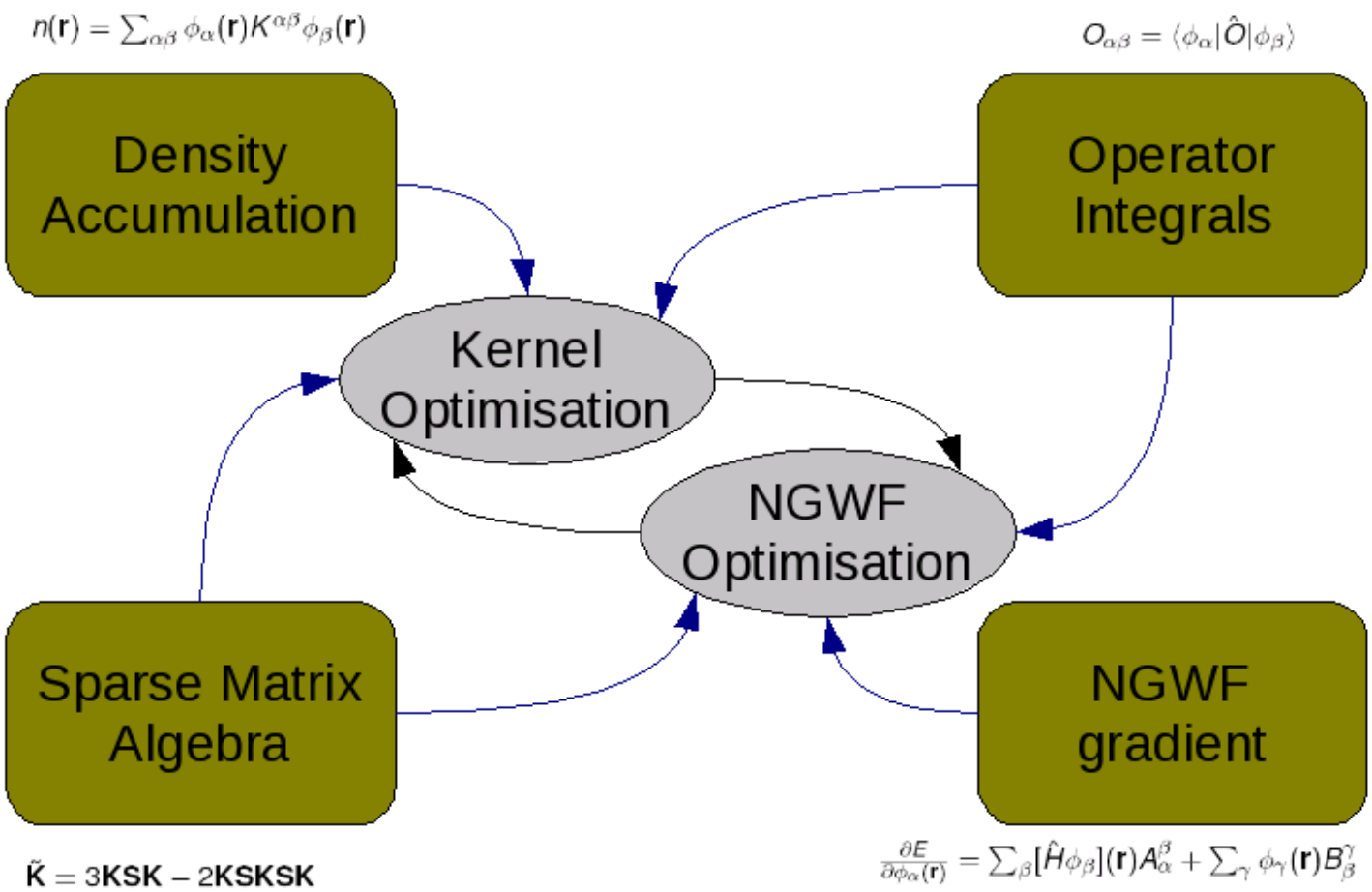
Main Total Energy Minimisation loop consists of two nested optimisations:

$$L[\{\phi_\alpha\}] = \min_{\mathbf{K}} E[\mathbf{K}, \{\phi_\alpha\}]$$



$$E_{\min} = \min_{\{\phi_\alpha\}} L[\{\phi_\alpha\}]$$

Tasks


$n(\mathbf{r}) = \sum_{\alpha\beta} \phi_\alpha(\mathbf{r}) K^{\alpha\beta} \phi_\beta(\mathbf{r})$
$O_{\alpha\beta} = \langle \phi_\alpha | \hat{O} | \phi_\beta \rangle$
Density Accumulation
Operator Integrals
Kernel Optimisation
NGWF Optimisation
Sparse Matrix Algebra
NGWF gradient
$\tilde{\mathbf{K}} = 3\mathbf{KSK} - 2\mathbf{KSKSK}$
$\frac{\partial E}{\partial \phi_\alpha(\mathbf{r})} = \sum_\beta [\hat{H}\phi_\beta](\mathbf{r}) A^\beta_\alpha + \sum_\gamma \phi_\gamma(\mathbf{r}) B^\gamma_\beta$

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

Main tasks

- FFTBox Fourier Transforms
  - Density, Local Potential, Kinetic, NL PSP projectors
- 'Batch' Row sums
  - Density, NGWF Gradient, Operator Integrals
- Local Potential Extraction / Density Deposition
  - Density, Local Potential, NGWF Gradient
- NL-PSP Projector integrals (esp in NGWF gradient)
  - NGWF gradient
- Sparse Algebra
  - Kernel Optimization, NGWF gradient
- Whole Cell FFTs (v minor).

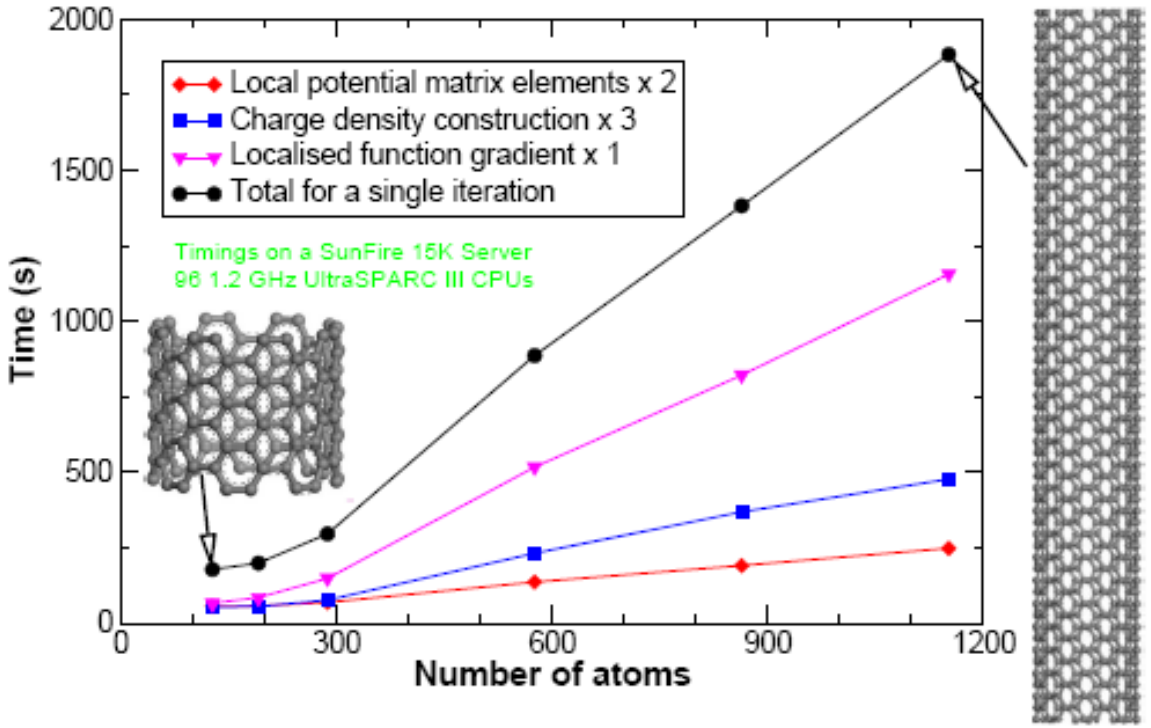
Time-Limiting Tasks

Different tasks dominate in different regimes

- Small to medium isolated systems (biomolecules, clusters etc)
  - Usually FFT limited.
- Densely overlapping solids (large NGWF radii)
  - Dominated by NGWF overlap sums
- Enormous isolated systems
  - Can become dominated by sparse algebra (terms like KSKSK are barely sparse at all even for an enormous system).

Time-Limiting Tasks

As an example, breakdown of time for an ideal system:



Communications Strategy

‘Hide’ Comms behind Computation: messages passed in background while computational work is executing.

‘Non-blocking’ routines (minimal use of broadcasting).

Ongoing work to improve efficiency of Comms

NGWF Overlap Sums

Density Accumulation, Operator Integrals, and part of the NGWF gradient all share a similar structure:

Require $\phi_\beta(\mathbf{r})[\hat{O}\phi_\alpha](\mathbf{r})$ for all overlapping $\phi_\alpha(\mathbf{r})$, $\phi_\beta(\mathbf{r})$

Density requires $\sum_{\alpha,\beta} \phi_\beta(\mathbf{r}) K^{\alpha\beta} \phi_\alpha(\mathbf{r})$

Kinetic contribution to $\mathbf{H}$ requires $\int d\mathbf{r}\, \phi_\beta(\mathbf{r}) \nabla^2 \phi_\alpha(\mathbf{r})$

Local potential matrix requires $\int d\mathbf{r}\, \phi_\beta(\mathbf{r})\, V_{\text{loc}}(\mathbf{r}) \phi_\alpha(\mathbf{r})$

Recent Algorithmic Improvements

- ‘Planned’ sums for NGWF overlap sums.
- Reduction of multi-stage NWGF deposition to single-step.
- Matrix Comms improvements.
- ‘Dense’ Matrix algebra.

Outline

Efficient Load Balancing

- In homogeneous systems, try to have Ncpus close to a divisor of Natoms.

  Less important the more inhomogeneous the system is

- Make sure all cpus run same speed

  On multi-core systems, ensure the MPI tasks optimally distributed between cores: memory bandwidth can be enough of a problem that better results can sometimes be obtained on just, eg two cores of a quad core chip).

Single Node Performance

- Maximise single node FFT performance.

  FFTW3 is usually safest, but results from it can be very compiler optimisation dependent. Intel MKL is very good where available. Other vendor specific versions are supported (and more can be if needed).

- On multicore machines, do not expect eg 4$\times$ performance from 4 cores of a quad-core system compared to 1 core.
- However, 4 $\times$ 4 cores can in principal give nearly 4$\times$ the performance of 4 cores.

Communications

- Batch System: Increase size of a ‘batch’ of NGWFs.

  Reduces total comms at cost of increased memory requirements (choose as high as you can, generally).

  Use *density_batch_size*, *locpot_int_batch_size*, *kinetic_int_batch_size*, *ngwf_grad_batch_size*

- ’Planned’ sums very often faster. use *planned_sums* : T
- Check the performance of dense matrices, with *dense_matrices* : T.

  Especially for large kernel cutoffs, sparsity levels to make sparse algebra worthwhile are only reached at tens of thousands of atoms.

  NB for small systems, sparse algebra is a negligible part of the calculation so test it with 1 iteration of something big.

Summary

- Highly efficient parallel scaling to thousands of atoms spread over hundreds of processors.
- Development work ongoing to improve parallel efficiency.