Fortran Expo: 15 Jun 2012

Conquest

— order N ab initio Electronic Structure simulation code for quantum mechanical modelling in large scale

Lianheng Tong

Overview

- Overview of Conquest project
- Brief Introduction to DFT
- Structure of Conquest
- Parallelisation and Linear Scaling
- Applications

Conquest

- Density Functional Theory Code: simulating electronic structures of materials
- Linear Scaling
- Written in Fortran 90
- Pure MPI parallelisation at moment, OpenMP+MPI hybrid in works
- Available as BETA release on HPC platforms: HECToR, K-Computer, NIMS simulator
- Supported by dCSE, HECToR
- Website: http://www.order-n.org/

Developers

- David Bowler (UCL)
- Lianheng Tong (UCL)
- Conn O'Rourke (UCL)
- Umberto Terranova (UCL)
- Veronika Brazdova (UCL)
- Mike Gillan (UCL)
- Lionel Truflandier (Bordeaux, France)
- Tsuyoshi Miyazaki (NIMS, Japan)
- Michiaki Arita (NIMS, Japan)

Density Functional Theory

- Nano-scale: Quantum Mechanics becomes important.
- Solving Schrödinger equation

$$\hat{H}\psi(\mathbf{r}) = E\psi(\mathbf{r})$$

- Multiparticle wavefunction approach: Number of parameters: $M = p^{3N}, \quad 3 \leq p \leq 10$
- Ground state electron charge density enough to fully system — Density Functional Theory (DFT). Nobel Price in Chemistry 1998.

Density Functional Theory

- To find Ground state energy, instead of solving

$$\hat{H}\psi(\mathbf{r}) = E\psi(\mathbf{r})$$

- We write:

$$E_{\mathrm{KS}} = E_{\mathrm{K}}[\rho(\mathbf{r})] + E_{\mathrm{H}}[\rho(\mathbf{r})] + E_{\mathrm{xc}}[\rho(\mathbf{r})] + E_{\mathrm{pseudo}}([\rho(\mathbf{r})], \mathbf{R}_i) + E_{\mathrm{ion}}(\mathbf{R}_i, \mathbf{R}_j)$$

and minimise the total energy with respect to density $\rho(\mathbf{r})$.

- Practical to work with Hamiltonian

$$\mathbf{H} = -\frac{1}{2}\boldsymbol{\nabla}^2 + \mathbf{V}_{\mathrm{H}} + \mathbf{V}_{\mathrm{xc}} + \mathbf{V}_{\mathrm{pseudo}}, \quad \mathbf{V}_{\mathrm{X}} = \frac{\delta E_{\mathrm{X}}}{\delta\rho(\mathbf{r})}$$

and minimise $E = \mathrm{tr}(\boldsymbol{\rho}\mathbf{H})$

Conquest

- We use density matrix

$$\rho(\mathbf{r}_1, \mathbf{r}_2) = \sum_n f_n \psi_n(\mathbf{r}_1)\psi_n^*(\mathbf{r}_2), \quad \rho(\mathbf{r}) = \rho(\mathbf{r}, \mathbf{r})$$

- We work directly with it using support functions

$$\rho(\mathbf{r}_1, \mathbf{r}_2) = \sum_{i\alpha j\beta} \phi_{i\alpha}(\mathbf{r}_1) K^{i\alpha j\beta} \phi_{j\beta}(\mathbf{r}_2)$$

- Locality imposed via radius of support functions and cutoff on $K^{i\alpha j\beta}$

- Support functions represented by basis functions

$$\phi_{i\alpha} = \sum_n C_{i\alpha}^n \chi^n(\mathbf{r})$$

Basis Sets

- We now have two basis sets in Conquest:
  - Blips (piece-wise cubic splines defined on a grid moving with atoms)

$$\theta_0(x) = \begin{cases} 1 - \frac{3}{2}x^2 + \frac{3}{4}|x|^3 & \text{if} \quad 0 < |x| < 1 \\ \frac{1}{4}(2 - |x|)^3 & \text{if} \quad 1 < |x| < 2 \\ 0 & \text{if} \quad 2 < |x| \end{cases}$$

$$\Theta\left(\mathbf{r}\right) = \theta_0(x)\theta_0(y)\theta_0(z)$$

  - Pseudo Atomic Orbitals, PAOs (radial term multiplied by spherical harmonic)
- PAOs allow many analytic or local calculations (efficient)
- Blips allow systematically improving basis set

Minimisation Procedures

- Three levels:
  - Innermost loop: minimise $E = \mathrm{tr}(\mathbf{KH})$ w.r.t. $\mathbf{K}$
  - Middle loop: seek self-consistent charge and potential
  - Outer loop: minimise energy w.r.t. basis set—vary $C_{i\alpha}^{n}$

Linear Scaling

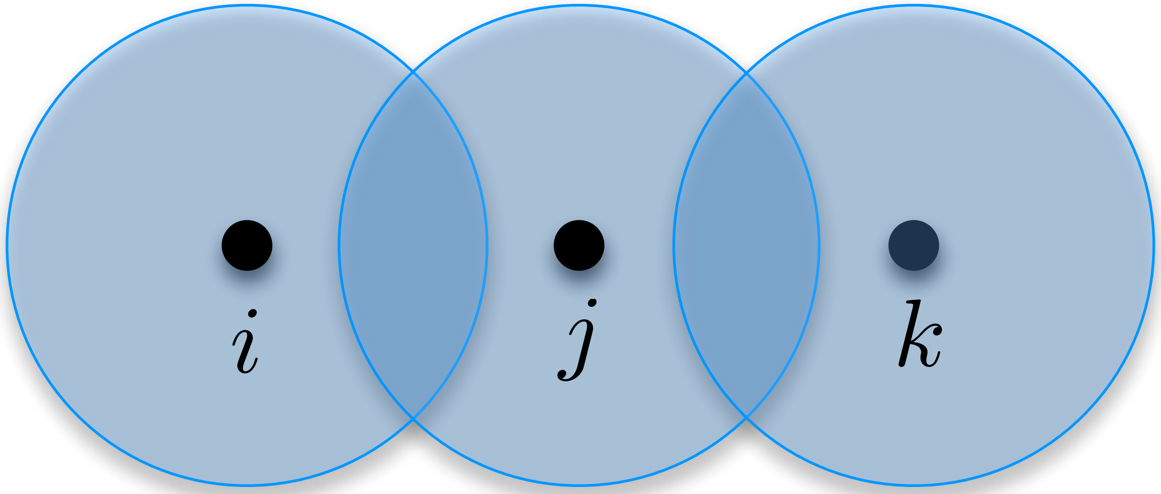
- Normal DFT calculations: Cost is of order N^3 to N^2
- Large systems: needs linear ordering
- *Localisation* is key: everything must have finite range
- Density is local: $\rho(\mathbf{r}_1, \mathbf{r}_2) \to 0, \quad \|\mathbf{r}_1 - \mathbf{r}_2\| \to \infty$
- We *impose*: $\rho(\mathbf{r}_1, \mathbf{r}_2) = 0, \quad \|\mathbf{r}_1 - \mathbf{r}_2\| \geq R_{\mathrm{cut}}$
- Interactions are ranged

Linear Scaling

- In practice each atom has a *halo*
- Support functions are *strictly local*

$$\phi_{i\alpha}(\mathbf{r}) = 0, \|\mathbf{r} - \mathbf{R}_i\| \geq R_{\mathrm{c}}^{\phi}$$

- *Only non-zero matrix elements are stored.*
- *Sparse matrix multiplication*



$$L^{i\alpha j\beta} \neq 0$$

$$L^{j\beta k\gamma} \neq 0 \quad R_{\mathrm{cut}} = \max(|R_i^{\mathrm{halo}} - R_j^{\mathrm{halo}}|)$$

$$L^{i\alpha k\gamma} = 0$$

Parallelisation

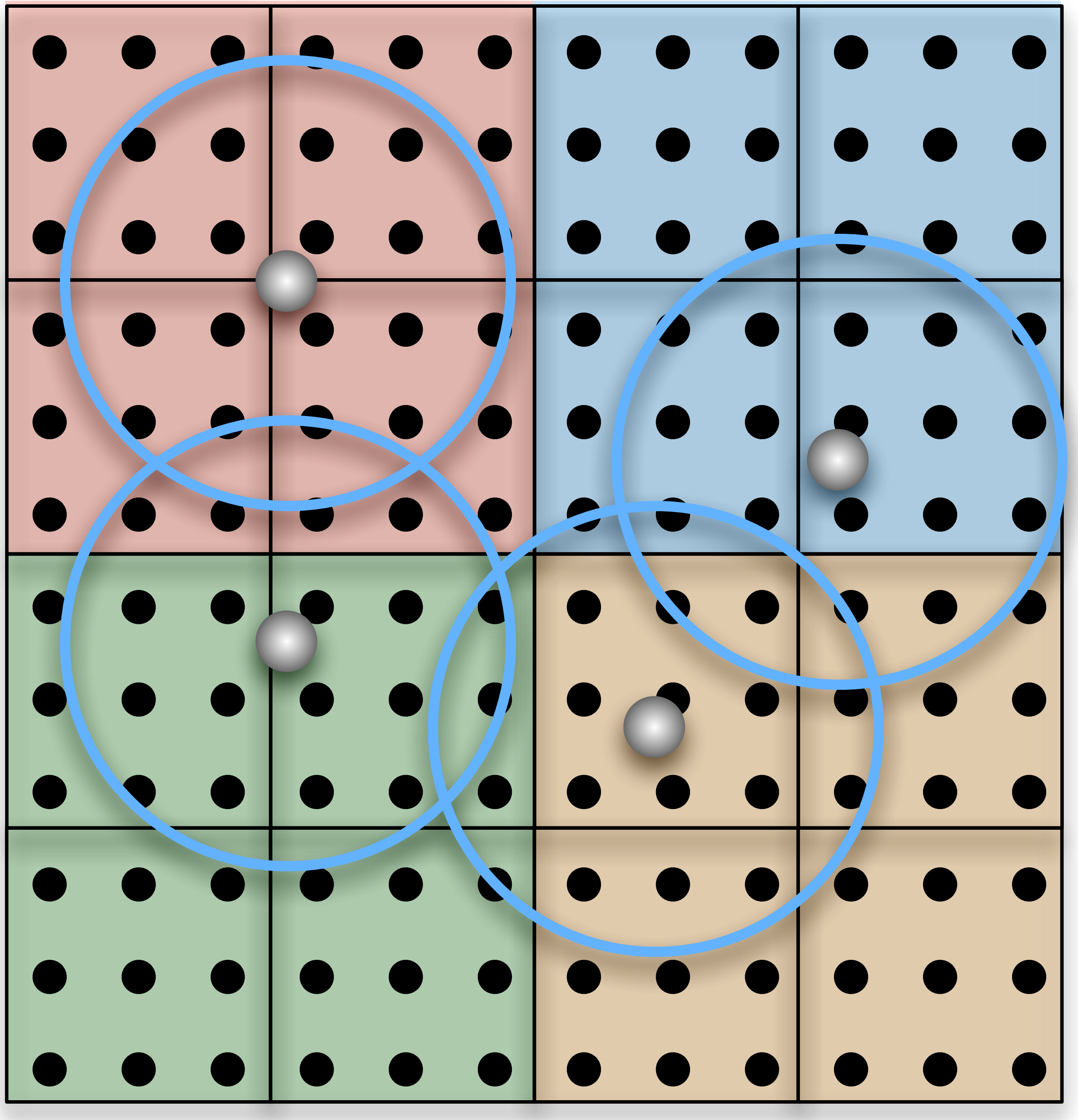
- There are three main areas in Conquest
  - Integration (on 3D integration/FFT grid)

$$H_{i\alpha j\beta} = \int \mathrm{d}^3\mathbf{r}\, \phi_{i\alpha}(\mathbf{r})\hat{H}\phi_{j\beta}(\mathbf{r})$$

$$S_{i\alpha j\beta} = \int \mathrm{d}^3\mathbf{r}\, \phi_{i\alpha}(\mathbf{r})\phi_{j\beta}(\mathbf{r})$$

  - Matrix multiplication
  - FFTs (on 3D integration/FFT grid)

Parallelisation: Integration/FFT grid

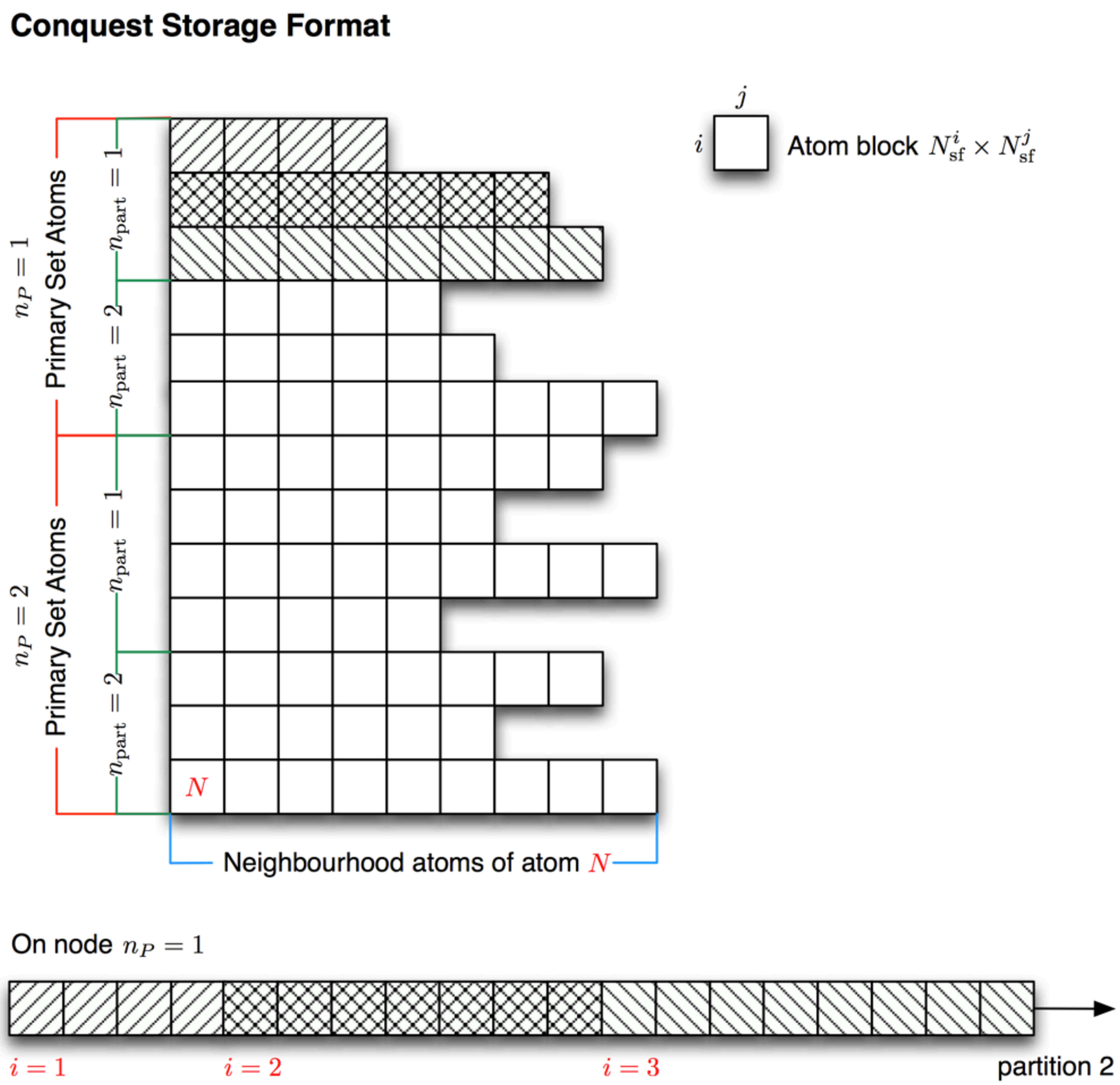


- Grid divided into *blocks* (for efficiency)
- Space divided into partitions, each partition is then allocated to a node
- Set of all blocks in charge by a node is called a *domain*
- A node stores value of *all* support functions at grid points within its domain
- Each node is responsible for *partial contributions* to *all* integrals (matrix elements) involving support functions touching its domain. Calculations are local

Parallelisation: Matrices

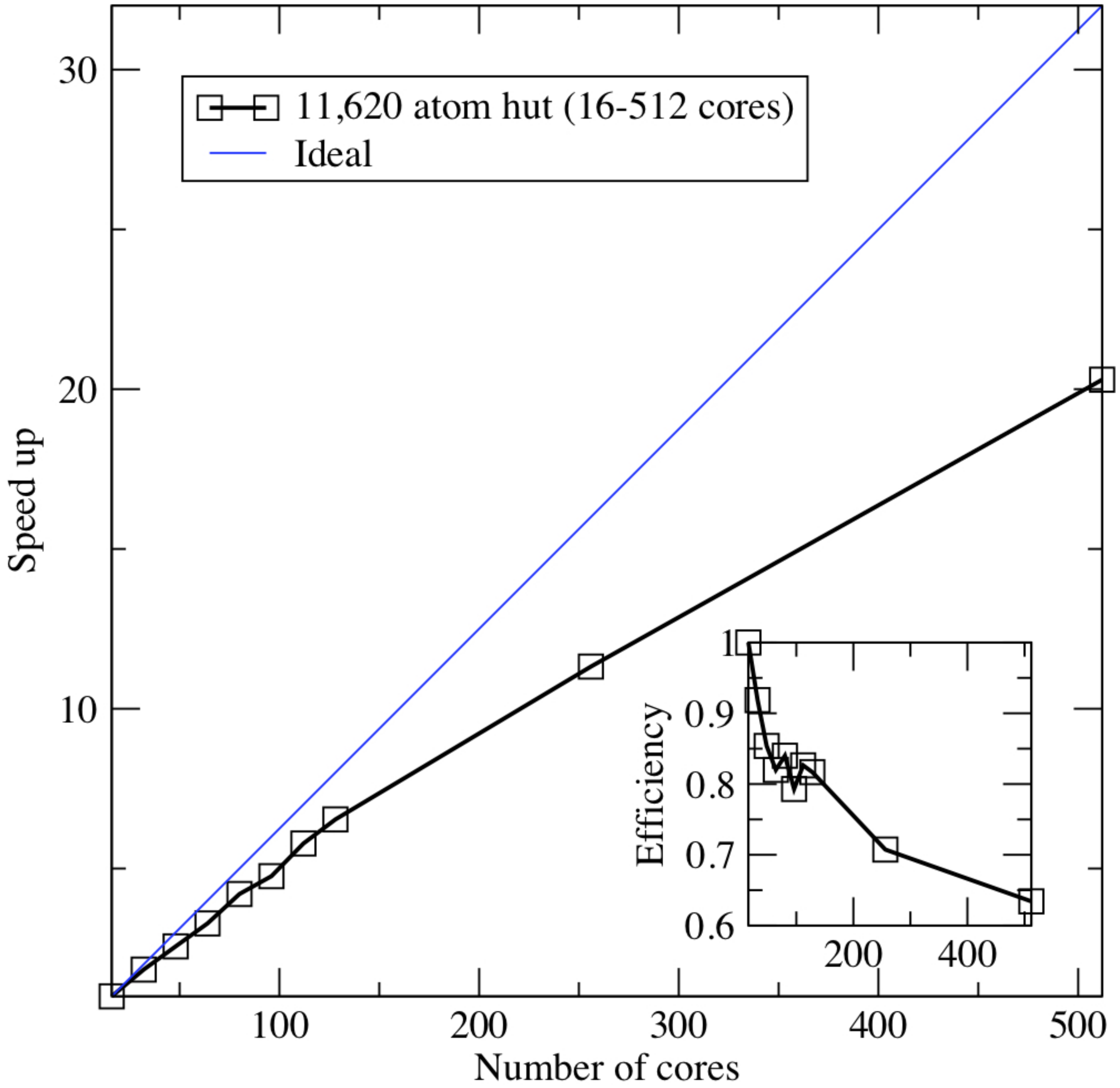
- Divides space into partitions, each partition has a set of atoms
- Each node in charge of a set of partitions, and hence a set of atoms
- Matrix distributed to nodes according *rows* associated to the atoms in charge
- Results of integrations need to be redistributed to follow matrix format
- Matrix multiplication needs data fetch from other nodes

$$C^{i\alpha}{}_{j\beta} = \sum_{k\gamma} A^{i\alpha}{}_{k\gamma} B^{k\gamma}{}_{j\beta}$$



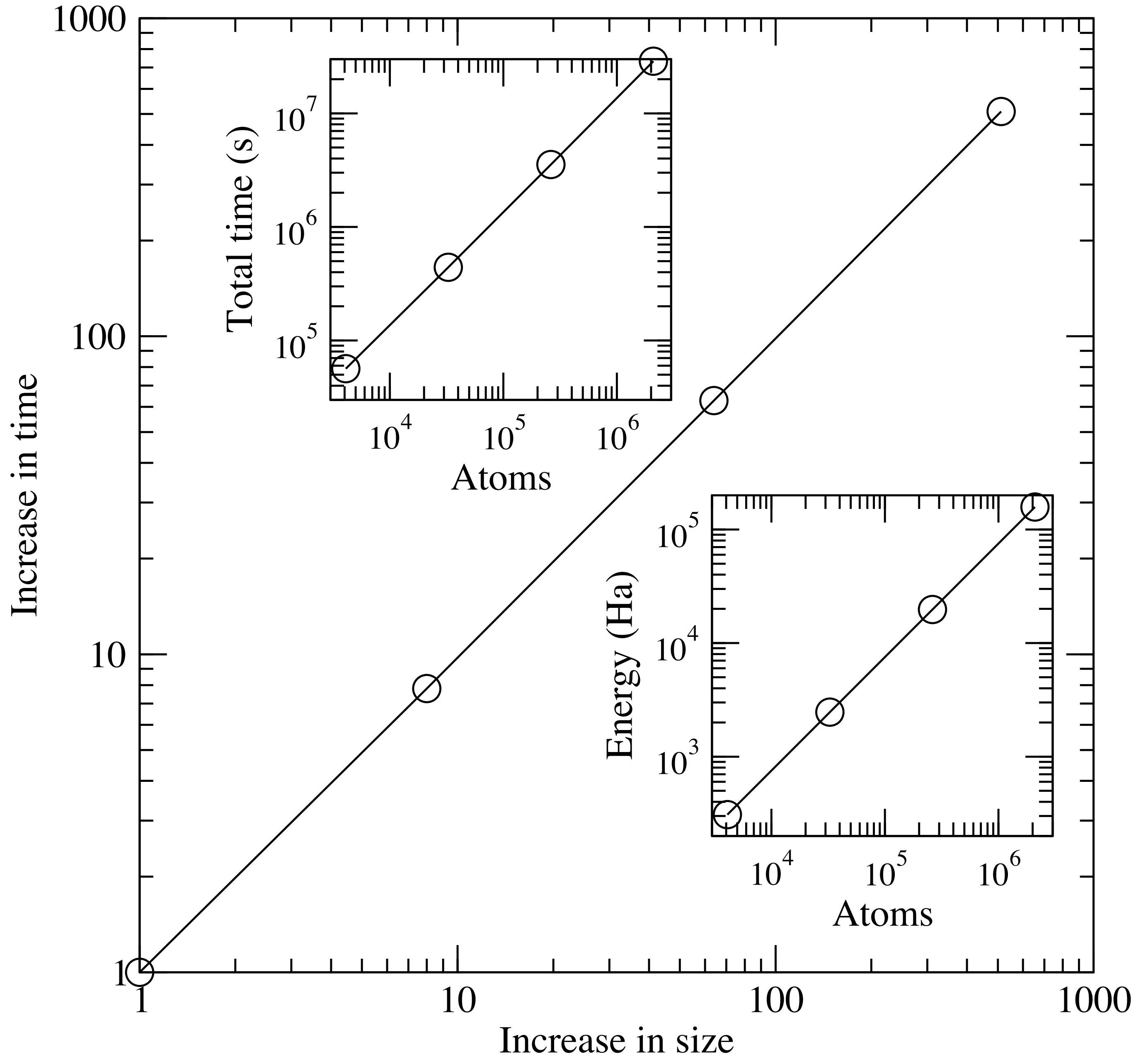
Strong Scaling

- Ge hut cluster on Si(001) surface
- 11,620 atoms in unit cell
- Using between 16 to 512 cores



Weak Scaling

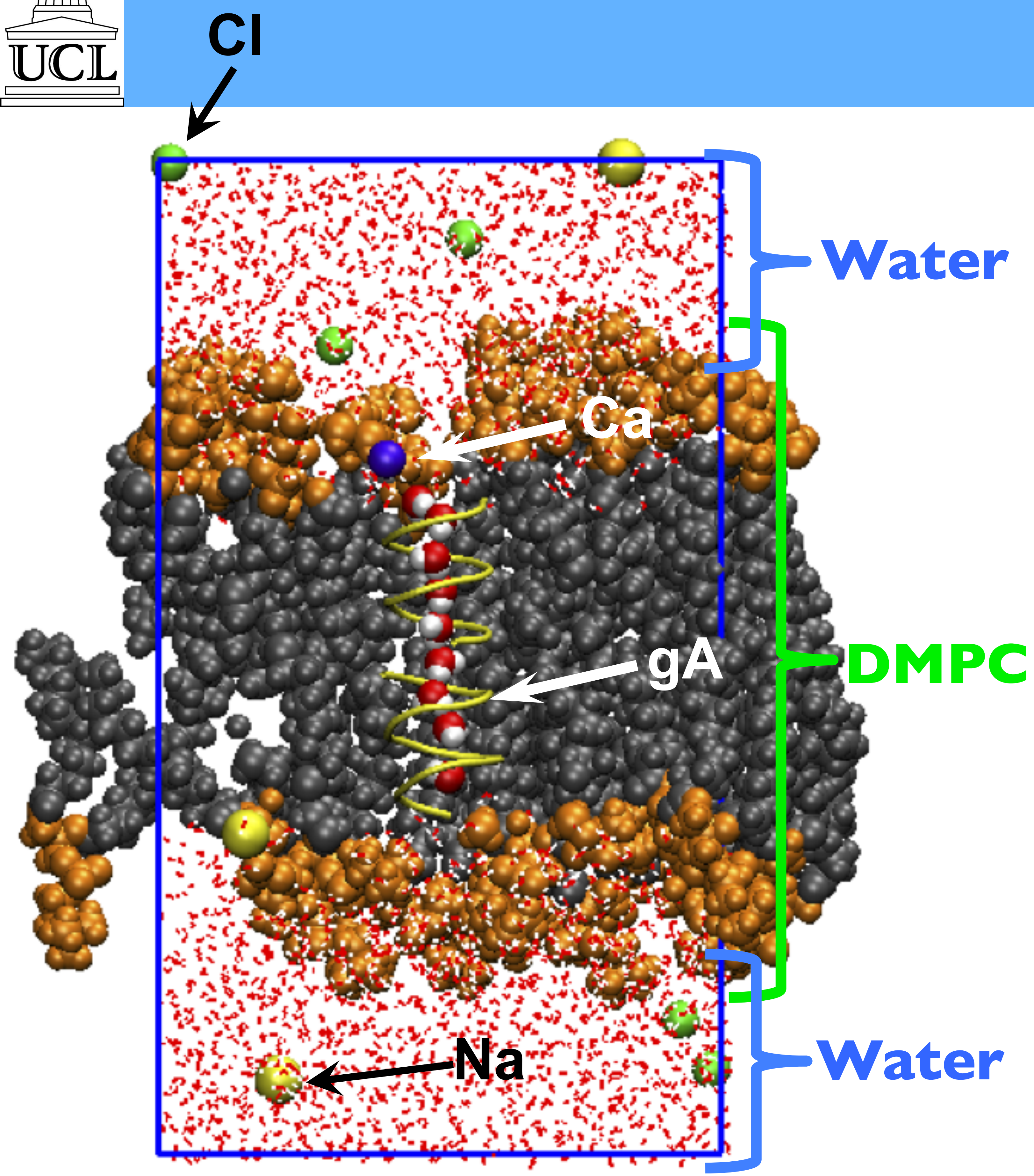
- 512 Si atoms per core (memory limited)
- Four support functions per basis function, slightly corse grid
- Self-consistency done for small cells
- Graph shows total time (sum of times on each core)



Million Atoms DFT

Atoms	Time/core (s)	Energy (Ha)	Cores
4,096	7068.878	-308.268	8
32,768	6893.759	-2,466.150	64
262,144	6931.418	-19,729.202	512
2,097,152	7032.496	-157,833.618	4096

HECToR Phase 2a



Michiaki Arita and Tsuyoshi Miyazaki, 2011

- Dimyristoyl-sn-glycero-phosphocholine
- Phospholipid that incorporates choline as head group.
- Important part of cell membrane
- Hydrophilic heads in outer surfaces of the bilayer, hydrophobic in between.
- gramicidin A ion channel allows selective ions (mono-valency cations) to pass through cell membrane
- Over 15000 atoms, full DFT (SCF) calculation

DMPC-gA in Water

- Basis: Optimised SZP from DZP (using diagonalisation, piece-wise)
- XC Functional: PBE(+D2)
- Cores: 512, NIMS Simulator 1
- Intel Xeon processor Nehalem-EP (2.8 GHz), 4 cores/node, 2.85GB per core
- SCF computation time: < 24 Hrs

Summary

- Presented some details of Conquest
  - Linear scaling DFT code
  - Excellent scaling on HPC platforms
  - Capable of performing full quantum mechanical calculations on millions of atom

Summary

- On Fortran 90
  - Reasons we choose Fortran 90
    - HPC platforms supports Fortran or C
    - Compatibility with existing libraries
    - Array arithmetics
    - Modules, interfaces and also optional variables
  - Issues (personal experiences)
    - Some compilers can be too forgiving
    - Allows F77 syntax, which some times can lead to confusion
    - Question: automatic arrays vs. allocatable arrays—stack or heap?

Acknowledgements

- Conquest: David Bowler, Tsuyoshi Miyazaki, Mike Gillan
- Funding:
  - EPSRC
  - HECToR (NAG) dCSE

Thank You!