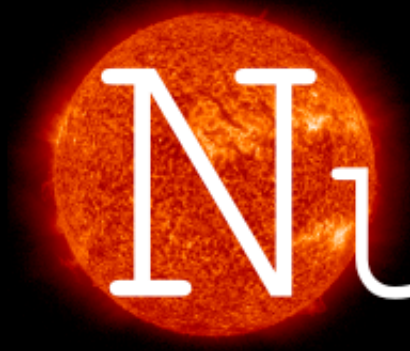


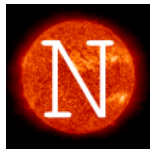
Nu-FuSE

Porting the DFT code CASTEP to GPGPUs

Toni Collis

acollis@epcc.ed.ac.uk

EPCC, University of Edinburgh

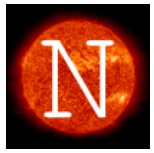


http://www.nu-fuse.com

CASTEP and GPGPUs

Outline

- Why are we interested in CASTEP and Density Functional Theory codes.
- Brief introduction to CASTEP underlying computational problems.
- The OpenACC implementation

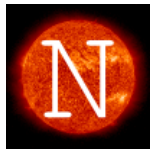


CASTEP: a DFT code

http://www.nu-fuse.com

- CASTEP is a commercial and academic software package
- Capable of Density Functional Theory (DFT) and plane wave basis set calculations.
- Calculates the structure and motions of materials by the use of electronic structure (atom positions are dictated by their electrons).
- Modern CASTEP is a re-write of the original serial code, developed by Universities of York, Durham, St. Andrews, Cambridge and Rutherford Labs

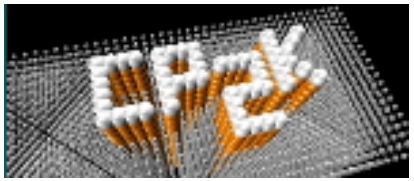
cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY



http://www.nu-fuse.com

CASTEP: a DFT code

- DFT/*ab initio* software packages are one of the largest users of HECToR (UK national supercomputing service, based at University of Edinburgh).
- Codes such as CASTEP, VASP and CP2K. All involve solving a Hamiltonian to explain the electronic structure.
- DFT codes are becoming more complex and with more functionality.



CASTEP

HECToR

http://www.nu-fuse.com

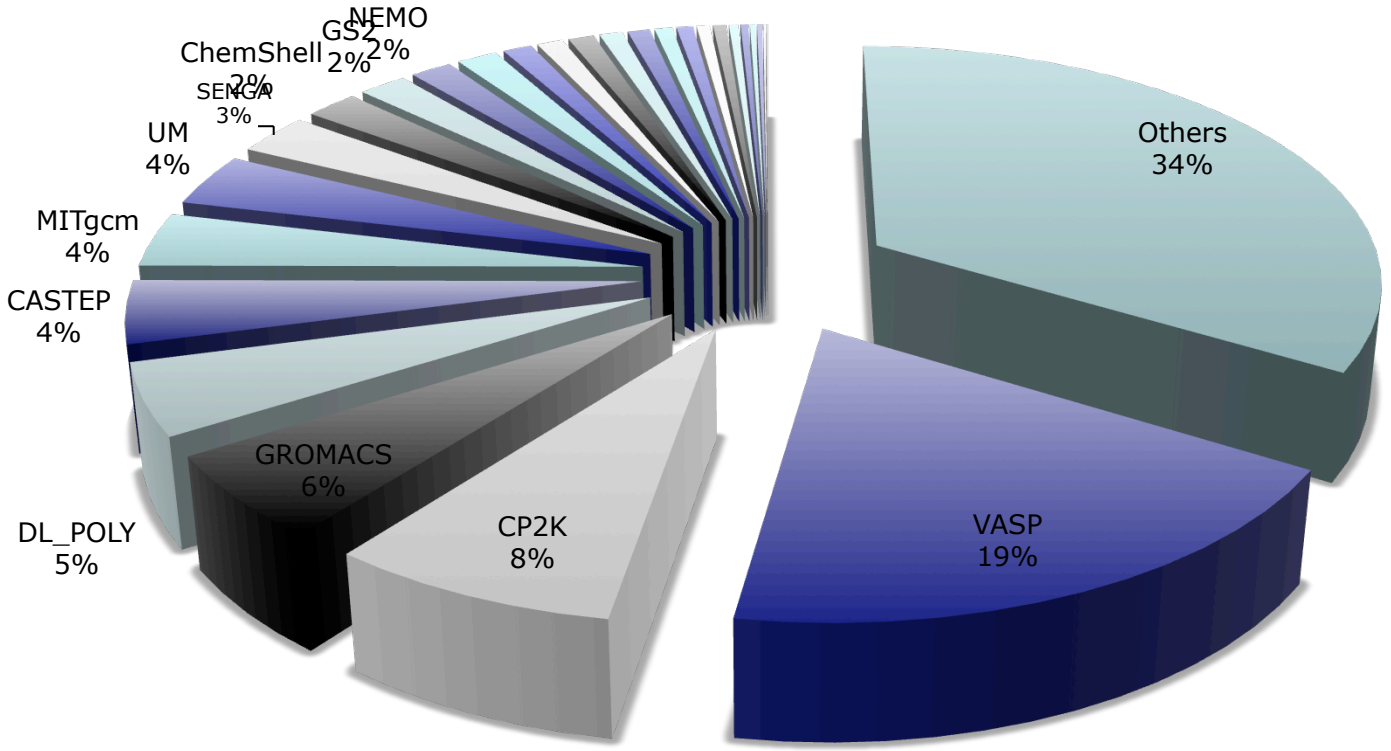
- UK National HPC Service
- Currently 30-cabinet Cray XE6 system
  - 90,112 cores
- Each node has
  - 2×16-core AMD Opterons
    (2.3GHz Interlagos)
  - 32 GB memory
- Peak of over 800 TF and 90 TB of memory

HECToR usage statistics

http://www.nu-fuse.com

Phase 3 statistics (Nov 2011 - Apr 2013)

Ab initio codes (VASP, CP2K, CASTEP, ONETEP, NWChem, Quantum Espresso, GAMESS-US, SIESTA, GAMESS-UK, MOLPRO)

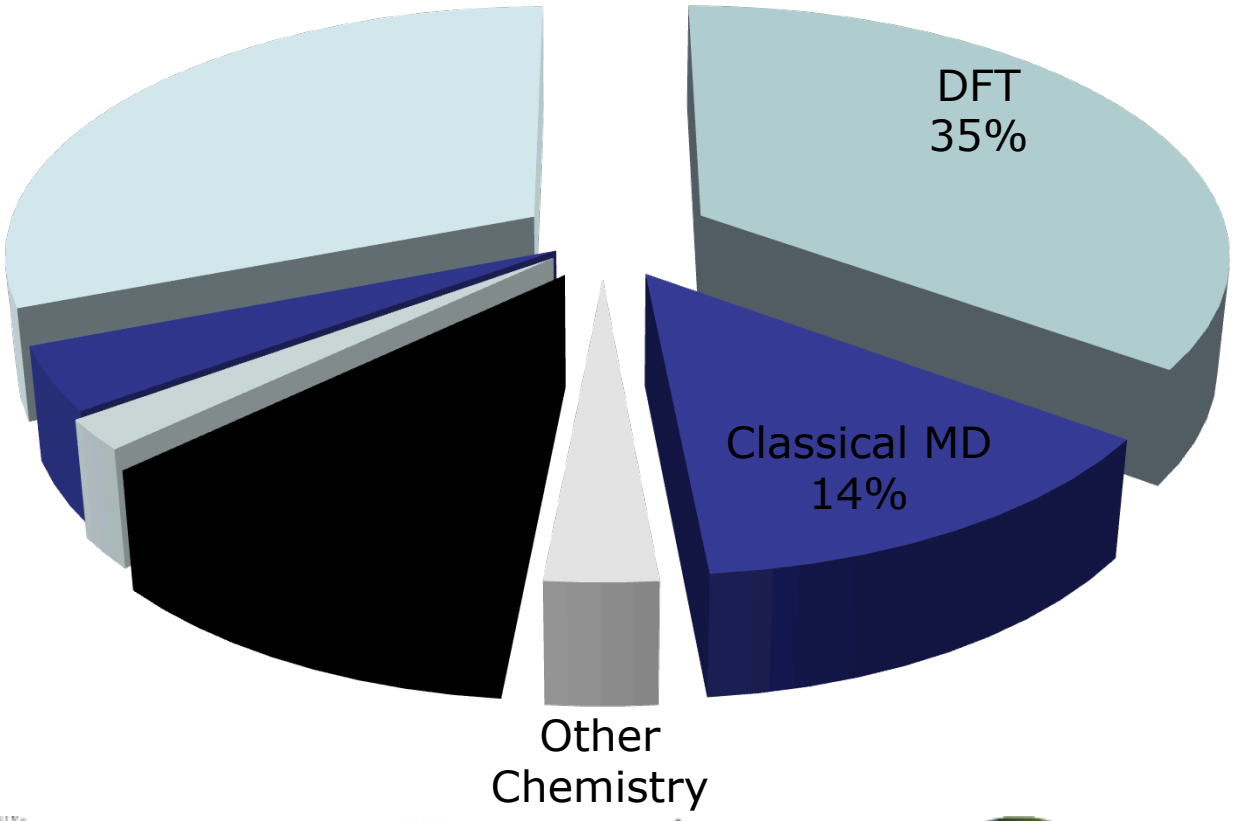


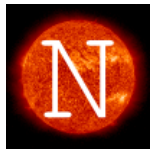
HECToR usage statistics

http://www.nu-fuse.com

Phase 3 statistics (Nov 2011 - Apr 2013)

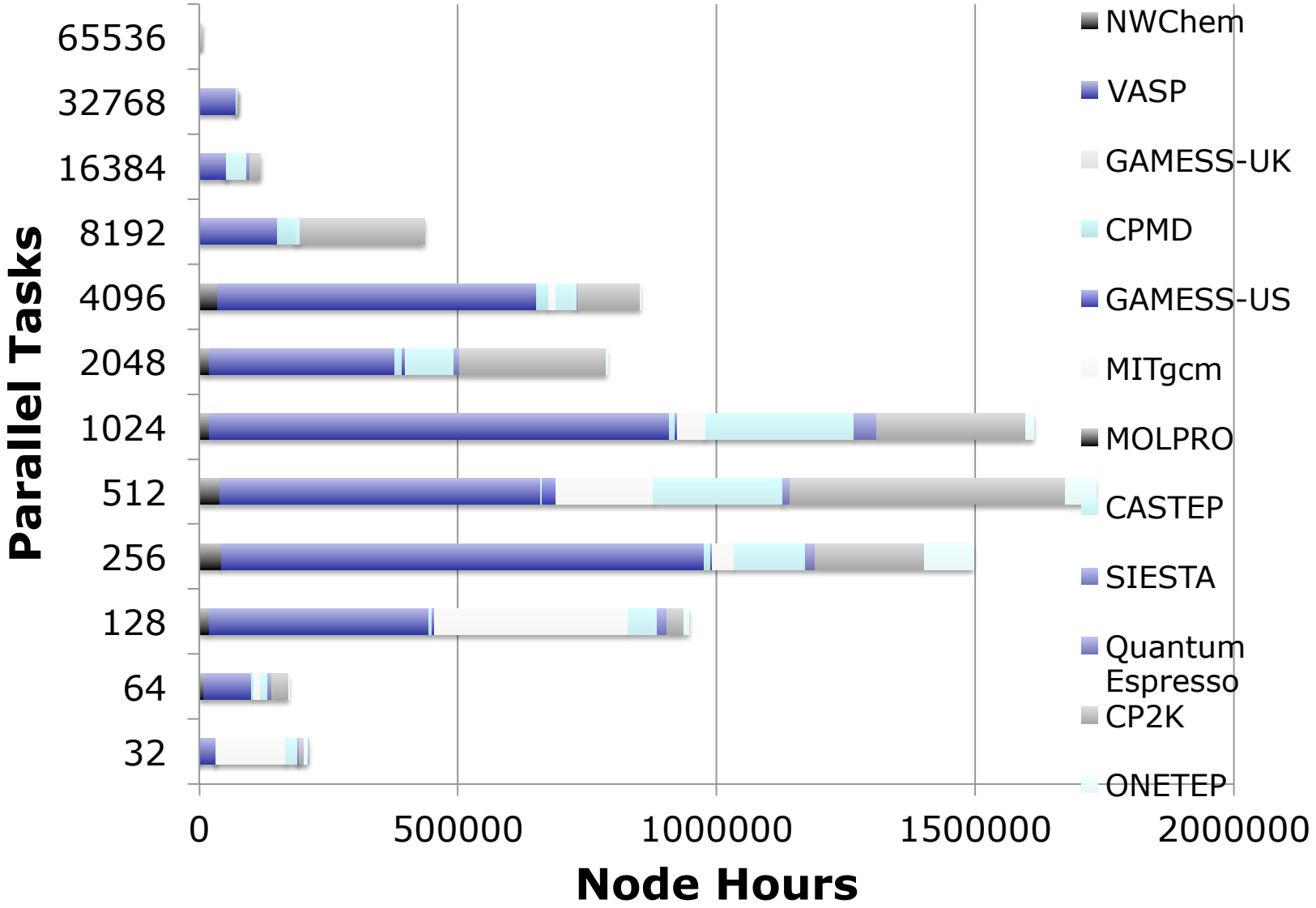
35% of the Chemistry software on HECToR is using DFT methods.

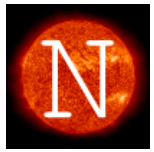




http://www.nu-fuse.com

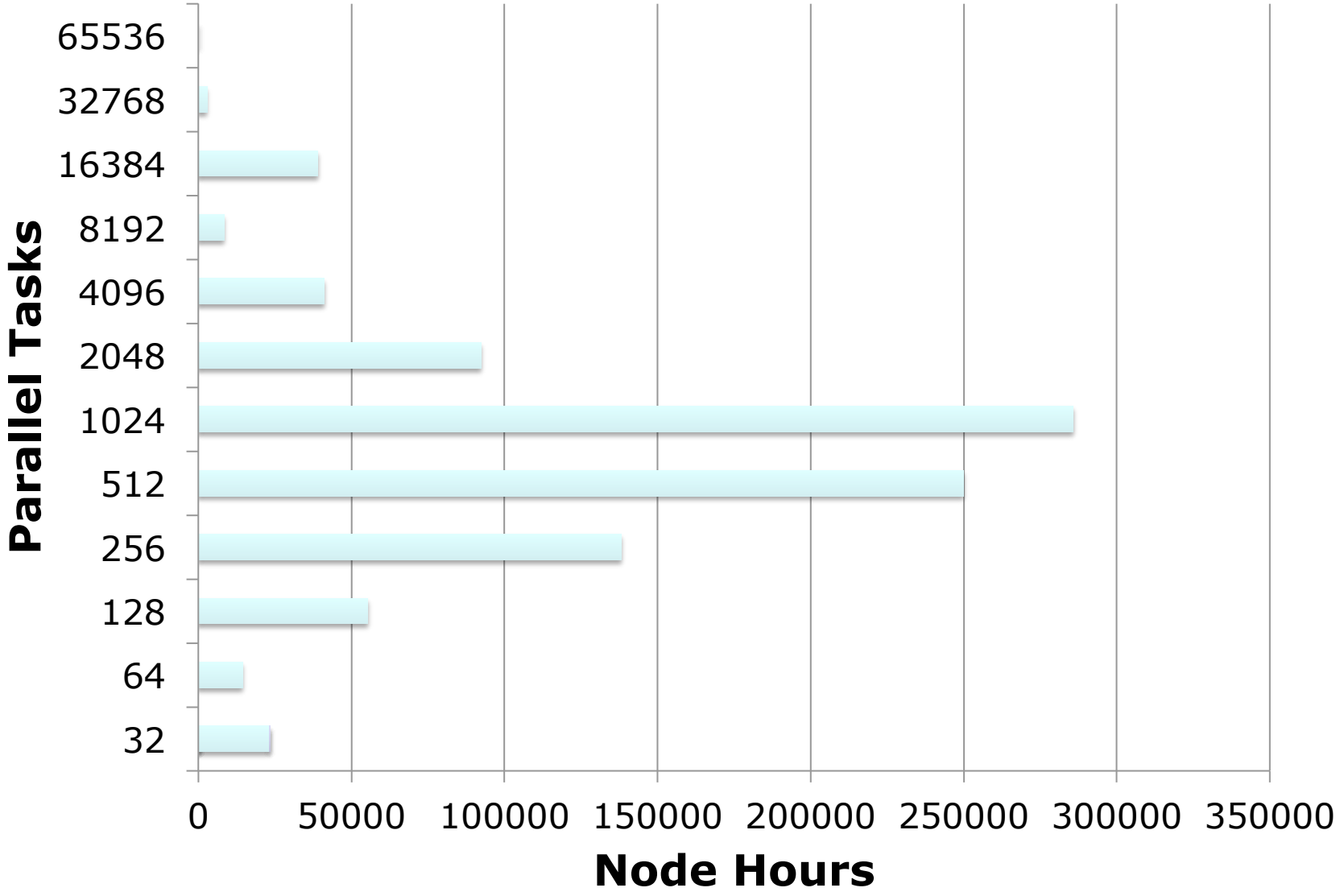
HECToR Usage Statistics

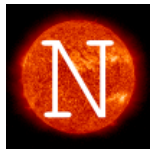




http://www.nu-fuse.com

HECToR Usage Statistics - CASTEP

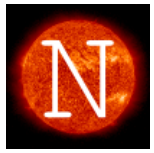




http://www.nu-fuse.com

Usage statistics: *ab initio codes*

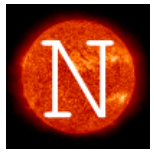
- On HECToR, *ab initio* codes most commonly use between 32 and 64 nodes (1024to 2048 cores); max cores used is 66536 (2048 nodes) using CP2K (between November2011 and April2013).
- Simulations generally computed by chemists/physicists are of up to 100's of atoms only.
- Thousand(s) of atom simulations are possible, but typically not used as this requires many nodes, resulting in excessive message passing synchronisation.
- CASTEP simulation of 64 atoms (Ti-Al-V): on 64 cores 13.5% of execution time is spent performing MPI_alltoallv for FFTs. Scaling 128 cores causes MPI to dominate, using 50% of the execution time, mostly due to poor synchronisation.
- For castep, typical usage of 1024 cores (32 nodes). Largest job using 32,768 cores (1024 nodes).
- Future technologies are likely to use accelerators – how will this affect DFT codes?



The issues with DFT calculations

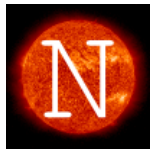
- Investigation of how to use alternative architectures is key to improving use of large-scale systems for larger simulations in DFT codes.
- Codes scale with $O(N^3)$ for N electrons.
- Typical benchmark results:
  - Benchmark 1: non-local wavefunction-based exchange correlation screening potential calculation of 8 Si atoms – total 32 electrons, on 8 CPUs.
  - Benchmark 2: 8 Ti, 64 Al and 2 V atoms. 596 electrons. No k-point parallelisation possible as only 1 k-point, on 64 CPUs.

	Benchmark 1	Benchmark 2
BLAS routines (matrix manipulation)	12.1%	19.8%
FFTs (using FFTW3 – includes comms)	38.7%	19.2%
Communications	26.6%	23.1%
(of which MPI_alltoall_V)	10.8%	13.5%



The issues with DFT calculations

- *First principles* calculations of condensed matter systems built up from calculations of the charged particle structure (positive nucleus and negatively charged electrons).
- Interactions between atoms and molecules (chemical and molecular bonding) results from the properties of the charged particles.
- Accurate calculation of these properties results in the ability to calculate complex physical phenomena of condensed matter systems.
  - Modelled with simple quantum mechanics – but large numerical calculations (Schrödinger's scales exponentially).
  - Electrons behave like point charge particles.
  - Complete system of N electrons and N_I nuclei is represented by the many body Schrödinger equation.
  - Born-Oppenheimer approximation assumes nuclei can be classed as classical particles.
  - Introduce fictitious auxiliary system, to reduce the number of electrons that are explicitly dealt with.
  - This fictitious system is a set of particles whose properties are identical to those of electrons, except that the electron-electron repulsive interaction is switched off.
  - Use ground state calculation only to reduce complexity of the calculation.



Principles of DFT

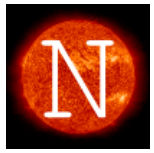
The principles of DFT

- Use Born-Oppenheimer Approximation – only model electronic structure
- Calculate a static potential, *V*, within which the electrons move.
- V then enables description of the *stationary* electronic state as a wavefunction:
- The wavefunction $\Psi(\vec{r}_1, \ldots, \vec{r}_N)$ is set up to satisfy Schrödinger's equation

$$\hat{H}\Psi = \left[\hat{T} + \hat{V} + \hat{U}\right]\Psi = \left[\sum_i^N (-\frac{\hbar^2}{2m_i}\nabla_i^2) + \sum_i^N V(\vec{r}_i) + \sum_{i<j}^N U(\vec{r}_i, \vec{r}_j)\right]\Psi = E\Psi$$

- This scales exponentially. To reduce complexity, this needs to be simplified so it can be calculated computationally.
- Define local effective (Kohn-Sham) external potential, where non-interacting particles move.
- The Kohn–Sham equation is defined by a local effective (fictitious) external potential in which the non-interacting particles move.
- Wavefunctions independent, *BUT* still need to satisfy exchange anti-symmetry. Result in single-particle wavefunctions that satisfy Schrodinger, called Kohn-Sham orbitals:

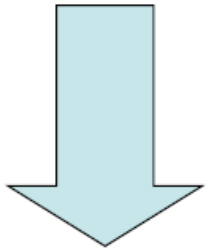
$$\left(-\frac{\hbar^2}{2m}\nabla^2 + v_{\text{eff}}(\mathbf{r})\right)\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r})$$



The issues with DFT calculations

Many Body Schrodinger Equation (exponential scaling)

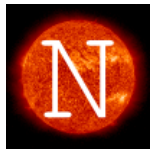
$$\{-\sum_i \frac{1}{2}\nabla_i^2 + \sum_{i,j} \frac{1}{|r_i - r_j|} + \sum_{i,I} \frac{Z}{|r_i - R_I|}\}\Psi(r_1,..r_N) = E\Psi(r_1,..r_N)$$



Kohn Sham Equation (65): The many body ground state problem can be mapped onto a single particle problem with the same electron density and a different effective potential (cubic scaling).

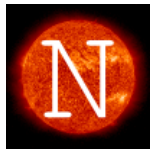
$$\{-\frac{1}{2}\nabla^2 + \int \frac{\rho(r')}{|r - r'|}dr' + \sum_I \frac{Z}{|r - R_I|} + V_{XC}\}\psi_i(r) = E_i\psi_i(r)$$

$$\rho(r) = \sum_i |\psi_i(r)|^2 = |\Psi(r_1,..r_N)|^2$$



The issues with DFT calculations

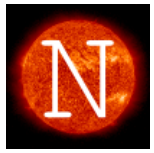
- DFT is nominally O($N^2 ln\ N$) or O(N^3) depending on system size
- Pseudo-Potential Plane-wave
  - Good coverage, no region bias
  - Coverage where not needed (cubic scaling)
  - Relies heavily on 3D FFTs (requires orthoganilsation, diagnolisation)
- Operations have fine-grain data parallelisation:
  - FFT, BLAS, Scatter-gather
- Includes non standard DFT calculations such as non local exchange, making the calculations a lot more expensive both in memory and CPU time.
- May seem like an obvious candidate because of fine-grain parallelisation, except for sparse data sets and communication overheads due to large data structures.
- Currently ported DFT software:
  - Quantum Espresso http://www.quantum-espresso.org
    - integrated suite of open source packages. Currently GPU acceleration available for the plain wave self consistent field (PWscf) code
  - Nwchem – Quantum Chemistry scalable code. Uses atom centred Gaussian type orbitals & Car Parrinello
    - http://www.nwchem-sw.org/index.php/Benchmarks#Current_developments_for_high_accuracy:_GPGPU_and_alternative_task_schedulers
  - VASP (Vienna *ab initio* simulation package) https://www.vasp.at/
    - Part of Hartree-Foch calculation has been ported, but no results released and not in main package release.



CASTEP: initial accelerator investigation

- Replace blas calls with cula
  - (cuda-blas library http://www.culatools.com/)
- Replace fft calls with cufft
  - Small simulation, to fit on one CPU, no MPI calls. 4 Ti atoms, 2 O atoms, total of 32 electrons.
    - No device calls runtime = 14.6s
    - Cula blas calls runtime = 31.1s
    - Cula blas and cufft calls runtime = 418s.

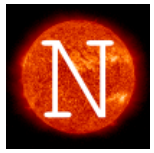
Majority of the increased runtime was due to data transfer.



GPU*ification* of CASTEP

http://www.nu-fuse.com

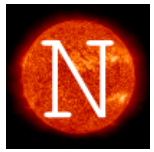
- Aim:
  - remove data transfer problems by placing most of the large data structures on the GPU.
  - Use OpenACC kernels, cula blas and cufft.
- Code structure:
  - 531,000 lines of modular Fortran 90/95 with MPI
  - 134 modules, over 1974 subroutines, 237 functions.
  - Parallelised over k-points, bands and plane waves
  - Able to calculate geometry optimisation, finite temperature molecular dynamics, and a wide range of derived electronic structure properties.
  - Runs on supercomputers and PCs.
  - Compiles in serial or with MPI.
  - Majority of computation in self-consistent field computation loop, using ~50 subroutines.



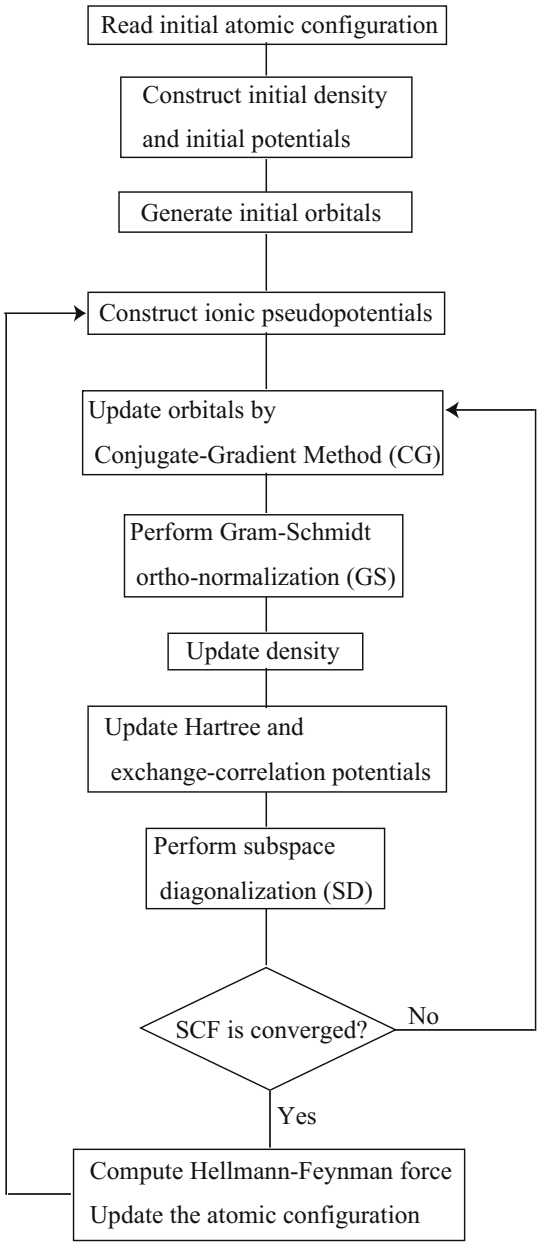
GPU*ification* of CASTEP

Data structures on device

- Wavefunctions:
  - `complex(kind=dp) :: Wavefunction%coeffs(:,:,:,:)`
  - `complex(kind=dp) :: Wavefunction%beta_phi(:,:,:,:)`
  - `real(kind=dp) :: Wavefunction%beta_phi_at_gamma(:,:,:,:)`
  - `logical :: Wavefunction%have_beta_phi(:,:)`
  - `complex(kind=dp) :: Wavefunctionslice%coeffs(:,:)`
  - `complex(kind=dp) :: Wavefunctionslice%realspace_coeffs(:,:)`
  - `real(kind=dp) :: Wavefunctionslice%realspace_coeffs_at_gamma (:,:)`
  - `logical :: Wavefunctionslice%have_realspace(:)`
  - `complex(kind=dp) :: Wavefunctionslice%beta_phi(:,:)`
  - `real(kind=dp) :: Wavefunctionslice%beta_phi_at_gamma(:,:)`
- Bands
  - `complex(kind=dp) :: coeffs(:)`
  - `complex(kind=dp) :: beta_phi(:)`
  - `real(kind=dp)  :: beta_phi_at_gamma(:)`

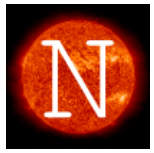


http://www.nu-fuse.com



Majority of calculation is within a 'Self Consistent Field' Loop

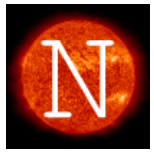
- Place central loop on device
- Aim to only remove atomic configuration and key electronic properties from the device
- Place OpenACC kernals around key computational stages



GPU*ification* of CASTEP

http://www.nu-fuse.com

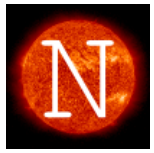
```
subroutine wave_copy_wv_wv_ks
……
!$acc kernels present_or_copy(wvfn_dst, wvfn_src)
!Map  reduced representation of coefficients on k-point
        do nb=1,nbands_to_copy
          recip_grid = cmplx_0
          call basis_recip_reduced_to_grid(wvfn_src%coeffs
(:,nb,nk_s,ns_s),nk_src,recip_grid,'STND')
          call basis_recip_grid_to_reduced(recip_grid,'STND',wvfn_dst%coeffs
(:,nb,nk_d,ns_d),nk_dst)
        end do
……
    ! copy rotation data
……
      do nb=1,nbands_to_copy
        do nb2=1,nbands_to_copy
          wvfn_dst%rotation(nb,wvfn_dst%node_band_index
(nb2,id_in_bnd_group),nk_dst,ns_dst) = &
                                      & wvfn_src%rotation(nb,wvfn_src
%node_band_index(nb2,id_in_bnd_group),nk_src,ns_src)
        end do
      end do
……
!$acc end kernels

end subroutine wave_copy_wv_wv_ks
```



GPU*ification* of CASTEP

The problems with porting

- CASTEP uses functions that are not supported on devices, such as the use of '`optional`' types when passing data to subroutines followed by '`if present`' statements.
  - Resolved by creating copies of subroutines with and without optional arguments.
- Specifying arrays with `dimension(*)` when passing to subroutines
  - Resolved by specifying correct dimension structuer, sometimes requiring mulitiple copies of subroutines
- Sometimes the limitations of what is on and off the device results in multiple`!$acc kernel` regions very close together, and not the entire subroutines, which is not necessarily very efficient. Will require a lot of fine tuning to improve performance.
- Serial only so far.



GPU*ification* of CASTEP

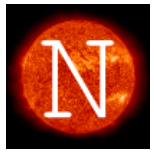
http://www.nu-fuse.com

Still an ongoing project

- Very closed to having the first version of the software ported to device
- Expect this to be optimised to improve performance and minimise data transfer
- Found a PGI compiler bug
  - Found bug in openACC compiler due to the inclusion of too many nested case/select statements in a kernel.
  - Waiting for a resolution.

Next step: OpenACC+MPI implementation.

cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY

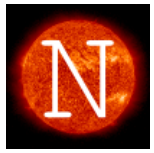


Summary

http://www.nu-fuse.com

Porting CASTEP to GPGPUs using OpenACC and CUDA libraries

- DFT software is one of the largest users of UK supercomputing facilities so we need to consider if these will work on accelerators.
- Close to having a version finished which will require iterative improvement to investigate performance of DFT software on accelerators.



http://www.nu-fuse.com

Acknowledgements

Nu-FuSE is funded by the G8 Exascale Software Projects funding

The University of Edinburgh and EPSRC has provided additional funding towards porting CASTEP to GPGPUs.

Collaborators:

Adrian Jackson (University of Edinburgh)

Graeme Ackland (University of Edinburgh)

Derek Hepburn (University of Edinburgh)

Stewart Clark (University of Durham)

Keith Refson (STFC/RAL)

http://www.nu-fuse.com

cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY

http://www.nu-fuse.com

cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY