Computational Science and Engineering (CSE)

Dr Ian Reid

Chief Commercial Officer

Ian.Reid@nag.co.uk

nag®

Experts in numerical algorithms and HPC services

Agenda

- Introduction to NAG
- HECToR CSE Support
  - Training
  - CSE – core and distributed
  - Examples
- Summary

NAG Background

- Founded 1970
  - Co-operative software project
  - Not-for-profit organisation
  - Surpluses fund on-going research
- ~£8m financial turnover
- ~100 employees
  - ~65% developers/technical consultants
  - Oxford (HQ), Manchester, UK; Chicago, USA; Tokyo, Japan; Taipei, Taiwan

NAG 1970–2010
40TH ANNIVERSARY
Celebrating 40 years of numerical excellence

Products & Services

- Numerical and Statistical Libraries
  - Over 1600 user-callable components
  - Callable from many environments (e.g. MATLAB, Excel)
- Consulting Services
  - Code development, tuning, tailoring
  - Testing and validation
- HPC Services
  - Procurement advice, market watch, benchmarking
  - Computational Science and Engineering (CSE) support
- Experts in Numerical Engineering

HECToR

HIGH END COMPUTING TERASCALE RESOURCE

A Research Councils UK High End Computing Service

HECToR

- UK Research National Service
  - £113m over 6 years 2007-13
  - Cray XT/XE series - ~0.8PF late 2011
- NAG advised on hardware procurement and undertook benchmarking process
- NAG provides CSE effort
  - 120 person years
  - Central and Distributed support model
  - Performance tuning, scaling user codes etc

HECToR: UK national HPC service

HECToR

Academic user community

Helpdesk

(N A G)

Computational Science & Engineering Support & Training: 120 person years

System management & hosting (UoE HPCx)

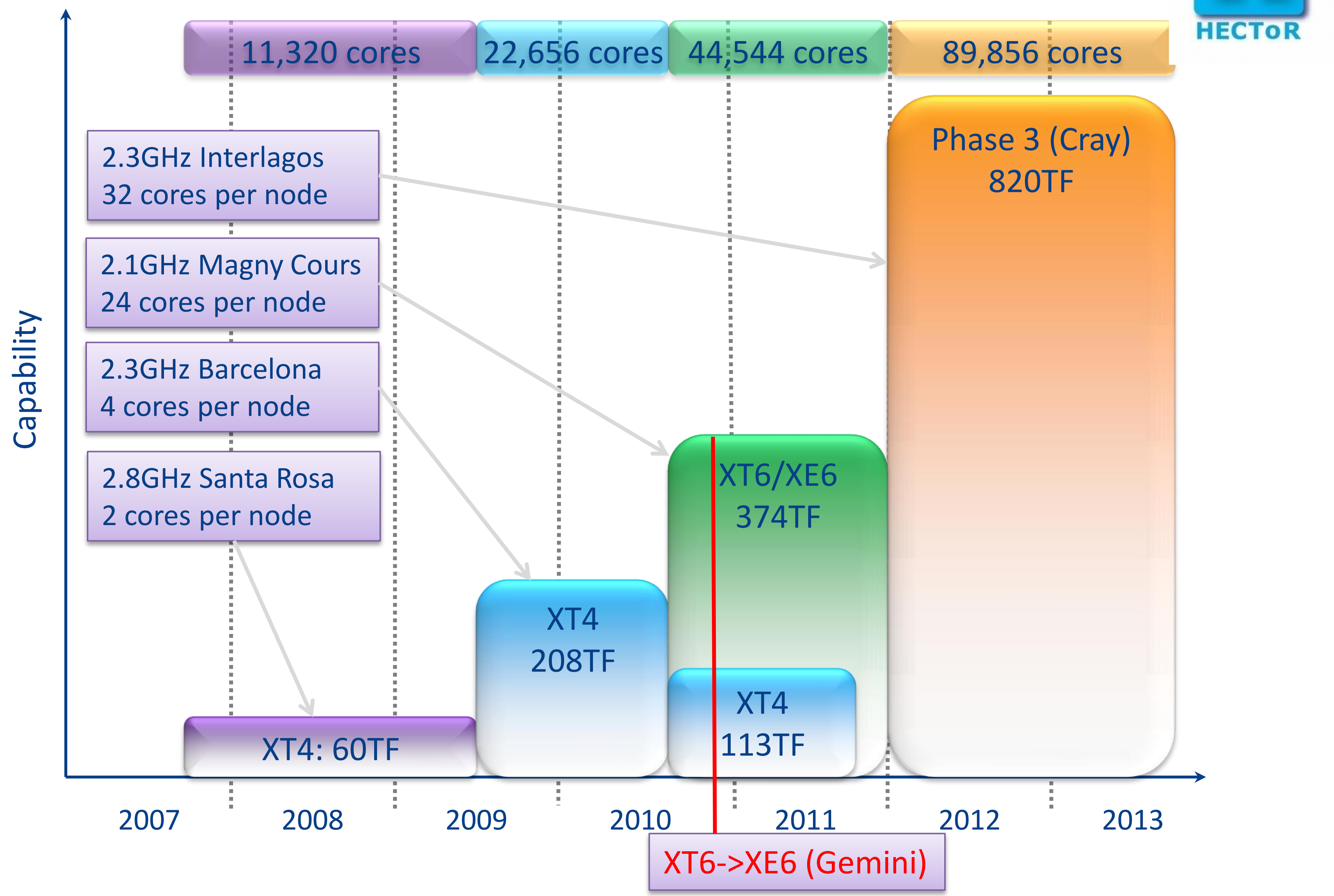
Cray Centre of Excellence

Phase 2b: 374TF Cray XE6

Moving in late 2011 to Phase 3: ~820TF Cray

2007-2013, ~£113M

Updated HECToR Technology Roadmap



HECToR CSE Service: Overview

HECTOR

Core Support

- Helpdesk - applications queries
- Assistance with porting, tuning, migration (2a→2b →3), …

Training

- HECToR, Parallel programming, Optimisation, Software engineering, …
- Held locally whenever sensible

Distributed Support

- Dedicated 6-24 month projects to scale, renovate, restructure applications
- Projects selected by independent panel with non-UK chairman
- 35 projects completed, 19 in progress, 42 person-years delivered (Aug‘11)
- ~75% embedded in the user community

HECToR Training

- Aimed at HECToR users but open to all
- Delivered by experienced practitioners
- Over 1000 attendees to date
- At least 72 course days per year
- Mainly delivered at user sites
- Also organise summer schools, teach an HPC module within Southampton Doctoral Training Centre, ...
- Collaborating with new EPSRC Short Course Centre for HPC

Typical Course Structure

- Held locally whenever sensible
- 1-5 days
  - complementary courses run back-to-back
- Typically 12-30 places
  - depends on course and venue
- Large practical content (hands-on)
  - aim for roughly one trainer for 10 students
- Printed course material provided

Training Courses

- HECToR Specific
  - Introduction to HECToR
  - Programming the X2 (Vector)
  - Transitioning to the Cray XE6
- Programming
  - Fortran 95
  - Parallel Programming w/MPI
  - OpenMP
  - Parallel I/O
  - Co-Array Fortran
  - CUDA Programming
  - OpenCL Programming
- Application Specific
  - DL_POLY
  - Exploiting Parallel CASTEP on large-scale HPC
- Other
  - Debugging, Profiling and Optimisation
  - Multicore
  - Core Algorithms for High Performance Scientific Computing
  - Best Practice in HPC Software Development
  - Scientific Visualisation

Attendee Feedback (~600 responses)

- 1=poor 2=fair 3=good 4=very good 5=excellent
  - Presentation: 4.08
  - Materials: 4.13
  - Exercises: 4.01
  - Organisation: 4.15
  - Overall rating: 4.14
- 3=useful 4=very useful 5=exactly what I need
  - Usefulness: 3.84
- 1=too slow 3=about right 5=too fast
  - Pace: 3.24

Other Training Stats

- 1047 attendees from 46 institutions
- 77 courses run so far ...
- ... delivered at 14 Universities plus NAG offices
- Over half attendees (557) attended “basic programming” courses (Fortran, MPI, OpenMP)

HECToR dCSE – early results

HECToR

Quantum Monte-Carlo **CASINO**	Materials Science **CASTEP**	Ocean Modelling **NEMO**
Up to 4x faster Scaling to 40,000 cores	4x scalability 4x faster	25% faster
12 person months	8 person months	6 person months
Initial savings of nearly £1M for one project	£2.4M saved for remainder of project	Potential £3M savings across all users

Codes Benefiting from dCSE

- ACPIM
- CABARET
- CASINO
- CASTEP
- ChemShell
- CITCOM
- Code_Saturne
- CP2K
- CRYSTAL
- DL_POLY_3
- EBL
- GAMESS-UK
- GLOMAP
- HELIUM
- ICOM
- Incompact3d
- LEM
- MRBV
- NEMO
- OpenFoam
- PRMAT
- SS3F
- SWT
- WRF

Knowledge Transfer is Key

- FFT scalability: e.g. Incompact3d, DSTAR, CABARET, (SS3F, SWT)
- Improved I/O: e.g. DL_POLY, Nemo
- OpenMP multi-core enhancements: e.g. Helium, GloMAP
- Shared memory: e.g. CASINO, VASP, CASTEP
- Improved data decompositions (load balancing): e.g. Fluidity-ICOM, CABARET, CARP
- New data parallelism: e.g. VASP

FFTs – 2d Decomposition

- Many important codes use FFTs
- Ning Li (NAG) recognised that 3d FFTs (e.g. from CFD) were only being decomposed in 1d (v. poor scaling)
- Ning developed a method to perform a 2d data decomposition whilst hiding the communications and data distribution
  - General-purpose 2D domain decomposition library has been implemented
  - Knowledge transfer to several other codes
- More details: www.2decomp.org

FFTs-2d Example

- Incompact3d: Direct Numerical Simulations (DNS) of turbulent fluid flows
  - Implicit finite difference solver for flow through fractal generated geometries – specialised geometries
  - Originally used 1D slab decomposition with good scaling, but limited to ~1,000 cores
  - With 2d-decomp almost perfect scaling up to 16,384 cores for a fixed problem size
  - And good scaling for a fixed workload per process (next slide)

New FFTs with Incompact3D

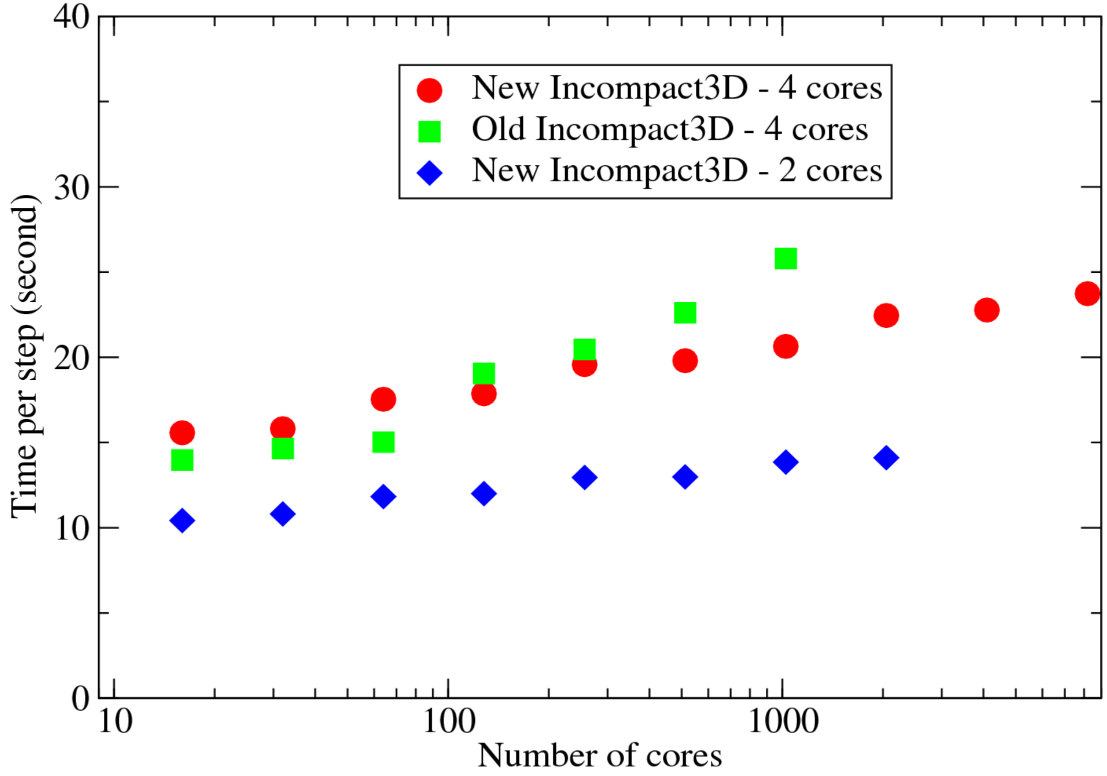


Figure: Weak scaling with constant work load per core (varying problem size)

Input/Output Example

- DL_POLY_4: molecular dynamics simulation
  - Generic parallelisation (for short-ranged interactions) based on spatial domain decomposition and linked cells
- Problem
  - 36 MD steps/s without I/O; **4** MD steps/s with I/O
    - (14.6 million particle system on 16,384 processors)
    - ~0.5s per MD timestep; ~1800s for I/O per ~100-1000 timesteps
- Solution
  - Gather data onto subset I/O processors
    - Do in batches so as to avoid memory overhead
    - Sort in parallel across the I/O processors
  - Result: **32** MD steps/s with I/O

Shared-memory (multi-core) Example

- CASINO: Quantum Monte Carlo (QMC) code for computing the properties of physical models with a large number of atoms, e.g. crystals or nanoclusters
  - Over 2GB of orbital data is stored for access by each core
  - But multi-core => less memory per core => use less cores per node (expensive/wasteful)
  - Solution – use shared memory on node
  - CASINO now runs on largest HECToR queues and also on Jaguar in the US.

New Data Parallelism Example

- VASP: ab initio quantum mechanical total energy calculations and molecular dynamics (MD) simulations with pseudopotentials and a plane wave basis set.
  - One of the most widely used applications on HECToR
  - Scalability limited to ~256 cores
  - Implemented new parallelisation over k-points
  - 2x speed-up on same core count on XE6
  - 5x speed-up on 1600+ cores for unit cell of Litharge (α-PbO) with a total of 4 atoms using 108 k-points
  - (Not applicable to all systems)

Parallelising a Serial Code Example

- CABARET: Compact Accurate Boundary Adjusting high REsolution Technique to solve compressible Navier-Stokes equations
  - A parallel version of CABARET has been implemented to pass data between internal boundaries (cell faces & sides)
  - A 3D backward-facing step test case of 51.2M cells, retains 80% parallel efficiency on 1000 cores
  - Simulations can now be performed on grids at least 512 times larger than previously possible.
  - Smaller scale simulations can now be performed in a **few minutes** that would otherwise have taken **several days**

Parallel Decompositions Example

- CARP – Large scale simulations of Cardiac Bioelectric Activity
  - Investigating treatments for disease and causes – e.g. defibrillation
  - MRI data converted to an unstructured finite element mesh with 10M unknowns (grid points)
  - Solve coupled pdes (elliptic and parabolic) and odes by using the PETSc parallel library
  - Poor performance above 64 cores due to
    - Load imbalance
    - I/O
    - Pre-conditioning for the elliptic pde

Unstructured Parallel Decompositions

- Improved parallel decomposition, I/O and explicit solver for pde
- Simulations now scale to 16,000 cores

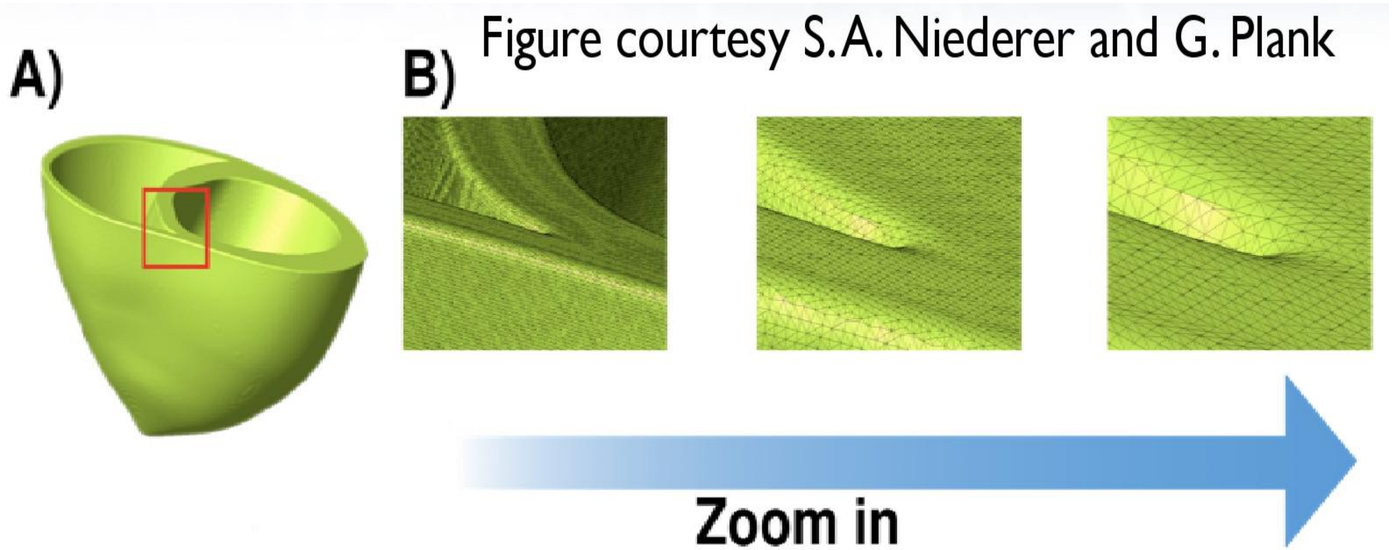


Figure: Model of a human heart, 26M unknowns

Unstructured Parallel Decompositions

- Simulations now lag real time by factor of 280 (16000 cores), previously 4300 (1000 cores)
- 1 second activity takes 5 minutes, previously 74 minutes

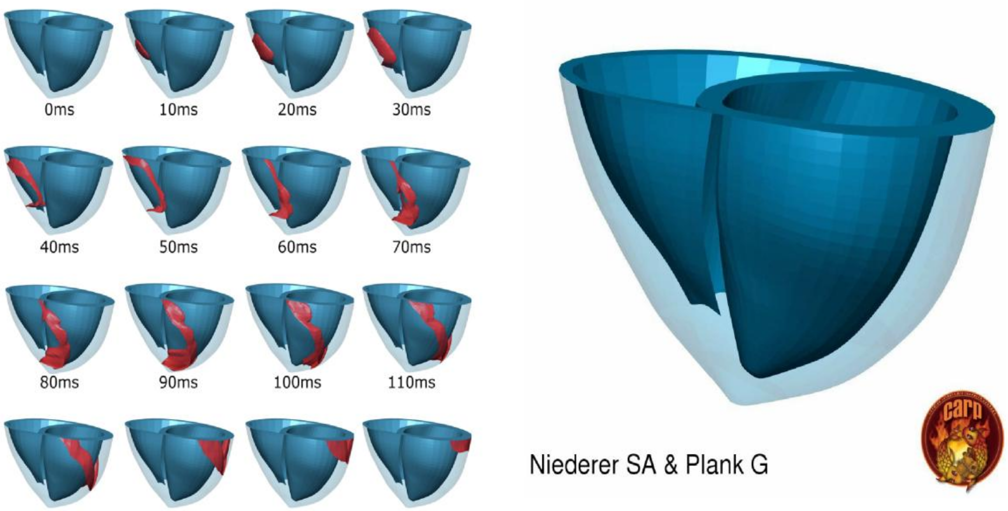


Figure: Model of a human heart compared to real time

Some CSE successes so far

HECToR

- **Speed and Scalability** of Key Materials Science Code (CASTEP) Quadrupled
- Speed and **I/O Performance** of Oceanography Code (NEMO) Enhanced
- Performance of Quantum Monte-Carlo Application (CASINO) **Quadrupled**
- **Speed and Scalability** of Materials Science Simulations (CP2K) Enhanced
- **Performance** of Atmospheric Chemistry Simulations (GLOMAP/TOMCAT) Enhanced
- **Performance** of Geodynamic Thermal Convection Simulations (CITCOM) Enhanced
- Scalability of Fluid Turbulence Simulations (EBL) **Enhanced up to 40x**
- Simulations of Catalytic Chemistry with ChemShell **8x Faster**
- Scalability of Ocean Modelling Application (Fluidity-ICOM) **Dramatically Improved**
- Performance of Molecular Dynamics Application (DL_POLY_3) **20x Faster**
- **Capabilities** of Key Materials Science Application (CASTEP) Significantly Enhanced
- Performance of Heart Modelling Application (CARP) **20x Faster**
- Performance of Turbulent Fluid Flow Simulations (Incompact3D) **Improved by factor of 6x**
- **Performance and Capabilities** of Materials Science Code (Conquest) Enhanced
- Aircraft Noise Simulations (CABARET) **Get Faster and Reach Bigger Models**

http://www.hector.ac.uk/cse/reports/

CSE Service Experience

HECToR

- Strong CSE support allows researchers/engineers to concentrate on the research/design
  - HPC skills development/transfer
- Developing local expertise is important for on-going sustainability
- Multicore is a real challenge – and then manycore …
  - Even established HPC codes often need work
  - Spectacular gains are possible (i.e. costly not to take them)
- Core + Distributed CSE model works very well

Further Information

- Full details about dCSE support
  - hector.ac.uk/cse/distributedcse/
- Current and completed projects
  - hector.ac.uk/cse/distributedcse/projects/
- Detailed technical reports from completed projects
  - hector.ac.uk/cse/reports/

Summary

- The hardware escalator has ended
- CSE is critical to achieving good code performance
- ... and good science

Thank You – Any Questions?

Ian.Reid@nag.co.uk