The University Logo

Different versions of the University Logo

Full colour print logo

The logo must not be reproduced less than 40 mm wide.

40mm

For applications such as narrow book spines or pens, the words 'University of Leicester' should be used in place of the logo.

Logo colours:
Pantone 200 C and Pantone 431 C, these will be converted to process for full colour material.

Inverse print logo

Mono print logo (printed in black only)

The logo must always be reproduced in its entirety and must not be distorted.

There are three variations of the logo: the full colour logo; the solid, mono logo; and also a white inverse version.

The full colour logo is only to be used on a white background. The solid, mono logo is to be used on a pale background colour and the white inverse version is to be used on a solid colour.
University of Leicester and DiRAC Catalyst

- DiRAC provides HPC resources for the UK theory communities in astrophysics, particle physics, cosmology and nuclear physics

- Leicester hosts DiRAC Data Intensive service: Catalyst@Leicester plans focus on data intensive workflows

- Main goals:
  - Port set of key DiRAC community codes to Thunder X2 and provide guidance to other users/communities
  - Provide feedback on performance of Catalyst systems as a national-scale HPC resource, including integration with other DiRAC resources (high-performance file system, etc)
  - Technical: explore use of BeeGFS and Moab/Torque scheduler
Configuration

- Catalyst Arm cluster currently set up as standalone system
- Access open to selected DiRAC users and Leicester research groups
- InfiniBand spine ready to integrate with DiRAC Data Intensive service in due course
  - Will access main DiRAC lustre storage.
- Storage using BeeGGFS
- Currently using PBS but working on Moab/Torque
Leicester/DiRAC Catalyst target codes and research areas

- **DiRAC codes** - astrophysics, particle physics, cosmology:
  - **sphNG** - star formation simulations
  - **SWIFT** - cosmological N-body+Hydrodynamics simulations of galaxy formation and evolution
  - **TROVE** - molecular line-list calculations for molecules such as methane for interpretation of exoplanet spectra
  - **GRID** - lattice-Quantum ChromoDynamics (QCD) calculations of properties of fundamental particles from first principles

- **Codes from across Leicester research community:**
  - **Earth Observation Science** - processing and modelling of satellite data, including forest cover mapping
  - **Computational Chemistry** - image processing for Cryo Electron-Microscopy with EMAN2 and Relion codes; extraction of information from biological NMR data using Relax code
  - **Engineering** - Detached Eddy Simulations of bandwidth-limited, compressible, turbulent flows to model aircraft noise with Cosmic code; post-processing of CFD simulations with Antares code
Hackathons

- A key goal of Catalyst is to assess viability of operating the Arm cluster as part of a national HPC service
- Two hackathons held so far - 10 codes worked on in depth:
  - Smoothed Particle Hydrodynamics (SWIFT, SphNG)
  - Unstructured mesh code for hydrodynamics (Arepo)
  - Lattice-based particle-physics calculations (Grid, Latfield2d)
  - Grid-based relativistic dynamics code (GRChombo)
  - Monte-Carlo particle transport code (OpenMC)
  - Molecular rovibrational energy calculator (TROVE)
  - Finite difference multigrid Poisson solver (DL_MG)
  - Finite-volume Euler equation solver for low mach-number astrophysical flows (Seven Leagues Hydrodynamics code)
- Majority of codes ported easily to Arm architecture and showed good performance
User Experiences

• Basic porting quite straightforward in general
• Sometimes need to experiment to find best compiler options as impact of flags is not always that expected from x86 experience
• For Fortran with flang compiler, *pointer vs allocatable* makes significant performance difference on Arm
• Some parts of code may be x86 specific (and you might have forgotten that!!)
• NEON porting is not difficult and is worth the effort in terms of performance
  • ~20% performance boost for GRChombo
• Arm Forge tools very helpful for profiling and optimising
• Performance relative to x86 often varies during a workflow as characteristics of calculation change
• As with other platform transitions, vendor support is essential and was particularly useful during the hackathons
Cosmological Simulations on Arm
The SWIFT code
Borrow (Durham), Kegerreis (Durham), Schaller (Leiden)

- Adheres strictly to the C99 standard, so was an 'easy' port; only changes needed were to non-standards compliant sections of code (lesson learned!)
- OpenMPI 4.0.1 working
- SWIFT visualisation toolchain uses python routines
  - relies on Numba JIT compiler - took some additional work to install
- BeeGFS availability means system is ready for full-scale production runs
• SWIFT: New open-source cosmological simulation code
  - includes hydrodynamics and gravity (N-body)
  - also used for planetary collision simulations
  - hybrid MPI + threads code using task-based parallelism.

• Nov: First cosmological simulation on Catalyst cluster with one billion particles including full galaxy formation physics
• Star formation rate-density is a primary output of cosmological simulations

• Results from SWIFT calculation on Arm consistent with calculation on x86 starting from same initial conditions
All runs used ARM clang, and two ranks per node (i.e. one per socket) with OpenMPI.

Sweet spot appears to be with 2 SMT threads per processor.
• SWIFT uses pthreads with customised internal thread handling
  - Customised pinning worked on Arm without modification
• Inclusion of libamath gives significant performance improvement with Arm Clang compiler
Star formation on Arm

sphNG

Matthew Bate (Exeter)

- Smooth Particle Hydrodynamics (SPH) code
  - Typically used for star and planet formation, accretion discs

- Multi-physics:
  - Gravity, hydrodynamics, sink particles
  - (non-)ideal MHD
  - Radiative transfer (flux-limited diffusion)
  - Dust/gas fluid mixtures (1-fluid and 2-fluid)
  - Dark matter

- Hybrid MPI/OpenMP code
  - Fortran 77+
  - No other libraries or dependencies
sphNG on ARM

Smaller values are better

- Broadwell 28-core 2.4GHz (ifort v16.0.3, Isca/Exeter)
- Skylake 36-core 2.3GHz (ifort v18.0.3, DIAL)
- ARM-Flang (v19.0, Catalyst)
- ARM-Flang (v19.0, Catalyst) -armpl

```
                      Time to solution
                      30
                      22.5
                      15
                      7.5
                      0
```

- Hydro (1N, 8MPI)
  0.5 million particles, Gravity+hydro only
- Orion (2N, 8MPI)
  5 million particles, Radiative transfer + ISM
- Clust-0.4pc (8N, 32MPI)
  35 million particles, Radiative transfer
• Lattice Quantum ChromoDynamics (QCD) code
  • applies 'Wilson flow' to lattice configurations
• Calculate various quantities as a function of this flow: topological charge, action density with Wilson action, action density with an improved plaquette action.
• Science goal is to use the Wilson flow in order to calculate the viscosity of the QCD medium vs temperature.
OpenQCD FASTSUM

Speedup vs Number of Cores

- Alice (Haswell) - Intel Compiler
- ARM - gcc Compiler
- Sunbird (Skylake) - Intel Compiler

Alan Kirby
Application porting: OpenQCD FASTSUM

Time Taken vs Number of Cores

- Alice (Haswell) - Intel Compiler
- ARM - gcc Compiler
- Sunbird (Skylake) - Intel Compiler

Alan Kirby
Summary

• Experience for technical team of initially standing up the cluster was no more difficult than standard installations

• User experience of code porting has been very positive in terms of ease of porting and performance obtained

• Further system work and code optimisation on-going

• For more information, or to request access contact:

  Mark Wilkinson, DiRAC Director
  mark.wilkinson@leicester.ac.uk