Architecture Comparison Exercise (ACE)

Loïs Steenman-Clark

Head of Computational Modelling Services (CMS)
National Centre for Atmospheric Science (NCAS)
Department of Meteorology
University of Reading

ACE

• is an architectural comparison exercise for the broad UK modelling community using the national academic HPC platforms

• is a study of a few application on a few diverse HPC platforms

OUTLINE

- *put ACE in context*
- *outline some key findings from ACE*
- *look at the limitations of ACE for meteorological code*

1. Where does ACE fit in the application benchmarking scenario?

Complexity/effort

→ *Application Modelling*

AIM: to extrapolate application performance for new application code developments or problems sizes or new HPC hardware or software

→ *RAPS*

AIM: to explore meteorological application performance in collaboration with HPC vendors

→***ACE***

AIM: to provide a snap shot of the performance of a group of applications on current diverse HPC architectures

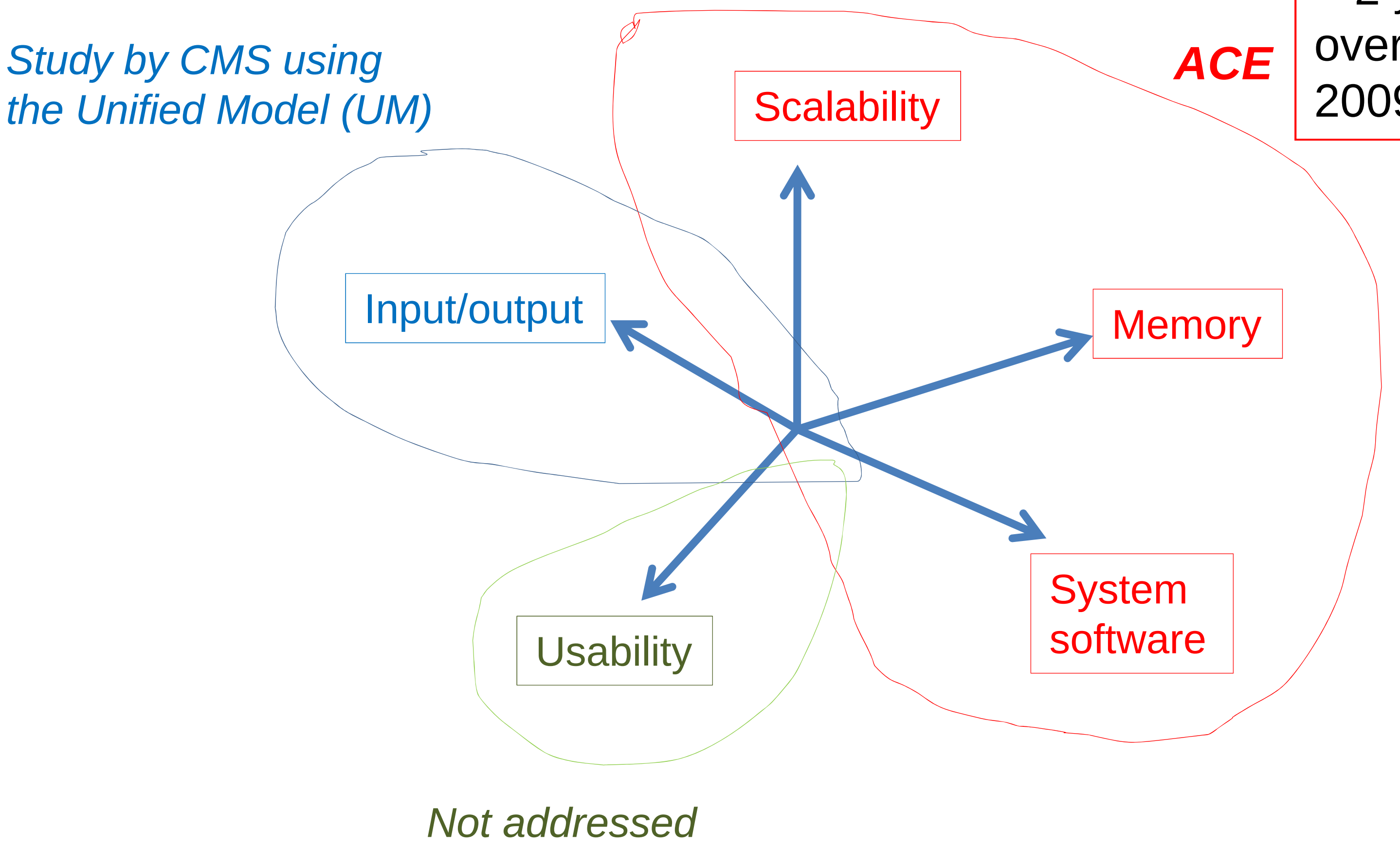
→ *benchmarking*

AIM: to predict performance of specific applications on new HPC platforms

2. What performance issues did ACE explore?

Issues that were meaningful to both applications and HPC

ACE was a small project with ~ 2 years FTE effort over ~9 months in 2009/2010



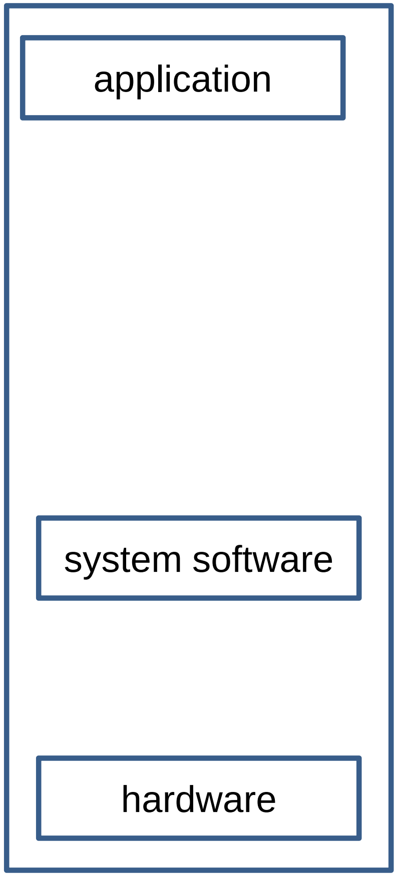
3. Applications and HPC systems used in ACE

Application	Description
SENGA	CFD code
CASTEP	Ab-initio Car-Parrinello code
GADGET	Cosmological simulation
AMBER	Bio-systems molecular dynamics package
HELIUM	Atomic calculations

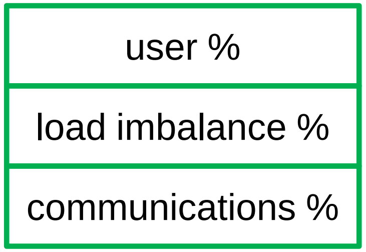
HPC System	Description
HPCx	IBM Power5 (closed January 2010)
HECToR/Jaguar	Cray XT4/XT5
Darwin/Merlin	Large University Clusters (Cambridge/Cardiff)
Jugene	IBM BlueGeneP
JuRoPa	Nehalem quad core Cluster

4. What are the ACE key achievements?

- Methodology for modelling the whole process of running case studies with applications



- exploring generic performance analysis tools for gathering evidence
- building simple performance model of the application using three sample cases: small, medium and large



- explore performance with different compilers
- with different MPI libraries
- influence of different maths libraries (where used)

- use HPCC synthetic benchmarks to characterise systems in conjunction with IMB benchmarks
- correlate hardware characteristics with the simple application performance model

4. What are the ACE key achievements?

SUMMARY

- large variation of performance of collectives on the different HPC architectures explored

- TAU and CrayPat were found to be very useful tools for performance analysis and both gave similar results

- the software environment can have a large effect on the performance of an application. In ACE the MPI environment was seen to have a large effect on some applications.

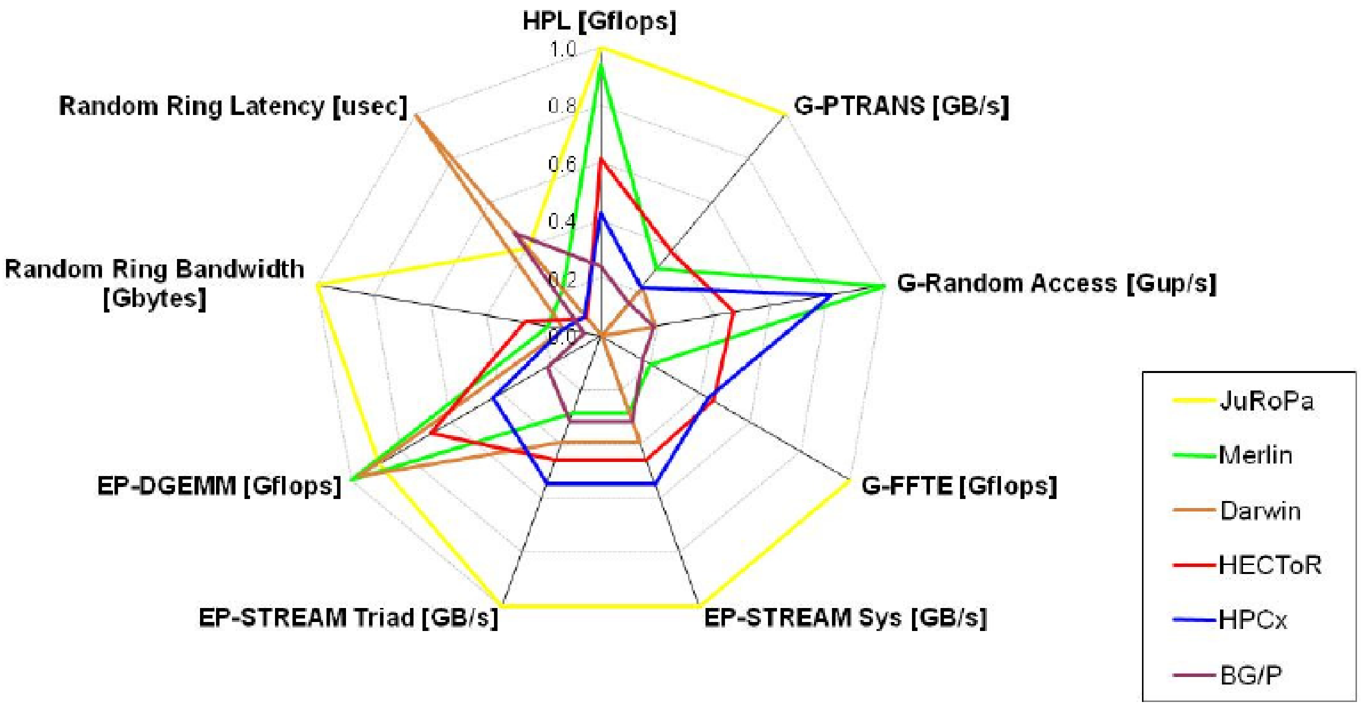
- the problem sizes of the cases were not always optimal and needed more consideration

- correlations can be observed linking application performance and the HPC attribute performance a determined by synthetic benchmarks

4. What are the ACE key achievements?

• Kiviat diagrams enabled a comparison of different HPC systems at a glance

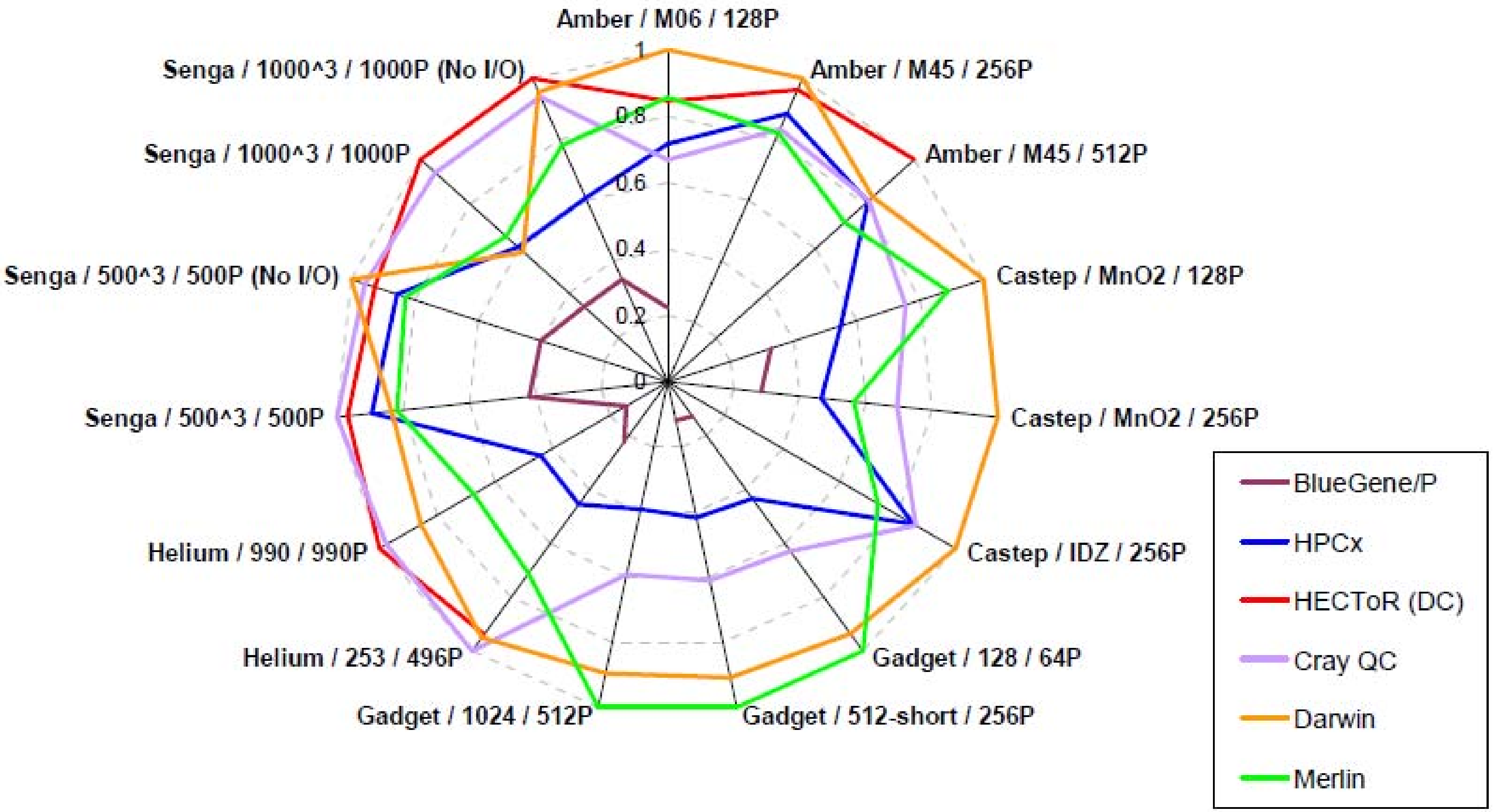
a) HPCC benchmark results for different HPC platforms



4. What are the ACE key achievements?

• Kiviat diagrams enabled a comparison of different applications at a glance

b) Application performance for different problem sizes on different HPC platforms



4. What are the ACE key achievements?

- Understanding of performance variability issues across a number of applications and HPC platforms, which provides evidence for investment in both software development and HPC provision

Dependency	Characterisation	Performance Variability
HPC architecture	• Scalability Memory Core speed Interconnect	*ACE* up to 200%
System software	• Compiler • MPI implementation • maths libraries	*ACE* 10-20% *ACE* ~20% *ACE* ~9%
Usability	• MTBF • jitter • slowdown	*ACE* ~10% on some HPC platforms
I/O	• Application • I/O hardware architecture • jitter	

5. What performance issues were NOT addressed by ACE?

Two critical issues for data intensive high resolution applications such as weather and climate models are

- Input/Output
- Throughput / Use-ability

Why were these not explored?

- Applications and I/O hardware were considered too hard to characterise within this current project (non-quiesent systems, limited disk space, challenge of changing application I/O strategies). However the methodology of Shan, Antypas and Shalf (NERSC) could be adapted to enable the Unified Model (UM) as well as SENGA with I/O to be included in a future ACE type study.

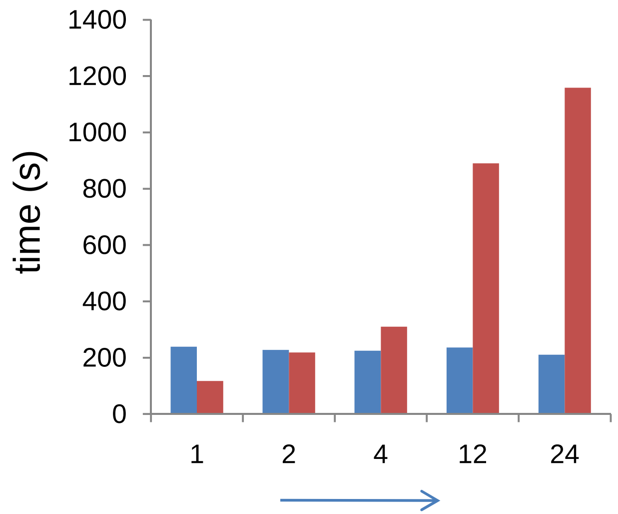
- Throughput or use-ability are again hard to characterise and they are a function of HPC service delivery and administration

5. What performance issues were NOT addressed by ACE?

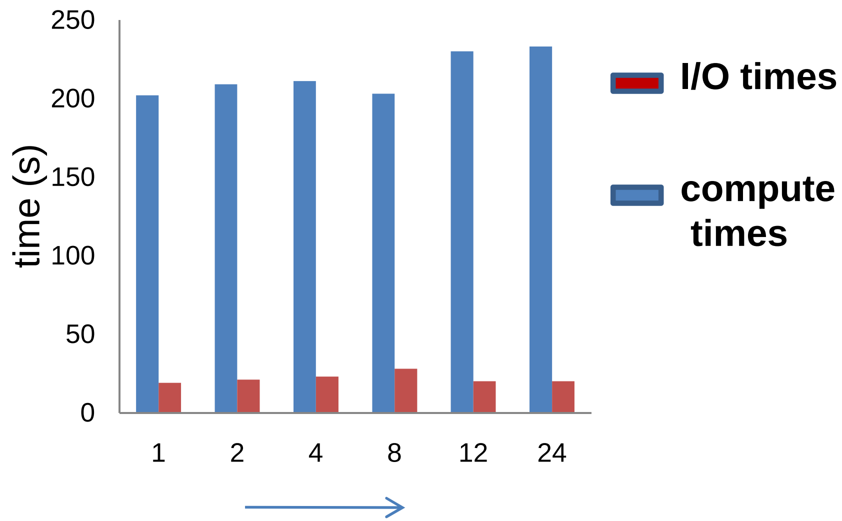
Dependency	Characterisation	Performance Variability
HPC architecture	• Scalability Memory Core speed Interconnect	***ACE*** up to 200% UM up to ~200%
System software	• Compiler • MPI implementation	***ACE*** ~20% UM ~20% ***ACE*** ~20% UM variable (but needs more investigation)
Usability	• MTBF • jitter • slowdown (throughput)	Variable with service delivery ***ACE*** ~10% UM ~10% UM >200% -1000%
I/O	• Application • I/O hardware architecture • jitter	UM up to 1000% (Need to apply Shan et al, NERSC methodology) UM ~40% on some HPC platforms

Applications can change I/O strategies

UM with a single writer I/O strategy



UM an I/O server strategy



Increasing output size and frequency

- Extend the IOR synthetic benchmarks from LLNL (used by Shan et al) to accommodate a greater range I/O strategies in a further ACE project to explore the performance dependency of the UM.
- Explore problem cases where I/O is an issue

The ACE project was

- funded by EPSRC (lead research council for UK academic HPC provision)

- managed by cross UK research council panels

- carried out by
  - EPCC (University of Edinburgh)
  - STFC Daresbury Laboratory
  - ARCA (University of Cardiff)
  - University of Cambridge HPC service

- undertaken in collaboration with many HPC services

- results will be made available via the EPSRC web site (www.epsrc.ac.uk)