UK National HPC Benchmarks

Andy Turner
2016

1. Introduction

In this document, we propose updates to the ARCHER procurement benchmarks by comparing the original set of benchmarks used in the ARCHER procurement to the usage profile on the service to date.

First we look at the original benchmark suite and comment on its coverage before providing a summary of the application usage on ARCHER to date, how these applications match to broad application types and how they match to the large scientific consortia. We analyse the usage pattern with respect to the original benchmark suite and then propose an updated set. Finally, we outline the next steps in the process to update the benchmark suite.

2. Original ARCHER Benchmark Suite

The ARCHER benchmark suite was chosen to be representative of the likely workload on ARCHER. The benchmark suite was made up of both applications benchmarks and synthetic benchmarks.

For the ARCHER procurement, all of the application benchmarks typically ran on 400-600 nodes (~10,000-15,000 cores) apart from the Met Office Unified Model (UM) benchmark that ran on ~150 nodes (~3500 cores):

- **CASTEP:** A general-purpose DFT-based materials science application. Written in Fortran with MPI and OpenMP parallelism. This application is memory-bound in most configurations and also stresses the interconnect through its heavy use of MPI all-to-all based collective operations.
- **CP2K:** Similar to CASTEP, a general-purpose DFT based materials science application. Written in Fortran with MPI and OpenMP parallelism. This application is memory-bound in most configurations and also stresses the interconnect through its heavy use of MPI all-to-all based collective operations.
- **DL_POLY:** A classical molecular mechanics-based materials science application. Written in Fortran with MPI and OpenMP parallelism. It also supports a GPGPU (CUDA) version. In most configurations this application is memory bound, but it can also be floating point performance bound with very large particle numbers and can become interconnect latency bound with small numbers of particles per parallel domain.
- **SENGA** Application to study combustion, combines computational fluid dynamics (CFD) with combustion chemistry models. Written in Fortran with MPI parallelism. This application is memory bound in most configurations.
- **Met Office UM:** Climate modelling code developed at the UK Met Office. Written in Fortran with MPI and OpenMP parallelism. The benchmark specifically excluded I/O performance, so in this configuration the application is memory bound.

The synthetic benchmarks were provided by the HPC Challenge suite:

- **HPC Challenge** (HPCC) Benchmarks:
  - HPL: Floating point performance
  - DGEMM: Floating point performance
  - Streams: Memory performance
  - PTRANS: Collective communications performance
  - RandomAccess: Memory performance
  - FFT: Memory performance
  - Comms. Bandwidth/Latency: Communications performance

Comments

The list shows that the ARCHER benchmarks have a bias towards materials science (not surprising as a large portion of the system is used for materials science by both EPSRC and NERC users) and also a heavy bias towards Fortran applications (again, not surprising as over 70% of the use on UK national HPC systems is generally from Fortran applications). Finally, almost all the application benchmarks are memory-bound to some degree. Even though the choice can be justified in terms of use profiles, it would be beneficial to see if we are able to choose a representative set that has broader coverage than the list above. It is also notable that none of the benchmarks evaluate I/O performance.

In the next section we will look at the usage patterns on ARCHER to assess if there are any changes we can propose that will make the benchmark set better meet the following criteria:

- Match the application usage profile on ARCHER.

- Provide broad coverage across different performance-critical system components: floating point performance, memory performance, interconnect performance, I/O performance, compiler performance, parallel library performance.
- Represent the usage of the range of research communities that use the service.

3. ARCHER Usage Data

All the usage data in this section is taken from the application usage monitoring tool installed on ARCHER. For a description of the tool, please see:

http://www.archer.ac.uk/documentation/white-papers/app-usage/UKParallelApplications.pdf

Top Applications by Usage

Table 1 below shows the applications on ARCHER that have used more than 1% of the total kAU to date (in order of decreasing usage).

Table 1: Applications using more than 1% of the total kAU usage.

Application	% Use	Science Area	Programming Language
VASP	17.1%	Materials Science/Chemistry	Fortran
CP2K	7.1%	Materials Science/Chemistry	Fortran
Gromacs	6.4%	Biomolecular Science	C/C++
CASTEP	6.4%	Materials Science/Chemistry	Fortran
Met Office UM	4.3%	Climate/Ocean Modelling	Fortran
HiPSTAR/SBLI	3.1%	Engineering (CFD)	Fortran
ONETEP	3.0%	Materials Science/Chemistry	Fortran
LAMMPS	2.8%	Materials Science/Engineering	C++
WRF	2.7%	Climate/Ocean Modelling	Fortran
Oasis	2.6%	Climate/Ocean Modelling	Fortran
NEMO	2.2%	Climate/Ocean Modelling	Fortran
CASINO	2.1%	Materials Science/Chemistry	Fortran
HYDRA	1.9%	Engineering (CFD)	Fortran
NAMD	1.8%	Biomolecular Science	C++
Quantum Espresso	1.6%	Materials Science/Chemistry	Fortran
OpenFOAM	1.4%	Engineering (CFD)	C++
Nektar++	1.3%	Engineering (CFD)	C++
PDNS3D	1.3%	Engineering (CFD)	Fortran
Code_Saturne	1.2%	Engineering (CFD)	Fortran
MITgcm	1.1%	Climate/Ocean Modelling	Fortran

This table immediately reveals the heavy use of Fortran on ARCHER with the majority of the remaining use coming from C++ applications. The only exception is Gromacs that still has a large amount of C.

We can also broadly match these applications to the areas of the architecture that their performance depends on, see Table 2. Note that for many of the applications listed, the component that performance is dependent on is also strongly influenced by the choice of input options so the benchmark inputs need to be carefully chosen.

Table 2: List of top applications and the system components that their performance can depend on.

Application	Performance depends on	Notes
VASP	Memory, Collective MPI Comms.	Xeon Phi, GPGPU
CP2K	Memory, Collective MPI Comms.	Xeon Phi, GPGPU, PRACE Benchmark, Existing ARCHER Benchmark
Gromacs	Compute, Memory, Point-to-Point MPI Comms.	Xeon Phi, GPGPU
CASTEP	Memory, Collective MPI Comms.	Existing ARCHER Benchmark
Met Office UM	I/O, Compute	Exisiting ARCHER Benchmark
HiPSTAR/SBLI	Compute, I/O	
ONETEP	Memory, Collective MPI Comms.	
LAMMPS	Compute, Memory, Point-to-Point MPI Comms.	Xeon Phi, GPGPU
WRF	I/O, Compute	
Oasis	I/O, Compute	Met Office UM coupled to NEMO
NEMO	I/O, Compute	
CASINO	Memory, Compute	
HYDRA	Compute, I/O	
NAMD	Compute, Memory, Point-to-Point MPI Comms.	Xeon Phi, GPGPU
Quantum Espresso	Memory, Collective MPI Comms.	
OpenFOAM	Compute, I/O	Does not scale to large core counts
Nektar++	Compute, I/O	
PDNS3D	Compute, I/O	
Code_Saturne	Compute, I/O	Xeon Phi, PRACE Benchmark
MITgcm	I/O, Compute	

Usage by Application Type

The application monitoring usage tool also provides a summary of usage by (broad) application type. Table 3 shows the major application types on ARCHER and their percentage usage. Note that LAMMPS is presented in its own category as this application has such broad use that it is hard to place: LAMMPS is used for materials science, biomolecular simulation and mesoscale engineering research.

Table 3: Broad application areas with usage more than 1% usage on ARCHER and major applications. Applications in parenthesis do not feature in the top application table above.

Application Type	% Use	Applications
Quantum Materials Modelling	36.8%	VASP, CP2K, CASTEP, ONETEP, CASINO, Quantum Espresso
Climate/Ocean Modelling	13.3%	Met Office UM, WRF, Oasis, NEMO, MITgcm
Computational Fluid Dynamics	12.3%	HiPSTAR/SBLI, HYDRA, OpenFOAM, Nektar++, PDNS3D, Code_Saturne
Biomolecular Simulation	8.7%	Gromacs, NAMD
Classical Materials Modelling	2.8%	LAMMPS
Quantum Chemistry	1.8%	(NWChem, GAMESS)
Plasma Science	1.5%	(EPOCH, GS2, OSIRIS)

Application Usage by Consortia

We can also look at the applications used by the EPSRC/NERC scientific consortia. Table 4 shows which of the top applications are used by the consortia.

Table 4: Applications used by major scientific consortia on ARCHER. Applications in parenthesis do not feature in the top application table above.

Consortium	Research Area	Major Applications
UKTC (e01)	Engineering (CFD)	HiPSTAR/SBLI, PDNS3D, Nektar++, HYDRA, Code_Saturne, OpenFOAM
MCC (e05)	Materials Science/Chemistry	VASP, CP2K, CASTEP, LAMMPS, Quantum Espresso
UKCP (e89)	Materials Science/Chemistry	CASTEP, ONETEP, VASP, CP2K
HECBioSim (e280)	Biomolecular Science	Gromacs, NAMD, LAMMPS
PPC (e281)	Plasma Physics	(EPOCH, GS2, OSIRIS)
UKCOMES (e283)	Mesoscopic Engineering	LAMMPS, (DL_MESO, HemeLB)
UKCTRF (e305)	Combustion	OpenFOAM, (SENGA)
Oceanography (n01)	Oceanography	Met Office UM, NEMO, MITgcm
Atmospheric and Polar Science (n02)	Climate Modelling	Met Office UM, Oasis, WRF
Mineral and Geo Physics (n03)	Mineral Physics/Geology	VASP, CASINO, (SPECFEM3D)

4. Analysis

Application Benchmarks

We start from the following assumptions:

- The number of application benchmarks should be small, 4-5 applications. We have limited ourselves to a maximum of 5 in this proposal and indicated which of these could be dropped to allow for an addition of 1 more benchmark if required (for example an additional scientific community joins the procurement process in the future).
- The application benchmarks chosen should reflect the usage pattern on ARCHER.
- The application benchmarks chosen should, where possible, provide broad coverage in terms of system stress (e.g. compute performance, memory performance, interconnect performance, I/O performance) and in terms of compiler functionality (e.g. Fortran, C, C++).
- Application benchmarks must be able to scale beyond 15,000 cores to provide adequate tests of capability performance on next generation HPC technologies.
- Access to the source code for applications used must be able to be made available to vendors with the minimal overhead.

The data from the analysis shows that based on broad application area usage (Table 3) we should be considering at least one application benchmark from each of the following areas:

- Quantum Materials Modelling
- Climate/Ocean Modelling
- Computational Fluid Dynamics
- Biomolecular Simulation

Original ARCHER Application Benchmarks: Comparison to Current Use Pattern

Comparing the original ARCHER application benchmark set to the current application usage and the applications employed by the different consortia it can be seen that three of the consortia are not well represented in the benchmark suite:

- UK Turbulence Consortium – no application in the benchmark suite.
    - SENGA could be considered to be partially representative but is not an ideal match and is not used to any large degree on ARCHER.
- HECBioSim – no biomolecular simulation package in the benchmark suite.
    - DL_POLY could be considered to be partially representative but is not generally used for biomolecular simulations on ARCHER.
- Plasma Physics Consortium – no plasma physics application in the benchmark suite.

Additionally, other coverage is not well represented in the original ARCHER application benchmarks:

- There is no parallel I/O benchmark as the version of the Met Office UM used in the benchmark suite has I/O specifically excluded.
- All the benchmarks are Fortran-based. Although the majority of usage is from Fortran applications, it would be useful to include benchmark applications that use the C/C++ compilers.
- The majority of the application benchmarks are memory bound so other performance characteristics of the system are not well evaluated for real applications.

Synthetic Benchmarks

Synthetic benchmarks should be able to quantify the probable maximum performance levels for the main potential hardware limitations to performance and scaling. These are:

- Compute (floating point) performance
- Memory performance (bandwidth and latency)
- Interconnect performance (bandwidth and latency)
- I/O performance (bandwidth)

Original ARCHER Synthetic Benchmarks: Coverage
The HPCC suite was used as the synthetic benchmarks in the original ARCHER benchmarks which covered the compute, memory and interconnect performance. The I/O performance was evaluated though a separate process in the ARCHER procurement using the "IOR" application.

5. Proposal

From the assumptions and analysis in Section 5 above we are able to propose an updated benchmark suite for UK National HPC procurements.

Application Benchmarks

Below we propose five application benchmarks from the areas described in Section 5 above. If an additional application benchmark was required that is not in this set (for example, a new community needs to be included) then we recommend that the CP2K application benchmark is the one that is replaced (more details in the CP2K section below). For each of the choices we provide a rationale behind the proposal.

Quantum Materials Modelling: CASTEP
https://www.castep.org/

CASTEP is a general-purpose DFT-based materials science application. Written in Fortran with MPI and shared memory parallelism using OpenMP. CASTEP has also been ported to Intel Xeon Phi (KNL) and GPGPU. This application is memory-bound in most configurations and also stresses the interconnect through its heavy use of MPI all-to-all based collective operations. CASTEP was chosen as:

- VASP was discounted from selection as the licensing model for the application means that it is difficult to get access to the source code for vendors.
- The CASTEP algorithms and parallelisation are similar to those used in VASP, the highest use application in this area.
- CASTEP has been used as a benchmark in a number of previous UK National Tier-1 HPC procurements and in the ACE benchmarking projects so a large amount of historical performance data already exists.
- The application can also be run on novel architectures such as Intel Xeon Phi and GPGPU.

The CASTEP benchmark models the performance of the following high usage applications:

- VASP, CP2K, ONETEP, Quantum Espresso

CASTEP performance is of interest for the following scientific consortia on ARCHER:

- MCC (e05), UKCP (e89), Mineral and Geo Physics (n03)

Quantum Materials Modelling: CP2K
https://www.cp2k.org/

CP2K is an Open Source general-purpose DFT-based materials science application. Written in Fortran with MPI and OpenMP parallelism. CP2K has also been ported to Intel Xeon Phi (KNL) and GPGPU. This application is memory-bound in most configurations.

CP2K is included in the benchmark suite in addition to CASTEP as:

- There is little to choose between the two applications in terms of usage and they are used by different research areas within the materials science community.
- The algorithms and parallelisation strategy used in CP2K are substantially different from those used in CASTEP and VASP.
- It is already part of the PRACE application benchmark suite and this would allow for comparison across a wide range of systems.

- Extensive development work has been performed on Xeon Phi systems by Intel Parallel Computing Centres (IPCCs).

If an application had to be dropped from the benchmark suite to make way for another application, then CP2K should be the one dropped as:

- There are two applications in the materials science area in the suite (CASTEP and CP2K) so the community would still be represented.
- CASTEP should be kept as its performance is a better model for that of VASP, the most heavily used application in this area.

CP2K performance is of interest for the following scientific consortia on ARCHER:

MCC (e05), UKCP (e89), Mineral and Geo Physics (n03)

Climate/Ocean Modelling: OASIS3-MCT (Met Office UM coupled to NEMO)
https://verc.enes.org/oasis
http://www.metoffice.gov.uk/research/modelling-systems/unified-model
http://www.nemo-ocean.eu/

OASIS3-MCT is a parallel earth system coupling framework; Met Office UM is a climate modelling code developed at the UK Met Office; NEMO is an ocean modelling code. All components are written in Fortran with MPI parallelism, Met Office UM and NEMO also support OpenMP parallelism. This benchmark should include the use of parallel I/O servers in the Met Office UM and XIOS in NEMO so that it becomes I/O bound in the same way as the application as used by users. This benchmark is included as:

- OASIS3-MCT coupled models are heavily used by the NCAS consortia.
- Met Office UM is heavily used by the NCAS and NOC user communities and NEMO is heavily used by the NOC user community.
- None of the application benchmarks in the original set stress the I/O performance of the system. This benchmark tests the I/O performance of the key I/O bound applications in the ARCHER user community

This benchmark models the performance of the other high usage applications:

- WRF, MITgcm

Met Office UM performance is of interest for the following scientific consortia on ARCHER:

- Oceanography (n01), Atmospheric and Polar Science (n02)

Computational Fluid Dynamics: OpenSBLI
https://arxiv.org/abs/1609.01277

OpenSBLI is a high level framework for finite-difference based models, particularly for CFD simulations. It uses a Python-based Domain Specific Language (DSL) which can then generate C++ source code with (optionally) OpenMP, CUDA, OpenCL or OpenACC components for a variety of computer architectures (e.g. CPU, GPGPU). This application is generally compute bound but certain phases in the calculation are I/O bound.

OpenSBLI was chosen in consultation with users for the following reasons:

- OpenSBLI is most representative of future capability work within the UKTC (the largest CFD user group on Tier-1 UK National HPC).
- OpenSBLI uses a DSL code generation approach, this programming model is rising in importance and it is useful to have a benchmark that uses this approach.
- OpenSBLI stresses the C++ compilers.

- As the DSL approach can target a range of different processor architectures the benchmark can be run across a wide range of HPC systems.
- HiPSTAR/SBLI are still actively used by the community but are not as representative of the capability requirements of the community.
- OpenFOAM was discounted as it is not able to scale to the required number of cores.

The OpenSBLI benchmark models the performance of the following high usage applications:

- HiPSTAR/SBLI, PDNS3D, HYDRA, OpenFOAM, Code_Saturne

OpenSBLI performance is of interest for the following scientific consortia on ARCHER:

- UKTC (e01), UKCTRF (e305)

Biomolecular Simulation: Gromacs
http://www.gromacs.org/

A classical molecular mechanics-based biomolecular simulation application Written in C/C++ with MPI and OpenMP parallelism. It also supports a GPGPU (CUDA) and Xeon Phi (KNL) versions. In most configurations this application is memory bound, but it can also be floating point performance bound with very large particle numbers and can become interconnect latency bound with small numbers of particles per parallel domain.

- Gromacs is the most heavily used biomolecular simulation package on ARCHER.
- Benchmarks can be chosen to stress different aspects of the system as required.
- Gromacs provides a test of the C/C++ compiler.
- The application can also be run efficiently on novel architectures such as GPGPU and Xeon Phi.

Gromacs benchmark models the performance of the following high usage applications:

- LAMMPS, NAMD

Thus, Gromacs performance is of interest for the following scientific consortia on ARCHER:

- MCC (e05), HECBioSim (e280), UKCOMES (e283)

Synthetic Benchamrks

As described above, the main role of synthetic benchmarks is to stress all the hardware components that can potentially limit performance to provide upper limits on expected application performance. They also play a role in stressing system and hosting infrastructure.

We propose keeping the original HPCC synthetic benchmark suite and adding the *benchio* synthetic parallel I/O benchmark.

Compute, memory, interconnect synthetic benchmarks: HPC Challenge
http://icl.cs.utk.edu/hpcc/

HPCC includes the following components:

- HPL: Floating point performance, power and cooling infrastructure stress test
- DGEMM: Floating point performance
- Streams: Memory performance
- PTRANS: Collective communications performance
- RandomAccess: Memory performance
- FFT: Memory performance
- Comms. Bandwidth/Latency: Communications performance

HPCC should be retained as it contains standard benchmarks that are used to evaluate HPC system performance worldwide.

I/O synthetic benchmark: benchio
https://github.com/EPCCed/benchio

benchio is synthetic parallel I/O benchmark that evaluates the performance of MPIIO, HDF5, and NetCDF.

- Parallel I/O is now one of the key performance characteristics of any HPC system.
- Investigations have revealed that the standard parallel I/O synthetic benchmark "IOR" is not a useful model of real parallel I/O performance for HPC applications.
- benchio is designed to model commonly used parallel I/O patterns in real applications.
- benchio can test MPIIO, HDF5, and NetCDF performance. These are the three most commonly used parallel I/O libraries on ARCHER.

Note: The "IOR" benchmark is often used to evaluate parallel I/O performance on systems. This benchmark was considered but was not suggested because:

- Most importantly, the IOR model of parallel IO is very simplistic and is a poor model of how applications actually perform parallel IO. The data layout is so simple that it does not stress the parallel IO libraries to any real extent.
- IOR has so many options that it very difficult to interpret any results. This also gives a huge amount of flexibility to vendors to choose a configuration to give "best" performance and hampers the ability to compare results across systems.
- Test IOR runs using parallel IO on ARCHER have given results that do not make any sense (for example, performance numbers above theoretical peak). Consulting with Cray, we have not yet been able to work out what the software is doing to give these spurious results.

Summary of Differences from Original Benchmarks

In this section we summarise the main changes from the original benchmarks.

Application benchmarks:

- Gromacs has been added in place of DL_POLY. DL_POLY is not used to a significant fraction on ARCHER whereas Gromacs is used extensively by the biomolecular simulation community. We would expect Gromacs to provide a good model of DL_POLY performance. Gromacs has the additional advantage of testing the C/C++ compiler performance.
- OpenSBLI has replaced SENGA. SENGA is not used to any significant degree on ARCHER and we would expect its demands/performance to be well modelled by OpenSBLI. OpenSBLI tests the DSL, code generation model that is becoming increasingly popular and will be more representative of the capability usage of the UKTC community going forwards.
- We recommend that the Met Office Unified Model benchmark includes the parallel I/O component that was excluded from the ARCHER benchmarks as this is the key factor in its performance on a production system.
- We provide a recommendation of an application benchmark that can be dropped (CP2K) to make space for an additional, different benchmark if required.

Synthetic benchmarks:

- The EPCC parallel I/O benchmark "benchio" has been added to add an application-relevant test of parallel I/O performance using the most common APIs (MPIIO, NetCDF, HDF5).

6. Next Steps

Although we propose the applications to use for benchmarking here, this does not include the inputs for the benchmarks themselves. The next steps in this process are:

- Approach appropriate user groups need to provide suitable benchmark inputs.
- Finally, the benchmarks should be baselined on ARCHER.

Benchmarking on ARCHER: Considerations

When running the benchmarks on ARCHER, care has to be taken as the performance on a production system that has been in operation for a number of years may not be directly comparable to performance on a brand new, empty system.

Below is a brief summary of how this could impact particular performance characteristics and what we may be able to do to mitigate the issues.

- **Compute and Memory Performance:** Tests of compute and memory performance are not generally a problem as these resources are allocated for exclusive use by a particular job.

- **Interconnect Performance:** Interconnect is a shared resource and so the performance of a benchmark can, conceivably be impacted by other jobs running on the system. We can measure performance variation by repeating the benchmark multiple times. This variation is a useful measure in itself and, along with theoretical performance numbers, should allow us to evaluate performance properly.

- **I/O Performance:** I/O is more problematic as the performance is not only affected by other users' jobs (as it is a shared resource) but also by age: as the file system fills up, the performance will drop. As for interconnect performance, repeated runs would be used to assess performance variation in production (which is interesting in itself) and we have access to the theoretical performance numbers to help evaluate the benchmark results.