

I/O Performance Benchmarking and Investigation on Multiple HPC Architectures



# Document Information and Version History

|  |  |
| --- | --- |
| **Version:** | 0.2 |
| **Status** | Draft |
| **Author(s):** | Bryan Lawrence , Chris Maynard , Andy Turner, Xu Guo, Dominic Sloan-Murphy, Juan Rodriguez Herrera |
| **Reviewer(s)** | David Henty |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Comments, Changes, Status** | **Authors, contributors, reviewers** |
| 0.1 | 2016-06-22 | Initial draft | Andy Turner |
| 0.2 | 2016-09-27 | Edits to abstract and system introductions | Dominic Sloan-Murphy |

# Abstract

I/O performance plays a key role in many scientific simulations and the bottleneck of I/O is an important challenge to solve towards Exascale computing. It is therefore necessary for CoEs and scientific communities with high I/O requirements to understand the usage pattern of existing HPC systems and applications to be suitably equipped to make informed plans for the future.

Theoretical performance numbers assume a clean formatted file system with no contention from other users. Obviously, when used in full production, this level of performance will not be attained.

One of the goals of this paper is to provide insight into the performance of the file systems *in production*. To answer questions such as: What is the maximum performance actually experienced? What variation in performance do users experience?

To that end, this paper details the I/O performance of multiple HPC architectures through testing a set of selected I/O benchmarks and representative scientific applications. Results are presented from the following systems:

* ARCHER: the UK national supercomputing service, using a Lustre filesystem.
* BlueGene/Q: the flagship machine of the UK DiRAC consortia, using an IBM GPFS filesystem.
* COSMA: a further DiRAC resource, using GPFS.
* UK-RDF DAC: the Data Analytic Cluster attached to the UK Research Data Facility, using GPFS.
* Cirrus: an industry-focused machine housed at the EPCC computing facility, using Lustre.
* \*\*\*Description of scientific application machine\*\*\*

In the Lustre case, a range of stripe counts and sizes are tested. GPFS provides less scope for user tuning, hence figures are given under the default configuration.

We benchmark a simple MPI-IO parallel test (produced by the EPCC) that writes a three-dimensional distributed dataset to a single shared file. On supported systems, we further measure and compare the performance of HDF5 and NetCDF equivalent implementations.

\*\*\*Description of scientific applications from ESiWACE and Met Office\*\*\*

We find \*\*\*analysis and conclusions\*\*\*

# HPC Systems

## ARCHER

The /work file systems on ARCHER use the Lustre technology in the form of Sonexion parallel file system appliances. The theoretical sustained performance (in terms of bandwidth) of Sonexion Lustre file systems is determined by the number of SSUs (Scalable Storage Units) that make up the file system. ARCHER has four Sonexion file systems:

* fs1: 1 SSU, theoretical sustained = 5 GB/s
* fs2: 6 SSU, theoretical sustained = 30 GB/s
* fs3: 6 SSU, theoretical sustained = 30 GB/s
* fs4: 7 SSU, theoretical sustained = 35 GB/s

# Parallel I/O benchmark: benchio

The parallel I/O performance of the file systems was evaluated by the *benchio* application developed at EPCC. This was chosen ahead of IOR for a number of reasons:

* The parallel I/O decomposition can be varied to better model actual user applications. IOR uses an extremely simplistic 1D data decomposition that does not model user codes and does not test the performance of MPIIO collective operations that are key to real performance.
* The IOR code is very opaque, this makes it very difficult to draw useful conclusions as to what variations in performance are due to.
* benchio is also able to evaluate the performance of HDF5 and NetCDF, two libraries that support parallel I/O and are commonly used by user communities on ARCHER. Note that this paper currently only covers MPIIO performance.

The benchio source code is Open Source and is available on GitHub.

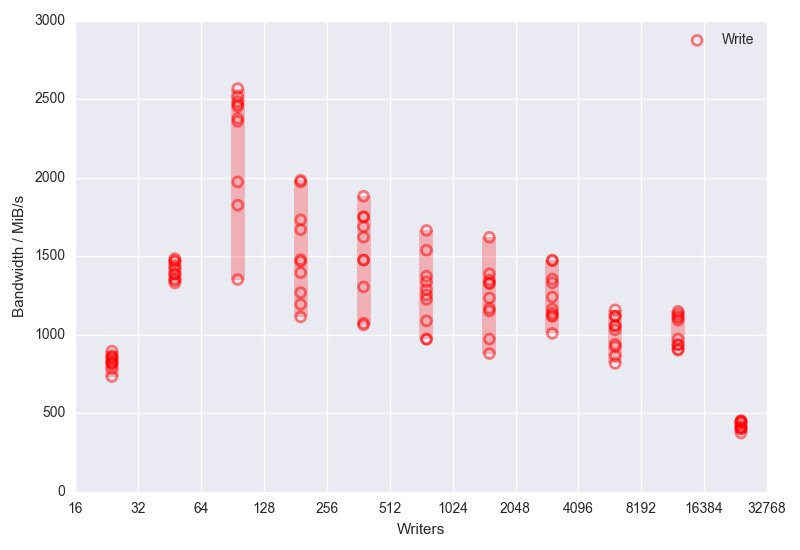
[Description of what benchio actually does]

# ARCHER Parallel Write Performance

Using the default Lustre settings on ARCHER:

* Stripe size: 1 MiB
* Number of stripes: 4

we see the performance shown in Figure and listed in Table.



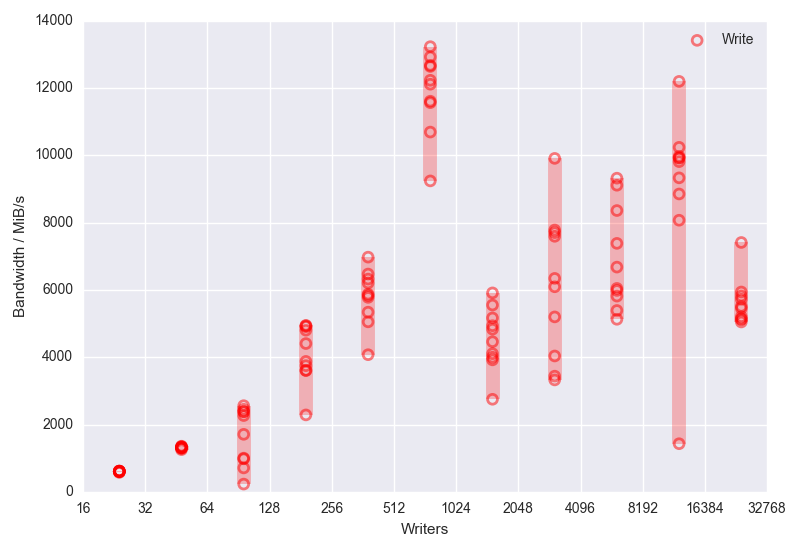
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Write Bandwidth (MiB/s)** | | | |  |
| **Writers** | **Min.** | **Median** | **Max.** | **Mean** | **Count** |
| 24 | 733.939 | 827.945 | 896.015 | 828.069 | 10 |
| 48 | 1328.688 | 1403.117 | 1484.661 | 1407.661 | 10 |
| 96 | 1351.340 | 2414.556 | 2567.143 | 2236.714 | 10 |
| 192 | 1113.092 | 1471.873 | 1982.988 | 1526.920 | 10 |
| 384 | 1061.639 | 1548.425 | 1881.732 | 1508.003 | 10 |
| 768 | 970.448 | 1270.291 | 1663.967 | 1270.813 | 10 |
| 1536 | 880.089 | 1279.216 | 1620.391 | 1241.438 | 10 |
| 3072 | 1008.584 | 1202.115 | 1475.161 | 1242.080 | 10 |
| 6144 | 817.562 | 1044.328 | 1158.771 | 1008.712 | 10 |
| 12288 | 901.767 | 1032.680 | 1149.057 | 1025.447 | 10 |
| 24576 | 374.109 | 423.413 | 452.824 | 422.868 | 10 |

Using the default stripe settings on ARCHER, the maximum write performance that can be achieved is just over 2,500 MiB/s, well short of the theoretical sustained performance of 30,000 MiB/s.

As described in Parallel I/O Performance on ARCHER, to get the best parallel write performance we must use as many stripes as possible. This is achieved on Lustre by setting the striping to “-1”which stripes over all available OSTs. We repeated the benchmarks with:

* Stripe size: 1 MiB
* Number of stripes: -1 (corresponds to 48 on fs3)

The performance for this configuration is shown in Figure and Table.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Write Bandwidth (MiB/s)** | | | |  |
| **Writers** | **Min.** | **Median** | **Max.** | **Mean** | **Count** |
| 24 | 576.868 | 612.247 | 615.717 | 605.801 | 10 |
| 48 | 1252.985 | 1318.389 | 1355.754 | 1313.825 | 10 |
| 96 | 230.472 | 1986.700 | 2559.369 | 1666.729 | 10 |
| 192 | 2284.795 | 4138.974 | 4943.626 | 4109.961 | 10 |
| 384 | 4073.371 | 5852.313 | 6971.013 | 5790.246 | 10 |
| 768 | 9235.570 | 12166.017 | 13222.881 | 11886.082 | 10 |
| 1536 | 2749.108 | 4645.424 | 5909.421 | 4561.950 | 10 |
| 3072 | 3323.747 | 6215.290 | 9904.133 | 6139.148 | 10 |
| 6144 | 5125.016 | 6358.722 | 9316.783 | 6918.379 | 10 |
| 12288 | 1429.390 | 9859.596 | 12192.030 | 8969.973 | 10 |
| 24576 | 5045.687 | 5480.116 | 7406.748 | 5633.765 | 10 |

When using the maximum number of stripes, we see much improved performance (compared to the default stripe count of 4) with a maximum write bandwidth of over 13,000 MiB/s with 576 cores (32 nodes) writing simultaneously. This is still less than 50% of the advertised sustained bandwidth of 30,000 GiB/s for this file system.