benchmark_c: A tool to compare I/O performance from MPI-IO, HDF5 and ADIOS2

Shrey Bhardwaj [1]
Supervisors: Dr. Paul Bartholomew [1] Prof. Mark Parsons [1]
[1]EPCC, University of Edinburgh

Introduction

Current Petascale HPC systems perform 10^{15} computations per second. However future challenges require even faster computational speeds. One of the major roadblocks to achieving these speeds is the I/O bottleneck.

This I/O bottleneck can be seen in figure 1 obtained from the CFD code, Xcompact3D [5].

This poster introduces benchmark_c [2] which has been developed to benchmark I/O backends to find improvements in the I/O bandwidth.

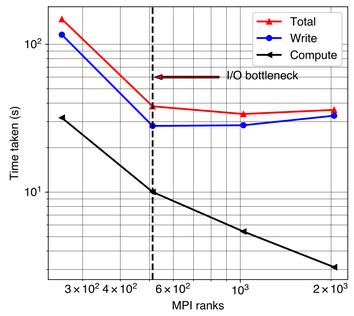
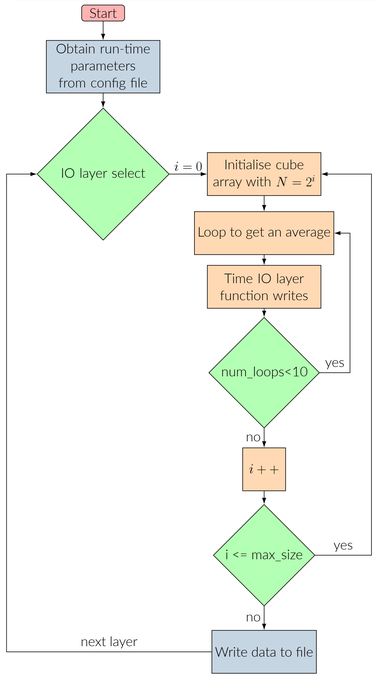


Figure 1:Parallel performance of Xcompact3D for increasing MPI ranks

benchmark_c



- benchmark_c [2] writes increasing array sizes using different I/O backends.
- It is derived from benchio [1], a fortran based application.
- A data array is passed to either MPI, HDF5, ADIOS2 HDF5 IO engine or ADIOS2 BP4 IO engine for writing to disk.
- The results were obtained by submitting this job 3 times and averaged to account for any system-wide noise.

Machines used	NextGenIO [4]	ARM Fulhame Cluster [3]
Total number of nodes	34	64
Cores per node	48	64
Memory per node (GB)	196	256
Compute environment	gnu/10.2.0 intel-mpi /2021.3.0 HDF5/1.12.0 ADIOS2/2.7.1	gnu/9.2.0 openmpi /4.0.2 HDF5/1.12.0 ADIOS2/2.7.1

Table 1:Details of HPC machines and modules used

Comparison of different backends

First different I/O backends were investigated for any improvement in I/O performance. This was conducted over multiple node configurations using NextGenIO with full Lustre striping across the run directory. In this experiment, the MPI ranks were used as follows; 1 node in serial (1 Processor), 1 node half full (24 Processors), 1 full node, 2 full nodes, 4 full nodes and 8 full nodes.

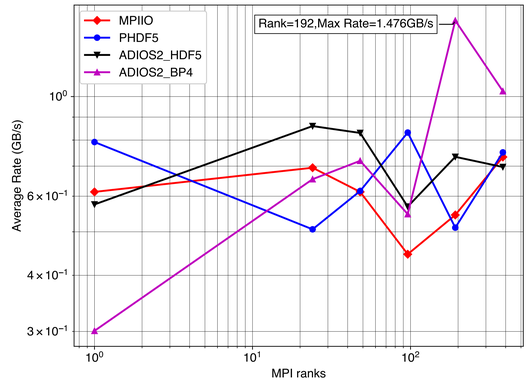


Figure 2:Comparison of achieved bandwidth using backends with local array size of 0.13GB

Comparison with different Machines

Next, this experiment was repeated on different machines, NextGenIO HPC system and ARM Fulhame Cluster. In addition to this, the jobs were run with maximum and minimum striping in both the machines with a local array size of 0.13GB with the same node configurations as before.

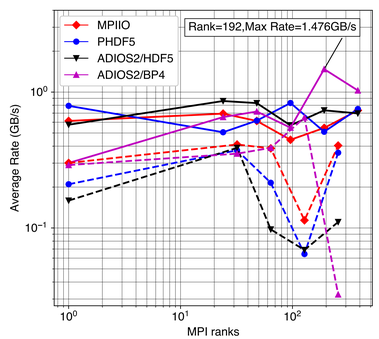


Figure 3:Benchmarking results on NextGenIO and Fulhame (marked by dashed line)

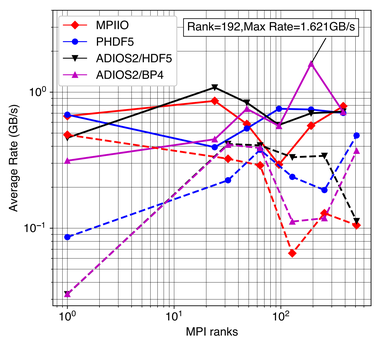


Figure 4:Benchmarking results on NextGenIO and Fulhame (marked by dashed line)

Speedup comparison of ADIOS2 I/O engines

Lastly, the ADIOS2 I/O engines were compared relative to HDF5 for performance advantages using their default settings. The experiments were run on NextGenIO with different MPI ranks and full Lustre striping across the run directory. In figure 5, two job sizes are compared, 1 MPI rank (serial) and 384 MPI ranks (8 Nodes) to compare the benefits of the two I/O engines with varying levels of parallelism upto a local array size of 0.13GB.

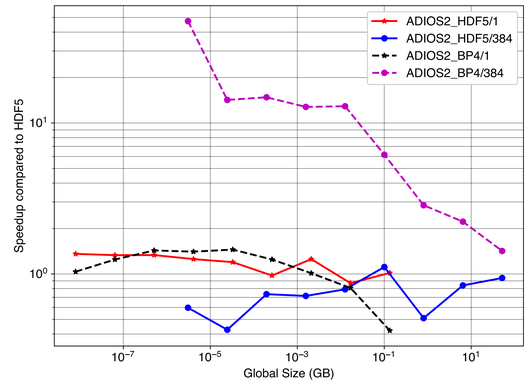


Figure 5:Speedup achieved w.r.t. I/O rates from HDF5 with local data size upto 0.13GB

Conclusions

It is observed that ADIOS2 BP4 I/O engine provides much better bandwidth than the other I/O layer backends. This is possibly due to factors such as the innovative BP4 native metadata system and write buffering system. From figure 4 it is observed that better rates are obtained by using NextGenIO compared to Fulhame.

In the future, larger sized arrays would be used for benchmarking. It would be useful to investigate the benefits of ADIOS2 BP4 I/O engine and its many configurable options. It is also planned to investigate advanced hardware such as the NVRAM storage of NextGenIO [6].

Acknowledgments

The Fulhame HPE Apollo 70 system is supplied to EPCC, the supercomputing centre at the University of Edinburgh, as part of the Catalyst UK programme, a collaboration with Hewlett Packard Enterprise, Arm and SUSE to accelerate the adoption of Arm based supercomputer applications in the UK.
The NEXTGenIO system was funded by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671951.
This work was supported by an EPCC funded studentship as part of the ASiMoV project. Funding from EPCC is gratefully acknowledged.

References

[1] benchio. `https://github.com/EPCCed/benchio.git`.
[2] benchmark_c. `https://github.com/sb15895/benchmark_c.git`.
[3] Fulhame. `https://www.epcc.ed.ac.uk/facilities/other-facilities/fulhame`.
[4] Nextgenio. `http://www.nextgenio.eu`.
[5] Xcompact3d. `github.com/xcompact3d/Incompact3d`.
[6] Adrian Jackson, Michèle Weiland, Mark Parsons, and Bernhard Homölle. An Architecture for High Performance Computing and Data Systems Using Byte-Addressable Persistent Memory. 2019. doi:10.1007/978-3-030-34356-9_21.