Parallel IO Performance on ARCHER2

David Henty

1 Introduction

The aim is to investigate how parallel IO performance is affected by a number of factors:

- the Lustre file system settings;
- the choice of MPI-IO, HDF5 or NetCDF;
- data aggregation within the Cray MPI-IO library.

Running the exercises should also give you some feeling for the kinds of IO rates achievable in practice, enabling you more easily to judge whether the IO phases of your own codes can or should be improved.

2 Test code

The test code is called `benchio` and is a simple Fortran program that performs parallel IO using a number of serial and parallel approaches.

For details, see `README.md` at `https://github.com/davidhenty/benchio`.

3 Setup

On ARCHER2 **you must work in the /work/ file system**, i.e. `cd /work/ta048/ta048/username/`.

Obtain the code from github:

```
git clone https://github.com/davidhenty/benchio
```

Before compiling, you must load some additional modules:

```
module load cray-hdf5-parallel
module load cray-netcdf-hdf5parallel
```

Now you should be able to compile:

```
make -f Makefile-archer2
```

Before running the code, you should configure the three output directories as follows:

- remove any striping on `unstriped` (i.e. set it to 1): `lfs setstripe -c 1 unstriped`
- `striped` should be set to a stripe count of 2: `lfs setstripe -c 2 striped`
- set striping on `fullstriped` to the maximum (i.e. system-selected) value: `lfs setstripe -c -1 fullstriped`

4 Initial exercise

You should be able to submit a job using the supplied SLURM script:

```
sbatch archer2.job
```

This will run a very simple test with serial IO of a 1 GiB file from a single process:

```
./benchio 512 512 512 local serial unstriped
```

What IO rates do you see from a single process?

5 Basic IO performance

We can try and estimate the maximum IO bandwidth from a node (128 CPU-cores) using the following settings:

```
#SBATCH --tasks-per-node=128
...
./benchio 256 256 256 local proc unstriped
```

It is a good idea to reduce the local array sizes as the total amount of data can become very large when running on hundreds of processes.

We can see if the limiting factor is the speed of the disks or the bandwidth from a single node by running on multiple nodes:

```
#SBATCH --nodes=2
#SBATCH --tasks-per-node=128
...
./benchio 256 256 256 local proc unstriped
```

How do the bandwidths from these tests compare?

Other things you could look at:

- To test if the MDS is a limiting factor you could write the same amount of data but using fewer processes, e.g. a 256^3 array with only 16 tasks per node.
- For serial IO we would not expect striping to be beneficial. What happens if you re-run the tests using the `striped` directory?

6 MPI-IO performance

For parallel IO we would expect the bandwidth to scale with the number of stripes.

Try the following tests:

```
#SBATCH --nodes=1
#SBATCH --tasks-per-node=128
...
./benchio 128 128 128 local mpiio
```

which will perform MPI-IO to all three of the directories. Now try with 2, 4 and 8 nodes - what IO rates do you see?

Other things you could look at:

- We expect that using non-collective IO routines will be very slow. Edit the file `mpiio.f90` and replace `MPI_File_write_all` with `MPI_File_write`.

  Note that the performance could be so slow that it is hard to measure (e.g. it takes too long to run). If so, try running on smaller numbers of tasks.

- How are the IO rates affected if you do not use the `native` data format, e.g. you replace this with `external32`?

7 Configuring the Lustre filesystem

Although by far the most important factors for achieving good parallel IO rates are to use collective IO and stripe the file across multiple OSTs, there are other settings that can be changed.

The Cray MPI-IO library can oupout useful statistics on how it performed your IO - to get this debug output, set the following variable in your SLURM script:

```
export MPICH_MPIIO_STATS=1
```

- Does increasing the stripe size have an effect, e.g. what happens of you use 4 MiB stripes instead of the default 1 MiB?
- Try using the alternative implementation of the Cray MPI library by siwthcing modules in your SLURM script (you do not need to recompile)

  ```
  module swap craype-network-ofi craype-network-ucx
  module swap cray-mpich cray-mpich-ucx
  ```

- By default, the MPI-IO library only allocates a single IO process for each stripe. This may not be optimal - you can try increasing this number, e.g. to double the number of IO processes:

  ```
  export MPICH_MPIIO_HINTS=*:cray_cb_nodes_multiplier=2
  ```

8 Alternative IO libraries

Repeat some of the MPI-IO experiments using a higher-level library such as HDF5. For example, to run on a single node:

```
#SBATCH --nodes=1
#SBATCH --tasks-per-node=128
...
./benchio 128 128 128 local hdf5
```

Are the IO rates similar to MPI-IO?