Group Assignment #4: Parallel I/O

Christopher D. Carothers
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, New York U.S.A. 12180-3590
Email: chrisc@cs.rpi.edu

March 10, 2017

DUE DATE: Noon, Friday, March 24th, 2017

1 Group Assignment Description

For this assignment, you can work in a group of up to 4 students. As a group, you are to develop an MPI-based C program which performs the following parallel I/O computation using the IBM Blue Gene/Q and the collective `MPI_File_write_at_all` and `MPI_File_read_at_all` with 64 MPI ranks per node for the following configurations:

1.1 Single File

- Processor Block size: 1M, 2M, 4M, 8M and 16M. Note, these are dummy blocks of data.
- Blocks to write/read: 16 per MPI rank.
- MPI Rank Counts: 512 (8 nodes), 1024 (16 nodes), 2048 (32 nodes), 4096 (64 nodes) and 8192 (128 nodes)
- Total experiments: 25 write and 25 read. *Have the "reads" use data created in previous "write" experiment.*

1.2 16 Files

With 16 files, the number of MPI ranks per file will range between 32 and 512.

- Processor Block size: 1M, 2M, 4M, 8M and 16M. Note, these are dummy blocks of data.
- Blocks to write/read: 16 per MPI rank.
- MPI Rank Counts: 512 (8 nodes), 1024 (16 nodes), 2048 (32 nodes), 4096 (64 nodes) and 8192 (128 nodes)
- Total experiments: 25 write and 25 read. *Have the "reads" use data created in previous "write" experiment.*

1.3 64 Files

With 64 files, the number of MPI ranks per file will range between 8 and 128.

- Processor Block size: 1M, 2M, 4M, 8M and 16M. Note, these are dummy blocks of data.
- Blocks to write/read: 16 per MPI rank.
- MPI Rank Counts: 512 (8 nodes), 1024 (16 nodes), 2048 (32 nodes), 4096 (64 nodes) and 8192 (128 nodes)
- Total experiments: 25 write and 25 read. *Have the "reads" use data created in previous "write" experiment.*

1.4 128 Files

With 128 files, the number of MPI ranks per file will range between 4 and 64.

- Processor Block size: 1M, 2M, 4M, 8M and 16M. Note, these are dummy blocks of data.
- Blocks to write/read: 16 per MPI rank.
- MPI Rank Counts: 512 (8 nodes), 1024 (16 nodes), 2048 (32 nodes), 4096 (64 nodes) and 8192 (128 nodes).
- Total experiments: 25 write and 25 read. *Have the "reads" use data created in previous "write" experiment.*

2 Performance Measurement

For this assignment, you need to time using the BG/Q cycle counter, the parallel I/O operations. To use this timer and overall order of operations for this assignment, see the following example below.

```
#include<hwi/include/bqc/A2_inlines.h>
....
unsigned long long start_cycle_time=0;
unsigned long long end_cycle_time=0;
unsigned long long total_cycle_time=0;
....
MPI_File_open();

start_cycle_time = GetTimeBase();
MPI_File_write_at_all() or MPI_File_read_at_all();
end_cycle_time = GetTimeBase();

total_cycle_time = end_cycle_time - start_cycle_time;
MPI_Allreduce( total_cycle_time );

MPI_File_close();
```

The clock rate on the BG/Q is 1.6 GHz.!!!! You need to use that that convert the cycle times to seconds and ultimately bandwidth. For example, if it takes 3.2 billion clock cycles to complete the I/O operation that is writing 16, 8MB blocks using 1024 MPI ranks. The total time is 3.2 billion cycles / 1.6 billion (BG/Q clock rate) which is 2 seconds total. The total data is $1024 * 16 * 8MB = 128GB$. So, the bandwidth is 128 GB in 2 seconds or 64 GB/sec. Note that 1024^3 bytes is 1 GB.

Use an `MPI_Allreduce` operation to compute the MAX cycles across all ranks and use that for your time and bandwidth calculations.

Last, do not include the time to open or close your files, only the time to perform the collective read and write operations.

3 Graphing Details

NOTE: Please clean up your data files in `scratch` after each run since the amount storage consumed by the largest run is 2TB

Measure the execution time of each read and write run using the hardware clock information provided above, convert to a bandwidth. Next, plot the bandwidth performance results of each read data set and write data set onto separate graphs. Here, each 3-D graph will be a function of blocksize and MPI rank count (X and Y axis) with the data bandwidth being the Z axis. You will have 25 plot points per graph.

For graphing software, Excel has the ability to make good 3-D graphs as does Gnuplot.

With read and write graphs for each of the 4 file count configurations, you will have 8 graphs total.

In your report, explain the performance trends you are observing for each graph. Some questions to consider:

- Are reads slower than writes ?
- Does performance increase or decrease as you increase the number of MPI ranks and if so what the peak.
- Does performance increase or decrease as you increase the number total files being written and if so, what is the peak performance ?

Again, you can do this in teams of up to 4 students per team.

4 HAND-IN INSTRUCTIONS and GRADING CRITERIA

Leave your code and write-up (PDF format) in your **assignment4** sub-directory on 1 team member's account on `kratos.cs.rpi.edu`. Make sure you include all team member names on the project report.

Finally, we'll be using the following check list and point deductions when grading your assignments.

- **CORRECT SUBMISSION: 10 points.** Source code is inside your **assignment4** directory. The code compiles without errors. Note, we will stub out the `GetTimeBase()` function on kratos.
- **CORRECT RESULT: 40 points.** The code produces the intended result.
- **ANALYSIS: 50 points**. All graphs are displayed with labels. An explanation of what the graphs represent is presented.
- **Other Point deductions:**

- -5pts: no labels and no mention of labels.
- -20pts: If no graphs exist.
- -30pts: If no explanation (this is the main point of the project).
- -2pts: If the source code is not in a file, but in the PDF.