

Performance comparison of Collective vs Non-collective (Interleaved mode)

Christian Chilan
chilan@ncsa.uiuc.edu
April 27, 2005

Although in many cases POSIX is the API that yields the best performance due to its low overhead, there are cases in which MPIIO and PHDF5 perform much better by taking advantage of the access pattern and the parallel capabilities of the platform.

The configuration for one of such cases is the following:

Processes 4
Block size 32KB
File size 128MB per process
Mode Collective - Interleaved
Transfer size 32KB:16MB
System copper

As shown in Figure1, MPI and PHDF5 perform better than POSIX for transfer sizes larger than 128KB by combining several write operations into a single request which reduces the high costs associated with latency.

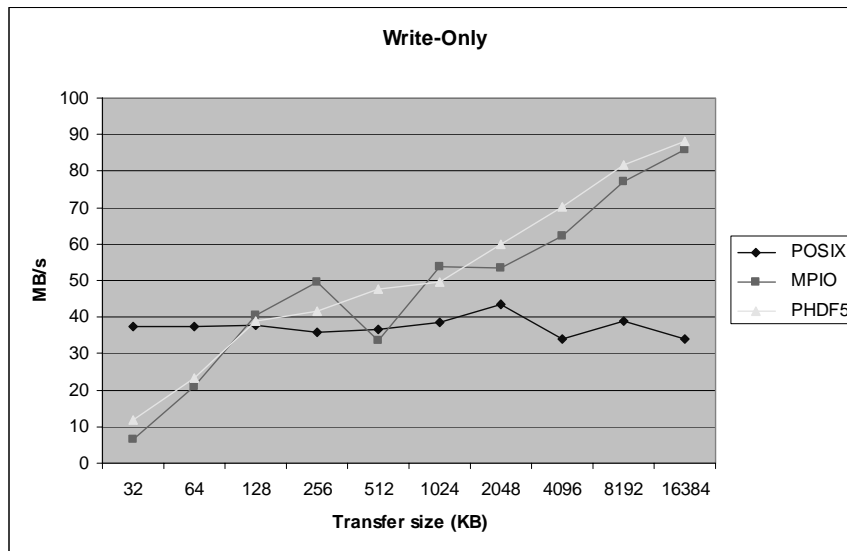


Figure 1 Throughput in collective interleaved mode

As expected, this advantage is only available in collective mode. Figure 2 shows that in non-collective interleaved mode, MPIIO and PHDF5 throughput is close to the one of POSIX. Note that POSIX measurements remain approximately the same regardless collective mode is enabled or not.

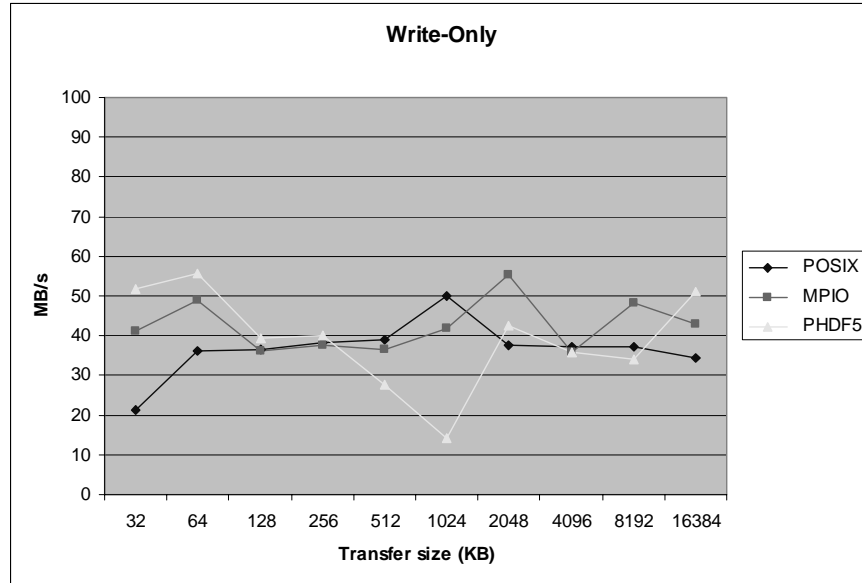


Figure 2 Throughput in non-collective interleaved mode

Modification of h5perf

The MPIO and POSIX write tests in h5perf have similar loop structures. When the collective mode is not enabled, the interleaved option forces each process to write every block at a time.

In order to increase the throughput of MPIO, we modified the code such that the write operations do not go through loop iterations. A way to achieve this is to use `MPI_File_Set_View` so that each process can write all the blocks within the transfer buffer using a single MPI call. As Figure 3 shows, this modification did not yield good performance. In fact, it caused MPIO to be consistently slower than POSIX for transfer sizes larger than 128KB.

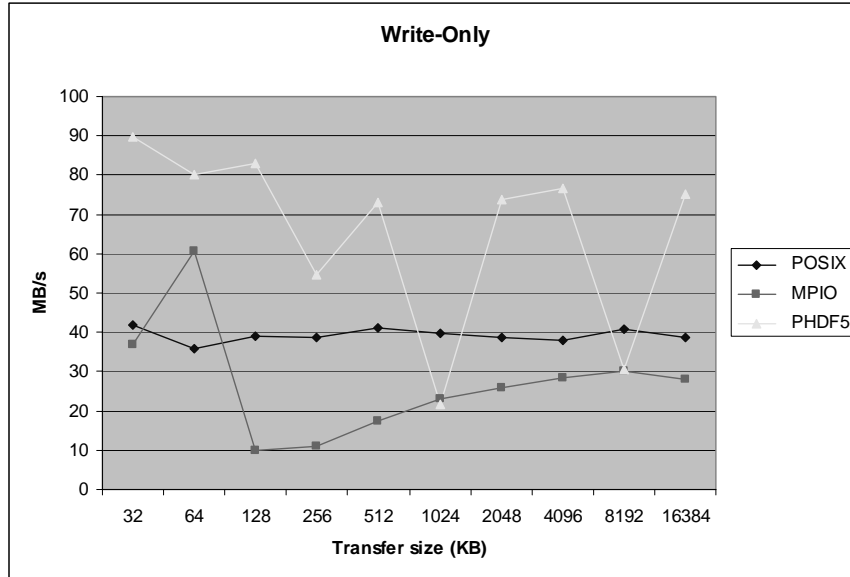


Figure 3 Throughput in non-collective interleaved mode (`File_Set_View` in MPIO)

We ran this test several times to reproduce the behavior shown in the Figure 3. While POSIX and MPIO remained approximately the same, the performance of PHDF5 appeared to be unstable.

Tests using the MPI-POSIX driver

We also carried out tests using the MPI-POSIX driver for PHDF5. Since in this case PHDF5 does not use MPI, no collective actions are executed. Its performance is very similar to the one obtained by POSIX as shown in Figure 4.

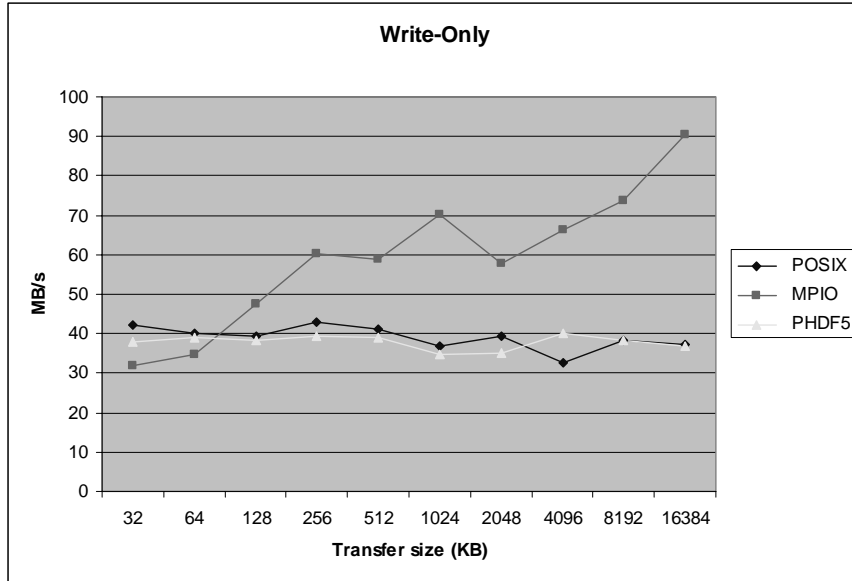


Figure 4 Throughput in collective interleaved mode (MPI-POSIX driver)

As expected, Figure 5 shows that performance measurements are almost the same for POSIX and PHDF5 in non-collective mode.

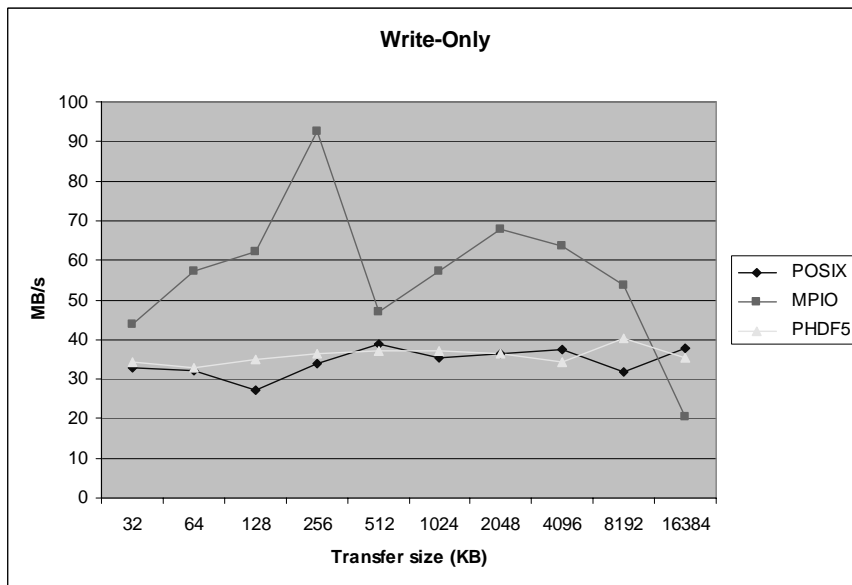


Figure 5 Throughput in non-collective interleaved mode (MPI-POSIX driver)

For the executed tests, we see that the performance of MPI (and PHDF5 when using the MPI driver) is unstable.