Using RAM Area Network to Reduce Synchronization Costs in Collective I/O Operations

Sergio Servantez†, Richard J Zamora◆, François Tessier*, Zhiling Lan†

†Illinois Institute of Technology, ◆Argonne National Laboratory, *Swiss National Supercomputing Centre

Overview

Collective operations improve I/O performance by merging the requests of ranks in a parallel application. This research explores using a RAM area network as an aggregation layer in performing collective I/O operations. A RAM area network (RAN) expands the conventional memory hierarchy by adding a remote level. This disaggregated architecture offers new data transfer possibilities by providing all compute nodes with a globally addressable memory space.

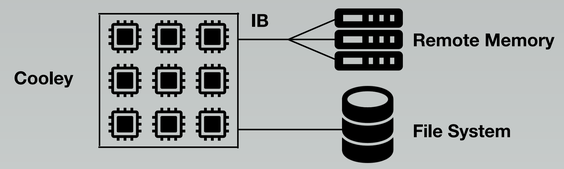

Figure 1. Illustration of Cooley architecture. Cooley's 126 compute nodes are connected to remote memory via an Infiniband network.

Two-Phase Collective I/O

Collective I/O operations typically are performed in two phases. Using RAN instead of DRAM for data aggregation implies a third phase to return data from remote memory. Data is moved over several iterations called rounds. Each rank must coordinate at every round which incurs a synchronization cost.

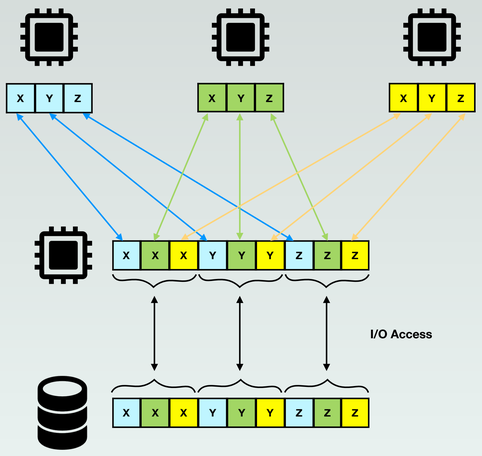

Figure 2. A diagram of two-phase collective I/O. In Phase 1 (top to middle) contiguous segments of the file are organized into an aggregator. In Phase 2 (middle to bottom) these segments are written to the file system.

Preliminary Data

As the size of contiguous chunks decrease, more write operations are required to organize segments of the file. This increase in write frequency translates into a slight increase in transfer time. The overall performance for each chuck size is significantly diminished by synchronization costs.

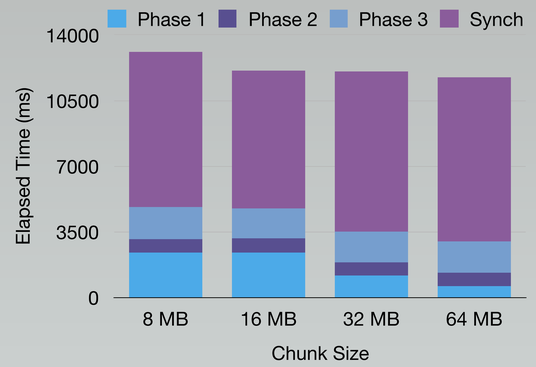

Figure 3. Performance of collective write using remote memory with varying chunk size.

The MPI implementation of collective write outperformed the three-phase counterpart where chunk size was 16 MB or greater. However, the three-phase write operation experienced a slower decline in performance as chunk size decreased. The yellow line in Figure 4 represents the upper bound on performance if synchronization costs were eliminated from the three-phase implementation. A significant reduction or elimination of these synchronization costs would translate into a drastic increase in performance.

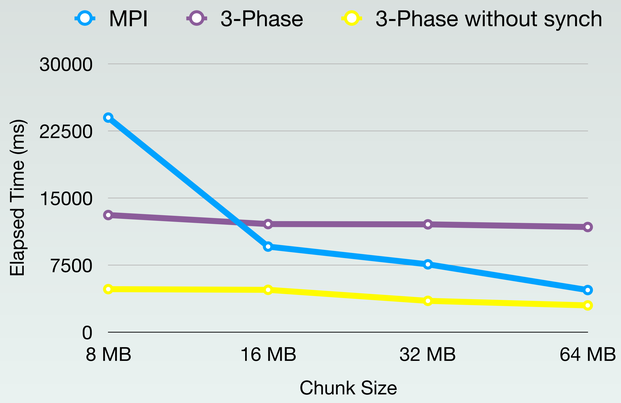

Figure 4. Performance of different collective write operations with varying chunk size.

New Three-Phase Algorithm

Here we present a new algorithm specifically designed for a disaggregated architecture which minimizes synchronization costs among ranks. The ranks must coordinate with each other once to signal that all local data has been transferred to remote memory. Once this barrier has been coordinated, each aggregator is free to collect its data independent of other aggregators.

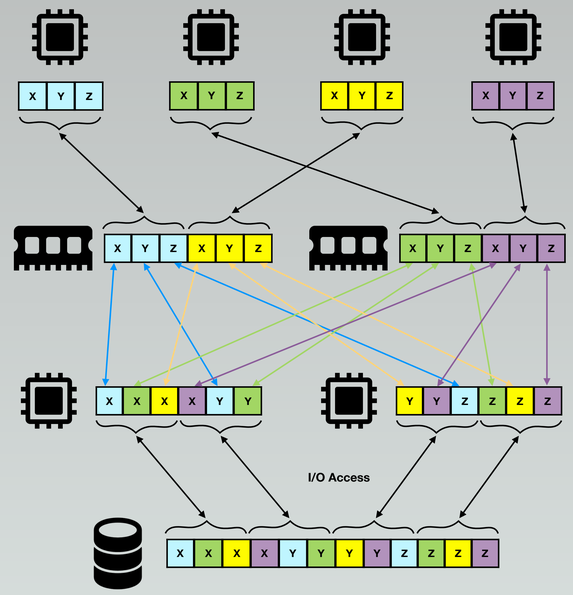

Figure 5. A diagram of three-phase collective I/O. In Phase 1 (top to upper-middle) local data is pooled into remote memory. In Phase 2 (upper-middle to lower-middle) aggregators organize contiguous segments of the file by reading from remote memory. In Phase 3 (lower-middle to bottom) these segments are written to the file system.

Ongoing Work

We currently have a working implementation of the three-phase algorithm described above. We have also recently completed a RAN benchmark to better understand read/write behavior across remote memory. The focus of our research is now on optimizing the performance of this implementation as it scales to many nodes.

This work is supported in part by US National Science Foundation grants CCF-1422009 and the U.S. Department of Energy, Office of Science.