Programming of Supercomputers

Assignment 4: MPI Collectives and MPI-IO

Prof. Michael Gerndt

Isaias A. Compres Urena

Assignment 3 Discussions

- Defining a baseline
  - Variability in measurements?
  - Problem size and process count scalability differences?
  - Haswell vs. Sandy Bridge?
- Point-to-point and One-Sided communication performance
  - Maximum theoretical overlap? Achievable in practice?
  - Which provided the best overall solution?

MPI Collectives

- Communication that involves a group of processes
  - All processes of a communicator participate simultaneously
  - Operations on a shared piece or set of information
  - These substitute complex sequences of point-to-point calls
  - Additional optimization opportunities for MPI implementers
  - There is a ‘root’ process in many cases
- Large and robust set of collectives provided by MPI
  - Main advantage of MPI versus other message-passing libraries
  - MPI-3 introduced non-blocking versions
- There are 3 main types of collective operations:
  1. Synchronization
  2. Data movement
  3. Global computation

Parallel IO with MPI-IO

- Provides a parallel file system neutral API
- Typical approach for IO are far from ideal:
  1. Root process manages IO and distributes data, or
  2. Each process creates a separate file for its partition, or
  3. A proprietary parallel IO API and implementation is used
- IO scalability issues, complexity and portability are common
- MPI-IO provides operations for:
  - Synchronization
  - Data movement
  - Data layout definitions
    - With MPI Datatypes
- The programmer can rely MPI-IO and let the implementation streamline and optimize file accesses in a portable manner
  - Open MPI and MPICH rely on the ROMIO implementation

Assignment 4 Motivation and Hints

- MPI Collectives
  - Users implementing collectives themselves is a common pitfall
    - Extra validation required on their custom algorithm
    - Users tend to implement naïve algorithms
  - MPI implementations are well researched and tested
    - Use hardware acceleration when possible
    - Minimize communication and algorithm iterations
- Parallel IO with MPI-IO
  - IO is a mayor bottleneck in many domains
    - Data Analytics (“Big Data”, data mining, statistics, etc.)
  - Large scale simulations in scientific computing
    - Terabytes of input and output data
    - Checkpoint and restart policies
  - MPI-IO takes advantage of fast network performance to streamline IO operations
  - MPI-IO abstracts the parallel file system (GPFS in SuperMUC).

Questions and Answers