TMPIFile: A New Parallel I/O Solution in ROOT

Yunsong Wang[1], Amit Bashyal[2], Taylor Childers[3],
Peter Van Gemmeren[3], Vakho Tsulaia[1], Philippe Canal [4]
[1] Lawrence Berkeley National Laboratory
[2] Oregon State University
[3] Argonne National Laboratory
[4] Fermi National Accelerator Laboratory

Overview

Communication among processes is generating considerable interest in the scientific computing community due to the increasing use of distributed memory systems. In the field of high energy physics (HEP), however, little research has been addressed on this topic. More precisely in ROOT I/O [1], the de facto standard for data persistence in HEP applications, no such feature is provided. In order to perform efficient and robust inter-process communications, we introduce TMPIFile into ROOT where Message Passing Interface (MPI) is used to pass data across the entire distributed system.

Message Passing Interface

- MPI:
  - A specification of message passing libraries
  - Communications among processes with separate address spaces
  - Explicit parallelism using distributed memory model
- Reasons for Using MPI [2]:
  - Standardization - The only standard that widely supported on all HPC platforms
  - Portability - No code modification required for a different platform that supports the MPI standard
  - Performance Opportunities - Vendor implementations will exploit native hardware features to optimize performance
  - Functionality – Rich set of features
  - Availability - Both vendor and public domain are available:
    - Open-source implementations: MPICH and OpenMPI
    - Vendors: Intel MPI, IBM Blue Gene MPI, Cray MPI, etc.

Why TMPIFile

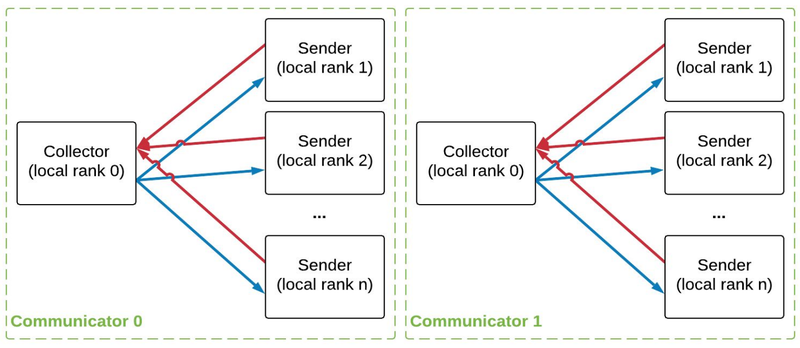
- Uniformed I/O solution: Due to the lack of cross-process communication in ROOT, HEP simulations tend to create solutions individually for each use case which results in unnecessary duplicates
- Consistent data handling: Extra procedures beyond ROOT are often required to post-process data which negatively affect the I/O consistency and introduce undesired difficulties for code debugging and maintenance

ROOT::TMPIFile at a Glance

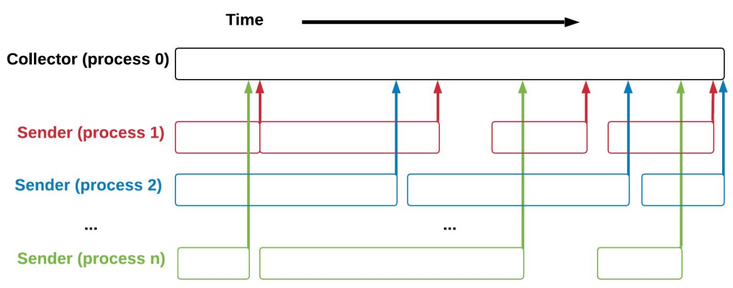
- Class derived from ROOT::TMemFile
- Use MPI to perform cross-process communications
- Write data into disk in TFile as output
- Minimum work unit: one Sender and one Collector
- Each Collector or Sender has its own unique MPI rank (address space)

- Collector:
  - Always the rank 0
  - Receive buffers from workers
  - Reopen/Create the output file from disk
  - Merge buffers into the output and write it into disk
- Sender:
  - Rank 1 or plus
  - Prepare a buffer for input data (events filled with TTrees/TH1Ds)
  - Send the buffer to its Collector via MPI functionalities

Features

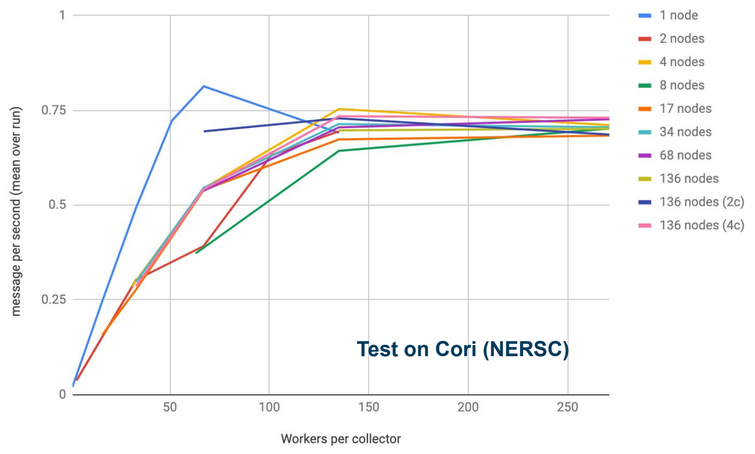


- Multi-communicator to improve merge concurrency:
  - Number of Collectors set by users
  - Data collections are local for each communicator group
  - Multiple Collectors (local rank 0) allow concurrent merging



- Non-blocking communications with MPI_Isend() and MPI_Irecv() to hide data transfer overhead
- Dynamic receiving with MPI_Probe on a first-come, first-served basis
- Use TMemFile::GetEND() to retrieve the minimal buffer size
- Consistent file compression level for both Senders and Collectors:
  - Constantly activate "fast merge" process on the Collector side
  - Significantly reduce the merge time from ~ 200 s to ~ 1 ms
- Auto data collection (send, receive, merge and write) when closing TMPIFile

Performance



- System setup:
  - Theta: 4392 nodes of KNL (64-core); peak of 11.7 PFlops; Cray Aries with Dragonfly.
  - Cori KNL: 9668 nodes of KNL 7250 (68-core); peak of 29.5 PFlops; Cray Aries with Dragonfly.
- Simulation-like test case:
  - ~ 2 MB per event
  - 10 events per sync/message
  - Sleep ~ 5 s between two sends to fake real-case workload
- Peak performance: about 1 message per second per collector which includes receiving the message and merging it with the final output
- Better & more stable performance on Theta due to smaller cluster size and less I/O-intensive workloads
- Scale up to ~ 130 Senders per Collector

Reference

1. Canal, P., Bockelman, B., & Brun, R. (2011). ROOT I/O: The fast and furious. In Journal of Physics: Conference Series (Vol. 331, No. 4, p. 042005). IOP Publishing.
2. Forum MPI. (2012). MPI: A Message Passing Interface Standard Version 3.0.