

I/O Operation in Phylanx

Alireza Kheirkhahan

Current Status

- Potentially large files as input or output
- Various file format is needed based on project
- Beside binary format, CSV and HDF5 formats are implemented
- Format chosen based on current need.

Current Status

- Comma-Separated Values (CSV)
 - Implemented by using *boost.spirit*
 - Could be used as basis for other text format such as *Matrix Market ...*

- HDF5
 - Developer dropped support for C++ API
 - The current API is not thread safe
 - We used HighFive, a modern C++/C++11 friendly interface
 - Adopted HighFive to support blaze matrix

Other Formats

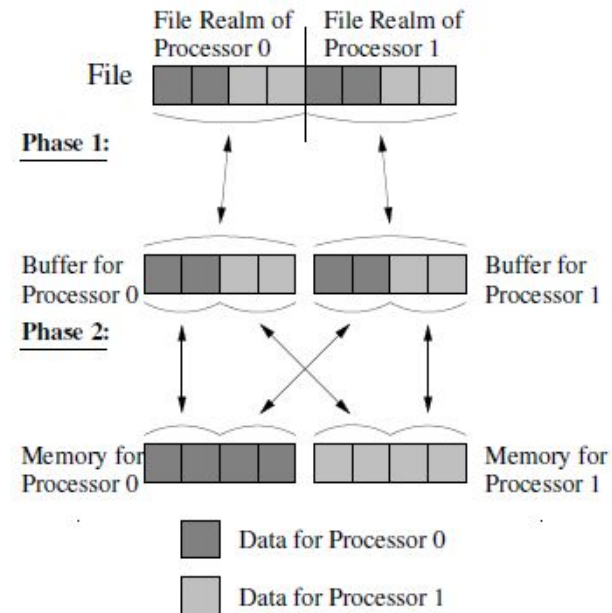
- NumPy, has both text and binary format.
- ARFF (Attribute-Relation File Format), used by WEKA
- Pandas Dataframe
- TensorFlow Dataset

Distributed I/O

- We need a way to describe a shared file in a distributed environment
- We need a uniform means to access a shared file in a distributed system
- We need a mechanism to distribute the I/O load between localities
- We need a mechanism to transfer I/O data between localities

Collective I/O

- Each locality has non-interleaving file realm
- Each locality take responsibility of it's portion and access directly to underlying file system
- For other portions, each locality communicate to the responsible locality rather than accessing the file system directly
- Data Sieving
- Request aggregation by two-phase I/O



Collective I/O Implementation

- Defined in MPI-IO
- Multiple implementation for MPI-IO
- None of the Implementation are non-blocking
- All communication between processes done by inter-process messages
- Synchronization on each step
- Not implemented in a task-based runtime system such as HPX

Thank you!