UCRL-PRES-145541

HPSS MPI-IO: A Standard Parallel Interface to HPSS File Systems

October 11, 2001

Bill Loewe

wel@llnl.gov

(925) 422-5587

This work was performed under the auspices of the
U.S. Department of Energy
by the University of California
Lawrence Livermore National Laboratory
under contract No. W-405 Eng-48.

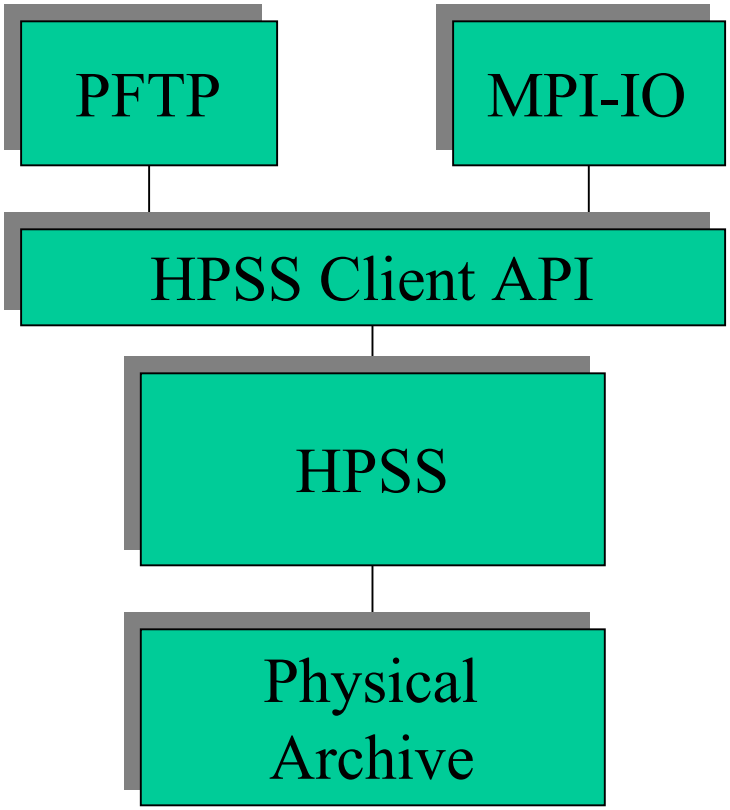
Outline

- Introduction to HPSS MPI-IO
  - History
  - Alternative Interfaces
  - Description
- Functionality
  - MPI Datatype Abstractions
  - Parallelism with Collective Operations
  - Nonblocking Accesses
- Design
  - Interfacing to HPSS
    - Steps for Opening Files
    - Steps for Collective Read/Write
- Summary


Introduction

- History of HPSS MPI-IO
  - HPSS MPI-IO is the MPI-IO implementation for use with HPSS.
  - Work was started in 1995 at LLNL
  - High-level user interface for the HPSS file system
  - Formally a subsystem of HPSS
- Alternative to existing Interfaces
  - Offers an alternative interface to HPSS Client API library
  - FTP, PFTP, HSI, etc.

MPI-IO Interface to HPSS



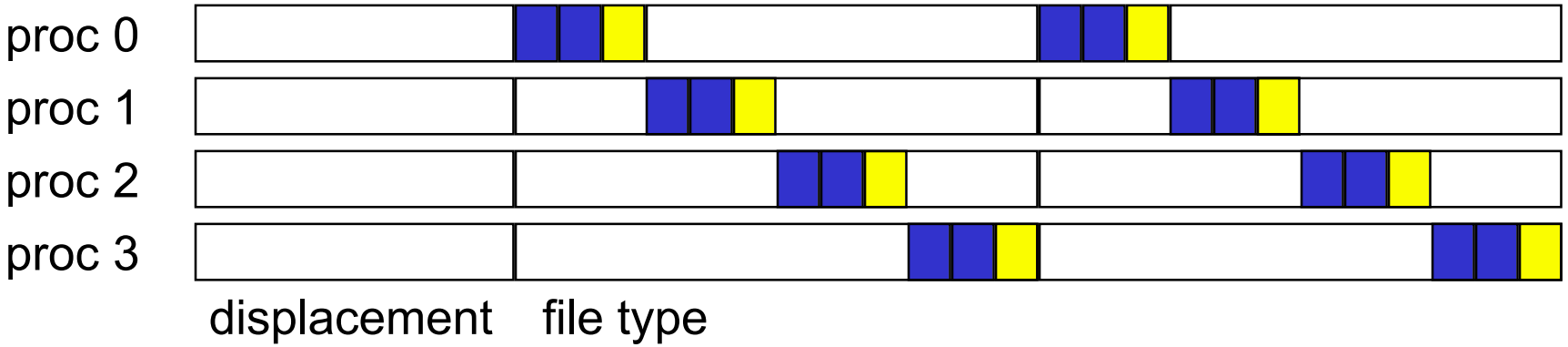
Description of HPSS MPI-IO

- Coordinates access to HPSS files from multiple processes.
- Provides nonblocking accesses to HPSS files.
- Offers the functionality of MPI-IO to HPSS users.

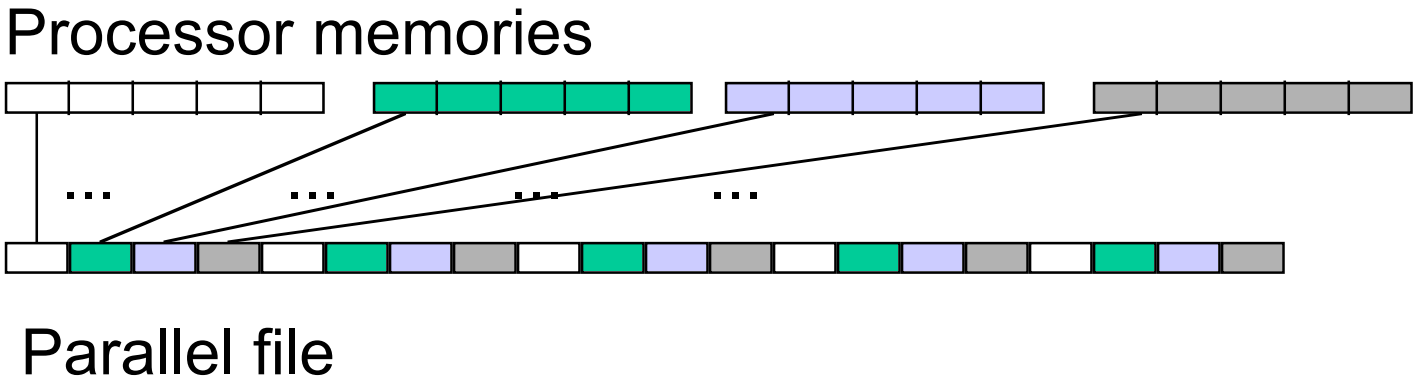

MPI Data Types

- Etype – Unit of data access and positioning used within a file.
- Buftype – Describes the layout within the program of the data to be written.
- Filetype – Basis for partitioning a file among processes and defines a template for accessing the file.
- Fileview – Set of data visible and accessible for each process.

Data Access Using MPI Data Types



Discontiguous access



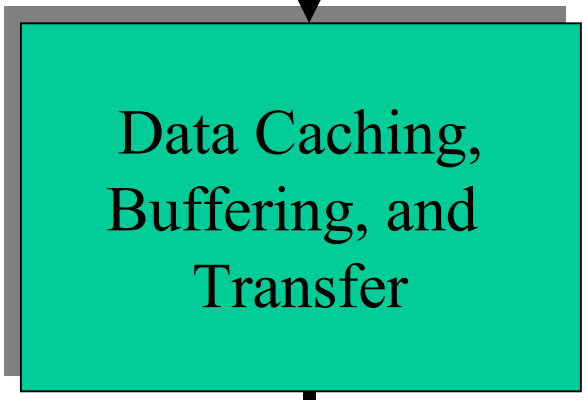

Functionality

- HPSS MPI-IO is a complete implementation of the MPI-IO portion of the MPI-2 standard.
- Includes implementations of other portions of MPI-2
- Data Placement using Patterns of MPI Datatypes
  - MPI Datatype abstraction paradigm
  - Patterns facilitate noncontiguous access to memory and file

- Coordinated Access
  - Coordinates and simplifies parallel access to HPSS files from multiple processes
  - Collective Operations (reads/writes/opens) executed by all nodes
  - Provides functionality to coalesce multiple small data accesses into single large access
  - Shared file pointers
- Thread-based support for Nonblocking Operations
  - Enables nonblocking accesses
  - Allows overlapping of I/O with computations
  - Can be split collective or noncollective

Data Movement

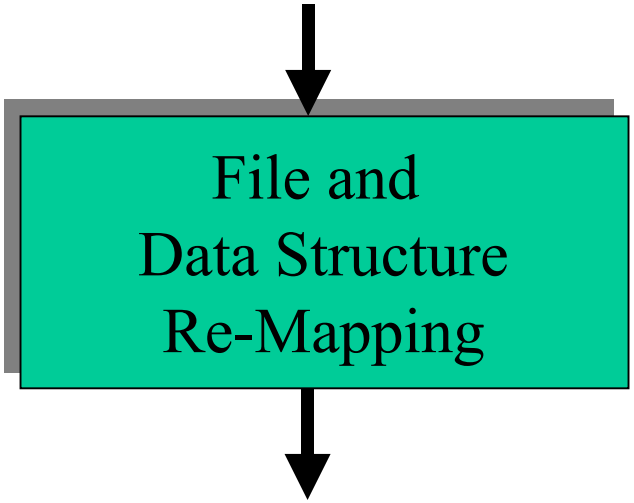
MPI_File_read_...
MPI_File_write_...



HPSS Client API
Commands

Data Structure Interface

MPI Datatypes
MPI File Views



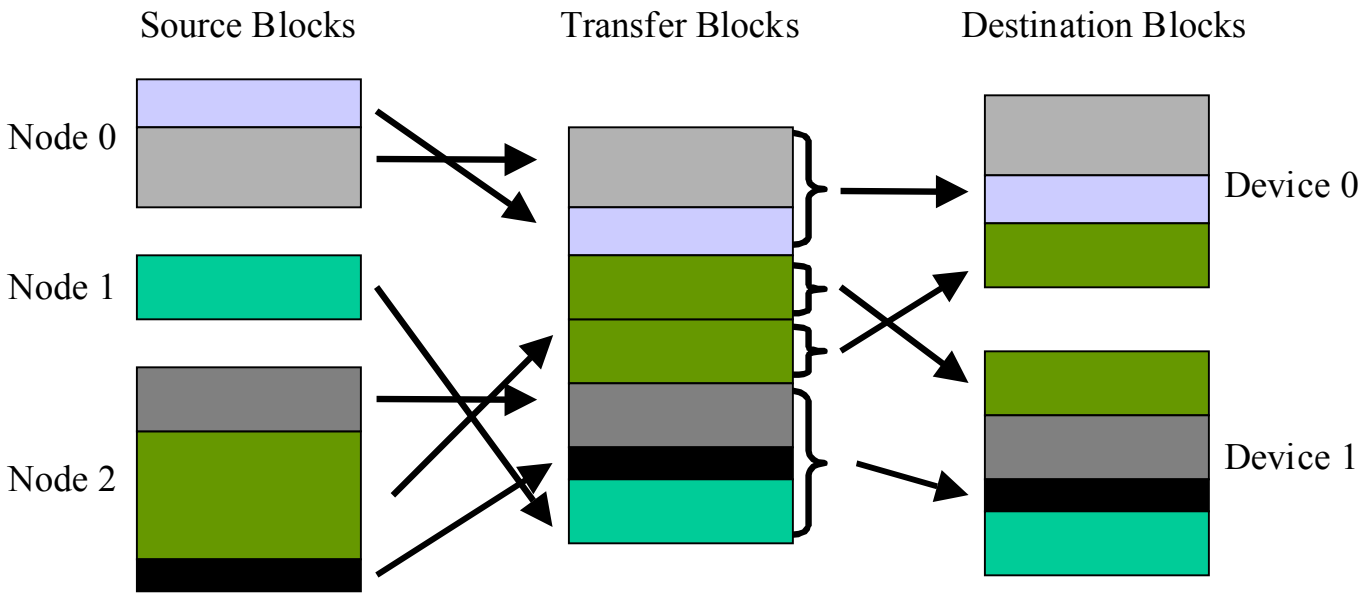
HPSS IOD
(Input/Output Descriptor)


I/O Descriptors

- Data structure that determines the details of a data transfer from the application side to the HPSS side.
- Treats this transfer as a mapping from the source to the destination.
- Created using MPI Datatypes

Simple Transfer

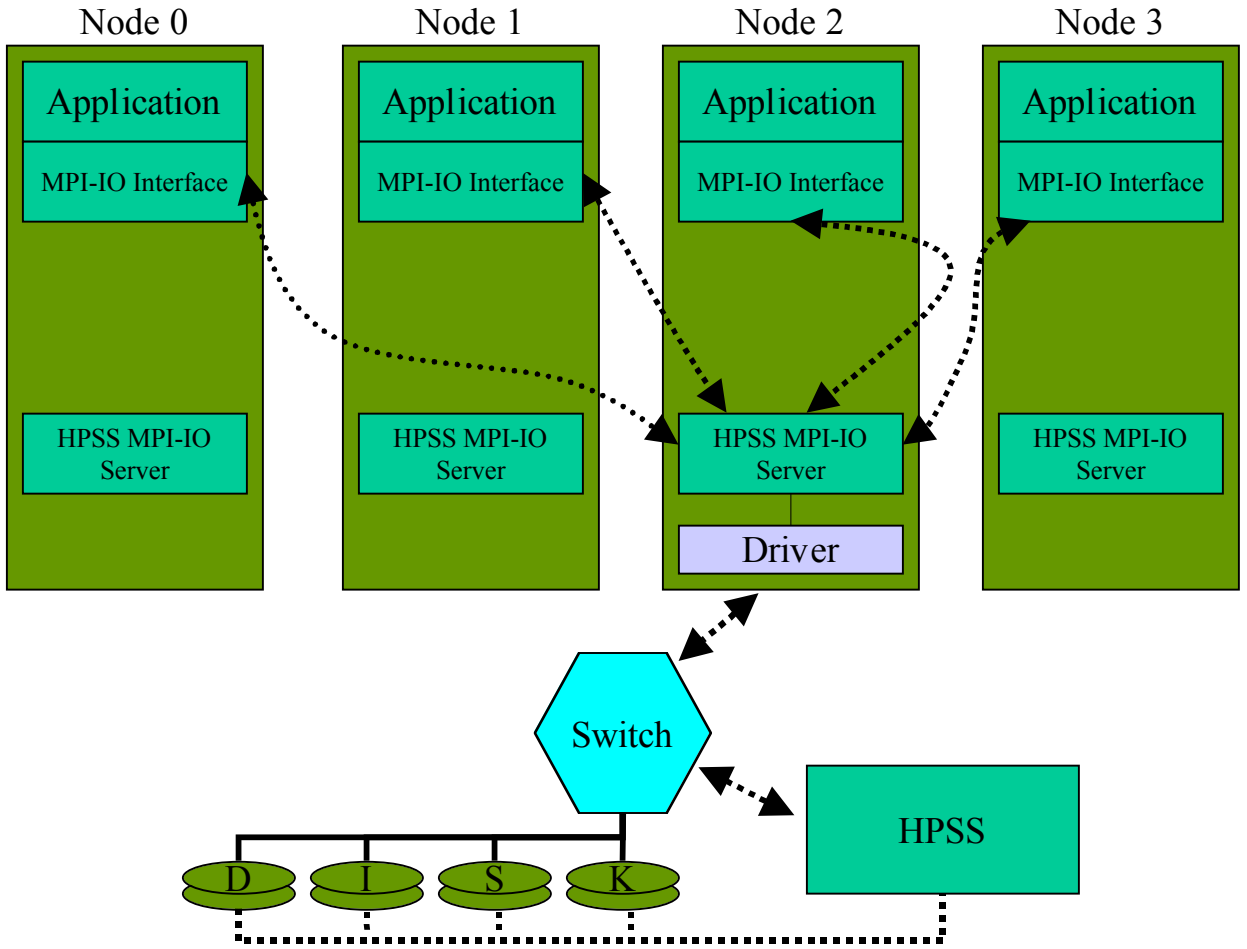
Distributed Transfer



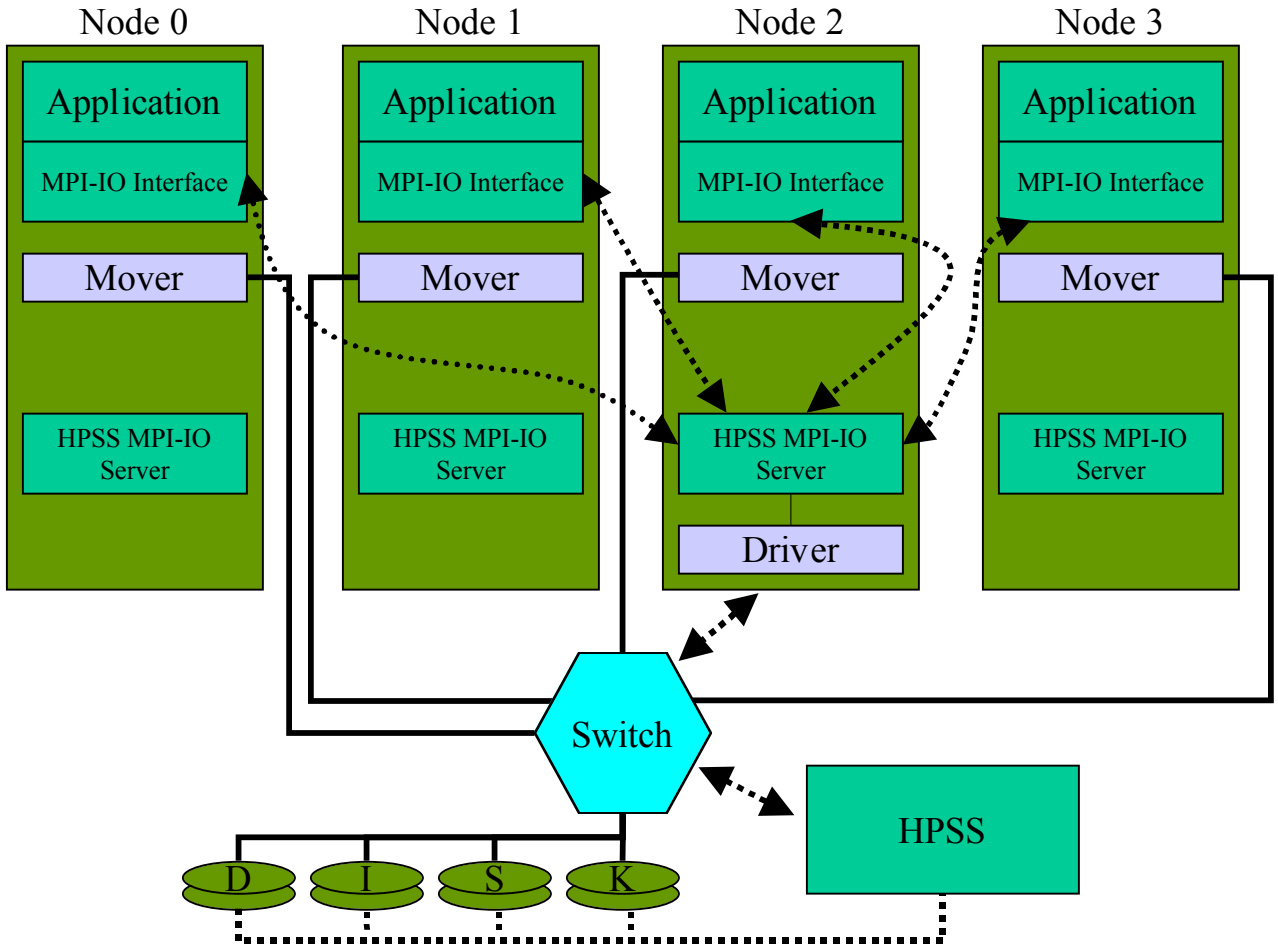

Steps to File Creation:

```
MPI_Init();
    MPI_File_open();
        MPI_File_write_all();
    MPI_File_close();
MPI_Finalize();
```

Architecture



Architecture



Summary

- Alternative interface to HPSS using MPI-IO
- Complete implementation of MPI-IO from the MPI-2 standard
- Allows parallelism in collective operations
- Distributes server load among processors
- Centralized control with parallel data transfers
- Allows non-blocking calls

Additional Information

– Linda Stanberry <lstanberry@llnl.gov>
– Bill Loewe <wel@llnl.gov>