MPI

Message Passing Interface

Jing Liu

TDB & LMB, Uppsala University

Programming of Parallel Computers , Feb 2016

UPPSALA UNIVERSITET

Dep. of Information Technology

Previous …

- Communicators
- Send and Receive: blocking/non-blocking
- Other Point-to-Point Functions
- Global Functions
- Datatypes
- Topology
- Timing

UPPSALA UNIVERSITET

Dep. of Information Technology

Outline

- One-sided communication
- Dynamic process creation
- Parallel I/O
- Example

jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

One-sided communication

- What is two-sided communication?
  - The data movement has to be specified on both sides.
  - P2P functions: Send/Recv
  - Global functions: Bcast, Scatter, …
- One-sided?
  - A process access another process address space without any explicit participation in that communication operation by the remote process.

UPPSALA UNIVERSITET

Dep. of Information Technology

One-sided communication

- Advantage:
  - Direct remote memory access (RMA)
  - No hand-shaking
  - Flexible and dynamic data distribution
  - No extra buffer
- How?
  - Create accessing window
  - Define the pattern
  - Sync (fence)
  - Reduction
  - Lighter syncs

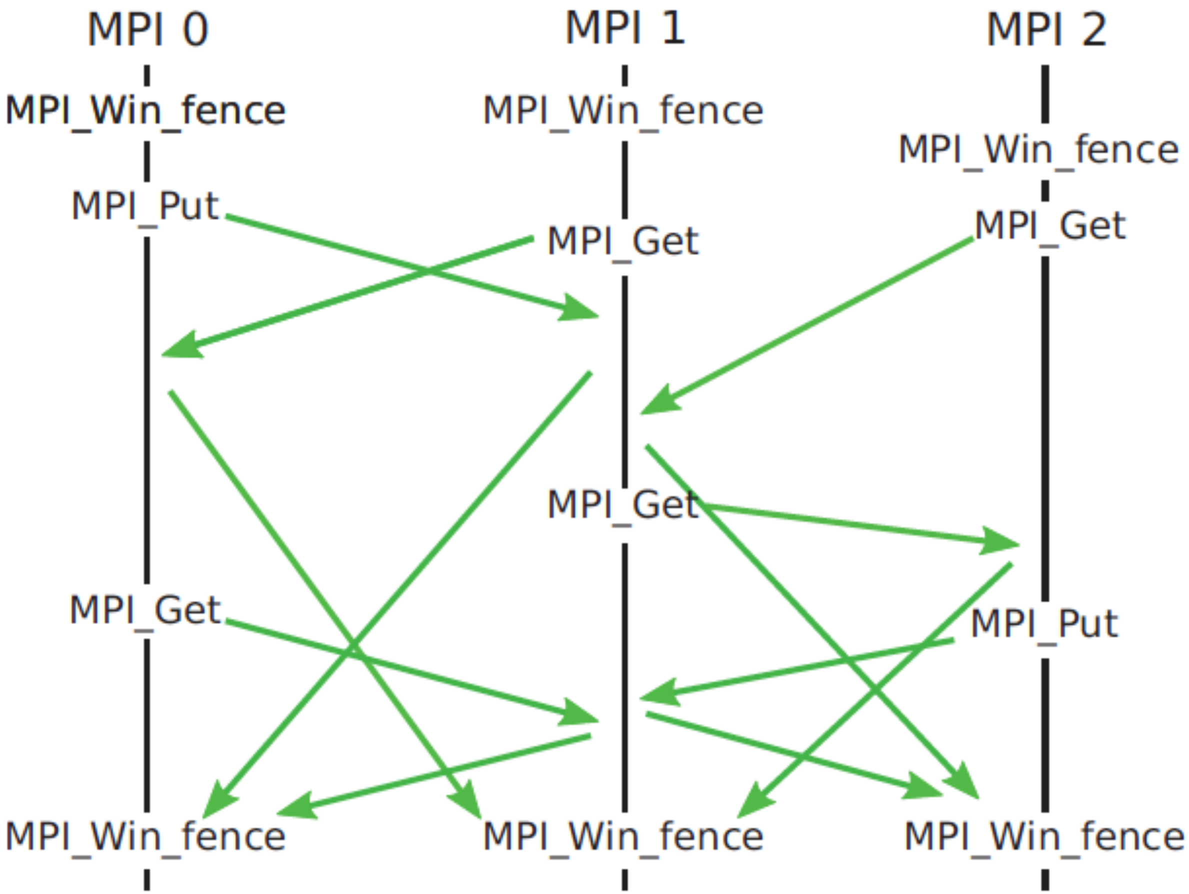
jing.liu@it.uu.se/@icm.uu.se

One-sided communication

- Synchronization
  - MPI_Win_fence
    - Similar to MPI_Barrier call
    - All communication calls should be inside a par of MPI_Win_fence
  - Post-Start-Complete-Wait
    - Synchronize within a group
  - Lock-Unlock
    - like mutexes or other concurrent operations

jing.liu@it.uu.se/@icm.uu.se

MPI_Win_fence



Demo:
Matrix
transpose
and
accumulate

Demo_fence

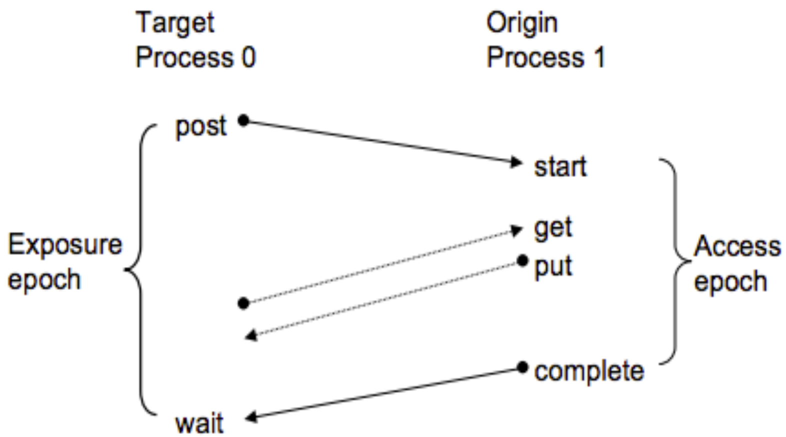
jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

Post-Start-Complete-Wait

- Work within a group (local and lighter than MPI_Fence)
  - Origin:
    - MPI_Win_start
    - MPI_Win_complete
  - Target:
    - MPI_Win_post
    - MPI_Win_wait
- Demo: Demo_put



jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

Outline

- One-sided communication
- Dynamic process creation
- Parallel I/O
- Example

jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

Dynamic Process Creation

- A MPI process can spawn new MPI processes at run time which starts running a new program.
  - MPI_Comm_spawn(...);
  - MPI_Comm_spawn_multiple(...)

Demo

jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

Outline

- One-sided communication
- Dynamic process creation
- Parallel I/O
- Example

jing.liu@it.uu.se/@icm.uu.se

UPPSALA UNIVERSITET

Dep. of Information Technology

Parallel I/O

The I/O Challenge:

- Problems are increasingly computationally challenging
  - Large parallel machines needed to perform calculations
  - Critical to leverage parallelism in all phases
- Data access is a huge challenge
  - Using parallelism to obtain performance
  - Finding usable, efficient, portable interfaces
  - Understanding and tuning I/O
- Data stored in a single simulation for some projects: – O(100) TB !!

Parallel I/O

- I/O approach
  - Gather all data to one process
  - Each process write to one file
  - All process write to one file
- MPI-I/O: the Basics
  - MPI-IO provides a low-level
  - Simply compile and link as you in any normal MPI program.

jing.liu@it.uu.se/@icm.uu.se

- MPI-IO can be done in 2 basic ways :
  - Independent – For independent I/O each MPI task is handling the I/O independently using non- collective calls like MPI_File_write() and MPI_File_read().
  - Collective – When doing collective I/O all MPI tasks participating in I/O has to call the same routines. Basic routines are MPI_File_write_all() and MPI_File_read_all()

Demo

Outline

- One-sided communication
- Dynamic process creation
- Parallel I/O
- Example

jing.liu@it.uu.se/@icm.uu.se

Case study

- The number of primes
  - A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.
  - n % j != 0, j = [2,3,…,n-1]
  - MPI_Bcast
  - MPI_Reduce

Demo

jing.liu@it.uu.se/@icm.uu.se