

CS3210

Parallel Computing

Changes from Monday in Green



Lab 5

Mon (4pm)

Tues (2pm)

Admin Updates

- **Lab 4 submission due today 11.59pm**
 - For ex6, just pick a data distribution you **expect** to do as well or better than the current distribution
- Assignment 1 part 1 grades released on LumiNUS
 - Weightage: 8% (grade reported in LumiNUS is out of 16)
 - Comments and grade breakdown in grading remarks
 - Bonus (for AVX) awarded separately
 - **If you have questions about the grading, write in to Prof via email ASAP - no changes accepted after 29 Oct (Thursday)**

Admin

Roadmap

- Please avoid using machines assigned to other lab pairs
 - You can use any FFA machines with the given MPI programs
 - Hint: take a look at **ompi-top**
- Today's lab
 - Part 1: Collective Communication
 - Part 2: Managing Communicators
 - Part 3: Cartesian Virtual Topology
- No lab submission this week - just explore the programs

Part 1

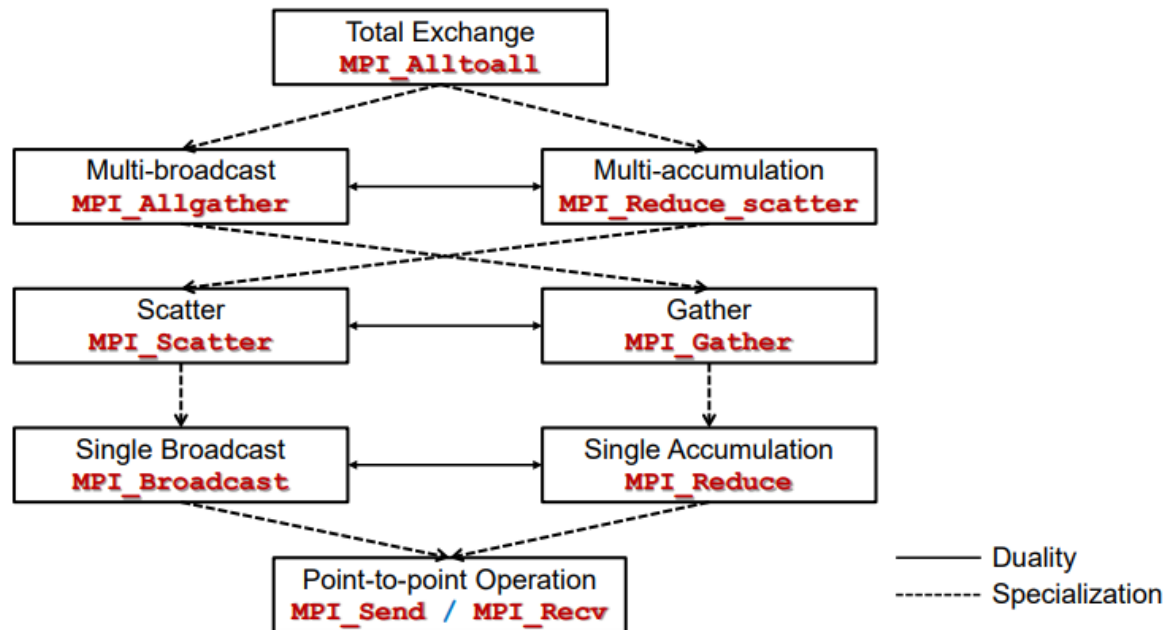
Collective Communication

- Three main types
 - Synchronisation operations
 - Data movement (distribution) operations
 - Collective computation (data movement with reduction)
- Synchronisation operations: only barrier
 - Blocking variant: **MPI_Barrier(MPI_Comm Comm)**
 - Non-blocking variant: **MPI_Ibarrier(MPI_Comm comm, MPI_Request *request)**
 - Non-blocking variant enforces barrier semantics at following completion call (**MPI_Test** or **MPI_Wait**) with **request**

Part 1

Collective Communication

- Data movement and collective computation operations
 - Collective computation: data movement with reduction (with a binary, associative and commutative operation)



Part 2

Managing Communicators

- An **MPI_Group** denotes a set of processes
 - All processes in a group have an associated group rank
 - Can operate on groups with set operations: union, intersection, difference, inclusive/exclusive ranges (by rank)
- An **MPI_Comm** comprises an **MPI_Group** with an associated context
 - All processes in a communicator have an associated rank
 - Create a new communicator with **MPI_Comm_create** with an **MPI_Group**; duplicate with **MPI_Comm_dup**
 - Compare two communicators with **MPI_Comm_compare**

Part 3

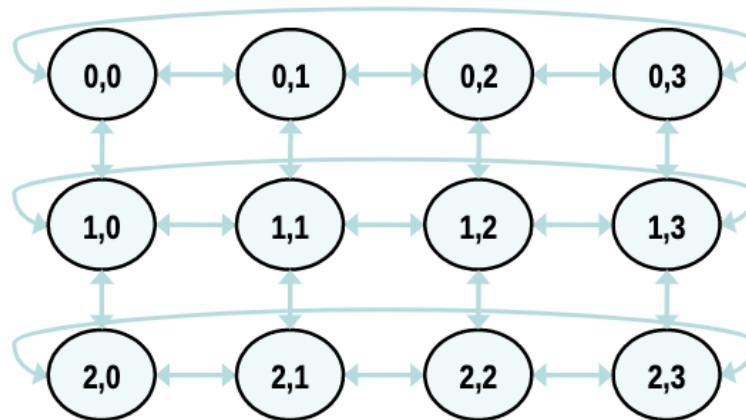
Virtual Topologies

- A virtual topology is some programmer-defined mapping of MPI processes to a geometric space
 - As *virtual* implies, there is no relation between the virtual topology and the underlying hardware organisation (layout)
 - Purpose: facilitate structured data access and message-passing communication for tasks exhibiting this topology
- Explicitly handled by programmer
 - OpenMPI supports Cartesian and graph virtual topologies
 - Today: explore only n -dimensional Cartesian topologies

Part 3

Cartesian Virtual Topology

- Cartesian virtual topology
 - Create with `MPI_Cart_create(MPI_Comm comm_old, int nDims, const int dims[], const int periods[], int reorder, MPI_Comm *comm_cart)`
 - `comm_cart` is result communicator with topology information



Part 3

Cartesian Virtual Topology

- Cartesian virtual topology information
 - Process in Cartesian topology of some rank can retrieve its Cartesian coordinates with **MPI_Cart_coords**
- What do you think **MPI_Cart_shift** does?
 - Difference between periodic (n -dimensional toroid) vs non-periodic Cartesian topology (n -dimensional Cartesian mesh)
 - **If periodic, returns ranks of source and destination processes when shifting along a specified dimension by some amount**
 - If not periodic, shift may exceed bounds along a dimension and return **MPI_PROC_NULL**

CS3210

Parallel Computing

Thank you! Any questions?



Lab 5

Mon (4pm)

Tues (2pm)

bit.ly/cs3210-t01-qn