

Programming of Supercomputers

Assignment 3:

MPI Point-to-Point and One-Sided Communication

Prof. Michael Gerndt

Isaias A. Compres Urena

Schedule Updates

- **Assignment 2:**
 - Discussion today
- **Assignment 3 and 4:**
 - Core programming and optimization assignments
 - Deadlines: 22 of December, and 27 of January
 - Need to discuss our sessions
 - Both will be introduced today

Assignment 2 Discussions

- Bugs in parallel programming
 - Were you familiar with them?
 - What did you learn about floating point arithmetic?
 - Are all of them parallel programming specific?
- What do you think about TotalView?
 - Is it worth learning?
 - Good OpenMP support?
 - Good MPI support?
 - Did you explore more than what was in the assignment?
- Alternative: Allinea DDT
 - Also in SuperMUC if you want to try it

MPI Point-to-Point Communication

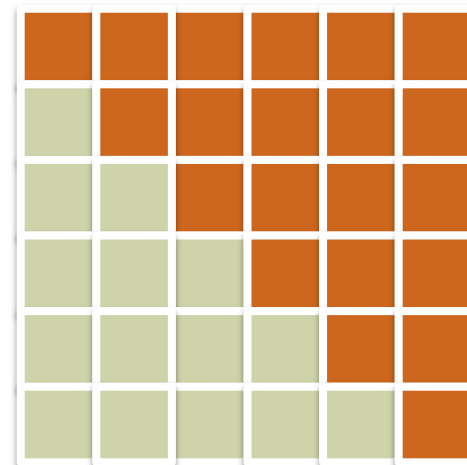
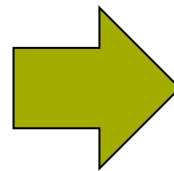
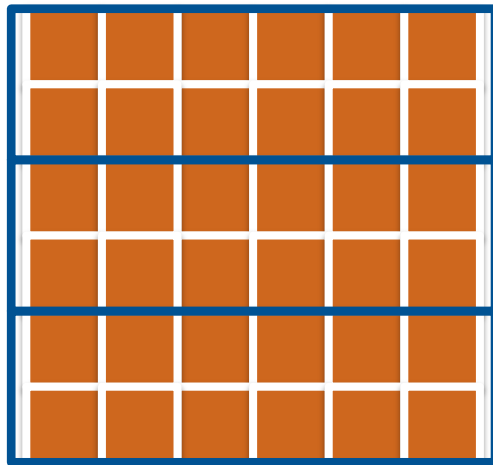
- Covered in the Parallel Programming lecture
- Large set of operations in the standard
 - Send, receive and combined operations
 - Receives matching sends
 - Blocking and non-blocking
 - Wait and probe
 - Buffered, ready, synchronous
- Part 1 of the assignment
 - Focus on blocking and non-blocking communication
 - Convert blocking to non-blocking communication
 - Try to achieve overlap
 - Perfect overlap is the target
 - Theoretical double performance

MPI One-Sided Communication

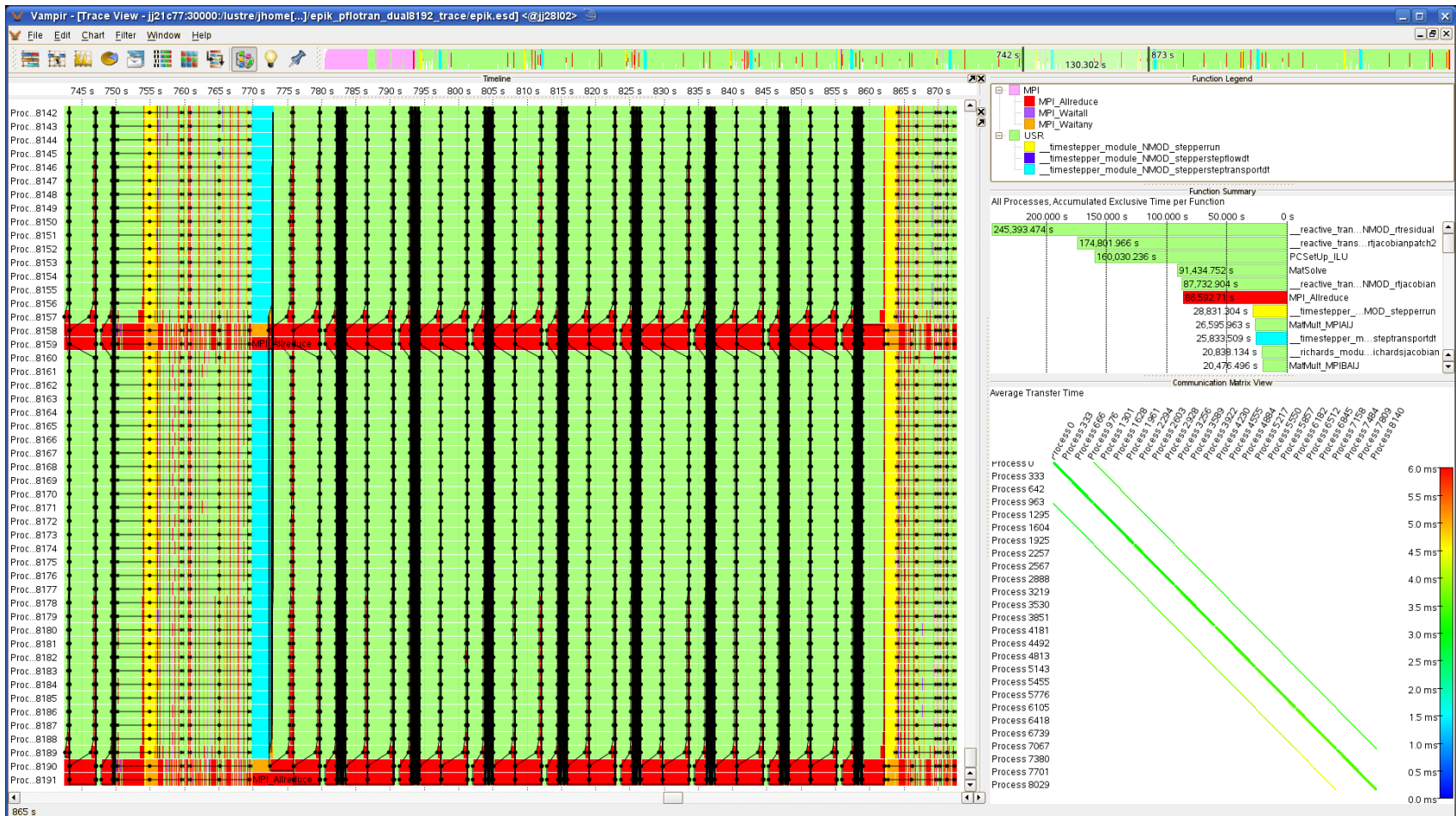
- Also covered in the Parallel Programming lecture
- Remote Memory Access (RMA)
- Operations are non-blocking
 - Remote process not blocked during transfer
 - Possible overlap with theoretical double performance
- No need for matching operations at the receiver
- Also a large set of operations in the standard
 - Put and get operations
 - Direct access to memory of other processes through windows
- Multiple benefits vs. point to point communication
 - Aim to reduce synchronization with bulk transfers and no direct matching
 - Aim to reduce data movement by eliminating intermediate buffering
 - Can simplify programming since only one side of the communication is specified
 - Can better benefit for RMA hardware features in some NICs

Gaussian Elimination with MPI

- Distributed Gaussian Elimination
 - Implementation given
 - MPI blocking point-to-point
- Tasks:
 - Convert to non-blocking
 - Convert to one-sided
- Scale of the programming task
 - 4 nodes
 - 64 processes
 - Use Intel MPI and test at multiple process counts in a single job



Recommended: Vampir Trace or Intel Trace Analyzer



Questions so far?

