ECE 574 – Cluster Computing
Lecture 14

Vince Weaver

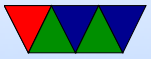
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

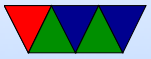
22 October 2015

Announcements

- Homework #3 and #4 Grades out soon
- How is HW#5 going?
- HW#6 will be posted next week (MPI)
  Need to get pi-cluster going



Midterm Review



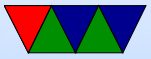
MPI continued

Some references

https://computing.llnl.gov/tutorials/mpi/

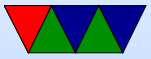
http://moss.csc.ncsu.edu/~mueller/cluster/mpi.guide.pdf

https://cvw.cac.cornell.edu/MPIcc/default



Efficient way of getting data to all processes

- master send to each individual, take a while
- some sort of tree, 0 to 1 and 2, 1 sends to 3 and 4, etc.
- use broadcast instead

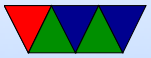


Collective Communication

- All must participate or there can be problems.
- Do not take tag arguments
- Can only operate on MPI defined data types, not custom
- Operations
  - Synchronization – all processes wait
  - Data Movement – broadcast, scatter-gather

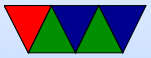
    scatter = take one structure and split among processes

    gather = take data from all processes and combine it
  - Reduction – one process combines results of all others



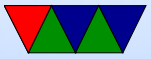
MPI_Barrier()

- All processes wait at this point.
- `MPI_Barrier (comm)`



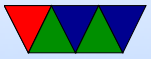
MPI_Bcast()

- `MPI_Bcast (&buffer,count,datatype,root,comm)`
- Sends data from the *root* process to each other process.
- Is blocking; when encountering a Bcast all nodes wait until they have received the data.



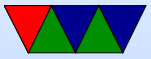
MPI_Scatter() / MPI_Gather()

- `MPI_Scatter (&sendbuf,sendcnt,sendtype,&recvbuf recvcnt,recvtype,root,comm)`
- Copies `sencnt` sized chunks of sendbuf to each processes `recvbuf`
- `MPI_Gather (&sendbuf,sendcnt,sendtype,&recvbuf, recvcount,recvtype,root,comm)`

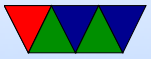


MPI_Reduce()

- `MPI_Reduce( void* send_data, void* recv_data, int count, MPI_Datatype datatype, MPI_Op op, in root, MPI_Comm communicator)`
- Operations
  - MPI_MAX,MPI_MIN – max, min
  - MPI_SUM – sum
  - MPI_PROD – product
  - MPI_LAND, MPI_BAND – logical/bitwise and
  - MPI_LOR,MPI_BOR – logical/bitwise OR

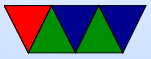


- MPI_LXOR,MPI_BXOR – logical/bitwise XOR
- MPI_MAXLOC,MPI_MINLOC – value and location



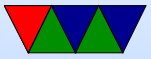
MPI_Allgather()

Gathers, but then broadcasts the result to all.

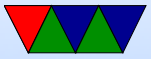


MPI_Allreduce()

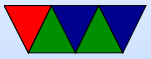
- Like an MPI_Reduce followed by an MPI_Bcast
- `MPI_Allreduce( void* send_data, void* recv_data, int count, MPI_Datatype datatype, MPI_Op op, MPI communicator)`
- Once the reduction is done, broadcasts the results to all processes



MPI_Reduce_scatter()

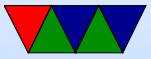


MPI_Alltoall()



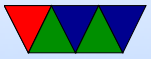
MPI_Scatterv()

Send non-contiguous chunks.



MPI_Scan()

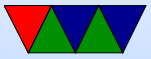
Lets you do partial reductions.



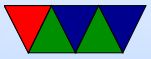
Custom Data Types

You can create custom data types that aren't the MPI default, sort of like structures.

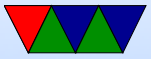
Open question: can you just cast your data into integers and uncast on the other side?



Groups vs Communicators

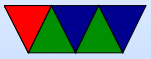


Virtual Topologies



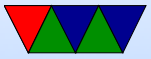
Examples

See the provided tar file with example code.

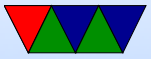


Running MPI code

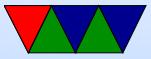
- `mpiexec -n 4 ./test_mpi`
- You'll often see `mpirun` instead. Some implementations have that, but it's not the official standard way.



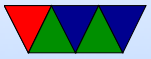
Send Example



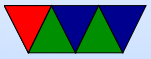
Blocking vs NonBlock Example?



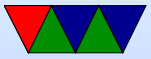
Wtime Example



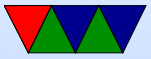
Barrier Example



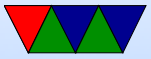
Bcast Example



Scatter Example



Gather Example



Reduce Example