Assignment MPI: Collective and Pt-to-Pt APIs

The purpose of this assignment is for you to learn more about

- implementing a dynamic workload partitioning scheme in MPI (here Master-Worker)

As usual all time measurements are to be performed on the cluster.

To be able to compile and run an MPI program on mamba, you need to add the line module load mpich/3.3.1 at the end of the file .bashrc located in the home directory of your account on mamba. (log off and back in afterward.)

To compile an MPI application, use the mpicc compiler in C and the mpicxx compiler in C++. They also serve as linker. To run an MPI application using 19 processes, you can run mpirun -n 19 ./myprogram. But you will need to have a proper node allocation first. And if you have a proper node allocation then specifying -n is not necessary because the cluster scheduler does that for you.

To queue an MPI job on mamba, you will need to specify how many nodes and how many cores per node you plan on using. You could use qsub -l nodes=2:ppn=16 to request two nodes with 16 cores each, or qsub -l procs=32 if you only care about having 32 cores independently of where they are. You can also use qsub -l nodes=4:ppn=4, to have 4 cores from 4 different nodes. There is currently a cap on mamba that prevents you from requiring more than 32 processes in total. This is abstracted in the scaffolding.

1 Numerical Integration : Master-Worker

A master-worker system enables dynamic scheduling of applications. The idea is that one of the MPI processes (usually rank 0) will be responsible for giving work to the other MPI processes and collect results.

Usually the master process starts by sending one chunk of work to all the workers and when one of the worker provides the result of that chunk of work, the master process will send a new chunk of work to the worker that just completed the work.

The workers will wait for a chunk of work to perform, perform that chunk of work, and return the result to the master node.

Pay attention that the master node needs to notify the worker nodes when there is no more work to perform so the workers can quit gracefully.

Question: Adapt the numerical integration to make it schedule the calculation using a master-worker system. Write that code in `master_worker/mpi_master_worker.cpp`. Use a granularity (size of the chunk) that you deem appropriate.

Question: Run and time that program on mamba and compute speedup. Use the `make run` and `make plot` scripts to do so.

2 Extra Credit: Advanced Master-Worker (30 pts)

One of the issue with such a master-worker implementation is that the worker process will be idle until the master provides the next chunk. To avoid this, it is common that the master will give more than a single chunk of work (say, 3 chunks of work) to each worker. That way, when a worker sends a result to the master node, the worker still has some chunks of work (here, 2 chunks) to perform.

Question: Adapt the numerical integration code to use advanced scheduling.

Question: Run and time that program on mamba. And generate speedup tables.