Assignment MPI: Point to point

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

- implementing a dynamic workload partitioning scheme in MPI (here Master-Worker).
- implementing operations using regular communication patterns.
- the impact of data partitioning.

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

When scheduling a job on mamba, the jobs have a fixed amount of memory available (for the entire job), you can request additional memory using `-l mem=120GB`. This request a TOTAL of 120GB for the whole job (as opposed to 120GB per process or per node). (The necessary parameters are given in the scaffolding.)

Strong scaling experiment. An experiment is a strong scaling experiment when you measure the speedup an algorithm achieves when you increase the number of resources. All the experiments we conducted so far were strong scaling experiments. Usually these are reported using a speedup chart.

Weak scaling experiment. An experiment is a weak scaling experiment when you increase the computational requirement of the problem proportionally to the number of resources allocated to the problem. Usually these are reported using a (processor,time) chart. The computation scales if the curve is flat.

1 Preliminary: ping pong

Question: Write a code that is to be run on two processes, where:

- process 0 read a value from the command line parameters;
- process 0 sends that value to process 1;
- process 1 reads it, add 2 to it;
- send it back to process 0;
- and process 0 prints it.

Write that code in `pingpong/mpi_ping_pong.cpp`.
Question: Test that code with `make run_1x2`.

2 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 that program on mamba using `make bench`. Once the jobs have completed `make table` and `make plot` to obtain tabulated times and speedup charts.

3 2D heat equation

The problem is defined on a discrete 2D space of size $n \times n$; let's call it H. Initialize H^0 using the provided functions. The kth iteration of the heat equation is defined by H^k is defined by

$$\begin{aligned} H^k[i][j] = \frac{1}{5}(&H^{k-1}[i-1][j]+ \\ &H^{k-1}[i][j-1] + H^{k-1}[i][j] + H^{k-1}[i][j+1]+ \\ &H^{k-1}[i+1][j]) \end{aligned}$$

(Take the elements out of the array as being $H^{k-1}[i][j]$)

Note that the implementation probably needs to keep H^k and H^{k-1} in memory.

Question: Implement a distributed memory version of the 2D heat equation problem. Write the code in `heat/mpi_heat.cpp`. Partition the data in blocks similarly to the matrix multiplication case.

(Hint: implement it sequentially first to understand the problem clearly.)

Question: Perform strong and weak scaling experiment to compute H^5 with `make bench`. Once the jobs are completed, plot performance chart with `make plot`.

Question: Get additional result on strong scaling with `make table_strong`. Does the code scale strongly?

Question: Get additional result on weak scaling with `make table_strong`. Does the code scale weakly?

Question: Describe how you would increase communication and computation overlap?

4 Extra Credit: Advanced Master-Worker

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.