Assignment: MPI - Algorithm and Partitioning.

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

- complex algorithms for distributed memory
- 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 (and not 120GB per node).

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 2D heat equation

A 2D heat equation is similar to the 1D equation from distributed memory algorithm assignment.

The problem is defined on a discrete 2D space of size $n \times n$; let's call it H. Initialize H in some fashion (random works). The kth iteration of the heat equation is defined by H^k is defined by

$$\begin{aligned} H^k[i][j] = \frac{1}{9}(&H^{k-1}[i-1][j-1] + H^{k-1}[i-1][j] + H^{k-1}[i-1][j+1] \\ &+H^{k-1}[i][j-1] + H^{k-1}[i][j] + H^{k-1}[i][j+1] \\ &+H^{k-1}[i+1][j-1] + H^{k-1}[i+1][j] + H^{k-1}[i+1][j+1]) \end{aligned}$$

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

The implementation probably need to keep H^k and H^{k-1} in memory.

Question: Implement a distributed memory version of the 2D heat equation problem. Partition the data the way you like better.

Question: Perform a strong scaling experiment to compute H^{20} from 1 core to 32 cores. (Feel free to restrict number of cores to particular numbers that matches your implementation.) Pick n such that H is about 1GB large, 10GB large, and 50GB large.

Question: Perform a weak scaling experiment to compute H^{20} from 1 core to 32 cores. (Feel free to restrict number of cores to particular numbers that matches your implementation.) Pick n such that on one core H is about 500MB large, 1GB large, and 2GB large.

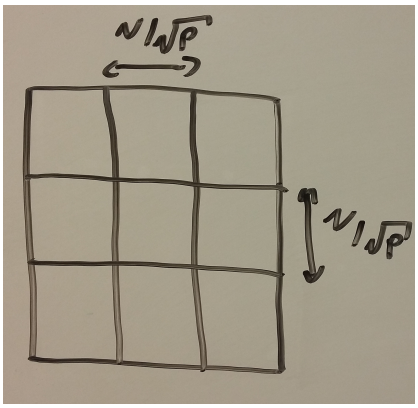
Question: How would you increase communication and computation overlap ?

2 Matrix multiplication

The problem is to compute iterated matrix multiplication defined by $x^k = Ax^{k-1}$, where A is a random matrix of size $n \times n$ and x^k is a vector of size n. Pick x^0 randomly.

For reference, $x^k = Ax^{k-1}$ is computed using $x^k[i] = \sum_j A[i][j]x^{k-1}[j]$. Or in other words, to compute $x^k[i]$ multiply element wise the ith row of the matrix by x^{k-1} and sum the values.

You should partition the data using blocks:



blocks

Question: Implement iterated matrix multiplication for the block decomposition partitioning scheme. Communicators are a particularly efficient way of implementing this block decomposition. For information see

- `http://mpitutorial.com/tutorials/introduction-to-groups-and-communicators/`
- `man MPI_Comm_split`
- `man MPI_Comm_free`

Question: Perform a strong scaling experiment to compute x^{20} from 1 core to 32 cores. (Feel free to restrict number of cores to particular numbers that matches your implementation.) Pick n such that A is about 1GB large, 20GB large, and 80GB large.

Question: Perform a weak scaling experiment to compute x^{20} from 1 core to 32 cores. (Feel free to restrict number of cores to particular numbers that matches your implementation.) Pick n such that on one core A is about 1GB large, 2GB large, and 4GB large.

Question: How would you increase communication and computation overlap ?