You can download these files using wget from the command line. This way you can easily download them to remote accounts, instead of downloading them to your computer and then uploading them to the remote account. You can also use the -c command to resume a partially downloaded file, if the download quit for any reason, for example:

wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz

will download chr1.fa.gz to the current directory, and continue the download if it was previously broken.

You can get the seed files (the reads) from:

www.ebi.ac.uk/ena

enter in ERP001953

To download them from the command line (from the above website):

wget -c ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR192/ERR192339/ERR192339.fastq.gz

wget -c ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR192/ERR192340/ERR192340.fastq.gz

...

You can download the mouse reference genome here:

http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/

you want to download the files chr*.fa.gz, which are gripped files of each mouse chromosome in FASTA format, e.g.:

wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz

wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr2.fa.gz

...

MPI Assignment 3 - Nov 2, 2017

You need to take a sequential C++ heat simulation and update it to run in parallel across multiple processes using MPI.

Download:

http://people.cs.und.edu/~tdesell/files/heat_diffusion_mpi/heat_diffusion.cxx

You can compile these on the lab machines as follows:

mpicxx heat_diffusion -o heat_diffusion

And then run, with:

./heat_diffusion <simulation name> <height> <width> <vertical slices> <horizontal slices> <time steps>

For example:

./heat_diffusion simtest 20 20 1 1 10

This will generate a number of files which are the values in the heat simulation for each time step. You can use these to validate that your MPI version is generating the correct and same results.

Read through the source file and understand what's going on. You need to parallelize the main loop which performs the simulation.

Copy this code into a new file 'heat_diffusion_mpi.cxx' that will perform a parallel version of this simulation. When you run this, you'll set the vertical and horizontal slices, and then when running your number of MPI processes needs to be equal to (horizontal_slices * vertical slices).

You should parallelize your code by dividing up the heat simulation by splicing it into the specified number of rectangles. You will need to use MPI_Send/MPI_Recv to do what is sometimes called a "halo" transfer. Each rectangle needs the surrounding values to be able to calculate it's values for the next time step.

Note that each process will need to initialize it's own smaller sized array. To validate your results, you will need to transfer data from each other process using MPI_Allgather and then write the entire simulation to the output file.

You can assume that the width and height are evenly divisible by the vertical and horizontal slice values.
Bonus (10%): Handle non-evenly divisible dimensions.
Bonus (20%): Update the code to perform a 3 dimensional heat simulation. In the command line arguments include a depth value, as well as a depth slices value.
When complete, upload your source file in a zip file to blackboard.

OpenCL Assignment 1 - Nov 9, 2017

Incoming.

OpenCL Assignment 2 - Nov 23, 2017

You need to take a sequential C++ heat diffusion simulation and update it's display loop to perform the diffusion on the GPU. Download: