You have (or will shortly) have access to COMET.

Read the COMET user guide:

<https://portal.xsede.org/sdsc-comet>

figure out how to copy of your existing Calculate PI MPI on there.

(please refer to:

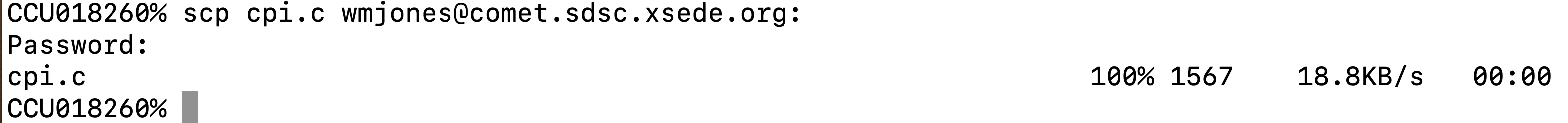
<https://portal.xsede.org/web/xup/documentation-overview>

(you will probably have to submit MFA to do this. I did the DUO phone app, since I’ve used this before to VPN into Clemson, etc …)

<https://portal.xsede.org/mfa>

for info about command line copying and also using Globus ...

You can also SCP your files over there like this:



Once your file is copied over, here are some notes about running a job in interactive mode on COMET:

*This will get you to the point that you can use COMET to submit a job (interactively versus using a script), and also will give you some hints about sizes and data types, and memory space.*

First, I got a lot of this from:

<https://portal.xsede.org/sdsc-comet>

(searching for “scratch” and also “interactive”)

Doing it the “interactive way”

First SSH to the login node of COMET:

CCU018260% ssh -l wmjones comet.sdsc.xsede.org

Password:

Last login: Wed Dec 4 11:21:04 2019 from 199.120.30.232

Rocks 7.0 (Manzanita)

Profile built 13:03 03-Dec-2019

Kickstarted 14:18 03-Dec-2019

WELCOME TO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-----/ \_\_\_\_/ \_\_ \/ |/ / \_\_\_\_/\_ \_\_/

--/ / / / / / /|\_/ / \_\_/ / /

/ /\_\_\_/ /\_/ / / / / /\_\_\_ / /

\\_\_\_\_/\\_\_\_\_/\_/ /\_/\_\_\_\_\_/ /\_/

[wmjones@comet-ln3 ~]$

Then just to check that I’m in my working directory:

[wmjones@comet-ln3 ~]$ pwd

/home/wmjones

[wmjones@comet-ln3 ~]$

Here’s the cpi.c file that I just copied over:

[wmjones@comet-ln3 ~]$ ls -la cpi.c

-rw-r--r-- 1 wmjones ccu100 1567 Jan 21 11:26 cpi.c

[wmjones@comet-ln3 ~]$

So, from there, I’m going to make a folder to keep this file. you may want to organize it into some structure. I put it in my examples folder. Then going to check to see which moduels are loaded:

[wmjones@comet-ln3 examples]$ module list

Currently Loaded Modulefiles:

1) intel/2018.1.163 2) mvapich2\_ib/2.3.2

[wmjones@comet-ln3 examples]$

I compiled it, and it worked:

[wmjones@comet-ln3 examples]$ mpicc -o cpi cpi.c

[wmjones@comet-ln3 examples]$ ls

cpi cpi.c MPI

[wmjones@comet-ln3 examples]$

We DO NOT WANT TO RUN THE PROGRAM HERE ON THE LOGIN NODE.

We want to schedule an interactive session on a compute node, as per the Usage Guide.

[wmjones@comet-ln3 examples]$ srun --partition=compute --pty --nodes=1 --ntasks-per-node=24 -t 00:30:00 --wait=0 --export=ALL /bin/bash  
srun: job 30990554 queued and waiting for resources   
srun: job 30990554 has been allocated resources  
[wmjones@comet-16-46 examples]$

Now we see that I’m on a different node: “comet-16-46”

So, recompile just to be sure that works:

[wmjones@comet-16-46 examples]$ mpicc -o cpi cpi.c

[wmjones@comet-16-46 examples]$

It does. So now let’s run it:

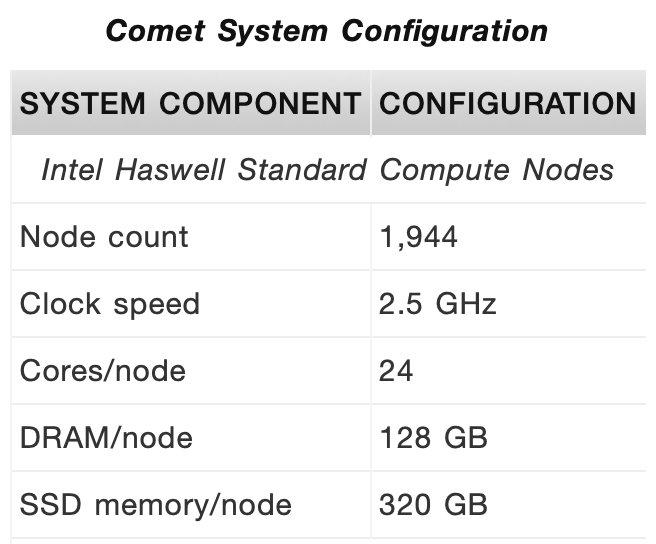
[wmjones@comet-16-46 examples]$ mpicc -o cpi cpi.c  
[wmjones@comet-16-46 examples]$ mpirun -np 2 ./cpi  
Process 0 of 2 is on comet-16-46.sdsc.edu  
Process 1 of 2 is on comet-16-46.sdsc.edu  
pi is approximately 3.1415926535900072, Error is 0.0000000000002141  
wall clock time = 2.424622  
[wmjones@comet-16-46 examples]$

So now, we’ve got the timing data for np=2.

What about np=1 ?

[wmjones@comet-12-66 examples]$ mpirun -np 1 ./cpi  
Process 0 of 1 is on comet-12-66.sdsc.edu  
pi is approximately 3.1415926535899708, Error is 0.0000000000001776  
wall clock time = 4.847824  
[wmjones@comet-12-66 examples]$

So, with COMET on a standard compute node, you have:



24 cores. So I want you to recollect the data for np = {1,2,3,4,5,6,7,8,16}, take 3 time measurements per run (like three timings for 1, 3 timings for 2, etc, and then find the average and the use these timings to plot out the execution time, speedup, and efficiencies as a function of np, just like you did for the prior assignment, but now on COMET instead.

Record the data, save all artifacts, create the plots, and then write a Word (or PDF) report that includes the data, plots showing what the results were. Zip all artifacts together and submit into Moodle.

WMJ