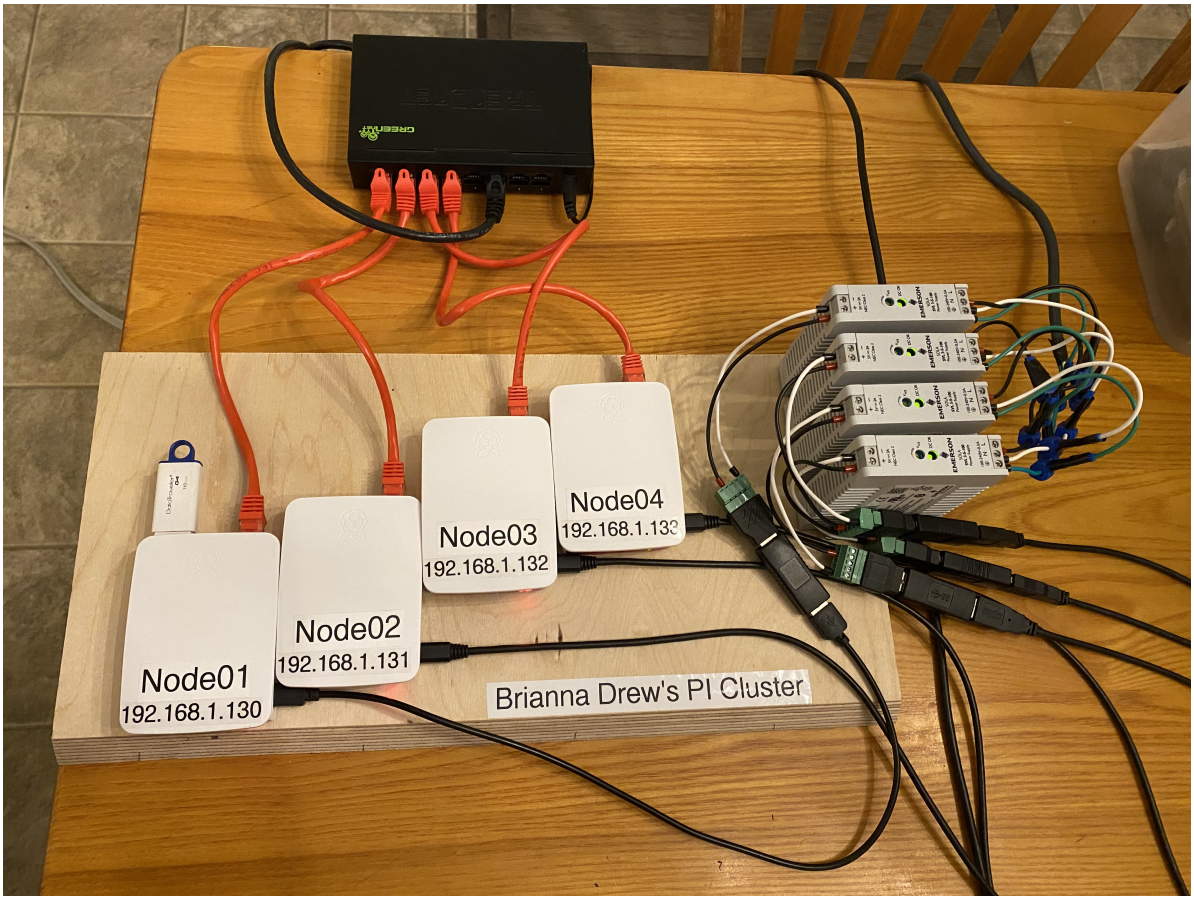
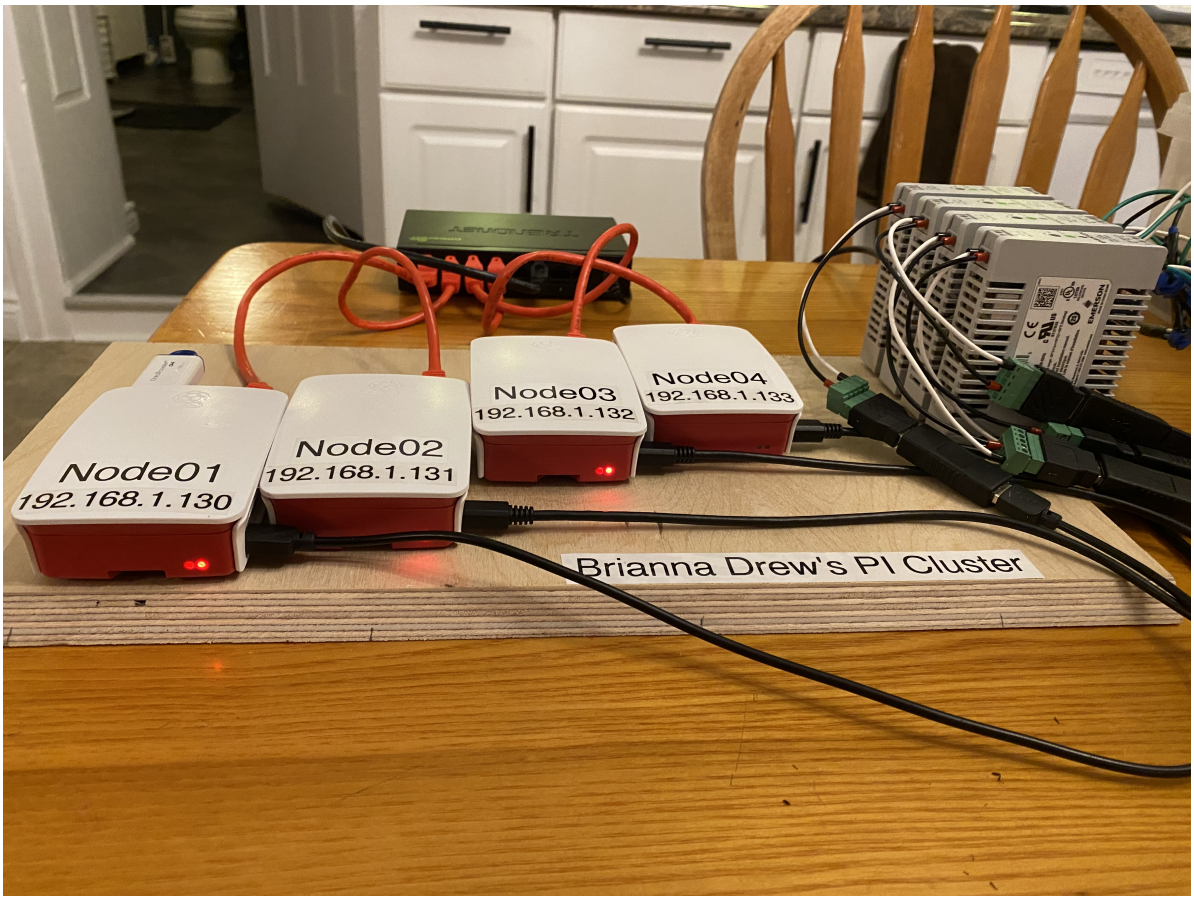
HPC Final Project - Raspberry Pi Cluster & SLURM

By Brianna Drew

Building and Setting Up the Cluster





The purpose of this project was to build a cluster computer with Raspberry Pis and schedule some jobs with SLURM. I had access to 4 Raspberry Pis along with power supplies, SD cards for each of the Pis, a network switch, access point, and ethernet cables to connect the devices.

Hardware

Quantity	Description
4	Raspberry Pi model B Rev 1.2
4	Emerson 5v, 3 amp power supplies
1	Trendnet 8 port switch
4	Patriot 32GB SD cards
1	16GB thumb drive for a shared folder across the cluster

plus associated power cabling and Cat 5e network cables.

The Raspberry Pi boots from an SD card so the operating system must be flashed to the SD card of each Pi. The program that I used to flash the SD cards with the OS (Raspbian Buster) is called *balenaEtcher*. The ethernet switch was connected to the home network and the plan was to access the cluster through ssh. After flashing the SD cards, ssh needs to be enabled. To do that, a blank file called ssh must exist in the boot partition of the card. The boot partition can be viewed and edited in Windows, so the SD card was inserted on my laptop and an empty file called ssh was created in the boot partition. This was done on each of the 4 SD cards. The next step was to hook up all the devices. Power cables from the power supplies were connected to the Pis. SD cards were inserted into the Pis. Ethernet cables were connected from the Pis to the switch and from the switch to the network. On powerup of each of the Pis, the default network IP is determined from the network DHCP server so initially I don't know what each of the Pis IP addresses are. To determine the IP addresses, I did a network scan with a program called *Advance IP Scanner*. After finding out what the IP addresses were for each of the Pis, I went into the router and set up static DHCP by entering the MAC address of each of the Pis along with the desired IP address from the pool of freed up IP addresses. This way, every time the Pi boots, the router sees its MAC address and assigns it the desired IP address. I chose:

Node	Address
node01	192.168.1.130
node02	192.168.1.131
node03	192.168.1.132
node04	192.168.1.133

Now, every time I boot a Pi, it gets the IP address that I assigned. The next step is to ssh into each of the Pis and run "sudo raspi-config" to change the time zone, locale, and expand the file system to use the entire size of the SD card. I use the program *Putty* to ssh in and the username will be pi and the password will be raspberry by default. The Pis should each have a unique hostname, so we will change this in three steps:

> sudo hostname node01 (or node02, node03, node04)
>
> sudo nano /etc/hostname and modify the hostname to the appropriate node name
>
> sudo nano /etc/hosts and change raspberrypi to the appropriate node name

Next, make sure ntp is installed on each Pi so that all times are correct with:

```
sudo apt install ntpdate -y
```

and reboot each of the Pis by running:

```
sudo reboot
```

then run:

```
sudo apt update
```

on each of the Pis to ensure repositories etc. are up to date. When logging into each Pi, you should be greeted with pi@node01, pi@node02, pi@node03, and pi@node04.

Shared Storage

Each node needs to be able to access a shared set of files which means the USB thumb drive needs to be set up as a share across all 4 Pis. The USB drive will reside on node01 (the master). With the USB drive installed and while I'm logged into node01, I run the command:

```
lsblk
```

This displays details about all existing block devices, one of which is the USB drive.

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda         8:0    1 14.4G  0 disk
└─sda1      8:1    1 14.4G  0 part /clusterfs
mmcblk0     179:0    0 29.1G  0 disk
├─mmcblk0p1 179:1    0  256M  0 part /boot
└─mmcblk0p2 179:2    0 28.9G  0 part /
```

As can be seen above, sda1 which is the USB drive is mounted in a directory called /clusterfs which is accessible across all of the nodes. For this to happen, a number of steps have to happen:

1. Format the drive to ext4 by running

   ```
   sudo mkfs.ext4 /dev/sda1
   ```

2. Create the mount directory and set permissions:

   ```
   sudo mkdir /clusterfs
   sudo chown nobody.nogroup –R /clusterfs
   sudo chmod 777 –R /clusterfs
   ```

Next, to automate mounting of the USB drive at boot, some modifications have to be done to a file called /etc/fstab. We need the unique ID identifier of the USB drive first. By running:

```
blkid
```

we see:

```
/dev/mmcblk0p1: LABEL_FATBOOT="boot" LABEL="boot" UUID="C839-E506"
BLOCK_SIZE="512" TYPE="vfat" PARTUUID="3d2644fc-01"
```

> /dev/mmcblk0p2: LABEL="rootfs" UUID="568caafd-bab1-46cb-921b-cd257b61f505" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="3d2644fc-02"
>
> /dev/sda1: UUID="c17f712a-899b-4f36-bbc4-903674b7e158" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="b3437f59-8656-934a-9168-0e690a719ee6"

The UUID for /dev/sda1 is what we need to include in the /etc/fstab file. So:

```
sudo nano /etc/fstab
```

and add the line: *UUID=c17f712a-899b-4f36-bbc4-903674b7e158 /clusterfs ext4 defaults 0 2*. Save the file and exit. To mount the drive without rebooting, enter:

```
sudo mount –a
```

Now that the drive is mounted, we can make sure permissions are set on the mounted folder.

```
sudo chown nobody.nogroup –R /clusterfs
sudo chmod –R 766 /clusterfs
```

Next, edit the /etc/exports file to add the line: */clusterfs 192.168.1.0/24(rw,sync,no_root_squash,no_subtree_check)* and then update the NFS kernel server:

```
sudo exportfs –a
```

To set up the share on all the clients (node02,node03,node04), install on each of the clients:

```
sudo apt install nfs-common –y
sudo mkdir /clusterfs
sudo chown nobody.nogroup /clusterfs
sudo chmod –R 777 /clusterfs
```

Now we need to modify the file /etc/fstab on each of the client nodes by adding the line:

192.168.1.130:/clusterfs /clusterfs nfs defaults 0 0

Now, after running:

```
sudo mount –a
```

on all of the clients, if a file is created or changed in /clusterfs it should be the same across all nodes.

Installing SLURM

Since node01 is going to be the master, we need to ssh in and start setting it up.

1. Edit the /etc/hosts file and add the following lines:

   192.168.1.131 node02

   192.168.1.132 node03

   192.168.1.133 node04

2. Install the SLURM packages:

   ```
   cd /etc/slurm-llnl
   cp /usr/share/doc/slurm-client/examples/slurm.conf.simple.gz
   ```

> gzip -d slurm.conf.simple.gz
>
> mv slurm.conf.simple slurm.conf

3. Edit the file /etc/slurm-llnl/slurm.conf as follows: Modify the first configuration line to include the hostname of the master node and its IP address: *SlurmctldHost=node01(192.168.1.130)* and then edit the SelectType field as follows:

```
SelectType=select/cons_res
SelectTypeParameters=CR_Core
```

   Set the cluster name: *ClusterName=glmdev*. At the end of the file, delete the compute node entry and add the following lines:

```
NodeName=node01 NodeAddr=192.168.1.130 CPUs=4 State=UNKNOWN
NodeName=node02 NodeAddr=192.168.1.131 CPUs=4 State=UNKNOWN
NodeName=node03 NodeAddr=192.168.1.132 CPUs=4 State=UNKNOWN
NodeName=node04 NodeAddr=192.168.1.133 CPUs=4 State=UNKNOWN
```

   Delete the example partition in the file and create a default partition with 3 nodes by adding:

```
PartitionName=mycluster Nodes=node[02-04] Default=YES MaxTime=INFINITE
State=UP
```

4. Configure cgroups support by creating a file /etc/slurm-llnl/cgroup.conf with the contents:

```
CgroupMountpoint="/sys/fs/cgroup"
CgroupAutomount=yes
CgroupReleaseAgentDir="/etc/slurm-llnl/cgroup"
AllowedDevicesFile="/etc/slurm-llnl/cgroup_allowed_devices_file.conf"
ConstrainCores=no
TaskAffinity=no
ConstrainRAMSpace=yes
ConstrainSwapSpace=no
ConstrainDevices=no
AllowedRamSpace=100
AllowedSwapSpace=0
MaxRAMPercent=100
MaxSwapPercent=100
MinRAMSpace=30
```

5. Whitelist system devices by creating the file /etc/slurm-llnl/cgroup_allowed_devices_file.conf and adding contents as follows:

```
/dev/null
/dev/urandom
/dev/zero
/dev/sda*
/dev/cpu/*/*
/dev/pts/*
/clusterfs*
```

6. Copy the files to the shared storage:

> sudo cp slurm.conf cgroup.conf cgroup_allowed_devices_file.conf /clusterfs

> sudo cp /etc/munge/munge.key /clusterfs

7. Enable and start Munge:

> sudo systemctl enable munge
>
> sudo systemctl start munge

8. Enable and start the SLURM daemon:

> sudo systemctl enable slurmd
>
> sudo systemctl start slurmd

9. Enable and start the SLURM daemon:

> sudo systemctl enable slurmctld
>
> sudo systemctl start slurmctld

Configure Compute Nodes (Nodes 02-04)

1. Install the SLURM Client as follows:

> sudo apt install slurmd slurm-client –y

2. Update the /etc/hosts file on each of the nodes indicating IP address and matching node name i.e.:

```
for node02:                  for node03:                  fornode04:
192.168.1.130  node01        192.168.1.130  node01        192.168.1.130
node01
192.168.1.132  node03        192.168.1.131  node02        192.168.1.131
node02
192.168.1.133  node04        192.168.1.133  node04        192.168.1.132
node03
```

3. Copy the following from shared storage so that nodes 2-4 match node01:

> sudo cp /clusterfs/munge.key /etc/munge/munge.key
>
> sudo cp /clusterfs/slurm.conf /etc/slurm-llnl/slurm.conf
>
> sudo cp /clusterfs/cgroup* /etc/slurm-llnl

4. Enable and start Munge:

> sudo systemctl enable munge
>
> sudo systemctl start munge

5. Start the SLURM Daemon:

> sudo systemctl enable slurmd
>
> sudo systemctl start slurmd

6. Run the following from the Master node to show that SLURM is running:\

> sinfo

If everything is running, we see the following:

> PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
>
> mycluster* up infinite 3 idle nodes[02-04]

SLURM is now operational. I had to make a shellscript file in the home directory of the Master node to start SLURM on boot as it isn't started automatically by default. I called the file startslurm and the contents are:

> sudo systemctl start slurmd
>
> sudo systemctl start slurmctld

The file needs execute permissions, so I set permissions as follows:

> sudo chmod +x ./startslurm

To start SLURM at boot, run the following:

> ./startslurm

Scheduling Jobs with SLURM

Simple Jobs + Hello World

helloworld.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=mycluster
cd $SLURM_SUBMIT_DIR
echo "Hello, World!" > helloworld.txt
```

pi@node01: /clusterfs

```
Using username "pi".
Authenticating with public key "rsa-key-20211122"
Linux node01 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 14 17:27:31 2021 from 192.168.1.254

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@node01:~ $ sinfo
PARTITION  AVAIL  TIMELIMIT  NODES  STATE NODELIST
mycluster*    up   infinite      3   idle node[02-04]
pi@node01:~ $ srun --nodes=3 hostname
node02
node04
node03
pi@node01:~ $ srun --ntasks=3 hostname
node02
node02
node02
pi@node01:~ $ srun --nodes=2 --ntasks-per-node=3 hostname
node02
node03
node02
node02
node03
node03
pi@node01:~ $ cd /clusterfs/
pi@node01:/clusterfs $ sbatch ./helloworld.sh
Submitted batch job 228
pi@node01:/clusterfs $
```

helloworld.txt

```
Hello, World!
```

R Plots

generate.R

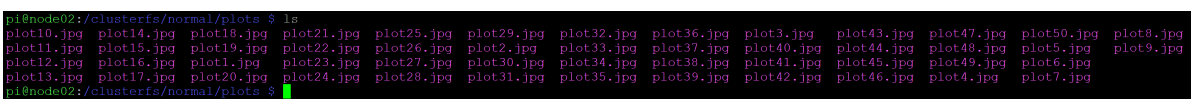
```
arg = commandArgs(TRUE)
samples = rep(NA, 100000)
for ( i in 1:100000 ){ samples[i] = mean(rexp(40, 0.2)) }
jpeg(paste('plots/', arg, '.jpg', sep=""))
hist(samples, main="", prob=T, color="darkred")
lines(density(samples), col="darkblue", lwd=3)
dev.off()
```

submit.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=mycluster
cd $SLURM_SUBMIT_DIR
mkdir plots
R --vanilla -f generate.R --args "plot$SLURM_ARRAY_TASK_ID"
```

pi@node02: /clusterfs/normal

```
slurm-8_12.out  slurm-8_17.out  slurm-8_21.out  slurm-8_26.out  slurm-8_30.out
pi@node02:/clusterfs/normal $ mkdir plots
mkdir: cannot create directory ‘plots’: File exists
pi@node02:/clusterfs/normal $ R --vanilla -f generate.R --args "plot1"

R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: arm-unknown-linux-gnueabihf (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> arg = commandArgs(TRUE)
> samples = rep(NA, 100000)
> for ( i in 1:100000 ){ samples[i] = mean(rexp(40, 0.2)) }
> jpeg(paste('plots/', arg, '.jpg', sep=""))
> hist(samples, main="", prob=T, color="darkred")
Error in plot.new() : could not open file 'plots/plot1.jpg'
Calls: hist -> hist.default -> plot -> plot.histogram -> plot.new
Execution halted
pi@node02:/clusterfs/normal $ sbatch --array=[1-50] submit.sh
Submitted batch job 83
pi@node02:/clusterfs/normal $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
pi@node02:/clusterfs/normal $ sbatch --array=[1-50] submit.sh
Submitted batch job 133
pi@node02:/clusterfs/normal $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
       133_[33-50] mycluster submit.s       pi PD       0:00      1 (Resources)
            133_21 mycluster submit.s       pi  R       0:00      1 node02
            133_22 mycluster submit.s       pi  R       0:00      1 node04
            133_23 mycluster submit.s       pi  R       0:00      1 node02
            133_24 mycluster submit.s       pi  R       0:00      1 node03
            133_25 mycluster submit.s       pi  R       0:00      1 node04
            133_26 mycluster submit.s       pi  R       0:00      1 node03
            133_27 mycluster submit.s       pi  R       0:00      1 node04
            133_28 mycluster submit.s       pi  R       0:00      1 node02
            133_29 mycluster submit.s       pi  R       0:00      1 node03
            133_30 mycluster submit.s       pi  R       0:00      1 node02
            133_31 mycluster submit.s       pi  R       0:00      1 node03
            133_32 mycluster submit.s       pi  R       0:00      1 node04
pi@node02:/clusterfs/normal $
```



/plots

```
* see attatched folder /plots to view plots generated from generate.R *
```

Hello World in MPI

hello_mpi.c

```c
#include <stdio.h>
#include <mpi.h>
int main(int argc, char** argv){
    int node;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &node);
    printf("Hello World from Node %d!\n", node);
    MPI_Finalize();
}
```

sub_mpi.sh

```bash
#!/bin/bash
cd $SLURM_SUBMIT_DIR
# Print the node that starts the process
echo "Master node: $(hostname)"
# Run our program using OpenMPI.
# OpenMPI will automatically discover resources from SLURM.
mpirun a.out
```

pi@node02: /clusterfs

```
Using username "pi".
Authenticating with public key "rsa-key-20211122"
Linux node01 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 14 17:32:41 2021 from 192.168.1.254

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@node01:~ $ srun --pty bash
pi@node02:~ $ cd /clusterfs
pi@node02:/clusterfs $ mpicc hello_mpi.c
pi@node02:/clusterfs $ ls
a.out                           helloworld.sh   normal          slurm-82.out
calc-pi                         helloworld.txt  slurm-228.out   slurm.conf
cgroup_allowed_devices_file.conf  lost+found    slurm-60.out    sub_mpi.sh
cgroup.conf                     mike.txt        slurm-6.out     usr
hello_mpi.c                     munge.key       slurm-73.out
pi@node02:/clusterfs $ sbatch --nodes=3 --ntasks-per-node=2 sub_mpi.sh
Submitted batch job 230
pi@node02:/clusterfs $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
               229 mycluster     bash       pi  R       1:29      1 node02
               230 mycluster sub_mpi.       pi  R       0:00      3 node[02-04]
```

Calculate Pi in Python

calculate.py

```
from mpi4py import MPI
from math   import pi as PI
from numpy  import array

def comp_pi(n, myrank=0, nprocs=1):
    h = 1.0 / n
    s = 0.0
    for i in range(myrank + 1, n + 1, nprocs):
        x = h * (i - 0.5)
        s += 4.0 / (1.0 + x**2)
    return s * h

def prn_pi(pi, PI):
    message = "pi is approximately %.16f, error is %.16f"
    print  (message % (pi, abs(pi - PI)))

comm = MPI.COMM_WORLD
nprocs = comm.Get_size()
myrank = comm.Get_rank()

n    = array(0, dtype=int)
pi   = array(0, dtype=float)
mypi = array(0, dtype=float)

if myrank == 0:
    _n = 20 # Enter the number of intervals
    n.fill(_n)
comm.Bcast([n, MPI.INT], root=0)
_mypi = comp_pi(n, myrank, nprocs)
mypi.fill(_mypi)
comm.Reduce([mypi, MPI.DOUBLE], [pi, MPI.DOUBLE],
            op=MPI.SUM, root=0)
if myrank == 0:
    prn_pi(pi, PI)
```

sub_calc_pi.sh

```
#!/bin/bash
#SBATCH --ntasks=6
cd $SLURM_SUBMIT_DIR
mpiexec -n 6 /clusterfs/usr/bin/python3 calculate.py
```

```
pi@node01:/clusterfs/calc-pi $ ls
ass1_2         lab5.cpp       slurm-68.out  slurm-75.out
assn1_2.cpp    slurm-234.out  slurm-69.out  slurm-76.out
calculate.py   slurm-66.out   slurm-70.out  sub_calc_pi.sh
lab5           slurm-67.out   slurm-71.out  sub_install_pip.sh
pi@node01:/clusterfs/calc-pi $ sbatch sub_calc_pi.sh
Submitted batch job 236
pi@node01:/clusterfs/calc-pi $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
               236 mycluster sub_calc       pi  R       0:01      2 node[02-03]
pi@node01:/clusterfs/calc-pi $ cat slurm-236.out
pi is approximately 3.1418009868930938, error is 0.0002083333033007
pi@node01:/clusterfs/calc-pi $
```

Calculate Pi with MPI

pi.c

```c
#include <mpi.h>
#include <stdio.h>
#include <math.h>

#define NINTERVALS 10000

double f(double);
double f(double a)
{
    return (4.0 / (1.0 + a * a));
}

int main(int argc, char *argv[])
{
    int myid, numprocs, i;
    double PI25DT = 3.14159265358979323846264338;
    double mypi, pi, h, sum, x;
    double startwtime = 0.0, endwtime;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);

    startwtime = MPI_Wtime();

    h = 1.0 / (double) NINTERVALS;
    sum = 0.0;

    for (i = myid + 1; i <= NINTERVALS; i += numprocs) {
        x = h * ((double) i - 0.5);
        sum += f(x);
    }
    mypi = h * sum;

    MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

    if (myid == 0) {
        printf("pi is approximately %.16f, Error is %.16f\n", pi, fabs(pi -
PI25DT));
        endwtime = MPI_Wtime();
```

```
        printf("wall clock time = %f\n", endwtime - startwtime);
    }

    MPI_Finalize();
    return 0;
}
```

sub_mpi_pi.sh

```
#!/bin/bash
cd $SLURM_SUBMIT_DIR
echo "Master node: $(hostname)"
mpirun a.out
```

pi@node01: /clusterfs

```
Using username "pi".
Authenticating with public key "rsa-key-20211122"
Linux node01 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 14 20:08:24 2021 from 24.114.86.130

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@node01:~ $ cd /clusterfs
pi@node01:/clusterfs $ nano pi.c
pi@node01:/clusterfs $ srun --pty bash
pi@node02:/clusterfs $ cd /clusterfs
pi@node02:/clusterfs $ mpicxx pi.c
pi@node02:/clusterfs $ exit
exit
pi@node01:/clusterfs $ nano sub_mpi_pi.sh
pi@node01:/clusterfs $ sbatch --nodes=3 --ntasks-per-node=1 sub_mpi_pi.sh
Submitted batch job 244
pi@node01:/clusterfs $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
               244 mycluster sub_mpi_       pi  R       0:01      3 node[02-04]
pi@node01:/clusterfs $ ls
a.out                            lab4_sub.sh  slurm-228.out  slurm-73.out
calc-pi                          lab7.cpp     slurm-230.out  slurm-82.out
cgroup_allowed_devices_file.conf lost+found   slurm-238.out  slurm.conf
cgroup.conf                      mike.txt     slurm-240.out  sub_mpi_pi.sh
hello_mpi.c                      munge.key    slurm-244.out  sub_mpi.sh
helloworld.sh                    normal       slurm-60.out   usr
helloworld.txt                   pi.c         slurm-6.out
pi@node01:/clusterfs $ cat slurm-244.out
Master node: node02
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.018314
pi@node01:/clusterfs $
```

Testing One of My Programs - Lab #7 (same as Lab #4)

lab7.cpp

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
    int procNum, procRank;
    int m,n, size;
    int sumProc = 0, sumAll =0;
    char test[] = "hi";

    MPI_Status status;

    MPI_Init(NULL, NULL);
    // stores total number of threads in procNum
    MPI_Comm_size(MPI_COMM_WORLD, &procNum);
    // each thread stores its own rank in procRank
    MPI_Comm_rank(MPI_COMM_WORLD, &procRank);

    // get the size of the array in rank 0 this demo has to get the inputs 4 by
5
    if(procRank == 0)
    {
        printf("Type the array size \n")
        scanf("%i %i", &m, &n);
    }
    // send the array size to all threads
    MPI_Bcast(&m, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    // set the array static for this demo sums to 866
    float arr[4][5]={
    {50,55,62,70,85},
    {35,42,45,47,49},
    {32,33,36,37,38},
    {25,30,30,35,30}
    };

    // send the array out to all threads
    for (int i=0; i < m; i++)
    {
        MPI_Bcast(arr[i], n, MPI_INT, 0, MPI_COMM_WORLD);
    }

    // each thread sums its row
    for(int i = 0; i < n; i++)
    {
        sumProc += arr[procRank][i];
    }

    // have each thread output its name and sum
    printf("Hi, I'm %i ", procRank);
    printf("My sum is: %i", sumProc);
    printf("\n");
```

```
    //sum all back to thread 0
    MPI_Reduce(&sumProc, &sumAll, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);

    // if thread 0 sum all thread results together
    if (procRank == 0);
    {
        //print the output
        printf("sumAll = %i \n", sumAll);
    }

    //end program
    MPI_Finalize();
    return 0;
}
```

lab4_sub.sh

```
#!/bin/bash
#SBATCH --nodes=3
#SBATCH --ntasks-per-node=1
#SBATCH --partition=mycluster
cd $SLURM_SUBMIT_DIR
echo "Master node: $(hostname)"
mpirun a.out
```

```
pi@node02:/clusterfs $ nano lab7.cpp
pi@node02:/clusterfs $ mpicxx lab7.cpp
pi@node02:/clusterfs $ ls
a.out                           helloworld.txt  normal          slurm-73.out
calc-pi                         lab4_sub.sh     slurm-228.out   slurm-82.out
cgroup_allowed_devices_file.conf  lab7.cpp        slurm-230.out   slurm.conf
cgroup.conf                     lost+found      slurm-238.out   sub_mpi.sh
hello_mpi.c                     mike.txt        slurm-60.out    usr
helloworld.sh                   munge.key       slurm-6.out
pi@node02:/clusterfs $ exit
exit
pi@node01:/clusterfs $ sbatch lab4_sub.sh
Submitted batch job 240
pi@node01:/clusterfs $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
SON)
               240 mycluster lab4_sub       pi  R       0:01      3 node[02-04]
pi@node01:/clusterfs $ cat slurm-240.out
Master node: node02
Hi, I'm 0 My sum is: 322
Hi, I'm 1 My sum is: 218
Hi, I'm 2 My sum is: 176
sumAll = 0
sumAll = 716
sumAll = 0
pi@node01:/clusterfs $ nano lab7.cpp
pi@node01:/clusterfs $
```

References

https://glmdev.medium.com/building-a-raspberry-pi-cluster-784f0df9afbd

https://www.mcs.anl.gov/research/projects/mpi/mpi_ruby/talks/mcs/example3-1.html