CS546 Parallel and distributed Processing

Programming Assignment

Manual

In this assignment, we have to implement a simple MPI-IO program. The homework has two parts.

1. One is to simply gain some familiarity with MPI-IO. MPI-IO will likely be part of whatever I/O stack you will be using in an HPC environment. We want each MPI process to write its rank to a common file using The MPI-IO interface.

2. The second part is to build a parallel I/O benchmark using MPI-IO. In this Benchmark, we will open a temporary file, write random data to a file, and close the file. Then we will open it again, read the data from the file and close it again. All operations are performed by each rank to a common file.

This assignment is done in MPI, run on Amazon AWS(Which includes OrangeFS file system across all the instances). Here are the main programs to be executed:

1. first.c

2. hw6.c (Second Part)

Execution:

These programs are run on Amazon Aws (Which includes OrangeFS file system across all the instances, leveraging the configured EBS Storage).

Configuration of OrangeFS file system On Amazon AWS: To build and install your OrangeFS AMIs with Community Support from Amazon Web Services (AWS) Marketplace and we need to subscribe OrangeFS in AWS Marketplace.

1. First we need to create VPC since OrangeFS installation requires an AWS Virtual Private Cloud with a unique VPCID. The VPC needs a single public subnet and an Internet gateway.

- Create VPC on VPC Dashboard on AWS.

- Enter VPC Name (HW6)

- Enter Availability Zone - us-west-2a

- Note down Public subnet, VPCID- 10.0.0.0/24 , vpc-4e52482b

- Click Create VPC.

2. Now we need to choose the cluster size and Template. The AWS Marketplace enables AMIs that can support multiple product configurations with OrangeFS:

- 1 Node: Single Amazon Machine Instance (AMI)

- 4 Node: 4 AMIs, 5 TB of space

- 8 Node: 8 AMIs, 10 TB of space

- 16 Node: 16 AMIs, 20 TB of space

For 32 Nodes,I setup additional clients on windows.

3. Now, we have create AWS Cloud Formation Stack, to build and install OrangeFS in an AWS Cloud Formation Stack. Follow the instructions to do from the AWS Cloud Formation Console.

- Enter Stack Name: MPI I/O

- Enter Template URL based on the number of nodes.

- Enter the subnet ID of an existing public subnet in VPC that has a connected Internet gateway.

- Enter Private subnet ID.

- Select the instance type, selected c1.xlarge.

- Enter VPCID,

- Enter existing EC2 Key Pair to enable SSH access: chiru.pem

Now, click create Stack button to create stack. Once it created, cloud stack configuration created with all the required pieces to provide a NAT instance into VPC.now we need to launch the EC2 instances it and automatically configure an OrangeFS file system across all the instances, leveraging the configured EBS Storage.

1. Now, Log in to the EC2 instance with your AWS key as follows:

```
chmod.exe 400 chiru.pem

ssh -i "chiru.pem" ec2-user@ec2-35-162-227-245.us-west-2.compute.amazonaws.com
```

-Use ssh again to access one of the OrangeFS instances.

2. now,we can test the installation by listing the contents of /mnt/orangefs:

```
ls /mnt/orangefs
```

we can see a Lost+Found directory here, you have successfully created the file system.

3.we can install additional clients to access the file system if we need.

4.Now,we need to setup cluster of n nodes.the cluster computes all the instances in a cluster in same subnet.

-Add the all the nodes in internal ip address (35-162-227-245) to etc/hosts like procss1 process2 …….etc

-we need to create a script to submit files from master instance to all other instances. Moreover, transfer chiru.pem to all the nodes and they can use password-less ssh to communicate.

5. Now,we need to Install and Run MPI

```
sudo yum install mpich2

sudo yum install mpich2-devel
```

This will pull in the GCC compiler, the MPI runtimes, and the mpicc wrapper.and have to do this on all compute nodes. We have to add the following lines to .bashrc:

```
export PATH=/usr/lib64/openmpi/bin:$PATH

export LD_LIBRARY_PATH=/usr/lib64/openmpi/lib
```

Now execute the both the parts as run as follows.

1.firstpart.c

Compile the below code with "mpicc first.c -o first –lm.Now run "mpiexec -np 16 ./first abc.txt " .Here we need to give one command line argument i.e outputfile for rank).

To check output of a file we need to enter the command : od -td -v abc.txt

2.hw6.c

Compile the below code with : mpicc hw6.c -o hw6 –lm.

Now run mpirun -np 1 ./hw6 output1.txt 2 output.txt" , Here we need to give three command line arguments i.e outputfile1, 1 MB for Rank, and output file name for timings.