Lab 7

Create Cluster of MPI Nodes using Docker Containers

Export/Load image to/from a tar file

- docker **save** -o <path for generated **tar** file> <image name>
- docker **load** -i <path to image **tar** file>

Create Custom Container

- **-p**: set bound ports range on host.
- **--mount type**: set shared folder path in host.
- **--shm-size:** set max accessible memory.

```
docker run
--name <container_tag_name>
-p5000-5300:5000-5300 -p6000-6300:6000-6300
--mount type=bind,source=<path in host>,target=<path in container>
--shm-size <max memory in GB>
-it <image_name>
```

Exercise 01

Run Lab02 exercise 3 Server and Client under 2 containers

1. Pull docker base ubuntu images : docker pull python
2. Create docker container from the pulled image.
3. Copy **Lab02** exercise **3** files to container.
4. Run client and servers inside your container
5. Commit the container into a new image with name **clientserver-py-server-img**
6. Create 2 containers from client-server-py image
7. Run the 2 containers, 1 of them as a server and the other as a client.

Exercise 01 Sol

1. docker pull python
2. docker run --name clientServerPY_server --mount type=bind,source=<source code_path>,target=/DS_COMP6231 --shm-size 2GB **-it** python **bash**
3. apt-get update
4. apt-get install nano net-tools iputils-ping
5. docker commit clientServerPY_server clientserver-py-server-img
6. docker run --name clientServerPY_**client01** --mount type=bind,source=<source code_path>,target=/DS_COMP6231 --shm-size 30GB -it clientserver-py-server-img **bash**
7. **Ping <client_IP>**
8. Change server IP
9. Run the server and clients

Exercise 02

1. Follow the steps of exercise 1 to run the MPI `pandas_mpi.py` with a cluster of 1 container as server and 2 containers as workers.

Exercise 02 Sol

1. Pull image:
    1. Empty image: docker pull dhna/mpi4py
    2. A ready image docker pull husseinabdallah2/mpi4py-cluster:master
2. docker run --name mpi4py_node<x> --mount type=bind,source=<**Source_Code_Path**> ,target=/DS_COMP6231 -it dhna/mpi4py
3. Update your Ubuntu:
    1. Use **passwd to c**hange the password of container it will be asked during ssh
    2. apt-get update
    3. apt-get install nano net-tools iputils-ping openssh-client openssh-server
    4. service ssh start
    5. service ssh stop
4. Update pandas: pip install pandas==1.5.0
5. Enable password-less SSH between containers
    1. nano /etc/ssh/sshd_config
    2. Change the line "**PermitRootLogin yes**"
    3. service ssh start
    4. Test ssh: **ssh root@<container_ip>**
    5. At Server Side:
        1. Create **machinefile** at path ~/ with the ip-addresses of all nodes:

            <container_1_ip>
            <container_2_ip>
            …..

        2. ssh-keygen -t rsa
        3. ssh-copy-id -i ~/.ssh/id_rsa.pub root@**<container_ip>**
        4. eval 'ssh-agent'
        5. **Now Try ssh without password**
    6. mpiexec -n 3 -machinefile ~/machinefile python -m mpi4py pandas_mpi.py

Exercises 03

1. Run Matrix-Multiplication example with 1 container as master and 5 containers as workers. Compare the MPI and Serial execution Times.

Exercise 03

1. Configure your Cluster
2. mpiexec -n 6 -machinefile ~/machinefile python -m mpi4py matrix_multiplication_MPI.py