

LAPORAN PRAKTIKUM MESSAGE PASSING INTERFACE (MPI)



Disusun Oleh :

Sanvic Dicaprio (09011282227081)

Dosen Pengampu :

Adi Hermansyah, S.Kom., M.T.

**Fakultas Ilmu Komputer
Jurusan Sistem Komputer
Universitas Sriwijaya Tahun 2023-2024**

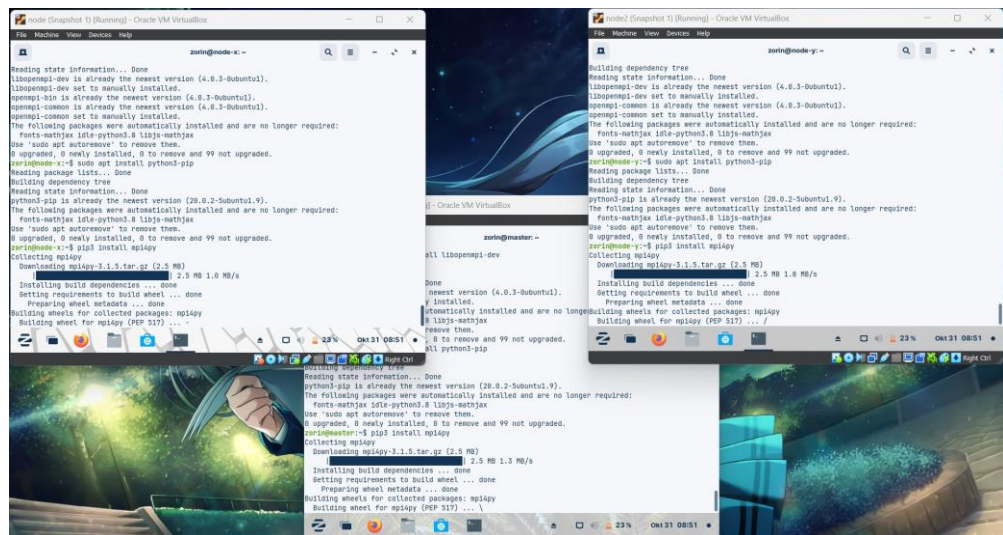
Dalam laporan praktikum ini, kami akan menjelaskan cara instalasi MPI, menghubungkan VM melalui jaringan host only dan bridged adapter via ssh, dan menjalankan program python bubble sort di MPI secara paralel via layanan ssh localhost dari openssh-server.

1. Set up MPI

A. MPI via Host Only Adapter

Jaringan *host only adapter* biasanya dipakai lebih dari satu VM yang aktif di satu komputer. Kelebihan dari host only adapter adalah fleksibilitas dan dapat terus aktif pada kondisi wifi dari komputer utama mati/nonaktif. Sebagai wadah praktikum, kami akan menggunakan ZorinOS sebagai VM utama yang telah digandakan sebanyak 2 kali. Berikut adalah cara untuk menginstall dan menjalankan pemrosesan paralel MPI via host only adapter:

- Ubah hostname pada ketiga VM dan update sistem. Install MPI dengan mengetik perintah “`sudo apt install openmpi-bin openmpi-common libopenmpi-dev`”. Install juga library python untuk MPI dengan “`pip3 install mpi4py`.”



- Jika telah terinstall, cek MPI dengan “`mpirun --version`”.

```
zorin@master:~/Desktop$ mpirun --version
mpirun (Open MPI) 4.0.3
Report bugs to http://www.open-mpi.org/community/help/
```

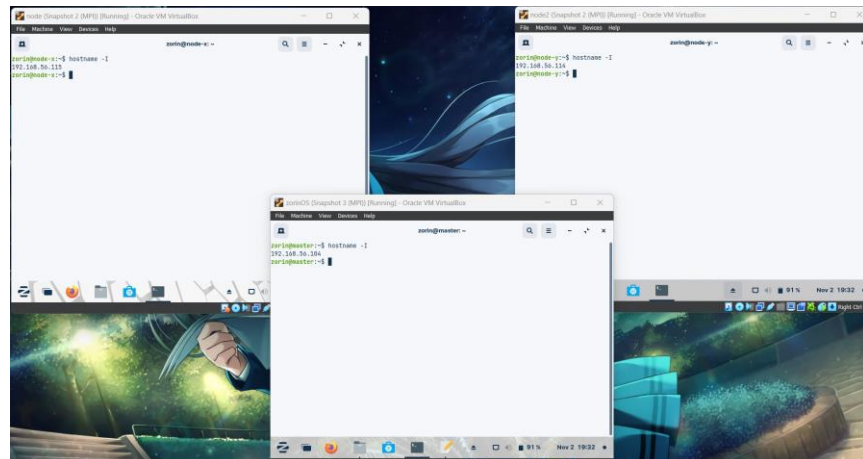
- Pastikan untuk mengecek versi mpi4py dengan perintah python dibawah ini.

```
zorin@master:~/Desktop$ python3
Python 3.8.10 (default, May 26 2023, 14:05:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import mpi4py
>>> print(mpi4py.__version__)
3.1.5
```

- Untuk mengetahui apakah mpi berjalan, ketik perintah benchmark seperti “mpixec -n <jumlah core proci> python3 -m mpi4py.bench helloworld.”.

```
zorin@master:~/Desktop$ mpixec -n 1 python3 -m mpi4py.bench helloworld
Hello, World! I am process 0 of 1 on master.
```

- Install semua mpi dan openssh-server pada semua node yang ingin terhubung. Tentukkan 1 komputer yang berperan sebagai master. Perlihatkan semua ip koneksi menggunakan perintah “hostname -I”.



- Masuk dan tambahkan ip pada masing-masing VM dengan “sudo nano /etc/hosts”.



Menambah IP pada file sistem hosts di slave_1

```
zorin@node-y: ~  
GNU nano 4.8 /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 zorin-VirtualBox  
  
192.168.56.104 master  
192.168.56.115 node-x  
192.168.56.114 node-y  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Menambah IP pada file sistem hosts di slave_2

```
zorin@master: ~  
GNU nano 4.8 /etc/hosts Modified  
127.0.0.1 localhost  
127.0.1.1 zorin-VirtualBox  
  
192.168.56.104 master  
192.168.56.115 node-x  
192.168.56.114 node-y  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Menambah IP pada file sistem hosts di master

- Simpan dan tutup semua file “/etc/hosts”. Kembali ke VM master, aktifkan ssh dan buka “ssh localhost”.

```
zorin@master: ~  
zorin@master:~$ ssh localhost  
Welcome to Zorin OS 16.3 (GNU/Linux 5.15.0-83-generic x86_64)  
  
* Website:      https://zorin.com  
* Help:         https://help.zorin.com  
  
Expanded Security Maintenance for Applications is not enabled.  
  
104 updates can be applied immediately.  
88 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Your Hardware Enablement Stack (HWE) is supported until April 2025.  
Anda telah berhasil melakukan login user  
  
Last login: Thu Nov  2 19:31:32 2023 from 192.168.56.114  
zorin@master:~$
```

- Ketik “ssh-keygen -t rsa” untuk mencetak ssh. Setelah itu, ketik perintah “ssh <user>@<hostname>” kepada semua VM. Input password dan tunggu hingga pesan konfirmasi telah muncul di terminal.

```

zorin@master:~$ ssh-copy-id zorin@master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zorin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that a
re already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote
system.
(if you think this is a mistake, you may want to use -f option)

zorin@master:~$ ssh-copy-id zorin@node-x
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zorin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that a
re already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote
system.
(if you think this is a mistake, you may want to use -f option)

zorin@master:~$ ssh-copy-id zorin@node-y
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zorin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that a
re already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote
system.
(if you think this is a mistake, you may want to use -f option)

```

- Lakukan pengujian sekali lagi pada terminal di master, uji coba apakah mpi berjalan di multi-computer dengan memasukkan perintah “mpirun -n <jumlah_komputer> -host <nama_host1>,<nama_host2>,<nama_host_lainnya> python3 -m mpi4py.bench helloworld”. Proses dapat dikatakan berhasil berjalan secara paralel apabila muncul pesan output seperti ini.

```

zorin@master:~$ mpirun -n 3 -host master,node-x,node-y python3 -m mpi4py.bench helloworld
Hello, World! I am process 0 of 3 on master.
Hello, World! I am process 1 of 3 on node-x.
Hello, World! I am process 2 of 3 on node-y.

```

B. MPI via Bridged Adapter

Bridged adapter dapat menjadi alternatif apabila ingin banyak VM yang terhubung pada komputer yang berbeda. Untuk dapat melakukan koneksi secara *bridged adapter*, pastikan koneksi jaringan baik secara *wired* maupun *wireless* sama pada komputer yang ingin dihubungkan. Kali ini, kami memakai ubuntu server yang di *remote* oleh CMD (Command Prompt). Berikut adalah cara menyiapkan MPI dengan *bridged adapter*:

- Update sistem terlebih dahulu, perlihatkan hostname pada komputer yang ingin terhubung.

```

ubuntu@ubuntu:~$ hostname -I
10.8.142.80

```

- Ketik “sudo nano /etc/hosts” untuk dapat mengedit jaringan host pada sistem yang terhubung di semua VM.


```
ubuntu@ubuntu:~$ hostname -I
10.8.142.80
ubuntu@ubuntu:~$ sudo nano /etc/hosts
[sudo] password for ubuntu:
```

Perintah untuk mengedit file sistem host

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 ubuntu

10.8.142.80 ubuntu
10.8.143.159 muhtadinserver
```

Menambahkan jaringan yang ingin disambungkan pada sistem

- Simpan dan tutup isi “/etc/hosts”. Install openssh-server untuk semua VM agar dapat dihubungkan. Setelah openssh-server diinstall, buat dan login sebagai user yang baru di semua VM. Untuk membuat user ketik “sudo adduser <nama_user>”, memberikan akses root “sudo usermod -aG sudo <nama_user>”, dan menggunakan user “su - <nama user>”.

```
ubuntu@ubuntu:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
```

Menginstall openssh-server

```
ubuntu@ubuntu:~$ su - job
Password:
```

Masuk sebagai user yang baru

- Ketik “ssh-keygen -t rsa” pada 1 VM master untuk dapat menghubungkannya dengan VM lain. Ganti direktori menjadi “/home/<nama_user>/.ssh”. dan hubungkan ssh mereka dengan mengetik “cat id_rsa.pub | ssh <nama_user>@<host> “mkdir .ssh; cat >> .ssh/authorized_keys””. Jika hasil output setelah memasukkan password “... file exists”, artinya file telah dibuat sebelumnya.

```
job@ubuntu:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/job/.ssh/id_rsa):
/home/job/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/job/.ssh/id_rsa
Your public key has been saved in /home/job/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:72v5qEhKn+MhHpBG7wsmNyCRF8BDvXo8mxTmqVmNQw0 job@ubuntu
The key's randomart image is:
+---[RSA 3072]-----+
|+oo
|+ E
|o + +
|+ B .
|o X * S
|. + & . .
|. @ X o ..
|* B B.+ .o.
|+. . .o+o.
+---[SHA256]-----+
```

Mengetik perintah kunci untuk ssh

```

job@ubuntu:~$ cd .ssh
job@ubuntu:~/.ssh$ cat id_rsa.pub | ssh job@ubuntu "mkdir .ssh; cat >> .ssh/authorized_keys"
job@ubuntu's password:
mkdir: cannot create directory '.ssh': File exists
job@ubuntu:~/.ssh$ cat id_rsa.pub | ssh job@muhtadinserver "mkdir .ssh; cat >> .ssh/authorized_keys"
job@muhtadinserver's password:
mkdir: cannot create directory '.ssh': File exists

```

Memasukkan authorized key pada semua VM

- Install MPI “sudo apt install openmpi-bin libopenmpi-dev” dan library python “pip3 install mpi4py” pada VM yang telah terhubung.

```

job@ubuntu:~$ sudo apt install openmpi-bin libopenmpi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libopenmpi-dev is already the newest version (4.1.2-2ubuntu1).
openmpi-bin is already the newest version (4.1.2-2ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
job@ubuntu:~$ sudo apt install python3-pip
pip install mpi4py
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: mpi4py in ./local/lib/python3.10/site-packages (3.1.5)

```

- Uji MPI setelah penginstalan berhasil dilakukan. Normalnya ia akan mengeluarkan output seperti dibawah ini.

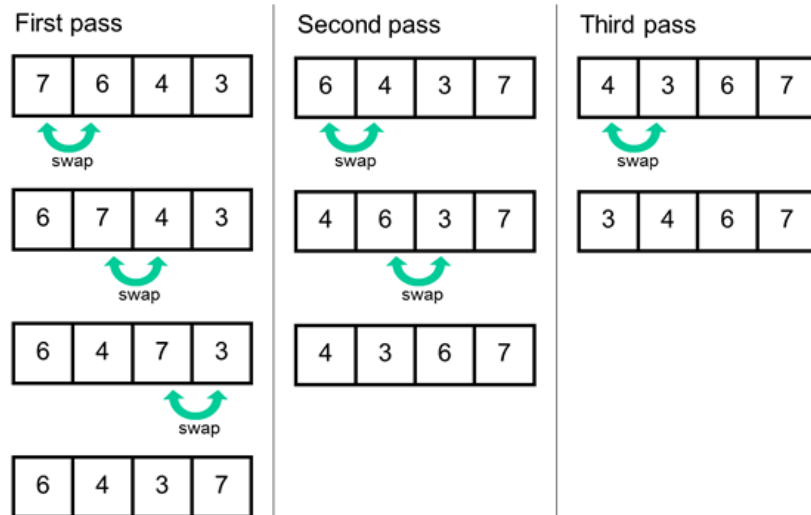
```

job@ubuntu:~$ mpirun -n 2 -host ubuntu,muhtadinserver python3 -m mpi4py.bench helloworld
Hello, World! I am process 0 of 2 on ubuntu.
Hello, World! I am process 1 of 2 on muhtadinserver.

```

2. Menjalankan program python bubble sort dengan MPI

Ada 2 program yang akan digunakan dalam menjalankan program MPI. Program tersebut menggunakan bahasa pemrograman python dengan konsep bubble sort. Bubble sort adalah kemampuan pengurutan data dengan cara menukar posisi dua nilai yang bersebelahan jika urutan data tidak sesuai.



Proses eksekusi dari bubble sort

Program pertama akan dieksekusi pada 2 kondisi jaringan, sementara program kedua hanya diuji pada 1 kondisi jaringan. Hal yang harus diperhatikan adalah kedua program harus berada dalam lokasi yang sama. Mungkin akan mudah mengakali kesamaan ini jika menggunakan *host only adapter*. Namun beda kondisi pada *bridged adapter* jika ada banyak VM tersambung dalam 1 jaringan. Untuk itu sebelum program python dijalankan, disarankan untuk dapat membuat direktori yang dapat dipakai pada semua VM.

Salah satu caranya adalah dengan menginstall NFS. Pertama, buat dulu direktori sebagai direktori bersama. Kemudian install *nfs-kernel-server* pada semua VM yang tersambung.

```
job@ubuntu:~/.ssh$ cd -
/home/job
job@ubuntu:~$ mkdir python
mkdir: cannot create directory 'python': File exists
job@ubuntu:~$ sudo apt install nfs-kernel-server
[sudo] password for job:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-kernel-server is already the newest version (1:2.6.1-1ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
```

Ketik “`sudo nano /etc/exports`” dan berikan akses pada direktori yang telah dibuat dengan `<lokasi_direktori> *(rw, sync, no_root_squash, no_subtree_check)`.


```

GNU nano 6.2 /etc/exports
# /etc/exports: the access control list for filesystems which may be export>
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no>
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/job/python *(rw,sync,no_root_squash,no_subtree_check)

```

Restart service NFS agar dapat digunakan.

```

job@ubuntu:~$ sudo exportfs -a
job@ubuntu:~$ sudo systemctl restart nfs-kernel-server
job@ubuntu:~$

```

Pada VM yang terhubung selain master, ketik “sudo mount <nama_host_master>:<direktori_master_yang _dimounting> <direktori_yang _jadi_wadah_mounting>”. Cek area mounting dengan “df -h”. jika muncul direktori master yang termounting, artinya perintah “mount” berhasil dieksekusi. Artinya setiap ada program baik dari master maupun node yang dimasukkan dalam direktori, orang lain dapat mengaksesnya melalui direktori mounting.

```

ubuntu:/home/job/python          9.8G  5.1G  4.2G  55% /home/job/python

```

Contoh program 1. Program ini akan mensortir data inputan berbentuk list yang diacak menjadi urutan data dari yang tertinggi hingga data terendah. Jika dieksekusi secara paralel, ia akan muncul sebanyak 3 kali dengan urutan list yang sama.

```

1 def bubble_sort(arr):
2     arr_len = len(arr)
3     for i in range(arr_len-1):
4         flag = 0
5         for j in range(0, arr_len-i-1):
6             if arr[j] > arr[j+1]:
7                 arr[j+1], arr[j] = arr[j], arr[j+1]
8                 flag = 1
9             if flag == 0:
10                break
11     return arr
12
13 arr = [5, 3, 4, 1, 2, 10, 83, 86, 83, 76, 64, 52, 64, 86]
14 print("List sorted with bubble sort in ascending order: ", bubble_sort(arr))

```

Isi dari program 1 di ZorinOS

```

zorin@master:~$ mpirun -n 3 -host master,node-x,node-y python3 /home/python/bubble_up.py
List sorted with bubble sort in ascending order: [1, 2, 3, 4, 5, 10, 52, 64, 64, 76, 83, 83, 86,
86]
List sorted with bubble sort in ascending order: [1, 2, 3, 4, 5, 10, 52, 64, 64, 76, 83, 83, 86,
86]
List sorted with bubble sort in ascending order: [1, 2, 3, 4, 5, 10, 52, 64, 64, 76, 83, 83, 86,
86]

```

Output yang didapat setelah di eksekusi secara paralel melalui host only adapter

```
def bubble_sort(arr):
    arr_len = len(arr)
    for i in range(arr_len-1):
        flag = 0
        for j in range(0, arr_len-i-1):
            if arr[j] > arr[j+1]:
                arr[j+1], arr[j] = arr[j], arr[j+1]
                flag = 1
            if flag == 0:
                break
    return arr

arr = [5, 3, 4, 1, 2]
print("List sorted with bubble sort in ascending order: ", bubble_sort(arr))
```

Isi dari program 1 di Ubuntu Server

```
job@ubuntu:~$ mpirun -n 2 -host ubuntu,muhtadinserver python3 /home/job/python/cobain.py
List sorted with bubble sort in ascending order: [1, 2, 3, 4, 5]
List sorted with bubble sort in ascending order: [1, 2, 3, 4, 5]
```

Output yang didapat setelah di eksekusi secara paralel via bridged adapter

Contoh program 2. Program ini akan menerima inputan berupa angka. Pengguna akan menentukan berapa banyak elemen yang akan diinput. Kemudian, pengguna akan memasukkan data tersebut hingga akhir. Contoh jika elemennya adalah 8, maka pengguna harus memasukkan inputan angka hingga 8 elemen. Begitu juga seterusnya. Elemen yang diinput akan disortir dan ditampilkan dalam bentuk list sesuai dengan urutan dari angka terendah hingga angka tertinggi.

```
1 from mpi4py import MPI
2 import numpy as np
3
4 def parallel_bubble_sort(data, comm):
5     rank = comm.Get_rank()
6     size = comm.Get_size()
7
8     local_data = np.array_split(data, size)[rank]
9
10    n = len(local_data)
11    for i in range(n):
12        for j in range(0, n - i - 1):
13            if local_data[j] > local_data[j + 1]:
14                local_data[j], local_data[j + 1] = local_data[j + 1], local_data[j]
15
16    sorted_data = comm.gather(local_data, root=0)
17
18    if rank == 0:
19        merged_data = np.concatenate(sorted_data)
20        merged_data.sort()
21        return merged_data
22    else:
23        return None
24
25 if __name__ == "__main__":
26     comm = MPI.COMM_WORLD
27     rank = comm.Get_rank()
28
29     if rank == 0:
30         try:
31             num_elements = int(input("Masukkan jumlah elemen dalam array: "))
32             data = []
33             for i in range(num_elements):
34                 element = int(input(f"Masukkan elemen array bilangan {i + 1}: "))
35                 data.append(element)
36         except ValueError:
37             print("Input harus berupa angka.")
38             exit(1)
39
40     else:
41         data = None
42
43     data = comm.bcast(data, root=0)
44
45     sorted_data = parallel_bubble_sort(data, comm)
46
47     if rank == 0:
48         print("Array yang diurutkan:", sorted_data)
```

Isi codingan program 2 (1)

Isi codingan program 2 (2)

```
zorin@master: ~/Desktop
zorin@master:~/Desktop$ mpiexec -n 3 -host master,node-X,node-Y python3 /home/python/cobainó.py
Masukkan jumlah elemen dalam array: 6
Masukkan elemen array bilangan 1: 45
Masukkan elemen array bilangan 2: 65
Masukkan elemen array bilangan 3: 76
Masukkan elemen array bilangan 4: 54
Masukkan elemen array bilangan 5: 32
Masukkan elemen array bilangan 6: 45
Array yang diurutkan: [32 45 45 54 65 76]
zorin@master:~/Desktop$
```

Output program 2 setelah dijalankan secara paralel