

Arsitektur & Organisasi Komputer

“ OPEN MPI (Parallel Computing)”



Dosen Pembimbing :

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

Disusun Oleh :

Prita Salma (09030582226036)

Fakultas Ilmu Komputer

Program Studi Teknik Komputer

UNIVERSITAS SRIWIJAYA 2022/2023

OPEN MPI (Parallel Computing)

1.1 ANGGOTA KELOMPOK :

- Satryo Pangestu (09030582226008) : MASTER
- Prita Salma (09030582226036) : SLAVE1
- Dwi Aurelia (09030582226038) : SLAVE2
- Azza Adliyah (09030582226040) : SLAVE3

1.2 TUJUAN

1. Menganalisis Konsep Open MPI
2. Mengevaluasi Kinerja Parallel Computing dengan Open MPI
3. Mengidentifikasi Aplikasi dan Kasus Penggunaan Open MPI

1.3 MANFAAT

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1.2.1 Kontribusi pada Pemahaman Konsep Komputasi Paralel

Memberikan kontribusi pada pemahaman umum tentang konsep komputasi paralel, khususnya melalui eksplorasi mendalam terhadap Open MPI sebagai salah satu platform yang relevan.

1.2.2 Panduan Implementasi Efektif Open MPI

Menyediakan panduan praktis untuk implementasi Open MPI secara efektif, membantu pemrogram dan peneliti dalam memahami cara mengoptimalkan kinerja sistem paralel.

1.2.3 Informasi Penting untuk Pengembangan Aplikasi

Memberikan informasi yang penting bagi pengembang aplikasi untuk memahami potensi dan batasan Open MPI, sehingga dapat mengembangkan aplikasi yang sesuai dengan prinsip-prinsip komputasi paralel.

1.4 DASAR TEORI

Open MPI, singkatan dari Open Message Passing Interface, adalah perangkat lunak yang dirancang untuk mendukung komputasi paralel dan distribusi tugas di lingkungan komputasi berskala besar. Konsep dasar komputasi paralel melibatkan pemecahan tugas besar menjadi bagian-bagian kecil yang dapat dijalankan secara bersamaan. Open MPI menawarkan model pemrograman paralel yang fleksibel, memungkinkan pengembang untuk mengimplementasikan kode yang dapat dijalankan secara efisien di beberapa prosesor.

Open MPI didasarkan pada arsitektur pesan-pengiriman yang memungkinkan proses berkomunikasi dan berkoordinasi satu sama lain untuk menyelesaikan tugas secara bersamaan. Dengan memanfaatkan model ini, Open MPI dapat diterapkan dalam berbagai konteks, mulai dari penelitian ilmiah dan simulasi hingga pengolahan data besar dan pemrosesan paralel di industri. Selain itu, dasar teori juga mencakup konsep-konsep fundamental seperti pengelolaan sumber daya, pengiriman pesan, dan sinkronisasi proses. Pemahaman mendalam terhadap konsep ini menjadi kunci dalam memahami cara Open MPI beroperasi dan bagaimana pengembang dapat memanfaatkannya secara optimal. Dengan dasar teori yang kokoh ini, penelitian ini akan menjelajahi implementasi Open MPI sebagai solusi yang efektif untuk mengatasi tantangan komputasi paralel dalam lingkungan modern.

1.5 PRATIKUM

Berikut ini Langkah pengerjaannya :

TUTORIAL DOWNLOAD PYTHON DARI UBUNTU

1.

```
Last login: Wed Nov 15 01:32:53 2023 from 10.1.10.26
prita@prita:~$ sudo apt-get update
[sudo] password for prita:
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
```

2.

```
prita@prita:~$ sudo apt install python3.9 python3.9-dev python3-
dev python3.9-
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

3.

```
Successfully installed mpi4py 3.1.0
(ENV3) prita@prita:~$ mpiexec -n 4 python -m mpi4py.bench hellow
orld
Hello, World! I am process 0 of 4 on prita.
Hello, World! I am process 1 of 4 on prita.
Hello, World! I am process 2 of 4 on prita.
Hello, World! I am process 3 of 4 on prita.
(ENV3) prita@prita:~$ nano prita.py
(ENV3) prita@prita:~$ python3 prita.py
Sorted array is:
11 12 22 25 34 64 90 (ENV3) prita@prita:~$ |
```

4.

```
1 $ sudo apt-get update
2 $ sudo apt install python3.9 python3.9-dev python3-dev python3.9-
  venv python3.8-venv
3 $ python3 -m venv ~/ENV3
4 $ source ENV3/bin/activate
5 (ENV3) $ sudo apt-get install -y mpich-doc mpich
6 (ENV3) $ pip install mpi4py -U
```

5.

```
prita@prita:~$ sudo apt install python3.9
```

6.

```
prita@prita:~$ sudo apt install python3.9-dev
```

7.

```
prita@prita:~$ sudo apt install python3-dev
```

8.

```
prita@prita:~$ sudo apt install python3.9-venv
```

9.

```
prita@prita:~$ sudo apt install python3.8-venv
```

10.

```
prita@prita:~$ python3 -m venv ~/ENV3
prita@prita:~$ source ENV3/bin/activate
(ENV3) prita@prita:~$ sudo apt-get install -y mpich-doc mpich
Reading package lists... Done
Building dependency tree
Reading state information... Done
mpich is already the newest version (3.3.2-2build1).
mpich-doc is already the newest version (3.3.2-2build1).
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
```

11.

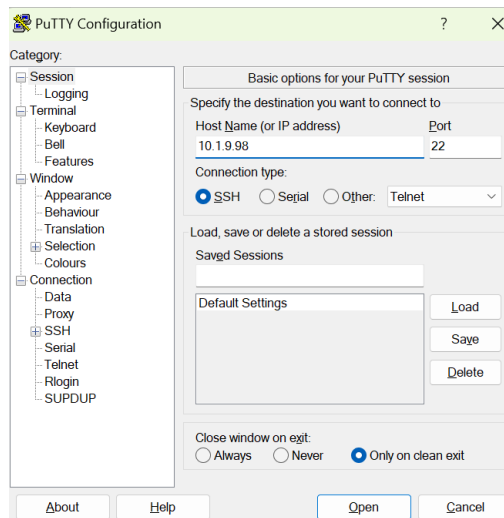
```
(ENV3) prita@prita:~$ p install mpi4py -U
p: command not found
(ENV3) prita@prita:~$ pip install mpi4py -U
Collecting mpi4py
```

12.

```
Successfully built mpi4py
Installing collected packages: mpi4py
Successfully installed mpi4py-3.1.5
(ENV3) prita@prita:~$ mpiexec -n 4 python -m mpi4py.bench hellow
orld
Hello, World! I am process 0 of 4 on prita.
Hello, World! I am process 1 of 4 on prita.
Hello, World! I am process 2 of 4 on prita.
Hello, World! I am process 3 of 4 on prita.
(ENV3) prita@prita:~$ nano prita.py
(ENV3) prita@prita:~$ python3 prita.py
Sorted array is:
11 12 22 25 34 64 90 (ENV3) prita@prita:~$
```

TUTORIAL MENJADI SLAVE

1.



2.

```
login as: prita
prita@10.1.9.98's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-166-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue 21 Nov 2023 08:28:26 AM UTC

System load:  0.0               Processes:    146
Usage of /:   33.1% of 11.21GB   Users logged in: 1
Memory usage: 3%               IPv4 address for enp0s3: 10.1.9.98
Swap usage:   0%
```

3.

```
Last login: Tue Nov 21 07:53:22 2023 from 10.1.9.244
prita@prita:~$ sudo apt-get update
[sudo] password for prita:
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 114 kB in 1s (82.6 kB/s)
Reading package lists... Done
prita@prita:~$ sudo adduser prita
adduser: The user `prita' already exists.
prita@prita:~$ sudo usermod -aG sudo prita
prita@prita:~$ su - prita
Password:
```

4.

```
done
prita@prita:~$ sudo apt install net-tools vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

5.

```
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
prita@prita:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:9.2p1-4ubuntu0.0).
```

6.

```
prita@prita:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/prita/.ssh/id_rsa): /home/prita/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/prita/.ssh/id_rsa
Your public key has been saved in /home/prita/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ruYEhK4vLrKIwbPL3n6+kZWVLzSWdOte4VCj41TAt8c prita@prita
The key's randomart image is:
+---[RSA 3072]-----+
|      ..o.+      |
|      . +.=..    |
|      . .  B *.o  |
|      . .  = * +..E |
|      . .  oS. + o. |
|..   .o. o .      |
|oo   o. . .       |
|Bo+  oo.          |
|OO+oo=+          |
+---[SHA256]-----+
```

7.

```
-bash: /home/prita/bubble: Is a directory
prita@prita:~/ssh$ sudo exportfs -a
prita@prita:~/ssh$ sudo systemctl restart nfs-kernel-server
prita@prita:~/ssh$ sudo apt install nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version (1:1.3.4-2.5ubuntu3.5).
nfs-common set to manually installed.
The following packages were automatically installed and are no longer required:
linux-headers-5.4.0-139 linux-headers-5.4.0-139-generic
linux-image-5.4.0-139-generic linux-modules-5.4.0-139-generic
linux-modules-extra-5.4.0-139-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

8.

```
prita@prita:~/ssh$ sudo mount master:/home/prita/bubble /home/prita/bubble
prita@prita:~/ssh$ sudo apt install openmpi-bin libopenmpi-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
linux-headers-5.4.0-139 linux-headers-5.4.0-139-generic
```

9.

```
prita@prita:~/.ssh$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.10).
The following packages were automatically installed and are no longer required:
linux-headers-5.4.0-139 linux-headers-5.4.0-139-generic
linux-image-5.4.0-139-generic linux-modules-5.4.0-139-generic
linux-modules-extra-5.4.0-139-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
prita@prita:~/.ssh$ pip install mpi4py
Processing /home/prita/.cache/pip/wheels/56/8b/e8/e2ac5b75cacca356a2f270568f7
ac5b6c77c6aa223bf8d2bca/mpi4py-3.1.5-cp38-cp38-linux_x86_64.whl
Installing collected packages: mpi4py
Successfully installed mpi4py-3.1.5
```

10.

```
prita@prita: ~/bubble
GNU nano 4.8 bubble.py Modified
from mpi4py import MPI

def parallel_bubble_sort(arr):
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()
    size = comm.Get_size()

    local_arr = arr[rank::size]

    for i in range(len(local_arr)):
        for j in range(0, len(local_arr) - i - 1):
            if local_arr[j] > local_arr[j + 1]:
                local_arr[j], local_arr[j + 1] = local_arr[j + 1], local_arr[j]

    sorted_arr = comm.gather(local_arr, root=0)

    if rank == 0:
        combined_arr = [item for sublist in sorted_arr for item in sublist]
        combined_arr.sort()
        for i in range(len(arr)):
            arr[i] = combined_arr[i]

if __name__ == '__main__':
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()

    if rank == 0:
        arr = [5, 3, 4, 1, 2]
    else:
        arr = None

    arr = comm.bcast(arr, root=0) # Broadcast the arr from rank 0 to all nodes
    comm.barrier()

    parallel_bubble_sort(arr)

    if rank == 0:
        print(f"List sorted with bubble sort in ascending order: {arr}")

File Name to Write: bubble.py
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

11.

```
prita@prita:~/.ssh$ touch /home/prita/bubble/bubble.py
prita@prita:~/.ssh$ cd /home/prita/bubble
prita@prita:~/bubble$ nano bubble.py
```

```

prita@prita:~/bubble$ mpirun -np 4 -host master,slave1,slave2,slave3 python3
bubble.pympirun -np 4 -host master,slave1,slave2,slave3 python3 bubble.py
ssh: Could not resolve hostname master: Temporary failure in name resolution
ssh: Could not resolve hostname slave1: Temporary failure in name resolution
-----
ORTE was unable to reliably start one or more daemons.
This usually is caused by:

* not finding the required libraries and/or binaries on
  one or more nodes. Please check your PATH and LD_LIBRARY_PATH
  settings, or configure OMPI with --enable-orterun-prefix-by-default

* lack of authority to execute on one or more specified nodes.
  Please verify your allocation and authorities.

* the inability to write startup files into /tmp (--tmpdir/orte_tmpdir_base).
  Please check with your sys admin to determine the correct location to use.

* compilation of the orted with dynamic libraries when static are required
  (e.g., on Cray). Please check your configure cmd line and consider using
  one of the contrib/platform definitions for your system type.

* an inability to create a connection back to mpirun due to a
  lack of common network interfaces and/or no route found between
  them. Please check network connectivity (including firewalls
  and network routing requirements).
-----
ORTE does not know how to route a message to the specified daemon
located on the indicated node:

my node:   prita
target node: slave1

This is usually an internal programming error that should be
reported to the developers. In the meantime, a workaround may
be to set the MCA param routed=direct on the command line or
in your environment. We apologize for the problem.
-----

```

1.6 ANALISIS

Open MPI sebagai platform komputasi paralel, beberapa aspek kunci menjadi pusat perhatian. Pertama, pengkajian konsep dasar Open MPI mengungkapkan keunggulan sistem ini dalam mendukung komputasi paralel. Model pemrograman yang fleksibel dan kemampuan untuk mengelola tugas paralel membuatnya relevan dalam lingkungan komputasi modern.

Selanjutnya, evaluasi kinerja Open MPI mengarah pada pemahaman mendalam tentang sejauh mana platform ini mampu meningkatkan kecepatan dan efisiensi komputasi paralel. Analisis benchmark dan studi kasus memungkinkan kita untuk memahami potensi pengoptimalan kinerja serta mengidentifikasi situasi di mana Open MPI menunjukkan keunggulan. Dalam konteks aplikasi, penelitian ini mencakup identifikasi kasus penggunaan Open MPI yang praktis dan relevan. Dari kebutuhan ilmu pengetahuan hingga aplikasi industri, analisis ini menyoroti fleksibilitas Open MPI dalam mendukung berbagai aplikasi komputasi paralel.

Selain itu, analisis ini melibatkan penelusuran dampak implementasi Open MPI pada pengembangan aplikasi. Kelebihan dan batasan Open MPI dalam konteks ini dibahas, memberikan wawasan yang berguna bagi pengembang dalam menghadapi tantangan dalam mengoptimalkan kinerja aplikasi mereka. Secara keseluruhan, analisis ini menyajikan pemahaman holistik tentang potensi dan kendala Open MPI dalam konteks komputasi paralel. Dengan memahami aspek-aspek ini, kita dapat merancang implementasi yang efektif dan mengambil manfaat maksimal dari kemampuan komputasi paralel yang ditawarkan oleh Open MPI.

1.7 KESIMPULAN

Secara keseluruhan, penelitian ini menggambarkan bahwa Open MPI memiliki peran yang signifikan dalam menghadirkan solusi efisien untuk komputasi paralel. Analisis mendalam konsep dasar, kinerja, dan aplikasi Open MPI mengungkapkan bahwa platform ini bukan hanya mendukung model pemrograman paralel yang fleksibel, tetapi juga memberikan peningkatan kinerja yang nyata dalam eksekusi tugas paralel. Melalui evaluasi kinerja, terlihat bahwa Open MPI mampu mengoptimalkan penggunaan sumber daya secara efisien, merangsang peningkatan kecepatan dan efisiensi komputasi. Kasus penggunaan yang diidentifikasi memperkuat aplikabilitas Open MPI dalam berbagai bidang, menunjukkan keberagaman dan fleksibilitasnya.

Kesimpulannya, implementasi Open MPI memberikan kontribusi positif dalam mendukung perkembangan komputasi paralel. Kelebihan platform ini dalam manajemen tugas paralel, fleksibilitas pemrograman, dan kinerja tinggi menjadikannya pilihan yang layak untuk proyek-proyek yang memerlukan pengolahan data besar dan tugas-tugas komputasi intensif. Dengan demikian, Open MPI bukan hanya merupakan alat yang kuat untuk pengembang dan peneliti, tetapi juga berpotensi menggerakkan evolusi lebih lanjut dalam paradigma komputasi paralel.