INDONESIAN E-HEALTH GRID MODEL TEST BED ON GRID LIPI MACHINE

Angga Pradipta
Sri Chusri Haryanti
Ummi Azizah Rachmawati

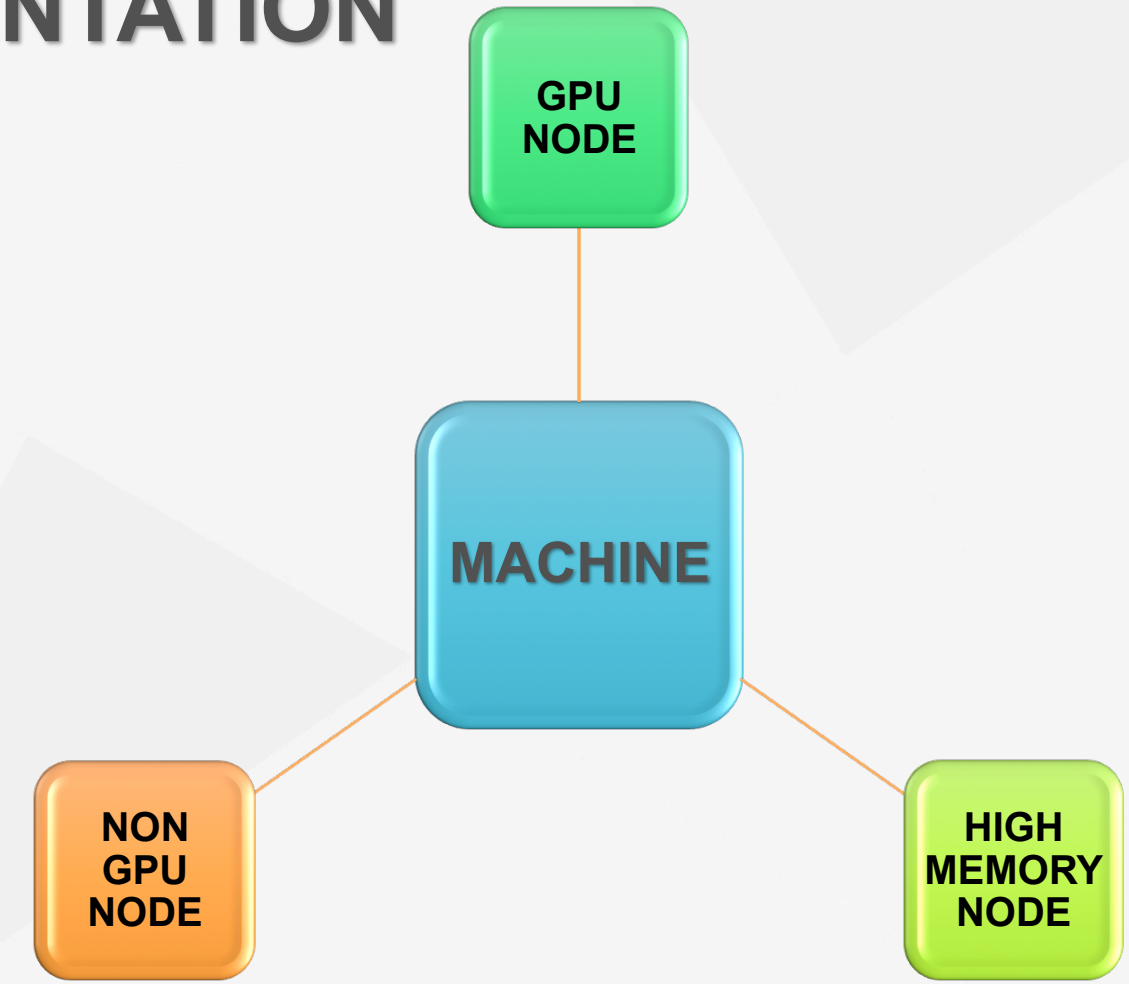
PRAGMA 34
May 11, 2018
Akihabara, Tokyo

Previous Work

- Inter-Hospital Network Design in Indonesia Using the NS-3 Simulator. (Yunan, 2015)
- Software Defined Network for Inter-Hospital Network in Indonesia. (Agung, 2016)
- Designing E-Health Grid Using OpenFlow for Hospitals in Indonesia by Province. (Azzahra, 2017)
- Indonesian E-Health Grid Model Test Bed on Grid LIPI Machine

GRID LIPI MACHINE FOR **E-HEALTH GRID IMPLEMENTATION**

Grid LIPI provides facilities to the researchers to use Grid machines with specifications that fit the research needs. In the LIPI Grid engine, this research uses GPU Node, non GPU Node and High Memory Node.

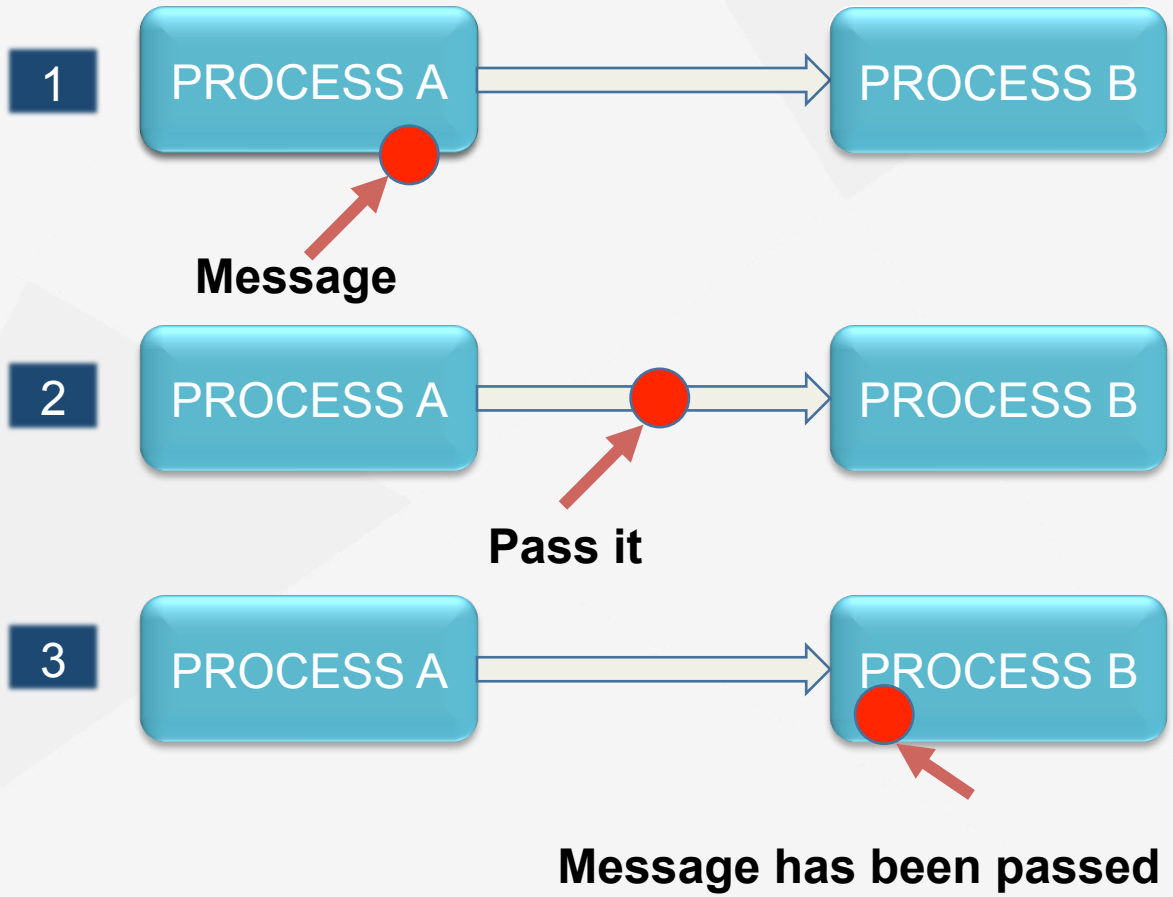


LIPI

http://grid.lipi.go.id/

Message Passing Interface (MPI)

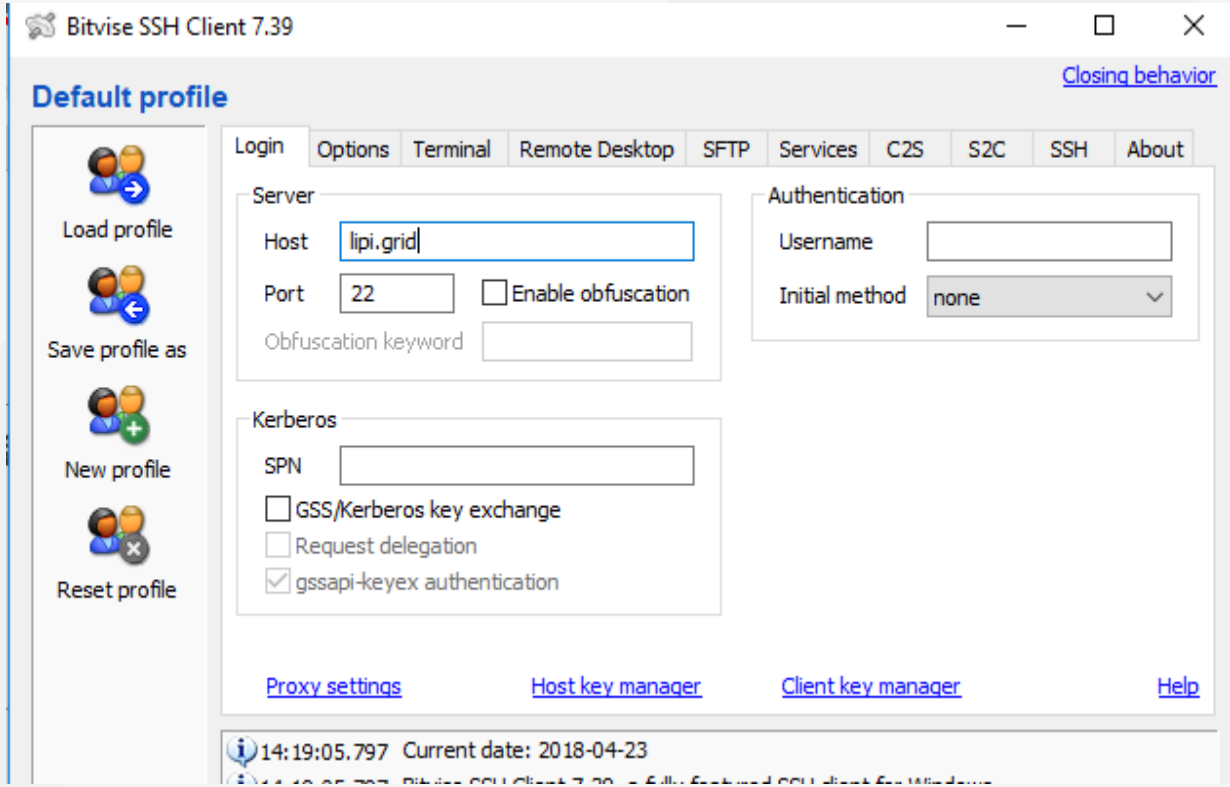
The LIPI Grid engine is one of the best options for implementing message passing interfaces (MPI) using a parallel system. MPI is one of the Application Programming Interface (API) that can support High Performance Computing (HPC).



SOFTWARE

Implementation using Bitvise as SSH Client to connect to Grid system

Default configuration settings



HARDWARE : **CIBINONG SITE**

Basic Nodes:

80 node
2 processors per node, 4 cores per processors
Dual Intel Xeon E5-2609 2,4 GHz
8 GB RAM DDR3-1600
500 GB HD SATA
Dual Gigabit interconnection
Linux (CentOS)

High Memory Node:

8 Node
2 processors per node, 8 cores per processors
Dual Intel Xeon E5-2640 2 GHz
256 GB RAM DDR3-1600
2 x 300 GB HD SAS
Dual Gigabit interconnection
Linux (CentOS)

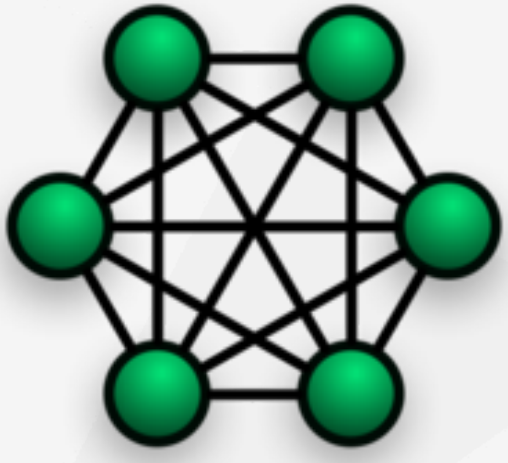
GPU Node:

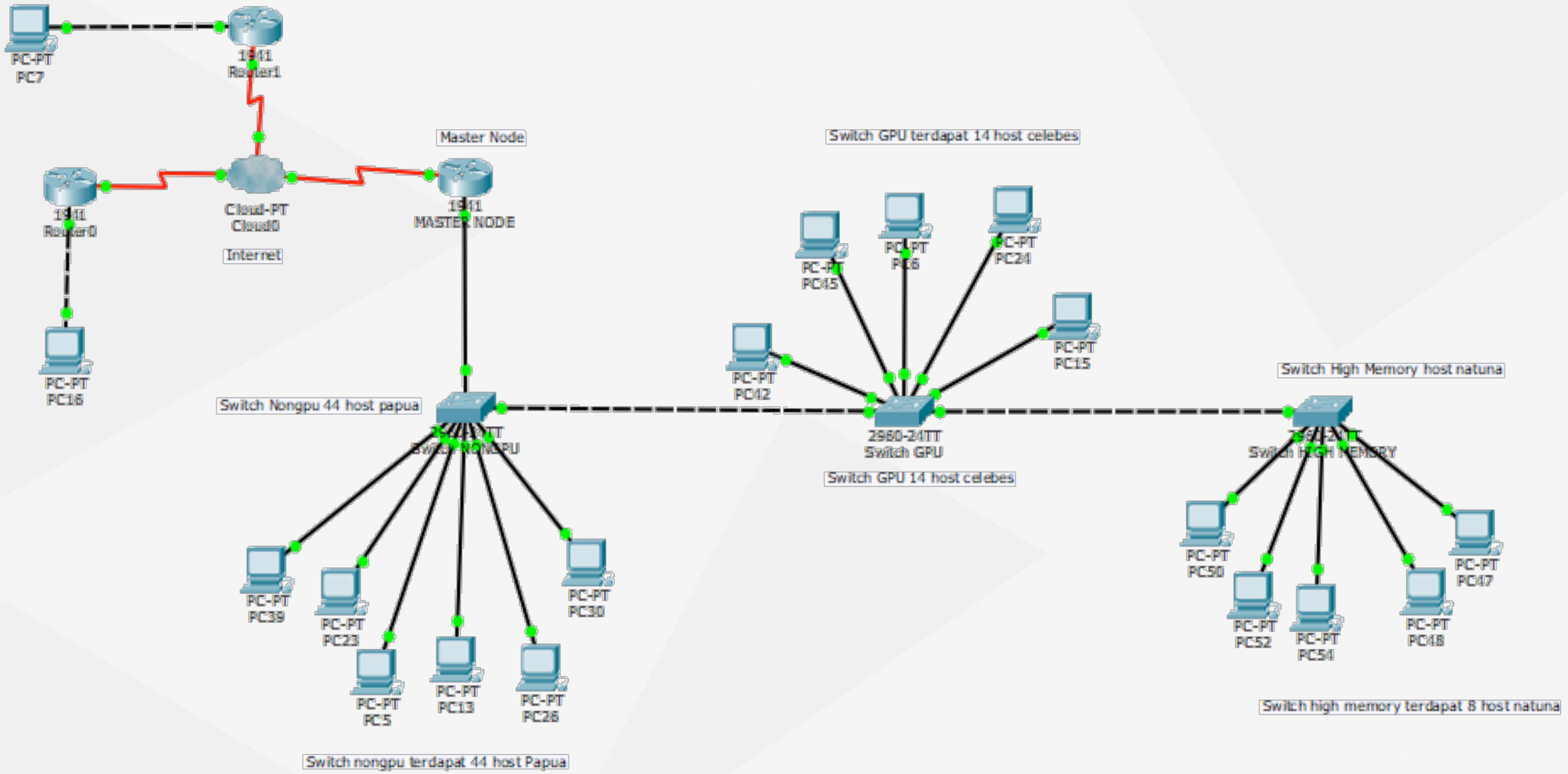
16 node
2 processors per node, 4 cores per processors
Dual Intel Xeon E5-2609 2,4 GHz
8 GB RAM DDR3-1600
500 GB HD SATA
Dual Gigabit interconnection
NVIDIA Tesla M2075 GPGPU
Linux (CentOS)

Master Node:

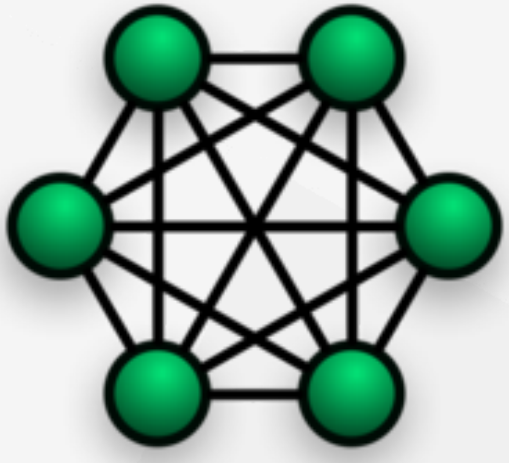
2 node
2 processors per node, 8 cores per processors
Dual Intel Xeon E5-2650 2,0 GHz
128 GB RAM DDR3-1600
24 TB HD SATA (Raw), RAID 5
Dual 10 Gigabit interconnection
Linux (CentOS)

TOPOLOGY DESIGN

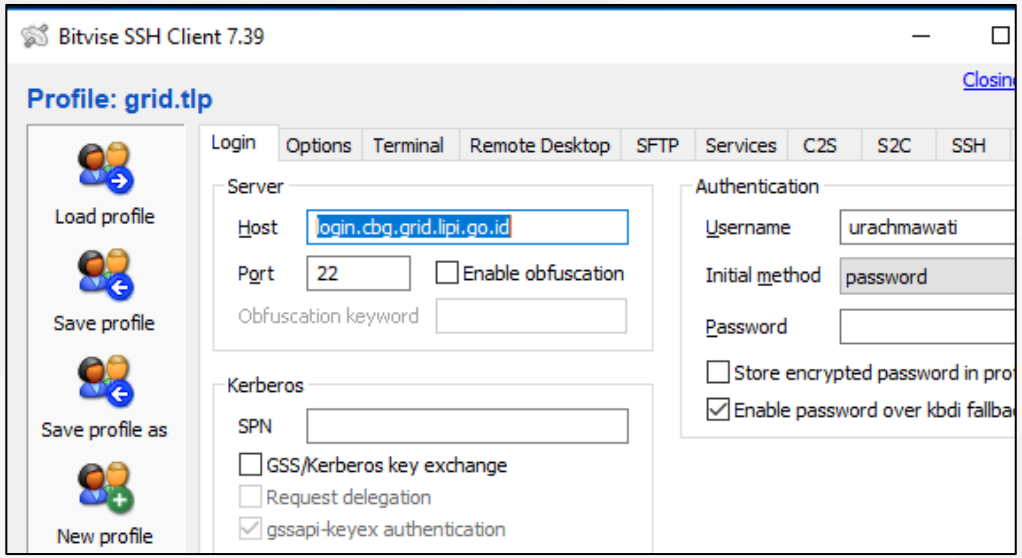


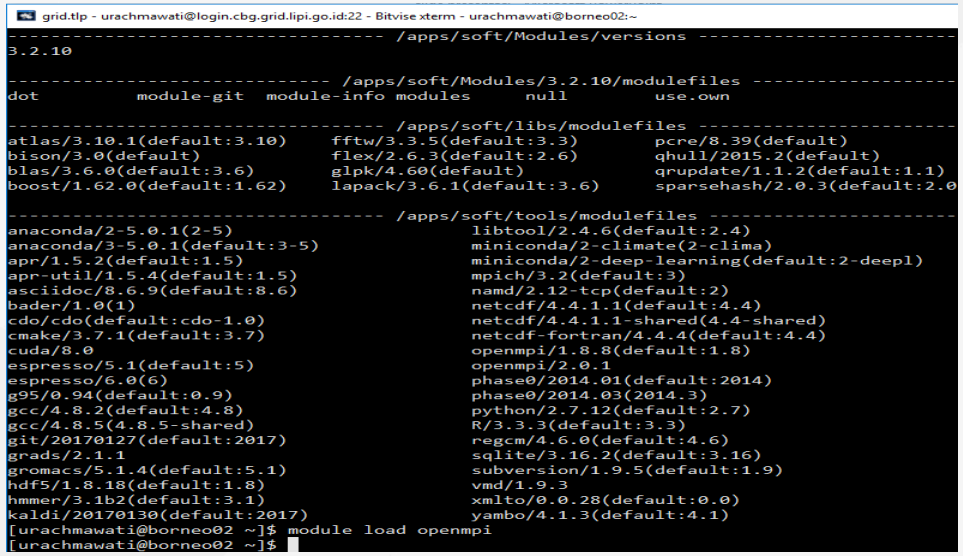

PC-PT
PC7
1941
Router1
Master Node
Switch GPU terdapat 14 host celebes
1941
Router0
Cloud-PT
Cloud0
Internet
1941
MASTER NODE
PC-PT
PC16
PC-PT
PC15
PC-PT
PC6
PC-PT
PC24
PC-PT
PC15
PC-PT
PC12
Switch High Memory host natuna
Switch Nongpu 44 host papua
2960-24TT
Switch GPU
Switch GPU 14 host celebes
PC-PT
PC39
PC-PT
PC23
PC-PT
PC30
PC-PT
PC5
PC-PT
PC13
PC-PT
PC26
PC-PT
PC50
PC-PT
PC47
PC-PT
PC52
PC-PT
PC54
PC-PT
PC48
Switch nongpu terdapat 44 host Papua
Switch high memory terdapat 8 host natuna

IMPLEMENTATION



STAGE OF IMPLEMENTATION





- Login to the grid system
- SSH master

- Module avail (to view all of module list)
- Module load openmpi

CREATE PBS FILE

- Nano run.pbs

```
#!/bin/bash

### This is parameter for job management
### Don't remove '#' sign before PBS
### The '###' is a comment

### PBS Parameters
### -N  Job's name
### -o  Output name
### -e  Error name
### -q  queue name : gpu or nongpu
### -l  resource
### Example:

#PBS -N "hello"
#PBS -q nongpu
#PBS -l walltime=04:00:00
#PBS -l nodes=8:ppn=1
```

```
### load  module you need
### you can check the available module by command: module avail

module load openmpi

### Call your script/code
### Example:

echo "Hostname = $HOSTNAME"
echo "Number of nodes = $NP"
echo "Start = `date`"

mpirun ../hello

echo "Finish = `date`"
```

CREATE **PROCESSING TIME** FILE .C

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char *argv[]) {
   double time1, time2,duration,global;
   int npes, myrank;
   MPI_Init(&argc, &argv);
   time1 = MPI_Wtime();

  /*Deklarasi rank proses */
    MPI_Comm_size(MPI_COMM_WORLD, &npes);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
  //printf("From process %d out of %d, Hello World!\n", myrank, npes);


   MPI_Barrier(MPI_COMM_WORLD);
   time2 = MPI_Wtime();
   duration = time2 - time1;
   MPI_Reduce(&duration,&global,1,MPI_DOUBLE,MPI_MAX,0,MPI_COMM_WORLD);
```

```
   char processor_name[MPI_MAX_PROCESSOR_NAME];
   int name_len;
   MPI_Get_processor_name(processor_name, &name_len);

   printf("Waktu proses : processor %s at %d is %f \n", processor_name,myrank,duration);
   MPI_Finalize();
   return 0;
}
```

COMPILE **RUNTIME.C**

```
[urachmawati@borneo02 prosestime]$ ls
4  8  run.pbs  runtime.c  test
[urachmawati@borneo02 prosestime]$ mpicc runtime.c -o runtime
[urachmawati@borneo02 prosestime]$ ls
4  8  run.pbs  runtime  runtime.c  test
[urachmawati@borneo02 prosestime]$
```

SUBMIT **RUN.PBS** TO THE GRID SYSTEM

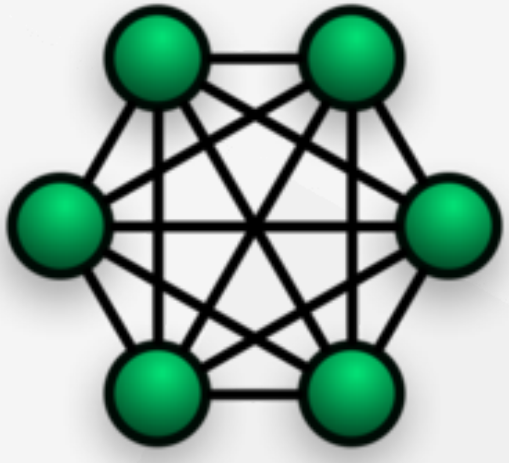
```
[urachmawati@borneo02 prosestime]$ qsub run.pbs
15317.borneo02.cbg.grid.lipi.go.id
[urachmawati@borneo02 prosestime]$
```

THE **RESULT**

```
[urachmawati@borneo02 prosestime]$ ls
4  8  8.e15317  8.o15317  run.pbs  runtime  runtime.c  test
[urachmawati@borneo02 prosestime]$
```

```
[urachmawati@borneo02 prosestime]$ cat 8.o15317
Hostname = papua01.cbg.grid.lipi.go.id
Number of nodes = 4
Start = Tue Apr 24 00:57:07 WIB 2018
Waktu proses : processor papua01.cbg.grid.lipi.go.id at 0 is 0.000091
Waktu proses : processor papua01.cbg.grid.lipi.go.id at 1 is 0.000083
Waktu proses : processor papua01.cbg.grid.lipi.go.id at 2 is 0.000113
Waktu proses : processor papua01.cbg.grid.lipi.go.id at 3 is 0.000044
Finish = Tue Apr 24 00:57:07 WIB 2018
[urachmawati@borneo02 prosestime]$
```

TESTING



HOSTS GPU

```
# Local IP for worker node gpgpu
10.100.3.1      celebes01.cbg.grid.lipi.go.id          celebes01
10.100.3.2      celebes02.cbg.grid.lipi.go.id          celebes02
10.100.3.3      celebes03.cbg.grid.lipi.go.id          celebes03
10.100.3.4      celebes04.cbg.grid.lipi.go.id          celebes04
10.100.3.5      celebes05.cbg.grid.lipi.go.id          celebes05
10.100.3.6      celebes06.cbg.grid.lipi.go.id          celebes06
10.100.3.7      celebes07.cbg.grid.lipi.go.id          celebes07
10.100.3.8      celebes08.cbg.grid.lipi.go.id          celebes08
10.100.3.9      celebes09.cbg.grid.lipi.go.id          celebes09
10.100.3.10     celebes10.cbg.grid.lipi.go.id          celebes10
10.100.3.11     celebes11.cbg.grid.lipi.go.id          celebes11
10.100.3.12     celebes12.cbg.grid.lipi.go.id          celebes12
10.100.3.13     celebes13.cbg.grid.lipi.go.id          celebes13
10.100.3.14     celebes14.cbg.grid.lipi.go.id          celebes14
```

HOSTS NON GPU

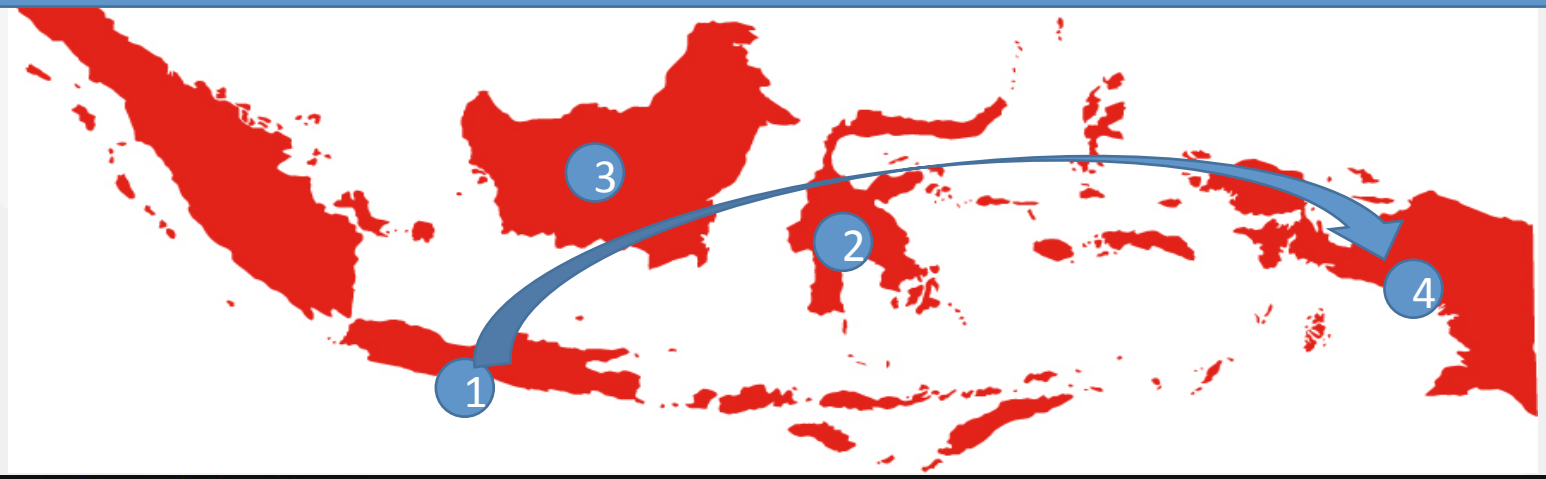
```
# Local IP for worker node nogpgpu
10.100.2.1      papua01.cbg.grid.lipi.go.id          papua01
10.100.2.2      papua02.cbg.grid.lipi.go.id          papua02
10.100.2.3      papua03.cbg.grid.lipi.go.id          papua03
10.100.2.4      papua04.cbg.grid.lipi.go.id          papua04
10.100.2.5      papua05.cbg.grid.lipi.go.id          papua05
10.100.2.6      papua06.cbg.grid.lipi.go.id          papua06
10.100.2.7      papua07.cbg.grid.lipi.go.id          papua07
10.100.2.8      papua08.cbg.grid.lipi.go.id          papua08
10.100.2.9      papua09.cbg.grid.lipi.go.id          papua09
10.100.2.10     papua10.cbg.grid.lipi.go.id          papua10
10.100.2.11     papua11.cbg.grid.lipi.go.id          papua11
10.100.2.12     papua12.cbg.grid.lipi.go.id          papua12
10.100.2.13     papua13.cbg.grid.lipi.go.id          papua13
10.100.2.14     papua14.cbg.grid.lipi.go.id          papua14
```

HOSTS HIGH MEMORY

```
# Local IP for high memory node
10.100.6.1      natuna01.cbg.grid.lipi.go.id          natuna01
10.100.6.2      natuna02.cbg.grid.lipi.go.id          natuna02
10.100.6.3      natuna03.cbg.grid.lipi.go.id          natuna03
10.100.6.4      natuna04.cbg.grid.lipi.go.id          natuna04
10.100.6.5      natuna05.cbg.grid.lipi.go.id          natuna05
10.100.6.6      natuna06.cbg.grid.lipi.go.id          natuna06
10.100.6.7      natuna07.cbg.grid.lipi.go.id          natuna07
10.100.6.8      natuna08.cbg.grid.lipi.go.id          natuna08
```

SCENARIO TESTING

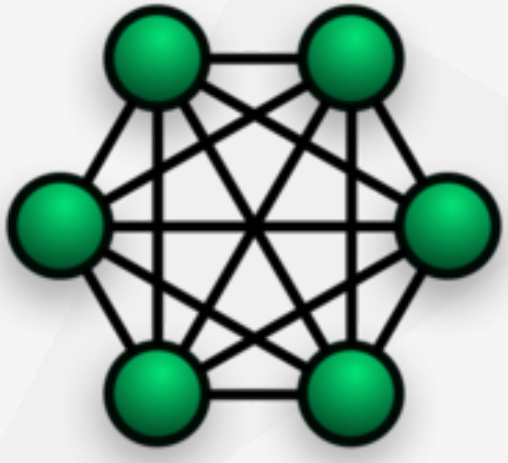
No	METHOD	TYPES OF HOST	PARAMATER
1	Vertical	Nongpu	Processing Time
		Gpu	Communication Time
2	Horizontal	Nongpu	Processing Time
		Gpu	Processing Time
		High Memory	Processing Time



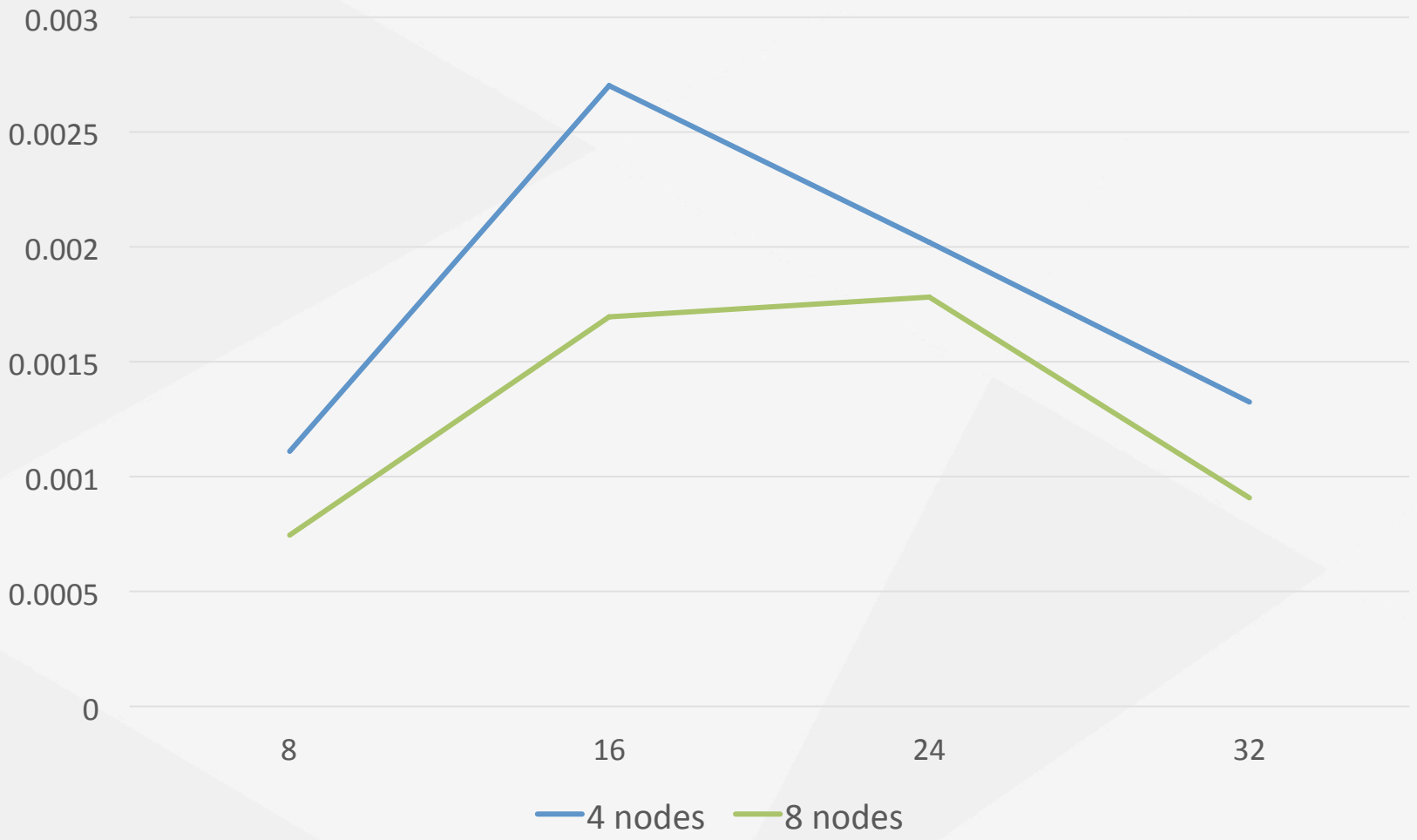

```
[urachmawati@borneo02 ~]$ ls
art  art2  fileasli  hello  hello.c  hostfile  test2.pbs  test.qsub
[urachmawati@borneo02 ~]$
```

```
[urachmawati@borneo02 ~]$
```

RESULT

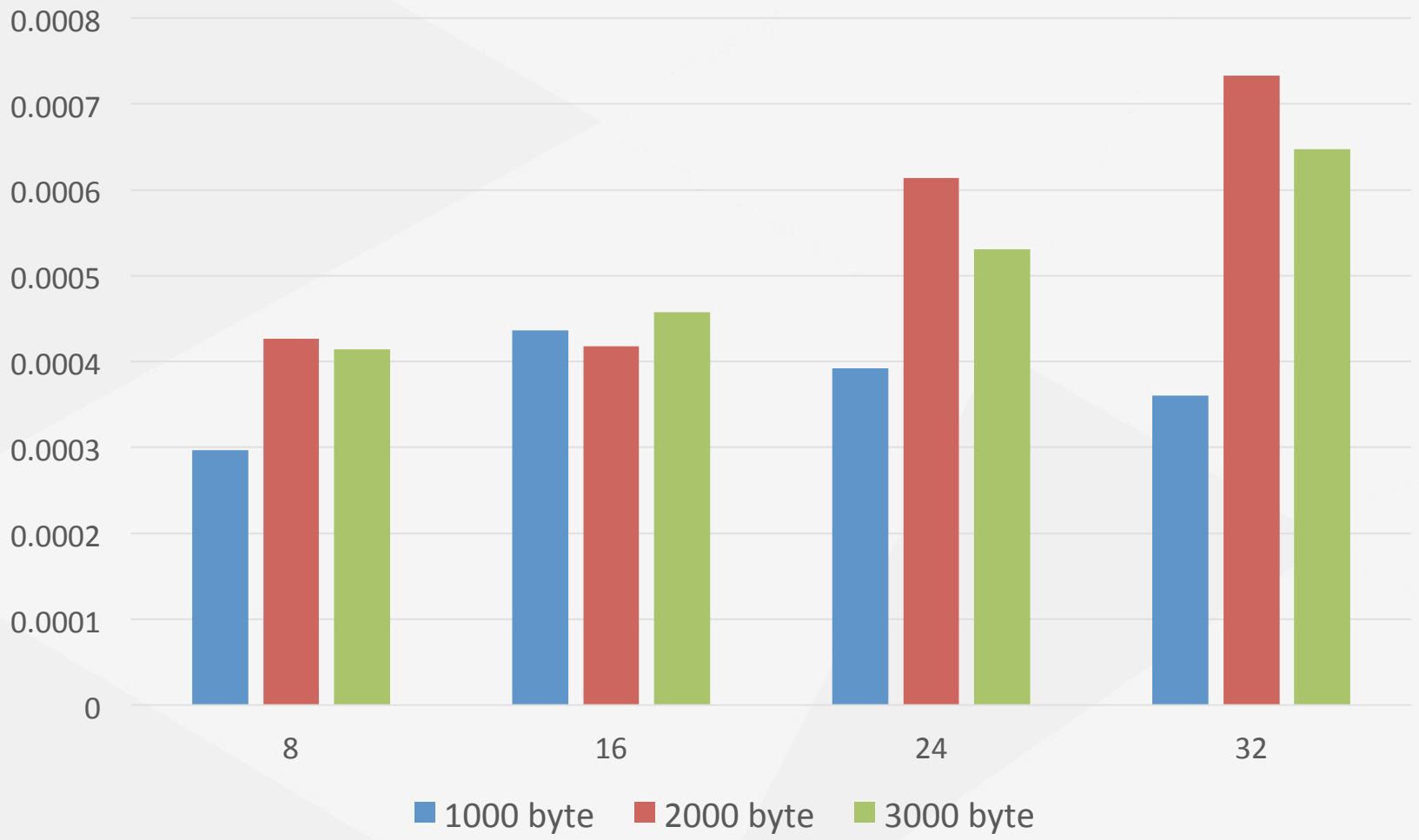


RESULT #1



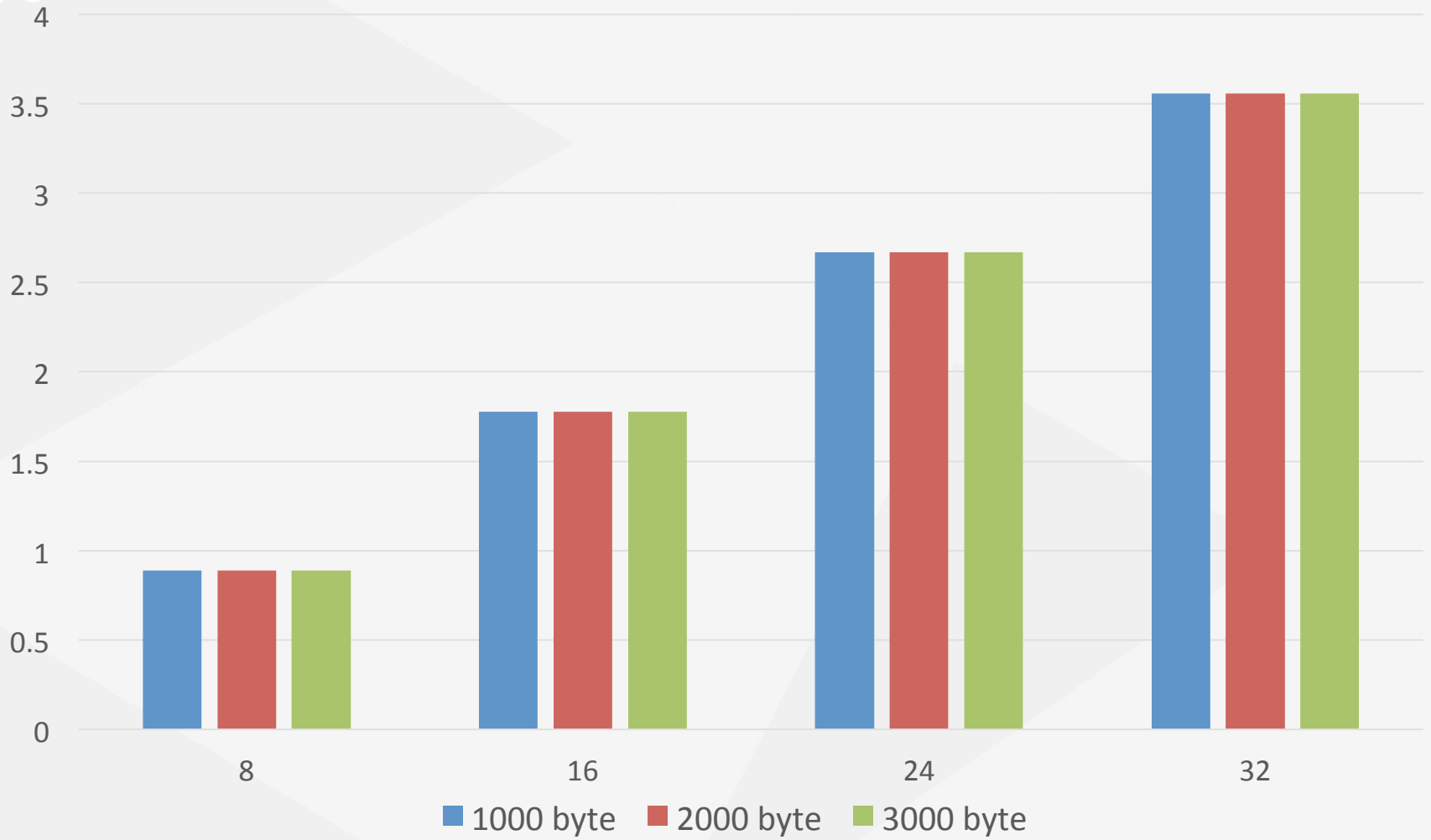
Graph comparison processing time 4 and 8 nodes with vertical method (nongpu / gpu).

RESULT #2



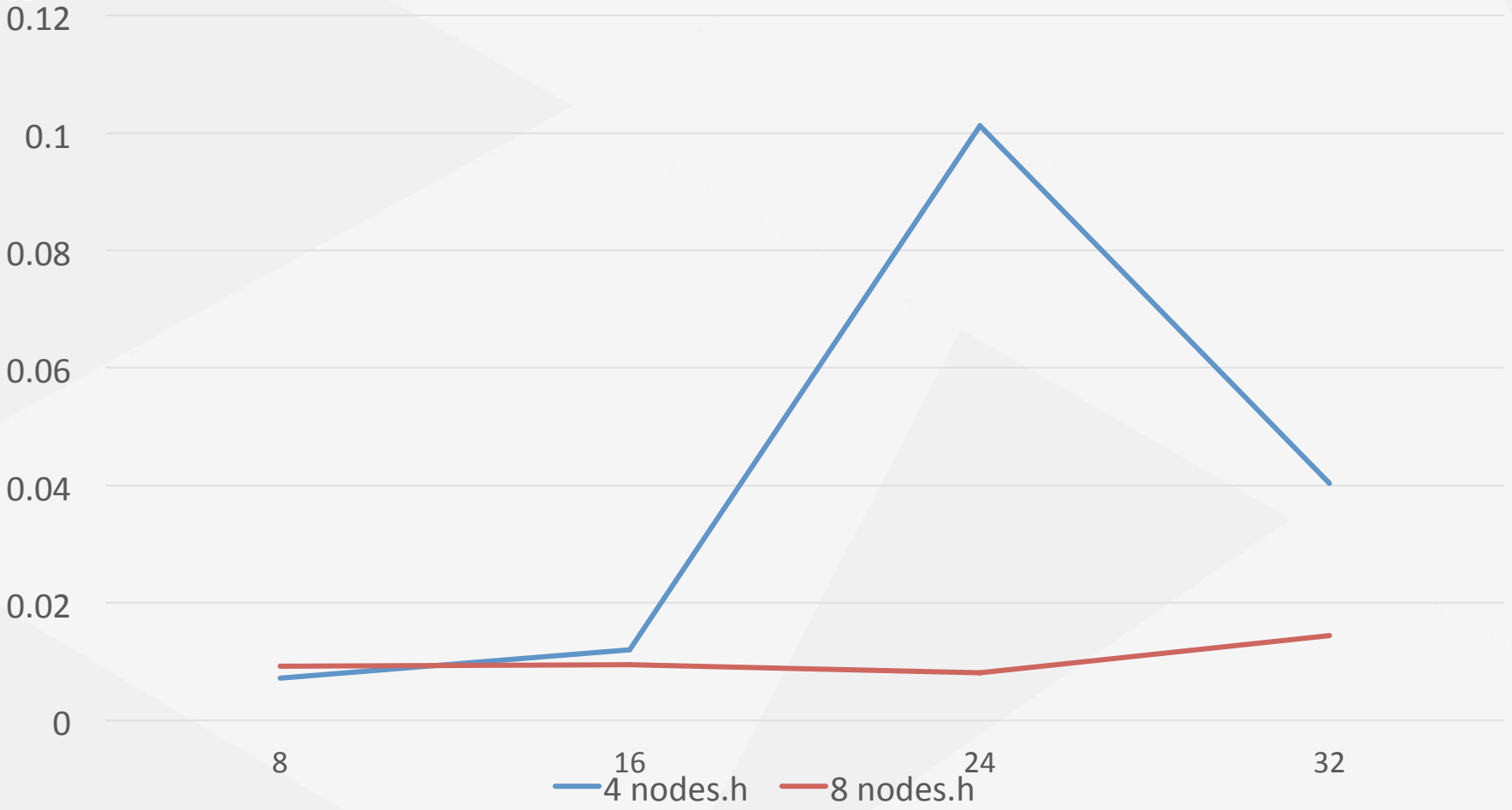
Communication time graph 4 vertical method node (nongpu / gpu).

RESULT #3



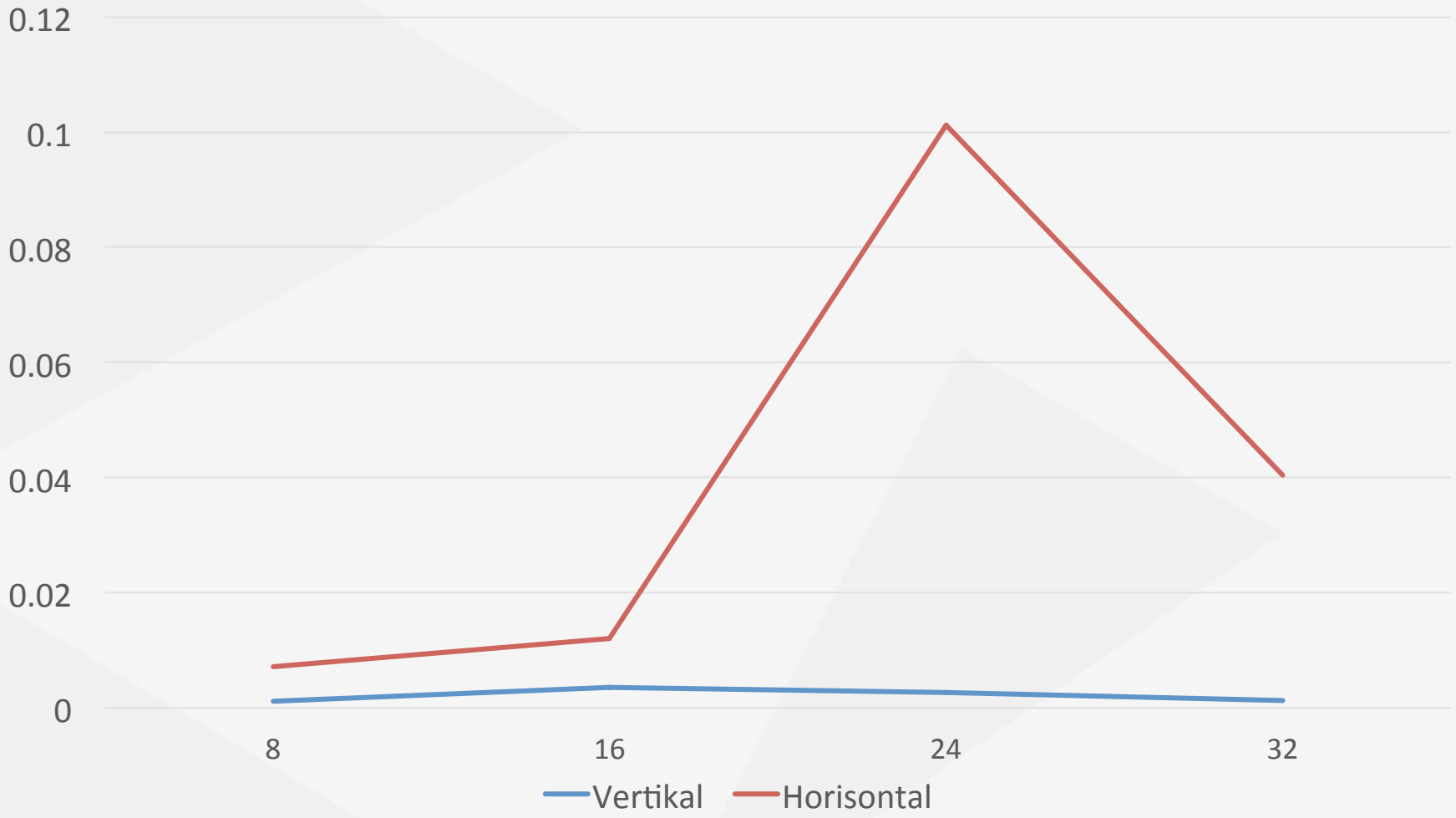
Communication time graph of 8 vertical method nodes (nongpu / gpu)

RESULT #4



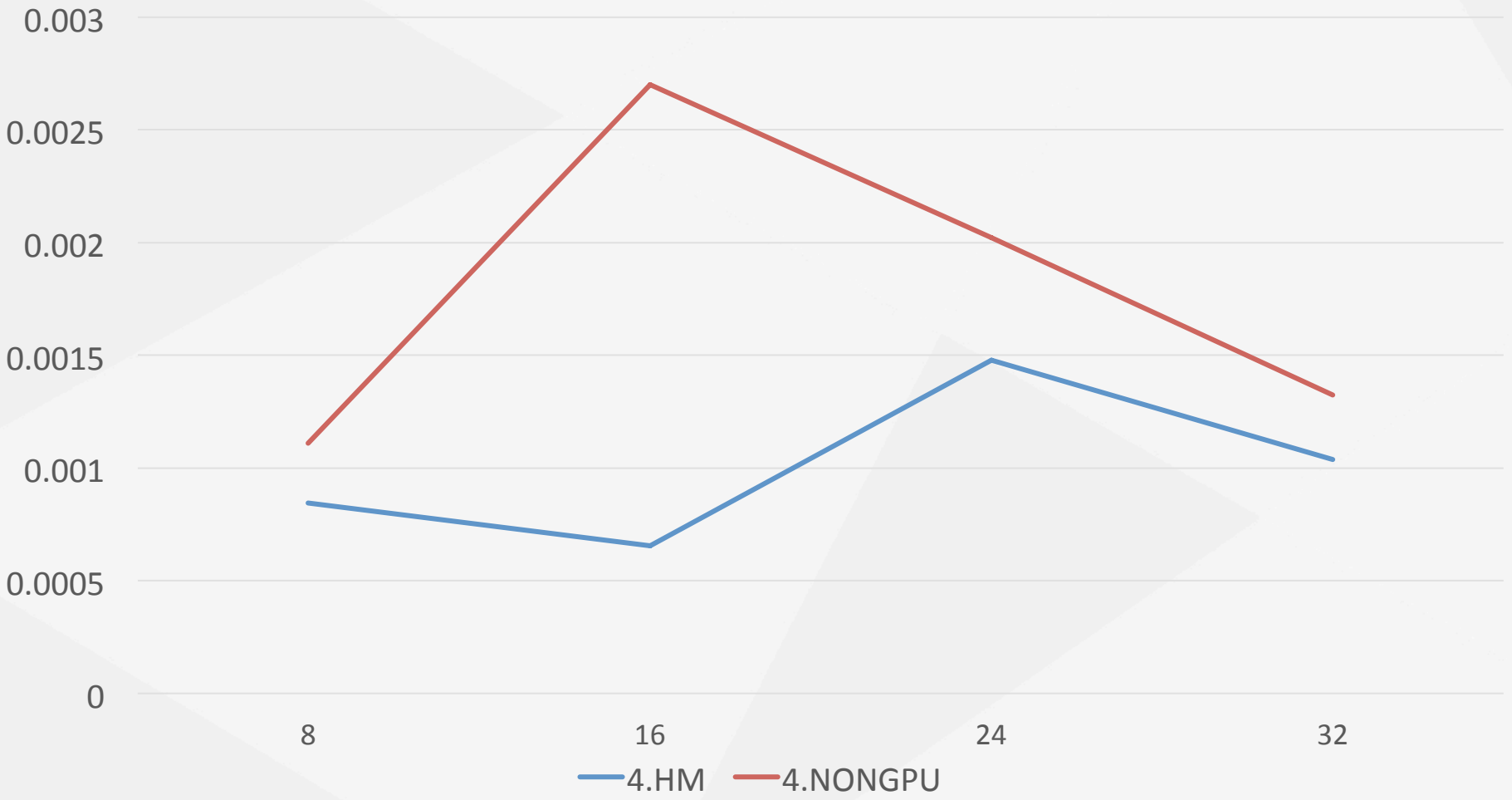
The comparison graph of processing time 4 and 8 nodes with horizontal method (nongpu / gpu)

RESULT #5



Graph comparison processing time 4 nodes on vertical and horizontal methods

RESULT #6



Graph comparison processing time 4 node high memory with nongpu.

CONCLUSION

- Part of Indonesian E-Health Grid model could be implemented on Grid LIPI with MPI, limited to 16 nodes with multiple processors.
- Average processing time is less than 1 second using vertical method. Processing time is increasing on 16 processors related to host in charge of the task (nongpu and gpu).
- Processing time using horizontal method is longer than the vertical method.
- Communication time test using vertical method is less than 1 second on 4 nodes and 3.5 seconds for 8 nodes.

REFERENCES

- Rachmawati, U.A., Haryanti, S.C. and Aini, N., 2016. E-Health Grid Network Topology Based on Referral Hospital Clustering in Indonesia. JCP, 11(6), pp. 513-519.. pada Maret 2017
- Keuchkerian, Samuel., 2010. *The Future of Healthcare: eHealth and Grid Computing*. Diakses dari http://www.gridtalk.org/Documents/ehealth.pdf pada April 2017.
- Wijaya, Wahyu.,2013..Laporan Triple Matrix Adjacency MPI. Diakses dari https://www.academia.edu/5568066/Laporan_Triple_Matrix_Adjacency_MPI pada Desember 2016.
- Tujuan dari LIPI. Diakses dari http://lipi.go.id/tentang/visimisi pada Maret 2017.
- Siregar., 2010. Definisi Rumah Sakit. Diakses dari repository.usu.ac.id/ bitstream/123456789/18308/4/Chapter%20II.pdf pada Mei 2017.
- Azzahra, Nabilla.,2017. Perancangan e-Health Grid dengan Menggunakan OpenFlow untuk Rumah Sakit di Indonesia Berdasarkan Provinsi serta Tinjauannya Menurut Agama Islam pada Maret 2017.
- Rachmawati, U.A., Haryanti, S.C., Agung, H. and Suhartanto, H., 2017. Software Defined e-Health Grid Networking Design Based on Referral Hospital in Indonesia. 10(4) pada Agustus 2017.

THANK YOU