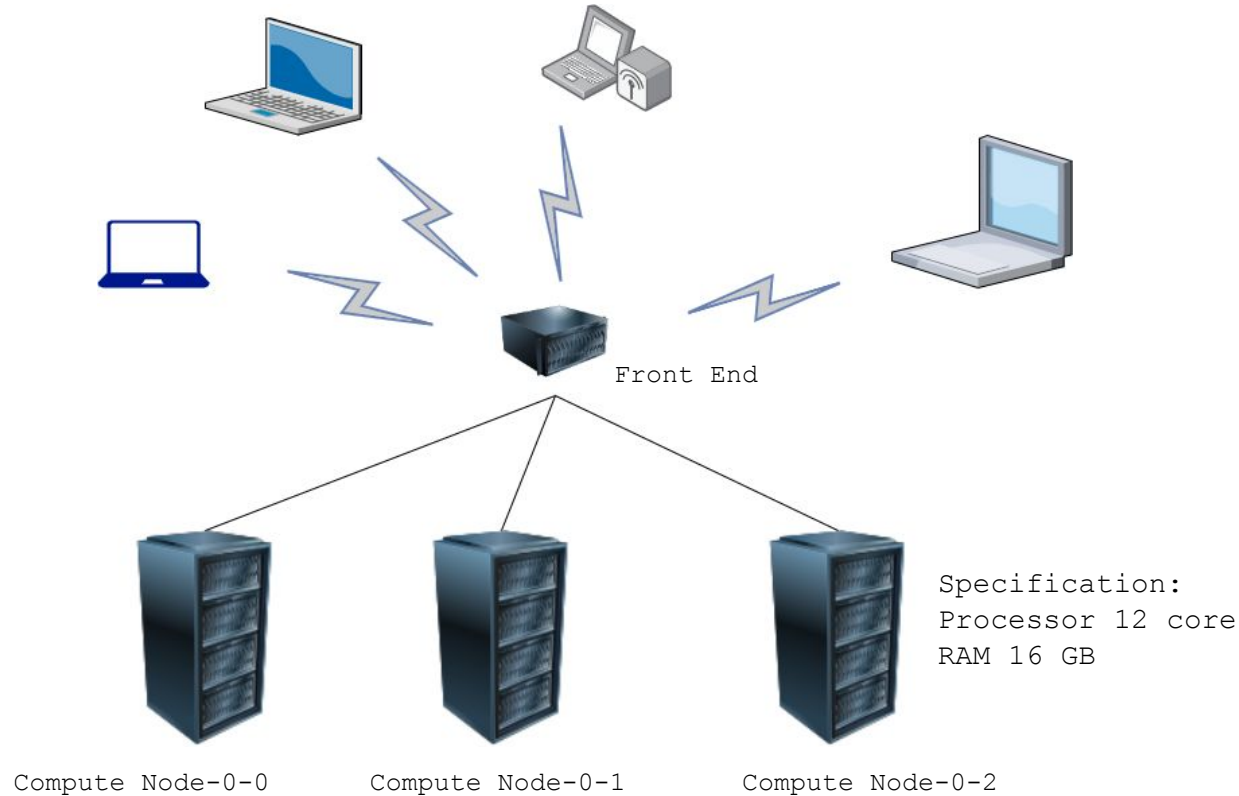


**KSCC 2020
Miscellaneous
v1.1**

Warat Puengtambol

Architecture



Practice

Compute Node: 3

Processor: 4 x 3

Source Code

Verification

Compute Node: 3

Processor: 12 x 3

WinSCP

The WinSCP interface displays a file transfer session. The top menu bar includes Local, Mark, Files, Commands, Session, Options, Remote, and Help. Below the menu is a toolbar with icons for Synchronize, Queue, and Transfer Settings. The main window is divided into two panes. The left pane shows the local file system (D:\Data) with a list of files including interfaces.txt, introduction.txt, languages.txt, library.txt, operation_mask.txt, portable.txt, protocols.txt, public_key.txt, remote_command.txt, and requirements.txt. The right pane shows the remote file system (/home/mprikryl/httpdocs/wiki/) with a list of files including .., wiki, .htaccess, administration.txt, after_installation.txt, announcement_winscp55.txt, announcement_winscp57.txt, awards.txt, commandline.txt, and config.txt. A Queue window at the bottom shows the progress of the transfer, listing the source, destination, size, time, speed, and progress for each operation.

Operation	Source	Destination	Transferred	Time	Speed	Progress
	/home/mprikryl/httpdocs/for...	D:\Documents\backup*.*	2 KB			Completed
	D:\Documents\wiki	/home/mprikryl/httpdocs...	29 KB	0:00:06	3,91 KB/s	52%
	D:\Documents\wiki\config.txt	/home/mprikryl/httpdocs...	5 KB			30%
	D:\Documents\movies\Movie\...	/home/mprikryl/httpdocs...	6 395 KB	0:07:49	44,6 MB/s	8%

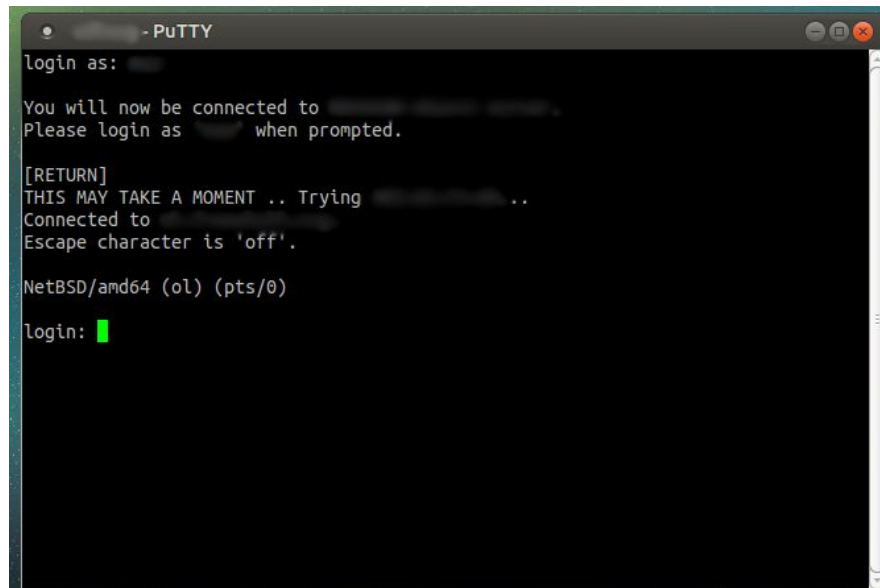
28,0 KB of 162 KB in 7 of 52

21,4 KB of 162 KB in 4 of 52

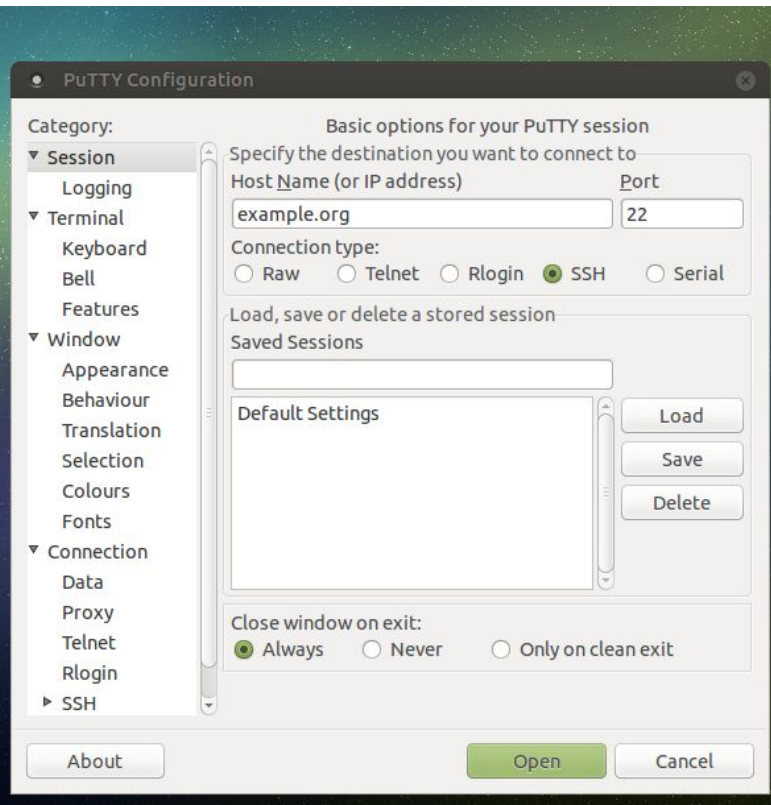
Queue (2)

SFTP-3 0:04:07

Putty



```
- PuTTY  
login as:   
  
You will now be connected to   
Please login as   
when prompted.  
  
[RETURN]  
THIS MAY TAKE A MOMENT .. Trying ..  
Connected to   
Escape character is 'off'.  
  
NetBSD/amd64 (ol) (pts/0)  
login: █
```



PuTTY Configuration

Category: Basic options for your PuTTY session

▼ Session
Specify the destination you want to connect to
Host Name (or IP address) Port
example.org 22

Connection type:
☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session
Saved Sessions
Default Settings [Load] [Save] [Delete]

Close window on exit:
☒ Always ☐ Never ☐ Only on clean exit

[About] [Open] [Cancel]

Compile c

```
$ gcc <.c file> -o <output filename> <include lib>
```

```
$gcc virus.c -o virus -lm
```

```
$./virus
```

MPI

```
$ mpicc <.c file> -o <output filename> <include lib>
```

```
$mpicc virus.c -o virus -lm
```

```
$mpirun -np <number of processor> <output filename>
```

```
$mpirun -np 2 virus.c
```

Open MP

```
$gcc -o <output filename> -fopenmp <.c file> <include lib>
```

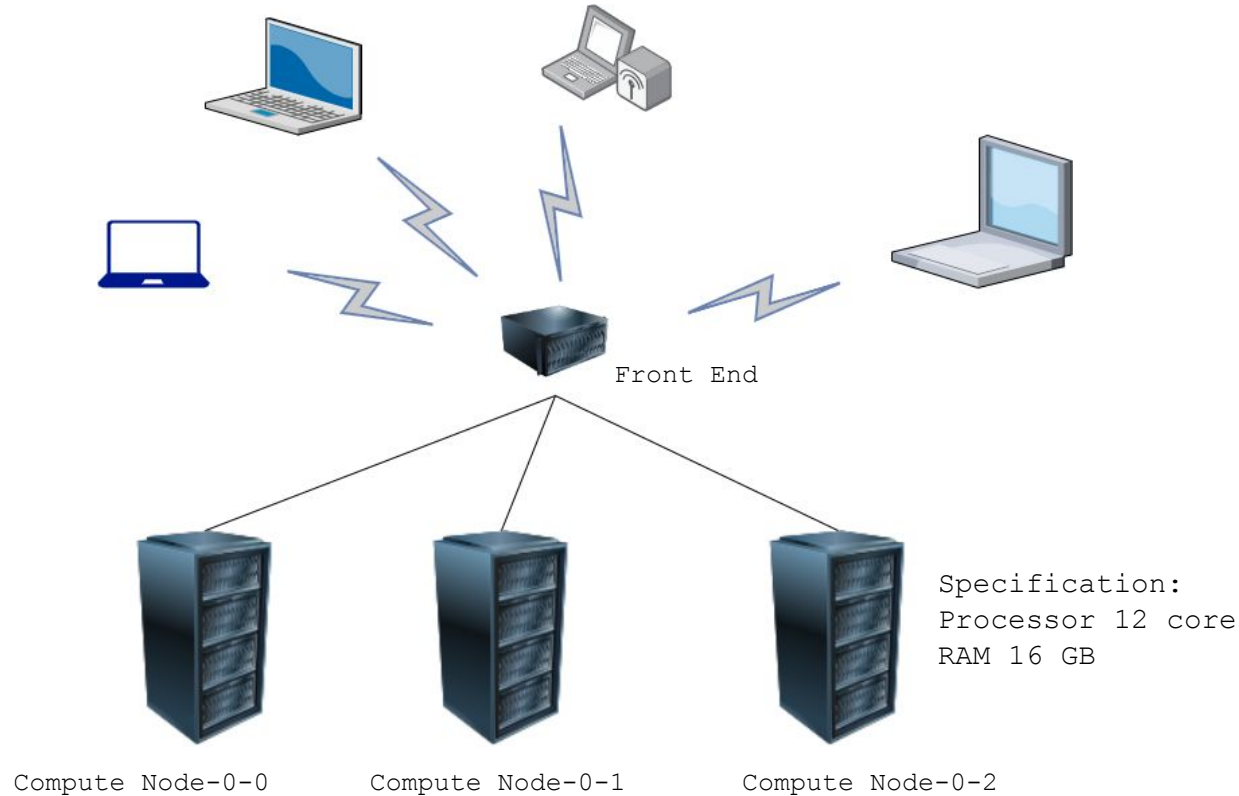
```
$gcc -o virus hello -fopen virus.c -lm
```

```
$export OMP_NUM_THREADS=2
```

```
$./virus
```

Job File

Architecture



Practice

Compute Node: 3

Processor: 4 x 3

Source Code

Verification

Compute Node: 3

Processor: 12 x 3

Job File

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N [your_job_name]
```

```
#$ -q [queue_name.q]
```

```
[your_program] [argv1] [argv2] [argv3] ...
```

Job File

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N Normal Virus
```

```
#$ -q kscq.q
```

```
./virus < seed.txt
```

Job File mpi

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N [your_job_name]
```

```
#$ -q [queue_name]
```

```
#$ -pe mpi [number_of_slots]
```

```
mpirun -np $NSLOTS [your_program] [argv1] [argv2] [argv3] ...
```

Job File mpi

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N mpi_virus
```

```
#$ -q ksc.c.q
```

```
#$ -pe mpi 2
```

```
mpirun -np $NSLOTS ./virus < seed.txt
```

Job File OpenMP

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N [your_job_name]
```

```
#$ -q [queue_name]
```

```
#$ -pe mpi [number_of_slots]
```

```
export OMP_NUM_THREADS=$NSLOTS && [your_program] [argv1] [argv2]
```

Job File OpenMP

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N OpenMP_Virus
```

```
#$ -q ksc.c.q
```

```
#$ -pe mpi 2
```

```
export OMP_NUM_THREADS=$NSLOTS && ./virus < seed.txt
```

Job File OpenMP with MPI

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N [your_job_name]
```

```
#$ -q [queue_name]
```

```
#$ -pe mpi [number_of_slots]
```

```
export OMP_NUM_THREADS=[processor per computenode] && mpirun -n [number of compute node] -x  
OMP_NUM_SLOTS -pernode [your_program]
```


Job File OpenMP with MPI

```
#!/bin/bash
```

```
#$ -cwd
```

```
#$ -N OpenMP_MPI
```

```
#$ -q ksc.c.q
```

```
#$ -pe mpi 12
```

```
export OMP_NUM_THREADS=4 && mpirun -n 3 -x OMP_NUM_SLOTS -pernode ./virus < seed.txt
```

Submit job

```
$ qsub <job file name>
```

```
$ qsub jobfile.txt
```

Happy Coding!!