

R lab simulation seminar

Chapter 0: How to use cluster computer

Takeshi Kawasaki

Department of Physics, Nagoya University

Last-update : September 15, 2022

Agenda

1 How to access to the R lab work station?

- VPN connection
- Useful tools
- SSH to the cluster

2 How to use the cluster?

- Compilation of your codes.
- How to submit your job – qsub.

Agenda

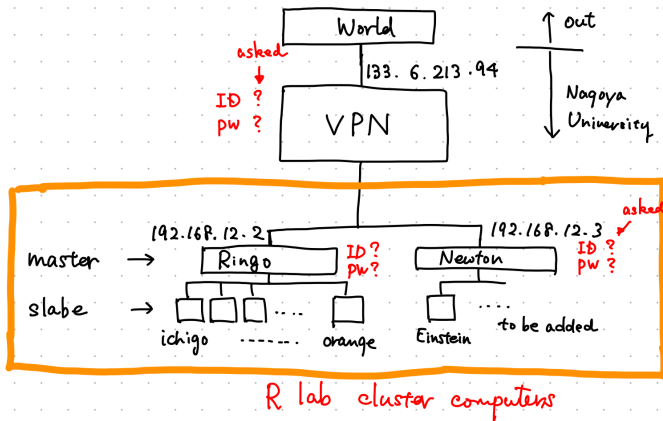
1 How to access to the R lab work station?

- VPN connection
- Useful tools
- SSH to the cluster

2 How to use the cluster?

- Compilation of your codes.
- How to submit your job – qsub.

1. How to access to the R lab work station?



☒ 1: R lab cluster system.

1.1. VPN connection

Information of the VPN router, connecting to the R lab cluster computer:

- IP address: 133.6.213.94
- VPN type: Pre-shared key L2TP/IPsec
- Pre-shared key (公開鍵) : broadstation
- ID: guest (or your sir name)
- PW: guest (or your sir name)

1.2. Useful tools

Windows users

- Teraterm (for terminal)
- Winscp (for scp)

Mac users

- File zilla (for scp)

1.3. SSH to the cluster

R lab cluster computer (ringo):

- IP address: 192.168.12.2
- ID: your sir name
- PW: your sir name

R lab cluster computer (newton):

- IP address: 192.168.12.3
- ID: your sir name
- PW: your sir name

リスト 1: "ssh" to the cluster computer (ringo).

```
1 ssh kawasaki@192.168.12.2
```

リスト 2: "ssh" to the cluster computer (newton).

```
1 ssh kawasaki@192.168.12.3
```

Agenda

1 How to access to the R lab work station?

- VPN connection
- Useful tools
- SSH to the cluster

2 How to use the cluster?

- Compilation of your codes.
- How to submit your job – qsub.

2.1. Compilation of your codes.

Compile this sample program.

リスト 3: A sample program. "pi.cpp"

```
1 # include <stdio.h>
2 # include <stdlib.h>
3 # include <math.h>
4 # include <iostream>
5 #include <iomanip>
6 # include <fstream>
7 # include <time.h>
8 using namespace std;
9 //A Monte Carlo simulation
10 int main(void){
11     int i, count = 0, max = 1e+7;
12     double x,y,z,pi;
13     srand(time(NULL));
14     for(i=0;i<max;i++){
15         x = (double)rand()/RAND_MAX;
16         y = (double)rand()/RAND_MAX;
17         z = x*x + y*y;
18         if(z<1)
19             count++; }
20     pi = (double)count / max * 4;
21     printf("%.20f\n",pi); // by C, %.20f    Display with 20 decimal precision
22     cout<< std::setprecision(21) << pi << endl; // by C++
23     return 0;
24 }
```

2.1. Compilation of your codes.

リスト 4: How to compile a code.

```
1  icc -O3 pi.c  
2  icc -O3 pi.cpp -o pi.out
```

compilers

- icc: intel c/c++ compiler
- gcc: GNU compiler collection

compile options

- -O3: One of the optimization options (i.e., -O -O0 -O1 -O2 -O3 -Os -Ofast -Og). -O3 is most used.
- -o: Specifies the name of the output file.

2.2. How to submit your job – qsub.

Prepare a "xxxxx.sh" file for submitting job.

リスト 5: Shell script file for submitting jobs. (eg: qsub.sh)

```
1 #!/bin/bash
2 #
3 # $ -cwd
4 # $ -j y
5 # $ -S /bin/bash
6 # $ -M kawasaki@r.phys.nagoya-u.ac.jp
7 # $ -m ea
8 # $ -V
9 #
10 # $ -q all.q@einstein
11 ./pi.out
```

リスト 6: Way for submitting a job with "qsub.sh".

```
1 qsub qsub.sh
```

リスト 7: Way for checking the status of your jobs.

```
1 qstat
```