

(index.htm)

(index.htm)


NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

hackUST & hardUST

User Manual on TH-2

Simplified Version

2018/3/2



(index.htm)

Table of List

(index.htm)

1	Login.....	3
1.1	VPN Verification	3
1.1.1	Configuration for Windows OS	3
1.1.2	Configuration for Mac OS X (SSLVPN)	3
1.1.3	Configuration for MAC OS X (IPSECVPN)	5
1.1.4	Configuration for CentOS/Redhat	7
1.1.5	Configuration for Ubuntu/Ubuntu Kylin	10
1.2	Login from Client Software.....	12
1.2.1	Login with Xshell (Xmanager-Xshell).....	12
1.2.2	Login with PUTTY.....	13
1.2.3	Login with Linux or Mac OS X	14
2	File Transfer.....	15
2.1	File System.....	15
2.2	Data Transfer	15
2.2.1	FileZilla Login	15
2.2.2	WinSCP Login	18
3	Environment variables management tool - module.....	20
3.1	Introduction.....	20
3.2	Basic command.....	20
4	Compiler.....	22
4.1	Intel Compiler.....	22
4.2	GCC Compiler	23
4.3	MPI compile environment	23
5	Job submission.....	25
5.1	Check node status with yhinfo or yhi	25
5.2	Check job status with yhqueue or yhq	25
5.3	Interactive processing job with yhrun.....	25
5.3.1	Introduction.....	25
5.3.2	General options of yhrun	25
5.3.3	Examples.....	27
5.4	Batch processing job with yhbatch.....	28
5.4.1	Introduction.....	28
5.4.2	Examples.....	28
5.5	Resource occupy command with yhalloc	29
5.5.1	Introduction.....	29
5.5.2	Examples.....	29
5.6	Cancel job with yhcancel	29
6	FAQ.....	31

(index.htm)

1 Login

(index.htm)

1.1 VPN Verification

1.1.1 Configuration for Windows OS

- A. Visit the webpage <https://vpn3.nscc-gz.cn:4433>, log in with your VPN username and password, then download the installation file of the VPN client and install;
- B. Run “**Hillstone Secure Connect**”, configure and input the following parameters and then click **Login**.
 - **Saved Connection:** It provides the previous connection information. You may select a connection from the drop-down menu.
 - **Server:** Input the IP address **vpn3.nscc-gz.cn**
 - **Port:** Input the port number **4433**
 - **Username:** Input the login name
 - **Password:** Input the corresponding password



- C. Once a successful connection is established, an icon  will be displayed at the right bottom of the Desktop.

1.1.2 Configuration for Mac OS X (SSLVPN)

- A. Download “SCVPN OSX 1.0.4.dmg” (SSLVPN Client for Mac OS X) at

(index.htm)

<http://pan.baidu.com/s/1o8RI3gQ> (password: b2vh), and install.

(index.htm)



B. Run “**Hillstone Secure Connect**”, input the following parameters and then click OK.

- **Name:** Enter a name for the connection
- **Description:** Enter the corresponding description for this connection
- **Server:** Input the IP address **vpn3.nscc-gz.cn**
- **Port:** Input the port number **4433**
- **Username:** Input the login name
- **Password:** Input the corresponding password
- **Remember Password:** Select the check box to remember the password



hackUST & hardUST

C. Once a successful connection is established, Connection established will be shown in the status bar”.

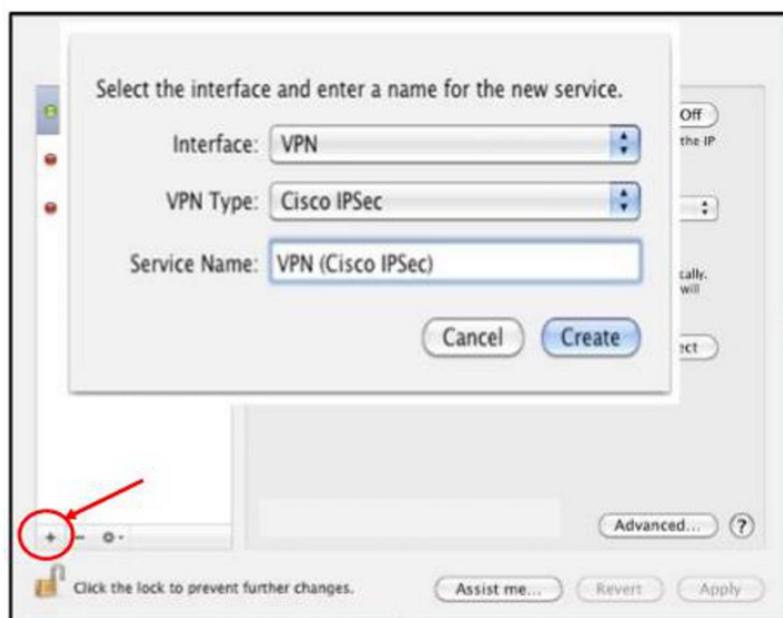
(index.htm)



1.1.3 Configuration for MAC OS X (IPSECVPN)

hackUST & hardUST

- Open System Preferences and click Network
- Click on the + sign in the lower left to add a new service.
 - Select Interface to be **VPN**
 - Select VPN Type to be **Cisco IPSec**
 - Click **Create**.



(index.htm)

- C. Enter **vpn3.nscc-gz.cn** as the Server Address

(index.htm)



Enter the username and the corresponding password into the Account Name and Password boxes respectively.

Click on **Authentication Settings**.



hackUST & hardUST

- D. In the Authentication Settings dialogue, enter **gzcszx@123** as the Shared Secret and **hillstone** as the Group Name

Click **OK**.

(index.htm)

(index.htm)



E. Once a successful connection is established, a welcome dialogue will appear.

Click **OK**.



hackUST & hardUST

1.1.4 Configuration for CentOS/Redhat

A. Login as a root user to install VPN client by the following command:

For 32-bit CentOS/Redhat:

```
# rpm -ivh http://kartolo.sby.datautama.net.id/EPEL/6/i386/vpnc-0.5.3-4.el6.i686.rpm
```

For 64-bit CentOS/Redhat:

```
# rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/vpnc-0.5.3-4.el6.x86_64.rpm
```

(index.htm)

- B. Type in the command **vpnc** to run the VPN client once installation is completed.

(index.htm)



Input the following parameters accordingly:

- **Enter IPSec gateway address: `vpn3.nscc-gz.cn`**
- **Enter IPSec ID: `hillstone`**
- **Enter IPSec secret: `gzcszx@123`**
- **Enter username:** Input the login name
- **Enter password:** Input the corresponding password

```
root@localhost:~/Desktop
File Edit View Search Terminal Help
[root@localhost Desktop]# rpm -ivh http://kartolo.sby.datautama.net.id/EPEL/6/i386/vpnc-0.5.3-4.el6.i686.rpm
Retrieving http://kartolo.sby.datautama.net.id/EPEL/6/i386/vpnc-0.5.3-4.el6.i686.rpm
warning: /var/tmp/rpm-tmp.meFhv8: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY
Preparing... ################################################ [100%]
1:vpnc ################################################ [100%]
[root@localhost Desktop]# vpnc
Enter IPSec gateway address: 10.88.16.100
Enter IPSec ID for 10.88.16.100: hillstone
Enter IPSec secret for hillstone@10.88.16.100:
Enter username for 10.88.16.100: xauth
Enter password for xauth@10.88.16.100:
Connect Banner:
| Welcome to StoneOS

VPNC started in background (pid: 3766)...
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
```

- C. Type in the command **ifconfig**.

Once a successful connection is established, you will be able to trace the tunnel interface and the corresponding IP address.

(index.htm)

(index.htm)



hackUST & hardUST

```
root@localhost:~/Desktop
File Edit View Search Terminal Help
[root@localhost Desktop]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0C:29:D5:B6:27
          inet addr:10.88.16.105 Bcast:10.88.16.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed5:b627/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:11274 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1515 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3609516 (3.4 MiB) TX bytes:97439 (95.1 KiB)
          Interrupt:19 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1716 (1.6 KiB) TX bytes:1716 (1.6 KiB)

tun0     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.16.1.14 P-t-P:172.16.1.14 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1412 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 b) TX bytes:1100 (1.0 KiB)

[root@localhost Desktop]#
```

D. To disconnect the VPN client, type in the command **vpnc-disconnect**.

```
[root@localhost Desktop]# vpnc-disconnect
Terminating vpnc daemon (pid: 3766)
[root@localhost Desktop]#
```

E. To uninstall the VPN client, type in the following command:

For 32-bit CentOS / Redhat:

#rpm -e vpnc-0.5.3-4.el6.i686

For 32-bit CentOS / Redhat:

#rpm -e vpnc-0.5.3-4.el6.x86_64

```
[root@localhost Desktop]# rpm -e vpnc-0.5.3-4.el6.i686
[root@localhost Desktop]# vpnc
bash: /usr/sbin/vpnc: No such file or directory
[root@localhost Desktop]#
```

(index.htm)

1.1.5 Configuration for Ubuntu/Ubuntu Kylin

- A. Login as a root user by using the command **su – root**

(index.htm)



Type in the command **apt-get install vpnc** to install the VPN client

Type in the command **vpnc** to run the VPN client once installation is completed.

- **Enter IPSec gateway address:** **vpn3.nscc-gz.cn**
- **Enter IPSec ID:** **hillstone**
- **Enter IPSec secret:** **gzcszx@123**
- **Enter username:** Input the login name
- **Enter password:** Input the corresponding password

```
tjchen@ubuntu:~$  
tjchen@ubuntu:~$  
tjchen@ubuntu:~$ su - root  
Password:  
root@ubuntu:~# apt-get install vpnc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  vpnc  
0 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.  
Need to get 0 B/87.0 kB of archives.  
After this operation, 251 kB of additional disk space will be used.  
Selecting previously unselected package vpnc.  
(Reading database ... 57330 files and directories currently installed.)  
Preparing to unpack .../vpnc_0.5.3r512-2ubuntu1_amd64.deb ...  
Unpacking vpnc (0.5.3r512-2ubuntu1) ...  
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...  
Setting up vpnc (0.5.3r512-2ubuntu1) ...  
root@ubuntu:~# vpnc  
Enter IPSec gateway address: 10.88.16.100  
Enter IPSec ID for 10.88.16.100: hillstone  
Enter IPSec secret for hillstone@10.88.16.100:  
Enter username for 10.88.16.100: xauth  
Enter password for xauth@10.88.16.100:  
Connect Banner:  
! Welcome to StoneOS  
  
VPNC started in background (pid: 3890)...  
root@ubuntu:~#
```

hackUST & hardUST

- B. Type in the command **ifconfig**.

Once a successful connection is established, you will be able to trace the tunnel interface and the IP address.

(index.htm)



(index.htm)

```
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:2f:c1:b6
          inet addr:10.88.16.103 Bcast:10.88.16.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe2f:c1b6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:7866 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:550994 (550.9 KB) TX bytes:10902 (10.9 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

tun0     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.16.1.16 P-t-P:172.16.1.16 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1412 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@ubuntu:~# _
```

hackUST & hardUST

C. To disconnect the VPN client, type in the command **vpnc-disconnect**.

```
root@ubuntu:~# vpnc-disconnect
Terminating vpnc daemon (pid: 3890)
root@ubuntu:~#
```

D. To uninstall the VPN client, type in the command **apt-get remove vpnc**.

```
root@ubuntu:~# apt-get remove vpnc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  iproute
Use 'apt-get autoremove' to remove it.
The following packages will be REMOVED:
  vpnc
0 upgraded, 0 newly installed, 1 to remove and 136 not upgraded.
After this operation, 251 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 57347 files and directories currently installed.)
Removing vpnc (0.5.3r512-2ubuntu1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
root@ubuntu:~# vpnc
-su: /usr/sbin/vpnc: No such file or directory
root@ubuntu:~# _
```

(index.htm)

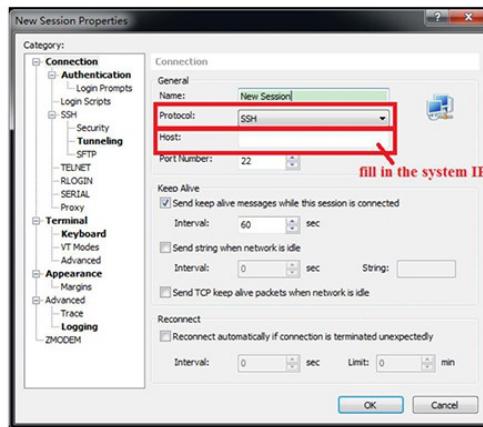


1.2 Login from Client Software

To further guarantee the security of data, users are only allowed to login with SSH. SSH client software such as Xmanager, PUTTY, SecureCRT, SSH Secure Shell Client are recommended. Login steps are as the following:

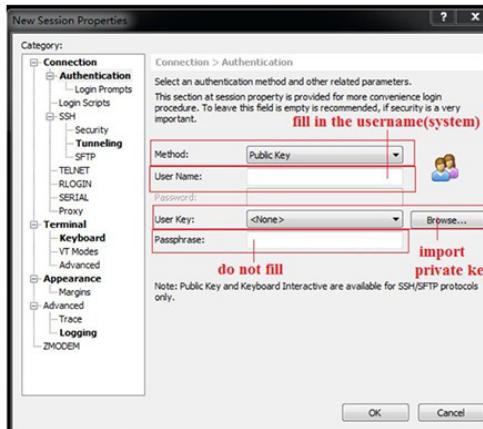
1.2.1 Login with Xshell (Xmanager-Xshell)

Run “Xshell”, click **New** in the toolbar. Go to **General** under the category **Connection**, select **SSH** under **Protocol** and fill in the following parameters:



- Hosts:** 172.16.22.11;
- Host Number:** 5566;

Select **Public Key** in **Method of Authentication**, and fill in **User Name** with your system username. Import the Private Key (.id file), and then click OK.

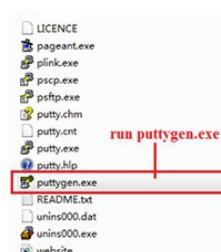


(index.htm)

1.2.2 Login with PUTTY

(index.htm)

To login with PUTTY, the Private Key should be converted first. You need to run “**puttygen.exe**” in the installation folder of PUTTY.



Click “**Load**” to import your Private Key. Then click “**Save private key**” to save the converted key.

hackUST & hardUST



Run “**PUTTY**” and select **Session**. Fill in the following information:

- **Hosts: 172.16.22.11;**
- **Host Number: 5566;**

(index.htm)

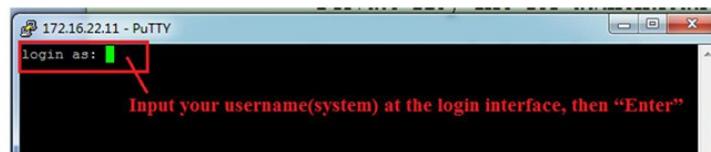
(index.htm)



Select **Connection -> SSH-> Auth**, then click the **Browse** button under **Private Key file for authentication**, import the converted Private Key file, and click Open to start the login interface.



Input your system username on the login interface, and then **Enter**.



1.2.3 Login with Linux or Mac OS X

If you are using Linux or Mac OS X, the Private Key file should be authorized with certain permission. The command is:

\$chmod 400 PrivateKey

Then modify some system files:

- 1) Modify the “ssh_config” file with root permission.
 - **Linux:** \$sudo vi /etc/ssh/ssh_config
 - **Mac OS X:** \$sudo vi /etc/ssh_config

(index.htm)

2) Add two lines below in the “**ssh_config**” file and save.

```
StrictHostKeyChecking no
```

```
UserKnownHostsFile /dev/null
```

(index.htm)



Open another client interface to login, select the Private Key file with ssh command:

```
ssh -i PrivateKey username@systemIP
```

Note: The Private Key is recommended to be regenerated after first login for the security of data. Details are in the FAQ of “Regenerate Private Key”.

2 File Transfer

2.1 File System

The partition of /BIGDATA1 is for data storage and job submission. The main directory of this partition can be quoted by using /BIGDATA1/username in your script. For free users, the size is 200GB~300GB. For premium users, the quota would be correlate with user’s requirement.

2.2 Data Transfer

Files on TH-2 can be upload or download through sftp client. You can either use the built in file transfer function of Xmanager, SSH Secure Shell Client, or data transfer software like FileZilla, WinScp, FlashFTP. (These free software are recommended for supporting resume broken transfer.)

2.2.1 FileZilla Login

Login steps of FileZilla are follows:

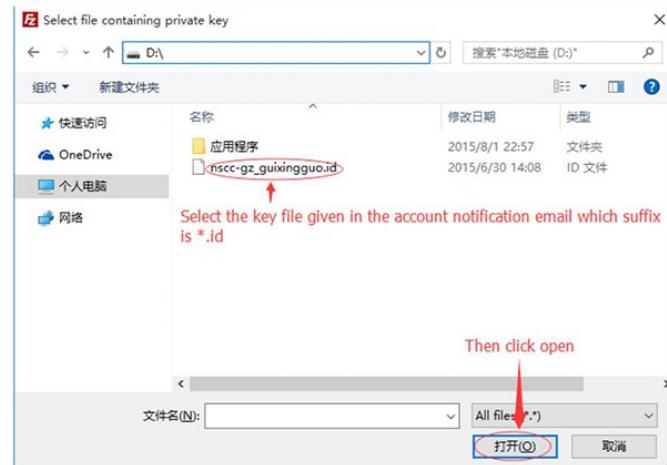
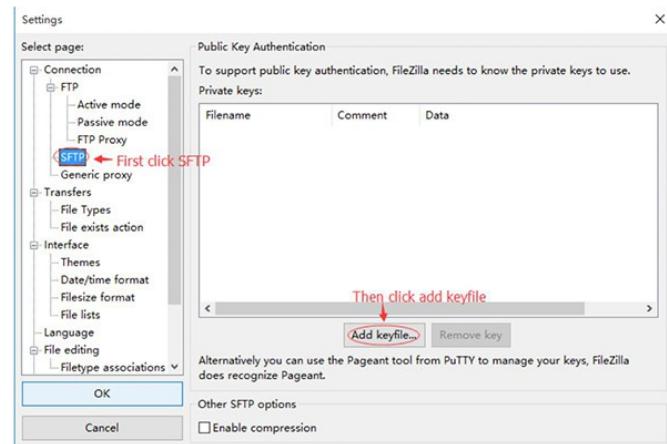


(index.htm)

(index.htm)



hackUST & hardUST

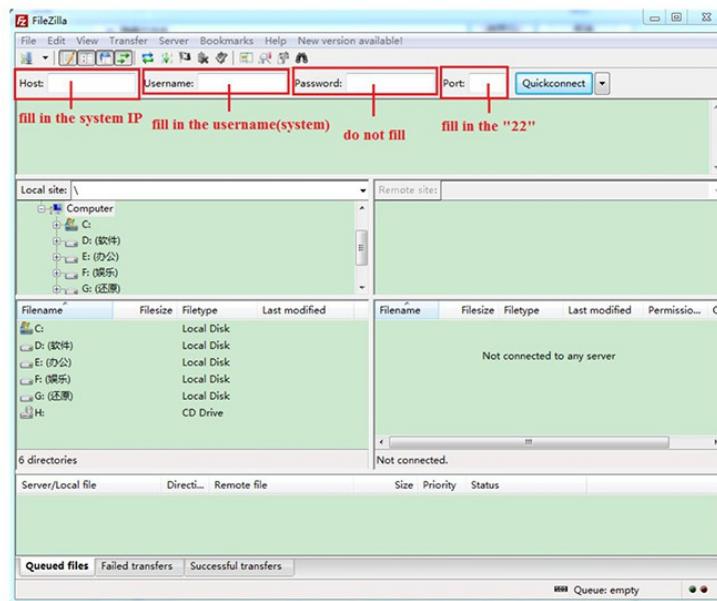
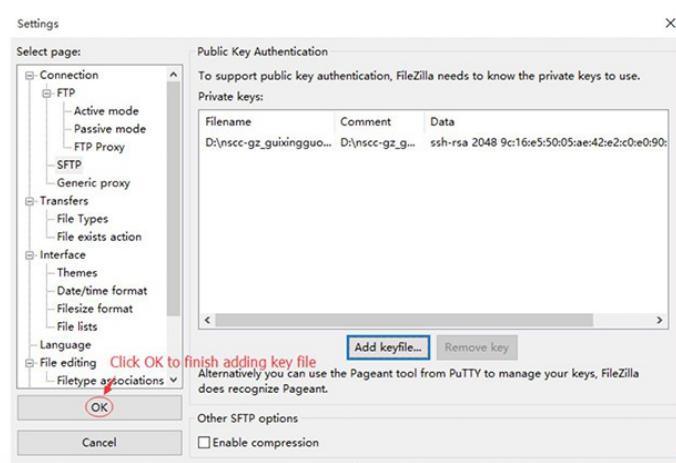


(index.htm)

(index.htm)



hackUST & hardUST

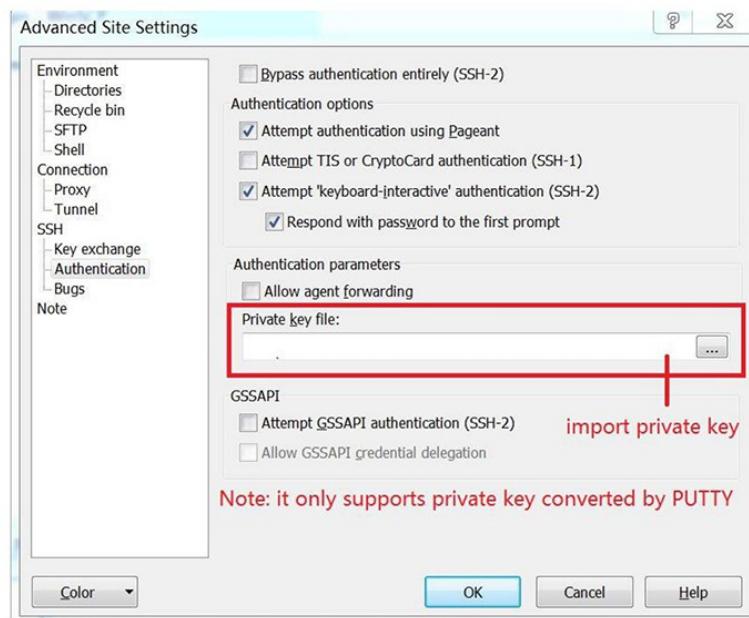
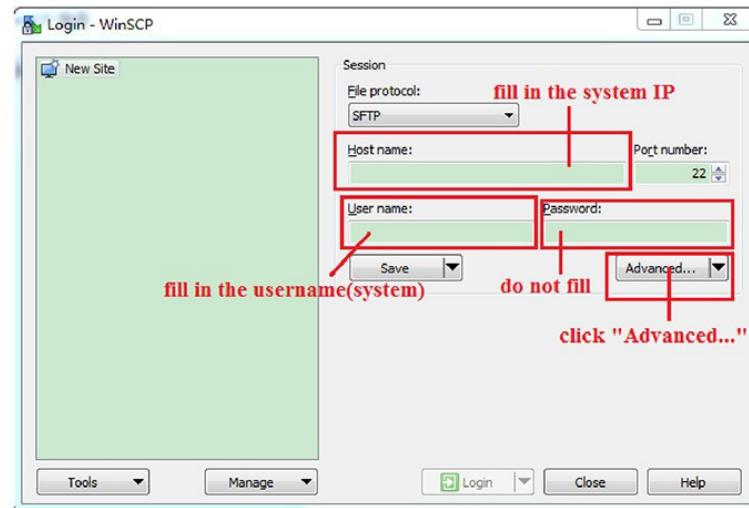


(index.htm)

2.2.2 WinSCP Login

Login steps of WinSCP are as follows:

(index.htm)



Notes:

1. SFTP service is provided 24 hours per day once login with VPN verification.

The storage space of TH-2 is only for temporary data storage. Considering the limitation of storage space and the security of your data, please save your important or sensitive data on your own computer and clear your space in time.

(index.htm)

3. When you are going to download large amount of files, pack your files first. It will accelerate the whole process and decrease the possibility of error. The command would be \$tar -cvf file.tar file. Here “file” is the file or directory that is to be packed, “file.tar” is the packed file.
4. The command for checking disk usage is as the following:

```
$lfs quota -uh `whoami` /BIGDATA1/username
```

(index.htm)



hackUST & hardUST

(index.htm)

3 Environment variables management tool - module

(index.htm)



3.1 Introduction

Users on TH-2 may need to set up different environment variables according to their needs, when the default configuration could not satisfy their requirements. We have installed module software on TH-2 to manage environment variables.

Module configures modulefile to support dynamic modification of environment variables, which controls the dependency of different software on environment variables. Users can set up their desired environment through simple commands, which would increase the efficiency of software transplantation.

3.2 Basic command

hackUST & hardUST

The module tool has been configured over login service nodes, usage are follows:

\$module avail: List available modulefile

```
[testuser1@ln1%tinhe2-B ~]$ module avail
----- /usr/share/Modules/modulefiles -----
FFTW/3.3.3          MPI/Intel/MPICH/3.1-icc11-dbg hdf5/1.8.11-intel      netcdf/4.0.1-CF-V13
FFTW/3.3.4          MPI/Intel/mpi/4.1.2.0.040 intel-compilers/13.0.0      netcdf/4.3.0-C
FFTW/3.3.4-intel   NAMD/2.9                         intel-tools/14.0.1      netcdf/4.3.0-C-ohndf5
FFTW/3.3.4-openmp  OpenFOAM/2.2.2                  PETSc/3.4.4           netcdf/4.3.2-CF
FFTW/3.3.4-openmp-O3 PETSc/3.4.4                  cmake/2.6-patch4       lammps/14-Feb14        null
MPI/Gnu/MPICH/3.1   cmake/2.8                         module-cvs            module-info
MPI/Intel/MPICH/3.1 cmake/2.8                         module-info          modules
MPI/Intel/MPICH/3.1-dbg dot                           module              mpich2-x86_64
MPI/Intel/MPICH/3.1-icc11 emacs                         module              mpich2-x86_64
[testuser1@ln1%tinhe2-B ~]$ █
```

\$module load [modulesfile]: Load the desired modulefile.

```
[testuser1@ln1%tinhe2-B ~]$ icc -v
icc version 14.0.1 (gcc version 4.4.6 compatibility)
[testuser1@ln1%tinhe2-B ~]$ module load intel-compilers/13.0.0
[testuser1@ln1%tinhe2-B ~]$ icc -v
icc version 13.0.0 (gcc version 4.4.6 compatibility)
[testuser1@ln1%tinhe2-B ~]$ █
```

Check the environment of loaded modulefile (for example, OpenFOAM/2.2.2)

```
[testuser1@ln1%tinhe2-B ~]$ blockMesh
-bash: blockMesh: command not found
[testuser1@ln1%tinhe2-B ~]$ module load OpenFOAM/2.2.2
[testuser1@ln1%tinhe2-B ~]$ which blockMesh
/vol-th/COMMON_software/OpenFOAM-2.2.2/platforms/linux64iccDPOpt/bin/blockMesh
[testuser1@ln1%tinhe2-B ~]$ █
```

Other usage of module can be found from command: **\$module --help**.

(index.htm)

(index.htm)



```
[testuser1@ln1tinhe2-B ~]$ module --help
Modules Release 3.2.7 2009-07-30 (Copyright GNU GPL v2 1991);
Usage: module [ switches ] [ subcommand ] [subcommand-args ]
Switches:
-H|--help      this usage info
-V|--version   modules version & configuration options
-f|--force     force active dependency resolution
-t|--terse     terse format avail and list format
-l|--long      long format avail and list format
-h|--human    readable format avail and list format
-v|--verbose   enable verbose messages
```

hackUST & hardUST

(index.htm)

4 Compiler

(index.htm)



Currently , GNU and Intel compiler are installed on TH-2, which support C, C++. Fortran 77 and Fortran 90. Besides, TH-2 supports both OpenMP and MPI for parallel programming. OpenMP supports shared memory multiprocessing, which can only be parallelized in single computation node. The maximum threads of OpenMP cannot be larger than core number of one node. While MPI supports distributed memory parallel processing, one job can run over one or several nodes. The maximum threads is only limited by the core limitation for different users.

In addition, parallel processing of OpenMP need the support from complier. Both GNU and Intel compiler support this standard.

hackUST & hardUST

4.1 Intel Compiler

Three version of Intel compilers have been installed on TH-2. They are intel 11.1, intel 13 and intel 14. Intel 14 has been configured as the default compiler for user. If there is no specific requirement, user do not need to configure the environment of compiler.

User can also use other version of Intel compiler through loading module. The commands are:

For Intel 13: \$module load intel-compilers/13.0.0

For Intel 11.1: \$module load intel-compilers/11.1

Note: user can use command \$which to find the path of compiling command.

For example, \$which icc will return the path of command \$icc. To identify the version of compiler, please add -v or -V after the compile command, like \$icc -v, \$ifort -V. Details of Intel compiler can be find through command: \$icc –help.

(index.htm)



4.2 GCC Compiler

The default version of GNU compiler on TH-2 is 4.4.6. All related compile command can be found in the directory of /usr/bin/.

4.3 MPI compile environment

TH-2 adopts autonomous interconnected high-speed network, thus the parallel compilation of MPI is based on Intel or GNU compiler.

MPI based on Intel compiler is installed in the directory /usr/local/mpi3. To pursue highest efficiency, MPI installed in this directory is autonomous implementation, while the ground layer is compiled by Intel compiler. The version of /usr/local/mpi3 is recommend for high efficiency.

hackUST & hardUST

Notes for MPI compilation:

1. *The default MPI is in the directory of /usr/local/mpi3, which deploys Intel 14 compiler. The library for this MPI is static, users do not need to care about the problem of dynamic library connection. (If your compilation need to use dynamic library, please use mpi3-dynamic. Load module MPI/Intel/MPICH/3.1-dyn to configure that environment.*
2. *The system also provides other version of MPI, like mpi3-dbg, mpi3-dynamic, mpi3-gcc, mpi3-icc11, mpi3-icc11, mpi3-large and Intel developed impi. Here, mpi3-dbg calls Intel 14 compiler for debugging. mpi3-dynamic calls Intel 14 compiler with dynamic MPI library. mpi3-gcc calls GNU compiler with static library. mpi3-icc11 calls Intel 11 compiler with static MPI library. mpi3-large calls Intel 14 compiler with static library, which supports over 2GB address space and enable the option of -mcmodel=large. Users can switch the version of MPI through module tool.*

(index.htm)

3. If users are going to use other version of MPI, like openmpi1.4.8, mpich2-1.3.1, you need to install your own software. However, the performance may be much lower than the built in MPI of TH-2.

(index.htm)



The compile command of MPI automatically contains the path of stand MPI head files, and connect with necessary MPI message interface library. These parameters do not need to be specified in user's command.

If user is compiling MPI code with "makefile" or "autoconf", you need to change some parameters. like change CC, CXX, F77, F90 to mpicc, mpicxx, mpif77, mpif90 in "makefile" or in the process of configure for "autoconf".

(index.htm)

5 Job submission

(index.htm)



5.1 Check node status with yhinfo or yhi

The command `yhi` is the shortcut for `yhinfo`, which can be used to check the

status of node on TH-2, specifically:

```
[testuser1@ln1:tinhe2-B ~]$ yhi
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
TRIAL up infinite 4 drain* cn[555,764-765,893]
TRIAL up infinite 2 down* cn[751,892]
TRIAL up infinite 1 drng cn686
TRIAL up infinite 33 drain cn[512,519,521,530,557,561,593,598,621,654-656,674,680-681,713-714,723,731,745-746,7
49,752-753,772,800-801,841,847-848,853-854,864]
TRIAL up infinite 344 idle cn[513-518,520,522-529,531-554,556,558-560,562-592,594-597,599-620,622-653,657-673,6
75-679,682-685,687-712,715-722,724-730,732-744,747-748,750,754-763,766-771,773-799,802-840,842-846,849-852,855-863,865-891,89
4-895]
```

“PARTITION” -- shows partitions that users have access to;

“NODES” -- shows the number of nodes;

“STATE” -- “idle” indicates the nodes that are available; “allocated” shows nodes that are currently deployed by users; “drain”/“down”/“drng” shows nodes that are currently not available;

“NODELIST” -- the ID of the nodes

5.2 Check job status with yhqueue or yhq

```
[testuser1@ln1:tinhe2-B ~]$ yhq
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
382974 MIC scut_yrk scut_zpj_1 CG 0:16 1 cn0
388474 SHARE xhpcg_MP root CG 3:11 1 cn8177
```

5.3 Interactive processing job with yhrun

5.3.1 Introduction

Interactive job submission: execute command `yhrun` in the shell interface, the main command would be:

`$yhrun [options] program`

5.3.2 General options of yhrun

`yhrun` contains several options, main options are as follows:

- `-n, --ntasks=number`

(index.htm)

It defines the number of tasks, which requests resource for “number” tasks.

The default request is to assign each task one core.

(index.htm)



- `-c, --cpus-per-task=ncpus`

It tells the resources management system that each task require “ncpus” processors. If not specified, this value is one.

- `-N, --nodes=minnodes[-maxnodes]`

Submitted job requires at least “minnodes” nodes, while “maxnodes” limits the maximum number that one job can deploy (like `-N 2-4` or `--nodes=2-4`).

These two values can be the same (like `-N 2` or `--nodes=2-2`). However, this request will not be satisfied if it exceed the nodes limitation of submitted partition, and the job will not be submitted. If `-N` is not specified, system will assign enough nodes to satisfy the requirement from parameter `-n` and `-c`.

Once the requirement of job falls in the limitation and incurs no delay, as many nodes as possible will be allocated.

- `-p, --partition=partition name`

It specifies the partition of resource. If not, job will run in the default partition.

- `-w, --nodelist=node name list`

It defines the ID of nodes. Assigned resource will include these nodes. This list can be defined by node ID separated by commas, or node list (like `cn[1-5, 7,...]`), or file name. When “/” is included in the parameter, it will be recognized as file name. If the maximum node number is specified, like `-N 1-2`, while the file contains extra two nodes, the submitted job will only run on first two nodes.

- `-x, --exclude=node name list`



(index.htm)

It lists the nodes that will be excluded for a job. When it contains “/”, the parameter will be recognized as a file name. yhrun will submit the job request to control process, and start all processes over remote nodes. If the request cannot be satisfied instantly, yhrun will start until all the resource requested is available. If user adds the --immediate option, yhrun will terminate the job once there are not enough resource.

- -h, --help

User can learn more options of yhrun from the command \$yhrun -h or \$yhrun --help.

5.3.3 Examples

- 1) Run “hostname” in the partition MIC with specified number of task:

```
[testuser1@ln2%tinhe2-B ~]$ yhrun -n 4 -p MIC hostname
cn6
cn6
cn6
cn6
[testuser1@ln2%tinhe2-B ~]$ yhrun --ntasks 4 -p MIC hostname
cn6
cn6
cn6
cn6
[testuser1@ln2%tinhe2-B ~]$ █
```

- 2) Run “hostname” in the partition of MIC on nodes “cn[145-157]”:

```
[testuser1@ln2%tinhe2-B ~]$ yhrun -n 3 -w cn[145-147] -p MIC hostname
cn145
cn146
cn147
[testuser1@ln2%tinhe2-B ~]$ █
```

- 3) Run “hostname” in the partition of MIC with four tasks, each node is assigned with one task. In addition, nodes “cn[145-147]” are included in the assigned nodes:

```
[testuser1@ln2%tinhe2-B ~]$ yhrun -n 4 -N 4 -w cn[145-147] -p MIC hostname
cn146
cn147
cn148
cn145
[testuser1@ln2%tinhe2-B ~]$ █
```

- 4) Run “hostname” in the partition of MIC with four tasks, each node is assigned with one task. In addition, nodes “cn[145-147]” are excluded from the assigned nodes:

```
[testuser1@ln2%tinhe2-B ~]$ yhrun -n 4 -N 4 -x cn[145-147] -p MIC hostname
cn202
cn203
cn204
cn205
[testuser1@ln2%tinhe2-B ~]$ █
```

(index.htm)



(index.htm)

5.4 Batch processing job with yhbatch

5.4.1 Introduction

Batch processing job means user need to compile job script first, which defines the resource requirement and run job in the background. The command for batch processing is yhbatch. The command interface will be returned once user enter this command, while the job is scheduling at the same time. Once the resource requirement is satisfied, the system will load and execute the user's script on the first allocated computation node (not on login service node).

The script of batch processing is a text file, whose first line starts with “#!” . It defines the interpretation program, like “sh”, “bash”. Since the environment of computation node is simplified, it only supports “sh” and “bash”.

5.4.2 Examples

Content of a user script “mybash.sh” are follows:

```
[testuser1@ln1 tinhhe2-B test]$ vi mybash.sh
#!/bin/bash
yhrun -n 4 -N 4 -p MIC hostname
```

According to this script, user identifies that the job needs four nodes of partition MIC.

```
[testuser1@ln1 tinhhe2-B test]$ ll
total 4
-rw-rw-r-- 1 testuser1 testuser1 56 Jun 13 10:49 mybash.sh
[testuser1@ln1 tinhhe2-B test]$
```

Note:

The text file “mybash.sh” should be authorized with executable permission. The command is:

```
$chmod +x mybash.sh
```

```
[testuser1@ln1 tinhhe2-B test]$ chmod +x mybash.sh
[testuser1@ln1 tinhhe2-B test]$ ll
total 4
-rwxrwxr-x 1 testuser1 testuser1 56 Jun 13 10:49 mybash.sh
[testuser1@ln1 tinhhe2-B test]$
```

The command for yhbatch is:

```
yhbatch -N 4 -p MIC ./mybash.sh
```

Once computation starts, an “.out” file whose name starts with “slurm” will be produced in the directory.

(index.htm)

User can learn more options through command \$yhbatch –help.

(index.htm)

5.5 Resource occupy command with yhalloc



5.5.1 Introduction

This command allows user to occupy computation resource before submitting job (CPU hours will start to count from this moment).

5.5.2 Examples

The command of yhalloc is used in follow ways:

Occupy resource with following command:

```
[testuser1@ln1 test]$ yhalloc -N 1
yhalloc: Granted job allocation 391753
```

Check the ID of submitted job with command yhq, which is “391753” on node “cn1483”.

```
[testuser1@ln1 test]$ yhq
      SPUZ61      SHARE xnpccg.MW      root   R    1:04:25    1b cn1483-4/,5b,58-59,6/-68, /2-/9
      391753      WORK   bash    testuser1   R    0:19    1 cn1483
```

User can log in the computation node through following command and execute program:

\$ssh NODELIST

```
[testuser1@ln1 test]$ ssh cn1483
The authenticity of host 'cn1483' (<no hostip for proxy command>) can't be established.
RSA key fingerprint is f1:dd:eb:26:58:f8:d3:67:c0:fd:66:8d:11:f:dc.
Are you sure you want to continue connecting (yes/no)? yes
```

5.6 Cancel job with yhcancel

hackUST & hardUST

User can use the command yhcancel to cancel one's own job, the format should be : \$yhcancel JOBID

User can get “JOBID” from the command yhq. For queuing jobs, cancelling job will simply mark those jobs as “CANCELLED”. For running or suspending jobs, yhcancel will terminate all steps of a job, including interpret batch processing script. Then system will mark job as “CANCELLED” and reclaim allocated nodes. Generally speaking, batch processing job will be terminated immediately. The process

(index.htm)

of yhrun for interactive processing jobs will be terminated once informed of job cancellation. The process of occupying resource for yhalloc will not quit until the job finishes. However, when cancelling a job, control process will inform the resource allocation process of yhrun or yhalloc. User can terminate a job through adding the option --kill-command for yhalloc.

(index.htm)



(index.htm)

6 FAQ

(index.htm)



hackUST & hardUST

- ✓ **VPN account connects successfully, while the client software cannot connect to TH-2.**

Please first check whether your computer has installed any security software. If so, please close these programs and reconnect VPN. If this problem still exists, please ping system IP and check packet loss. If packet loss is high, the problem is due to your own network speed. If that value is low, please contact with our customer service.

- ✓ **Webpage to check used CPU hours and job history**

To check used and remaining CPU hours and job history of your account, please visit the following webpage when connected to VPN:

<https://172.16.22.11:10021/index/>

- ✓ **Command to access directory or file is slow, like the command ls.**

There are three reasons for this problem: (1) the network is slow with high latency. (2) There is a large amount of I/O operations which block communication. (3) Excess amount of files in the directory. For case (1) and (2), it will back to normal after a few moments. For case (3), users are recommended to save files in different directories.

- ✓ **Regenerate Private Key**

Private Key can be regenerated. User need to follow steps below once login TH-2.

Step 1:

```
$ cd ~/.ssh
```

```
$ tar cvf bak.tar *
```

Step 2:

```
$ ssh-keygen -t rsa    ( keep pushing "Enter" )
```

(index.htm)

```
$ cd ~/.ssh
$ cp id_rsa.pub authorized_keys
$ chmod 400 authorized_keys
$ cd ..
$ chmod 700 -R .ssh
```

(index.htm)

**Step 3:**

Copy the content of “id_rsa” (for Private Key Linux version) to local text file.

Create a new terminal to login with Private Key file. If it works, the regeneration of Private Key is success. If not, please repeat step 2 and 3. If it fails after several trials, please unpack the packet generated in the first step (The command would be \$tar -xvf bak.tar), and use the old Private Key to login.

- ✓ **Error of “Invalid partition name specified”**

User are suggested to check whether that partition is available with command yhi. If not, it means your CPU hours has reached the limitation.

- ✓ **Error of “Failed to allocate resources: User's group not permitted to use this partition”**

When submit a job, user should add a parameter -p partition before code name. The available partition can be found through command yhi.

- ✓ **Submit job with yhrun, the job is killed once the interface closes.**

The job submission mode of yhrun is interactive processing. The job will be killed once the submitting interface closes. Users are recommended to submit job with batch processing mode yhbatch if it require long run time. The submitted script with yhbatch will run on the first allocated node. Job will still run even after the shell interface closes.

(index.htm)

- ✓ **Failure of job on multiple nodes with yhbatch.**

(index.htm)



The first step of command yhbatch is resource allocation. For jobs run on multiple nodes, user should add parameter -N in the command, which is:

```
$yhbatch -N nodenum -n prnum -p partition job.sh
```

- ✓ **User cannot login computation node.**

User are not allowed to login computation node unless the node is allocated by the resource management system. If user wants to login computation node before job submission, yhalloc should be used to allocate nodes first.

- ✓ **The job is killed once the quit yhalloc.**

The main difference between yhalloc and yhbatch is that: Job will perform on allocated nodes once the requirement for resource is satisfied. It is good for job that requires specification of node or other resource. Allocated resource and running job will quit once the login shell exit.

- ✓ **How to handle some error of job.**

Normal error messages like: “No enough endpoint resources”, “Job credential expired”, “bus error” can be handled by excluding error nodes with adding parameter “-x” when submitting job. For example, user can add parameter -x cn1 to avoid allocating node “cn1”. Information of error nodes can be found in the log file. Users are welcome to report errors to our customer service for further analyze and treatment.

- ✓ **How to handle missing library for some jobs.**

User can find relevant library with command locate and add the path in directory of /BIGDATA1 into environment variable “LD_LIBRARY_PATH”. If it asks for system library, user can try following command before job submission:

```
$source /BIGDATA1/app/osenv/ln1/set2.sh.
```

(index.htm)

If it still does not work, please copy the missing library file to your own directory, like “~/lib”, and configure the environment variable as:

(index.htm)

```
$export LD_LIBRARY_PATH=~/lib:$LD_LIBRARY_PATH
```



✓ **Job submission mode of TH-2.**

TH-2 adopts exclusive job submission mode. Allocated node will be exclusively used by single user. It means that processors on allocated node cannot be used by other users even if they are not fully deployed.

✓ **The status of job is “CG” after job quit.**

The status of CG means that some processes are not fully terminated, but it would not affect normal job quit.

✓ **Part of processes are killed when job finishes, and then quit.**

User should check whether system completes all necessary output. The reason for such problem is that some processes complete computation earlier, while others are still running. However system recognizes that as the whole job has been finished. If those processes cannot finish in a short time range, they are forced to be terminated.

POWERED BY CICAPP

(index.htm)



(index.htm)

hackUST & hardUST