

# 天河系列超级计算机入门使用培训

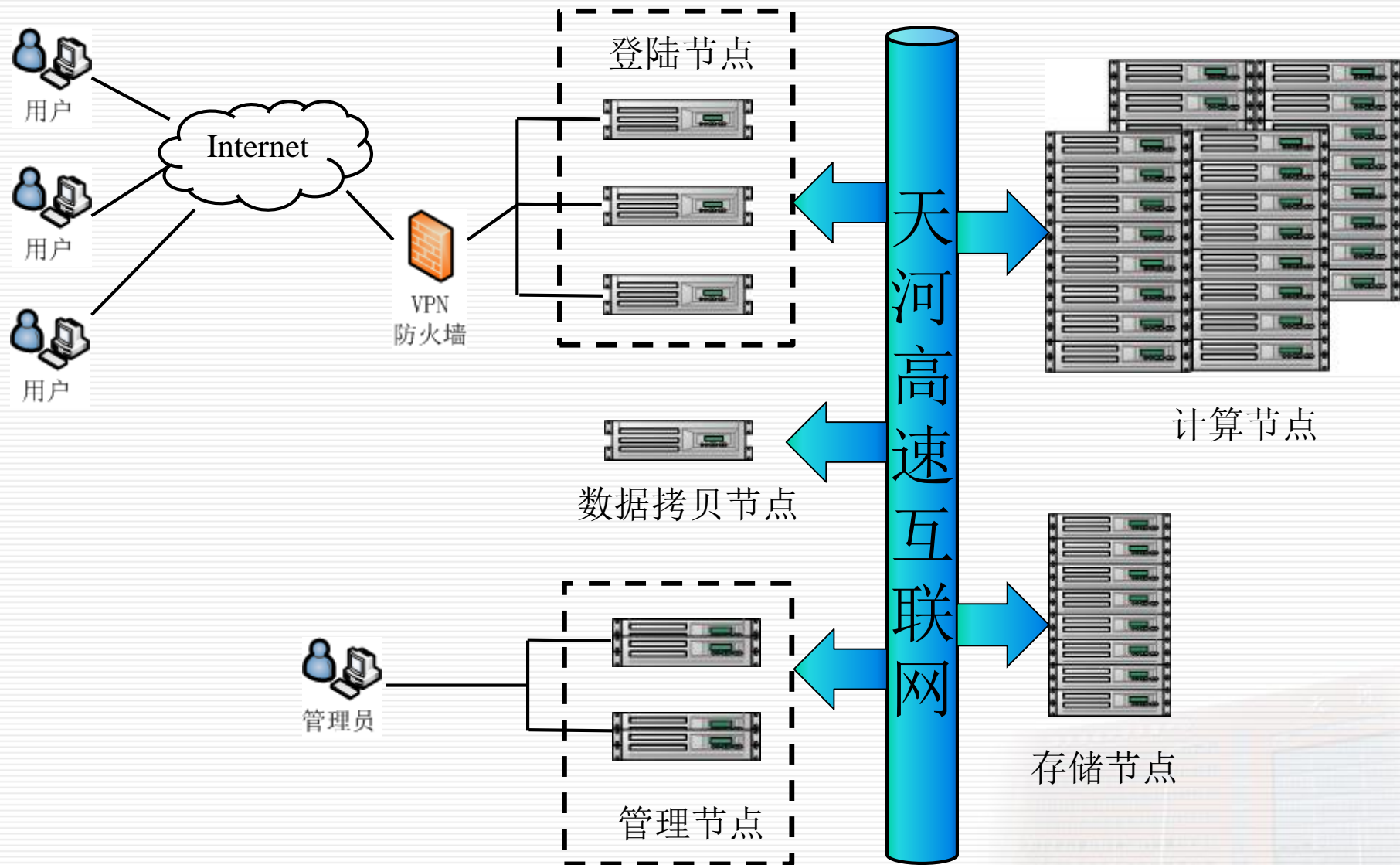
高性能计算部 郑刚

zhenggang@nsc-tj.cn



- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍

- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍



## ◆ 登陆节点

登陆节点为用户提供一个登陆系统的平台，用户可以通过Internet网络登录VPN，然后通过SSH终端登录到登陆节点上；麒麟操作系统，具备软件编译环境，用户可以登录的是LN1, LN2, LN3, LN8, LN9；

LN1, LN2, LN8作为用户的登陆节点；

LN3, LN9作为用户的登陆节点，同时额外提供数据传输功能；

允许操作：在登陆节点上用户可以进行软件编译与调试,环境变量配置作业提交，文件编辑，结果查看等操作。

禁止操作：禁止用户直接在登陆节点上运行计算程序。

## ◆ 计算节点

表 1-1 计算节点配置

<b>CPU*2</b>	型号	Intel Xeon X5670	核心数量	6	主频:	2.93GHz
<b>GPU</b>	型号	Fermi M2050	核心数量	448	显存:	3GB
<b>内存</b>	24 GB / 48GB					
<b>硬盘</b>	无					
<b>操作系统</b>	版本	RHEL 5.3	内核	2.6.32-358.123.4-TH		

CPU*2	型号	2*Intel Xeon CPU E5-2690 v4 @ 2.60GHz 14cores		
L3 缓存	35MB		内存	128GB
GPU	2*Tesla K80， 即 4*GK210			
显存	4*12GB			
驱动版本	375.26			
编译环境	CUDA 8.0			
硬盘	无			
操作系统	版本	RedHat 7.2	内核	3.10.0

### ◆存储节点

采用分布式存储文件系统，该文件系统由多个存储节点构成，对外提供一个统一的大分区，供所有登陆节点与计算节点进行挂载。

### ◆文件系统

集中式管理的分布式lustre文件系统；目前天河共有四套文件系统。

### ◆互联网络

天河高速自主互联网络，单点通信带宽80Gbps，低通信延迟。

### ◆作业管理系统

基于开源slurm进行二次开发的作业调度管理系统。

- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍



### ◆ 共享目录

TH1A     /vol-th、/vol6、/WORK、/THL4.

HPC     /THL6、/THL7

### ◆ 用户根目录（以vol-th为例）

/vol-th/home

### ◆ 常用软件安装目录

/vol-th/software

### ◆ 常用动态链接库目录

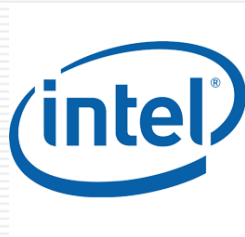
/vol-th/lib

/vol-th/lib/mklem64t

/vol-th/intel\_composer\_xe\_2013.0.079\_lib

## ● 编译器

在TH-1A大系统的登陆节点中，目前安装了Intel编译器、GCC编译器以及PGI编译器。用户可根据程序需求，选择相应的编译器进行编译和应用程序开发，由于目前上线的天河系统广泛采用了Intel的CPU，因此在编译中除特定需要，建议用户首选Intel编译器。另外，在TH-1A大系统的登陆节点上还提供了MPI并行编译环境，以及针对GPU的CUDA编译环境。





## ● 1) Intel编译器

用户在登陆节点上使用Intel编译器进行程序编译时需添加如下环境变量声明（以 Composer\_xe\_2013.0.079 为例）

```
source /opt/intel/composer_xe_2013.0.079/bin/iccvars.sh intel64  
source /opt/intel/composer_xe_2013.0.079/bin/ifortvars.sh intel64  
source /opt/intel/composer_xe_2013.0.079/mkl/bin/mklvars.sh intel64
```

用户在计算节点上提交作业运行时，如需要调用Intel编译器的动态库，则需要添加如下环境变量声明：

```
export LD_LIBRARY_PATH=/vol-th/intel_composer_xe_2013.0.079_lib:$LD_LIBRARY_PATH  
export LD_LIBRARY_PATH=/vol-th/lib:$LD_LIBRARY_PATH
```



## ● 2) GCC编译器



系统自带 4.4.7 版本

`/usr/bin/gcc`

共享目录 4.6.1/4.9.4/6.4.0 版本

```
# gcc4.6.1
```

```
export PATH=/vol-th/software/gcc-4.6.1/bin:$PATH
```

```
export LD_LIBRARY_PATH=/vol-th/software/gcc-4.6.1/lib64:$LD_LIBRARY_PATH
```

```
# gcc4.9.4
```

```
export PATH=/vol-th/software/gcc-4.9.4/bin:$PATH
```

```
export LD_LIBRARY_PATH=/vol-th/software/gcc-4.9.4/lib64:$LD_LIBRARY_PATH
```



### ● 3) PGI编译器

PGI 16.4:

TH-1A 大系统上的 ln2 登陆节点上安装有 PGI 编译器，版本为16.4。

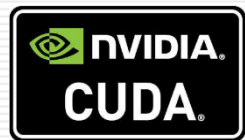
目前购买的 license 与 ln2 登陆节点的物理地址绑定。

```
export PGI=/opt/pgi-2016-164-ln2/linux86-64/16.4
```

```
export PATH=$PGI/bin:$PATH
```

```
export LD_LIBRARY_PATH=/vol-th/pgi-2016-164-ln2-lib:$LD_LIBRARY_PATH
```

```
export MANPATH=$PGI/man:$MANPATH
```



#### ● 4) CUDA编译器

CUDA编译环境包含三个部分，编译器、SDK和设备驱动；目前计算节点CUDA编译环境和设备驱动，均已经更新至CUDA7.5。用户使用CUDA进行程序编译时需添加如下环境变量声明（以cuda7.5为例）：

```
export PATH=/vol-th/software/cuda/cuda-7.5/bin:$PATH
```

```
export LD_LIBRARY_PATH=/vol-th/software/cuda/cuda-7.5/lib64:$LD_LIBRARY_PATH
```

## ● 5) MPI编译器

由于 TH-1A 采用了自主互连的高速网络，因此底层 MPI 为自主实现。使用天河系统提供的 MPI 进行并行编译可以充分发挥天河高速网的性能，提高并行效率，推荐用户使用基于 Intel 编译器编译的 MPI，如 MPI/mpich/intel2013。

## ● 5) MPI编译器

类别	编译器	编译器版本	MPI版本实现	目录名称
基于天河高速网	Intel	11.1	MPICH2-1.4.1	mpi_1.4.1_intel_11.1
		13.1	MPICH-3.0.4	mpi-intel2013
	GCC	4.1	MPICH2-1.2.1	mpi_1.2.1_gcc_4.1
		4.6	MPICH2-1.4.1	mpi_1.4.1_gcc_4.6
		4.9	MPICH-3.0.4	mpich_3.0.4_gcc_49
	PGI	16.04	MPICH-3.0.4	mpi-pgi-2016-164-ln2



## ● 编译、运行并行程序举例

用户使用Intel2013编译器+mpi-intel2013编译器进行程序编译运行

```
source /opt/intel/composer_xe_2013.0.079/bin/iccvars.sh intel64
```

```
source /opt/intel/composer_xe_2013.0.079/bin/ifortvars.sh intel64
```

```
source /opt/intel/composer_xe_2013.0.079/mkl/bin/mklvars.sh intel64
```

```
export MPI_HOME=/vol-th/software/mpi/mpi-intel2013
```

```
export PATH=$MPI_HOME/bin:$PATH
```

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

```
export LD_LIBRARY_PATH=/vol-th/intel_composer_xe_2013.0.079_lib:/vol-th/lib:$LD_LIBRARY_PATH
```

## ● Module 常用命令

命令	作用
module av	查看系统中可用的软件
module help	查看具体软件的信息
module add   load	加载环境变量
module rm   unload	卸载环境变量
module list	显示用户已加载的环境变量
module show	显示具体的 Module 配置文件
module swap   switch	替换环境变量
module purge	卸载当前 shell 环境下的所有环境变量

## ● module av

```
[liqf@th-hpc2-ln0 ~]$ module av
```

```
----- /THL7/software/modules/3.2.10/Modules/3.2.10/THL7/Applications -----
abinit/8.2.2-icc16-mvapich2(default)  espresso/5.4.0-icc16-openmpi2  hifiles/1.0.0-icc16-IMPI5.1(default)  phonopy/1.11.10.4-py27-gcc4.8
activeTcl/8.5.18(default)             espresso/6.1-icc16-IMPI5.1     hifiles/1.0.0-icc16-mvapich2          phonopy/1.11.2-py27-gcc4.8(default)
autodock/4.2.6(default)               espresso/6.1-icc16-mvapich2    hmmer/3.1b2-icc16-mvapich2(default)   plink2/20170320(default)
autodock_vina/1.1.2(default)          espresso/6.1-icc16-openmpi2    impute2/2.3.2-static(default)         samtools/1.2-gcc4.8(default)
bcftools/1.2-gcc4.8(default)          espresso/6.1-icc17-IMPI2017.4  lammps/11Aug17-icc16-IMPI5.1          scalasca/2.3.1-icc16-IMPI5.1(default)
beagle/4.0(default)                  espresso/6.2.1-icc16-IMPI5.1   lammps/11May18-icc16-IMPI5.1          scorep/3.0-icc16-IMPI5.1(default)
bedtools2/2.25.0-gcc4.8              espresso/6.2.1-icc16-mvapich2  lammps/17Nov16-icc16-IMPI5.1(default) siesta/4.0-icc16-IMPI5.1(default)
bedtools2/2.26.0-gcc4.8(default)      espresso/6.2.1-icc16-openmpi2  lammps/22Jul16-default-icc16-IMPI5.1  siesta/4.0-icc16-mvapich2
bedtools2/4.2.6                      espresso/6.2.1-icc17-IMPI2017.4 lammps/22Jul16-default-icc16-mvapich2  siesta/4.0-icc17-IMPI2017.4
berkeleyGW/1.0.6-real-icc16-mvapich2  espresso/6.3-icc17-IMPI2017.4 lastz/1.04.00-gcc4.8(default)         tophat/2.1.1(default)
blast/2.6.0(default)                 fastqc/0.11.5                 namd/2.12-icc16-IMPI5.1(default)       varScan/2.3.9(default)
blat/36(default)                     ffmpeg/3.2.4-gcc4.8(default)   namd/2.12-icc16-mvapich2              vcftools/0.1.14-icc16(default)
boltztrap/1.2.5-icc16(default)        fishingCNV/1.5.3(default)      ncview/2.1.7-icc16-mvapich2(default)  velvet/1.2.10-gcc48(default)
bowtie/1.2.0-gcc4.8(default)          gromacs/4.6.7-double-icc16-IMPI5.1  nektar++/4.4.1-icc16-mvapich2(default) vmd/1.9.3
bowtie2/2.3.1-gcc4.8(default)         gromacs/4.6.7-single-icc16-IMPI5.1  nwdchem/6.6-icc16-mvapich2(default)   vmd/1.9.3-Text-mode(default)
breakdancer/1.4.5-gcc4.8             gromacs/5.1.2-double-icc16-IMPI5.1  openexr/2.2.0-icc16(default)          vtstscripts/935(default)
bwa/0.7.15-gcc4.8(default)           gromacs/5.1.2-double-icc16-IMPI5.1-AVX256  openfoam/3.0.1-gcc4.8-mvapich2       wannier90/1.2-icc16(default)
cnvnator/0.3-gcc4.8(default)         gromacs/5.1.2-double-icc16-mvapich2  openfoam/3.0.1-icc16-mvapich2(default) wannier90/2.0.0-icc16-IMPI5.1
control-freec/10.5(default)          gromacs/5.1.2-single-icc16-IMPI5.1(default)  openmx/3.8-icc17-IMPI2017.4(default)  wannier90/2.1.0-icc16-IMPI5.1
cp2k/4.1-icc16-IMPI5.1(default)      gromacs/5.1.2-single-icc16-IMPI5.1-AVX256  packmol/17.076-gcc4.8                wps/3.8-icc16-mvapich2(default)
espresso/5.4.0-icc16-IMPI5.1(default) gromacs/5.1.2-single-icc16-mvapich2  packmol/17.076-icc16(default)         wrf/3.8-emreal-dmpar-icc16-mvapich2(default)
espresso/5.4.0-icc16-mvapich2        harminv/1.4-icc16(default)      phono3py/1.10.9-py27-gcc4.8          yambo/4.1.4-icc16-IMPI5.1(default)
----- /THL7/software/modules/3.2.10/Modules/3.2.10/THL7/Compilers -----
GCC/4.8.5                             java/jdk1.8.0_121(default)      MPI/mpich-3.1/intel2016u3             MPI/mvapich2-2.2/pgi16.10          PGI/16.10(default)
Intel_compiler/16.0.3(default)        MPI/Intel/IMPI/5.1.3.210       MPI/mvapich2-2.2/gcc4.8.5            MPI/openmpi-2.0.2/gcc4.8.5       yasm/1.3.0-icc16(default)
Intel_compiler/16.0.4                 MPI/mpich-3.1/gcc4.8.5         MPI/mvapich2-2.2/intel2016u3         MPI/openmpi-2.0.2/intel2016u3
```

## ● module add / module list

```
[liqf@th-hpc2-ln0 ~]$ module add Intel_compiler
[liqf@th-hpc2-ln0 ~]$ module add MPI/Intel/IMPI/5.1.3.210
[liqf@th-hpc2-ln0 ~]$ module list
Currently Loaded Modulefiles:
  1) Intel_compiler/16.0.3          2) MPI/Intel/IMPI/5.1.3.210
[liqf@th-hpc2-ln0 ~]$
```

- module show

```
[liqf@th-hpc2-ln0 ~]$ module show wrf
-----
/THL7/software/modules/3.2.10/Modules/3.2.10/THL7/Applications/wrf/3.8-emreal-dmpar-icc16-mvapich2:
module-whatism wrf/3.8-emreal-dmpar-icc16-mvapich2
prereq Intel_compiler/16.0.3
prereq MPI/mvapich2-2.2/intel2016u3
prereq netcdf/4.4-icc16-mvapich2
conflict wrf
prepend-path PATH /THL7/software/wrf/3.8-emreal-dmpar-icc16-mvapich2/main
prepend-path PATH /THL7/software/wrf/3.8-emreal-dmpar-icc16-mvapich2/run
prepend-path INCLUDE /THL7/software/wrf/3.8-emreal-dmpar-icc16-mvapich2/inc
setenv WRF /THL7/software/wrf/3.8-emreal-dmpar-icc16-mvapich2
-----
```

- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍

## ◆ 前提条件

1.拥有VPN账号和密码；（网络环境）

2.拥有系统账号和密码；（登录系统）

3.windows系统登陆终端ssh工具

(Xmanage, SSHSecureShell, Putty等等)

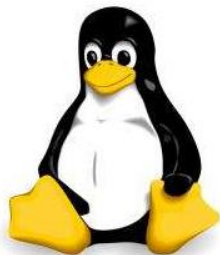
4.确认用户当前网络：

联通入口：<https://thvpn.nscs-tj.cn>

电信入口：<https://thvpn1.nscs-tj.cn>

## ◆ VPN 登陆

目前 TH-1A 系统已经同时接入了联通和电信双网络，用户根据自身网络接入商的不同可以选择不同的登陆域名来登录 VPN。





## ◆Windows系统登陆新VPN

推荐使用IE浏览器，以及基于IE核心的浏览器

联通用户浏览器输入 <https://thvpn.nsccln.cn>;

电信用户浏览器输入 <https://thvpn1.nsccln.cn>.



NSCC  
国家超级计算天津中心

登录SSL VPN

用户名

密码

登录

其它登录方式：

证书登录 USB-Key登录

- 自动安装组件失败，请手动 [下载安装组件](#)
- 登录异常，请下载修复工具 [尝试修复](#)
- 需要更多帮助信息，请 [点击这里](#)

输入用户名、密码登录SSL VPN

为IE浏览器安装控件（仅初次配置时需要安装）

使用IE浏览器或基于IE内核的浏览器登录，会提示安装Active X浏览器控件，如下图：





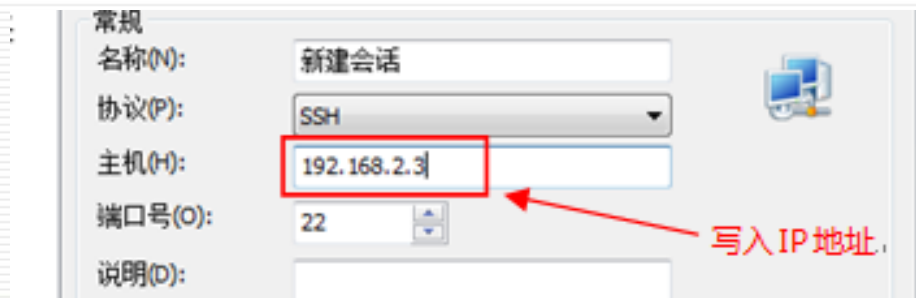
安装好 Active X 浏览器控件，用户就可以正常登陆VPN；

## ◆ Windows系统终端登陆

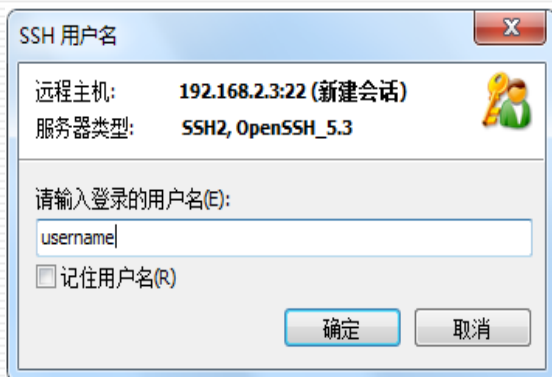
- 1) 再次登陆，输入VPN账号和密码，会成功登陆VPN网页。
- 2) 查看并记录不同登陆节点（例如TH-1A-LN1）的ip地址，如下图：



- 3) 打开ssh登录工具如Xshell等，点击新建会话按钮，输入资源对应的IP地址，点击确定。



继续输入系统登陆的账户名和账户密码



SSH 用户名

远程主机: 192.168.2.3:22 (新建会话)

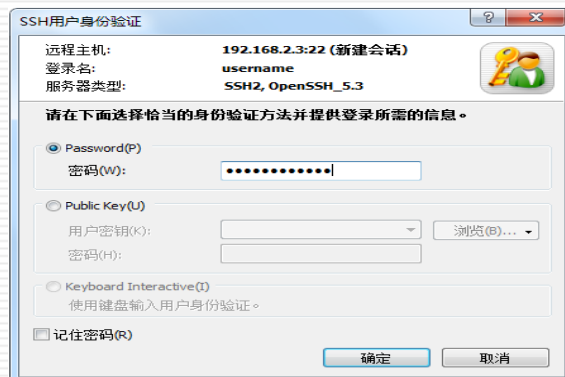
服务器类型: SSH2, OpenSSH\_5.3

请输入登录的用户名(E):

username

☐ 记住用户名(R)

确定 取消



SSH 用户身份验证

远程主机: 192.168.2.3:22 (新建会话)

登录名: username

服务器类型: SSH2, OpenSSH\_5.3

请在下面选择恰当的身份验证方法并提供登录所需的信息。

☒ Password(P)

密码(W): .....

☐ Public Key(U)

用户密钥(K): [下拉菜单] 浏览(B)...

密码(H): [输入框]

☐ Keyboard Interactive(I)

使用键盘输入用户身份验证。

☐ 记住密码(R)

确定 取消

登陆LN1后，会收到如下提示：

1.Welcome to TH-1A System of NSCC-TJ.

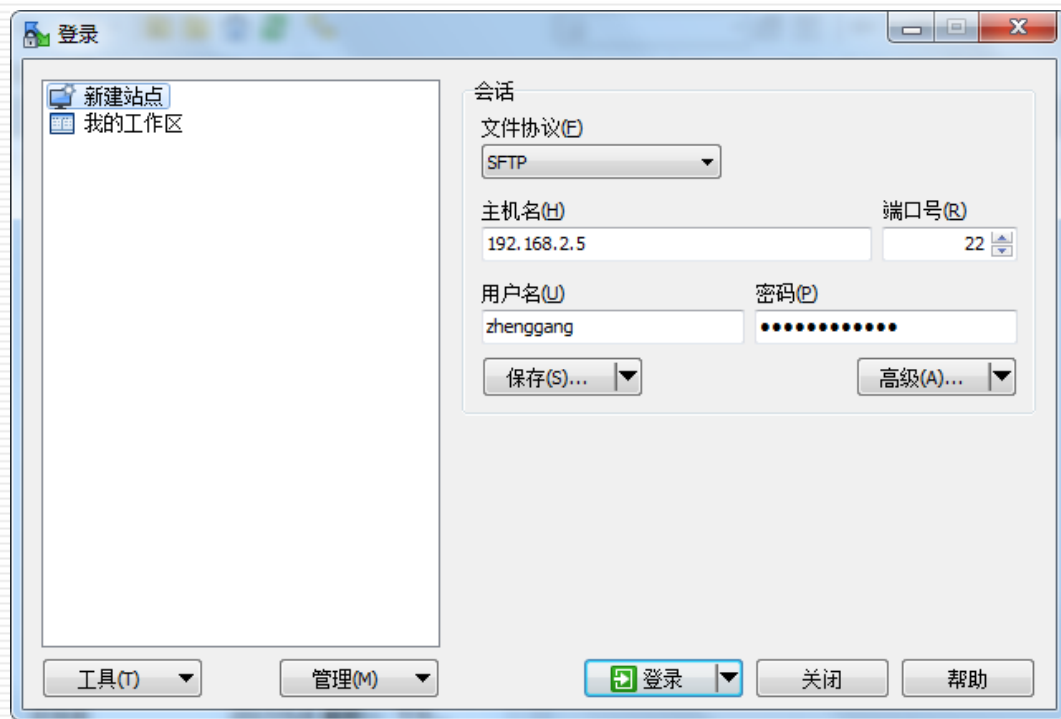
2.If you have any problem, you can send mail to support@nsccl-tj.gov.cn

之后您即可以开始编译、提交任务等操作。

```
Last login: Fri Apr 15 23:36:55 2016 from 10.200.200.67
*****
* 1.Welcome to TH-1A System of NSCC-TJ.
* 2.If you have any problem, you can send mail to support@nsccl-tj.gov.cn
*****
Disk quotas for group username (gid 1711):
  Filesystem  kbytes  quota  limit  grace  files  quota  limit  grace
  /vol-th    268435456  524288000  1073741824  -    500000  1000000  2000000
[username@ln1%tianhe ~]$
```

## ◆ 文件传输

建议用户使用WinScp的sftp数据传输软件（免费软件，网络容易下载，且该软件支持断点续传，推荐使用）



- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍

## 作业管理系统

- 以CPU核作为并行作业的资源分配单位，实现并行作业的调度运行。**在我们的系统中，为保证用户程序的稳定运行，计算结点采用抢占式的方式供用户使用。**
- 所有在计算结点中运行的串行或并行应用程序，都必须通过资源管理系统来提交运行。
- 资源管理系统针对排队的作业，根据实时运行资源状态和排队作业优先级，决定在哪些计算结点中何时运行应用程序。**优先级原则：先提交先算。**
- 不同的应用程序之间不存在资源的竞争冲突，用户也可以通过作业管理系统来监控应用程序的运行。如：yhc。

## ● 查看节点状态 --- yhi

用户可以使用yhi命令查看节点的使用情况



```
[liqf@th-hpc2-ln0 ~]$ yhi
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug2      up       30:00      4   idle cn[552-555]
TH_LONG2    up    10-00:00:0      1 down* cn924
TH_LONG2    up    10-00:00:0    401 alloc cn[572-623,625-645,647-655,658-6
60,963,965-971,973-997,1004-1034,1036-1063]
TH_LONG2    up    10-00:00:0     25  idle cn[624,646,656-657,669-670,673-6
[liqf@th-hpc2-ln0 ~]$
```

PARTITION 表示分区,      TIMELIMIT      表示该分区的时间限制,  
NODES      表示节点数,      STATE      表示节点运行状态。



## ● 查看节点状态 --- yhi



注意：

- 由于大型集群系统具备一定故障率，TH-1A大系统系统十分庞大，为了保证系统稳定性，分区中有限定任务执行时间的限制，因此建议用户为程序设立“断点”从而保证任务由于意外中断后，可以继续运算。
- 如果您的程序没有办法“续算”，而且运行时间超过时间队列的限制，请联系中心技术人员。





## ● 作业提交

目前 TH-1A 大系统部署的资源管理系统包括多种作业提交方式，包括交互模式 `yhrun` 和批处理模式 `yhbatch`

### yhrun 交互模式

用户的作业命令在提交节点上运行，关闭当前登陆终端会导致程序退出

### yhbatch 批处理模式

用户的作业脚本在分配的节点上运行，关闭当前登陆终端不会影响作业运行

● 交互式作业提交 --- yhrun



```
yhrun -N 20 -n 240 -p TH_NET ./a.out
```

- N 指定要运行的节点数
- n 指定要运行的进程数
- p 指定使用的队列
- w 请求指定列表中的结点 -w cn[32-51]
- x 排除指定列表中的结点 -x cn[32-51]
- J 指定作业的名字



## ● 批处理式作业提交 --- yhbatch

编写提交脚本

sub.sh, 使用yhbatch命令提交。

1.编写脚本sub.sh

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

```
yhrun -N 20 -n 240 -p TH_NET ./a.out
```

2.提交脚本

```
yhbatch -N 20 -n 240 -p TH_NET ./sub.sh
```

## ● 查看作业状态 --- yhq

用户可以使用yhq命令查看作业状态（仅限自己账户）



```
[zhenggang@ln1%tianhe ~]$ yhq
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
6167633	debug	test.sh	zhenggang	PD	0:00	1	(Priority)
6167645	TH_SR	1mp_th	zhenggang	R	2:30:48	8	cn[4901-4908]
6167836	TH_SR	1mp_th	zhenggang	R	2-12:30:39	20	cn[4457,4469-4488]

关键词	含义
JOBID	任务 ID
PARTITION	用户作业正在使用的计算分区
NAME	任务名称
USER	用户名
ST	作业状态：R: 运行状态, PD: 排队等待状态, S: 挂起状态, CG: 作业正在退出状态
NODES	结点个数
NODELIST	结点名字列表
REASON	Priority: 优先级 Association Resource Limit: 关联的资源限制已满 Resources: 当前可用资源不能满足作业需求 Partition Down: 作业所在的分区处于 down 状态

## ● 作业取消 --- yhcancel

取消已提交的作业

通过yhq命令查看作业队列，获取作业id号

取消作业命令：yhcancel 6167937

取消某用户全部作业命令：

yhcancel -u user



## ● 作业提交及操作视频介绍

### 天河系统编译、作业管理方法实例

国家超级计算天津中心 郑刚  
zhenggang@nscg-tj.gov.cn

- 一、系统平台架构
- 二、系统及编译环境
- 三、系统登陆
- 四、作业管理
- 五、天河应用环境完善介绍

类别	软件列表	路径
文件读写	zlib	/vol-th/software/io_tools/zlib/1.2.6
	hdf5	/vol-th/software/io_tools/hdf5/serial/1.8.11
		/vol-th/software/io_tools/hdf5/mpi/1.8.11
	netCDF	/vol-th/software/io_tools/netcdf/serial/4.1.2
		/vol-th/software/io_tools/netcdf/mpi/4.1.2
	pnetcdf	/vol-th/software/io_tools/pnetcdf/1.6.1
	ncl	/vol-th/software/io_tools/ncl/6.3.0
数据后处理	nco	/vol-th/software/io_tools/nco/4.4.1
	jasper	/vol-th/software/graphics/jasper/1.900.1
数学库	libpng	/vol-th/software/graphics/libpng/1.6.17
	fftw	/vol-th/software/libraries/fftw/serial/3.3.3
		/vol-th/software/libraries/fftw/mpi/3.3.3

注：以上安装依赖编译器的软件在安装时全部使用Intel Compiler 编译生成



类别	软件列表	路径	版本号
气候气象	WRF	/vol-th/software/WRF	3.4.1
		/vol-th/software/WRFV3.7.1/WRFV3	3.7.1
		/vol-th/software/WRFV3.8_pnetcdf	3.8
	CESM	/vol-th/software/CESM	1.2.2
	ROMS	/vol-th/software/roms-201504	201504
分子动力学模拟	NAMD	/vol-th/software/NAMD	2.9-2.12 CPU/GPU
	Gromacs	/vol-th/software/gromacs506_cpu or gromacs512_cpu	5.0.6/5.1.2 CPU
		/vol-th/software/gromacs506_gpu or groamcs512_gpu	5.0.6/5.1.2 GPU
	Lammps	/vol-th/software/lammps/lmp_mpi	2015.12.07
计算材料学	Quantum ESPRESSO	/vol-th/software/pwscf	5.0.2
		/vol-th/software/espresso-5.0.2	5.0.2
	Yambo	/vol-th/software/yambo	4.0.1
	SIESTA	/vol-th/software/Siesta/siesta-3.2	3.2
	Wannier90	/vol-th/software/wannier90/wannier90-1.2.x	1.2
		/vol-th/software/wannier90/wannier90-2.0.x	2.0
	BerkeleyGW	/vol-th/software/BerkeleyGW	1.0.6
	ABINIT	/vol-th/software/Abinit/abinit	7.6.1
	CP2K	/vol-th/software/CP2K/cp2k.popt	2.2
		/vol-th/software/CP2K/cp2k.popt-2.5	2.5
	CPMD	/vol-th/software/cpmd.x	3.13.2
计算化学	Nwchem	/vol-th/software/Nwchem	6.0
药物设计	AutoDOCK	/vol-th/software/AutoDock	4.0
	DOCK	/vol-th/software/Dock	6.0
生物信息	Blast	/vol-th/software/blast-2.2.23	2.2.23
	Samtools	/vol-th/software/samtools-1.2	1.2
	Bowtie	/vol-th/software/bowtie-1.0.0	1.0
	Bowtie2	/vol-th/software/bowtie2-2.2.0	2.2.2
计算流体力学	OpenFOAM	/vol-th/software/OpenFOAM	源码包
		/vol-th/software/OpenFOAM_ThirdParty	第三方包

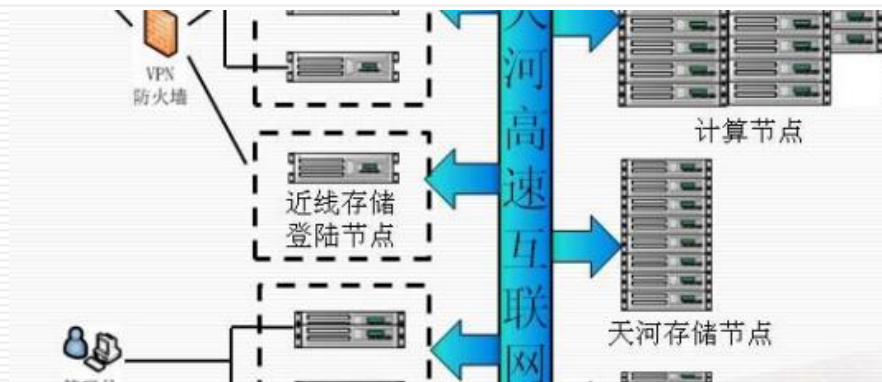
## 1) HPC系统和天河三号原型样机系统

随着天河一号系统应用范围的不断扩展，以及用户计算需求的增加，中心于2017年3月开始搭建了新的HPC集群。目前已经有两套上线使用的HPC系统，预计今后还会继续推出。

同时，天河三号原型样机系统18年已在我中心部署完成，采用自主飞腾CPU、自主天河高速互联、自主麒麟操作系统。目前大多数领域应用移植并测试成功，可以提供测试机会。

## 2) 近线存储

海量近线存储容量为1.3PB，与天河系统其他存储通过高速网互联，主要用于存放用户历史数据，其平台架构如下图1所示。近线存储已经对外开放，若用户需要使用该存储，请与应用部联系。



### 3) 中心存储资源管理制度优化

存储资源管理进一步优化，天河系统上的用户账号逐渐增加，系统的存储资源逐渐趋于饱和。当系统总存储空间利用率达80%以上时，系统的整体稳定性就会受到一定影响，进而会影响用户作业运行的稳定性。为了更为高效、科学地利用存储资源，中心应用部制定了存储资源管理方案，针对磁盘空间的不同使用率，启动分等级预警，并制定了相应的预警方案，保障系统存储的稳定性。

### 4) 中心删除备份机制

为了更好的保护用户的数据，防止用户数据误删等操作的发生，中心开发了yhrm命令，该命令除了与系统自带的rm命令使用方法一样，使用该命令删除数据后，会自动把删除的数据迁移到类似windows系统的回收站里面进行15天的保存，如果用户发现删除的数据还有用，那就可以通过回收站把这些数据恢复。比如vol6下的用户账号test\_user在执行yhrm删除数据后，在/vol6/.Recycle/test\_user，删除的数据存放在以删除时间命名的文件夹中，找到指定文件后即可cp到自己想存放的位置。

### 5) 核心数据备份

为了防止由于不可抗力因素造成用户数据的损坏或丢失，进一步加强用户数据的安全性，中心部署了一套可用存储，为每一位合同用户提供核心数据备份服务，同时提供核心数据恢复服务，以保障用户核心数据的安全。



# 感谢您的支持与合作

国家超级计算天津中心 郑刚

zhenggang@nsc-tj.cn





国家超级计算天津中心官方微信