

国家超级计算广州中心/广州超级计算中心

天河二号简明使用手册

V 1.1

钟英

2019 年 1 月 12 日

使用须知

1. “天河二号”超级计算机（以下简称“天河二号”）的**结点使用方式为独占式**，即计算结点分配给用户以后，不能为其他用户所用；**机时的计算也将以结点为基本单元**。比如，使用 2 结点共 12 核（每结点使用 6 核）运行 1 小时的程序，机时为 $2*24*1=48$ （核*小时），而非 $2*6*1=12$ （核*小时）。因此用户需要合理利用自己的资源。

2. **用户所分配机时仅为机时使用**。如委托研发、修改程序代码、编写复杂脚本等额外服务，会产生一定的服务费用。如有需求，请发邮件到 techsupport@nscg-gz.cn 洽商。

3. **有少量机时但不能满足作业运行完成时，按照中心目前规定：账号有剩余机时且还有少量短时间作业需要运行，作业可正常提交；若剩余机时已不能满足少量短时间作业运行完成，对于已经提交的作业仍会保障运行完成**。对于不希望继续运行需结束作业，用户需自行停止需要结束的作业。对于已无机时的账号，使用 yhi 和 yhq 无法查看分区和作业状态，此时若还需查看作业状态信息，请使用“`yhq -a`”命令查看。

4. **超算中心不提供商业软件的安装适配、售后及应用支持服务**。您可以自行安装和使用商业软件，或与该软件的售后服务团队联系寻求协助。用户自行安装软件带来的版权问题请自负责任。

5. 超算中心根据实际情况对基于“天河二号”部署的开源软件的安装进行一定程度上的协助，但软件的算法合理性、精度、并行效率、使用方法等软件自身问题需要自行解决。常用的开源软件的使用方法可以查阅超算中心官

网上的相关说明，如：<http://www.nscg-gz.cn/newsdetail.html?7311>

（Quantum ESPRESSO）。

6. 如您的任何行为对超算中心的财产和声誉等方面造成了任何损失，超算中心将依法追究相关责任。以上条例解释权归超算中心所有。

目 录

1	登录.....	1
1.1	VPN 验证.....	1
1.2	终端登录.....	1
1.2.1	PUTTY 登录.....	1
1.2.2	Xshell (Xmanager-XShell) 登录.....	3
1.2.3	Linux 或苹果系统登录.....	4
1.3	备注.....	4
2	文件传输.....	6
2.1	文件系统.....	6
2.2	数据传输.....	6
2.2.1	FileZilla 登录.....	7
2.2.2	WinSCP 登录.....	8
3	环境变量管理工具 module.....	10
3.1	简介.....	10
3.2	基本命令.....	10
4	编译器.....	11
4.1	Intel 编译器.....	11
4.2	GCC 编译器.....	11
4.3	MPI 编译环境.....	11
5	作业提交.....	13
5.1	结点状态查看 yhinfo 或 yhi.....	13
5.2	作业状态信息查看 yhqueue 或 yhq.....	13
5.3	交互式作业提交 yhrun.....	14
5.3.1	简介.....	14
5.3.2	yhrun 常用选项.....	14
5.3.3	使用示例.....	15
5.4	批处理作业 yhbatch.....	16
5.4.1	简介.....	16
5.4.2	使用示例.....	16
5.5	结点资源抢占命令 yhalloc.....	17
5.5.1	简介.....	17
5.5.2	使用示例.....	17
5.6	任务取消 yhcancel.....	17
5.7	备注.....	18
6	常见上机问题 (FAQ)	19

1 登录

1.1 VPN 验证

VPN 登录操作步骤请见《VPN 客户端使用手册》。

1.2 终端登录

以上成功与天河二号建立了 VPN 安全链接后，为了进一步保证用户的数据安全，中心不允许 telnet 等方式登录服务器，必须通过 ssh 登录方式来使用中心资源。用户可以使用 ssh 客户端软件（如 Putty、SecureCRT、Xmanager）来登录系统。登录步骤如下：

Step 1: 从管理员处获取认证文件，切忌传播。

Step 2: 使用终端工具连接，通过使用系统管理员提供的 Private Key 文件（随账号通知邮件附件给出）进行认证，具体操作可参考下例。

注意：

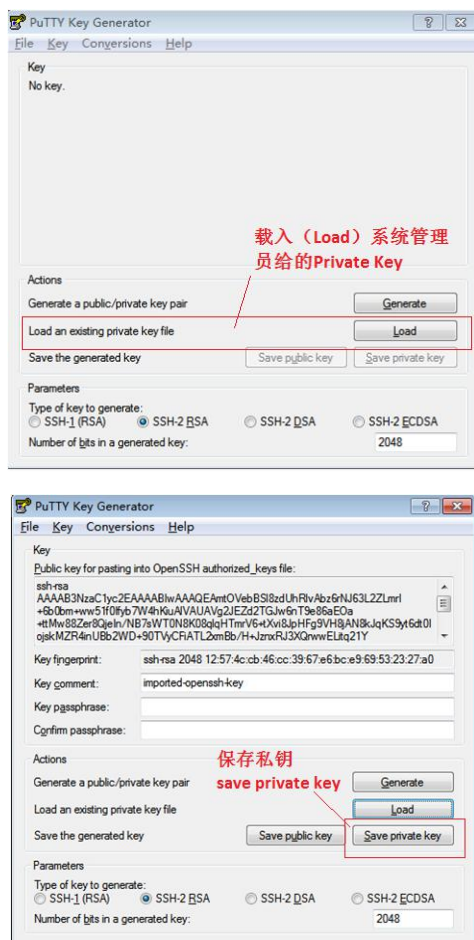
下文提到的系统 IP 为 172. 16. 22. 11，端口号为 5566。

1.2.1 PUTTY 登录

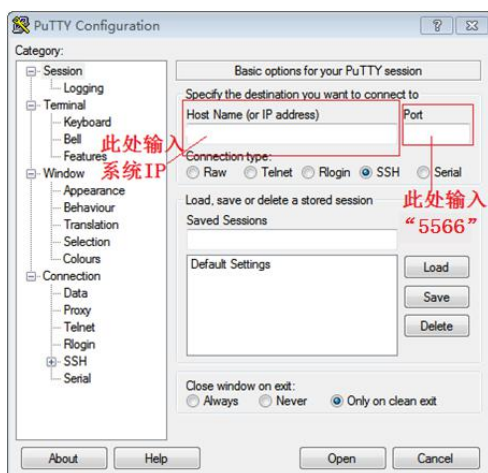
用 PUTTY 登录首先需要转换 Private Key 文件。打开 PUTTY 的安装路径，运行 PUTTYGEN.EXE 程序进行 Private Key 文件转换。



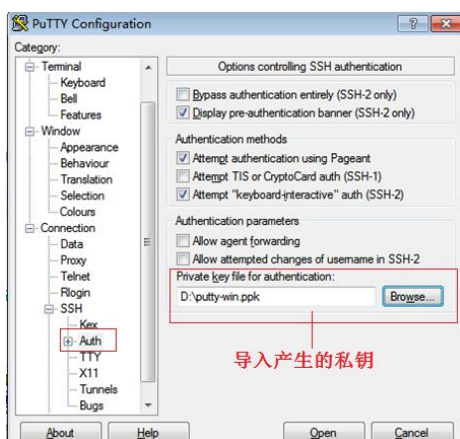
点击“Load”载入 Private Key 文件，点击“Save private key”保存转换后的 Key 文件。



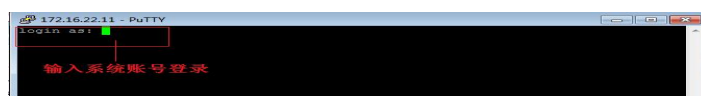
运行 PUTTY.EXE，选择 “Session”。在 “Host Name” 处填写系统 IP，“Port” 处填写 “5566”。



选择 “Connection” -> “SSH” -> “Auth”，然后点击 “Private key file for authentication” 处的 “Browse” 选择转换后的 Private Key 文件，然后点击 “Open” 打开登录界面。

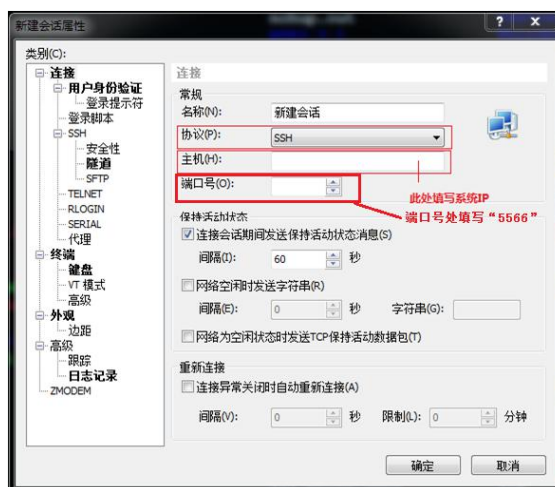


在登录界面里输入系统账号，回车即可登录。

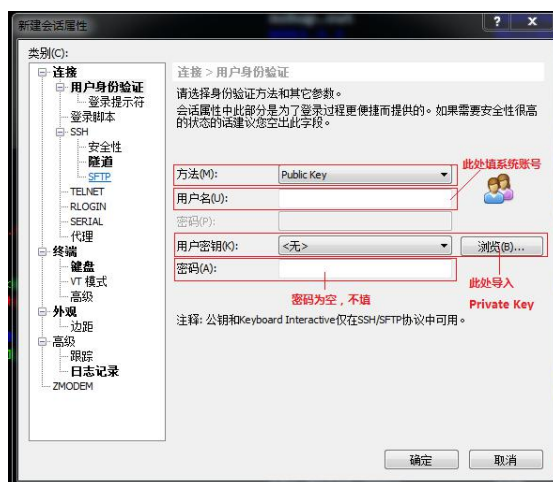


1.2.2 Xshell (Xmanager-XShell) 登录

打开 XShell，点击工具栏的“新建”选项。在“连接”的“常规”里，“协议”选择 SSH，然后在“主机”处填写系统 IP。



在“用户身份验证”这里，“方法”选择 Public Key，然后点击“浏览”选择得到的 Private Key 文件，然后点击“确定”即可登录。



1.2.3 Linux 或苹果系统登录

如果是 Linux 或者苹果系统，首先需要给 Private Key 文件设置权限，命令如下：

```
chmod 400 PrivateKey
```

然后对系统文件做如下修改：

1) 用 root 权限修改 ssh_config 文件（linux 系统路径为/etc/ssh/ssh_config，苹果系统路径为/etc/ssh_config）：

Linux 系统：sudo vi /etc/ssh/ssh_config

苹果系统：sudo vi /etc/ssh_config

2) 在 ssh_config 文件中增加如下两行并保存：

```
StrictHostKeyChecking no
```

```
UserKnownHostsFile /dev/null
```

重新打开一个终端界面进行登录，使用“ssh 命令的 -i 选项”来指定 Private Key 文件，命令如下所示：

```
ssh -p 5566 -i PrivateKey 系统账号@系统 IP
```

1.3 备注

1) 建议用户在第一次登录时重新生成 Private Key，具体操作见下文 FAQ 中的“重新生成 Private Key”，避免 Private Key 泄露导致数据泄露。

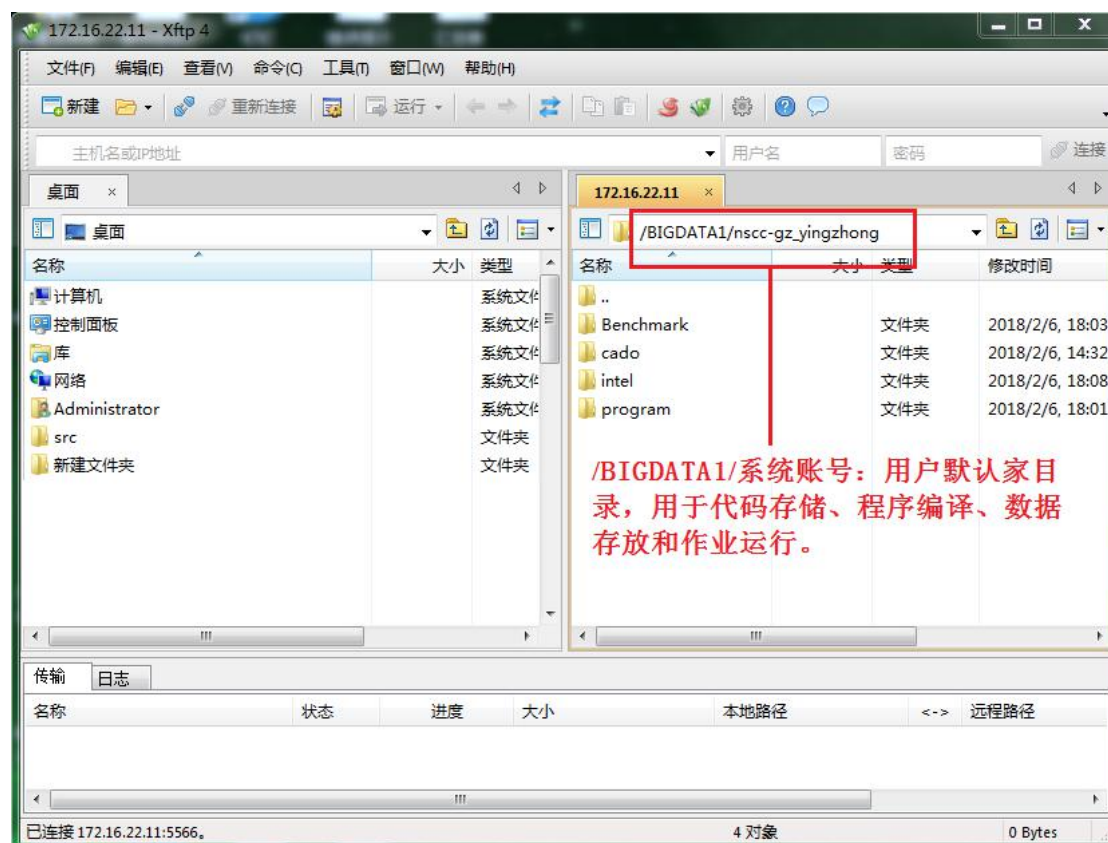
2) 不要修改家目录、密钥所在文件夹和密钥文件等的属主、属组及读写权限，否则可能会出现因密钥不符合安全规范而不能登陆的问题。

3) 用户可在登录服务器进行作业提交和程序编译等操作，但请注意系统资源的使用情况（可使用 **top** 命令查看），不要过度使用。请不要进行类似于使用超过 8 核进行编译、同时编译 20 多个程序等操作，这类操作会耗费大量系统资源，影响用户的体验。若经发现，可能会杀掉用户相关进程；严重者将禁止账号登录。

2 文件传输

2.1 文件系统

天河二号高性能计算平台文件系统为/BIGDATA1。



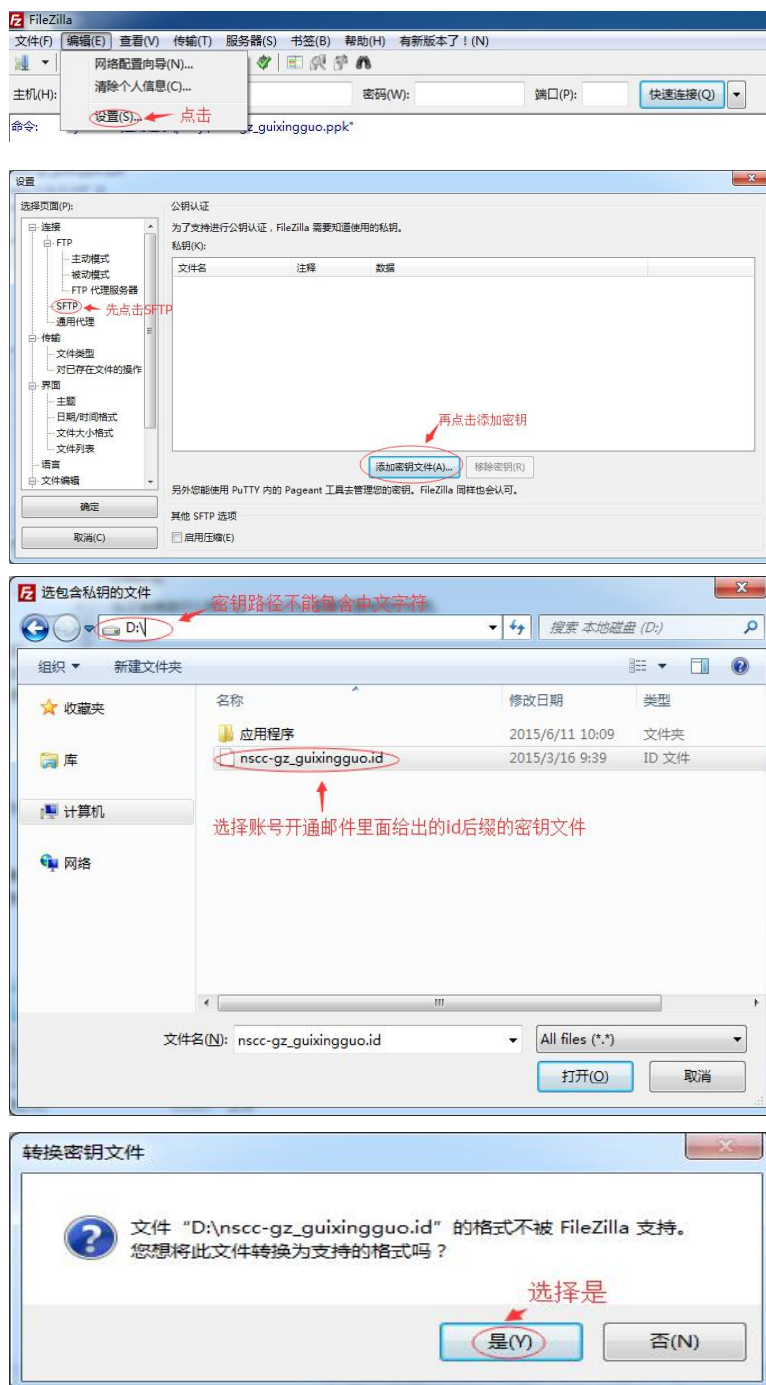
该分区下的用户主目录为用户默认家目录，用于代码存储、程序编译、数据存放和作业运行。在脚本中使用“/BIGDATA1/系统账号”引用该分区下的用户主目录。对于试用用户，其磁盘限额每租户 1TB（租户相当于账号组，一个租户可以包含一个或者多个账号）；对于签约用户，其磁盘限额将根据用户的具体需求进行配置。所有登录服务结点和计算结点都可以访问该分区下的文件。

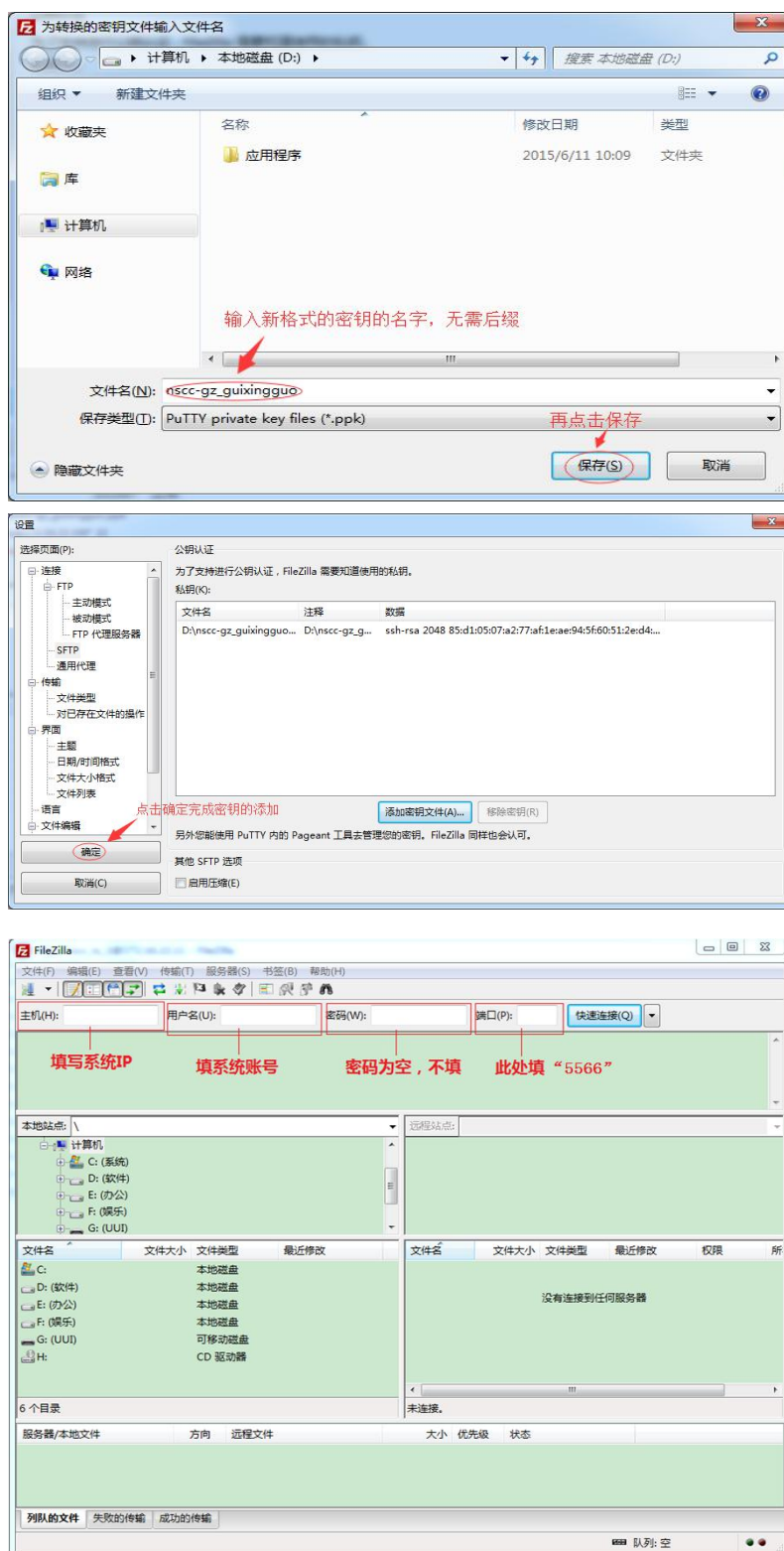
2.2 数据传输

从外部机器向天河二号上传或下载文件，可以使用 sftp 客户端，例如使用 FileZilla、WinScp、FlashFTP 等的 sftp 数据传输软件，或者使用 Xmanager 等本身自带的文件传输功能。

2.2.1 FileZilla 登录

FileZilla 的登录步骤见下图：

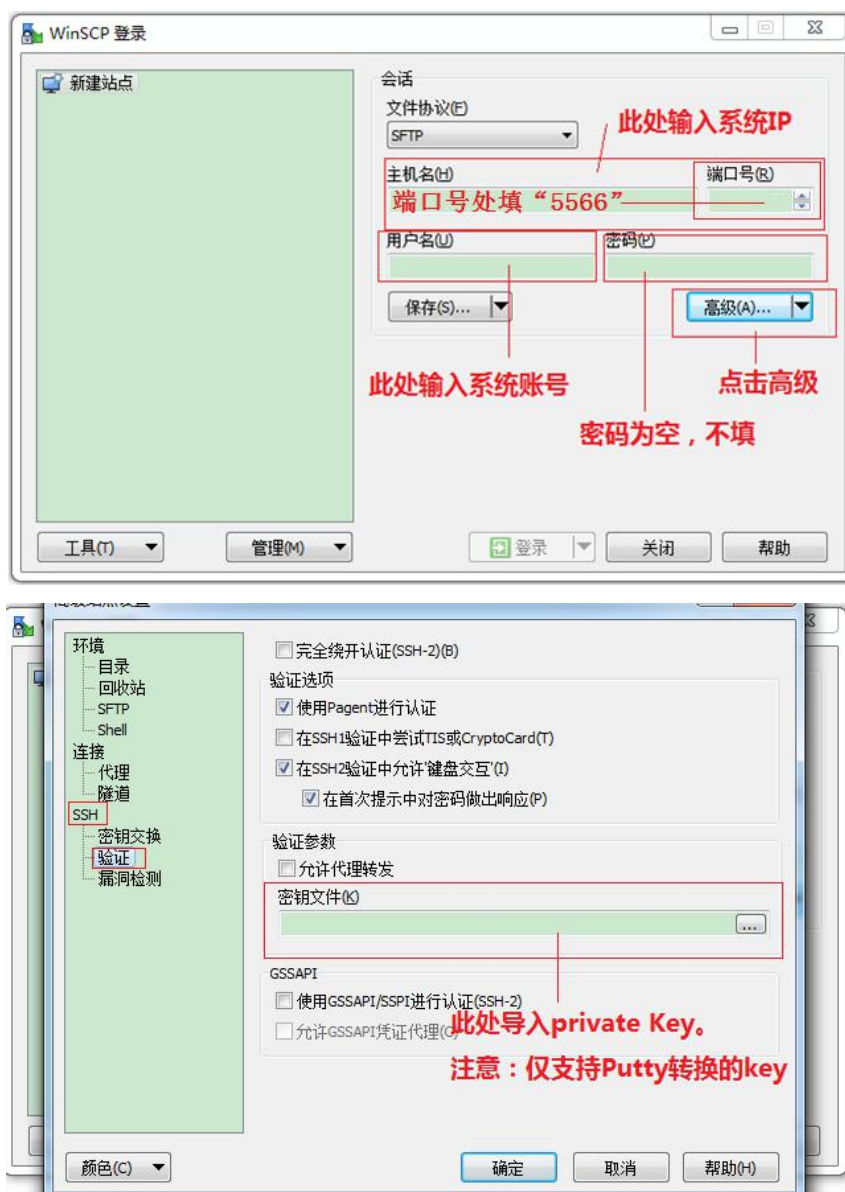




注意：密钥所在路径不能含有中文。

2.2.2 WinSCP 登录

WinSCP 的登录步骤见下图：



注意：

1. SFTP 服务是全天 24 小时可以连接的，需提前使用 VPN 验证方可登录。天河二号中的存储空间只作为数据的临时存储，鉴于存储空间容量限制和数据安全的考虑，请用户及时把重要数据或敏感数据保存到自己的计算机中，并及时清理自己的存储空间。
2. 在有大量文件需要下载时，建议使用 tar 命令进行打包，以加快下载速度，减少出错几率。该命令为：tar -cvf file.tar file，其中 file 为需要打包压缩的文件或目录，file.tar 为打包后的文件。

3 环境变量管理工具 module

3.1 简介

由于不同用户在天河二号上可能需要使用不同的软件环境，配置不同的环境变量，默认配置无法满足所有用户的需要，因而在天河二号上安装了 **module** 软件来进行对环境变量的管理，方便用户更好地使用机器。

module 通过配置 **modulefile** 支持环境变量的动态修改，能够控制软件不同版本对环境变量的依赖关系。用户通过简单的命令即可获得适于自己环境变量设置，因而提高了用户移植软件的效率。

3.2 基本命令

已经在登录服务结点上配置好 **module** 工具，主要用法如下：

module avail：查看可用的模块的列表。

```
[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/impi/4.1.2.040    intel-compilers/13.0.0  netcdf/4.3.0-C
FFTW/3.3.4-intel    NAMD/2.9                   intel-compilers/14.0.1  netcdf/4.3.0-C-nohdf5
FFTW/3.3.4-openmp   OpenFOAM/2.2.2             intel_tools/14.0.1     netcdf/4.3.2-CF
FFTW/3.3.4-openmp-O3 PETSc/3.4.4                lammps/14-Feb14        null
MPI/Gnu/MPICH/3.1   cmake/2.6-patch4           module-cvs             use.own
MPI/Intel/MPICH/3.1 cmake/2.8                  module-info
MPI/Intel/MPICH/3.1-dbg dot                          modules
MPI/Intel/MPICH/3.1-icc11 emacs                      mpich2-x86_64
[testuser1@ln1%tinhe2-B ~]$
```

module load [modulesfile]：能够加载需要使用的 **modulefiles**。

```
[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 ~]$
```

使用 **module** 加载软件（OpenFOAM/2.2.2）的配置环境。

```
[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 ~]$
```

module 其它用法，可在 **help** 中查询。

```
[testuser1@ln1%tinhe2-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
```

4 编译器

目前，天河二号系统已配置 GNU 和 Intel 编译器，支持 C，C++，Fortran77 和 Fortran90 语言程序的开发。同时，天河二号系统支持 OpenMP 和 MPI 两种并行编程模式。其中 OpenMP 为共享内存方式，仅能在一个计算结点内并行，最大线程数不能超过结点处理器核心数；MPI 是分布式内存并行，计算作业可以在一个或者若干个结点上进行，最大进程数仅受用户帐号所能调用的 CPU 总数限制。

共享内存的 OpenMP 并行方式通常由编译器来支持，目前 GNU 和 Intel 的编译器均已实现了对该标准的支持。

4.1 Intel 编译器

天河二号系统上已配置多个版本的 Intel 编译器。其中，系统已设置了 intel 14 为用户默认编译器。若无特殊要求，用户登录后无需设置编译器环境。

用户若想使用其他版本的 Intel 编译器可使用 module 进行环境加载。具体命令如下：

使用 Intel 13 编译器：module load intelcompiler/13.0.1

注意：查找编译命令所在的路径可以使用 which 命令，例如“which icc”将返回当前使用的 icc 命令所在的具体路径。确认编译器的版本请在编译命令后使用 -v 或者 -V 参数，例如“icc -v”、“ifort -V”，Intel 编译器的详细命令行调用则可以用“icc --help”获得。

4.2 GCC 编译器

天河二号上默认安装的 GNU 编译器版本是 4.8.5，相关的编译命令都安装到 /usr/bin 目录中。

4.3 MPI 编译环境

由于天河二号采用了自主互连的高速网络，因此底层 MPI 为自主实现，基于 Intel 编译器和 GNU 编译器进行编译，所有 mpi 版本均安装目录在

/BIGDATA1/app/MPI/mpich 路径下，为了追求最高效率，该目录下的 **mpi** 为自主实现的 **mpi** 版本。基本使用时（运行程序没特殊要求时）推荐使用 /BIGDATA1/app/MPI/mpich/3.2.1-icc-14.0.2-dynamic 版本，有较高的效率。

并行 **mpi** 编译环境使用注意事项：

1. 系统默认使用/BIGDATA1/app/MPI/mpich/3.2.1-icc-14.0.2-dynamic 目录下的 **mpi**。该 **mpi** 调用 Intel 14 编译器，且该 **mpi** 的库均为动态库。

2. 天河二号具备自主高速互联网络，并提供 **MPI** 编程环境，如用户必须使用其他版本 **mpi**，比如 **openmpi** 等，也可以自己安装并部署。用自行 **mpi** 编译的程序，同样可以利用高速互联网络的虚拟以太网运算任务，但性能会较天河二号自主 **MPI** 低很多。

MPI 编译命令内部会自动包含 **MPI** 标准头文件所在的路径，并自动连接所需的 **MPI** 通信接口库，所以不需要用户在命令行参数中指定。

如果用户使用 **makefile** 或 **autoconf** 编译 **MPI** 并行程序，还可以将 **makefile** 中的 **CC**, **CXX**, **F77**, **F90** 等变量设置成 **mpicc**, **mpicxx**, **mpif77**, **mpif90**，或这在 **autoconf** 的 **configure** 过程前设置 **CC**, **CXX**, **F77** 和 **F90** 等环境变量为 **mpicc**, **mpicxx**, **mpif77** 和 **mpif90** 等。

5 作业提交

5.1 结点状态查看 yhinfo 或 yhi

yhi 为 yhinfo 命令的简写，用户用其查看结点状态。

```
[nscs_ts_2@lon26%tianhe2-H ~]$ yhi
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
docker_128 up infinite 75 idle cn[6912-6927,6929-6930,6932-69
-7039]
localdisk up infinite 33 alloc cn[7040-7041,7046-7055,7059,70
localdisk up infinite 30 idle cn[7056,7063-7064,7066-7067,70
,7154-7157,7166-7167]
bigdata* up infinite 15 comp cn[7417,7419,7469-7470,7832,83
bigdata* up infinite 2 drng cn[7322,8820]
bigdata* up infinite 1092 alloc cn[7297-7298,7301-7305,7309,73
,7355-7360,7362-7364,7369-7370,7372,7377-7381,7383-7385,7388-7390,7393,7
1,7453-7454,7456,7458-7459,7465,7467-7468,7471-7479,7488,7490,7496-7503,
92-7596,7601-7608,7612,7614-7615,7618-7620,7622,7625-7626,7629-7630,7635
690,7694,7696-7698,7700-7703,7705,7707-7709,7711-7715,7717-7731,7733-773
7810-7811,7813-7816,7818-7820,7823-7824,7826-7831,7833-7837,7839,7843-78
-7902,7904-7917,7919-7931,7933-7936,7941-7943,7945-7947,7951,7955,7958-7
4,8006-8007,8009,8012,8014-8016,8018-8023,8025,8027-8032,8044,8050,8057,
13-8116,8118-8121,8124,8126,8129-8134,8136-8139,8141,8143-8144,8148-8149
211,8213-8217,8220-8224,8228-8229,8233-8237,8239-8240,8243-8247,8256,825
8297,8299-8306,8308-8310,8315-8316,8324-8326,8334-8336,8340-8341,8345,83
-8389,8393,8396-8400,8403-8404,8406-8407,8413-8415,8418,8420-8431,8433-8
8,8500-8506,8509-8510,8512-8513,8515-8517,8519,8521-8524,8531-8532,8534,
80-8582,8584-8585,8591,8599,8605,8608-8609,8613-8616,8619-8620,8623-8625
673,8675-8678,8680,8683-8686,8690,8692-8693,8695-8706,8708-8711,8714-871
8779,8781-8782,8786-8789,8792-8793,8795-8798,8802-8805,8807-8814,8817-88
,8865-8866,8882,8885-8887,8902-8905,8907-8909,8913-8916,8918-8920,8926-8
4,9016-9017,9023-9029,9032-9033,9036,9039-9043,9057-9058,9060-9061,9065-
62-9163,9166,9169-9173,9175-9182,9187-9188,9193-9194,9198-9210,9213-9215
bigdata* up infinite 52 idle cn[7986,8067,8145,8241,8277,83
,8899,8939-8942,8977-8980,9019-9020,9067,9069,9074,9083,9101-9104,9119,9
free up infinite 4 comp cn[9059,9106-9107,9120]
free up infinite 1 drng cn8820
```

其中 PARTITION 表示分区，NODES 表示结点数，NODELIST 为结点列表，STATE 表示结点运行状态。其中，idle 表示结点处于空闲状态，allocated 表示结点已经分配了一个或多个作业。

5.2 作业状态信息查看 yhqueue 或 yhq

yhq 为 yhqueue 命令的简写，用户用其查看作业运行情况。

```
[nscs_ts_2@lon26 ~]$ yhq
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
1222563 bigdata bash nscs_ts_2 R 0:33 1 cn7955
[nscs_ts_2@lon26 ~]$
```

其中 JOBID 表示任务 ID，Name 表示任务名称，USER 为用户，TIME 为已运行时间，NODES 表示占用结点数，NODELIST 为任务运行的结点列表。

作业状态通常显示 R、PD、S、CG 或者 JobHeldAdmin。R 表示作业正在执行；PD 表示作业处于排队状态；S 表示作业被挂起；CG 表示作业已结束，等待资源释放；JobHeldAdmin 表示作业因为资源调度由管理员暂时锁定，待资源满足时会被释放并进入作业调度。

注意：推荐使用“yhq -a”查看作业状态信息。对于已无机时的账号，使用 yhi 和 yhq 无法查看分区和作业状态，此时若还需查看作业状态信息，使用“yhq -a”命令可查看。

5.3 交互式作业提交 yhrun

5.3.1 简介

交互式提交作业：在 shell 窗口中执行 yhrun 命令，主要命令格式如下：

```
yhrun [options] program
```

5.3.2 yhrun 常用选项

yhrun 包括多个选项，其中最常用的选项主要有以下几个：

- **-n, --ntasks=number**

指定要运行的任务数。请求为 number 个任务分配资源，默认为每个任务一个处理器核。

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

告知资源管理系统控制进程，作业步的每个任务需要 ncpus 个处理器核。若未指定此选项，则控制进程默认为每个任务分配一个处理器核。

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

请求为作业至少分配 minnodes 个结点。调度器可能觉得在多于 minnodes 个结点上运行作业。可以通过 maxnodes 限制最多分配的结点数目（例如“-N 2-4”或“--nodes=2-4”）。最少和最多结点数目可以相同以指定特定的结点数目（例如，“-N 2”或“--nodes=2-2”将请求两个且仅两个结点）。分区的结点数目限制将覆盖作业的请求。如果作业的结点限制超出了分区中配置的结点数目，作业将被拒绝。如果没有指定-N，缺省行为是分配足够多的结点以满足-n 和-c 参数的需求。在允许的限制范围内以及不延迟作业开始运行的前提下，作业将被分配尽可能多的结点。

- **-p, --partition=partition name**

在指定分区中分配资源。如未指定，则由控制进程在系统默认分区中分配资源。

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

请求指定的结点名字列表。作业分配资源中将至少包含这些结点。列表可以用逗号分隔的结点名或结点范围（如 `cn[1-5,7,...]`）指定，或者用文件名指定。如果参数中包含“/”字符，则会被当作文件名。如果指定了最大结点数如 `-N 1-2`，但是文件中有多余 2 个结点，则请求列表中只使用前 2 个结点。

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

不要将指定的结点分配给作业。如果包含“/”字符，参数将被当作文件名。

`yhrun` 将把作业请求提交到控制进程，然后在远程结点上启动所有进程。如果资源请求不能立即被满足，`yhrun` 将阻塞等待，直到资源可用以运行作业。如果指定了 `--immediate` 选项，则 `yhrun` 将在资源不是立即可用时终止。

- **-h, --help**

若需使用 `yhrun` 更多选项，可通过“`yhrun -h`”或“`yhrun --help`”查看。

5.3.3 使用示例

1) 在分区 `bigdata` 上指定任务数运行 `hostname`:

```
[nscs_ts_2@lon26%tianhe2-H ~]$ yhrun -n 4 -p bigdata hostname
cn8121
cn8121
cn8121
cn8121
[nscs_ts_2@lon26%tianhe2-H ~]$
```

2) 在分区 `bigdata`，结点 `cn[8977-8980]` 上运行 `hostname`:

```
[nscs_ts_2@lon26%tianhe2-H ~]$ yhrun -n 4 -w cn[8977-8980] -p bigdata hostname
cn8980
cn8978
cn8979
cn8977
[nscs_ts_2@lon26%tianhe2-H ~]$
```

3) 在 `bigdata` 分区，运行 4 任务的 `hostname`，每个结点一个任务，分配的结点中至少包含结点 `cn[8977-8988]`:

```
[nscs_ts_2@lon26%tianhe2-H ~]$ yhrun -n 4 -N 4 -w cn[8977-8988] -p bigdata hostname
yhrun: Warning: can't run 4 processes on 12 nodes, setting nnodes to 4
cn8977
cn8978
cn8980
cn8979
[nscs_ts_2@lon26%tianhe2-H ~]$
```

4) 在 **bigdata** 分区, 运行 4 任务的 **hostname**, 每个结点一个任务, 分配的结点中不包含结点 **cn[8977-8988]**:

```
[nscs_ts_2@lon26%tianhe2-H ~]$ yhrun -n 4 -N 4 -x cn[8977-8988] -p bigdata hostname
cn9104
cn9102
cn9101
cn9103
[nscs_ts_2@lon26%tianhe2-H ~]$
```

5.4 批处理作业 yhbatch

5.4.1 简介

批处理作业是指用户编写作业脚本, 指定资源需求约束, 提交后台执行作业。提交批处理作业的命令为 **yhbatch**, 用户提交命令即返回命令行窗口, 但此时作业在进入调度状态, 在资源满足要求时, 分配完计算结点之后, 系统将在所分配的第一个计算结点 (而不是登录结点) 上加载执行用户的作业脚本。

批处理作业的脚本为一个文本文件, 脚本第一行以“**#!/**”字符开头, 并制定脚本文件的解释程序, 如 **sh**, **bash**。由于计算节点为精简环境, 只提供 **sh** 和 **bash** 的默认支持。

5.4.2 使用示例

例如用户的脚本名为 **mybash.sh**, 内容如下:

```
[nscs_ts_2@lon26%tianhe2-H TEST1]$ vi mybash.sh
#!/bin/bash
yhrun -N 4 -n 4 -p bigdata hostname
```

根据该脚本用户提交批处理作业, 需要明确申请的资源为 **bigdata** 分区的 4 个结点。

```
[nscs_ts_2@lon26%tianhe2-H TEST1]$ ll
total 4
-rw-r--r-- 1 nscs_ts_2 nscs_ts 48 Jan 16 15:40 mybash.sh
[nscs_ts_2@lon26%tianhe2-H TEST1]$
```

注意: 需给该文本文件设置 **mybash.sh** 可执行权限, 利用命令: **chmod +x mybash.sh**

```
[nscs_ts_2@lon26%tianhe2-H TEST1]$ chmod +x mybash.sh
[nscs_ts_2@lon26%tianhe2-H TEST1]$ ll
total 4
-rwxr-xr-x 1 nscs_ts_2 nscs_ts 48 Jan 16 15:40 mybash.sh
[nscs_ts_2@lon26%tianhe2-H TEST1]$
```


用户 yhbatch 批处理命令如下：

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

计算开始后，工作目录中会生成以 slurm 开头的.out 文件为输出文件。

更多选项，用户可以通过 yhbatch --help 命令查看。

5.5 结点资源抢占命令 yhalloc

5.5.1 简介

该命令支持用户在提交作业前，抢占所需计算资源（此时开始计算所用机时）。

5.5.2 使用示例

yhalloc 提交方式如下：

首先申请资源，执行如下命令：

```
[nscs_ts_2@lon26%tianhe2-H TEST1]$ yhalloc -N 1 -p bigdata
yhalloc: Granted job allocation 1222747
[nscs_ts_2@lon26 TEST1]$
```

通过 yhq 查看相应的 jobID 为 1222747，结点为 cn8145。

```
[nscs_ts_2@lon26 TEST1]$ yhq
      JOBID PARTITION   NAME     USER ST      TIME  NODES NODELIST(REASON)
      1222747    bigdata   bash     nscs_ts_2  R      0:09      1 cn8145
[nscs_ts_2@lon26 TEST1]$
```

用户可以选择如下方式：

```
[nscs_ts_2@lon26 TEST1]$ ssh cn8145
Warning: Permanently added 'cn8145' (RSA) to the list of known hosts.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (en_US.UTF-8): No such file or directory
-bash: warning: setlocale: LC_COLLATE: cannot change locale (en_US.UTF-8): No such file or directory
-bash: warning: setlocale: LC_MESSAGES: cannot change locale (en_US.UTF-8): No such file or directory
-bash: warning: setlocale: LC_NUMERIC: cannot change locale (en_US.UTF-8): No such file or directory
-bash: warning: setlocale: LC_TIME: cannot change locale (en_US.UTF-8): No such file or directory
[nscs_ts_2@cn8145%tianhe2-H ~]$
```

切换到 cn8145 结点，之后执行程序。

5.6 任务取消 yhcanceled

用户使用 yhcanceled 命令取消自己的作业。命令格式如下：

```
yhcanceled jobid
```

jobid 可通过 yhq 获得。对于排队作业，取消作业将简单地把作业标记为 CANCELLED 状态而结束作业。对于运行中或挂起的作业，取消作业将终止作业的所有作业步，包括批处理作业脚本，将作业标记为 CANCELLED 状态，并

回收分配给作业的结点。一般地，批处理作业将会马上终止；交互作业的 `yhrun` 进程将会感知到任务的退出而终止；抢占结点资源的 `yhallocc` 进程不会自动退出，除非作业所执行的用户命令因作业或任务的结束而终止。但是在作业被取消时，控制进程都会发送通知消息给分配资源的 `yhrun` 或 `yhallocc` 进程。用户可以选择通过 `yhallocc` 的 `--kill-command` 选项设置在收到通知时向所执行的命令发送信号将其终止。

5.7 备注

由于手册篇幅限制，只列出了对于绝大多数用户比较重要的相关内容，如您有其他需求也可以联系超算中心技术人员。

重要提示：

1) 请不要在登录结点直接运行可执行程序（极大的影响其他用户的登录和使用效率）。

2) 如无特殊需要，请使用批处理方式（`yhbatch`）提交任务，如果有任何问题请联系超算中心技术人员。

3) 请保存好运行程序的 `log` 文件，从而方便超算中心技术人员在作业出问题后，协助解决问题。

4) 若需登录计算结点运行程序，需要先分配计算结点，方可登录。

5) 提交时需在提交命令中加入参数选项“-p 分区名”，即提交命令应为“`yhurn -p 分区名 ...`”或者“`yhbatch -p 分区名`”。同时，推荐用户使用 `yhbatch` 方式提交作业。分区名请通过 `yhi` 命令查看获得，其中 **PARTITION 一栏对应的就是分区，如下图所示。**

```
[nscc_ts_2@lon26 TEST1]$ yhi
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
docker_128 up infinite 75 idle cn[6912-6927,6929-6930,6932-6933,6936-6949,6952-6962,-7039] 分区名
localdisk up infinite 50 alloc cn[7040-7041,7046-7055,7063-7064,7066-7067,7073-7080,-7155,7160-7167]
localdisk up infinite 13 idle cn[7056,7059,7069,7071,7082,7091,7126,7138,7140,7147,
bigdata* up infinite 15 comp cn[7417,7419,7469-7470,7832,8318-8321,8552,8555,9059,
bigdata* up infinite 2 drng cn[7322,8820]
bigdata* up infinite 1104 alloc cn[7297-7298,7301-7305,7309,7311-7312,7314,7317,7320-
,7355-7360,7362-7364,7369-7370,7372,7377-7381,7383-7385,7388-7390,7393,7397-7400,7402-7408,7410
1,7453-7454,7456,7458-7459,7465,7467-7468,7471-7479,7488,7490,7496-7503,7505-7506,7510,7552-755
92-7596,7601-7608,7612,7614-7615,7618-7620,7622,7625-7626,7629-7630,7635-7637,7640,7642-7644,76
690,7694,7696-7698,7700-7703,7705,7707-7709,7711-7715,7717-7731,7733-7735,7743,7746-7750,7753-7
755,7757-7758,7760-7761,7763-7764,7766-7767,7769-7770,7772-7773,7775-7776,7778-7779,7781-7782,7784-7785,7787-7788,7790-7791,7793-7794,7796-7797,7799-7800,7802-7803,7805-7806,7808-7809,7811-7812,7814-7815,7817-7818,7820-7821,7823-7824,7826-7827,7829-7830,7832-7833,7835-7836,7838-7839,7841-7842,7844-7845,7847-7848,7850-7851,7853-7854,7856-7857,7859-7860,7862-7863,7865-7866,7868-7869,7871-7872,7874-7875,7877-7878,7880-7881,7883-7884,7886-7887,7889-7890,7892-7893,7895-7896,7898-7899,7901-7902,7904-7905,7907-7908,7910-7911,7913-7914,7916-7917,7919-7920,7922-7923,7925-7926,7928-7929,7931-7932,7934-7935,7937-7938,7940-7941,7943-7944,7946-7947,7949-7950,7952-7953,7955-7956,7958-7959,7961-7962,7964-7965,7967-7968,7970-7971,7973-7974,7976-7977,7979-7980,7982-7983,7985-7986,7988-7989,7991-7992,7994-7995,7997-7998,7999-8000]
```

6 常见上机问题（FAQ）

✓ VPN 账号连接成功，但是终端工具连接不了天河二号。

若出现该现象，首先请查看您的电脑是否安装 360 卫士、安全卫士等软件，若安装了请先将软件关闭，再重新连接 VPN；若上一步完成后仍无法连接，请 ping 系统 IP，查看丢包率，若丢包率很高则是您的网速导致，若丢包率低，则请联系中心相关人员排查。

✓ 机时以及磁盘限额查询。

在浏览器中输入天河星光网址（<https://starlight.nscg-gz.cn/>），然后注册并登录天河星光账号，按提示绑定 HPC 账号，机时及磁盘存储使用情况会在天河星光账户首页显示。



✓ 如果遇到一些作业运行时报库无法找到，如何处理？

用户可通过 locate 命令查找相应的库，并将在/BIGDATA1 目录下的对应的库路径加入环境变量 LD_LIBRARY_PATH 中。如果是系统的库文件，可以尝试命令“source /BIGDATA1/app/osenv/set.sh”然后再提交作业。如果还是不行，可将缺少的库拷贝到自己的文件夹如~/lib 中，并设置环境变量：“export LD_LIBRARY_PATH=~/lib:\$LD_LIBRARY_PATH”。

例如：

1. 假设 libgfortran.so.3 等系统库找不到：

error while loading shared libraries: **libgfortran.so.3**: cannot open shared object file: No such file or directory

解决办法如下：

\$ locate libgfortran.so.3

/usr/lib/gcc/x86_64-redhat-linux/4.4.4/libgfortran.so

/usr/lib/gcc/x86_64-redhat-linux/4.4.4/32/libgfortran.so

/usr/lib64/libgfortran.so.3

/usr/lib64/libgfortran.so.3.0.0

\$ source /BIGDATA1/app/osenv/set.sh

然后提交作业。

2. 假设 libmkl_sequential.so 等 intel 库找不到：

error while loading shared libraries: **libmkl_sequential.so**: cannot open shared object file: No such file or directory

解决办法如下：

\$ locate libmkl_sequential.so

/opt/intel/composer_xe_2013_sp1.2.144/mkl/lib/ia32/libmkl_sequential.so

/opt/intel/composer_xe_2013_sp1.2.144/mkl/lib/intel64/libmkl_sequential.so

/opt/intel/composer_xe_2013_sp1.2.144/mkl/lib/mic/libmkl_sequential.so

\$ module load intelcompiler/14.0.2

\$ module load intelcompiler/mkl-14

然后提交作业。

✓ “ls” 等访问文件夹操作很慢。

出现“ls” 等访问文件夹操作慢的原因主要有 3 个：一是网络慢，网络时延大；二是有大量的 IO 操作正在进行，造成 IO 阻塞；三是该文件夹下的文件过多（有成千上万个文件）。若是原因一和二，通常等一段时间后即可恢复正常；若是原因三，则需用户将自己文件夹下的文件分开存放。

✓ 重新生成 Private Key

Private Key 可以重新生成，用户在登录天河二号系统后，按照如下步骤操作
第一步：

\$ cd ~/.ssh


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

第二步：

```
$ ssh-keygen -t rsa (一直输入回车；若出现“Overwrite (y/n)?”，请输入“y”。)
```

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

```
$ cp id_rsa.pub authorized_keys
```

```
$ chmod 400 authorized_keys
```

```
$ cd ..
```

```
$ chmod 700 -R .ssh
```

第三步：

将 id_rsa (Private Key linux 系统版) 的内容复制到本地的文本文件中，新建一个终端用此新文件用作 Private Key 文件登录。若登录成功则新的 Private Key 生成成功。若不成功，重复第二、三步操作。若多次操作不成功，请解压第一步中生成的压缩包 (\$ tar -xvf bak.tar)，并使用旧 Private Key 登录。

✓ 提交作业报“Invalid partition name specified”。

报该错时，建议用户先用“yhi”查看是否可以看见自己所在的分区。若无法看见分区，则是您的机时已到限制。

✓ 提交作业报“Failed to allocate resources: User's group not permitted to use this partition”。

用户提交作业时通常需要加“-p 分区名”这一参数，同时该参数应写在程序名前。分区可用“yhi”来查看所在分区。

✓ 采用 yhrun 提交作业，关闭界面后，再次登录时发现作业被 killed。

yhrun 是交互式提交作业模式，一旦作业提交的界面关闭作业就会被 killed。若需要较长时间运行的作业，建议用户采用 yhbatch 批处理提交方式。yhbatch 负责资源分配，yhbatch 获取资源后会在获取资源的第一个结点运行提交的脚本，当前登录 shell 断开后，加载作业仍可正常运行。

✓ 采用 yhbatch 提交多结点作业失败的原因。

采用 yhbatch 提交作业首先进行的是分配资源，因此对于多结点作业，采用 yhbatch 提交时应在提交命令中指定 -N 参数，即提交命令是“yhbatch -N nodenum -n pronum -p pardition job.sh”。

✓ **计算结点无法登录。**

目前我们对计算结点做了限制，除非用户分配了计算结点，否则无法登录。用户若想登录计算结点再算题，首先需要用 `yhalloc` 分配结点，方可登录结点算题。

✓ **yhalloc 分配资源，退出 yhalloc 后发现作业断掉。**

`yhalloc` 与 `yhbatch` 最主要的区别是，`yhalloc` 命令资源请求被满足时，直接在提交作业的结点执行相应任务，适合需要指定运行结点和其他资源限制，并有特定命令的作业。当当前登录 `shell` 断开后，申请获得的资源以及加载作业任务会退出。

✓ **如果遇到一些作业报错，应该如何处理？**

较为常见的报错如：“No enough endpoint resources”，“Job credential expired”，“bus error”，用户可以通过日志找到相关的报错结点，在提交作业命令中使用参数“-x 结点名称”剔除掉问题结点重新进行作业提交，如“-x cn1”表示在我申请的资源中不使用 `cn1` 这个结点。如遇到相关报错问题也希望您能及时与我们进行联系，并提供您的报错日志信息(并加上错误发生的时间,提交命令等信息)，以便我们进行有效的分析和处理。

✓ **天河二号作业提交模式。**

目前天河二号系统采用独占式作业提交模式，即作业一旦提交到计算结点，则该结点被您独占使用。也就是说，一旦作业提交到计算结点，即使该结点的 CPU 核没有用满，其他人的作业也无法提交上去。

✓ **作业退出后仍显示 CG 状态，是否影响作业退出？**

CG 状态是作业退出时，部分结点上的进程没有完全停止导致，并不影响作业的正常退出。

✓ **作业完成退出时显示部分进程被 killed，然后退出。**

这种情况下，用户首先应检查所需的输出是否已正常输出完成。导致这种情况出现的原因是有部分进程先完成了计算而提前结束，而当一个作业的部分进程结束，系统默认为作业已完成，在一定时间内其他进程若不结束，则会被强制结束。