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 1.0

钟英

2018 年 2 月 10 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@nscc-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.nscc-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 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：

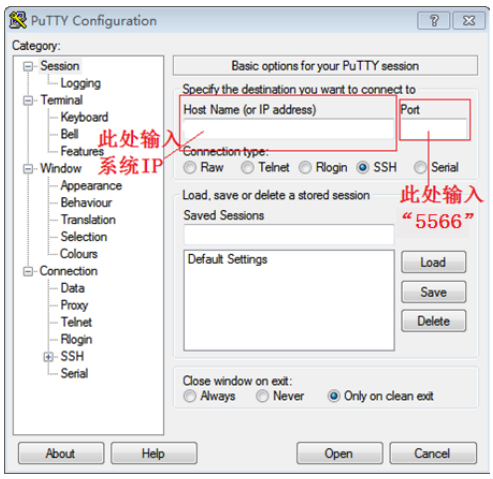
下文提到的系统 IP 为 172.16.22.11，端口号为 5566。

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“22”。

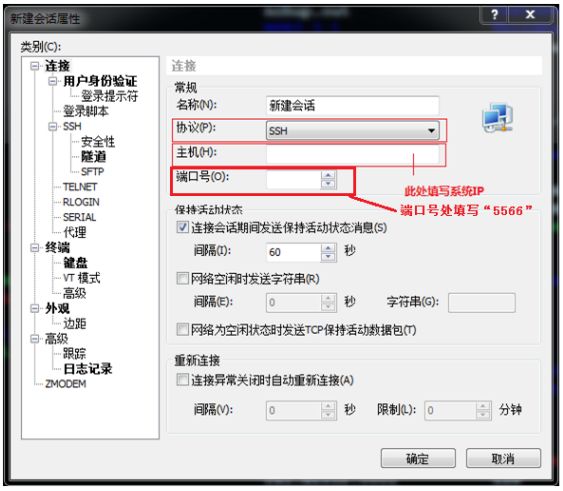


选择“Connection”->“SSH”->“Auth”，然后点击“Private key file for authentication”处的“Browse”选择转换后的 Private Key 文件，然后点击“Open”打开登录界面。

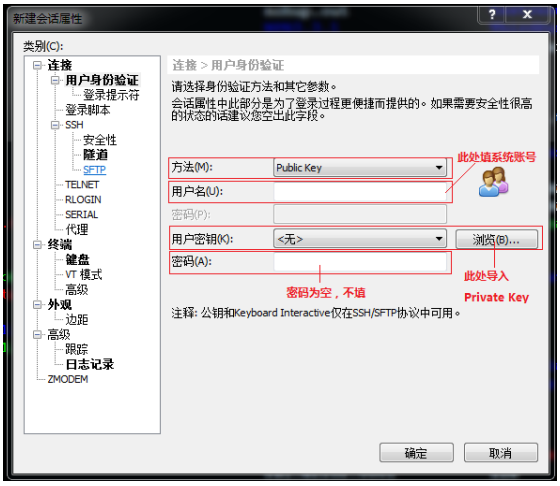
在登录界面里输入系统账号，回车即可登录。

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

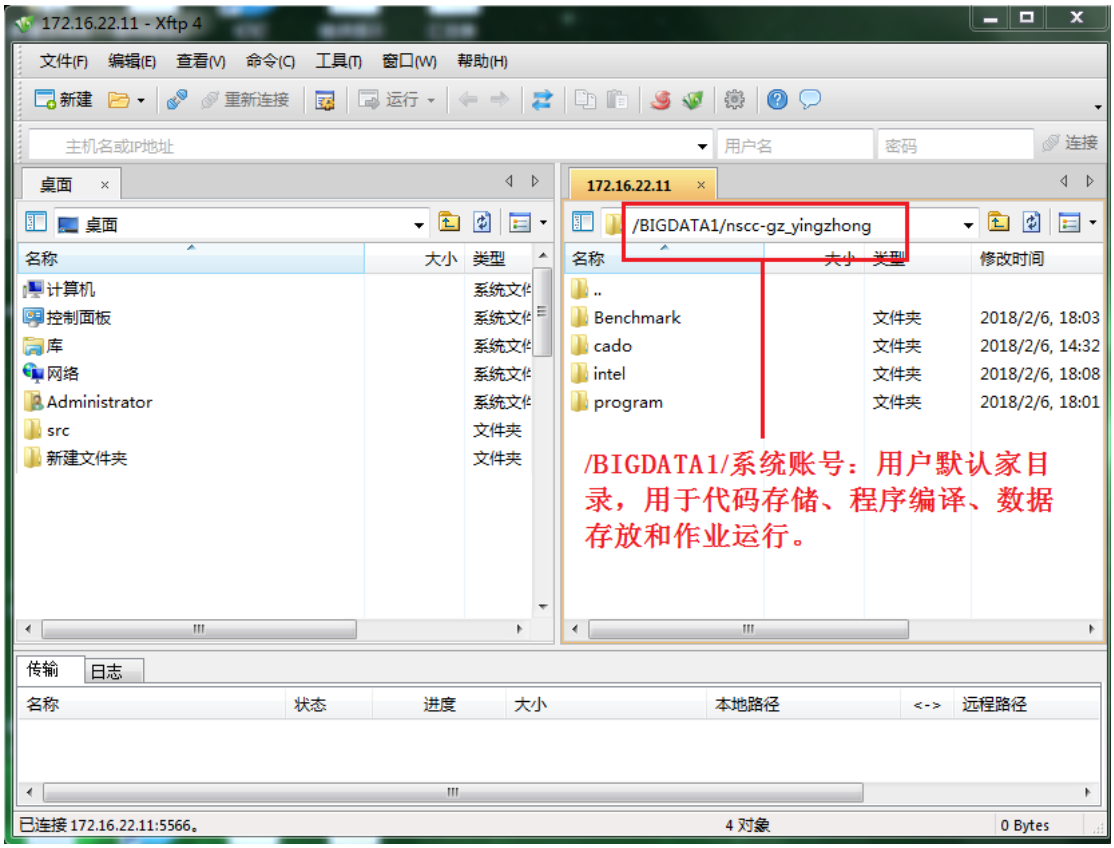
注意：

建议用户在第一次登录后重新生成 Private Key，具体操作见下文 FAQ 中的“重新生成 Private Key”，以避免 Private Key 泄露导致的数据泄露问题。

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。



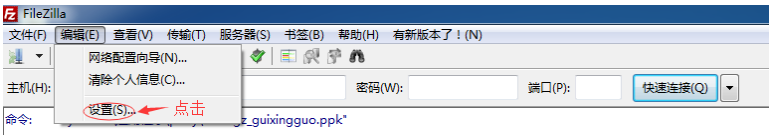
该分区下的用户主目录为用户默认家目录，用于代码存储、程序编译、数据存放和作业运行。在脚本中使用“/BIGDATA1/系统账号”引用该分区下的用户主目录。对于试用用户，其磁盘限额每用户不超过 1TB；对于签约用户，其磁盘限额将根据用户的具体需求进行配置。所有登录服务结点和计算结点都可以访问该分区下的文件。

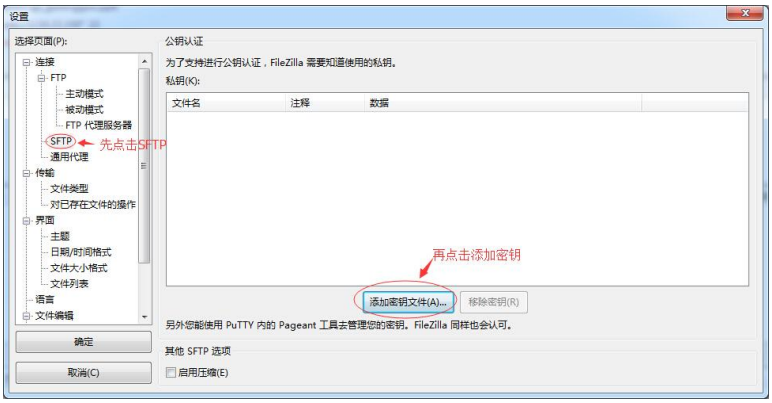
2.2 数据传输

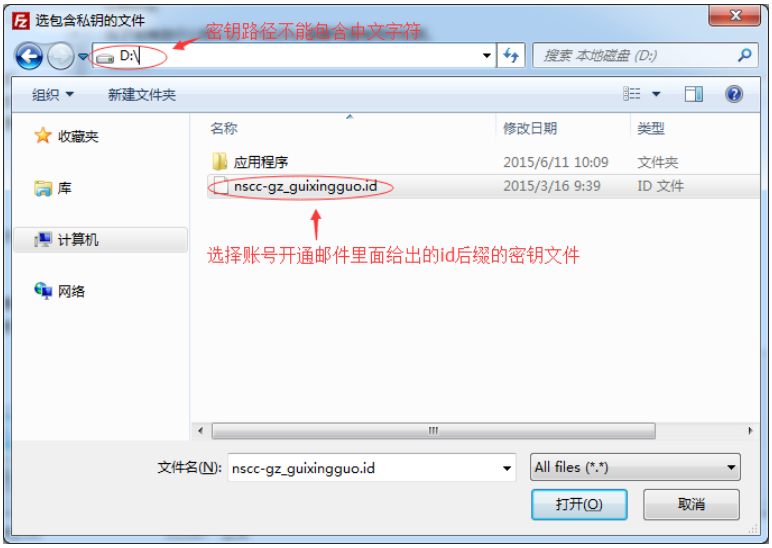
从外部机器向天河二号上传或下载文件，可以使用 sftp 客户端，例如使用 FileZilla、WinScp、FlashFTP 等的 sftp 数据传输软件，或者使用 Xmanager 等本身自带的文件传输功能。

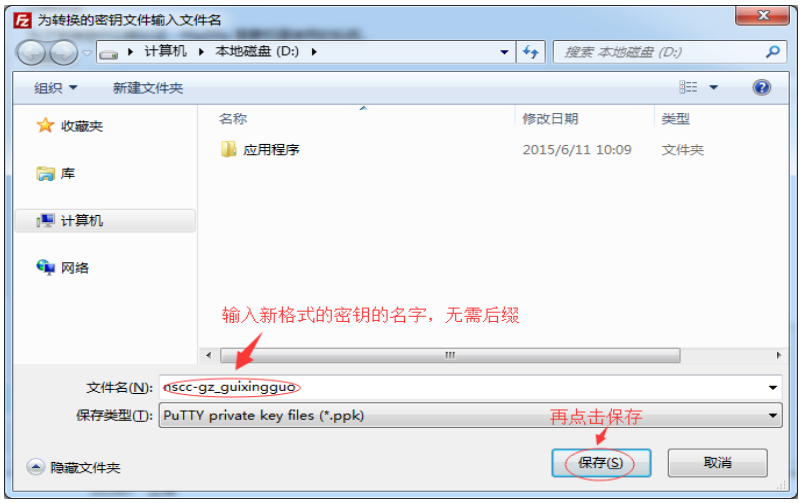
2.2.1 FileZilla 登录

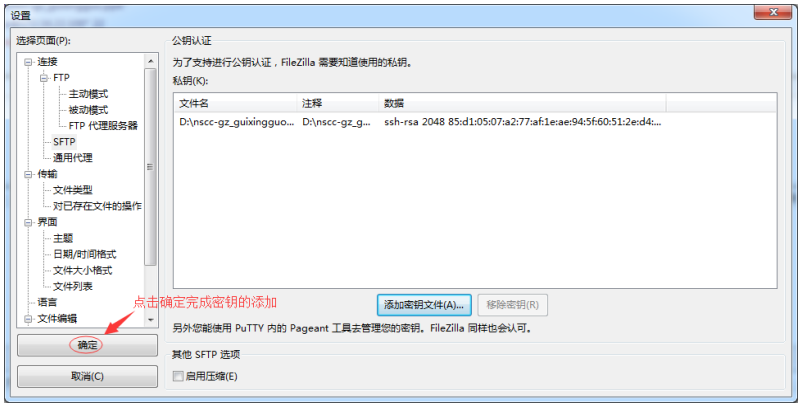
FileZilla 的登录步骤见下图：

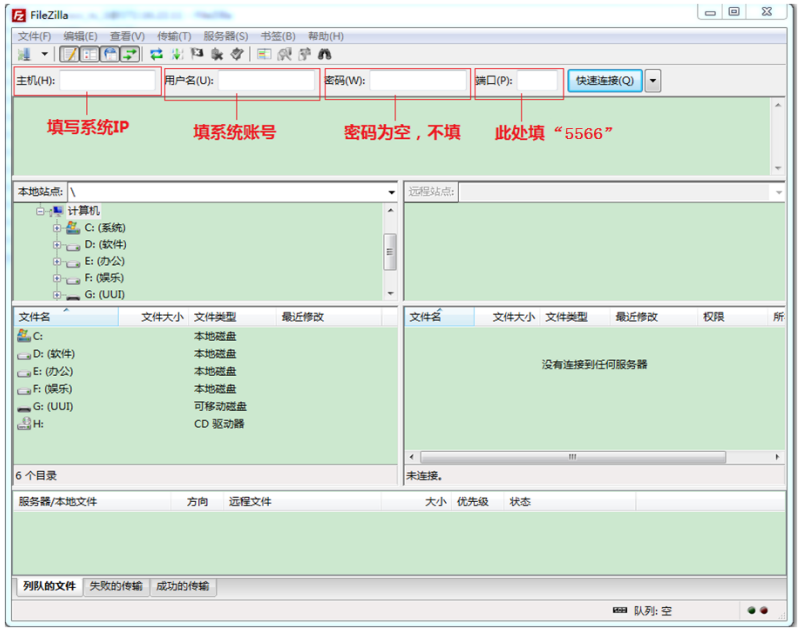








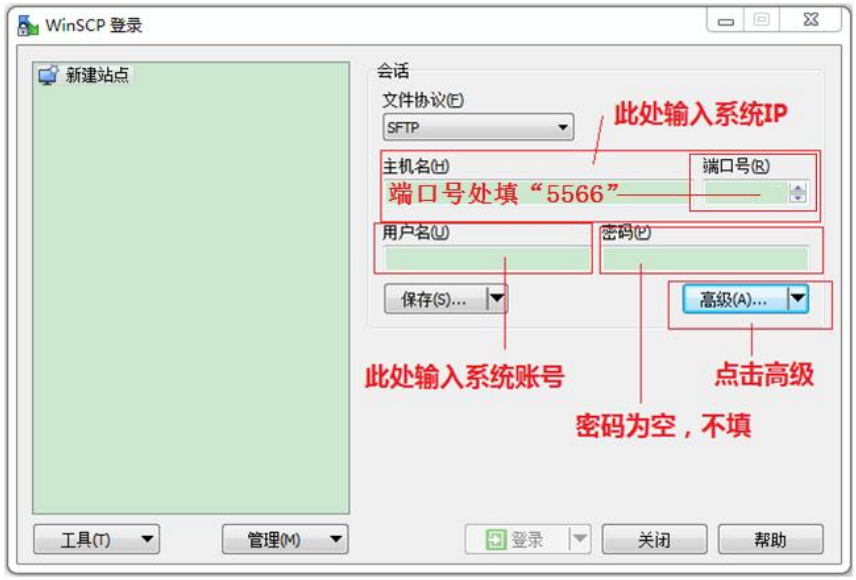


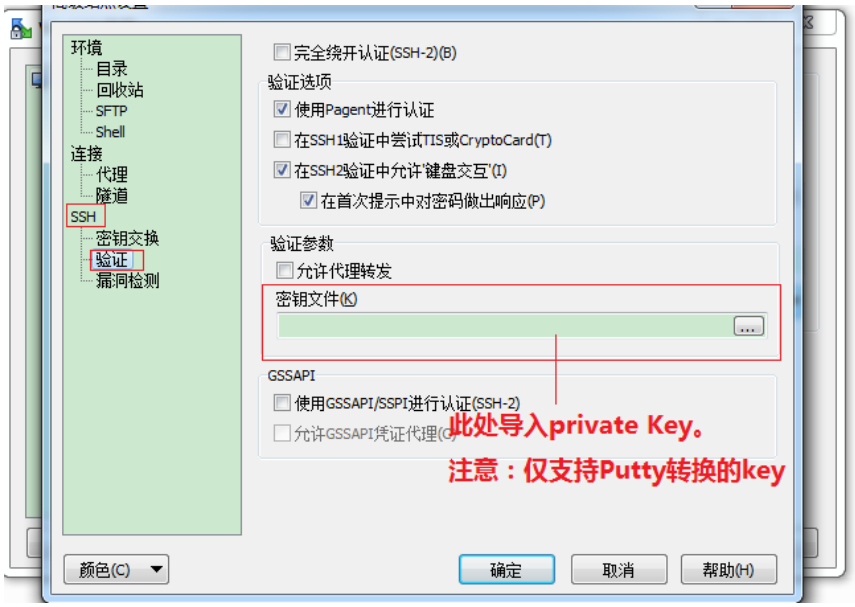


注意：密钥所在路径不能含有中文。

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。

```
[nscc-gz_yingzhong@lon5 ~]$ yhi
PARTITION  AVAIL  TIMELIMIT  NODES  STATE NODELIST
docker_128    up   infinite      1 drain* cn7008
docker_128    up   infinite     22  down* cn[6926,6930,6932,6934,6948,6950,6954-69
55,6974-6975,6985,6987,6994,6997-6998,7000,7003,7006,7009,7012,7015,7020]
docker_128    up   infinite    105   idle cn[6912-6925,6927-6929,6931,6933,6935-69
47,6949,6951-6953,6956-6973,6976-6984,6986,6988-6993,6995-6996,6999,7001-7002,7004
-7005,7007,7010-7011,7013-7014,7016-7019,7021-7039]
localdisk     up   infinite     13 drain* cn[7168,7178-7179,7200-7201,7218-7221,72
40,7244-7245,7273]
localdisk     up   infinite    140  down* cn[7040-7167,7180,7208,7210-7211,7242,72
48,7252,7267-7268,7270,7272,7286]
localdisk     up   infinite      1  drain cn7190
localdisk     up   infinite      1  alloc cn7209
localdisk     up   infinite    101   idle cn[7169-7177,7181-7189,7191-7199,7202-72
07,7212-7217,7222-7239,7241,7243,7246-7247,7249-7251,7253-7266,7269,7271,7274-7285
,7287-7295]
bigdata*      up   infinite    146 drain* cn[7353,7381,7478,7514-7515,7569,7590,76
08,7615,7632,7642,7644,7660,7670,7672,7678,7716,7745,7760,7778-7783,7792-7793,7803
,7824-7825,7827,7830,7832-7837,7854,7868,7881,7900,7912,7930,7951,7988,7996,8014,8
032,8037,8104,8135,8165,8172,8191-8207,8224-8239,8256-8271,8280,8282-8319,8412,842
2,8424,8437]
bigdata*      up   infinite   1023  down* cn[7298,7302,7307-7313,7315-7318,7321,73
24-7325,7330-7331,7335,7339-7340,7342,7348,7354,7363,7373-7374,7392,7401-7405,7409
,7412,7417,7422,7424-7425,7428-7429,7432,7436,7438-7439,7441-7452,7454-7455,7459-7
460,7462,7465-7468,7470,7472,7475-7477,7482-7485,7489,7498,7502-7510,7512-7513,751
7-7518,7523,7525,7533-7542,7546-7547,7556,7558-7559,7562-7568,7570-7571,7581-7582,
7586,7593,7595,7610,7631,7635,7640-7641,7652,7656,7661,7667,7684,7689,7695-7699,77
05,7707-7709,7714,7717-7731,7733-7736,7754-7756,7758,7775,7808,7810,7820-7823,7826
,7828-7829,7831,7838-7841,7843,7846-7847,7852-7853,7861,7863,7874,7886,7893-7894,7
914,7917,7919,7931,7946,7948-7949,7955-7956,7958-7963,7974,7991,7993,8002-8006,802
4,8027,8040,8042,8068-8071,8078,8082,8085,8091-8092,8096,8105,8107,8119,8121,8123,
8125,8130,8139,8148-8149,8179-8181,8183-8184,8326,8333,8369-8370,8372,8388-8391,83
97,8400,8403,8421,8426,8435,8438-8439,8448-8451,8453-8507,8509-8692,8694-8869,8872
-9215]
bigdata*      up   infinite      1   drng cn7314
bigdata*      up   infinite      7  drain cn[7516,7530,7618,8360,8436,8870-8871]
bigdata*      up   infinite    743   idle cn[7296-7297,7299-7301,7303-7306,7319-73
20,7322-7323,7326-7329,7332-7334,7336-7338,7341,7343-7347,7349-7352,7355-7362,7364
-7372,7375-7380,7382-7391,7393-7400,7406-7408,7410-7411,7413-7416,7418-7421,7423,7
```

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。

```
[nscc-gz_yingzhong@lon5 ~]$ yhq
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(R
EASON)
              1033    bigdata     bash nscc-gz_ying  R      57:56      1 cn7314
[nscc-gz_yingzhong@lon5 ~]$
```

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。

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：

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ yhrun -n 4 -p bigdata hostname
cn7314
cn7314
cn7314
cn7314
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

2）在分区 bigdata，结点 cn[7303-7306]上运行 hostname：

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ yhrun -n 4 -w cn[7303-7306] -p bigdata hostname
cn7303
cn7304
cn7306
cn7305
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

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[7303-7304]：

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ yhrun -n 4 -N 4 -w cn[7303-7304] -p bigdata hostname
cn7303
cn7304
cn7305
cn7306
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

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[7303-7304]：

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ yhrun -n 4 -N 4 -x cn[7303-7304] -p bigdata hostname
cn7327
cn7326
cn7328
cn7329
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

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：

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ 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。

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ ll
total 4
-rw-r--r-- 1 nscc-gz_yingzhong nscc-gz 48 Feb  3 11:50 mybash.sh
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

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

```
[nscc-gz_yingzhong@lon5%tianhe2-H test]$ ll
total 4
-rwxr-xr-x 1 nscc-gz_yingzhong nscc-gz 48 Feb  3 11:50 mybash.sh
[nscc-gz_yingzhong@lon5%tianhe2-H test]$
```

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：

```
[nscc-gz_yingzhong@lon5 ~]$ yhalloc -N 1 -p bigdata
yhalloc: Granted job allocation 1051
```

通过 yhq 查看相应的 jobID 为 1051，结点为 cn7314。

```
[nscc-gz_yingzhong@lon5 ~]$ yhq
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(REASON)
              1051   bigdata     bash nscc-gz_ying  R       0:07      1 cn7314
```

用户可以选择如下方式：

```
[nscc-gz_yingzhong@lon5 ~]$ ssh cn7314
The authenticity of host 'cn7314 (<no hostip for proxy command>)' can't be established.
RSA key fingerprint is f1:dd:eb:26:58:f8:d3:67:c0:fd:fd:66:8d:11:4f:dc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'cn7314' (RSA) to the list of known hosts.
Last login: Sat Feb  3 17:10:21 2018 from cn13725
[nscc-gz_yingzhong@cn7314%tianhe2-H ~]$
```

切换到 cn7314 结点，之后执行程序。

5.6 任务取消 yhcancel

用户使用 yhcancel 命令取消自己的作业。命令格式如下：

yhcancel　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 yhalloc 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 yhalloc 进程。用户可以选择通过 yhalloc 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是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-gz_yingzhong@lon5 ~]$ yhi
PARTITION  AVAIL  TIMELIMIT  NODES  STATE NODELIST
docker_128    up   infinite      1 drain* cn7008
docker_128    up   infinite     22  down* cn[6926,6930,6932,6934,6948,6950,6954-69
55,6974-6975,6985,6987,6994,6997-6998,7000,7003,7006,7009,7012,7015,7020]
docker_128    up   infinite    105   idle cn[6912-6925,6927-6929,6931,6933,6935-69
47,6949,6951-6953,6956-6973,6976-6984,6986,6988-6993,6995-6996,6999,7001-7002,7004
-7005,7007,7010-7011,7013-7014,7016-7019,7021-7039]
```

分区名

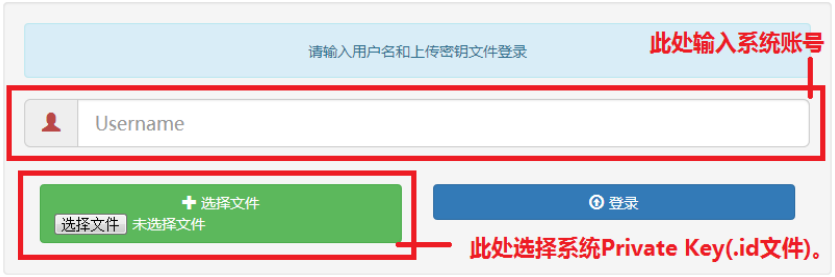
6 常见上机问题（FAQ）

✓ **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。**

登录 VPN 后，在**非 IE 浏览器网页**中输入网址（**http://172.16.22.11:10021/**）并打开。如图所示，输入系统账号以及选择秘钥（Private Key）文件后，点击登录即可查看机时以及磁盘限额情况。



✓ **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/ln1/set2.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/ln1/set2.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。