



国家超级计算天津中心

HPC1 系统用户手册

二〇一七年九月



目 录

1 HPC1 系统运行环境	1
1.1 平台架构	1
1.1.1 登陆结点	1
1.1.2 计算结点	1
1.1.3 互联网络	2
1.1.4 存储	2
1.2 系统环境	2
1.3 软件环境	3
1.3.1 编译器	3
1.3.2 MPI 编译环境	4
1.3.3 基础应用软件	5
1.3.4 常用科学计算软件	7
2 系统访问方式	9
2.1 基本条件	9
2.2 登陆 VPN	9
2.2.1 新版 VPN 登陆	10
2.2.1.1 Windows 操作系统	10
2.2.1.2 Linux 操作系统	18
2.2.1.3 MAC 操作系统	23
2.2.3 VPN 登陆注意事项	26
2.3 登陆服务器和数据传输	27
2.3.1 登陆服务器	27
2.3.2 文件传输	28
2.4 用户帐号密码修改	28
3 作业提交	29
3.1 使用限制	29
3.1.1 计算分区	29
3.1.2 用户权限限制	29
3.2 状态查看命令	30
3.2.1 结点状态查看 yhinfo 或 yhi	30
3.2.2 作业状态信息查看 yhqueue 或 yhq	30



3.3 提交作业	32
3.3.1 批处理作业 yhbatch	33
3.3.1.1 MPI 并行作业	35
3.3.1.2 openMP 并行作业	36
3.3.1.3 MPI+openMP 并行作业	36
3.3.2 交互式作业提交 yhrun	37
3.3.3 应用软件作业提交举例	38
3.3.4 其他典型脚本举例	42
3.4 任务取消 yhcancel	43
4 常见问题	44
4.1 VPN 登陆问题	44
4.2 系统登陆问题	44
4.3 编译问题	45
4.4 作业运行问题	47
4.5 存储问题	50
5 技术支持	51
附录 A 常用 Unix 命令	52
A1 基本命令	52
A2 目录操作	52
A3 文件创建、复制与删除	52
A4 文件属性	52
A5 文件显示与连接	53
A6 文件查找与比较	53
A7 文件压缩与备份	53
A8 输入输出重定向	53
附录 B 常用 vi 命令	54
B1 进入与退出 vi	54
B2 移动光标	54
B3 正文输入、删除、替换、恢复和查找命令	54
B4 行编辑命令	55



HPC1 系统用户手册

1 HPC1 系统运行环境

1.1 平台架构

HPC1 系统用户服务部分包括：登陆结点、计算结点两个部分，各自详述如下：

1.1.1 登陆结点

表 1-1 登陆结点配置

CPU*2	型号	2* Intel Xeon CPU E5-2680 v2 @ 2.80GHz 20cores		
L3 缓存	25MB			
内存	64GB			
操作系统	版本	RedHat 7.2	内核	3.10.0-327.el7.x86_64

作用：登陆结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** 终端登录到登陆结点上。

允许操作：在登陆结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，数据传输等操作。

禁止操作：禁止用户直接在登陆结点上运行计算程序。

登陆结点名称：**th-hpc1-ln0**，IP 地址 **192.168.2.101**。

1.1.2 计算结点

计算结点挂载共享存储，具备软件运行所需的运行环境，具体配置如表 1-2 所示：

表 1-2 计算结点配置

CPU*2	型号	2*Intel Xeon CPU E5-2690 v4 @ 2.60GHz 14cores		
L3 缓存	35MB			
内存	128GB			
操作系统	版本	RedHat 7.2	内核	3.10.0



作用：计算结点为用户提供一个大规模并行计算资源，用户可以将自己的作业通过作业调度系统提交到计算结点上运行。计算结点上具备程序运行所需的运行环境，但不具备软件编译环境。

用户在无作业的情况下无法直接登录到计算结点上，但可以通过 **ssh** 服务登录到正在运行用户自身作业的计算结点上查看自己程序的运行情况。

1.1.3 互联网络

采用商用 InfiniBand ConnectX FDR 56GB/s 互联网络，支持各类开源科学计算软件和商用软件。

1.1.4 存储

HPC1 系统采用 Lustre 分布式存储文件系统，该文件系统由多个存储结点构成，对外提供一个统一的大分区，供所有登陆结点与计算结点进行挂载。

作用：**Lustre** 高性能存储用于用户进行计算任务的数据存储，用户在登陆系统后看到的数据存储在 **Lustre** 存储中。

注意：为了满足并行计算对共享存储的高速读写需求，**Lustre** 分布式存储不提供多副本服务，因此无法长期保证用户数据的安全，建议用户及时拷贝自己的核心数据结果。

1.2 系统环境

HPC1 系统有一个共享存储，对应一个共享目录/THL6，计算结点均可访问。
除该共享目录外，计算结点无法访问登陆结点上的其他目录。

- 常用软件安装目录：/THL6/software

该目录属于共享目录的下级目录，存放用户常用的软件或编译器等。该目录仅提供用户普遍需要使用的常用软件，会随着用户的需求不断更新，软件由管理员负责安装，为用户提供使用。

1.3 软件环境

HPC1 系统采用 Environment modules 工具用来快速设置和修改用户编译运行环境。Environment modules 通过加载和卸载 modulefile（环境配置文件）可直接改变用户的环境变量，用户不需要频繁修改.bashrc 文件，从而避免误操作。

如下为 Modules 常用命令，如果用户计划使用某一软件，先在 shell 命令行下输入 module av 查看所需软件是否已在系统中安装；若已安装，则通过 module help 命令具体查看所需软件的版本、依赖环境等；然后在 shell 命令行下通过 module add 加载环境变量，最后提交脚本。

表 1-3 Modules 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 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

通过 module av 可查看系统可用的软件，命名方式遵循“软件名称/软件版本-编译器版本-MPI 版本”，以 namd/2.12-icc16-IMPI5.1 为例，该软件为 NAMD，版本 2.12，使用 Intel 编译器和 Intel 的 MPI 编译生成。

用户退出当前 shell 环境后，已加载的环境变量会取消，可以将 module add 加载命令加入到.bashrc 文件中，使得变量的加载永久生效。

目前由于计算节点上没有安装 Modules 环境，因此以上 Modules 相关命令在计算节点上无法执行。

1.3.1 编译器

表 1-4 编译器环境



编译器	版本	加载命令
Intel	16.0.3	module add Intel_compiler
GCC	4.8.5	/usr/bin/gcc
PGI	Community Edition Version 16.10	module add PGI

备注：推荐使用 Intel 编译器。如需使用 Intel MKL，使用命令 `module add MKL` 即可。

1.3.2 MPI 编译环境

表 1-5 MPI 编译环境

MPI 版本	调用的后端编译器	对应 module	加载命令
IMPI-5.1.3.210	Intel	MPI/Intel/IMPI/5.1.3.210	module add Intel_compiler module add MPI/Intel/IMPI
mvapich2-2.2	Intel	MPI/mvapich2-2.2/intel2016u3	module add Intel_compiler module add MPI/mvapich2-2.2/intel2016u3
	GCC	MPI/mvapich2-2.2/gcc4.8.5	module add MPI/mvapich2-2.2/gcc4.8.5
openmpi-2.0.2	Intel	MPI/openmpi-2.0.2/intel2016u3	module add Intel_compiler module add MPI/openmpi-2.0.2/intel2016u3
	GCC	MPI/openmpi-2.0.2/gcc4.8.5	module add MPI/openmpi-2.0.2/gcc4.8.5

备注：若使用后端调用 Intel 编译器的 MPI 实现，需确保使用 `mpiicc/mpiifort` 等命令；

若使用后端调用 GCC 编译器的 MPI 实现，需确保使用 `mpicc/mpif90` 等命令。

注意事项：

1. 由于系统计算节点采用 Intel 芯片，结合内部应用测试，建议使用 Intel 编译器及基于 Intel 编译器生成的 Intel 自带的 MPI 版本（IMPI，上表中加粗部分），这样计算和通信效率一般会较高。使用如 mvapich/openmpi 等的效率比 IMPI 低约 10%。

如果用户使用 makefile 或 autoconf 编译 MPI 并行程序，若使用 Intel 编译器，应将 makefile 中的 CC、CXX、F77、F90 等变量设置成 mpiicc、mpiicpc、mpiifort

和 mpiifort；若使用 GCC 编译器，应将 makefile 中的 CC、CXX、F77、F90 等变量设置成 mpicc、mpicxx、mpif77 和 mpif90 等。

1.3.3 基础应用软件

表 1-6 基础应用软件列表

类别	软件列表	加载命令	说明
文件读写	HDF5	module add Intel_compiler module add MPI/mvapich2-2.2/intel2016u3 module add hdf5	高效存储和分发科学数据的数据格式
	netCDF	module add Intel_compiler module add MPI/mvapich2-2.2/intel2016u3 module add netcdf	面向数组型并适于网络共享的数据的描述和编码标准
	nco	module add nco	操纵和分析 netcdf 等格式的数据工具
数据后处理	JasPer	module add Intel_compiler module add jasper	专门用于 JPEG-2000 Part-1 标准的图像编解码的工具
	libPNG	module add Intel_compiler module add libpng	PNG 图像文件存储格式标准的官方库
	Ncview	module add Intel_compiler/16.0.3 module add netcdf/4.4-icc16-mvapich2 module add ncview	用于查看 netcdf 文件及绘图
	Udunits	module add udunits	用于气候气象数据的后处理工具
数学库	BLAS	module add Intel_compiler module add blas	基础线性代数子程序库
	Lapack	module add Intel_compiler module add lapack	高性能的线性代数计算库
	ARPACK	module add arpack	大规模特征值求解程序库
	Boost	module add Intel_compiler module add MPI/Intel/IMPI module add boost	C++标准扩展库
	Hypre	module add Intel_compiler module add MPI/mvapich2-2.2/intel2016u3 module add hypre	网格划分、数据重拍、稀疏矩阵求解库
	Metis	module add Intel_compiler module add metis	网格划分、数据重拍、稀疏矩阵求解库（串行版）

	Parmetis	module add Intel_compiler module add parmetis	网格划分、数据重拍、稀疏矩阵求解库（并行版）
	Gsl	module add Intel_compiler module add gsl	GNU Scientific Library
	Libint	module add Intel_compiler module add libint	高斯函数求解，常用于 CP2K 等软件中
	Libxc	module add Intel_compiler module add libxc	密度泛函类软件的接口库
	FFTW	方式 1: module add Intel_compiler module add MPI/Intel/IMPI module add fftw 方式 2: module add Intel_compiler module add MPI/mvapich2-2.2/intel2016u3 module add fftw/3.3.4-icc16-mvapich2	快速傅里叶变换数学库
性能分析	scorep	module add scorep	性能分析工具
	scalasca	module add scalasca	性能分析工具
	tau	module add tau	性能分析工具
	collect	module add collectl	性能分析工具
	valgrind	module add valgrind	内存分析检测工具
工具类	Paraview	module add paraview	可视化工具
	VTK	module add vtk	可视化工具
	rar	module add rar	rar 文件压缩解压缩工具
	p7zip	module add p7zip	7z 文件压缩解压缩工具，也可解压缩 iso 文件
	libxml2	module add libxml2	C 语言的 XML 程式库
	glog	module add glog	C++开源日志库
	libiconv	module add libiconv	编码转换库
	muparser	module add muparser	跨平台的公式解析库
	ghostscript	module add ghostscript	PostScript 语言解释器
	fish	module add fish	UNIX/Linux 系统下的 SHELL 程序
	openexr	module add Intel_compiler/16.0.3 module add ilmbase module add openexr	用于计算机成像应用的高动态范围（HDR）图像文件格式

	julia	module add julia	面向科学计算的高性能动态高级程序设计语言
	perl	module add perl	计算机程序语言
	root	module add root	模块化科学软件框架
	Ocaml	module add ocaml	CamI 编程语言的主要实现
	cnvnator	module add cnvnator	寻找 CNV 和通过映射读取深度覆盖的基因分型的工具

1.3.4 常用科学计算软件

HPC1 系统配备有较为完备的应用软件环境，可支持用户进行各类主流应用程序的运行。部分软件如下，我中心技术人员会定期更新软件环境以满足用户需求。

表 1-7 常用科学计算软件列表

类别	软件列表	版本号
分子动力学模拟	NAMD	2.12
	Gromacs	5.1.2
	Lammps	2016.7.22 2017.8.11
	VMD	1.9.3
	Packmol	17.076
计算材料学	Quantum ESPRESSO	5.4.1 6.1
	SIESTA	4.0
	CP2K	4.1
天气预报	WRF	3.8
	WPS	3.8
计算化学	Nwchem	6.6
生物信息	Bedtools2	2.25.0
	hmmer	3.1b2
	impute2	2.3.2
	lastz	1.04.00

	plink2	20170320
	tophat	2.1.1
	vcftools	0.1.14
	beagle	4.0
	VarScan	2.3.9
	FishingCNV	1.5.3
	control-freec	10.5
	vcftools	0.1.14
	blat	36
	Trimmomatic	0.36
	cnvnator	0.3

用户使用须知：

1. 以上仅列出了部分在系统上已安装的软件，用户可通过设置环境变量直接使用。用户如有其他软件的移植、安装、兼容性测试，可直接联系相关工作人员。
2. 各类商业软件（如 VASP、Material Studio、Gaussian09）均可正常在我中心高性能计算平台上使用，但由于中心尚未购买对应软件版权，用户若使用这类商业软件，需提前购买版权获取使用许可，并仅限于该用户自己使用，且用户须对合法性负责，中心技术人员可协助完成软件的部署和测试。

2 系统访问方式

为了更好的保证用户的数据安全，中心采用 SSL VPN 实现远程用户对系统的登陆访问，用户需要首先登陆 VPN 后，才能使用中心资源。下面将详细介绍用户如何登陆 HPC1 系统。

2.1 基本条件

用户需要具备的基本条件如下：

1. 经过了中心用户基本审查创建流程，并填写了**相应的文件和协议**。
2. 具备一个 **VPN 账号及密码**。（用户如经过了审查创建流程，会收到用户账号创建成功的 email，里面会有 VPN 及系统的账号及密码）
3. 具备一个**系统用户及密码**。

具备上了上述条件，您就可以尝试登陆至 **HPC1 系统使用系统资源**。**首先登陆 VPN，然后登陆服务器**，步骤及所需软件如下节描述。

2.2 登陆 VPN

目前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。

另外，VPN 可支持手机登陆，目前安卓/苹果系统的手机均可登陆 VPN，这样可以方便用户及时查看作业状态。限于篇幅，手机登陆 VPN 的方式未在此列出，有需求的用户可联系 service@nscc-tj.gov.cn 索要手机登陆 VPN 的操作说明。

2.2.1 新版 VPN 登陆

2.2.1.1 Windows 操作系统

A 使用浏览器登陆 VPN

A-1 IE 及 IE 内核浏览器登录天河 SSL VPN

1) 打开中心网站主页，登陆 SSL VPN

登陆中心官网：<http://www.nsc-tj.gov.cn/>，在右上角用户 VPN 登录选项点击“新 VPN 入口”。或直接登录新 VPN 的网址：联通：<https://thvpn.nsc-tj.cn> 电信：<https://thvpn1.nsc-tj.cn>。



2) 输入用户名、密码登录 SSL VPN。



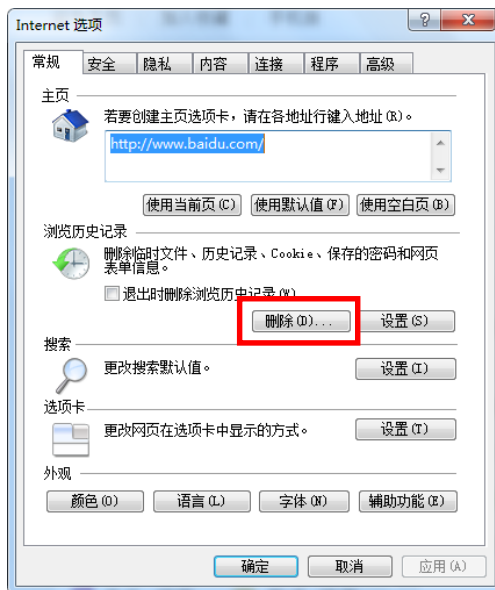
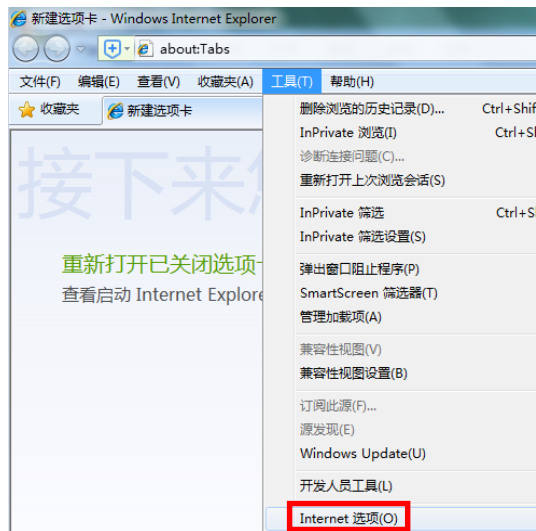
3) 为 IE 浏览器安装控件（仅初次配置时需要安装）。

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：

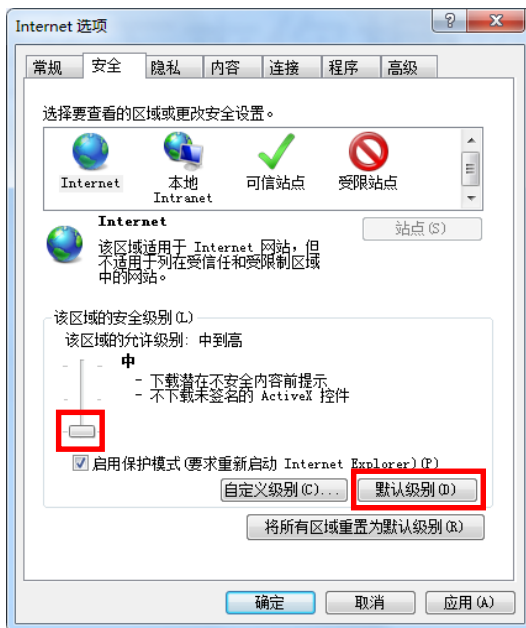


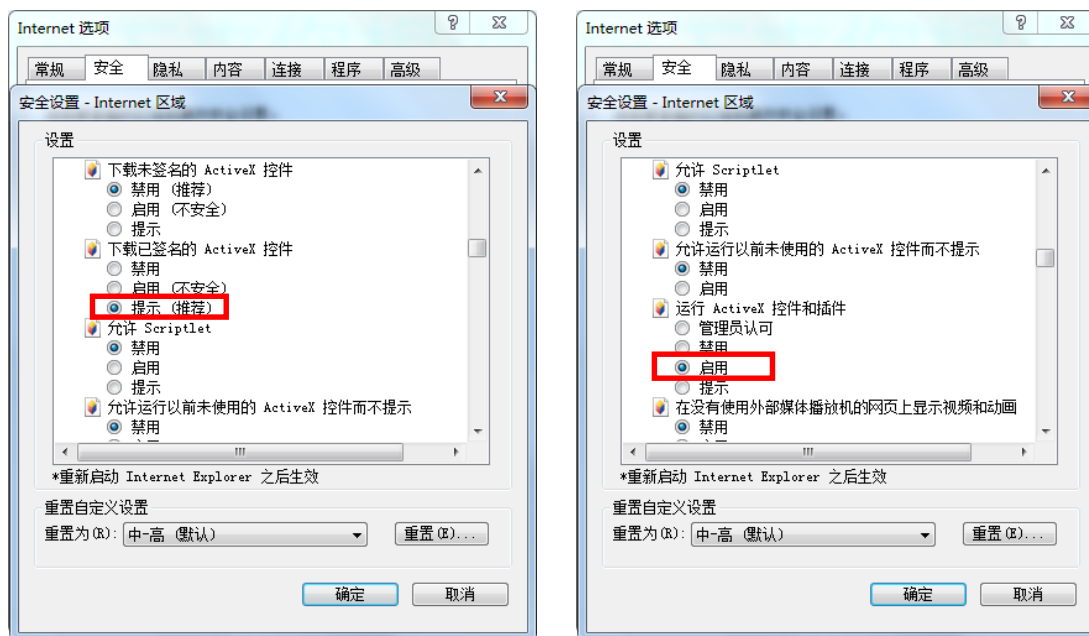
安装完成后，即可成功登陆 VPN。如果没有看到安装提示或安装失败，请参考下面处理步骤：

(1) 清理 IE 浏览器缓存,选择“菜单栏→工具→Internet 选项→常规→浏览历史记录→删除”，如下图：



(2) 浏览器安全级别设置太高，调整浏览器的安全级别至 "中" 或 "默认"。或修改下浏览器安全设置，允许下载和运行 Active X 控件和插件；（切换到 "Internet 选项" 的 "安全" 选项卡）



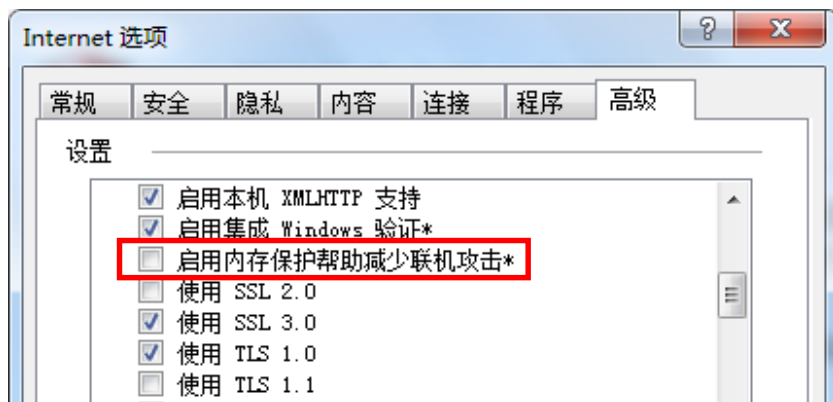


(3) 浏览器插件拦截控件下载，如上网安全助手等；

(4) 浏览器开启了增强保护模式，选择 "Internet 选项" 的 "高级" 标签页，将列表中的 "启用增强保护模式" 的勾选取消；



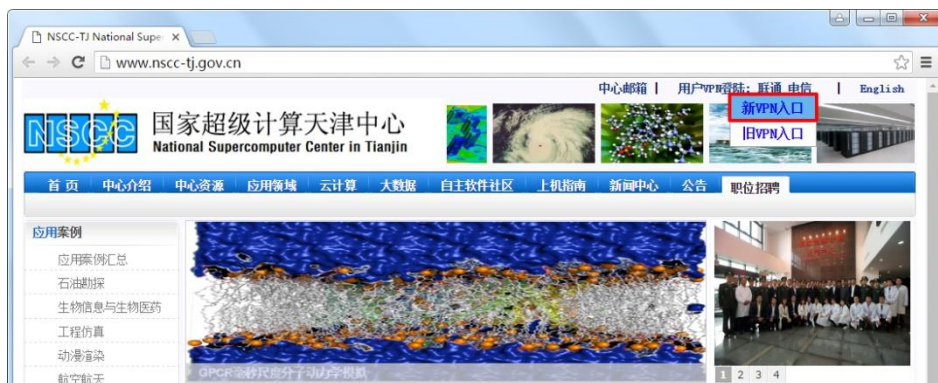
(5) 如果在 IE 8 下安装控件时出现崩溃，选择 "Internet 选项" 的 "高级" 标签页，将列表中的 "启用内存保护帮助减少联机攻击" 的勾选取消；



小提示：IE 浏览器版本要求 Internet Explorer 6.0 和以上版本，低于 Internet Explorer 6.0 版本建议升级。

A-2 其他浏览器登录天河 SSL VPN（例如 chrome 、firefox 等）

1) 登陆中心官网：<http://www.nsccl-tj.gov.cn/>，在右上角用户 VPN 登录选项点击“新 VPN 入口”。或直接登录新 VPN 的网址：联通：<https://thvpn.nsccl-tj.cn> 电信：<https://thvpn1.nsccl-tj.cn>。



2) 打开 sslvpn 页面，单击“下载安装组件”手动安装客户端。



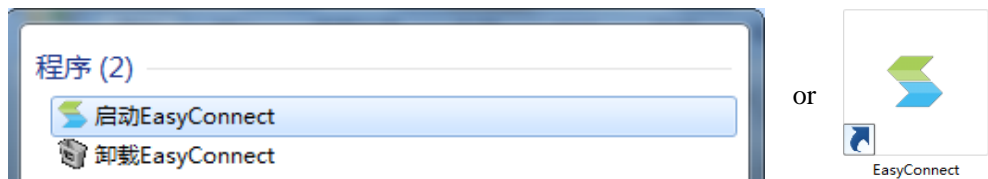
3) 系统会自动下载 EasyConnect 组件并安装，安装后即可成功登陆 ssl vpn。



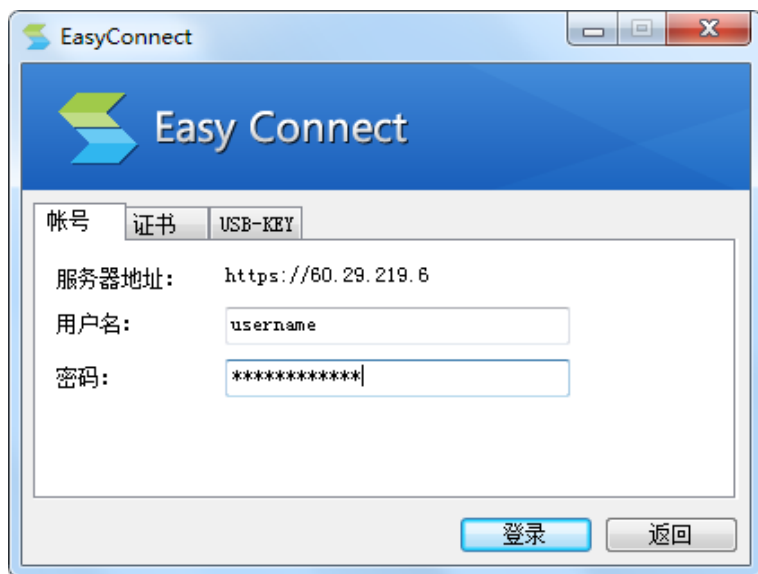
B 通过“Easy Connect”快捷方式登陆 VPN

第一次正常登陆 VPN 后，以后再次登陆可直接通过点击在桌面中的“Easy Connect”快捷方式，或在 windows 开始菜单寻找 Easy Connect 软件启动，输入账号和密码即可。

1) 在 Windows 的开始菜单（或桌面快捷方式）启用 Easy Connect 软件。



2) 输入用户名和密码，点击"登陆"后，会自动打开浏览器，跳转到 VPN 登陆成功页面。



C 访问 SSL VPN 资源登陆 HPC1 系统

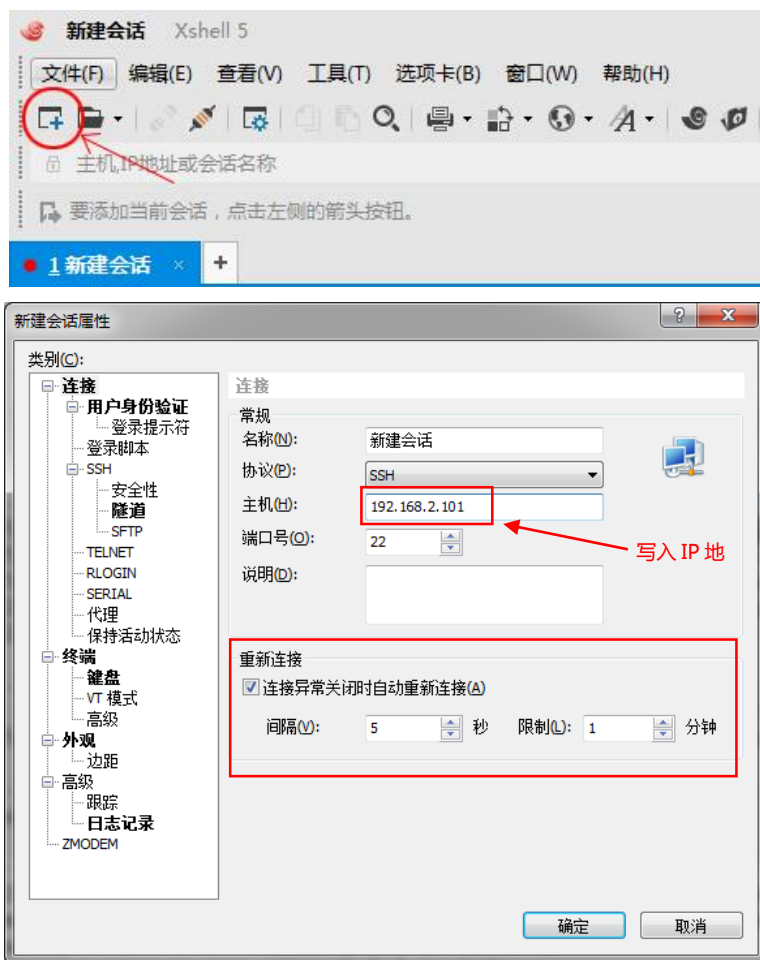
1) 按照以上方式成功登陆中心的 VPN 后，用户则可以通过 ssh 服务登陆天河系统登陆结点来使用中心资源。

其中 VPN 网页左上角“资源组列表”下有“TH-HPC1 系统”。请点击“TH-HPC1 系统”查看可用登陆节点（th-hpc1-ln0）的 IP 地址，如下图：



2) 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。另外，为避免 shell 终端长时间无人操作自动中断问题，建议按如

下红框设定自动重新连接。



3) 继续输入系统登陆的账户名和账户密码。



4) 顺利登陆系统。

2.2.1.2 Linux 操作系统

目前新版 VPN 在 Linux 系统下，对 32 位操作系统完全兼容，具体操作过程可参见下面 A 小节；针对 64 位操作系统，支持主流操作系统如 Ubuntu、CentOS、Fedora，但只能通过在命令行下运行特定脚本实现 VPN 登陆，具体操作过程参见下面 B 小节。

A 使用浏览器登陆 VPN（32 位操作系统）

1) 以 firefox 浏览器为例：登陆中心官网：<http://www.nsccl-tj.gov.cn/>，在右上角用户 vpn 登录选项点击“新 VPN 入口”。或直接登录新 VPN 的网址：联通：<https://thvpn.nsccl-tj.cn> 电信：<https://thvpn1.nsccl-tj.cn>。



2) 输入用户名、密码登录 SSL VPN



3) 下载 Java 虚拟机安装包

使用浏览器登录，系统如果没有安装 Java 虚拟机，会登录失败。请参考下图，下载 Java 安装包。



4) 下载完成后，打开终端，按下面命令操作：

```

如果当前用户不是管理员，则通过su 命令切换至管理员，再执行安装文件，操作如下：
[xxx@localhost ~]$ su
Password: <输入管理员密码>

[root@localhost root]# cd Downloads/    <切换至下载目录>
[root@localhost Downloads]# ls -l
total 21128
-rw-r--r-- 1 xxx xxx 21632626 2016-04-15 10:00 jre-for-linux.bin
[root@localhost Downloads]# chmod u+x jre-for-linux.bin    <给予执行权限>
[root@localhost Downloads]# ls -l
total 21128
-rwxr--r-- 1 xxx xxx 21632626 2016-04-15 10:00 jre-for-linux.bin
[root@localhost Downloads]# ./jre-for-linux.bin    <执行安装文件>
...
Creating jre1.6.0_27/lib/rt.jar
Creating jre1.6.0_27/lib/jsse.jar
Creating jre1.6.0_27/lib/charsets.jar
Creating jre1.6.0_27/lib/ext/localedata.jar
Creating jre1.6.0_27/lib/plugin.jar
Creating jre1.6.0_27/lib/javaws.jar
Creating jre1.6.0_27/lib/deploy.jar    <安装过程>

Done.
Linux JRE installed successfully.    <安装成功提示>
[root@localhost Downloads]#

```

小提示：

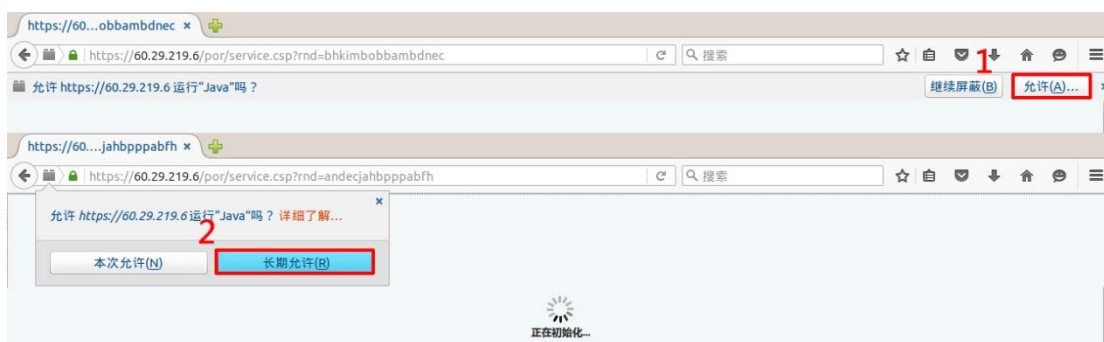
Ubuntu 系统，默认不启用管理员帐号，通过 su 命令无法提权，请使用 sudo 命令进行提权。

```

[xxx@localhost ~]$ su
[xxx@localhost ~]$ sudo ./jre-for-linux.bin

```

5) 重新启动 firefox 浏览器，登陆 SSL VPN，进行相关设置。



如果为非 root 用户，会弹出终端，按要求输入 root 密码即可。



如果弹出"警告-安全", 请勾选“始终信任”, 然后点击“是”, 见下图:



如果弹出"安全信息"警告, 请勾选“始终信任”, 然后点击“运行”, 见下图:



点击运行后, 会自动启动应用程序, 完成登陆。至此 Java 虚拟机配置完成。

B 使用命令行下运行特定脚本登陆 VPN (64 位操作系统)

有需求的用户可以与我们联系获取所需的脚本文件 `install_vpn_liunx_x86.sh`。
详见“四、联系我们”。

1) 安装运行 VPN 所需的 32 位库, 不同操作系统对应的命令有所不同, 具体如下
(需要切换为 root 权限)

a) CentOS 7 和 Redhat 7:

```
# yum install glibc.i686
```

```
# yum install libstdc++-4.8.5-4.el7.i686
```

b) Fedora 22 workstation:

```
# yum install glibc.i686
```

```
# yum install libstdc++-4.8.5-4.el7.i686
```

```
# yum install libstdc++.so.6
```

c) Ubuntu 16 和 Debian 8:

```
# apt-get install lib32stdc++6
```

d) Linux Mint 17:

跳过此步骤。

2) 将 `install_vpn_linux_x86.sh` 文件拷贝到 Linux 系统中（需要切换为 root 权限）

执行:

```
# chmod +x install_vpn_linux_x86.sh
```

```
# ./install_vpn_linux_x86.sh
```

安装完成

3) 登录 VPN 的过程（使用普通用户登陆）

(1) 设置 vpn 地址

```
$ svpnc -h thvpn.nscg-tj.cn:443
```

(2) 设置用户名密码

方法 1)

```
$ svpnc -u your_username:your_ssh_password
```

方法 2)

```
$ svpnc -U
```

```
[info] setting user from interactive input
```

```
please input username:your_username
```

```
get name ok, name username
```

```
please input password: your_ssh_password
```

```
getpwd ok, your_ssh_password
```

```
[info] set username ok, username your_username
```

```
[info] set password ok, password your_ssh_password
```

(3) 登陆 vpn

```
$ svpn -i
```

(4) 查询 vpn 运行状态

```
$ svpn -s
```

(5) 注销 vpn

```
$ svpn -o
```

(6) 退出 VPN

```
$ svpn -q
```

备注：使用-q 退出 VPN 后下次登录需要再设置用户名密码。

C 访问 SSL VPN 资源登陆天河系统

1) 按照以上方式成功登陆中心的 VPN 后，用户则可以通过 ssh 服务登陆天河系统登陆结点来使用中心资源。

其中 VPN 网页左上角“资源组列表”下有“TH-HPC1 系统”。请点击“TH-HPC1 系统”查看可用登陆节点（th-hpc1-ln0）的 IP 地址，如下图：



2) 打开 Linux 终端，使用 ssh 命令，输入用户名、IP 地址，然后输入系统登录密码即可。命令为“ssh yourusername@192.168.2.101”。

2.2.1.3 MAC 操作系统

A 使用浏览器登陆 VPN

1) 以 Safari 浏览器为例：登陆中心官网：<http://www.nsc-tj.gov.cn/>，在右上角用户 vpn 登录选项点击“新 VPN 入口”。或直接登录新 VPN 的网址：联通：

<https://thvpn.nscc-tj.cn> 电信: <https://thvpn1.nscc-tj.cn>。



2) 输入用户名、密码登录 SSL VPN



3) Safari 浏览器安装插件

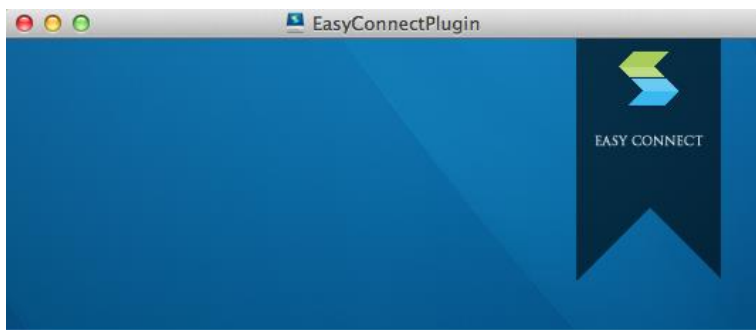
在登录页面点击“下载安装组件”，打开“安装浏览器控件”页面，点击“下载”，获得最新版本的 Safari 浏览器控件 EasyConnectPlugin.dmg，如下图：



也可以直接登录，在登录过程中，按照提示下载“Mac OS X 平台安装包”，或者不安装浏览器控件，仅使用 Web 服务。



浏览器控件 EasyConnectPlugin.dmg 会下载到“下载”目录，当下载完成时，双击程序包图标，启动安装向导，按照安装向导的提示操作，完成安装。



Install EasyConnect Plugin.pkg

当出现如下界面时，表示控件已安装完成，此时打开浏览器重新登录即可。



小提示：目前只支持 Safari 浏览器，请使用 Safari 浏览器登录。

2.2.3 VPN 登陆注意事项

VPN 登陆注意事项如下：

1. 在国家超级计算天津中心网站 www.nscg-tj.gov.cn 右上角有“用户 VPN 登陆”选项，用户也可以从那里点击进入（联通与电信用户请分别选择适合自己的网络接入）。
2. 如果您在 Windows 下安装有 360 安全卫士等防护软件，当有提示时，请点允许，来保证 VPN 的正常使用。
3. VPN 用户默认限制，**同时使用上限为 5 人**。（一台 PC 机只允许登录一个



VPN 账号) 如果您有特殊需求, 需要更多的人同时使用, 请告知中心。由于 VPN 账号同时允许多人登陆, 因此不允许修改密码, 您如果需要修改密码, 请联系中心技术人员。

4. 在登陆 VPN 后, 通过软件登陆系统, 进行编译、提交任务等操作时, 请不要关闭浏览器或退出 VPN, 否则会断开连接。

5. 系统默认用户登陆 VPN 后, 如不进行任何操作(登录系统等操作), 30 分钟后会自动下线。(如果您通过 ssh 软件连接天河登陆结点则不会自动下线)

6. 如果您需要退出 VPN 时, 请点击 VPN 右上角的“退出”, 而不要直接关闭浏览器。(关闭浏览器系统会保留该用户 5 分钟, 如果在该段时间内超过 VPN 用户限制 5, 您将短时无法登陆)

7. 您在使用中遇到问题可以参考“4.1 小节的 VPN 登陆问题解决”, 如果还无法解决则可以 email 或电话咨询。(相关联系方式见“5 小节的 技术支持”)

2.3 登陆服务器和数据传输

2.3.1 登陆服务器

按照以上方式成功登陆中心的 VPN 后, 用户则可以通过 ssh 服务登陆天河系统登陆结点来使用中心资源。为了保证用户的数据安全, 中心不提供 telnet 等其他连接方式。

中心资源通过 **TCP 应用** 的方式供用户使用, 用户可以使用 ssh 客户端软件(如 Xmanager、SecureCRT、Putty)来登录系统。SSH Secure Shell Client, SecureCRT, Putty 等均为免费软件, 网络上均有下载。

登录时, Host Name 项填写登陆节点对应的 IP 地址。

特别注意:

1、为了保障系统安全, 用户密码连续输入错误 5 次以后, 将被禁止登录 10 分钟。

2、系统会监控用户的行为, 如果用户故意实施对系统造成危害的行为, 我们

保留对该用户采取法律手段的权利。

2.3.2 文件传输

目前 HPC1 系统使用 th-hpc1-ln0 登陆节点提供数据的上传、下载服务。**Linux 和 Mac 用户可以直接使用 scp 等命令拷贝数据，此处不再详述。**

Windows 用户：从外部客户端向系统中上传或下载文件，可以使用 sftp 客户端，例如 SSH Secure Shell Client 等本身自带的文件传输功能。或者使用 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）。

同登陆服务器类似，Host Name 项填写登陆节点对应的 IP 地址。

2.4 用户帐号密码修改

目前系统采用 LDAP 进行用户管理，新创建的用户第一次登陆结点时会创建相应的工作目录。用户可以通过“passwd”命令修改用户密码，以 ncps 用户为例，举例说明如下：

```
[ncps@th-hpc1-ln0 ~]$ passwd
Changing password for user ncps.
Enter login(LDAP) password:
New password:
Re-enter new password:
LDAP password information changed for ncps
passwd: all authentication tokens updated successfully.
```

首先需要输入中心给分配的账户密码，之后再输入新的密码，重复输入一次后，就会显示密码更新成功。出于安全考虑，系统密码更新必须遵守如下规则：

密码至少位数 16 位；2、至少包含 3 种不同字符。

特别提示：为了保障用户的数据安全，您需要保证您的系统用户密码不外泄，希望您能经常更换系统用户密码。

如需更换 VPN 账号密码，请告知中心技术人员，我们帮您更换。

3 作业提交

HPC1 系统上的作业管理系统以计算结点作为并行作业的资源分配单位，实现并行作业的调度运行。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。资源管理系统首先将用户提交的应用程序构造成作业进行排队处理，然后根据系统的实时运行资源状态，决定何时以及在哪些计算结点中加载应用程序的运行，不同的应用程序之间不存在资源的竞争冲突，用户可以通过作业管理系统来监控应用程序的运行。

3.1 使用限制

3.1.1 计算分区

目前可用计算分区为 TH_HPC1，该分区对作业无最大运行时间限制，用户可以使用“`yhi -l`”或“`yhcontrol show partition partition_name`”命令，看到相应的分区的详细信息。

注意：

由于大型集群系统具备一定故障率，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。

3.1.2 用户权限限制

除了上述的分区限制，目前还根据用户的申请情况，针对用户做了一定的限制，该限制主要基于用户和中心签订合同的规模。

包括：最多可以使用的结点数、最多可以使用的核数、单个任务最多可以使用的结点数、单个任务最多可以使用的核数等。

通过命令“`yhacctmgr list association user myUserName`”可查看账号 `myUserName` 的具体权限设置。用户只有查看自己账号的权限，无查询其他账号的

权限。

用户在使用过程中，如果有超出自己合同范围内的计算规模的计算需求，请基于自己的需求，向中心提出申请，中心会根据用户需要审查后，进行一定的修改。

为了保证系统和用户数据的安全，目前普通用户不能在没有申请资源时，就 `ssh` 链接到计算结点，只有分配了相应的计算结点资源后，才能 `ssh` 到指定计算结点。

3.2 状态查看命令

在用户提交作业前，应先查看系统的使用情况，这样利于用户根据系统使用情况，进行选择。

3.2.1 结点状态查看 `yhinfo` 或 `yhi`

`yhi` 为 `yhinfo` 命令的简写，用户可以使用 `yhi` 或者 `yhinfo` 命令查看结点的使用情况，从而根据情况做出选择。可以通过命令 `yhi -l` 获得结点更为详细的信息。

表 3-5 `yhi` 输出的关键词说明

关键词	含义
PARTITION	用户可用的计算分区
AVAIL	可用状态：up 表示可用；down 表示不可用
TIMELIMIT	该分区的作业最大运行时长限制
NODES	结点数量
STAT	结点状态： down: 不可用状态 idle: 空闲状态 alloc: 被分配状态 drain: 排空状态，表示该类结点不再分配到其他作业
NODELIST	结点名字列表

3.2.2 作业状态信息查看 yhqqueue 或 yhq

yhqueue 或 yhq 命令用于查看系统中，各计算结点的运行情况。

表 3-6 yhq 输出的关键词说明

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	排队状态： AssociationResourceLimit: 关联的资源限制已满 Resources: 当前可用资源不能满足作业需求 Dependency: 作业的依赖关系未满足 PartitionDown: 作业所在的分区处于 down 状态

总结来讲，用户作业的状态主要有如下几种：

PD: 排队，pending

R: 运行中，running

S: 挂起中，suspended

CA: 被取消，canceled

CD: 成功结束，completed

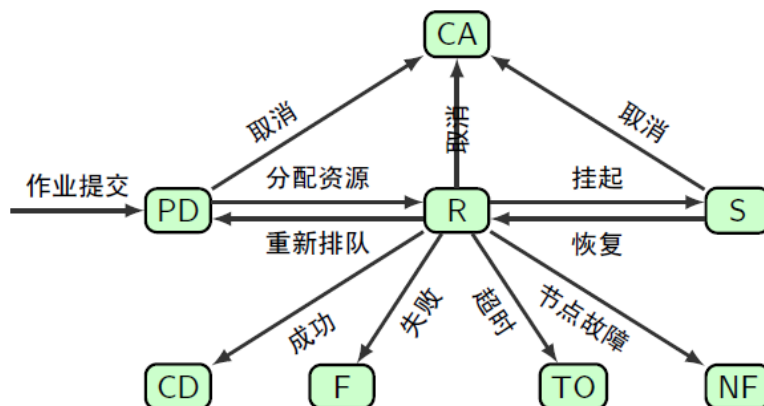
F: 失败结束，failed

TD: 超时，timeout

NF: 因节点故障而运行失败，node_fail

作业状态转换的详细图如下，由于 CD, CA, F 这三个作业状态持续时间很短，

因此使用 `yhq` 命令可能会观察不到这些状态。



用户可以使用 `yhq` 查看自己提交的作业，为了保证用户的数据安全，普通用户通过 `yhq` 只能看到自己提交的作业。

查看作业明细：

用户可以通过如下命令来查看自己提交的作业明细

```
yhcontrol show jobs jobid
```

其中 `jobid` 表示作业的 id 号，用户根据自己作业的情况填入即可，之后用户即可以看到该作业十分详细的信息。

注意：用户作业如果长时间为 **CG** 状态，表示作业没有正常退出，系统管理员会定期扫描 **CG** 作业并处理，请用户耐心等待；用户作业如果变成 **S** 状态，表示系统管理员在维护系统，维护完成后会将用户作业恢复，对用户作业不会造成影响。

3.3 提交作业

目前 HPC1 系统部署的资源管理系统包括多种作业提交方式，包括批处理作业提交方式 `yhbatch` 和交互作业提交方式 `yhrun`。作业终止方式为 `yhcancel` 命令，需要获取作业的 `jobid`，可以通过 `yhq` 命令查看获得。

本手册，为了简化和方便用户，只对相关命令做简单介绍，用户如需更多参数选择，则可以通过响应命令后加入 `--help` 的方式，获取帮助信息，或查阅 SLURM 相关资料。

3.3.1 批处理作业 yhbatch

注意：

如果没有交互需求，请使用 **yhbatch** 提交任务。**yhbatch** 提交的作业终端关闭时不会受到影响，登陆结点 **down** 机时也不会受到影响，强烈推荐使用 **yhbatch** 提交任务。

yhbatch向资源管理系统提交一个批处理脚本，yhbatch将在脚本成功提交到资源管理系统控制进程并分配作业JobID后立即退出。

批处理脚本可能不会被立刻分配资源，而是在排队作业队列中等待，直到资源需求得到满足。当批处理脚本被分配资源后，资源管理系统将在所分配的第一个结点上运行批处理脚本。

yhbatch 运行的主要格式如下：

yhbatch [options] program

yhbatch 包括多个选项，用户最常使用的选项如下：

-n, --ntasks=ntasks
指定要运行的进程数。请求 yhrun 分配/加载 ntasks 个进程。省缺的情况是每个 CPU 核运行一个进程，但是-c 参数将改变此省缺值。
-N, --nodes=minnodes[-maxnodes]
请求为此作业至少分配 minnodes 个结点。调度器可能决定在多于 minnodes 个结点上启动作业。可以通过指定 maxnodes 限制最多分配的结点数（如 “--nodes=2-4”）。最少和最多结点数可以相同以便指定确切的结点数（如 “--nodes=2-2” 将请求两个并且仅仅两个结点）。如果没有指定-N，省缺的行为是分配足够的结点以满足-n 选项的要求。
-p, --partition=partition
从分区 partition 请求资源。如未指定，则省缺为默认分区。
-t, --time=minutes
设置作业的运行时间限制为 minutes 分钟。省缺值为分区的时间限制值。当到达时间限制时，作业的进程将被发送 SIGTERM 以及 SIGKILL 信号终止执行。完整格式为--time=days-hours:minutes:seconds，建议包机时用户使用该选



项。
-D, --chdir=path 加载的作业进程在执行前将工作目录改变到 <code>path</code> 。省缺情况下作业 <code>yhrun</code> 进程的当前工作目录。
-l, --label 在标准输出/标准错误的每行之前添加任务号。通常，远程任务的标准输出和标准错误通过行缓冲直接传递到 <code>yhrun</code> 的标准输出和标准错误。 <code>--label</code> 选项将在每行输出前面添加远程任务的 ID。
-J, --job-name=jobname 指定作业的名字。省缺值是可执行程序的名字 <code>program</code> 。
-W, --wait=seconds 指定在第一个任务退出后，到终止所有剩余任务之前的等待时间。0 表示无限等待（60 秒后将发出一个警告）。省缺值可由系统配置文件中的参数设置。此选项用于确保作业在一个或多个任务提前退出时能够及时终止。
-w, --nodelist=nodelist filename 请求指定列表中的结点。分配给作业的将至少包含这些结点。 <code>nodelist</code> 可以是逗号分割的结点列表或范围表达式（如 <code>cn[1-5,7,12]</code> ）。如果包含 “/” 字符，则 <code>nodelist</code> 将会被当作是一个文件名，其中包含了所请求的结点列表。
-x, --exclude=nodelist filename 排除指定列表中的结点。分配给作业的将不会包含这些结点。
--checkpoint-path=path 指定任务检查点映像文件的保存目录。省缺为任务的当前工作目录。
--checkpoint-period=number[h m] 指定对作业进行自动周期性检查点操作。如果 <code>number</code> 后没有跟时间单位，则默认为 h（小时）。
--restart-path=path 指定本次任务加载为从以前的检查点映像恢复执行。 <code>path</code> 为检查点映像文件所在的路径。
--exclusive 此作业不能与其它运行的作业共享结点，加入此选项，则表示用户需要针对此作业使用独占的处理器，如果没有足够的处理器，则作业的启动将会被推迟。

以上选项中，由以 **-N** , **-n**, **-p**, **-w**, **-x** 等选项最常用，**-N** 指定结点数，**-n** 指定进程数，**-p** 指定分区名，**-w** 指定结点列表，**-x** 指定不参加分配的结点列表（用于排除自己认为有问题的结点）。

用户在 **yhbatch** 的参数中指定资源分配的需求约束，编写的作业脚本中，也可以使用 **yhrun** 命令加载计算作业，此时 **yhrun** 通过环境变量感知已经分配了资源，从而直接创建作业而不再次提交作业。

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**, **rsh** , **csh** 等。

这种作业提交方式，适合提交绝大多数作业。如果需要连续执行多个任务的作业，用户可以在脚本中提交多个任务，逐个计算。

如前所述，系统中作业的运行分成**两步：资源分配与任务加载**。批处理作业使用 **yhbatch** 提交脚本的方式运行，**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。

3.3.1.1 MPI 并行作业

举例一：

假设用户可执行文件为 **a.out**，需使用 28 个进程并行计算，编写提交脚本 **sub.sh** 如下：

```
#!/bin/bash
yhrun -n 28 -p TH_HPC1 a.out
```

加载环境变量，并提交批处理命令：

```
module add Intel_compiler
module add MPI/Intel/IMPI
yhbatch -n 28 -p TH_HPC1 -J example1 ./sub.sh
```

计算过程中，脚本所在的工作目录中默认会生成以 **slurm** 开头的 **.out** 文件，屏幕输出的信息会保存到该文件中。

注意：

yhbatch 申请的资源必须不小于 **sub.sh** 脚本中 **yhrun** 申请的资源。

3.3.1.2 openMP 并行作业

openMP 支持共享式内存并行，因此单纯的 openMP 多线程并行程序只能在单计算结点上运行。由于每个计算结点是 28 个处理器核心数，因此最大线程数设置不能超过 28。

如果用户的程序支持该并行方式，若用户可执行文件为 **a.out**，需使用 28 个 openMP 多线程并行计算。编写提交脚本 **sub.sh** 如下：

```
#!/bin/bash
export OMP_NUM_THREADS=28
yhrun -N 1 -n 1 -c 28 -p TH_HPC1 a.out
```

提交批处理命令如下：

```
yhbatch -N 1 -n 1 -p TH_HPC1 ./sub.sh
```

3.3.1.3 MPI+openMP 并行作业

如果用户的程序支持该并行方式，若用户可执行文件为 **a.out**，需使用 8 个进程并行计算，每个进程下开启 14 个 openMP 线程，则应使用的计算结点数为 $8 \times 14 / 28 = 4$ 。编写提交脚本 **sub.sh** 如下：

```
#!/bin/bash
export OMP_NUM_THREADS=14
yhrun -N 4 -n 8 -c 14 -p TH_HPC1 a.out
```

加载环境变量，并提交批处理命令：

```
module add Intel_compiler
module add MPI/Intel/IMPI
yhbatch -N 4 -n 8 -p TH_HPC1 ./sub.sh
```


注意：HPC1 系统上的资源使用抢占式调度方式，即作业在结点上哪怕只运行了一个核的进程，其他作业也无法再分配到该结点上。

特别提示：

批处理作业提交模式，使用范围很广，由于手册篇幅限制，不能详述，如果您在提交批处理作业的过程中遇到了任何问题，请联系中心技术人员。

3.3.2 交互式作业提交 yhrun

对于交互式作业，资源分配与任务加载两步均通过 yhrun 命令进行：当在登录 shell 中执行 yhrun 命令时，yhrun 首先向系统提交作业请求并等待资源分配，然后在所分配的结点上加载作业任务。

yhrun 运行的主要格式如下：

yhrun [options] program

yhrun 包括多个选项，与 yhbatch 类似。

示例：

1) 在分区 TH_HPC1，使用两个结点上运行 hostname

```
$ yhrun -N 2 -n 28 -p TH_HPC1 hostname
yhrun: job 4385 queued and waiting for resources
yhrun: job 4385 has been allocated resources
cn4
cn4
...
cn5
```

特别注意：

1. yhrun 基本可以替代 mpirun，使用 1.3.2 章节推荐的系统自带的 mpi 实现编译的程序，完全可以使用 yhrun 提交任务，而不需使用 mpirun。

2. yhrun 为交互式作业提交方式，用户如需要和程序进行交互，则选择直接使用 yhrun 提交任务，**如果不需要交互，则需使用批处理作业提交方式。**

3. yhrun 提交的作业，如果没有进行输入输出的重定向，在关闭登陆客户端软件时，会导致任务中断，因此如无特殊需要，在直接使用 yhrun 提交任务时，重定向输入输出，并保留相应的 log 文件，方便遇到问题时，技术人员及时解决。

重定向举例如下：

```
yhrun -p TH_HPC1 -N 4 -n 112 ./a.out >log 2>&1 &
```

>为重定向符号，2>&1 表示标准错误输出重定向至标准输出，最后的&表示后台提交方式，这样保证了该任务在登陆客户端关闭时依然保持不中断。

4. 再次提示，如无特殊需要请使用批处理作业 `yhbatch` 提交方式，`yhbatch` 提交的作业终端关闭后不会受到影响。

3.3.3 应用软件作业提交举例

1. 应用软件 LAMMPS 使用

1) 在登陆节点命令行下加载 LAMMPS 所需环境变量：

```
$ module add Intel_compiler/16.0.3
$ module add MPI/Intel/IMPI/5.1.3.210
$ module add MKL/16.0.3
$ module add lammmps/17Nov16-icc16-IMPI5.1
```

说明：

从 lammmps 的版本名称 17Nov16-icc16-IMPI5.1 可以看出：

- 它的版本号是 17Nov16，即 2016-11-17 发布的版本。用户可以依据需求更换其他版本。
- 它所使用的编译器为 intel compiler 2016 和 intel mpi 5.1，所以需要加载相关的 module 环境。
- 它编译的时候使用到了 intel mkl 数学库来加速计算，故需要加载 MKL 的 module 环境。

2) 编写任务脚本 run.sh 如下：

```
#!/bin/sh
yhrun -N 2 -n 56 lmp < in.test
```

说明：

- 第一行：它是一个用/bin/sh 来解析的脚本文件。
- 第二行：-N 2 表示 2 个节点；-n 56 表示共 56 cpu 核；lmp 是可执行程序的名字；in.test 是输入文件名。

3) 登陆节点命令行下提交任务：

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

2. 应用软件 GROMACS 使用

- 1) 在登陆节点命令行下加载 GROMACS 所需环境变量:

```
$ module add Intel_compiler/16.0.3
$ module add MPI/Intel/IMPI/5.1.3.210
$ module add fftw/3.3.4-icc16-IMPI5.1
$ module add gromacs/5.1.2-single-icc16-IMPI5.1
```

- 2) 编写任务脚本 run.sh 如下:

```
#!/bin/sh
gmh_mpi grompp -f rf_verlet.mdp
yhrun -N2 -n56 gmh_mpi mdrun -ntomp 1
```

说明:

- 第二行: 用 gmh_mpi grompp 进行前期处理。
- 第三行: 用 gmh_mpi mdrun 来计算, -ntomp 1 表示每个 mpi 进程启用一个 openmp 线程。
- 用户根据自己的需求将相关的 gmh 处理命令写入 run.sh 脚本即可。

- 3) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

3. 应用软件 NAMD 使用

- 1) 在登陆节点命令行下加载 NAMD 所需环境变量:

```
$ module add Intel_compiler
$ module add MPI/Intel/IMPI
$ module add namd/2.12-icc16-IMPI5.1
```

- 2) 编写任务脚本 run.sh 如下:

```
#!/bin/sh
yhrun -N2 -n56 namd2
```

- 3) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

4. 应用软件 **espresso** 使用

- 1) 在登陆节点命令行下加载 espresso 所需环境变量:

```
$ module add Intel_compiler  
$ module add MPI/Intel/IMPI  
$ module add espresso/5.4.0-icc16-IMPI5.1
```

- 2) 编写任务脚本 run.sh 如下:

```
#!/bin/sh  
yhrun -N2 -n56 pw.x < test.in > test.out # 以 pw.x 为例
```

- 3) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

5. 应用软件 **Siesta** 使用

- 1) 在登陆节点命令行下加载 Siesta 所需环境变量:

```
$ module add Intel_compiler  
$ module add MPI/mvapich2-2.2/intel2016u3  
$ module add siesta/4.0-icc16-mvapich2
```

- 2) 编写任务脚本 run.sh 如下:

```
#!/bin/sh  
yhrun -N2 -n56 siesta
```

- 3) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

6. 应用软件 CP2K 使用

- 1) 在登陆节点命令行下加载 CP2K 所需环境变量:

```
$ module add Intel_compiler
$ module add MPI/Intel/IMPI
$ module add cp2k/4.1-icc16-IMPI5.1
```

- 2) 编写任务脚本 run.sh 如下:

```
#!/bin/sh
yhrun -N2 -n56 cp2k.popt input.inp # input.inp 为输入文件名, 如 H2O-64.inp
```

- 3) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

7. 应用软件 WRF 使用

- 1) 在登陆节点命令行下加载 WRF 所需环境变量:

```
$ module add Intel_compiler
$ module add MPI/mvapich2-2.2/intel2016u3
$ module add netcdf/4.4-icc16-mvapich2
$ module add wrf/3.8-emreal-dmpar-icc16-mvapich2
```

- 2) 使用 module help 命令可以得到 wrf 的相关信息

```
$ module help wrf/3.8-emreal-dmpar-icc16-mvapich2
```

- 3) 将 wrf 文件夹下的 run 目录拷贝到用户的目录下:

```
$ cp /THL6/software/wrf/3.8-emreal-dmpar-icc16-mvapich2/run .
```

- 4) 依据用户需求修改 namelist.input 及相关配置文件

- 5) 编写任务脚本 run.sh 如下:

```
#!/bin/sh  
yhrun -N2 -n56 wrf.exe
```

6) 登陆节点命令行下提交任务:

```
$ yhbatch -N 2 -n 56 -p TH_HPC1 run.sh
```

3.3.4 其他典型脚本举例

如果有单节点内需并行运行多个任务的需求, 可参考如下脚本 `test.sh`, 该脚本实现在一个节点内并行提交并运行 28 个作业。`test.sh` 内容如下:

```
#!/bin/bash  
# can be changed by user as need  
EXE=./a.out  
numJobsPerNode=28  
partition=TH_HPC1  
myOutFile=slurm-${SLURM_JOB_ID}.out  
#  
  
# Do NOT change  
loop=`expr ${numJobsPerNode} - 1`  
for y in `seq 0 $loop`  
do  
    num=`expr $y + 1`  
    time yhrun -N 1 -n 1 -p $partition $EXE > run${num}.log 2>&1 &  
done  
#  
wait
```

在命令行下提交该脚本的命令为 `yhbatch -N 1 -p TH_HPC1 test.sh`

3.4 任务取消 yhcancel

yhcancel 取消用户运行的任务，命令为 `yhcancel jobid`。

yhcancel 命令强制取消任务后，`slurm-jobid.out` 文件中显示的信息如图 3-1 所示：

```
yhrun: Force Terminated job 118230
slurmd[cn0]: *** STEP 118230.0 CANCELLED AT 2010-07-20T15:42:11 ***
yhrun: error: cn6: tasks 48-55: Terminated
yhrun: error: cn4: tasks 32-39: Terminated
yhrun: error: cn7: tasks 56-63: Terminated
yhrun: error: cn3: tasks 24-31: Terminated
yhrun: error: cn1: tasks 8-15: Terminated
yhrun: error: cn0: tasks 0-7: Terminated
yhrun: error: cn5: tasks 40-47: Terminated
yhrun: error: cn2: tasks 16-23: Terminated
yhrun: XXX: job done
```

图 3-1 任务取消后显示信息

4 常见问题

4.1 VPN 登陆问题

Q: Windows 系统登陆 VPN 无法加载插件或报错

A: 按照用户手册确认是否打开了浏览器的允许加载插件选项，并确保您机器的安全软件、杀毒软件等不会阻止插件的正常运行。

Q: 登录 VPN 提示“超出人数限制”

A: VPN 用户默认最多允许 5 个人同时登陆，超过这个限制就会报错。但如果用户登录 VPN 后没有点击“退出”按钮正常退出，而是直接关掉页面，则会保持该次连接会话直到超时，此时也会占用一个登录连接数。

Q: Linux 系统登陆 VPN 不成功

A: linux 用户登录 VPN 需要使用 root 账户，并且同时关闭自己机器的 sshd 服务，常用版本的关闭方法为“/etc/init.d/sshd stop”。具体请参考用户手册或者 VPN 登陆下面的“SSL VPN 设置向导”。linux 系统的登陆与 windows 大致相同，但其需要 java 的支持。

Q: 在网页（<http://www.nscg-tj.gov.cn/>）下 VPN 登陆天河一号服务器，提示“本地用户有效期已经过期”

A: 是由于 VPN 已经到期，请联系与您联系的相关工程师，申请开通 VPN。

4.2 系统登陆问题

Q: 登陆结点 home 目录下看不到原有的用户文件

A: 这是由于登录结点启动后还没有挂载相应的共享存储，请用户先退出系统，稍等待后再重新进行登录。

Q: 在登陆结点执行“ls; cd”这样的基本命令会卡死

A: 此时用户可以尝试更换其他登陆结点，如果仍然有问题请联系系统管理员。原因之一可能是登录结点负载比较重，造成使用终端命令受到影响；原因之二可能是用户客户端的网络负载比较重，出现比较严重的网络延迟；原因之三可能是系统

的存储正在进行恢复调整。

Q: 登陆结点无法连通

A: 这有可能是用户在登陆结点上运行非法程序导致结点宕机，我们会实时对系统进行监控，出现这种情况请用户更换其他登陆结点。建议用户不要在登陆结点上运行任何计算，一旦查到并影响到其他人的使用，则会进行警告，屡次不改者可能会被封号。

4.3 编译问题

Q: 使用 mpif90 编译并程序，提示说 command not found

A: 原因为用户未设置 mpi 环境或设置错误。可参见 1.3.2 章节中的设置方式，使用 modules 方式加载。

Q: 常见的“undefined reference to”问题解决办法

A: 1) 目标文件缺失：当进行可执行程序链接时，链接命令中找不到某个函数所在源代码的目标文件`***.o`，出现“undefined reference to ***”错误。

解决办法：找到需要的目标文件`***.o`，然后将其添加到链接命令中。

2) 库文件缺失：当进行可执行程序链接时，链接命令中找不到某个库函数所在的库文件，出现“undefined reference to ***”错误。

解决办法：找到需要的库文件，然后将其绝对路径使用`-L` 和`-l` 添加到链接命令中，请务必确保该路径是计算结点可访问的共享目录。

3) 多个库文件连接顺序：当进行可执行程序链接时，所需目标文件和库文件均齐全，但由于链接顺序问题，出现库文件依赖报错，从而同样会出现“undefined reference to ***”错误。

解决办法：在链接命令中给出所依赖的库时，需要注意库之间的依赖顺序，依赖其他库的库一定要放到被依赖库的前面。

4) C++代码链接C库文件：若库文件或目标文件由c代码生成的，则在c++代码中链接目标函数时，也会碰到 undefined reference 的问题。

解决办法：由于c++代码调用c语言库的函数，链接目标文件时候找不到，出现“undefined reference to ***”错误。只需在调用C语言库函数的源码文件中，把

与 c 语言库相关的头文件包含添加一个 extern "C"的声明即可。

Q: 在链接 MKL 的支持单线程的库时，报错：undefined reference to dgemm_***

A: 原因为 MKL 库没有被有效链接。需先加载 Intel 编译器和 MKL 环境变量，下面为一个使用范例：

```
$ module add Intel_compiler
$ module add MKL
$ cat demo.sh
#!/bin/bash
mpif90 -c ./Main/exe.f90
mpif90 -o exe exe.o \
    -lmkl_intel_lp64 -lmkl_core -lmkl_sequential -lpthread -lm -ldl \
    -Wl,-rpath, -lmkl_intel_lp64 -lmkl_core -lmkl_sequential -lpthread -lm -ldl
```

注：关于具体使用 MKL 的编译、链接选项具体可参考 <https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

Q: 在使用 MKL 的支持多线程的库时，报错：libmkl_intel_thread.so: undefined reference to `__kmpc_ok_to_fork'

A: 原因为在使用 MKL 的多线程库时，没有有效使用-openmp 选项。下面为一个范例脚本：

```
$ module add Intel_compiler
$ module add MKL
$ cat demo.sh
#!/bin/bash
mpif90 -c -openmp ./Main/exe.f90
mpif90 -o exe -openmp exe.o \
    -lmkl_intel_lp64 -lmkl_core -lmkl_intel_thread -lpthread -lm -ldl \
    -Wl,-rpath, -lmkl_intel_lp64 -lmkl_core -lmkl_intel_thread -lpthread -lm -ldl
```

注：如果在编译和链接阶段不加入-openmp 选项，就会包相关多线程的函数无法找到。关于具体使用 MKL 的编译、链接选项具体可参考 <https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

Q: 在编译程序时，报错：undefined reference to drandgaussian

A: 原因为 ACML 库没有被有效链接。drandgaussian、drandgamma 等均为

AMD ACML 下的库函数，用户可自行下载和安装对应编译器下的 ACML 库。下面为一个范例脚本：

```
> cat demo.sh
#!/bin/bash
ACML_DIR=/yourPath/acml/5.2.0
gfortran -c ./test.f90 -I${ACML_DIR}/include
gfortran -o a.out test.o \
    -L${ACML_DIR}/lib -lacml_mp \
    -Wl,-rpath,${ACML_DIR}/lib,-L${ACML_DIR}/lib -lacml_mp
```

4.4 作业运行问题

Q：为什么我用 **yhi** 命令看到很多 **idle** 的结点，但是我提交作业后，作业不能立即执行

A：HPC1 系统的作业调度原则是先来先服务，因为已经有用户优先于您提交作业，并且已经在排队，所以 **idle** 的结点需要优先分配给优先提交作业的用户，即使用户提交作业需求结点数小于空余结点数，仍然因为先来先服务原则，处于等待状态。

Q：我用 **yhrun** 提交完作业以后，网络或终端断掉后，我的作业也相应停止

A：请您使用 **yhbatch** 命令提交作业，这样您就不会出现网络或终端断掉后作业停止的现象。

Q：请问你们的系统上的内存最大是多少 G？

A：HPC1 系统的每个计算结点上的内存是 125G，如果您的并行程序需要比 125G 大的内存，您可以通过使用更多的结点来分散您的内存的使用。

Q：作业在某一个时间点后无输出

A：导致这种现象的原因很多，需要具体问题具体分析，其中一种可能是由于用户磁盘配额已满，无法写入数据造成的，因此需要用户及时清理自己的数据，如果并非这种情况请您邮件或者电话与我们的系统管理人员取得联系，我们将进行进一步查看。

Q：作业断开，**slurm** 日志中出现“DUE TO TIME LIMIT”报错信息

A：这是因为作业运行时间超过队列最大运行时间限制，请注意您所在队列的

运行时间限制以及您作业已运行时间。命令行下输入 `yhi`，输出结果的第三列 `TIMELIMIT` 即为不同分区下的最大作业运行限制。请您经常查看作业运行状态，以确保您的作业在正常运行。

Q：作业断开，slurm 日志中出现“Not enough endpoint resources”报错信息

A：这是由于结点有残留进程，结点的 `endpoint` 没有正常释放。用户提交可以加 `-x` 剔除问题结点，然后联系管理员进行解决。

Q：运行作业报错“slurmd[cn4234]: Cannot send after transport endpoint shutdown”

A：结点异常。请重新提交作业，并同时使用 `-x cn4234` 把报错结点过滤掉。

Q：作业断开，slurm 日志中出现“Group ID not found on host”报错信息

A：这是由于计算结点的 `passwd` 和 `group` 没有与管理结点同步导致。用户提交可以加 `-x` 剔除问题结点，然后联系管理员进行解决。

Q：作业断开，slurm 日志中出现“couldn't chdir to `/vol-th/home/your_username`: No such file or directory: going to /tmp instead”报错信息

A：这是由于计算结点没有挂载用户所在的共享存储。用户提交可以加 `-x` 剔除问题结点，然后联系管理员进行解决。

Q：作业断开，slurm 日志中出现“yhrun: error: Task launch for 2440965.0 failed on node cn2892: Job credential expired”报错信息

A：这是由于计算结点时间没有与管理结点同步。用户提交可以加 `-x` 剔除问题结点，然后联系管理员进行解决。

Q：作业断开，slurm 日志中出现“bus error”报错信息

A：导致“bus error”的报错原因很多，具体问题需要使用工具排查。用户提交可以加 `-x` 剔除问题结点，然后联系管理员进行解决。

Q：运行作业报错“fortrtl: severe (41): insufficient virtual memory”

A：运行作业的内存不足，请尝试多使用结点，每个结点上少使用核数来提交运行。

Q：运行作业提示“error while loading shared libraries: libXXX.so: cannot open shared object file: No such file or directory”



A: 需要用户将动态链接库的路径添加到自己运行的环境变量中，假设缺少 `x` 库，先“locate `x`”找到该链接库的地址 `$DIR`，请确保 `$DIR` 为共享目录！然后将所需库对应的软件通过 `module add` 方式加载。

在计算时找不到动态库是因为计算结点和登陆结点的软件环境有所不同。链接器在处理动态库时将链接时路径（Link-time path）和运行时路径（Run-time path）分开，`-L` 只是指定了程序链接时库的路径，并不影响程序执行时库的路径；`-Wl,-rpath` 指定程序运行时库的路径，该库的路径信息保存在可执行文件中，运行时它会直接到该路径查找库；也可使用 `LD_LIBRARY_PATH` 环境变量来指定动态库在运行时的搜索路径。

Q: 提交的作业总是被自动退出

A: 您如果是使用 `yhrun` 提交任务，那么终端关闭、脚本终止都会导致任务被杀掉。建议用户使用 `yhbatch` 的提交方式，`yhbatch` 提交的任务，终端关闭甚至登陆结点宕机都不会对已提交的作业有影响。另外，还有可能是您提交的作业所分配的计算结点有问题导致自动退出，请您仔细查看产生的日志文件的报错信息，是否属于以上问题中的一种，并采取相应的处理。

Q: 查看有可用结点，但作业却一直处于 PD 状态

A: HPC1 系统的资源管理器采用“先进先出”的作业调度方式，作业处于 PD 状态说明在用户前面有其他用户先提交了作业，并且之前的用户作业超出了目前的可用资源总数，请用户耐心等待。根据用户资源需求，系统管理人员也会定期进行资源调整，降低作业排队时间。

Q: 作业运行的状态为 CG，自己也杀不掉这个作业，是否影响我的使用，这个问题如何解决

A: CG 状态是作业正在退出，用户如果杀不掉这些 CG 状态的作业也没关系，它并不影响您的使用，管理员会集中定期处理这个 CG 状态的作业。

Q: 任务突然运行速度变慢

A: 可能是该任务使用的计算结点内存报错，可换其他结点重新提交任务。

Q: 在计算结点上运行程序，找不到某些命令，比如说提示 `bc: Command not found`



A: 复制登录结点上的 bc 命令到自己账户下，设置好该命令的环境变量后，重新运行就可以找到命令。

Q: 提交作业后，提示 “yhbatch: error: Batch job submission failed: User's group not permitted to use this partition”

A: 用户没有权限使用提交作业时-p 参数后面指定的队列，请使用 yhi 命令检查您可以使用的队列。

Q: PBS 作业系统里查看运行的结点名称的变量 \$PBS_NODELINE，在 HPC1 系统里对应哪一个变量

A: \$SLURM_NPROCS，它与 PBS 的 \$PBS_NODELINE 是一样的功能。

4.5 存储问题

Q: 作业运行提示“forrtl: Input/output error”

A: 可能是存储某一时刻压力较大，造成 IO 错误，请您重新提交作业。

Q: 作业运行时报错：forrtl: No space left on device, forrtl: severe (38): error during write, unit 12，但是同样的作业再次提交时可能就正常运行完成。

A: 该问题主要由文件系统中某一 OST 存储已满导致，请联系与您对接的工程师或系统管理员。

Lustre 文件系统由若干 IO 服务器（Object Storage Services）和 Object Storage Targets(OST)组成。当对一个文件进行读写操作时，为了提高 IO 效率，文件系统会自动将该文件的读写操作分割成多个，在多个 OST 上并发实现。如果在该过程中，使用到的某一 OST 出现问题，就会发生读写错误。

5 技术支持

由于用户手册篇幅有限，只列出了用户使用系统的基本方法以及常见问题和解决方法，很难面面俱到，还请您能够谅解。如果您在系统使用过程中遇到任何问题，都可以及时与中心技术人员取得联系。中心技术人员会在收到用户问题反馈后的 24 小时工作时间内给予回复。

1. 合同、资源申请使用、应用软件相关问题联系方式：

Email: service@nscg-tj.gov.cn

电话：022-65375561

2. 系统使用、作业运行相关问题联系方式：

Email: support@nscg-tj.gov.cn

电话：022-65375560，18302248223

重点提示：为了能尽快使您的问题得到定位解决，请您在通过邮件或电话联系中心技术人员时提供以下基本信息（建议拷贝表格填写）：

系统用户名	作业 jobid	作业日志路径	作业提交命令	错误现象

包括：系统用户名，出错作业号（jobid），出错作业日志路径，作业提交命令，错误现象描述等信息。

在此，特别感谢您对我们工作的信任与支持，同时也祝您在天河系统使用过程中工作愉快！希望我们能够共同携手推动我国并行计算技术的发展，合理使用并行计算资源在各个科研及工业领域不断创新突破！

附录 A 常用 Unix 命令

A1 基本命令

date: 显示日期和当前时间，命令格式：\$date。
who: 查询当前登录在系统中的用户信息，命令格式：\$who。
w: 查询当前登录在系统中的用户行为，命令格式：\$w。
write: 将消息直接发送到另一个用户的终端上，命令格式：
 \$write username
 Hello: We have a meeting at Room 412.
 键入 Ctrl-D 结束输入消息，在 username 用户终端上可以看到上述信息。
mesg: 选择是拒绝还是接受由 write 发来的消息，命令格式：
 \$mesg n 拒绝由 write 发来的消息；
 \$mesg y 允许别的用户发送消息；
 \$mesg 报告当前是否允许别的用户向你的终端发送消息。
ps: 用于查看当前系统中的活跃进程，命令格式：\$ps [options]。
kill: 终止指定进程，命令格式：\$kill [-signal] pid。

A2 目录操作

mkdir: 创建目录，命令格式：\$mkdir directory ...。
rmdir: 删除目录，命令格式：\$rmdir directory ...。。。
pwd: 显示当前工作目录，命令格式：\$pwd。
ls: 显示目录内容，命令格式：\$ls [options] [names]，选项可合用。
cd: 改变工作目录，命令格式：\$cd [directory]。

A3 文件创建、复制与删除

touch: 创建内容为空的文件，命令格式：\$touch 文件名。
rm: 删除文件或目录，命令格式：\$rm [-r] [-f] [-i] file ...。
cp: 复制文件或目录，命令格式：\$cp [-i] [-r] file1 [file2...] target
 -r。 如果 file 为目录，则 cp 将复制该目录及其所有文件。
mv: 文件的搬移或更名，命令格式：\$mv file1 target。

A4 文件属性

chmod: 改变文件的读、写或执行权限，命令格式：
 \$chmod [who] operator [permission] file-list。
chown: 改变文件的属主，命令格式：\$chown [-R] [-h] owner file...。



chgrp: 改变文件的组主，命令格式: `$chgrp [-R] [-h] group file...`。

A5 文件显示与连接

cat: 用于显示文件或文件连接，命令格式:

`$cat file1 file2` 显示 `file1` 和 `file2` 的内容;

`$cat file1 file2 > file3` 将 `file1` 和 `file2` 合并成 `file3`。

more: 显示文件的内容，命令格式: `$more 文件名`。

head: 显示文件的前几行，命令格式为: `$head [-n] [file...]`。

tail: 将文件从指定位置开始的内容全部显示到屏幕上:

`$tail [+n] [lbc] file` 从文件头加上 `n` 处开始显示;

`$tail [-n] [lbc] file` 从文件尾减去 `n` 处开始显示;

`$tail -f file` 间隔 1 秒循环显示文件新内容。

ln: 建立指定文件的硬链接或符号链接，命令格式:

`$ln [-s] [-f] [-n] file target`。

A6 文件查找与比较

grep: 查找字符串，命令格式: `$grep pattern files`。

find: 从指定目录开始，递归地从子目录寻找匹配文件，命令格式:

`$find dirname option-list`。

diff: 比较两个文本文件的差异，命令格式: `diff [options] file1 file2`。

A7 文件压缩与备份

compress: 进行文件压缩，命令格式: `$compress [-cfv] filename`。

uncompress: 解压缩文件，命令格式: `$uncompress [-cfv] filename`。

tar: 用于建立磁带档案（文件系统的备份），或存到档案媒介或从档案媒介中读取文件，命令格式: `$tar c|t|x [bvf] [tarfile] [bsize] [file_list]`。

A8 输入输出重定向

`<`: 输入改向，命令格式: `$command < file`。

`>`: 输出改向，更新指定文件内容，命令格式: `$command > file`。

`>>`: 输出改向，将执行结果接到指定文件内容后面，命令格式:

`$command >> file`。

附录 B 常用 vi 命令

B1 进入与退出 vi

进入 vi 命令格式：\$ vi filename

vi 中的退出命令有以下几种：

- :q 退出。当文件已被修改，vi 将在屏幕的底行显示提示信息。
- :q! 强行退出。
- :w 回写文件但不退出。
- :wq 回写文件并退出。
- :x 与 wq 相同，回写文件并退出。

B2 移动光标

- ↑或 k 键 把当前光标向上移动一行，保持光标的列位置。
- ↓或 j 键 把当前光标向下移动一行，保持光标的列位置。
- 或 l 键 把当前光标向右移动一个字符。
- ←或 h 键 把当前光标向左移动一个字符。
- \$键 把当前光标移动到该行行末。
- ^键 把当前光标移动到该行行首。
- w 键 把当前光标移动到该行的下一个字的首字符上。
- b 键 把当前光标移动到该行的上一个字的首字符上。
- e 键 把当前光标移动到该行的该字的末尾字符上。
- ^F 向前滚动一整屏正文。
- ^D 向下滚动半个屏正文。
- ^B 向后滚动一整屏正文。
- ^U 向上滚动半个屏正文。

在用 k、j、l、h 四个键时，可以在它们的前面加一个数字，这样在需要多次移动光标时不必多次按移动命令键。

B3 正文输入、删除、替换、恢复和查找命令

- a 在光标的后面开始插入正文。
- A 在光标所在行的行首插入正文。
- I 在光标的前面开始插入正文。
- I 在光标所在行的行末插入正文。
- o 在光标所在行（当前行）的下一行的行首开始插入正文。
- O 在光标所在行（当前行）的上一行的行首开始插入正文。
- Esc 退出输入方式。
- Backspace 输入方式下删除字符。

x	删除当前光标所在的字符。
nx	删除从当前光标开始的 n 个字符，n 为要删除的字符数。
dw	删除当前光标所在的字。
ndw	删除从当前光标开始的 n 个字，n 为要删除的字数。
dd	删除当前光标所在行。
ndd	删除从当前光标开始的 n 行，n 为要删除的行数。
rx	用 x 替代当前光标所在的字符。
nrx	用 x 替换 n 个字符，在替换完第 n 个字符后该命令自动停止。
u	废除最近的命令，恢复被修改或删除的内容。
U	把当前行恢复到修改它之前的状态。
/pattern	在缓冲区中向下查找指定的字符串 pattern。
?pattern	在缓冲区中向上查找指定的字符串 pattern。
n	重复上一次查找命令。
N	以相反的查找方向重复上一次查找命令。

B4 行编辑命令

键入“:”，并在屏幕底部的“:”号提示符下输入行编辑命令。

: set nu	显示正文的行号。
: set nonu	取消行号。
: 1, \$p	显示缓冲区的整个内容。
: r wqb	将文件 wqb 中的内容读入缓冲区，插入当前光标下。
: 1, 5w clh	将正文中 1 到 5 行的内容写到名为 clh 的文件中去。
: 2, 5d	将正文中 2 到 5 行删除。
: 2, 5t8	将正文中 2 到 5 行复制到第 8 行的后面。
: ! ls	暂时转出 vi 编辑器，执行 shell 命令 ls。
: sh	暂时转出 vi 编辑器，执行 shell 命令，键入 ^D 或 exit 返回正文。