

高性能计算平台用户手册

南京航空航天大学高性能计算中心

2023 年 2 月

目 录

1	平台资源概述.....	1
1.1	应用软件资源.....	1
2	使用流程及规范.....	2
2.1	使用流程	2
2.2	使用规范	5
3	常用科学计算软件使用	6
3.1	GNU 编译器套件	6
3.2	Gromacs	7
3.3	vasp.....	9
	附录 1 Linux 常用命令.....	11
	附录 2 常用脚本.....	12
	附录 3 modulefile（使用场景）	14

1 平台资源概述

南京航空航天大学高性能计算平台（以下简称平台）是依托工信部十三五信息化专项--南京航空航天大学智慧校园建设的校级科学仪器公共平台，平台包含了 CPU0 队列（部署于云平台）、CPU1 队列（部署于平台一期）、CPU2 队列（部署于平台二期）、GPU 队列（部署于平台一期）、GPU 人工智能队列（部署于 HPC-AI 平台）五部分，可提供 CPU 计算和 GPU 计算所需资源。

1.1 应用软件资源

CPU0 队列、CPU1 队列、CPU2 队列普通节点上已安装常用应用软件，CPU2 队列 Windows 节点需联系管理员配置账户并安装所需软件。

附录 1 提供了常用的 Linux 系统命令，用户可在命令行界面登陆成功后参考并操作。

附录 2 提供 vasp、lammps、fluent、lsdyna 脚本样本，用户可以直接复制使用脚本样本。

附录 3 提供了 module 常见的指令和几个 module 的使用案例，并演示了如何编写 modulefile 来管理自己的软件环境，方便加载应用软件所需的环境变量。

针对需要使用 python 的用户特别说明：集群无法连接公网下载第三方模块，集群提供了 anaconda3 环境供用户使用，可使用 module load 命令加载 anaconda3 的环境。如 anaconda3 中缺少第三方模块，请联系管理员协助安装所需的第三方模块。

2 使用流程及规范

2.1 使用流程

1) 用户开户：首次使用平台的用户需通过网上办事大厅“高性能计算平台用户申请”流程提交开户申请，经相关责任部门审核处理后完成开户。学生用户需绑定教师账户后方可使用。



图 1 用户平台申请

用户可根据使用需要选择不同队列，需输入个人邮箱用于接收开户信息和通知信息，点击提交按钮开始申请流程。

高性能计算平台用户申请			
申请人信息			
姓名	<input type="text"/>	单位	信息化处（信息化技术中心） * 此项必须填写。
学/工号	<input type="text"/>	邮箱	<input type="text"/>
固定电话	<input type="text"/>	移动电话	<input type="text"/>
是否为项目负责人		<input checked="" type="radio"/> 是 <input type="radio"/> 否	
项目信息			
项目名称	<input type="text"/>	项目编号	<input type="text"/>
所需软件及版本	<input type="text"/>		
项目简介	<input type="text"/>		
集群类型			
开户集群	<input type="checkbox"/> CPU0 集群 <input type="checkbox"/> CPU1 集群 <input type="checkbox"/> CPU2 集群 <input type="checkbox"/> GPU 集群 <input type="checkbox"/> GPU 人工智能集群(仅适用于深度学习方向)		
<input type="checkbox"/> 本人同意遵守《南京航空航天大学高性能计算平台管理条例（试行）》			
<input type="checkbox"/> 本人承诺，计算内容不涉及国家秘密信息，如有违反，愿承担相应责任。			

图 2 申请页面

2) 签署协议：教师用户通过网上办事大厅“高性能计算平台技术服务协议”流程与信息化处签署技术服务协议，约定使用模式、计算资源限额和服务内容。已绑定教师的学生用户无需签署技术服务协议，由绑定的教师用户负责约定服务相关内容。

3) 使用平台：用户签署平台服务协议后，通过信息化处主页查阅《高性能计算平台使用手册》，登录平台确定软硬件环境，提交计算作业。学生用户使用记录关联到教师账户。**校外用户登录需要先登录校园 VPN。**

高性能计算平台鼓励用户在发表依托平台产出的成果时标注平台信息，在正文或致谢中注明“研究工作得到南京航空航天大学高性能计算平台支持”（**This work is partially supported by High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics**）。

4) 缴费：计算任务完成后，用户应根据使用情况缴纳计算费用。高性能计算中心定期向未完成缴费的用户发送缴费提醒，并暂停未及时缴费用户的平台使用权限。基本流程为：**a)系统登录—>b)使用确认—>c)费用缴纳**。详情请参考计算中心主页的仪器设备使用预约及参考操作手册。

a)系统登录：首先用户需要登录南京航空航天大学大型仪器设备共享管理平台，校内用户通过统一身份认证登录，如图 3 所示：



图 3 统一身份认证登录大仪平台

校外用户进入大仪平台官网后点击左上角“校外”按钮，完成注册后输入账户密码登录，如图 4 所示：



图 4 校外用户大仪入口

b)使用确认：在“结算中心”模块中点击“个人消费记录”，进入个人消费记录页面。用户核对使用及消费信息，确认无误即点击“确认”按钮。



图 5 使用确认

c)费用缴纳：仪器管理员生成结算单并提交财务后，校内用户可在“结算中心--待缴费用”模块，打印结算单。结算单由财务账号负责人签字并加盖印鉴后，提交至高性能计算中心（本部提交至综合楼 710，江宁提交至 1 号楼 8 楼办公室），由高性能计算中心汇总至国资处实验室管理科（本部综合楼 326），并统一提交财务处结算。



图 6 校内用户费用缴纳

校外用户可在“结算中心--待缴费用”模块，查看待缴费情况，并可通过扫描缴费和对公汇款进行费用结算。

(1) 扫描缴费：点击缴费按钮，弹出缴费二维码，扫码即可进行缴费。

(2) 对公汇款：

账户名称：南京航空航天大学

开户银行：交通银行南京御道街支行

银行账号：320006639010149000354

交行行号：301301000394

5) 销户：若不再需要继续使用高性能计算平台资源，可主动联系管理员进行注销账户。对连续两年未使用平台的用户，高性能计算中心将自动终止协议、销户并删除用户存放在计算平台上的所有信息和数据。

2.2 使用规范

1) 用户应遵守《中华人民共和国保守国家秘密法》等国家保密相关法律法规，禁止使用平台处理涉密数据或信息。

2) 用户应遵守《中华人民共和国网络安全法》等网络安全相关法律法规，禁止使用平台从事违法犯罪活动。

3) 用户使用平台时，一律通过安装在相应集群上的作业管理系统提交作业，进行计算和获取结果，不可绕过作业管理系统使用平台资源。

4) 不得擅自转让、出借账号，将口令随意告诉他人；也不得借用他人账号使用计算资源。

5) 不得窃取他人账号或口令，非法入侵他人账号，阅读他人文件，窃取他人计算和研究成果或受法律保护的资源。

6) 用户不得利用平台制造和传播计算机病毒；禁止破坏数据、破坏程序或其他恶意行为。

7) 用户应高度重视作业和结果的安全，及时收取计算结果，防止数据损坏、丢失或泄露。

8) 增强自我保护意识，及时反映违反用户行为规范的人和事。平台有权停止账号的使用权限，并对违规用户做出相应的处罚。

9) 用户有义务及时反馈平台存在的问题，有义务积极配合平台进行的统计、测试等各项工作。

3 常用科学计算软件使用

3.1 GNU 编译器套件

GCC (GNU Compiler Collection, GNU 编译器套件) 是由 GNU 开发的编程语言译器。GNU 编译器套件包括 C、C++、Objective-C、Fortran、Java、Ada 和 Go 语言前端, 也包括了这些语言的库 (如 libstdc++, libgcj 等)。

安装 GCC 操作方法如下 (因计算平台是内网环境, 无法连接互联网, 可以通过下载安装包后按照使用手册文件上传下载内容的方法进行上传, 再进行安装操作):

```
# 获取源码
export VERSION=9.3.0
wget https://mirrors.aliyun.com/gnu/gcc/gcc-${VERSION}/gcc-${VERSION}.tar.gz

# 解压
tar xf gcc-${VERSION}.tar.gz
cd gcc-${VERSION}

# 编译安装
./contrib/download_prerequisites
# 如果没有办法下载依赖, 可以尝试修改./contrib/download_prerequisites 文件
# 把 base_url='ftp://gcc.gnu.org/pub/gcc/infrastructure/' 改成
# base_url='http://gcc.gnu.org/pub/gcc/infrastructure/'
# 定义安装路径
export PREFIX=/gpfs/share/software/gcc/9.3.0
./configure --prefix=${PREFIX} --enable-languages=c,c++,fortran --build=x86_64-linux --
disable-multilib
make
make install
```

环境变量设置如下:

```
PATH
CPATH
LIBRARY_PATH
LD_LIBRARY_PATH
```

GCC 使用方法如下: 平台上安装了多个版本的 gcc, 可以直接通过 module 加载使用; 加载 gcc 编译器代码如下:

```
module load gcc/7.3.0
```

测试代码 test.c

```
#include <stdio.h>
int main(void)
```



```
{  
    printf("Hello world!\n");  
    return 0;  
}
```

使用 gcc 编译器编译连接

```
gcc test.c -o test
```

运行编译好的程序

```
./test
```

3.2 Gromacs

Gromacs 是用于研究生物分子体系的分子动力学程序包。它可以用分子动力学、随机动力学或者路径积分方法模拟溶液或晶体中的任意分子，进行分子能量的最小化，分析构象等。同时它是一个功能强大的分子动力学的模拟软件，其在模拟大量分子系统的牛顿运动方面具有极大的优势。它的模拟程序包包含 Gromacs 力场(蛋白质、核苷酸、糖等)，研究的范围可以包括玻璃和液晶、到聚合物、晶体和生物分子溶液。

Gromacs CPU 版本安装方法如下（因计算平台是内网环境，无法连接互联网，可以通过下载安装包后按照使用手册文件上传下载内容的方法进行上传，再进行安装操作）：

```
# 获取源码  
export VERSION=2021.3  
wget --no-check-certificate https://ftp.gromacs.org/gromacs/gromacs-${VERSION}.tar.gz  
  
# 解压  
tar xf gromacs-${VERSION}.tar.gz  
cd gromacs-${VERSION}  
  
# 编译安装  
# 导入编译环境，avx512 需要高版本 gcc 支持  
module load openmpi/3.1.1  
module load cmake/3.18.3  
module load gcc/7.3.0  
  
# cpu 版本安装  
export CC=gcc CXX=g++  
mkdir build ; cd build  
  
# -DGMX_SIMD=AVX_512 需要在支持 avx512 的 CPU 处理器上才可编译成功
```

```

PREFIX=/gpfs/share/software/gromacs/2021.3/gcc-7.2.0/openmpi-3.1.3
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DGMX_MPI=ON -DGMX_SIMD=AVX_512 -
DCMAKE_INSTALL_PREFIX=$PREFIX
make && make install
GPU 版本安装
# gpu 版本的编译需要在 gpu 节点上执行
# 获取源码
export VERSION=2021.3
wget --no-check-certificate https://ftp.gromacs.org/gromacs/gromacs-${VERSION}.tar.gz

# 解压
tar xf gromacs-${VERSION}.tar.gz
cd gromacs-${VERSION}

# 编译安装
# 导入编译环境，avx512 需要高版本 gcc 支持
module load openmpi/3.1.1
module load cmake/3.18.3
module load gcc/7.3.0

# cpu 版本安装
export CC=gcc CXX=g++
mkdir build_gpu ; cd build_gpu

# -DGMX_SIMD=AVX_512 需要在支持 avx512 的 CPU 处理器上才可编译成功
PREFIX=/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/openmpi_3.1.3
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DGMX_GPU=CUDA -DGMX_MPI=ON -
DGMX_SIMD=AVX_512 -DCMAKE_INSTALL_PREFIX=$PREFIX
make && make install

```

Gromacs 提交作业步骤如下：

step 1. 创建工作目录；

```

mkdir gromacsJob1
cd gromacsJob

```

step 2. 将运行 gromacs 需要的相关文件上传到该文件夹下；

step 3. 在该文件夹下编写作业脚本，并命名为 gromacs.sh, 脚本内容如下：

```

#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
module load gcc/7.3.0

```

```
module load openmpi/3.1.1
/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/opempi_3.1.3/bin/GMXRC.bash
mpirun -np 48
/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/opempi_3.1.3/bin/gmx_mpi mdrun -deffnm
md1-cluster0 -dlb yes
```

step 4. 提交作业:

```
sbatch gromacs.sh
```

3.3 vasp

VASP 是维也纳大学 Hafner 小组开发的进行电子结构计算和量子力学-分子动力学模拟软件包。它是目前材料模拟和计算物质科学研究中最流行的商用软件之一。

安装 vasp 操作方法如下（因计算平台是内网环境，无法连接互联网，可以通过下载安装包后按照使用手册文件上传下载内容的方法进行上传，再进行安装操作）：

```
# 导入编译器
module load intel/2017u5
tar xf vasp.5.4.4.tar.gz
cd vasp.5.4.4
cp arch/makefile.include.linux_intel ./makefile.include
make all
# 编译完成后会在 vasp.5.4.4 的文件夹下的 bin 文件夹里生成 vasp_gam、vasp_ncl、
vasp_std 三个可自行文件。

# gpu 版本安装
module load intel/2017u5
tar xf vasp.5.4.4.tar.gz
cd vasp.5.4.4
cp arch/makefile.include.linux_intel makefile.include
# 修改 -openmp 为 -qopenmp
make gpu
# GPU 版本使用的时候需要载入 cuda，高版本如 2018 的 intel 编译器编译时会报错
```

VASP 提交作业步骤如下:

step 1. 创建工作目录;

```
mkdir vaspJob1
cd vaspJob1
```

step 2. 将运行 vasp 需要的相关文件上传到该文件夹下;

step 3. 在该文件夹下编写作业脚本，并命名为 vaspJob1.sh，脚本内容如

下:

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
```

```
# 导入运行环境
module load intel/2017u5
module load vasp/5.4.4
```

```
# MPI 跨节点运行
mpirun -n 64 vasp_std
```

step 4. 提交作业:

```
sbatch vaspJob1.sh
```

附录 1 Linux 常用命令

在 Linux 系统下命令行界面登陆成功后，即可进行操作，下面列举了常用的 Linux 系统命令：

1. 查看目录下有哪些文件：ls
2. 查看文件属性：ls -l
3. 切换目录：cd 目录名称，如 cd /data/software
4. 查看文件：cat 文件名，如 cat /data/home/zhouhao/testfile
5. 查看文件：more 文件名，如 more /data/home/zhouhao/testfile，空格键向下翻页
6. 删除文件：rm 文件名，如 rm /data/home/zhouhao/testfile
7. 删除文件夹：rm 文件夹名，如 rm -r /data/home/zhouhao/testdir
8. 编辑文件：vim 命令，vim 常用有两种模式：
 - 1) 阅读模式，用 vim 打开一个文件后就进入到阅读模式，在阅读模式下可以通过上下键调整页面阅读所打开的文件内容。
 - 2) 编辑模式，需要输入特定的键才能对文件内容进行编辑。
 - 3) 阅读模式——>编辑模式：
 - a 键：当前光标所在处的后一个字符处编辑内容
 - i 键：当前光标所在处编辑内容
 - o 键：当前光标所在行的下一行编辑内容
 - O 键：当前光标所在行的上一行编辑内容
 - 4) 编辑模式——>阅读模式：Esc 键
 - 5) 退出 vim 编辑界面，需要在阅读模式下先输入冒号，冒号后面可输入以下字符
 - w 保存不退出
 - wq 保存退出
 - wq! 保存并强制退出
 - q 不保存退出
 - q! 不保存强制退出

附录 2 常用脚本

vasp 脚本试列

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
module load intel/2017u5
module load vasp/5.4.4
mpirun vasp_std
scontrol show job $SLURM_JOBID
```

lammmps 脚本试列

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
module load intel/2017u5
module load lammmps/7Aug19
mpirun -np 80 lmp_mpi < in.cutting
scontrol show job $SLURM_JOBID
```

fluent 脚本试列

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
srun hostname -s > hostlist
/data/software/ansys18.0/v180/fluent/bin/fluent -ssh -g 3ddp -t64 -cnf=hostlist -mpi=intel -i
fluent.jou
scontrol show job $SLURM_JOBID
```

注：fluent.jou 文件需要单独编写，如有问题请联系集群管理员。

lsdyna 试列

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
```

```
#SBATCH --output=%J.out
```

```
unset SLURM_GTIDS
```

```
ulimit -s
```

```
module load intel/2017u5
```

```
PATH=$PATH:/data/software/ansys/17.2/v172/ansys/bin
```

```
lsdyna -np 64 memory=200000000 -mpi intelmpi i=111-1-1-3-v172.k -lsdynampp
```

附录 3 modulefile（使用场景）

平台使用 Environment Modules 以模块的形式对环境变量进行管理。在高性能计算集群系统中，安装有多种软件及其不同版本，它们需要设置不同的环境变量，Environment Module 可以将这些环境变量做成模块文件（modulefile）。模块可以被加载(load)、卸载(unload)、切换(switch)等，这些操作会改变相应的环境变量设置，让用户方便的在不同环境之间切换。相比将环境变量设置写入/etc/profile、~/.bashrc 或 ~/.bash_profile，Environment Module 操作只影响当前用户的当前登录环境；相比直接 source 文件，Environment Module 的操作可以撤销（卸载）。普通用户还可以自己编写 module，具有很好的定制性。用户不仅可以在命令行中 Environment Modules，也能在作业提交脚本中使用，对编译和计算环境都能够很好的控制。

首先介绍 module 常见的一些指令，接着介绍几个 module 的几个使用案例，最后介绍如何编写 modulefile 来管理自己的软件环境。

首先介绍 module 常见的一些指令：

```
module help      # 显示帮助信息
module avail     # 显示已经安装的软件环境
module load      # 导入相应的软件环境
module unload    # 删除相应的软件环境
module list      # 列出已经导入的软件环境
module purge     # 清除所有已经导入的软件环境
module switch [mod1] mod2 # 删除 mod1 并导入 mod2
```

接着介绍几个 module 的使用例子：

查看集群现有软件活库：

```
module avail
```

查看集群可用的 intel 版本：

```
module avail intel
```

导入 intel/2017u5 软件环境：

```
module load intel/2017u5
```

清除所有通过 module 导入的软件环境：

```
module purge
```

最后介绍如何编写 modulefile 来管理自己的软件环境。首先，创建目录用来存放自己的 modulefile：

```
mkdir ${HOME}/mymodulefiles # 创建目录用于放自己的 module file
```



```

echo "export MODULEPATH=${HOME}/mymodulefiles:\$MODULEPATH" >> ~/.bashrc
source ~/.bashrc # 或者退出重新登录即可
# 编写自己的第一个 module file
cd ${HOME}/mymodulefiles
vim myfirstmodulefile

```

然后在创建好的目录下编写 modulefile, 假设 /share/home/test/soft/gcc/7.2.0 安装了 gcc 编译器, 则可以这么编写 modulefile:

```

#%Module1.0
##
##
module-whatis "my first modulefile"

set topdir "/share/home/test/soft/gcc/7.2.0"
prepend-path PATH "${topdir}/bin"
prepend-path LIBRARY_PATH "${topdir}/lib"
prepend-path LD_LIBRARY_PATH "${topdir}/lib"
prepend-path LIBRARY_PATH "${topdir}/lib64"
prepend-path LD_LIBRARY_PATH "${topdir}/lib64"
prepend-path CPATH "${topdir}/include"
prepend-path CMAKE_PREFIX_PATH "${topdir}"
setenv CC "${topdir}/bin/gcc"
setenv CXX "${topdir}/bin/g++"
setenv FC "${topdir}/bin/gfortran"
setenv F77 "${topdir}/bin/gfortran"
setenv F90 "${topdir}/bin/gfortran"

```

编写好后执行 module avail 即可查看到刚刚写好的 modulefile 了。以下为编写 modulefile 文件常见的语法:

```

set          # 设置 modulefile 内部的变量
setenv       # 设置环境变量
prepend-path # 效果类似于 export PATH=xxx:$PATH
append-path  # 效果类似 export PATH=$PATH:xxx

```