中科曙光 Sugon

地球系统数值模拟装置使用培训

携手成就梦想

课程内容

中科曙光 Sugon

01 • 装置整体情况介绍

02 • 装置系统环境的使用

03 • 装置调度系统的使用

04 • 常见问题说明

装置总体介绍

中科曙光 Sugon

总体：

根据地球系统科学数值模拟对计算、数据处理、存储、可视化计算的需求，建设峰值计算能力超过15PFLOPS、存储容量超过80PB的高性能计算机系统，用于研究与地球科学系统数值模拟相适应的系统架构、耦合计算等关键技术，以及开发地球模式气候模拟等数值模拟的并行软件、研究全球气候变化的公共计算平台。

内容：

- 模式计算分系统承担地球系统模式的计算任务，拥有海量计算资源，提供强大的计算能力。
- 网络交换分系统负责计算过程当中的高速数据通信，同时提供系统管理通路和网络安全防护。
- 数据存储分系统存储各种数据，为所有数据提供可持续、高速访问的全局空间。
- 支撑管理分系统包含硬件和软件，提供模式开发和系统的运行必要的软硬件支撑。
- 基础设施分系统为整个大装置的运行提供必要的制冷和配电解决方案。

模式计算分系统

模式计算分系统峰值计算能力15PFlops，达到20万核，主要承担地球系统模式的计算任务，拥有海量计算资源，提供强大的计算能力。本系统总体上采用可动态重构的分区式机群系统来应对地球系统模式的不同计算需求，各分区既可独立使用，分别支持不同应用，又可以实现高效耦合，协同完成复杂模式计算。

CX52-G40-LP刀片

通用计算节点
256G内存，1360台
512G内存，120台
1024G内存，120台

H620-G30

一期计算节点
256G内存，600台

CX52-D40-LP刀片

专用加速计算节点
2CPU+2DCU，1160台
128G内存

- 模拟器系统共包含**3360**个计算节点，可提供数 **215040** 个CPU核心，**2320**块DCU加速卡

地球系统数值模拟装置

基础设施软硬件布局图

预装软件列表

中科曙光 Sugon

```
------------------------------------------------------------------------ /public/software/modules ------------------------------------------------------------------------
apps/anaconda3/5.3.0                                    compiler/intel/2021.3.1                         mathlib/netcdf/gnu/4.4.1
apps/esmf/intelmpi/7.0.0                                compiler/intel/2022.1.0                         mathlib/netcdf/intel/4.4.1
apps/esmf/intelmpi/7.0.1                                compiler/rocm/4.0                               mathlib/netcdf/intel/4.7.4
apps/esmf/intelmpi/8.0.1                                mathlib/antlr/gnu/2.7.7                         mathlib/netcdf/intel/4.8.1
apps/lammps/intelmpi/29Sep2021                          mathlib/antlr/intel/2.7.7                       mathlib/pio/gnu/hpcx-2.7.4-gcc7.3.1-2.5.1
apps/Lammps-DCU2/hpcx-gcc-7.3.1                         mathlib/cdo/intel/1.10.19                       mathlib/pio/gnu/openmpi-4.0.4-gcc4.8.5-2.5.1
apps/m4/universal/1.4.18                                mathlib/grib_api/intel/1.19.0                   mathlib/pio/intel/2.5.1
apps/ncl_ncarg/6.3.0                                    mathlib/gsl/intel/2.6                           mathlib/pnetcdf/gnu/hpcx-2.7.4-gcc7.3.1-1.12.1
apps/ncl_ncarg/6.6.2                                    mathlib/hdf4/gnu/4.2.13                         mathlib/pnetcdf/gnu/openmpi-4.0.4-gcc4.8.5-1.12.1
apps/nco/gnu/4.8.1                                      mathlib/hdf4/intel/4.2.13                       mathlib/pnetcdf/intel/1.12.1
apps/nco/intel/4.8.1                                    mathlib/hdf4/intel/4.2.15                       mathlib/pnetcdf/intel/1.12.3
apps/ncview/gnu/2.1.7                                   mathlib/hdf5/gnu/1.8.20                         mathlib/szip/gnu/2.1.1
apps/ncview/intel/2.1.7                                 mathlib/hdf5/intel/1.10.3                       mathlib/szip/intel/2.1.1
apps/PyTorch/1.7.mmcv/pytorch-1.7-mmcv1.3.8-rocm-4.0.1  mathlib/hdf5/intel/1.12.0                       mathlib/udunits/gnu/2.2.28
apps/PyTorch/dtk-22.10/1.10.0a0                         mathlib/hdf5/intel/1.12.2                       mathlib/udunits/intel/2.2.28
apps/singularity/3.8.0                                  mathlib/hdf5/intel/1.8.20                       mathlib/wgrib2/2.0.8
apps/TensorFlow/tf1.15.3-rocm4.0.1/hpcx-2.7.4-gcc-7.3.1 mathlib/jasper/gnu/1.900.1                     mathlib/zlib/gnu/1.2.11
apps/TensorFlow/tf2.5.0-rocm4.0.1/hpcx-2.7.4-gcc-7.3.1  mathlib/jasper/intel/1.900.1                    mathlib/zlib/intel/1.2.11
benchmark/imb/intelmpi/2017                             mathlib/jpeg/gnu/9a                             mpi/intelmpi/2017.4.239
compiler/cmake/3.20.1                                   mathlib/jpeg/intel/9a                           mpi/intelmpi/2021.3.0
compiler/dtk/22.04.2                                    mathlib/lapack/gnu/3.9.1                        mpi/intelmpi/2021.3.1
compiler/dtk/22.10.1                                    mathlib/lapack/intel/3.9.1                      mpi/intelmpi/2022.1.0
compiler/gcc/7.3.1                                      mathlib/libpng/gnu/1.2.12                       mpi/openmpi/gnu/4.0.4
compiler/intel/2021.3.0                                 mathlib/libpng/intel/1.2.12

------------------------------------------------------------------------ /opt/hpc/software/modules ------------------------------------------------------------------------
compiler/devtoolset/7.3.1       compiler/rocm/3.3               mpi/hpcx/2.7.4/intel-2017.5.239
compiler/intel/2017.5.239       mpi/hpcx/2.7.4/gcc-7.3.1        mpi/openmpi/4.0.4/gcc-7.3.1
```

课程内容

集群登录

步骤1： 安装模拟器专用vpn客户端

- VPN 客户端下载地址：
  - http://www.leagsoft.com/doc/article/103107.html（Linux系统推荐）

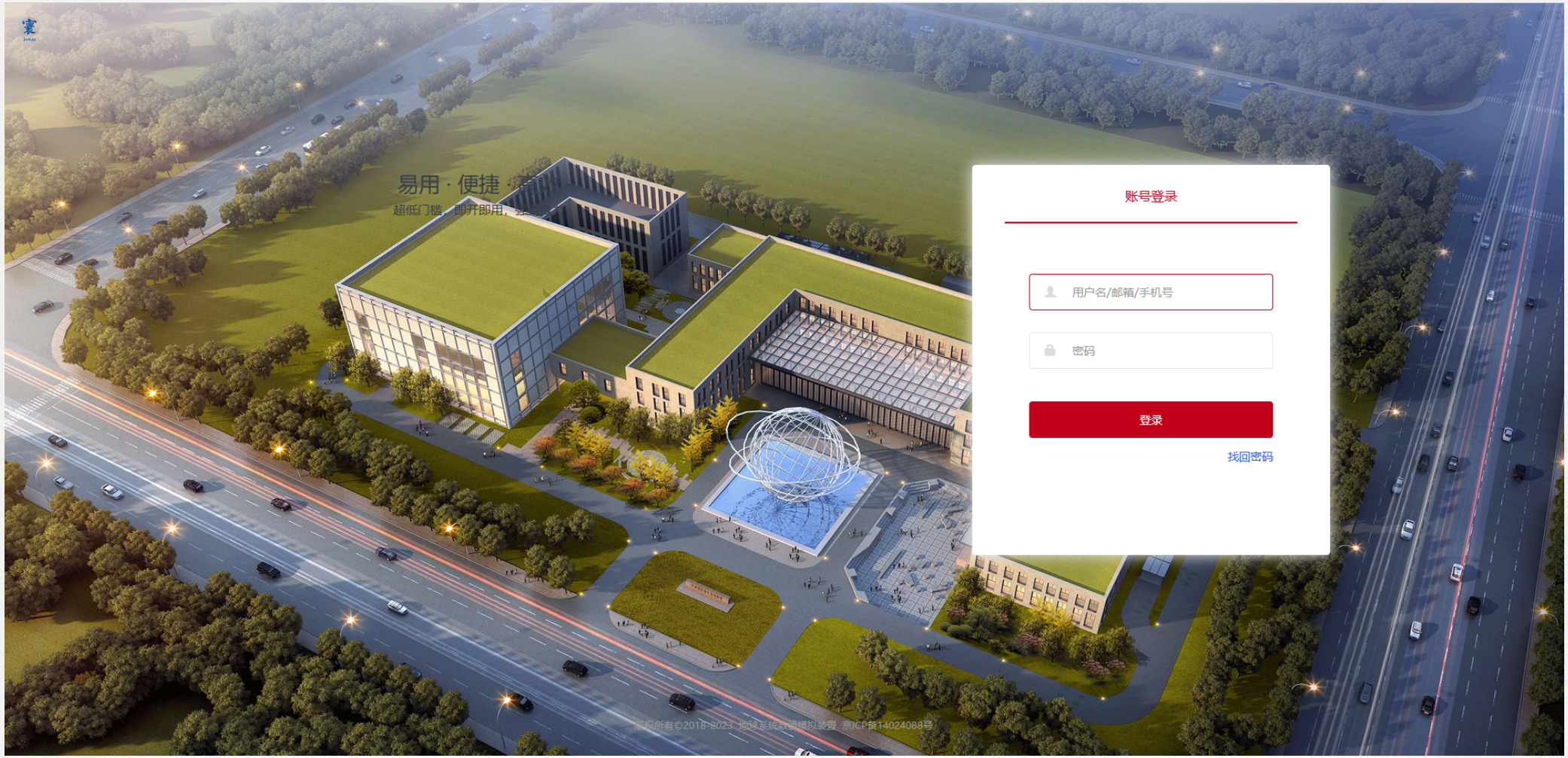
- SSL VPN 连接服务器地址：
  - 大气所北郊园区：
    网关地址：　　103.165.83.237
    Port：　　47240

  - 其他所/教育网/电信/联通/移动等：
    网关地址：　　cstnet.vpn.earthlab.iap.ac.cn
    Port：　　47240

集群登录

步骤2： 访问地球系统数值模拟装置在线平台

地址：https://earthlab-ac.iap.ac.cn



携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon

科学计算 | 作业提交 | 作业管理 | 文件管理（E-File） | 命令行「E-Shell」 | 支持与服务 | earthlab_user

快捷入口

编辑

模板提交

BASE 基础模板

>_ 命令行

作业状态

更多 >

作业总数 15 个

- 完成：0
- 运行：15
- 保留：0
- 排队：0
- 挂起：0
- 退出：0

地球系统数值模拟装置

可访问队列

数据更新时间：2023-06-06 19:59:23

占用 | 其它 | 空闲 | 展开

bigmem bigmem 空闲节点：77 总节点数：119 运行作业数：3 展开	**cpu_parallel** cpu_parallel 空闲节点：159 总节点数：1650 运行作业数：19 展开	**cpu_single** cpu_single 空闲节点：28 总节点数：570 运行作业数：30 展开	**dcu** dcu 空闲节点：572 总节点数：1160 运行作业数：14 展开
debug debug 空闲节点：30 总节点数：30 运行作业数：0 展开	**matlab** matlab 空闲节点：1 总节点数：1 运行作业数：0 展开		

携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon

可用资源

CPU

总核数: 25600

可用 24640　已用 960

加速卡

总卡数: 800

可用 790　已用 10

存储资源

个人主目录：/public/home/earthlab_user
已用 0.57%

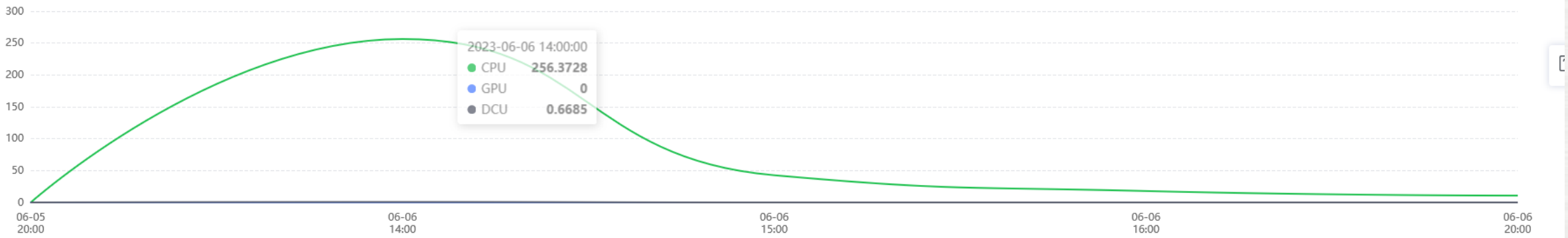
2.86 (GB)　500.0 (GB)

机时

最近30天机时消耗统计

CPU(核*时)：327.7346　GPU(卡*时)：0　DCU(卡*时)：0.6685

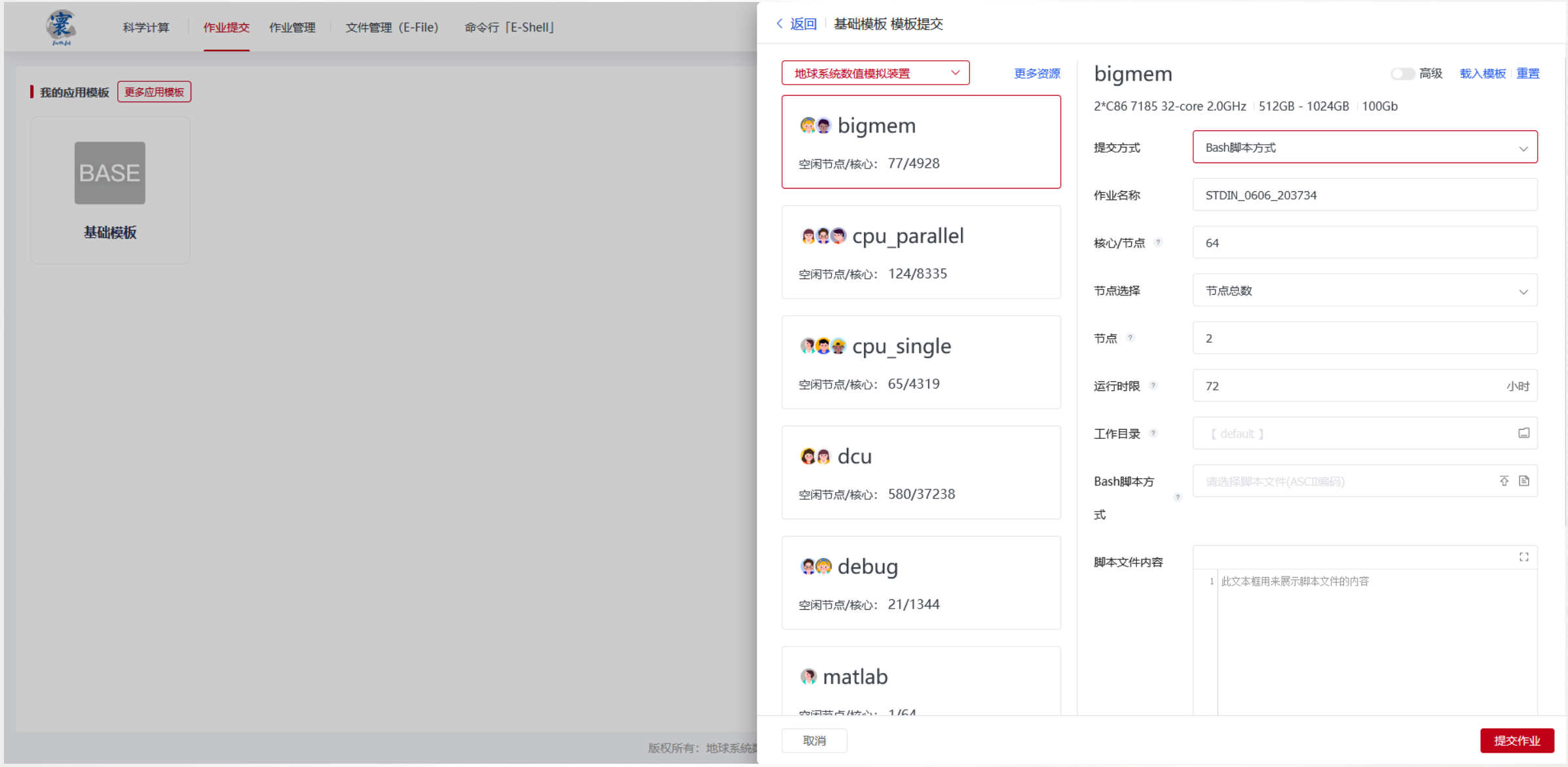
CPU(核*时)　GPU(卡*时)　DCU(卡*时)



携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon



携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon

科学计算 | 作业提交 | 作业管理 | 文件管理（E-File） | 命令行「E-Shell」 | 支持与服务 | earthlab_user

当前作业 | 历史作业

全部区域 | 展示组成员作业 | 作业ID | 搜索... | 全部状态 | 全部队列 | 30s

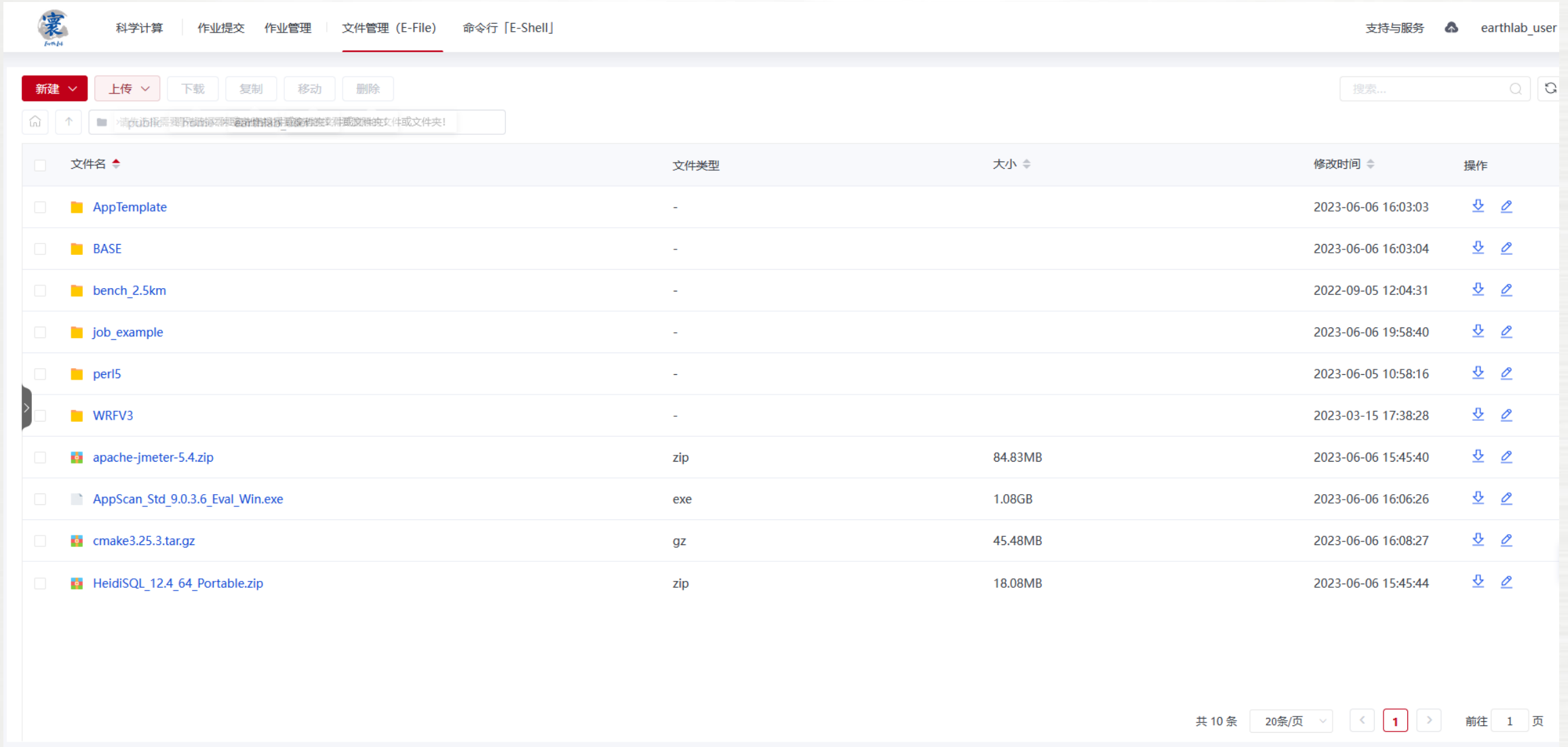
作业ID	作业名	应用	队列	区域	开始时间	运行时长	状态	操作
8327349	wrf	wrf	cpu_single	地球系统数值模拟装置	2023-06-06 20:20:54	00:14:01	运行	
8327210	testdcu	test	dcu	地球系统数值模拟装置	2023-06-06 19:59:02	00:35:53	运行	
8327206	testdcu	test	dcu	地球系统数值模拟装置	2023-06-06 19:59:02	00:35:53	运行	
8327207	testdcu	test	dcu	地球系统数值模拟装置	2023-06-06 19:59:02	00:35:53	运行	
8327208	testdcu	test	dcu	地球系统数值模拟装置	2023-06-06 19:59:02	00:35:53	运行	
8327209	testdcu	test	dcu	地球系统数值模拟装置	2023-06-06 19:59:02	00:35:53	运行	
8327179	testparallel	test	cpu_parallel	地球系统数值模拟装置	2023-06-06 19:58:51	00:36:04	运行	
8327180	testparallel	test	cpu_parallel	地球系统数值模拟装置	2023-06-06 19:58:51	00:36:04	运行	
8327181	testparallel	test	cpu_parallel	地球系统数值模拟装置	2023-06-06 19:58:51	00:36:04	运行	
8327182	testparallel	test	cpu_parallel	地球系统数值模拟装置	2023-06-06 19:58:51	00:36:04	运行	
8327183	testparallel	test	cpu_parallel	地球系统数值模拟装置	2023-06-06 19:58:51	00:36:04	运行	

共 11 条 | 20条/页 | 1 | 前往 1 页

携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon



携手成就梦想

地球系统数值模拟装置在线平台

中科曙光 Sugon

命令行（E-Shell） 文件管理（E-File）

1 login02 +

```
Last login: Tue Jun  6 20:29:57 2023 from admin09
[earthlab_user@login02 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           8327179 cpu_paral testpara earthlab  R      45:26      2 a3406n[08-09]
           8327180 cpu_paral testpara earthlab  R      45:26      2 a3412n[17-18]
           8327181 cpu_paral testpara earthlab  R      45:26      2 b3305r2n[7-8]
           8327182 cpu_paral testpara earthlab  R      45:26      2 a3106n[09-10]
           8327183 cpu_paral testpara earthlab  R      45:26      2 a3213n[12-13]
           8327349 cpu_singl      wrf earthlab  R      23:23      1 a3402n05
           8327206       dcu  testdcu earthlab  R      45:15      1 e2211r3n3
           8327207       dcu  testdcu earthlab  R      45:15      1 e2413r2n2
           8327208       dcu  testdcu earthlab  R      45:15      1 e2109r1n1
           8327209       dcu  testdcu earthlab  R      45:15      1 e2109r1n2
           8327210       dcu  testdcu earthlab  R      45:15      1 e2401r5n3
[earthlab_user@login02 ~]$
```

存储资源介绍

步骤2： 访问地球系统数值模拟装置在线平台

用户登录后：

家目录在　　/public/home/xxxx　　（500G空间配额 SSD分区）

数据目录　　/data1/xxxx　　（20TB空间配额 非SSD分区）

※ 存储空间也是一种计算资源，而且总容量是有限的，建议培养良好的使用习惯，定期清理无用或无效数据，不要把存储空间当成廉价的存储或网盘使用

※ 重要数据建议自行进行异地备份，对数据做删除等有风险操作前，建议先进行小范围验证再批量操作

计算资源介绍

队列名称	单节点配置	最大运行作业	最大运行时间	队列节点配置
cpu_single	64C 256GB mem	10	3天（72小时）	CPU 串行计算节点 单个作业 最多使用 1 个节点，64 核心
cpu_parallel	64C 256GB mem	10	30天（720小时）	CPU 并行计算节点 单个作业 最少使用 2 个节点，120核心
dcu	64C 128GB mem 2DCU	10	30天（720小时）	DCU加速节点，也可用于CPU计算 单作业资源不做限制
bigmem	64C 512GB – 1024GB	5	15天（360小时）	大内存节点 适用于内存占用 较高的程序，但作业资源不做限制
matlab	64C 512GB	5	1天（24小时）	包含1个大内存节点，matlab专用队列
debug	64C 256GB	1	1小时	程序调试专用节点

※ 计算队列与资源限制会随运营策略进行调整

密码重置与修改

地球系统数值模拟装置在线平台账号与计算集群系统账号密码不一致

在线平台账号密码支持自助修改与找回

计算集群系统账号仅支持自住修改不支持找回
自助修改可以在命令行（ E-shell ）使用passwd命令

如果忘记计算集群系统密码可发送邮件到支持邮箱由管理员进行重置

携手成就梦想

软件环境加载

环境变量

（environment variable）

是用来存储有关shell会话和工作环境的信息的一种特性

这项特性允许你在内存中存储数据以便程序或shell中运行的脚本能够轻松访问到它们。这也是存储持久数据的一种简便方法。

```
NFSCONF=/opt/clusconf/etc/nfs.cfg
MAIL=/var/spool/mail/root
PATH=/public/software/module-4.4.1/bin:/opt/clusconf/sbin
PWD=/root
IPMICONF=/opt/clusconf/etc/ipmi.cfg
LANG=en_US.UTF-8
MODULEPATH=/public/software/modules
LOADEDMODULES=
AUTOCLUSCONF=/opt/clusconf/etc/autoconf.cfg
HISTCONTROL=ignoredups
SHLVL=1
HOME=/root
BASH_ENV=/public/software/module-4.4.1/init/bash
LOGNAME=root
STARTWAITTIME=300
```

TIPS

＊ 系统环境变量名一般都采用大写表示
＊ 用户自定义的变量建议都用小写避免与系统环境变量重名导致的问题

软件环境加载

灵活的加载并使用环境变量

方法1、source 环境变量脚本

```
MPI_HOME=/public/software/mpi/intelmpi/2017.4.239
export  I_MPI_ROOT=${MPI_HOME}
export  PATH=${MPI_HOME}/intel64/bin:$PATH
export  LD_LIBRARY_PATH=${MPI_HOME}/intel64/lib:$LD_LIBRARY_PATH
export  MANPATH=${MPI_HOME}/man:$MANPATH
export  INCLUDE=${MPI_HOME}/intel64/include:$INCLUDE
```

```
[root@admin1 profile.d]# which mpirun
/usr/bin/which: no mpirun in (/usr/lib64/qt-3.3/bin:/public/software/NCL/6.6.2//bin:/opt
ched/sbin:/opt/gridview//pbs/dispatcher/bin/lsf_cmd:/opt/gridview//pbs/dispatcher/bin:/
idview/clusquota//bin:/opt/gridview/clusquota//sbin:/opt/clusconf/bin:/usr/local/sbin:/
[root@admin1 profile.d]# source /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
[root@admin1 profile.d]# which mpirun
/public/software/mpi/intelmpi/2017.4.239/intel64/bin/mpirun
```

软件环境加载

中科曙光 Sugon

灵活的加载并使用环境变量

方法1、source 环境变量脚本

仅支持单一shell

```
[root@admin1 ~]# echo $SHELL
/bin/csh
[root@admin1 ~]# source /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
MPI_HOME=/public/software//mpi/intelmpi/2017.4.239: Command not found.
MPI_HOME: Undefined variable.
```

同一环境变量可以多次加载，不会对有冲突的环境变量进行检查

```
[root@admin1 ~]# source /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
[root@admin1 ~]# source /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
[root@admin1 ~]# source /public/software/profile.d/mpi_openmpi-intel-2.1.2.sh
[root@admin1 ~]# source /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
[root@admin1 ~]# source /public/software/profile.d/mpi_openmpi-intel-2.1.2.sh
[root@admin1 ~]#
[root@admin1 ~]# echo $PATH
/public/software//mpi/openmpi/intel/2.1.2/bin:/public/software//mpi/intelmpi/2017.4.239/intel64/bin:/public
lic/software//mpi/intelmpi/2017.4.239/intel64/bin:/public/software//mpi/intelmpi/2017.4.239/intel64/bin:/pu
l64/bin:/public/software/module-4.4.1/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

软件环境加载

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

```
[root@admin1 ~]# cat /public/software/modules/mpi/intelmpi/2017.4.239
#%Module1.0

module-whatis   "IntelMPI MPI utilities for IB/intel compiler"

conflict mpi

set             MPI_HOME                /public/software//mpi/intelmpi/2017.4.239

setenv          I_MPI_ROOT              /public/software//mpi/intelmpi/2017.4.239
prepend-path    PATH                    ${MPI_HOME}/intel64/bin
prepend-path    LD_LIBRARY_PATH         ${MPI_HOME}/intel64/lib
prepend-path    MANPATH                 ${MPI_HOME}/man
prepend-path    INCLUDE                 ${MPI_HOME}/intel64/include
```

软件环境加载

灵活的使用加载环境变量

方法2、使用module命令控制环境变量

查看当前可用的modulefile

命令：module av or module avail

装置中预安装软件的modulefile路径

/public/software/modules

```
------------------------------------------------------------ /public/software/modules/ -------------------
apps/anaconda3/5.3.0                                        mathlib/jasper/intel/1.900.1
apps/esmf/intelmpi/7.0.0                                    mathlib/jpeg/gnu/9a
apps/m4/universal/1.4.18                                    mathlib/jpeg/intel/9a
apps/ncl_ncarg/6.3.0                                        mathlib/libpng/gnu/1.2.12
apps/nco/gnu/4.8.1                                          mathlib/libpng/intel/1.2.12
apps/nco/intel/4.8.1                                        mathlib/netcdf/gnu/4.4.1
apps/ncview/gnu/2.1.7                                       mathlib/netcdf/intel/4.4.1
apps/ncview/intel/2.1.7                                     mathlib/pio/gnu/hpcx-2.7.4-gcc7.3.1-2.5.1
apps/PyTorch/1.7.mmcv/pytorch-1.7-mmcv1.3.8-rocm-4.0.1      mathlib/pio/gnu/openmpi-4.0.4-gcc4.8.5-2.5.1
apps/TensorFlow/tf1.15.3-rocm4.0.1/hpcx-2.7.4-gcc-7.3.1     mathlib/pio/intel/2.5.1
apps/TensorFlow/tf2.5.0-rocm4.0.1/hpcx-2.7.4-gcc-7.3.1      mathlib/pnetcdf/gnu/hpcx-2.7.4-gcc7.3.1-1.12.1
benchmark/imb/intelmpi/2017                                 mathlib/pnetcdf/gnu/openmpi-4.0.4-gcc4.8.5-1.12.1
compiler/cmake/3.20.1                                       mathlib/pnetcdf/intel/1.12.1
compiler/rocm/4.0                                           mathlib/szip/gnu/2.1.1
mathlib/antlr/gnu/2.7.7                                     mathlib/szip/intel/2.1.1
mathlib/antlr/intel/2.7.7                                   mathlib/udunits/gnu/2.2.28
mathlib/cdo/intel/1.10.19                                   mathlib/udunits/intel/2.2.28
mathlib/grib_api/intel/1.19.0                               mathlib/wgrib2/2.0.8
mathlib/hdf4/gnu/4.2.13                                     mathlib/zlib/gnu/1.2.11
mathlib/hdf4/intel/4.2.13                                   mathlib/zlib/intel/1.2.11
mathlib/hdf5/gnu/1.8.20                                     mpi/intelmpi/2017.4.239
mathlib/hdf5/intel/1.8.20                                   mpi/openmpi/gnu/4.0.4
mathlib/jasper/gnu/1.900.1
```

软件环境加载

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

加载需要的modulefile

命令：**module add or module load**

module既支持bash也支持csh

```
[root@admin1 ~]# echo $SHELL
/bin/bash
[root@admin1 ~]# module load mpi/intelmpi/5.0.2.044
```

```
[root@admin1 ~]# echo $SHELL
/bin/csh
[root@admin1 ~]# module load mpi/intelmpi/5.0.2.044
```

module会根据modulefile文件的conflict字段对冲突的环境进行控制

```
[root@admin1 ~]# module load python/2.7
[root@admin1 ~]# module load python/3.7
Loading python/3.7
  ERROR: python/3.7 cannot be loaded due to a conflict.
    HINT: Might try "module unload python" first.
```

软件环境加载

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

查看当前已经加载的modulefile

命令：module li or module list

```
[root@admin2 ~]# module li
Currently Loaded Modulefiles:
  1) mpi/intelmpi/2017.4.239                      3) anaconda/5.3.0
  2) compiler/intel/intel-compiler-2017.5.239     4) cmake/3.16.6
```

软件环境加载

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

删除当前已经加载的modulefile

命令：**module rm or module unload**　　删除某一个或多个**modulefile**

```
[root@admin2 ~]# module unload cmake/3.16.6 anaconda/5.3.0
[root@admin2 ~]# module li
Currently Loaded Modulefiles:
  1) mpi/intelmpi/2017.4.239                    2) compiler/intel/intel-compiler-2017.5.239
```

删除所有已经加载的modulefile

命令: **module purge**

```
[root@admin2 novar_run.batch]# module li
Currently Loaded Modulefiles:
  1) mpi/intelmpi/2017.4.239                    2) compiler/intel/intel-compiler-2017.5.239
[root@admin2 novar_run.batch]# module purge
[root@admin2 novar_run.batch]# module li
No Modulefiles Currently Loaded.
```

软件环境加载

中科曙光 Sugon

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

替换当前已经加载的modulefile

命令：**module swap**

```
[root@admin1 ~]# module load python/2.7
[root@admin1 ~]# module load python/3.7
Loading python/3.7
  ERROR: python/3.7 cannot be loaded due to a conflict.
    HINT: Might try "module unload python" first.
[root@admin1 ~]# module swap python/3.7
```

某些版本中使用module swap 要求加载的modulefile的路径必须完全一样，否则会报错

```
[root@admin2 novar_run.batch]# module li
Currently Loaded Modulefiles:
  1) mpi/intelmpi/2017.4.239
[root@admin2 novar_run.batch]# module swap mpi/openmpi/intel/4.0.3
ModuleCmd_Switch.c(172):ERROR:152: Module 'mpi/openmpi/intel' is currently not loaded
```

软件环境加载

灵活的加载并使用环境变量

方法2、使用module命令控制环境变量

编写自己的modulefile

添加自定义的MODULEPATH

export MODULEPATH=[YOUR_NEW_MODULEPATH]:$MODULEPATH

或者

module use YOUR_NEW_MODULEPATH

```
[sugon@admin2 ~]$ export MODULEPATH=~/modules:$MODULEPATH
[sugon@admin2 ~]$ module av

------------------------------------------------------------ /public/home/sugon/modules --
python/2.7 python/3.7

------------------------------------------------------------ /public/software/modules ---
anaconda/5.2.0                                  mpi/openmpi/gnu/4.0.3_gcc4.8.5
anaconda/5.3.0                                  mpi/openmpi/gnu/4.0.3_gcc7.2
cmake/3.16.6                                    mpi/openmpi/intel/4.0.3
```

课程内容

调度系统命令介绍

中科曙光 Sugon

命令分类	命令	功能介绍
作业提交和控制	sbatch	提交脚本排队执行，批处理模式。
	salloc	创建资源申请并启动shell用于运行作业，交互模式
	srun	创建资源申请并启动作业（通常是MPI作业）
	scancel	取消作业或作业步。
系统状态	sinfo	查看节点和分区的状态。
	squeue	查看作业和作业步的状态。
	scontrol	查看或更新各种对象（如集群、分区、作业、等）的状态。

查询信息-sinfo

示例1: 默认格式查看分区的状态信息

sinfo

```
[sghpc2@login04 ~]$sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
serial       up   infinite     20   idle cmac[0011-0030]
normal       up   infinite      1 drain* cmac1174
normal       up   infinite    192  drain cmac[1091-1173,1175-1282,1304]
normal       up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
normal       up   infinite   1257   idle cmac[0049-0056,0089-0093,0102-1090,1283-1303,1305-1538]
operation    up   infinite      1 drain* cmac1174
operation    up   infinite    192  drain cmac[1091-1173,1175-1282,1304]
operation    up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
operation    up   infinite   1257   idle cmac[0049-0056,0089-0093,0102-1090,1283-1303,1305-1538]
sgtest       up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
sgtest       up   infinite    747   idle cmac[0049-0056,0089-0093,0102-0835]
[sghpc2@login04 ~]$
```

示例2: 长格式查看分区的状态信息（ --long / -l ）

sinfo -l

```
[sghpc2@login04 ~]$sinfo --long
Tue Mar 27 22:51:58 2018
PARTITION AVAIL  TIMELIMIT   JOB_SIZE ROOT OVERSUBS     GROUPS  NODES       STATE NODELIST
serial       up   infinite 1-infinite   no       NO        all      1       mixed cmac0011
serial       up   infinite 1-infinite   no       NO        all     19        idle cmac[0012-0030]
normal       up   infinite 1-infinite   no       NO        all      1       down* cmac0338
normal       up   infinite 1-infinite   no       NO        all      1    draining cmac0533
normal       up   infinite 1-infinite   no       NO        all      4     drained cmac[0059-0061,1304]
normal       up   infinite 1-infinite   no       NO        all     74   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545,1122-1137,1530]
normal       up   infinite 1-infinite   no       NO        all   1423        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-1121,1138-1303,1305-1400,1402-1529,1531-1538]
normal       up   infinite 1-infinite   no       NO        all      1        down cmac1401
operation    up   infinite 1-infinite   no       NO        all      1       down* cmac0338
operation    up   infinite 1-infinite   no       NO        all      1    draining cmac0533
operation    up   infinite 1-infinite   no       NO        all      4     drained cmac[0059-0061,1304]
operation    up   infinite 1-infinite   no       NO        all     74   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545,1122-1137,1530]
operation    up   infinite 1-infinite   no       NO        all   1423        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-1121,1138-1303,1305-1400,1402-1529,1531-1538]
operation    up   infinite 1-infinite   no       NO        all      1        down cmac1401
sgtest       up   infinite 1-infinite   no       NO        all      1       down* cmac0338
sgtest       up   infinite 1-infinite   no       NO        all      1    draining cmac0533
sgtest       up   infinite 1-infinite   no       NO        all      3     drained cmac[0059-0061]
sgtest       up   infinite 1-infinite   no       NO        all     57   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545]
sgtest       up   infinite 1-infinite   no       NO        all    739        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-0835]
[sghpc2@login04 ~]$
```

查询作业 squeue

squeue:查询排队和运行状态的作业

参数	解释
-j, --job=job(s)	已逗号分隔指定显示的jobid，默认显示全部
-n, --name=job_name(s)	逗号分隔指定的作业名称
-o, --format=format	指定显示的信息
-p, --partition=partition(s)	逗号分隔指定队列中的作业

```
[earthlab_user@login01 job_example]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           8313272 cpu_singl     test earthlab  R       0:01      1 a3104n03
```

查询资源-scontrol show

查看状态和配置命令
scontrol show <COMMAND>

COMMAND	解释
job	显示作业信息
node	显示节点信息
partition	显示队列信息

```
[earthlab_user@login01 job_example]$ scontrol show job 8313263
JobId=8313263 JobName=test
   UserId=earthlab_user(2348) GroupId=earthlab_user(2356) MCS_label=N/A
   Priority=2000 Nice=0 Account=earthlab_user QOS=user_earthlab_user
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=INVALID TimeLimit=3-00:00:00 TimeMin=N/A
   SubmitTime=2023-06-05T20:21:57 EligibleTime=2023-06-05T20:21:57
   AccrueTime=2023-06-05T20:21:57
   StartTime=2023-06-05T20:21:58 EndTime=2023-06-08T20:21:58 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-06-05T20:21:58
   Partition=cpu_single AllocNode:Sid=login01:52603
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=a3106n17
   BatchHost=a3106n17
   NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,mem=1800M,node=1,billing=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryCPU=1800M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/public/home/earthlab_user/job_example/sleep.slurm
   WorkDir=/public/home/earthlab_user/job_example
   Comment=test
   StdErr=/public/home/earthlab_user/job_example/slurm-8313263.out
   StdIn=/dev/null
   StdOut=/public/home/earthlab_user/job_example/slurm-8313263.out
   Power=
   NtasksPerTRES:0
```

```
[root@login01 ~]# scontrol show partition debug
PartitionName=debug
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=partition_debug_5869
   DefaultTime=01:00:00 DisableRootJobs=YES ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=01:00:00 MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED LLS=NO
   Nodes=a3112n[01-19],a3314n[08-18]
   PriorityJobFactor=1 PriorityTier=6000 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=1920 TotalNodes=30 SelectTypeParameters=NONE
   JobDefaults=(null)
   SuspendKeepIdle=n/a
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

```
[root@login01 ~]# scontrol show node b2108r3n1
NodeName=b2108r3n1 Arch=x86_64 CoresPerSocket=8
   CPUAlloc=64 CPUTot=64 CPULoad=1.14
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=10.1.118.17 NodeHostName=b2108r3n1 Version=20.11.8-1.5.3-20230305
   OS=Linux 3.10.0-957.el7.x86_64 #1 SMP Fri Oct 25 22:34:08 UTC 2019
   RealMemory=1029690 AllocMem=480000 FreeMem=1001102 Sockets=8 Boards=1
   MemSpecLimit=10240
   State=ALLOCATED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=bigmem
   BootTime=2023-05-16T12:41:14 SlurmdStartTime=2023-05-17T17:01:18
   CfgTRES=cpu=64,mem=1029690M,billing=64
   AllocTRES=cpu=64,mem=480000M
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
   Comment=(null)
```

SLURM-提交作业

srun	sbatch	salloc
交互式作业提交	批处理作业提交	节点资源获取

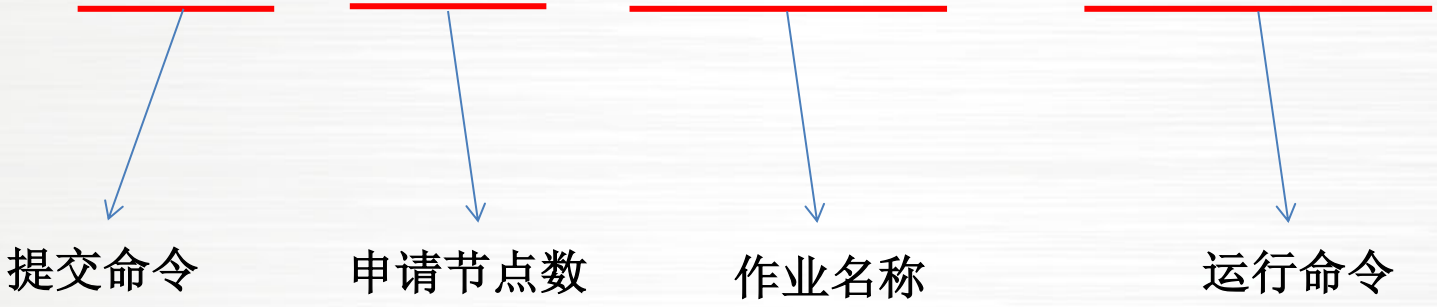
SLURM-提交作业参数

参数	参数解释
-J 或者 --job-name	指定作业名称
-p或者 --partition	指定队列资源
-N或者 --nodes=<number>	指定节点数量
-n或者 --ntasks=<number>	指定处理器数量
--ntask-per-node=<number>	指定每节点任务数，模拟器环境最大为64
--cpus-per-task=<number>	指定每任务CPU核心数对应多线程场景，可能需配合OMP_NUM_THREADS变量
--mem=xx	预留申请内存容量，单位MB、GB
--gres=dcu:<number>	申请DCU卡数，模拟器环境最大为2
-w或者 --nodelist=<node name list>	指定申请的节点，格式可以为node1，node2 或者node[1-10]
-x或者 --exclude=<node name list>	排除指定的节点，格式可以为node1，node2 或者node[1-10]
-o或者 --output=<filename pattern>	指定stdout的输出文件。如果指定的文件已经存在，它将被覆盖。
-e或者 --error=<filename pattern>	指定stderr的输出文件。如果指定的文件已经存在，它将被覆盖。
-t或者 --time=DD-HH:MM	指定作业运行时间，例如7-08:00代表7天8小时
--exclusive	设置节点独占

SLURM-srun

例：提交请求2个节点的并且指定作业的名称为job1

```
srun  -N 2  -J  job1  sleep 10
```



SLURM-sbatch

```
[earthlab@login01 ~]$ sbatch sleep.job      //sbatch 只接收脚本
Submitted batch job 19

[earthlab@login01 ~]$ cat sleep.job         //脚本格式示例
#!/bin/bash
#SBATCH -J sleep                            //指定作业名
#SBATCH -p debug                            //指定队列
#SBATCH --time=1                            //指定运行时间（分钟）
#SBATCH -N 2                                //请求节点数
#SBATCH -n 2                                //请求核心数
#SBATCH -o logs/%j.sleep                    //标准输出文件
#SBATCH -e logs/%j.sleep                    //错误输出文件

echo ${SLURM_JOB_NODELIST}                  //作业占用节点列表
echo  start on $(date)                      //开始时间
sleep 100                                   //执行命令
echo end on $(date)                         //结束时间
```

SLURM-salloc

```
[zlei@login01 software]$ salloc -p normal -n 1
salloc: Pending job allocation 308763
salloc: job 308763 queued and waiting for resources
salloc: job 308763 has been allocated resources
salloc: Granted job allocation 308763
salloc: Waiting for resource configuration
salloc: Nodes a3110n01 are ready for job
[zlei@login01 software]$ ssh a3110n01
Last login: Thu Nov 18 01:01:42 2021 from login01
[zlei@a3110n01 ~]$ module purge
[zlei@a3110n01 ~]$ module load compiler/intel/2017.5.239
[zlei@a3110n01 ~]$ module load mpi/intelmpi/2017.4.239
[zlei@a3110n01 ~]$ mpirun -np 1 hostname
a3110n01
[zlei@a3110n01 ~]$ exit
logout
Connection to a3110n01 closed.
[zlei@login01 software]$ exit
exit
salloc: Relinquishing job allocation 308763
```

1、提交任务
2、分配资源并生成jobid

3、登录分配节点执行任务

4、任务执行完成退出并回收资源

SLURM-查询作业sacct

参数	解释
-E, --endtime=end_time	查询在指定时间之前，任何状态的作业.如果通过-s参数指定状态则返回在此时间之前的指定状态的作业，有效格式为： HH:MM[:SS] [AM|PM] MMDD[YY] or MM/DD[/YY] or MM.DD[.YY] MM/DD[/YY]-HH:MM[:SS] YYYY-MM-DD[THH:MM[:SS]]
-S, --starttime= starttime	在指定时间后，任何状态的作业
-T, --truncate	如果一个job在 --starttime之前开始运行，开始时间将被截断为 --starttime，同样的作业结束时间 = --endtime
-o, --format	指定显示字段以逗号分隔

SLURM-命令sacct示例

```
[zlei@login01 software]$ sacct -S 2021-06-23 -E 2021-11-18
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
87422        Tensorflow       high       zlei         32     FAILED      1:0
87422.batch       batch                  zlei         16     FAILED      1:0
87422.extern     extern                  zlei         32  COMPLETED      0:0
87422.0           orted                  zlei          8  COMPLETED      0:0
87423        Tensorflow       high       zlei         32     FAILED      1:0
87423.batch       batch                  zlei         16     FAILED      1:0
87423.extern     extern                  zlei         32  COMPLETED      0:0
87423.0           orted                  zlei          8  COMPLETED      0:0
87431        Tensorflow       high       zlei         16     FAILED      2:0
87431.batch       batch                  zlei         16     FAILED      2:0
87431.extern     extern                  zlei         16  COMPLETED      0:0
87432        Tensorflow       high       zlei         16     FAILED      1:0
87432.batch       batch                  zlei         16     FAILED      1:0
87432.extern     extern                  zlei         16  COMPLETED      0:0
91407             bash     matlab       zlei          1  COMPLETED      0:0
91407.extern     extern                  zlei          1  COMPLETED      0:0
```

作业脚本示例

CPU：

/public/software/run.slurm-serial　　基于Intel编译器串行脚本示例

/public/software/run.slurm-intelmpi　基于Intel编译器+intelmpi环境示例

/public/software/run.slurm-hpcx　　基于intel编译器+hpcx（mpi）环境示例

/public/software/run.slurm-openmp　基于Intel+intelmpi+openmp使用场景的示例

DCU：

/public/software/run.slurm-DCU　　加速器使用脚本参考，绑定部分与single_process.sh使用

/public/software/single_process.sh　加速器卡绑定文件参考

脚本中的module 环境需要根据自己编译时依赖的软件环境保持一致

作业脚本示例1—通用作业脚本 IntelMPI

```
#!/bin/bash
#SBATCH -J JOB_NAME
#SBATCH -p normal
#SBATCH -N 2
#SBATCH -n 120
#SBATCH --ntasks-per-node=60
#SBATCH -o log.%j
#SBATCH -e log.%j
#SBATCH --exclusive
#SBATCH -t 7-24:00

module purge
module load compiler/intel/2017.5.239
module load mpi/intelmpi/2017.4.239
module load mathlib/netcdf/intel/4.4.1
#module load ...

export I_MPI_FABRICS=shm:dapl
export I_MPI_DAPL_UD=1
export I_MPI_DAPL_UD_RDMA_MIXED=1
export I_MPI_LARGE_SCALE_THRESHOLD=8192
export I_MPI_DAPL_UD_ACK_SEND_POOL_SIZE=8704
export I_MPI_DAPL_UD_ACK_RECV_POOL_SIZE=8704
export I_MPI_DAPL_UD_RNDV_EP_NUM=2
export DAPL_UCM_REP_TIME=8000
export DAPL_UCM_RTU_TIME=8000
export DAPL_UCM_RETRY=10
export DAPL_UCM_CQ_SIZE=2000
export DAPL_UCM_QP_SIZE=2000
export DAPL_UCM_DREQ_RETRY=4
export DAPL_UCM_DREP_TIME=200
export DAPL_UCM_WAIT_TIME=10000


scontrol show hostname > nd
NP=$SLURM_NPROCS
mpirun -np $NP -machinefile nd /path/to/app
```

指定作业名称
指定作业队列
指定作业申请的资源，包括节点数、进程数等
指定作业输出日志
设定申请节点独占
设定作业运行时间

加载或设置作业所需的环境变量

生成节点列表
作业执行语句

作业脚本示例2—通用作业脚本 OpenMPI

```
#!/bin/bash
#SBATCH -J JOB_NAME
#SBATCH -p normal
#SBATCH -N 4
#SBATCH -n 240
#SBATCH --ntasks-per-node=60
#SBATCH -o log.%j
#SBATCH -e log.%j
#SBATCH --exclusive
#SBATCH -t 7-24:00

module purge
module load compiler/intel/2017.5.239
module load mpi/hpcx/2.7.4/intel-2017.5.239
module load mathlib/netcdf/intel/4.4.1

mpirun -np 240 ./wrf.exe
```

指定作业名称
指定作业队列
指定作业申请的资源，包括节点数、进程数等
指定作业输出日志
设定申请节点独占
设定作业运行时间

加载或设置作业所需的环境变量

作业执行语句

作业脚本示例3—串行作业脚本

```
#!/bin/bash
#SBATCH -J Serial
#SBATCH -p normal
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -o log.%j
#SBATCH -e log.%j
#SBATCH -t 7-24:00

module purge
module load compiler/intel/2017.5.239
module load mathlib/netcdf/intel/4.4.1
#module load ...


srun /path/to/app

/path/to/app
```

指定作业名称
指定作业队列
指定作业申请的资源，包括节点数、进程数等
指定作业输出日志
设定作业运行时间

加载或设置作业所需的环境变量

对于串行作业，可以在脚本中只用srun执行任务也
可以直接运行可执行程序

作业脚本示例4—MPI/OpenMP作业脚本

```
#!/bin/bash
#SBATCH -J JOB_NAME
#SBATCH -p normal
#SBATCH -N 2
#SBATCH -n 120
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=30
#SBATCH -o log.%j
#SBATCH -e log.%j
#SBATCH --exclusive
#SBATCH -t 7-24:00

export OMP_NUM_THREADS=30     ##--cpus-per-task=30
module purge
module load compiler/intel/2017.5.239
module load mpi/intelmpi/2017.4.239
module load mathlib/netcdf/intel/4.4.1
#module load ...

export I_MPI_FABRICS=shm:dapl
export I_MPI_DAPL_UD=1
export I_MPI_DAPL_UD_RDMA_MIXED=1
export I_MPI_LARGE_SCALE_THRESHOLD=8192
export I_MPI_DAPL_UD_ACK_SEND_POOL_SIZE=8704
export I_MPI_DAPL_UD_ACK_RECV_POOL_SIZE=8704
export I_MPI_DAPL_UD_RNDV_EP_NUM=2
export DAPL_UCM_REP_TIME=8000
export DAPL_UCM_RTU_TIME=8000
export DAPL_UCM_RETRY=10
export DAPL_UCM_CQ_SIZE=2000
export DAPL_UCM_QP_SIZE=2000
export DAPL_UCM_DREQ_RETRY=4
export DAPL_UCM_DREP_TIME=200
export DAPL_UCM_WAIT_TIME=10000

scontrol show hostname > nd
NP=$SLURM_NPROCS
mpirun -np $NP -machinefile nd /path/to/app
```

指定作业名称
指定作业队列
指定作业申请的资源，包括节点数、进程数等
指定作业输出日志
设定申请节点独占
设定作业运行时间

通过OMP_NUM_THREADS变量设置单个进程发起的线程数
通常情况这个值应该与cpus--per-task相同
在模拟器环境中 ntask-per-node * cpus-per-task 最大不超过64

加载或设置作业所需的环境变量

生成节点列表
作业执行语句

作业脚本示例5—作业依赖关联

提交作业时可以指定作业之间的依赖关系，通过各种逻辑关系的设置来完成一个复杂的业务处理流程。

常用的逻辑关系包括：

after（依赖作业开始运行时）

afterok（依赖作业正常结束时）

afterany（依赖作业均完成）

afternotok（依赖作业非正常结束）

如例所示：

提交A、B、C三个作业，其依赖关系为：

B作业需要在A作业正常结束后才能运行，

C作业需要在作业B正常完成后开始

```
[zlei@login01 ~]$ cat depdencysubmit.sh
#! /bin/bash

AJobId=`sbatch JobA.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobA.slurm"
        exit 1
fi

BJobId=`sbatch --dependency=afterok:$AJobId JobB.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobB.slurm"
        exit 1
fi

CJobId=`sbatch --dependency=afterok:$BJobId JobC.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobC.slurm"
        exit 1
fi
```

```
[zlei@login01 ~]$ sh depdencysubmit.sh
[zlei@login01 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            308789    normal     JobB     zlei PD       0:00      1 (Dependency)
            308790    normal     JobC     zlei PD       0:00      1 (Dependency)
            308788    normal     JobA     zlei  R       0:05      1 a3110n01
[zlei@login01 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            308788    normal     JobA     zlei CG       0:32      1 a3110n01
            308789    normal     JobB     zlei PD       0:00      1 (Dependency)
            308790    normal     JobC     zlei PD       0:00      1 (Dependency)
[zlei@login01 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            308790    normal     JobC     zlei PD       0:00      1 (Dependency)
            308789    normal     JobB     zlei  R       0:06      1 a3110n01
```

课程内容

常见问题

问题：如何反馈需求及问题？

- 依托地模装置工作中遇到的使用问题或技术问题，可发送技术交流台账到支持邮箱：
  earthlab-techsupport@mail.iap.ac.cn

- 邮箱中须写明装置账号和联系方式及具体的问题描述

- 若涉及资源调整或其他运营类问题需装置办审批通过后方可支持

- 若涉及作业类问题，请提供以下详细信息：

  - 问题描述，提供具体的报错信息或截图
  - 作业ID及作业工作路径（需包含作业运行脚本及作业日志）

常见问题

问题：
用户A需要把自己家目录下的某个文件或者某个目录共享给用户B
为了方便他把自己家目录的权限设置成了777
设置完成后，用户A突然发现自己提交的作业无法正常运算了

原因是用户家目录过于开放的权限使得SSH无密码访问失效了
因为SSH使用秘钥的过程中会检查秘钥文件和上级目录的权限，如果权限过于开放，SSH协议会判断存在不安全的行为，导致秘钥认证失败

如果需要共享部分数据，可以将目录权限设置为755

常见问题

问题：是否可以在登录节点运行作业？

- 大装置是一个公共环境，登录节点是公共使用资源，严禁在登录节点运行任何高负载、高消耗的任务

问题：如何传输数据？

- 少量数据推荐使用sftp工具（例如xftp、filezilla等）或rsync等支持断点续传的工具进行传输
- 大量数据如无法通过服务器直接传输，请提供移动硬盘等存储介质进行拷贝，如有此类需求请提前发数据台账到支持邮箱并提供以下信息：
  - 联系方式（手机或电话）
  - 大装置系统账号（集群登录账号，非VPN账号）
  - 存储介质类型（如服务器硬盘、移动硬盘、NAS设备等）
  - 运维工程师评估后，会跟您进行联系

常见问题

问题：数据拷贝速度慢？

- 数据拷贝的速度是一个综合所有因素的复杂结果，可能取决于以下多个方面：
  - 网络
  - 存储介质类型与性能
  - 大装置存储的负载
  - 传输协议的特性
  - 传输工具的性能
  - 传输文件的类型（文件数多少）
  - 传输两端的系统性能与负载
  - …

- 建议如下：
  - 拷贝数据尽可能精简（减少无用数据传输）
  - 使用支持断点续传的工具，启用多进程传输
  - 文件数较多目录先进行打包
  - 大量数据直接使用单独介质进行拷贝

常见问题

问题：模拟器上如何安装软件？

- 模拟器上提供了常用的编译器和部分数学库，可以使用module av命令进行查看
- 如果没有您需要的软件或数学库，可以自行在家目录编译安装
- 如果需要安装支持，请发技术台账到支持邮箱，安装支持要求如下：
  - 提供软件的安装介质
  - 提供软件的官网地址与安装手册或文档
  - 商业软件请提供可用的正式授权

问题：其他集群的软件直接拷贝能否使用？

- 异构操作系统软件不可混用
- 原则上操作系统与软件环境（依赖其他软件）差别不大的情况下可以使用，需要实际测试
- 建议在大装置环境上重新进行编译

常见问题

问题：使用调度系统提交作业是否直接可以并行计算？

- 调度系统仅负责分配计算资源并不直接参与任务计算的过程
- 是否可以并行计算取决于程序本身

问题：计算任务是不是使用的节点或核心数越多计算就越快？

- 最能够发挥装置计算能力的作业是扩展性好的作业
- 并行计算的效率取决于程序和算例本身的扩展性
- 需要通过实际测试与调试找到针对作业本身最合适的规模
- 建议提交作业时每节点预留2个核心用于系统自身服务开销

中科曙光
Sugon

谢谢！

IT技术及解决方案的领导者
数据中国百城百行的发起者
中科院产业化联盟的推动者
安全可控信息系统的践行者

Sugon

携手成就梦想