中科曙光
Sugon

高性能计算机基础培训
高性能集群使用和管理

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

- 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

装置软、硬件环境

IT设备名称	数量	说明
通用计算节点1	1960台	双路计算节点（2颗海光处理器，内存256GB）
通用计算节点2	240台	双路计算节点（2颗海光处理器，120台内存为512GB，120台内存为1024GB）
专用加速节点	1160台	双路异构计算节点（2颗海光处理器，2颗海光DCU加速卡，内存128GB）
高速计算网络	1套	800口HDR InfiniBand核心交换机、40口HDR InfiniBand接入交换机和连接线缆
管理监控网络	1套	核心交换机，万兆交换机，千兆交换机和连接线缆
在线存储	3台	ParaStor存储管理节点
	20台	ParaStor存储元数据节点
	228台	ParaStor存储数据节点
离线存储	1套	离线磁带库，提供50PB容量
系统管理节点	16台	H620-G30机架式服务器
系统登陆节点	16台	H620-G30机架式服务器
系统服务节点	32台	H620-G30机架式服务器

类别	软件
操作系统	CentOS Linux release 7.6.1810
编译器	Intel compiler
	GNU-4.8.5
数学库	Hdf-4.2.13
	Hdf5-1.8.20
	Jasper-1.900.1
	NetCDF-4.1.3
	NetCDF-4.4.1
	PnetCDF-1.12.1
	Zlib-1.2.11
MPI	IntelMPI
	HPCX-2.7.4
	OpenMPI-4.0.4
异构计算环境	ROCm-3.3

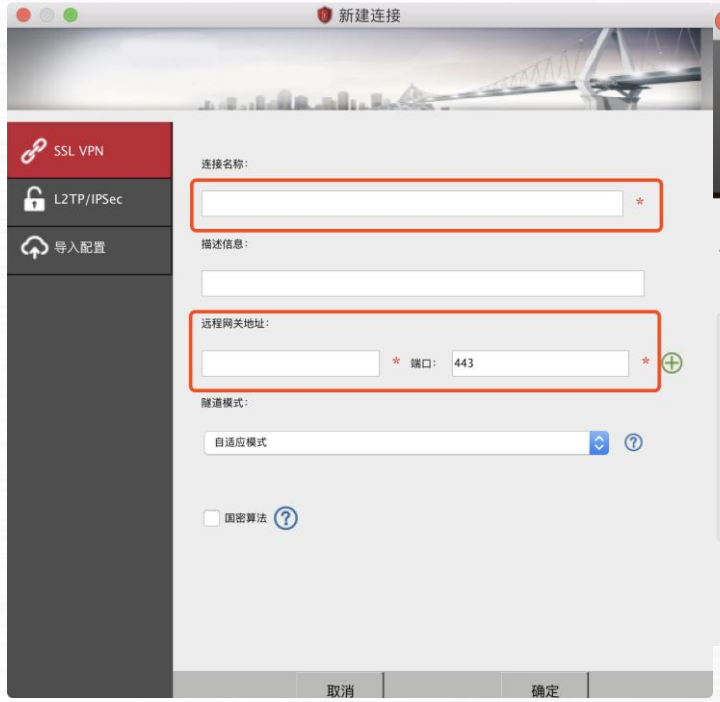
课程内容

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：
  - https://dell-r610-119-27.iap.ac.cn/secoclient.zip
  - 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：
    IP： 103.165.83.237
    Port： 47240
  - 其他所/教育网/电信/联通/移动等：
    IP： cstnet.vpn.earthlab.iap.ac.cn
    Port： 47240



集群登录

步骤2： 使用SSH 或基于 SSH 方式的客户端工具通过域名登录节点：

登录域名地址：login.earthlab.iap.ac.cn　　使用系统用户名和密码登录

用户登录后：家目录在　/public/home/xxxx　　（500G空间配额 SSD分区）

数据目录　/data/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，建议大家养成良好的使用习惯，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

计算队列划分

队列名称	状态	资源限制	最大运行时长	队列说明
normal	启用	无限制	24:00:00	包含全部3360个计算节点
high	启用	无限制	24:00:00	包含全部1160个DCU节点
bigmem	启用	无限制	24:00:00	包含239个大内存节点（512GB、1024GB）
matlab	启用	无限制	24:00:00	包含1个大内存节点，matlab专用队列

※ 计算队列与资源限制可能会随运营策略随时调整，可以使用sinfo命令查看

SSH客户端工具使用

win10 cmd or powershell

优点：

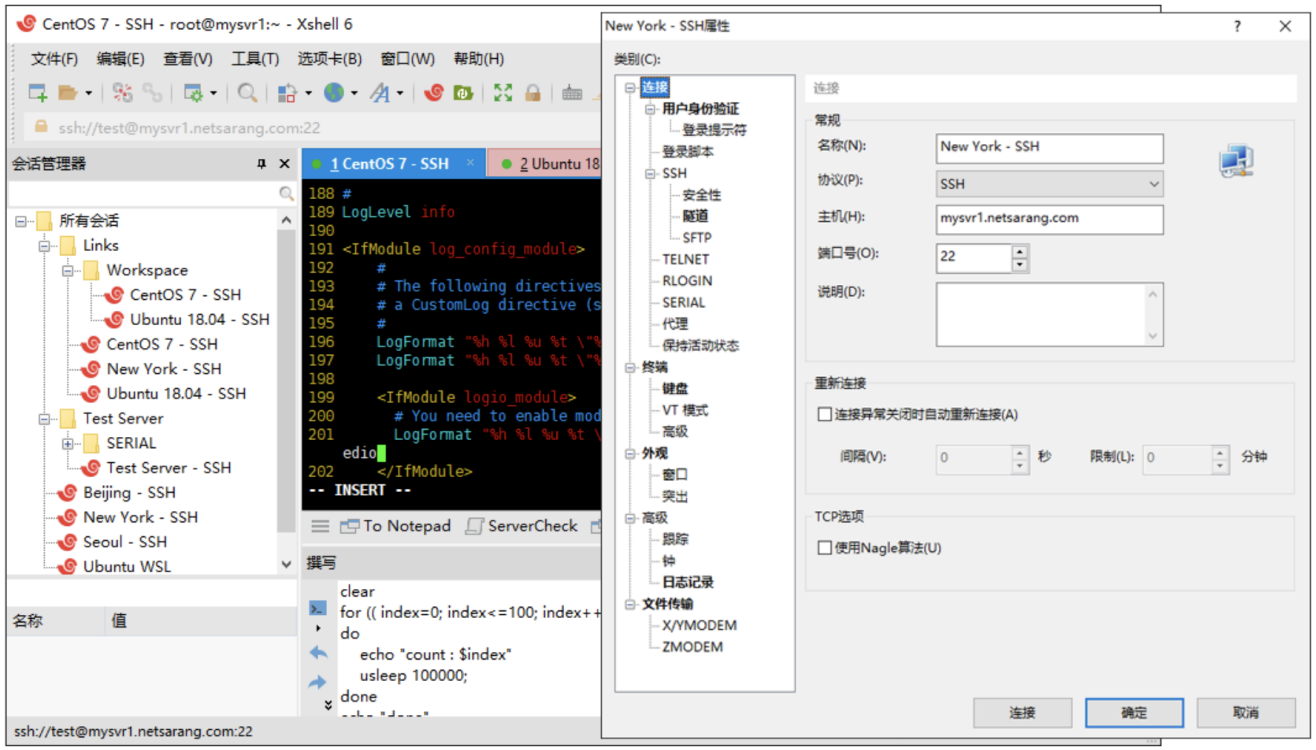
系统自带工具，不用再单独安装

缺点：

功能有限，不支持其他扩展功能

SSH客户端工具使用

XSHELL （官网地址 https://www.netsarang.com/zh/xshell)

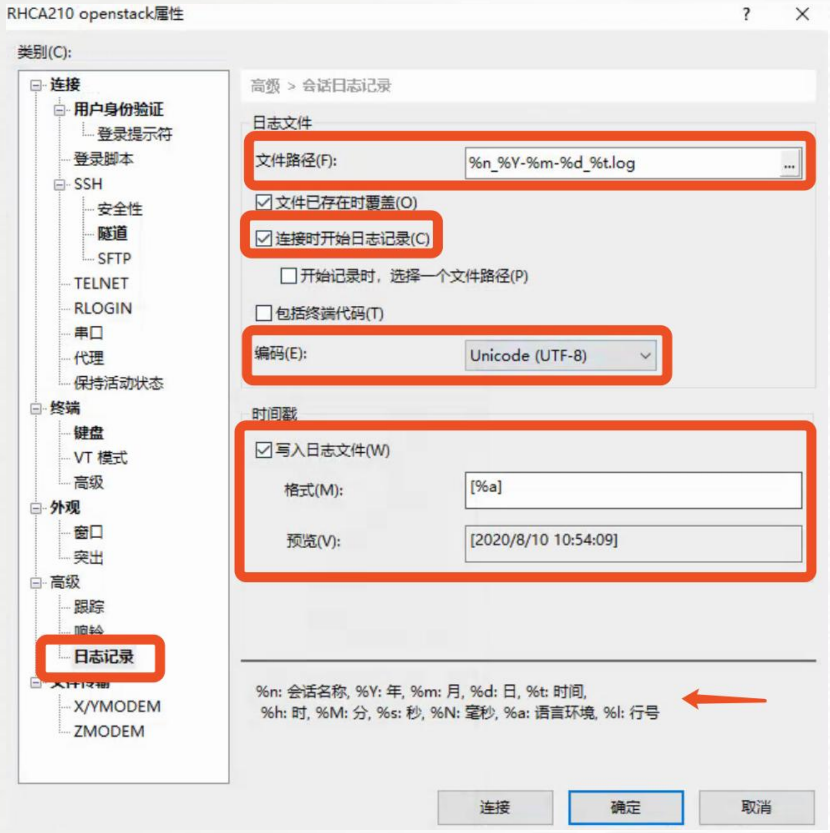


优点：
功能齐全，支持会话保存，支持图形转发

缺点：
非免费软件，完全功能需要安装多个套件

SSH客户端工具使用

配置SSH客户端的操作日志



右键点击会话，选择属性

在菜单中选择日志记录

右边界面可以调整以下内容：

日志路径

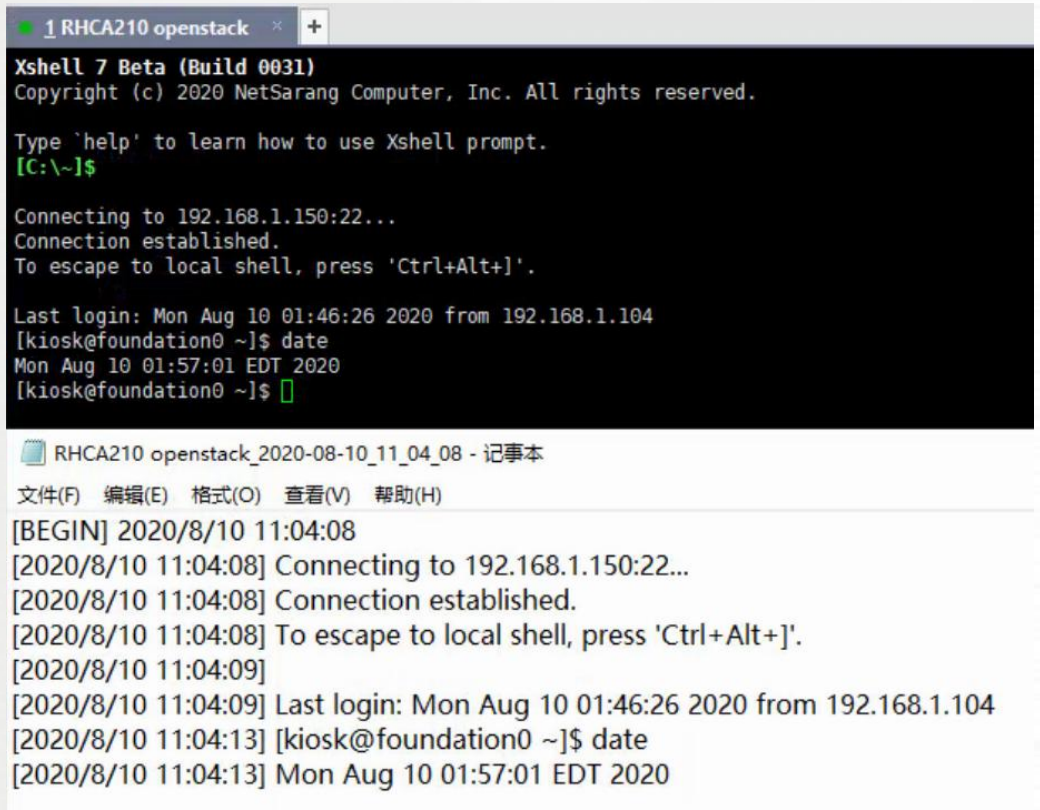
勾选连接时打开日志记录

日志文件编码

日志文件时间戳格式

SSH客户端工具使用

配置SSH客户端的操作日志



关于操作日志的注意事项：

1、Windows系统里日志路径尽量不要放在系统盘（C盘）

2、日志文件定期清理避免过多占用磁盘空间

SSH客户端工具使用

集群文件的上传和下载

常用工具介绍：

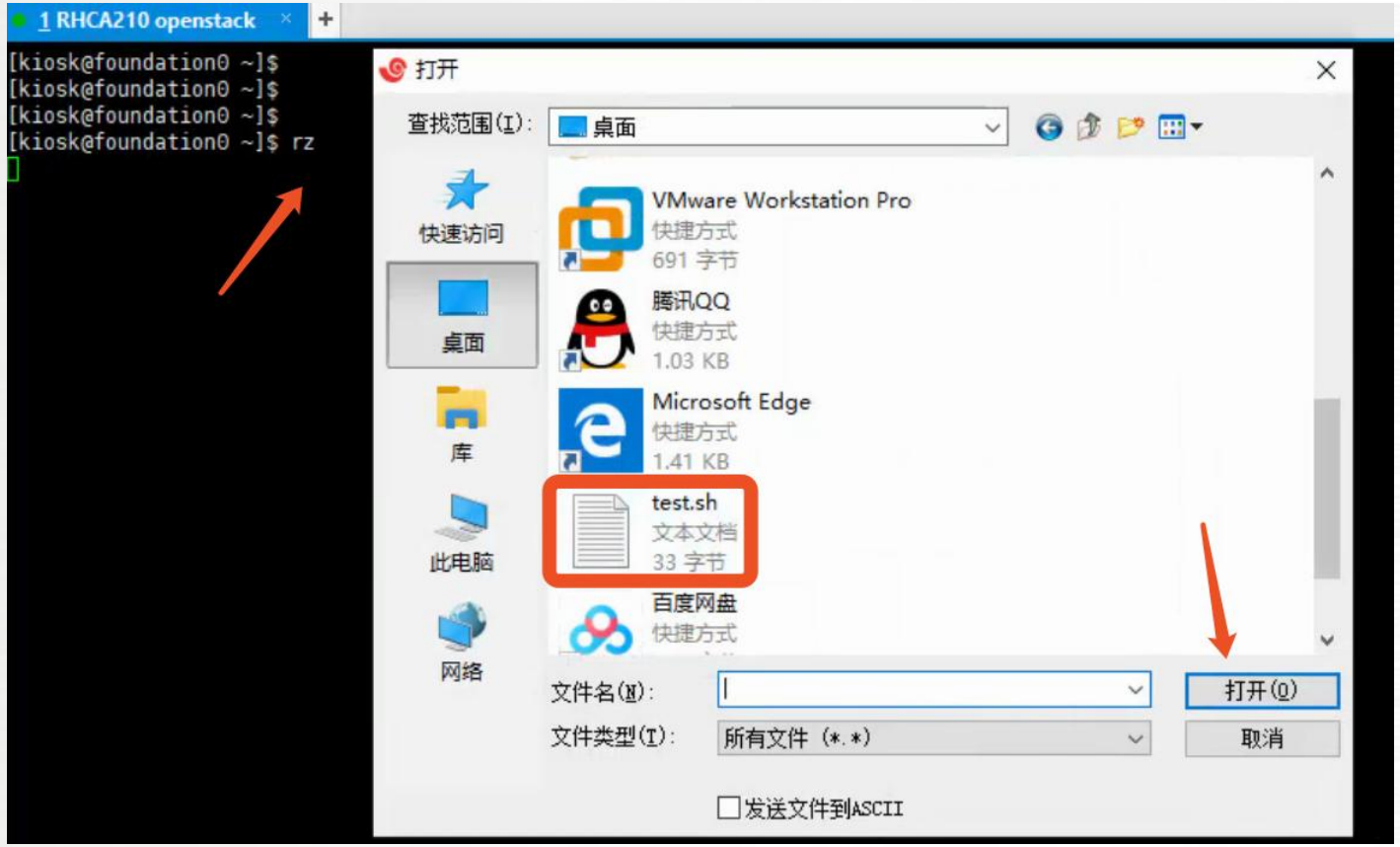
xftp：xshell组件，可以跟xshell结合，连接后能直接打开

filezilla：免费工具，功能强大

lrzsz： 系统自带工具，不用单独安装，使用方便
不是所有**SSH**客户端都支持
不支持超过**2GB**文件传输
不支持断点续传
仅支持文件操作，不支持目录操作
适用于小文件传输，比如源代码，文本文件，日志文件等

SSH客户端工具使用

rz sz使用方法介绍



上传文件

命令：**rz**

只能上传到系统当前路径

不建议跨节点进行上传

不支持上传目录

如果是**windows**环境的文件上传后

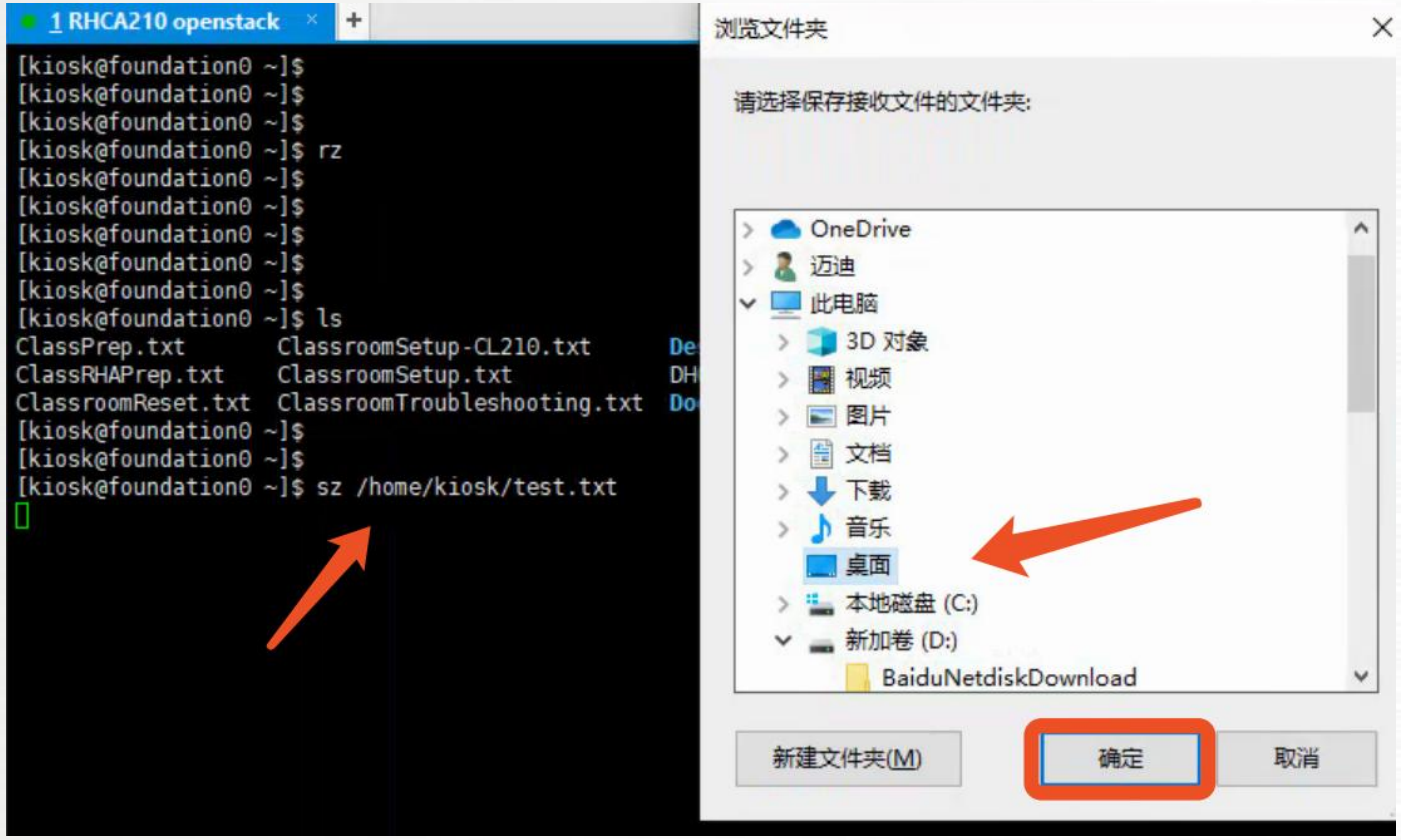
建议使用**dos2unix**进行格式转化，

避免因不同系统导致的格式问题

命令：**dos2unix [filename]**

SSH客户端工具使用

rz sz使用方法介绍



下载文件

命令：**sz [path to file]**

不支持下载目录

如果是下载到**windows**环境的文件建议下载前使用**unix2dos**进行转码操作，避免因不同格式导致的问题

命令：**unix2dos [path to file]**

Linux基本概念 SSH无密码访问

出于安全考虑用户登录操作系统需要做认证

认证的方式有很多种，其中比较典型的两种方式：

1、密码认证

2、秘钥认证

我们所说的SSH无密码访问并不是绕开操作系统的安全认证，而是改用一种比密码认证相对更加安全的方式秘钥认证进行访问

Linux基本概念 SSH无密码访问

秘钥文件的路径 在用户家目录下的.ssh目录中 ~/.ssh

```
[sugon@admin2 .ssh]$ ls -ald /public/home/sugon/.ssh/
drwx------ 2 sugon users 8192 4月  24 14:35 /public/home/sugon/.ssh/
```

注意目录的权限是700，也就是说除了属主本身，任何其他用户都没有任何权限访问

目录中包含的文件及其权限

```
[sugon@admin2 .ssh]$ ls -al /public/home/sugon/.ssh/
总用量 27
drwx------  2 sugon users 8192 4月  24 14:35 .
drwx------ 21 sugon users 8192 8月  11 10:04 ..
-rw-------  1 sugon users  395 4月  23 11:32 authorized_keys
-rw-------  1 sugon users   25 4月  23 11:32 config
-rw-------  1 sugon users 1679 4月  23 11:32 id_rsa
-rw-r--r--  1 sugon users  395 4月  23 11:32 id_rsa.pub
-rw-r--r--  1 sugon users 7427 6月  29 16:28 known_hosts
```

Linux基本概念 SSH无密码访问

authorized_keys	记录公钥文件的信息
id_rsa.pub	公钥文件
id_rsa	私钥文件

```
[root@admin1 ~]# cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvgQelE2y+CjP+PpGLtQGyU2axh7B/LA18Kq6rsCI1Sr1EUWK
4ayhcIV2+BxY5NXyqu4jOxyfIGs/N6UNDcxfOosLIAL1hlO2Zde190MfniIrU9gs
NlA8ucAD4Lhcsu7zkKBkkXbIsaviG+tww0TNyDY3RumoWQiyHJuaRuMhGAU4FbpJ
KlGCO9IpD6AzUamQmk0FclghAmpesFTFcBnd3zt/2QtYVIJ/gsOhN0h24jtu/dYD
oOJdZqDQ2TetEBSNiY5VYGa88B8lIPrrd9mqnMxKMFM0x2UPyaAIoFsSOhX2PBzX
JPiPETOexrBfWEFQX/5N3IZfZg1G5MINGlqXCwIDAQABAoIBACL0bwupH3s3Rsru
f2qt5nGxrP81OUpawBF91sm+8W0Fcf+zY1EQiMJCSxhpXpD6KqH3HNQK99rm38JE
wpn/NqV2Xp2M7c3AKJWjkP3jgzV746yQ39E+AW3KPOxh/euMuZGpb9aym0jT7cxl
C2Fgjtp93r1UzntSdjF9Dmxr6/KW8gBOevHK1x3mbO28eUN3TqZUGKDggqUMNTOE
/TcvGPf/7jwSxr6OjndhGYx1/R5NPoXtfxBNO0utvzfAMdIkRufdMbWHyt0zGx6N
elA6+5u/SFnHTsCL74jj6NnS2PBpjSkFCe2WvMbC1Trc28xmmmg24DMe5+UQitTf
7yS1LIECgYEA5Fs3Zpcd11pfIAe/KmnYFkEpaES5O0MwO73+oHVJ0udlwlz5IbBS
yhoDagIeVL33ZG/uo+0V9o70c4AeR3FtUS/kKzANfGkHweH5mF9zIte46Wus8D5z
uh3VzjsCV7b7AJMB7E6cxoazDMUhQYytWkosRHhf3osCg8DlhhKxqZkCgYEA1QS8
Zz2h5XiweDW3GN/98KFydSb4r5vq7XEvkUVLtTNxCty1O4oVRYGPyClz/rd/Dg3n
RJG7ggDCufuC2vP4acndQ3E1Znk0bem3Pb1nzs3ZBjPslZFhHCa0nf15oBzJz7ld
yPoElns4zH3cxvaBTOp5sYYk6RLr2PBUtn79VEMCgYAtUy2b30K9C+L1qJP5Byks
waJGX92kNaHNOdjgYGdWxTcdbPbGCQlls+QZlnIOv+XquYzUXKqPVbCPfE6GP61D
AA55w//pQa+YfBHefJONxuOc8RJSXRL7ThCx7mR4pIkNhCBNlwznGrVPQ5FmQVXn
JfHDZWEh+tV08USdZ/h7gQKBgFP5EPg14JtgL1jD/BDPRADW9BLgsQoxEbRWeSTw
Rz8AL2DTzpHRBj+nKnICfVp9JZX5vyBVvrAp8pl0X4pGwLebPyE2AJIL17I4fubU
wpZQqWv7jhkL/Vcmnu7lHEwU7S146lKu1J4ppyNdvmmkhSY4DI8BzY0fy19ypkJm
3yCXAoGBAK5+WD6A5xjb6bAFvSlvJEk976O66RavwdSiW60PzQCtGWu28+BPeI9n
TguKpofhrqkv6FfWMug87ZwC3gc16REz9JVI5TTYIMKYa2I4zlhLgeJzOmVH42Y6
Ruc0ODRC4SWk/d0h5l0oz+8++ZH56KWY/bVKfqA2i4L6vI4zKPkA
-----END RSA PRIVATE KEY-----
```

```
[root@admin1 ~]# cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+BB6UTbL4KM/4+kYu1AbJTZrGHsH8sDXwqrquwIjVKvURRYrhrKFwhXb4
w2UDy5wAPguFyy7vOQoGSRdsixq+Ib63DDRM3INjdG6ahZCLIcm5pG4yEYBTgVukkqUYI70ikPoDNRqZCaTQVyWCECal6wVM
ZrzwHyUg+ut32aqczEowUzTHZQ/JoAigWxI6FfY8HNck+I8RM57GsF9YQVBf/k3chl9mDUbkwg0aWpcL root@admin1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDPrP3ptPuPsSA3t9FMZoofRpQwNEfZlO78xiXXHry8mEILkolJatiQmRXW
SQwCmTlO+FOMCLZAeeQhXymmOBIWtWiKGD6pqNxzwa483HrDJLrDbvOG+VL4ug20/u5z28pKR8c+7zrmlvqpqs1f7EuANn9I
haNY3AQzp6WQ2oxy7eJaNhyd8IkTGkcysQZwZ4K8//C856F6TrMvqMYG1RPCkDSWLvh6qozMahU9ItWP root@admin1
```

```
[root@admin1 ~]# cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+BB6UTbL4KM/4+kYu1AbJTZrGHsH8sDXwqrquwIjVKvURRYrhrKFwhX
w2UDy5wAPguFyy7vOQoGSRdsixq+Ib63DDRM3INjdG6ahZCLIcm5pG4yEYBTgVukkqUYI70ikPoDNRqZCaTQVyWCECal6v
ZrzwHyUg+ut32aqczEowUzTHZQ/JoAigWxI6FfY8HNck+I8RM57GsF9YQVBf/k3chl9mDUbkwg0aWpcL root@admin1
```

Linux基本概念 SSH无密码访问

SSH无密码访问 – 常见问题

场景：
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：
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

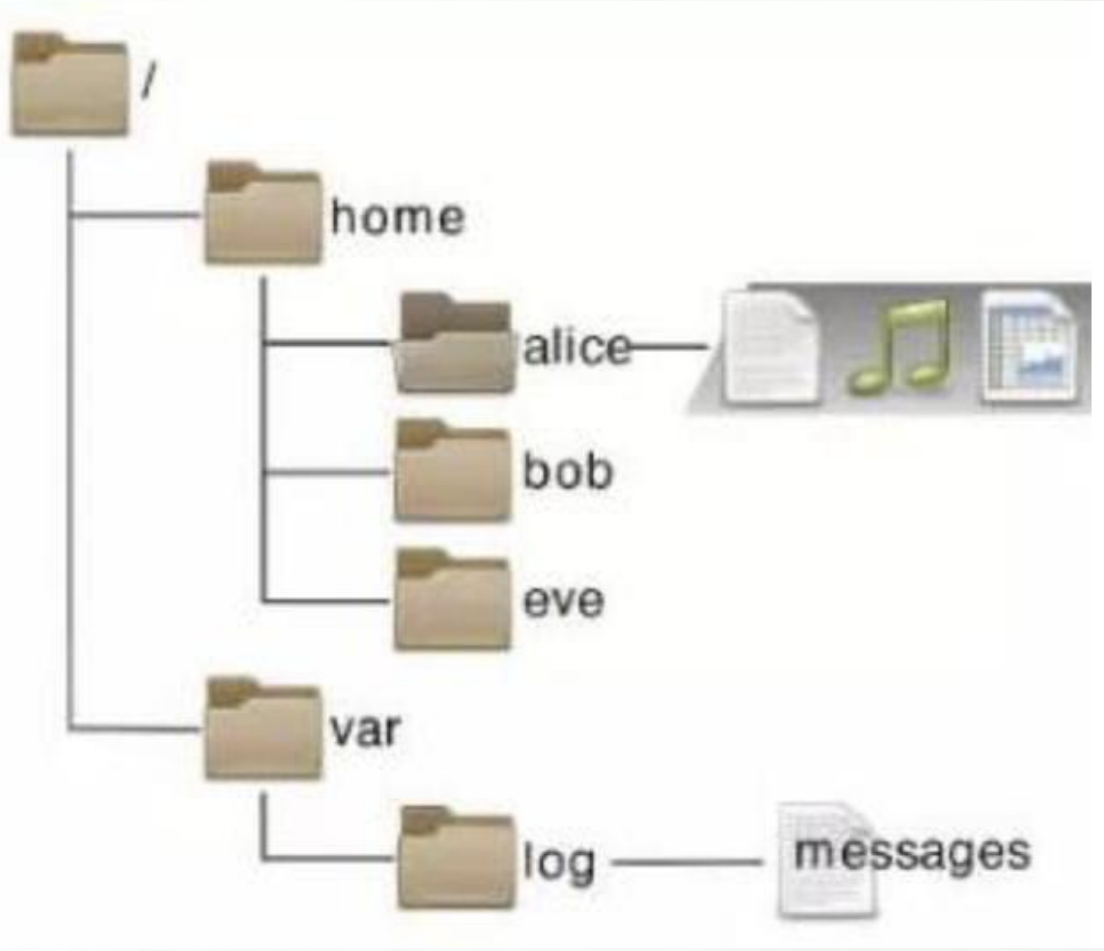
Linux基本概念 路径

什么是路径?
文件或者目录在系统中的唯一位置

Linux中路径分为两种:

绝对路径是完全限定名称从根目录开始(/)开始，指定到达唯一单个文件所需要遍历的每个子目录

相对路径仅指定从工作目录到达该文件所需的路径



Linux基本概念 路径

路径中的几个特殊符号：

/	Linux中文件系统的根路径
~	代表用户的家目录，比如/public/home/user
.	代表当前目录
..	代表上一级目录

tips:

＊ 在Linux操作系统中，大小写字母是敏感的
＊ 对于标准的Linux文件系统，文件路径名长度（包含所有/ 字符）不可超过4096字节
＊ 路径名中通过 / 分隔开的每一部分的长度不可超过255字节
＊ 文件名称可以使用任何UTF-8编码的Unicode字符，但 “/” 和Null字符除外。

Linux基本概念 环境变量

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

Linux基本概念 环境变量

根据作用域的不同，环境变量分为两种：

全局环境变量

全局环境变量对于shell会话和所有生成的子shell都是可见的

```
[root@admin1 ~]# export a=1
[root@admin1 ~]# bash
[root@admin1 ~]# echo $a
1
[root@admin1 ~]#
```

局部环境变量

局部变量则只对创建它们的 shell会话可见

```
[root@admin1 ~]# a=1
[root@admin1 ~]# bash
[root@admin1 ~]# echo $a

[root@admin1 ~]#
```

全局环境变量对那些所创建的子shell需要获取父shell信息的程序来说非常有用

Linux基本概念 环境变量

中科曙光 Sugon

常用环境变量介绍：

PATH	决定了shell将到哪些目录中寻找命令或程序
LIBRARY_PATH	用于在***程序编译期间***查找动态链接库时指定查找共享库的路径
LD_LIBRARY_PATH	用于在***程序加载运行期间***查找动态链接库时指定查找共享库的路径
INCLUDE	用于在***程序编译期间***查看头文件的路径
MANPATH	用于指定软件帮助文档的路径

Linux基本概念 环境变量

引用和查看环境变量

基本语法：

引用变量　　$变量名

查看变量值　echo $变量名　or　echo ${变量名}　or　echo "${变量名}"

```
[root@admin1 ~]# echo $PATH
/opt/clusconf/sbin:/opt/clusconf/bin:/public/software/module-4.4.1/bin:/opt/clusconf/sbin
```

Linux基本概念 环境变量

中科曙光 Sugon

设置环境变量

基本语法：变量名="变量值"

注意等号两边是没有空格的

定义局部变量

my_variable="Hello World"

定义全局变量

export my_variable="Hello World"

修改系统的全局环境变量

export PATH=/root/bin:$PATH

```
[root@admin1 ~]# echo $PATH
/usr/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin
[root@admin1 ~]# export PATH=/root/newbin
[root@admin1 ~]# echo $PATH
/root/newbin
[root@admin1 ~]# date
bash: date: command not found
```

```
[root@node1 ~]# export PATH=/root/newbin:$PATH
[root@node1 ~]# echo $PATH
/root/newbin:/usr/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin
[root@node1 ~]# date
Tue Aug 11 07:54:26 CST 2020
[root@node1 ~]# newcommand
Tue Aug 11 07:54:31 CST 2020
```

Linux基本概念 环境变量

灵活的加载并使用环境变量

方法1、source 环境变量脚本

```
[root@admin1 profile.d]# cat /public/software/profile.d/mpi_intelmpi-2017.4.239.sh
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
```

Linux基本概念 环境变量

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
```

Linux基本概念 环境变量

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
```

Linux基本概念 环境变量

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
```

Linux基本概念 环境变量

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.
```

Linux基本概念 环境变量

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

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
```

Linux基本概念 环境变量

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.
```

Linux基本概念 环境变量

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
```

Linux基本概念 环境变量

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

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

调度系统概述-基本概念

■ 资源（Resource）

- ✓ 作业运行过程中使用的可量化实体都是资源；
- ✓ 包括硬件资源（节点、内存、CPU 、GPU等）和软件资源（ License ）；

■ 集群（Cluster）

- ✓ 包含计算、存储、网络等各种资源实体且彼此联系的资源集合；
- ✓ 在物理上，一般由计算处理、互联通信、I/O 存储、操作系统、编译器、运行环境、开发工具等多个软硬件子系统组成；
- ✓ 节点是集群的基本组成单位，从角色上一般可以划分为管理节点、登陆节点、计算节点、存储节点等。

■ 作业（Job）

- ✓ 物理构成，一组关联的资源分配请求，以及一组关联的处理过程；
- ✓ 交互方式，可以分为交互式作业和非交互式作业；
- ✓ 资源使用，可以分为串行作业和并行作业；

调度系统概述-基本概念

■ 分区（Partition）

- ✓ 带名称的作业容器；
- ✓ 用户访问控制；
- ✓ 资源使用限制；

■ 作业调度系统（Job Schedule System）

- ✓ 负责监控和管理集群中资源和作业的软件系统；
- ✓ 通常由资源管理器、调度器、任务执行器，以及用户命令和API组成；

调度系统概述-主要作用

- **单一系统映像**
  - ✓ 解决集群结构松散问题；
  - ✓ 统一用户接口，使用简化；
- **系统资源整合**
  - ✓ 管理异构资源和异构系统；
- **多任务管理**
  - ✓ 统一管理任务，避免冲突；
- **资源访问控制**
  - ✓ 基于策略的资源访问控制；

调度系统是面向集群的操作系统。

调度系统命令介绍

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步（通常是MPI作业）。
	sattach	连接正在运行的作业步的标准输出和错误等。
	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-查询命令sinfo

sinfo命令参数（1/3）	
-a, --all	查看所有分区信息（含隐藏分区），对应环境变量SINFO_ALL。
-d, --dead	查看dead状态（通信异常）的节点和分区的信息，与-r参数对应。
-h, --noheader	打印分区（或节点）信息时不打印表头。
-i <seconds>, --iterate=<seconds>	定时打印状态信息。
-l, --long	打印分区（或节点）的详细信息。
-n <nodes>, --nodes=<nodes>	查看指定节点的信息。
-N, --Node	以面向节点（默认为分区）的格式打印信息，每行一个节点。
-p <partition>, --partition=<partition>	查看指定分区的状态，对应环境变量SINFO_PARTITION。

查询信息-查询命令sinfo

sinfo命令参数介绍（2/3）	
-r, --responding	查看计算节点（内部通信）正常的节点和分区的状态，与-d参数对应。
-s, --summarize	查看分区中节点状态的摘要信息。
-S <sort_list>, --sort=<sort_list>	设置打印状态信息时排序的规则，如sinfo -S “#P，-t”。参数可以通过设置环境变量SINFO_SORT实现。
-t <states> , --states=<states>	查询指定节点状态的分区或节点的信息，常见状态包括：**ALLOC/ALLOCTED,COMP/COMPLETING,DOWN,DRAIN/DRAINED/DRAINING,ERR/ERROR,FAIL,IDLE,MIX/MIXED,UNK/UNKNOWN**。
--federation	显示所有集群的分区（或节点）的信息，对应变量SINFO_FEDERATION。
--local	仅显示当前集群的分区（或节点）的信息，对应环境变量SINFO_LOCAL，优先级高于--federation。
-M, --clusters=<string>	显示多个集群的分区（或节点）的 信息，对应环境变量SLURM_CLUSTERS。取值“all”代表所有集群。

查询信息-查询命令sinfo

Sinfo参数介绍（3/3）	
-o <output_format>, --format= <output_format>	按照指定的字段和格式显示信息，对应变量**SINFO_FORMAT**。 格式为“%[[.]size]<type> %[[.]size]<type> …”。 其中，点号（.）表示右对齐，size表示字段长度，type为代表特定字段的字符（或字符串）。示例如下： sinfo -o %all 以竖线分隔的形式显示所有字段。 sinfo -o "%9P %.5a %.10l %.6D %.6t %N" 显示内容：分区名-分区状态 -最大运行时间-节点数-节点状态-节点列表。
-O <output_format>, --Format= <output_format>	按照指定的格式显示状态信息。 格式为“type[:[.]size],type[:[.]size],…”。 其中，type为代表特定字段的字段名，点号（.）表示右对齐，size表示字段长度。示例如下： sinfo -O all sinfo -O Partition:9,available:.6,time:.11,nodes:.6,statecompact:.6,nodelist:.12

查询信息-查询命令sinfo

格式化字段说明（1/5）		
字段名（-O）	格式化（-o）	字段说明
allocmem		节点已分配内存
allocnodes	%S	队列提交节点
available	%a	分区状态
cluster	%V	集群名
cpus	%c	节点CPU核数
cpusload	%O	节点负载
freemem	%e	节点空闲内存
cpusstate	%C	CPU统计A/I/O/T
cores	%Y	Cores/Socket
defaulttime	%L	默认运行时间

格式化字段说明（2/5）		
字段名（-O）	格式化（-o）	字段说明
disk	%d	临时盘大小MB
features	%f	节点属性
features_act	%b	Active features
groups	%g	分区用户组
gres	%G	一般性资源
maxcpuspernode	%B	单节点最大核数
memory	%m	节点总内存
nodes	%D	节点数
nodeaddr	%o	节点地址
nodeai	%A	节点统计A/I

查询信息-查询命令sinfo

格式化字段说明（3/5）		
字段名（-O）	格式化（-o）	字段说明
nodeaiot	%F	节点统计AIOT
nodehost	%n	节点主机名列表
nodelist	%N	节点名列表
oversubscribe	%h	资源超额认购
partition	%P	分区名
partitionname	%R	分区名
port		节点TCP端口
preemptmode	%M	分区抢占模式
priorityjobfactor	%I (i)	分区优先级因子
prioritytier	%p	分区调度优先级

格式化字段说明（4/5）		
字段名（-O）	格式化（-o）	字段说明
reason	%E	节点不可用原因
root	%r	仅root可用资源
size	%s	作业节点数限制
statecompact	%t	节点状态，缩写
statelong	%T	节点状态
sockets	%X	节点Socket数
socketcorethread	%z	CPU配置，S:C:T
time	%l (L)	最大运行时间
timestamp	%H	节点不可用时间
threads	%Z	单核超线程数

查询信息-查询命令sinfo

格式化字段说明（5/5）		
字段名（-O）	格式化（-o）	字段说明
user	%u	节点不可用状态的配置用户
userlong	%U	同user，额外显示用户ID
version	%v	软件版本
weight	%w	节点优先级
all	%all	全体字段，竖线分隔

查询信息-查看分区

中科曙光 Sugon

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 ~]$
```

查询信息-查看分区

示例3: 查看分区的摘要信息（--summarize）。

sinfo --summarize

```
[sghpc2@login04 ~]$sinfo --summarize
PARTITION AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
serial       up   infinite        1/19/0/20  cmac[0011-0030]
normal       up   infinite   91/1407/6/1504  cmac[0035-1538]
operation    up   infinite   91/1407/6/1504  cmac[0035-1538]
sgtest       up   infinite     74/723/4/801  cmac[0035-0835]
[sghpc2@login04 ~]$
```

示例4: 自定义分区摘要信息显示，添加CPU状态分布信息。

sinfo -o "%9P %.5a %.10l %.16F %.24C %N"

```
[root@login_a04 ~]#  sinfo -o "%9P %.5a %.10l %.16F %.24C  %N"
PARTITION AVAIL  TIMELIMIT   NODES(A/I/O/T)            CPUS(A/I/O/T)  NODELIST
serial       up   infinite        16/8/0/24           20/748/0/768  cmac[0011-0034]
serial_op    up   infinite        16/8/0/24           20/748/0/768  cmac[0011-0034]
largemem     up   infinite     194/32/0/226       6168/1064/0/7232  cmac[0035-0260]
normal       up   infinite  1091/413/0/1504     34806/13322/0/48128  cmac[0035-1538]
operation    up   infinite  1091/413/0/1504     34806/13322/0/48128  cmac[0035-1538]
[root@login_a04 ~]#
```

查询信息-查看分区

中科曙光 Sugon

示例5: 查询特定分区特定状态的节点状态信息。
sinfo -p P1,P2 -t T1,T2

```
[sghpc2@login04 ~]$sinfo -p operation,normal -t idle,alloc
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal       up   infinite      1   drng cmac0533
normal       up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
normal       up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
operation    up   infinite      1   drng cmac0533
operation    up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
operation    up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
[sghpc2@login04 ~]$
```

查询作业 squeue

squeue:查询排队和运行状态的作业

参数	解释
-A, --account=account(s)	查询指定账号的作业，默认全部账号下的作业
-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
-u, --user=user_name(s)	逗号分隔指定用户的作业

```
[zlei@login01 job_example]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            308712    normal     test     zlei  R       0:01      1 a3110n01
```

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

SLURM队列-scontrol show

查询指定队列命令:
scontrol show partition <name>

```
[root@gpunode2 ~]# scontrol show partition debug
PartitionName=debug
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=YES QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=gpunode[1,2]
   PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=80 TotalNodes=2 SelectTypeParameters=NONE
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

SLURM节点-scontrol show

查询指定节点:
scontrol show node <name>

```
[root@gpunode2 ~]# scontrol show node gpunode1
NodeName=gpunode1 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUErr=0 CPUTot=40 CPULoad=0.37
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=10.0.35.187 NodeHostName=gpunode1 Version=17.02
   OS=Linux RealMemory=30000 AllocMem=0 FreeMem=6194 Sockets=40 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=debug,nvidia2
   BootTime=2017-08-07T11:49:16 SlurmdStartTime=2017-08-22T15:23:13
   CfgTRES=cpu=40,mem=30000M
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

SLURM作业-scontrol show

查询指定作业:
scontrol show job

```
[root@gv11 ~]# scontrol show job
JobId=119 JobName=sleep
   UserId=sugon(1000) GroupId=docker(1000) MCS_label=N/A
   Priority=4294901738 Nice=0 Account=physics QOS=testpartitionqos
   JobState=FAILED Reason=NonZeroExitCode Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=0 Reboot=0 ExitCode=1:0
   RunTime=00:00:00 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2017-08-24T20:23:07 EligibleTime=2017-08-24T20:23:07
   StartTime=2017-08-24T20:23:07 EndTime=2017-08-24T20:23:07 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=nvidia AllocNode:Sid=gv11:4295
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gv11
   BatchHost=gv11
   NumNodes=1 NumCPUs=1 NumTasks=0 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,node=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=sleep
   WorkDir=/home/sugon
   Power=
```

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

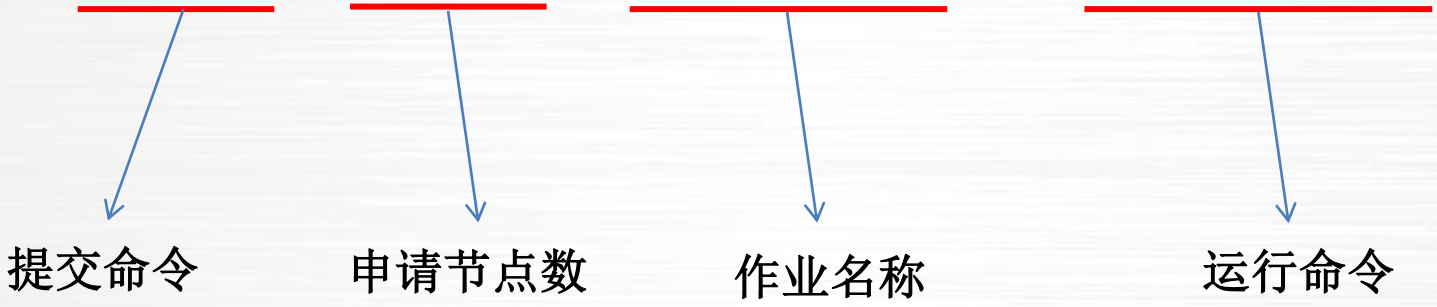
SLURM-提交作业参数

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

```
[sugon@gpunode1 ~]$ sbatch sleep.job      //sbatch 只接收脚本
Submitted batch job 19

[sugon@gpunode1 ~]$ 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。

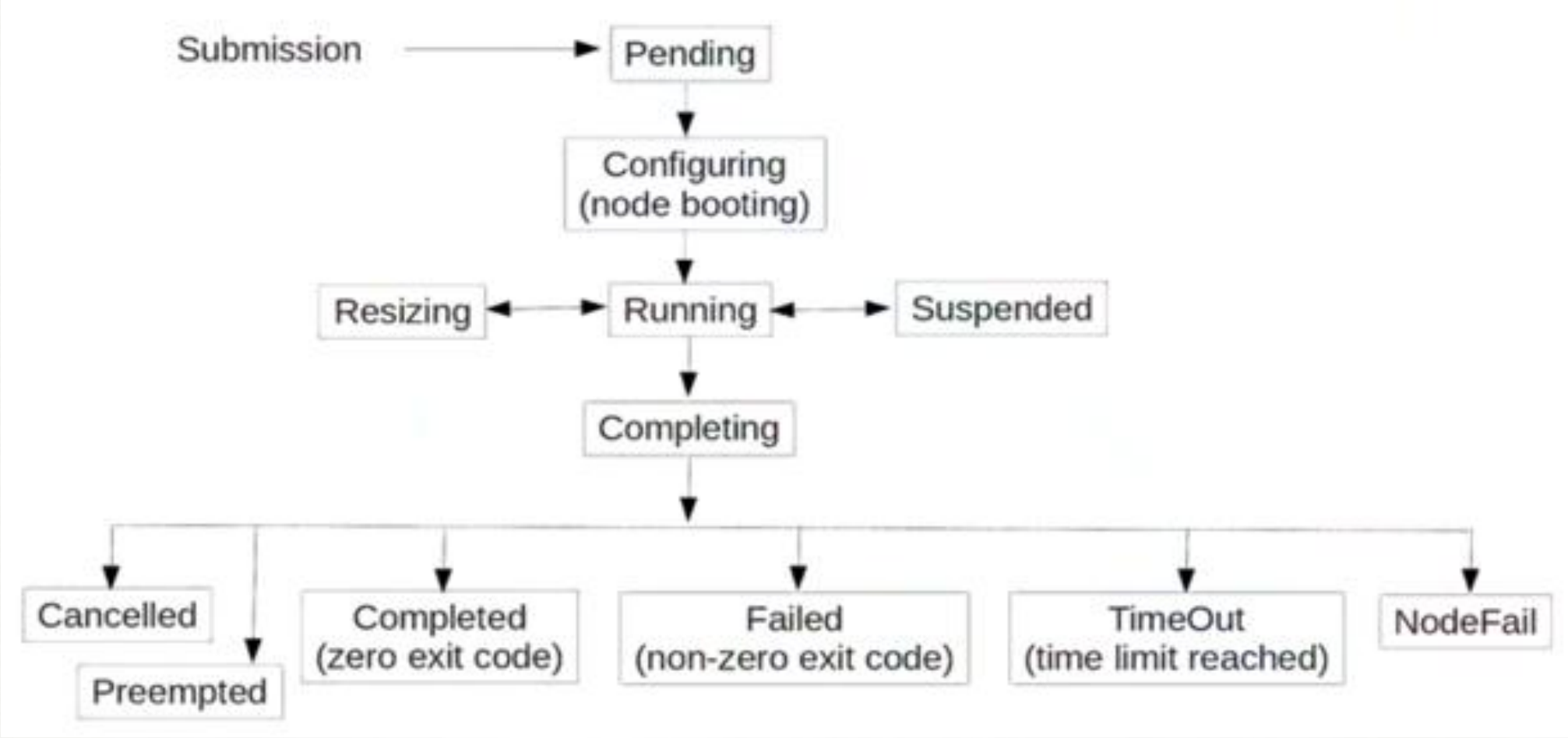
常用的逻辑关系包括：
after（依赖作业开始运行时）
afterok（依赖作业正常结束时）
afterany（依赖作业均完成）
afternotok（依赖作业非正常结束）

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
```

作业状态变化


Submission
Pending
Configuring
(node booting)
Resizing
Running
Suspended
Completing
Cancelled
Preempted
Completed
(zero exit code)
Failed
(non-zero exit code)
TimeOut
(time limit reached)
NodeFail

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

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：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