

澳门科技大学建设项目 高性能计算平台 用户使用手册

曙光星云信息产业（北京）有限公司

二零一七年三月

法律条款

1. 对本文档的任何使用均被视为完全理解并接受本文档列举的所有法律条款。
2. 本文档的所有权利归作者所有，作者保留所有权利。
3. 若本文档的保密级别为“公开发布”，允许学术使用，未经作者书面授权，禁止商业使用。若本文档的保密级别为“内部资料”，仅授权曙光星云信息产业（北京）有限公司的在职员工使用。若本文档的保密级别为“核心资料”，未经作者书面授权，禁止任何形式的使用。
4. 对本文档的使用形式包括但不限于存储、出版、复制、传播、展示、引用、编辑。使用过程中不得对本文档作任何增减编辑，引用时需注明出处。商业使用是指在商业活动中或有商业目的活动中的使用。学术使用是指以技术交流或学术研究为目的的使用。
5. 实施任何侵权行为的法人或自然人都必须向作者支付赔偿金，赔偿金计算方法为：
赔偿金 = 涉案人次 × 涉案时长（天） × 涉案文档份数 × 受众人次 × 基数 × 1 元人民币，
涉案人次、涉案时长、涉案文档份数、受众人次小于 1 时，按 1 计算。若保密级别为“公开发布”，基数为 100；若保密级别为“内部资料”，基数为 1000；若保密级别为“核心资料”，基数为 10000。
6. 对举报侵权行为、提供有价值证据的自然人或法人，作者承诺奖励案件实际赔偿金的 50%。
7. 作者不保证文档内容的正确性。对文档的使用后果，作者不承担任何责任。
8. 涉及本文档的法律纠纷由作者所在地法院裁决。
9. 本文档所列举法律条款的最终解释权归作者所有。

目 录

目 录	III
1. 集群系统概述.....	1
2. 基础软件环境.....	2
2.1 基础软件环境.....	2
2.2 应用软件环境.....	3
3. 作业调度系统.....	4
3.1 作业调度系统概述.....	4
3.2 计算队列分布.....	4
4. 用户上机操作.....	4
4.1 用户登录.....	4
4.2 文件传输.....	5
4.2.1 基于 GridView Web 界面进行文件传输.....	5
4.2.2 FTP 服务传输文件.....	6
4.3 提交作业.....	7
4.3.1 用 GridView web 界面提交作业.....	7
4.3.2 用命令行方式提交作业.....	13
4.4 作业管理.....	16
4.4.1 用 GridView Web 界面管理作业.....	16
4.4.2 用命令行方式管理作业.....	20
4.5 源码编译.....	22
4.5.1 c/fortran 编译器汇总.....	22
4.5.2 mpi 编译器汇总.....	22
4.5.3 环境变量的选取.....	23
附录 Linux 常用命令.....	24
1.1 浏览目录命令.....	24
1.2 浏览文件命令.....	25
1.3 目录操作命令.....	26
1.4 文件操作命令.....	26
1.5 查找类命令.....	27
1.6 用法帮助命令.....	28
1.7 打包、解包, 压缩、解压缩命令.....	28
1.8 时间相关命令.....	29
1.9 系统信息类命令.....	30
1.10 网络通讯类命令.....	30
1.11 软件包管理命令.....	31
1.12 编辑器命令 (vim)	32
1.13 用户管理命令.....	33
1.14 用户组管理命令.....	33
1.15 输入/输出重定向与管道命令.....	34

1. 集群系统概述

澳门科技大学 HPC 集群系统整体计算能力理论峰值为 171.568Tflops。集群系统采用曙光自主设计的 HPC 集群系统，架构主体设计包括计算系统、存储系统、网络系统和管理系统。计算系统包含刀片服务器(160 台)，2 台八路胖节点服务器和 2 台 GPU 节点服务器。存储系统采用 1 套曙光 ParaStor 并行文件系统，总存储裸容量达 360TB。管理系统配置 2 台集群登录节点和 2 台集群管理节点。计算高速互连网络采用 12 台 36 口 FDR(56Gb/s)交换机组成 3D-Torus 互连，并行配置 1 套千兆集群管理网络和 1 套集群监控网络。

其中 160 台双路计算刀片，配置目前 x86 体系中计算速度较快的 Intel Xeon E5 2600v3 系列处理器，并采用 2.5GHz 主频 2680 v3 12 核处理器，配置 64GB 内存，主要负责超高计算量需求；2 台 GPU 节点加速服务器，每个节点配置 2 个 Intel Xeon 2680 v3 12 核处理器，4 块 NVIDIA Tesla K40，64GB 内存；2 台曙光 8 路服务器，每节点配置 8 颗 Intel Xeon E7-8857 v2 12c 处理器，2T 大内存，满足大内存业务的计算需求。

共享存储采用专门的聚合带宽高达 4G/s 的并行存储系统，能同时满足高速数据传输和数据安全的需求。

曙光 Gridview 综合管理系统实现集群资源监控与管理，以及集成 HPC 云计算平台，满足在线运维，用户集中管理，以及大规模作业管理的需求。

本文主要介绍在集群系统上，普通用户提交作业的方法，及注意事项。

2. 基础软件环境

2.1 基础软件环境

集群系统所有登录、管理节点和计算节点的操作系统，统一部署为 64 位 CentOS6.6，提供标准的 64 位 Linux 操作环境。

集群系统部署的编译器既包含 Linux 操作系统自带的 GNU 的 c/c++/fortran 编译器，也提供了 Intel/MPI 多个版本的 c/c++/fortran 的编译器。

集群系统还部署了 Lapack/Scalapack /fftw/mkl 等多个版本的数学库。

此外，集群系统还部署了 openmpi/intelmpi/等多个版本的 mpi 库，可以支持 OpenMP 和 MPI 两种并行方式。

基础软件安装路径详见下表：

分类	软件名	版本	安装路径	环境变量
GPU 节点	CUDA	8.0	/usr/local/cuda	/etc/profile.d/cuda-8.0_env.sh
IB 驱动	ofed	3.4.0	默认	无
集群管理软件	clusterm	2.0.0	/public/sourcecode	无
	clusconf	2.0.1	/opt/clusconf	自动添加
	Gridview	4.0.1	/opt/gridview	自动添加
编译器	GNU	4.4.7	/usr	系统自带
	GNU	4.9.1	/public/software/compiler/gnu	支持 module 管理
	Intel	2016.2.181	/public/software/compiler/intel	
MPI 并行环境	OpenMPI	1.6.5	/public/software/mpi	
	MPICH	3.2		
	IntelMPI	5.1.3		
	OpenMPI	2.0.1		
	MVAPICH	2.3b		
数学库	MKL	11.0	/public/software/mathlib	支持 module 管理
	FFTW	3.3.4		
	NETCDF	4.4.0		
	Lapack	3.4.2		
测试软件	HPL	2.2	/public/software/benchmark	支持 module 管理

	IMB	3.2.4		
	Stream	5.10		
	IOZone	3.430		

2.2 应用软件环境

软件名	版本	安装路径	环境变量
R	3.2.1	/public/software/apps/R	支持 module 管理
Python	2.7.12	/public/software/apps/python	
Gromacs	5.1.3	/public/software/apps/gromacs/	
NAMD	2.10	/public/software/apps/namd	
WRF	3.6.1	/public/software/apps/wrf	/public/software/profile.d/apps_wrf-3.6.1.sh

特别说明：

module 管理工具可以用于加载所安装软件的环境变量，常用命令：

module avail 查看安装的软件

module load <module> 加载环境变量

module load <module> 卸载环境变量

3. 作业调度系统

3.1 作业调度系统概述

为了有效利用澳门科技大学超算中心的集群系统计算能力，合理分配计算资源，采用基于pbs/maui的作业调度系统。这样，用户仅需把注意力放在作业内容上，而计算节点和核心数的分配都由作业调度系统完成，既方便用户使用，也提高集群系统整体的利用效率。

澳门科技大学超算中心的集群系统既提供基于web界面的作业提交和管理模式，也提供基于命令行的作业提交和管理模式，方便不同类型的作业和用户的使用要求。

3.2 计算队列分布

目前，澳门科技大学超算中心按计算节点种类不同分成3个开放队列。

parallel队列为并行任务队列，包括155个刀片计算节点，每个节点可以提供24个核心；该队列共可以提供3720个cpu计算核心。

serial队列为串行任务队列，包括5个刀片计算节点，每个节点可以提供24个核心；该队列共可以提供120个cpu计算核心。

gpu队列，包括2台GPU计算节点，每个节点配置两颗12核心处理器，4块NVIDIA Tesla K40。

fat队列，包括2台八路胖节点，每节点提供96核心处理器，2T内存。

作业完全按照提交时间的先后顺序运行，实行资源预约机制，不限制作业运行时间。

4. 用户上机操作

在澳门科技大学超算中心集群系统上，用户提交作业的基本步骤如下所示：

- 准备模型数据文件和作业脚本文件。
- 用户账号创建和登录
- 传输文件到集群系统登录节点上
- 提交作业
- 作业管理
- 作业结果查看，将结果文件下载到本地硬盘

其中，准备模型数据文件和作业脚本文件，用户可以在本地完成。用户账号的创建需要联系集群系统管理员。本文主要介绍其余几个步骤的进行方法。

4.1 用户登录

用户可以使用注册的用户名和密码来登陆GridView Web界面，或者用ssh的方式采用客户端软件来登录节点：



用户通过客户端软件 ssh 登录到节点后，会自动登录到自身的家目录：`/public/home/用户名`；

在该路径下系统已经自动生成三个文件夹：`software`，`sourcecode`，和 `app`。建议用户将使用的软件装到 `software` 文件夹，将源代码相关文件放到 `sourcecode` 目录下，将用户的算例等文件放到 `app` 文件夹下。其中 `software` 目录下内置 `profile.d` 文件夹，其中包含集群系统的所有环境变量文件。

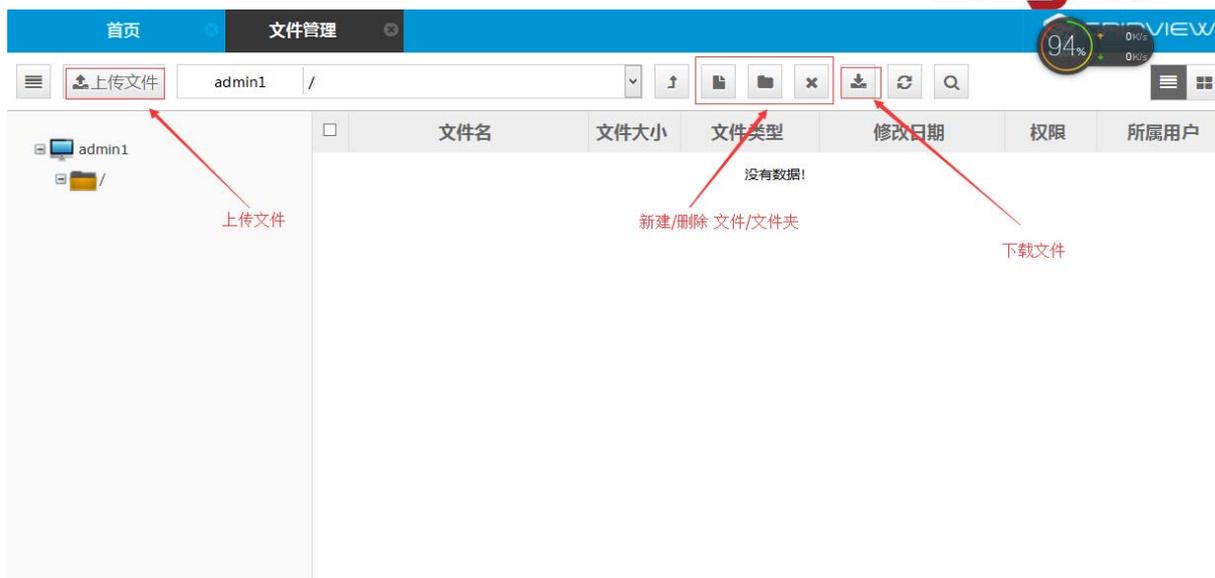
4.2 文件传输

4.2.1 基于 GridView Web 界面进行文件传输

用户用用户名和密码登录到 GridView 的 web 界面后，可以选择如下操作，进入用户的文件管理界面：



在该界面下可以进行上传文件/下载文件/以及新建文件夹，删除文件夹等，如下图所示：

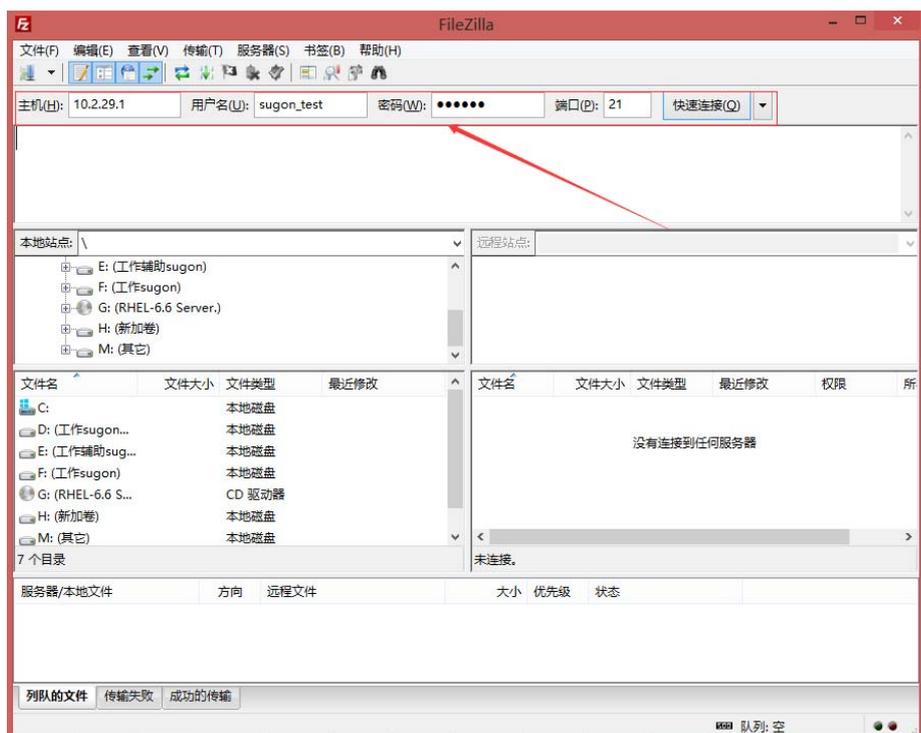


4.2.2 FTP 服务传输文件

用户可以通过 FTP 服务进行文件的传输。登录时，需要使用主机名和主机密码。注意 FTP 服务是 24 小时连续服务的。

在 windows 系统下，推荐用户使用免费的 ftp 软件 Filezilla，下载的官方网址为：
<http://sourceforge.net/projects/filezilla/>

安装 Filezilla 软件后，打开软件界面，输入主机地址 10.20.102.11 以及用户名和密码，完成后单击快速链接，或者回车键既可以登陆 ftp 站点。在文件菜单中，可以选择“站点管理器”保存 ftp 地址，用户名和密码。下次登录时只需要从“站点管理器”中选择相应的 FTP 地址即可以直接登陆。



用户也可以选择使用其他 FTP 软件如 cuteftp, flashfxp 等进行登录。当需要下载或上传的文件数量较多时, 建议用 tar 命令将文件进行打包后下载或上传, 加快文件传输速度, 减少文件传输的出错概率。

注意, 因为 windows 操作系统和 linux 操作系统对于个别字符的处理方式不同, 上传的文本文件在 linux 系统下, 需要用 dos2unix 命令先处理一下, 然后再继续使用, 否则可能导致作业提交出错。

4.3 提交作业

在集群系统上提交作业, 用户有两种方式可供选择。一种是通过 GridView Web 界面提交, 一种是通过命令行的方式, 用 pbs 命令提交。

用户严禁使用任何前台或者后台方式直接运行程序, 所有计算任务必须通过 GridView web 界面提交或命令行用 PBS 提交。否则, 后果自负。

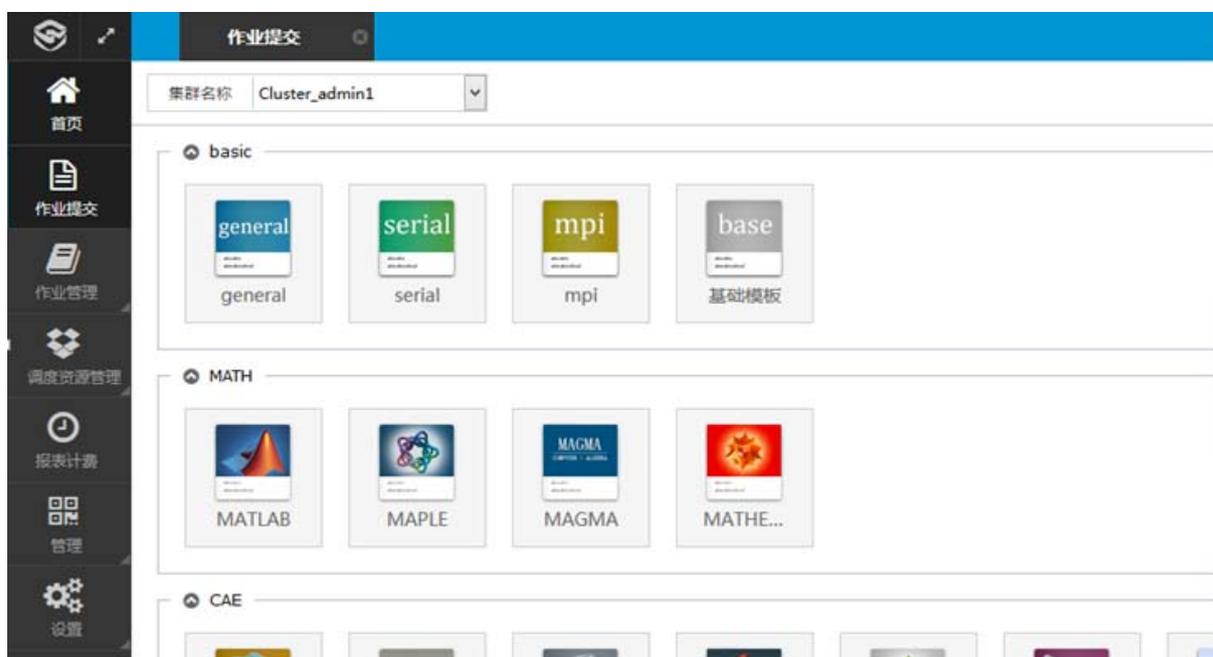
具体提交作业方法如下所示。

4.3.1 用 GridView web 界面提交作业

用户通过 GridView web 界面提交作业, 登陆方式: [http:// 10.50.18.1:6080](http://10.50.18.1:6080)

用户名: *username* 密码: *passwd*

系统缺省可用 portal 有四种, general/serial/mpi/基础模板, 如下图所示:



其中:

- general 为通用模板, 既可提交并行作业, 也可以提交串行作业。
- Serial 为串行作业使用模板, 可以提交串行和支持 openmp 的作业, 串行作业仅能一个核心内运行, openmp 仅能在一个单独的节点上运行。

- Mpi 为并行作业使用模板。可以提交并行作业，作业可以在多个节点上并行执行。
- 基础模板同样可以提交并行作业和串行作业。

4.3.1.1 基础模板作业提交

用户选择基础模板后，出现如下界面：

集群名称	Cluster_mgmt34	▼
作业名称	STDIN	
队列	score	▼
工作路径	/home/sugon01	Q
提交方式	命令行方式	▼
*命令行方式	sleep 100	
*所需执行时间	0	小时 10 分钟
节点选择	节点数	▼
节点数	1	
每个节点处理器数	1	
每个节点GPU数	请输入大于0的正整数	

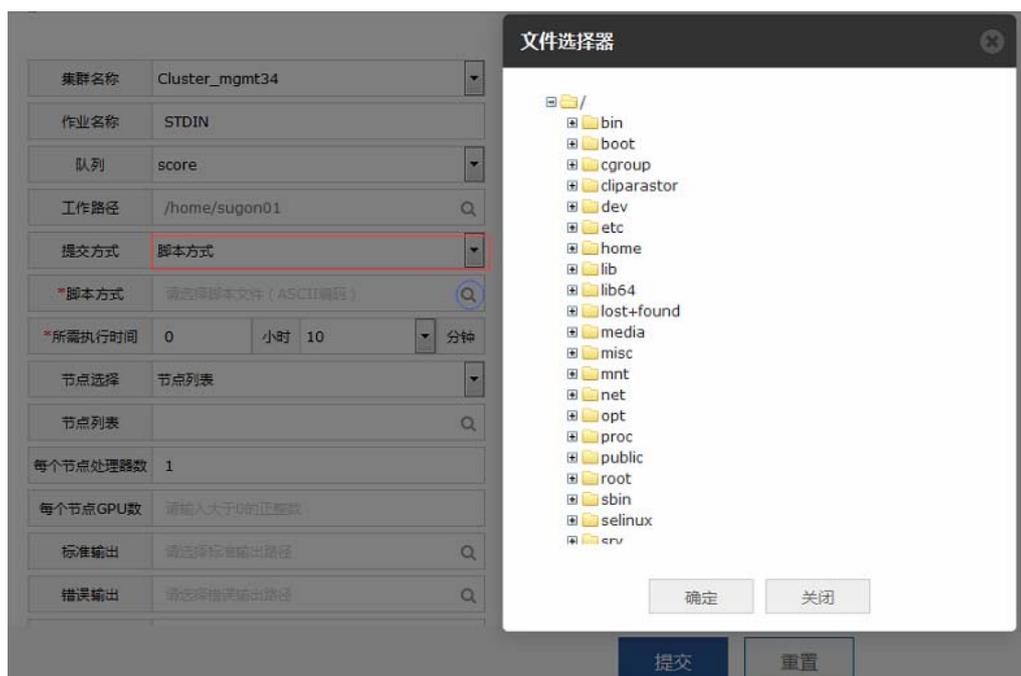
提交
重置

基础模板作业提交时，用户可以选择默认的集群（目前，大工超算中心集群系统为单集群），填写作业名称，或使用缺省名。用户可以根据自身作业的需要，选择不同的作业队列，关于队列的详细说明参见 3.2 节计算队列分布。工作路径默认为用户的家目录，用户可以根据需要选择。作业提交方式有命令行方式和脚本方式两种。用户仅能选择其中的一种。

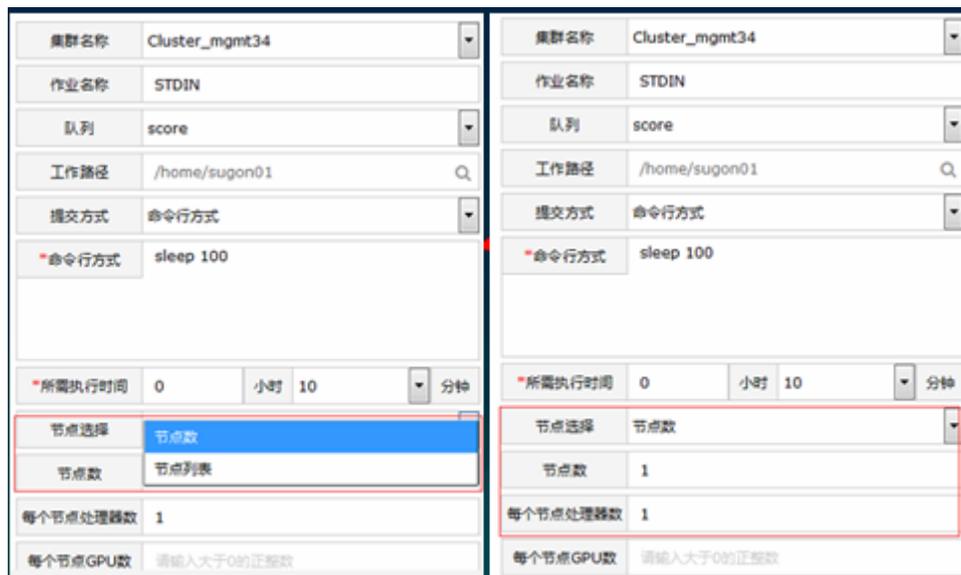
集群名称	Cluster_mgmt34	▼	队列	score	▼
作业名称	STDIN		工作路径	score	▼
队列	score	▼	snode	▼	
工作路径	/home/sugon01	Q	提交方式	命令行方式	▼
提交方式	命令行方式	▼	*命令行方式	命令行方式	▼
*命令行方式	sleep 100		脚本方式	脚本方式	▼
*所需执行时间	0	小时 10 分钟			
节点选择	节点数	▼			
节点数	1				
每个节点处理器数	1				
每个节点GPU数	请输入大于0的正整数				

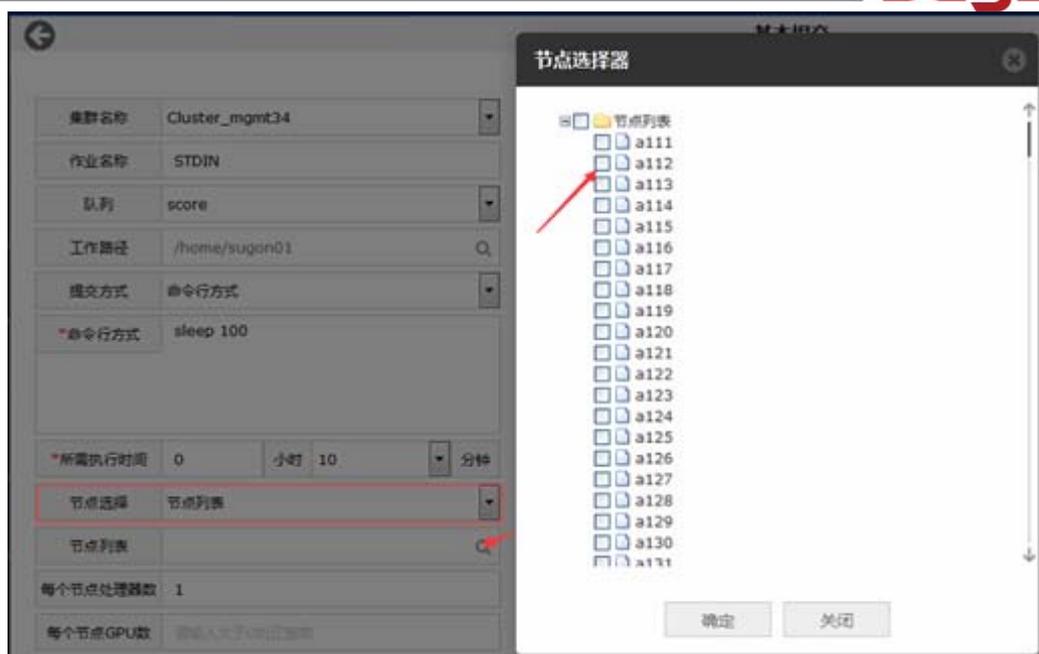
提交
重置

如果选择用脚本方式提交作业，用户可以选择已经写好的脚本文件，如下图所示：

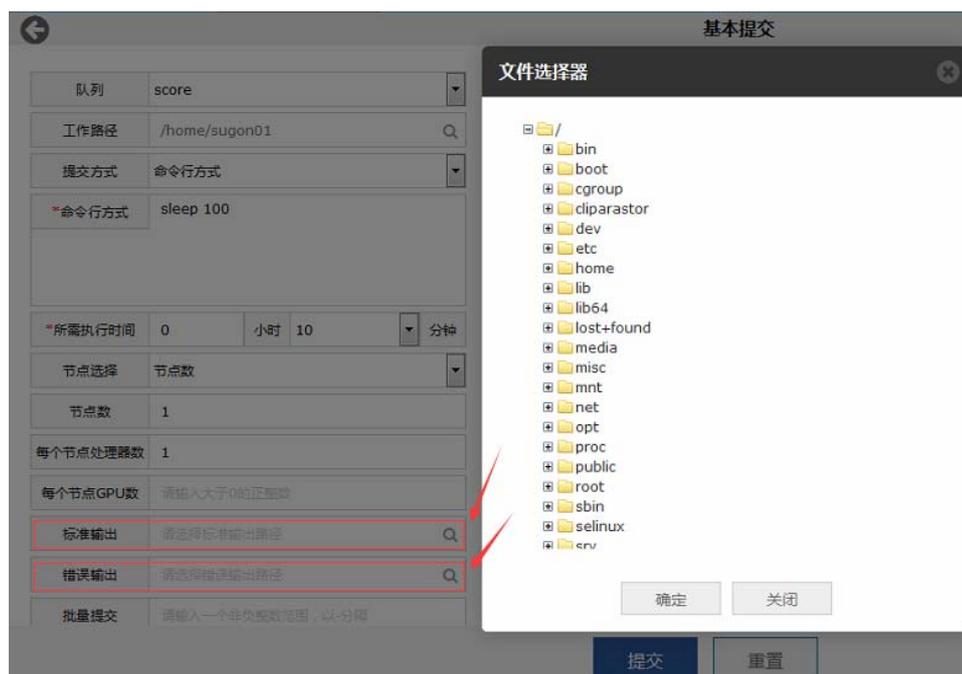


用户根据自己作业情况输入作业所需的执行时间，然后可以选择作业运行的节点数或节点列表。如下图所示：





节点数或节点列表选择完毕后，用户可以指定标准输出文件和错误输出文件名字和路径，也可以用缺省设置。



4.3.1.2 Serial portal 模板作业提交

Serial portal 模板作业提交界面如下，具体提交方式同基础模板方式类似：

The screenshot shows the ClusPortal SERIAL & OpenMP job submission interface. It is titled 'serial' and includes the following sections:

- Available Resource**
- Job Schedule Parameters**
 - *Nnodes: 1
 - *Cores/Node: 1
 - *Wall Time: 1:0:0
 - *Queue: (dropdown menu)
 - *Name: serial_1012_133555
 - Buttons: WinScp File Manager, Web File Manager
- Run Parameters**
 - Program: /bin/hostname (with Browse... button)
 - Arguments: (empty field)
 - Working DIR: /public/home/sugon (with Browse... button)
 - Output File: serial_1012_133555.txt
- Remote Visualization Parameters**
- Advanced Parameters**

At the bottom, there are three buttons: 预设参数 (Pre-set parameters), 提交 (Submit), and 重置 (Reset).

CPU Time: 如果部署了 ClusQuota 集群资源配额计费系统，将显示您目前可用的机时配额，计量单位为“CPU*Hours”。例如，系统显示目前可用配额为 120 CPU*Hours，表明最多可以用 12 个 CPU 并行运算 10 小时。本次计算任务结束之后，将按照“CPU 并行数*实际运行时间”扣除相应的机时配额。队列状态的“Charge Rate”一栏表示它们在 ClusQuota 系统中的计费比率，例如，优先级高的工作队列其计费比率也相应要高一些。

Nnodes: 本次计算任务需要使用多少个节点。本 portal 中只能选 1

Cores/Node: 本次计算任务每个节点需要使用多少 CPU 核。

Wall Time: 本次计算任务预计将运行多长时间。根据系统的调度策略，WallTime 较短的任务将有机会优先运行；不过须注意，一旦 WallTime 时间到了而程序尚未运行结束，本次任务将被强行终止。因此请合理预估 WallTime 的长短。此外，如果部署了 ClusQuota 集群资源配额计费系统，本次任务申请的机时资源不允许超过您目前可用的机时配额。

Queue: 本次计算任务将使用的工作队列。

Name: 本次计算任务的名称。

WinScp File Manager: 启动 WinSCP 程序上传/下载计算任务的输入输出文件。

Web File Manager: 启动文件管理

Program: 选择本次计算任务的可执行程序或命令。

Arguments: 如果应用程序运行时需要提供自定义的参数，请在此输入。

Working DIR: 本次计算任务的工作目录。

Output File: 计算过程中的标准输出和标准错误输出信息，将被重定向保存为文件。

PBS Options: 如果需要手动添加 PBS 作业的高级参数，可在此处设置。这类参数的行首必须包含“#PBS”关键字，将被加到 PBS 脚本文件的开始处。该选项默认无需设置。

Pre Commands: 如果运行 mpirun 命令之前需要做前处理操作，可在此处设置相关命令参数，命令行格式必须遵循 bash 脚本规范。该选项默认无需设置。

Post Commands: 如果在 mpirun 命令运行结束之后需要做后处理操作，可在此处设置相关命令参数，命令行格式必须遵循 bash 脚本规范。该选项默认无需设置。

4.3.1.3 MPI portal 模板作业提交

Mpi portal 模板作业提交界面如下，具体提交方式同基础模板方式类似：

The screenshot shows the MPI portal configuration interface. It includes sections for 'Job Schedule Parameters' (Nodes, Cores/Node, Wall Time, Queue, Name) and 'Run Parameters' (MPI Type, Remote Shell, Communication, Other Parameters, MPI Envfile, MPI Program, Arguments). The 'Run Parameters' section is highlighted with a red box, showing 'MPI Type' set to 'openmpi', 'Remote Shell' set to 'SSH', 'Communication' set to 'InfiniBand', and 'Other Parameters' with 'Share Memory' and 'CPU Binding' checked.

多数参数说明请见 Serial Portal 模板作业提交，下列参数为 MPI portal 模板独有的，说明如下：

MPI Type: 选择 MPI 并行环境，如 Open MPI 或 Intel MPI。

Remote Shell: 多节点并行任务，MPI 初始化并行环境时，节点之间的访问模式。建议采用默认的 SSH 模式。

Communication: 节点连接方式，InfiniBand 和 TCP 两种方式。建议采用默认的 InfiniBand 方式。

Other Paramasters: 多节点并行任务，节点之间数据交换采用何种网络。如果勾选“Share Memory”选项，表示同一节点内的 MPI 进程采用共享内存方式进行数据交换；如果勾选“CPU Binding”选项，表示将 MPI 进程与固定的 CPU 核心绑定，防止进程漂移。开启这两个选项通常可以提高 MPI 程序的运行速度。

MPI Envfile: 文件环境变量配置文件。用户可浏览集群，选择配置文件。

MPI Program: 选择本次计算任务的可执行程序。本 Portal 安装后提供基本的 MPI 版本 CPI 可进行测试，用户也可浏览选择集群中的其他 MPI 程序。

Arguments: 如果 MPI 应用程序运行时需要提供自定义的参数，请在此输入。

MPI Options: 如果需要手动添加 MPI 并行时的高级参数，可在此处设置，这些参数将被传递成为 mpirun 命令行参数的一部分。该选项默认无需设置。

Pre Commands: 如果运行 mpirun 命令之前需要做前处理操作，可在此处设置相关命令参数，命令行格式必须遵循 bash 脚本规范。该选项默认无需设置。

Post Commands: 如果在 mpirun 命令运行结束之后需要做后处理操作，可在此处设置相关命令参数，命令行格式必须遵循 bash 脚本规范。该选项默认无需设置。

4.3.1.4 General portal 模板作业提交

General portal 模板作业提交界面如下，具体提交方式同基础模板方式类似：

参数说明同基础模板/Serial portal 模板/MPI portal 模板。

4.3.2 用命令行方式提交作业

用户可以通过 ssh 客户端软件直接登录到集群系统的登录节点上，用命令行的方式提交作业。请注意，用户可以在登录节点查看文件，编辑文件，查看作业，查看资源使用情况等。但用户不允许在登陆节点上运行计算程序。

在登录节点上提交作业只允许通过 pbs 作业调度系统提交作业。Pbs 提交作业是通过 qsub 命令来执行，qsub 命令通过脚本文件提交作业到作业管理系统，具体格式如下：

```
qsub <PBS 作业脚本>
```

PBS 作业脚本本质上是一个 shell 脚本，注释行以“#”开头，pbs 运行参数以“#PBS”开头。PBS 作业脚本里可以直接调用 shell 命令和系统命令。脚本里运行参数说明如下表所示：

运行参数	说明
-a <作业开始运行的时间>	向 PBS 系统指定作业运行的开始时间。 作业运行时间格式为：[[[CC]YY]MM]DD]hhmm[.SS]
-A <用户名>	使用不同的用户来提交作业，缺省使用当前用户名
-o <标准输出文件的路径> -e <标准错误输出的路径>	该参数指定标准错误输出的位置，缺省的情况下，PBS 系统把标准输出和标准错误输出放在用户 qsub 命令提交作业的目录下。 标准错误输出：<作业名>.o<作业号> 标准错误输出：<作业名>.e<作业号> 路径使用如下格式标准： [<节点名>:]<路径名>
-N <作业名>	指定提交的作业名
-q <目标队列>	指定作业提交的目标队列，其中目标队列可以是目标队列、目标节点名或者是目标节点上的队列。如果目标队列是一个路由队

	列，那么服务器可能把作业路由到新的队列中。如果该参数没有指定，命令 qsub 会把作业脚本提交到缺省的队列中。
-l <申请资源列表>	<p>该参数指定作业脚本申请的 PBS 系统资源列表。</p> <p>申请资源列表使用如下格式：</p> <p style="text-align: center;"><资源名>[=[<数量>]][, 资源名[=[<数量>]], ...]</p> <p>例如作业希望申请在双路节点上申请 5 个 CPU 资源的情况，则可以在脚本中如下：</p> <pre>#PBS -l nodes=2:ppn=2+1:ppn=1</pre>
登陆 SHELL 继承来的变量	包括\$HOME, \$LANG, \$LOGNAME, \$PATH, \$MAIL, \$SHELL 和\$TZ。
\$PBS_0_HOST	qsub 提交的节点名称
\$PBS_0_QUEUE	qsub 提交的作业的最初队列名称
\$PBS_0_WORKDIR	qsub 提交的作业的绝对路径
\$PBS_JOBID	作业被 PBS 系统指定的作业号
\$PBS_JOBNAME	用户指定的作业名，可以在作业提交的时候用 qsub -N <作业名>指定，或者在 PBS 脚本中加入#PBS -N <作业名>。
\$PBS_NODEFILE	<p>PBS 系统指定的作业运行的节点名。该变量在并行机和机群中使用。当在 PBS 脚本中用#PBS -l nodes=2:ppn=2 指定程序运行的节点数时，可以使用\$PBS_NODEFILE 在脚本中引用 PBS 系统指定的作业运行的节点名。比如：</p> <pre>#PBS -l nodes=2:ppn=2 mpirun -np 4 -machinefile \$PBS_NODEFILE <程序名></pre>
\$PBS_QUEUE	PBS 脚本在执行时的队列名

在集群计算系统上提交的作业通常分为如下几类：

- 普通的串行程序，仅使用一个计算核心即可。
- 同一节点内运行的 OpenMP 或基于 threads 的共享内存程序，仅使用一个节点内的多个核心。
- 利用消息传递方式的跨节点的 MPI 并行程序。
- 复杂的 OpenMP+MPI 混合并行程序。

4.3.2.1 普通的串行程序作业脚本示例：

脚本	脚本说明
#PBS -N test	指定作业名：test

#PBS -l nodes=1:ppn=1	指定作业资源：1 个节点，1 个核心
#PBS -l walltime=12:00:00	指定作业运行时间 12 天
#PBS -q blades	指定作业运行队列 blades
cd \$HOME/test/	进入工作目录 \$HOME/test
./a.out > \$HOME/result/a.result	执行命令，并将结果重定向到 ./a.out > \$HOME/result/a.result 文件中。

上面为一个普通的串行作业脚本示例，用户可以通过 qsub 命令，加上该脚本的文件名，就可提交作业。脚本中给指定了作业名称，作业所需资源，作业的运行时间，作业运行所用队列，以及作业执行的目录。用户把作业的可执行文件和目录更换到用户的自己的内容就可成功提交用户自己的作业。

4.3.2.2 共享内存并行作业

脚本	脚本说明
#PBS -N test_OpenMP	指定作业名：test_OpenMP
#PBS -l nodes=1:ppn=10	指定作业资源：1 个节点，10 个核心
#PBS -l walltime=12:00:00	指定作业运行时间 12 天
#PBS -q blades	指定作业运行队列 blades
export OMP_NUM_THREADS=10	设置 OpenMP 线程数环境变量
cd \$HOME/test_OpenMP/	进入工作目录 \$HOME/test_OpenMP
./a.out > \$HOME/result/a.result.OpenMP	执行命令，并将结果重定向到 ./a.out > \$HOME/result/a.result.OpenMP 文件中。

该类作业包括 OpenMP 并行方式的作业，也包括不使用 OpenMP 而是通过 POSIX 等系统底层所编写的多线程程序。

用户请注意，这里的用户申请的节点数，核心数，需要同 OMP_NUM_THREADS 一致，且该数值不应该超出队列中单节点的物理核心数。同时，还要注意用户可执行程序中输入文件如需要设定 OpenMP 的核心数，也要同上面的参数设置一致。

4.3.2.3 MPI 并行作业

脚本	脚本说明
#PBS -N test_MPI	指定作业名：test_MPI
#PBS -l nodes=2:ppn=20	指定作业资源：2 个节点，40 个核心
#PBS -l walltime=12:00:00	指定作业运行时间 12 天
#PBS -q blades	指定作业运行队列 blades
echo This jobs is \$PBS_JOBID@\$PBS_QUEUE	打印作业名字和作业所属队列信息
cd \$PBS_O_WORKDIR	进入工作目录 \$PBS_O_WORKDIR

mpirun -np 40 -machinefile \$PBS_NODEFILE ./vasp	运行 Vasp 可执行文件, 指定 40 个线程, 以及所用的节点名
--	------------------------------------

该类作业为 MPI 并行方式的作业, 请用户注意用户设定的核心数数值不应该超出队列中单节点的物理核心数。

4.3.2.4 OpenMP+MPI 混合同行作业

脚本	脚本说明
#PBS -N test_OMP_MPI	指定作业名: test_OMP_MPI
#PBS -l nodes=2:ppn=20	指定作业资源: 2 个节点, 40 个核心
#PBS -l walltime=12:00:00	指定作业运行时间 12 天
#PBS -q blades	指定作业运行队列 blades
echo This jobs is \$PBS_JOBID@\$PBS_QUEUE	打印作业名字和作业所属队列信息
cd \$PBS_O_WORKDIR	进入工作目录 \$PBS_O_WORKDIR
mpirun -np 40 -machinefile \$PBS_NODEFILE --mca btl self,openib,sm ./test_OMP_MPI	运行 test_OMP_MPI 可执行文件, 指定 40 个线程, 以及所用的节点名, 并指定共享内存, 和用 InfiniBand 网络计算

该类作业为 OpenMP+MPI 混合同行方式的作业, 请用户注意用户设定的核心数数值不应该超出队列中单节点的物理核心数。

4.3.2.5 其它类的作业提交

集群系统除了刀片服务器外, 还包括胖节点。

对于胖节点, 需要用户选择合适的队列提交作业(作业队列必须包含用户打算使用的节点)。

4.4 作业管理

在集群系统上进行作业管理, 用户有两种方式可供选择。一种是通过 GridView Web 界面, 一种是通过命令行的方式, 用 pbs 命令管理查询。

4.4.1 用 GridView Web 界面管理作业

4.4.1.1 用户作业状态查询

提交过作业后, 可以点击作业列表菜单查询作业状态, 如下图所示:

The screenshot displays the '作业列表' (Job List) page in the Sugon Gridview ClusPortal. The interface includes a sidebar with navigation options like '作业提交' (Job Submission) and '作业管理' (Job Management). The main content area shows job submission parameters such as 'Available Resource', 'Job Schedule Parameters' (including *Nnodes and *Queue), and 'Run Parameters' (including 'Run Script' and 'WorkingDIR'). A dropdown menu is open over the '作业列表' option, showing sub-options like '状态视图' (Status View) and '历史作业' (History Jobs). Below the submission form is a table listing jobs with columns for Job ID, Job Name, Owner, Queue, Status, Start Time, and Run Time. The table shows several jobs in a '运行' (Running) state.

作业ID	作业名	所有者	队列	状态	开始时间	运行时间	操作
50882	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50883	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50884	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50885	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50886	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50887	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50888	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:45	[操作图标]
50889	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:41	[操作图标]
50890	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	27-04:23:41	[操作图标]
50891	ensf	zifawang...	silicon_c	运行	2015-10-30T10:54:...	24-05:35:51	[操作图标]

4.4.1.2 用户作业管理

用户可以对列表中对队列进行操作。注意用户可以看到其他用户作业的运行状态，但用户并不能对其他用户的作业进行操作，仅能对于自己的作业进行操作。操作方式是点选左上角的快捷按钮，可以删除作业，保留作业，挂起作业，恢复作业，释放作业，以及重新运行作业等。如下图所示：

This screenshot shows a closer view of the job list table. Red arrows point to the '操作' (Action) column, highlighting icons for '保留' (Keep), '释放' (Release), '挂起' (Suspend), '恢复' (Resume), '重新运行' (Restart), and '删除' (Delete). Below the table, the '作业信息' (Job Information) section is expanded for job ID 50882, showing details like '作业名: ensf', '提交时间', '所有者', '所属队列', 'VNC会话', '作业类型', and '占用物理内存'. To the right, a '文件列表 (/public/hom...)' section is shown with the message '没有数据!' (No data!).

作业ID	作业名	所有者	队列	状态	开始时间	运行时间	操作
50906	ensf	zifawang...	silicon_c	运行	2015-10-27T12:06:...	26-21:48:08	[操作图标]
50973	I4804	linzh@ad...	silicon_c	运行	2015-10-28T16:16:...	25-17:37:56	[操作图标]

作业信息	
作业名:	ensf
作业ID	50882
提交时间	2015-10-27T12:06:22
所有者	zifawang@admin1
所属队列	silicon_c
VNC会话	
作业类型	
占用物理内存	

文件列表 (/public/hom...)	
文件名	操作
没有数据!	

4.4.1.3 用户可用资源申请

用户还可以对自身可用资源进行申请(最大作业数, 最大核心数), 需要点击左侧的资源申请菜单项:

The screenshot shows the Gridview interface with the '资源申请' (Resource Application) menu item highlighted in the left sidebar. The main content area displays a table of running jobs with columns for Job ID, Job Name, Owner, Queue, Status, Start Time, and Run Time. A red arrow points from the '资源申请' menu item to the '资源申请' option in the dropdown menu.

作业ID	作业名	所有者	队列	状态	开始时间	运行时间	操作
50882	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:15	
50883	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:15	
50884	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:15	
50887	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:15	
50888	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:15	
50889	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:11	
50890	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:28:11	
50891	ensf	zifawang...	silcon_c	运行	2015-10-30T10:54:...	24-05:40:21	

提交申请后, 等待系统管理员审批通过, 即可使用申请的新的资源量。

4.4.1.4 用户历史作业查询和报表生成

用户可以通过点击左侧的历史作业菜单, 来查询自己的历史作业运行情况, 如下图所示:

The screenshot shows the Gridview interface with the '历史作业' (Historical Jobs) menu item highlighted in the left sidebar. The main content area displays a table of historical jobs with columns for Job ID, Job Name, Owner, Queue, Status, Start Time, and Run Time. A red arrow points from the '历史作业' menu item to the '历史作业' option in the dropdown menu.

作业ID	作业名	所有者	队列	状态	开始时间	运行时间	操作
50886	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:29:45	
50887	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:29:45	
50888	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:29:45	
50889	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:29:41	
50890	ensf	zifawang...	silcon_c	运行	2015-10-27T12:06:...	27-04:29:41	
50891	ensf	zifawang...	silcon_c	运行	2015-10-30T10:54:...	24-05:41:51	

The screenshot shows two views of the job management interface. The top view is a list of jobs with columns for Job ID, Job Name, Queue Name, Start Time, End Time, Execution Node, and Action. A dropdown menu is open over the 'Queue Name' column, showing options like '最近一小时', '最近一天', '今天', '昨天', '最近一周', '上一周', '最近一个月', '上个月', '全部', and '自定义'. The bottom view shows a similar list but with a calendar overlay for July 2015, allowing users to filter jobs by date. The calendar shows the 1st, 2nd, 3rd, and 4th of the month.

用户还可对历史作业按条件查询，并生成报表。

The screenshot shows the '生成报表' (Generate Report) dialog box. It includes a '模板类型' (Template Type) dropdown set to '用户作业统计详表'. There are sections for '时间过滤' (Time Filter) with '时间方式' (Time Method) set to '预定义' (Predefined) and '时间段' (Time Period) set to '上一小时' (Last Hour). The '基本信息过滤' (Basic Information Filter) section has '用户名' (Username) set to 'sugon'. The '报表导出方式' (Report Export Method) section has '导出HTML' (Export HTML) selected. Buttons for '生成报表' (Generate Report) and '关闭' (Close) are at the bottom.

用户作业统计报表详表

时间范围: 2015-07-02 14:00:00--2015-07-02 15:00:00

生成时间: 2015-07-02 15:10:19

操作人员: wjm

用户: wjm

日期	作业总数	核数	CPU使用时间(小时)	WallTime(小时)	内存使用量(MB)	Efficiency(%)
2015-07-02	9	9	0.0000	5.3917	24.91	0.00
合计	9	9	0.0000	5.3917	24.91	0.00

用户作业统计报表总表

时间范围: 2015-07-02 14:00:00--2015-07-02 15:00:00

生成时间: 2015-07-02 15:10:46

操作人员: wjm

用户名	队列名	CPU使用时间		WallTime		内存使用量		Efficiency
		数值(小时)	百分比(%)	数值(小时)	百分比(%)	数值(MB)	百分比(%)	
wjm	lew	0.0000	0.00	5.3917	100.00	24.91	100.00	0.00
	总计(按队列)	0.0000		5.3917		24.91		0.00
总计(按用户)		0.0000		5.3917		24.91		0.00

4.4.2 用命令行方式管理作业

4.4.1.1 用户作业状态查询

提交过作业后，用户可以查看作业状态：

```
qstat
```

```
[root@login1 bin]# qstat
Job ID          Name          User          Time Use S Queue
-----
1002.mgmt34     TT           sugon         00:00:00 C score
1003.mgmt34     TT           sugon         0 R score
```

```
[root@login1 bin]# qstat -a
```

```
mgmt34:
Job ID          Username      Queue         Jobname       SessID  NDS  TSK  Req'd  Req'd  S  Elap
-----
1002.mgmt34     sugon        score         TT            0       1   24   --    01:00:00 C  --
1003.mgmt34     sugon        score         TT            18442   1   1    --    01:00:00 C  --
```

Job Id 作业标识符 PBS 自动指定

Username 用户名

Queue 队列名

Jobname 作业名 由作业提交人规定的名称

SessID Session 标识符 仅当作业运行时有值

NDS 使用的节点数

TSK 使用的 cpu 数或 task 数

Req' d Memory 作业所需的内存数

Req' d Time 作业所需的 cpu 时间或 wall time

S 作业状态

Elap Time 作业已经运行的时间

查看节点状态：

```
pestat
```

```
[root@login1 bin]# pestat
node      state  load   phymem  ncpus  allmem  resi  usrs  tasks  jobidlist
a110     free  0.07  129010  24    129010  11229  7/2   0      0
a111     free  0.00  129010  24    129010  11228  6/1   0      0
a112     free  0.03  129010  24    129010  11224  6/1   0      0
a113     free  0.06  129011  24    129011  11228  6/1   0      0
a114     free  0.03  129010  24    129010  11230  6/1   0      0
a115     free  0.00  129010  24    129010  11231  6/1   0      0
a116     free  0.11  129011  24    129011  11226  6/1   0      0
a117     free  0.02  129011  24    129011  11229  6/1   0      0
a118     free  0.15  129010  24    129010  11226  6/1   0      0
a119     free  0.02  129011  24    129011  11225  6/1   0      0
a120     free  0.03  129010  24    129010  11226  6/1   0      0
a121     free  0.02  129010  24    129010  11226  6/1   0      0
a122     free  0.00  129010  24    129010  11227  6/1   0      0
a123     free  0.02  129010  24    129010  11240  6/1   0      0
a124     free  0.00  129010  24    129010  11227  6/1   0      0
a125     free  0.00  129010  24    129010  11226  6/1   0      0
a126     free  0.02  129010  24    129010  11228  6/1   0      0
a127     free  0.01  129010  24    129010  11232  6/1   0      0
a128     free  0.00  129010  24    129010  11224  6/1   0      0
a129     free  0.08  129010  24    129010  11233  6/1   0      0
```

node 节点名
state 节点状态
Load 节点负载
Phymem 物理内 MB
ncpus cpu/核心数
Allmem 分配内存数
Resi 寄存器数
usrs 用户数
Tasks 作业数
Joblist 作业列表

4.4.1.2 用户作业管理

用户可以对列表中对列进行操作。注意用户可以看到其他用户作业的运行状态，但用户并不能对其他用户的作业进行操作，仅能对于自己的作业进行操作。操作方式是点选左上角的快捷按钮，可以删除作业，保留作业，挂起作业，恢复作业，释放作业，以及重新运行作业等。如下图所示：

删除作业

```
qdel -p jobid
```

交换作业顺序

```
qorder 111.node1 112.node1
```

挂起作业：

```
qhold 111.node1
```

取消作业挂起

```
qrls 111.node1
```

更改作业运行队列：

```
qmove high 111.node1
```

4.5 源码编译

有些用户需要对自行安装源码编译的软件，或编译自编的代码，需要进行源码编译。目前在集群系统上部署的编译器和编译命令如下表所示：环境变量的选取。

4.5.1 c/fortran 编译器汇总

澳门科技大学超算中心集群系统 C/fortran 编译器汇总

编译器	安装目录	版本	相关命令
GNU C	/usr/bin	4.4.7	gcc myprog.c
GNU C++	/usr/bin	4.4.7	g++ myprog.cpp
GNU Fortran	/usr/bin	4.4.7	gfortran myprog.f
Intel C/C++/Fortran	/public/software/compiler/intel/composer_xe_2015.2.164/	15.0.2	icc/icpc myprog.c
	/public/software/compiler/intel/composer_xe_2015.2.164/	15.0.2	ifort myprog.f

4.5.2 mpi 编译器汇总

澳门科技大学超算中心集群系统 MPI 编译器汇总

MPI 实现 方式	安装目录	版本	MPI 编译器	
			c 编译器	Fortran 编译器
mvapich2	/public/software/mpi/mvapich 2/2.1/intel	2.1	mpicc mpicxx	mpif77 mpif90
	/public/software/mpi/mvapich 2/2.1/gnu	2.1	mpicc mpicxx	mpif77 mpif90
mpich	/public/software/mpi/mpich/3 .1.4/intel	3.1.4	mpicc mpicxx	mpif77 mpif90
	/public/software/mpi/mpich/3 .1.4/gnu	3.1.4	mpicc mpicxx	mpif77 mpif90
openmpi	/public/software/mpi/openmpi /1.6.5/intel	1.6.5	mpicc mpicxx	mpif77 mpif90
	/public/software/mpi/openmpi /1.6.5/gnu	1.6.5	mpicc mpicxx	mpif77 mpif90
	/public/software/mpi/openmpi /1.8.7/intel	1.8.7	mpicc mpicxx	mpif77 mpif90
	/public/software/mpi/openmpi	1.10.	mpicc	mpif77

	/1.10.2/gnu	2	mpicxx	mpif90
IntelMPI	/public/software/mpi/intelmpi/5.0.2.044/	5.0.2	mpiicc mpiicpc	mpiifort
			mpicc mpigcc mpicxx mpigxx	mpif77 mpif90 mpifc

4.5.3 环境变量的选取

澳门科技大学超算中心集群系统 C/fortran 编译器环境变量的选取

编译器	版本	环境变量选取命令
GNU	4.4.7	系统缺省值
Intel	15.0.2	source software/profile.d/compiler_intel-composer_xe_2015.2.164.sh

澳门科技大学超算中心集群系统 mpi 编译器环境变量的选取

MPI 实现方式	版本	环境变量选取命令
mvapich2	Intel 2.1	source software/profile.d/mpi_mvapich2-2.1-intel.sh
	Gnu 2.1	source software/profile.d/mpi_mvapich2-2.1-gnul.sh
mpich	Intel 3.1.4	source software/profile.d/mpi_mpich-3.1.4-intel.sh
	Gnu 3.1.4	source software/profile.d/mpi_mpich-3.1.4-gnu.sh
openmpi	Intel 1.6.5	source software/profile.d/mpi_openmpi-1.6.5-intel.sh
	Gnu 1.6.5	source software/profile.d/mpi_openmpi-1.6.5-gnu.sh
	Intel 1.8.7	source software/profile.d/mpi_openmpi-1.8.7-intel.sh
	Gnu 1.10.2	source software/profile.d/mpi_openmpi-1.10.2-gnu.sh
IntelMPI	5.0.2	source software/profile.d/mpi_intelmpi-5.0.2.044.sh

澳门科技大学超算中心集群系统数学库环境变量的选取

数学库	版本	环境变量选取命令
-----	----	----------

fftw	float 2.1.5	source software/profile.d/mathlib_fftw-2.1.5-float.sh
	double 2.15	source software/profile.d/mathlib_fftw-2.1.5-double.sh
	float 3.3.3	source software/profile.d/mathlib_fftw-3.3.3float.sh
	float 3.3.3	source software/profile.d/mathlib_fftw-3.3.3-float.sh
lapack	Intel 3.4.2	source software/profile.d/mathlib_lapack-3.4.2-intel.sh
	Gnu 3.4.2	source software/profile.d/mathlib_lapack-3.4.2-gpu.sh

附录 Linux 常用命令

1.1 浏览目录命令

用户使用命令行所做的大部分工作是用来定位、列出、创建以及删除文件和目录，下面列举最为常用的这类命令及其解释，更为详细的用法请参见 Linux 有关书籍。

➤ `ls [options] [directory]` 列出文件

常用的命令参数选项有 `-l`, `-a`, `-t` 等。`ls` 代表 `list`。

`ls -la` —— 给出当前目录下所有文件的一个长列表，包括以句点开头的隐藏文件。

`ls -l *.doc` —— 列出当前目录下以字母 `.doc` 结尾的所有文件。

`ls -a` —— 显示当前目录所有文件及目录。

`ls -d` —— 将目录像文件一样显示，而不显示该目录下的文件。

`ls -R` —— 列出所有子目录下的文件。

`ls -t` —— 将文件依建立时间之先后次序列出。

`ls -ltr s*` —— 列当前目录下任何名称是 `s` 开头的文件，愈新的文件排愈后。

➤ `cd [directory]` 切换目录

`cd` 代表 `change directory`。

`cd ~` —— 切换到用户家目录。

`cd /tmp` —— 切换到目录 `/tmp`。

`cd ..` —— 切换到上一层目录

`cd /` —— 切换到系统根目录
`cd /usr/bin` —— 切换到/usr/bin 目录。

1.2 浏览文件命令

➤ `cat [textfile]` 显示文本文件内容
`cat` 代表 `catenate`。

`cat /etc/passwd` —— 显示文本文件 `passwd` 中的内容。

`cat test.txt | more` —— 逐页显示 `test.txt` 文件中的内容。

`cat test.txt >>test1.txt` —— 将 `test.txt` 的内容附加到 `test1.txt` 文件之后。

`cat a.txt b.txt >readme.txt` —— 将文件 `a.txt` 和 `b.txt` 合并成 `readme.txt` 文件。

➤ `more [textfile]` 和 `less [textfile]` 逐屏显示文本文件内容

`more` 命令和 `less` 命令都是用于要显示的内容会超过一个画面长度的情况。`more` 命令让画面在显示满一页时暂停，此时可按空格键继续显示下一个画面；而 `less` 命令除了可以按空格键向下显示文件外，还可以利用上下键来卷动文件。二者都使用热键 `q` 退出。

`more /etc/passwd` —— 显示 `etc` 目录下文本文件 `passwd` 中的内容。

`ls -al | more` —— 以长格形式显示当前目录下的所有内容，显示满一个画面便暂停，可按空格键继续显示下一画面。按热键 `q` 退出。

`less /etc/named.conf` —— 显示 `etc` 目录下文本文件 `named.conf` 中的内容。

`ls -al | less` —— 以长格形式显示当前目录下的所有内容，用户可按上下键浏览。按热键 `q` 退出。

➤ `head [files]` 和 `tail [files]` 查看文件前几行和后几行的内容

`head` 和 `tail` 命令用于查看从文件头或文件尾开始的指定数量的行的内容。

`head -10 /etc/passwd` —— 显示/etc/passwd 文件的前 10 行内容。

`tail -10 /etc/passwd` —— 显示/etc/passwd 文件的倒数 10 行内容。

`tail +10 /etc/passwd` —— 显示/etc/passwd 文件的从第 10 行开始到末尾的内容。

`head -20 file | tail -10 /etc/passwd` —— 结合 `head` 与 `tail` 命令，显示/etc/passwd 文件的第 11 行到第 20 行的内容。

`tail -f /usr/tmp/logs/daemon_log.txt` —— 使用参数 `-f` 时，`tail` 不会回传结束信号，除非我们去自行去中断它；相反地，它会一直不停地继续显示，直到发现文件自它最后一次被读取后，又被加入新的内容时。一般用于监视日志文件的动态更新，有实时监视的效果。本例用于显示 `/usr/tmp/logs/daemon_log.txt` 文件的动态更新。

1.3 目录操作命令

- `pwd` 显示用户目前所在的工作目录的绝对路径名称。

`pwd` 代表 `print working directory`

- `mkdir [-p] [directory]` 创建目录

`mkdir` 代表 `make directory`。

`mkdir mydir` —— 在当前目录下建立 `mydir` 目录。

`mkdir -p one/two/three` —— 在当前目录下建立指定的嵌套子目录。

- `rmdir [-p] [directory]` 删除目录

删除“空”的子目录。`rmdir` 代表 `remove directory`。

`rmdir mydir` —— 删除“空”的子目录 `mydir`。

`rmdir -p one/two/three` —— 删除“空”的嵌套子目录 `one/two/three`。

注意：选项“-p”表示可以递归删除多层子目录，但删除的目录须为空目录，且须具有对该目录的写入权限。

1.4 文件操作命令

- `cp [source] [target]` 复制文件

将一个文件、多个文件或目录复制到另一个地方。`cp` 代表 `copy`。

`cp test1 test2` —— 将文件 `test1` 复制成新文件 `test2`。

`cp test3 /home/bible/` —— 将文件 `test3` 从当前目录复制到 `/home/bible/` 目录中。

`cp -r dir1(目录) dir2(目录)` —— 复制目录 `dir1` 为目录 `dir2`。`-r` 参数表示递归。

注意：`cp` 命令默认将覆盖已存在的文件，加 `-i` 参数表示覆盖前将与用户交互。

- `mv [source] [target]` 移动文件，文件改名

将文件及目录移到另一目录下，或更改文件及目录的名称。`mv` 代表 `move`。

`mv afile bfile` —— 将文件 `afile` 改名成新文件 `bfile`。

`mv afile /tmp` —— 将文件 `afile` 从当前目录移动到 `/tmp/` 目录下。

`mv afile ../` —— 将文件 `afile` 移动到上层目录。

`mv dir1 ../` —— 将目录 `dir1` 移动到上层目录。

- `rm [files]` 删除文件或目录

删除目录需要加 `-r` 选项，强制删除用 `-f`。`rm` 代表 `remove`。

`rm myfiles` —— 删除 `myfiles` 文件。

`rm *` —— 删除当前目录下的所有未隐藏文件。

`rm -f *.txt` —— 强制删除所有以后缀名为 `txt` 文件。

`rm -rf mydir` —— 删除目录 `mydir` 及其下的所有内容。

`rm -ia*` —— 删除当前目录下所有以字母 a 开头的文件，`-i` 选项表示将与用户交互。

➤ `ln [-s] [source] [target]` 建立链接

在文件和目录之间建立链接，参数 `-s` 为建立软链接（符号链接）。`ln` 代表 link。

`ln -s /usr/share/doc doc` —— 创建链接文件 `doc`，并指向目录 `/usr/share/doc`。

`ln -s afile linkafile` —— 为文件 `afile` 创建名为 `linkafile` 的软链接

`ln afile bfile` —— 为文件 `afile` 创建名为 `bfile` 的硬链接

`ln /usr/share/test hard` —— 创建一个硬链接文件 `hard`，这时对于 `test` 文件对应的存储区域来说，又多了一个文件指向它。

➤ `touch [options] [filename]` 新建一个文本文件

新建一个文本文件或修改文件的存取/修改的时间记录值。

`touch *` —— 将当前目录下的文件时间修改为系统的当前时间。

`touch -d 20100101 test` —— 将 `test` 文件的日期改为 2010 年 1 月 1 日。

`touch abc` —— 若 `abc` 文件存在，则修改为系统的当前时间；若不存在，则生成一个为当前时间的空文件。

➤ `file [filename]` 查看 `filename` 文件的类型

1.5 查找类命令

➤ `grep 'string' [file]` 在文件中搜索匹配的字符串位置（所在行）并输出到屏幕

`grep` 代表 (global regular expression print, 全局正则表达式打印)。

`grep bible /etc/exports` —— 查找文件 `/etc/exports` 中包含字符串 `bible` 的所有行。

`grep -v ^# /etc/apache2/httpd.conf` —— 在主 Apache 配置文件中，查找所有非注释行。

`tail -100 /var/log/apache/access.log | grep 404` —— 在 Web 服务器日志的后 100 行中查找包含字符串 `404` 的行，`404` 代表 Web 服务器的“文件没找到”代码。

`tail -100 /var/log/apache/access.log | grep -v googlebot` —— 在 Web 服务器的后 100 行中，查看没有被 Google 搜索引擎访问的行。

`rpm -qa | grep httpd` —— 搜索已安装的 rpm 包中含有 `httpd` 字符串的文件名。

➤ `find name [filename]` 和 `locate [file]` 查找文件或目录

`find` 用来查找文件或目录。`locate` 用于快速查找定位文件，但只能搜索文件名。

`find ./ -name httpd.conf` —— 搜索当前目录下名为 `httpd.conf` 的文件并显示结果。

`find /etc -name httpd.conf` —— 搜索 `/etc` 目录下名为 `httpd.conf` 的文件并显示结果。

`find . | grep page` —— 在当前目录及其子目录中，查找文件名包含字符串 `page` 的文件。

locate traceroute —— 在系统任何地方查找文件名包含字符串 traceroute 的文件。

- whereis [options] 查找程序的源、二进制文件或手册
whereis 命令在指定的目录中查找程序的源、二进制文件或手册。
- whereis passwd —— 将和 passwd 文件相关的文件都查找出来。
- whereis -b passwd —— 只将二进制文件查找出来。

1.6 用法帮助命令

- man [command] 查看 command 命令的说明文档
man 代表 manual page
- info [command] 查看 command 命令的说明文档
info 代表 information
- whatis [command] 在 whatis 资料库（手册）中搜寻指定命令的简短描述。

1.7 打包、解包，压缩、解压缩命令

➤ tar [options] [filename] 打包命令。
tar 代表 tape archive。它能够将用户所指定的文件或目录打包成一个文件，但不做压缩。
一般 Linux/Unix 上常将打包命令 tar 与压缩 gzip 联合使用。Tar 不仅可以打包文件，也可以将硬盘数据备份。tar 命令常用参数：

- c: 创建一个新 tar 文件
- v: 显示运行过程的信息
- f: 指定文件名
- z: 调用 gzip 压缩命令进行压缩或解压
- j: 调用 bzip2 压缩命令进行压缩或解压
- t: 查看压缩文件的内容
- x: 解开 tar 文件
- p: 使用原文件的原来属性（属性不会依据使用者而变）

tar -cvf test.tar * —— 将所有文件打包成 test.tar，扩展名 .tar 需自行加上。

tar -zcvf test.tar.gz * —— 将所有文件打包并调用 gzip 命令压缩成为 test.tar.gz。

tar -tf test.tar —— 查看 test.tar 文件中包括了哪些文件。

tar -xvf test.tar —— 将 test.tar 文件解开。

tar -zxvf foo.tar.gz —— 将 foo.tar.gz 解压缩。

tar -jxvf foo.tar.bz2 —— 将 foo.tar.bz2 解压缩。

`tar -cvf /tmp/etc.tar /etc` —— 将整个/etc 目录下的文件全部打包成为 /tmp/etc.tar。

`tar -zcvf /tmp/etc.tar.gz /etc` —— 将整个/etc 目录下的文件全部打包并调用用 `gzip` 命令压缩成为/tmp/etc.tar.gz。

`tar -zxvpf /tmp/etc.tar.gz /etc` —— 将/etc/内的所有文件备份下来，并且保存其权限。参数 `-p` 非常重要，尤其是当需要保留原文件的属性时！

➤ `gzip [options] [filename]` 压缩和解压缩命令。

通过压缩减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。`gzip` 和 `gunzip` 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令。`gzip` 代表 GNU zip。GNU 是 Gnu is Not Unix 的缩写，GNU

Project 是自由软件基金会（Free Software Foundation）的一部分，它对 Linux 下的许多编程工具负责。

各选项的含义：

`-c`：将压缩结果写入到标准输出上，原文件保持不变。缺省时 `gzip` 将原文件压缩为.gz 文件，并删除原文件。

`-r`：递归式地查找指定目录并压缩其中的所有文件或者是解压缩。

`-d`：解压缩指定文件。

`-t`：测试压缩文件的完整性。

`-v`：对每一个压缩和解压的文件，显示文件名和压缩比。

`gzip usr.tar` —— 压缩一个文件 `usr.tar`，此时压缩文件的扩展名为 `.tar.gz`。

`gzip -v /mnt/lgx/a1.doc` —— 压缩文件/mnt/lgx/a1.doc，此时压缩文件的扩展名为 `.gz`。

`gzip -d /mnt/lgx/a1.doc.gz` —— 解压缩文件/mnt/lgx/a1.doc.gz

1.8 时间相关命令

➤ `date` 显示/修改当前的系统时间

`date` —— 查看系统当前时间。

`date 121010232009.10` —— 将时间更改为 12 月 10 日 10 点 23 分 10 秒 2009 年[月日時分年.秒]。

➤ `cal` 显示日历

`cal` —— 显示当月日历。

`cal 7 2007` —— 显示 2007 年 7 月份的日历。

`cal 2010` —— 显示 2010 年全年的日历。

- hwclock 显示当前的硬件时钟

hwclock --show —— 查看硬件当前时钟。

hwclock --set --date="01/17/2010 13:26:00" —— 设置硬件时钟，格式 hwclock --set --date="月/日/年 时:分:秒"。

hwclock --hctosys —— 硬件时钟与系统时间同步。--hctosys 表示 Hardware Clock to SYStem clock。

hwclock --systohc —— 系统时间和硬件时钟同步。

- ntpdate 同步网络时钟

ntpdate 210.72.145.44 —— 与 ntp 时间服务器进行时间同步。210.72.145.44 是中国国家授时中心的官方服务器。（需要安装 ntp 的软件包）。

1.9 系统信息类命令

- dmesg 显示系统开机信息命令

dmesg 代表 diagnostic message。显示系统诊断信息、操作系统版本号、物理内存的大小及其它信息。

- df 用于查看文件系统的各个分区的占用情况。df 代表 disk free。

df -hl —— 查看磁盘剩余空间信息。

df -T —— 显示分区类型。

- fdisk 磁盘分区工具

fdisk -l —— 显示所有硬盘的分区情况。

- du [options] [directory or filename...] 显示指定的目录或文件所占用的磁盘空间。

du 代表 disk usage

- free 查看系统内存，虚拟内存（交换空间）的大小占用情况

- who 或 w 查看当前系统中有哪些用户登录

who —— 显示登录的用户名、登录终端和登录时间。

who -uH —— 带有标题栏的登录用户的详情，其中 -u 选项指定显示用户空闲时间。

1.10 网络通讯类命令

- ifconfig 显示和设置网络设备

ifconfig eth0 192.168.0.1 —— 将第一块网卡的 IP 地址设置为 192.168.0.1。

ifconfig eth0 down —— 关闭第一块网卡。

`ifconfig eth0 up` —— 启用第一块网卡。

`ifconfig eth0 netmask 255.255.255.0` —— 将第一块网卡的子网掩码设置为 255.255.255.0。

`ifconfig eth0 192.168.0.1 netmask 255.255.255.0` —— 同时设置 IP 地址和子网掩码。

`ifconfig eth0 -broadcast 192.168.0.255` —— 将第一块网卡的广播地址设置为 192.168.0.255。

➤ `route` 显示和设置路由

`route add 0.0.0.0 gw 网关地址` —— 增加一个默认路由。

`route del 0.0.0.0 gw 网关地址` —— 删除一个默认路由。

`route` —— 显示当前路由表。

➤ `ping [options] [主机名/IP 地址]`，检测是否能够与远端机器建立网络通讯连接

➤ `netstat [options]` 查看网络状态

`netstat -i` ——interface，显示网络界面信息表单。

`netstat -s` ——statistic，显示网络工作信息统计表。

`netstat -t` ——tcp，显示 TCP 传输协议的连接状态。

`netstat -r` ——route，显示路由表。

➤ `traceroute [远程主机 IP 地址或域名]` 跟踪路由

➤ `ftp` 文件传输

➤ `telnet [主机名/IP 地址]` 登录到远程计算机

➤ `finger` 查询远程计算机（通常是运行 Linux/UNIX 的计算机）上用户的详细信息。

1.11 软件包管理命令

RPM 的全名是 Red Hat Package Manager。利用 RPM 命令，可以安装、删除、升级管理软件，支持在线安装和升级软件。通过 RPM 包管理可以知道软件包包含哪些文件，也可以查询系统中的某个文件属于哪个软件包，可以查询系统中的软件包是否安装及安装的版本。具体用法请参见 Linux 相关书籍。下面列举一些 rpm 的基本用法。

➤ 我们得到一个新软件，在安装之前，一般都要先查看一下这个软件包是做什么的，可以用这条命令查看：

```
rpm -qpi strace-4.5.18-10.13.x86_64.rpm
```

系统将会列出这个软件包的详细资料。

➤ 我们可以用下面这条命令查看软件包将会在系统里安装哪些文件：

```
rpm -qpl strace-4.5.18-10.13.x86_64.rpm
```

- 安装该软件包：
`rpm -ivh strace-4.5.18-10.13.x86_64.rpm`
- 如果系统已经安装该软件包的低版本，可以用下面的命令进行升级安装：
`rpm -Uvh strace-4.5.18-10.13.x86_64.rpm`
安装某个软件时，RPM会自动处理包的依赖关系，如果不想进行依赖检查，可以给rpm加上 `--nodeps` 参数，想要强制安装可以加上 `--force` 参数。
- 卸载某个安装过的软件，只需执行 `rpm -e <文件名>` 命令即可。
`rpm -e strace`
- 如果不小心误删了某些包的系统文件，可以用下列命令查看有哪些文件损坏，以便进行修复安装。
`rpm -Va`
- 下面这条命令行可以帮助我们快速判定某个文件是属于哪个软件包：
`rpm -qf <文件名>`
- 如果想查看当前系统已经安装了哪些rpm包，可以执行：
`rpm -qa`
也可以与 `grep` 联用，进行查找操作：
`rpm -qa | grep strace`

1.12 编辑器命令 (vim)

在 Linux 下编写文本或语言程序，首先必须选择一种文本编辑器。VIM 编辑器是工作在字符模式下的高效文本编辑器，它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制。

在命令行里输入 `vim` 即可调用 VIM 编辑器：

`vim` —— 调用 `vim`，可以进行编辑工作，编辑完成后可以保存到新文件；

`vim foo.txt` —— 如果文件已经存在，则打开编辑，如果文件不存在，则新建编辑；

VIM 有几种基本工作模式，在 VIM 里头执行 `:help mode` 可以看到 VIM 的所有模式，主要有如下几种。

- Normal mode 即通常所谓的命令模式，在此模式使用 `a`、`i`、`A`、`I`、`o`、`O` 等进入 Insert mode。
- Insert mode 即通常所谓的编辑模式，在此模式使用 `ESC` 进入 Normal mode 或者 `Ctrl-o` 临时进入 Normal mode。
- Command-line mode 命令行模式，在 Normal mode 下按冒号进入，按 `ESC` 取消执行命令或者回车执行命令，然后回到 Normal mode。
- Visual mode 即选择模式（注意跟用鼠标选择不同），用 `v`、`V`、`C-v` 或 `C-q` 进入
- Select mode 鼠标选择
- Replace mode 在 Normal mode 下按 `R` 进入，按 `ESC` 返回 Normal mode，相当于 Windows 下命令行中按 `Insert` 键进入的覆盖模式

从 VI/VIM 中退出：按 `ESC` 确认返回到 Normal mode，然后

- `:wq` 保存并退出

- :q! 不保存并退出
- :x 退出，如果文件更改则保存
- ZZ 退出，如果文件更改则保存（按住Shift，再按两次z）

VIM 的命令是非常具有美感的，下面是几个例子：

- i 在光标前插入；I 在行首插入
- a 在光标后插入；A 在行末插入
- 在下一行插入；O 在上一行插入
- x 删除当前字符；X 删除前一个字符

类似的还有 b, B, ge, gE, w, W, e, E, f, F, t, T, 这些命令都可以用 :help cmdname 查到帮助。

- dd 删除一行
- yy 拷贝一行
- h j k l 左下上右移动光标
- Ctrl-w h, Ctrl-w j, Ctrl-w k, Ctrl-w l 切换到左下上右窗口
- VI/VIM的命令大多都可以带一个数字前缀或者一个数字范围，比如：
- 5dd 从当前行开始删除5行
- 5yy 从当前行开始拷贝5行
- 1,3d 删除1至3行
- 1,3y 拷贝1至3行（.代表当前行，\$代表最后一行）

VIM 的功能非常强大，想进一步了解和学习的 VIM，可以参考 VIM 相关文档。

1.13 用户管理命令

useradd 添加用户

userdel 删除用户

passwd 为用户设置密码

usermod 修改用户的登录名、用户的家目录等

id 查看用户的 UID、GID 及所归属的用户组

pwck 校验用户配置文件/etc/passwd 和/etc/shadow 文件内容是否合法或完整

chfn 更改用户信息工具。可以留下真实的姓名、办公室、电话等资料。

su 用户切换工具。表示 substitute user

sudo 通过另一个用户来执行命令（execute a command as another user）

finger 查看用户信息工具

1.14 用户组管理命令

groupadd 添加用户组

groupdel 删除用户组

groupmod 修改用户组信息

groups 显示用户所属的组

grpck 校验组账号文件 (/etc/group) 和影子文件 (/etc/gshadow) 的一致性和正确性。

1.15 输入/输出重定向与管道命令

➤ [command]<inputfile 输入重定向

输入重定向。输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以来自键盘，而来自一个指定的文件。所以说，输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

➤ [command]>outputfile 输出重定向

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

ls -lR >dirtree.list —— 创建一个包含目录树列表的文件。

➤ [command] >>outputfile 输出追加重定向

为避免输出重定向中指定文件只能存放当前命令的输出重定向的内容，shell 提供了输出重定向的一种追加手段。输出追加重定向与输出重定向的功能非常相似，区别仅在于输出追加重定向的功能是把命令（或可执行程序）的输出结果追加到指定文件的最后，而该文件原有内容不被破坏。如果文件不存在，那么就创建它，如果存在，那么就追加到文件后边。

➤ [command1] | [command2] 把 command1 执行的结果作为输入送到 command2 中执行。