高性能计算入门

大型科学仪器共享平台

准备工作

- 申请账号
  - https://v.ruc.edu.cn/servcenter#/form/draw/8438
- 安装SSH登录软件
  - http://183.174.229.251/cluster-support/cluster-login/
- 熟悉我们的在线使用手册
  - http://183.174.229.251/cluster-support/

目录

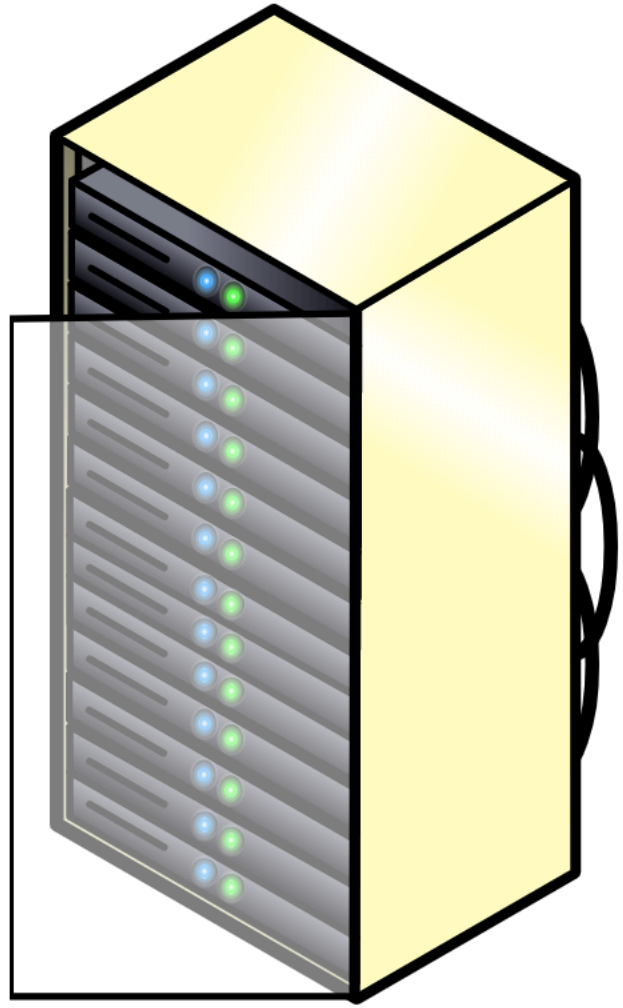
- 高性能计算案例
- 高性能计算集群简介
  - 硬件架构
  - 术语解释
- 快速开始
  - 登陆、上传数据和代码
  - 提交作业
  - 软件使用
  - Python/R、Jupyter、深度学习与容器

一些使用高性能计算的案例

- 统计
  - 统计机器学习
- 化学
  - 分子动力学模拟
- 生物信息学
  - 长非编码RNA的鉴定
- 中文/新闻
  - 使用计算机对文章进行分类
- …

双核酷睿 i5 + 8GB 内存
256G SSD硬盘

V.S.



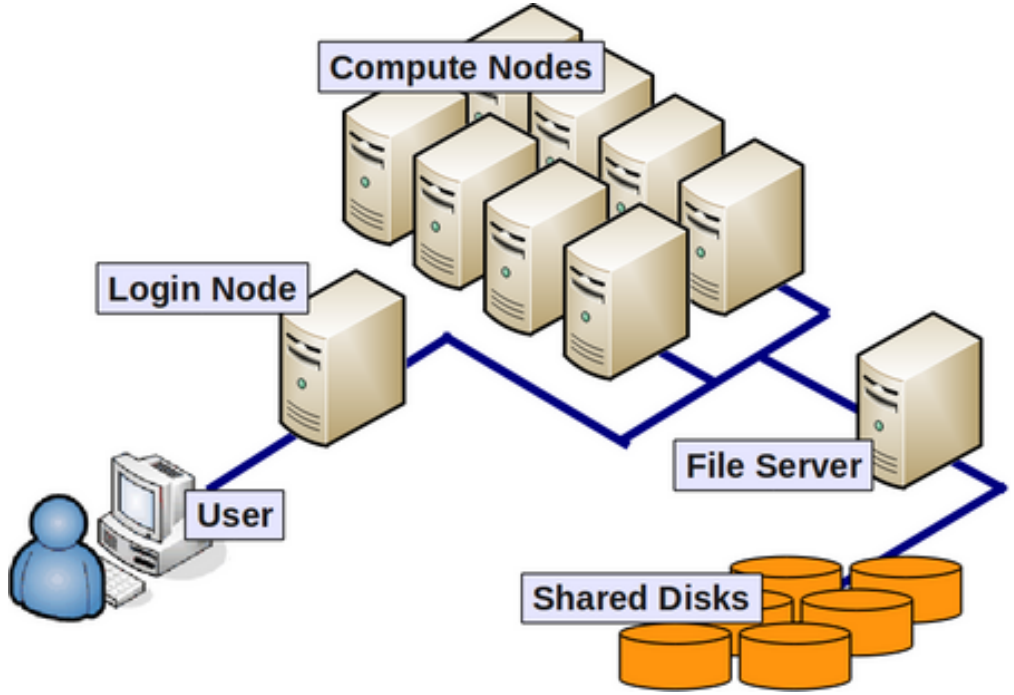
（24核至强 + 128GB 内存）✖55
150T 存储空间

为何选择我们

- 涵盖多学科的计算软件
- 软件教学和问题解答
- 强劲的高性能计算硬件
- 专业的数据中心部署方案

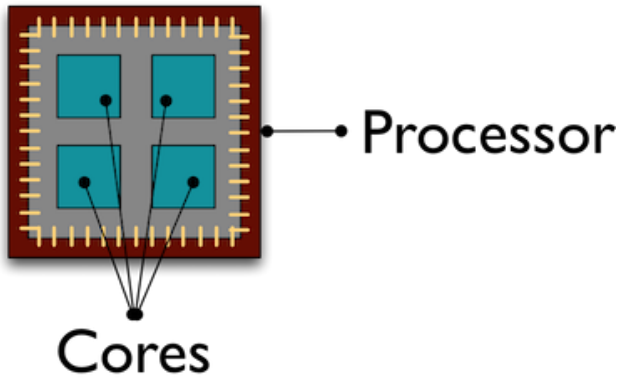
什么是高性能计算集群？

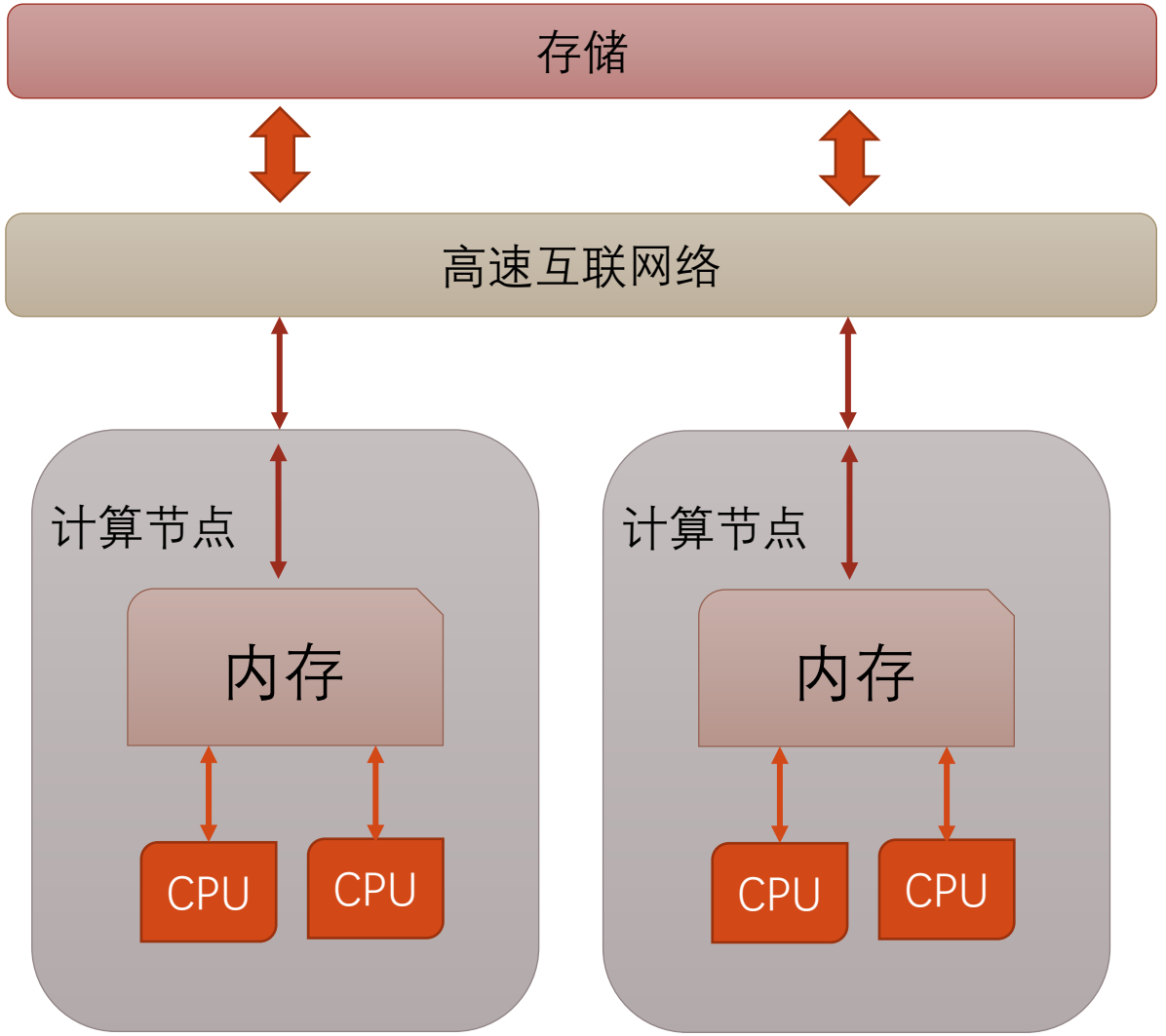
- 由一组计算性能强劲的计算机通过高速网络连接后组成性能强劲的集群。
  - 登录节点
  - 计算节点
  - 存储节点



集群硬件架构

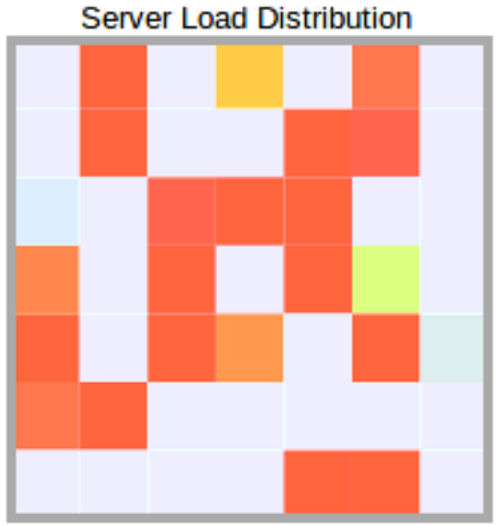
- 节点：某台物理计算机
- CPU：Central Processing Unit
  - 由多个核心（Core）组成
- 节点间使用高速互联网络连接





术语解释

- 程序：
  - 能够被计算机CPU执行
  - 给定一个输入，程序经过计算执行，产生一个输出
  - 一般需要编译器/解释器将代码转化成可执行程序
- 软件：
  - 商业公司或开源社区编写的大型程序，被我们用来进行科学计算
- 作业：
  - 计算资源被分配到某个用户的某个程序上来执行，被称为启动作业
- 调度：
  - 调度器：计算节点像共享单车一样被共享，需要一种机制来分配这些计算资源
  - 作业被分配到某些节点上进行计算



集群服务器负载热力图

串行程序和并行程序

- 请王小明同学来计算4道题目。
  - 小明每次算一道题花费1s，总共耗时4s。（串行）
- 请第二组的韩梅梅，李雷来计算4道题目。
  - 小两口相视一笑花费0.1s （互相通信）
  - 然后每个人做两道题，花费2s
  - 总共花费2.1s （并行）

- 并行程序可以加快问题求解速度，但不是所有计算都可以并行。
  - 使用MPI框架和OpenMP编程来并行计算。
- 推荐阅读 张林波等 《并行计算导论》

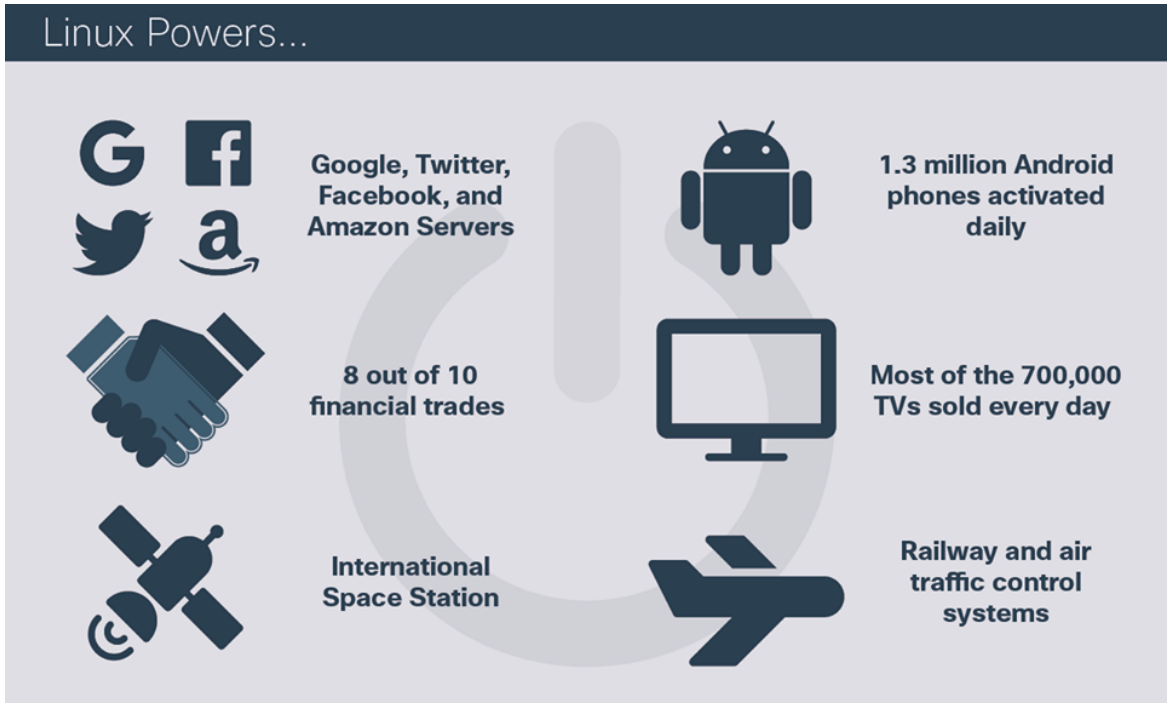
集群支持的编程语言

- C/C++ 大型项目选用的编程语言，运行快，调试编写慢
- Fortran 传统科学计算语言
- Linux Shell脚本
  - 处理数据利器，把人从重复性工作中解放，提升工作效率
- Python/R 数据处理、分析、可视化，机器学习
- Julia 科学计算新秀，开源，速度快
- Matlab 传统科学计算软件

与集群交互

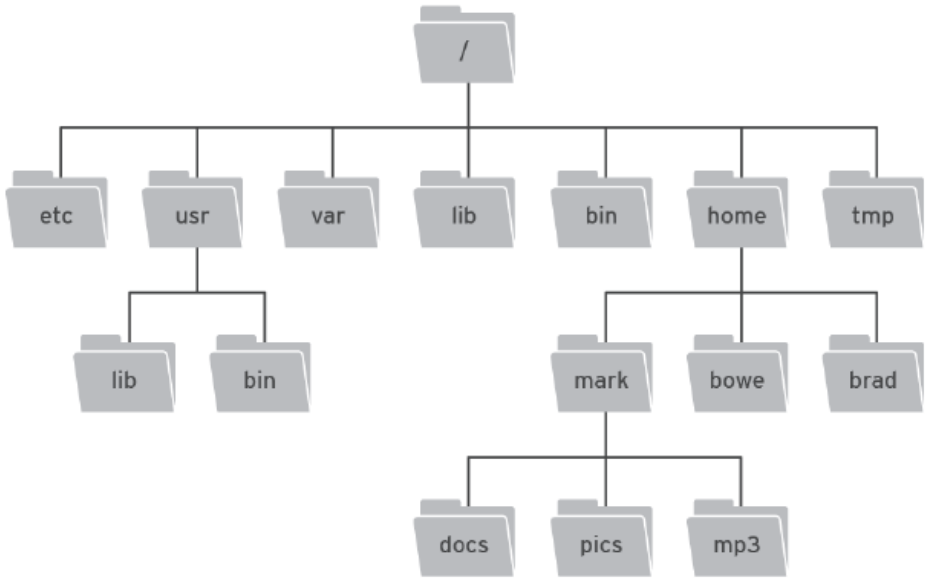
- Linux
- 使用SSH协议登录集群
  - Windows：Xshell、MobaXterm、 Putty、WinScp
  - Mac / Linux：iTerm、Terminal
- 使用scp、rsync等命令上传文件
- 使用wget命令下载文件

- 《鸟哥的Linux私房菜》



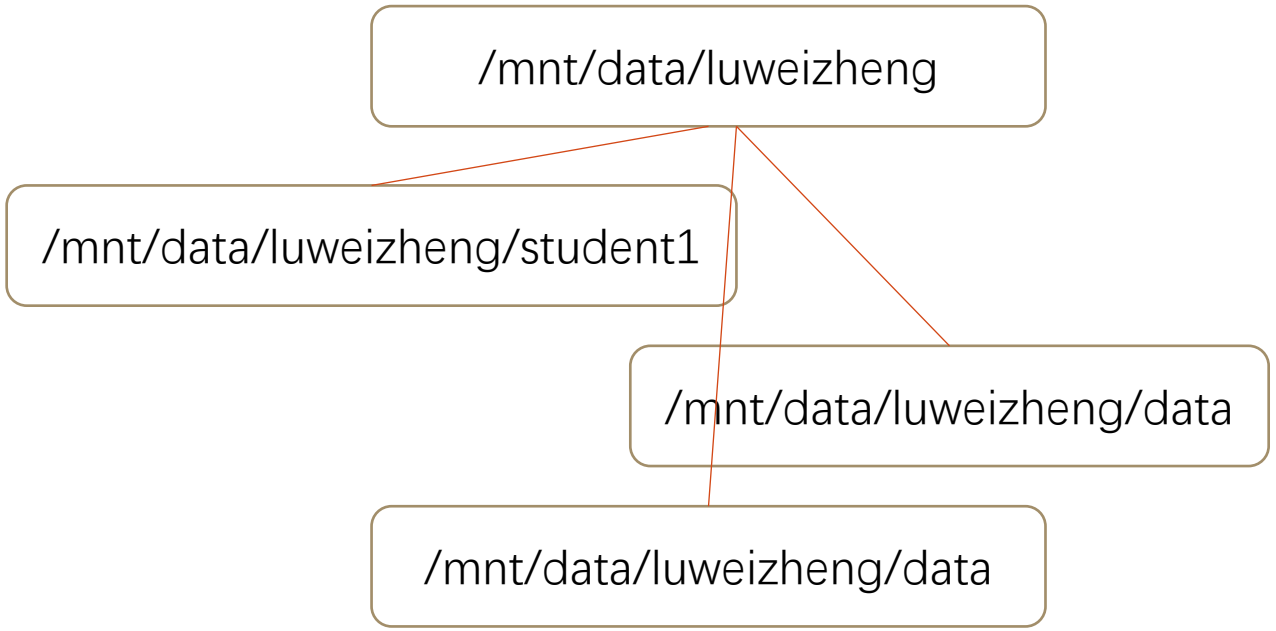
Linux目录介绍

- 树形结构
- / 最顶层目录
- /etc 系统配置文件目录
- /usr /bin /sbin 一些重要程序的安装目录
- /lib /lib64 依赖库
- cd 切换目录
- pwd 查看当前目录
- . 当前目录
- .. 上一层目录
- ~ 等于用户home目录



集群数据管理

- home
  - 代码、脚本、Python、R依赖包
- /mnt/data 150T空间
  - 大型软件
  - 大文件数据集、输入输出文件等
  - 需要创建属于自己的文件夹 mkdir
- /public 公共目录：公用软件、配置文件



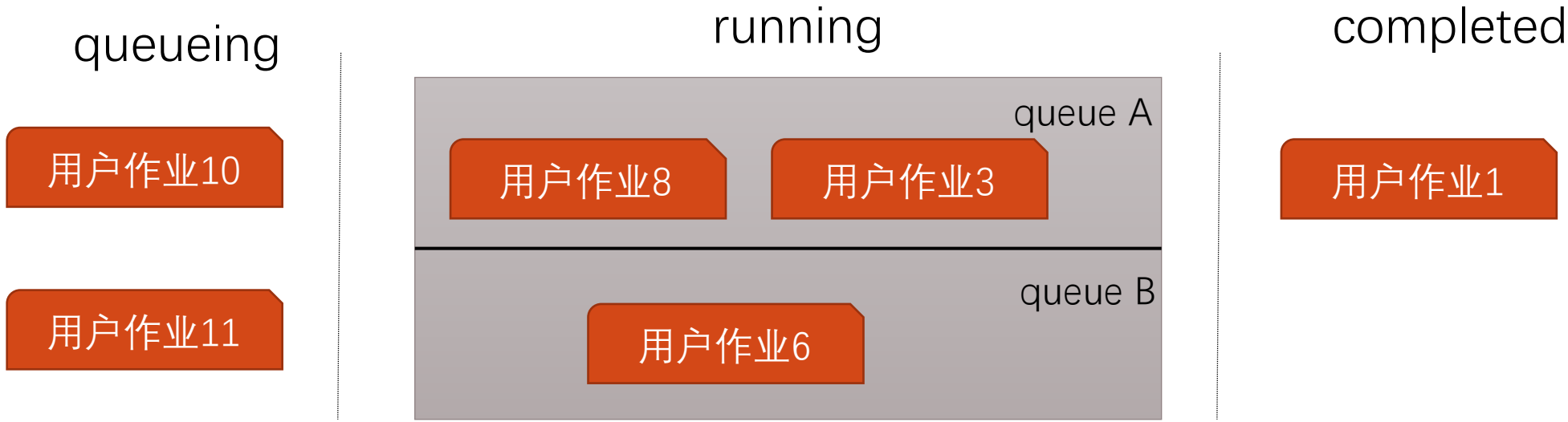

```
drwxr-xr-x 2 luweizheng users 6 3月   12 2019 data
drwxr-xr-x 2 luweizheng users 6 3月   12 2019 student1
drwxr-xr-x 2 luweizheng users 6 3月   12 2019 student2
drwxr-xr-x 2 luweizheng users 6 3月   12 2019 student3
```

作业调度器

- **Torque/PBS**

  http://docs.adaptivecomputing.com/torque/4-2-10/help.htm

- 计算资源分配、作业调度
- 类似于银行叫号机



提交作业

- `git clone https://github.com/masterstevelu/cluster-tutorials.git`
- `cd cluster-tutorials/job-scheduler/`
- `qsub -l nodes=1:ppn=2 hello.sh`
- 作业参数

作业参数

- 作业名
- 队列名
- 申请资源
- 邮箱
- 输入输出文件

属性	取值	说明
-l	资源申请参数，通常为多个参数	设定作业所需资源，详见资源申请部分
-N	作业名称	设定作业名称，用以区别你和其他人的作业名
-o	程序运行后标准输出的文件路径	设定作业的标准输出文件路径，如不设置，则输出文件为 作业名.o+作业id
-e	程序运行后错误日志输出的文件路径	设定作业的标准错误文件路径，如不设置，则输出文件为 作业名.e+作业id
-p	-1024到1023之间的整数	设定作业的优先级，数值越大优先级越高
-q	队列名	设定作业队列名称
-M	邮箱	作业执行结果通过邮件通知用户
-m	e、b、a三个选项	邮件选项，b表示作业开始运行时发送提醒邮件，e表示作业结束时发送提醒邮件，a表示当作业被系统管理员杀死时发送提醒邮件

给作业分配计算资源

- `-l nodes=1:ppn=24`
  - 1个节点 24个核心
- `-l nodes=2:ppn=24`
  - 2个节点 共48个核心
- 队列：
  - default普通节点24个核心
  - pnode胖节点40个核心
- 队列将会不断调整

资源	取值	说明
nodes	节点	设定作业所需节点资源
ppn	数值，应小于每个节点上的CPU核数	设定每个节点所需要的CPU核数
walltime	hh:mm:ss	设定作业所需的最大墙上时间，从进程从开始运行到结束，时钟走过的时间
pmem	正整数，后面可跟b、kb、mb、gb	设定作业在每个核心上所需最大内存
cput	hh:mm:ss	设定作业所需的最大CPU时间，用户的作业获得了CPU资源以后，所有CPU执行时间的总和

一些有用的PBS命令

- `qstat [-f] job_id`
  - 作业运行状态：R(Running)、Q(Queueing)、C(Completed)
- `qdel job_id`
- `pbsnodes [-l all/free]`
- `qsub –I` 交互模式 调试程序时使用

输出结果

- `stdout` 程序中`print`正确输出
- `stderr` 运行不成功的报错信息

软件加载

- 类似图书馆查阅书籍：
  - 在检索系统中查找书架号
  - 找到对应书架并找到对应的书
- 环境变量PATH：在Linux系统中找到软件安装的位置

- `source xxx`
- `module load xxx`

一个提交作业的样例

- 作业名
- 运行队列
- 输出文件
- 获取计算资源
- 邮件提示
- 加载环境变量
- 运行程序

```
#!/bin/bash

### file: example.sh
### set job name
#PBS -N example-job
### set output files
#PBS -o example.stdout
#PBS -e example.stderr
### set queue name
#PBS -q default
### set computing resources
#PBS -l nodes=2:ppn=4
### send mail
#PBS -M xxx@ruc.edu.cn
#PBS -m eba
### enter job's working directory
cd $PBS_O_WORKDIR
### get the number of processors
NP=`cat $PBS_NODEFILE | wc -l`

### add environment variables
module load mpi/intelmpi-2015
### 程序执行部分，不同程序的执行方法不同，请根据你的程序来修改下面一行
### 如对你的脚本有疑问，请联系我们
### run an example mpi job
mpirun -np $NP -machinefile $PBS_NODEFILE ./mpi
```

Anaconda

- 数据科学及相关依赖包
- 使用Anaconda创建和使用自己的环境
  - 包含常用科学计算库 且经过10倍硬件加速
  - 尽量不要自己编译安装
- `module load anaconda/5.3.0`

```
#PBS -N iris_classification_example
### use one node for this job
#PBS -l nodes=1
#PBS -q default
#PBS -o iris.stdout
#PBS -e iris.stderr

cd $PBS_O_WORKDIR
### set python in anaconda and 'test_evn_2' environment
module load anaconda/5.3.0
source activate test_env_2

python3 iris_with_xgb.py
```

Python

- Base环境
- 在base环境安装包
  - pip install --user your_package -i https://pypi.douban.com/simple
- 从base环境中克隆出自己的环境
  - conda create --name test_env_2 --clone base
  - source activate test_env_2
  - source deactivate test_env_2
- 使用Anaconda时默认是base环境！请注意你的环境！

包	版本	包	版本
gensim	3.7.1	nltk	3.3.0
gsl	2.4	numpy	1.15.4
hdf5	1.10.2	openblas	0.3.3
jieba	0.39	pandas	0.23.4
jupyter	1.0.0	python	3.7.0
matplotlib	3.0.2	scikit-learn	0.20.2
mkl	2019.1	scipy	1.2.0
mpi4py	3.0.0	seaborn	0.9.0
networkx	2.1	xgboost	0.81

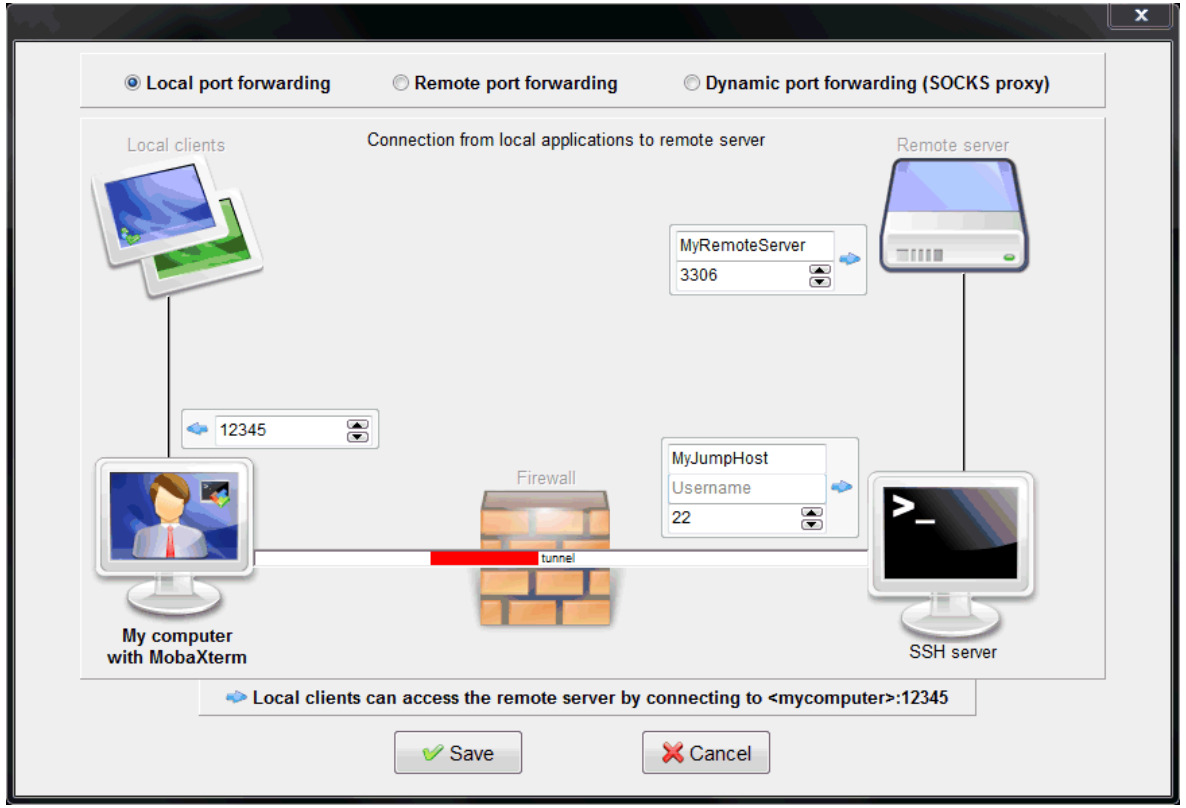
R & Julia

- Anaconda base环境已安装
- R
  - http://183.174.229.251/cluster-support/manual/r/
  - 默认单进程 未并行加速
  - `Rscript example.R`
  - `install.packages('your_package')`
- Julia
  - `julia example.jl`
  - 支持并行

```
import Pkg
Pkg.add("Calculus")
```

Jupyter

- Web交互编程工具 数据可视化
- 使用Jupyter步骤
  - 使用我们提供的脚本提交作业
  - 开启SSH隧道
  - 访问http://localhost:port/
- `cd cluster-tutorials/jupyter/`
- `sh jupyter.sh`



Singularity容器

- 兼容docker，将所需操作系统、依赖库、软件等信息都打包到一个镜像中
- 镜像像虚拟机一样被快速启动
- 解决多个工作环境不一致问题
- 像执行普通命令一样，在容器中执行命令：
  - ```
    /usr/local/bin/singularity exec
    /mnt/data/container_library/deep_learning/tf-1.10.0-py35
    python3 -c 'import tensorflow as tf;
    print(tf.__version__)'
    ```
- 镜像位置 **/mnt/data/container_library**
- 制作镜像

  https://www.sylabs.io/guides/2.6/user-guide/container_recipes.html
    ```

机器学习

- TensorFlow、PyTorch以及其他一切依赖这些包的PyPi包请在Singularity容器中运行
- 在物理机上直接运行xgboost、scikit-learn等包

工作流程

- 编写代码、准备数据集
- 将代码编译成可执行文件（Optional）
- 测试代码在小量数据上能够运行成功
- 编写作业提交脚本
- 使用qsub提交作业
- 看其他paper、刷网页、吃饭睡觉…
- 作业执行完毕，查看输出文件

调试程序

- 使用小量数据
- 使用交互模式 `qsub –I –l nodes=1`
- 单元测试，即先测试好某个模块
- 打印中间结果
- **注意阅读文档！**

Be a nice cluster citizen!

- 集群上请使用调度软件！
- 不要在登录节点（251）上直接运行作业，会影响他人使用！
  - python xxx.py
  - R
  - ./your_software
  - singularity exec
  - …
- 合理申请计算资源，不要一次提交过多任务！
- 尽量使用我们提供的Python、R，避免重复安装！
- 家目录空间有限，数据和输入输出放在/mnt/data下！
- 不要动别人目录下的内容！

致谢

- 用户在发布科研成果或发布论文时，在成果中注明（例如在致谢中注明）：

  本研究工作得到中国人民大学高性能计算校级公共平台支持（Resources supporting this work were provided by High-performance Computing Platform of Renmin University of China）

- 论文被接受后请及时通知我们，学校将对标注平台支持的论文予以奖励

The work is supported by the National Natural Science Foundation of China (No. 21673286), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China (program No. 16XNLQ04), and High-performance Computing Platform of Renmin University of China.

项目申请

- 科研项目申请书中，一般都需要填写开展研究的保障条件或资源条件。如果用户有此需求，建议按照下述对高性能计算集群的介绍文字来写：

本项目承担单位中国人民大学建设有校级高性能计算集群，由学校专门的单位和专职技术人员管理和维护，同时学校实施了相应的制度来规范计算资源的管理和使用。中国人民大学高性能计算集群（rmdx-cluster）由55台计算节点（其中四路计算节点4台、双路计算节点50台）、1台存储节点和1台管理节点组成，总CPU核数1456颗，内存共4608GB，存储空间150TB，理论计算峰值25625 GFLOPs。集群设备、技术支持、管理制度三位一体，为本项目研究工作的顺利开展提供了基础资源的保证。

未来计划

- Linux使用入门：vim、shell脚本
- Python入门
- Python数据科学入门
- 并行编程入门
- …

如有任何问题，请联系我们！

在线手册：http://183.174.229.251/cluster-support/

电话： 010-82503969

邮箱：hpc@ruc.edu.cn

微信：

鲁蔚征 18610081936

石源 happy_tomjerry