

一、运行 R 作业

作业只能由普通用户提交

1. 编写串行作业脚本

创建文件，例如文件名是：R.pbs

```
1 #!/bin/bash
2 #PBS -N testjob
3 #PBS -l mem=1g
4 #PBS -l nodes=1:ppn=1
5 #PBS -l walltime=1:00:00
6 Rscript script.R
```

参数说明：

第一行：脚本注释，说明使用 bash 来执行这个脚本

第二~五行不是注释，是 PBS 的说明部分

第二行：作业的名字，即 testjob

第三行：作业申请的内存资源，1GB 的内存

第四行：申请一个节点，一个核。

第五行：作业运行的时间是 1 个小时，过了 1 个小时如果没有运行完，那么调度系统会把它 kill 掉。

第六行：运行程序的命令行，其中这个 script.R 是用户自己编写的 R 脚本文件。其文件路径最好写全路径而不是相对路径，如/data/user03/Test/script.R

第二~五行可以省略。

2. 提交作业脚本

使用 `qsub` 命令提交作业

```
qsub R.pbs
```

3. 查看作业状态

`qstat` 命令查看作业的状态，如果是 R 表示在运行，Q 表示在排队

使用 `checkjob ID`，查看为什么在排队

`qstat -f JobID` 查看作业的详细情况，包括提交时间，哪些节点在运行这个作业等。

`qdel ID` 删除 R 和 Q 的作业

`qhold ID` 把 Q 的作业挂起

`qr1s ID` 把挂起的作业释放

4. 安装 R 包。

(1) 从 CRAN 手动下载 R 包，并将其上传到服务器上（例如路径 `/data/user/Rlib`）；

(2) 打开 R

(3) 输入 `install.packages("/data/user/Rlib/pkgs", repos=NULL, type="source",`

`lib="/data/user/Rlib")`。（第一个参数要写 R 包所上传的全路径，最后 `lib` 的参数值为需要将 R 包安装的路径）

5. 加载 R 包。

加载 R 包。 `library(package,lib.loc = '/data/user/Rlib')`

6.并行 R 程序示例。

```
library(iterators,lib.loc="/data/user/Rlib")
library(foreach,lib.loc="/data/user/Rlib")
library(doParallel,lib.loc="/data/user/Rlib")
svd_f <- function()
{
  m <- matrix(nrow=1000, ncol=1000)
  for (i in 1 : 1000)
  {
    m[i, ] <- rnorm(1000, 1, 0.5)
  }
  svd(m)
}
cl <- makeCluster(28)
registerDoParallel(cl)
clusterEvalQ(cl, .libPaths("/data/user/Rlib"))
svd_result <- foreach(i=1:1000) %dopar%
{
  svd_f()
}
stopCluster(cl)
```

二、运行 Matlab 作业

作业只能由普通用户提交

1. 编写串行作业脚本

创建文件，例如文件名是：test.pbs

```
1 #!/bin/bash
2 #PBS -q batch
3 #PBS -N testjob
4 #PBS -l mem=1g
```

```
5 #PBS -l walltime=1:00:00  
6 cd $PBS_O_WORKDIR  
7 matlab -nodisplay -r test
```

参数说明：

第一行：脚本注释，说明使用 bash 来执行这个脚本

第二~五行不是注释，是 PBS 的说明部分

第二行：表示作业运行的队列，即 batch 队列

第三行：作业的名字，即 testjob

第四行：作业申请的内存资源，1GB 的内存

第五行：作业运行的时间是 1 个小时，过了 1 个小时如果没有运行完，那么调度系统会把它 kill 掉。

第六行：切换到提交作业的目录

第七行：运行程序的命令行，其中这个 test 是要运行的 Matlab 脚本文件。

第二~五行可以省略。

2、编写 array 脚本（重复同样的脚本若干次）

编写提交作业脚本，例如：test.pbs

```
1 #!/bin/bash  
2 #PBS -l batch
```

```
3 #PBS -l nodes=1:ppn=1
4 #PBS -t 1-20
5 cd $PBS_O_WORKDIR
6 matlab -nodisplay -r test
```

第 1 行表示下面的命令由 bash 解析器来执行

第 2 行不是注释，把作业提交到哪个队列，即 batch 队列

第 3 行不是注释，指定资源，表示我要申请的资源，即：1 个节点，每个节点 1 个核心来运行我的作业

第 4 行表示要重复运行脚本 20 次。

第 5 行切换到提交作业的目录。

第 6 行运行 test.m 的 Matlab 脚本。要省略.m。

3. 编写并行脚本

编写提交作业脚本，例如：test.pbs

```
1 #!/bin/bash
2 #PBS -l batch
3 #PBS -l nodes=1:ppn=20
4 cd $PBS_O_WORKDIR
5 matlab -nodisplay -r test
```

参数说明：

第一行：脚本注释，说明使用 bash 来执行这个脚本

第二行：表示作业运行的队列，即 batch 队列

第三行：要求资源为 20 个核。

第四行：切换到提交作业的目录

第五行：运行程序的命令行，其中这个 test 是要运行的 Matlab 脚本文件。

Matlab 脚本 test.m 如下：

```
c=parpool(20)

n = 200;
A = 500;
a = zeros(n);
parfor i = 1:n
    a(i) = max(abs(eig(rand(A))));
end
delete(c)
```

4. 安装包和调用包

Matlab 可以将别人的文件放在自己的本地目录运行。

三、运行 C++ 任务。

1. 编写串行作业脚本

```
1 #!/bin/bash

2 #PBS -q batch

3 #PBS -N testjob

4 #PBS -l mem=1g

5 #PBS -l walltime=1:00:00
```

```
6 cd $PBS_O_WORKDIR
```

```
7 /data/user04/jiaze/TopWords.exe -DataFile WLS.txt -ItemMode cen -  
MaxWordLen 6 -MinWordFreq 3 -ErrorLevelEM 1E-6 -IterationNumEM 100 -  
RoundNum_Prune 3 -IterationNumEM_Prune 10 -WordBoundaryThreshold 0.5 -  
WordSeparator 124 -MarkItemFilePath  
/data/user04/jiaze/StdMarkItemSet.txt
```

参数说明：

第一行：脚本注释，说明使用 bash 来执行这个脚本

第二~五行不是注释，是 PBS 的说明部分

第二行：表示作业运行的队列，即 batch 队列

第三行：作业的名字，即 testjob

第四行：作业申请的内存资源，1GB 的内存

第五行：作业运行的时间是 1 个小时，过了 1 个小时如果没有运行完，那么调度系统会把它 kill 掉。

第六行：切换到提交作业的目录。

第七行：运行程序的命令行。其中这个 TopWords.exe 是要运行的可执行文件，后面是用户输入的一些参数，和用户编写的 C++ 程序有关。

第二~五行可以省略。

2. 提交并行任务，使用 openMPI。

使用 openMPI 时会报错，可能与 openMPI 的版本有关。

3. 安装库与调用库。

需按照无 root 权限安装库的方法执行。参考

<https://wenku.baidu.com/view/1763a12e5ef7ba0d4b733bb4.html>

四、运行 python 任务。

1. 编写串行作业脚本

创建文件，例如文件名是：python.pbs

```
1 #!/bin/bash
2 #PBS -N testjob
3 #PBS -l mem=1g
4 #PBS -l nodes=1:ppn=1
5 #PBS -l walltime=1:00:00
6 python testpython.py
```

参数说明：

第一行：脚本注释，说明使用 bash 来执行这个脚本

第二~五行不是注释，是 PBS 的说明部分

第二行：作业的名字，即 testjob

第三行：作业申请的内存资源，1GB 的内存

第四行：申请一个节点，一个核。

第五行：作业运行的时间是 1 个小时，过了 1 个小时如果没有运行完，那么调度系统会把它 kill 掉。

第六行：运行程序的命令行，其中这个 `testpython.py` 是用户自己编写的 python 脚本文件。其文件路径最好写全路径而不是相对路径，如

```
/data/user03/Test/testpython.py
```

第二~五行可以省略。

2. 安装 python module。

1. 下载相应的模块。解压模块。
2. 进入到相应的文件夹。运行 `python setup.py install --user`