

# DAMPE 服务器用户指南

## DAMPE 服务器用户指南

科大 DAMPE 服务器进行了升级改造，系统版本从 CentOS-6 升级到 CentOS-7，因此 DAMPE software 及依赖软件进行了一系列的重装。目前，DAMPE software 已经能够正常运行，HTCondor 作业调度系统也已经准备好。

关于服务器的使用，这篇文档将做一些简单介绍。

### 关于旧系统环境下编译的软件

由于之前的服务器操作系统为 CentOS 6，这次升级到 CentOS 7 在 Linux 内核上就是一个不小的变动，很多的软件在编译安装的时候都依赖于系统内核。之前有些用户可能为了满足自己的需求，自行安装了一些软件例如 ROOT、Geant4 等。

现在如果还想使用之前的软件，需要重新编译，如果暂时不需要重新编译，可以使用默认安装的软件，用户默认加载下面的命令到配置文件中

```
source /export/setenv/setenv.sh
```

该配置文件指向了服务器目前编译安装的大部分软件：

```
#!/bin/bash

#module load gcc-4.9.2
export LD_LIBRARY_PATH=/usr/local/lib:/usr/local/lib64:$LD_LIBRARY_PATH
PATH=/usr/local/Trolltech/Qt-4.8.7/bin:$PATH

export PATH

source /export/Install/root-5.34/bin/thisroot.sh
source /export/Install/geant4.10.02.p01/share/Geant4-10.2.1/geant4make/geant4make.sh
source /export/Install/DMPSW-Install/bin/thisdmpsw.sh
source /export/Install/xerces-c-3.1.1/bashrc
source /export/Install/CLHEP-2.3.1.0/bashrc
```

```
source /export/Install/cmake-3.7.2/bashrc
source /export/Install/fftw/bashrc
```

## 关于 Condor 提交作业

### Condor 提交作业模板

简单来说，利用 Condor 提交作业就是把前台运行程序的命令，通过格式化的脚本，批量提交到后台。例如下面的前台命令是执行 JobOpt\_MipSel.py，后面接上两个参数，一个是日期对应的 file list 目录及文件名，另一个是生成文件的目录和文件名（一部分）。

```
./JobOpt_MipSel.py newfilelist/201602/20160212_filelist CombinedSelFile/test
```

考虑到 file list 是一天一天排序有规律可循，可以批量化提交作业。Condor 提交作业的模板有很多，用户可以自行研究其他模板，下面介绍其中一种模板，文件目录：

```
/home/lcm/Job_cmd # 文件目录

$lcm:Job_cmd$ tree # 目录结构
.
├─ exam.cmd # 作业描述文件
├─ FCP_Orbit_Sel.sh # 作业执行文件
├─ printfile # 输出文件目录，不同类型可以指派到不同目录
├─ 0.err # error 日志文件，程序出错之后会将错误信息写入
├─ 0.log # Condor 调用资源的日志文件，标明作业运行时间，占用内存等
├─ 0.out # 程序运行时的输出文件，例如 cout 信息流
├─ 1.err
├─ 1.log
├─ 1.out
├─ 2.err
├─ 2.log
├─ 2.out

1 directory, 11 files
```

- 作业执行文件，这个脚本可以根据用户的作业进行调整，不做过多说明

```

1 #!/bin/bash
2
3
4 Month=201603
5 process=$1
6 ((id=process+1))
7 if [[ $id -lt 10 ]];
8 then
9     ID=${Month}0${id}
10 else
11     ID=${Month}${id}
12 fi
13
14 cd ../
15 ./JobOpt_MipSel.py ./newfilelist/${Month}/${ID}_filelist
    CombinedSelFile/${Month}/Rawdata_${ID}_test 16
17 #sleep 3
18

```

- 作业描述文件 exam.cmd

```

1 Universe           = vanilla           # 声明作业类型
2 Notification       = Never            # 不发送通知
3 GetEnv             = True             # 复制用户环境配置
4 nice_user         = True             # 降低当前作业优先级
5 next_job_start_delay = 3             # 提交一个作业后停顿
6 Executable        = ./FCP_Orbit_Sel.sh # 作业执行文件
7 Arguments         = $(Process)       # 传入参数, 这里是进程
8 Output            = ./printfile/$(Process).out
9 Error             = ./printfile/$(Process).err
10 Log              = ./printfile/$(Process).log
11 Queue            = 3                 # 提交作业数量

```

- 提交作业

```
condor_submit exam.cmd
```

## 常用命令

```

condor_q # 查看当前用户作业运行情况
condor_q -submitter <user> # 查看 user 的作业运行情况
condor_q -allusers # 查看所有用户的作业运行情况
condor_rm <user> # 删除用户 <user> 的所有作业
condor_rm <JobID> # 删除自己的某一个作业

```

```
condor_status # 查看服务器节点核心占用情况

condor_userprio -most # 查看用户作业优先级
condor_userprio -all # 查看正在作业的用户优先级
condor_userprio -allusers #查看所有提交过作业的用户优先级
# 以上三个命令相似，不同在于展示的参数，具体优先级参数的含义可以参考 HTCondor 的用户手册。注意这里优先级由计算机自行调配，用户切勿随意更改
```

## 作业优先级问题

```
condor_userprio -setfactor <user>@dampe.local 1000
# 设置用户优先级，参数有用，如果提交作业太多，或者占用核心时间过长可以使用这个命令降低优先级（增大最后的参数，例如 1e+11），作业运行完毕可以调回 默认参数 1000
```

## nice\_user 设置

主动降低作业优先级，在exam.cmd 文件中主动加上下面这句

```
nice_user = True
```

适用条件：

- **模拟作业一律加上 nice\_user**

Example file: exam.cmd

```
1 Universe           = vanilla
2 Notification       = Never
3 GetEnv             = True
4 nice_user          = True
5 next_job_start_delay = 3
6 Executable         = ./FCP_Orbit_Sel.sh
7 Arguments          =$(Process)
8 Output             =./printfile/$(Process).out
9 Error              =./printfile/$(Process).err
10 Log                =./printfile/$(Process).log
11 Queue             31
```

因为由 nice\_user 的设计，用户在提交作业之前如果发现服务器资源被占满，可以先通过命令 condor\_userprio 查看是否有 nice\_user 的作业，如果有，则可以酌情提交作业

```
lcm:Job_cmd$ condor_userprio
Last Priority Update: 1/5 20:47
User Name                Effective Priority Res Total Usage Time Since
                        Priority Factor In Use (wghted-hrs) Last Usage
-----
wulb11@dampe.local      27646.61 1000.00 95 1322.77 <now>
yzhou@dampe.local      54685.07 1000.00 1 2380.44 <now>
weiyf@dampe.local      70358.88 1000.00 29 1573.55 0+00:05
nice-user.weiyf@dampe.local 6.16286e+09 1e+10 200 4.04 <now>
nice-user.wulb11@dampe.local 8.83301e+10 1e+10 200 311.63 <now>
-----
Number of users: 5                    525 5592.43 0+23:59
```

有些同学登录服务器的时候碰到密钥错误的问题，登录不上，会出现在利用 MacOS 或者 Linux 终端内网登录 DAMPE 和外网通过 UI 登录 DAMPE 的过程中，如附件中的图片所示。

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
SHA256:LkUg1F [REDACTED].
Please contact your system administrator.
Add correct host key in /Users/[REDACTED]/.ssh/known_hosts to get rid of this m
essage.
Offending RSA key in /Users/[REDACTED]/.ssh/known_hosts:1
RSA host key for 202.141.163.196 has changed and you have requested strict check
ing.
Host key verification failed.
```

原因是：服务器重装了系统，原有的密钥已经失效，登录需要建立新的密钥，但是服务器 IP 地址又没变，导致密钥无法顺利建立。

这个问题的解决办法是：

- **Mac & Linux 终端直接登录 DAMPE**，在用户主目录下，利用文本编辑器打开 **.ssh/known\_hosts** 删除旧的内网登录 DAMPE 的密钥，重新登录即可。
- **外网通过 UI 登录 DAMPE**，在 UI 的个人目录下，同样利用文本编辑器打开 **.ssh/known\_hosts** 删除旧的内网登录 DAMPE 的密钥，重新登录即可。

**注意：**密钥的开头会有服务器的 IP 或者域名，连接多个服务器的用户，请注意甄别！

另外，Windows 系统利用 XShell 登录，似乎会自动更新密钥，暂时不会碰到这个问题。