

# Start your work in BESIII

李小康\*

高能所计算集群个人目录: `/afs/ihep.ac.cn/users/l/lixk33`

2025 年 2 月 26 日

- 2024-03-01 v2.0

- 2025-02-25 v3.0

由于高能所计算集群的 Linux 发行版从 CentOS 改成 Alma, 交作业脚本有变动。

---

\* 在使用本指导文档的过程中, 有任何问题可以通过邮箱联系本人 2216183193@qq.com。

# 目录

<b>1</b>	<b>申请账号</b>	<b>3</b>
<b>2</b>	<b>远程访问</b>	<b>4</b>
<b>3</b>	<b>简单指令</b>	<b>5</b>
<b>4</b>	<b>配置环境</b>	<b>7</b>
4.1	获取 BOSS . . . . .	7
4.2	配置 TestRelease . . . . .	9
4.3	快速配置环境 . . . . .	9
<b>5</b>	<b>简单事例分析</b>	<b>9</b>
<b>6</b>	<b>进阶事例分析</b>	<b>11</b>
6.1	模拟和重建 . . . . .	11
6.1.1	decaycard . . . . .	11
6.1.2	初始化 . . . . .	12
6.1.3	运行作业 . . . . .	12
6.2	分析 . . . . .	13
6.2.1	dstlist . . . . .	14
6.2.2	编译算法包 . . . . .	14
6.2.3	运行作业 . . . . .	15
<b>7</b>	<b>文件管理</b>	<b>16</b>

# 1 申请账号

在[高能所统一认证系统](#)注册账号，以登录高能所允许的相关网站浏览内容。

### Registration

\*

Email

Email Account

\*

Name

First Name or F

lastName

\*

Gender

☒ Male

☐ Female

\*

Identity

IHEP Users ☐ Yes ☐ No

\*

Password

Account Password

Weak

Good

Strong

\*

Confirm Password

Country or Region

Asia:China

▼

\*

Home Institute

\*

Security Control

Collaboration

Contact Person

\*

Image Text

68xn

Refresh

如上图，其中选择

- Identity : No→Student
- Home Institute : Sun Yat-Sen University
- Security Control : Collaboration→BES3

高能所统一认证账号注册成功后，如下图，还需要申请 AFS 集群账号，以获得在高能所计算集群上运行作业的权限。

## 账号安全



### 密保邮箱 (已设置)

设置并验证密保邮箱后，您可以使用密保邮箱找回密码。

lixk33@mail2.sysu.edu.cn

[更改](#)



### VPN服务

申请VPN，您可以使用VPN账号远程办公。[高能所人员申请VPN](#) [非高能所人员申请VPN](#)



### 申请集群账号

计算集群服务	实验组	申请时间
AFS	BES	2021-10-02 21:02:55

## 2 远程访问

访问高能所的计算集群需要借助 Linux 操作系统<sup>1</sup>，因此首先需要有一个 Linux 终端。对于 Windows 用户<sup>2</sup>，如果你对 Linux 不熟悉<sup>3</sup>，推荐使用以下软件组合<sup>4</sup>

- Xshell：提供 Linux 终端，用于访问高能所计算集群。
- Xftp<sup>5</sup>：提供类 Windows 的文件夹图形页面，便于上传、下载文件。
- Xming：提供图形界面，用于显示作图脚本的结果。

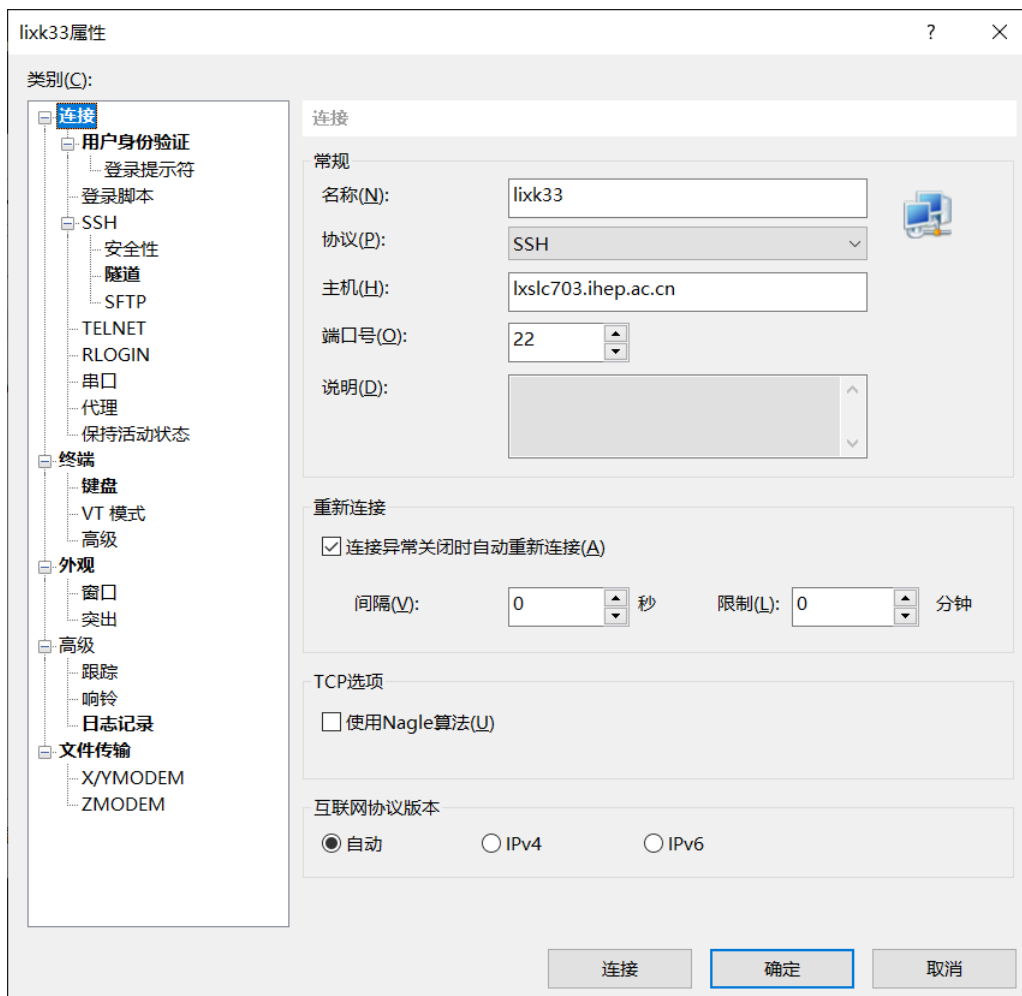
<sup>1</sup>Linux 在安全性和管理便捷性都有不可比拟的优势，尤其是在处理高能物理实验的海量数据时。

<sup>2</sup>macOS 用户的终端本身就是与 Linux 兼容的 Unix，不需要借助其他软件，具体参考[高能所计算环境使用手册-3.1.2.2 macOS 下登录集群](#)。原生的 mac 终端没有那么方便，建议辅助软件 iTerm 使用

<sup>3</sup>如果你想在 Windows 系统上使用更加原生的 Linux 终端，Windows Subsystem for Linux (WSL) 是一个不错的选择，需要自行折腾。

<sup>4</sup>Xshell 和 Xftp 有[家庭/学校免费版](#)，而 Xming 本身就是免费的。Xshell 和 Xftp 会自动绑定，而 Xshell+Xming 的绑定则需要手动操作，具体自己上网查。

<sup>5</sup>Xftp 不是必须的，因为 Linux 操作系统可以通过指令上传和下载文件。



打开 Xshell 以后新建会话，就会出现上图所示的窗口，其中

- 名称：随意
- 主机：beslogin.ihep.ac.cn
- （左侧栏）用户身份验证：高能所注册的用户名和密码

点击连接即可成功访问高能所。

任何远程访问方法都至少要实现三个功能

- ssh 登录（能够记录要登录的主机、账号、密码、登录指令）
- 图形转发（通常使用 x11）
- 文件传输（通常使用 sftp）

### 3 简单指令

Linux 的基础是命令行模式，需要输入指令与操作系统交互，有别于 Windows 系统图形界面的鼠标点击为主。以下只罗列一些常用的指令。

- `ls` 即 `list`，以下指令会展示当前目录下的文件夹和文件。

```
|ls
```

- `cd` 即 `change directory`，以下指令会从当前目录移动至目录 `directory_name` 目录下。

```
|cd directory_name
```

- `pwd` 即 `print working directory`，以下指令会打印当前所在目录。

```
|pwd
```

- `mkdir` 即 `make directory`，以下指令会在当前目录下创建一个目录 `directory_name`。

```
|mkdir directory_name
```

- `rm` 即 `remove`，以下指令会删除目录 `directory_name`。若不加 `-r` 则是删除文件，对下面的 `cp` 指令同理。

```
|rm -r directory_name
```

- `cp` 即 `copy`，以下指令将目录 `directory_name` 下的 `file_name` 复制到当前目录下<sup>6</sup>。

```
|cp directory_name/file_name .
```

- `vim` 是 Linux 系统中的一种编辑器，以下指令打开文件 `file_name`。

```
|vim file_name
```

学习该编辑器需要另外的知识，以下只罗列最必要的功能。

- 按 `i` 键——进入插入模式，此时可以对文件进行编辑。
- 按 `Esc` 键——进入命令行模式，此时不能对文件进行编辑。
- 输入 `:q` ——在命令行模式下，该操作可以退出文件。若输入 `:wq`，则保存修改后退出。

---

<sup>6</sup>注意最后有一个英文句号`.`，代表当前目录。`..` 则代表上一级目录。

## 4 配置环境

主要的目录如下

- /: 整个服务器的根目录。
- ~: 500MB, 个人的 home 目录<sup>7</sup>, 一般用于储存 cmthome<sup>8</sup>和方便实用的 shell 脚本<sup>9</sup>。
- /workfs2/bes/\$USER: 5GB, 一般用于储存算法包<sup>10</sup>。
- /besfs?/users/\$USER: 其中? 处可能是 5, 该目录大小未知, 有需要再用。
- /scratchfs/bes/\$USER: 500GB, 一般用于储存大量的作业。

学会管理自己的目录系统, 合适的命名和分级是十分重要的, 可以到各位学长学姐的目录下借鉴。这对于工作能否顺利进行有重大影响。此外, 文件目录繁多, 写一个日志记录自己每次的重要操作将会对减少浪费时间、提升工作效率具有明显的帮助。

以下以 BOSS 7.0.8 版本<sup>11</sup>为例, 配置环境。

### 4.1 获取 BOSS

- 创建目录

其中 cvmfs-7.0.8<sup>12</sup>后续将用于存放该版本下需要使用的算法包, cmthome 用于存放不同版本的 BOSS。

```
cd /workfs2/bes/$USER
mkdir cvmfs-7.0.8
cd ~
mkdir cmthome
```

- 复制 BOSS 7.0.8 版本到个人目录

```
cd cmthome
cp -r /cvmfs/bes3.ihep.ac.cn/bes3sw/cmthome/cmthome-7.0.8 .
```

---

<sup>7</sup>其从根目录开始的绝对路径为/afs/ihep.ac.cn/users/l/lixk33

<sup>8</sup>cmt 的英文全称为 Configuration Management Tool, 就是编译管理工具。

<sup>9</sup>人与操作系统的交互通过中介 shell, 它将命令解析后让操作系统运行。.sh 结尾的文件即 shell 脚本, 是一系列 Linux 下的指令的集合, 运行 shell 脚本需要使用 source 指令。

<sup>10</sup>其中 \$USER 是已经定义的全局变量, 可以等价替换为你注册高能所统一认证账号所使用的用户名。

<sup>11</sup>BOSS 的英文全称为 BES Offline Software System。

<sup>12</sup>cvmfs 的英文全称为 CernVM File System, 这是由 CERN 开发的文件系统, 帮助高能物理在用于运行数据处理应用程序的全球分布式计算基础设施上部署软件。

- 修改

```
cd cmthome-7.0.8  
vim requirements
```

打开该文件后，在 43<sup>13</sup>行附近找到如下内容。43 行中的 `user_name` 修改为自己的用户名，去掉 46 和 47 行前的 `#`，修改结果如下。

```
43 #Add your worarea to CMTPATH  
44 macro WorkArea "/workfs2/bes/user_name/cvmfs-7.0.8/"  
45 #Add dev area to the front of your CMTPATH (but first  
    remove any previously defined devarea which is under your  
    home directory)  
46 path_remove CMTPATH "${WorkArea}"  
47 path_prepend CMTPATH "${WorkArea}"
```

修改完成后运行 shell 脚本和编译

```
source setupCMT.sh  
cmt config  
source setup.sh
```

再修改<sup>14</sup>

```
vim setupCVS.sh
```

将第一行修改为

```
1 export CVSROOT=':pserver:bes3@koala.ihep.ac.cn:/bes/bes'
```

运行 shell 脚本

```
source setupCVS.sh
```

- 检查

```
echo $CMTPATH
```

若设置正确，则以上指令运行后会输出 4 个路径，以 `:` 为间隔，且第一个路径为设置好的如下的 `$WorkArea` 目录

```
/workfs2/bes/user_name/cvmfs-7.0.8/
```

---

<sup>13</sup>vim 编辑器的命令行模式下输入 `:set nu` 可以显示行号。

<sup>14</sup>CVS 的英文全称为 Concurrent Versions System，是版本控制系统。



## 4.2 配置 TestRelease

```
cd /workfs2/bes/$USER/cvmfs-7.0.8/
cp -r /cvmfs/bes3.ihep.ac.cn/bes3sw/Boss/7.0.8/TestRelease/ .
cd TestRelease/TestRelease-00-00-95/cmt
cmt config
source setup.sh
echo $TESTRELEASEROOT
```

若成功则会输出算法路径

/workfs2/bes/\$USER/cvmfs-7.0.8/TestRelease/TestRelease-00-00-95

## 4.3 快速配置环境

将配置环境的指令写成一个 shell 文件，每次登录计算集群时运行一次，即可快速配置环境。

```
cd ~
cp /afs/ihep.ac.cn/users/l/lixk33/bes.sh .
```

将 shell 文件复制过来，将文件中的 lixk33 替换为自己的用户名。以后每次登录计算集群时运行以下指令即可。

```
source bes.sh
```

此外，创建软链接，将算法包和作业的存储目录链接到个人目录下，方便后续从个人目录下进入。

```
cd ~
ln -s /workfs2/bes/lixk33 workfs
ln -s /scratchfs/bes/lixk33 scratchfs
```

## 5 简单事例分析

简单来讲，工作流程可以分为 3 个步骤：Simulation → Reconstruction → Analysis。

Simulation 就是通过 Monte Carlo 方法模拟产生信号事例。Reconstruction 就是根据模拟产生的事例在探测系统上留下的电子学信息来重建物理信息（包括径迹重建、顶点重建、能量刻度等），最终达到模拟真实数据的目的。而 Analysis 就是对得到物理信息进行分析，内容比较广泛，不唯一，这也是最主要的工作。

接下来通过官方提供的一个简单事例走通这个流程。

- 使用本地的 BOSS

```
cd ~/workfs/cvmfs-7.0.8/TestRelease/TestRelease-00-00-95/run
boss.exe HelloWorldOptions.txt
```

boss.exe 即本地的 boss 程序，运行脚本，输出 successfully。

- 目标反应道  $J/\psi \rightarrow \rho + \pi$

- Simulation

.dec 文件指定所要模拟的反应道和理论模型<sup>15</sup>，可以打开来浏览一下。

```
vim rhopi.dec
```

jobOptions 文件中包含运行作业的指令，因此需要指定反应道（即 .dec 文件目录）、使用的算法包目录等等相关的目录。模拟、重建和分析的作业都需要使用到。

模拟没有输入，运行根据理论模型生成一定量  $J/\psi \rightarrow \rho + \pi$  的 Monte Carlo<sup>16</sup> 数据，结合探测器响应，输出探测的原始信息 .rtraw 文件

```
boss.exe jobOptions_sim.txt
```

→

rhopi.rtraw

- Reconstruction

重建需要输入模拟的结果，根据电子学信息重建物理信息，输出 .dst 文件

```
boss.exe jobOptions_rec.txt
```

→

rhopi.dst

- Analysis

分析需要输入重建的结果，对物理信息进行处理，输出 .root 文件

```
boss.exe jobOptions_ana_rhopi.txt
```

→

rhophi\_ana.root

使用 ROOT<sup>17</sup> 打开分析结果，作直方图观察结果<sup>18</sup>

```
root -l rhopi_ana.root
```

```
fit4c->Draw("mpi0")      \\ pi0 的不变质量
```

---

<sup>15</sup> 本例中只产生一个反应道的数据。对于产生理论上所有已知反应的模拟，其 .dec 文件是极其冗长的，因此有官方版本。

<sup>16</sup> 由于量子理论的结果都具有统计性，因此产生的数据也要满足一定的随机性，随机方法称为 Monte Carlo 方法，简称 MC。

<sup>17</sup> ROOT 是 CERN 基于 C++ 开发的专用于处理高能物理实验海量数据的分析框架，可以认为是 C++ 编译器附带专用自定义类和算法包。其使用技巧繁多，需要查阅 ROOT 官网学习。另外，ChatGPT 也具有 ROOT 相关知识，因此直接询问 ChatGPT 可以免去阅读参考文档，大大提高效率。由于 ROOT 是基于 C++ 语言的，因此对于 C++ 的类和指针等概念要有简单的认识。

<sup>18</sup> root 指令进入 ROOT 环境中的命令行模式，.q 退出。ROOT 环境下指令一般前面都要加 .。

```
fit5c->Draw("mrhp")      \\ rho0+pi0的不变质量
.q
```

至此完成了一个简单的分析。

## 6 进阶事例分析

上一节通过官方提供的一个简单的例子介绍了数据处理的 3 个步骤，其中几乎所有需要的文件、脚本、结果都放在同一个目录下，且都是通过本地的 BOSS 运行。在自己实际分析的过程中，由于文件繁多，所需要建立的目录系统是较为复杂的，需要次第搭建。而大量的数据处理作业则需要提交到计算集群进行处理，等待返回结果。

需要处理的数据分为 3 个部分：sigmc<sup>19</sup>、incsmc<sup>20</sup>、data。

其中 sigmc 就是需要寻找或测量的目标反应道 MC 数据，数据量较小，可以自行模拟-重建-分析；incsmc 是官方产生的包含母粒子所有反应道的 MC 数据，数据量庞大，因此官方已经完成模拟-重建，给出.dst 文件，只需要分析；data 是探测器运行的真实数据，数据量同样庞大，也给出.dst 文件。data 的处理类似 incsmc，但一般需要分析完 MC 数据后方可使用，刚开始用不到。

接下来以寻找  $J/\psi \rightarrow \bar{D}^0 + \bar{K}^{*0}$  为例，介绍 09 年的 sigmc 的处理。该分析的主目录如下，所需要的所有脚本都可以复制到自己对应的目录下。没有的目录自己创建后再复制过来。<sup>21</sup>

```
/afs/ihep.ac.cn/users/l/lixk33/scratchfs/work/D0Kstar0
```

### 6.1 模拟和重建

#### 6.1.1 decaycard

decaycard 就是描述计算机在产生信号道时，整个衰变的拓扑结构，包括母粒子衰变成什么中间态粒子，中间态粒子衰变成什么末态粒子，每个衰变过程又依据什么衰变模型。由于中间态粒子一般是不稳定且寿命极短的，而探测器只能探测到  $e\mu\pi Kpn$  等长寿的末态粒子，因此需要根据末态去重建中间态粒子。

先将下面总的 decaycard 复制到自己对应目录下<sup>22</sup>，其中包含系统默认的总 DECAY.DEC 文件。

```
/afs/ihep.ac.cn/users/l/lixk33/decaycard/decaycards
```

<sup>19</sup>英文全称为 Signal Monte Carlo。

<sup>20</sup>英文全程为 Inclusive Monte Carlo。

<sup>21</sup>总的原则是所有复制过来的文件都可以打开看看，将调用的目录复制过来，再将调用目录改为自己的目录。

<sup>22</sup>如无特别说明，均需复制到自己的对应目录下，以免出错。

$$J/\psi \rightarrow \begin{cases} \bar{D}^0 \rightarrow K^+ e^- \bar{\nu}_e \\ \bar{K}^{*0} \rightarrow K^- \pi^+ \end{cases}.$$

自定义 decaycard 文件目录如下<sup>2324</sup>。本例选择用  $K^+ e^- \bar{\nu}_e$  去重建  $\bar{D}^0$ ，用  $K^- \pi^+$  去重建  $\bar{K}^{*0}$ <sup>25</sup>。复制过来。

```
/afs/ihep.ac.cn/users/l/lixk33/scratchfs/work/D0Kstar0/sigmc/09/decaycard
/D0Kstar0.dec
```

### 6.1.2 初始化

进入如下的 09 年 sigmc 目录，复制一些必要的目录过来，并做一些修改。

```
/afs/ihep.ac.cn/users/l/lixk33/scratchfs/work/D0Kstar0/sigmc/09/
→ sim_ini/
→ rec_ini/
→ ana_ini/
```

- sim\_ini/run\_template\_tail.txt 中需要将调用的 D0Kstar0.dec 文件改为自己路径下的。
- 将其最后一行的每份作业事例数改为 1000。
- rec\_ini 不需要改动。

### 6.1.3 运行作业<sup>26</sup>

首先是产生和测试作业。

- 复制一些有用的脚本  
将如下的目录复制过来，里面放着产生作业、检查作业的脚本。复制过来之后重新 source bes.sh，更新 /bin 文件夹内的脚本到环境变量。  
/afs/ihep.ac.cn/users/l/lixk33/bin
- 在 sigmc/09 目录下，用以下指令产生作业
- 用以下指令<sup>27</sup>产生模拟和重建的作业

<sup>23</sup>在上面的目录中也有，但在每年的 sigmc 目录下还放了一个，方便使用。

<sup>24</sup>注意：其中/afs/ihep.ac.cn/users/l/lixk33/下的 scratchfs 是软链接而非目录，其链接的绝对路径是/scratchfs/bes/lixk33。

<sup>25</sup>当然，这样并不能得到所有的  $\bar{D}^0, \bar{K}^{*0}$ ，但是没有关系，在计算分支比的时候会考虑到这一点。

<sup>26</sup>本小节有较大改动，如果你发现生成结果与我对应目录下的结果不同，不必担心，这很可能是自然现象。此外，因为工作遗留问题，我的目录下可能放着很多无关的脚本。如无必要，请勿无脑复制，只按照本教程指引复制有用的文件即可，以保持目录整洁。

<sup>27</sup>该指令在 v3.0 中更新。

```
| gensimrec.sh job_name job_num
```

→ joblist/\*\_sim\_\*.txt

→ joblist/\*\_rec\_\*.txt

job\_name 可用信号道反应命名, job\_num 可设为 20 份作业, 其中 \* 代表不定、任意。以上操作还会产生 3 个文件夹

→ rtraw/ dst/ joblist/

- 在正式提交之前, 修改每份作业的产生事例数为 100, 进行本地低事例测试。用以下指令运行 1 个模拟作业, 没有输入, 输出原始的电子学信息文件。

```
| boss.exe *_sim_0.txt
```

→ rtraw/\*\_sim\_0.rtraw

再用以下指令运行重建的作业, 输入 rtraw 文件, 输出重建的物理信息文件。

```
| boss.exe *_rec_0.txt
```

→ dst/\*\_rec\_0.dst

以此确保编译运行正常, 然后将单份作业事例数改回 1000。

接着是提交和检查作业。

- 进入 09/joblist 目录, 用以下指令提交作业。

```
| source subjob
```

在作业运行期间, 可以用以下快捷指令查询作业完成情况。

```
| hep_q -u $USER
```

- 全部作业完成后, 用以下快捷指令<sup>28</sup>检查作业完成情况, 成功还是失败。

```
| cks
```

注意, 同名的结果返回时是会覆盖的, 因此需要及时保存。

## 6.2 分析

前一节仅针对 sigmc, 从这一节开始, 还要考虑 incsmc。只是由于后者数据量庞大, dst 文件已经由官方产生并保存<sup>29</sup>。

---

<sup>28</sup>该快捷指令运行的脚本是 check\_simrec.sh, 存放在 ~/bin 中, 快捷指令的定义在 bes.sh 中。

<sup>29</sup>在 [Offline Software Group](#) 的网站上可以查询调用。但是本例中已经生成好 dstlist, 只需要复制即可, 不需要上。

### 6.2.1 dstlist

接下来合并 dst 文件目录，生成 dstlist，用于分析作业的调用。对于 sigmc，进入 sigmc/09 目录，将目录换为自己存放 dst 文件的目录，用以下指令将 dst 目录下的 dst 文件的绝对目录生成 dstlist。

```
ls 个人存放dst文件的路径/*.dst > dstlist
```

→

sigmc/09/dstlist

### 6.2.2 编译算法包

在实际操作中，常常是先用成熟的算法包测试通过整个流程，在此基础上再去考虑修改。

- D0Kstar0Alg

- 进入以下路径，复制所需要的两个算法包<sup>30</sup>

/afs/ihep.ac.cn/users/l/lixk33/workfs/cvmfs-7.0.8

→

McDecayModeSvc/ D0Kstar0Alg/

- 进入 D0Kstar0Alg/D0Kstar0Alg-00-00-01/cmt 目录下，用以下指令编译算法包。另一个算法包同理。最后的运行结果中显示 successfully 即可。

```
cmt config
source setup.sh
cmt make
```

- TestRelease<sup>31</sup>

- 调用新增的算法包，进入/TestRelease/TestRelease-00-00-95/cmt 目录下，打开 requirements 文件，在 for Analysis 一栏下添加两行

```
#===== for Analysis =====
use      McDecayModeSvc  McDecayModeSvc-*
use      D0Kstar0Alg     D0Kstar0Alg-*
```

- 在 TestRelease 的 cmt 目录下，按照 D0Kstar0Alg 的方法编译。

每次修改算法包都需要重复以上的编译过程，新的算法才能生效。

---

<sup>30</sup>前者是普遍需要的，后者本分析专用的。

<sup>31</sup>这是调用所有算法包的总算法包。

### 6.2.3 运行作业

首先是产生和测试作业。

- 在 `sigmc/09` 目录下用以下指令<sup>32</sup>产生承担分析任务的作业

```
|split.sh filelist job_num
```

→ `joblist/*_ana_*.txt`

其中 `filelist` 就是前面生成的 `dstlist`。对于 `job_num` 和之前相同。以上操作还会产生 2 个文件夹

→ `root/`

→ `joblist_in/`

- 对于 `sigmc`，还可以接着进行低事例测试，以验证算法包正常。  
在 `sigmc/09` 目录下，用<sup>33</sup>6.1.3产生的低事例 `*_rec_0.dst` 文件，参照<sup>34</sup>6.2.1的方法生成 `dstlist`。然后根据以上方法生成分析的作业，用以下指令在本地运行作业，输入 `*_rec_0.dst`，输出分析结果 `root` 文件。

```
|boss.exe ana_*.txt
```

→ `root/ana_*.root`

接着是提交和检查作业。

- 进入 `sigmc/09/joblist` 目录，参照<sup>35</sup>6.1.3使用相同的指令提交和查询作业。
- 全部作业完成后，会生成分析结果。

→ `root/ana_*.root`

在 BOSS 7.0.5<sup>34</sup>版本中可以用以下指令打开图形窗口查看 `root` 变量的分布直方图。

```
|root -l rootfile  
|TBrowser t
```

其中 `rootfile` 是运行结果 `root` 文件的文件名。

- `09/joblist` 目录下中以下快捷指令<sup>35</sup>检查作业完成情况。

```
|cka
```

---

<sup>32</sup>该指令在 v3.0 中更新。

<sup>33</sup>指令中的 `*.dst` 要换为具体的测试 `.dst` 文件，`dstlist` 中应该只有一个 `dst` 文件。

<sup>34</sup>想用的话，需要自己另外配置，但不是必须的。

<sup>35</sup>该快捷指令运行的脚本是 `check_ana.sh`，`~/bin` 中。

最后是合并结果并作图。

- 在 `sigmc/09` 目录下打开脚本 `hadd_all.C` 修改 `chain -> Add("*")` 的路径为自己得到的 `*_ana_*.root`, 用以下指令运行脚本<sup>36</sup>, 合并分析作业返回的 `root` 文件

```
| root -l hadd_all.C
```

→

`all.root`

- 作图

```
| root -l all.root
|.ls                                \\ 展示root文件的内容
|d0kstar0->GetEntries()             \\ 获取事例数
|d0kstar0->Show(0)                  \\ 展示第一个事例的信息
|d0kstar0->Draw("D0_m>>(60,1.82,1.91)") \\ 作D0质量的直方图
```

至此完成了 `sigmc` 的初步分析。

## 7 文件管理

从 `v2.0` 到 `v3.0`, 由于提交作业的脚本的改变, 目录会变得比较繁杂。也许在完成上述步骤之后你已经发现了。而本教程撰写之初, 尚且处于 `v2.0` 的脚本, 因此本教程的示例分析中目录结构也是旧貌。建议移步以下目录参考新的目录结构

`/afs/ihep.ac.cn/users/l/lixk33/scratchfs/work/RCLFV/psip2etau/sigmc/09`

---

<sup>36</sup>运行 C++ 脚本使用 `root -l` 指令, 运行 shell 脚本使用 `source` 指令