中国科学技术大学
University of Science and Technology of China

# 模拟重建与Shell脚本入门

孙睿

*University of Science and Technology of China*

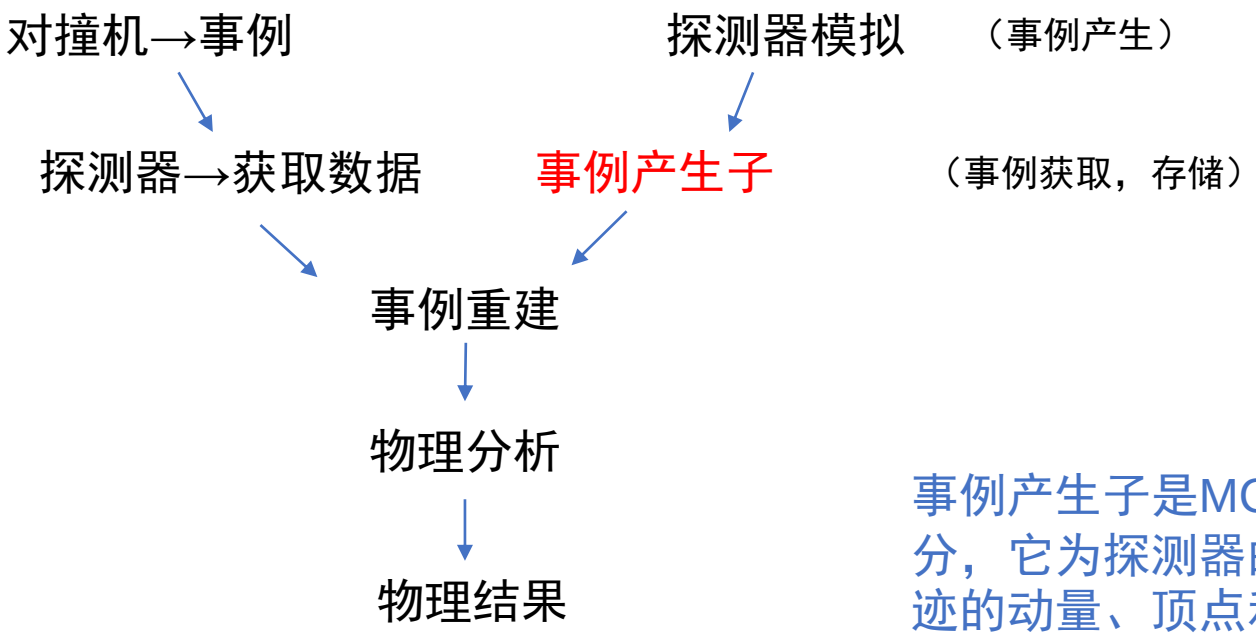
2023/7/13

# Monte Carlo方法



Monte Carlo（MC）方法是按抽样调查法求取统计值来推定未知特性量的计算方法。基于此，可以对复杂的物理系统进行模拟，以近似得到真实的情形。

粒子物理实验中的一般流程




事例产生子是MC模拟的一个重要组成部分，它为探测器的事例模拟提供各条径迹的动量、顶点和粒子种类的信息

# 信号MC产生过程



编写产生子（一般用现成的即可）

↓

编写衰变卡（decay card）

↓ .dec

模拟（simulation）

↓ .rtraw

重建（reconstruction）

↓ .dst

分析（analysis）.cxx

↓

.root

# 模拟



## $TESTRELEASEROOT/run/jobOptions_sim.txt

```
1  //DENG Zi-yan 2008-03-17
2
3  #include "$OFFLINEEVENTLOOPMGRROOT/share/OfflineEventLoopMgr_Option.txt"
4
5  //*************job options for generator (KKMC)***************
6  #include "$KKMCROOT/share/jobOptions_KKMC.txt"
7  KKMC.CMSEnergy = 3.097;
8  KKMC.BeamEnergySpread=0.0008;
9  KKMC.NumberOfEventPrinted=1;
10 KKMC.GenerateJPsi=true;
11
12 //*************job options for EvtGen***************
13 #include "$BESEVTGENROOT/share/BesEvtGen.txt"
14 EvtDecay.userDecayTableName = "rhopi.dec";
15
16 //*************job options for random number***********************
17 BesRndmGenSvc.RndmSeed = 100;
18
19 //*************job options for detector simulation*****************
20 #include "$BESSIMROOT/share/G4Svc_BesSim.txt"
21
22 //configure for calibration constants
23 #include "$CALIBSVCROOT/share/calibConfig_sim.txt"
24
25 // run ID
26 RealizationSvc.RunIdList = {-9989};
27
28 #include "$ROOTIOROOT/share/jobOptions_Digi2Root.txt"
29 RootCnvSvc.digiRootOutputFile = "rhopi.rtraw";
30
31
32 // OUTPUT PRINTOUT LEVEL
33 // Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
34 MessageSvc.OutputLevel  = 5;
35
36 // Number of events to be processed (default is 10)
37 ApplicationMgr.EvtMax = 50;
```

产生子

束流质心能量

束流能散

共振态产生类型

衰变卡

随机数种子

Run号

产生文件的路径

信息输出等级

事例数

Decay 母粒子的名称
Br x1 x2 ..xn  衰变模型 (参数);
Enddecay……End

根据需要使用的data，找到对应的run号填在{}中。
例如：{-9947,0,-10878}
查询网址：
https://docbes3.ihep.ac.cn/~offlinesoftware/index.php/Production

# 模拟



衰变卡（$TESTRELEASEROOT/run/rhopi.dec)

```
Decay J/psi
  0.3333 rho0 pi0 HELAMP 1.0 0.0 0.0 0.0 -1.0 0.0;
  0.3333 rho+ pi- HELAMP 1.0 0.0 0.0 0.0 -1.0 0.0;
  0.3333 rho- pi+ HELAMP 1.0 0.0 0.0 0.0 -1.0 0.0;
Enddecay

End
```

注：1.衰变卡片中粒子的名称必须按照EvtGen粒子表(可见: $BESEVTGENROOT/share/pdt.table )中的定义填写。

2.衰变模型必须是EvtGen中的注册模型(可见: $BESEVTGENROOT/share/DECAY.DEC )，其引用及参数必须按照手册中的格式要求填写。

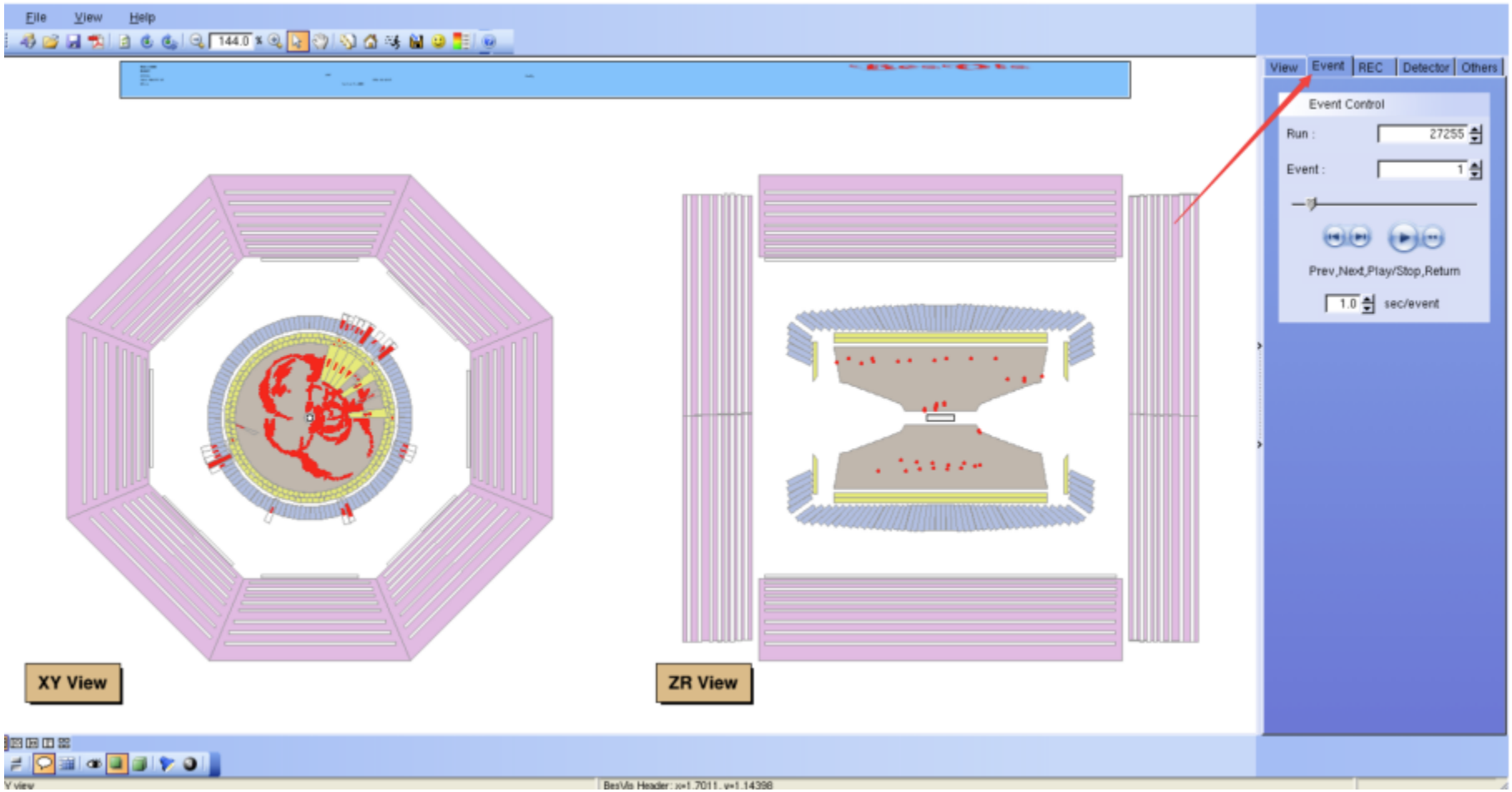
3.如果某个母粒子的衰变道分支比之和不等于1，EvtGen平台将会对这些道的分支比重新归一。

4.如果在DECAY.DEC中找不到的过程，暂用PHSP模型。

# 模拟



模拟产生.rtraw文件。可用besvis.exe查看图像



# 重建



$TESTRELEASEROOT/run/jobOptions_rec.txt

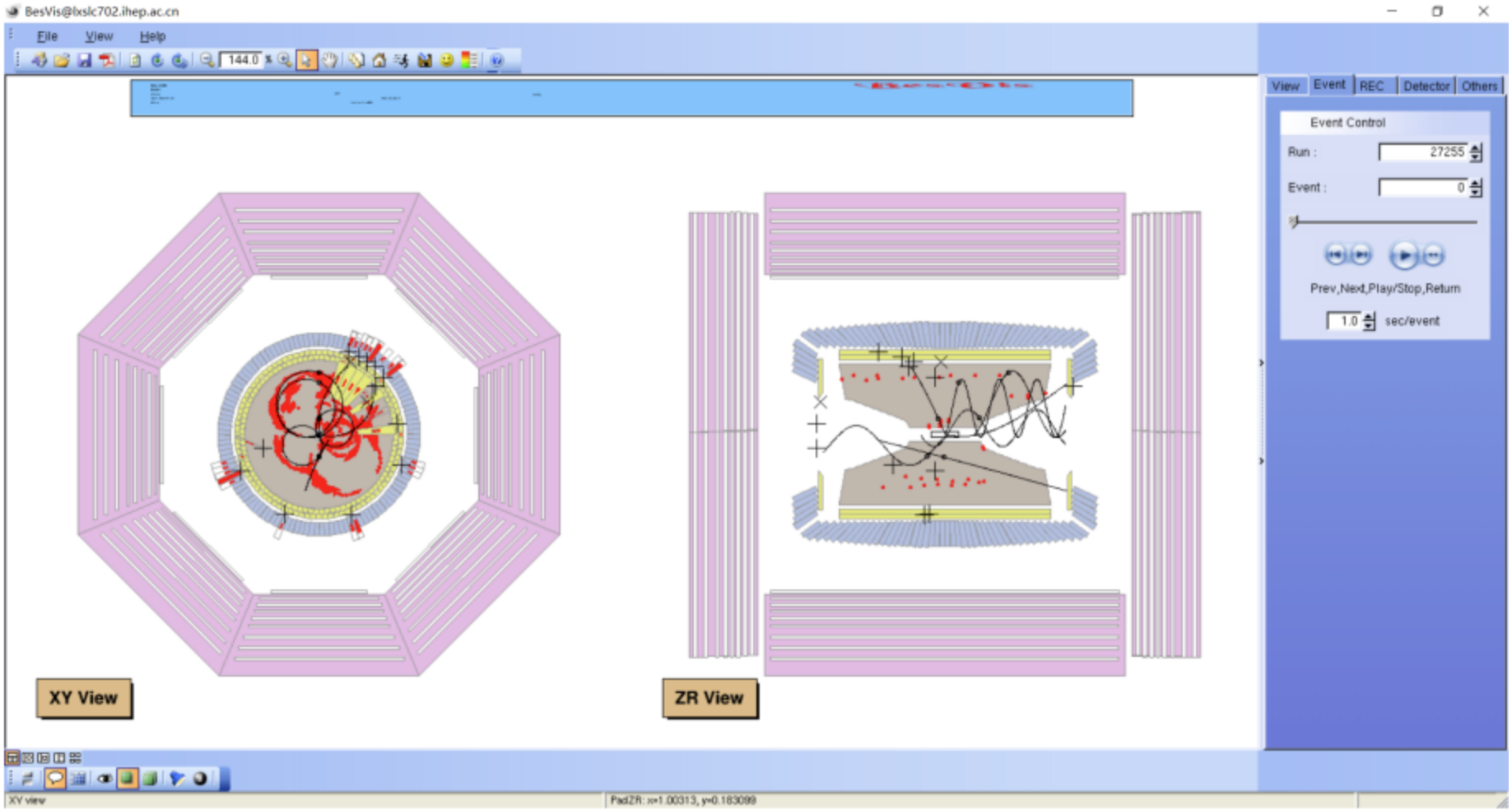
```
//**************job options for random number**************************
BesRndmGenSvc.RndmSeed = 100;     随机数种子，与对应的sim文件保持一致

//Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
MessageSvc.OutputLevel = 2;   信息输出等级

//ROOT input data file
EventCnvSvc.digiRootInputFile = {"rhopi.rtraw"}; 输入文件，即模拟的输出文件

//ROOT output data file
EventCnvSvc.digiRootOutputFile ="rhopi.dst";   模拟的输出文件

//Number of events to be processed (default is 10)
ApplicationMgr.EvtMax = 50;     事例数
```

注：1.重建的事例数，如果写-1则默认与模拟文件中的事例数一致。
2.若不更改输出文件的路径，则默认为程序运行的位置。

# 重建



重建产生.dst文件。可用besvis.exe查看图像



# 分析



$TESTRELEASEROOT/run/jobOptions_rec.txt

```
1  #include "$ROOTIOROOT/share/jobOptions_ReadRec.txt"
2  #include "$VERTEXFITROOT/share/jobOptions_VertexDbSvc.txt"
3  #include "$MAGNETICFIELDROOT/share/MagneticField.txt"
4  #include "$ABSCORROOT/share/jobOptions_AbsCor.txt"
5  #include "$RHOPIALGROOT/share/jobOptions_Rhopi.txt"   放入自己的分析算法
6
7  // Input REC or DST file name
8  EventCnvSvc.digiRootInputFile = {"rhopi.dst"};   重建输出文件的路径
9
10 // Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
11 MessageSvc.OutputLevel = 5;
12
13 // Number of events to be processed (default is 10)
14 ApplicationMgr.EvtMax = 50;   事例数，-1表示全部事例
15
                                          输出文件的路径
16 ApplicationMgr.HistogramPersistency = "ROOT";
17 NTupleSvc.Output = { "FILE1 DATAFILE='rhopi_ana.root' OPT='NEW' TYP='ROOT'"};
```

# 分析



$RHOPIALGROOT/share/jobOptions_Rhopi.txt

```
1  #include "$VERTEXFITROOT/share/jobOptions_VertexDbSvc.txt"
2  ApplicationMgr.DLLs += {"RhopiAlg"};   算法包的名称
3  ApplicationMgr.TopAlg += { "Rhopi" };  算法程序.cxx的名称
4
5  Rhopi.Vr0cut = 1.0;
6  Rhopi.Vz0cut = 5.0;
7
8  Rhopi.EnergyThreshold = 0.04;
9  Rhopi.GammaPhiCut = 20.0;
10 Rhopi.GammaThetaCut = 20.0;          “开关”
11 Rhopi.GammaAngleCut = 20.0;
12
13 Rhopi.Test4C = 1;
14 Rhopi.Test5C = 1;
15 Rhopi.CheckDedx = 1;
16 Rhopi.CheckTof = 1;
```

/cvmfs/bes3.ihep.ac.cn/bes3sw/Boss/7.0.8/Analysis/Physics/RhopiAlg/RhopiAlg-00-00-23/src/Rhopi.cxx

注：如果在科大服务器上运行作业（包括模拟重建分析），需添加以下这句话：

DatabaseSvc.Host="10.1.2.12";

# 成功标志



看输出日志的最后两行

```
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
```

```
================================================================================================
           MBrA:    Detailed statistics for all branches
================================================================================================
  KF     AveWt     ERela     WtSup      Wt<0   Wt>Wmax       Ntot       Nacc   Nneg   Nove   Nzer
   4  0.031421  0.090038 2.030      0.000000  0.019550       1696         51      0      3    106
All:  0.031421  0.090038 2.030      0.000000  0.019550       1696         51      0      3    106
================================================================================================
BesSim::finalize(), total events in this run: 50
BesDetectorConstruction::~BesDetectorConstruction()
G4 kernel has come to Quit state.
EventSelector       ERROR ............. releaseContext Not Implemented .............
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
```

模拟

```
total event number is : 50
total track number is : 99     RecMdcTrack number is : 4     RecMdcKalTrack number is :95
Total event:50
PrimaryVertex     SUCCESS ==================================
PrimaryVertex     SUCCESS survived event :50 3 2 1 0 0 0 0 0 0
PrimaryVertex     SUCCESS ==================================
HltEventMaker     SUCCESS 0 events are converted.
DstHltMaker       SUCCESS 50 events are converted.
ApplicationMgr       INFO Application Manager Finalized successfully
ApplicationMgr       INFO Application Manager Terminated successfully
```

重建

# 利用集群跑作业



• 作业提交 hep_sub jobscript
jobscript: 作业脚本名，可以是绝对路径文件名也可以是相对路径文件名

• 作业状态查询 hep_q –u
-u: 指定查看某用户的作业，默认为当前用户。
例如: hep_q –u 可以查看自己的作业

• 作业删除 hep_rm jobs
jobs: 指定要删除的作业id，支持指定多个作业id同时删除。
例如: hep_rm 12345 12345.6
hep_rm –a: 删除当前用户所有作业

• 查看作业时长限制:
hep_clus –g bes --walltime

如果提交标准的boss作业，可使用更简化的boss.condor命令
例：boss.condor joboptions.txt

参考：HTCondor作业

| 实验 | 短作业(short)时长限制(小时) | 普通作业时长限制(小时) | mid作业时长限制(小时)及资源使用量限制(百分比) |
|---|---|---|---|
| BES | <0.5 | <40 | <100:10% |

# 环境设置



HepJob涉及的所有命令都在以下目录，建议
将该目录加入用户环境变量 PATH 中：

bash 用户

```
$ export PATH=/afs/ihep.ac.cn/soft/common/sysgroup/hep_job/bin:$PATH
```

tcsh 用户

```
$ setenv PATH /afs/ihep.ac.cn/soft/common/sysgroup/hep_job/bin:$PATH
```

在科大服务器需要用这句命令配置hepjob的环境
source /cvmfs/common.ihep.ac.cn/software/hepjob/setup_hepjob.csh ustc

# 批量提交作业



为了充分利用计算集群的资源，节约工作时间，实际工作中我们往往需要提交大量作业。而我们不可能一个个手动提交，因此可以借助脚本实现自动批量提交。

Shell作为Linux系统的命令解释器，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。不仅如此，Shell有自己的编程语言用于对命令的编辑，可以编写脚本处理高复杂度工作。

考虑到我们只需要进行轻量化的工作，且基本所有Linux系统自带Shell，因此后续我们将介绍如何用Shell脚本（这里是bash）批量提交作业。

后面的脚本文件是我自己用过的，仅供参考，大家需要根据自己的需求进行适当的修改

# 模拟作业脚本



```
//DENG Zi-yan 2008-03-17

#include "$OFFLINEEVENTLOOPMGRROOT/share/OfflineEventLoopMgr_Option.txt"

//*************job options for generator (KKMC)**************
#include "$KKMCROOT/share/jobOptions_KKMC.txt"
KKMC.CMSEnergy = 3.097;
KKMC.BeamEnergySpread=0.0008;
KKMC.NumberOfEventPrinted=1;
KKMC.GenerateJPsi=true;

//*************job options for EvtGen**************
#include "$BESEVTGENROOT/share/BesEvtGen.txt"
EvtDecay.userDecayTableName = "/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/sim/xi.dec";


//*************job options for random number***********************

//*************job options for detector simulation*****************
#include "$BESSIMROOT/share/G4Svc_BesSim.txt"

//configure for calibration constants
#include "$CALIBSVCROOT/share/calibConfig_sim.txt"

// run ID
//RealizationSvc.RunIdList = {-9947,0,-10878,-27255,0,-28236};
RealizationSvc.RunIdList = {-52940,0,-54976,-55861,0,-56546,-56788,0,-59015};

#include "$ROOTIOROOT/share/jobOptions_Digi2Root.txt"


// OUTPUT PRINTOUT LEVEL
// Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
MessageSvc.OutputLevel  = 6;

// Number of events to be processed (default is 10)
ApplicationMgr.EvtMax = 5000;
DatabaseSvc.Host="10.1.2.12";
```

/ustcfs/BES3User/undergraduate/rsun/introduction/sim.head

与前面模拟部分的文件基本一致，一些需要修改的量放在后面的脚本中

# 模拟作业脚本



/ustcfs/BES3User/undergraduate/rsun/introduction/sim.sh

```
#!/bin/bash   规定使用哪种shell
set +x
cd /ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/sim/round19
wpath=/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/sim
cp -r $wpath/run.head ./
let i=0
let j=0
for ((j = 0; j < 30; j++)); do
  mkdir ${j} && cd ${j}
  for ((i = 0; i < 20; i++)); do
    let index=1000+i+j*20
    cat ../run.head >run_${index}.txt     将run.head写入新的文件
    echo "BesRndmGenSvc.RndmSeed = ${index};" >>run_${index}.txt   随机数种子
    echo "RootCnvSvc.digiRootOutputFile = \"sigMC_Xi_${index}.rtraw\";" >>run_${index}.txt
    boss.condor run_${index}.txt
  done
  cd ../
done
```

注：chmod +x “jobscript” 赋予文件可执行权限

# 重建作业脚本



```
//input ROOT MC data
#include "$ROOTIOROOT/share/jobOptions_ReadRoot.txt"
#include "$OFFLINEEVENTLOOPMGRROOT/share/OfflineEventLoopMgr_Option.txt"

// background mixing
#include "$BESEVENTMIXERROOT/share/jobOptions_EventMixer_rec.txt"
MixerAlg.ReplaceDataPath="/ustcfs/bes3data/randomtrg/";

#include "$CALIBSVCROOT/share/job-CalibData.txt"
#include "$MAGNETICFIELDROOT/share/MagneticField.txt"
#include "$ESTIMEALGROOT/share/job_EsTimeAlg.txt"

// PAT+TSF+HOUGH method for MDC reconstruction
#include "$MDCHOUGHFINDERROOT/share/jobOptions_MdcPatTsfHoughRec.txt"

#include "$KALFITALGROOT/share/job_kalfit_numf_data.txt"
#include "$MDCDEDXALGROOT/share/job_dedx_all.txt"
#include "$TRKEXTALGROOT/share/TrkExtAlgOption.txt"
#include "$TOFRECROOT/share/jobOptions_TofRec.txt"
#include "$TOFENERGYRECROOT/share/TofEnergyRecOptions_MC.txt"
#include "$EMCRECROOT/share/EmcRecOptions.txt"
#include "$MUCRECALGROOT/share/jobOptions_MucRec.txt"

#include "$EVENTASSEMBLYROOT/share/EventAssembly.txt"
#include "$PRIMARYVERTEXALGROOT/share/jobOptions_kalman.txt"
#include "$VEEVERTEXALGROOT/share/jobOptions_veeVertex.txt"

#include "$HLTMAKERALGROOT/share/jobOptions_HltMakerAlg.txt"
#include "$EVENTNAVIGATORROOT/share/EventNavigator.txt"

//output ROOT REC data
#include "$ROOTIOROOT/share/jobOptions_Dst2Root.txt"

//configure of calibration constants for MC
#include "$CALIBSVCROOT/share/calibConfig_rec_mc.txt"
ApplicationMgr.EvtMax = -1;

//*************job options for random number**********************

//Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
MessageSvc.OutputLevel = 6;
DatabaseSvc.Host="10.1.2.12";
```

/ustcfs/BES3User/undergraduate/rsun/introduction/rec.head

事例数-1，与对应模拟文件的事例数一致

# 重建作业脚本



/ustcfs/BES3User/undergraduate/rsun/introduction/rec.sh

```
#!/bin/bash
set +x
round=round19
cd /ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/rec/${round}
wpath=/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/rec
sim=/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/sim
cp -r $wpath/run.head ./
let i=0
let j=0
let num=$(find ${sim}/${round} -name "*.rtraw" | wc -l)
let line=num/20
for ((j = 0; j < $line; j++)); do
  if [ -d "${sim}/${round}/${j}" ]; then
    mkdir ${j} && cd ${j}
    for ((i = 0; i < 20; i++)); do
      let index=1000+i+20*j
      if [ -f "${sim}/${round}/${j}/sigMC_Xi_${index}.rtraw" ]; then
        cat ../run.head >run_${index}.txt
        echo "BesRndmGenSvc.RndmSeed = ${index};" >>run_${index}.txt
        echo "EventCnvSvc.digiRootInputFile = {\"${sim}/${round}/${j}/sigMC_Xi_${index}.rtraw\"};" >>run_${index}.txt
        echo "EventCnvSvc.digiRootOutputFile = \"rec_${index}.dst\";" >>run_${index}.txt
        boss.condor run_${index}.txt
      fi
    done
    cd ../
  fi
done
```

# 分析作业脚本



/ustcfs/BES3User/undergraduate/rsun/introduction/ana.head

```
#include "$ROOTIOROOT/share/jobOptions_ReadRec.txt"
#include "$MAGNETICFIELDROOT/share/MagneticField.txt"
#include "$ABSCORROOT/share/jobOptions_AbsCor.txt"
#include "/home/rsun/workarea/7.0.8/Analysis/Physics/XiWork/XiWork-00-00-03/share/jobOptions_Xi.txt"

DatabaseSvc.Host="10.1.2.12";
MessageSvc.OutputLevel = 6;
MessageSvc.useColors = false;

ApplicationMgr.EvtMax = -1;
ApplicationMgr.DLLS += {"RootHistCnv"};
ApplicationMgr.HistogramPersistency = "ROOT";
```

```
#include "$VERTEXFITROOT/share/jobOptions_VertexDbSvc.txt"
#include "$EVENTWRITERROOT/share/jobOptions_EventWriter.txt"
ApplicationMgr.DLLs += {"XiWork"};
ApplicationMgr.TopAlg += { "Xi" };

WriteMyDst.ItemList = {
  "/Event/EventHeader#1",
  "/Event/Dst/DstMdcTrackCol#1",
  "/Event/Dst/DstMdcKalTrackCol#1",
  "/Event/Dst/DstMdcDedxCol#1",
  "/Event/Dst/DstTofTrackCol#1",
  "/Event/Dst/DstEmcShowerCol#1",
  "/Event/Dst/DstMucTrackCol#1",
  "/Event/Dst/DstExtTrackCol#1",
  "/Event/EvtRec/EvtRecEvent#1",
  "/Event/EvtRec/EvtRecTrackCol#1",
  "/Event/EvtRec/EvtRecPrimaryVertex#1",
  "/Event/EvtRec/EvtRecVeeVertexCol#1",
  "/Event/Trig/TrigData#1",
  "/Event/Hlt/DstHltInf#1",
  "/Event/MC/McParticleCol"
};
```

jobOptions_Xi.txt

# 分析作业脚本



```
#!/bin/bash
set +x
cd /ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/ana
round="roundtruth"
todolist="todolist12.txt"
workplace=/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/ana/${round}
todo=/ustcfs/BES3User/undergraduate/rsun/sigMC_Xi/ana/${todolist}

if [[ ! -d ${round} ]]; then
    mkdir ${round}
fi

if [[ $(ls -A ./${round} | wc -w) -ne 0 ]]; then
    rm -rf ${round}
    mkdir ${round}
fi
cd ./${round}

cp -r ../run.head ./
cp -r ${todo} ./

sed 's/^/"&/g' ${todo} >./tmp.txt
sed 's/$/&"/g' tmp.txt >./${todolist}
rm -rf tmp.txt

let dirnum=100
let num=1
let include=20
let i=0
let j=0
let n=$(cat ./${todolist} | wc -l)
let line=n/num
let nn=n%num
```

/ustcfs/BES3User/undergraduate/rsun/introduction/ana.sh

# 分析作业脚本



```
for ((i = 0; i <= line; i++)); do
    let index=1000+i
    cat ./run.head >run_${index}.txt
    echo -e "\nEventCnvSvc.digiRootInputFile = {" >>run_${index}.txt
    for ((j = 1; j <= num; j++)); do
        let tmp=i*num+j
        sed -n "${tmp},${tmp}p" ./${todolist} >>run_${index}.txt
        if [[ i -eq line && j -eq nn ]]; then
            echo "};" >>run_${index}.txt
            echo "NTupleSvc.Output = {\"FILE1 DATAFILE = '${workplace}/${dirnum:1:2}/ana_${index}.root' OPT='NEW' TYPE='ROOT'\"};" >>run_${index}.txt
            mkdir ${dirnum:1:2}
            mv run_*.txt ./${dirnum:1:2}
            let dirnum=dirnum+1
            break 2
        fi
        if [[ j -lt ${num} ]]; then
            echo "," >>run_${index}.txt
        else
            echo "};" >>run_${index}.txt
        fi
    done
    echo "NTupleSvc.Output = {\"FILE1 DATAFILE = '${workplace}/${dirnum:1:2}/ana_${index}.root' OPT='NEW' TYPE='ROOT'\"};" >>run_${index}.txt
    let ndir=$i+1
    if [[ tmp -eq n ]]; then
        mkdir ${dirnum:1:2}
        mv run_*.txt ./${dirnum:1:2}
        break
    fi
    if [[ ndir%${include} -eq 0 ]]; then
        mkdir ${dirnum:1:2}
        mv run_*.txt ./${dirnum:1:2}
        let dirnum=dirnum+1
```

```
find ./ -name "run_*.txt" >tmp.txt
sort tmp.txt >boss.txt
rm -rf tmp.txt
let nrun=$(cat boss.txt | wc -l)
for ((i = 1; i <= nrun; i++)); do
    boss.condor $(sed -n "${i},${i}p" boss.txt)
done
```