

# A Proposal for More-easily Producing Samples from Parton-Level Generators

Soushi Tsuno

Simulation Meeting on Nov.4.2004

# Introduction

Last time, I talked about the interface directory to pass through the **MCProd** from ME generators. (See web talk in the last simulation meeting(Oct.21).)

Using this module (**LesHouchesModule**), we can expect

- 1) minimum effort to make production data,
- 2) easy to check all parameters in PROD data,
  - ➡ Utility to use same PS / decay model at least.
  - ➡ All parameters controlled one tcl files.
- 3) minimum data size.
  - ➡ We do not need to save HEPG separately.
  - ➡ Maintenance will be simpler.

I can provide the procedure how to directory make PROD data from “pev file”

But one question was rising up.

Q) How do we manage multiple run number??

That makes things complicated.

## Multi run number

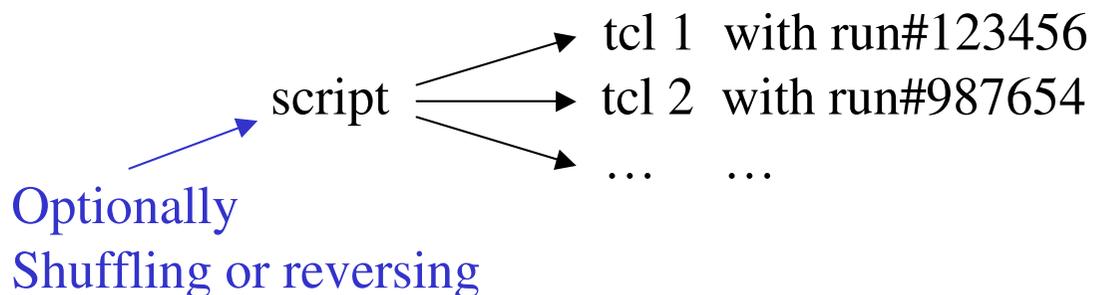
There are two different procedures.

- 1) Official MC Production runs on the farm (DCF).
- 2) End-user uses CAF system.

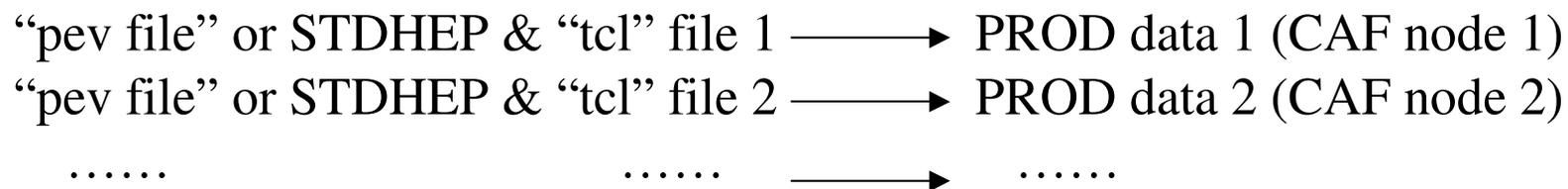
# CAF Production

Now, the **LesHouchesModule** is a part of **MCPProd**.

Thus, a certain “script” may create the “tcl” files according to run numbers.



Then, throw them into **CAF jobs**.

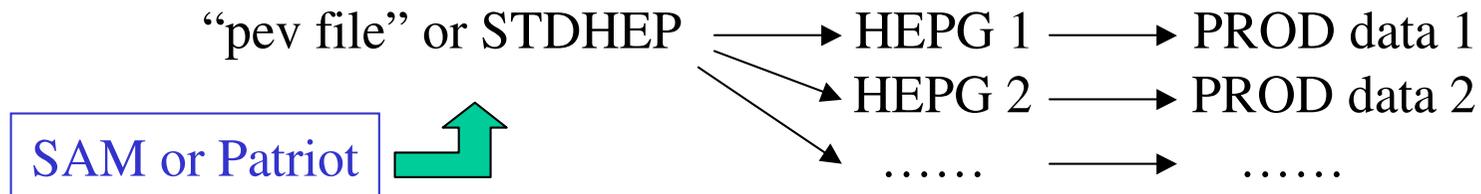


 No need to save HEPG separately.  
The “tcl” files contain all parameters settings.

**This is almost ready. Everyone can use it in 5.3.3.**

# Farm Production

Currently, it is difficult to change run numbers in one input file.  
So that we need to make HEPG files with the different run numbers.



At this point, the file management is bothering work even if it handles automatically. (also lots of log files...)

We have

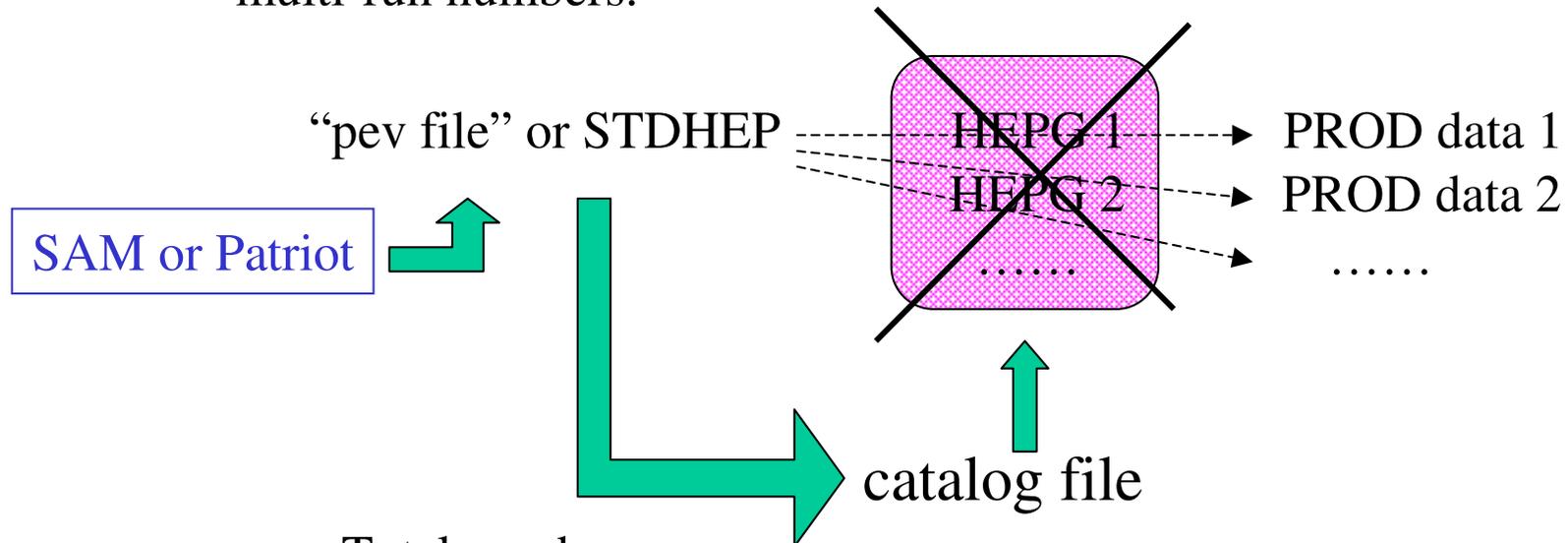
- 1) slow speed,
- 2) large disk space,
- 3) lots of log files.

Strange logic.

Why does HEPG has run numbers?  
HEPG only contains physics events.  
(should be independent from run#.)  
Run number should be assigned at  
the detector simulation.

# Proposal on Farm Production

Instead of making HEPG files, prepare the catalog file containing multi-run numbers.



Total number  
of events

If we know the total number of events which we want, the run number is simply assigned.

We can expect

- 1) speed up,
- 2) small disk space,
- 3) doubly reduced log files.

Smart logic

“pev” or STDHEP is physics events without run number.

# Test it for future MC production

**MadGraph** or whatever

**GRAPPA** samples:

Each 100 k events.

W + 0,1,2,3,4 jets

Z + 0,1,2,3,4 jets

WW + 0,1,2 jets

ZZ + 0,1,2 jets

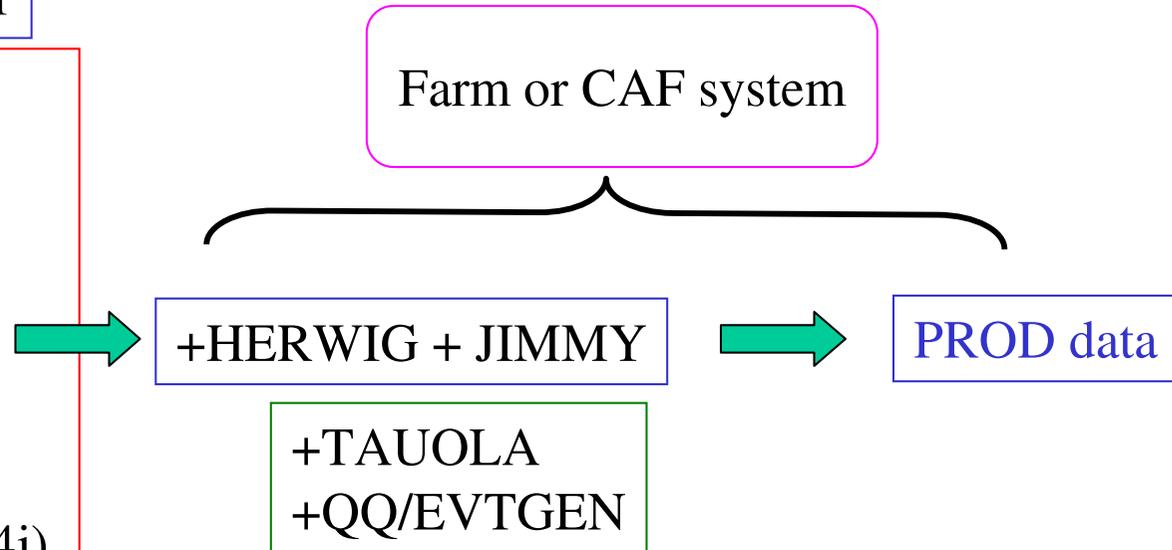
ZW + 0,1,2 jets

t t (6-body) (1+4j)

t t + jet (7-body) (1+5j)

... whatever you need.

Bosons decay into e,  $\mu$ ,  $\tau$ .



- 1) The test must be needed for future standard.
- 2) **JIMMY** (HERWIG multiple interaction model) can be tested at the same time.
- 3) **GRAPPA** only treats “real” width correction and 3x3 CKM mixings at CDF. (More reliable tool to estimate b-fraction in Method II.)

# Summary

LesHouchesModule makes MCProd simpler and easier.

For MC production with multi run numbers,

- 1) In CAF production, it's ready to use.
- 2) In farm production, it requires some modifications if we do not make HEPG.

Time scale depends on (2).

We need to test it.

# LesHouches Package

Now, HERWIG.6.5 is ready to use LHA format.  
we extend “pythia\_i/pythia\_service” to HERWIG.

## LesHouches Package

We do not need “pythia\_i/pythia\_service” anymore.

Same as before, LesHouchesModule only works for the interface.  
LesHouchesModule requests :

1) (unweighted) event file from the ME-generator. **That's it!!**

We can select PYTHIA or HERWIG via tcl file of LesHoucesMoule.

If it does not work, check the two core subroutine, **UPINIT/UPEVNT**  
between CDF code and your code. (if else, I will describe later.)

# What we expect

- 1) **Coding will be simple.** Because LesHouches does not need to run the ME-generators inside CDF-offline. Users just need to prepare the event file which was produced outside CDF-offline code.

➡ easy to debug , guarantee the right operation of ME-generators.

- 2) **MC production will be much smarter.** We do not need HEPG files (and STDHEP files) before making Production data. LesHouches just allows us to start the (unweighted) event file from ME-generators.

➡ the storing file size will be minimum.

- 3) **Easy for a systematic study of MC.** At least, ME has no systematics. So that all parameters are controlled by the tcl files.

➡ In our current scheme (to start from HEPG/STDHEP), we can not trace what parameters are used in showering MCs.

➡ On the other hand, in LesHouchesModule, just look at tcfs, and if we want to change the parameters for the systematic study, just need to change the tcfs.

# Parameters in LesHouchesModule

- LesGenType** : Select ME-generators of your input event file.  
Alpgen/CompHep/MadGraph/Grappa/MCFM/USER
- LesDatFile** : Name of your input event file.
- LesPSModel** : Set showering-MC of PYTHIA or HERWIG.
- firstEventToRead** : The starting event number to store in HEPG.
- putResonance** : Set resonance particles (W or Z) in the event history.
- scaleOfKtClus** : Set energy scale for the parton shower based on kt-clus algo..
- useTAUOLA** : Use or not. The helicity info. from ME is connected correctly.
- useStdHep** : Read from STDHEP files or write the STDHEP file.
- StdhepFileName** : The name of STDHEP file to read or write.
- NumOfStdEvent** : The event number to read or write in STDHEP file.
- PythiaMenu** : Menu of PYTHIA. This is exactly same as CDF-PYTHIA.
- HerwigMenu** : Menu of HERWIG. This is exactly same as CDF-HERWIG.
- (useJIMMY)** : (Use JIMMY underlying event model for HERWIG.)

# How to include your generator (case A)

In case if your event file does not work in LesHouchesModule,

1) Check the event file format. Type,

```
~% diff [your generator]/UPINIT LesHouches/src/[your generator]_UPINIT
~% diff [your generator]/UPEVNT LesHouches/src/[your generator]_UPEVNT
```

If you see a lot of lines, your event format is not suitable in LesHouchesModule.

**But!**

To use it, replace your **UPINIT/UPEVNT** (and the other related subroutines) into LesHouches/src/[your generator]\_les.F as the subroutine name of [your generator]\_UPINIT/UPEVNT.

**It should work.**

## How to include your generator (case B)

In case if your generator does not exist in LesHouchesModule,

- 1) Copy **UPINIT** and **UPEVNT** in your generator into LesHouchesModule/src/**user\_les.F** with the name of **USER\_UPINIT** and **USER\_UPEVNT**.
- 2) Run it with the “**LesGenType set USER**” in tcl file.

More specified details is in <http://nkek15.fnal.gov/~tsuno/leshouches.html>