Running MadGraph Guide
For the Texas A&M Phenomenology Computing

by:
Ziqing Hong
zqhong@tamu.edu
Jorge Morales
georgemm01@physics.tamu.edu

July, 2014

This guide is intended to be used with the Texas A&M Brazos Cluster, but the information here can easily be generalized for use with any system.

Introduction

This guide will show how to run Mad Graph; this platform is self-sufficient, meaning it doesn't really need to be installed, but rather the files are generated from the website and untarred. We show the appropriate steps necessary to efficiently submit to a cluster: 1) make a tarball for cluster submission 2) make a PBS script that submits multiple single-core jobs of 100,000 events with different random seeds, as suggested by MadGraph authors.

Getting the MadGraph Code

We are only interested in generator-level information. No Pythia-PGS is run in this procedure. Generally, one can generate a code package from:
http://madgraph.hep.uiuc.edu/new_gen_proc_card.html

Choose the process, and submit your specifications, until you see: "Now a process is generated". However, our example process: TopBSM is a bit more complicated, it isn't implemented in the web interface. Start from downloading the whole MG5 package

The first several steps can be done on a local computer, or on hurr (might be slower).

1. Go to http://madgraph.hep.uiuc.edu/index.html , register and download the version you want.
2. Copy the files over to your home directory (or wherever you want to set this up).

```
[zqhong@hurr:~/MadGraph]$ scp MG5_aMC_v2.1.1.tar.gz username@hurr.tamu.edu:~/.
```

3. Log in to hurr (with an xterm option enabled)

```
[zqhong@hurr:~/MadGraph]$ ssh -Y username@hurr.tamu.edu
```

, make the directory you want to work in, and untar the files

```
[zqhong@hurr:~/MadGraph]$ tar -xvf MG5_aMC_v2.1.1.tar.gz
```

4. Go to the directory, and list the contents, you should see:

```
[zqhong@hurr:~/MadGraph]$ cd MG5_aMC_v2_1_1/
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ ls
aloha  doc.tgz  INSTALL   MadSpin   proc_card.dat  tests       VERSION
bin    HELAS    LICENSE   mg5decay  README         UpdateNotes.txt
doc    input    madgraph  models    Template       vendor
```

5. Go to models, and copy the UFO files over (note, you might not need UFO models). I got the model from Adam Falkowski, together with **run_card.dat** and **param_card.dat** in the same directory. The UFO models can be created with FeynRules https://feynrules.irmp.ucl.ac.be/

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ cd models

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/models]$ cp ~/MadGraph/store/Axigluon_Com_UFO.zip .

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/models]$ unzip Axigluon_Com_UFO.zip

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/models]$ cd ..
```

6. Load the python module on Brazos (this is maintained by the Brazos Administrators, sometimes the most recent version is not the one recommended for MadGraph, check http://brazos.tamu.edu/software/modules.html for the available modules on the Cluster.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ module load python
Python 2.7.3 (GCC, 64-bit)
Setup environment for Python 2.7.3 (64-bit)
Environment variables set:
 PATH LD_LIBRARY_PATH MANPATH PYTHONHOME PYTHONPATH
----------------------------------------------------------
```

7. Change run mode to single machine, this will make your jobs more efficient since you will submit several single-core jobs. Keep it as 2 if you're running on your own machine and want to use as many cores as possible, but cannot 'access' as many as in the cluster. Edit the file, and look fot the line in which you set the run mode.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ vi input/mg5_configuration.txt

 run_mode = 0
```

Now you need to make the environment to generate the events. Note that here you see an example for importing non-common models, you may also just use Standard Model generation (most basic form in MadGraph), but you should take the appropriate precautions for your own project (most models are available downloading the packages). Once you have this output you can generate them not only interactively but also submitting to the Cluster.

Obtaining the MadGraph Tarball for Massive-Running

8. Run / Open MadGraph

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ bin/mg5_aMC
```

9. If needed import the new UFO model (or similar).

```
MG5_aMC>  import model Axigluon_Com_UFO
```

10. Generate/Execute a simulated process, by defining the decay you want to study, for example in this case we want pp → tt~ → dilepton. Later you can also specify initial particles (pp~), energy and PDF in run_card.dat

```
MG5_aMC> generate p p > t t~ QED=0, ( t > w+ b , w+ > l+ vl ), (t~ > w- b~, w- > l- vl~)
MG5_aMC> output
MG5_aMC> quit
```

11. Now, a process has been generated. Go to the directory usually named **PROC_sm_0** or similar, in our example: **PROC_Axigluon_Com_UFO_0** .

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ ls
aloha    HELAS   madgraph  PROC_Axigluon_Com_UFO_0  Template       VERSION
bin      input   MadSpin   proc_card.dat            tests
doc      INSTALL  mg5decay  py.py                   UpdateNotes.txt
doc.tgz  LICENSE  models   README                   vendor

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1]$ cd PROC_Axigluon_Com_UFO_0

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ ls
bin     HTML        madevent.tar.gz  README.gridpack     Source
Cards   index.html  MGMEVersion.txt  README.lhapdf       SubProcesses
Events  lib         README           README.systematics  TemplateVersion.txt
```

12. Information regarding the process generated can be viewed in index.html

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ w3m index.html
```

13. Copy over run_card.dat and param_card.dat to the process directory, for some 'standard' processes it might already be there, but make sure it is.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ cp
~/MadGraph/store/run_card.dat Cards/.

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ cp
~/MadGraph/store/param_card.dat Cards/.
```

You will need to modify the 'cards' appropriately. This will make the detailed configurations about the physics of the process and other details of your jobs. The most common files you might be interested on are: **param_card.dat   proc_card_mgb.dat   run_card.dat** .

14. Edit run_card.dat

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ vi Cards/run_card.dat
```

14a. We want to run MadGraph on cluster, so change gridpac to true. Switch it to false in case you want to run it interactively. Keep nevets less than 100,000 as suggested by MadGraph authors, if you need lot's of events run multiple times with different seeds, more details later. Tweak the following lines:

```
 .true.     = gridpack  !True = setting up the grid pack
 100000 = nevents ! Number of unweighted events requested
```

15. Tune the parameters of the new model in param_card.dat . This is all about the physics specifications that you want.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ vi Cards/param_card.dat
```

16. Again, change run mode to single machine for running on cluster, this time in the configuration file me5_configuration.txt

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ vi Cards/me5_configuration.txt
```

```
 run_mode = 0
```

17. Generate the gridpack tarball (-f means no question asked).

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ bin/generate_events -f
```

* In the **PROC_sm_0** directory you find the same: **/bin/generate_events -f**

NOTE: This could take a while on hurr. If it takes too long, your process might be killed, if this happens, you can

a) Run everything above on your own machine and copy the output tarball to hurr
b) Request an 'interactive session' by:

```
qsub -q general -l nodes=1:ppn=8 -l walltime=01:00:00 -I -X
```

and then doing this step.

c) Sumbit a PBS job doing this step.

In the end, you'll get a tarball.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ ls
bin         index.html       README.gridpack        SubProcesses
Cards       lib              README.lhapdf          TemplateVersion.txt
crossx.html  madevent.tar.gz  README.systematics
Events      MGMEVersion.txt  run_01_gridpack.tar.gz
HTML        README           Source
```

The cross-section is calculated and stored in crossx.html

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ w3m crossx.html
```

18. The tarball (run_01_gridpack.tar.gz) is self-contained. That's the only thing we need for massive event generation. To prevent messing up with the whole MadGraph package, move it to somewhere else.

```
[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ mkdir
```

```
~/MadGraph/PROC

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ mv
run_01_gridpack.tar.gz ~/MadGraph/PROC/.

[zqhong@hurr:~/MadGraph/MG5_aMC_v2_1_1/PROC_Axigluon_Com_UFO_0]$ cd ~/MadGraph/PROC/

[zqhong@hurr:~/MadGraph/PROC]$ ls
run_01_gridpack.tar.gz
```

* If you're generating the gridpack tarball on your own machine, copy it over to somewhere on hurr and proceed from here.

19. Untar the tarball

```
[zqhong@hurr:~/MadGraph/PROC]$ tar -xzvf run_01_gridpack.tar.gz
[zqhong@hurr:~/MadGraph/PROC]$ ls
madevent  README.gridpack  run_01_gridpack.tar.gz  run.sh
```

20. Go to madevent and (re)compile, clean and re-make the tarball

```
[zqhong@hurr:~/MadGraph/PROC]$ cd madevent/

[zqhong@hurr:~/MadGraph/PROC/madevent]$ bin/compile

[zqhong@hurr:~/MadGraph/PROC/madevent]$ bin/clean4grid

[zqhong@hurr:~/MadGraph/PROC/madevent]$ cd ..

[zqhong@hurr:~/MadGraph/PROC]$ rm run_01_gridpack.tar.gz

[zqhong@hurr:~/MadGraph/PROC]$ tar -czvf !$ *
```

Scripting to run in the Cluster

Now run_01_gridpack.tar.gz is ready for massive event production!

21. To run pbs jobs, create a .pbs file. A sample file can be found as below. First go to a directory for your pbs script, and make the pbs config file in it.

```
[zqhong@hurr:~/MadGraph/PROC]$ cd ~/pbs
[zqhong@hurr:~/pbs]$ vi madevent.pbs

#PBS -l nodes=1:ppn=1
#PBS -l walltime=3:00:00
#PBS -q hepxrt
#PBS -t 1-10

. /etc/profile.d/modules.sh
```

```
module load python

mkdir /tmp/$PBS_JOBID
chmod 700 /tmp/$PBS_JOBID

cd /tmp/$PBS_JOBID

cp ~/MadGraph/PROC/run_01_gridpack.tar.gz .

tar -xzvf run_01_gridpack.tar.gz

rm run_01_gridpack.tar.gz

./run.sh 100000 $PBS_ARRAYID

ls

mv events.lhe.gz events_$PBS_JOBID.lhe.gz

cp events_$PBS_JOBID.lhe.gz /fdata/hepx/store/user/zqhong/MG_output/.

#tar -czvf run_01_gridpack_$PBS_JOBID.tar.gz *

#mv run_01_gridpack_$PBS_JOBID.tar.gz /fdata/hepx/store/user/zqhong/MG_output

rm -r /tmp/$PBS_JOBID

exit 0
```

* This pbs file specifies 10 sub-jobs (with -t 1-10), and uses the sub-job number ($PBS_ARRAYID) as random seed.
* Each sub-job generates 100k events, as specified in ./run.sh 100000 $PBS_ARRAYID
* The job uses local harddisk mounted on /tmp, and automatically clean up the area afterwards.
* The output events.lhe.gz is renamed with $PBS_JOBID attached to the filename, so that multiple output files do not overwrite each other.
* The output files are moved to a specified location (/fdata/hepx/store/user/zqhong/MG_output/. in this case). Get the directory prepared and make sure you have write permission there.

22. Finally! Submit the pbs job with qsub

```
[zqhong@hurr:~/pbs]$ qsub madevent.pbs
3558517.brazos.tamu.edu
```

* The output files will show up in the area you specified. They're zipped lhe files.
* Run gunzip on them, and use your favorite lhe file processor to proceed.