



Gbasf2 usage: running on the grid

Mario Merola, Benjamin Oberhof, Silvio Pardi, Antonio Budano

7th Belle II Italian meeting 04/05/17



Gbasf2: getting started



 Check you have the certificates and proper settings to run on the grid:

https://confluence.desy.de/pages/viewpage.action?spaceKey=Bl&title=Computing+GettingStarted

Gbasf2 tutorial at the last B2GM as starting point:

https://kds.kek.jp/indico/event/23336/session/54/?slotId=0#20170208

 Reminder: from MC8 on, data will be accessible only via grid (small amount of BB MC will be transferred to KEK)



Gbasf2: tests performed



Simple analysis job on MC8 with release-00-08-00

Production jobs with custom arguments

More complex analysis using Full Event Interpretation

 Run with user-defined modules / variables (not in the central release)



Run simple analysis example on MC8



- Reconstruction of B0 → J/psi Ks (J/psi → mu mu and Ks → pi pi)
- Perform kinematic cuts and save the info into ntuples
- Usage (example taken from tutorial adapted to MC8):

```
gbasf2 ./ana_jmumuks.py -p analysis0 -i / belle/MC/release-00-08-00/DB00000208/MC8/prod00000962/s00/e0000/4S/r00000/mixed/sub00
```

• gbasf2 creates n jobs, one for each mdst file, like:

```
basf2 -i mdst_000044_prod00000223_task00000043.root ana_jmumuks-43.py
```

Download jobs output with:

```
gb2_job_output -p analysis0
gb2_ds_get /belle/user/merola/analysis0
```

Everything works smoothly



Run MC production jobs with custom arguments (1)



5

- It might be useful for tests before the official production starts, or if need signal samples with different BG configurations and/or with old releases
- In my case, for example, I needed to produce B \rightarrow tau nu without beam background (with an old release)
- Have to specify by hand the number of events and the seed for generation, through the option "--basf2opt". Also, I have to submit one job for each generation, with a command like:





Run MC production jobs with custom arguments (2)



- In this case the jobs submission and splitting is not handled automatically as for the first example, but it works.
- N.B. by default gbasf2 does not accept custom input arguments of the script you want to run, but only basf2 standard args (-n, -i, -l, etc.). If you want to use custom arguments (for example the seed) you need to modify the basf2helper module.



Test using FEI (1)



Example: B → tau nu analysis on MC8

```
gbasf2 ./BtagBsig.py -f basf2_final_path_without_selection.pickle
--basf2opt=" -l ERROR -n 10000 " -p merola_taunufei_charged_1 -n 50
-i /belle/MC/release-00-08-
00/DB00000208/MC8/prod00000964/s00/e0000/4S/r00000/charged/sub02
```

- The arguments are: supplemental files, basf2 options, project name, number of files per job, path of the input files
- The script needs the FEI training weights (~100MB), which can be linked with -f command as well... but the limit file size is 10MB (because they are temporary copied to the dirac server and then to the worker node of the site where the jobs will run)



Test using FEI (2)



Ω

- Copy the file to a specific SE (for example with gb2_ds_generate command) bypassing the dirac server.
- The FEI weights can be uploaded to the condition database (as the FEI developer, Thomas Keck, did with the old trainings):

```
conditionsdb tag create test_merola

conditionsdb upload test_merola /home/belle2/tkeck/fei/
Belle2_Generic_2017_Track14_1/fei_local_database

and can be accessed via:
use_central_database('test_merola', LogLevel.WARNING, 'fei_database')
```

Good solution in this particular case, but in general?





FEI training on the grid



- In the future the FEI training itself should run over the grid
- The FEI training includes many reconstruction steps, with each one
 using the output of the previous step and storing temporary files for
 the next steps (up to 1-2 TB of data stored).
- We need to test it on the grid as soon as possible



Run with user defined modules/variables



10

Test with new module:

gbasf2 ./test_jpsimumu_grid.py -p oberhof_analysis1 -n 1 -i /belle/MC/release-00-08-00/DB00000208/MC8/prod00001999/s00/e0000/4S/r00000/1191820000/sub00 -f libReconstructMissing.so KmodularAnalysis.py

- libReconstructMissing.so is the shared lib object corresponding to the new module and KmodularAnalysis.py contains the definition of the function using the module
- Test with new variable: defined a new variable in PIDVariables.cc (new Likelihood ratio for PID), then tried to run linking the library with —f libanalysis.so → problems with re-definition of variables





Conclusions



- Use the grid and send feedback
- Some features are missing and need to be developed
- Suggest to subscribe to comp-users-forum@belle2.org