Gbasf2 Tutorial

2022 Belle II Summer Workshop

J. Guilliams

Based on tutorials given by T. Hara, S. Cunliffe, J. Bennett, K. Huang & M. Villanueva

THE UNIVERSITY of MISSISSIPPI

If you need help

- There are Confluence pages with additional information:
  - Gbasf2 mainpage
  - Gbasf2 documentation (still under construction)
  - Instructions for gbasf2 analysis
  - Gbasf2 FAQ and troubleshooting page
  - Computing glossary
- See the previous gbasf2 tutorials
- Please join the comp users forum
  - Ask questions, receive announcements on new release and system issues, etc.
- Ask, and answer, questions at questions.belle2.org

Keep in Mind!

- A local computing system, the grid is **NOT**
  - Once submitted, your jobs are assigned to computing systems around the world
  - If your jobs are bad, all sites to which they are distributed will be affected
  - Therefore, **always test your jobs locally prior to submitting to the grid**
    - As a protection against this, we have scout jobs - submits 10 jobs with a few events per job; if at least 5 scout jobs successfully complete, the remain jobs are released to run
    - If scout jobs fail, check the logs of the scout jobs

Prerequisites

- Do you have all that you need to work on the grid?
  - Everything you need can be found on the Computing GettingStarted Confluence page
- The prerequisites are:
  - A system with SL6 or CentOS 7 (more info on next slide)
  - A valid grid certificate issued within a year and installed in `~/.globus` and on the web browser (more info on next slide)
  - Belle Virtual Organization (VO) membership registered or renewed within a year at the VOMS server
  - Registration in DIRAC

Prerequisites

- System doesn't have SL6?
  - If the system you are using has **Singularity** available, you can work with SL6 by using

```
singularity shell --cleanenv --bind /cvmfs:/cvmfs docker://sl:6
```

- Valid grid certificate issued within a year and installed in `~/.globus` and on the web browser
  - For DIRAC, the certificate must be in PEM format

```
ls -l ~/.globus

total 8
-r--r--r-- 1 justing justing 2011 June 16 10:45 usercert.pem
-r-------- 1 justing justing 1978 June 16 10:47 userkey.pem
```

Make sure that your user key is only readable by you!

If not `-r--------`, **use** `chmod 400 userkey.pem`

```
openssl x509 -in ~/.globus/usercert.pem -noout -subject -dates

subject=DC = org, DC = cilogon, C = US, O = Brookhaven National Laboratory, CN = Justin Guilliams A21194426
notBefore=June 16 15:24:14 2022 GMT
notAfter=Jun 18 03:29:14 2023 GMT
```

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally
     A. If successful, prepare to submit with gbasf2 environment
  II. Locate the input dataset(s) you wish to use on the grid
  III. Submit jobs to the grid
     A. Monitoring your jobs
  IV. Download output to perform offline analysis

Before starting, a brief note

- For the examples/exercises to follow, we will use the **uDST** file format
  - uDST (user Data Summary Table): format type that, by applying certain selection cuts on an input dataset (usually of mDST format), contains a select amount of events useful for a certain type of analysis

I. Develop basf2 steering file; Test it locally

- For this tutorial, we will borrow from a steering script used in the b2starterkit to reconstruct $D^{*+} \to [D^0 \to K^-\pi^+\pi^-\pi^+]\pi^+$
  - If you wish, you can use the steering script located under `~justin/public/tutorial2022` (on KEKCC)

```
$ cat gbasf2Tutorial2022.py
#!/usr/bin/env python3

######################################################################
#
#
# This is a template for the US Belle II Summer School 2022
# It is intended as a starting point for an analysis of
#
# D*+ -> D0 pi+
#        |
#        +-> K- pi+ pi- pi+
#
######################################################################
```

I. Develop basf2 steering file; Test it locally

- For this tutorial, we will borrow from a steering script used in the b2starterkit to reconstruct $D^{*+} \to [D^0 \to K^-\pi^+\pi^-\pi^+]\pi^+$
- Now we test locally to see if the script runs properly
  - We will use the latest light release: `light-2205-abys`
  - Set up the basf2 environment by using

    `source /cvmfs/belle.cern.ch/sl6/tools/b2setup light-2205-abys`
  - To make everyone's life easier, the input/output files have been provided in the script; no need to use `-i` and `-o` flags this time

```
import basf2 as b2
import modularAnalysis as ma
import variables.collections as vc
import variables.utils as vu

# Define the path
mypath = b2.create_path()

infile = '/group/belle2/dataprod/MC/SkimTraining/mixed_BGx1.mdst_000001_prod00009434_task10020000001.root'
output_file = 'Dst2D0pi_D02k3pi.root'
```

I. Develop basf2 steering file; Test it locally

- For this tutorial, we will borrow from a steering script used in the b2starterkit to reconstruct $D^{*+} \to [D^0 \to K^-\pi^+\pi^-\pi^+]\pi^+$
- Now we test locally to see if the script runs properly
  - We will use the latest light release: `light-2205-abys`
  - Set up the basf2 environment by using
  - To make everyone's life easier, the input/output files have been provided in the script; no need to use `-i` and `-o` flags this time
  - Now, all we need to do is execute

    ```
    basf2 ~justin/public/tutorial2022/gbasf2Tutorial2022.py
    ```

I. Develop basf2 steering file; Test it locally

- For this tutorial, we will borrow from a steering script used in the b2starterkit to reconstruct $D^{*+} \to [D^0 \to K^-\pi^+\pi^-\pi^+]\pi^+$
- Now we test locally to see if the script runs properly
- If executed correctly, the job should run successfully

```
[INFO] Writing NTuple dsttree
[WARNING] There were 1418 tracks skipped because of zero charge for K-:all  { module: ParticleLoader_K-:myk }
[WARNING] There were 1286 tracks skipped because of zero charge for pi+:all  { module: ParticleLoader_pi+:mypi }
[INFO] ===Error Summary=============================================================
[WARNING] There were 1286 tracks skipped because of zero charge for pi+:all  { module: ParticleLoader_pi+:mypi }
[WARNING] There were 1418 tracks skipped because of zero charge for K-:all  { module: ParticleLoader_K-:myk }
[INFO] ================================================================================
[WARNING] in total, 2 warnings occurred during processing
=================================================================================================================
Name                                                                |      Calls | Memory(MB) |    Time(s) |     Time(ms)/Call
=================================================================================================================
RootInput                                                           |     200001 |         16 |      59.45 |    0.30 +-   2.58
ProgressBar                                                         |     200000 |          0 |       1.24 |    0.01 +-   0.00
ParticleLoader_pi+:mypi                                             |     200000 |         75 |      28.40 |    0.14 +-   6.89
PListCopy_pi+:mypi                                                  |     200000 |          0 |       3.24 |    0.02 +-   0.00
ParticleSelector_applyCuts_pi+:mypi                                 |     200000 |          0 |       9.97 |    0.05 +-   0.06
ParticleLoader_K-:myk                                               |     200000 |          0 |      19.61 |    0.10 +-   0.06
PListCopy_K-:myk                                                    |     200000 |          0 |       2.84 |    0.01 +-   0.00
ParticleSelector_applyCuts_K-:myk                                   |     200000 |          0 |       8.69 |    0.04 +-   0.01
ParticleCombiner_D0:K3pi -> K-:myk pi+:mypi pi-:mypi pi+:mypi       |     200000 |          0 |      67.86 |    0.34 +-   0.48
ParticleCombiner_D*+:D0pi -> D0:K3pi pi+:mypi                       |     200000 |          0 |      49.66 |    0.25 +-   0.59
MCMatch_D*+:D0pi                                                    |     200000 |          0 |       1.40 |    0.01 +-   0.01
VariablesToNtuple_D*+:D0pi                                          |     200000 |          0 |       4.05 |    0.02 +-   0.14
=================================================================================================================
Total                                                               |     200001 |         91 |     290.63 |    1.45 +-   7.47
=================================================================================================================
```

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment
  II. Locate the input dataset(s) you wish to use on the grid
  III. Submit jobs to the grid
     A. Monitoring your jobs
  IV. Download output to perform offline analysis

Preparing to submit with gbasf2

- Set your gbasf2 environment
  - Two options are available to you to set up your gbasf2 environment
    - CVMFS installation (*recommended*)
    - Local installation

CVMFS installation (recommended)

- Use the preinstalled CVMFS version on KEKCC (or wherever CVMFS is available)

Note

The two following commands **must** be used **each time you start a session**

```
source /cvmfs/belle.kek.jp/grid/gbasf2/pro/tools/setup.sh
gb2_proxy_init -g belle
```

Local installation

- If you would like to use a local installation, some assembly (configuration) is required

> **Note**
>
> The following commands are **only** needed **when installing gbasf2**

```
mkdir <path_to_install> && cd <path_to_install>
wget -N http://dirac.cc.kek.jp/dirac/dirac-install.py
python dirac-install.py -V Belle-KEK
source bashrc && dirac-proxy-init -x
dirac-configure --cfg defaults-Belle-KEK.cfg
```

- After installation, set the gbasf2 environment
  - The detailed procedure can be found here

> **Note**
>
> The two following commands **must** be used **each time you start a session**

```
source ~/<path_to_install>/BelleDIRAC/gbasf2/tools/setup
gb2_proxy_init -g belle
```

Preparing to submit with gbasf2

- Set your gbasf2 environment
- After installation, set the gbasf2 environment
- To confirm if the set up worked properly, we can execute `gb2_proxy_info`

```
$ gb2_proxy_info
subject      : /DC=org/DC=cilogon/C=US/O=Brookhaven National Laboratory/CN=Justin Guilliams A21194426/CN=2408129987/CN=2269844749
issuer       : /DC=org/DC=cilogon/C=US/O=Brookhaven National Laboratory/CN=Justin Guilliams A21194426/CN=2408129987
identity     : /DC=org/DC=cilogon/C=US/O=Brookhaven National Laboratory/CN=Justin Guilliams A21194426
timeleft     : 23:28:58
DIRAC group  : belle
rfc          : True
path         : /tmp/x509up_u47759
username     : justing
properties   : NormalUser
VOMS         : True
VOMS fqan    : ['/belle']
Succeed with return value:
0
```

- You should see your proxy info, the time remaining in the session (by default, the session lasts for 24 hours), your DIRAC group, and so on

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment ✓
  II. Locate the input dataset(s) you wish to use on the grid
  III. Submit jobs to the grid
     A. Monitoring your jobs
  IV. Download output to perform offline analysis

II. Locate input dataset(s) on the grid

- The most common task as user of the grid is the submission of jobs with input files, which can be found
  - From the official Belle II MC campaigns
  - From the official data reprocessing and skims
- Files on the grid are distributed, as well as the available resources
- Luckily, as a user, you don't have to worry about the physical location of files
  - A file catalog keeps record of where the files are located
- Let’s look at how data is handled on the grid

Datasets and Datablocks on the grid

- On the grid, files are classified inside datasets
- Every dataset is located using a **logical path name (LPN)**
  - LPN: a virtual path used to handle files distributed along the grid sites
- The first portion of the LPN locates the dataset, always beginning with `/belle`
- Examples of dataset LPNs:
  - `/belle/MC/release-04-00-03/DB00000757/MC13a/prod00009434/s00/e1003/4S/r00000/mixed/mdst`
  - `/belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst`

Datasets and Datablocks on the grid

- Each dataset is subdivided into one, or more, datablock(s)
  - Each datablock contains a maximum of 1000 files
  - If a dataset contains more than 1000 files, at least it will be subdivided into at least two datablocks
  - Datablocks are labeled as `subXX`, with an incremental number per each one
    - For example

```
$ gb2_ds_list /belle/MC/release-02-00-01/DB00000411/MC11/prod00005678/s00/e0000/4S/r00000/mixed/mdst
/belle/MC/release-02-00-01/DB00000411/MC11/prod00005678/s00/e0000/4S/r00000/mixed/mdst/sub00
/belle/MC/release-02-00-01/DB00000411/MC11/prod00005678/s00/e0000/4S/r00000/mixed/mdst/sub01
/belle/MC/release-02-00-01/DB00000411/MC11/prod00005678/s00/e0000/4S/r00000/mixed/mdst/sub02
/belle/MC/release-02-00-01/DB00000411/MC11/prod00005678/s00/e0000/4S/r00000/mixed/mdst/sub03
```

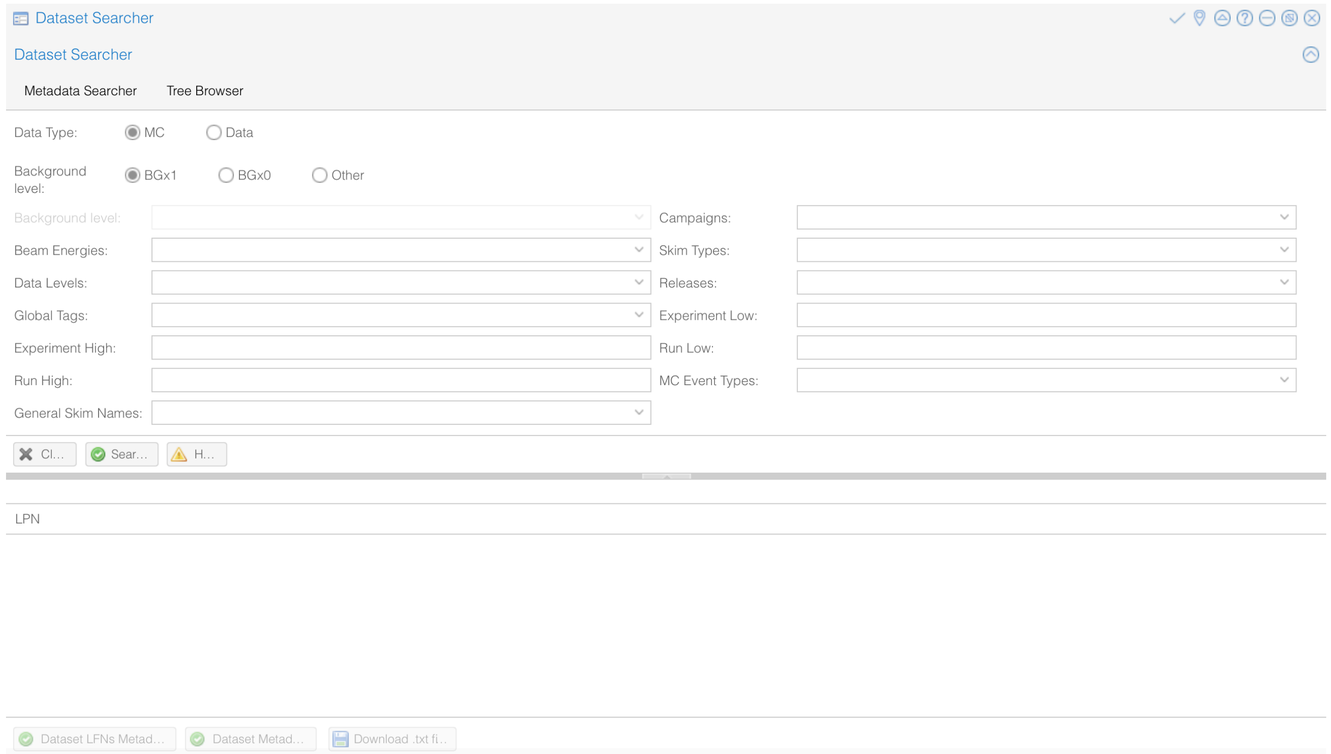
Datasets and Datablocks on the grid

- With the latest releases of gbasf2, a project can be submitted **per dataset** or **per datablock**
  - If submitted **per dataset**, all datablocks within the specified dataset will be resolved
- Inside the project, gbasf2 will produce jobs file-by-file
- The number of output files in the project will be the number of files in the input datablock
  - Or, if submitted per dataset, the number of files within the resolved datablock(s) within the input dataset
- So then, how do we locate MC/data samples?

II. Locate input dataset(s) on the grid

1.The Dataset searcher webapp

- To locate datasets on the grid, we use the Dataset searcher on the DIRAC web portal (Menu icon at bottom left -> BelleDIRACApps -> Dataset searcher)
  - Here, you have the option to search either data or MC, samples with/without beam background (BGx1/ BGx0) and other options to better refine your search



1. The Dataset searcher webapp

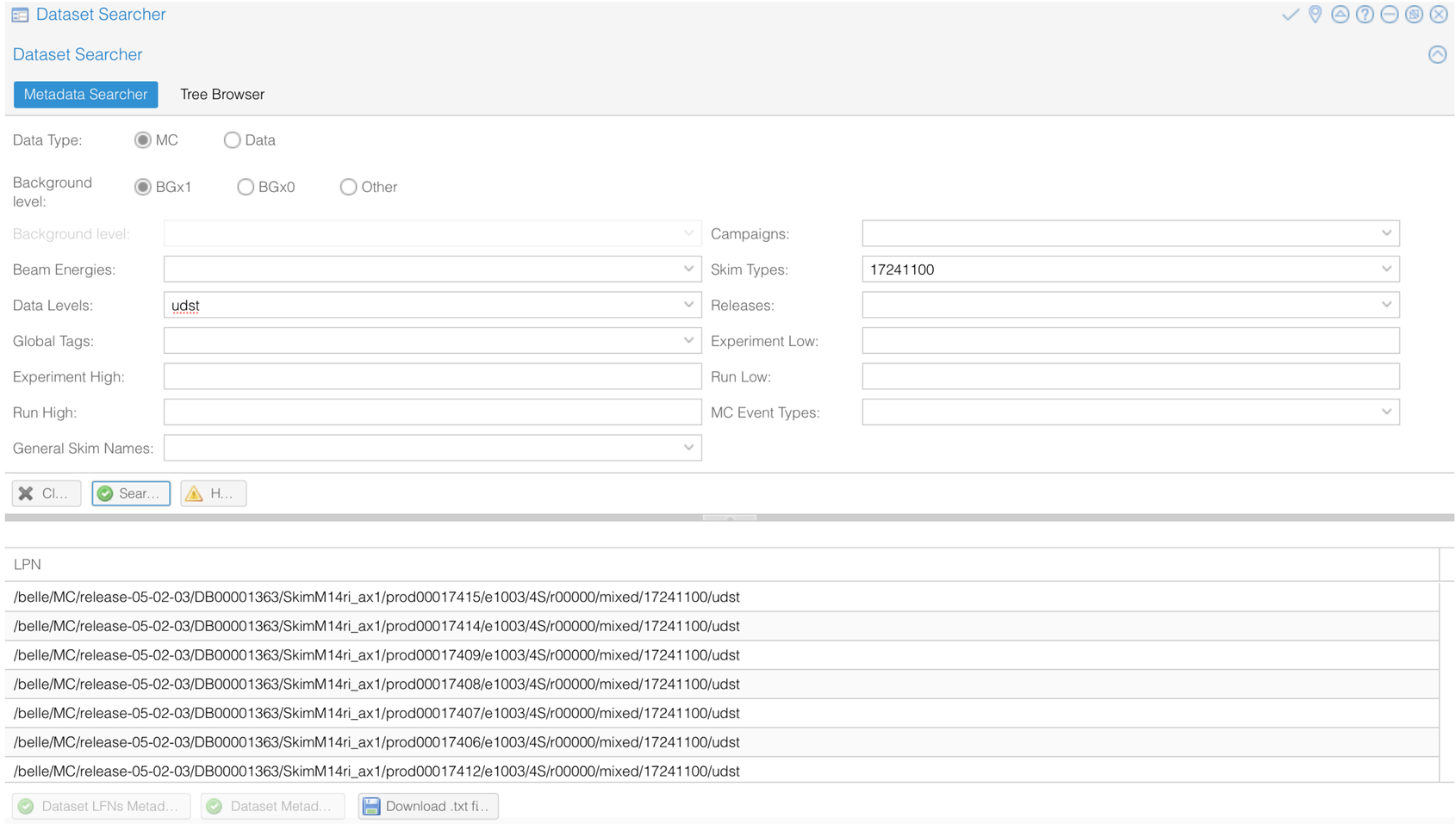
Exercise

Using the Dataset searcher webapp, obtain the first LPN you see for the skim sample that we used earlier for the decay mode `D*+ → [D0 → K−π+π−π+]π+` using the following

```
* Skim Type: 17241100
* Data Type: MC
* Background overlay: BGx1
* Data level: udst
* MC Event Types: mixed
```

1. The Dataset searcher webapp

- **Solution:** `/belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst`



II. Locate input dataset(s) on the grid

1. The Dataset searcher webapp
2. `gb2_ds_search`

2. gb2_ds_search

```
$ gb2_ds_search dataset --help
usage: gb2_ds_search dataset [-h] [-o OUTPUT_FILE] [--campaign CAMPAIGN]
                             [--data_type DATA_TYPE] [--data_level DATA_LEVEL]
                             [--run_high RUN_HIGH] [--exp_high EXP_HIGH]
                             [--run_low RUN_LOW] [--exp_low EXP_LOW]
                             [--mc_event MC_EVENT] [--skim_decay SKIM_DECAY]
                             [--general_skim GENERAL_SKIM]
                             [--beam_energy BEAM_ENERGY]
                             [--global_tag GLOBAL_TAG] [--release RELEASE]
                             [--bkg_level BKG_LEVEL]

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Output a text file containing all matching datasets.
  --campaign CAMPAIGN   The MC or Data production campaign name.
  --data_type DATA_TYPE
                        mc or data
  --data_level DATA_LEVEL
                        udst, mdst, etc
  --run_high RUN_HIGH   The highest allowed run number(INTEGER
                        VALUE)(inclusive).
  --exp_high EXP_HIGH   The highest allowed Experiment number (INTEGER VALUE)
                        (inclusive).
  --run_low RUN_LOW     The lowest allowed Run number (INTEGER VALUE)
                        (inclusive).
  --exp_low EXP_LOW     The highest allowed Experiment number (INTEGER VALUE)
                        (inclusive).
  --mc_event MC_EVENT   The MC event type ("uubar", "1110043100", etc) used
                        for
  --skim_decay SKIM_DECAY
                        The skim type used to reconstruct and select events.
  --general_skim GENERAL_SKIM
                        The general skim name (not in use currently!)
  --beam_energy BEAM_ENERGY
                        4S, 5S, etc
  --global_tag GLOBAL_TAG
                        The global tag used to create the dataset.
  --release RELEASE     The basf2 release used to create the dataset.
  --bkg_level BKG_LEVEL
                        Background Level for MC .
```

- A command-line tool that interacts with the Dataset searcher
- You can see how to use this tool by executing `gb2_ds_search dataset --help`

2. gb2_ds_search

- A command-line tool that interacts with the Dataset searcher
- You can see how to use this tool by executing `gb2_ds_search dataset –help`
- Example:

```
$ gb2_ds_search dataset --data_type mc --skim_decay 14120601 --campaign SkimM13ax1 --beam_energy 4S --mc_event uubar --bkg_level BGx1
Matching datasets found:
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013046/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013047/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013048/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013049/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013050/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013051/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013052/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013053/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013054/e1003/4S/r00000/uubar/14120601/udst
/belle/MC/release-04-02-00/DB00000898/SkimM13ax1/prod00013055/e1003/4S/r00000/uubar/14120601/udst
```

2. gb2_ds_search

Exercise

This time using the command-line tool, obtain the first LPN you see for the skim sample that we used earlier for the decay mode `D*+ → [D0 → K−π+π−π+]π+` using the following

```
* Skim Type: 17241100
* Data Type: MC
* Background overlay: BGx1
* Data level: udst
* MC Event Types: mixed
```

2. gb2_ds_search

- **Solution**:

```
$ gb2_ds_search dataset --data_type mc --skim_decay 17241100 --mc_event mixed --bkg_level BGx1
Matching datasets found:
/belle/MC/release-05-02-18/DB00001363/SkimM14ri_ax1/prod00023415/s00/e1003/4S/r00000/mixed/17241100/udst
/belle/MC/release-05-02-18/DB00001363/SkimM14ri_ax1/prod00023414/s00/e1003/4S/r00000/mixed/17241100/udst
/belle/MC/release-05-02-18/DB00001363/SkimM14ri_ax1/prod00023416/s00/e1003/4S/r00000/mixed/17241100/udst
/belle/MC/release-05-02-18/DB00001363/SkimM14ri_dx1/prod00024206/s00/e1003/4S/r00000/mixed/17241100/udst
```

2. gb2_ds_search

- If we want additional info for one of the datasets that we just searched for, we can use `gb2_ds_query_dataset`
- For Example:

```
$ gb2_ds_query_dataset -l /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst
udst
       dataset: /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst
       creationDate: 2021-05-06 10:40:36
       lastUpdate: 2021-05-11 10:31:19
       nFiles: 68
       size: 71086246048
       status: good
       productionId: 17415
       transformationId: 453586
       owner: g:belle_skim
       mc: SkimM14ri_ax1
       stream:
       dataType: mc
       dataLevel: udst
       beamEnergy: 4S
       mcEventType: mixed
       generalSkimName:
       skimDecayMode: 17241100
       release: release-05-02-03
       dbGlobalTag: DB00001363
       sourceCode:
       sourceCodeRevision:
       steeringFile: skim/SkimM14ri_ax1/release-05-02-03/SkimScripts/CharmLow_Skim.py
       steeringFileRevision:
       experimentLow: 1003
       experimentHigh: 1003
       runLow: 0
       runHigh: 0
       logLfn:
       parentDatasets:
       description: SkimM14ri_ax1 CharmLow skim on MC14_mixedBGx1_b10.
```

II. Locate input dataset(s) on the grid

1. The Dataset searcher webapp
2. `gb2_ds_search`
3. Collections

3. Collections

- A container which holds related datasets
  - Example: ‘Moriond_2021’
    - /belle/Data/release-03-02-02/DB00000654/proc9/prod00008521/e0008/4S/r00043/mdst
    - /belle/Data/release-03-02-02/DB00000654/proc9/prod00008521/e0008/4S/r00044/mdst
    - ....
- Datasets belonging to a specific collection can be viewed using

```
gb2_ds_search collection --list_all_collections /belle/collection/XXX/*
```

3. Collections

- A container which holds related datasets
- Datasets belonging to a specific collection can be viewed using

```
gb2_ds_search collection --list_all_collections /belle/collection/Data/*

/belle/collection/Data/Moriond2022_all_4Soffres_v1
/belle/collection/Data/Moriond2022_all_4S_v1
/belle/collection/Data/Moriond2022_hadron_4Soffres_v1
/belle/collection/Data/Moriond2022_hadron_4S_v1
/belle/collection/Data/proc13_chunk1_all_4S_10601400_v2
/belle/collection/Data/proc13_chunk1_all_4S_offres_v2
/belle/collection/Data/proc13_chunk1_all_4S_v2
/belle/collection/Data/proc13_chunk1_had_4S_10601300_v2
/belle/collection/Data/proc13_chunk1_had_4S_10601500_v2
/belle/collection/Data/proc13_chunk1_had_4S_offres_v2
/belle/collection/Data/proc13_chunk1_had_4S_v2
/belle/collection/Data/proc13_chunk2_all_4S_offres_v1
/belle/collection/Data/proc13_chunk2_all_4S_v1
/belle/collection/Data/proc13_chunk2_had_4S_10601300_v1
/belle/collection/Data/proc13_chunk2_had_4S_10601500_v1
/belle/collection/Data/proc13_chunk2_had_4S_offres_v1
/belle/collection/Data/proc13_chunk2_had_4S_v1
```

3. Collections

- A container which holds related datasets
- Datasets belonging to a specific collection can be viewed using `gb2_ds_search collection --list_all_collections /belle/collection/XXX/*`
- Collections can be used directly with gbasf2 job submission by passing the collection name to the -i flag

```
gbasf2 yourSteeringScript.py -p myProjName \
       -i /belle/collection/Data/Moriond2022_hadron_4S_v1 \
       -s light-2201-venus
```

- More information about collections can be seen here

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment ✓
  II. Locate the input dataset(s) you wish to use on the grid ✓
  III. Submit jobs to the grid
     A. Monitoring your jobs
  IV. Download output to perform offline analysis

III. Submit jobs to the grid

- Submission on the command-line has the basic form

```
gbasf2 <your_steering_file.py> -p <project_name> -i <input_dataset> -s <available_basf2_release>
```

  - Here, we will be submitting the dataset LPN as input; it should be specified with the `-i` flag
  - We will use `—dryrun` to see if everything looks ok prior to submission

```
$ gbasf2 ~justin/public/tutorial2022/grid/gbasf2Tutorial2022.grid.py \
         -p gb2Tutorial2022_Dst2D0pi_D02k3pi \
         -i /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst \
         -s light-2205-abys --dryrun
```

III. Submit jobs to the grid

- Since everything looks good, time to submit the jobs
  - **Note**: I am using a wildcard as submission where I am only using one file; you do not need to do likewise

```
$ gbasf2 ~justin/public/tutorial2022/grid/gbasf2Tutorial2022.grid.py \
         -p gb2Tutorial2022_Dst2D0pi_D02k3pi \
         -i '/belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/*01.root' \
         -s light-2205-abys
```

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment ✓
  II. Locate the input dataset(s) you wish to use on the grid ✓
  III. Submit jobs to the grid ✓
     A. Monitoring your jobs
  IV. Download output to perform offline analysis

Monitoring your jobs

- How can you see the status of your jobs? Like searching for datasets, there are two ways to check the status of jobs, also
  1. Command-line gb2 tools
  2. The Webapp

1. Command-line gb2 tools

- From the command-line, we can use `gb2_project_summary` and `gb2_job_status` (along with the flag `-p` to specify the project name)

```
$ gb2_project_summary -p gb2Tutorial_Dst2D0pi_D02k3pi
          Project                Owner    Status    Done    Fail    Run    Wait    Submission Time(UTC)    Duration
=============================================================================================================
gb2Tutorial_Dst2D0pi_D02k3pi    justing    Good      1       0       0      0       2022-07-12 18:37:19     00:12:25
```

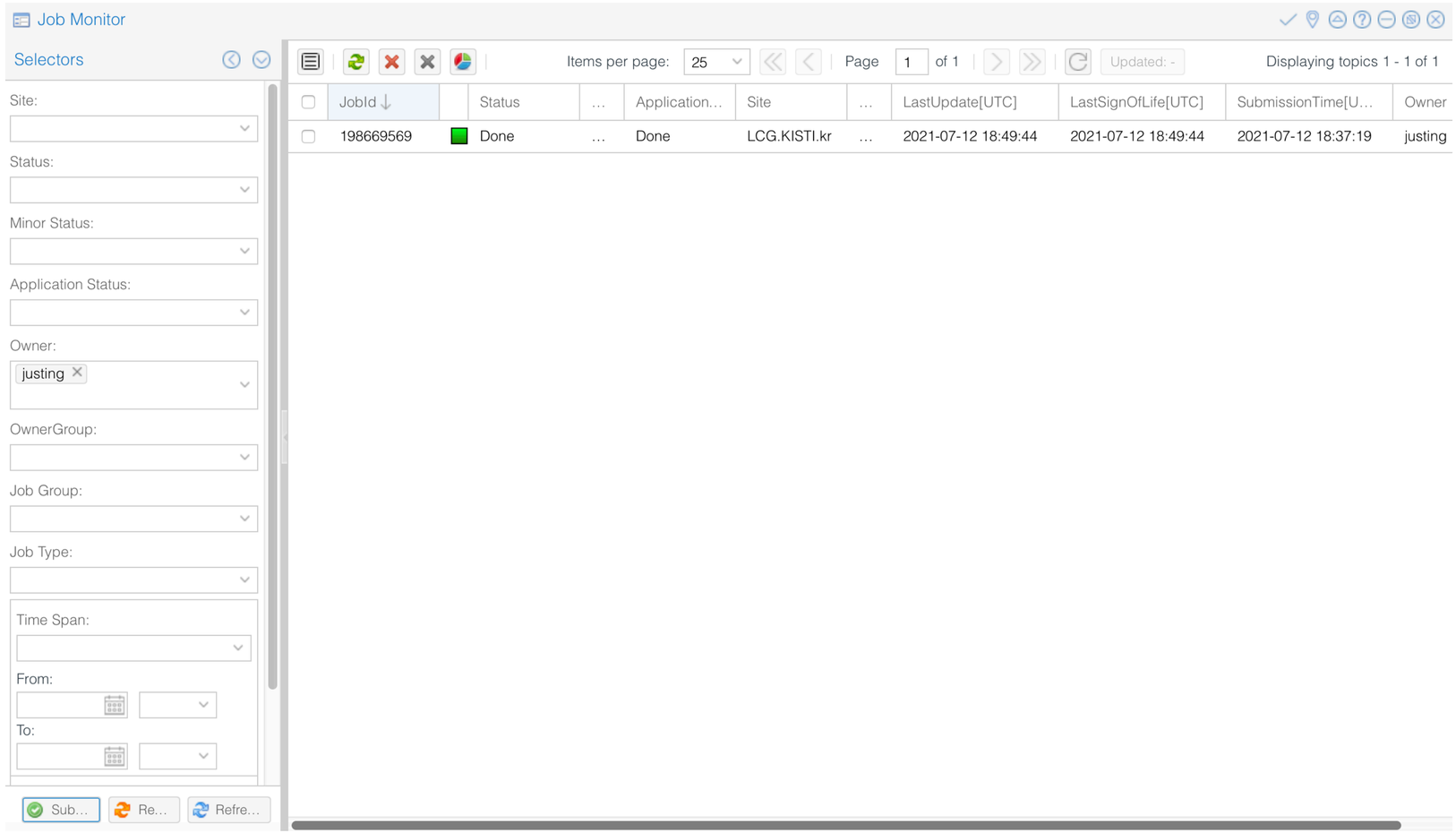
```
$ gb2_job_status -p gb2Tutorial_Dst2D0pi_D02k3pi
 Job id      Status       MinorStatus        ApplicationStatus        Site
=============================================================================
198669569    Done      Execution Complete    Done                 LCG.KISTI.kr

--- Summary of Selected Jobs ---
Completed:0  Deleted:0  Done:1  Failed:0  Killed:0  Running:0  Scouting:0  Stalled:0  Waiting:0
```

Monitoring your jobs

- How can you see the status of your jobs? Like searching for datasets, there are two ways to check the status of jobs, also
  1. Command-line gb2 tools
  2. The Webapp

2. The Webapp

- Use Job Monitor on the DIRAC web portal
  (Menu icon at bottom left -> Applications -> Job Monitor)



Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment ✓
  II. Locate the input dataset(s) you wish to use on the grid ✓
  III. Submit jobs to the grid ✓
     A. Monitoring your jobs ✓
  IV. Download output to perform offline analysis

IV. Downloading output

- When your jobs finish, you will be able to handle the output
- You can list the output by using `gb2_ds_list`
  - The output files will be located in your user space `/belle/user/<username>/<project_name>`

```
$ gb2_ds_list /belle/user/justing/gb2Tutorial_Dst2D0pi_D02k3pi/sub00
/belle/user/justing/gb2Tutorial_Dst2D0pi_D02k3pi/sub00/ntuple_00000_job198669569_00.root
```

IV. Downloading output

- Now, to download the files, use `gb2_ds_get`

```
# Here, we will create a directory to store the files of the tutorial under our home directory
mkdir -p ~/gbasf2Tutorial2022 && cd ~/gbasf2Tutorial2022

# Downloading the files
gb2_ds_get /belle/user/justing/gb2Tutorial2022_Dst2D0pi_D02k3pi
```

- **NOTE:** you can submit jobs or download files from any local machine where gbasf2 is installed

gb2_ds_rm

- Tool allowing users to delete datasets/ datablocks/files from grid
- **Be thoughtful of others - don't clog up resources others need also**
  - After they are finished running, locally download your jobs from the grid
  - Delete your datasets **as soon as possible**
    - avoid needless clogging of memory space on storage elements

```
$ gb2_ds_rm --usage
usage: gb2_ds_rm.py [-h] [-v] [--usage] [-f] [-u USER] [-r {MC,data,user}]
                    [--noBar]
                    dataset [dataset ...]

positional arguments:
  dataset               specify dataset(s) name

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose         increase verbosity (up to -vv)
  --usage               show detailed usage
  -f, --force           skip confirmation
  -u USER, --user USER  specify user name
  -r {MC,data,user}, --subcate {MC,data,user}
                        specify a dataset category
  --noBar               disable status bar


Asynchronously removes files and metadata associated with the dataset or project name provided.
All replicas on the SEs are deleted.

Examples::

  $ gb2_ds_rm project_name
  $ gb2_ds_rm "/belle/user/hideki/project_*"
  $ gb2_ds_rm -u somebody project_name
  $ gb2_ds_rm -f project_name
```

gb2_ds_rm

- Tool allowing users to delete datasets/datablocks/files from grid
- **Be thoughtful of others - don't clog up resources others need also**
- Formerly, tool ***deleted file-by-file - very slow/inefficient***
- Integrated tool with Rucio client tools, adding ***asynchronous deletion functionality***
  - ***fast/efficient deletion*** - test showed deletion of dataset performed 78 times faster than previous implementation

- **Old**

```
real            78m40.617s
user            6m18.085s
sys             0m35.607s
```

- **New**

```
$ time gb2_ds_rm new_gb2_ds_rm
LFNs to remove:

                Dataset                   Files
=============================================
/belle/user/justing/new_gb2_ds_rm/sub00   803
Do you want to remove following files:
Please type [Y] or [N]: Y
Successfully removed FileCatalog/AMGA entries, FileCatalog/AMGA directories for dataset /belle/user/justing/new_gb2_ds_rm/sub00
Successfully removed FileCatalog/AMGA entries, FileCatalog/AMGA directories for dataset /belle/user/justing/new_gb2_ds_rm
                 LFN                      result
=============================================
Succeed
/belle/user/justing/new_gb2_ds_rm/sub00   OK
/belle/user/justing/new_gb2_ds_rm         OK

real            1m0.846s
user            0m3.947s
sys             0m0.463s
```

4720 sec vs. 60 sec

For this dataset, the new implementation was at least 78 times faster*!

* Results may vary

Part I: Submitting your first Jobs (using MC)

- The usual workflow is:
  I. Develop a basf2 steering file (same file used with gbasf2, also); Test it locally ✓
     A. If successful, prepare to submit with gbasf2 environment ✓
  II. Locate the input dataset(s) you wish to use on the grid ✓
  III. Submit jobs to the grid ✓
     A. Monitoring your jobs ✓
  IV. Download output to perform offline analysis ✓

Part II: Submitting Jobs using data

- Why discuss this when we know how to submit jobs already?
  - Running over data is a bit different
- Running over signal samples, generic MC and, in our case, skimmed MC samples is fairly straightforward since it requires few LFNs as input
- However, running over data is technically more complicated since every run corresponds to a dataset
  - So, running over data could require dealing with thousands of LFNs
- For example, looking at proc12 data:

```
[justin@ccw01 ~]$ gb2_ds_search dataset --data_type data --campaign proc12 --general_skim hadron --beam_energy 4S
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03399/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03399/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03400/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03400/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03402/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018884/e0012/4S/r03402/hadron/10601300/mdst
.
.
.
```

Part II: Submitting Jobs using data

1. Handling large datasets

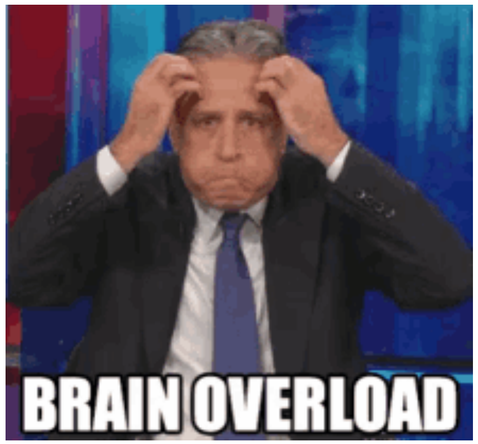
1. Handling large datasets

- To make life easier, we will use proc12 data only from experiment 12

```
gb2_ds_search dataset --data_type data --campaign proc12 --skim_decay "" \
                      --general_skim hadron --beam_energy 4S --exp_low 12 --exp_high 12 \
                      --output tutorial2022/lpns_list/proc12_exp12.list
```

```
cat tutorial2022/lpns_list/proc12_exp12.list | wc -l
1397
```

So many datasets!!



1. Handling large datasets

- Recall: with the latest releases of gbasf2, there is no longer a need to append `/subXX` to the dataset LPN
- If you are dealing with several datasets and you want (or need) to append the LPNs with `/subXX`, there is a quick and easy way to do so

1. Handling large datasets

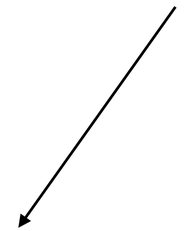
- Like in our example, there are 1397 datasets

```
$ head tutorial2022/lpns_list/proc12_exp12.list
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01000/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01001/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01021/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01022/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01149/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01150/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01151/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01152/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01154/hadron/mdst
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01155/hadron/mdst
```

- But what if we want to only run over one datablock, `/sub00`?
  - We can do the following to quickly append `/sub00` to all of these LPNs

```
$ sed -i 's/mdst/mdst\/sub00/g' tutorial2022/lpns_list/proc12_exp12.list

$ head tutorial2022/lpns_list/proc12_exp12.list
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01000/hadron/mdst/sub00
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01001/hadron/mdst/sub00
/belle/Data/release-05-01-22/DB00001779/proc12/prod00018881/e0012/4S/r01021/hadron/mdst/sub00
```

Again, this is no longer strictly required, but is still available to you if you need it



1. Handling large datasets

For submission of jobs with lists of LFNs as input, use gbasf2 with `--input_dslist`.

Since there are 1397 datasets in our example, there are a few possibilities for treating a large number of dataset LPNs and/or files

- Splitting the list of LPNs (or, files if you have a list of files)
  - GNU command `split` (*which we will use to divide up the list of LPNs*)
- Using more than one file as input per job
  - Flag `-n` (*which we will also use to reduce the number of jobs submitted per LPN*)

First, we will divide the list of LPNs to avoid bottlenecking resources on the grid

1. Handling large datasets

```
# use the -l flag to divide the list by a given number of lines (100 lines = 100 dataset LPNs per file)
# use the --additional-suffix flag to provide explicit file extension for the divided files (use .list or .lst)
# (optional) use the -d flag tells split to use numbers instead of letters when naming the divided files

split -l 100 -d --additional-suffix=.lst tutorial2022/lpns_list/proc12_exp12.list \
                                         tutorial2022/lpns_list/proc12_exp12_


ls -l tutorial2022/lpns_list/
```

1. Handling large datasets

Now, let's try submitting the jobs using just one of the subdivided lists (proc12_exp12_00.lst).

Additionally, we will use the flag `-n 2` with the `gbasf2` command

> This will use two files as input per job, cutting the number of jobs for our project to 56 jobs (101 -> 56 jobs)

Let's try submitting jobs with the same steering file for proc12, exp 12:

```
gbasf2 tutorial2022/grid/gbasf2Tutorial2021.grid.py \
       -p gb2Tutorial2022_Dst2D0pi_D02k3pi \
       --input_dslist tutorial2022/lpns_list/proc12_exp12_00.lst \
       -s light-2205-abys -n 2 --dryrun
```

Part III: Dealing with Issues

1. Rescheduling jobs

1. Rescheduling jobs

- Sometimes, *stuff* happens
- Jobs can fail for several reasons, like
  - Timeout in the transfer of a file between sites
  - Central service not available, or down, for a short period of time
  - An issue in the site hosting the job
  - etc.

1. Rescheduling jobs

- If you see that some of your jobs failed …

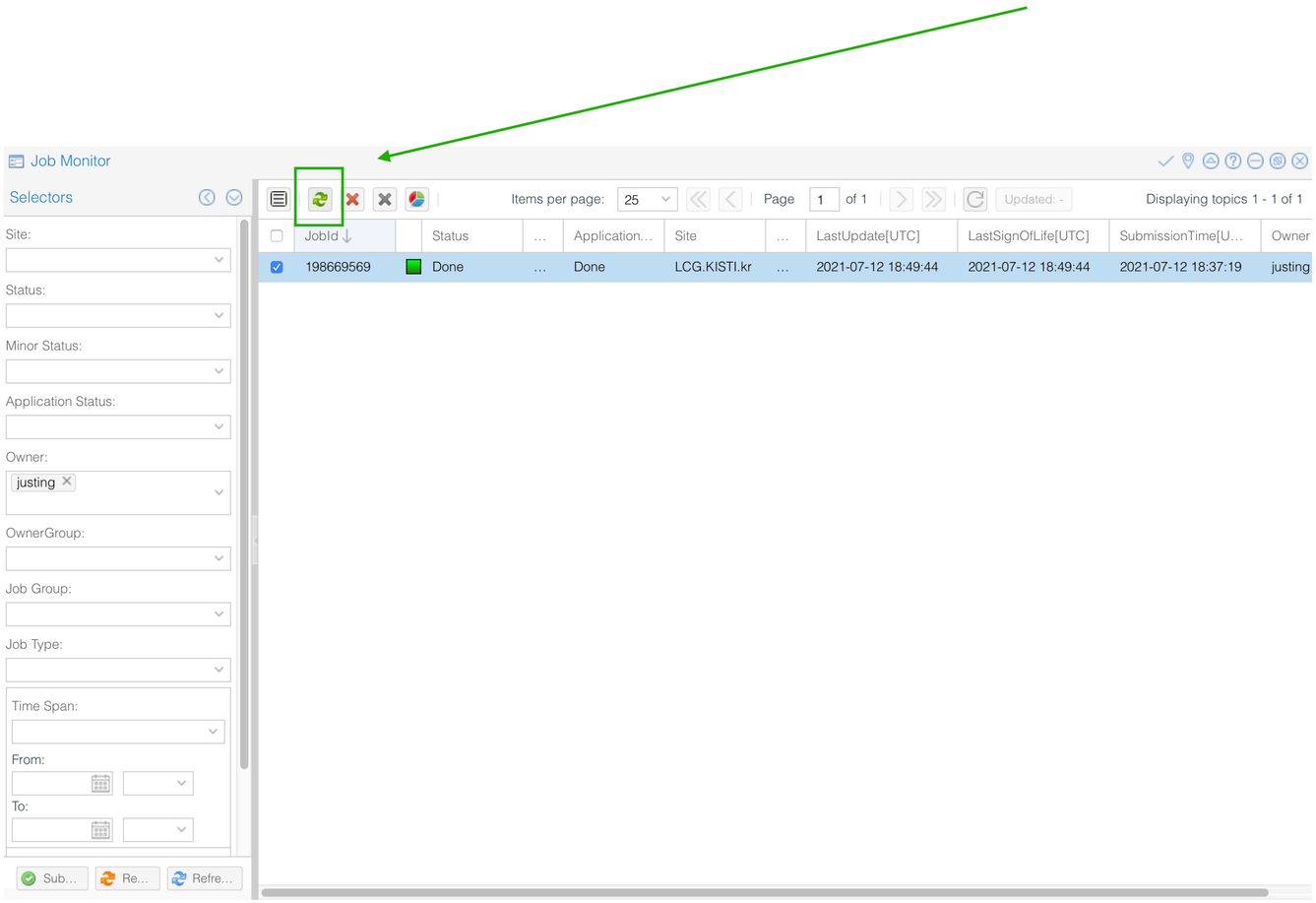
```
[justin@ccw01 ~]$ gb2_project_summary -p gb2Tutorial_Dst2D0pi_D02k3pi
          Project                Owner    Status    Done   Fail   Run   Wait   Submission Time(UTC)   Duration
=============================================================================================================
gb2Tutorial_Dst2D0pi_D02k3pi   justing   Good      1      0      0     0      2022-07-12 18:37:19    00:12:25
```

… you can use `gb2_job_reschedule -p <project name>`

```
[justin@ccw01 ~]$ gb2_job_reschedule --usage | tail -n 13
Resubmit failed jobs or projects.
Only jobs which have fatal status (Failed, Killed, Stalled) are affected.
Exact same sandbox and parameters are reused. Thus you may need to submit different job if they are wrong.

By default, select only your jobs in current group.
Please switch group and user name by options.
All user's jobs are specified by '-u all'.

Examples:

% gb2_job_reschedule -j 723428,723429
% gb2_job_reschedule -p project1 -u user
```

1. Rescheduling jobs

- Along with `gb2_ds_reschedule`, you can also use the Dataset searcher
  - In Job Monitor, select the failed job and then click the "Reschedule" button

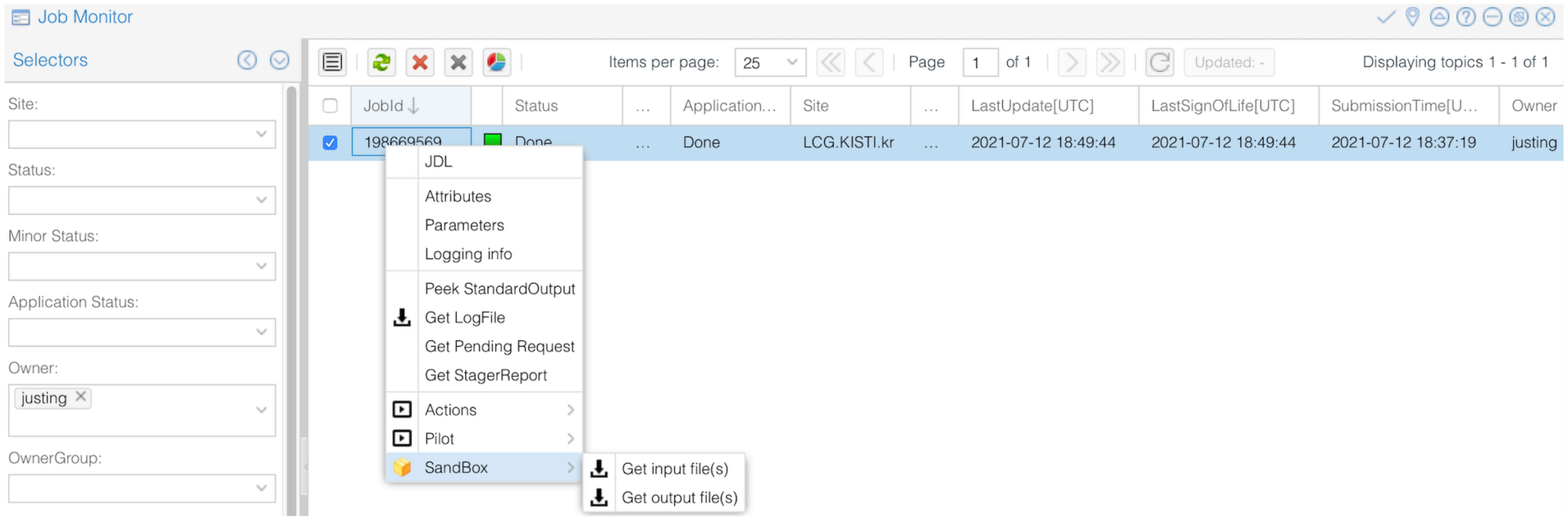


Part III: Dealing with Issues

1. Rescheduling jobs
2. What if all jobs failed?

2. What if all jobs failed?

- If **ALL** your jobs failed, there is probably something wrong with the steering file or the gbasf2 arguments
  - Did you test your steering script locally before submitting jobs to the grid?
- A handy way to see what the issue was is (if possible) downloading the output sandbox
  - It contains the logs related to your job



2. What if all jobs failed?

- If **ALL** your jobs failed, there is probably something wrong with the steering file or the gbasf2 arguments
- A handy way to see what the issue was is (if possible) downloading the output sandbox
- You can also retrieve the output log files from the command-line using `gb2_job_output`

```
[justin@ccw01 ~]$ gb2_job_output -j 198669569
download output sandbox below ./log/JOBID
1 jobs are selected.
Please wait...
                            Result for jobs: ['198669569']
================================================================================
Downloaded: "Job output sandbox retrieved in /gpfs/home/belle2/justin/log/198669569”

[justin@ccw01 ~]$ ls -l ~justin/public/tutorial2022/grid/log/198669569/
total 33
-rw-r--r-- 1 justin b2_belle2 3623 Jul 13 03:38 job.info
-rw-r--r-- 1 justin b2_belle2 8159 Jul 13 03:49 Script1_basf2helper.py.log
-rw-r--r-- 1 justin b2_belle2 1485 Jul 13 03:49 std.out
```

- Then you can look at the log by using
`cat ~justin/public/tutorial2022/grid/log/198669569/Script1_basf2helper.py.log`

Part III: Dealing with Issues

1. Rescheduling jobs
2. What if all jobs failed?
3. What to do if you get stuck?

3. What to do if you get stuck?

- If you happen to get stuck, contact the comp-user-forum
  - When asking your question include:
    - Your user name
    - Project name (or job id)
    - Details about the errors you are seeing;
      (all the details the experts will need to identify the issue)

There are other tools are available

- You can see all the tools available to you from the command-line (gb2 + tab + tab)
- Recall: you can use `–help` and `–usage` to see what each tool does

```
[justin@ccw01 ~]$ gb2_
gb2_admin_fts_monitor          gb2_ds_rep                  gb2_job_kill                gb2_prod_cancelInputFile    gb2_prod_uploadFile
gb2_admin_fts_submit           gb2_ds_rm                   gb2_job_output              gb2_prod_chains             gb2_project_analysis
gb2_admin_remove_amga_dir      gb2_ds_rm_rep               gb2_job_parameters          gb2_prod_downloadFile       gb2_project_summary
gb2_check_downtime             gb2_ds_sanitize             gb2_job_reschedule          gb2_prod_expected           gb2_proxy_destroy
gb2_check_release              gb2_ds_search               gb2_job_status              gb2_prod_extend             gb2_proxy_info
gb2_ds_count_events            gb2_ds_searcher_create      gb2_job_test                gb2_prod_listFile           gb2_proxy_init
gb2_ds_du                      gb2_ds_searcher_delete      gb2_list_destse             gb2_prod_logging            gb2_req_summary
gb2_ds_generate                gb2_ds_searcher_update      gb2_list_queue              gb2_prod_register           gb2_se_list
gb2_ds_get                     gb2_ds_set_datablock_meta   gb2_list_service            gb2_prod_releases           gb2_se_surl
gb2_ds_list                    gb2_ds_set_dataset_meta     gb2_list_site               gb2_prod_restart            gb2_site_analysis
gb2_ds_put                     gb2_ds_set_file_meta        gb2_pilot_summary           gb2_prod_show_metadata      gb2_site_summary
gb2_ds_query_datablock         gb2_ds_siteForecast         gb2_postInstall             gb2_prod_showTransfer       gb2_transformation_summary
gb2_ds_query_dataset           gb2_ds_sync                 gb2_prod_approve            gb2_prod_status             gb2_update
gb2_ds_query_file              gb2_ds_verify               gb2_prod_campaigns          gb2_prod_stop
gb2_ds_register_dataset_meta   gb2_job_delete              gb2_prod_cancel             gb2_prod_summary
```

gb2_ds_rep - Another useful tool

```
$ gb2_ds_rep --usage
usage: gb2_ds_rep [-h] [-v] [--usage] [-s SE] -d SE [-u USER]
                  [-r {MC,data,user}] [-b] [--sole] [--noBar] [-f]
                  dataset [dataset ...]

positional arguments:
  dataset               specify dataset(s) name

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose         increase verbosity (up to -vv)
  --usage               show detailed usage
  -s SE, --src_se SE    source SE
  -d SE, --dst_se SE    destination SE
  -u USER, --user USER  specify user name
  -r {MC,data,user}, --subcate {MC,data,user}
                        specify a dataset category
  -b, --bulk            request for asynchronus operation
  --sole                make sole replica
  --noBar               disable status bar
  -f, --force           skip confirmation

Replicate dataset to other SE.

Examples:

% gb2_ds_rep -d KEK2-SE dataset1
% gb2_ds_rep -d PNNL-SE -u username dataset1
% gb2_ds_rep -d KMI-SE "/belle/user/hideki/dataset*"
```

- Replicates your dataset to another storage element (SE)
  - Can help speed up download of dataset (and the files within) by replicating your dataset to an SE closer to you

Final Remarks

- Help us!
  - Provide your feedback to improve the tools and make them more user-friendly
  - Report issues if/when you see them
  - Take DP and DP Expert shifts
- **Thank you!**