Intro to gbasf2

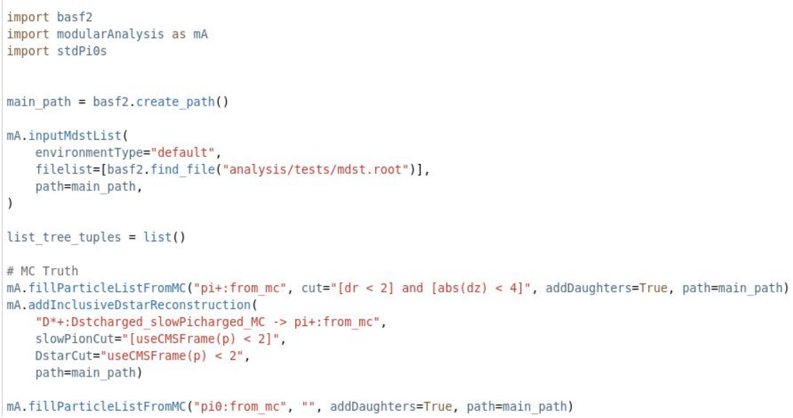
Anil Panta
Justin Guilliams
Jake Bennett

Belle II Summer Workshop
July 13, 2021

THE UNIVERSITY of MISSISSIPPI

Hands-on progress so far:

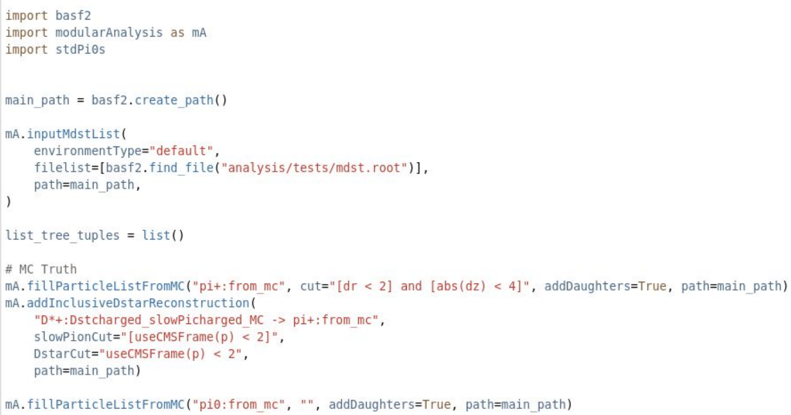
- Your steering script is prepared
- Using basf2 modules



```
import basf2
import modularAnalysis as mA
import stdPi0s


main_path = basf2.create_path()

mA.inputMdstList(
    environmentType="default",
    filelist=[basf2.find_file("analysis/tests/mdst.root")],
    path=main_path,
)

list_tree_tuples = list()

# MC Truth
mA.fillParticleListFromMC("pi+:from_mc", cut="[dr < 2] and [abs(dz) < 4]", addDaughters=True, path=main_path)
mA.addInclusiveDstarReconstruction(
    "D*+:Dstcharged_slowPicharged_MC -> pi+:from_mc",
    slowPionCut="[useCMSFrame(p) < 2]",
    DstarCut="useCMSFrame(p) < 2",
    path=main_path)

mA.fillParticleListFromMC("pi0:from_mc", "", addDaughters=True, path=main_path)
```

Hands-on progress so far:

- Your steering script is prepared
- Using basf2 modules
- With Input files on local resources



```
import basf2
import modularAnalysis as mA
import stdPi0s


main_path = basf2.create_path()

mA.inputMdstList(
    environmentType="default",
    filelist=[basf2.find_file("analysis/tests/mdst.root")],
    path=main_path,
)

list_tree_tuples = list()

# MC Truth
mA.fillParticleListFromMC("pi+:from_mc", cut="[dr < 2] and [abs(dz) < 4]", addDaughters=True, path=main_path)
mA.addInclusiveDstarReconstruction(
    "D*+:Dstcharged_slowPicharged_MC -> pi+:from_mc",
    slowPionCut="[useCMSFrame(p) < 2]",
    DstarCut="useCMSFrame(p) < 2",
    path=main_path)

mA.fillParticleListFromMC("pi0:from_mc", "", addDaughters=True, path=main_path)
```

```
$ basf2 steering_script.py -i myinputfiles.root
```

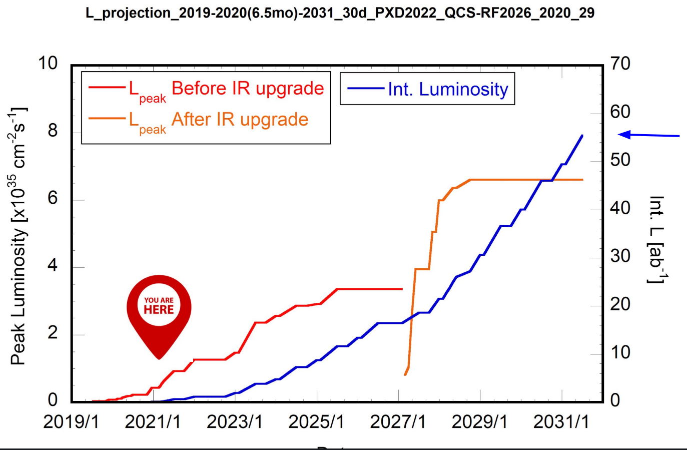
```
$ basf2 my_analysis_script.py -i /local/path/my_input_files.root
```

Can a whole analysis be performed locally?

(on a single computing center or machine)

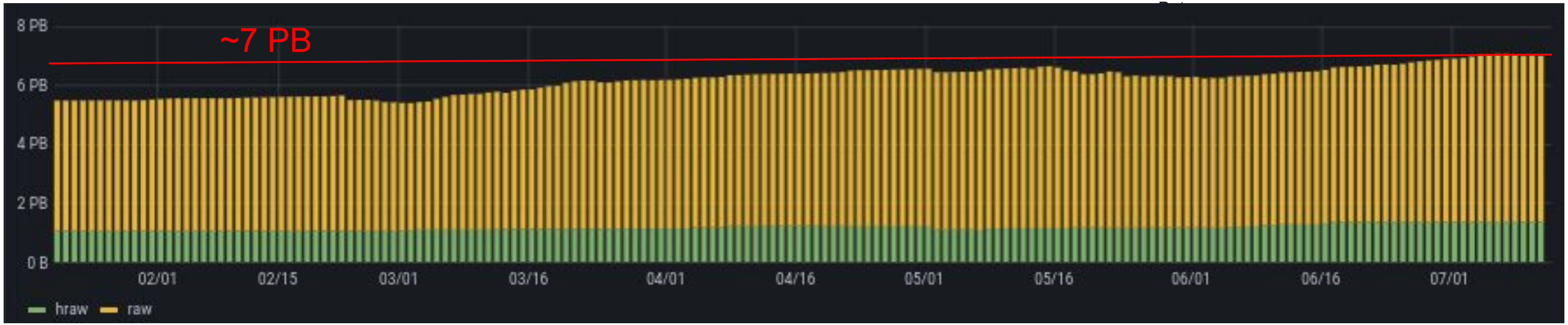
Belle II:

- Target integrated luminosity: 50 /ab
- Massive data volume ~ **hundreds of PetaBytes** (PB)
  -> Huge storage required
  -> Huge computing power required
- Collaborators all over the world
  -> Data distributed around the world



> 50 /ab

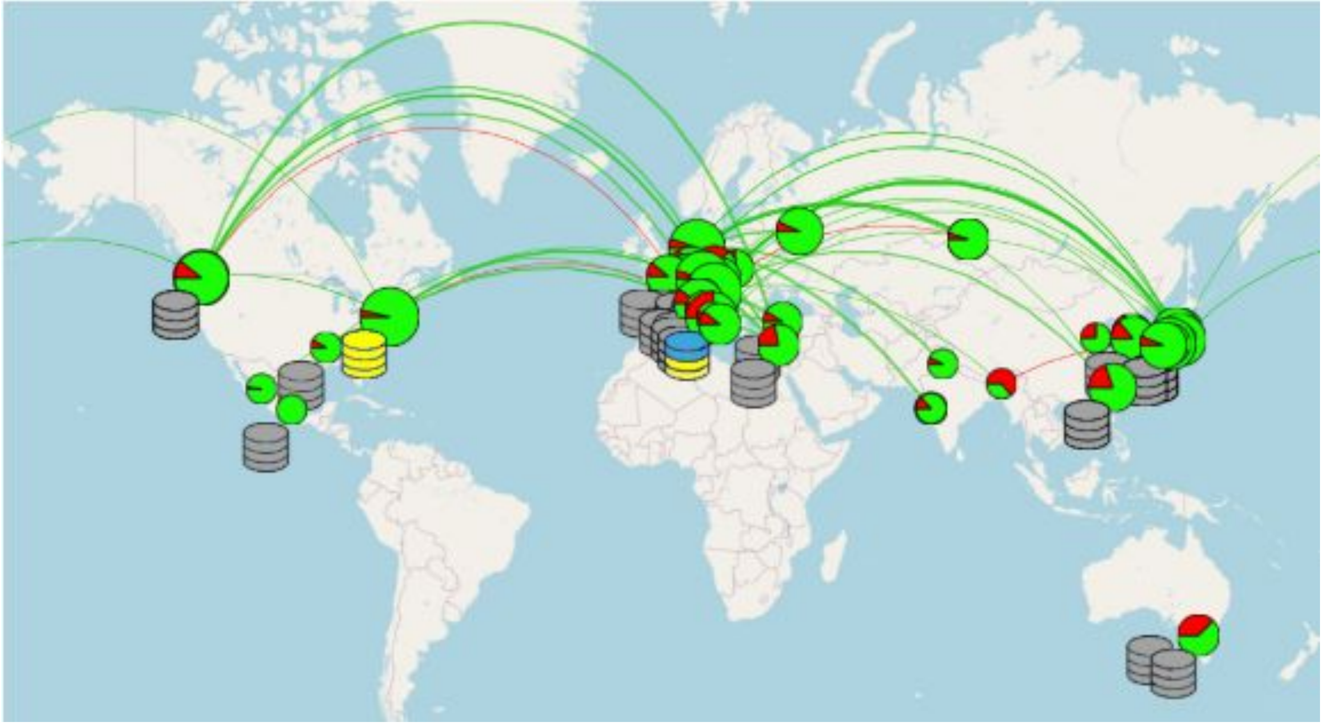
Raw and hRaw so far:



~7 PB

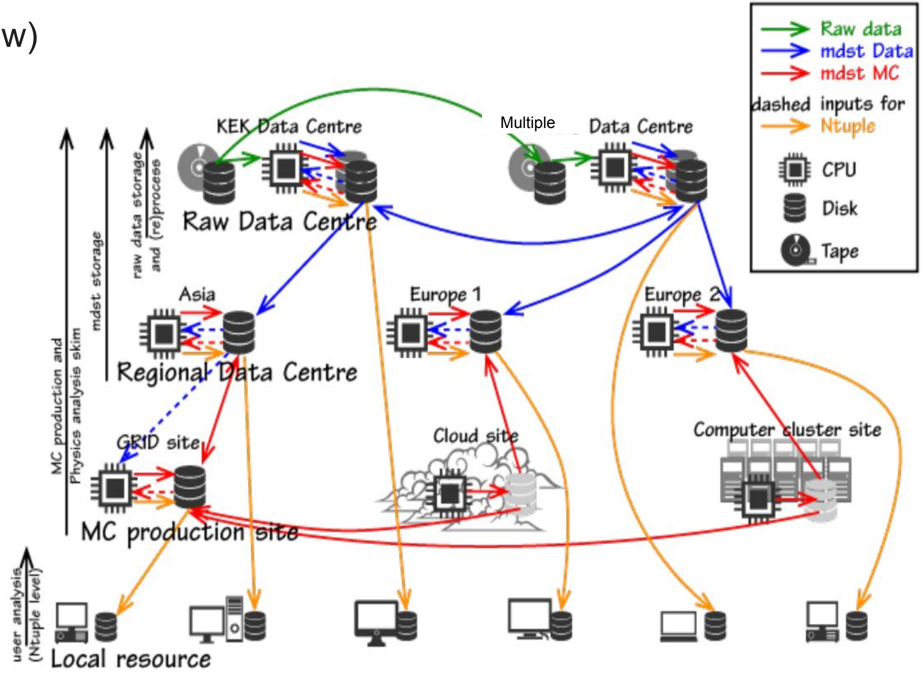
Distributed ("Grid") Computing:

- Loosely-coupled "super virtual computer"
- Collection of heterogeneous resources for computing, storage, cataloging etc.



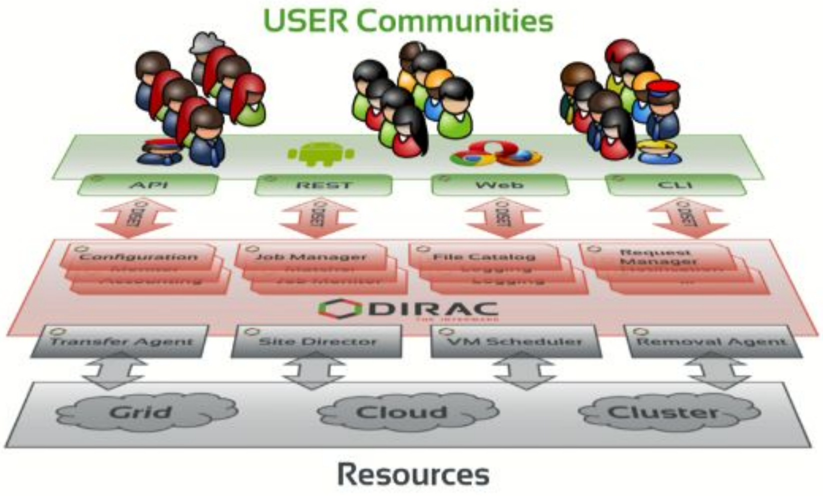
Belle II Computing Model:

- One full copy of all raw data is stored at KEK
- Another replica distributed over multiple (6 for now) raw data centers (BNL, DESY, etc.)
- Processed data stored in regional data centers



BelleDIRAC:

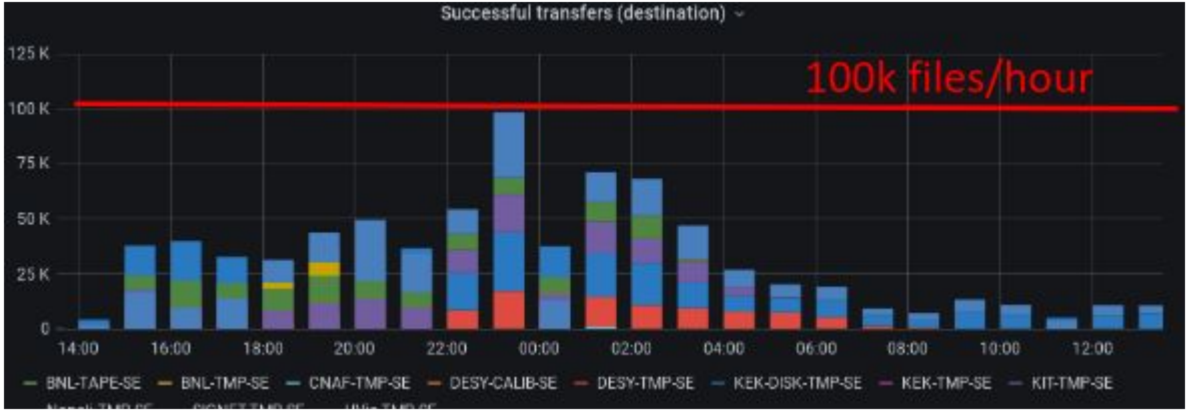
- Distributed Infrastructure with Remote Agent Control (DIRAC)
  - acts as a layer between user and resources
  - provides grid solution like:
    Workload Management (WMS),
    Account Management, etc.



- **BelleDIRAC** is an extension of DIRAC
- It is Belle II specific to fulfill our needs, e.g. production management

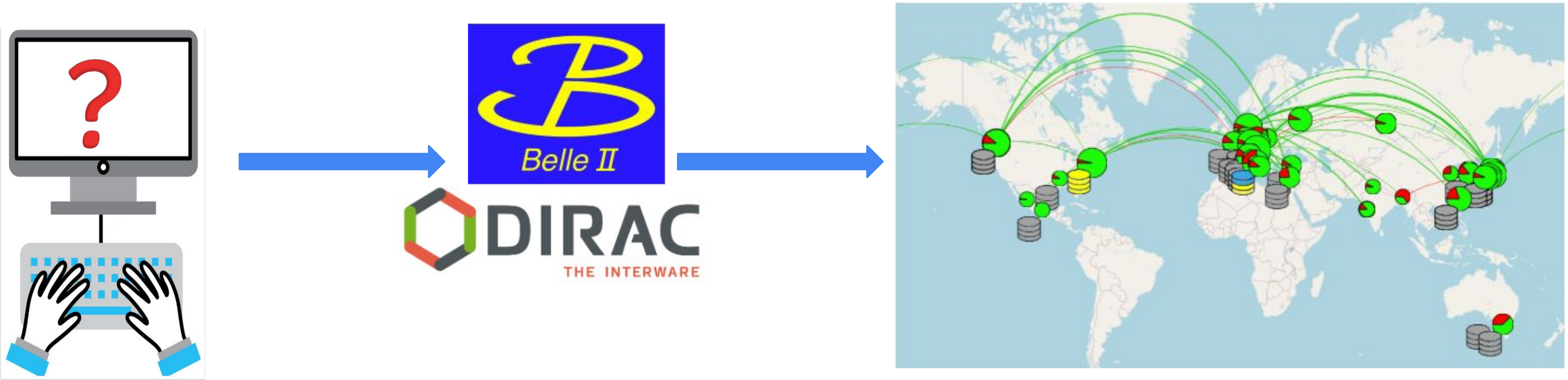
Rucio:

- From Jan 2021, Belle II started using Rucio
- Rucio is a Distributed Data Management software
- Rucio acts as a File Catalog (RFC)



File transfer a few hours after the transition

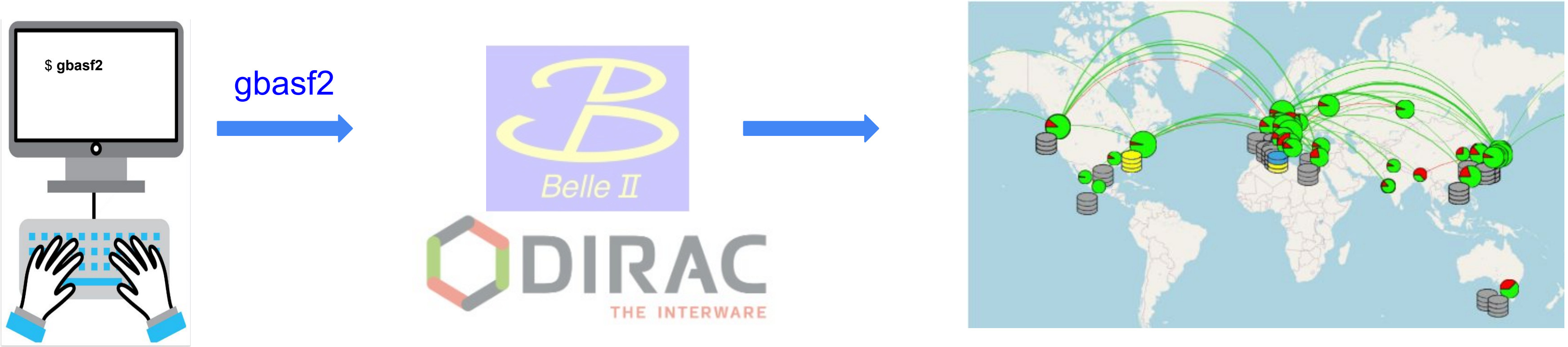
How to interact with the grid?



What tool to use to interact with BelleDIRAC -> Grid ?

gbasf2

- gbasf2 is **grid-based basf2**
- Extends basf2 from local resources to the grid
- Runs the same steering files as for basf2



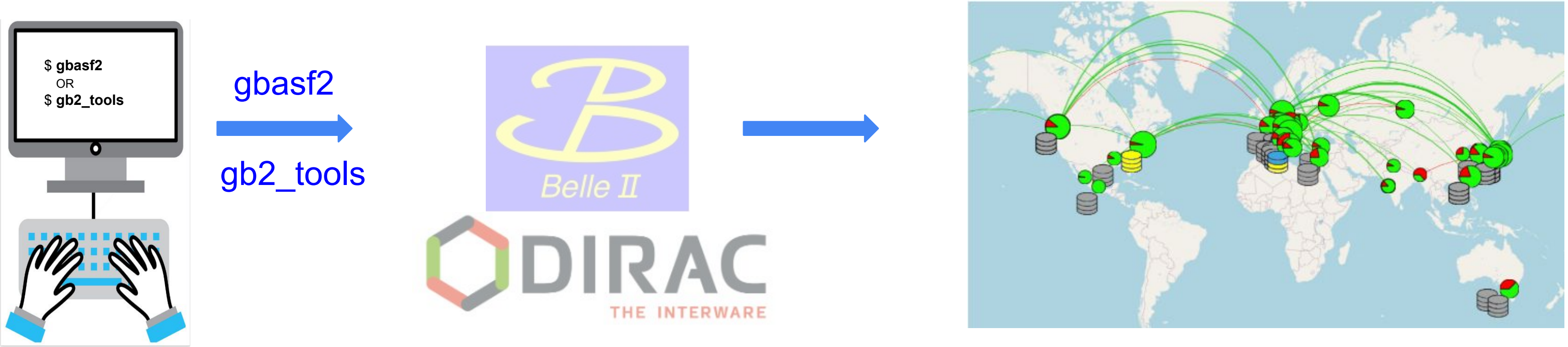
```
$ gbasf2
```

gbasf2

- gbasf2 is **grid-based basf2**
- Extends basf2 from local resources to the grid
- Runs the same steering files as for basf2

gb2_tools

- gb2_tools: **set of BelleDIRAC cmd line tools**
- Used to monitor, manipulate jobs on the grid
- Additional functionality for grid management

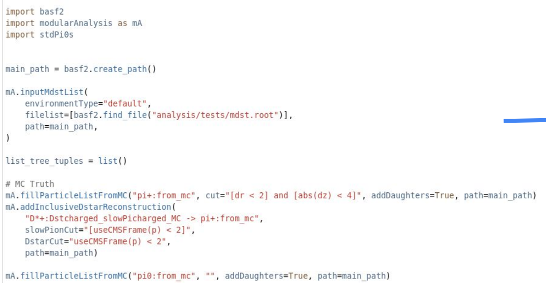


```
$ gbasf2
  OR
$ gb2_tools
```

gbasf2 workflow:

- Create a basf2 steering file
- Run basf2 locally and make sure it works
- Find the input files on the grid
- Submit a gbasf2 project

Steering file



```
import basf2
import modularAnalysis as mA
import stdPi0s


main_path = basf2.create_path()

mA.inputMdstList(
    environmentType="default",
    filelist=[basf2.find_file("analysis/tests/mdst.root")],
    path=main_path,
)

list_tree_tuples = list()

# MC Truth
mA.fillParticleListFromMC("pi+:from_mc", cut="[dr < 2] and [abs(dz) < 4]", addDaughters=True, path=main_path)
mA.addInclusiveDstarReconstruction(
    "D*+:Dstcharged_slowPicharged_MC -> pi+:from_mc",
    slowPionCut="[useCMSFrame(p) < 2]",
    DstarCut="useCMSFrame(p) < 2",
    path=main_path)

mA.fillParticleListFromMC("pi0:from_mc", "", addDaughters=True, path=main_path)
```

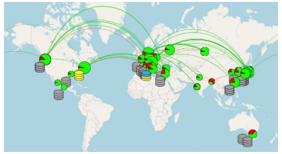
→ basf2

Locally
Tested [‘OK’]

→ **Find Input Files on the Grid**

→ **gbasf2**

↓



How to use gbasf2 ?

- Fulfill the prerequisites:
  1. A valid grid certificate issued within a year and installed in ~/.globus and in a web browser
  2. Belle VO membership registered or renewed within a year: https://voms.cc.kek.jp:8443/voms/belle
  3. DIRAC registration: https://dirac.cc.kek.jp:8443/DIRAC/
- Install gbasf2 or use pre-installed version in CVMFS. gbasf2install

```
$ gbasf2 <steering_scripy.py>  -i <input_path> -p <project_name> -s <basf2_release>
```

- **project_name:** name assigned by you
- **basf2_release:** any available Basf2 software version
- **input_path:** the logical file name (LFN) of the files to analyze

Detail in hands on by Justin.

Input files:

- In basf2 where **input files are local:**

```
$ basf2 my_analysis_script.py -i /local/input/files.root
```

- In gbasf2 where **input files are in storage element in grid:**

```
$ gbasf2 my_analysis_script.py -i  <input_files> -s <basf2_release> -p <project_name>
```

<input_files>
WHERE??
WHAT??

Input files location ‘WHERE’?

- The physical path name (**PFN**) of the file is the actual location of files on grid. For example, **srm://dcache.ijs.si:8443/srm/managerv2?SFN=/pnfs/ijs.si/belle/DATA/belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_000068_prod00017415_task10020000069.root**

- This looks **complicated**. And identifying a PFN for every file is cumbersome and **not appropriate for gbasf2 use**.

Input files location 'WHERE'?

- The physical path name (**PFN**) of the file is the actual location of files on grid. For example,
  srm://dcache.ijs.si:8443/srm/managerv2?SFN=/pnfs/ijs.si/belle/DATA/belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_000068_prod00017415_task10020000069.root

- This looks **complicated**. And identifying a PFN for every file is cumbersome and **not appropriate for gbasf2 use**.

- To keep track of the location of files and replicas, we use a **File Catalog (FC)**, specifically the Rucio FC
- Abstraction of PFN -> **Logical File Name (LFN),** looks like:

  /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_000068_prod00017415_task10020000069.root

- The above path is enough to locate files and replicas on the grid
- Each path is unique

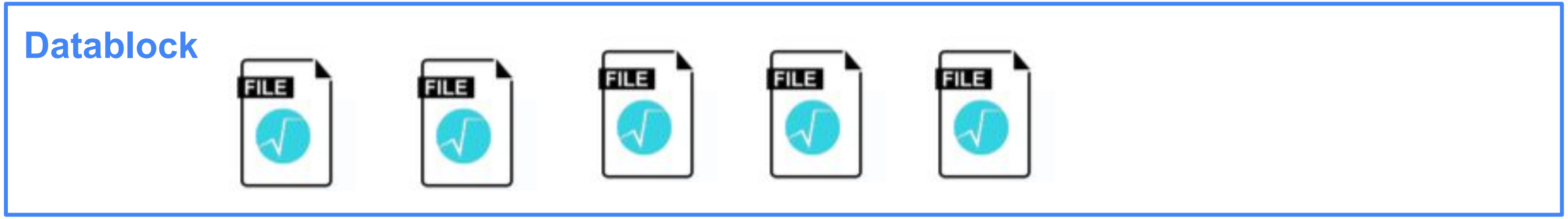
Terminology:

- Units of data management:
  - **File** : LFN (Logical File Name), smallest unit of data

    /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_prod00017415_task10020000069.root

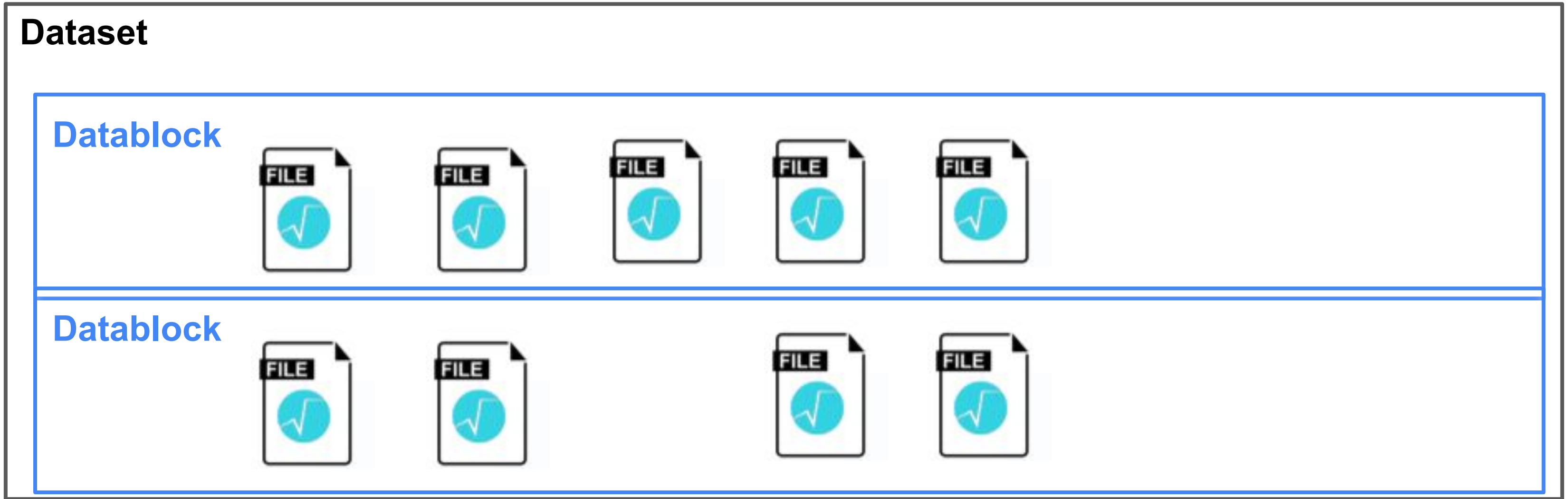
Terminology:

- Units of data management:
  - File : LFN (Logical File Name), smallest unit of data
    /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_prod00017415_task10020000069.root
  - **Datablock :** LPN (Logical Path Name), a block of at most 1000 files
    /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00



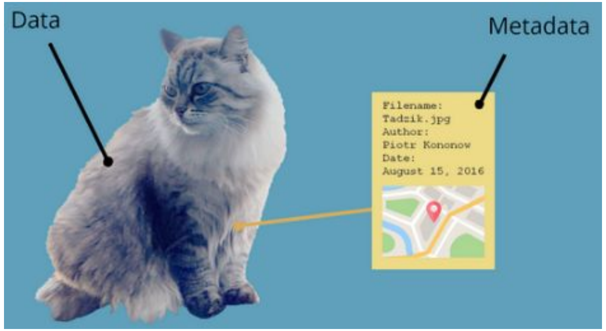
Terminology:

- Units of data management:
    - File : LFN (Logical File Name), smallest unit of data
      /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00/udst_prod00017415_task10020000069.root
    - Datablock **:** LPN (Logical Path Name), a block of at most 1000 files
      /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst/sub00
    - **Dataset :** LPN (Logical Path Name), consists at least 1 dataset
      /belle/MC/release-05-02-03/DB00001363/SkimM14ri_ax1/prod00017415/e1003/4S/r00000/mixed/17241100/udst

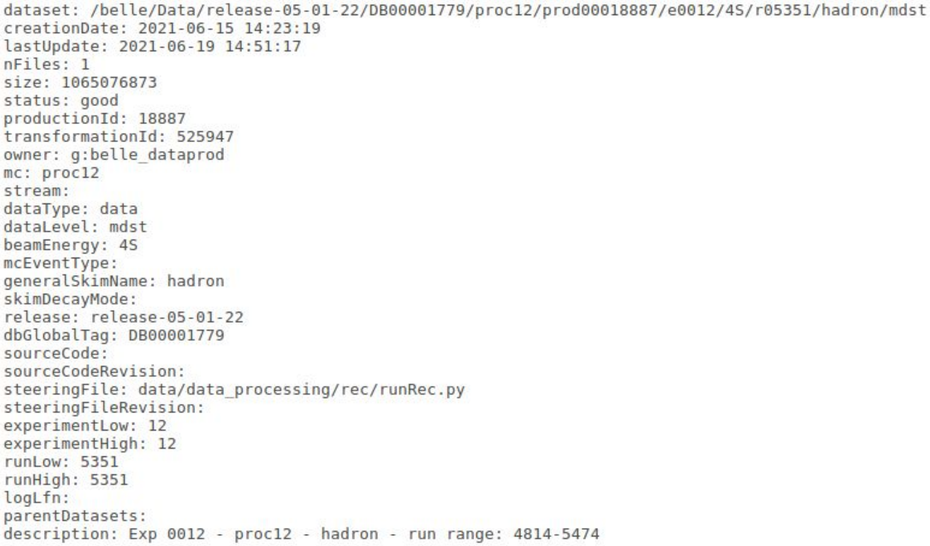


MetaWHAT?

- What kind of files do the LFNs represent?
  -> Metadata, details or description of files/datablocks
- Each LFN and LPN has metadata associated with it
- All metadata are described here



Metadata of dataset:



```
dataset: /belle/Data/release-05-01-22/DB00001779/proc12/prod00018887/e0012/4S/r05351/hadron/mdst
creationDate: 2021-06-15 14:23:19
lastUpdate: 2021-06-19 14:51:17
nFiles: 1
size: 1065076873
status: good
productionId: 18887
transformationId: 525947
owner: g:belle_dataprod
mc: proc12
stream:
dataType: data
dataLevel: mdst
beamEnergy: 4S
mcEventType:
generalSkimName: hadron
skimDecayMode:
release: release-05-01-22
dbGlobalTag: DB00001779
sourceCode:
sourceCodeRevision:
steeringFile: data/data_processing/rec/runRec.py
steeringFileRevision:
experimentLow: 12
experimentHigh: 12
runLow: 5351
runHigh: 5351
logLfn:
parentDatasets:
description: Exp 0012 - proc12 - hadron - run range: 4814-5474
```

Dataset-Searcher (DSS)

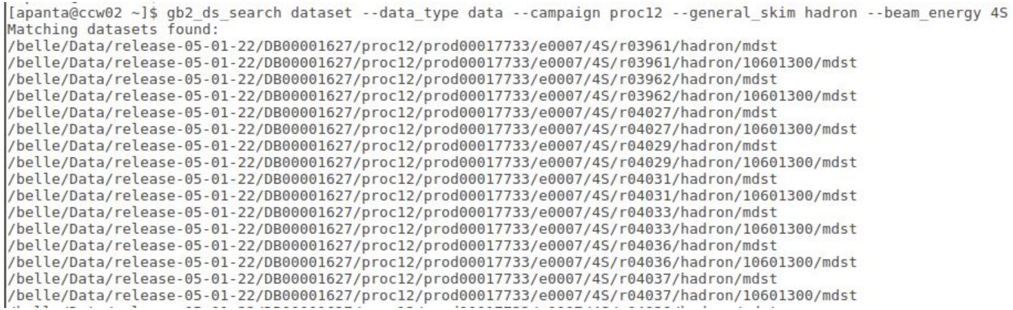
- The DSS is the **tool to find datasets on the grid via their associated metadata**
  - Users select/input relevant metadata of interest
  - Returns a list of datasets matching the metadata
  - Use the dataset as input to gbasf2

* Details in the hands-on session by Justin

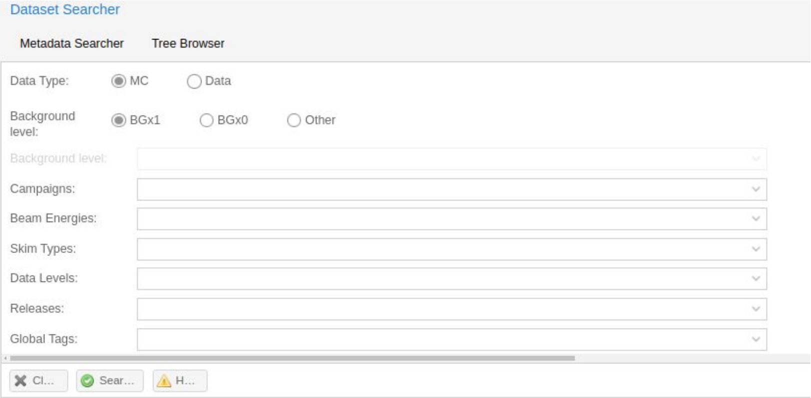
- Two ways to use the DSS

1. gb2 tool: **gb2_ds_search**

2. DIRAC **WebAPP:** **DSS**



```
[apanta@ccw02 ~]$ gb2_ds_search dataset --data_type data --campaign proc12 --general_skim hadron --beam_energy 4S
Matching datasets found:
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r03961/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r03961/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r03962/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r03962/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04027/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04027/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04029/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04029/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04031/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04031/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04033/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04033/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04036/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04036/hadron/10601300/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04037/hadron/mdst
/belle/Data/release-05-01-22/DB00001627/proc12/prod00017733/e0007/4S/r04037/hadron/10601300/mdst
```



Dataset Searcher

Data Type: ● MC ○ Data

Background level: ● BGx1 ○ BGx0 ○ Other

Background level:

Campaigns:

Beam Energies:

Skim Types:

Data Levels:

Releases:

Global Tags:

Lets move to the hands-on session for the DSS and gbasf2