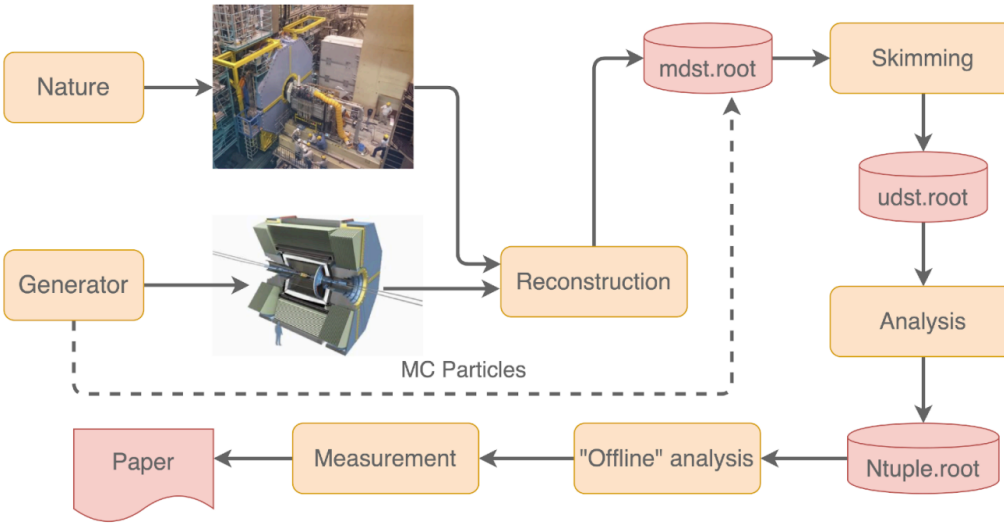
Belle II Analysis Models

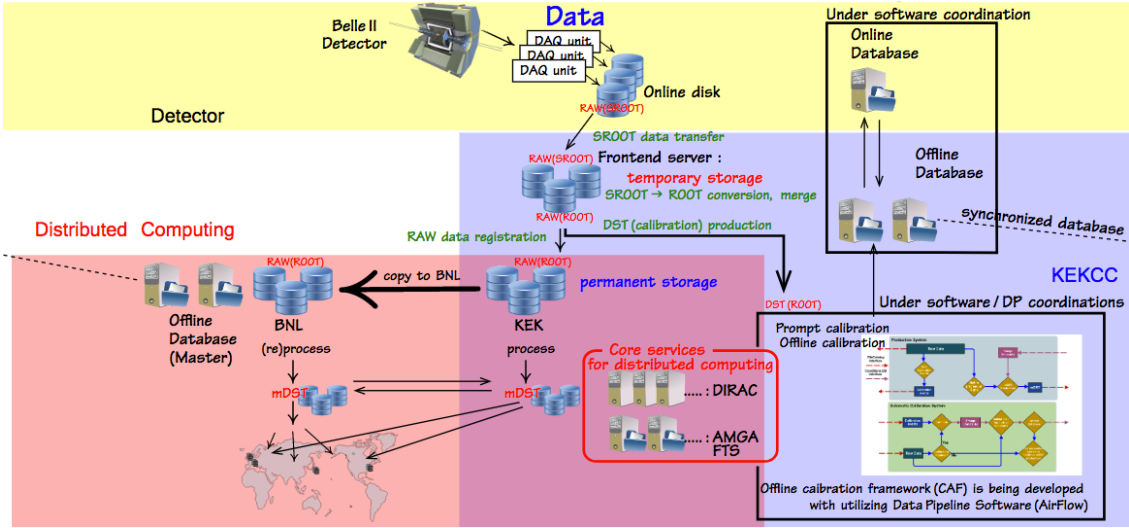
Frank Meier

ROOT Users Workshop
10 May 2022

Analysis Workflow



Belle II data flow



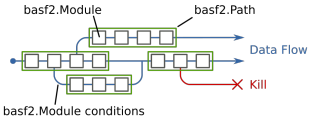
Data formats

In general, Belle II output is stored in ROOT files containing subsets of objects

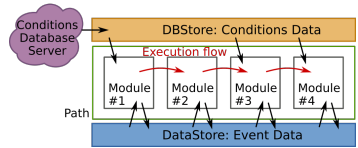
- **RAW**: raw data containing detector information.
  - ~70 kB/event
  - raw data set during 2019-2021 operation: 5 PB
- **cDST**: calibration Data Summary Table
  - ~120 kB/event
  - contains objects needed for calibration
- **mDST**: mini Data Summary Table
  - ~15 kB/event
  - strictly controlled subset of objects necessary for analysis
- **uDST**: user Data Summary Table
  - ~20 kB/event
  - mDST objects + analysis objects (ParticleLists)
  - produced from skims and intended for physics results

Belle II Analysis Software Framework (basf2)

- more than just an “analysis” framework
- performs unpacking of raw data, reconstruction (tracking, calorimeter clustering, PID), ...
- split into packages for all subdetectors as well as analysis and mva
  - can easily use our analysis tools on HLT (High Level Trigger) nodes



- C++ modules that are setup and configured via python scripts
- basf2 links against defined set of external libraries
  - ROOT, EvtGen, Pythia, Geant4, ...
  - many python packages



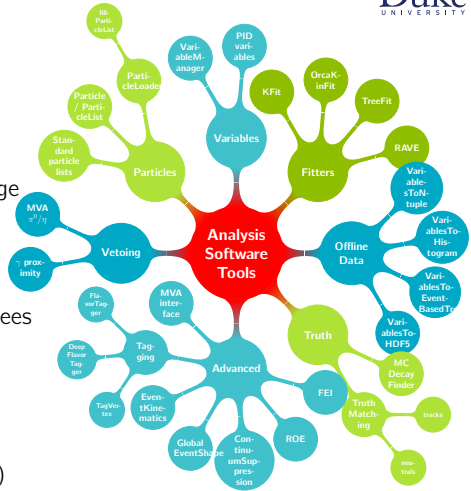
- basf2 source code and documentation publicly available at https://github.com/belle2/basf2 and https://software.belle2.org

Software releases of basf2

- major releases once a year with very thorough validation containing all software changes that are merged to the main branch (after approval of librarian)
- one or two minor releases per major releases with limited amount of new features, usually for specific purpose
- patch releases mostly for bug fixes, especially for data-taking and calibration during data-taking synchronized with maintenance days
- light releases
  - for introduction of new data analysis features
  - contain only framework, mdst, mva, analysis, skim, geometry, online_book, and b2bii packages
  - no unpacking or digitization $\Rightarrow$ only mdst and udst can be processed
  - new version about every two months

Belle II analysis package

- load data
- create particles from mdst dataobjects
- combine particles using decay strings
- apply selection cuts implemented with domain-specific language
- perform high-level analysis operations
  - vertex fits
  - (flavor) tagging
- store variables in ntuples mostly as candidate-based ROOT trees



- analysis package common for all physics working groups (WG)
  - encourage WG-specific code to be pushed to basf2
  - option to (easily) run specialized modules/variables on grid
- B2BII converts Belle data format into Belle II data format

Belle II analysis package

```python
import basf2
import modularAnalysis as ma
import stdV0s
import vertex
import variables.collections as vc

# create path
main = basf2.create_path()

# load input data
ma.inputMdst('inputfile.mdst.root', path=main)

# create ParticleLists
ma.fillParticleList('mu+:good', 'PIDmu > 0.9', path=main)
stdV0s.stdKshorts(fitter='TreeFit', path=main)

# combine particles
ma.reconstructDecay('J/psi:wide -> mu+:good mu-:good', 'abs(dM) < 0.01', path=main)
ma.reconstructDecay('B0:final -> J/psi:wide K_S0:merged', 'Mbc > 5.24', path=main)

# perform vertex fit
vertex.treeFit('B0:final', conf_level=0, massConstraint=['J/psi', 'K_S0'], path=main)

# store variables in output ntuple
ma.variablesToNtuple('B0:final', variables=vc.kinematics, filename='outputfile.root', treename='tree', path=main)

# process events
basf2.process(main)
```

- analysis package common for all physics working groups (WG)
  - encourage WG-specific code to be pushed to basf2
  - option to (easily) run specialized modules/variables on grid
- B2BII converts Belle data format into Belle II data format

Offline analysis

- performed at kekcc (Japan), at National Analysis Computing Facilities (at DESY or INFN), or on local machines
- no coherent Belle II offline analysis framework
  - analysts are mostly on their own on how to get the physics results out of the data
  - surveys showed that people use full range of options: C++, python, ROOT macros, Jupyter notebooks, ...
- MVA method interfaces provided within basf2 (FastBDT, TMVA, ...)
- fitting tools task force
- systematics framework