Computing and Software at Belle II

CHEP 2019 - Adelaide

David Dossett
david.dossett@unimelb.edu.au

The University of Melbourne
The Collaboration

981 active members, 118 institutes, 26 countries

You are here
The Belle II Experiment

• **B Factory Flavour Physics**
  - CKM matrix
  - CP violation
  - Rare B decays
  - Lepton Flavour Violation (τ physics)

• **Precision measurements test SM theory → indirect search for new physics**

• **Better precision requires a LOT of data**

Background subtracted Δt for $B^0 \rightarrow J/\psi K_L$ [arXiv:1212.5342]
The SuperKEKB Accelerator

Design Luminosity $8 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$

$\sim 50 \text{ ab}^{-1}$ total
The Belle II Detector

EM Calorimeter
CsI(Tl), waveform sampling electronics

KL and muon detector
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC
(end-caps, inner 2 barrel layers)

Vertex Detector
2 layers Si Pixels (DEPFET) +
4 layers Si double sided strip DSSD

Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (forward)

Central Drift Chamber
Smaller cell size, long lever arm

electrons (7 GeV)

positrons (4 GeV)

Belle II TDR, arXiv:1011.0352
Software Tools
Analysis Software - basf2

- Common software repository (+ externals) for simulation, reconstruction, analysis tools
  - Individual package sub-directories
  - Core framework is shared
- ROOT for I/O
- C++ heavy lifting in event loop
- Declarative Python steering files
- Multiprocessing of events (no multi-threading)
- Conversion tool for Belle data (b2bii)
  - Can use old data with new MVA and decay reconstruction tools
Jupyter Integration - basf2

In [ ]:

```python
import basf2

from modularAnalysis import *

from variables.utils import create_aliases_for_selected

filename = "http://ckpwaw.physik.uni-karlsruhe.de/~nbraun/BBJpsiKs_mu_20x8_r1000.root"

filelistSI0 = [filename]

main_path = create_path()

queue = handler.create_queue()

inputMdstList("default", filelist=filelistSI0, path=main_path)

fillParticleList("pi+-loose", ‘pid > 0.01’, path=main_path)

fillParticleList("mu+-loose", ‘muid > 0.01’, path=main_path)

# reconstruct Ks -> pi+ pi- decay
# keep only candidates with 0.4 < M(pipi) < 0.6 GeV
reconstructDecay("K_500:pi+-loose", ‘0.4 < M < 0.6’, path=main_path)

# reconstruct J/ψ -> mu+ mu- decay
# keep only candidates with 3.0 < M(mumu) < 3.2 GeV
reconstructDecay("J/psi:mumu -> mu+-loose", ‘3.0 < M < 3.2’, path=main_path)

# reconstruct B0 -> J/ψ Ks decay
# keep only candidates with 5.2 < M(J/psiKs) < 5.4 GeV
reconstructDecay("B0:jpsiKs -> J/psi:mumu K_500:pi+-", ‘5.2 < M < 5.4’, path=main_path)

# perform B0 kinematic vertex fit using only the mu+ mu-
# keep candidates only passing C.L. value of the fit > 0.0 (no cut)
vertexRave( ‘B0:jpsiKs’, path=main_path)

# build the rest of the event associated to the B0
buildRestOfEvent("B0:jpsiKs", path=main_path)

# perform MC matching (MC truth association). Always before Tag/
```
Conditions DB

- Conditions payloads (files) are collected under a “Global Tag” string
- Each payload is valid for a run range
  - Interval of Validity (IoV)
- basf2 downloads necessary payloads at the start of a job
- Brookhaven National Laboratory (BNL) hosts the server
- Investigating contents of tags and payloads is easy and available to all members via CLI in basf2

Track 2: 7th Nov 15:00 - CDB Server Infrastructure
Calibration and Alignment Framework (CAF)

- A basic calibration follows a nice workflow
- We have the CAF to help automate these common procedures

- Run Collector
- Run Collector
- Run Collector

Input Data → Run Collector → Collected Data → Run Algorithm(s) → Constants

Option to parallelise on batch system over input files

basf2 process(es)

histograms, ntuples, TTree etc

subprocess
Calibration and Alignment Framework (CAF)

- Multiple dependent calibrations gets much more complex!
- Simple Python steering files for declaring calibrations
- Restart from checkpoint available

```python
...  
cal_b.depends_on(cal_a)  # cal_a MUST complete before cal_b
fw = CAF()
fw.add_calibration(cal_a)
fw.add_calibration(cal_b)
fw.backend = LSF()  # Configurable if needed
fw.run()
```
Collaborative Tools

• Keeping us in sync with each other

• Many websites and tools:
  – Bitbucket
  – Confluence (wiki)
  – Jira
  – Indico
  – Document Server (Invenio)
  – Askbot (FAQ)
  – ...

• DESY laboratory hosts most of them

Track 8: Today 12:15 - Belle II Collaborative Tools
Data Processing
Belle → Belle II

• **Belle**
  - ~1 PB raw data + MC
  - KEK Computing Centre (KEKCC) as the single analysis centre

• **Belle II**
  - ~130 PB raw data (2 copies) + MC
  - Worldwide Grid usage with user analysis

• *A huge increase in complexity in how the experiment's data is processed and accessed*
Distributed Components

- Human
- Production Manager
- Data Manager
- End User

- Belle II Software Interfaces
- BelleDIRAC
- BelleRawDIRAC

- Production Management
- Fabrication
- Client Tools
- Web Portal
- Monitoring

- B2Raw Data Management

- Distributed Data Management

- VMDIRAC
- WMS
- AMGA
- CVMFS
- DMS
- RMS
- FTS
- LFC

- Grid Services

- Sites, Computing Elements, and Storage Elements

- Cloud Site
- VCYCLE
- Cloud I/F
- Cluster
- DIRAC Slave
- CE

- Local I/O
- Or Remote I/O

- SE Transfers

04/11/2019
CHEP 2019
• Software High Level Trigger (HLT) reconstruction code uses basf2

• Flags of event types in HLT result:
  - hadron, μμ, γγ, ...
  - Used to create calibration subsets offline

• DAQ system writes out Sequential ROOT (SROOT) file format

  Track 1: 5\textsuperscript{th} Nov 11:45 - Initial Belle II HLT Performance
Handed by Belle II Core Computing (B2CC)
B2CC Monitoring

PostgreSQL + Django + ReactJS + pandas + plotly
Offline → Grid: BelleRawDIRAC

Poster - Raw Data Management System

KEKCC Offline Storage → Upload → KEK TMP Grid Storage → Replication → BNL Raw Data

 KEK TMP Grid Storage → Replication → KEK Raw Data

 KEK TMP Grid Storage → Register Replicas → LFC

 KEKCC Offline Storage → Register Metadata → AMGA Metadata Catalogue
Distributed Data Management (DDM)

- A set of DIRAC Agents/Services/DBs implemented in BelleDIRAC
- MC production and Raw data replication utilise it heavily
Distributed Data Management (DDM)

- Uses DIRAC RMS
- Bulk LFC operations
- SE Health + Performance monitoring
- Simple data distribution system
- Investigation of Rucio as a future replacement underway

Track 4: Today 11:30 - Distributed Data Management
Plenary: Today 16:30 - Rucio Outside ATLAS
Dataset Searching

- Searchable database of official datasets for grid usage
- Made in DIRAC DB/Service with a DIRAC webapp on top
  - Faster searching than direct catalogue queries
  - Simply remove datasets that are not suitable for user analysis
  - Initial command line tools to make CRUD operations on database easy
- Download LPNs for usage with gbasf2
Dataset Searching

Poster: Dataset Searching at Belle II
Prompt Processing

- Getting a good first mDST production requires up to date calibration constants
Prompt Processing

- **Detectors are moving towards CAF for production of payloads**
  - Standardising code style + scripts
  - Avoiding custom bash/Python submission scripts

- **cDST common file format**
  - Guided by common needs of all post-tracking calibrations
  - Reduces unnecessary reconstruction
  - Mixture of raw + track information

- **Organisation has been mostly through Jira tasks**
  - Great for communication and notification
Prompt Calibration

- Quite a complex workflow
- It is not strictly enforced → leads to errors
  - Misconfiguration of scripts due to confusion + human error
  - Wrong payloads uploaded to database → delays while we fix it
- Instead we want to:
  - Enforce a strict order and basf2 release
  - Run default CAF scripts (minimal configuration by a human)
  - Sign-off and check payload IoVs for problems
  - Produce cDSTs on a run range
- Leave room for future automation of steps that currently need an expert
Apache Airflow Usage

- A very flexible way of defining workflows of heterogeneous tasks
- Good web monitoring of tasks comes pre-made (Flask)
- Added web based interface for calibration experts to:
  - Submit tasks
  - Sign-off on results
- Background workflows handle continuous collection of information on new data/software releases

Track 2: 7\textsuperscript{th} Nov 14:00 - Prompt Calibration Automation
Conclusion

- We're very busy!
- All the pieces work, but improvements will hopefully free up some manpower
- Data is here...
<table>
<thead>
<tr>
<th>Topic</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating the Belle II Collaborative Services and Tools</td>
<td>Data quality monitors of vertex detectors</td>
</tr>
<tr>
<td>Performance of the High Level Trigger</td>
<td>Prompt calibration automation</td>
</tr>
<tr>
<td>Experience Supporting Conditions DB server Infrastructure</td>
<td>Lepton identification</td>
</tr>
<tr>
<td>Distributed data management</td>
<td>Grid Networking</td>
</tr>
<tr>
<td>Generation of pixel detector background data with a GAN</td>
<td>Selective background Monte Carlo simulation</td>
</tr>
<tr>
<td>User documentation and training</td>
<td>Alignment for the first precision measurements</td>
</tr>
<tr>
<td>Belle2VR: An Interactive Virtual Reality Visualization of GEANT4 Event Histories</td>
<td>Dataset Searching Webapp</td>
</tr>
<tr>
<td>Status of the Belle II simulation library</td>
<td>Raw Data Management System</td>
</tr>
<tr>
<td>Performance of tracking on collision data</td>
<td></td>
</tr>
<tr>
<td>Generative Adversarial Network for Background Generation in the KLM</td>
<td></td>
</tr>
</tbody>
</table>
Backup
Main Distributed Components

- **BelleRawDIRAC**
  - DIRAC extension
  - Performs first upload to grid storage and metadata setting

- **BelleDIRAC**
  - DIRAC extension
  - Production System, DDM, client tools

- **gbasf2**
  - Command line tool to easily submit basf2 steering files to the grid
  - Part of BelleDIRAC repository

- **Catalogues**
  - LFC (files)
  - AMGA (metadata about datasets+files)
Code Development

- Atlassian Bitbucket (Git) + Jira + Bamboo (+ Buildbot)
- Any member can contribute by making a Jira task and opening a Pull Request
  - Unit tests + compilation of branch automatically run
  - Librarian of package must be a reviewer
  - Easy to find blocking tasks for releases and responsible people
- Nightly validation of analysis software
- Software quality shifter checking for commits that caused shifts in validation plots:
  - Event size, reconstruction time, ...
- Sphinx code documentation (Doxygen for C++ classes)
### Pull requests

**Filter by:**  
- **All**  
- David Dossett  
- Target branch  
- I'm reviewing

<table>
<thead>
<tr>
<th>Summary</th>
<th>Reviewers</th>
<th>Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERGED</td>
<td>1</td>
<td>✔️</td>
</tr>
<tr>
<td>David Dossett - #5079, last updated on 17 Oct 2019</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>MERGED</td>
<td>3</td>
<td>✔️</td>
</tr>
<tr>
<td>David Dossett - #5052, last updated on 12 Oct 2019</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>MERGED</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>David Dossett - #4810, last updated on 26 Sep 2019</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>MERGED</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>David Dossett - #4913, last updated on 24 Sep 2019</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>