Preservation of the software dependencies and data provenance at the LHCb experiment

Ana Trisovic
CERN & University of Cambridge
on behalf of the LHCb collaboration
Outline

- The data preservation initiative
- Introduction to the LHCb software stack and data production
- Development and functionality of the database
- Software virtualization
Introduction

The data preservation initiative

- Preservation of:
  - Experimental & simulated data
  - Software & documentation
  - Analysis & publications
- Assist reproducible research
- Preserve the workflows

Agenda

- Building a database to
  - keep the LHCb software metadata
  - document data production
  - record data provenance
- Allow the data processing outside the LHCb infrastructure

Preservation of the software dependencies and data provenance at the LHCb experiment

Ana Trisovic
The LHCb software stack

- Based on Gaudi framework
- Supports:
  - event data processing applications that run in real time high level triggers,
  - the data and Monte Carlo production in the offline system and
  - the physics analyses performed by the users

<table>
<thead>
<tr>
<th>Applications/AppConfig</th>
<th>Component Libraries</th>
<th>Frameworks</th>
<th>LHCbSys (Data_Dictionary, Event_Model, Detector_Description, Conditions_Database)</th>
<th>Gaudi (GaudiPython)</th>
<th>Online Monitoring and Commissioning Lovell (Velo) Orwell (Calo) Panoptes (Rich) Vetra (Velo, ST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauss Simulation DecFiles</td>
<td>Boole Digitization</td>
<td>Alignment</td>
<td>Bender Python analysis</td>
<td>Erasmus Analyses repository</td>
<td>Panoramix Event display</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brunel Reconstruction</td>
<td>Analysis</td>
<td>Stripping</td>
<td>Hlt</td>
</tr>
</tbody>
</table>

Preservation of the software dependencies and data provenance at the LHCb experiment

Ana Trisovic
The LHCb Data and Monte Carlo production

Data production pipeline:

Monte Carlo pipeline:

The names of stages are followed by the names of applications used in the stages. HLT stands for High Level Trigger. Turbo is a new streaming strategy where events are reconstructed in the trigger.
Graph database

Introduction to Neo4j
- Neo4j is an open source NOSQL graph database, implemented in Java
- Robust, scalable, high performance
- Large and active community

Implementation
- Data collected from LHCb TWiki and other sources
- Differences in metadata in run I and run II
- Software changes in time
Database structure

Capturing the software stack dependencies and links to the data

Snapshot from Neo4j web browser

Preservation of the software dependencies and data provenance at the LHCb experiment

Ana Trisovic
Implementation and functionality

- Interface implemented in py2neo
- Transition to neo4j 3

From top to bottom of the LHCb software stack:

```
MATCH p = (a:Application{project:'BENDER'})-[r:REQUIRES*..]-> (d:Framework{project:'GAUDI'})
RETURN p LIMIT 1
```

The code matches a path from an application Bender to Gaudi framework and returns the graph on the right.
Web portal (demo)

- Search engine for processing passes and production metadata
- Recommendation of the latest data sets

Figure: Web interface. Obsolete processing passes are marked in red.
The big picture

Providing the information on data production and the software, and linking the monitoring system Dirac and bookkeeping to the CERN analysis preservation (CAP) portal, users and containerised software in Docker
Graph mining

What can we learn by looking at the graph?

- The most used software versions
- CVMFS management - identifying obsolete software that should be archived
- List of applications that should run with particular subset of the data (eg. collision16)

Use-cases:
- Troubleshooting (eg. identifying the data affected by faulty software component)
Graph visualisation

- Visualisation tool built on top of the database
- Allows the users to explore the graph by themselves
- Implemented with the Collaboration spotting team at CERN: Jean-Marie Le Goff, Adam Agocs, Dimitrios Dardanis
- More about Collaboration spotting
Running the LHCb data production in Docker containers

Docker

- **Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run**
- Actively used at LHCb for nightly builds

Running the LHCb data production on Docker

- Encapsulate the software for future use
- Requirements
  - Docker
  - CVMFS
- Containers that were used are LHCb official at DockerHub

Preservation of the software dependencies and data provenance at the LHCb experiment

Ana Trisovic
Preserving the workflows

• Capturing the data production workflow in a declarative way with the Yadage project and Lukas Heinrich

• Input information for each of the stages are accessible from the db

• On the right are Monte Carlo events created in the workflow

• Yadage is on Github

• Lukas’ talk at CHEP
Summary

- Collecting the metadata about the current system and data production needed for the long term preservation
- Supporting data processing outside the Grid and LHCb infrastructure

Thank you for your attention

For more information about the project and feedback, please contact me by email at ana.trisovic@cern.ch