



Deployment and validation of Git CondDB

M. Clemencic *on behalf of LHCb*

May 16, 2017

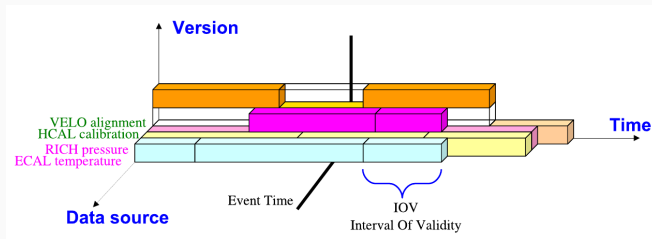
CERN - LHCb

- We have been using COOL + SQLite for 12 years
- Alternative technologies to be investigated ([LHCBPS-1586](#))
- A Git based implementation is being commissioned ([LBOPG-72](#))

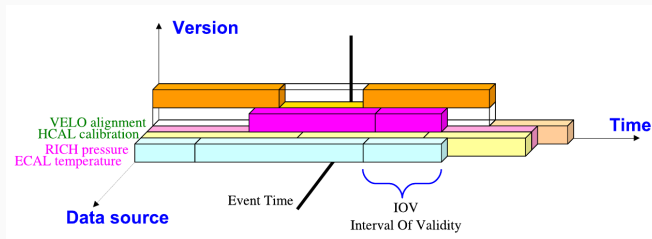
Table of contents

1. A Git CondDB
2. COOL + SQLite vs. Git CondDB
3. Status
4. WIP: Workflows examples

A Git CondDB



- 3 dimensions
 - filesystem-like structure
 - versions
 - time evolution (IOVs)
- SQL backend: Oracle, MySQL, SQLite



- 3 dimensions
 - filesystem-like structure
 - versions
 - time evolution (IOVs)
- SQL backend: Oracle, MySQL, SQLite

- Distributed Version Control System
 - filesystem structure with versions
 - each clone contains all versions
 - tags and branches
 - built-in incremental synchronization
- A local database with 2 out of 3 CondDB dimensions
 - the 3rd dimension (IOVs) can be simulated

GitConddb Layout

with IOVs

no IOV

IOVs indirection for faster lookup

reference to payload

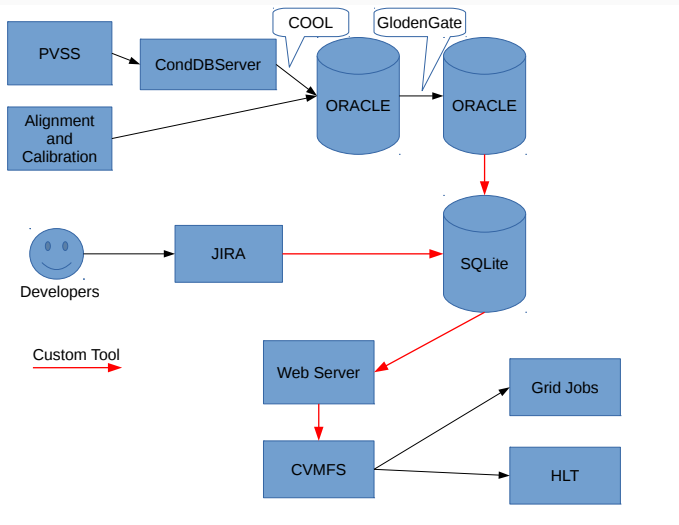
```
MomentumSmear.xml
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE D0DB SYSTEM "git:/DTD/structure.dtd">
3 <D0DB>
4 ...
5 <!--<condition name = "MomentumSmear">
6 ...
7 <!--<param name = "Smear" type = "Histo1D" comment="Smear factors, valid forever/no smearing">
8 <!--<param name=' 'name': 'h1', 'title': 'Smear Factors, (no smearing)',
9 <!--<param name=' 'X' : { 'nbins': 1 , 'low' : 0 , 'high' : 1e+10 },
10 <!--<param name=' 'bins' : [ ( 0 , 0 ) , ( 0 , 0 ) , ( 0 , 0 ) ]
11 <!--</param>
12 ...
13 <!--</condition>

MomentumScale.xml
1 IOVs
2 12938400000000000000 2011-01
3 13253760000000000000 2012-01
4 13569984000000000000 2013-01
5

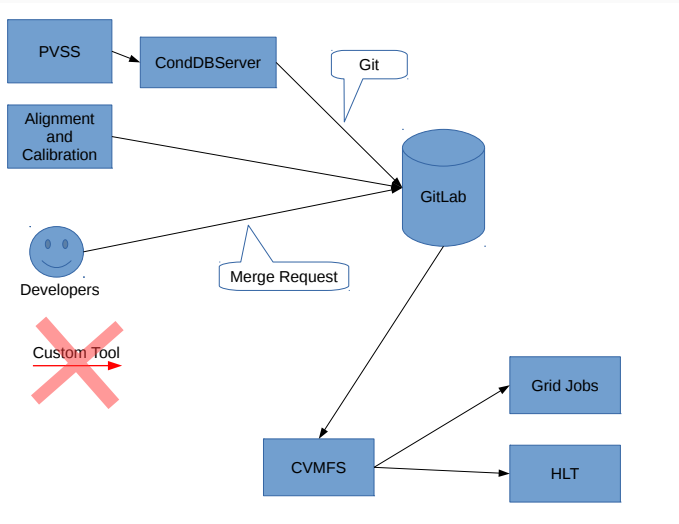
fa4733e60a
1 <?xml version="1.0"?>
2 <!DOCTYPE D0DB SYSTEM "git:/DTD/structure.dtd">
3 <D0DB>
4 <!--<condition name="MomentumScale">
5 <!--<param comment="Global delta scale Reco14 2k+11" type="double" name="Delta">0.0001</param>
6 <!--<param comment="Calibration for IdpPlus Reco14 2k+11" type="Histo2D" name="IdpPlus">{ 'name' :
7 'X' : { 'edges' : [ -0.3 , -0.15 , -0.1 , -0.075 , -0.05 , -0.025 , 0 , 0.025 , 0.05 , 0.075 , 0.1 , 0
8 ...
9 'Y' : { 'edges' : [ -0.25 , -0.1 , 0 , 0.1 , 0.25 ] } ,
10 'bins' : -
11 ...
```


COOL + SQLite vs. Git Conddb

Infrastructure: COOL



Infrastructure: Git



Performance

Speed measurements using LoadDDDB test (includes XML parsing):

COOL only	45.974s	-
DDDB in Git	34.947s	-34%
DDDB + LHCBCOND + DQFLAGS in Git	33.162s	-38%
Git only	33.024s	-38%

Space on Disk (Git includes Upgrade conditions)

	SQLite	Git bare (runtime)	Git checkout (development)
DDDB	34M	2.9M	20M
LHCBCOND	730M	285M	1.9G
ONLINE	2.3G	357M	4.1G
SIMCOND	25M	8.6M	23M
DQFLAGS	399K	128K	436K

Status

- Integrated in LHCb v42r2
 - see [lhcb/LHCb!557](#)
- Git clones of COOL ready for all partitions (except CALIBOFF)
 - `/cvmfs/lhcb.cern.ch/lib/lhcb/git-conddb.test`
- Backward compatibility
 - fallback on SQLite if Git repositories are missing
 - new software will access old tags from Git
 - old software will not see new tags
(backport planned for specific versions)
- Online tools adapted
 - publishing (PVSS and calibrations) to both Git Conddb and COOL
 - MooreOnline and BrunelOnline configurations ready

We need that experts test it (see [LBOPG-72](#))

enable Git CondDB

```
export GITCONDDBPATH=/cvmfs/lhcb.cern.ch/lib/lhcb/git-conddb.test
```

disable Git CondDB

```
export NO_GIT_CONDDB=1
```

More instructions available on [TWiki](#)

Pending tasks can be found in [LHCbps-1709](#)

WIP: Workflows examples

Not much different from any Git-based workflow:

```
git clone ssh://git@gitlab.cern.ch:7999/lhcb-conddb/LHCBCOND.git
```

Edit the files, then test:

```
export CMAKE_PREFIX_PATH=${PWD}:${CMAKE_PREFIX_PATH}
lb-run Project/version gaudirun.py MyOptions.py \
  --option 'from Configurables import CondDB; CondDB().Tags["LHCBCOND"] = ""'
```

Publish the changes:

```
cd LHCBCOND
git commit -a -m 'my super improvements'
git push origin HEAD:super-stuff
```

Create a merge request

Not much different from any Git-based workflow:

```
git clone ssh://git@gitlab.cern.ch:7999/lhcb-conddb/LHCBCOND.git
```

Edit the files, then test:

```
export CMAKE_PREFIX_PATH=${PWD}:${CMAKE_PREFIX_PATH}
lb-run Project/version gaudirun.py MyOptions.py \
  --option 'from Configurables import CondDB; CondDB().Tags["LHCBCOND"] = ""'
```

Publish the changes:

```
cd LHCBCOND
git commit -a -m 'my super improvements'
git push origin HEAD:super-stuff
```

Notes:

- Git CondDBs found automatically

Create a merge request

Not much different from any Git-based workflow:

```
git clone ssh://git@gitlab.cern.ch:7999/lhcb-conddb/LHCBCOND.git
```

Edit the files, then test:

```
export CMAKE_PREFIX_PATH=${PWD}:${CMAKE_PREFIX_PATH}
lb-run Project/version gaudirun.py MyOptions.py \
  --option 'from Configurables import CondDB; CondDB().Tags["LHCBCOND"] = ""'
```

Publish the changes:

```
cd LHCBCOND
git commit -a -m 'my super improvements'
git push origin HEAD:super-stuff
```

Notes:

- Git CondDBs found automatically
- *empty* tag means “use files”

Create a merge request

Modify IOVs

- Files without IOVs are stored as files
- Files with IOVs are stored as directories
 - mandatory file **IOVs**
 - each line is “start-time path-to-payload”
 - path-to-payload may point to a directory containing **IOVs**
 - payloads stored as files
 - no constraint on names or location
 - currently using first part of SHA1 sum as name
 - it’s better if they are in the directory they refer to

- CondDBBrowser support (read-only?)
- Automatic consistency checks (GitLab-CI)
 - monotonic entries in IOVs
 - all referenced payloads actually present
 - no unreferenced payloads
 - no dates in XML comments
 - ...
- Describe how to work with [sparse checkout](#)

Conclusions

Summary

- Git Conddb almost ready for production
 - already used at the PIT (Moore, Brunel)
 - need feedback from detector experts ([LBOPG-72](#))
 - enabled automatically for everybody ASAP
- Missing features for developers to be addressed soon
 - feedback and workflows examples are needed ([LHCBPS-1709](#))