“ObjectCmp”: A tool to extract differences between C++ objects

Sandro Wenzel

based on previous work of

Marian Ivanov, Peter Hristov

ALICE OFFLINE WEEK; 30.03.2017

Disclaimer

- Effort in progress and not complete !
- Based on existing work !
- Talk targeted to get some feedback: Is this useful ? what features do we need ?
- Probably a tool mostly related to QA …

Motivation

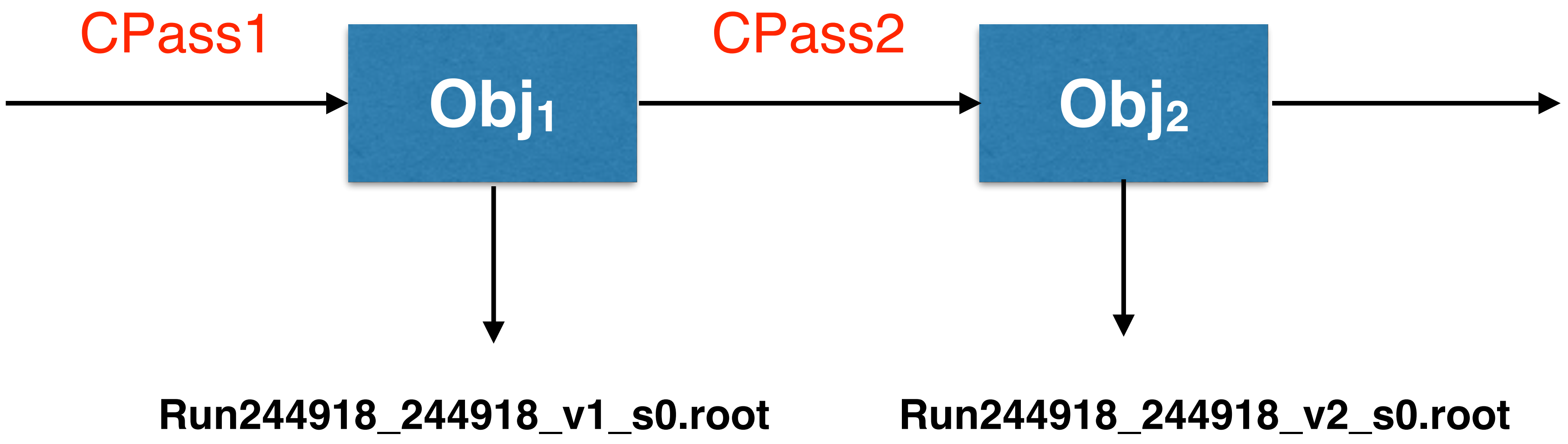
- Most of our data is encapsulated in C++ objects and serialised to disc using ROOT I/O …

- We want to test for bitwise compatibility of objects sitting in different root files …
  - **MUCH** more complicated than comparing the file in binary mode (because of timestamps and other information that ROOT adds)!!

- We want to obtain a measure how much objects in different root files differ…

The OCDB use-case

- OCDB files are written using **objects** deriving from **AliCDBEntry**
  - change often; are used for calibration purposes



CPass1

Obj$_1$

CPass2

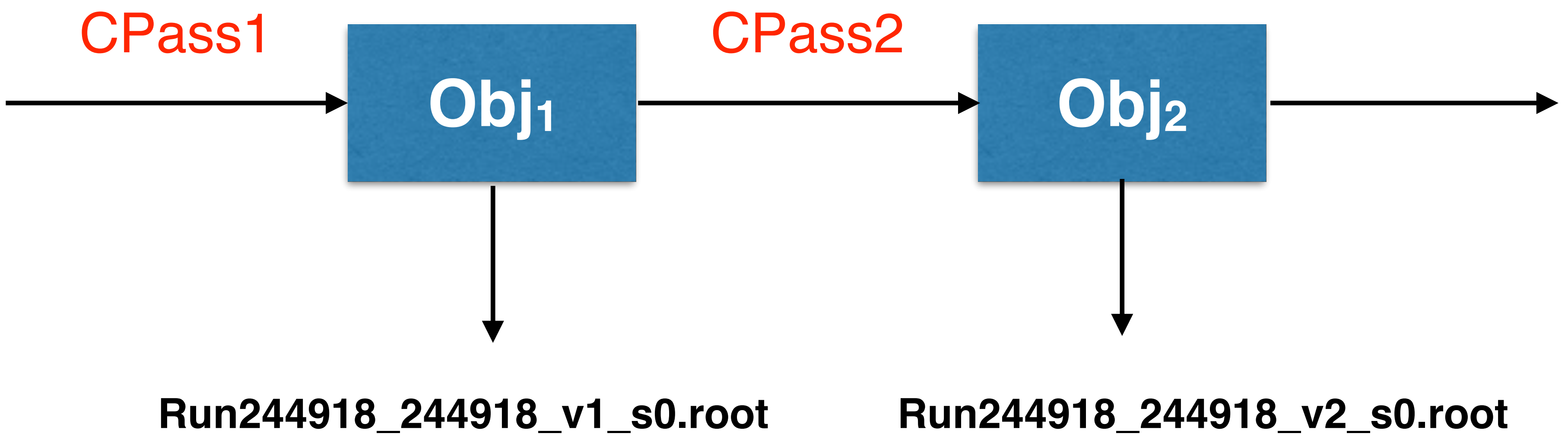
Obj$_2$

Run244918_244918_v1_s0.root

Run244918_244918_v2_s0.root

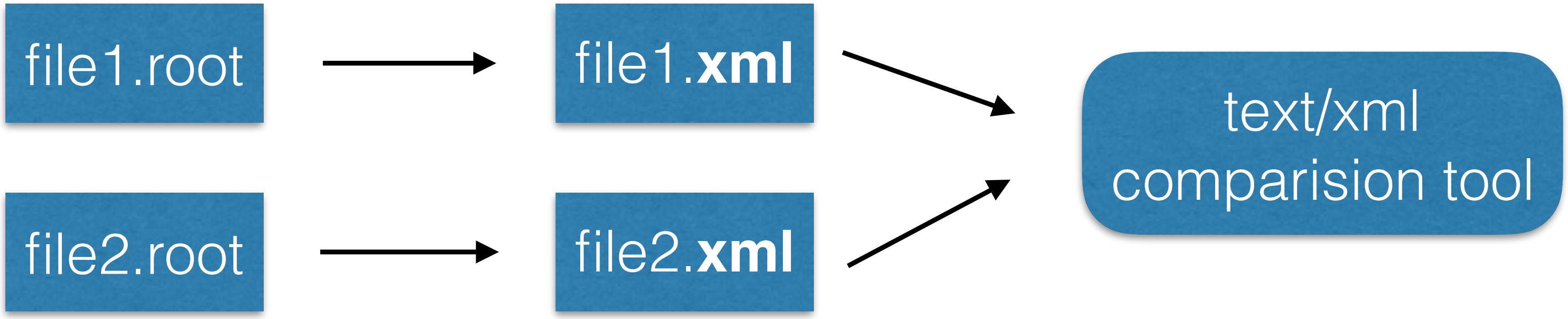
The OCDB use-case

- OCDB files are written using **objects** deriving from **AliCDBEntry**
  - change often; are used for calibration purposes



- **Can we have a generic tool allowing us**
  - to **extract**, **visualize** and **process the differences** between different object versions?
  - to see if **calibration process converges**? $||\mathrm{Obj}_1 - \mathrm{Obj}_2||_{\mathrm{subspace}} \to 0$

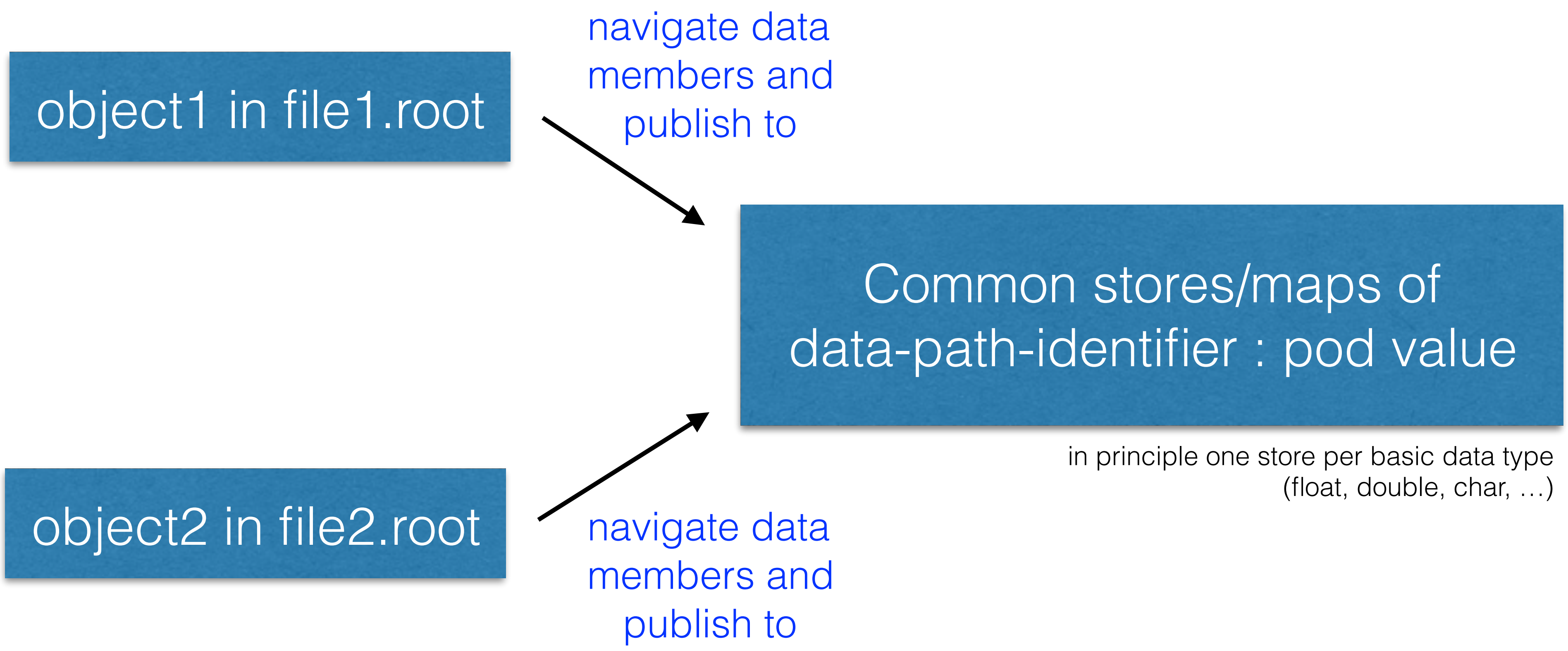
Idea based on text tools



- Pro:
  - simple “diff” might be a good enough in simple cases
- Contra:
  - multi stage process
  - intermediate text output size can be huge
  - still requires external tools (xml parser) to parse and extract differences between 2 xml files
  - text handling is slow
  - loss of precision for double/float values
  - text encoding might be difficult to interpret and diff (for example compression scheme for std::bitset…)

Idea using introspection

- Use **introspection** capabilities of ROOT in order to loop and hierarchically navigate through data members of C++ objects
  - publish encountered (leaf) data to in-memory stores/map(s)
  - use this data-store to easily retrieve differences



object1 in file1.root

navigate data members and publish to

Common stores/maps of data-path-identifier : pod value

in principle one store per basic data type
(float, double, char, …)

object2 in file2.root

navigate data members and publish to

How it looks like : Simple Printing

```cpp
struct Point {
  Point() = default;
  Point(double x, double y) : mX(x), mY(y) {}
  double mX = -110.;
  double mY = 10;
};

struct Track {
  int mId = -1;
  std::vector<Point> mPoints{Point(-1,10), Point(2,10)};
};
```

```
"PrintObject ../test/Track.root Track"

Track.mId{int} : -1
Track.mPoints{vector<Point>}[0].mX{double} : -1
Track.mPoints{vector<Point>}[0].mY{double} : 10
Track.mPoints{vector<Point>}[1].mX{double} : 2
Track.mPoints{vector<Point>}[1].mY{double} : 10
```

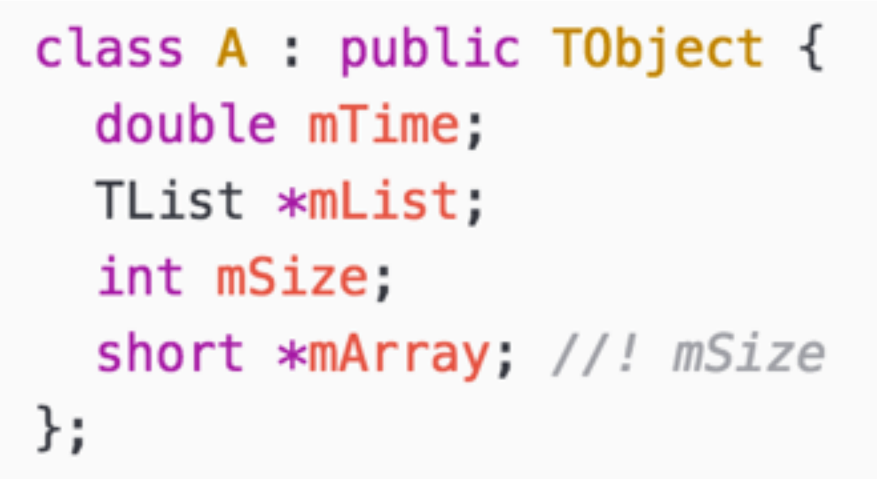
unique data identifier ↗

↖ pod value

Developments done

https://gitlab.cern.ch/swenzel/ObjectCmp/

- Got some code from Marian/Peter which could already navigate and print C++ objects
  - for C++ classes deriving from TObject
  - for basic datatypes
  - ROOT collections (TList, …)

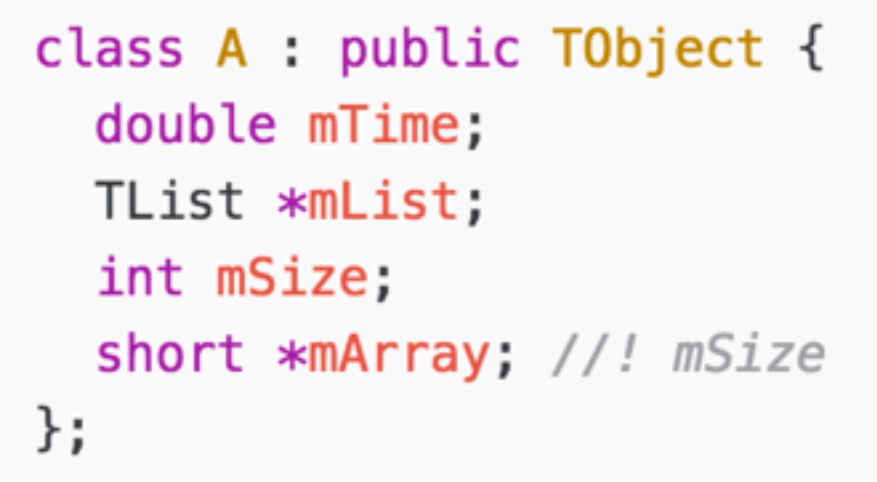


```
class A : public TObject {
  double mTime;
  TList *mList;
  int mSize;
  short *mArray; //! mSize
};
```

Developments done

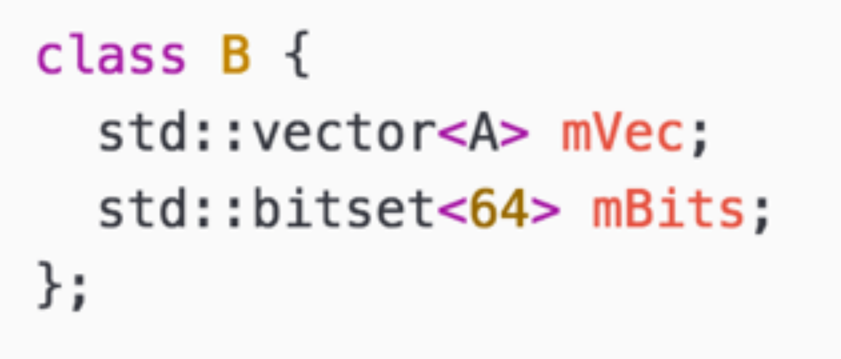
https://gitlab.cern.ch/swenzel/ObjectCmp/

- Got some code from Marian/Peter which could already navigate and print C++ objects
  - for C++ classes deriving from TObject
  - for basic datatypes
  - ROOT collections (TList, …)



```
class A : public TObject {
  double mTime;
  TList *mList;
  int mSize;
  short *mArray; //! mSize
};
```

- Extended it
  - to no longer require TObject inheritance
  - to handle stl containers (vector, maps,…)
  - with the logic to compare 2 objects in memory



```
class B {
  std::vector<A> mVec;
  std::bitset<64> mBits;
};
```

OCDB diff

```
.CDBCompare Run244918_244918_v2_s0.root Run244918_244918_v3_s0.root
```

(/alice/data/2015/OCDB/TOF/Calib/RunParams/)

```
DIFFERENCE FOR KEY AliCDBEntry.fId{AliCDBId}.fVersion{int} ABSOLUTE -1

//…

DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0{float*}[0] ABSOLUTE -7.72662
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0{float*}[1] ABSOLUTE -3.46628
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0{float*}[2] ABSOLUTE -2.82877

//…

DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fTOFResolution{float*}[0] ABSOLUTE 26.0465
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fTOFResolution{float*}[1] ABSOLUTE 30.7094
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fTOFResolution{float*}[2] ABSOLUTE 26.1067

//…

DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0Spread{float*}[19] ABSOLUTE -0.0553589
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0Spread{float*}[20] ABSOLUTE 1.43106
DIFFERENCE FOR KEY AliCDBEntry.fObject{AliTOFRunParams}.fT0Spread{float*}[21] ABSOLUTE 1.00253

//…
```

Limitations + Plans

- We need to have the dictionary library in order to compare the object
  - This is in principle not needed … in contact with the ROOT team to get rid of this restriction
  - In future it might be possible to compare objects from root files without having the dictionary
- (Probably) no support for TTree/TBranches yet …
- Need application layer to process differences …
  - somewhat use-case specific …
  - might be best done using python which can query a “json-diff” using functional techniques