QA tools overview

Jacek Otwinowski
(for DPG QA tools)

Outline

- Roadmap and timelines (Run2)
- QA data storage
- Software release validation
- Offline QA
- Online QA (see Raymond talk)
- Service tasks
- Outlook

Goal:
Develop common set of tools for the software validation, online and offline QA for Run2 and Run3.

Roadmap and timelines (Run2)

- Massive tests of tools (Feb 2017)
  - Elasticsearch + Swan
  - ROOT tree based DB + Swan
  - Overwatch + Elasticsearch
- Release validation (April 2017)
  - Include MC information
  - Based on trending information
- Offline QA (April 2017)
  - Based on trending information
- Online QA (Sep 2017)
  - Based on trending information
  - Trigger alarms

QA data storage

Matteo Concas and Dario Berzano

- QA output files stored on AFS and EOS file systems
- QA offline data regularly pushed to AFS using rsync
- EOS and AFS synchronized based on rsync
  - Github: mconcas/qa-sync/blob/master/qa-sync.sh
  - Backward compatibility issue

    http://aliqamod.web.cern.ch/aliqamod/data/2016/LHC16t/logbook.root (old)

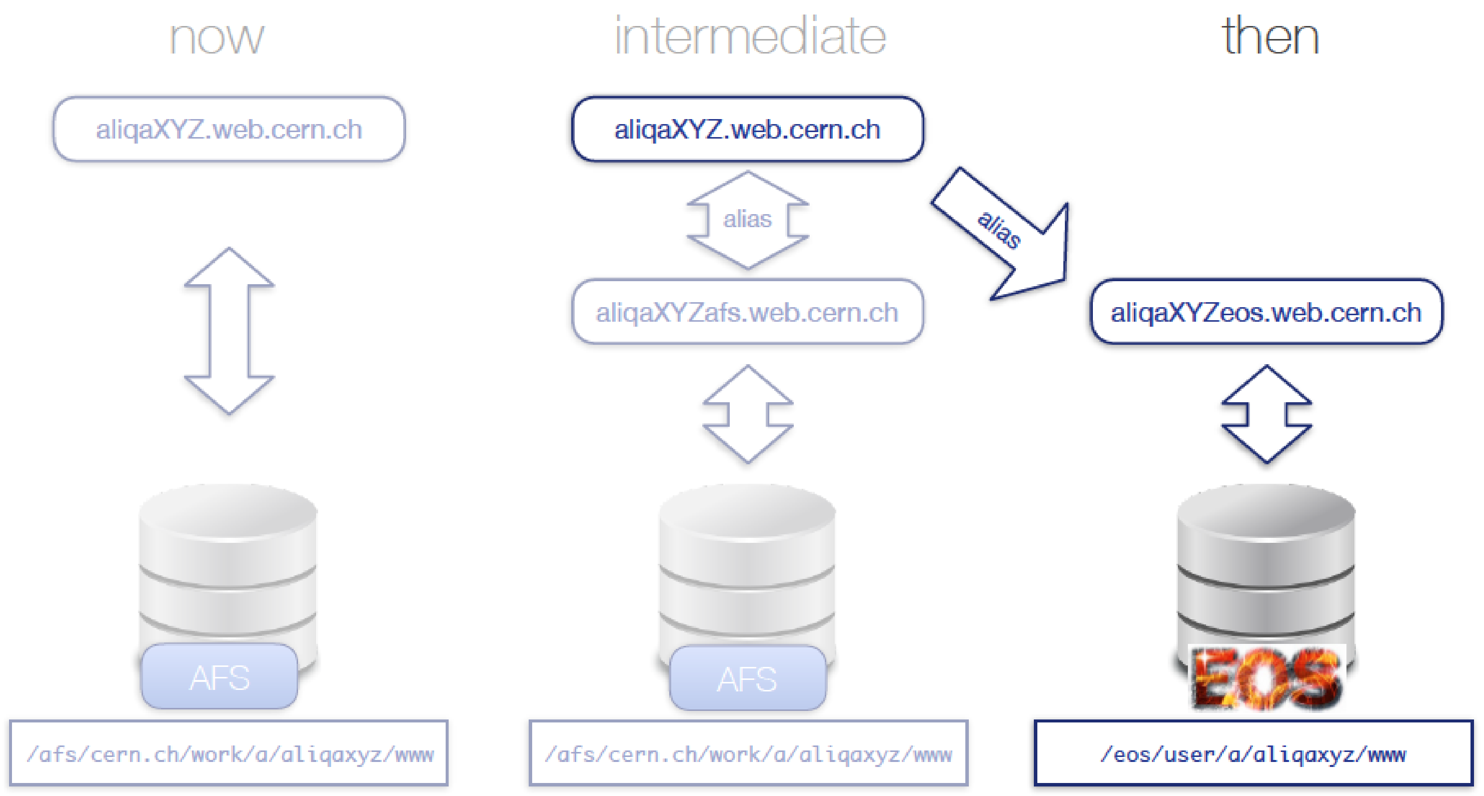
    http://aliqamod.web.cern.ch/data/2016/LHC16t/logbook.root (new)
  - Quota on #inodes increased to 10^7 (file archives should be considered)
- EOS will be the only storage after tests
- EOS for QA files from software validation and online QA

https://indico.cern.ch/event/608363

Migration from AFS to EOS

Matteo Concas and Dario Berzano

Websites migration - Schema



Software release validation

- Based on reference data and MC
  - MC to be implemented
  - Full processing chain
  - Data: Calibration, reconstruction, QA
  - MC: Event generation, Geant3 transport, reconstruction, QA
- Run for each ALICE software release
  - Preparation for continues builds

Software release validation on MC

Matteo Concas, Jihyun Bhom, DPG and offline core

- Setup performance generators (reference MC)
  - Pythia jets from user defined parton distributions implemented (Marian Ivanov)
  - Heavy-flavor signal enhancement to be considered
- Run full processing chain
  - Minimum configuration based on JDL scripts
- QA output
  - Completeness checks
  - RAM and CPU
  - Detector and tracking QA
- Data sets comparison based on trending information
  - Regression tests
  - Triggering alarms

https://indico.cern.ch/event/593667
https://indico.cern.ch/event/617942

Offline QA

Marian Ivanov, Hans Beck
Iwona Sputowska, Jacek Otwinowski

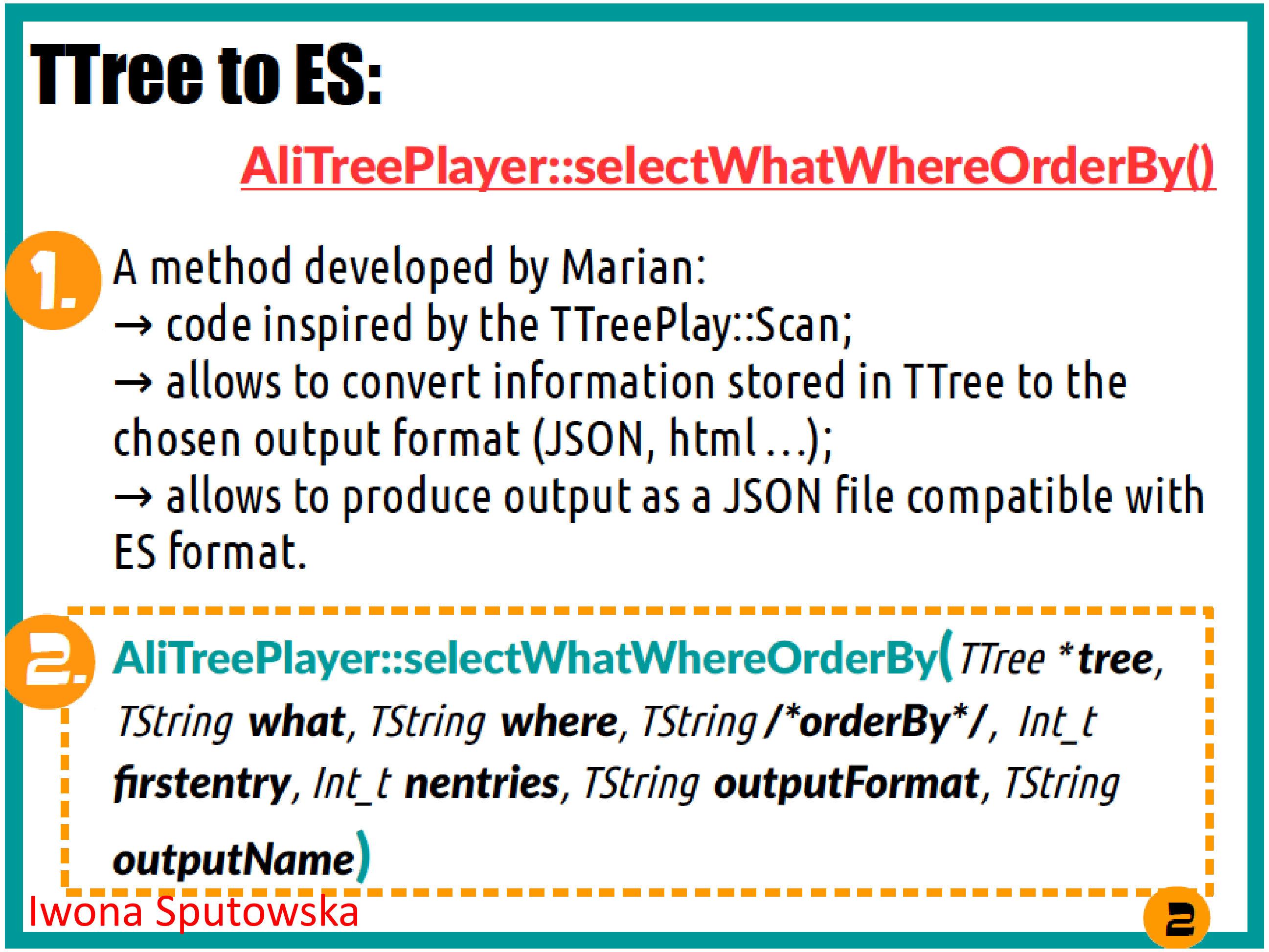
- Detector, trigger, calibration, tracking, PID and analysis QA
- QA tools developments
  - QA based on Elasticsearch (https://www.elastic.co)
  - ROOT tree based DB (TPC QA generalization)
- To run on local cluster and grid systems

QA based on Elasticsearch

- QA trending information stored in ROOT trees (trending.root)
  - Numbers, vectors, graphs
- Data/reconstruction flow
  - trending.root → JSON → Elasticsearch → ROOT → SWAN
- ROOT → JSON conversion implemented (Marian Ivanov)
- Elasticsearch query
  - Python client tested
  - C++ client to be tested
  - Interface based on TFormula considered
  - Massive tests ongoing
- Parent-child relationships in Elasticsearch
  - Example tested based on Elasticsearch 5.2

    https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-parent-field.html
  - To be tested on ALICE cases

ROOT → JSON conversion



TTree to ES:

AliTreePlayer::selectWhatWhereOrderBy()

1. A method developed by Marian:
→ code inspired by the TTreePlay::Scan;
→ allows to convert information stored in TTree to the chosen output format (JSON, html…);
→ allows to produce output as a JSON file compatible with ES format.

2. AliTreePlayer::selectWhatWhereOrderBy(*TTree* ***tree**, *TString* **what**, *TString* **where**, *TString* **/*orderBy*/**, *Int_t* **firstentry**, *Int_t* **nentries**, *TString* **outputFormat**, *TString* **outputName**)

2

Iwona Sputowska

Conversion possible for all ROOT objects.

https://indico.cern.ch/event/612476
https://indico.cern.ch/event/614413

Elasticsearch query – Python Client

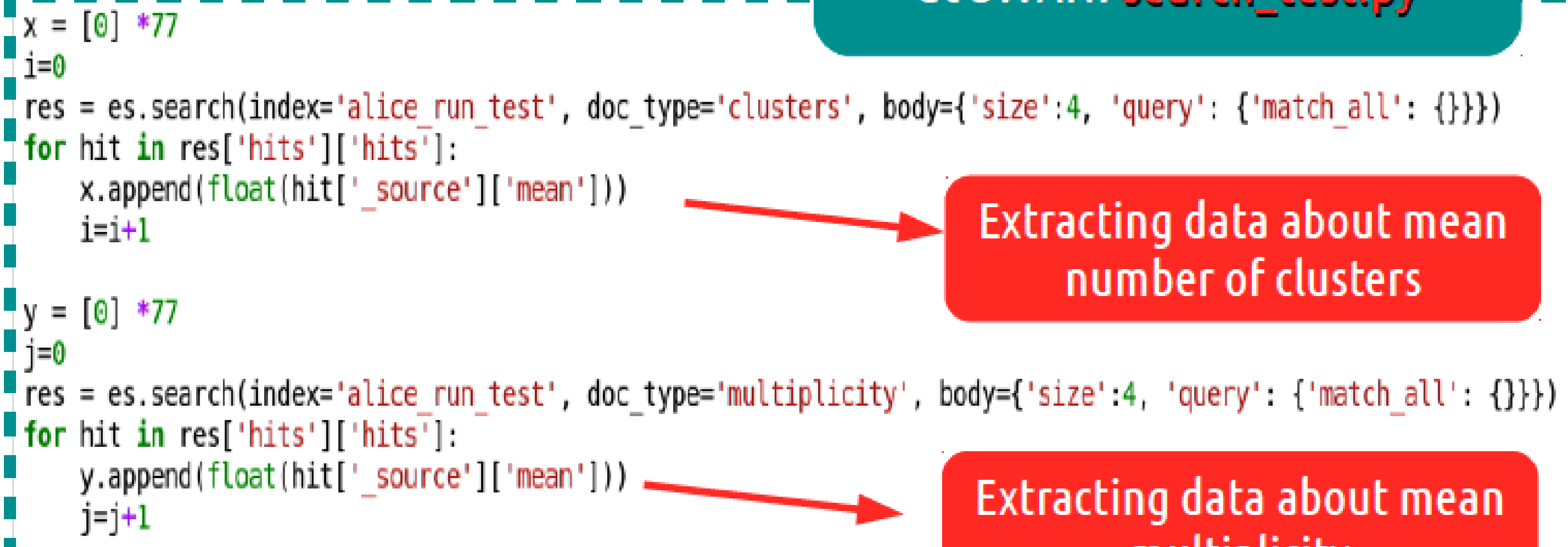
Iwona Sputowska

https://indico.cern.ch/event/589855

Phyton + Elasticsearch → elasticsearch-py package:

```
from elasticsearch import Elasticsearch
```

Fragment of the script running at SWAN: search_test.py



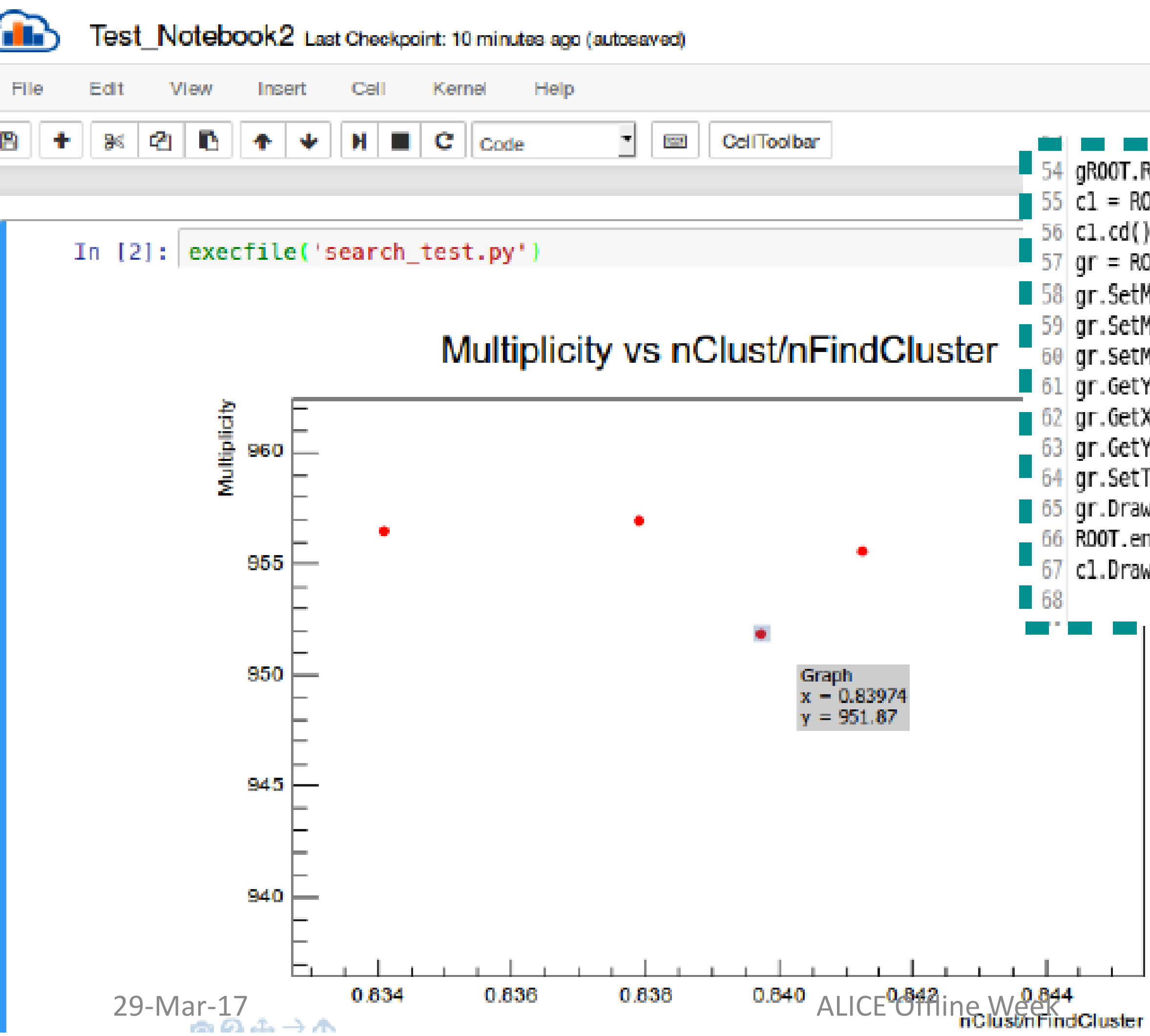
```
x = [0] *77
i=0
res = es.search(index='alice_run_test', doc_type='clusters', body={'size':4, 'query': {'match_all': {}}})
for hit in res['hits']['hits']:
    x.append(float(hit['_source']['mean']))
    i=i+1

y = [0] *77
j=0
res = es.search(index='alice_run_test', doc_type='multiplicity', body={'size':4, 'query': {'match_all': {}}})
for hit in res['hits']['hits']:
    y.append(float(hit['_source']['mean']))
    j=j+1
```

Extracting data about mean number of clusters

Extracting data about mean multiplicity

Elasticsearch + SWAN

The visualization of data using Jupiter Notebook in SWAN



Test_Notebook2 Last Checkpoint: 10 minutes ago (autosaved)
Control Panel
Logout
File Edit View Insert Cell Kernel Help
Python 2
Code
CellToolbar
Multiplicity vs nClust/nFindCluster
Multiplicity
nClust/nFindCluster
Graph
x = 0.83974
y = 951.87

```
In [2]: execfile('search_test.py')
```

```
gROOT.Reset()
c1 = ROOT.TCanvas( 'c1', 'Example with Formula', 200, 10, 700, 500 )
c1.cd()
gr = ROOT.TGraph(len(x),array('d', x),array('d', y))
gr.SetMarkerSize(0.7)
gr.SetMarkerStyle(8)
gr.SetMarkerColor(2)
gr.GetYaxis().SetTitle("Multiplicity")
gr.GetXaxis().SetTitle("nClust/nFindCluster")
gr.GetYaxis().SetTitleOffset(1.5)
gr.SetTitle("Multiplicity vs nClust/nFindCluster")
gr.Draw("AP")
ROOT.enableJSVis()
c1.Draw()
```

Iwona Sputowska

https://indico.cern.ch/event/589855
https://indico.cern.ch/event/617701

Demonstration: copy-paste
https://indico.cern.ch/event/617701/:2176118

ROOT tree based DB (TPC QA generalization)

- Input information stored in the ROOT files
  - Relations (e.g. TPC QA vs. TRD QA) made with friend trees
  - Inner joins (SQL) to combine columns from one or more tables in a rational database implemented
- Access to the external information provided (RCT, Logbook,…)
  - https://alice.its.cern.ch/jira/browse/ATO-46
  - Standardized cross queries possible (e.g. TPC QA vs. RCT)
- Time series query support for tree using AliTreePlayer
  - https://alice.its.cern.ch/jira/browse/ATO-382
- Data selection (acceptance cuts) based on the custom parameterizations (AliNDLocalRegression)
  - https://alice.its.cern.ch/jira/browse/PWGPP-163

See Marian talk

Service task 1

Task: Development of a trending and alarm framework for online HLT QA

Several QA components on the HLT provide QA data for different detectors at discrete times as simple 1 or multi-dimensional histogram. These data are visualized using the overwatch web application. The scope of the task is to implement a component processing the histograms and extracting trending information. These trending information will be sent to an elasticsearch database for visualization. In addition, by comparing to limits defined by users, automatic alarms should be raised (E-Mail to detector responsible with alarm message + Log in database). The alarm handling should be in testing phase during the pp data taking period end of 2017.

Service task 2

Task: Development and tests of tools for data sets comparison based on performance parameterization maps.

Tracking performance observed in MC simulations is different from that for real data. In some regions of phase space (phi, eta, q/pt) the discrepancies are larger than intrinsic errors and should be removed from the analysis. Each data set collected by ALICE should be validated by comparing to anchor MC based on performance parameterization maps (PWGPP-163). In order to perform such comparison for updated reconstruction algorithms the input data (filtered raw data) should be prepared for each ALICE data set.

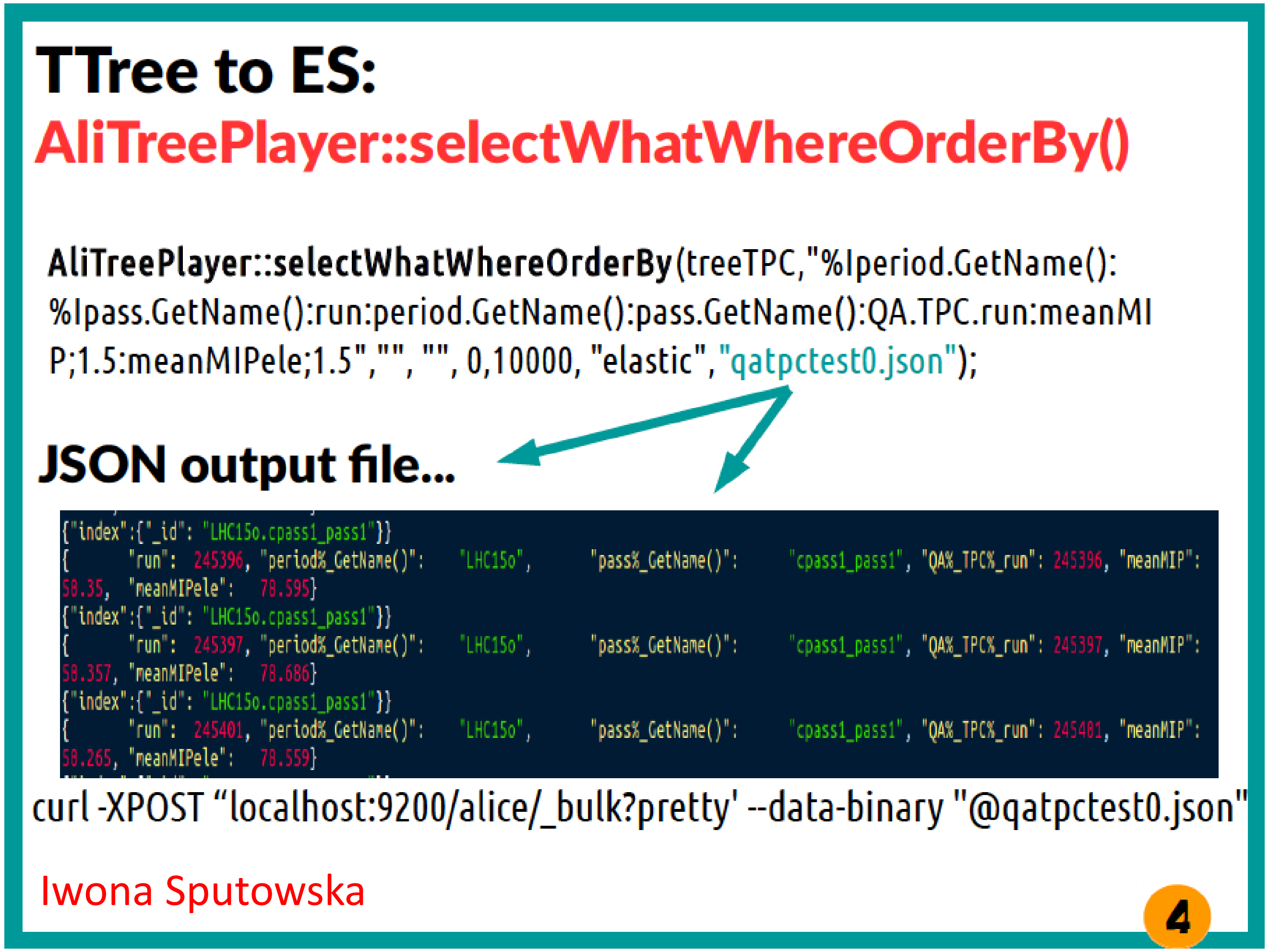
Outlook

- Development is ongoing on several fronts
- Requirements for Run3 are also discussed
  - https://indico.cern.ch/event/591882/
- Connection to JSROOT developers and CERN monitoring/Elastic stack teams established (join meetings)
- Regular meetings (Wednesdays 2:30 PM)
- Email list alice-dpg-qa-tools@cern.ch
- Dedicated JIRA project (ADQT)

Please join and participate in developments!

Backup

ROOT → JSON conversion



TTree to ES:
AliTreePlayer::selectWhatWhereOrderBy()

```
AliTreePlayer::selectWhatWhereOrderBy(treeTPC,"%Iperiod.GetName():
%Ipass.GetName():run:period.GetName():pass.GetName():QA.TPC.run:meanMI
P;1.5:meanMIPele;1.5","", "", 0,10000, "elastic","qatpctest0.json");
```

JSON output file...

```
{"index":{"_id": "LHC15o.cpass1_pass1"}}
{       "run":  245396, "period%_GetName()":    "LHC15o",       "pass%_GetName()":      "cpass1_pass1", "QA%_TPC%_run": 245396, "meanMIP":
58.35,  "meanMIPele":   78.595}
{"index":{"_id": "LHC15o.cpass1_pass1"}}
{       "run":  245397, "period%_GetName()":    "LHC15o",       "pass%_GetName()":      "cpass1_pass1", "QA%_TPC%_run": 245397, "meanMIP":
58.357, "meanMIPele":   78.686}
{"index":{"_id": "LHC15o.cpass1_pass1"}}
{       "run":  245401, "period%_GetName()":    "LHC15o",       "pass%_GetName()":      "cpass1_pass1", "QA%_TPC%_run": 245401, "meanMIP":
58.265, "meanMIPele":   78.559}
```

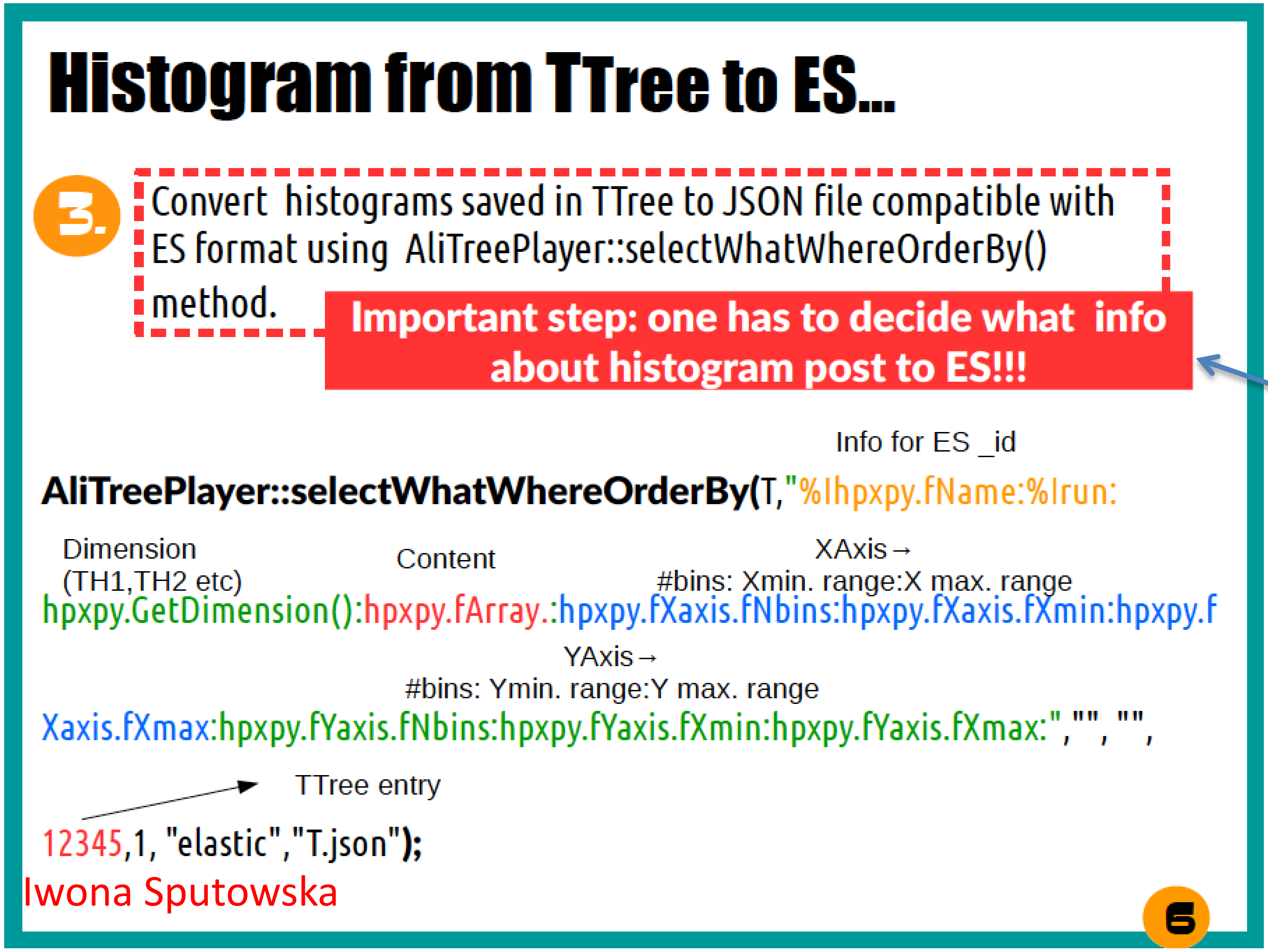
```
curl -XPOST “localhost:9200/alice/_bulk?pretty' --data-binary "@qatpctest0.json"
```

Iwona Sputowska

4

https://indico.cern.ch/event/614413

ROOT → JSON conversion (histograms)



Histogram from TTree to ES...

3. Convert histograms saved in TTree to JSON file compatible with ES format using AliTreePlayer::selectWhatWhereOrderBy() method.

Important step: one has to decide what info about histogram post to ES!!!

Info for ES _id

```
AliTreePlayer::selectWhatWhereOrderBy(T,"%Ihpxpy.fName:%Irun:
hpxpy.GetDimension():hpxpy.fArray.:hpxpy.fXaxis.fNbins:hpxpy.fXaxis.fXmin:hpxpy.f
Xaxis.fXmax:hpxpy.fYaxis.fNbins:hpxpy.fYaxis.fXmin:hpxpy.fYaxis.fXmax:","", "",
12345,1, "elastic","T.json");
```

Dimension (TH1,TH2 etc)
Content
XAxis→ #bins: Xmin. range:X max. range
YAxis→ #bins: Ymin. range:Y max. range
TTree entry
6

Iwona Sputowska

Extend trending.root info with QA histograms

What JSON document content?

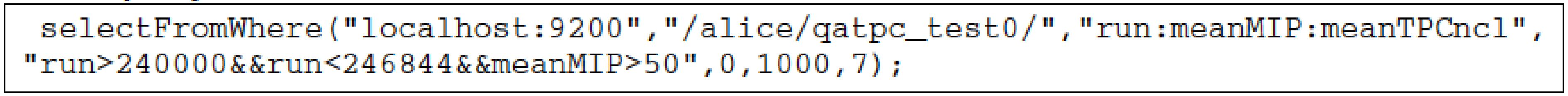
https://indico.cern.ch/event/614413

Elasticsearch query

- Official Elasticsearch Python Client
  - https://elasticsearch-py.readthedocs.io/en/master
- Non official Elasticsearch c++ Client (to be tested)
  - https://github.com/QHedgeTech/cpp-elasticsearch/tree/master/src
  - To be installed in AliRoot
- C++ querry based on TFormula (to be implemented)

Iwona

Marian



```
 selectFromWhere("localhost:9200","/alice/qatpc_test0/","run:meanMIP:meanTPCncl",
"run>240000&&run<246844&&meanMIP>50",0,1000,7);
```

https://indico.cern.ch/event/612476