Data Mining as a Service with ROOT

E. Tejedor, D. Piparo, P. Mató
on behalf of the ROOT team – EP-SFT
L. Mascetti, J. Moscicki, M. Lamanna – IT-ST
ACAT 2016. UTFSM, Valparaíso (Chile)
18/01/2016
• The DMaaS project

• Integration of ROOT with the Notebook technology
  – ROOTbooks for data analysis

• A new service: ROOT & Jupyter Notebooks within the CERN IT services’ portfolio
  – Spotlight on storage: EOS, CERNBox
Introduction:
The “Notebook”
Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: \[
\sum_{n=-\infty}^{\infty} |x(n)|^2
\]
Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: \[ \sum_{n=-\infty}^{\infty} |x(n)|^2 \]

In [1]: def thisFunction():
    return 42
Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: \[ \sum_{n=-\infty}^{\infty} |x(n)|^2 \]

In [1]: def thisFunction():
   
   return 42

This is a notebook in Python
Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code: \[ \sum_{n=-\infty}^{\infty} |x(n)|^2 \]

In [1]: def thisFunction():
    return 42

In [2]: thisFunction()

Out[2]: 42
We can invoke commands in the shell...

```bash
def thisFunction():
    return 42

thisFunction()
```

```
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg > SF.jpg
```
In [1]: def thisFunction():
    return 42

In [2]: thisFunction()
Out[2]: 42

In [3]: ```bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
> SF.jpg
```

<table>
<thead>
<tr>
<th>% Total</th>
<th>% Received</th>
<th>% Xferd</th>
<th>Average Speed</th>
<th>Time</th>
<th>Time</th>
<th>Current</th>
<th>Dload</th>
<th>Upload</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spent</td>
<td>Left</td>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>128k</td>
<td>100</td>
<td>128k</td>
<td>0</td>
<td>0</td>
<td>2731k</td>
<td>0</td>
<td>--:--:--</td>
<td>--:--:--</td>
</tr>
<tr>
<td>--:--:--</td>
<td>--:--:--</td>
<td>--:--:--</td>
<td>2787k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In [1]: def thisFunction():
   return 42

In [2]: thisFunction()

Out[2]: 42

In [3]: 
%%bash
    curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
    > SF.jpg

<table>
<thead>
<tr>
<th>% Total</th>
<th>% Received</th>
<th>% Xferd</th>
<th>Average Speed</th>
<th>Time</th>
<th>Time</th>
<th>Current</th>
<th>Dload</th>
<th>Upload</th>
<th>Total</th>
</tr>
</thead>
</table>
| 100    | 128k       | 100     | 128k          | 0            | 0    | 2731k   | 0     | --:--:-| --:--:
|        |            |         |               |              |      |         |       |        | 2787k |

In [4]: from IPython.display import Image

Image(filename="./SF.jpg",width=225)
In [1]:
   def thisFunction():
       return 42

In [2]: thisFunction()

Out[2]: 42

In [3]:
   %bash
   curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
   > SF.jpg

   % Total  % Received % Xferd  Average Speed   Time    Time  Spent  Left  Speed
   dload  upload   total
   100   128k  100  128k   0   0 2731k  0  --:--:--  128k  0  2787k

In [4]: from IPython.display import Image
   Image(filename="./SF.jpg",width=225)

Out[4]:
In [1]:

In [2]: 42

In [3]:
```bash
curl rootaasdemo.web.cern.ch/rootaasdemo/SaasFee.jpg \
   > SF.jpg
```

```
% Total  % Received % Xferdé Time  Time  Current   Speed
     0     0     0     0     0     0    0.00kB/s   0.00kB/s
Spent   Left  Speed
100 128k 100 128k 0 0 2731k 0 --:--:-- --:--:-- --:--:-- 2787k
```

In [4]:
```python
from IPython.display import Image
Image(filename="./SF.jpg",width=225)
```

Out[4]:

![Image](SF.jpg)
Data mining with ROOT “as a service”

*Interface:* Jupyter Notebooks

**Goals:**

- **Use** ROOT only with a web browser
  - Platform independent ROOT-based data analysis
  - Calculations, input and results “in the cloud”
- **Allow easy sharing of scientific results:** plots, data, code
  - Storage is crucial
- **Simplify teaching** of data processing and programming
- **Potential integration with other analysis ecosystems:** R, Python, …
Integration of ROOT with Notebooks

ROOT

JupyROOT
(ROOT-Jupyter integration)
A Choice of Kernels

Select items to perform actions on them.

- PresentationNotebooks
- cernbox
- HowTo_ROOT-Notebooks.ipynb
- HowTo_ROOT-Notebooks_Long.ipynb
- My First Notebook.ipynb
- Untitled.ipynb
In [1]: import ROOT # This triggers the integration layer

Welcome to JupyROOT 6.07/03
In [1]: import ROOT  # This triggers the integration layer

Welcome to JupyROOT 6.07/03

In [ ]:

```cpp
auto myHisto = TH1;
```

C++ Cells in Python Notebooks

ROOT Tab Completion
In [1]: import ROOT # This triggers the integration layer

Welcome to JupyROOT 6.07/03

In [2]: %cpp
   auto myHisto = TH1F("h","MyData:X;Y",64,-4,4); // C++11
In [1]: import ROOT # This triggers the integration layer

Welcome to JupyROOT 6.07/03

In [2]: %%cpp
   auto myHisto = TH1F("h","MyData;X;Y",64,-4,4); // C++11

In [3]: h = ROOT.myHisto # Find the variable back in Python!
h.FillRandom("gaus")
c = ROOT.TCanvas()
h.Draw()
c.Draw()
Seamless display of graphics
In [4]: ```cpp
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
In [4]:

```cpp
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

In [5]:

```python
f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```
In [4]:

```cpp
double myG(double* x, double* par) {
    auto res = (x[0] - par[1]) / par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

In [5]:

```cpp
f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S")  # Capture printouts
```

```
FCN=47.4997 FROM MIGRAD STATUS=CONVERGED 69 CALLS 70 TO TAL
EDM=2.04372e-09 STRATEGY= 1 ERROR MATRIX ACC
```

```
EXT PARAMETER             STEP       FIRST
 NO.  NAME   VALUE  ERROR   SIZE      DERIVATIVE
1 N    2.46469e+02 4.31493e+00 1.19092e+02  -5.38026e-06
2 mu   1.04793e-02 1.43576e-02 4.87640e-05   4.15093e-03
3 sigma 1.00316e+00 1.03818e-02 2.86307e-05  -2.55310e-04
```
In [6]: ROOT.enableJSVvis()  # Not active by default yet!
c.Draw()
ROOT.disableJSVvis()

My Data

<table>
<thead>
<tr>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entries 5000</td>
</tr>
<tr>
<td>Mean 0.008500</td>
</tr>
<tr>
<td>RMS 1.014</td>
</tr>
</tbody>
</table>

bin = 38
x = [0.6250, 0.7500]
entries = 198
```cpp
In [1]:
TCanvas c1("c1","c1",600,400);
TH2F hcontz("hcontz","Option CONTZ example ",40,-4,4,40,-20,20);
Float_t px, py;
for (Int_t i = 0; i < 25000; i++) {
    gRandom->Rannor(px,py);
    hcontz.Fill(px-1.5*py);
    hcontz.Fill(2+0.5*px,2*py-10.,0.1);
}
hcontz.Draw("CONTZ");
c1.Draw();
```
```cpp
hcontz.Draw("CONTZ");
c1.Draw();
```
Try It Out! – ROOT Binder

Anonymous access

http://mybinder.org/repo/cernphsft/rootbinder

ROOT is a framework for data processing, born at CERN, at the heart of high-energy physics research. Every day, thousands of physicists use ROOT applications to analyze petabytes of data or to perform simulations.

Try a ROOTbook now: choose your favourite language!

Python

C++

More ROOTBooks!

3D Geometries Visualization

Modelling and Fitting: Higgs

View, Create and Run ROOTbooks!
Follow some simple instructions in:
https://root.cern.ch/how/how-create-rootbook
and…

$ root --notebook

This command:
1. Starts a local notebook server
2. Connects to it via the browser

Provides a ROOT C++ kernel and the rest of ROOTbook goodies
• ROOTbooks How-Tos
  https://root.cern.ch/howtos#Jupyter%20Notebooks

• ROOT bindings for Jupyter

• ROOT C++ Kernel
  https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages

• Examples (15 already) from the new ROOT Tutorials can be found at:
  https://root.cern.ch/code-examples#notebooks

  both in Python and C++ (and mixed!)
DMaaS and CERN services’ Portfolio
Integration With CERN’s Ecosystem

- Centrally provide ROOT as a Service
- Authentication with CERN credentials
- Infrastructure: virtual machines in OpenStack Cloud
- Software distribution: CVMFS
- Storage access: CERNBox, EOS
  - All data potentially available!
- Synergy with document sharing
  - Notebook visualiser already available in Indico
Pilot Service (Distributed)

- **CERN Auth**
- **Web Portal (jupyterhub)**
- **Container Scheduler**
- **CERN Cloud**

**Notebook Container**
- **jupyter**
- **docker**

**ROOT**
- Data Analysis Framework

**CERNBox**

**CernVM File system**

**EOS**
Select items to perform actions on them.

- [ ] ..
- [ ] tutorials
- [ ] HowTo_ROOT-Notebooks.ipynb
- [ ] My First Notebook.ipynb
CERNBox: Browse

Files ▼ Help & Download Clients

All files

Favorites

Shared with you

Shared with others

Shared by link

Deleted files

New ▼

Name ▲ Size Modified

- .ipynb_checkpoints 0 kB 2 minutes ago
- My First Notebook.ipynb 0 kB a few seconds ago
- tutorials 0 kB 7 days ago
- HowTo_ROOT-Notebooks.ipynb 214 kB 7 days ago
- My First Notebook.ipynb 24 kB 2 minutes ago
- myOutputFile.root 6 kB 3 minutes ago
- myPlot.pdf 14 kB 3 minutes ago
- Define a custom environment via a web form:

LCG release
- 82 ROOTaaS6

Platform
- x86_64 SLC6 GCC4.9 Opt

Spawn

N releases

CernVM File system

~150 packages
• CERN Summer Student Program, ROOT tutorial: Interactive notebooks offered
  – 50 participants, perfect scaling, a success!
  https://indico.cern.ch/event/407519

• Data Science @ LHC Workshop, Multivariate analysis tutorial
  http://indico.cern.ch/event/395374/

• E-Planet exchange @ UERJ, Brazil
  – 30 participants, every day for a week, 3h a day
  https://indico.cern.ch/event/402660/
• ROOT is now integrated with notebooks
  – Python and C++ interactive shells
  – Tab completion, C++/Python integration, syntax highlighting, graphics inlining
  – Available now!
  – Future: leverage ROOT-R integration, TMVA, …

• Integration with the CERN services portfolio
  – Collaborating with IT department
    • Special thanks to IT-ST
  – Plan to deliver a Pilot Service in 2016