S44: High Energy Physics Big Data and the ATLAS experiment with a hands-on tutorial - 2

Data Science School Göttingen on 21.07.2017

Gen Kawamura
II.Physikalisches Institut, Universität Göttingen

Overview of Hands-on (30-40 mins)

- Let's start Hands-on
  - Your Grid environments
  - Hello World AMI (ATLAS Metadata Interface)
  - Hello World Athena job
  - Hello World PyRoot
    - Plot of electron Pt distribution
  - Hello World prun Grid job
  - Hello World PyRoot Grid job

Your Grid environments

VMs at Göttingen University

- Loging in your VM

```
## USE your user account
ssh ...@...VM
```

Prepare your environments

- Reading ATLAS environments from CVMFS

```
## Cloning materials
$ cd
$ git clone https://github.com/GenKawamura/DataScienceSummerSchool_ATLAS_2017
$ cd DataScienceSummerSchool_ATLAS_2017
```

Setup CVMFS

- Reading ATLAS environments from CVMFS

```
## Alias to setupCVMFS
setupCVMFS(){
   export LCG_LOCATION=
   export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
   source $ATLAS_LOCAL_ROOT_BASE/user/atlasLocalSetup.sh ""

   ## Using EMI LCG package
   source ${ATLAS_LOCAL_ROOT_BASE}/packageSetups/atlasLocalEmiSetup.sh --emiVersion ${emiVersionVal}
}

## Or, use a script
. setupATLASHandsOn.sh


## Using CVMFS (with EMI LCG client tools)
setupCVMFS
```

Hands-on exercise grid certificate

- Checking your certificate and VO

```
##  Copying a proxy certificate
export X509_USER_PROXY=/tmp/x509_cert_$UID
cp -v grid_proxy $X509_USER_PROXY
chmod 600 $X509_USER_PROXY

## Check your VOMS proxy
voms-proxy-info -all

## Read X509 attributes if you are interested
openssl x509 -in $X509_USER_PROXY -text | less
```

How it works

- You get a temporary key of a door now

Hello World AMI (ATLAS Metadata Interface)

Hands-on exercise
pyAMI Interface

- AMI CLI interface

```
##  Loading the pyAMI environment
$ lsetup pyami

##  Search data of 2016 and period A1
$ ami list datasets data16_13TeV%periodA1.%
   data16_13TeV.periodA1.physics_Main.PhysCont.AOD.t0pro20_v01
   data16_13TeV.periodA1.physics_Main.PhysCont.DAOD_STDM2.grp16_v01_p2623
   data16_13TeV.periodA1.physics_Main.PhysCont.DAOD_STDM4.grp16_v01_p2623
   data16_13TeV.periodA1.physics_Main.PhysCont.DAOD_STDM5.grp16_v01_p2623
   data16_13TeV.periodA1.physics_Main.PhysCont.DAOD_STDM7.grp16_v01_p2623
```

Hands-on exercise check metadata by pyAMI

```
##  Show metadata of a dataset
$ ami show dataset info data16_13TeV.00284285.physics_Main.merge.AOD.f662_m1453_r8067_p2645
   logicalDatasetName : data16_13TeV.00284285.physics_Main.merge.AOD.f662_m1453_r8067_p2645
   nFiles : 0
   totalEvents : 0
   totalSize : NULL
   runNumber : 284285
   period : J6
   prodsysStatus : NO EVENTS YET
   dataType : AOD
   beamType : NULL
   conditionsTag : NULL
   geometryVersion : NULL
   streamName : physics_Main
   version : f662_m1453_r8067_p2645
   lastModified : 2016-06-09 18:35:05
   amiStatus : VALID
   created : 2016-06-09 18:35:04
   inContainer : 0
   added_comment : NULL
   keyword : NULL
   prodsysIdentifier_0: 8650873
   taskStatus_0 : UNKNOWN:METADATA ERROR
   TIDState_0 : added
   task_lastModified_0: 2016-06-10 09:24:25
```

Hands-on exercise check metadata by pyAMI

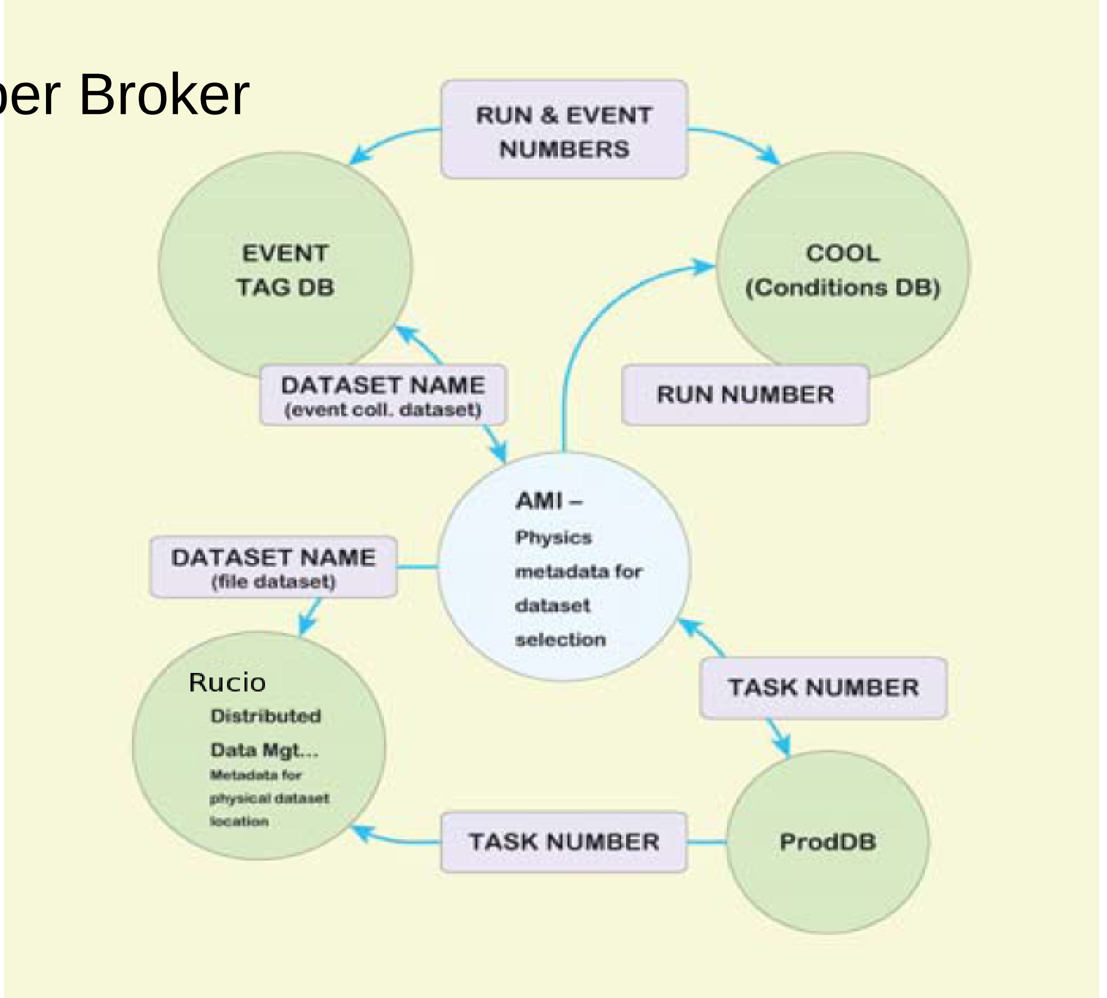
Show RAWs

$ ami show dataset prov data16_13TeV.00284285.physics_Main.merge.AOD.f662_m1453_r8067_p2645

...

How it works

- Applications
  - The Monte-Carlo Dataset Number Broker
  - The ATLAS Metadata directory
  - Tag collector
- ProdDB
  - For Monte-Carlo simulation



S. Albrand, T. Doherty, J. Fulachier, F. Lambert. The ATLAS Metadata Interface. Interna-tional Conference on Computing in High Energy and Nuclear Physics (CHEP-07), Sep 2007, Victoria, Canada. IOP Publishing, 120, pp.072003, 2008, <10.1088/1742-6596/120/7/072003>. <in2p3-00192624>

Hello World Athena Job

Hands-on exercise
simple Athena job

- Only 5 events by Pythia MC generator

```
## Setup an Athena release
$ asetup 17.2.4,here,setup

## Run Pythia MC event generator
$ athena ajob_options/jobOptions.pythia16.py
```

Do not need this exercise, but it works on Grid as well

- Athena job by PanDA client
  - pathena

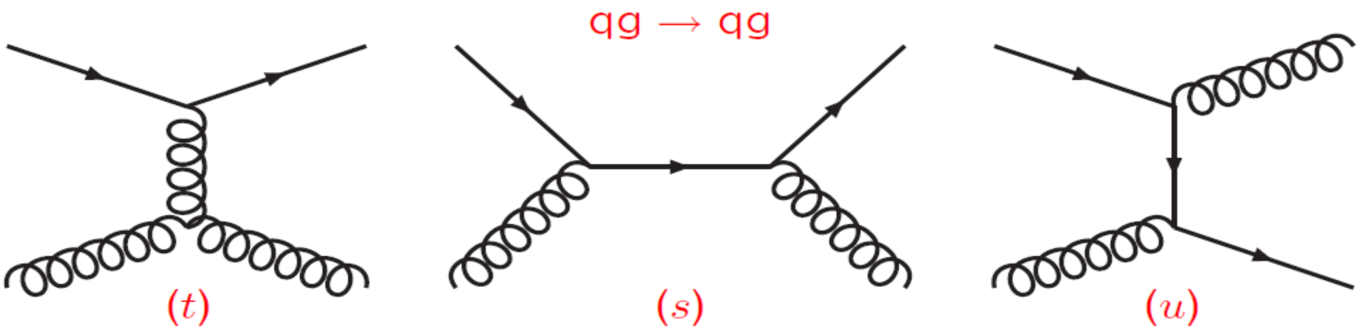
```
## Loading PanDA client
$ lsetup panda

## For example, you can seamlessly run Athena code on Grid
$ pathena ajob_options/jobOptions.pythia16.py  --outDS=user.gkawamur.evgen.pool.pythia.v1.$$ --split 5
```

How it works - 1

- Quantum mechanics
  - Each event is depending on event probability

A given initial and final state typically can be related via several separate intermediate histories, e.g.



Cross section $\sigma \propto |A_t + A_s + A_u|^2 \neq |A_t|^2 + |A_s|^2 + |A_u|^2$.

Interference $\Rightarrow$ not possible to know which path process took.

If one amplitude dominates then approximate simplifications (e.g. A_t dominates for scattering angle $\rightarrow 0$).

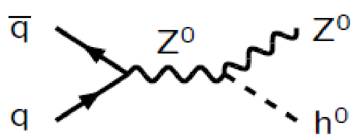
$$\text{Trick}: \quad \sigma_t \propto |A_t + A_s + A_u|^2 \frac{|A_t|^2}{|A_t|^2 + |A_s|^2 + |A_u|^2}$$

How it works - 2

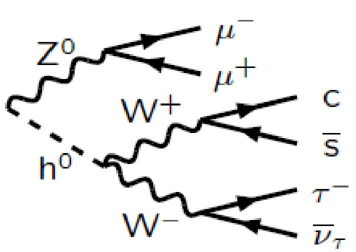
- The main physics components

Structure of the basic generation process:
(**Not** in physical time order, but $\sim$ by order of consideration.)

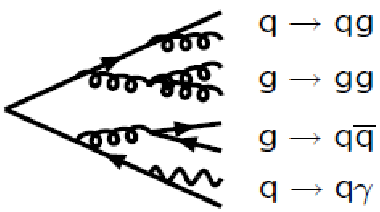
1) Hard subprocess:
$|\mathcal{M}|^2$, Breit-Wigners,
parton densities.



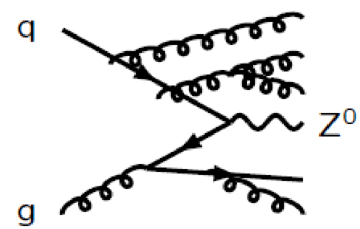
2) Resonance decays:
includes correlations.



3) Final-state parton showers.



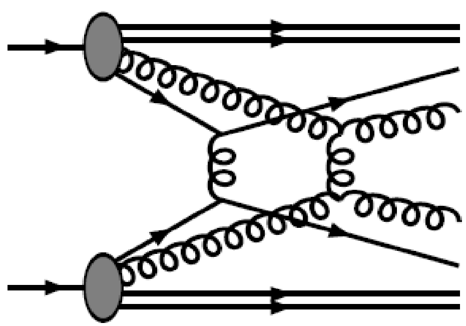
4) Initial-state parton showers.



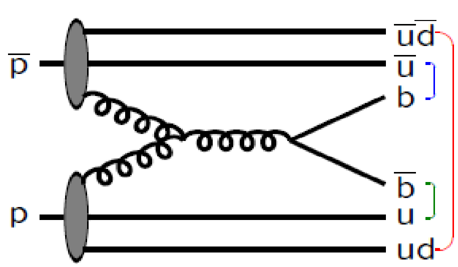
How it works - 3

- The main physics components

5) Multiple parton–parton interactions.

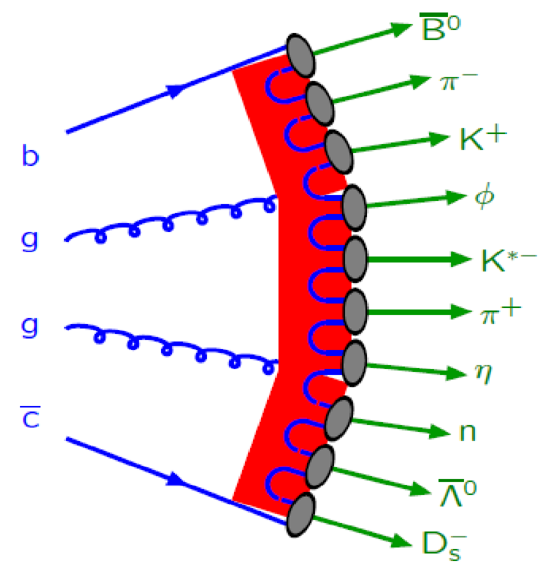


6) Beam remnants, with colour connections.

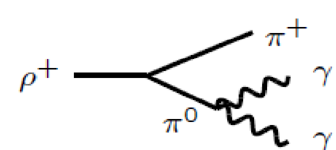


5) + 6) $\approx$ Underlying Event

7) Hadronization



8) Ordinary decays: hadronic, τ, charm, . . .



Hello World PyRoot

Hands-on exercise
PyRoot example

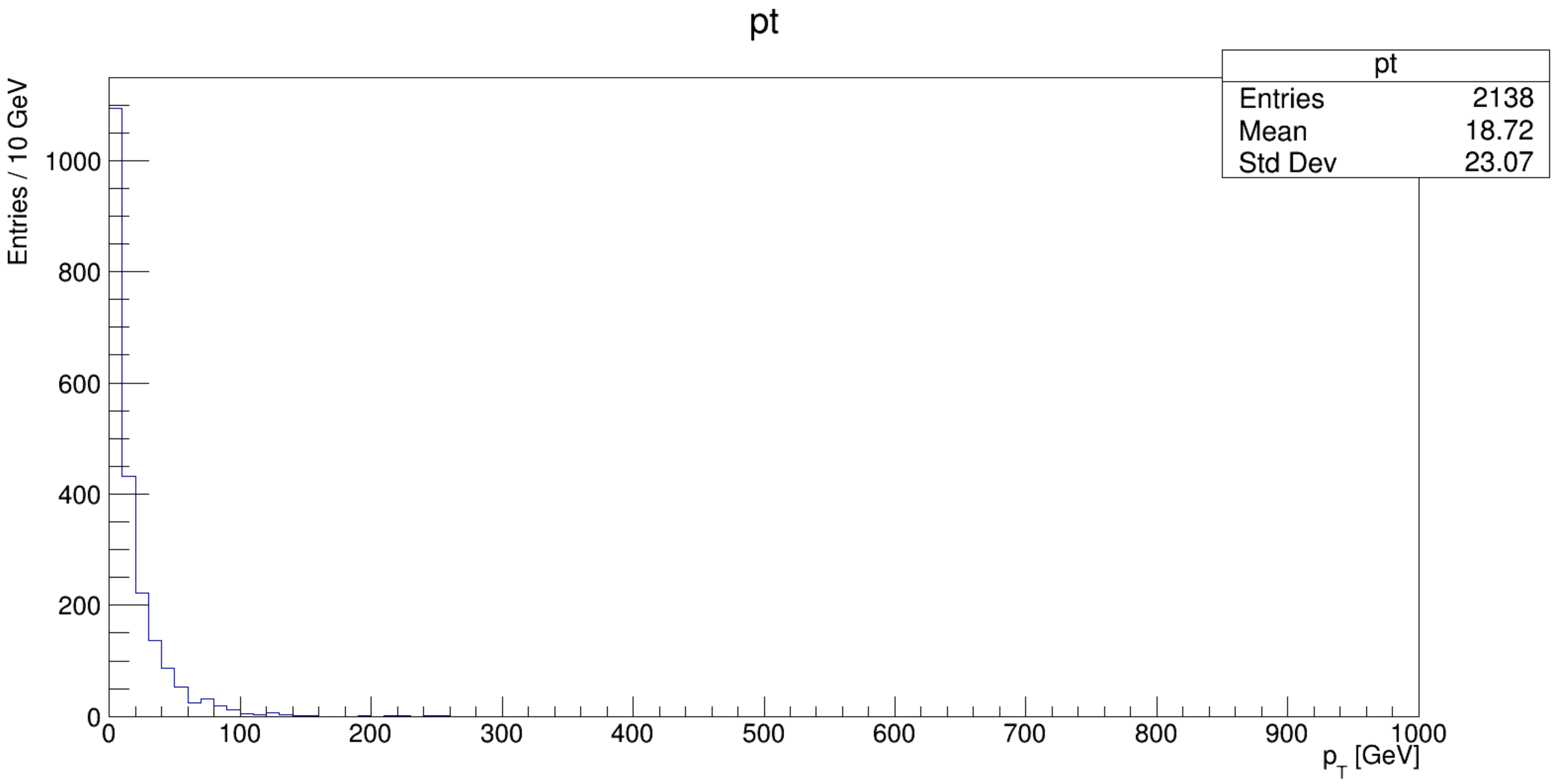
```
## Making PyRoot environments
 $ cd pyroot
 $ source pyroot_env.sh

## Getting a sample
 $ ./get-sample-files.sh -n 1
 $ ls valid2.117050.PowhegPythia_P2011C_ttbar.digit.AOD.e2657_s1933_s1964_r5534/* > input.txt

## Extracting and counting electron energy
 $ less xAOD_electron_hist_example.py
 $ ./xAOD_electron_hist_example.py -i input.txt -o hist.root

## Plotting electron energy distribution. (this may not work in VMs)
 $ root hist.root
 root [1] TBrowser t
```

Plot of electron Pt distribution



How it works

- Just looping entries (events) in a Root tree and counting electron Pt in histogram object

*** xAOD_electron_hist_example.py**

```
# Make the "transient tree":
t = ROOT.xAOD.MakeTransientTree( f, treeName)

print( "Number of input events: %s" % t.GetEntries() )
for entry in xrange( t.GetEntries() ):
    t.GetEntry( entry )
    print( "Processing run #%i, event #%i" % ( t.EventInfo.runNumber(), t.EventInfo.eventNumber() ) )
    print( "Number of electrons: %i" % len( t.ElectronCollection ) )
    # loop over electron collection
    for el in t.ElectronCollection:
        pthist.Fill(el.pt()/1000.)
        pass # end for loop over electron collection
    pass # end loop over entries
f.Close()
pass
```

Hello World prun Grid Job

Hands-on exercise
Using ATLAS client tools

- First “Hello world” job by PanDA client

PanDA client
lsetup panda

Make a Python script
cat hello_world.py

```
#!/usr/bin/python
print "Hello world!"
```

chmod 755 hello_world.py
./hello_world.py

Hello world!

Submitting a prun job
prun --outDS user.gkawamur.pruntest.$$ --exec hello_world.py

```
INFO : gathering files under /home/gen/tmp/for_new_comer
INFO : upload source files
INFO : submit
INFO : succeeded. new jediTaskID=5107461
```

Submitting 5 prun jobs
prun --outDS user.gkawamur.pruntest.$$ --exec hello_world.py –nJobs=5

What will happen?

- On PanDA web interface, we can find the jobs

jobstatus (1)	finished (2)
minramcount (1)	1-2GB (1)
outputfiletype (2)	? (1) log (1)
priorityrange (2)	1000:1099 (1) 2000:2099 (1)
processingtype (1)	panda-client-0.5.72-jedi-athena (2)
prodsourcelabel (2)	panda (1) user (1)
produsername (1)	Gen Kawamura (2)
reqid (1)	94 (2)
specialhandling (1)	ddm:rucio (2)
transformation (2)	buildJob-00-00-03 (1) runAthena-00-00-12 (1)

Prodsys Jobs Handling

Job list Sort by PandaID, time since last state change, ascending mod time, descending mod time, priority, attemptnr, ascending duration, descending duration

PanDA ID Attempt#	Owner Group	Request Task ID	Transformation	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Mod	Cloud Site
3131853110 Attempt 1	Gen Kawamura	94 10262517	runAthena-00-00-12	finished	2016-12-19 14:51	0:0:10:54	0:0:01:14	12-19 15:00	DE ANALY_IEPSAS-Kosice online HC.Blacklist.set.online
	Job name: user.gkawamur.tutorial2016.12/.3131853110 #1								
	Datasets: **Out:** user.gkawamur.tutorial2016.12.log.112492285								
3131853105 Attempt 0	Gen Kawamura	94 10262517	buildJob-00-00-03	finished	2016-12-19 14:51	0:0:02:38	0:0:03:17	12-19 15:00	DE ANALY_IEPSAS-Kosice online HC.Blacklist.set.online
	Job name: user.gkawamur.tutorial2016.12/ #0								
	Datasets: **Out:** panda.1219145126.858464.lib._10262517								

Hello World PyRoot Grid Job

Hands-on exercise
PyRoot with Grid

- First “Hello world” PyRoot job by PanDA client

```
## Making PyRoot environments
 $ inDS="valid2.117050.PowhegPythia_P2011C_ttbar.digit.AOD.e2657_s1933_s1964_r5534"
 $ outDS="user.gkawamur.DStutorial.pyroot.xAOD.v0.1_$$"
 $ infile="input.txt"
 $ outfile="hist.root"
 $ prun --useRootCore --inDS=$inDS --forceStaged \
--outDS=$outDS --outputs=$outfile --nFiles=100 --nFilesPerJob=1 \
--exec="echo %IN > $infile; xAOD_electron_hist_example.py -i $infile -o $outfile"
```

How it works

- Executing a task processing processing events per job (per file)