

# Some useful computing how to's

Yoann Kermaïdic

DUNE France analysis workshop

April 18<sup>th</sup>, 2023

# Introduction

- The goal is to give some practical examples on what I had to do when I developed, tested and ran DUNE code
- The corresponding DUNE tutorial:  
<https://dune.github.io/computing-training-basics-short/02-storage-spaces/index.html>  
<https://dune.github.io/computing-training-basics-short/05-expert-in-the-room-larsoft/index.html>
- Few custom scripts collection found at
  - <https://dune-france.pages.in2p3.fr/analysis-workshop/>
  - Please contribute =)
  - Can always dig in the computing tutorials and adapt them to your case

# Interactive work with VNC session

```
-----
- Setup VNC session tutorial -
- Y.Kermaidic -
- (09/02/2022) -
-----

# Aim to speed-up a lot the event display of LArSoft or LARION

0. Setup your SSH configuration:

0. Adapted from https://cdcvs.fnal.gov/redmine/projects/sbndcode/wiki/Viewing_events_remote

1. On your local machine, put these lines into your ~/.ssh/config - change XX by a 2 digits

Host dune04
HostName dune04.fnal.gov
User your_username
ForwardAgent yes
ForwardX11 yes
ForwardX11Trusted yes
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
LocalForward 5901 localhost:59XX

2. On your local machine, make sure your /etc/krb5.conf file is properly set:

Look at https://authentication.fnal.gov/krb5conf/
Add the relevant configuration file to your /etc/krb5.conf
Try to log via:

kinit -a -r7d your_username@FNAL.GOV
-> you should be asked for your Kerberos password
ssh dune04
-> you should be logged in without being asked for any password

2. On your remote machine, put these lines into your ~/.bash_profile

#VNC stuff
VNCNUM=XX #CHANGE THIS NUMBER TO WHATEVER VNC SERVER NUMBER YOU PICKED
if [[ 'hostname' == *"gpvm"* ]] #only start VNC servers on the gpvms (i.e. not on the b
then
export DISPLAY=localhost:$VNCNUM #Export the display to point to the VNC server
if [ `lsor -1 -P -n | grep $(expr 5900 + ${VNCNUM}) | wc -l` -eq 0 -o `lsor -1 -P -n
then
echo "vncserver:$VNCNUM not running. Starting now...."
vncserver:$VNCNUM -localhost -bs #Check if the VNC server is running and start
else
echo "vncserver:$VNCNUM already running (hopefully owned by you). Not attempting
fi
fi

2. Log in on your remote machine, start a VNC session and a test display (like ROOT TBrowser):

kinit -a -r7d your_username@FNAL.GOV
ssh dune04

vncserver :XX -localhost -bs
-> setup a custom/dedicated password "MYPWD"

source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
setup dunes v09_42_02_00-q e20:prof
root
new TBrowser

3. On your local machine, open the VNC window:

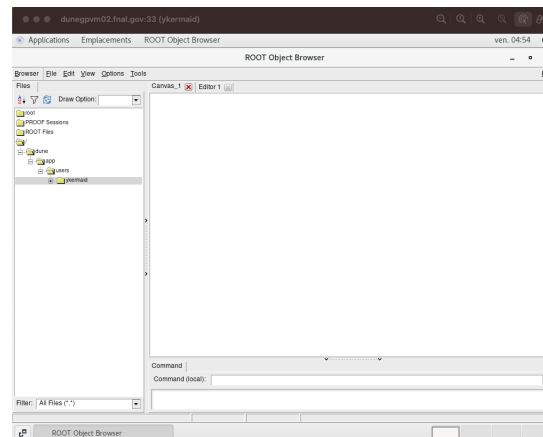
open vnc://localhost:5901
-> Enter your custom password "MYPWD"
-> You should see a Scientific Linux window opening
```

- **Reference:**

[https://cdcvs.fnal.gov/redmine/projects/sbndcode/wiki/Viewing\\_events\\_remotely\\_with\\_VNC](https://cdcvs.fnal.gov/redmine/projects/sbndcode/wiki/Viewing_events_remotely_with_VNC)

- **First point to get settled with when working outside of your lab**

- Only transfer remote display (« no data » transfer)
- Speed-up by large factor (no lag)



# Setup DUNE software

- On our data centers, CVMFS is available.
- I have the following lines in a `~/.larsoft_profile` file that I source

```
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
export DUNEVERSION=v09_72_00d00
export DUNEQUALIFIER="e20:prof"
setup dunesw $DUNEVERSION -q $DUNEQUALIFIER
```

- You can then easily execute

```
lar --help
```

# Develop your ROOT macros

- Found it useful to learn/manipulate LArSoft objects in a light way

- Use of the ‘Gallery’ environment to loop through ‘art::Event’

```
R__ADD_INCLUDE_PATH("gallery/Event.h")
```

```
for (gallery::Event ev(filenamees); !ev.atEnd(); ev.next()) {  
    your processing  
}
```

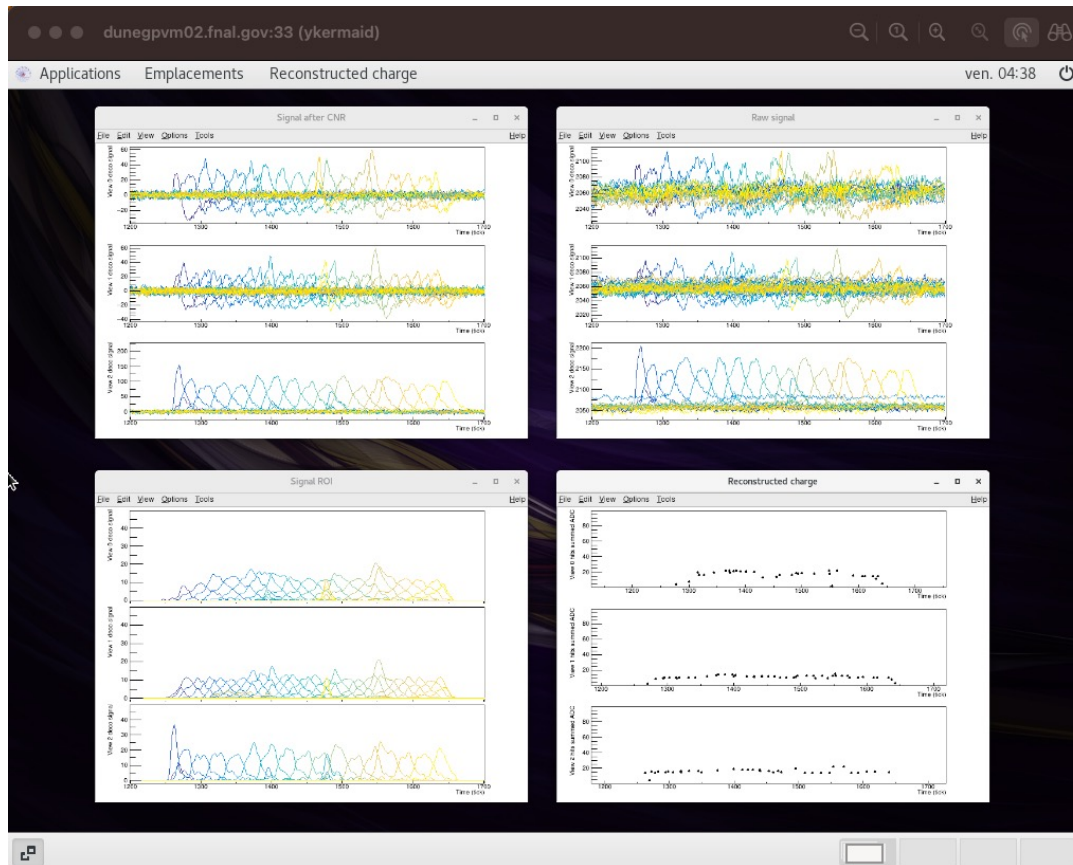
- Easy to execute:

```
root -l  
[0] .L YourMacro.C  
[1] YourMacro()
```

# Develop your ROOT macros

- See e.g. <https://gitlab.in2p3.fr/dune-france/analysis-workshop/-/blob/master/macros/PlotDecoPulses.C>  
 .L PlotDecoPulses.C  
 PlotDecoPulses(10894,50,5,1200,1700,true)

Pulses after  
CNR



Raw pulses

Pulses after  
deconvolution

Corresponding  
hit charge

# Develop your Module

## Implement a local LArSoft

```
-----
- Compile local LArSoft tutorial -
- T.Houdy -
- (3/02/2023) -
-----

1. Login at CCIN2P3/CERN/FNAL

-----

For your first time:

-----

2. Choose the place you want to settle your local LArSoft software

$ mkdir dune_workdir
$ cd dune_workdir
$ source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
$ mrb newDev -v v09_63_00d01 -q prof:e20
$ source localProducts_larsoft_v09_63_00d01_prof_e20/setup
$ echo $MRB_PROJECT
return: "
MRB_Project =
...
"

-----

3. To get the source code

$ cd $MRB_SOURCE
$ mrb g dunesw
$ cd dunesw
$ git checkout develop -b local_tibo_version

-----

4. Download only what you want to modify, the rest will be herited from the corresponding version

----- To download DUNE/dunereco -----
$ cd $MRB_SOURCE
$ mrb g dunereco
$ cd dunereco
$ git checkout develop -b local_tibo_reco

----- To download DUNE/dunesim -----
$ cd $MRB_SOURCE
$ mrb g dunesim
$ cd dunesim
$ git checkout develop -b local_tibo_sim

-----

5. Check compatibility between versions of source codes
$ cd $MRB_SOURCE
$ cd dunesw
$ mrbsetenv

-----

6. If no error messages, start building
$ cd $MRB_BUILDDIR
$ mrb install -j3
----- If installation is successful -----
$ mrbslp
$ cd ..
$ tar
```

- Make your code usable by anybody in the collaboration – people may be interested to reproduce your study
- Straightforward tutorials, don't hesitate to try!
- Let's try on a simple module
  - Print out the event number

# Develop your Module

- Once you have your local LArSoft installation:

```
-bash-4.2$ cetskelgen
```

```
-v
```

```
-d /dune/app/users/ykermaid/sw/dune_workdir/srcs/protoduneana/protoduneana/verticaldrift/workshop/
```

```
-e beginJob -e endJob analyzer myns::PrintEventNumber
```

```
INFO: Adding required member function void analyze(art::Event const& e) override
```

```
INFO: Adding optional member function void beginJob() override
```

```
INFO: Adding optional member function void endJob() override
```

```
INFO: Wrote
```

```
/dune/app/users/ykermaid/sw/dune_workdir/srcs/protoduneana/protoduneana/verticaldrift/workshop//PrintEventNumber_module.cc
```

```
//////////////////////////////////////////////////////////////////  
// Class:      PrintEventNumber  
// Plugin Type: analyzer (Unknown Unknown)  
// File:       PrintEventNumber_module.cc  
//  
// Generated at Thu Apr 13 15:42:05 2023 by Yoann Kermaidic using cetskelgen  
// from version .  
//////////////////////////////////////////////////////////////////  
#include "art/Framework/Core/EDAnalyzer.h"  
#include "art/Framework/Core/ModuleMacros.h"  
#include "art/Framework/Principal/Event.h"  
#include "art/Framework/Principal/Handle.h"  
#include "art/Framework/Principal/Run.h"  
#include "art/Framework/Principal/SubRun.h"  
#include "canvas/Utilities/InputTag.h"  
#include "fhiclcpp/ParameterSet.h"  
#include "messagefacility/MessageLogger/MessageLogger.h"  
  
namespace myns {  
  class PrintEventNumber;  
}  
  
class myns::PrintEventNumber : public art::EDAnalyzer {  
public:  
  explicit PrintEventNumber(fhicl::ParameterSet const& p);  
  // The compiler-generated destructor is fine for non-base  
  // classes without bare pointers or other resource use.  
  
  // Plugins should not be copied or assigned.  
  PrintEventNumber(PrintEventNumber const&) = delete;  
  PrintEventNumber(PrintEventNumber&&) = delete;  
  PrintEventNumber& operator=(PrintEventNumber const&) = delete;  
  PrintEventNumber& operator=(PrintEventNumber&&) = delete;  
  
  // Required functions.  
  void analyze(art::Event const& e) override;  
  
  // Selected optional functions.  
  void beginJob() override;  
  void endJob() override;  
};
```

```
private:  
  // Declare member data here.  
};  
  
myns::PrintEventNumber::PrintEventNumber(fhicl::ParameterSet const& p)  
  : EDAnalyzer{p} // ,  
  // More initializers here.  
{  
  // Call appropriate consumes<>() for any products to be retrieved by this module.  
}  
  
void myns::PrintEventNumber::analyze(art::Event const& e)  
{  
  // Implementation of required member function here.  
}  
  
void myns::PrintEventNumber::beginJob()  
{  
  // Implementation of optional member function here.  
}  
  
void myns::PrintEventNumber::endJob()  
{  
  // Implementation of optional member function here.  
}  
  
DEFINE_ART_MODULE(myns::PrintEventNumber)
```



# Develop your Module

- Update the namespace (myns -> dunefrworkshop)
- Add your custom input parameters (fLogLevel) set in the FHICL file
- Initiate the code and write your event analyser (here cout only)

```
////////////////////////////////////  
// Class:      PrintEventNumber  
// Plugin Type: analyzer (Unknown Unknown)  
// File:       PrintEventNumber_module.cc  
//  
// Generated at Thu Apr 13 15:42:05 2023 by Yoann Kermaidic using cetskelgen  
// from version .  
////////////////////////////////////  
  
#include "art/Framework/Core/EDAnalyzer.h"  
#include "art/Framework/Core/ModuleMacros.h"  
#include "art/Framework/Principal/Event.h"  
#include "art/Framework/Principal/Handle.h"  
#include "art/Framework/Principal/Run.h"  
#include "art/Framework/Principal/SubRun.h"  
#include "canvas/Utilities/InputTag.h"  
#include "fhiclcpp/ParameterSet.h"  
#include "messagefacility/MessageLogger/MessageLogger.h"  
  
namespace dunefrworkshop {  
    class PrintEventNumber;  
}  
  
class dunefrworkshop::PrintEventNumber : public art::EDAnalyzer {  
public:  
    explicit PrintEventNumber(fhicl::ParameterSet const& p);  
    // The compiler-generated destructor is fine for non-base  
    // classes without bare pointers or other resource use.  
  
    // Plugins should not be copied or assigned.  
    PrintEventNumber(PrintEventNumber const&) = delete;  
    PrintEventNumber(PrintEventNumber&&) = delete;  
    PrintEventNumber& operator=(PrintEventNumber const&) = delete;  
    PrintEventNumber& operator=(PrintEventNumber&&) = delete;  
  
    // Required functions.  
    void analyze(art::Event const& e) override;  
  
    // Selected optional functions.  
    void beginJob() override;  
    void endJob() override;  
};
```

```
private:  
    // Declare member data here.  
    int fLogLevel;  
};  
  
dunefrworkshop::PrintEventNumber::PrintEventNumber(fhicl::ParameterSet const& p)  
    : EDAnalyzer{p},  
    fLogLevel(p.get< int >("LogLevel"))  
    // More initializers here.  
{  
    // Call appropriate consumes<>() for any products to be retrieved by this module.  
}  
  
void dunefrworkshop::PrintEventNumber::analyze(art::Event const& e)  
{  
    // Implementation of required member function here.  
    if( fLogLevel >= 2 ) std::cout << "Start analysing event " << fEventNum << " ..." << std::endl;  
}  
  
void dunefrworkshop::PrintEventNumber::beginJob()  
{  
    // Implementation of optional member function here.  
    if( fLogLevel >= 1 ) std::cout << "My module is starting" << std::endl;  
}  
  
void dunefrworkshop::PrintEventNumber::endJob()  
{  
    // Implementation of optional member function here.  
    if( fLogLevel >= 1 ) std::cout << "My module is done" << std::endl;  
}  
  
DEFINE_ART_MODULE(dunefrworkshop::PrintEventNumber)
```

# Compile your Module

- Add your module into the CMakeLists

- In <https://github.com/YoannKermaidic/protoduneana/blob/develop/protoduneana/verticaldrift/CMakeLists.txt>  
add\_subdirectory(checks)  
add\_subdirectory(workshop)
- In verticaldrift/workshop/CMakeLists.txt

```
cet_build_plugin(  
    PrintEventNumber art::module  
    lardataobj::RecoBase  
    lardata::ArtDataHelper  
    lardata::Utilities  
    art::Framework_Core  
    art::Framework_Principal  
    art::Framework_Services_Registry  
    art_root_io::tfile_support  
    ROOT::Core  
    art_root_io::TFileService_service  
    art::Persistency_Common  
    canvas::canvas  
    art::Persistency_Provenance  
    art::Utilities  
    messagefacility::MF_MessageLogger  
    cetlib::cetlib  
    cetlib_except::cetlib_except  
    ROOT_BASIC_LIB_LIST  
    dune prototypes::ProtoDUNEDataUtils  
    ProtoDUNEUtilities  
    BASENAME_ONLY  
)  
install_fhicl()  
install_source()
```

<- copy/paste of working example  
could potentially be simplified

# Compile your Module

- Setup your environment:

- source /dune/app/users/ykermaid/sw/dune\_workdir/localProducts\_larsoft\_v09\_65\_03d00\_prof\_e20/setup
- mrbsetenv && mrbslp
- cd \$MRB\_SOURCE && mrbsetenv && cd \$MRB\_BUILDDIR && mrb install -j8

- Now compiling

```
INFO: install prefix = /dune/app/users/ykermaid/sw/dune_workdir/localProducts_larsoft_v09_65_03d00_prof_e20
-----
INFO: stage cmake for MRB project larsoft v09_65_03d00
-----
```

... (can take time at the « generation » stage)

```
-----
INFO: stage install SUCCESS for MRB project larsoft v09_65_03d00
-----
```

# Setup a minimal FHICL file

```
#include "services_dune.fcl"

process_name: WSPrintEvent

services:{
  TFileService:          { fileName: "ws_print_evt_output.root" }
  TimeTracker:           {}
  RandomNumberGenerator: {}
  ExptGeoHelperInterface: @local::dune_geometry_helper
  Geometry:              @local::dunecrpcb_geo
  LArPropertiesService:   @local::dunefd_properties
  DetectorClocksService: @local::protodune_detectorclocks
  DetectorPropertiesService: @local::protodune_detproperties
  SpaceCharge:           @local::dunefd_spacecharge
}

source:
{
  module_type: RootInput
  maxEvents: -1
  fileNames: [ "detsim.root" ]
}

outputs: {}

physics:
{
  analyzers:
  {
    printevt:
    {
      module_type: "PrintEventNumber"
      LogLevel: 2
    }
  }
  analysis: [ printevt ]
  end_paths: [ analysis ]
}
```

- Your module!
- Your input parameter!

# Setup a minimal FHICL file

```
#include "services_dune.fcl"

process_name: WSPrintEvent

services:{
  TFileService:          { fileName: "ws_print_evt_output.root" }
  TimeTracker:           {}
  RandomNumberGenerator: {}
  ExptGeoHelperInterface: @local::dune_geometry_helper
  Geometry:              @local::dunecrpcb_geo
  LArPropertiesService:   @local::dunefd_properties
  DetectorClocksService: @local::protodune_detectorclocks
  DetectorPropertiesService: @local::protodune_detproperties
  SpaceCharge:           @local::dunefd_spacecharge
}

source:
{
  module_type: RootInput
  maxEvents: -1
  fileNames: [ "detsim.root" ]
}

outputs: {}

physics:
{
  analyzers:
  {
    printevt:
    {
      module_type: "PrintEventNumber"
      LogLevel: 2
    }
  }
  analysis: [ printevt ]
  end_paths: [ analysis ]
}
```

- One of the most important block to check! (i.e. the one you may have less control on)

- Your module!
- Your input parameter!

- We will have time to discuss FHICL settings more in depth in subsequent sessions ;-)

# Ready to go! (at FNAL)

```
lar -c ws_print_evt.fcl
xroot://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/dunepro/vd-coldbox-top/full-
reconstructed/2023/detector/test/VD_coldbox_CRP2_CRP3_2022/00/00/17/27/1727_100_a_cb_reco_67284240_0_2023-04-
11T124909Z.root
```

```
My module is starting
Begin processing the 1st record. run: 1727 subRun: 0 event: 23761 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23761 ...
Begin processing the 2nd record. run: 1727 subRun: 0 event: 23765 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23765 ...
Begin processing the 3rd record. run: 1727 subRun: 0 event: 23769 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23769 ...
```

```
Begin processing the 60th record. run: 1727 subRun: 0 event: 23997 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23997 ...
14-Apr-2023 10:14:37 CEST Closed input file "xroot://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/dunepro/vd-coldbox-top/fu
ll-reconstructed/2023/detector/test/VD_coldbox_CRP2_CRP3_2022/00/00/17/27/1727_100_a_cb_reco_67284240_0_2023-04-11T124909Z.root"
My module is done
```

```
=====
TimeTracker printout (sec)           Min           Avg           Max           Median           RMS           nEvts
=====
Full event                          0.000114279    0.0018671     0.0201962     0.000198546     0.00463903    60
source:RootInput(read)              8.0453e-05    0.00180225    0.0201239     0.00013743      0.00461853    60
end_path:printevt:PrintEventNumber  4.869e-06     1.01829e-05   5.2277e-05    7.0385e-06      9.03464e-06   60
=====
```

```
TrigReport ----- Event summary -----
TrigReport Events total = 60 passed = 60 failed = 0 Zambelli
```

```
TrigReport ----- Modules in End-path -----
TrigReport Run Success Error Name
TrigReport 60 60 0 printevt
```

```
TimeReport ----- Time summary [sec] -----
TimeReport CPU = 0.034642 Real = 0.145842
```

```
MemReport ----- Memory summary [base-10 MB] -----
MemReport VmPeak = 1981.48 VmHWM = 738.456
```

```
Art has completed and will exit with status 0.
```

- Now compiling



# Ready to go! (at FNAL)

```
lar -c ws_print_evt.fcl
xroot://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/dunepro/vd-coldbox-top/full-
reconstructed/2023/detector/test/VD_coldbox_CRP2_CRP3_2022/00/00/17/27/1727_100_a_cb_reco_67284240_0_2023-04-
11T124909Z.root
```

```
My module is starting
Begin processing the 1st record. run: 1727 subRun: 0 event: 23761 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23761 ...
Begin processing the 2nd record. run: 1727 subRun: 0 event: 23765 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23765 ...
Begin processing the 3rd record. run: 1727 subRun: 0 event: 23769 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23769 ...
```

```
Begin processing the 60th record. run: 1727 subRun: 0 event: 23997 at 14-Apr-2023 10:14:37 CEST
Start analysing event 23997 ...
14-Apr-2023 10:14:37 CEST Closed input file "xroot://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/dunepro/vd-coldbox-top/fu
ll-reconstructed/2023/detector/test/VD_coldbox_CRP2_CRP3_2022/00/00/17/27/1727_100_a_cb_reco_67284240_0_2023-04-11T124909Z.root"
My module is done
```

```
=====
TimeTracker printout (sec)           Min           Avg           Max           Median           RMS           nEvts
=====
Full event                          0.000114279    0.0018671     0.0201962     0.000198546     0.00463903    60
source:RootInput(read)              8.0453e-05    0.00180225    0.0201239     0.00013743      0.00461853    60
end_path:printevt:PrintEventNumber  4.869e-06     1.01829e-05  5.2277e-05    7.0385e-06      9.03464e-06   60
=====
```

```
TrigReport ----- Event summary -----
TrigReport Events total = 60 passed = 60 failed = 0 (ambigu)
```

```
TrigReport ----- Modules in End-path -----
TrigReport Run Success Error Name
TrigReport 60 60 0 printevt
```

```
TimeReport ----- Time summary [sec] -----
TimeReport CPU = 0.034642 Real = 0.145842
```

```
MemReport ----- Memory summary [base-10 MB] -----
MemReport VmPeak = 1981.48 VmHWM = 738.456
```

```
Art has completed and will exit with status 0.
```

- Hum, why is this file name looking so complicated?

# XRootD

## Create reconstructed files list

```
#!/bin/bash

#-----
#- Reconstructed files listing -
#-      Y.Kermaidic      -
#-      (23/03/2023)    -
#-----

source ~/.larsoft_profile

# you may need to run 'kx509' first for validating your certificate to access the samweb database

SCRIPT_DIR=$( pwd )

PROD_KEY=2023-03
TOP_RUN=17

INPUT_DIR=/pnfs/dune/tape_backed/dunepro/vd-coldbox-top/full-reconstructed/2023/detector/test/VA
OUTPUT_DIR=/your/path/where/to/store/the/xrootd/list/of/files

echo "INPUT_DIR: ${INPUT_DIR}"
cd ${INPUT_DIR}

RUNS=$( ls )
for RUN in ${RUNS[@]}
do
  echo "RUN: ${RUN}"
  cd ${RUN}
  FILES=$( ls ${TOP_RUN}-${RUN}*_${PROD_KEY}*.root )
  for FILE in ${FILES[@]}
  do
    FULL_PATH=$( samweb get-file-access-url --schema=xroot ${FILE} )
    echo "FILE: ${FILE} -> ${FULL_PATH}"
    echo "${FULL_PATH}" >> ${OUTPUT_DIR}/${TOP_RUN}/${RUN}.list
  done
done
cd -
done
cd ${SCRIPT_DIR}
```

- **Run your code interactively on xrootd file list**

The eXtended ROOT daemon is software framework designed for accessing data from various architectures and in a complete scalable way (in size and performance).

XRootD is most suitable for read-only data access.

- **Done via**

`samweb get-file-access-url --schema=xroot ${FILE}`

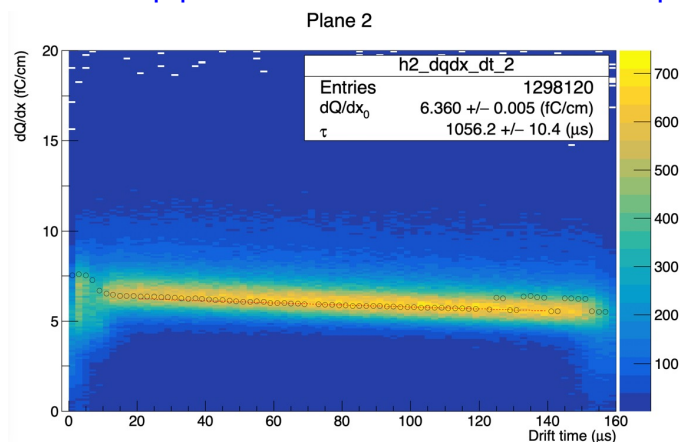
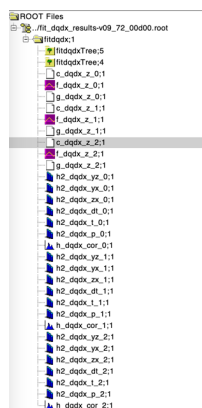
Moving to RUCIO (location) & MetaCat (catalog)

- You will need a grid proxy to submit jobs and access data in dCache via xrootd or ifdh  
`setup_fnal_security`



# Concrete example

- Data quality for ProtoDUNE-VD
  - We created a new directory for vertical drift in ‘protoduneana’  
<https://github.com/DUNE/protoduneana/tree/develop/protoduneana/verticaldrift>
  - Run-wise electron lifetime stability through dQ/dx attenuation fit
    - surely something similar was already done for ProtoDUNE-SP but we have some specificities in the VD design
    - Needed a new module running on a full VD-CB dataset:  
[https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/FitdQdx\\_module.cc](https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/FitdQdx_module.cc)  
[https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/pdvd\\_fit\\_dqdx.fc](https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/pdvd_fit_dqdx.fc)



# Concrete example

- Data quality for ProtoDUNE-VD
  - **We created a new directory for vertical drift in ‘protoduneana’**  
<https://github.com/DUNE/protoduneana/tree/develop/protoduneana/verticaldrift>
  - **Run-wise electron lifetime stability through  $dQ/dx$  attenuation fit**
    - surely something similar was already done for ProtoDUNE-SP but we have some specificities in the VD design
    - Needed a new module running on a full VD-CB dataset:  
[https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/FitdQdx\\_module.cc](https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/FitdQdx_module.cc)  
[https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/pdvd\\_fit\\_dqdx.fcl](https://github.com/DUNE/protoduneana/blob/develop/protoduneana/verticaldrift/checks/pdvd_fit_dqdx.fcl)
  - **Developped with mrb (see tuto), then the code was copied in my fork**  
<https://github.com/YoannKermaidic/protoduneana>
  - **Finally made a ‘Pull Request’ and merged into DUNE software by Tom**  
<https://github.com/DUNE/protoduneana/pull/16>

# Job submission

## Run jobs @FNAL

```
#!/bin/bash

#-----
#-          SubmitFNALjobs  -
#-          Y.Kermaidic    -
#-          (23/03/2023)    -
#-----

SCRIPT_PATH=/dune/app/users/username/scripts

TOP_RUN=17
RECO_PATH=/pnfs/dune/tape_backed/dunepro/path/to/your/reco/data/00/00/${TOP_RUN}
OUTPUT_PATH=/pnfs/dune/resilient/users/username/your/output/path

RUNS=$(ls ${RECO_PATH});

source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
setup jobsub_client

for RUN in ${RUNS[@]};
do
  # Assumes that you have the list of files you want to process in OUTPUT_PATH, and called "${TOP_RUN}
  RUNLISTJOB="${TOP_RUN}/${RUN}_reco.list"

  RUNLIST="${OUTPUT_PATH}/${RUNLISTJOB}"

  OUTPUT_FILE="output-filename_${TOP_RUN}/${RUN}.root"

  echo "
  echo "SubmitFNALjobs.sh : "
  echo " - RUNLIST: ${RUNLIST} "
  echo " - OUTPUT_FILE: ${OUTPUT_FILE}"
  echo "
  # Interactive start for initial check of the script
  #bash ${SCRIPT_PATH}/RunFNALscript.sh ${RUNLISTJOB} ${OUTPUT_FILE}

  # Job submission command
  jobsub_submit -N 1 -M --OS=SL7 --group=dune --expected-lifetime=3h --resource-provides=usage.
done;
```

- **Simple command example:**

```
jobsub_submit
-N 1          # number of copies of this job
-M
--OS=SL7     # Scientific Linux 7
--group=dune
--expected-lifetime=3h
--resource-provides=
usage_model=DEDICATED,OPPORTUNISTIC
file://${SCRIPT_PATH}/your_script.sh
${ARGS}
```

- My use case: ProtoDUNE-DP diffusion analysis (~30 k files)
- Please comment on your usage (e.g. use of tar files?)

# Retrieve job output

```
#!/bin/bash
USER_PATH=/pnfs/dune/resilient/users/ykermaid
PROD_PATH=${_CONDOR_SCRATCH_DIR}
#PROD_PATH=/dune/app/users/username/test_jobsub # for initial interactive test

# source DUNE software to get access to ifdh
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
export DUNEVERSION=v09.65.03d00
export DUNEQUALIFIER="e20:prof"
setup dunesw $DUNEVERSION -q $DUNEQUALIFIER

# make a subdirectory of the Condor scratch directory
mkdir ${PROD_PATH}/ykermaid_products
mkdir ${PROD_PATH}/ykermaid_products/.upsfiles
cat > ${PROD_PATH}/ykermaid_products/.upsfiles/dbconfig << 'EOF'
FILE = DBCONFIG
AUTHORIZED_NODES = *
VERSION_SUBDIR = 1
PROD_DIR_PREFIX = ${UPS_THIS_DB}
EOF

cd ${PROD_PATH}/ykermaid_products

# copy tar ball products to custom grid accessible folder
ifdh cp ${USER_PATH}/dunesw-09.65.03d00-slf7-x86_64-e20-prof.tar.bz2 ${PROD_PATH}/ykermaid_products/
ifdh cp ${USER_PATH}/protoduneana-09.65.03d00-slf7-x86_64-e20-prof.tar.bz2 ${PROD_PATH}/ykermaid_products/
ifdh cp ${USER_PATH}/pddp_diffusion_analysis.fcl ${PROD_PATH}/
ifdh cp ${USER_PATH}/pddp/data/reco/$1 ${PROD_PATH}/

# untar products
tar -jxf dunesw-09.65.03d00-slf7-x86_64-e20-prof.tar.bz2
tar -jxf protoduneana-09.65.03d00-slf7-x86_64-e20-prof.tar.bz2

export PRODUCTS=${PROD_PATH}/ykermaid_products:$PRODUCTS
echo "PRODUCTS: ${PRODUCTS}"

setup dunesw $DUNEVERSION -q $DUNEQUALIFIER

lar -c ${PROD_PATH}/pddp_diffusion_analysis.fcl -S ${PROD_PATH}/$1 -T ${PROD_PATH}/$2
ifdh cp ${PROD_PATH}/$2 ${USER_PATH}/pddp/data/reco/
```

- **Idfh**

Another useful data handling command you will soon come across is ifdh. This stands for Intensity Frontier Data Handling. It is a tool suite that facilitates selecting the appropriate data transfer method from many possibilities while protecting shared resources from overload. You may see *ifdhc*, where *c* refers to *client*.

- Allows to forward files from *user space* to *production space* back and forth

# Conclusion

- Setup a VNC session once for all
- Two approaches for analysing data shown: ROOT macros / Module
- Good practice to access files: 'xrootd' using 'samweb' tool
- Job submission useful for testing code or run on small sample  
-> contact the 'DUNE production team' for large scale work  
(e.g. large simulation of ProtoDUNE-VD sample)