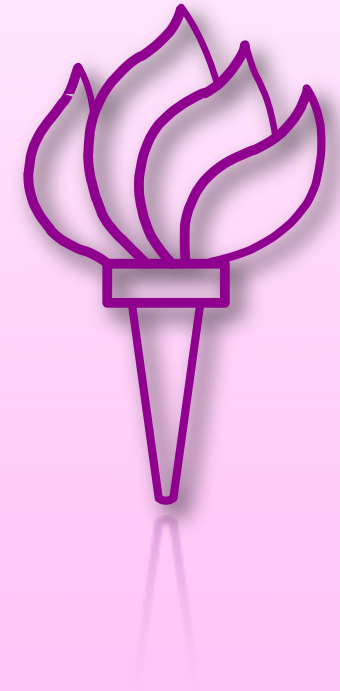


Tier3 examples and benchmarks



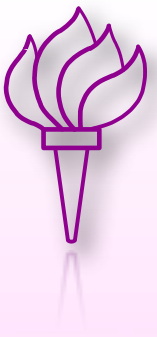
Attila Krasznahorkay
on behalf of a few people



NEW YORK UNIVERSITY

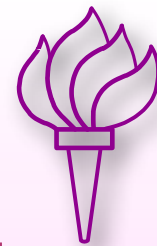


Overview



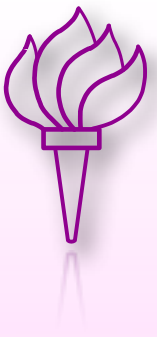
- **Tier3** analysis use cases
 - Creating D3PDs
 - Processing D3PDs
- **Documentation areas**
 - Generic examples based on `TTree::MakeClass()` and `TTree::MakeSelector()`
 - General tips for writing performant analysis code
 - A full-blown D3PD analysis code (L1DiMuon)

Introduction



- Why provide **Tier3** specific examples?
 - USATLAS **Tier3s** are mainly designed for ntuple (D3PD) analysis
 - The physics workbook contained instructions practically only for Athena
- Check if the current analysis methods fit into the **Tier3** design, and make suggestions to the physics groups about improvements to their code if they don't
- All this **will** become part of the analysis workbook ([link](#))

D3PD creation/handling



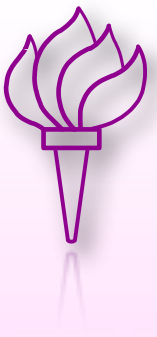
- TWiki [link](#)
- Only test jobs are envisaged to be run on **Tier3s**, or jobs on special datasets that are only available locally
- Panda (pathena) very efficient in quickly producing custom D3PD datasets
 - But many analyses should just start from one of the D3PD types created by the production system
- The details of how to get these datasets to the **Tier3s** is not part of the working group's mandate

Generic examples



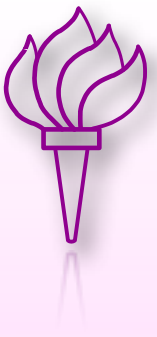
- [link1](#), [link2](#)
- Simple instructions for how to start writing an analysis from scratch
- Geared towards students who just started learning about ROOT and the data formats of ROOT
- Will have to update the instructions about `TSelector` a bit, as the current example will have problems when running with PROOF(-Lite)

SFrame examples



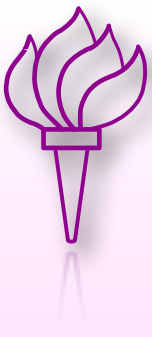
- SFrame is a light-weight “framework” for writing analysis code in (<http://sframe.sourceforge.net>)
- Used by a number of groups inside ATLAS
- Provides a **lot** of help in writing analysis code that can be run either on one processor core, on PROOF-Lite, or a full-blown PROOF cluster
- The generic documentation is on its own Wiki: <http://sourceforge.net/apps/mediawiki/sframe/>

SFrame in the workbook



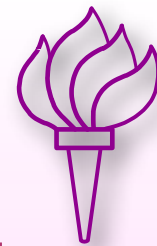
- Placeholder page exists already ([link](#))
- Will have to be extended to at least show the SFrame example developed within the **Tier3** PROOF working group
 - This is currently documented [here](#)
- Should focus on the following points:
 - Writing an SFrame package from scratch (a number of helper scripts exist for this)
 - Writing code in a PROOF-compatible way
 - Setting up a job to use PROOF(-Lite)

Tips for improving SW



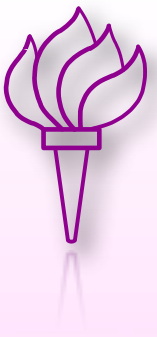
- The idea is to provide generic instructions to improve the performance of any kind of D3PD analysis code ([link](#))
- Should evolve with time, but already has all the basic advice
 - Don't **ever** use `TTree::GetEntry(...)`
 - Only read variables that your analysis needs
 - Do optimal event selection
 - Keeping the local ntuples organised

The L1DiMuon code



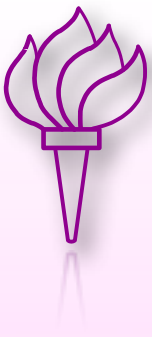
- An old analysis of mine which creates configuration files for the MuCTPI hardware and simulation
 - Was written as a set of CMT packages, as I wanted to use it on multiple platforms from the start
 - Depends on ROOT and XercesC
 - Some of the code is taken from the MuCTPI simulation, which uses XercesC for XML handling
 - Both of these should be available on the **Tier3** machines. XercesC is part of the ATLAS releases, but can also be installed from the SL5 software repository.
- Provides a realistic example of how to do a “technical” analysis - Only for advanced users

Getting L1DiMuon



- TWiki explaining how to retrieve/use the code:
<https://twiki.cern.ch/twiki/bin/view/Main/L1DiMuonAnalysis>
- Code location in SVN:
<https://svnweb.cern.ch/trac/atlasusr/browser/krasznaa/L1DiMuon>
- See backup slides for compilation instructions...

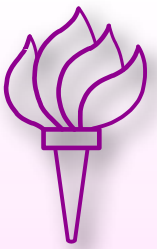
The L1DiMuon job



- Generate the special D3PDs (MuctpiD3PD) as input
 - Job0 that can be used for processing data AODs is in the directory: [DataAnalysis/athena/](#)
- Retrieve the produced D3PD in some **Tier3**-friendly way
- Set up the D3PD processing job:
 - Create an XML file describing the input files to the job:

```
l1dimu_dataInputCreator -o myInput.xml /dir/*.root
```
 - Get a GoodRunsLists XML for your dataset

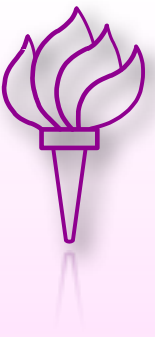
The L1DiMuon job



- Run the MuCTPI Look Up Table creator executable:
 - To run without using PROOF-Lite:

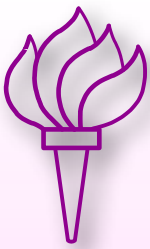
```
l1dimu_createLUT -x lut.xml -i myInput.xml -g  
LUTAnalysis/Geometry/nominal.xml -l myGRL.xml
```
 - To run using PROOF-Lite: Just add a “-p” to the argument list of `l1dimu_createLUT`
- Creates an XML file with the configuration for the MuCTPI (simulation)
 - The LUT extraction is a quite elaborate process, but has to produce the exact same XML output on all the sites (wether using PROOF-Lite or not)

Examples in L1DiMuon



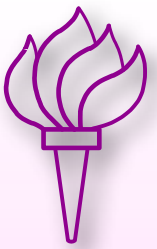
- The code demonstrates how to do the following in a D3PD processing code:
 - Use CMT in an Athena independent way
 - Write code in a platform independent way using CMT
 - Calculate complicated quantities on PROOF -- the code creates big STL structures which have to be merged amongst the PROOF worker nodes
- All the code is available for users to copy-paste into their projects

Summary



- A few of us started working on providing examples for using **Tier3** sites efficiently
- Will continue adding simple examples
- Creating realistic examples is by no means easy
 - A full analysis code usually depends on a number of libraries
 - Codes are usually only set up to work in the environment of the developer, little effort is put into making them flexible
 - The really interesting parts of an analysis are usually not automated -> Very hard to put them into an out-of-the-box example...

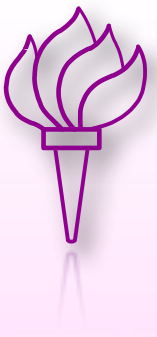
Backup



Compiling L1DiMuon

- On lxplus:
 - Set up a version of CMT
 - In an empty directory execute:

```
svn co svn+ssh://svn.cern.ch/repos/atlasusr/krasznaa/L1DiMuon/trunk ./
source setup_CERN.sh
./checkout_LUTAnalysis.sh
source setup_CERN.sh
cd LUTAnalysis/cmt
cmt br make
```

Compiling L1DiMuon

- In “standalone” mode (when not on lxplus):
 - Set up a version of CMT
 - In an empty directory execute:

```
svn co svn+ssh://svn.cern.ch/repos/atlasusr/krasznaa/L1DiMuon/trunk ./  
# Edit setup_STANDALONE.sh to match your system  
source setup_STANDALONE.sh  
./checkout_LUTAnalysis.sh  
source setup_STANDALONE.sh  
cd LUTAnalysis/cmt  
cmt br make
```