Standalone Ntuple analysis

Doug Benjamin
(Duke University)

Stand alone compiled root analysis

- Using Sergei Chekanov’s code and ntuples for this example
- Input Ntuples created in Athena either on grid or in your local cluster
- Compiled C++ code used for fast processing of the files
    - Single threaded processing
    - Straight forward migration toward Proof Analysis
- Accessing data stored in xrootd system

Makefile – xrootd hooks

- Makefile needs to have Netx library (not included by default)

```
include ${ROOTSYS}/test/Makefile.arch
# Root variables
ROOTCFLAGS    = $(shell root-config --nonew --cflags)
ROOTLIBS      = $(shell root-config --libs) -lNetx
ROOTGTTLIBS   = $(shell root-config --nonew --glibs)
# Assign or add variables
CXXFLAGS     += $(ROOTCFLAGS)
LIBS         += $(ROOTLIBS)
```

Source code changes for xrootd

- Data files passed to program from external file with list of input files

```
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00001.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00002.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00003.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00004.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00005.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00006.root
root://ascvmxrdr.hep.anl.gov//xrootd/ascwrk1/input_data/input_1_00007.root
```

- Changes to source code:

```
#include <TChain.h>
#include <TXNetFile.h>
#include <TFile.h>

 //TFile f(glob.ntup[i].c_str());
 TXNetFile f(glob.ntup[i].c_str());
 TTree *tree = (TTree *)f.Get("Ntuple");
```

Compiling

- Setup user environment

```
setupATLAS
localSetupGcc --gccVersion=gcc432_x86_64_slc5
localSetupPython --pythonVersion=2.5.2
localSetupROOT --rootVersion=5.26.00-slc5-gcc4.3
```

- Compiling and Linking
  - Make … very simple

Run in batch system with bare condor

- Create a tarball with exe and other needed files
- Condor Job classad file:

```
# The executable we want to run.
Executable      =  /users/benjamin/condor/ntuple_testjob/ntuple_test_andy.sh

# Environment for job
Environment="jobid=$(Process) datanode=ascwrk1 num_files=100"

# The argument to pass to the executable.
Arguments       = "ascwrk1"

# Input file given to the job.
Input           = /users/benjamin/condor/ntuple_testjob/main.tgz

# The job's stdout is sent to this file.
Output          = ntuple_test_xrootd_ascwrk1_running_on_ascwrk0_same_list.$(Cluster).$(Process).out

# The job's stderr is sent to this file.
Error           = ntuple_test_xrootd_ascwrk1_running_on_ascwrk0_same_list.$(Cluster).$(Process).err

# The condor log file for this job, useful when debugging.
Log             = ntuple_test_xrootd_ascwrk1_running_on_ascwrk0_same_list.$(Cluster).$(Process).log
```

Shell script used by Condor

```
#!/bin/bash
shopt -s expand_aliases

exe=main-xrootd
#
#  setup the job environment
#
export ATLAS_LOCAL_ROOT_BASE=/export/share/atlas/ATLASLocalRootBase
alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh'

#  Define which grid client to use

export ALRB_useGridSW=wlcg-client

setupATLAS

localSetupGcc --gccVersion=gcc432_x86_64_slc5
localSetupROOT --rootVersion=5.26.00-slc5-gcc4.3

# run executable
time ./$exe
```

Tail of stdout for job

```
TH1.Print Name  = track_den1a, Entries= 14041620, Total sum= 1.40416e+07
TH1.Print Name  = track_den2a, Entries= 12334193, Total sum= 1.23342e+07
TH1.Print Name  = track_den3a, Entries= 5042531, Total sum= 5.04253e+06
TH1.Print Name  = track_den1pt, Entries= 35163482, Total sum= 3.51083e+07
TH1.Print Name  = track_den2pt, Entries= 30749190, Total sum= 3.0694e+07
TH1.Print Name  = track_den3pt, Entries= 12361240, Total sum= 1.23061e+07
TH1.Print Name  = gamma_den1, Entries= 80832, Total sum= 80832
TH1.Print Name  = gamma_den2, Entries= 80832, Total sum= 80832
TH1.Print Name  = gamma_den3, Entries= 80832, Total sum= 80832
Total selected events=19857222
real time=3532.26
Number of MBytes read = 99011.5
Processing Rate (MB/s) = 28.0306
Job running on host - ascwrk1.hep.anl.gov finished at Mon Mar 29 21:20:47
  CDT 2010
```

Processing rate – 5620 Hz

Conclusion

- Standalone Root straight forward to use
- High performance (~5K/events per second)