PROOF tutorial
Network I/O Basics

Gerardo Ganis, CERN, PH-SFT
gerardo.ganis@cern.ch

Introduction

- Data files are usually stored in high capacity Storage Elements
  - Only a small fraction can be copied locally
- SE are accessed via network
- Network is the bottleneck today, even at 10 Gb/s
- Need efficient way to access data remotely, i.e. over the network
  - Caching
  - Selective read-ahead

TFile derivatives

- ROOT provides TFile interfaces with all major backends via protocol driven plug-in technology
- The static method

  TFile *TFile::Open(const char *url, ...)

  decodes the URL and load the appropriate plug-in based on the protocol
- It returns an instance of TFile that can used as if the file were local, but reading over the network
- Plug-ins drivers are located under

  $ROOTSYS/etc/plugins/TFile

$ROOTSYS/etc/plugins/TFile

- Current content

```
$ ls $ROOTSYS/etc/plugins/TFile
P010_TWebFile.C    P050_TGFALFile.C  P090_TSQLFile.C  P130_TAS3File.C
P020_TRFIOFile.C   P060_TChirpFile.C P100_TXNetFile.C P140_TGSFile.C
P030_TCastorFile.C P070_TAlienFile.C P110_THDFSFile.C
P040_TDCacheFile.C P080_TXMLFile.C   P120_TNetFile.C
```

- Protocol driven, e.g. P010_TWebFile.C

```
$ cat $ROOTSYS/root/etc/plugins/TFile/P010_TWebFile.C
void P010_TWebFile()
{
   gPluginMgr->AddHandler("TFile", "^http:", "TWebFile",
      "Net", "TWebFile(const char*,Option_t*)");
}
```

  - If the protocol is 'http' use TWebFile from libNet with the indicated constructor
  - See http://root.cern.ch/root/html/TWebFile.html

Example with http

- Try to open
  http://root.cern.ch/files/h1/dstarmb.root

  with new TFile …

```
root[] f = new TFile("http://root.cern.ch/files/h1/dstarmb.root")
Error in <TFile::TFile>: file files/h1/dstarmb.root does not exist
```

  with TFile::Open() …

```
root[] f = TFile::Open("http://root.cern.ch/files/h1/dstarmb.root")
root[] f->ls()
TWebFile**              http://root.cern.ch/files/h1/dstarmb.root
 TWebFile*              http://root.cern.ch/files/h1/dstarmb.root
  KEY: TTree    h42;1   dstar
```

  directly with TWebFile()

```
root[] f = new TWebFile("http://root.cern.ch/files/h1/dstarmb.root")
(class TFile*)0x1018d2400
```

The 'root://' protocol

- The 'root://' protocol was originally chosen to indicate files open via the simple 'rootd' daemon
  - The port number 1094 is officially assigned to it
- Today 'rootd' is provided for legacy only, the 'root' protocol indicates direct access via XROOTD
  - PROOF still uses rootd to access to files on the nodes
- Direct access to 'rootd' can be obtained with the protocol 'rootd://'

Basic XROOTD usage

- Basic usage of XROOTD as data server is simple
- It can be useful, for example, to analyze files that you have on lxplus from anywhere
- On lxplus, make sure you have xrootd in your path
- Localize the directories with the files to access
- Create a configuration file xrd.cf like this

```
$ cat xrd.cf
# Otherwise we cannot create the admin files
all.adminpath /tmp/ganis/
all.pidpath  /tmp/ganis/
# Allow access to the following directories in readonly mode
# (add as many as needed, one directive per directory)
all.export /afs/cern.ch/work/g/ganis/public/public/data r/o
# Better to use a dedicate port
xrd.port 51094
```

Basic XROOTD (2)

- Start XROOTD on lxplus

```
lxplus414~$ xrootd -c xrd.cf
120417 10:57:06 001 Scalla is starting. . .
Copr.  2010 Stanford University, xrd version v3.1.0
++++++ xrootd anon@lxplus414.cern.ch initialization started.
Config using configuration file xrd.cf
=====> all.adminpath /tmp/ganis/
=====> xrd.port 51094
…
------ File system server initialization completed.
------ xrootd protocol initialization completed.
------ xrootd anon@lxplus414.cern.ch:51094 initialization
completed.
```

- We need a tunnel for the firewall: on the client

```
$ ssh -f4N -L 51094:lxplus414.cern.ch:51094 lxplus.cern.ch
```

Basic XROOTD (3)

- Ready to try from the client

```
root [0] f =
TFile::Open("root://localhost:51094//afs/cern.ch/work/g/ganis/p
ublic/data/event/http/event_1.root”)
...
(class TFile*)0x1018d4c00
root [1]
```

using the address of the tunnel (localhost:51094)

- You can start XROOTD in background mode

```
lxplus414~$ xrootd -c xrd.cf -b -l ~/local/xrd/xrd.log
```

so that it keeps running

Read-ahead, Caching

- Network-based systems usually provide transfer optimizations based on read-ahead and local caching
  - For example TXNetFile, the xrootd client ...
- The problem is cache misses in random access
- You can reduce those if you know what you need next
- TTrees (next module) statistically can make very good guesses about the needed info
- That's why ROOT provides the TTreeCache ...

Copying files

- Sometimes it may be more convenient to copy locally the files, if the fit on the local disk
- TFile::Cp is a useful static tool to use in macros
- Example: copy files from the ROOT HTTP server

```
{
   TString fin, fout;
   Int_t i = 1;
   for (; i<=50; i++) {
      fin.Form("http://root.cern.ch/files/data/event_%d.root", i);
      fout.Form("event_%d.root", i);
      TFile::Cp(fin, fout);
   }
}
```

  - See macros/copyEventHttp.C
- May require a tunnel to access all files ...

Getting files via tunneling

- There is a trick to go around the firewall
- Setup the dynamic channel outside ROOT

```
~$ ssh -fN4 -D 8000 tunnelHost
```

- Tell ROOT to use the tunnel

```
root [] gEnv->SetValue("XNet.SOCKS4Port", 8000)
```

- Open (or copy) the file