Challenges in Long Term Computing Models ROOT Solutions

First Workshop on Data Preservation

And Long Term Analysis in HEP

DESY

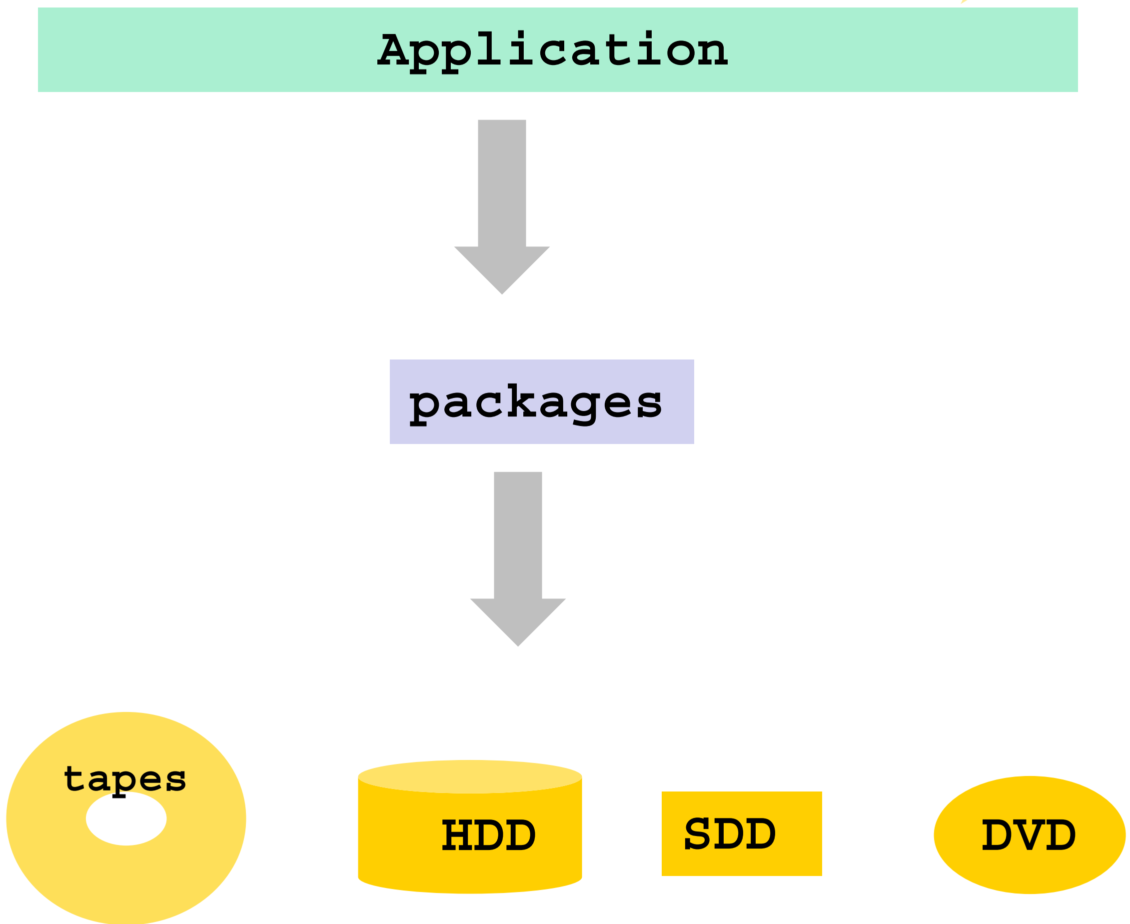
27 January 2009

René Brun

CERN

Everything changes

Atlas with more than 25000 classes



A long time ago

- With the advent of time sharing systems like VAX/VMS, then personal work stations, the request for interactive analysis systems is growing.
- Two systems emerge in 1982
- HTV at CERN (interactive histogram presenter)
- GEP at DESY (support for vectors/tuples)
- Attempt to combine these two systems in 1984 fails (GEP was written in PL1)

BOS, ZEBRA: The banks era

- Designed for Fortran aplications.
- Mainly 2 data types: ints and floats
- Could describe complex structures
- Self-described bank format (3 ints, then floats)
- Up to the application to interpret contents
- No schema evolution
- Not appropriate for languages supporting derived data types

The PAW era (1984 →)

- First version end 1984:
- No ntuples
- A command line interface KUIP
- An “abstract” interface to graphics (HIGZ) with implementations for the CORE system (US), GKS, Apollo graphics, PHIGS
- New data set format (Zebra/RZ) with support for row-wise ntuples

The PAW era (2)

- New version end 1986
- A ntuple query system (ntuple/plot)
- Column-wise ntuples
- Fortran interpreter: COMIS
- Successful system for the final steps of data analysis. Still in use today in many places.
- However, the PAW file format was not designed to support collections of large data sets.

LHC software: first steps: 1994

- First attempts to organize the software for LHC
- RD44 (Geant4) and RD45 (objy) projects
- Some gurus predict that Object-Oriented data bases will dominate the market in 2000
- Painful move to C++
- Commercial software approach (LHC++)
- Transient model = persistent model
- Central data base, single point of failure
- No schema evolution

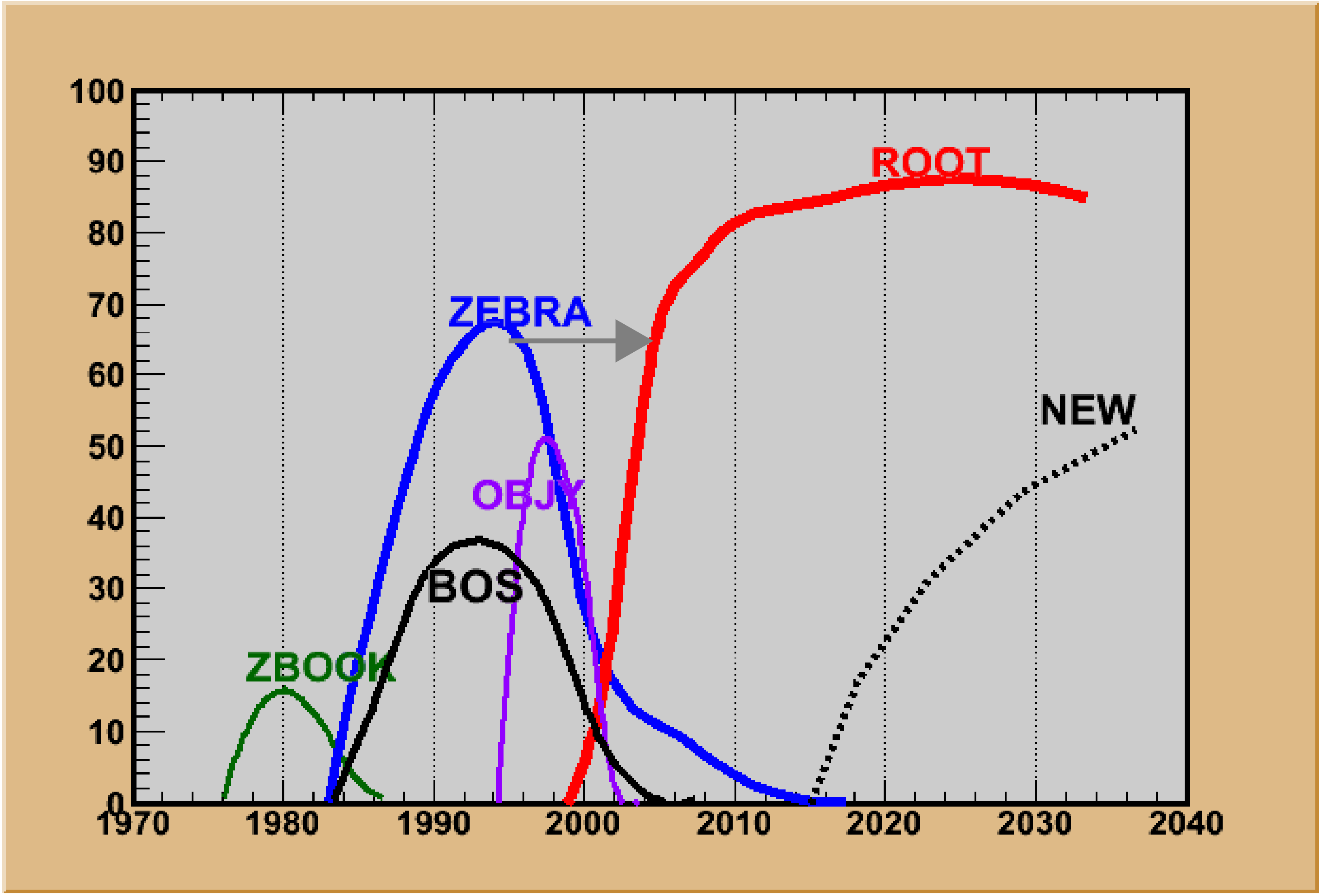
ROOT project starts: 1995

- Based on previous experience with PAW/GEANT
- Use C++ for interpreter
- Use lessons learned with PAW and ZEBRA I/O
- Design for very large data sets (mainly read-only)
- Object-Orientation; GUI, Graphics, I/O
- Quickly realize importance of object dictionaries
- From custom streamers to automatic schema evolution.
- Member-wise streaming vs object-wise streaming
- Query system very important
- Design for Parallelism and Network Access

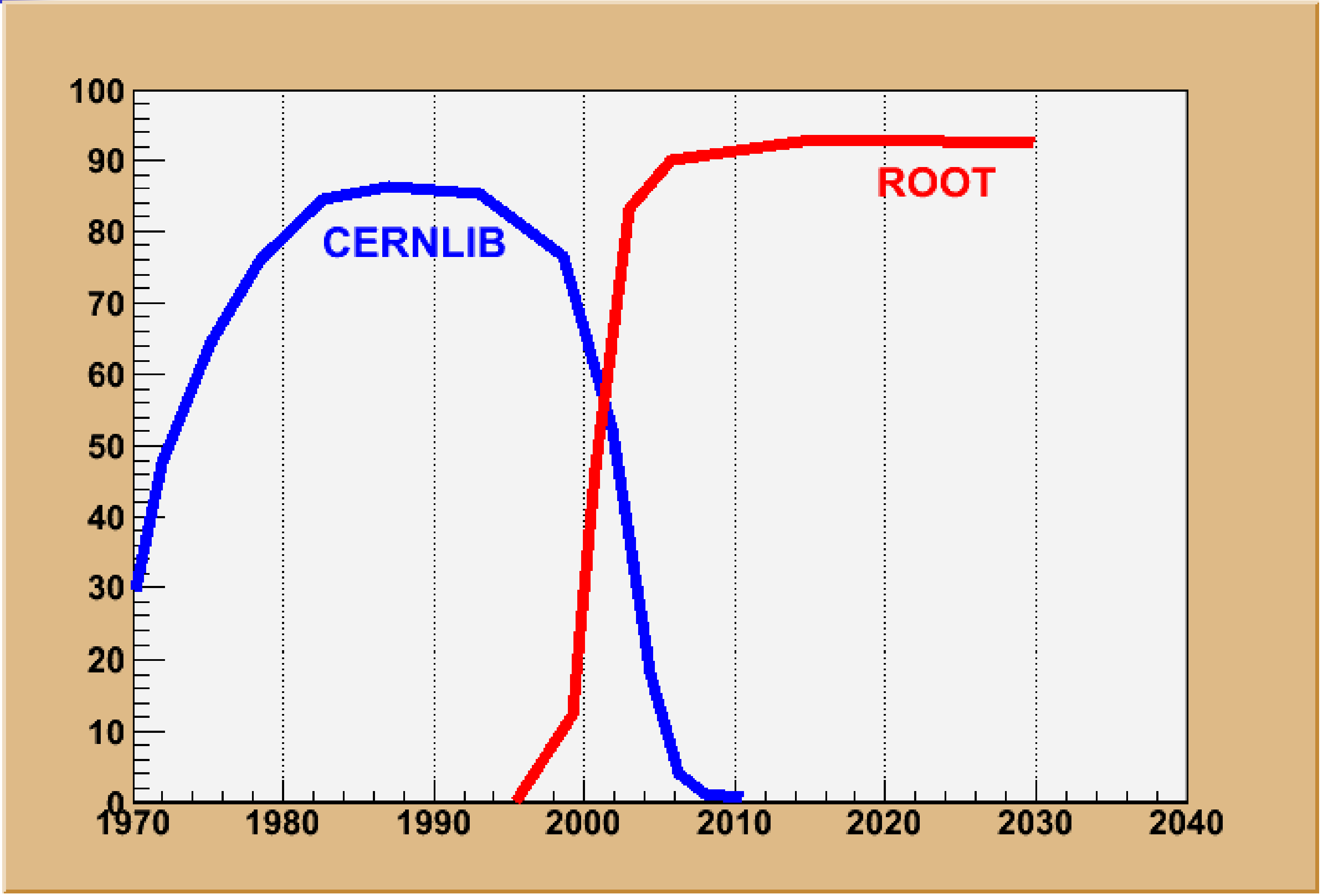
HDF, HDF5

- The Hierarchical Data Format is the standard in the astronomy, astrophysics world.
- Very good attempt to structure data sets in files
- BUT, missing the essential components to automatically build in memory complex data structures (application task!)
- No support for automatic schema evolution
- Efficient network access left to the application

I/O packages evolution in HEP



CERNLIB to ROOT



ROOT shared libs

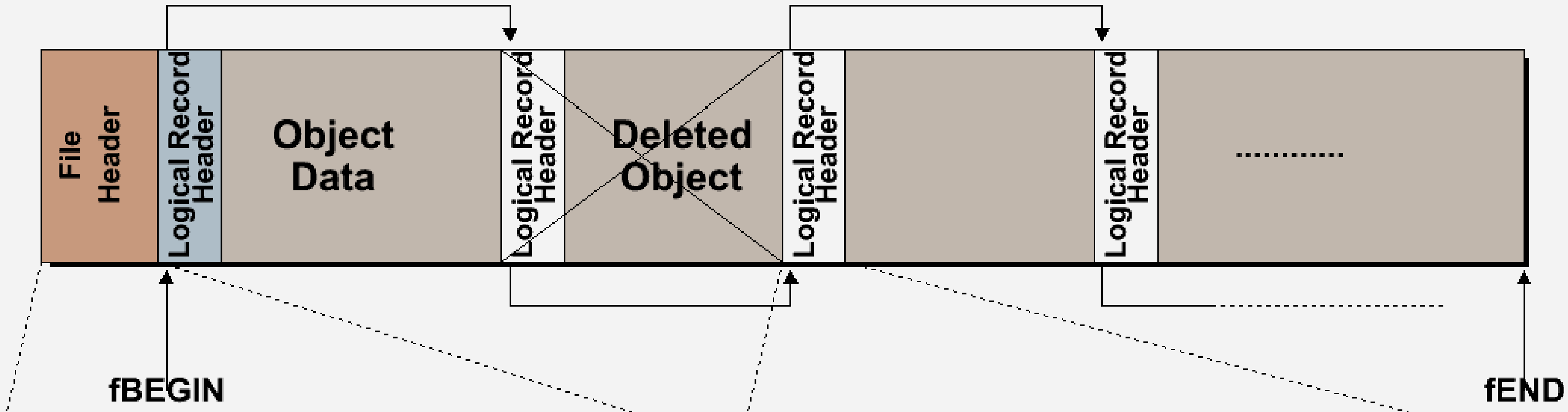
You dynamically load what you use

```
libASImage.so
libASImageGui.so
libCint.so
libCintex.so
libCore.so
libEG.so
libEGPythia6.so
libEGPythia8.so
libEve.so
libFFTW.so
libFTGL.so
libFitPanel.so
libFoam.so
libFumili.so
libGX11.so
libGX11TTF.so
libGdml.so
libGed.so
libGenVector.so
libGeom.so
libGeomBuilder.so
libGeomPainter.so
libGpad.so
libGraf.so
libGraf3d.so
libGui.so
libGuiBld.so
libGuiHtml.so
libHbook.so
libHist.so
libHistPainter.so
libHtml.so
libKrb5Auth.so
libMLP.so
libMathCore.so
libMathMore.so
libMatrix.so
libMinuit.so
libMinuit2.so
libNet.so
libNetx.so
libNew.so
libPhysics.so
libPostscript.so
libProof.so
libProofDraw.so
libProofPlayer.so
libProofx.so
libQuadp.so
libRGL.so
libRIO.so
libRLDAP.so
libRODBC.so
libRecorder.so
libReflex.so
libReflexDict.so
libRint.so
libRooFit.so
libRooFitCore.so
libRooStats.so
libRootAuth.so
libRuby.so
libSPlot.so
libSQL.so
libSessionViewer.so
libSmatrix.so
libSpectrum.so
libSpectrumPainter.so
libSrvAuth.so
libTMVA.so
libTable.so
libThread.so
libTree.so
libTreePlayer.so
libTreeViewer.so
libUnuran.so
libVMC.so
libX3d.so
libXMLIO.so
libXMLParser.so
libXrdBwm.so
libXrdClient.so
libXrdCrypto.so
libXrdCryptossl.so
libXrdOfs.so
libXrdProofd.so
libXrdRootd.so
libXrdSec.so
libXrdSecgsi.so
libXrdSecgsiGMAPLDAP.so
libXrdSeckrb5.so
libXrdSecpwd.so
libXrdSecsss.so
libXrdSecunix.so
libXrdSut.so
libdequeDict.so
liblistDict.so
libmap2Dict.so
libmapDict.so
libminicern.so
libmultimap2Dict.so
libmultimapDict.so
libmultisetDict.so
libsetDict.so
libvalarrayDict.so
libvectorDict.so
```

ROOT Files and Trees

Analysis model

ROOT File description



File Header

"root": Root File Identifier
fVersion: File version identifier
fBEGIN: Pointer to first data record
fEND: Pointer to first free word at EOF
fSeekFree: Pointer to FREE data record
fNbytesFree: Number of bytes in FREE
fNfree: Number of free data records
fNbytesName: Number of bytes in name/title
fUnits: Number of bytes for pointers
fCompress: Compression level

Logical Record Header (TKEY)

fNbytes: Length of compressed object
fVersion: Key version identifier
fObjLen: Length of uncompressed object
fDatime: Date/Time when written to store
fKeylen: Number of bytes for the key
fCycle : Cycle number
fSeekKey: Pointer to object on file
fSeekPdir: Pointer to directory on file
fClassName: class name of the object
fName: name of the object
fTitle: title of the object

TFile::Map

```
root [0] TFile *falice = TFile::Open("http://root.cern.ch/files/alice_ESDs.root")
root [1] falice->Map()
20070713/195136  At:100     N=120      TFile
20070713/195531  At:220     N=274      TH1D          CX =  7.38
20070713/195531  At:494     N=331      TH1D          CX =  2.46
20070713/195531  At:825     N=290      TH1D          CX =  2.80
….
20070713/195531  At:391047  N=1010     TH1D          CX =  3.75
Address = 392057        Nbytes = -889   =====G A P===========
20070713/195519  At:392946  N=2515     TBasket       CX = 195.48
20070713/195519  At:395461  N=23141    TBasket       CX =  1.31
20070713/195519  At:418602  N=2566     TBasket       CX = 10.40
20070713/195520  At:421168  N=2518     TBasket       CX = 195.24
20070713/195532  At:423686  N=2515     TBasket       CX = 195.48
20070713/195532  At:426201  N=2023     TBasket       CX = 15.36
20070713/195532  At:428224  N=2518     TBasket       CX = 195.24
20070713/195532  At:430742  N=375281   TTree         CX =  4.[illegible]
20070713/195532  At:806023  N=43823    TTree         CX = 1[illegible].84
20070713/195532  At:849846  N=6340     TH2F          CX = 100.63
20070713/195532  At:856186  N=951      TH1F          CX =  9.02
20070713/195532  At:857137  N=16537    StreamerInfo  CX =  3.74
20070713/195532  At:873674  N=1367     KeysList
20070713/195532  At:875041  N=1        END
root [2]
```

Classes dictionary

List of keys

TFile::ls

```
root [3] falice->ls()
  KEY: TH1D     logTRD_backfit;1
  KEY: TH1D     logTRD_refit;1
  KEY: TH1D     logTRD_clSearch;1
  KEY: TH1D     logTRD_X;1
  KEY: TH1D     logTRD_ncl;1
  KEY: TH1D     logTRD_nclTrack;1
  KEY: TH1D     logTRD_minYPos;1
  KEY: TH1D     logTRD_minYNeg;1
  KEY: TH2D     logTRD_minD;1
  KEY: TH1D     logTRD_minZ;1
  KEY: TH1D     logTRD deltaX;1
  KEY: TH1D     logTRD_xCl;1
  KEY: TH1D     logTRD_clY;1
  KEY: TH1D     logTRD_clZ;1
  KEY: TH1D     logTRD_clB;1
  KEY: TH1D     logTRD_clG;1
  KEY: TTree    esdTree;1       Tree with ESD objects
  KEY: TTree    HLTesdTree;1    Tree with HLT ESD objects
  KEY: TH2F     TOFDig_ClusMap;1
  KEY: TH1F     TOFDig_NClus;1
  KEY: TH1F     TOFDig_ClusTime;1
  KEY: TH1F     TOFDig ClusToT;1
  KEY: TH1F     TOFRec_NClusW;1
  KEY: TH1F     TOFRec_Dist;1
  KEY: TH2F     TOFDig_SigYVsP;1
  KEY: TH2F     TOFDig_SigZVsP;1
  KEY: TH2F     TOFDig_SigYVsPWin;1
  KEY: TH2F     TOFDig SigZVsPWin;1
```

TFile::MakeProject

```
(macbrun2) [253] root -l
root [0] TFile *falice = TFile::
root [1] falice->MakeProject("al
MakeProject has generated 26 cl
alice/MAKEP file has been gene
Shared lib alice/alice.so has bee
Shared lib alice/alice.so has bee
root [2] .!ls alice
AliESDCaloCluster.h      AliESDZD
AliESDCaloTrigger.h      AliESDca
AliESDEvent.h            AliESDfr
AliESDFMD.h              AliESDfr
AliESDHeader.h           AliESDki
AliESDMuonTrack.h        AliESDtr
AliESDPmdTrack.h         AliESDv0
AliESDRun.h              AliExter
AliESDTZERO.h            AliFMDFl
AliESDTrdTrack.h         AliFMDMa
AliESDVZERO.h            AliMulti
AliESDVertex.h           AliRawDa
root [3]
```

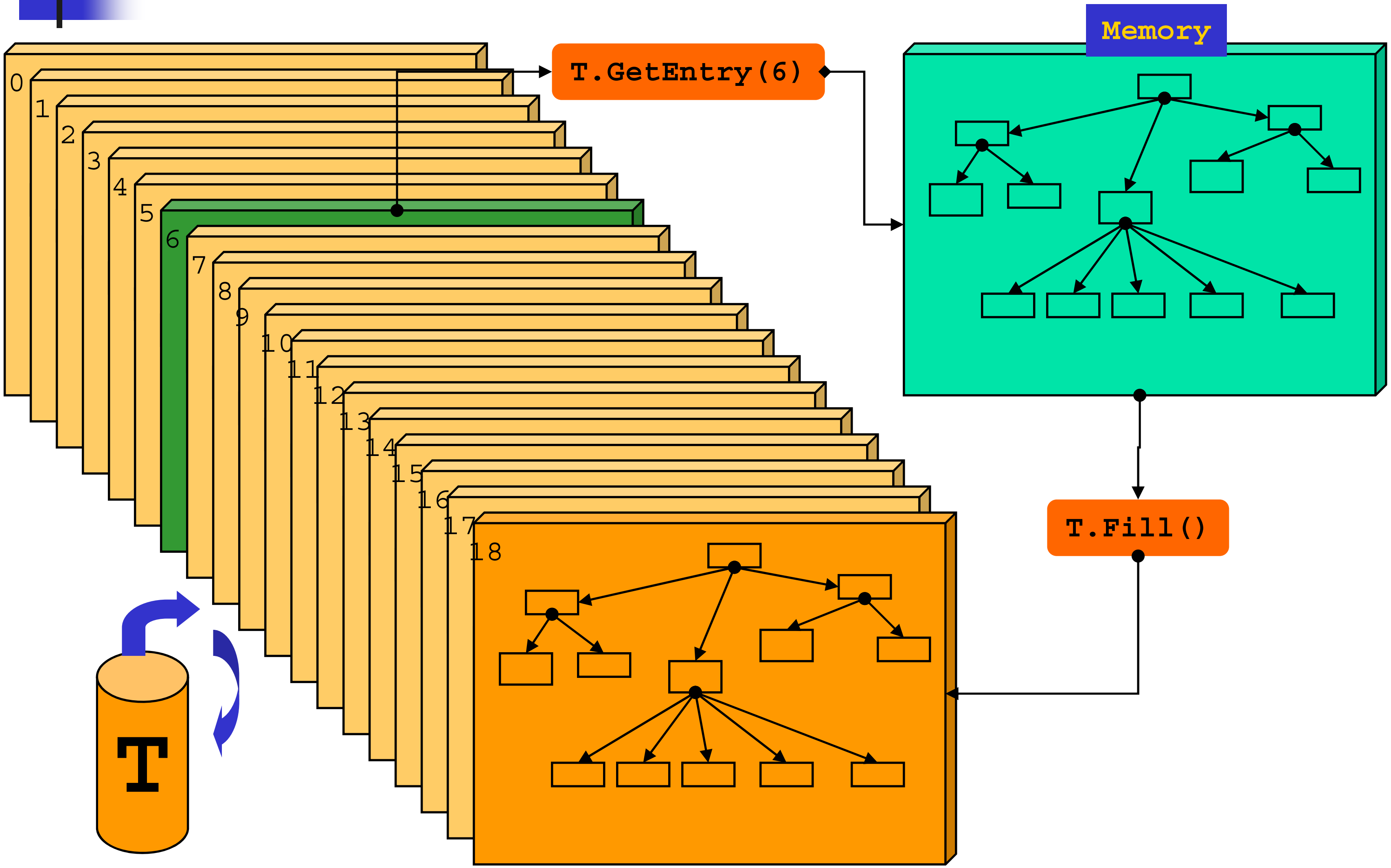
AliESDCaloCluster.h

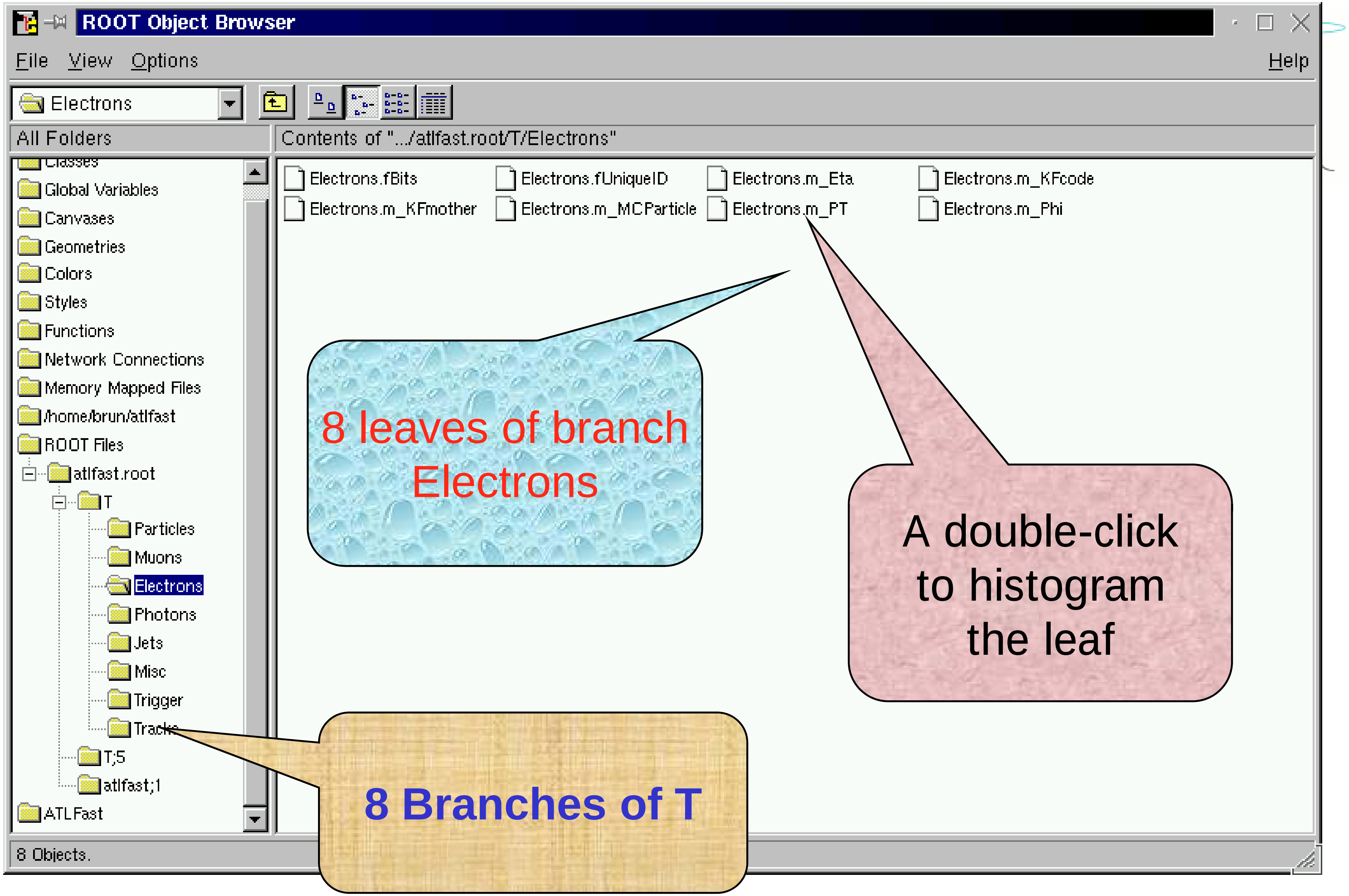
AliESDCaloCluster.h:2 <No selected symbol>

```
//////////////////////////////////////////////////////////
//   This class has been generated by TFile::MakeProject
//     (Sat Jan 24 15:24:51 2009 by ROOT version 5.23/01)
//      from the StreamerInfo in file http://root.cern.ch/files/alice_ESDs.root
//////////////////////////////////////////////////////////


#ifndef AliESDCaloCluster_h
#define AliESDCaloCluster_h
class AliESDCaloCluster;

#include "TObject.h"
#include "TArrayS.h"

class AliESDCaloCluster : public TObject {

public:
   Int_t      fID;                 //Unique Id of the cluster
   Int_t      fClusterType;        //Flag for different clustering versions
   Bool_t     fEMCALCluster;       //Is this is an EMCAL cluster?
   Bool_t     fPHOSCluster;        //Is this is a PHOS cluster?
   Float_t    fGlobalPos[3];       //position in global coordinate system
   Float_t    fEnergy;             //energy measured by calorimeter
   Float_t    fDispersion;         //cluster dispersion, for shape analysis
   Float_t    fChi2;               //chi2 of cluster fit
   Float_t    fPID[10];            //"detector response probabilities" (for the PID)
   Float_t    fM20;                //2-nd moment along the main eigen axis
   Float_t    fM02;                //2-nd moment along the second eigen axis
   Float_t    fM11;                //2-nd mixed moment Mxy
   UShort_t   fNExMax;             //number of (Ex-)maxima before unfolding
   Float_t    fEmcCpvDistance;     //the distance from PHOS EMC rec.point to the closest CPV rec.point
   Float_t    fDistToBadChannel;   //Distance to nearest bad channel
   TArrayS*   fTracksMatched;      //Index of tracks close to cluster. First entry is the most likely
   TArrayS*   fLabels;             //list of primaries that generated the cluster, ordered in deposit
   TArrayS*   fDigitAmplitude;     //digit energy (integer units)
   TArrayS*   fDigitTime;          //time of this digit (integer units)
   TArrayS*   fDigitIndex;         //calorimeter digit index

   AliESDCaloCluster();
   virtual ~AliESDCaloCluster();

   ClassDef(AliESDCaloCluster,5); // Generated by MakeProject.
};
#endif
```

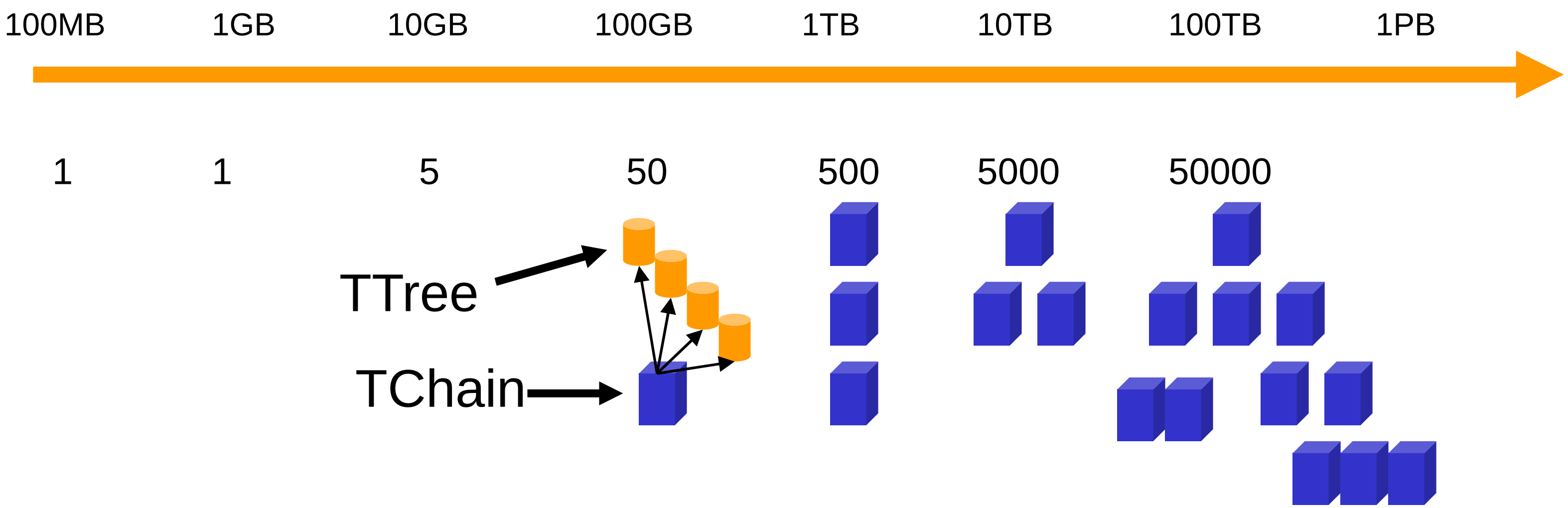
Memory <--> Tree

Each Node is a branch in the Tree




ROOT Object Browser
File View Options
Help
Electrons
All Folders
Contents of ".../atlfast.root/T/Electrons"
Global Variables
Canvases
Geometries
Colors
Styles
Functions
Network Connections
Memory Mapped Files
/home/brun/atlfast
ROOT Files
atlfast.root
T
Particles
Muons
Electrons
Photons
Jets
Misc
Trigger
T;5
atlfast;1
ATLFast
8 Objects.
Electrons.fBits
Electrons.fUniqueID
Electrons.m_Eta
Electrons.m_KFcode
Electrons.m_KFmother
Electrons.m_MCParticle
Electrons.m_PT
Electrons.m_Phi
8 leaves of branch Electrons
A double-click to histogram the leaf
8 Branches of T

Data Volume & Organisation



A TFile typically contains 1 TTree

A TChain is a collection of TTrees or/and TChains

A TChain is typically the result of a query to the file catalogue

Access Transparency

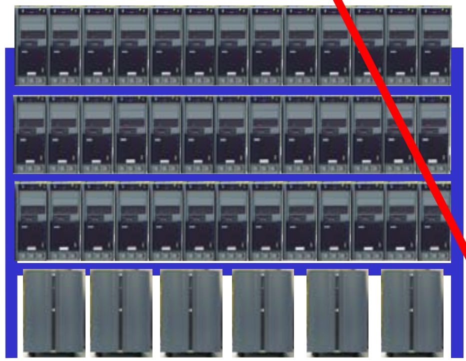
```
TFile *f1 = TFile::Open("local.root")

TFile *f2 = TFile::Open("root://cdfsga.fnal.gov/bigfile.root")

TFile *f3 = TFile::Open("rfio:/castor.cern.ch/alice/aap.root")

TFile *f4 = TFile::Open("dcache://main.desy.de/h1/run2001.root")

TFile *f5 = TFile::Open("chirp://hep.wisc.edu/data1.root")

TFile *f5 = TFile::Open("http://root.cern.ch/geom/atlas.root")
```

From the desktop to the GRID
Parallelism at all levels

Desktop

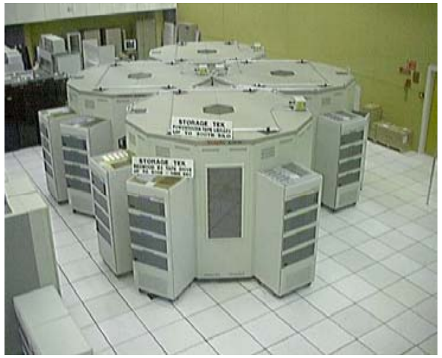
Online/Offline

Farms

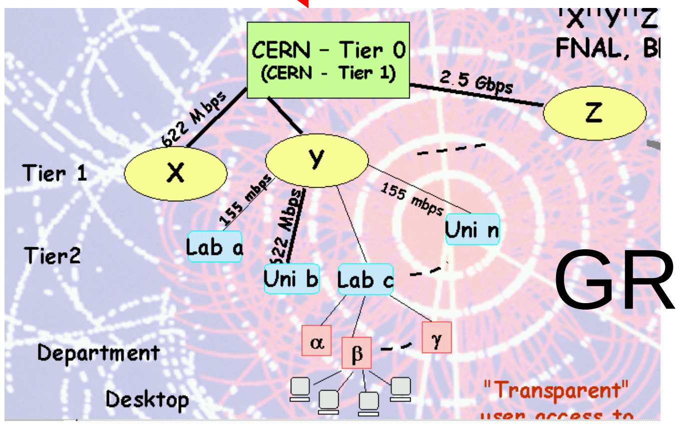


Local/remote

Storage



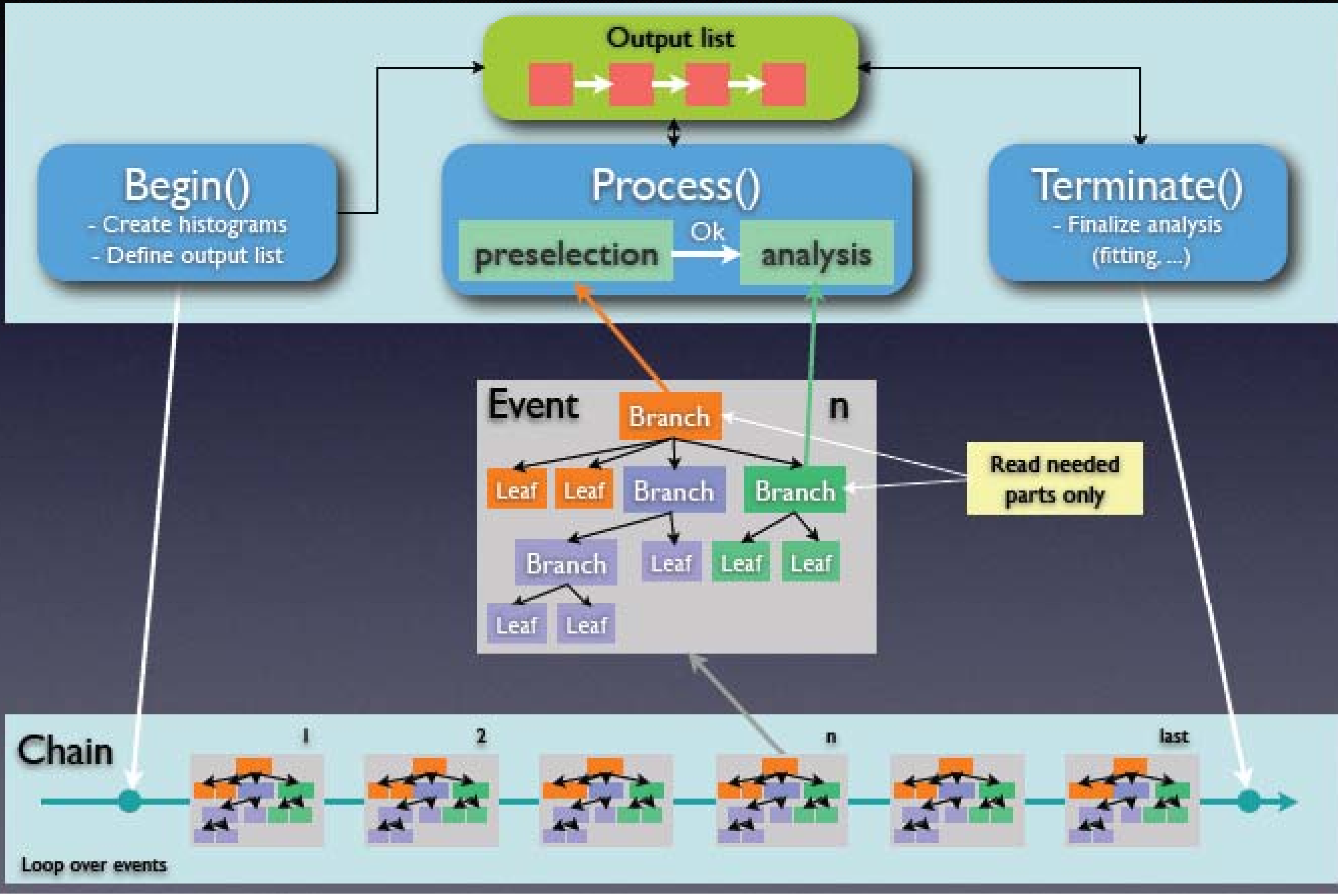
Data analysis tools must be able to exploit parallelism on the laptop, use remote CPUS in parallel, and storage elements and networks in a transparent way



GRID

The ROOT Data Model

Trees & Selectors

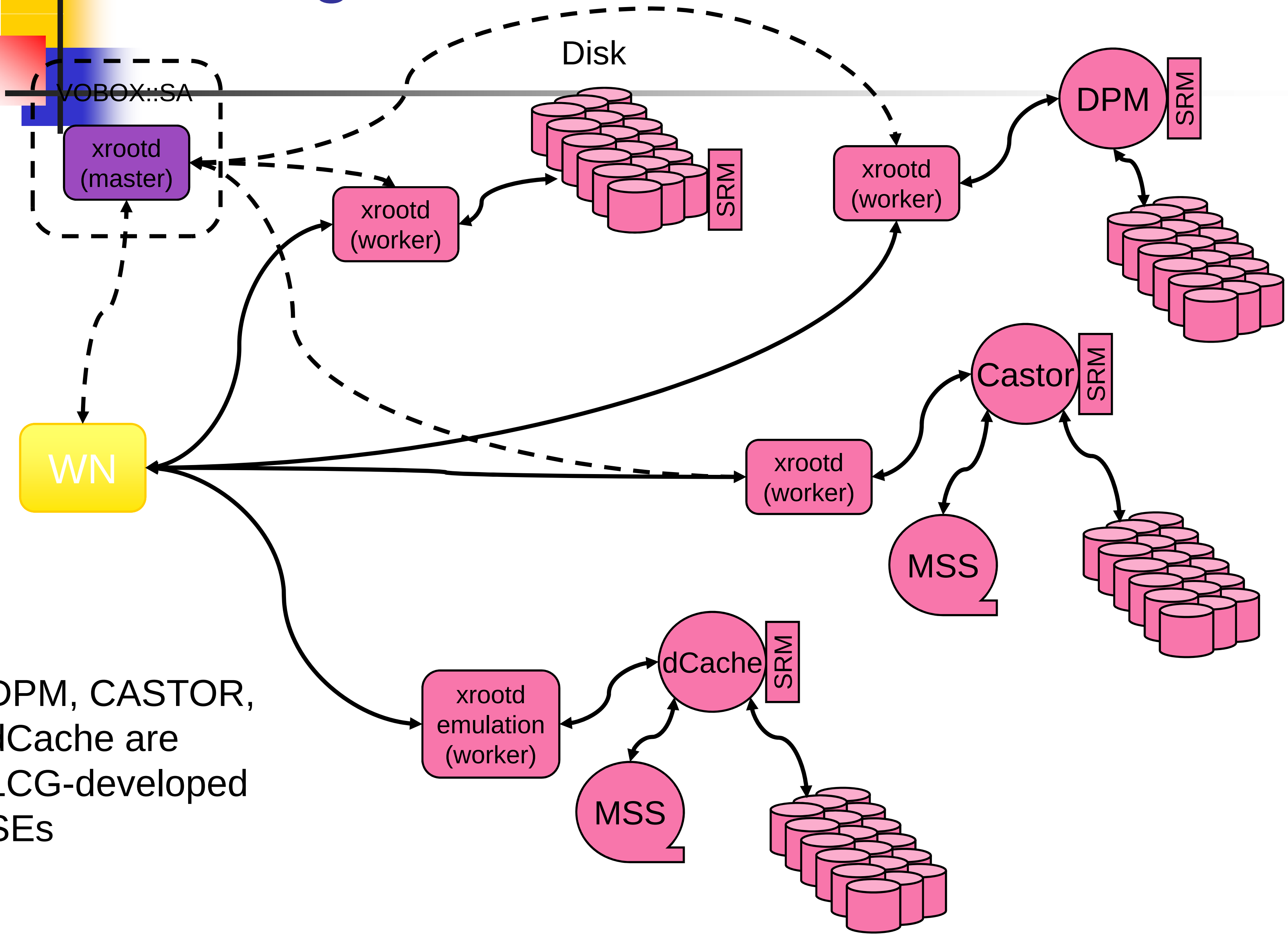


TSelector: Our recommended analysis skeleton

- Classes derived from TSelector can run locally and in PROOF

- **Begin()** — once on your client
  - **SlaveBegin()** — once on each slave
    - **Init(TTree* tree)** — for each tree
      - **Process(Long64_t entry)** — for each event
  - **SlaveTerminate()**
- **Terminate()**

Storage element

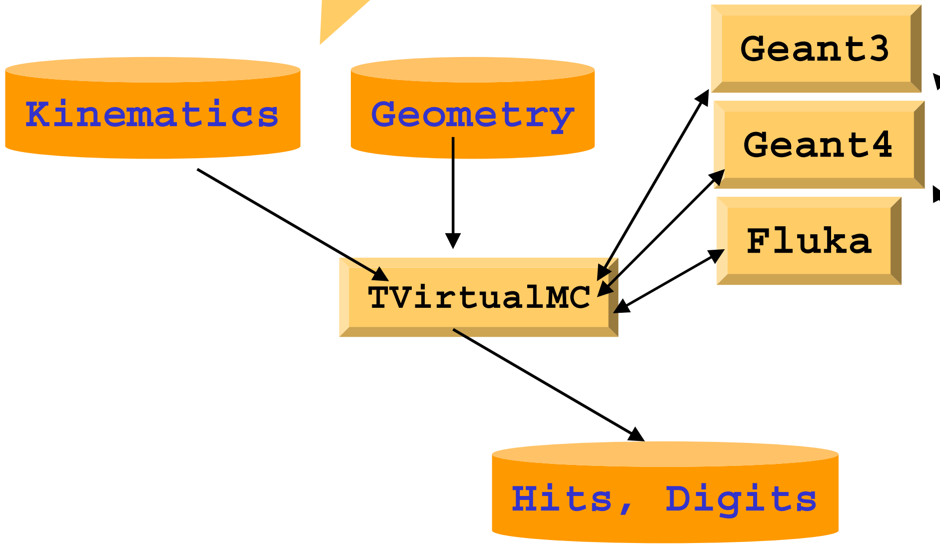


DPM, CASTOR, dCache are LCG-developed SEs

Virtual Monte Carlo

This strategy facilitates migration or comparisons with a common input and a common output

ROOT can provide a solid base for: geometry, visualisation, interactivity, interpreter and persistency



Dectector Simulation

Geant3.tar.gz includes an upgraded Geant3 with a C++ interface

Geant4_mc.tar.gz includes the TVirtualMC <-->Geant4 interface classes

Current developments

CINT Interpreter

- replace important components by LLVM, a Just In Time compiler (JIT)
- LLVM is gcc compliant compiler(Open source project, sponsor Apple)
- the interpreter is the compiler
- this will introduce a more robust C++ parser (C++0x compliant)
- JIT will provide an excellent replacement for the internals of TFormula,
- TTreeFormula (code generated and compiled in a few milliseconds)

PROOF, xrootd

- PROOF
  - -in production in Alice
  - -under evaluation by groups in Atlas, CMS
  - -close cooperation with the xrootd developers
  - -prototyping multi-core, SDD, HDD, zfs/Lustre, xrootd
- PROOF-LITE
  - -mini version of PROOF designed for multi-core laptops
  - -will become default-ROOT (automatic parallelism)
- XROOTD
  - -important developments, robustness
  - -caching and proxies wor LAN and WAN files

Fitting, Stats

- ROOFIT
- -many developments by Wouter Verkerke (now Atlas)

- ROOSTATS
- -common project Atlas, CMS, etc for a statistics library
- -in very active development
- -based on ROOFIT

We recommend

- Make your event model as simple as possible
- do not over-use the system (eg Atlas, CMS produce Trees with >6000 branches!)
- do not make derivations of the main ROOT objects like TH1, TTree. You must be able to browse your files with standard ROOT and without your classes.
- use TSelector as analysis model and not your own event loop
- think parallelism

```
for(int event=0;event<nevents;event++) {
   T->GetEvent(event);
   //do some analysis
}
```

```
T.Process(mySelector)
```

ROOT :Stability, Robustness

- Substantial manpower effort
  - 10 FTE at CERN + 1.5 at FNAL
- Support in place for the LHC era
- Still many developments
  - I/O robustness for LHC is vital
  - I/O performance improvements
- Micro and macro parallelism : a must

Doc + web site

- New ROOT web site will be introduced end of February
- (see snapshot at http://root.cern.ch/drupal)
- Better introduction for beginners
- Better tutorials
- Better look & feel