BERKELEY LAB

NERSC

NATIONAL ENERGY RESEARCH
SCIENTIFIC COMPUTING CENTER

National Energy Research Scientific Computing Center (NERSC)

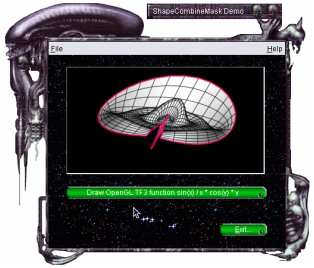
ROOT 2007 Report

Wim Lavrijsen

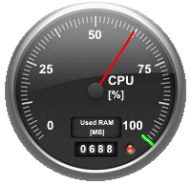
HENPC Group Meeting – LBNL 04/06/07

Office of Science
U.S. DEPARTMENT OF ENERGY

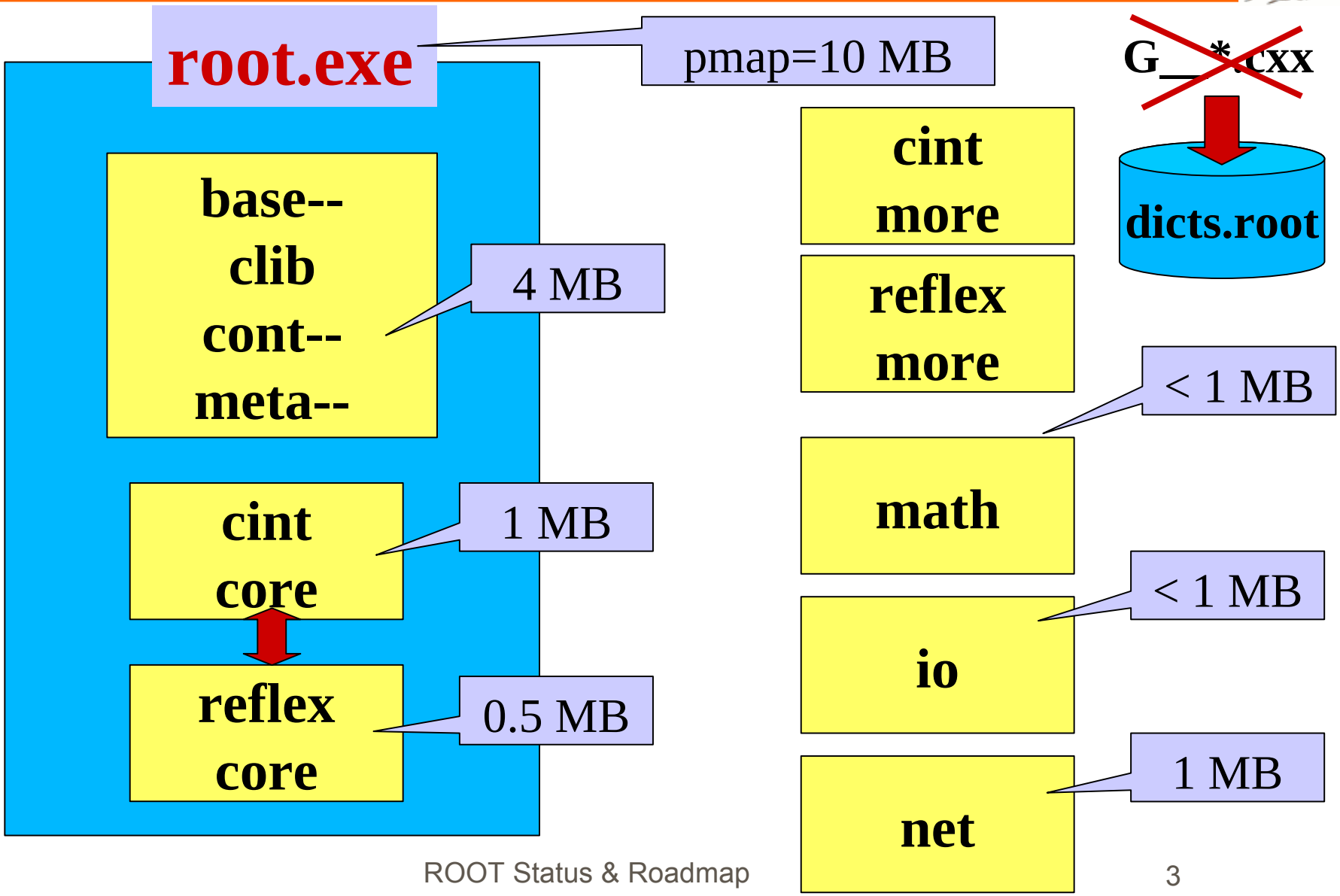
Overall Impression



- **Consolidation**
  - Library/Headers re-organization
    - Dependencies and configure cleanup
  - I/O, Math, Base, Meta, Plugin Mgr
    - Now done with SEAL
  - Reflex, Cint7, dictionary (memory) sizes
    - First work towards FFI/portable dicts
- **Cutification**
  - 8 Talks on GUI/Graphics/Event Display
- **Focus on analysis a little incoherent**
  - Way more on features than process



Libraries reorganization
(proposal phase2 == BOOT phase1)



SEAL Migration

- Last remaining item to be migrated was the SEAL plugin service
- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory declared via one macro in the user code
  - Load on demand via rootmap files (genmap)

libRIO

- New library containing all the ROOT classes to do basic Input/Output (ROOT 5.15/04 and above)
  - Includes TFile, TKey, TBufferFile, the Collection Proxies (for STL), etc.
  - TDirectory, TBuffer are now a pure abstract interface.
  - TDirectoryFile, TBufferFile are the concrete implementation
  - TFile derives from TDirectoryFile instead of TDirectory.
  - These change may be backward incompatible:
    - If you creates a TDirectory object, replace with TDirectoryFile
    - If you creates a TBuffer object, replace with TBufferFile
  - Dictionaries are NO longer dependent on any of the classes in libRIO

```
#if ROOT_VERSION_CODE >= ROOT_VERSION(5,15,0)
#include <TBufferFile.h>
#else
#include <TBuffer.h>
#endif
#if ROOT_VERSION_CODE >= ROOT_VERSION(5,15,0)
    TBufferFile b(TBuffer::kWrite,10000);
#else
    TBuffer b(TBuffer::kWrite,10000);
#endif
```

Autoderefencing

- TRef and TRefArray are now auto-dereferenced when used in TTree::Draw
  - Requires to enable TRef autoloading (by calling TTree::BranchRef)
  - For collections of references either a specific element of the collections may be specified or the entire collection may be scanned. (example 2.)
- Same framework can be used for any *Reference* classes (eg. POOL Ref)

- The TTreeFormula operator @ applied to a reference object allows to access internals of the reference object itself (example 3.)
- The dereference mechanism even works across distributed files (if supported by the reference type) (example 4.)
- Caveat for TRefArray and TRef:
  - To know the underlying type, the first entry of the TTree is read.
- Special Thanks To Markus Frank
- Special Thanks to DZero for testing the limits of the TRef mechanism

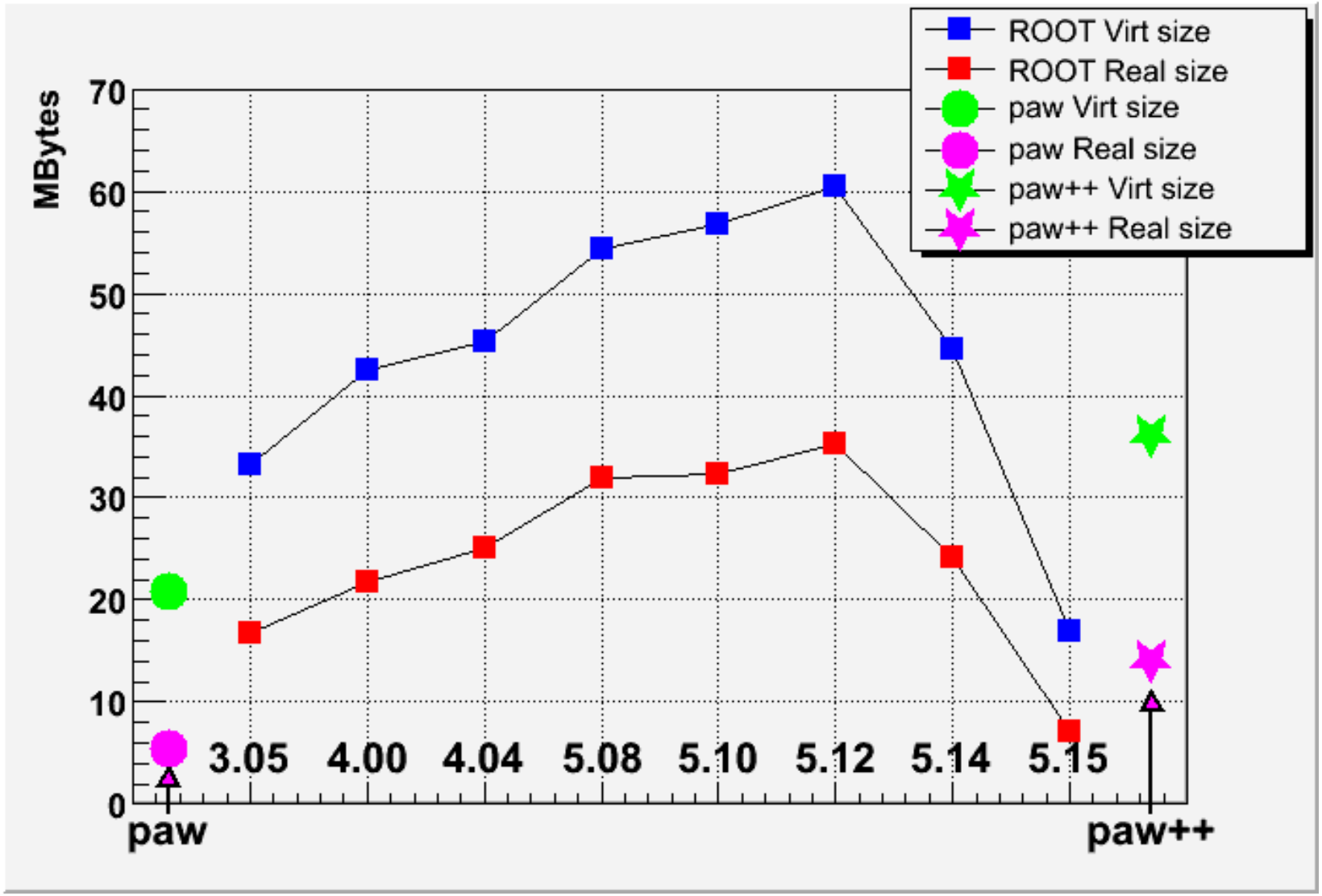
```
1: T->Scan("fLastTrack.GetPx()");
2: T->Scan("fMuons.GetPx():fMuons[0].GetPx()”);
3: T->Scan("fLastTrack.GetUniqueID():fLastTrack@.GetUniqueID()&0xFFFF");
4: T->Scan("fWebHistogram->GetRMS()");
```

Upcoming Features

- Continue Consolidations ☺

- Splitting STL collection of pointers
- Schema Evolution
  - To and From more combinations of containers (Double32 vs. double, ROOT containers).
- MakeProxy
  - Add support for STL containers
  - Add support for CINT-interpretation
- TTree
  - Indexing using bitmap algorithm (TBitMapIndex) from LBL.
  - TTree::Draw performance

Where are we today with 5.15/04 ?



More Impressions

- **Several dual-distributed packages**
  - xrootd, RooFit, TMVA
  - May want similar model for PyROOT ...
- **Experiment talks on AODs:**
  - Same, same, only differ in restrictions
    - Babar for President!
  - Even ALICE wants analysis w/o AliROOT
- **Little feedback from user community**
  - No heated discussion, low attendance
    - Note: ATLAS SW Week in Munich at same time

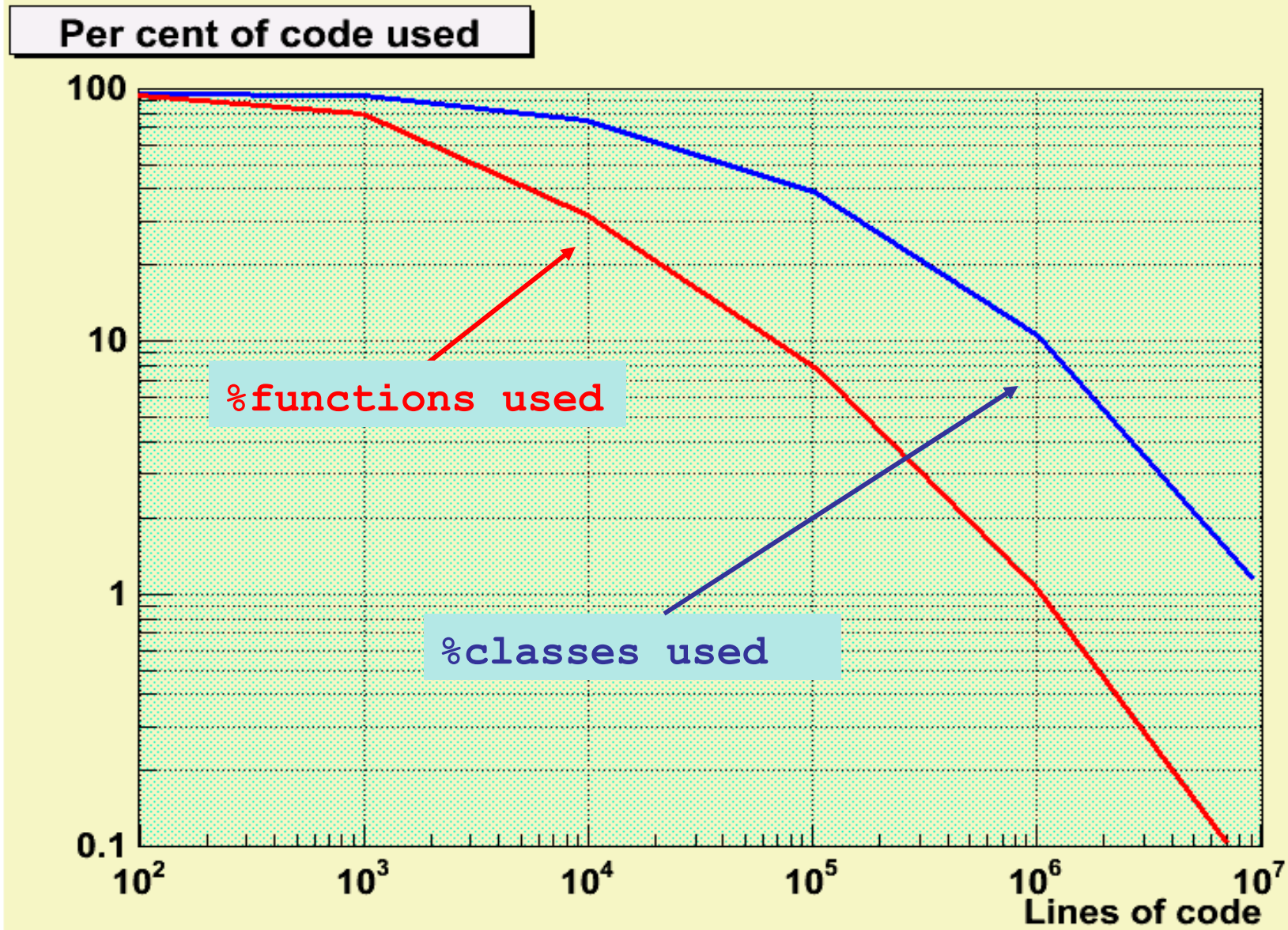
Next Big Thing(s)

- BOOT (introduced at CHEP'06)
  - Dynamic downloading/building of needed functionality only
  - On the fly compilation
- Multi-core and threading
  - Proof-based, interpreter support
    - So it'll mean work for me, too ...
  - Thread safety
  - Shared memory (?)
  - Nothing concrete yet, AFAICS

Some LHC parameters collected 1 year ago

	Alice	Atlas	CMS	ROOT
number of lines in header files	102282	698208	104923	153775
classes total	1815	8910	???	1500
classes in dict	1669	>4120 2140	835	1422
lines in dict	479849	455705	103057	698000
classes c++ lines	577882	1524866	277923	857390
total lines Classes+dict	1057731	???	380980	1553390
total f77 lines	736751	928574	???	3000
directories	540	19522	<500	958
comp time	25'	750'	90'	30'
lines compiled/s	1196	50 (70)	71	863

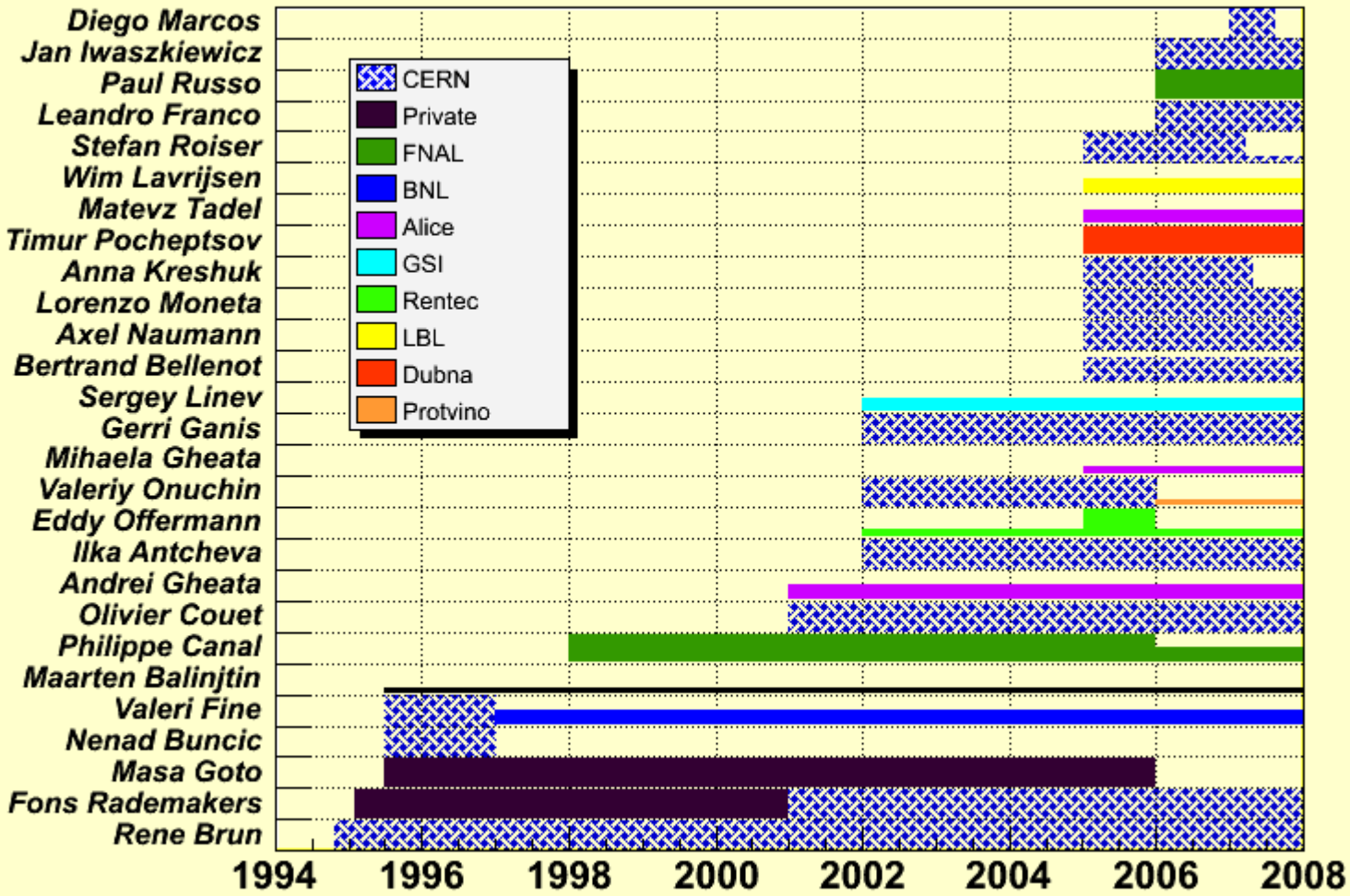
Fraction of code really used in one program



What is BOOT?

- Start from the source URL whenever possible
- But an option to download only a subset of the libs could be implemented too.
- Compile into local cache and reuse at next session.
- A tool is provided to convert a CVS source tree into a compact file that also includes the dictionary data structures and the classes/functions documentation.
- Compile with the best options for the local hardware.
- A front-end to C++ (with the corresponding C++ code generation) for very frequent use cases.

ROOT Team Evolution



ROOT Team today
(working 50% or more)

Interesting talks about all these topics

- **CORE**: Fons Rademakers, Leo Franco, Diego Marcos
- **DICT**: Axel Naumann, Philippe Canal, (Stefan Roiser)
- **I/O**: Philippe Canal(50%), Paul Russo
- **MATH**: Lorenzo Moneta, (Anna Kreshuk)
- **GEOM**: Andrei Gheata, Mihaela Gheata
- **GUI**: Ilka Antcheva, Bertrand Bellenot, (V.Onuchin(yy%))
- **GRAF**: Olivier Couet, Timur Potcheptsov, Matev Tadel(50%)
- **PROOF**: Fons, Gerri Ganis, Jan Iwaszkiewicz
- **PYROOT**: Wim Lavrijsen (xx%)

Resource

http://root.cern.ch/root/R2007/Welcome.html