

ROOT 6

Axel Naumann

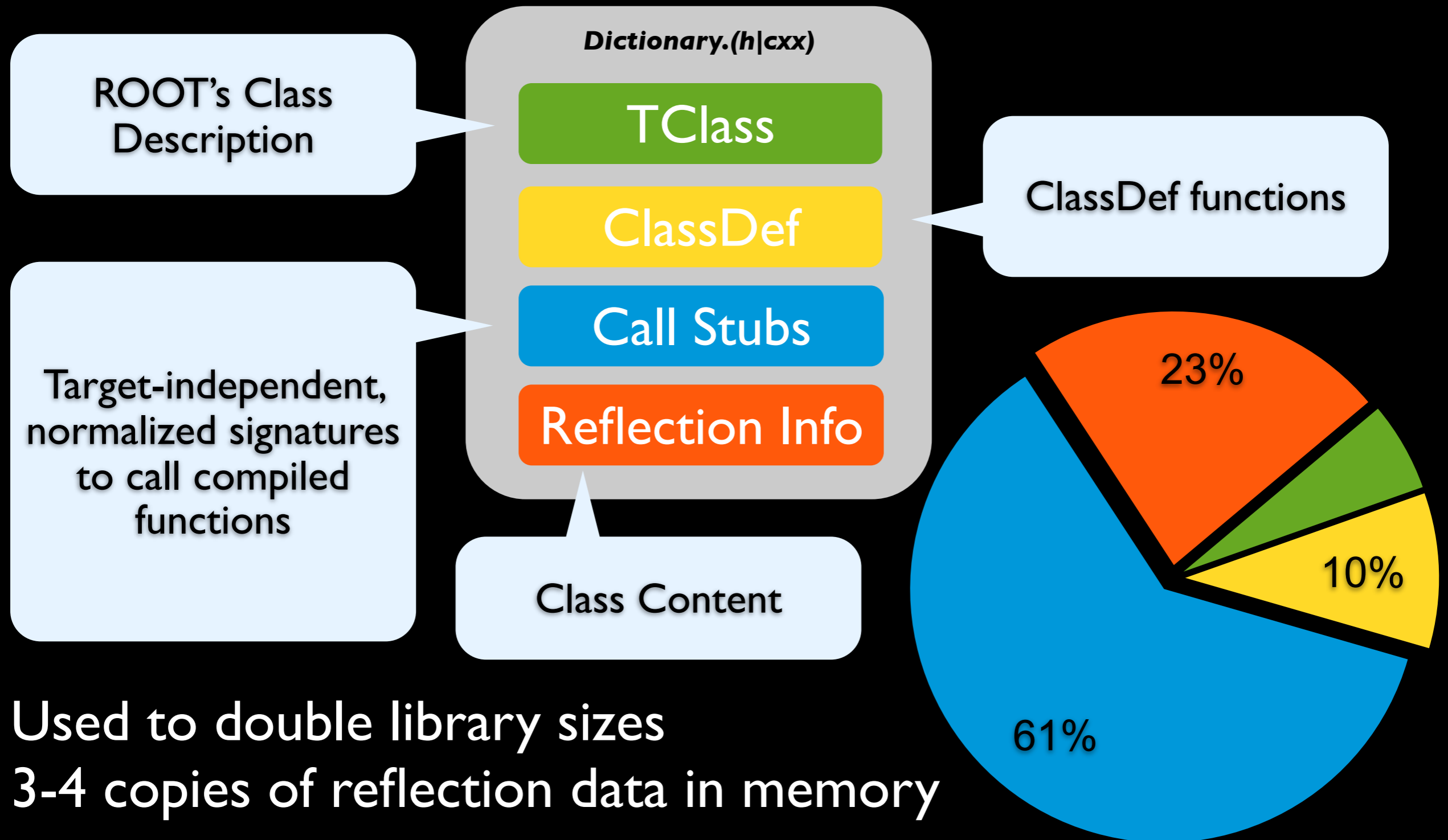
ROOT6

- Replace CINT by cling, libLLVM / libClang-based interpreter
 - reduce maintenance load
 - inherit feature set, correctness
- Backward compatible
 - exception: improved TClass instead of Reflex, Cintex

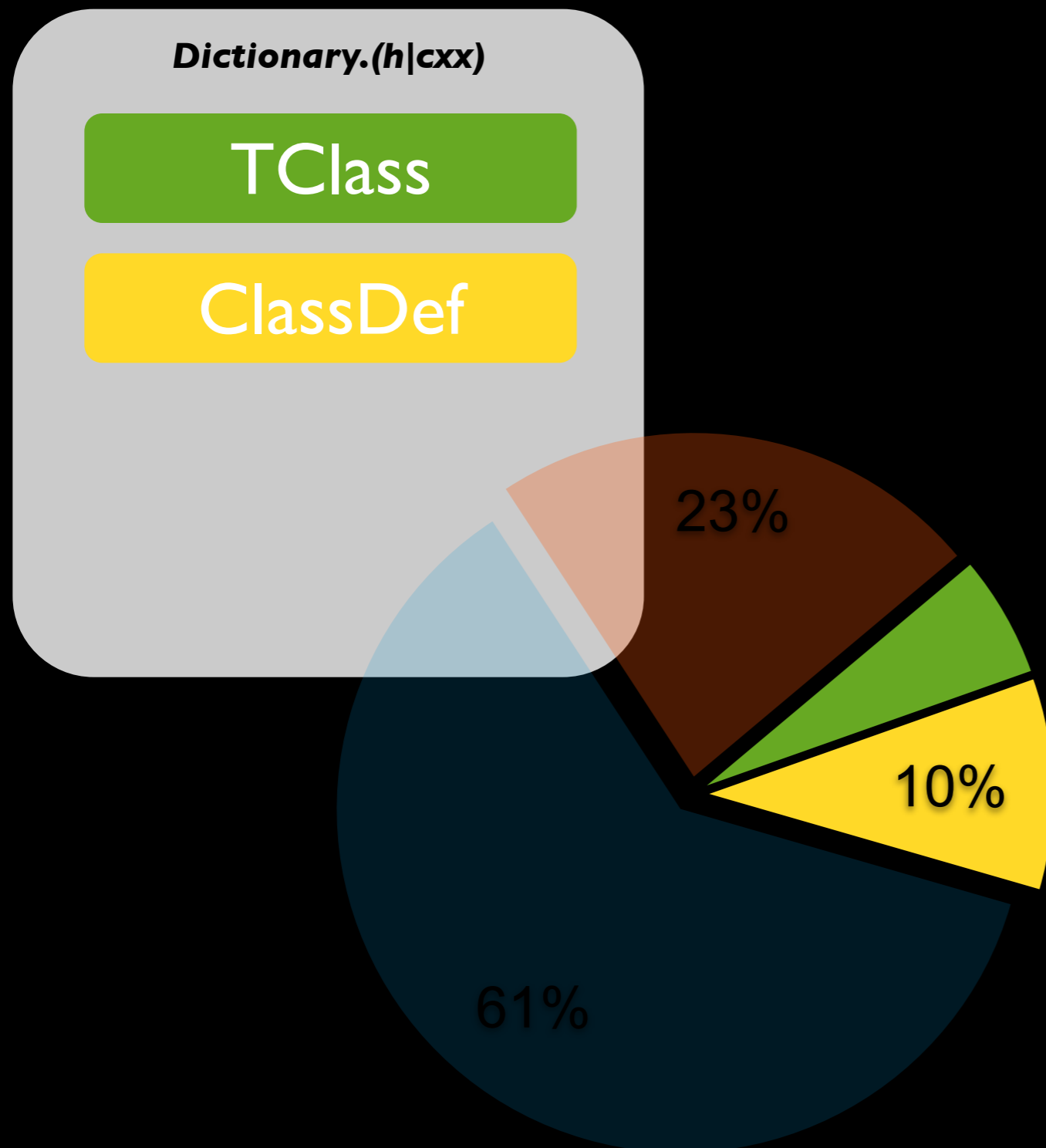
Dictionaries

- Type information now provided by clang
- Use clang for persistent format
 - precompiled headers (PCH)
 - or modules (PCM)
- Considerably reducing dictionary size

ROOT 5 Dictionaries



ROOT 6 Dictionaries



- Reflection data in PCH or PCM, 26MB PCH for all of ROOT (100MB libs)
- JIT for function calls
- on-demand template instantiation

ACLIC

- Remains
- Does exactly what it used to do:
 - calls your compiler
 - builds shared library
- Dictionary is of course ROOT6 dictionary

cling

- C++11 cannot be implemented in CINT
 - would need to re-write from scratch
 - implausible
- We must move, there is no alternative.

Status

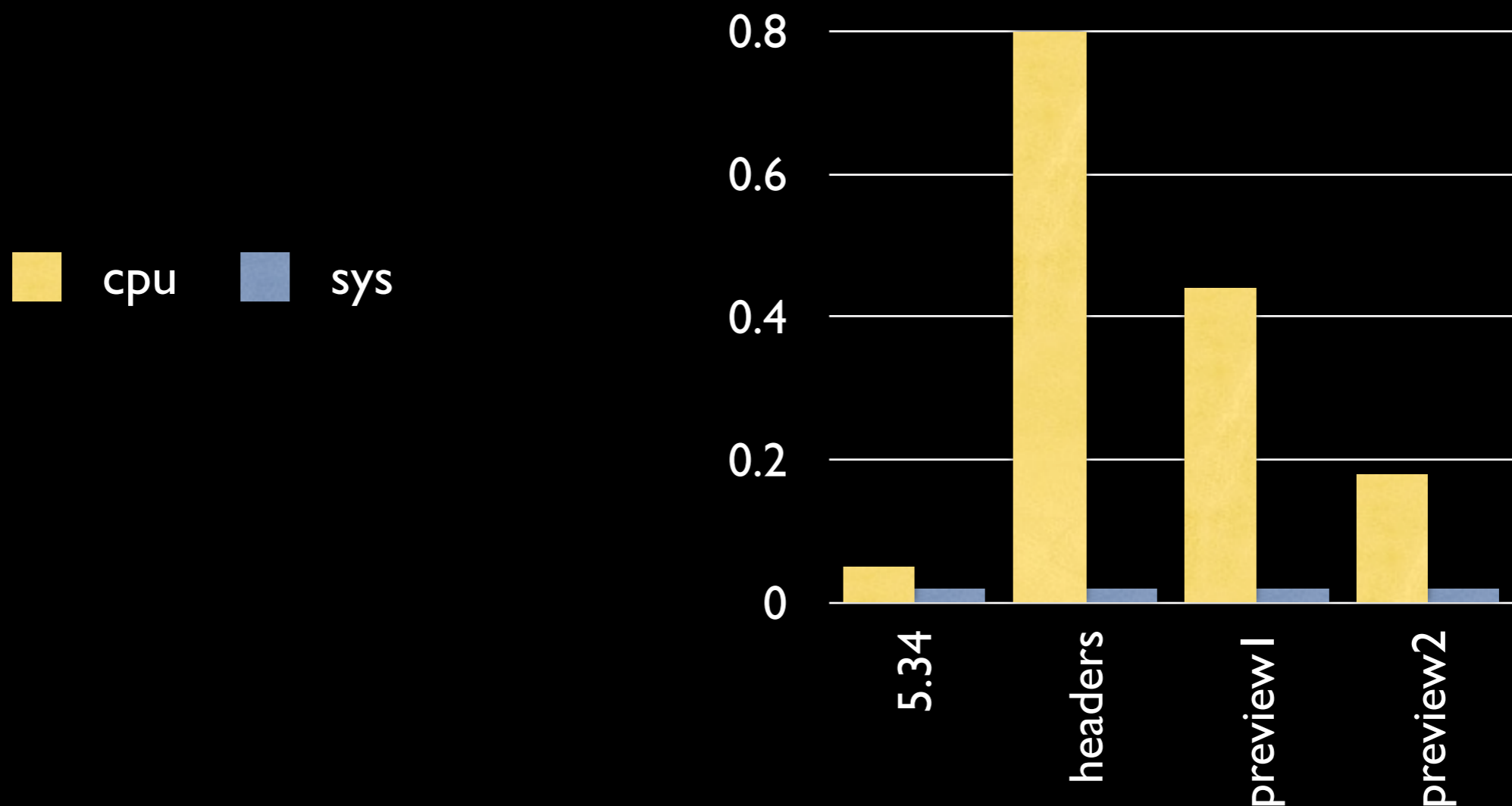
- C++ interpreter cling works since 2011
- CINT, Reflex, Cintex replaced by cling in ROOT trunk (aka ROOT 5.99)
- Preview1 released in March (ROOT Users Workshop)
- Preview2 released this week:
+autoloading, +usability, +stability

Rene's memory.C

- Prints ROOT proc's values after startup
 - CPU time: sys, user
 - RAM usage: resident, virtual
- No compute loop
 - just-in-time compiler / cling speed not seen

Performance: CPU

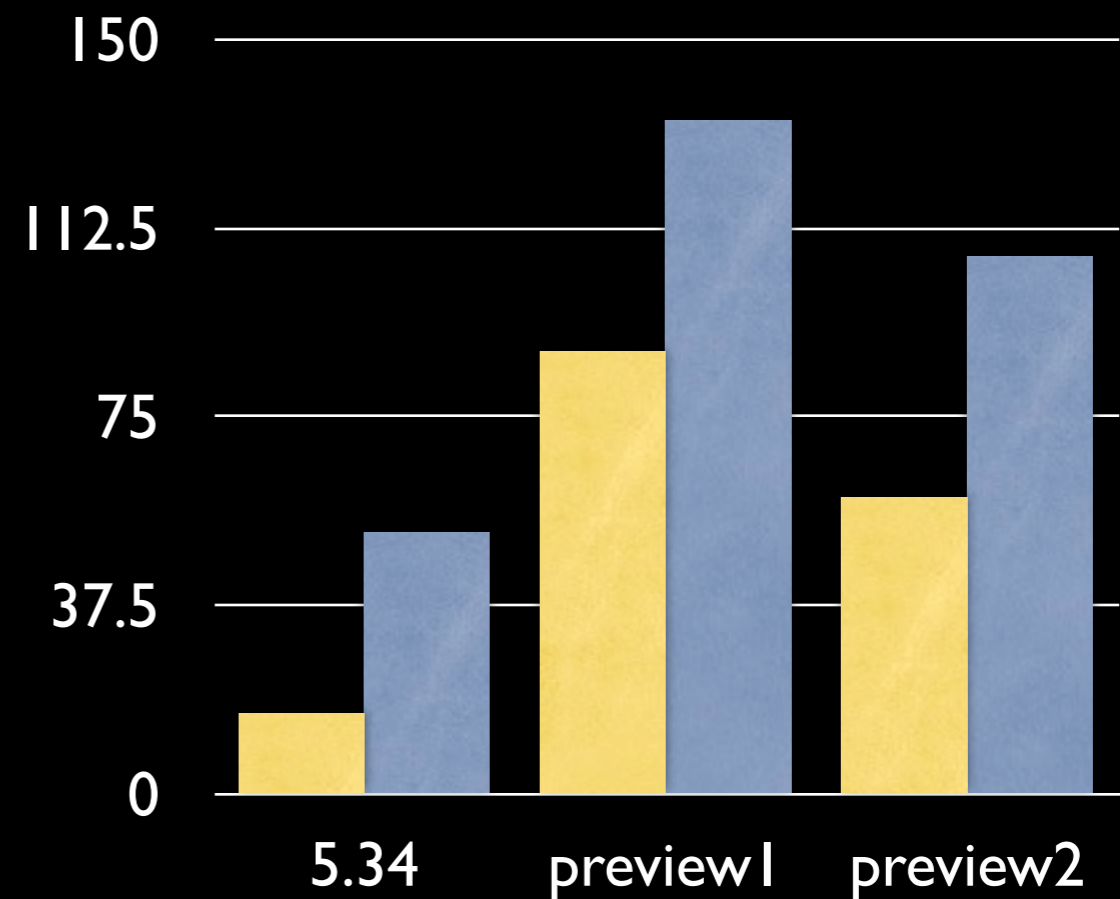
- `root -l -q memory.C [seconds]`



Performance: RAM

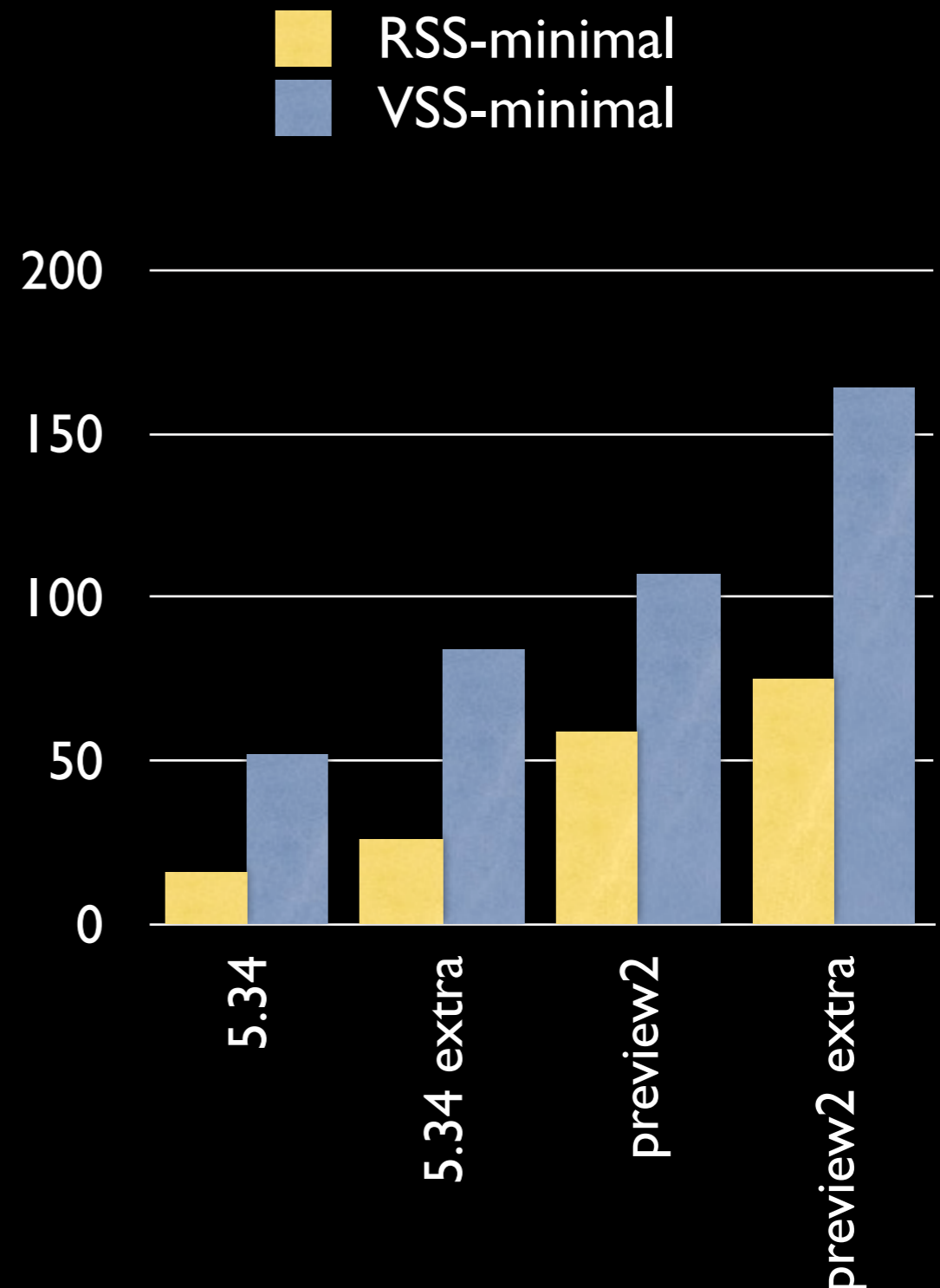
- `root -l -q memory.C [MB]`
- `libllvm + libclang:`
70 MB

■ RSS ■ VSS



Performance: RAM (2)

- basic: minimal libraries
- extra: loading GUI, Gpad, RIO, Net, Hist, Graf, Tree, Graf3d, Postscript, Matrix, Physics, Thread



Performance

- Have not spent time on performance but features
- Many obvious improvements
 - e.g. delay loading from PCH/PCM

Platforms

- Linux 32bit + 64bit
- MacOS
- ARM 32bit + 64bit (llvm; ROOT:TBC)
- PPC 32bit + 64bit (llvm; ROOT:TBC)
- NOT Windows: clang not there yet
 - Bertrand Bellenot tracks clang's progress

Current Features

- ROOT I/O works
- Most CINT features implemented in cling
 - autoloading of libraries and headers
 - automatic variables: `c = new TCanvas();`
 - `new TFile(...); hist->Draw()`

Feature Tracking

- Roottest 1513 logs (steps) in 506 tests
 - 129 occurrences of 28 missing features
- Jira, 102 issues open, 160 done
- Trello (technical), 63 open, 105 done
 - <http://cern.ch/go/Mh8G>
- Google Doc for planning

ROOT 6.0 Features

Required Features

- PCM
- Unloading: `.x code.C; .x code.C`

PCM with rootcling / genreflex

- Need to inform libclang about types
 - by parsing headers at runtime, or
 - by using PCH, or
 - by using PCM
- C++ calls to declare types to libclang equivalent to modules, but inefficient

PCH versus PCM

- Can only have one PCH but multiple PCMs
- PCM replaces #include
 - much more flexible
 - frameworks pull in ROOT, G4, ...



PCH versus PCM

- We want PCM.

PCM in the World

- We want PCM.
- And so does Apple: <http://llvm.org/devmtg/2012-11/Gregor-Modules.pdf>
- Dedicated Google (!) developer since April
 - from 6 to 20 commits / month!
- C++ Standards Committee Study Group

Remaining PCM Challenges

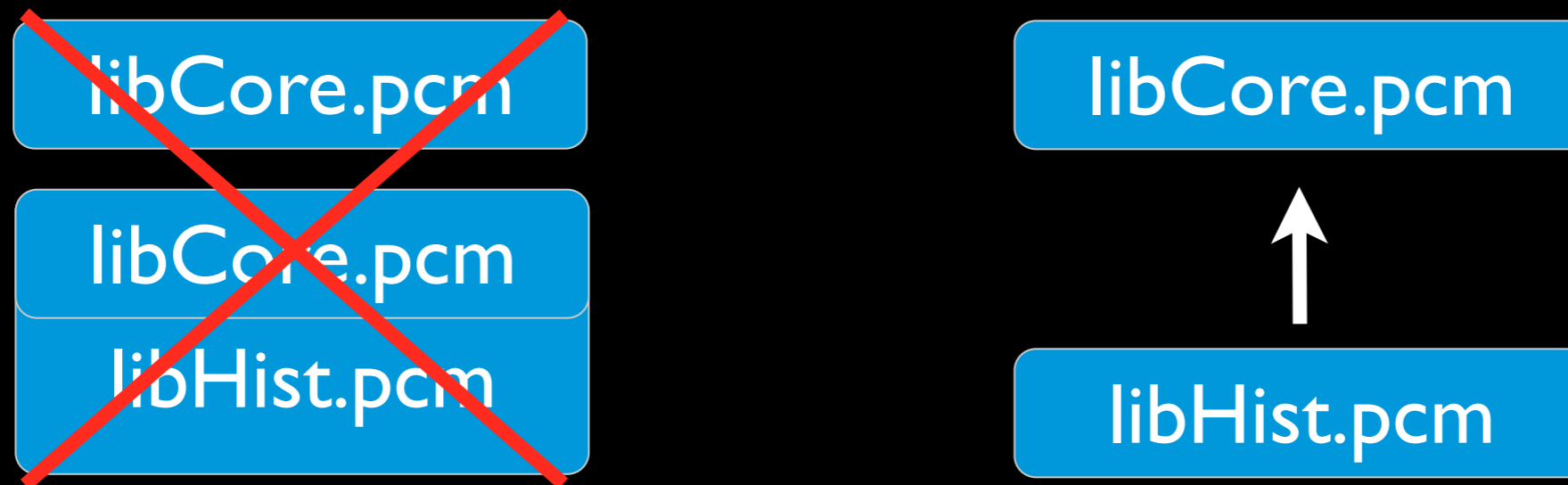
- Merging of identical definitions from multiple modules
 - `class TObject{} * 100` PCMs
- Can be worked around / reduced by using `libCore.pcm` for any `#include "TObject.h"`
- tiered PCMs: `libHist.pcm` uses `libCore.pcm` and `libMatrix.pcm`

PCM - Delivery?

- Combined effort
 - Apple: clang lead dev
 - Google: famous compiler expert
 - CERN: Axel
- No schedule, though
- And thus have backup plan

PCM Backup Plans

- Tiered PCMs to reduce merging



- ROOT currently using PCH, could be done on larger scale, too

PCM Backup Plans (2)

- Parse header files + all selection.xml / Linkdef.h at runtime; slow library loading
- Generate single PCH from ZIP file containing all headers + Linkdef / selection

Unloading: Scope

- Need to roll back interpreter state:
 - compiler (AST)
 - preprocessor (#define)
 - file manager (foo.h in #include “foo.h”)
 - execution engine (`__ZN3fooD1Ev`)
- Not part of clang

Unloading: Status

- Error recovery implements AST unloading; have design for missing parts
- Execution engine almost done

Unloading Backup Plan

- Mockup of unloading: namespace hiding
 - namespace N1 {class A {int i};}
 - namespace N2 {class A {float f};}
- Simpler implementation but limitations
 - #include <X> only works if X is in PCM / PCH
 - extern “C”,...

Reflex

Ingredients

- `genreflex`
- Reflex API

Reflex API

- Used by LHCb, CMS, ATLAS for type info
- Must be replaced by calls to TClass etc
- CMS has isolated all calls to Reflex
 - now uses TClass where possible
 - requests for some new TClass interfaces

TClass Interfaces

- Will be implemented in ROOT 6 and 5.34
 - Experiments can migrate away from Reflex API usage in ROOT 5
- List of missing interfaces for TClass will be provided by LHCb, ATLAS

genreflex

- Will use same sources as rootcling
- Dedicated selection.xml reader
 - Linkdef.h and selection.xml mapped to same selection data

Scheduling, Release Definition

Reliability of Estimates

- We don't go from 0 to 100 in 4 months
- Impressive amount of work already done
 - cling, ROOT I/O, roottest
- We are implementing feature by feature, passing test by test
- Converging

Problem of Estimate

- Much of this work is fundamental
 - reimplementing heart of ROOT
 - connecting two large, independent products != adding features to product
- No more blockers, but work, work, work
- Implementing first, refining later

2012

- Redefinition of feature set caused redefinition of release date
 - new: “As good as ROOT 5”
- Mis-estimation of modules

Experiments and 6.0

- Frameworks have different set of requirements than users
- Need genreflex, TClass API
- Need less interpreter features than physicists

Experiments' Migration

- ALICE: Peter Hristov and Axel
- ATLAS: Valeri Fine and Fons
- CMS: Bill Tanenbaum and Paul Russo
- LHCb: Marco Clemencic and Axel

ROOT 6.0

- ROOT 6.0 is user-oriented release
- As good as ROOT 5, incl PCM, unloading
- Mostly interpreter features

Separate Releases

- ROOT 6 beta for experiments: November 2013
- ROOT 6: May 2014
- Enables independent / parallel progress and early feedback

Release Planning

After ROOT 6

ROOT 6

ROOT 6 Beta if AiROOT requires interpreter features

ROOT 6 Beta

Doing

Done

Release Planning

- ATLAS, CMS, LHCb beta release:
November 2013
- ROOT6: May 2014
- If massive use of interpreter features
(unloading,...) in AliROOT:
May 2014 (ROOT 6)
else November 2013 (beta release)

Contributors

- Vassil Vassilev (100%): cling
 - fellow, contract ends March 2014
- Axel Naumann (90%)
- Philippe Canal (25%)
- Paul Russo (50%): ROOT / TCallFunc (“meta”)

Contributors (2)

- since June 1: Bertrand Bellenot (50%)
- starting July 1: Cristina Cristescu (100%), TS
- Wim Lavrijsen (50%): PyROOT

ROOT 6.00-beta

Work List

JIRA issue number

FTE estimate in weeks

- Autoloading namespaces, template instances **2w** *ROOT-4786 ROOT-4779*
- I/O of private classes **2w** *ROOT-4865*
- Finish genreflex **8w** *ROOT-165 ROOT-5266 ROOT-49*
- PluginSvc migration recipe **1w** *ROOT-5245*

ROOT 6.00-beta

Work List (2)

- CallFunc: return by value, performance **8w**
ROOT-4961 ROOT-4972 ROOT-4955 ROOT-5272
- TClass extensions: TEnum,
hasDictionary()... **8w** *ROOT-164*
- PyROOT - hinges not on PyROOT but on
prerequisite bugs in ROOT **4w** *ROOT-154*
- Modules / PCH for frameworks to run (but
not yet framework development) **4w** *ROOT-149*
ROOT-5161

ROOT 6.00-beta

Planning

- 7 FTE weeks short over 13 weeks
= 0.5 FTEs

ROOT6 Work List

- #include “Event.h”
gSystem->Load(“libEvent”); new Event() **6w**
ROOT-4691
- Improve unnamed macro handling **7w**
ROOT-4719 ROOT-4763 ROOT-4780 ROOT-4791 ROOT-4760
- #include “Test.C+” **4w** *ROOT-5273*
- foo(); >& out.txt **4w** *ROOT-5274*
- int A = 12; int A = 14 **4w** *ROOT-4803*

ROOT6 Work List (2)

- Finalize dynamic scopes:
TFile::Open("f.root"); hist->Draw() 8w
ROOT-5216 ROOT-5276
- Enhance recovery on input error 8w
ROOT-4704 ROOT-5029
- #include "A.h"; mv B.h A.h
#include "A.h" 2w *ROOT-5277*
- Unloading, namespace-based 9w *ROOT-5278*
- Module-based dictionary / alternative 20w
ROOT-149

ROOT6 Work List (3)

- Tutorials *4w*
- ROOT manual *4w* *ROOT-5167*
- NOTE: no proper unloading yet; would need an additional *16w* (of Vassil!) *ROOT-48*
 - Vassil scheduled 100% (down from 300%)

ROOT 6 Planning

- 20 FTE weeks short on 29 weeks
= 0.7 FTE
- Need 0.5 FTE + 0.7 FTE = 1.2 FTE from now to May to make both releases!

Summary

- ROOT 6 is converging
- We need additional hands
- Vassil is a central contributor
- Split milestones increase control and availability

Work Plan - Beta

Title	Assignee	Estimate
Add MetaProcessor support for .O 0 1 2..	BB	0.5
Review the issue of dictionary (I/O) for private classes.	XX	2
Autoloading of namespaces and template instances	VV	2
ClingWorkAroundCallfuncAndReturnByValue (See https://savannah.cern.ch/bugs/index.php?98317 and https://savannah.cern.ch/bugs/?98148)	PR	2
Cling testsuite doesn't propagate the return result when piping to FileCheck	BB	2
Verify that PCMs "block" #include	AN	1
Review the TClass initialization for STL collection when module are in place	AN	1
Add 1st class support for Enum type in core/meta	CC	8
Incremental PCM	PC	1
Incremental PCM	AN	1
Improve performance of CallFunc	PR	4
Improve performance of CallFunc	PC	2
genreflex CLI options	BB	2
genreflex selection.xml	XX	4
Support for GAUDI PluginSvc migration to ROOT's TPluginManager	XX	1
Report PCM issues to clang to test progress / collaboration	AN	6
	FR	0

Work Plan - ALICE / Interpreter

Title	Assignee	Estimate
Add pragma handler to the PP that does the main header file parsing to enable inline pragma handling (currently looking at those in the linkdef file)	PC	1
Resolve symbols at runtime from libraries loaded at runtime	XX	6
Improve detection of unnamed macro	BB	4
Add support for #include within unnamed macro (being parsed at global scope)	AN	2
Add support for #include "TestClass.cc+"	BB	4
canWrap: namespace, #if	PR	2
Fix handling of return statement in unnamed macro	AN	1
	FR	0
	VV	0
	CC	0

Work Plan - 6.0

Title	Assignee	Estimate
Add support in Cling for output redirection: ls(); >& output.txt	XX	4
In diagnostic report the original file name and line number rather than mentioning input_line	AN	0.5
Update ROOT manual For cling	FR	4
Uncache file lookup failures	VV	2
Add support for multiple .q level (.qqqqqq) with various degree of tear down being done (including none)	BB	1
ROOT prompt versus ProcessLine(): multi-line continuation	AN	1
Enhance error recovery	CC	8
Error handling and messages	BB	4
Review tutorials executions	XX	4
Prevent module reader from re-reading definitions	AN	
Re--add support for identical redeclaration	VV	4
Prevent spurious autoloading of library due to clang typo correcter	VV	2
Verify that ClingWorkAroundMultipleInclude is fixed by the Module.	AN	0.5
Test (and finish) support for dynamic scope	VV	8
Unloading mockup using namespaces	PR	6
Unloading mockup using namespaces	PC	3
Unloading in all parts	VV	0
Avoid crashes in 'simple' use of null pointer	VV	0
Implement PCM or alternative	AN	8
Implement PCM or alternative	XX	12
	CC	0
	PR	0

Resources

		13 ALICE@13		13 ROOT6		29	
Fraction	Name	Exp Have	Needs	Have	Needs	Have	Needs
0.8	AN	10.4	9	1.4	3	11.2	10
0.7	BB	9.1	4.5	4.6	8	7.8	5
0.6	CC	7.8	8	-0.2	0	9.4	8
0.2	FR	2.6	0	2.6	0	5.8	4
0.25	PC	3.25	3	0.25	1	3.25	3
0.5	PR	6.5	6	0.5	2	6.5	6
1	VV	13	2	11	0	15	16
1	XX	13	7	6	6	16	20
1	Grand Total	65.65	39.5	26.15	20	74.95	72
				15 ALICE@29		29	
	Name			Have	Needs	Have	Needs
	AN			3	3	11.2	10
	BB			6	8	7.8	5
	CC			1	0	9.4	8
	FR			3	0	5.8	4
	PC			0.75	1	3.25	3
	PR			1.5	2	6.5	6
	VV			15	0	15	16
	XX			8	6	16	20
	Grand Total			38.25	20	74.95	72