Cint : C++ interpreter

Frequently Asked Questions Digest

14 Oct 2002 at CERN

Masaharu Goto

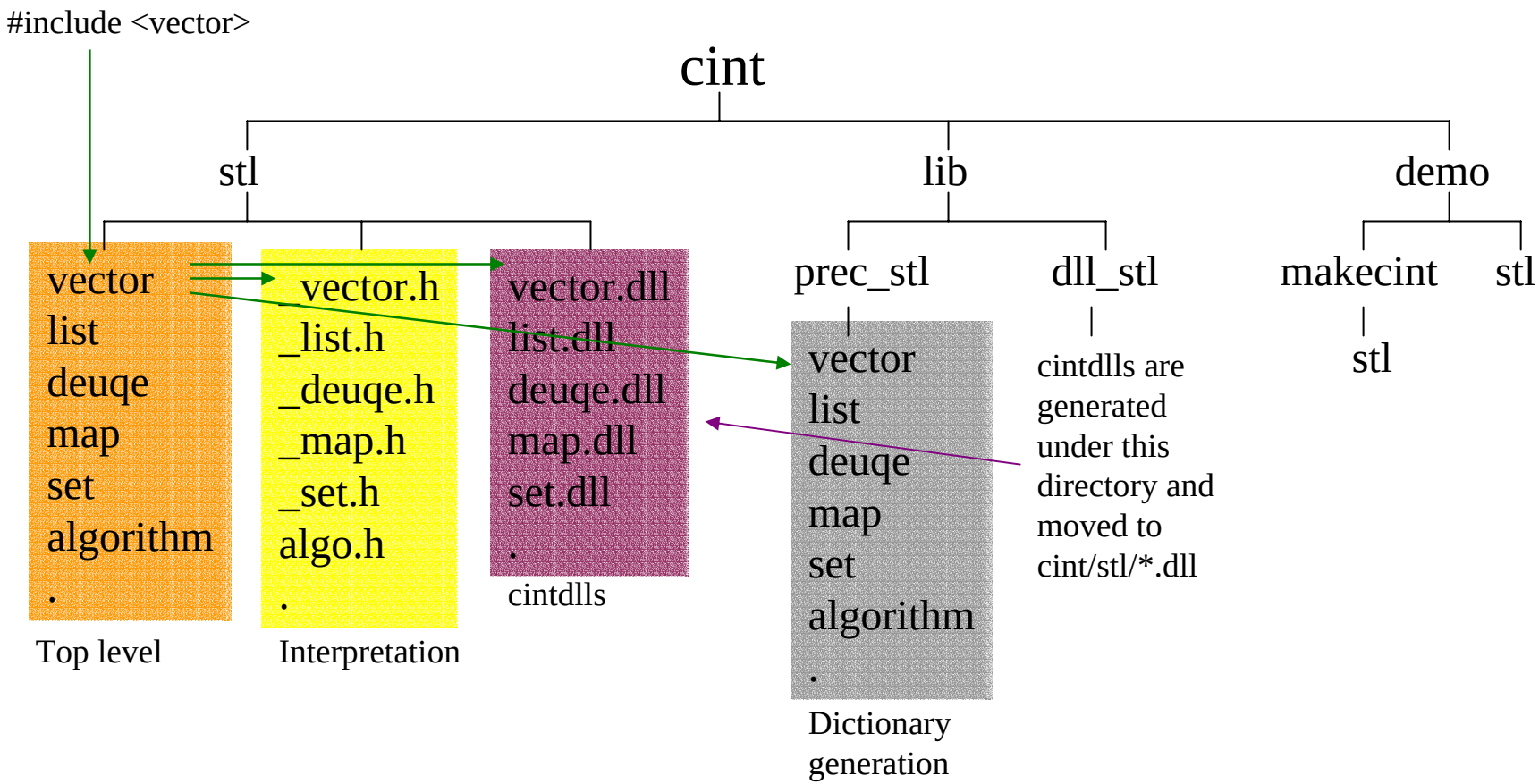
On going progress, bug fix

- Many activities since ROOT2001
  - Change ID 1550 to 1723

Frequently Asked Questions

- STL (dictionary generation)
- Optimization (bytecode) problems
- Multi-platform issues
  - Variable arguments
  - ‘long long’ and ‘long double’
- Basics of Cint (from non ROOT users)

How STL is implemented in CINT



Interpreting STL containers

- cint/stl/_[container].h are used
- Derived from 1994 HP implementation
  - No default arguments

    template<class T> class vector { };
- Cint can only interpret simple examples of vector and list

STL dictionary generation

- Files under cint/lib/prec_stl are used
- Derived from 1997 ANSI/ISO C++ draft
- All kinds of STL containers can be compiled with makecint/rootcint
  - vector,list,deque,map,multimap,set,multiset,queue,stack,valarray
- Many compilers have subtle deviation from the standard → #ifdef

STL progress

- Support new compilers (versions)
  - gcc 3.0 , 3.1/3.2 , RedHat 7.2 gcc 2.95
  - Borland C++ compiler 5.5
  - Intel C++ compiler
- More ANSI/ISO conformity
  - Along with new compiler support

STL suggestion

- Use compiled STL containers
  - With ROOT, install cintdlls
- Generic algorithms are interpreted
  - cint/stl/algo.h
- Report STL dictionary generation problems
  - Or, you could debug cint/lib/prec_stl/* by yourself

Optimization (bytecode) problems

- Class object instantiated in a loop

  for(int I=0;I<5;I++) { TXxxx a; }

  – Especially when if() statement is false in the 1st iteration

  for(int I=0;I<5;I++) { if(I) TXxxx a; }

- On going bug fix
  – Bug fix in cint 5.15.59

Optimization suggestion

- Use –O0 (optimization off) if speed is not critical
- Use script compiler if performance is important : Thanks to Philippe Canal
- Use new/delete for class object if possible

Multi-platform issues

- There are not many multi-platform issues
  - Because Cint is made platform independent
- Few exceptions are
  - Variable arguments
  - ‘long long’ , ‘long double’
  - STL (as explained previously)
  - 64bit issues may be sometimes overlooked

Variable arguments

- Implementation of variable arguments in dictionary is highly platform dependent
- Progress
  - va_arg for HP-UX is supported
    - This was the most difficult one
- Need to debug each platform in face
  - But, this is not easy

    Patience, patience, patience …

‘long long’, ‘long double’

- Supported by wrapper class
  - class G__longlong, G__longdouble
- Supported as an optional DLL component
  - include/long.dll ← compiled in lib/longlong dir
- Why not supporting this as basic component of Cint?
  - Still seeing multi-platform issues.
  - Investigation needed

Thank you

- Send your message to cint@root.cern.ch

- Please include following information
  - Cint version 5.15.xx (different from ROOT version)
  - CPU, OS, compiler, compiler version