A New C++11 Stack

David Chisnall

February 5, 2012
C++: Who Cares?

Hardware Vendors love ‘optimised’ C++ code that lets them sell more RAM and faster processors

Software Developers love the job security that complex template metaprogramming provides

Users want to run things like FireFox and OpenOffice
Why Do We Need a New C++ Stack?

- WG21 released a new standard while no one was looking
  - GCC 4.2.1 / GNU libstdc++ don’t support it
  - (And they’re GPL’d)
  - (And newer versions are GPLv3)
  - (Which is, like, bad and stuff)
I might have to do expensive rewrites or release my code if I accidentally violate it (fear)

- I don’t know if what I want to do will violate it (uncertainty)
- I’m not sure I understand this big blob of legalese (doubt)

GPLv3 comes with twice as much FUD as v2! (e.g. permanent termination of license in case of violation)
• I know exactly what I can do with this code (anything except claim I wrote it)
C++11 provides things like r-value references that allow some optimisation (move semantics)

- Supporting this touches almost every function / method in STL and amounts to an almost total rewrite of libstdc++
- So maybe we should actually do a complete rewrite of libstdc++?
- And all of the other bits too?
The GNU C++ Stack

C++ Source

\( g++ \)

C++ Programs

libstdc++

libsupc++

libc

kernel
The New C++ Stack

C++ Source → clang++ → libc++ → libcxxrt → libc → kernel → C++ Programs
What Needed Porting

- Libcxxrt was developed for FreeBSD, tested on Linux
- Libc++ was developed on Darwin, tested on... Darwin
- Clang++ was developed on many platforms, tested everywhere (even on Windows sometimes)
Libc++ and Locales

Libc++ uses a lot of _l libc functions

- These were missing on FreeBSD
- Implemented as a result of FreeBSD Foundation funding
- Some other bits were Darwin-libc specific
The Port

- Very minor changes to libc++
- Implement the missing bits in libc
- Modify the test suite to use locale names that are valid on FreeBSD and Darwin (e.g. en_US.UTF8 instead of en_US)
Libcxxrt and libc++ are in FreeBSD Trunk

- Test suite passes more tests than on Darwin (yay!)
- Atomics currently waiting for me to finish implementing them for C in clang
- Should be enabled in FreeBSD 9.1, default in 10
- Please test!
How To Test

# The old GNU stack:
$ g++ foo.cc
$ g++ foo.o

# Clang with the GNU libraries
$ clang++ foo.cc -stdlib=libstdc++
$ clang++ foo.o -stdlib=libstdc++

# The new stack
$ clang++ foo.cc -stdlib=libc++
$ clang++ foo.o -stdlib=libc++

Default for clang is currently libstdc++, but it won’t be forever...
Other Ports

Someone at Google started a GNU/Linux port, but glibc is too much pain to work with so they gave up.

- Ruben Van Boxem has worked on a Windows port (missing Windows ABI support in clang is currently the blocker there).
- Solaris port sponsored by Tbricks underway, should be finished in February.
- NetBSD? OpenBSD? Minix?
Questions?