redhat.

Libabigail & ABI compatibility

Taming the runtime linking problem

Ben Woodard Consulting Engineer
Dodji Seketeli Tools Engineer
Aug 2017

Problems due to ABI Compatibility

There are several problems cause by ABI compatibility problems either real or feared.

- Linking objects compiled by different compilers not recommended
  - Unforeseen problem at runtime
  - Need to compile libs for all compilers
- Library Version problems
  - Match library version with application
  - Match compiler

redhat.

Lots of compilers

We use a lot of compilers...

- GCC
- Intel
- PGI
- XL
- LLVM

redhat.

Lots of compilers

We use a lot of compilers...and a lot of different version of the compilers.

- GCC
  - 4.x -> 7.2
- Intel
  - 9.1, 1[0-4].[01], 1[5-8]
- PGI
  - 7.[01], [89].0, 10.[2,9], 11.{1,10}, 12.[18], 13.{1,2,6,10}, 14.{3,7,10}, 15.{1,5,7,10}, 16.[13], 17.x
- XL
  - ???
- LLVM
  - 3.[4-8], 4.x, 5.0

redhat.

Runtime Linking Problems

Linking objects compiled by different compilers can sometimes cause unexplained problems. Why?

- ABI is supposed to make this work properly.
  - This does work to some extent with system libraries
  - Other libraries may hit corner cases especially C++
- We are talking about ABI not API
  - API directly affects this.
  - But with C++ templates the API is the ABI.
- Thus linking objects compiled by different compilers is often not recommended
  - Need to compile libs for all compilers
  - Hard to diagnose problems

redhat.

Libabigail

Libabigail reads ELF, DWARF, & XML to build an IR of the ABI

- ELF, DWARF -> IR of ABI -> can be saved as XML
  - Other formats possible e.g. MacO or Windows
  - Not just DWARF to DWARF comparison
    - Comparison between compilers
    - Saved ABI XML represents all ABI elements
  - Compare saved ABI to ELF, DWARF for new object
    - Used to maintain ABI stability
    - No need to keep old objects around.
  - Kernels - we call this KABI
  - abipkgdiff/fedabipkgdiff
    - Used in production
    - fedabipkgdiff - should be easily adaptable to other build systems
    - Fedora dist.abicheck - CI for the distro

redhat.

Libabigail utilities

Libabigail is a C++ library and is easily to write utilities using it

- abidw - dump the ABI to stdout
  - Used to save copies ABI
  - “abidw --abidiff elfobject.so” compares an object to itself. Used to test libabigail
- abidiff - compare the ABI of one ELF or ABI XML file to another.
  - Compare two objects or saved ABI XML files.
- abicompat - does a changed ABI matter to the program
  - If the only changed functions and variables are not used the program, then it isn’t a problem.
  - Does not detect functional changes
- abipkgdiff - compares analogous objects from related packages or directories

redhat.

Runtime Linking Problems

Linking objects compiled by different compilers can sometimes cause unexplained problems.

Why? ABI is supposed to make this work properly.

- C works
  - C is very old & simple
  - Otherwise system libraries wouldn't work.
  - Mostly API problems and no symbol versioning
- C++
  - Standard based on ia64 work with updates
  - Willing to stand behind and support Clang & GCC compatibility.
  - RHEL7 has dual ABI pre/post C++11
- FORTRAN
  - Unknown
- Current GCC & Clang are compatible

redhat.

Dual ABI

The evilness that lingers....

Mostly a problem for RHEL7 where the system compiler is pre-gcc 5.1

Different than language standard i.e. -std=c++14

Libstdc++ supports dual-ABI on RHEL7 with DTS compilers.

General rule of thumb: recompile all C++ libraries with C++11

ABI tags

- Uses inline namespaces to encode ABI tags within the mangled names.
- Not just useful for libstdc++
- To help detect problems -Wabi-tag

redhat.

DWARF

- Not prescriptive
- Still evolving - mostly to deal with C++ challenges
- Inconsistency leads to false positives
  - Biggest problem with C++
  - Inconsistent names
- Weaknesses leads to false negatives
- Not normalized
  - e.g. spaces introduced in type names
- Quality is still a problem with some compilers
  - Leads to false positives and a few false negatives
  - Several compilers are awful to the point of being unusable
- GPU code by and large doesn’t have DWARF.

redhat.

Templates are hard

Example:

```
#include <string>
#include <vector>

std::vector<std::string> var;
```


redhat.

Templates are hard

```
dodji@adjoa:gcc-llvm$ ../../build/tools/abidiff test3-gcc.o test3-clang.o
Functions changes summary: 0 Removed, 0 Changed, 0 Added function
Variables changes summary: 0 Removed, 1 Changed, 0 Added variable
Function symbols changes summary: 11 Removed, 1 Added function symbols not referenced by debug info
Variable symbols changes summary: 0 Removed, 0 Added variable symbol not referenced by debug info

1 Changed variable:

 [C]'std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >,
std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > var' was changed to
'std::vector<std::__cxx11::basic_string<char>, std::allocator<std::__cxx11::basic_string<char> > > var' at test3.C:4:1:
  type of variable changed:
   type name changed from 'std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >,
std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > >' to
'std::vector<std::__cxx11::basic_string<char>, std::allocator<std::__cxx11::basic_string<char> > >'
   type size hasn't changed

   1 base class deletion:
     struct std::_Vector_base<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >,
std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > at stl_vector.h:74:1
   1 base class insertion:
     struct std::_Vector_base<std::__cxx11::basic_string<char>, std::allocator<std::__cxx11::basic_string<char> > > at
stl_vector.h:74:1
```


redhat.

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

redhat.

For help getting started, visit
http://brand.redhat.com/applications/presentations
to download the official Red Hat Presentation Guide

For more information on how to use this template within Google Slides, view this step-by-step guide.