# Using latest gcc and clang for great good

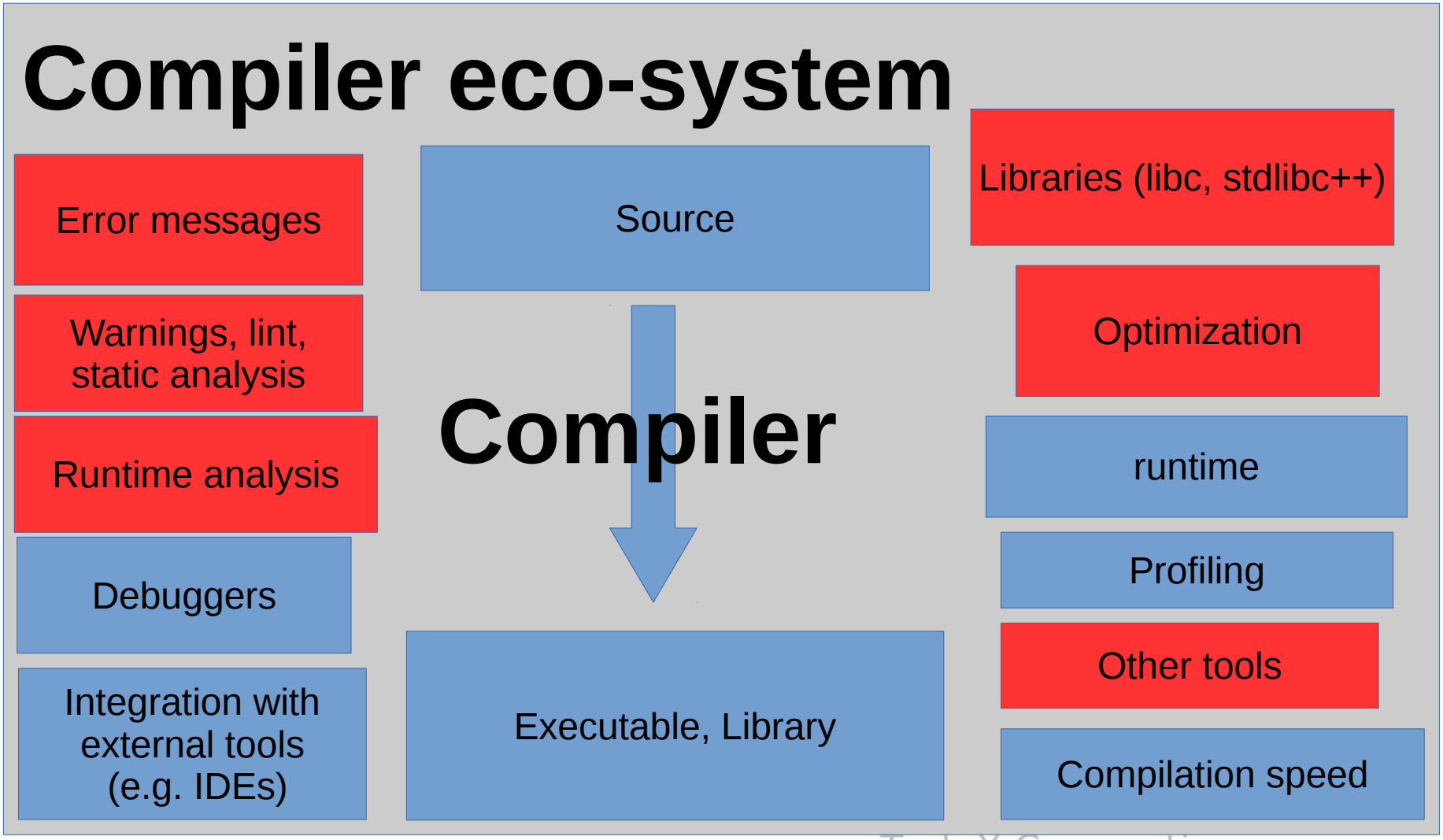
```
[dmeiser@longspeak ~]$ gcc --version
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[dmeiser@longspeak ~]$ clang --version
clang version 3.4 (trunk 194257)
Target: x86_64-unknown-linux-gnu
Thread model: posix
[dmeiser@longspeak ~]$ ls /usr/ | grep gcc
gcc46
gcc47
gcc48
gcc49
gcc_trunk
```

Dominic Meiser
7/21/2014
dmeiser@txcorp.com

TECH-X
SIMULATIONS EMPOWERING YOUR INNOVATIONS

# Compilers do much more than translate source to object code




# Major C++ compilers have improved their eco-system in key areas

- Standards conformance (clang, gcc, and cl mostly conform to C++11 and C++14)
- Faster STL due to move semantics
- Error and warning messages:
  - Fewer false positives
  - Better diagnostics
  - Better formatting
- “Sanitizer tools”: address sanitizer, thread sanitizer, undefined behavior sanitizer
- Link time optimization
- C++ specific optimizations (e.g. De-virtualization)
- Auto-vectorization
- Tools: clang-format, include-what-you-use
- IDE integration (clang backend for Visual Studio; QtCreator uses clang for intellisense)

=> There are many reasons for using latest versions of compilers

# Building latest gcc releases

```
wget http://www.netgull.com/gcc/releases/gcc-4.9.1/gcc-4.9.1.tar.gz
tar xfz gcc-4.9.1.tar.gz
cd gcc-4.9.1
contrib/download_prerequisites
./configure -help
./configure --prefix=/scr_ivy/gcc-4.9.1 \
      --enable-languages=c,c++,fortran,lto \
      --enable-lto --disable-lib-quadmath --disable-quadmath-support
make -j4
make install
```

# Building gcc trunk

- Lots of development occurring between major releases
- Also interesting work in branches (e.g. OpenACC)

```
svn checkout svn://gcc.gnu.org/svn/gcc/trunk gcc_trunk
cd gcc_trunk
contrib/download_prerequisites
./configure -help
./configure --prefix=/scr_ivy/gcc-trunk \
      --enable-languages=c,c++,fortran,lto \
      --enable-lto -disable-lib-quadmath \
      --disable-quadmath-support
make -j4
make install
```


# Building clang trunk

- http://clang.llvm.org/get_started.html
- I normally use libstdc++ (gcc's stl)
- But there is also libc++ (clang/llvm rewrite of stl)

# Using non-system compilers

- /scr_ivy/gcc-4.9.1/bin/c++ foo.cpp
- CC=/scr_ivy/gcc-4.9.1/bin/gcc CXX=... {make,configure}
- cmake -DCMAKE_CXX_COMPILER=/scr_ivy/gcc-4.9.1/bin/c++ ...
- Must link to correct runtime libraries (libc, stdlibc++)
  - Backwards compatible
  - Can change LD_LIBRARY_PATH
  - Use rpath
  - Can clobber old system libraries: cp /scr_ivy/gcc-4.9.1/lib64/* /usr/lib64
  - Or create links to new library location
- Sometimes need to upgrade gdb to debug executables created with newer compiler

# Address sanitizer

- Valgrind as a library (static or dynamic)
- Much faster than valgrind (about 2x slower than uninstrumented code)
- Very few false positives
- Terminates process upon encountering first error

# Clang-format

- Tex for source code
- Build on libtooling (full c++ understanding)
- Simple specification of coding standards
- More than just indentation
- Easy to integrate into vim and emacs
- Apply to diffs (e.g. svn commits)


# Include-what-you-use

- http://code.google.com/p/include-what-you-use/
- Include dependencies are a major cost for compiling large C++ code basis
- Include-what-you-use detects unneeded includes
- Helps finding superficial includes by showing which symbols come from which header
- Replaces includes with forward declarations
- Build on libtooling