LLVM and Clang
Advancing Compilers and Tools

Chris Lattner
http://llvm.org
October 25, 2013
LLVM is everywhere

- Industry
- Open Source
- Academia
… for many different things

• System compiler for Apple and FreeBSD platforms
• Used by most GPGPU implementations
• Many new language implementations
• Finding bugs in source code
• Special effects in movies
• Games, Playstation 4
So..., what is it?
What is a compiler?

**compiler**

*noun*

1. a person who compiles information (as for reference purposes): *a compiler of anthologies*.
2. **a computer program** that transforms human readable source code of another computer program into the machine readable code that a CPU can execute.

- Clang and GCC are compilers
- What is LLVM?
What is LLVM?

llvm.org is an open source umbrella project

• Provides useful tools:
  ▪ Assembler, linker, compiler, debugger, and more

• Strong community, with shared values:
  ▪ Common processes, patch review, etc
  ▪ Common design approaches
  ▪ Preference for MIT/BSD License

• LLVM is a compiler **infrastructure**!
Compiler Infrastructure 101
How does a compiler work?

- Frontend: Parse and validate source code
- Optimizer: Improve intermediate form
- Backend: Generate target specific code

Standard approach for at least 35 years!
In 2013, this is not good enough!

- Great compilers are a huge investment:
  - Source code analysis framework
  - Machine specific code generation
  - Performance optimization
- Other tools want these capabilities too!
  - Compiler “plugins” are not enough
Decomposing a processor target in LLVM

- JIT Support
- Compiler Support
- Assembler
- Disassembler
- Instruction Tables
Building an Assembler

Assembler

Command Line Interface  Common Assembler Logic

JIT Support

Compiler Support  Assembler  Disassembler

Instruction Tables

ARM

JIT Support

Compiler Support  Assembler  Disassembler

Instruction Tables

X86

JIT Support

Compiler Support  Assembler  Disassembler

Instruction Tables

PowerPC, Sparc, SystemZ, PTX, ...
Advantages of this Design

• One truth for instructions:
  ▪ New features (e.g. AVX-512) added in one place
  ▪ Assembler, disassembler, and compiler support all agree
• Compiler gets integrated assembler
• JIT encodings tested by static compiler
• Clients decide what features they need
Compiler Infrastructure?

- Library-based design
  - Modularity
  - Proper layering
  - Testability
- Follows “textbook” compiler design
  - Frontend, optimizer, backend
  - … with enforced layers
- Enables building things we never anticipated!
Applications of LLVM
mesa 3d - LLVMpipe Software Rasterizer

Benchmarks from phoronix.com

- **OpenArena v0.8.5**
  - Resolution: 640 x 480
  - Mesa 9.1.1 git 044048
  - Mesa 9.2 development git 064605f

- **World of Padman v1.2**
  - Resolution: 800 x 600
  - Mesa 9.1.1 git 004040
  - Mesa 9.2 development git 064605f

- **Urban Terror v4.1**
  - Resolution: 800 x 600
  - Mesa 9.1.1 git 044048
  - Mesa 9.2 development git 064605f

Benchmarks from phoronix.com
Open Shading Language

• Special effects rendering engine:
  ▪ Quality is everything
  ▪ Huge: > 200GB per scene
  ▪ 4-10 hours/frame
  ▪ Many thousands of cores

• Driven by Sony Pictures Imageworks
  ▪ Used in several well-known pictures

http://llvm.org/devmtg/2010-11/
Compile just about anything to Javascript!

Epic Citadel

rawson.js

https://github.com/kripken/emscripten/wiki
Commercial Language Implementation

- **Xcode**
  Apple
  C, C++, Objective-C

- **embarcadero**
  C++ Builder

- **OpenCL**
  Apple, Intel, AMD, NVidia, Rapidmind, Gallium3d, ...

- **NVIDIA CUDA**

- **Adobe Pixel Bender**

- **mono**
  C#, Cross Platform

- **LabVIEW**

- **REALbasic**
  Cross-platform that really works.
Research and Independent Languages

- The Glasgow Haskell Compiler
- Rust
- Julia
- pure
- Rubinius
- LLVM D compiler
- MacRuby
- LLVM Pascal Compiler
- Intel SPMD Program Compiler
Clang - “C Lang”uage Family

• Compiles C, C++, and Objective-C
  ▪ Drop-in compatible with GCC & Visual Studio (wip)

• Only compiler with:
  ▪ Full C++’11 language and library
  ▪ Modern Objective-C

• Follows the LLVM library-based “infrastructure” design
  ▪ Builds on powerful LLVM backend
  ▪ Reusable in other tools
Clang has great diagnostics

```
$ ksh - 105 % clang pointmain.c -o pointmain -g
Note: Clang defaults to using C99 mode with warnings enabled
In file included from pointmain.c:2:
./point.h:6:2: error: expected ';' after struct
  }
^*
pointmain.c:6:37: error: no member named 'horizontal' in 'struct Point'; did you mean 'horizontal'?  
  printf("%d, %d\n", pl.vertical, pl.horizontal);
  ^
In file included from pointmain.c:2:
./point.h:5:19: note: 'horizontal' declared here
  double vertical, horizontal;
^*
pointmain.c:6:11: warning: conversion specifies type 'int' but the argument has type 'double' [-Wformat]
  printf("%d, %d\n", pl.vertical, pl.horizontal);
  ^
pointmain.c:6:15: warning: conversion specifies type 'int' but the argument has type 'double' [-Wformat]
  printf("%d, %d\n", pl.vertical, pl.horizontal);
  ^
5 diagnostics generated.
$ ksh - 106 %
```
Clang compiles fast

**Time to Compile ImageMagick**

- GCC 4.8.1: 34.3 seconds
- Clang 3.3: 21 seconds

**Time to Compile PHP v5.2.9**

- GCC 4.8.1: 33.3 seconds
- Clang 3.3: 21 seconds

2x Faster!

http://www.phoronix.com/scan.php?page=article&item=intel_haswell_llvm33
Generates fast code

SciMark v2.0 - Composite Result

GCC 4.8.1
GCC 4.9.0 SVN 20130623
Clang 3.3
Clang 3.4 SVN 20130626

mflops, more is better

10% Faster!

20% Faster!

http://www.phoronix.com/scan.php?page=article&item=llvm_clang34_first
Clang Applications

- Clang static analyzer [http://clang-analyzer.llvm.org](http://clang-analyzer.llvm.org)

- Address Sanitizer [http://clang.llvm.org/docs/AddressSanitizer.html](http://clang.llvm.org/docs/AddressSanitizer.html)


- Many more...
and so much more...

http://lldb.llvm.org/  LLDB Debugger
http://lld.llvm.org/   LLD Linker
http://libcxx.llvm.org/ C++ Standard Library
http://compiler-rt.llvm.org/ Compiler Runtime
http://dragonegg.llvm.org/  GCC Plugin
http://openmp.llvm.org/   OpenMP Runtime

http://llvm.org/
LLVM Compiler Infrastructure
High technology in service of great applications and tools

http://llvm.org/