LLVM Performance Workshop at CGO 2025

Aditya Kumar

Logistics and Contacts

- Aditya (hiraditya at msn.com)
- Jose M Monsalve Diaz (jmonsalvediaz at anl.gov)
- Shilei Tian (i at tianshilei.me)
- Rafael Andres Herrera Guaitero (rafaelhg at udel.edu)
- Kevin Sala (ksala at llnl.gov)

Started in 2017

- To bring llvm developers focussed on performance optimizations.
    - Showcases new techniques, tools, and methodologies for improving performance of critical applications
- Presenters from major tech companies and universities worldwide

What types of people attend?

- Active developers of projects in the LLVM Umbrella (LLVM core, Clang, LLDB, libc++, compiler_rt, klee, lld, OpenMP, MLIR, etc).
- Anyone interested in using these as part of another project.
- Students and Researchers.
- Compiler, programming language, and runtime enthusiasts.
- Those interested in using compiler and toolchain technology in novel and interesting ways.

Topics

- Clang
- LLVM backends
- Runtime
- Emerging architectures
- Heterogeneous programming models
- Machine Learning Optimization Techniques
- Cryptography

What can we do to the workshop better?

Compiler improvements in RISC-V

Aditya Kumar

RISC-V Backend in LLVM

- A ‘default’ (rather than experimental) backend since LLVM/Clang 9.0 (Sep 2019).
- Current extension support status: https://llvm.org/docs/RISCVUsage.html
- RVA23 Ratified
  - Vector Extension
  - Hypervisor Extension

Toolchains

- LLVM Android toolchain
    - https://android.googlesource.com/platform/prebuilts/
- riscv-gnu-toolchain
    - https://github.com/riscv/riscv-gnu-toolchain
    -

Language runtimes

- Go - RISCV64 supported since 1.14 release
  - RISCV32 port not available
- Rust
  - tier 2 support for RISC-V RV64
  - RISCV32 support Experimental
  - Bare-metal supports more machines
    - https://github.com/riscv-rust/riscv-rust-quickstart
- Dart SDK targets RISCV64
- More Details
  - https://wiki.riscv.org/display/HOME/Language+Runtimes

Bootloaders

- U-boot - “virt” machine 32-bit/64-bit
  - https://github.com/u-boot/u-boot/blob/master/doc/board/emulation/qemu-riscv.rst
- OpenSBI - https://github.com/riscv-software-src/opensbi
- Coreboot

What is new in LLVM

- RISCVSExtWRemoval
- RISCVCodeGenPrepare
- RISCVExpandAtomicPseudoInsts
- RISCVInsertSETVLI
- RISCVRedundantCopyElimination
- RISCVMakeCompressible
- RISCVRVVInitUnde

Reference: https://riscv-europe.org/summit/2023/media/proceedings/plenary/2023-06-08-11h30-Alex-BRADBURY-slides.pdf

What is new in LLVM

- Auto vectorization with the loop vectorizer
  - Enabled upstream (Google SiFive collaboration)
  - Parallel downstream work with BSC (and others) on tail folding using LLVM vector predication intrinsics (setting VL rather than using masked loads and stores).
  - More tuning to be done
  - Has support for scalable vectors (and vector register grouping)

Reference: https://riscv-europe.org/summit/2023/media/proceedings/plenary/2023-06-08-11h30-Alex-BRADBURY-slides.pdf

What is new in LLVM

- BOLT: A post-link optimiser designed to speed up large application
- LLVM libc
- CI improvements
- llvm-mca
- Various newly supported ISA extensions. e.g.
  - z[f|d]inx, code size reduction extensions, vector crypto, zacas

Reference: https://riscv-europe.org/summit/2023/media/proceedings/plenary/2023-06-08-11h30-Alex-BRADBURY-slides.pdf

Contributing to RISC-V

- Mailing list / discourse discussion
  - https://lists.riscv.org/g/main
  - https://llvm.org/docs/GettingInvolved.html
  - https://discourse.llvm.org/tag/riscv
- Biweekly sync-up / coordination calls
- Vectorizer improvements
  - https://docs.google.com/document/d/1Glzy2JiWuysbD-HBWGUOkZqT09GJ4_Ljodr0lXD5XfQ
- SIGs
  - e.g., https://lists.riscv.org/g/sig-vector

Questions