BUD17-406
Toolchain Coordination
Maxim Kuvyrkov (Linaro), Siddhesh Poyarekar (Linaro)
Joey Ye (ARM)
Upstream Toolchain Coordination

- Linaro TCWG
  - GNU Toolchain
  - LLVM Toolchain
- Linaro Services
  - GNU Toolchain
- ARM
  - GNU Toolchain
  - LLVM Toolchain
- ST
  - GNU Toolchain
  - LLVM Toolchain
Linaro TCWG: GNU Toolchain

- GCC: IPA-CP improvements
- GCC: IPA pure-const improvements
- GCC: Gimple reassociation optimizations
- GCC: Investigate LTO performance
- Glibc: C11 threads support in Glibc
- Glibc: Continue code cleanup and consolidation
- GDB: Kernel Awareness
Linaro TCWG: LLVM Toolchain

- **LLD:** Make LLD self-hosting with Clang on ARM + testsuite
- **LLDB:** Complete hardware breakpoint support
- **LLVM:** Global ISel for ARM (basic coverage)
- **LLVM:** Investigate libomp vs libgomp
- **LLVM:** Buildbot improvements
Linaro TCWG: Other tools

- OpenOCD: AArch64 & GDB Remote debugging interoperability
- ILP32 ABI: Review Glibc patches, assist with delivery
- LuaJIT: Performance improvement investigation
Linaro Services: GCC

- GCC optimizations for falkor
  - Pipeline description
  - Tuning
  - Loop Array Prefetching

- Generic GCC work
  - Bug fixes in the scheduler, vectorizer
  - Interplay of loop distribution and vectorizer passes
    - Make alias analysis smarter
    - Reverse CSE when it affects loop distribution adversely
Linaro Services: Glibc

- Glibc optimizations for falkor and aarch64
  - Optimized routines for falkor
  - Tunables support for IFUNC selection overrides
  - Feature parity with x86 - sysconf cache info

- Generic glibc work
  - Systemwide tunables
  - Tightening environment variable processing in AT_SECURE processes
ARM Update to BUD17
TCWG

Joey Ye
Senior Engineer Manager
GNU development in 2016

▪ Push architecture features, Performance optimizations and defect fixes into GCC7
▪ Architecture features
  ▪ ARMv8.2-A full support
  ▪ ARMv8-M full support
  ▪ New architectures (SVE, ARMv8.3-A)
  ▪ Reviewing ILP32
▪ Performance optimizations
  ▪ Vectorizer enhancement for aarch64
  ▪ IVOpts for aarch32/aarch64
  ▪ Store widening
  ▪ If conversion enhancement
  ▪ Observed performance gain on SPEC
▪ Defect fixes
  ▪ Required as aarch64 becomes a primary platform
▪ 7 engineers attended GCC Cauldron
GNU focus 2017

- Push architecture features, optimizations and defect fixes into GCC8
- Architecture features enabling
- Optimizing for server benchmark
  - Improve CPU2006
  - Look into new version of SPEC benchmark
- Optimization tasks
  - Vect-math library enabling and optimization
  - Math function optimization (libm & newlib)
  - Vectorizer enhancement for aarch64
  - GCC prefetcher
  - Watching GCC aarch64 performance regression
- Cleanup GCC aarch64 correctness regression
LLVM development in late 2016, early 2017

- **Architecture features**
  - ARMv8.3-A full support – ongoing
  - SVE support – ongoing

- **Optimizations**
  - More code size improvements: see [Sjoerd Meijer’s upcoming presentation at EuroLLVM (March 27/28, Saarbrücken)](#).
  - Continuous tracking of top-of-trunk performance on a range of benchmarks and cores; reacting to get performance regressions fixed quickly. See [Kristof Beyls’s FOSDEM2017 presentation last month](#).

- **LLVM framework improvements**
  - Making it easier to write pipeline models for ARM code.
  - Contributing to GlobalISel development.

- **Other improvements**
  - Libc++: improve multi-threading support; making no-exceptions work.
  - Debug info at optimization: significant improvement on correctly describing variables in loops.
  - Android and Chromeos enabling: fixed a range of defects.
  - Infrastructure: LNT improvements.

- Three presentations at LLVM developers meetings and FOSDEM
- Contributed to the organization of US/Euro LLVM dev meetings and FOSDEM
LLVM Focus 2017

- Enable architecture features
- More code size optimizations
- Continuous tracking of top-of-trunk performance
- Contributing to GlobalISel development
- Android and Chromeos enabling
- Infrastructure: LNT improvements
- Optimizing libomp
Expect Linaro to carry on

- Continue current contribution from Linaro
  - LLD, GDB, Qemu
- Continue Linaro GNUTools binary release
  - AArch64
  - AArch32 A-profile
  - Linux target
- Continue GCC/LLVM quality testing
- Continue GCC optimizations
Additional help needed

- SVE enabling?
  - Qemu
  - ACLE in GCC
- More LLVM quality test?
  - Android build test and trunk perf test
- AArch32 R/M profile bare-metal binary release?
  - Linux, Windows and Mac host
- Optimize libgomp?
  - Find and verify major design flaws in GOMP
  - Verify applicability of per-thread task queues approach
  - Handle bigger scale project for re-designing GOMP to be more like LLVM libomp
  - Help to verify correctness of proposed changes (e.g. check that OpenMP3.1 Validation Suite test passes; tasking benchmarks, etc. are still working)
Thank You

#BUD17

For further information: www.linaro.org
BUD17 keynotes and videos on: connect.linaro.org