Hafnium build system

TF-A Tech Forum

Olivier Deprez

Jan 2022

Agenda

- Hafnium build system (recap)
- Problem statement
- Recent upstream changes
- Proposals

Hafnium build system (recap)

- Project transitioned from Google to trustedfirmware.org in May 2020.
- Normal world Hypervisor augmented to an FF-A secure partitioning kernel (aka SPM).
- The project is self contained with source code, test framework (incl. Linux dependency), test cases, CI scripts.
- Build system is gn (https://gn.googlesource.com/gn/) + ninja (https://ninja-build.org/)
- Split in submodules
  - https://git.trustedfirmware.org/hafnium
  - Hafnium (top level), driver, prebuilts, project/reference, dtc, googletest, linux
  - Cloning the top level project and all submodules is necessary to build and test.

Hafnium build system (recap)

- Prebuilts submodule contains a mix of x86_64 and Aarch64 binaries
  - Toolchains (x86_64 clang + gcc), build tools (x86_64), test binaries (AArch64) etc.
- Developer and production needs
  - Build all: Hypervisor+SPMC+test framework and tests (make PROJECT=reference)
  - Run tests (kokoro/test.sh and kokoro/test_spmc.sh)
  - Build all, run tests, run checkers (kokoro/build.sh). Used by jenkins. Vote at each patch submission.
- Builds 10+ targets in one go
  - secure_aem_v8a_fvp_clang, secure_aem_v8a_fvp_vhe_clang, secure_tc_clang, aem_v8a_fvp_clang, aarch64_linux_clang, aem_v8a_fvp_vhe_clang, android_aarch64_clang, host_fake_clang, qemu_aarch64_clang, qemu_aarch64_vhe_clang, rpi4_clang

Problem statement

- This design worked great during project bring up through 2020.
- Inherited the project legacy (hardly scalable)
  - Static LLVM/gcc toolchains stored in the repo (ensures "reproducibility").
  - Supports x86_64 host only.
  - Hardcoded tools paths in build files.
- New requirements emerging in 2021
  - Hafnium component productization (Total Compute, Yocto...)
  - Build Hafnium on Arm host.
  - Favor SPMC vs Hypervisor.
  - Dependency to 3rd party projects (googletest, dtc, linux...)
  - Prebuilt submodule is very large.
  - Clone and build time can be improved.

Q4'21 upstream changes

- Clang 9 to clang 12 migration
  - Fixed clang build/tidy errors hit with recent toolchain.
  - LLVM toolchain in prebuilt still required by the CI. "Reproducible builds".
  - Regular upgrades sourced from Android repo.
- Removed gcc dependency.
- Alternate (out of tree) tools paths
  - ninja and gn binaries can be provided though make command.
  - dtc binary provided through PATH
- Alternate (out of tree) toolchain
  - Provided through PATH
  - Mostly tested with official LLVM stock builds https://releases.llvm.org/download.html
  - Permits using a Yocto provided toolchain.
- The above permits building on Aarch64 host with a recent toolchain.

Proposals for next steps

- Reduce the prebuilt submodule footprint
  - Remove LLVM and gcc from prebuilt (1.5GB)
  - Developer or Jenkins/CI environment provides the LLVM toolchain to PATH.
- Build SPMC targets independently
  - Create project/spmc submodule
    - New submodule git tree (hafnium/project/spmc)
      - SPMC build only, not building the test framework
      - Or SPMC + tests baked by kokoro/test_spmc.h
    - Keep project/reference
      - Still builds all targets or only the Hypervisor targets.
  - Use a TARGET option on build command line.
    - e.g. secure_aem_v8a_fvp_clang
- Reduce dependency to 3rd party submodules (used by test framework)
  - dtc, googletest, linux

Thank You
Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה