Testing TF-A with QEMU in OpenCl
Contents

+ Running QEMU tests in OpenCI
  • Navigating the Jenkins UI
  • Navigating the test results

+ Running local QEMU tests

+ Changes in TF-A CI Scripts
  • Overview of the CI architecture
  • Changes to build and run packages

+ Final Comments and Conclusions
Running QEMU tests in OpenCI
Running QEMU tests in OpenCI
Navigating the Jenkins UI

+ Open the **tf-a-ci-gateway** job console in Jenkins:
  https://ci.trustedfirmware.org/job/tf-a-ci-gateway
+ Select **Build with Parameters** from the left menu
+ Enter **qemu-boot-tests** into the **TEST_GROUPS** input
+ Hit build!
Running QEMU tests in OpenCI

Navigating the Jenkins UI

+ Open the tf-a-ci-gateway job console in Jenkins: https://ci.trustedfirmware.org/job/tf-a-ci-gateway
+ Select Build with Parameters from the left menu
+ Enter qemu-boot-tests into the TEST_GROUPS input
+ Hit build!
Running QEMU tests in OpenCI

Navigating the test results

+ An interactive GUI is produced by the tf-a-ci-gateway job
+ Provides summary of individual test results
+ Results also provided in a JSON-format file
+ Specific test cases can be re-run directly from this console
Running QEMU tests in OpenCI

Navigating the test results

- The result in the status column is a link to the tf-a-builder that run the test
- Provides access to artefacts from the test and all logs
- Terminal symbol links to the console log of that job
- Results are retained for up to 15 days
Running QEMU tests in OpenCI

Navigating the test results

- LAVA populates results live; the plain logs can be downloaded after the run.
- The interactive LAVA console is available as soon as the tf-a-builder submits the job to LAVA.
Running QEMU tests in OpenCI

Navigating the test results
Running local QEMU tests
Running local QEMU tests

- Trust Firmware A and CI repository sources (v2.7 onwards).
- For existing tests, TFTF isn’t strictly necessary (pass dummy value).
- System QEMU for Aarch64 (> v4.2.1).

```bash
test_run=1 \nworkspace=$(mktemp -d) \ntfa_downloads="https://downloads.trustedfirmware.org/tf-a" \nnfs=$workspace \ntf_root="/path/to/trusted-firmware-a/" \ntftf_root="/path/to/tf-a-tests/" \ntest_groups=qemu-boot-tests/qemu-default:qemu-linux.rootfs-fip.uefi-virt \bash $ci_root/script/run_local_ci.sh
```
Running local QEMU tests

```
test_run=1
workspace=$(mktemp -d)
tfa_downloads="https://downloads.trustedfirmware.org/tf-a"

nfs=$workspace
tf_root="/path/to/trusted-firmware-a/"
tftf_root="/path/to/tf-a-tests/"
test_groups=qemu-boot-tests/qemu-default:qemu-linux.rootfs-fip.uefi-virt

bash $ci_root/script/run_local_ci.sh
```

Do not spawn expect instances to track serial output!

Directory to build and run test from.

OpenCI parameter - scratch & tool area

Individual test we would like to run.
Running local QEMU tests
Changes in TF-A CI Scripts
Changes in TF-A CI Scripts

Overview of the CI architecture

- Consists of a build and run phase.
- Build phase downloads and generates all the resources required to run the test.
- The run phase executes the output of the build phase.
- Information about the build phase is encoded in the test configuration.
Changes in TF-A CI Scripts
Overview of the CI architecture

Build configuration – plain text files in `tf_config` or `tftf_config` containing build parameters for TF-A and or TFTF.
Changes in TF-A CI Scripts

Overview of the CI architecture

```{tf-build-config | nil},tftf-build-config]:{run-config | nil}
```

Run configuration – specifies the steps necessary to setup the test environment i.e., choosing a non-secure payload.
Changes in TF-A CI Scripts

Overview of the CI architecture

- Test configurations of a similar nature are collected as a group.
- See the **group** subdirectory in the CI repository for already existing groups.
- Providing a group name to the CI scripts executes all tests thereunder i.e. qemu-boot-tests runs all QEMU tests under group/qemu-boot-tests
Changes in TF-A CI Scripts

Changes to build and run packages

1. **Boot QEMU from TF-A to the Linux shell on a local machine**
2. Automate build steps and fetching of pre-built resources
3. Setup reproducible test environment in CI
4. Automate job submission
Changes in TF-A CI Scripts

Changes to build and run packages

- There are two ways to launch QEMU guests: firmware based and kernel shortcut.
- \texttt{-kernel} in addition to \texttt{-bios} means that QEMU loads the kernel into DRAM and puts the kernel info into the \texttt{fw\_cfg} device.
- Bios is responsible for providing the kernel with the device tree.
- A binary consisting of BL1 and the FIP are used as the BootROM. Bootloader built into the FIP image as BL33.

```
$ qemu-system-aarch64 -machine 'secure=on, virtualization=on, gic-version=2' -cpu max -nographic -display none -d unimp -m 4G -M virt -smp 4 -append 'console=ttyAMA0,115200n8 root=/dev/vda earlycon' -bios qemu bios.bin -initrd rootfs.bin -kernel kernel.bin
```
Changes in TF-A CI Scripts

Changes to build and run packages

1. Boot QEMU from TF-A to the Linux shell on a local machine
2. Automate build steps and fetching of pre-built resources
3. Setup reproducible test environment in CI
4. Automate job submission
Changes in TF-A CI Scripts

Changes to build and run packages

What parts do we need?
- Kernel
- EDK II
- Root filesystem
- Boot image (bl1.bin + fip.bin)

Some parts not possible to build at runtime ie kernel and EDK II

Must build TF-A from source!
Changes in TF-A CI Scripts

Changes to build and run packages

- Introduced three fragments:
  - run_config/qemu-linux.rootfs
  - run_config/qemu-fip.uefi
  - run_config/qemu-virt

- Fetch and archive kernel, and root filesystem
- Fetch EDK II and build the FIP with it as BL33
- Generate the BootROM that is provided with the `-bios` option

```bash
post_tf_archive() {
  image="qemu_bios.bin" b11_path="$archive/b11.bin"
  fip_path="$archive/fip.bin" gen_qemu_image
}

run_config/qemu_fip.uefi

# Concatenate b11 and fip images to create a single BIOS consumed by OEMU.
# cp $b11_path "$image"
dd if=$fip_path of="$image" bs=64k seek=4
archive_file "$image"

qemu_utils.sh
```
Changes in TF-A CI Scripts
Changes to build and run packages

1. Boot QEMU from TF-A to the Linux shell on a local machine
2. Automate build steps and fetching of pre-built resources
3. **Setup reproducible test environment in CI**
4. Automate job submission
Changes in TF-A CI Scripts
Changes to build and run packages

https://tf.validation.linaro.org/scheduler/device_types
Changes in TF-A CI Scripts

Changes to build and run packages

- Each test job has a list of actions [1]:
  - Boot – boot the device and check prompts to see whether device has booted correctly
  - Deploy – download files to boot the device and prepare an overlay of files to run the test action
  - Test – clone and run tests for this job

- For boot action, it constructs the correct pipeline from a specified set

- User provides prompt to expect when system boots successfully... LAVA fails if it doesn’t find this.
Changes in TF-A CI Scripts

Changes to build and run packages

1. Boot QEMU from TF-A to the Linux shell on a local machine
2. Automate build steps and fetching of pre-built resources
3. Setup reproducible test environment in CI
4. Automate job submission
Changes in TF-A CI Scripts

Changes to build and run packages

- Jobs may be submitted to LAVA in one of three ways:
  1. Using the lavacli program
  2. XML-RPC API [2]
  3. Its web UI [3].

- LAVA jobs are defined in YAML files

- A job defines what software should be deployed on the device under test (DUT) and what actions should be performed there.

- Created a template job configuration for boot tests.

script/lava-templates/qemu-linux.yaml
Changes in TF-A CI Scripts

Changes to build and run packages
Conclusion

- To get started running tests, you might need to be added to this group:
  - Get in touch via the mailing list or directly to Joanna Farley to get added.

- Currently QEMU tests do not gate patch submission, long term strategy is to add them to L1/L2 tests once we are confident there are stable.

- There are still quite a few gaps that need to be filled
  - Expect script support
  - Measured Boot
  - OPTEE
  - U-Boot

- Interested, or keen to make it better? Contributions are welcome!
- We are also looking for some help in maintaining the infrastructure – if this is of interest, please do get in touch!
Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
Resources

1. https://validation.linaro.org/static/docs/v2/explain_first_job.html
2. https://tf.validation.linaro.org/static/docs/v2/data-export.html#xml-rpc