Exercising QEMU generated ACPI/SMBIOS tables using bios-bits

Anirban (Ani) Sinha
Principal Software Engineer, Virtualization, Red Hat
anisinha@redhat.com.

FOSDEM 2024
February 3 2024
Agenda

Focus of the talk

▸ Why use BIOS BITS to test QEMU?
▸ Implementation choices for the test framework
▸ Description of the test framework
▸ Overview of changes in BIOS BITS for the test to work
Why BIOS BITS

BIOS BITS

Software written by Josh Triplett

▸ Used by Intel/bios developers to test bios implementations in real physical HW boxes.
▸ Executes ACPI/SMBIOS tables in BIOS directly from grub2 without need for an OS.
▸ Uses acpica acpi interpreter in ring 0.
▸ Has a python environment in ring 0.
▸ No need to learn “bashish” – grub’s native scripting language. Can use python for tests.
▸ Python ACPICA extension. So tests can execute tables using ACPICA.
▸ All components built into an bootable iso which is then used to boot a VM and execute the tests.

Source:
https://www.youtube.com/watch?v=36QlepyUuhg
Why BIOS BITS

QEMU and BIOS BITS

Why use BIOS BITS for QEMU ACPI/SMBIOS Tests?

▸ Existing qtests only validate the ACPI/SMBIOS table blobs against golden master blobs.
▸ They do not actually execute the tables from within a running VM.
▸ We do not want to execute tables though an OS - we want to execute it directly.
▸ Using acpica extension from python scripts make it possible to execute acpi methods easily from python.
Agenda

Focus of the talk

- Why use BIOS BITS to test QEMU?
- Implementation choices for the test framework
- Description of the test framework
- Overview of changes in BIOS BITS for the test to work
Test implementation choices

QEMU and BIOS BITS repos

There are two repos at play - the QEMU repo and the bios-bits repo

- Bios-bits is maintained as a fork of the upstream repo since Josh does not maintain it upstream anymore.
- Lots of enhancements needed to be made to bios-bits fork (upgraded ACPICA/fixes to build issues etc).
- Bios-bits build system generates an iso which is immutable and new if new tests are added or modified, the iso needs to be rebuilt again.
- It is better to have QEMU repo contain the python test scripts as people make changes in ACPI in QEMU.

Note: All means to get in touch with Josh failed while I was working on this project.
Test implementation choices

QEMU and BIOS BITS repos - How to deal with two repositories?

Considerations:

- Do we have bios-bits repo as another submodule? Nooooooo!
  - People hate submodules! See
    [https://lore.kernel.org/all/d7a7b28f-a665-2567-0fb6-e31e7ecbb5c8@redhat.com/](https://lore.kernel.org/all/d7a7b28f-a665-2567-0fb6-e31e7ecbb5c8@redhat.com/)

- For every test added or modified to bits, the bits build should generate a new iso, the test should point to the new iso and then boot a vm with it.

- Going back and forth between two repos is complicated for ACPI developers who just want to add a test in QEMU for the changes they are making in ACPI.
Test implementation choices

QEMU and BIOS BITS repos - How to deal with two repositories?

Considerations:
- How to keep two repos in sync?
- Does ACPI developers care about how bios-bits work? Nooooooo!
- Do we want this new test to be an avocado test ("make check-avocado") or a unit test ("make check-qtest") test?
Test implementation choices

Avocado test or not?!

Considerations:

- Do we want this new test to be an avocado test (‘make check-avocado’) or a unit test (‘make check-qtest’) test?

  - avocado test framework has all the library to spawn a QEMU VM with proper arguments, run it and then terminate it.
  - Overhead of VM management handled by the avocado framework.
  - The framework also handles downloading artifacts that are needed to run the test.
  - ‘qtest’ is run more frequently by developers who make ACPI changes.
  - ‘qtest’ is better understood and familiar? Not everyone care about avocado integration tests.
Test implementation choices

Avocado test or not?!

Considerations:

▸ I started first with writing a unit test ("make check-qtest") using the qtest framework.
▸ Learnt all about it but also realized that it was not appropriate for the kind of test I wanted to write.
▸ Wrote a new python based test framework.
▸ Scrapped it once I learnt of the avocado framework and realized lot of the work is already handled in the framework itself (vm management).
▸ Chose avocado framework in the end.
Agenda

Focus of the talk

- Why use BIOS BITS to test QEMU?
- Implementation choices for the test framework
- Description of the test framework
- Overview of changes in BIOS BITS for the test to work
Description of the test framework

How the bios-bits avocado test works?

Main implementation details:

- Pre-built bios-bits artifacts that are not related to the test itself - grub binaries, acpica etc.
  - Built once and are only built when the need arises. Artifacts are maintained in gitlab.
  - There is a standard build script that uses a latest Ubuntu container to build entire bits iso.
  - The build process also generates these artifacts.
- ACPI tests that are run from within the bits VM are maintained in QEMU repository.
Description of the test framework

How the bios-bits avocado test works?

Main implementation details:

- The main bits avocado test drives everything
  - Downloads the artifacts.
  - Generates a new bits iso with the artifacts and the ACPI tests that are in QEMU repo.
  - Runs the QEMU VM with the iso generated.
  - The VM automatically runs the tests and pushes the results out using the isa-debugcon at address 0x403.
**Description of the test framework**

**QEMU Repository**
https://gitlab.com/qemu-project/qemu

- **Main avocado bits test that drives it all**
  tests/avocado/acpi-bits.py

- **Tests run from within bios-bits environment**
  (frequent updates)
  tests/avocado/acpi-bits/bits-tests/smbios.py2
  tests/avocado/acpi-bits/bits-tests/testacpi.py2
  tests/avocado/acpi-bits/bits-tests/testcpuid.py2
  tests/avocado/acpi-bits/bits-tests/smilatency.py2

- **QEMU VM**
  booted off with bits ISO running ACPI/SMBIOS tests.

**BIOS-BITS Repository**
https://gitlab.com/qemu-project/biosbits-bits

- **Branches:**
  - **build-scripts**:
    - contains bios-bits build script and Dockerfile (used only when manually building bits)
  - **qemu-bits**:
    - contains main bios-bits source with QEMU test enhancements
  - **master**:
    - original bios-bits source from upstream (not used).

**Build artifacts**
(updates are infrequent)

- **Bios-bits iso with tests**
  to boot QEMU VM with.

- **QEMU VM**
  booted off with bits ISO running ACPI/SMBIOS tests.
Description of the test framework

So why did we chose this design?

Advantages:

▸ No need to go back and forth between two repos. Developers can only use the QEMU repo.
▸ No need to understand how bios-bits work or how it is built.
▸ Quick turnaround in modifying and adding tests and testing changes all from within QEMU workspace.
▸ No need to use submodules.
▸ No need to build entire bits iso - prebuilt artifacts along with modified tests make it a quick process to generate the iso.
▸ It's a simple change to point the test framework to the new artifacts if it is required.
Description of the test framework

So why did we chose this design?

Disadvantages:
- Prebuilt artifacts means it only supports 64-bit x86 iso/test environment at the moment.
- Test is not architecture independent.
  - Supporting non-x86 platforms is non-trivial as bios-bits only supports x86 at this moment.
  - Did it ever support any platform other than x86? Probably not.
- Tool dependencies to build the iso file.

Suggestions for improvement is welcome!

We need contributors to add more tests.
Demo
Focus of the talk

- Why use BIOS BITS to test QEMU?
- Implementation choices for the test framework
- Description of the test framework
- Overview of changes in BIOS BITS for the test to work
Overview of the changes in bios-bits

Major bios-bits updates in the fork

▸ Numerous build fixes for the latest toolchain/compiler.
▸ Changes across all submodules - grub, python, libffi, acpica etc.
▸ A newer acpica that can support newer tables.
▸ Ability to push logs out of the isa-debugcon port 0x403 port in QEMU.
▸ Print logs on the console.
▸ Run tests and quit mode.
▸ Python upgrade to 3.7 would be nice but involves lot of work!
Useful resources

- Bios-bits test framework QEMU documentation
  https://www.qemu.org/docs/master/devel/acpi-bits.html

- ACPI/SMBIOS bios-bits tests in QEMU repo
  https://gitlab.com/qemu-project/qemu/-/tree/master/tests/avocado/acpi-bits/bits-tests?ref_type=heads

- BIOS-bits QEMU fork
  https://gitlab.com/qemu-project/biosbits-bits

- Bios-bits project page: https://biosbits.org/

- Bios-bits upstream source: https://github.com/biosbits/bits

- Josh’s presentation slides

- Josh’s talk on bios-bits: https://www.youtube.com/watch?v=36QlepyUuhg

- Intel’s bios-bits download page https://designintools.intel.com/bios-implementation-test-suite-bits.html
Thank you

Red Hat is the world’s leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.