ARM Servers Meet Industry Standards

ACPI Compliance Testing and LuvOS
Standards for ARMv8 Servers

- From ARM:
  - SBSA: Server Boot System Architecture
  - SBBR: Server Base Boot Requirements
  - PSCI: Power State Coordination Interface
  - [http://infocenter.arm.com](http://infocenter.arm.com)

- From UEFI:
  - UEFI: Unified Extensible Firmware Interface
  - ACPI: Advanced Configuration and Power Interface
  - [http://www.uefi.org/specsandtesttools](http://www.uefi.org/specsandtesttools)

- These all have ancillary standards
How are standards enforced?

- They’re not.
- Unless a great deal of shouting is involved.
- As a practical matter:
  - Self-discipline from vendors
  - Community interaction
  - Participation in standards definition
  - More shouting
- Standards compliance test suites:
  - You’re joking, right?
Specification Compliance Test Suites

- They exist here and there (SCT, BITS, FWTS, …)
- Where we have started at Linaro: luvOS and FWTS
  - luvOS
    - Controlled, repeatable test environment
    - Built around Yocto
    - Runs several test suites automatically
    - [https://01.org/linux-uefi-validation](https://01.org/linux-uefi-validation)
  - FWTS: FirmWare Test Suite
    - Built to catch problems that kept showing up
    - Lots of UEFI- and ACPI-specific test cases
    - Easily extensible
    - [https://wiki.ubuntu.com/FirmwareTestSuite/Reference](https://wiki.ubuntu.com/FirmwareTestSuite/Reference)
What we have added to luvOS

- LuvOS on AArch64
  - How to use luvOS images
    i. If you have your own custom kernel, just replace it in luv-live-image boot partition
    ii. You can compile your custom kernel using CPIO image, but you need to specify the location to store luv results.
  - Where to get luvOS images
    i. pre-built images available on Linaro server
  - How to add test suites
    i. This is same as adding any package to standard OE/Yocto distribution.
  - Source trees
    i. The luvOS git repositories are on GitHub just clone and build it
What we have added to FWTS

- **--arch** parameter to allow cross-testing
- **--acpicompliance** parameter for tests specifically checking for compliance with the ACPI specification
- ACPI compliance tests:
  - MADT fields and all of the subtables and their fields
  - RSDP fields (submitted upstream)
  - FADT fields (submitted upstream)
  - Reduced hardware mode fields within the FADT (submitted upstream)
  - Minor corrections to table and/or field structs
What we have added to FWTS

- **--arch** parameter to allow cross-testing
  
  ```
  $ sudo src/fwts --arch=arm64 -t /path/to/arm/acpi/tables
  ```

- **--acpicompliance** parameter for ACPI specification compliance testing
  
  ```
  $ sudo src/fwts --acpicompliance --arch=arm64
  ```
What we have added to FWTS

$ sudo src/fwts -f --acpicompliance -t ../seattle-tables/v0
Running 3 tests, results appended to results.log
Test: RSDP Root System Description Pointer test.
    RSDP Root System Description Pointer test.              8 passed, 1 failed
Test: MADT Multiple APIC Description Table (spec compliant).
    MADT checksum test.                                     1 passed
    MADT revision test.                                     2 passed
    MADT architecture minimum revision test.                1 passed
    MADT flags field reserved bits test.                    1 passed
    MADT subtable tests.                                    51 passed
Test: FADT Fixed ACPI Description Table tests.
    ACPI FADT Description Table flag info.                  1 info only
    FADT checksum test.                                     1 passed
    FADT revision test.                                     1 failed
    ACPI FADT Description Table tests.                      9 passed, 1 failed
Test FADT SCI_EN bit is enabled.                          1 skipped
Test FADT reset register.                                 1 skipped
What we have added to FWTS

Results generated by fwts: Version V16.01.00 (2016-01-13 08:23:12).

Test 5 of 5: MADT subtable tests.
PASSED: Test 5, MADT revision 3 is defined.
PASSED: Test 5, MADT subtable type 11 (GICC CPU Interface) is defined.
PASSED: Test 5, Subtable 1 of type 11 (GICC CPU Interface) is the correct length: 76
PASSED: Test 5, MADT GICC CPU Interface reserved field properly set to zero.
PASSED: Test 5, MADT GICC CPU Interface, flags, bits 2..31 are reserved and properly set to zero.
PASSED: Test 5, MADT GICC CPU Interface, is using a defined parking protocol version.
PASSED: Test 5, MADT subtable type 12 (GICD GIC Distributor) is defined.
PASSED: Test 5, Subtable 9 of type 12 (GICD GIC Distributor) is the correct length: 24
PASSED: Test 5, MADT GICD GIC Distributor reserved field properly set to zero.
PASSED: Test 5, MADT GICD GIC Distributor GIC version field is in 0..4.
PASSED: Test 5, MADT GICD GIC Distributor second reserved field is properly set to zero.
PASSED: Test 5, MADT subtable type 13 (GICv2m MSI Frame) is defined.
PASSED: Test 5, Subtable 10 of type 13 (GICv2m MSI Frame) is the correct length: 24
PASSED: Test 5, MADT GICv2m MSI Frame reserved field properly set to zero.
PASSED: Test 5, MADT GICv2m MSI Frame Frame ID 0x0 is unique as is required.
PASSED: Test 5, MADT GICv2m MSI Frame, flags, bits 1..31 are reserved and properly set to zero.

================================================================================
56 passed, 0 failed, 0 warning, 0 aborted, 0 skipped, 0 info only.
================================================================================
TODO List

- LuvOS
  - Add netconsole feature
  - Add NDCTL (NFIT defined control) test suite
  - Add support for network boots
  - Port BITS test suite on ARM64. BITS contains some patches which are not yet ported on top of latest ARM64 supported GRUB.
  - Add CHIPSEC test suite on ARM64. This is not yet completely open source. Some codes are under NDA

- FWTS
  - Removing #ifdef __x86__
  - More ACPI compliance tests for more tables
    - Priority is SBBR required tables, anything ARM-centric
  - SBBR-specific tests
  - Investigate version-specific testing
  - [https://git.linaro.org/people/al.stone/fwts.git](https://git.linaro.org/people/al.stone/fwts.git)
HELP!

- There is a LOT to be done!
- If you can share firmware, send us a copy!
- If you can get access to firmware/hardware, run the tests and tell us what you find!
- WRITE MORE TESTS!

...and I’ve finally run out of ‘!’s....