

Moving s390 kvm-unit-tests up the stack

—

Janosch Frank

Software Engineer

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark

symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

The following are trademarks or registered trademarks of other companies.

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Outline

- Current Usage
- Moving up and sideways?
- Initial changes
- Results
- Looking into the future

Current Usage

Current usage

Use Cases

- Testing architecture compliance
- Testing new facilities
- Regression testing
- Error reproduction
- Backport verification

Current usage

Why?

- Small
- Fast
- Dead simple to write

→ Perfect fit for tests without much configuration

Current usage

Environments

- KVM as (a) host
- **Development / Verification**
 - Manual single run (`./s390x-run s390x/test.elf`)
 - Manual full run (`./run_tests.sh`)
 - Mostly as hardware virtualized guests
- **CI**
 - Automated full run
 - Hardware and nested virtualization (VSIE)
 - Only on s390 hardware

Moving up and sideways

Moving up and sideways

Use Cases

- We have a **lot** of testing kernels
- Not all of them run on all hypervisors
- Most of them aren't open-source:
 - Architecture compliance assurance kernels
 - Architecture fuzzing kernels
 - Hand written loops
 - More than one framework similar to KVM unit tests...

Moving up and sideways

Environments

- We have a **lot** of environments to run tests on
 - Hypervisors (LPAR, z/VM, KVM)
 - Emulation (TCG, zPDT)
 - Simulation (lots...)
 - Various firmware levels on hardware also affect instruction execution
- Emulation of a instruction can vary widely depending on environment
- For example z/VM handles page table changes way different than KVM

Moving up and sideways

Goal & expectations

- Let's make kvm unit tests run on the maximum amount of environments!
 - We could uncover implementation differences in KVM
 - Some might lead us to implementation errors
 - In any case we will learn something when uncovering the reason of the difference
- Let's push adoption of kvm unit tests to other teams!
 - Maybe we can replace some hand made tests with the unit test framework
 - The more people use the tests, the more bugs we can find and fix

Initial changes

Initial changes

Boot / IPL

- Unit tests were only bootable with direct ELF kernel boot
 - Very KVM specific
 - Other hypervisors expect 8 or 16 bytes of boot information at specific locations (PSW)
 - BSS is not cleared
- Unit tests expect 64 bit zArch architecture
 - Depending on age of machine that might work out
 - But we want to be able to run tests also on old machines

→ Added binary test generation

→ Added switch to zArch and 64 bit

Initial changes

SCLP Console

- SCLP (console) emulation IRQs:
 - QEMU delivers the IRQ synchronously to the end of the instruction emulation
 - No other s390 hypervisor does that
 - KVM unit tests didn't wait for IRQ
- SCLP EBCDIC support:
 - Most s390 environments speak EBCDIC, KVM uses ASCII...

→ Added external sclp interrupt handling

→ Added EBCDIC output

Results

Results

SCLP Console

- SCLP identified as a low hanging target in initial changes
 - Immense architectural complexity
 - KVM / QEMU implementation was done to “make Linux guests work”
- Extended testing showed LPAR did output different error codes than QEMU / KVM
 - Error priority was wrong
 - Boundary checks were missing
 - Size error reporting was done in a wrong way

Results

Firmware testing

- Firmware developers needed help testing new features on IBM Z 15 before release
 - We ran KVM unit tests under KVM, LPAR and in simulation
 - In simulation we were able to find errors easier than in LPAR
 - Simulation lets us update firmware within seconds
- Improved development speed

Results

STHYI & Storage Keys

- STHYI:
 - KVM was running in a previously untested environment
 - Input for STHYI emulation was different in that environment
 - A STHYI output field was wrongly flagged as valid but contained only zeroes
 - KVM fix

- SKEYS
 - Storage key test under z/VM showed fails
 - I had made false assumptions when writing the test
 - Test was fixed

Looking into the future

Looking into the future

Possible additions

- Being able to run under more environments is great
- But now we need automatic runs for those environments:
 - CI inclusion of the unit tests under LPAR, z/VM and nested KVM
 - CI inclusion of TCG
 - Early execution of the tests for pre-release HW and SW
- Integration of these environments is harder than on KVM
- LPAR for example needs sftp or disk boot for tests
- z/VM could use the virtual punch card reader

Looking into the future

Possible additions

- Nested / VSIE (WIP)
- IO
 - VIRTIO
 - PCI
 - Crypto
- SMP stressors
 - Signaling
 - Yielding

Looking into the future

Caveats

- We might want to or already added a library for new tests
 - With additional library code comes complexity
 - With complexity we lose the ease of use
- We need to make sure that our changes do not affect our fellow developers

Go out and test!

Thank You

