

RISE and Yocto: qemuriscv64 Intermittent Failures Investigation

Overview

The Yocto Project Autobuilder (AB) runs a builds test matrix covering various architectures, toolchains, image types, and feature sets. One of these configurations involves “ptest” (package test) images, which essentially build suitable unit/integration tests from package sources into a minimal image to ensure functionality is retained, and identify any regressions as the project evolves. An example would be building “core-image-ptest-parted” for the parted utility, which is run with the “bitbake -c testimage core-image-ptest-parted” command.

Since enabling qemuriscv64 as a nightly test target, the Yocto Project has observed frequent, but intermittent, timeouts for qemuriscv64 guests running these ptest suites. Failures are usually observed when the host systems are under load from concurrent jobs. Other architectures such as qemux86-64 and qemuarm64 do not experience comparable failure rates, but (unlike qemuriscv64) the AB workers include host systems matching these architectures which can provide kvm acceleration to the guests. This document outlines observations and potential solutions for the qemuriscv64 intermittent ptest issue.

Related Links

- Yocto Project [Autobuilder](#) (AB)
- [yocto-autobuilder2](#) - main AB configuration repository
- [yocto-autobuilder-helper](#) - repository containing helper scripts and configs for the AB
- [Autobuilder Issue Summary](#) - dashboard tracking issues tagged as “AB-INT”.

About Intermittent Failure Rates

The Autobuilder build/test matrix runs daily as well as against any patch series batches being tested before merging (which would average 2 or 3 runs a day). The “frequent” failure rate, using the libpng package as an example, over the month of June is about 17% (5 total failures out of 30 days). Some failures were observed in clusters, while others occurred in a very short time window (one month or less) and never repeated.

These test suites also rarely experience failures on unloaded systems without artificial limiting (e.g. via **taskset**), unless functional regressions have been introduced.

Triage Findings

This section provides an extended look at qemuriscv64 ptest failures over the past year, with some highlights for specific packages. Some occurrences are outliers lacking enough evidence to identify a workaround or fix other than to either increase allocated test timeout limits, or to skip the test cases entirely. The packages detailed here should not be considered an exhaustive list of those prone to timeouts on qemuriscv64, especially as both upstream and Yocto Project-specific development continues.

Unless otherwise noted, the following details are accurate as of July 2026.

Intermittent ptest Failures by Package

Package	Count	First	Last	Notes
glib-2.0	2	03/29/2026	06/23/2026	Also seen on qemuarm64
gnutls	2	01/13/2026	05/02/2026	
gststreamer	5	01/16/2026	06/15/2026	Two different tests fail
libpng	14	03/25/2026	07/22/2026	
libxml2	8	06/20/2025	01/05/2026	increased timeout
parted	24	06/06/2025	07/08/2026	Most consistent fail. Patched
procps	2	05/08/2026	05/20/2026	Not a timeout
python3-cryptography	25	01/16/2026	03/09/2026	runtime buffering change
Rust selftest	2	08/09/2025	05/28/2026	Unusual, kernel segfault

python3-cryptography

- Notable: even under low/no load, qemuarm64 test runs were much faster than qemuriscv64: [comment 13 on issue 16164](#)
 - Many factors - could be a crypto-specific optimization problem, not unique to qemuriscv64 performance itself

- Eventually mitigated by [ptest-runner buffering fix](#)

parted

- Most prolific example. Five failures in June 2026, two already in July.
- Seven AB logs all showing same timeout after **t9041-undetected-in-use-16th-partition.sh** - actual failure likely in **t9042-dos-partition-limit.sh**: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#)
- Workaround: skip **t9042-dos-partition-limit.sh** on riscv64 only

libpng

- Five failures in June 2026
- Numerous log links on [Bugzilla issue](#)
- Timeout occurs in logs after **tests/pngimage-quick** test - this suggests that **tests/pngimage-full** is the failure point
 - Note that on a quick test with latest libpng18 branch (on x86-64 host) shows tests/pngimage-full pass, but tests/pngtest-all fails (passes in qemu riscv64 tests)
- Possible workaround with test timeout bump - needs review of source + Yocto implementation

libxml2

- Blueprint for upstream patching approach - submitted timeout logic upstream, backported [patch](#) for Yocto to carry over interim period
- Approach is non-ideal - adds extra Yocto maintainer burden, masks real issue (and reduces data set for further triage)

glib-2.0

- Similar to libxml2 in that the timeout was increased - this time very early due to being observed on both qemu riscv64 and qemu arm64
- [Fix](#) was “trivial” because the test suite uses gnome-desktop-testing-runner, which has a “-t” option for adjusting timeout.

gststreamer

- Multiple tests in the suite have failed, and it notably appears in the issue summary dashboard for other architectures many times

Long-Term Mitigation

- riscv64 build hosts with hypervisor in AB (through Hetzner) - timeline unknown
 - Puts qemu riscv64 “on par” with other architectures, since it would then have kvm acceleration
 - Depends additionally on official distro support - stability is essential
 - Likely need multiple if native riscv64 builds (i.e., not just runtime tests) are also desired. Yocto matrix includes selftests, multiple deployed distros for thorough coverage

Short-Term Mitigation

In rough order of feasibility (and including applied changes):

1. ptest-runner buffering change - [merged](#)
2. Per-package relaxation of timeouts - possible, implemented for some (e.g. [glib-2.0](#), [libxml2](#)), but only hides the real issue, and some test suites don't support this adjustment without upstream patches
3. Disable/skip problematic tests more aggressively
4. Drop ptests for particularly problematic recipes on RISC-V
5. QEMU optimizations - investigating viability of some changes (e.g. disabling TCG plugins) in Yocto based on feedback from Daniel Barboza. See the [Mitigation Testing](#) section for details.
6. **Provide extra (2-3) dedicated Hetzner servers exclusively for qemu riscv64 builds/tests in the AB cluster? Price has gone up by a factor of five since Yocto started using the service, so it may not be feasible**

Mitigation Testing

Under construction. Don't view the content in this section as exhaustive or final yet

cache=unsafe test

TCG plugins - what is it, how it was tested in source, results for source vs with testimage

Other potential performance gains - most not desirable (e.g. turning off vector), but need outlining anyway

Upstream - too far out-of-scope - expertise no higher than upstream maintainers, unlikely to find novel optimizations for riscv64 specifically

Maybe some per-package upstream optimization to do (e.g. cryptography)?

Runtime Comparisons: qemuarm64 vs qemuriscv64

Unless otherwise noted, these results are taken from **singular manual runs** of each test suite after building the corresponding ptest images on a 12-core AMD Ryzen 7900 machine with 64GB of RAM, where the system is otherwise under no load (other than normal system processes). There is some variance seen over repeated runs, but the idea here is just to get a ballpark runtime length ("DURATION"; marked in bold) when comparing qemuarm64 vs qemuriscv64.

General Notes: Building QEMU with Yocto

The openembedded-core layer contains multiple different recipes and include files for defining our QEMU configuration, including qemu-native (for user mode), qemu-system-native (for system mode), and a qemu recipe itself for on-target binaries. Each of these recipes inherits from one or more include files to build the complete set of configuration, compilation, and deployment options.

The cases described in this document fall under the qemu-system-native branch - we are building core-image-ptest-<recipe> images and using them to perform runtime testing with qemu-system-native, either via the **runqemu** helper script or with Yocto's **testimage** facilities. The data shown below was collected using the former (runqemu).

Package Test Runtime Comparison: Summary

Package	qemuarm64 (s)	qemuriscv64 (s)	Runtime Increase (%)
python3-cryptography	289	478	65
gnutls	456	721	58
parted	325	414	27
libxml2	140	203	45
coreutils	582	1003	72
gzip	47	73	55

python3-cryptography

qemuriscv64

```
=====
Testsuite summary
# TOTAL: 4441
# PASS: 4237
# SKIP: 204
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
DURATION: 478
END: /usr/lib/python3-cryptography/ptest
2026-07-13T16:23
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuriscv64:~#
```

qemuarm64

```
=====
Testsuite summary
# TOTAL: 4441
# PASS: 4237
# SKIP: 204
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
DURATION: 289
END: /usr/lib/python3-cryptography/ptest
2026-07-13T17:34
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuarm64:~#
```

gnutls

qemuriscv64

gnutls test summary:

total: 388
pass : 373
fail : 0
skip : 15

DURATION: 721

END: /usr/lib/gnutls/ptest
2026-07-13T21:26
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuriscv64:~#

qemuarm64

gnutls test summary:

total: 388
pass : 373
fail : 0
skip : 15

DURATION: 456

END: /usr/lib/gnutls/ptest
2026-07-13T17:47
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuarm64:~#

parted

Note: these results omit the recent patch included upstream to skip the t9042 test, so this represents the full suite which has seen failing intermittently

qemuriscv64

Testsuite summary for GNU parted 3.7

```
=====
# TOTAL: 100
# PASS: 86
# SKIP: 14
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

make: Leaving directory '/usr/lib/parted/pctest/tests'

DURATION: 414

END: /usr/lib/parted/pctest

2026-07-13T17:02

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuriscv64:~#

qemuarm64

Testsuite summary for GNU parted 3.7

```
=====
# TOTAL: 100
# PASS: 86
# SKIP: 14
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

make: Leaving directory '/usr/lib/parted/pctest/tests'

DURATION: 325

END: /usr/lib/parted/pctest

2026-07-13T17:56

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuarm64:~#

libxml2

qemuriscv64

Total 514 tests, no errors

PASS: testlimits

DURATION: 203

END: /usr/lib/libxml2/ptest

2026-07-13T17:07

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuriscv64:~#

qemuarm64

Total 514 tests, no errors

PASS: testlimits

DURATION: 140

END: /usr/lib/libxml2/ptest

2026-07-13T20:55

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuarm64:~#

coreutils

Note: This is a “control” test - coreutils has not been seen failing in ways comparable to the other packages in the list

qemuriscv64

Testsuite summary for GNU coreutils 9.11

```
=====
# TOTAL: 731
# PASS: 561
# SKIP: 170
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

make[1]: Leaving directory '/usr/lib/coreutils/ptest'

DURATION: 1003

END: /usr/lib/coreutils/ptest

2026-07-14T15:27

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuriscv64:~#

qemuarm64

Testsuite summary for GNU coreutils 9.11

```
=====
# TOTAL: 731
# PASS: 561
# SKIP: 170
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

make[1]: Leaving directory '/usr/lib/coreutils/ptest'

DURATION: 582

END: /usr/lib/coreutils/ptest

2026-07-14T17:18

STOP: ptest-runner

TOTAL: 1 FAIL: 0

root@qemuarm64:~#

gzip

Note: This is a “control” test - gzip has not been seen failing in ways comparable to other packages in the list

qemuriscv64

Testsuite summary for gzip 1.14

```
=====
# TOTAL: 30
# PASS: 28
# SKIP: 2
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
make[2]: Leaving directory '/usr/lib/gzip/ptest/src/tests'
make[1]: Leaving directory '/usr/lib/gzip/ptest/src/tests'
make: Leaving directory '/usr/lib/gzip/ptest/src/tests'
```

DURATION: 73

```
END: /usr/lib/gzip/ptest
2026-07-14T15:36
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuriscv64:~#
```

qemuarm64

Testsuite summary for gzip 1.14

```
=====
# TOTAL: 30
# PASS: 28
# SKIP: 2
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

```
make[2]: Leaving directory '/usr/lib/gzip/ptest/src/tests'
make[1]: Leaving directory '/usr/lib/gzip/ptest/src/tests'
make: Leaving directory '/usr/lib/gzip/ptest/src/tests'
```

DURATION: 47

```
END: /usr/lib/gzip/ptest
2026-07-14T17:26
STOP: ptest-runner
TOTAL: 1 FAIL: 0
root@qemuarm64:~#
```