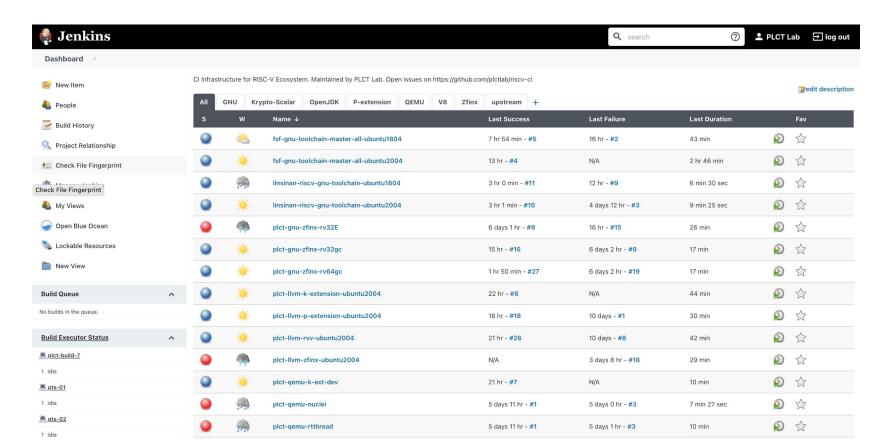
Testing Infrastructure for RISC-V Toolchains & Runtimes

Wei Wu
lazyparser@gmail.com
PLCT Lab

https://ci.rvperf.org

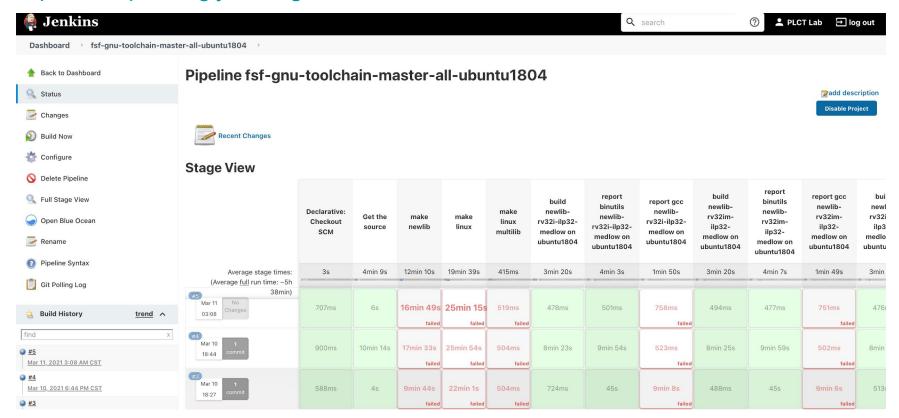
- Jenkins + 5 buildbots
 - Ubuntu 18.04 x 3
 - Ubuntu 20.04 x 2
 - Docker enabled on all buildbots
 - o 16c-128g-4t or 32c-192g-2t
- CI for github.com/riscv/riscv-gnu-toolchain had been established
 - Default master branch
 - o rvv-branch
 - fsf-trunk-daily-build
- Also has build jobs for V8, OpenJDK, and QEMU

Adding new jobs now. Suggestions & Issues are welcome!



Several failures observed

https://ci.rvperf.org/job/fsf-gnu-toolchain-master-all-ubuntu1804/



https://ci.rvperf.org/blue/organizations/jenkins/fsf-gnu-toolchain-master-all-ubuntu1804/5/pipeline/32

Stage Logs (make linux)

```
☐ Shell Script -- ./configure --prefix=$PWD/build-rv64-linux & make -i $(nproc) report-linux (self time 25min 15s)

                                -00 (internal compiler error)
FAIL: gfortran.dg/pr96711.f90
FAIL: gfortran.dg/pr96711.f90
                                -00 (test for excess errors)
FAIL: gfortran.dg/pr96711.f90
                                -01 (internal compiler error)
FAIL: gfortran.dg/pr96711.f90
                                -01 (test for excess errors)
                                     (internal compiler error)
FAIL: gfortran.dg/pr96711.f90
FAIL: gfortran.dg/pr96711.f90
                                -02 (test for excess errors)
FAIL: gfortran.dg/pr96711.f90
                                -03 -fomit-frame-pointer -funroll-loops -fpeel-loops -ftracer -finline-functions (internal compiler error)
FAIL: gfortran.dg/pr96711.f90
                                -03 -fomit-frame-pointer -funroll-loops -fpeel-loops -ftracer -finline-functions (test for excess errors)
FAIL: gfortran.dg/pr96711.f90
                                -03 -g (internal compiler error)
                                -03 -q (test for excess errors)
FAIL: gfortran.dg/pr96711.f90
                                -Os (internal compiler error)
FAIL: gfortran.dg/pr96711.f90
FAIL: gfortran.dg/pr96711.f90
                                -Os (test for excess errors)
               ====== Summary of gcc testsuite =======
                              # of unexpected case / # of unique unexpected case
                                       gcc I
                                                      q++ |
                                                                gfortran |
 rv64imafdc/ lp64d/ medlow |
                                         2 | 28 / 7 | 12 /
                                                                       1 |
Makefile: 920: recipe for target 'report-gcc-linux' failed
make: *** [report-qcc-linux] Error 1
```

Build scripts are open source

- https://github.com/plctlab/riscv-ci/blob/main/riscv-gnu-toolchain-master-ubuntu 1804.jenkinsfile
 - In a separate repo
- https://github.com/plctlab/riscv-gnu-toolchain/blob/fsf-trunk-daily-build/jenkinsfile.ubuntu1804
 - Directly in the repo
- https://github.com/plctlab/PLCT-Toolbox/tree/master/ci.rvperf.org/job-configs
 - o Bash scripts. Obsolete

Help to add more CI for GNU Toolchain

- Physical RISC-V machines/devboards will be available in late April
- More x86 VMs would be added
- One QA staff checks the results and sync to developers
- Need to add more testsuites, benchmarks, and build configs

https://github.com/plctlab/riscv-ci/issues/new

RISC-V All-In-One Project

Wei Wu
lazyparser@gmail.com
PLCT Lab

All-in-one Project in one page

- RISC-V has several ISA extension; extension has different status
- Draft, stable, frozen, ratified.
- Only ratified extensions are accepted
- Extensions in other status have a develop branch

Problem: If you want to use V, P, K, and B extensions, you have to merge them yourself. Very hard work.

Solution: Maintain a series of downstream repos/branches publicly

Roadmap

priv	Linux Kernel 2021H2	Hypervisor TBD.	
un-priv	Toolchain: GNU, LLVM	Simulator: Spike, QEMU	FPGA SoC: TBD.
	2021H1	2021H1	TBD.



Option 2: Create Two Vendor Branch on Upstream github/plctlab/

- Master branch keep current scheme, not accept unratified extension.
- One branch for all non-draft unratified extension.
 - e.g. v 0.9, b 0.92 and zfh 0.1 in this branch
- One branch for all draft unratified extension.
 - e.g. v 1.0-draft, b 0.93-draft in this branch.

The PLCT Lab will maintain a branch for this purpose.

- Pros:
 - Easier to use unratified ext.
- Cons:
 - Need extra effort when certain extension become ratified.
 - e.g. We have V, B and Zfh in branch, but V is ratified now, we need to extract the
 V part from the branch and merge into master.



Policy in Other Open Source Projects

Linux kernel

- Same policy as GNU toolchian, NO unratified extension and custom extension could upstream.
 - RFC patches is welcome for review, but won't merge until ratified.
 - Must maintained in downstream.
 - NO vendor/custom ext. allowed.

Plan: new RISC-V Machine that can enable all ratify/stable/draft extensions.

Qemu

fork a new machine/cpu in 2021Q1, enable V,P,B,K in 2021Q2

- Allow unratified extension upstream master branch, but must be non-draft version.
 - e.g. V-ext v0.9 is OK, V-ext 1.0-draft is not OK
 - Disable by default must enable by option -cpu rv64,x-v=true,vext_spec=v0.9
 - NO vendor/custom ext. allowed.

LLVM

- Allow unratified extension upstream, but must be non-draft version
- Must enable with an extra option: -menable-experimental-extension
- Must explicitly specify the extension version: -march=rv64gcv0p9
- No vendor/custom ext. there, but RISC-V LLVM maintainers are open mind for that.

Open source

- https://github.com/riscv/riscv-gcc/tree/riscv-gcc-experiment
- https://github.com/riscv/riscv-binutils-gdb/tree/riscv-binutils-experiment
- https://github.com/plctlab/riscv-gnu-toolchain/
- https://github.com/plctlab/riscv-qcc
- https://github.com/plctlab/riscv-binutils-gdb
- https://github.com/plctlab/riscv-glibc
- https://github.com/plctlab/riscv-newlib
- https://github.com/plctlab/plct-gemu
- https://github.com/plctlab/llvm-project
- (No kernel developer yet.)

Thanks!

- Issues and Comments are welcome!
- PRs are welcome!
- The downstream branches are never go upstream.
 - So, no copy assignments needed. (IMO)