

How Well Does RISC-V Perform?



Recent Comparisons Against Other Architectures



Jeremy Bennett, Craig Blackmore, Paolo Savini (Embecosm) Hugh O'Keeffe (Ashling)



How to measure compiler performance

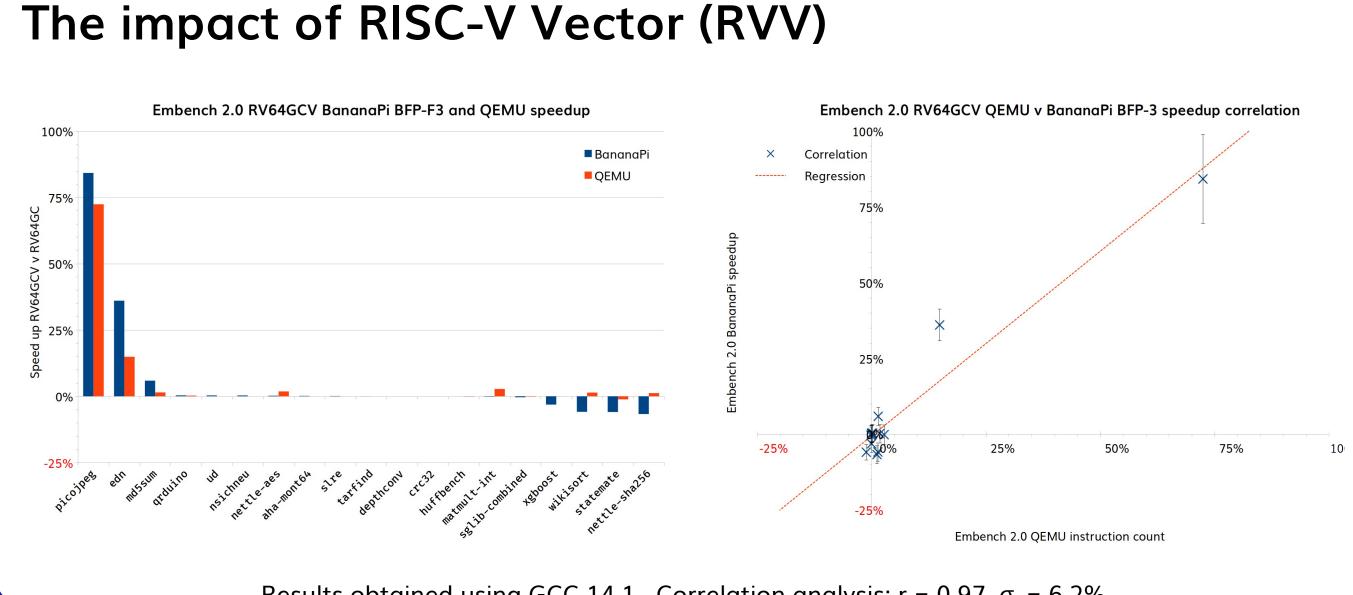
Key benchmark criteria:

- based on multiple real programs, which are open source, or at least readily available
- able to benchmark code speed and code size
- an easily understood and reproducible score, fairly based on the performance of all programs
- updated regularly, to avoid being gamed by compilers.

This year we focus exclusively on Embench and comparison to Arm, primarily for microcontrollers. There are not yet enough application class RV64 platforms to usefully run SPEC CPU, particularly for RISC-V Vector (RVV) 1.0. QEMU is widely used for this, but we show, using Embench, that, the correlation between QEMU instruction count and real hardware performance with RVV is not close enough to be useful for compiler development.

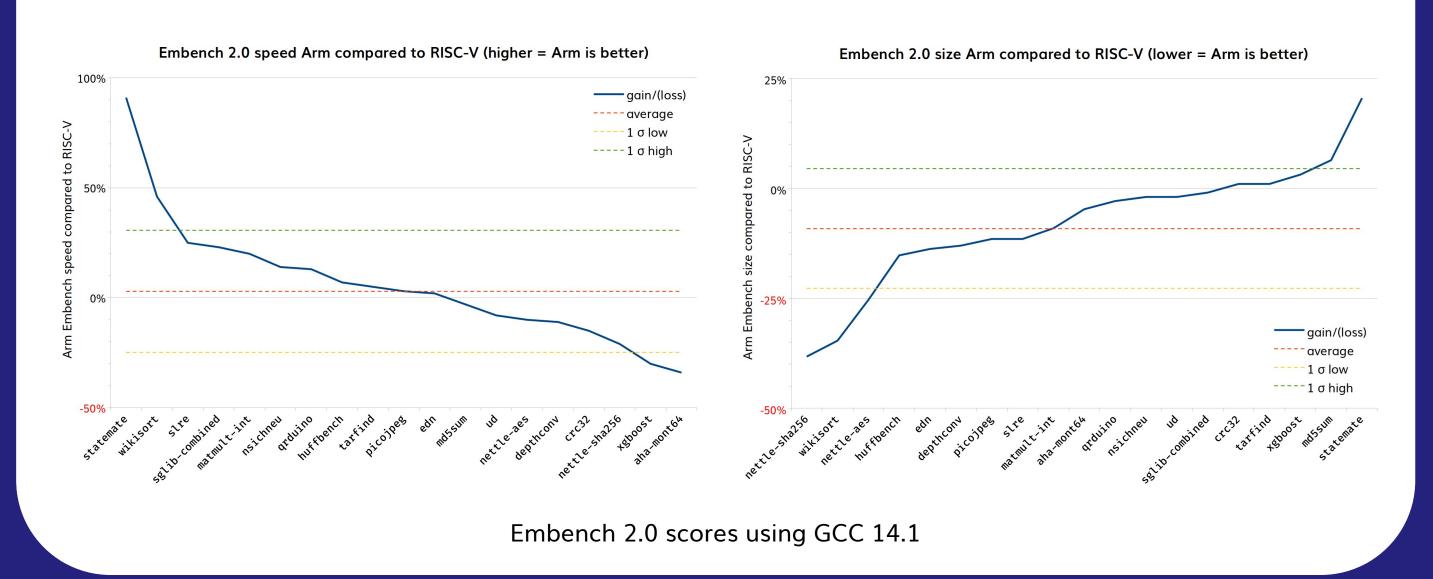
> **RISC-V** CV32E40Pv2 on Nexys A4 FPGA @ 15MHz BananaPi BPI-F3 @1.6GHz

Arm STM32F407 @ 16MHz



Results obtained using GCC 14.1. Correlation analysis: r = 0.97, $\sigma_r = 6.2\%$

GCC Arm Cortex M4 (no FPU) v RV32IMC



Community performance

We can use analysis of git repositories as a way to find out how active communities are. We can find contributors by counting author names or email addresses, constraining that to just commits unique to a particular release branch.

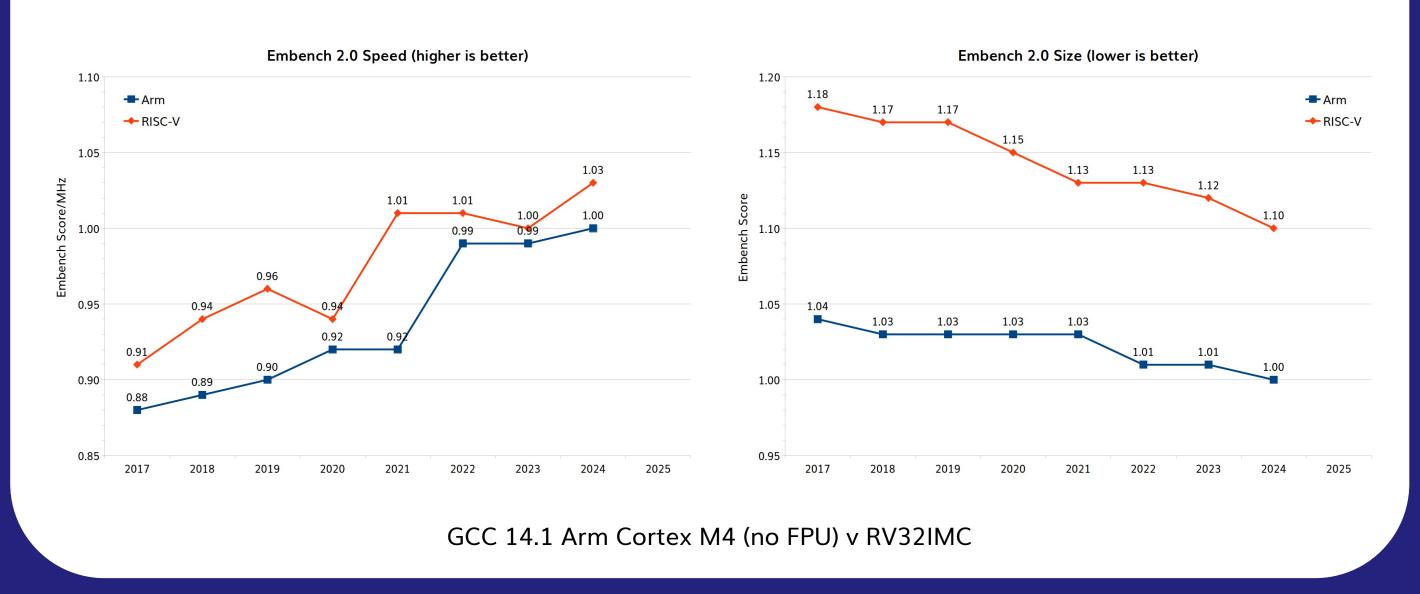
git log --no-merges remotes/upstream/release/20.x ^remotes/upstream/release/19.x \ --pretty="format:%an" | sort -u

git log --no-merges remotes/upstream/release/20.x ^remotes/upstream/release/19.x \ --pretty="format:%ae" | sort -u

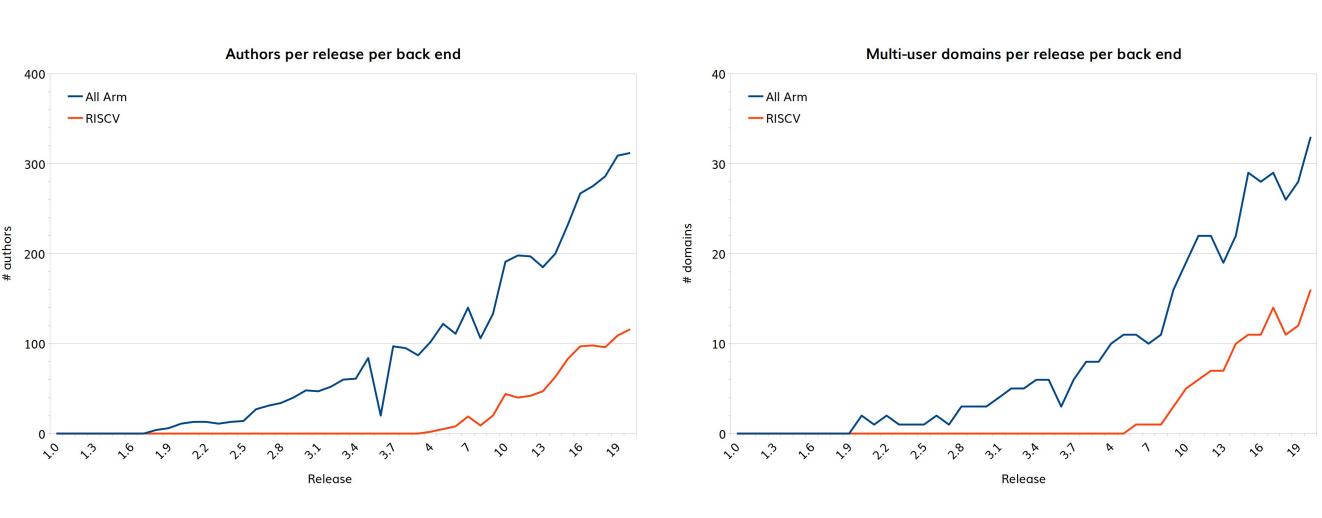
With the email data and some post-processing to remove generic email domains (gmail etc) and any domain with a single contributor, we can get an estimate of the number of different companies contributing.

With slightly different parameters, we can get data by year rather than release. By constraining to particular directories, we can break out the data by front end (different languages) and back end. We can even do semantic analysis on the contributors and the language of their messages to explore how inclusive the community is for different groups.

Compiler performance over time: Arm v RISC-V

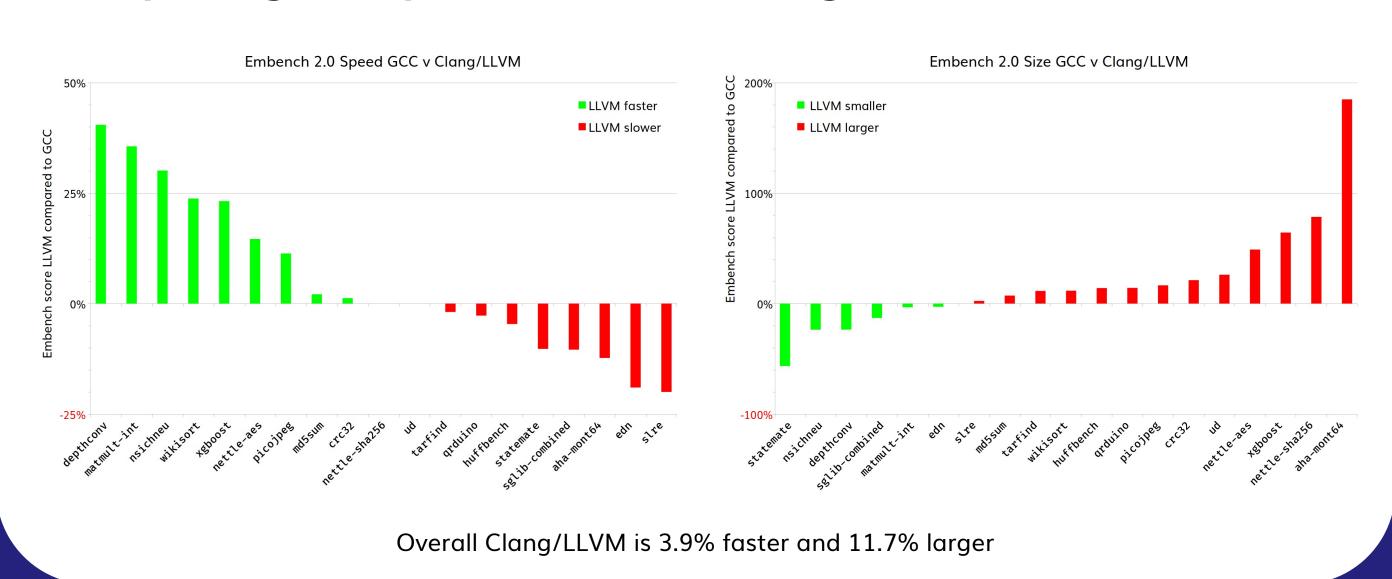


LLVM community activity RISC-V v Arm

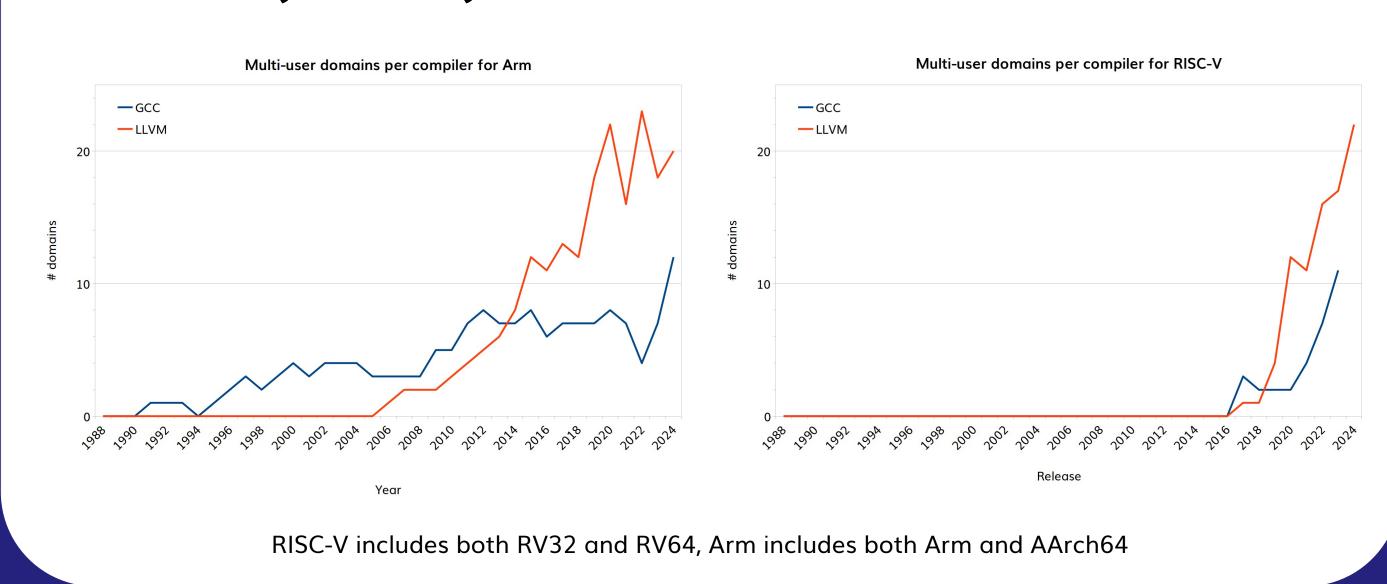


RISC-V includes both RV32 and RV64, Arm includes both Arm and AArch64

Comparing compilers: GCC v Clang/LLVM



Community activity GCC v LLVM



A huge thank you to our collaborators and supporters: The GCC and LLVM communities, the OpenHW Foundation and the Embench community.