Efficient Open-source RISC-V Trace Generation for Enabling Reuse in Computer Architecture Research

Gokulakrishnan Ranghamannar, Gopalakrishnan Srinivasan, Karthik Sankaranarayanan



Dept. of Computer Science and Engineering, Indian Institute of Technology, Madras

Efficient RISC-V Trace Generation

Gokulakrishnan R

Efficient RISC-V Trace Generation

OSCAR 2025

<ロト < 四ト < 三ト < 三ト

3

• **RISC-V Growth**: Two proprietary ISAs, x86 and ARM, dominate the world, but limit extension and customization. RISC-V is rapidly gaining traction as an open-source alternative.

- **RISC-V Growth**: Two proprietary ISAs, x86 and ARM, dominate the world, but limit extension and customization. RISC-V is rapidly gaining traction as an open-source alternative.
- Research Gap: Lack of industry-level toolchains for RISC-V compared to proprietary ISAs.

- **RISC-V Growth**: Two proprietary ISAs, x86 and ARM, dominate the world, but limit extension and customization. RISC-V is rapidly gaining traction as an open-source alternative.
- Research Gap: Lack of industry-level toolchains for RISC-V compared to proprietary ISAs.
- *µ*-arch Simulators: Execution-driven vs Trace-driven.



- **RISC-V Growth**: Two proprietary ISAs, x86 and ARM, dominate the world, but limit extension and customization. RISC-V is rapidly gaining traction as an open-source alternative.
- Research Gap: Lack of industry-level toolchains for RISC-V compared to proprietary ISAs.
- *µ*-arch Simulators: Execution-driven vs Trace-driven.



• **Our Pipeline**: Efficient trace generation for RISC-V μ -arch simulation.

Gokulakrishnan R

Efficient RISC-V Trace Generation

OSCAR 2025

<ロト < 四ト < 三ト < 三ト

3

• Spike:

- Reference functional model of the RISC-V foundation.
- Instruction-level model and hence quite slow.

∃ >

< A > <

э

• Spike:

- Reference functional model of the RISC-V foundation.
- Instruction-level model and hence quite slow.

Dromajo:

- Co-simulation infrastructure for RISC-V.
- Requires modifying the source code.
- Can't generate kernel traces.

• Spike:

- Reference functional model of the RISC-V foundation.
- Instruction-level model and hence quite slow.
- Dromajo:
 - Co-simulation infrastructure for RISC-V.
 - Requires modifying the source code.
 - Can't generate kernel traces.

• FireSim:

- FPGA-Accelerated Cycle-Exact Scale-Out System Simulation.
- Requires RTL.
- Specialized hardware knowledge required to generate application traces.



Gokulakrishnan R

Efficient RISC-V Trace Generation

イロト イヨト イヨト イヨト

3

• **Trace Format**: Sparta defines the Simple Trace Format (STF) and provides a C++ API to use it.

< 4 → <

- **Trace Format**: Sparta defines the Simple Trace Format (STF) and provides a C++ API to use it.
- **Trace-driven Component**: Any microarchitectural simulator that supports the STF trace format.

- **Trace Format**: Sparta defines the Simple Trace Format (STF) and provides a C++ API to use it.
- **Trace-driven Component**: Any microarchitectural simulator that supports the STF trace format.
- **Execution-driven Component**: QEMU full system emulator enables execution-driven high-speed functional modeling for RISC-V.

- **Trace Format**: Sparta defines the Simple Trace Format (STF) and provides a C++ API to use it.
- **Trace-driven Component**: Any microarchitectural simulator that supports the STF trace format.
- **Execution-driven Component**: QEMU full system emulator enables execution-driven high-speed functional modeling for RISC-V.
- **ROI Selection**: We use SimPoint to extract representative regions of interest.



SimPoint

- **Basic Block Vectors (BBV)**: The frequency map of the number of instructions executed in each basic block for a given interval of instructions.
- **Similarity Matrix**: (*x*, *y*) indicates the normalized distance between BBV *x* and BBV *y*.



Figure: Example BBV Similarity Matrix (100M intervals)

Design Choice - Quick Emulator (QEMU)

Image: A matrix and a matrix

æ

• QEMU allows running unmodified guest operating systems and supports emulating different CPU architectures.

- QEMU allows running unmodified guest operating systems and supports emulating different CPU architectures.
- **TCG Plugin Support**: QEMU supports TCG plugins that can register callbacks during code translation and execution

- QEMU allows running unmodified guest operating systems and supports emulating different CPU architectures.
- **TCG Plugin Support**: QEMU supports TCG plugins that can register callbacks during code translation and execution
- Two TCG Plugins:
 - To generate BBVs 1B instruction intervals to be consumed by SimPoint (Link).
 - To generate STF traces 1B warmup instructions per simpoint to be consumed by μ-arch simulator (Link).

6/12

- **Clustering algorithm**: We use both DBSCAN and K-Means algorithms to cluster the BBVs.
- **Distance metric**: To measure the distance between BBVs, we use the cosine distance.
- **Evaluating clusters**: To evaluate the clustering, we use silhouette scores. In case of DBSCAN, we assign outliers to their nearest clusters.

SPEC CPU2017 Results

Benchmark	Icount (Billions)	No. of Clusters	Silhouette Score	Slowdown
500.perlbench_r_checkspam	1489	1	NA	1.421
500.perlbench_r_diffmail	962	3	0.9412	1.458
500.perlbench_r_splitmail	877	3	0.9533	1.394
502.gcc_r_pp.opts-O2	304	7	0.6417	1.421
502.gcc_r_pp.opts-O3	254	6	0.6043	1.438
502.gcc_r_smaller.opts-O3	352	3	0.6617	1.363
502.gcc_r_ref32.opts-O3	349	19	0.664	1.423
502.gcc_r_ref32.opts-O5	242	23	0.5656	1.439
503.bwaves_r_1	521	11	0.8994	1.042
503.bwaves_r_2	822	2	0.8682	1.031
503.bwaves_r_3	641	2	0.8795	1.037
503.bwaves_r_4	780	9	0.8793	1.091
505.mcf_r	885	5	0.7789	1.300
507.cactuBSSN_r	4202	7	0.92	1.065
508.namd_r	2064	18	0.8417	1.059
511.povray_r	4878	1	NA	1.157
519.lbm_r	1485	1	NA	1.068
520.omnetpp_r	1283	1	NA	1.302
523.xalancbmk_r	1321	2	0.9745	1.446
526.blender_r	2143	5	0.9371	1.185
531.deepsjeng_r	2209	2	0.9069	1.349
538.imagick_r	4738	3	0.9965	0.959
541.leela_r	3200	1	NA	1.225
544.nab_r	2089	5	0.9441	1.100
548.exchange2_r	4230	2	0.9042	1.535
549.fotonik3d_r	2796	2	0.8254	1.019
554.roms_r	3300	6	0.8397	1.080
557.xz_r_cld	464	3	0.9737	1.219
557.xz_r_cpu2006docs	1160	18	0.5838	1.529
557.xz_r_input.combined	645	3	0.8255	1.400
Average	1690	5.8	0.83 • • • • • • 1.25 •	
okulakrishnan R	Efficient RISC-V Trace Generation		OSCAR 2025	

8/12

- We achieve an average silhouette score of 0.83 for SPEC CPU2017 rate benchmarks run with the reference input set.
- Instrumentation Speed: Billions of instructions per second.
- Trace Generation Speed: Millions of instructions per second.
- We achieve a two orders of magnitude reduction in the simulation load compared to the full runs.

Trace Generation for Linux Boot

- We generated traces for the boot of Ubuntu 25.04 (Plucky Puffin) image.
- Clustering analysis of this data results in 8 representative clusters with a silhouette score of 0.31.



Figure: Cosine distance similarity matrix for linux boot

Gokulakrishnan R

Efficient RISC-V Trace Generation

OSCAR 2025

Discussion and Future Work

- Work is part of the RISC-V Open Software for Architecture (ROSA) project at IIT Madras.
- RISC-V Trace Generator and Traces (Link):



• Future Work

- Need to address the large sizes of STF trace files (0.5 bytes per instruction recorded on average).
- Predictive capability of the generated trace snippets should be validated against the full application runs.
- For feedback or questions, please contact:
 - Gokul: rgokul.4204@gmail.com
 - Karthik: karthiksankaranarayanan@iitmpravartak.net
 - Gopal: sgopal@cse.iitm.ac.in

Thank You

イロト イヨト イヨト イヨト

æ