Reproducibility Report for SC25 Paper Zero-Value Code Specialization via Profile-Guided Control Data Flow Analysis

Quentin Guilloteau
INRIA
Lyon, France
Quentin.Guilloteau@inria.fr

Abstract

This reproducibility report provides details about the artifact evaluation done with regards to the Artifact Description and Evaluation appendix of SC25 paper Zero-Value Code Specialization via Profile-Guided Control Data Flow Analysis by Xin You, Yi Liu, Yufan Xu, kelun Lei, Depei Qian, Shaokang Du, Hailong Yang, and Zhongzhi Luan. The work was done as part of the Reproducibility Initiative of SC25. The author is a member of the SC25 Reproducibility Committee.

Keywords

Artifact Evaluation, Reproducibility

ACM Reference Format:
Quentin Guilloteau. 2025. Reproducibility Report for SC25 Paper Zero-Value Code Specialization via Profile-Guided Control Data Flow Analysis. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '25), November 16–21, 2025, St Louis, MO, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Overview of Reproduction of Artifacts

The following table provides an overview of each computational artifact's reproducibility status. Artifact IDs correspond to those in the AD/AE Appendices.

Artifact ID	Available	Functional	Reproduced
A_1	•	•	•
Badge awarded	yes	yes	yes

2 Reproduction of Computational Artifacts

2.1 Timeline

The artifact evaluation was conducted from July 21st, 2025 to August 22nd, 2025.

Platform	x86	ARM
CPU	Intel Xeon Gold 5220	Cavium ThunderX2 99xx
Cores	18	64
Memory	96 GiB	256 GiB
System	Debian 11, Linux 5.10	Debian 11, Linux 5.10

Table 1: Hardware and Software configurations used for the reproduction attempt.

2.2 Computational Environment and Resources

The reproduction attempts were carried out on the Grid'5000 testbed[1]. The experiments on `x86` architectures were executed on the `gros` cluster of the Nancy site[2], and the `ARM` architecture on the `pyxis` cluster of the Lyon site[3].

Table 1 summarizes the hardware and software configurations used for the reproduction attempt.

2.3 Details on Artifact Reproduction

2.3.1 Access to the Artifact. The Artifact is composed of several parts, located on GitHub and archived on Zenodo (https://zenodo.org/records/16936151):

- Source Code of ZeroSpec: https://github.com/mSKdowmM/ZeroSpec[4], archived as `ZeroSpec-master.zip` on Zenodo
- Source Code of the "workflow" and its scripts: https://github.com/mSKdowmM/ZeroSpec-cases-X86[5] and https://github.com/mSKdowmM/ZeroSpec-cases-ARM[6], archived as `ZeroSpec-cases-X86-master.zip` and `ZeroSpec-cases-ARM-master.zip` on Zenodo.
- Built Docker containers: `ZeroSpec_X86.tar.gz` and `ZeroSpec_ARM.tar.gz` on Zenodo
- Data gathered from the authors: `ZeroSpecData.tar.gz` on Zenodo

2.3.2 Containers and `Dockerfiles`. The `Dockerfiles` producing the image in the Zenodo archive are in the source code of ZeroSpec. While those images do build, I was unable to rerun the experiments with those images. The `run.sh` script completes but the experiments do not seem to complete:

[1]https://grid5000.fr
[2]https://www.grid5000.fr/w/Nancy:Hardware#gros
[3]https://www.grid5000.fr/w/Lyon:Hardware#pyxis
[4]commit: `9a78479eca22ce14819b67938fd543e23fa7f959`
[5]commit: `9c6620c3366cca8ff7ed7a162225ffa200dfd6fa`
[6]commit: `4e09f2f8059fa70dd955dc7acd23b72a1a4f9640`

SC '25, St Louis, MO, USA

ACM ISBN 979-8-4007-1466-5/2025/11
https://doi.org/10.1145/XXXXXXX.XXXXXXX

Programs	Speedup Paper		Speedup Reproduced	
	x86	ARM	x86	ARM
IS	3.04%	4.67%	0% (1.0)	1.6% (1.016)
UA	1.87%	8.04%	-14% (0.865)	-0.4% (0.963)
backprop	6.31%	21.3%	18% (1.179)	9% (1.087)
QuEST	11.27%	5.20%	6% (1.060)	5% (1.054)

Table 2: Reproduction of Table 4 of the Paper. The number in parenthesis are the raw data generated in the `speedup.csv` file.

```
$ cat speedup.csv
case,speedup
Error: time format is not correct for is.ori
quest, 1.5999999999999999
```

Hence, the `Dockerfiles` present in the source code might not actually be able to generate the same containers are the ones shared on Zenodo.

In the remaining of this reproduction attempt, we used the containers from the Zenodo archive.

2.3.3 Execution. The steps given by the authors to run the artifact are quite straightforward: executing the `run.sh` script.

One shortcoming is the lack of documentation and information about the progress of this script, or the absence of "checkpointing" mechanism. Hence, one will need to run this script to completion without information. In this reproduction attempt, the script ran between 8 and 10h, as written in the AD.

Note that the SPEC CPU 2017 is not present in the artifact and not explained how to integrate with it.

2.3.4 Artifact Outputs. The `run.sh` generates output files in the `results` folder:

- `static.pdf`: Top plot of Figure 7 of the paper
- `dynamic.pdf`: Bottom plot of Figure 7 of the paper
- `slowdown.pdf`: Top plot of Figure 8 of the paper
- `memory.pdf`: Bottom plot of Figure 8 of the paper
- `speedup.csv`: Data for Table 4 of the paper
- `estimate.csv`: Data for Table 5 of the paper
- `estimate_only.csv`: Partial data of `estimate.csv`

One shortcoming is that the script does not produce the CSV file associated for all the pdf files. It should be possible to save the data as csv by modifying the Python files (`scripts/*.py`). This shortcoming does not allow us to simply compare the plots from the paper with the plots from the reproduction, as the runs from the different architectures (`x86` and `ARM`) will each generate a PDF file.

2.4 Reproduced Results

This section shows the generated plots and data from the reproduction attempt and compare them to the figures from the paper.

The data from the reproduction attempt can be found on Zenodo: https://doi.org/10.5281/zenodo.16920003

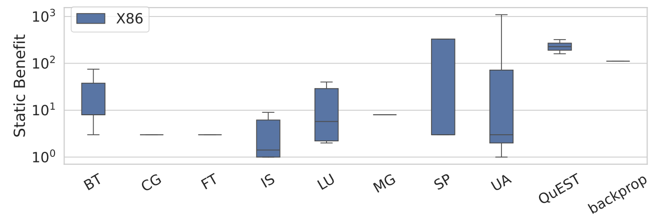

Figure 1: Static Results for `x86` (generated `static.pdf` file)

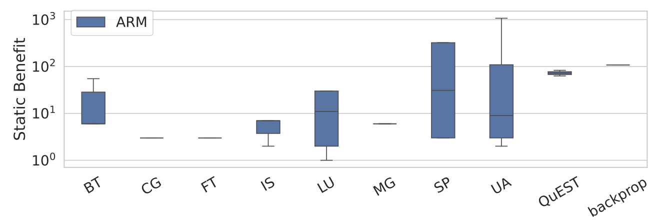

Figure 2: Static Results for `ARM` (generated `static.pdf` file)

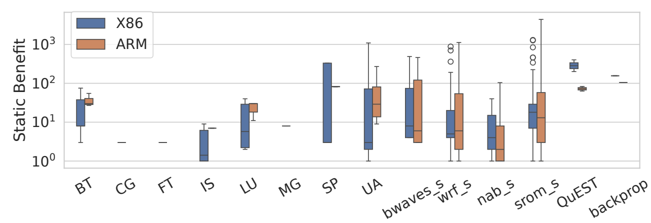

Figure 3: Static Results from the Paper

2.4.1 Table 4 of the Paper. Table 2 depicts the reproduction of Table 4 of the paper.

We suppose the differences come from the hardware differences.

2.4.2 Table 5. Table 3 depicts the reproduction of Table 5 of the paper.

We suppose the differences come from the hardware differences.

2.4.3 Static. Figures 1 and 2 are the reproduced figures of the top plot of Figure 7 from the paper (Figure 3 in this document).

Overall the relative behavior are similar. We might observe higher variation for the ARM results (e.g., `UA` and `LU`).

2.4.4 Dynamic. Figures 4 and 5 are the reproduced figures of the bottom plot of Figure 7 from the paper (Figure 6 in this document).

Note that the y-axis of the generated plots (Figure 4 and 5) are different.

Overall, the results are similar. We do observe greater variations for `ARM` and `SP`.

2.4.5 Slowdown. Figures 7 and 8 are the reproduced figures of the top plot of Figure 8 from the paper (Figure 9 in this document).

Results appear to be similar.

Programs	Benefit				Error			
	Paper		Reproduction		Paper		Reproduction	
	x86	ARM	x86	ARM	x86	ARM	x86	ARM
IS	1.45%	1.14%	0	0	1.59%	3.53%	0.0% (0.0)	-1.5% (-0.015789473684210 57)
UA	0.18%	0.30%	0	0	1.69%	7.74%	13.54% (0.13537117903930118)	3.65% (0.036544850498338 78)
BT	0.48%	0.71%	0	0	-	-	-	-
CG	0.13%	0	0	0	-	-	-	-
EP	0	0	0	0	-	-	-	-
FT	0.69%	0	0	0	-	-	-	-
LU	3.33%	3.06%	0	0	-	-	-	-
MG	0.06%	0	0	0	-	-	-	-
SP	0.01%	0.06%	0	0	-	-	-	-
backprop	29.80%	53.30%	0	0	-23.49%	-32.00%	-17.86% (-0.17857142857142 86)	-8.73% (-0.087301587301587 44)
QuEST	17.07%	8.28%	0	0	-5.80%	-3.08%	-6.02% (-0.060240963855421 55)	-5.39% (-0.053932584269662 964)

Table 3: Reproduction of Table 5 of the Paper. The number in parenthesis are the raw data generated in the `estimate.csv` file.

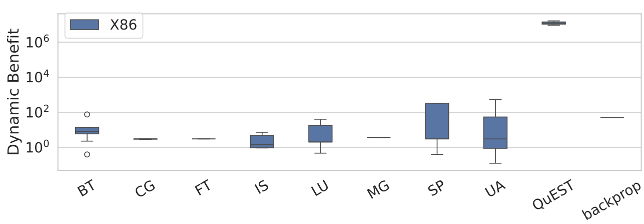

Figure 4: Dynamic Results for `x86` (generated `dynamic.pdf` file)

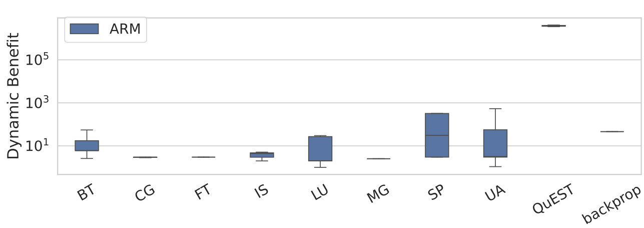

Figure 5: Dynamic Results for `ARM` (generated `dynamic.pdf` file)

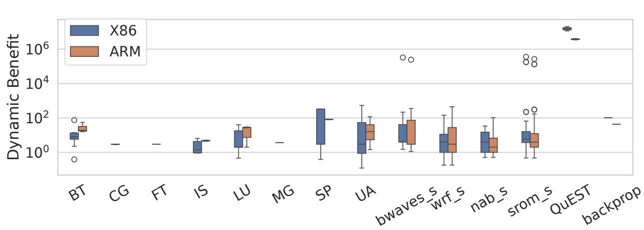

Figure 6: Dynamic Results from the Paper

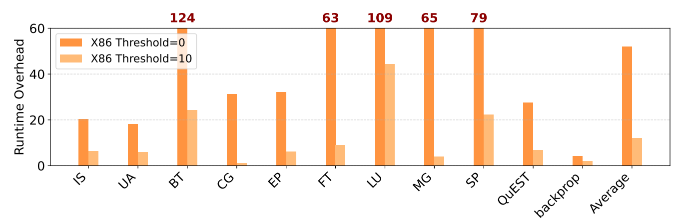

Figure 7: Slowdown Results for `x86` (generated `slowdown.pdf` file)

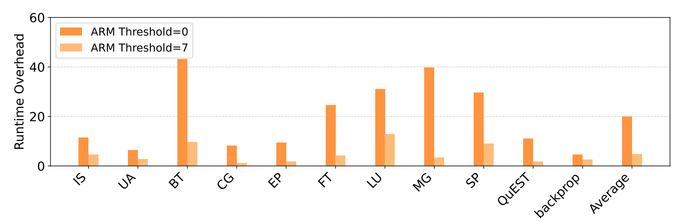

Figure 8: Slowdown Results for `ARM` (generated `slowdown.pdf` file)

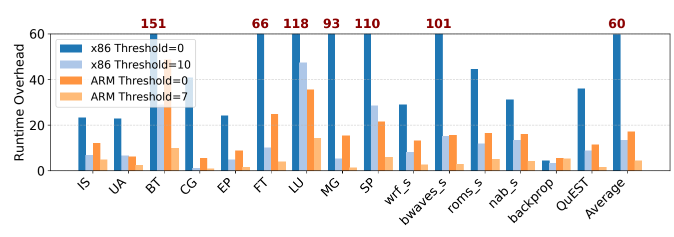

Figure 9: Slowdown Results from the Paper

2.4.6 Memory. Figures 10 and 11 are the reproduced figures of the bottom plot of Figure 8 from the paper (Figure 12 in this document).

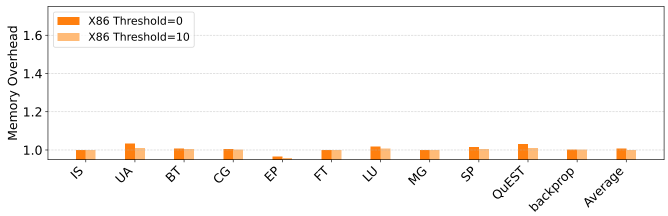

Figure 10: Memory Results for `x86` (generated `memory.pdf` file)

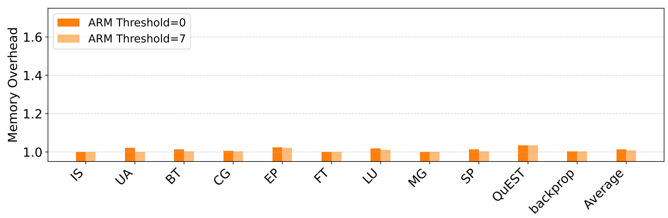

Figure 11: Memory Results for `ARM` (generated `memory.pdf` file)

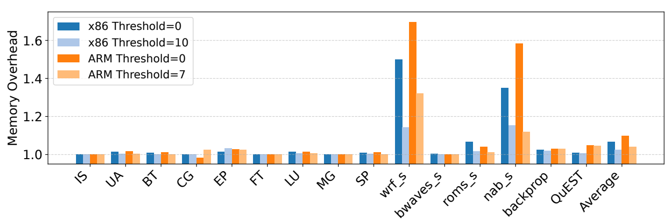

Figure 12: Memory Results from the Paper

Results are similar. The results for `EP` on `x86` are however different, as well as `CG` on `ARM` with a threshold of `0`.

3 Conclusion

3.1 Artifacts Available Badge

The different elements of the artifact are available.

It is however unclear if the `Dockerfiles` shared can actually regenerate the containers shared on Zenodo, or if it was simply a problem on our side (see Section 2.3.2).

3.2 Artifacts Evaluated-Functional Badge

The artifact does not contain the benchmarks from SPEC CPU 2017 which is presented in the paper.

The artifact could be improve to be more "reproducer" friendly, but was enough to regenerate the plots. A future researcher wanting to reuse the work presented in the paper should be able to do so with a bit of effort, but might be limited in the types of variation they could introduce in this artifact (e.g., versions of the compilers).

We suspect that the artifact was crafted especially for the artifact evaluation process and is not exactly what the authors used to run their experiments and generate their plots (see the difference in output between the content of the archive `ZeroSpecData.tar.gz` and what is generated by the artifact).

3.3 Artifacts Reproduced Badge

We managed to reproduce the behaviors presented in Figures 7 and 8 of the paper. The reproduction of Tables 4 and 5 of the paper is however unclear. But, again the SPEC CPU 2017 benchmark was missing.

3.4 Recommendation to the Reproducibility Chairs

In conclusion, and given the criteria of SC25[7], we think that the artifact can be safely be awarded the "Artifacts Available", the "Artifacts Evaluated-Functional". Concerning the "Artifacts Reproduced" badge: we think that the reproduced figures 7 and 8 of the paper are enough to award the badge to the artifact. However, the results from the reproduced Tables 4 and 5 are unclear, and the reproduced Figures 7 and 8 are missing the SPEC CPU 2017 benchmarks.

Acknowledgments

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

Disclaimer: This Reproducibility Report was crafted by volunteers with the goal of enhancing reproducibility in our research domain. The time period allocated for the reproducibility analysis was constrained by paper notification deadlines and camera-ready submission dates. Furthermore, the compute hours in the shared infrastructure (e.g., Chameleon Cloud) available to the authors of this report were limited and restricted the scope and quantity of experiments in the review phase. Consequently, the inability to reproduce certain artifacts within this evaluation should not be interpreted as definitive evidence of their irreproducibility. Limitations in the time allocated to this review and the compute resources available to the reviewers may have prevented a positive outcome. Furthermore, reviewers assess the reproducibility of the artifacts provided by the authors; however, they are not accountable for verifying that the artifacts support the main claims of the paper.

[7] https://sc25.supercomputing.org/program/papers/reproducibility-appendices-badges/