Reproducibility Report for ACM SIGMOD 2023 Paper: "GIO: Generating Efficient Matrix and Frame Readers for Custom Data Formats by Example"

SONSONG MO, Nanyang Technological University, Singapore

We successfully replicated the key performance outcomes presented in the experimental section of the original paper, utilizing the data files, codes, and a set of scripts supplied by the authors. Nevertheless, I encountered difficulties with the scripts intended for figure generation, leading to the partial creation of the figures.

1 INTRODUCTION

The examined paper [1] proposes a GIO framework for automatic matrix/frame reader generation by example. The authors have uploaded the necessary artifacts to GitHub, which qualifies the paper for the Artifacts Available badge. Additionally, they provide scripts to efficiently run the code and reproduce results, meeting the criteria for the Artifacts Evaluated badge. Finally, by comparing the results obtained from these scripts with those in the paper, researchers can achieve the same findings, demonstrating the reproducibility of the results and earning the paper the Results Reproduced badge. However, it is important to note that I encountered issues related to the `tikz` package when generating images on my end. Specifically, while Figures 8, 9, and Table 3 were successfully produced, Figures 10, 11, and 12 could not be generated due to these issues.

2 SUBMISSION

The reproducibility submission for the GIO framework paper, including code, data, and scripts necessary for reproducing the results, is available at the GitHub repository: https://github.com/damslab/reproducibility/tree/master/sigmod2023-GIO-p454. This repository provides all the resources required to replicate the findings of the paper.

3 HARDWARE AND SOFTWARE ENVIRONMENT

In Table 1, we have summarized and compared the experimental environments used in our reproducibility study with those described in the original paper, ensuring a clear understanding.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	AMD EPYC 7302	Intel i9-10900X
Cores	32	20
GHz	3.3	3.7
RAM	128GB	128GB
Storage	960GB SSD, 24TB HDD	1TB SSD, 4TB HDD
System	Ubuntu 20.04.1	Ubuntu 18.04.6
Java	OpenJDK 11	OpenJDK 11
Python	3.8	3.8
C/C++	clang++10	clang++6

4 REPRODUCIBILITY EVALUATION

4.1 Process

We followed the reproduction pathway outlined at https://github.com/damslab/reproducibility/tree/master/sigmod2023-GIO-p454. Except for the Yelp dataset, which requires manual downloading from https://www.yelp.com/dataset/download, all other aspects such as environment setup, dataset downloading, code execution, and image generation can be accomplished by executing the `runAll.sh` script.

4.2 Results

Due to issues encountered during image generation, we were only able to reproduce Figure 8, Table 3, and Figure 9 in [1]. For the other results, our local findings are in agreement with those reported in the original paper.

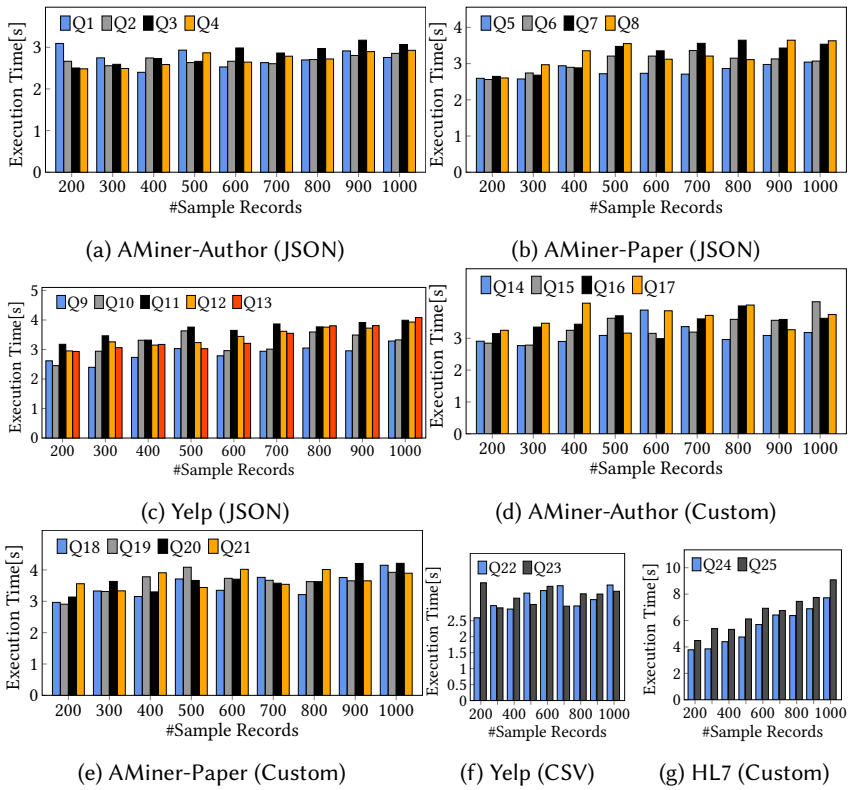


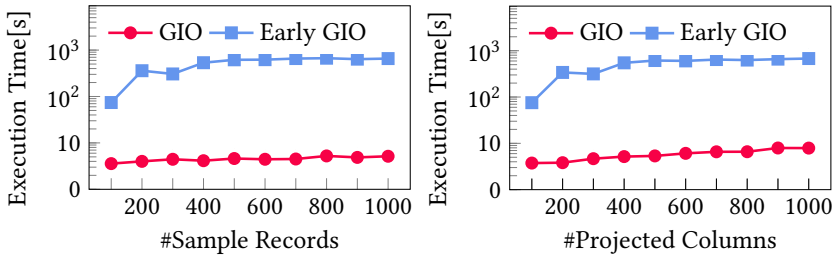
Fig. 1. Reproduction of Figure 8 in [1]: Q1-Q25 Execution Time for Identifying Mapping Rules.

5 SUMMARY

All experimental results were successfully reproduced on our machine, except for some issues encountered during image generation. It is recommended that the authors enhance the image generation scripts to ensure better compatibility and ease of use across diverse environments.

Fig. 2. Reproduction of Table 3 in [1]: Execution Time for Identifying Mapping Rules, from 1k to 10k Sample Records (sec).

#Rows	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25
1K	3.20	3.07	3.56	3.31	3.60	3.64	4.19	3.84	4.04	3.78	4.09	3.78	3.78	2.81	3.81	3.77	3.42	3.04	3.16	3.45	3.30	2.92	2.84	6.80	7.91
2K	3.76	3.97	4.79	3.90	4.32	4.22	4.90	3.90	4.87	4.58	4.60	5.35	5.02	3.48	3.92	4.79	4.04	4.41	4.21	4.60	4.65	3.00	3.26	17.31	16.50
3K	4.40	4.24	4.81	4.34	4.59	5.66	5.76	5.78	4.75	4.42	5.20	5.68	5.41	4.89	5.66	8.22	6.85	6.99	6.70	8.45	7.63	2.98	3.37	24.94	21.48
4K	3.89	5.31	4.52	4.18	4.77	6.36	6.46	5.52	5.02	5.06	6.44	5.43	6.03	10.46	11.68	11.48	12.04	13.64	13.20	14.83	14.92	3.76	4.36	32.20	27.62
5K	4.57	4.77	5.16	4.77	4.61	5.75	6.46	6.26	5.48	5.54	7.61	6.22	7.20	17.17	18.82	22.33	20.17	22.36	23.37	25.40	24.16	3.43	4.11	37.54	33.43
6K	5.05	5.05	5.33	5.41	5.82	6.73	7.00	7.53	6.66	6.13	7.27	6.75	7.79	27.59	31.43	35.08	31.04	37.14	36.14	39.70	38.81	3.91	4.21	49.75	44.57
7K	4.60	4.89	5.44	5.10	5.54	7.07	7.39	7.38	6.89	6.33	8.53	6.94	7.11	42.97	48.16	51.36	47.72	56.38	55.74	57.91	57.84	4.07	4.86	57.94	54.55
8K	4.62	5.27	5.69	5.31	5.67	6.91	7.69	7.31	6.54	6.18	8.56	8.07	7.28	61.42	70.45	70.40	68.99	80.16	80.37	80.12	82.10	4.30	5.20	73.93	69.27
9K	4.74	5.87	5.82	6.22	6.94	7.03	7.66	8.23	7.26	6.59	8.66	7.89	7.46	86.79	98.52	99.69	94.73	110.90	109.62	115.95	115.17	4.40	5.17	90.15	85.61
10K	5.27	5.45	6.21	5.66	6.00	8.04	7.77	7.48	7.13	6.88	8.73	7.62	7.61	121.62	132.25	130.92	128.66	145.69	147.89	155.83	155.34	4.91	5.44	113.69	107.66



(a) #Columns = 200 (b) #Sample Records = 200

Fig. 3. Reproduction of Figure 9 in [1]: Identification Overhead of Different GIO Versions.

REFERENCES

[1] Saeed Fathollahzadeh and Matthias Boehm. 2023. GIO: Generating Efficient Matrix and Frame Readers for Custom Data Formats by Example. *Proc. ACM Manag. Data* 1, 2 (2023), 120:1–120:26.