Reproducibility Report for ACM SIGMOD 2023 Paper: "Efficient GPU-Accelerated Subgraph Matching"

ALEXANDER KRAUSE, Technische Universität Dresden, Germany

The core thesis of the paper was easily reproducible. The authors provided a set of scripts that automatically execute the experiments, collect the data and plot the charts. The resulting figures were near identical to the submitted paper.

1 INTRODUCTION

The paper [1] studies a new approach to process Subgraph Matching, which is a common operation in graph data processing. There is a trend to utilize GPUs to accelerate this operation, however current algorithms tend to suboptimally use this resource and thus often terminate with the dreaded Out-of-Memory (OOM) error. The authors propose a combination of a Cuckoo-trie and alternating Breadth- and Depth-First search.

2 SUBMISSION

The submission contains the original paper and is accompanied by a zipped snapshot of the authors github repository. Within the provided `README.md` one can find precise instructions, which version of what software has to be available.

Further available:

- Shell scripts to automate experiment execution and chart generation
- Data is hosted by the authors: https://hkustconnect-my.sharepoint.com/:f:/g/personal/xsunax_connect_ust_hk/Ei6Ln-BjcgxIi9NiRxd6CAkB2DRr99BF-ig4QvOpeYAwNQ

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 compares our machine to one of the employed servers from the authors. The authors used an 8-GPU environment, we had a single server at our disposal.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel® Xeon® CPU E5-2683 v4 @ 2.10GHz	Intel® Xeon® Gold 6240R CPU @ 2.40GHz
cores	16 (32 Threads)	2x 24 (48 Threads)
RAM	256GB	768GB
GPU	NVIDIA GeForce RTX 2080 Ti	2x NVIDIA Quadro RTX 8000

4 REPRODUCIBILITY EVALUATION

The original experiments ran for about **60 days** within an 8-GPU environment, our equipment featured 2 GPUs. Due to hardware and time constraints, we ran the experiments for the core claims, i.e. Figure 8 and Figure 9 of the original paper.

4.1 Process

The unzipped archive can be deployed as-is on the compute server. We had to manually download and unzip the 6 datasets from the authors' sharepoint and move them to the correct directory. As a sidenote, our previously installed NVCC 11.7.0 was unable to run the code, however upgrading

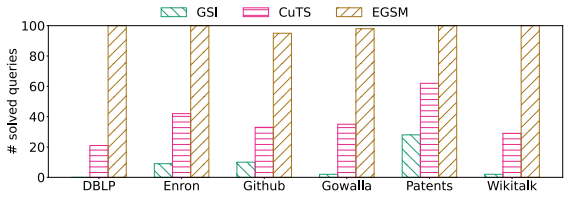


Fig. 8. Number of solved queries.

(a) Original Figure 8

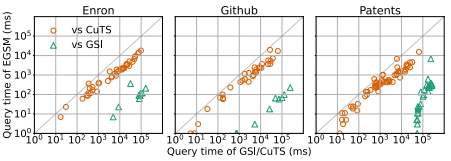
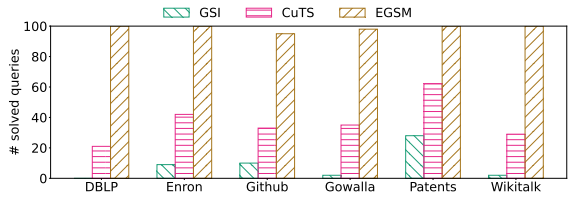
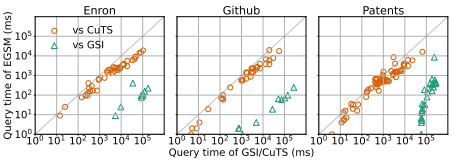


Fig. 9. Comparison on query time.

(b) Original Figure 9



(c) Reproduced Figure 8



(d) Reproduced Figure 9

Fig. 1. Comparison of the original and reproduced core figures

to 11.7.1 made it work. The code is split into three directories, one for each tool. We followed the `README.md` and manually built the code in the specified order. You may need to install `libreadline`, if it is not present on your system. The scripts exactly report the query outcome, i.e. if a query failed (e.g. due to OOM), it prompts **ERROR**. This little quirk can make it a bit irritating to distinguish if there is an actual error in the setup or if this was the expected outcome – which it actually is, since most queries of the competitors die due to OOM. We used screen as a terminal multiplexer to run the experiments in the background. After several days, we reattached to the screen session and obtained our results.

4.2 Results

Given sufficient time, the reproducibility environment replays all queries. We could easily comment-out all experiments, which do not belong to Figure 8 and Figure 9 to limit the overall runtime. Despite some minor runtime deviations between Figures 1b and 1d, we find that the original findings could be reproduced.

5 SUMMARY

The authors put much effort into the automation of the reproducibility setup, which made it overall very easy to recreate the corresponding experiments.

REFERENCES

[1] Xibo Sun and Qiong Luo. 2023. Efficient GPU-Accelerated Subgraph Matching. *Proc. ACM Manag. Data* 1, 2 (2023), 181:1–181:26.