Reproducibility Report for ACM SIGMOD 2022 Paper: "Givens QR Decomposition over Relational Databases"

KONSTANTINOS BITSAKOS, National Technical University of Athens, Greece
DIMITRIOS TSOUMAKOS, National Technical University of Athens, Greece

The reproducibility submission by the authors, their assistance in providing help when needed, and the provisioning of access to required hardware resources have substantially eased the task of reproducing the results of this work. In this report, we describe the steps taken and results that show the agreement in principle and detail of the reproducibility study to the original findings.

1 INTRODUCTION

In this report, reproducibility results are presented for the ACM SIGMOD 2022 paper: "Givens QR Decomposition over Relational Databases" [1] by Dan Olteanu (University of Zurich), Nils Vortmeier (University of Zurich), and Dorde Zivanovic (University of Oxford). The main goal of the paper is to describe and evaluate an algorithm for computing the upper-triangular matrix in the QR decomposition of a matrix that is produced from the natural join over two relational database tables. Specifically, the FIGARO algorithm improves performance and accuracy of the decomposition under specific conditions.

Our efforts have shown that the findings of this work can be fully reproduced and validated in different settings and input sets as those described in the original submission.

2 SUBMISSION

From a code perspective, gcc-10.1, cmake 3.13.4, python 3.7.3, and psql 11.12 are required to run the experiments. Moreover, the Intel MKL 2021.2.0, Boost 1.67.0, Eigen 3.3.7, Openblas 0.3.13, Thread Building Blocks 2018.0, nlohoman json library, and Gtest 1.8.1 C++ libraries must be installed to run experiments. The environment is tested on Linux OS and requires Docker and bash to run scripts. We also note that for *all* experiments to be able to run, a minimum of 500GB of disk space, 192GB of RAM, and 24 physical/48 logical cores are needed.

From a replicating experiments point of view, the reproducibility code is close to an ideal submission. There is a central `repro.sh` bash script, which will: (a) create and set up a new docker container as needed and (b) execute all the experiments at once, downloading/creating data and plotting figures where required. Yet, explicit instructions exist on how to do the setup, data acquisition, and experiments independently, if so desired.

A list with a summary of the submission contents is the following:

- GitHub repository with code and all scripts at https://github.com/Sigmod2022ReproFigaro/figaro,
- A general readme file at https://github.com/Sigmod2022ReproFigaro/figaro/blob/main-branch/README.md,
- The open-source code repository structure at https://github.com/Sigmod2022ReproFigaro/figaro/blob/main-branch/figaro-code/README.md,
- Instructions for setting up and running the system at https://github.com/Sigmod2022ReproFigaro/figaro/blob/main-branch/figaro-code/USAGE.md. This also describes how to download the three real data sets and generate the synthetic ones used in the paper.

Data sets used in the reviewed paper:

- Yelp and Favorita: see https://github.com/Sigmod2022ReproFigaro/figaro/blob/main-branch/figaro-code/USAGE.md

- Retailer: confidential
- Synthetic ones: see https://github.com/Sigmod2022ReproFigaro/figaro/blob/main-branch/figaro-code/USAGE.md

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 presents the Hardware and Software Environment used both in the paper's experimental section and during the reproducibility tests. We thank the authors for providing us with access to a docker container in their infrastructure with the following specifications:

Table 1. Hardware & Software environment

	Paper + Review
OS	Ubuntu 20.04
CPU	Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz
cores	48
Thread(s) per core	2
NUMA nodes	2
RAM	188GB
Storage	196GB

4 REPRODUCIBILITY EVALUATION

4.1 Process

The process that was followed was pretty straightforward and is described in the Submission Section: A Docker container was set up in the authors' infrastructure, and the execution of a subset of experiments was performed. Specifically, we chose a (quite substantial) subset of experiments to be executed (albeit with a smaller number of re-runs due to time limitations), including parts of every experiment (1–4) in the accepted paper. Relative to the issues encountered, we can only note one temporary mishap where the container could not utilize all required storage due to another container running at the same physical node. The authors' response in every matter was exemplary. We also note that all reproduced results have been saved in the reviewers' space.

4.2 Results

Experiment 1 was successfully reproduced in our experiments. Fig. 1 and the tables generated are qualitatively matching the original ones and correspond to the top Figure 4 as well as Figure 5 in the paper. Respective results follow the trends and within small deviations the figures reported in the accepted submission.

Experiment 2 (Figure 6 in the paper) was also successfully reproduced with a simple division required to produce the respective plot. The results follow the trends and are within small deviations compared to those reported in the accepted submission.

Experiment 3 was reproduced relative to the original data only (Table 2 – top part in the paper). In this case, the output comprises three files that must be parsed in order to identify the best and worst join order in each case so that the speedups can be correctly reported. The manually computed (we believe there could exist an easy script for this case) speedups are similar or slightly larger than the reported ones.

Finally, experiment 4 (Table 3 in the paper) was successfully reproduced. We confirm that the collected results match those reported in the accepted submission.

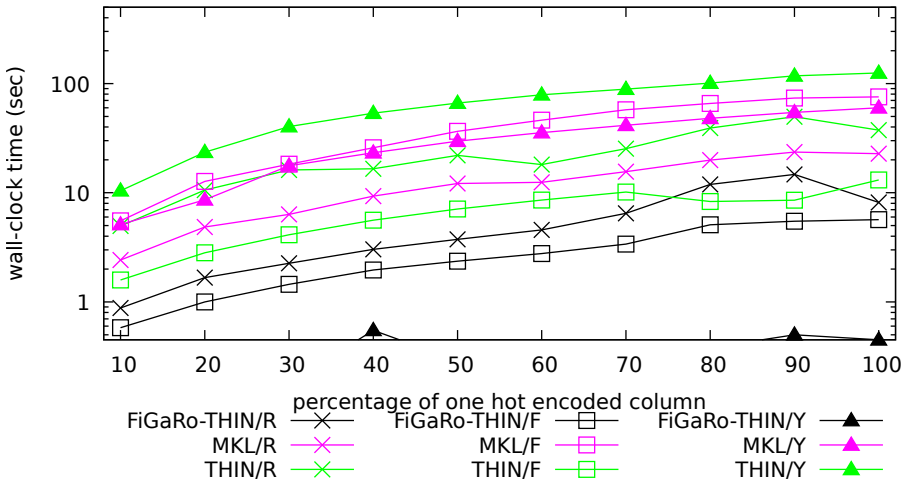

Fig. 1. Figure reproduced that corresponds to Figure 4(top) of the original submission.

5 SUMMARY

In the reviewers' opinion, the authors have made considerable efforts to submit a user-friendly and functional reproducibility package and provide us with the help and access to the hardware needed. Our findings agreed both in principle and detail with the ones reported in the accepted paper, marking this as a fully reproducible submission.

REFERENCES

[1] Dan Olteanu, Nils Vortmeier, and Dorde Zivanović. 2022. Givens QR Decomposition over Relational Databases. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1948–1961. https://doi.org/10.1145/3514221.3526144