Reproducibility Report for ACM SIGMOD 2022 Paper: "WeTune: Automatic Discovery and Verification of Query Rewrite Rules"

ĐORĐE ŽIVANOVIĆ, University of Oxford, United Kingdom

The core thesis of this paper was successfully reproduced. The time performance scales and the number of produced rules are comparable to the ones in the paper.

1 INTRODUCTION

This reproducibility report is for the paper "WeTune: Automatic Discovery and Verification of Query Rewrite Rules" authored by Zhaoguo Wang, Zhou Zhou, Yicun Yang, Haoran Ding, Gansen Hu, Ding Ding, Chuzhe Tang, Haibo Chen, Jinyang Li from the Institute of Parallel and Distributed Systems, Shanghai Jiao Tong University, China and Jinyang Li from the Department of Computer Science, New York University, United States [1]. The core thesis of this paper was successfully reproduced on the below-mentioned machine, which was different from the one used in the paper by the authors.

2 SUBMISSION

The paper results are easily reproduced on the specified OS (Ubuntu 20.04) by following instructions and evaluating the corresponding scripts. The setup of the paper system alongside the competitors is described in the corresponding repository. The submitted content is listed here:

- GitHub repository with code and scripts at https://ipads.se.sjtu.edu.cn:1312/opensource/wetune/. These scripts include the installation of necessary libraries, generating the data, evaluating the experiments, and generating the results in a suitable format.
- Data generators:
  - Synthetic dataset
  - https://ipads.se.sjtu.edu.cn:1312/opensource/wetune/-/blob/repro/testbed/src/main/java/wtune/testbed/runner/GenerateTableData.java;
- Data sources at URL: https://ipads.se.sjtu.edu.cn:1312/opensource/wetune/-/blob/repro/testbed/src/main/java/wtune/testbed/runner/GenerateTableData.java.

3 HARDWARE AND SOFTWARE ENVIRONMENT

In Table 1 is listed hardware used in the paper and used in the reproducibility review.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel E5-2650 v3 CPU	Intel Xeon Silver 4214
cores	20	24
GHz	2	1
RAM	126GiB	188GiB
Storage	SSD	SSD
OS	Ubuntu 20.04	Ubuntu 20.04
Virtualization	Docker	Docker

4 REPRODUCIBILITY EVALUATION

4.1 Process

The initial submission had detailed step-by-step instructions. However, I asked the authors to provide the containerized version of their reproducibility effort as a docker image. This allowed an easy evaluation of the experiments.

Initially, their code had a bug that did not allow the reproducibility of rules used in the optimization of queries. After several rounds of correspondence, the authors managed to fulfill all the requests and fix the bug.

4.2 Results

The main core thesis of the paper is reproducible on the hardware that differs from the one used in the paper. In particular, the following claims are successfully reproduced with results similar to the ones in the paper. I am quoting the whole original sentences from the paper.

(1) WeTune finds 1106 promising and non-reducible rules in 36 hours (on 120 CPU cores in total), among which 32 hours were spent in verification.
(2) For the 50 issues we have studied, WeTune can optimize 76% (38) of them, while MS SQL Server and Calcite can only optimize 46% (23) and 8%(4) of them.
(3) For 23 queries of them, the rewriting performed by WeTune can achieve a 23.8% - 95.2% latency reduction than the rewriting of Calcite itself.
(4) For workload A, WeTune can optimize at least 50% of the queries with more than 10% latency reduction and 17%, 18%, 30% reduction for workload B, C and D, respectively. WeTune can also optimize 13%-21% queries with at least a 90% latency reduction for all four workloads.

The reproduced results of the previously mentioned statements are:

(1) WeTune finds 1053 promising and non-reducible rules.
(2) For the 50 issues we have studied, WeTune can optimize 76% (38) of them, while MS SQL Server and Calcite can only optimize 22% (11) and 8%(4) of them.
(3) For 22 queries of them, the rewriting performed by WeTune can achieve a 23% - 97% latency reduction than the rewriting of Calcite itself.
(4) For workload A, WeTune can optimize at least 50% of the queries with more than 30% latency reduction and 65% reduction for workload B, respectively. WeTune can also optimize 25%-27% queries with at least a 90% latency reduction for two workloads.

5 SUMMARY

The core thesis of the paper is reproducible on the hardware that differs from the one used in the paper. The time performance scales and the number of produced rules are comparable to the ones in the paper. I want to thank the authors for all their effort to address all the suggestions.

REFERENCES

[1] Zhaoguo Wang, Zhou Zhou, Yicun Yang, Haoran Ding, Gansen Hu, Ding Ding, Chuzhe Tang, Haibo Chen, and Jinyang Li. 2022. WeTune: Automatic Discovery and Verification of Query Rewrite Rules. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 94–107. https://doi.org/10.1145/3514221.3526125