Reproducibility Report for ACM SIGMOD 2020 Paper: "Database Benchmarking for Supporting Real-Time Interactive Querying of Large Data"

DOGN XIE, The Pennsylvania State University, USA

Overall, the materials provided by authors are sufficient to reproduce the result of their paper despite there are some small glitches in their early documentations. The authors are responsible and worked close with me on making it more accessible to general users. It would be even better for this paper to provide their data generators rather than fixed data sets so that people can generate more patterns in the future.

1 INTRODUCTION

In this report, I will go through my experience of reproducing results in SIGMOD 2020 paper "Database Benchmarking for Support Real-Time Interactive Querying of Large Data" [1] by Battle et al. In particular, I go through the documented procedures authors shared publicly to run all workload benchmarks on my local machine through a Docker image. Generally, the setup and evaluation procedures are pretty straightforward and automated with some level of flexibility for users to explore. However, there are some small flaws in their original documentation, where I have to work with the authors to fix. On the other hand, it would be better for the authors providing their workload generator so that follow-up researchers can generate various pattern in the future.

2 SUBMISSION

The authors submit a full artifact to reproduce their results, including the source code, workload traces for evaluation, a docker image and a site with its documentations. Automatic benchmark running and report generating scripts are included within the docker image and source code. But, I did not find a script for generating evaluation figures. Data set used in the paper is also made public through an URL, which can be copied to the testing environment later manually.

A list with a summary of the submission contents is also useful. For example:

- Github Repository: https://github.com/leibatt/crossfilter-benchmark-public
- Docker Image Location: https://hub.docker.com/r/crossfilterbenchmark/crossfilter-benchmark
- Detailed deployment and running documentation: https://osf.io/9xerb/wiki/home/
- Data Set in Paper: https://osf.io/9xerb/files/
- Data Generators: N/A

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 lists the hardware and software environment used in the paper and my reproducing effort.

Table 1. Hardware & Software environment

	Paper			Repro Review
	Sever	Laptop	Docker	
CPU	Intel Xeon Silver 4116	Intel i7	N/A	Intel Xeon E5-2697 v4
cores	12	6	N/A	36
GHz	2.10GHz	2.20GHz	N/A	2.30GHz
RAM	20GB	16GB	N/A	256GB
Storage	N/A	SSD	N/A	SSD
OS	RHEL 7	OSX 10.14	Ubuntu 18.04	Ubuntu 18.04 (Docker)

4 REPRODUCIBILITY EVALUATION

4.1 Process

Overall, the process of reproducing the paper's evaluation is relatively easy. I used the Docker image prepared by the authors when deploying the running environment, which is pretty much a single button effort. Data set has to be download manually and copied into the running Docker container, which is less than ideal but still relatively easy. The running process is pretty much just to run two scripts and wait until the results coming out. Besides, the authors also left out enough space for users to tune different running parameters.

However, during my reproducing effort, I have encountered several technical issues including misleading documentation and buggy code. I worked with the authors to fix all of them and now it seems everything works fine. I would expect the author to update their documentation (which has partially done by now), source code, and docker image. My other suggestion for the authors is to release their data generating so that other researchers would have a general sense how all the evaluated dataset is generated and be able to craft their own workloads based on their own needs.

4.2 Results

The paper in this reproducing effort introduces a new benchmark. Thus, I would say the main contribution of this paper is to provide an easy-to-use benchmark framework for different type of DBMSs. On that end, the submitted artifact achieve both easy to deploy and run for all involving DBMS tested in the reproducing procedure. On the other end, this reproducing effort shows similar results (proportional to) what is shown in the paper. The only thing I would suggest to improve is to have a script to generate figures from the reported JSON/CSV results.

5 SUMMARY

Overall, it is a easy and pleasant procedure to reproducing the results of this paper. There are some technical issues I encountered but all of them are solved thanks to the author's commitment. My suggested for the authors at this point is to refine their documentation, code and Docker image, provide a automatic figure generating script, as well as potentially release their workload generator.

REFERENCES

[1] Leilani Battle, Philipp Eichmann, Marco Angelini, Tiziana Catarci, Giuseppe Santucci, Yukun Zheng, Carsten Binnig, Jean-Daniel Fekete, and Dominik Moritz. 2020. Database Benchmarking for Supporting Real-Time Interactive Querying of Large Data. In *SIGMOD*. ACM, 1571–1587. https://doi.org/10.1145/3318464.3389732