Reproducibility Report for ACM SIGMOD 2021 Paper: "EIRES: Efficient Integration of Remote Data in Event Stream Processing"

EMMANOUIL KRITHARAKIS and VASILIKI KALAVRI, Boston University, United States

The main results of the paper are reproducible with a few exceptions that we attribute to using scaled-down datasets for evaluation and differences in the experimental environment.

1 INTRODUCTION

This report describes the reproducibility process for the paper *EIRES: Efficient Integration of Remote Data in Event Stream Processing* [1] by Bo Zhao (Humboldt-Universität zu Berlin), Han van der Aa (Universität Mannheim), Thanh Tam Nguyen (Leibniz Universität Hannover), Quoc Viet Hung Nguyen (Griffith University) and Matthias Weidlich (Humboldt-Universität zu Berlin). After resolving some issues with the code setup and associated scripts, we were able to reproduce the paper's main results with a few exceptions.

2 SUBMISSION

The reproducibility submission consists of the following items:

- A GitHub repository including the EIRES code, scripts, and data at https://github.com/zbjob/EIRES.
- A README file at repository EIRES at https://github.com/zbjob/EIRES/blob/master/README.md. The README file includes instructions on installing prerequisite software and lists the code usage parameters. It also contains a description of the datasets used for evaluation and the scripts provided for running experiments and plotting results.
- The data generators are available at https://github.com/zbjob/EIRES/tree/master/run.
- The data sources are available at https://github.com/zbjob/EIRES/tree/master/data.

After cloning the repository, users need to download and configure the `boost` library. Once this step is completed successfully, users need to compile the EIRES code. After compilation, users can use a script to run all experiments. The script generates output files with latency and throughput measurements. Once the experiments are complete, users need to use another script to post-process the measurements. This analysis script generates another list of files which can then be used as input to plotting scripts.

The provided scripts largely automate the process of data generation and analysis. We ran into a few issues that we were able to eventually solve with the authors' help. We explain these issues and their fixes in detail in Section 4.

3 HARDWARE AND SOFTWARE ENVIRONMENT

We ran all experiments on the Google Cloud Compute Engine, using a `c2-standard-16` VM in zone `us-central1-a`. The environment parameters are shown in Table 1 alongside those used in the paper. The CPU utilization peaked at 16.9%, while the memory and SSD utilization peaked at 1.16% and 4.87%, respectively.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel(R) Xeon(R) E7 -4880	Intel Xeon (Cascade Lake, 2nd Gen)
cores	60	16
GHz	2.5	3.9
RAM	1TB	64GB
Storage	512 SSD	512 SSD

4 REPRODUCIBILITY EVALUATION

4.1 Process

The process of reproducing the results consists of five steps: (1) Configuring the `boost` library, (2) compiling the code, (3) running the code, (4) post analysis, and (5) plotting the results to generate the paper figures in the original paper.

Configuration. The `boost` configuration step initially failed due to a linking error caused by a binary file that was accidentally pushed to the project repository. in particular, we came across the error: *The 'bin/cep_match' failed to be generated based on the stdout message "Makefile:21: recipe for target 'bin/cep_match' failed".* After the authors pushed a fix to the repository, we followed the steps below successfully:

```
> cd EIRES
> wget https://boostorg.jfrog.io/artifactory/main/release/1.72.0/source/boost\_1
> \_72\_0.tar.gz
> tar zxvf boost\_1\_72\_0.tar.gz
> cd boost\_1\_72\_0
> ./bootstrap.sh
> ./b2
```

We then edited the `EIRES/src/EIRES_bushfire/Makefile` and updated the flags `BOOST` and `BOOSTLD` with the path in our machine. Assuming the `EIRES` is in `$HOME`, we set `BOOST = -I $(HOME)/EIRES/boost_1_72_0` and `BOOSTLD = -L $(HOME)/EIRES/boost_1_72_0/stage/lib`.

Compiling the code. We compiled the code without any difficulty following the instructions:

```
> cd EIRES
> sh compile.sh
```

Running the code. To run the experiments, we executed the `run_all.sh` script inside the `run` directory. During our first attempt, we found that the script kept running for nine days, continuously generating output to file `throughput_BL3_greedy_P5_1run.csv`. According to the authors, this file corresponds to the output of evaluating baseline 3 for query P5, which requires weeks to complete. To address this issue, the authors scaled down the data size and the query time window for P5. Further, to speed up the reproducibility reviewing process, they switched off printing event traces in the terminal, and modified the scripts to run the experiments in parallel. They provided two new scripts, `run_synthetic_1.sh` to execute remaining experiments, and `run_BL3-P5-P6.sh` to execute baseline 3.

Post analysis. To perform post-processing, we executed the `analyse_all.sh` script inside the `run` directory.

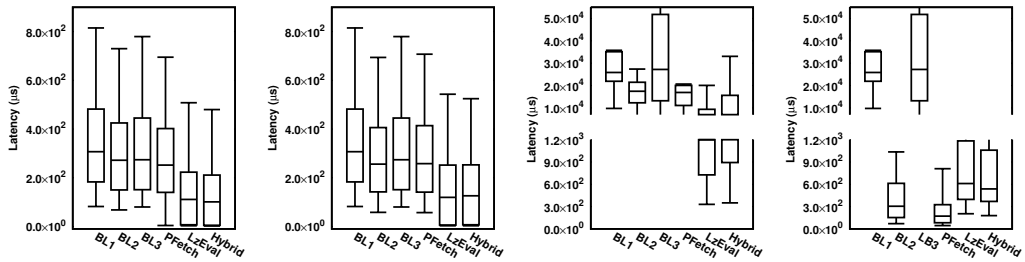

Fig. 1. Reproduced plots for Figure 5 in the paper

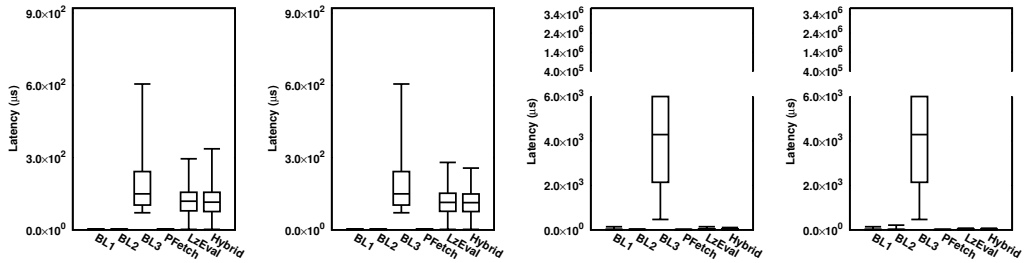

Fig. 2. Reproduced plots for Figure 6 in the paper

Plotting figures. To generate the paper figures, we ran the `plot_all.sh` script inside the `plot` directory. We were able to generate the paper plots with the exception of Figure 10 (a), as files `latency_day_fire_Hybrid_1_run.csv` and `latency_day_fire_LzEval_1_run.csv` were empty after running `analyse_all.sh`. The authors could not provide a solution to this issue before submitting this report.

4.2 Results

After resolving the issues detailed in Section 4.1, we were able to generate most of the paper's results. Next, we describe the main differences and variations in our plots.

Figure 5 shows the overall effectiveness and efficiency for Q1. Our reproduced plots look similar with the ones in the paper, with the exception of cost cache greedy (5c). In this case, the scaling behavior is different for the PFetch, LzEval, and Hybrid columns. The difference is close to one order of magnitude.

Figure 6 shows the overall effectiveness and efficiency for Q2. The plotted value for the BL1, BL2, and PFetch are very low and hardly readable on the reproduced plots. The same is true for LzEval and Hybrid in 5(c) and 5(d).

Figure 8, 9 seem to have an issue with the range of column values. We attribute these issues to the script plotting range configuration. The results of Figure 9 show value fluctuations in all columns. In particular, the latency median values are approximately increased in 9(a) and 9(b) by up to 200ms.

We attribute these variations to the following reasons: (a) scaling down the input data during step 3 of the reproducibility process, (b) using a less powerful machine than the one used for the paper evaluation, and (c) lack of automatic range adjustment in the plotting scripts.

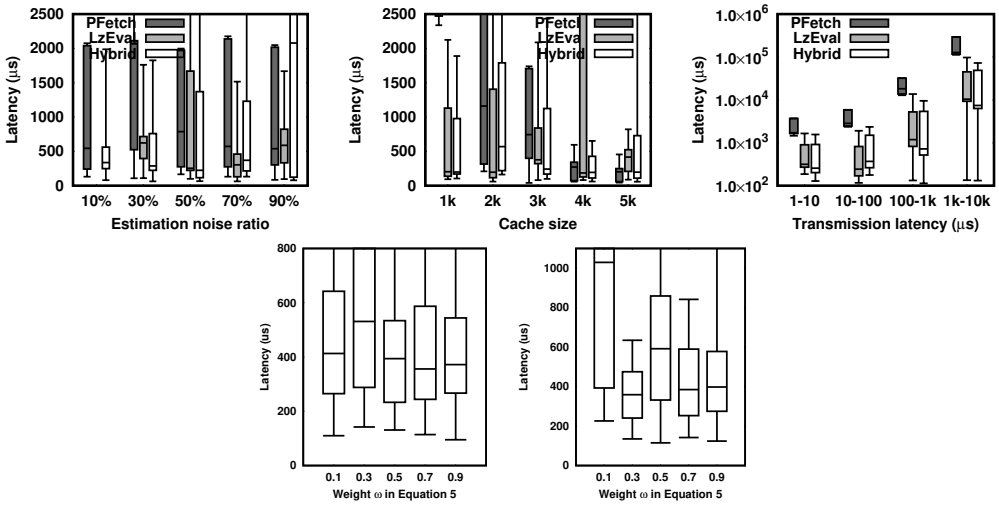

Fig. 3. Reproduced plots for Figures 8 and 9 in the paper

REFERENCES

[1] Bo Zhao, Han van der Aa, Thanh Tam Nguyen, Quoc Viet Hung Nguyen, and Matthias Weidlich. 2021. Eires: Efficient integration of remote data in event stream processing. In *Proceedings of the 2021 International Conference on Management of Data*. 2128–2141.