Reproducibility Report for ACM SIGMOD 2021 Paper: "Parallelizing Intra-Window Join on Multicores: An Experimental Study"

PARIS CARBONE, KTH Royal Institute of Sweden, Sweden

The paper is an experimental study that focuses on the performance of intra-window join operations on top of multicore architectures. All results in this work have been reproduced at a convincing degree. Furthermore, the entry featured fully automated and streamlined scripts with minimal parametrization limited to optional hardware parameters.

1 INTRODUCTION

This reproducibility report concerns the paper Parallelizing Intra-Window Join on Multicores [1], which is joint work between TU Berlin, German Research Centre for Artificial Intelligence, Singapore University of Technology, National University of Singapore, and ByteDance (Singapore). The paper features 50 performance plots, comparing different join strategies over varying workloads, app requirements, and hardware constraints. To that end, the authors propose a reasonable strategy selection scheme based on such parameters. The paper results were all reproduced to a convincing degree, making the overall contributions highly credible and correct.

2 SUBMISSION

The reproducibility submission [2] contained all source code and scripts necessary to run the result. Depending on the multicore hardware available, the main script could be parametrized to reserve a certain number of cores available in the underlying hardware. The main script required root access but it could execute everything in a reasonable manner. In more detail the script involved the following:

- Installing compiler and build packages.
- Downloading the input datasets (from a dropbox target link).
- Generating the simulated inputs for the experiments.
- Preparing the software binaries and system environment.
- Executing all experiment scenarios.
- Plotting all the results.

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 summarizes the hardware specifications of the original paper as well as those used in the reproduction of the results.

4 REPRODUCIBILITY EVALUATION

4.1 Process

The master script really did everything for the purpose. By default, the script would utilize 8 cores but I configured it to work with all 16 physical cores. The configuration and installation took a few minutes. The experiments finished after running overnight for up to roughly 15 hours. The output directories included log files and a large set of plots. Apart from occasional warnings, no errors occurred and everything went smoothly.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel Xeon Gold 612	Intel Xeon Gold 5218
cores	2(socket) x 12	16(32 hyperthreading)
GHz	2.6	2.3
cache	3xL3 19MB	
3xL3 22MB RAM	64GB DDR4 2666MHz	32GB DDR4 2666MHz
Storage	>100GB SSD	1.6TB NVMe SSD
OS	Ubuntu 18 (4.15.0 kernel)	Ubuntu 18 (4.15.0 kernel)

4.2 Results

During the evaluation, all results plotted in the original paper were double-checked. Some of the experiments exhibited reduced durations up to 20% due to the improved CPU speeds in the evaluation. The few small deviations that were observed do not really change the main outcomes of the paper and can be attributed to mere hardware differences (i.e., better cache and faster CPUs in the evaluation hardware). Minor yet noteworthy deviations include:

D1: Figure 8 - Cache Efficiency Profiling (YSB): in Partition (a) L3 misses per input tuple for SHJ-JB and PMJ-JB are significantly higher, close to ½ in ratio of the L2 misses exhibited. Evidently, this is attributed to hardware differences.

D2: Figure 14 - Impact of Window length: (a) SHJ-JM showcased the lowest rate of inputs/msec compared to PMJ-JM but that deviation does not change any important outcome.

D3: Figure 15 - Impact of sorting step (delta) of PMJ. The 30% bar plot showcased the minimum overall measure of cycles per input tuple (in the paper it looks irregularly spiked). This does not change the narrative and in fact, complements the outcome.

D4: Figure 19: Micro-architectural Analysis on Rovio. The execution time breakdown (a) shown in the paper indicates “Memory Bound” operations dominating almost 100% for NPJ, SHJ-JM, and SHJ-JB. In the reproduced experiment we witness 20-40% “core bound” and up to 10% “retiring” operations for these strategies. Nevertheless “memory-bound” is still taking the majority of the time for the same strategies so the outcome is not changed.

D5: Figure 21: Impact of SIMD. The PMJ-JB (AVX) showed higher cycle values in the reproduced execution. This could be attributed to minor hardware pipelining and SIMD execution differences.

5 SUMMARY

Overall most paper claims have been reproduced with minimal effort. This is a stellar entry in SIGMOD reproducibility, setting a high standard on experiment-heavy papers at SIGMOD.

REFERENCES

[1] Zhang, Shuhao, Yancan Mao, Jiong He, Philipp M. Grulich, Steffen Zeuch, Bingsheng He, Richard TB Ma, and Volker Markl. *Parallelizing Intra-Window Join on Multicores: An Experimental Study. In Proceedings of the 2021 International Conference on Management of Data, pp. 2089-2101.* 2021.

[2] Reproducibility Repository (accessed 14Feb 2022) https://github.com/intellistream/AllianceDB/releases/tag/v1