
MILOS NIKOLIC, University of Edinburgh, UK

The experiments are reproducible and support the key findings of the paper. The reproduced figures exhibit similar trends as those in the paper, with raw performance numbers (throughput and speedup) being lower in general, which can be attributed to hardware differences.

1 INTRODUCTION


2 SUBMISSION

The repository containing scripts and reproducibility instructions for this paper is available at:

https://github.com/cirquit/presto

The reproducibility repository contains scripts for setting up the experimental environment, cloning and building the source code, downloading and preparing the datasets1, running the experiments, regenerating all the plots, and recompiling the paper. The repository provides detailed instructions on how to run these scripts in either fully-automated or step-by-step mode.

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 shows the hardware and OS environments reported in the paper, recommended in the reproducibility instructions, and used in the reproducibility evaluation. The difference in the hardware specification affects raw performance numbers, in particular the throughput and speedup in multi-threaded experiments, but not the overall trends and key findings.

<table>
<thead>
<tr>
<th></th>
<th>Paper</th>
<th>Recommended</th>
<th>Repro Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Ubuntu 18.04</td>
<td>Linux</td>
<td>Ubuntu 18.04</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Xeon E5-2630 v3</td>
<td>Intel Xeon E5-2630 v3</td>
<td>Intel Xeon Silver 4210</td>
</tr>
<tr>
<td>Cores (physical)</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>GHz</td>
<td>2.4 GHz</td>
<td>2.4 GHz</td>
<td>2.2 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>78GB</td>
<td>80GB</td>
<td>250GB</td>
</tr>
<tr>
<td>Storage</td>
<td>HHD</td>
<td>NFS (5TB)</td>
<td>NFS (900GB)</td>
</tr>
</tbody>
</table>

1The datasets were hosted on various platforms: Kaggle (Imagenet), zenodo.org (Cube++), lrz.de – Leibniz Supercomputing Centre (openwebtext, commonvoice), tum.de – Technical University of Munich (CREAM), openslr.elda.org (Librispeech).
4 REPRODUCIBILITY EVALUATION

4.1 Process
We followed the step-by-step instructions provided in the reproducibility repository. Using the fully-automated script was impractical due to its lengthy execution of 28 days. The reproducibility evaluation included experiments over seven datasets, which were executed sequentially.

Running each experiment failed on the first attempt. We had to modify the docker file of each experiment to include "/bin/bash" before invoking "./Miniconda3-latest-Linux-x86_64.sh".

4.2 Results
The conclusions of the reproducibility evaluation are as follows.

- **Figure 6: Storage consumption.**
  This experiment is *fully reproducible*.

- **Figure 7: Profiling a synthetic dataset.**
  This experiment is *fully reproducible*.

- **Figure 8: Effects of caching on T4 throughput.**
  This experiment is *reproducible*. The trends are identical. For each pipeline, the throughput is lower than in the paper.

- **Figure 9: Online processing time**
  This experiment is *fully reproducible*. The trends are similar, with longer processing times in general. The effect of app-cache more pronounced than in the paper.

- **Figure 10: Storage consumption v. T4 throughput with compression.**
  This experiment is *fully reproducible*.

- **Figure 11: Multi-threaded scalability**
  This experiment is *reproducible*. The reproduced speedups have the same trend but are in general 20-30% lower than in the paper; e.g., maximum speedup was 4.1x vs. 5.8x from the paper.

- **Figure 12: Speedup at 8000 samples.**
  This experiment is *partially reproducible*. The reproduced speedups have the same trend and are in general 20-30% lower than those from the paper. But one exception is the CV2-JPG pipeline, where the speedups for the ‘unprocessed’ and ‘concatenated’ variants in the paper are around 8x whereas the reproduced speedups are around 4.2x.

- **Figure 13: Speedup of applying RMS to a synthetic dataset.**
  This experiment is *reproducible*. The numpy plot shows a slowdown, while the TensorFlow plot shows a speedup. The maximum reproduced speedup for TensorFlow is around 5x, lower than the 8x speedup reported in the paper.

- **Figure 14:**
  This experiment is *fully reproducible*.

REFERENCES