GIO: Generating Efficient Matrix and Frame Readers for Custom Data Formats by Example[1]

SAEED FATHOLLAHZADEH, Graz University of Technology & Know-Center GmbH, Austria
MATTHIAS BOEHM, Technische Universität Berlin, Germany

This guide provides instructions for replicating the paper's findings. To ensure successful execution, it is essential to meet specific setup requirements.

1 SOURCE CODE INFO

- **Repository:** Apache SystemDS (https://github.com/apache/systemds, commit: 82d9d130861be8e36d37a08c22cdd8d3231de6c2)
- **Reproducibility Repository:** https://github.com/damslab/reproducibility/tree/master/sigmod2023-GIO-p454
- **Programming Language:** Java, `Clang++10`, `Python 3.8`, `SystemDS`
- **Packages/Libraries Needed:** `JDK 11`, `cmake`, `RapidJSON`, `Clang++`, `Python`, `Git`, `Maven`, `pdflatex`, `unzip`, `unrar`, `xz-utils`

2 EXPERIMENT INDEX STRUCTURE

```
.
|── baselines
    |── JavaBaselines
    |── PythonPandas
    +── RapidJSONCPP
|── datagen
|── explocal
    |── exp1_micor_bench_identification
    |── exp2_identification_1k_10k
    |── exp3_early
    |── exp4_micro_bench
    |── exp5_systematic
    |── exp6_end_to_end
    +── plotting
|── load-had3.3-java11.sh
|── plots
|── run0LoadConfig.sh
|── run1SetupDependencies.sh
|── run2SetupBaseLines.sh
|── run3DownloadData.sh
|── run4GenerateData.sh
|── run5LocalExperiments.sh
|── run6PlotResults.sh
+── runAll.sh
```

3 INSTALL AND CONFIG

Hardware and Software Info: We ran all experiments on a server node an AMD EPYC 7302 CPU @ 3.0-3.3, GHz (16 physical/32 virtual cores) with 128 GB, two 480 GB SATA SSDs (system/home), and twelve 2 TB SATA HDDs. All reader experiments utilize a single SSD.

Setup and Experiments: The repository is pre-populated with the paper's experimental results ("`results`"), individual plots ("`plots`"), and SystemDS source code.

To acquire installation and config, follow these steps:

```
./run0LoadConfig.sh
./run1SetupDependencies.sh;
./run2SetupBaseLines.sh;
```

Note 1. "`run0LoadConfig.sh`" is crucial to execute before proceeding with other steps. It loads JVM memory settings and configures Hadoop settings. Please refrain from commenting out this line when running the steps

4 DATA PREPARATION

The datasets will be downloaded and generated automatically, follow these steps:

```
./run3DownloadData.sh;
./run4GenerateData.sh;
```

5 RUN EXPERIMENTS

To execute all experiments, simply run "`run5LocalExperiments.sh`". The experiments will commence, collecting output results.

```
./run5LocalExperiments.sh;
```

Note 2. Our configuration entails running all experiments five times, making it a more time-consuming process.

6 PLOTTING

For plotting, simply run "`run6PlotResults.sh`":

```
./run6PlotResults.sh;
```

Since the experiments are run five times, at the end of the experiments, we will merge all of them and create average results.

7 RUN ALL

The entire experimental evaluation can be run via "`runAll.sh`", which deletes the results and plots and performs setup, dataset download, dataset preparation, dataset generating, local experiments, and plotting. However, for a more controlled evaluation, we recommend running the individual steps separately.

```
./run0LoadConfig.sh
./run1SetupDependencies.sh;
./run2SetupBaseLines.sh;
./run3DownloadData.sh;
./run4GenerateData.sh;
./run5LocalExperiments.sh;
./run6PlotResults.sh;
```

Please be aware that this process may take a considerable amount of time, and you may prefer to have better control over the timing of each experiment.

Note 3. we are clearing the system cache using the command `"echo 3 >/proc/sys/vm/drop_caches && sync"`. Therefore, it is necessary to use `"sudo"` when executing `"runAll.sh"` (e.g., `"sudo runAll.sh"`).

REFERENCES

[1] Saeed Fathollahzadeh and Matthias Boehm. 2023. GIO: Generating Efficient Matrix and Frame Readers for Custom Data Formats by Example. *Proc. ACM Manag. Data* 1, 2 (2023), 120:1–120:26. https://doi.org/10.1145/3589265