

AWARE:

This is the guide to reproduce the paper. To make the execution work there is a few setup requirements. But first it is sincerely recommended to follow the latest guide at: <https://github.com/damslab/reproducibility.git>

To produce a full reproduction of the paper it is expected that one have access to a Spark (v 3.2.0) cluster with Hadoop (v 3.3.1) running Java 11. But any single machine should be able to run small scale experiments covering most of the experiments shown. For baseline experiments

The specifics of the resources used in the paper is: 6 Cluster nodes and a main node with 32 virtual cores and 128 GB RAM each. Local storage requirements is at least 100GB, and distributed HDFS is 2.0 TB.

The machines have similar specifications as m5a.8xlarge in AWS. In there it should be simple to start a spark cluster, but setting such up is not covered in this guide. Note that if one wants to, it has to be able to switch hadoop and spark versions, to run all baseline experiments.

Further dependencies are:

- Java 11 and 8 available on main node
- Maven 3.6+
- Git
- rsync (installed per default on Ubuntu)
- ssh (also installed per default on Ubuntu)
- Python 3.6+
- pdflatex - If you want to make the paper.

1 Verification

First we verify the setup is correct.

```
1 java -version
2 mvn -version
3 git --version
4 python3 --version
```

The output should look something like:

```

1 Me:~/github/reproducibility/sigmod2023-AWARE-p5$
   java -version
2 -versionopenjdk version "11.0.16" 2022-07-19
3 OpenJDK Runtime Environment (build 11.0.16+8-post-
   Ubuntu-0ubuntu120.04)
4 OpenJDK 64-Bit Server VM (build 11.0.16+8-post-
   Ubuntu-0ubuntu120.04, mixed mode, sharing)
5 Me:~/github/reproducibility/sigmod2023-AWARE-p5$
   mvn -version
6 Apache Maven 3.8.3 (
   ff8e977a158738155dc465c6a97ffaf31982d739)
7 Maven home: /home/baunsgaard/maven/mvn
8 Java version: 11.0.16, vendor: Ubuntu, runtime: /
   usr/lib/jvm/java-11-openjdk-amd64
9 Default locale: en_US, platform encoding: UTF-8
10 OS name: "linux", version: "5.15.0-46-generic",
   arch: "amd64", family: "unix"
11 Me:~/github/reproducibility/sigmod2023-AWARE-p5$
   git --version
12 git version 2.25.1
13 Me:~/github/reproducibility/sigmod2023-AWARE-p5$
   python3 --version
14 Python 3.8.10

```

For the distributed parts of the experiments further installs are needed: Verify the install of :

```
1 spark-submit --version
2 hdfs version
```

Output should look like:

```

1 $ spark-submit --version
2 WARNING: An illegal reflective access operation has occurred
3 WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/sbaunsgaard/spark-3.2.0-bin-hadoop3.2/jars/spark-unsafe_2.12-3.2.0.jar) to constructor java.nio.DirectByteBuffer(long,int)
4 WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
5 WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
6 WARNING: All illegal access operations will be denied in a future release
7 Welcome to
8
9      /---\
10     /___\__ --- ----/ ___\
11    _\ \/_ - \/_ - '/ ___\ '_/
12   /___/ .--/\_-/_-/ /\_\_\ version 3.2.0
13  /_/
14
15 Using Scala version 2.12.15, OpenJDK 64-Bit Server VM, 11.0.13
16 Branch HEAD
17 Compiled by user ubuntu on 2021-10-06T12:46:30Z
18 Revision 5d45a415f3a29898d92380380cfd82bfc7f579ea
19 Url https://github.com/apache/spark
20 Type --help for more information.
21
22 $ hdfs version
23 Hadoop 3.3.1
24 Source code repository https://github.com/apache/hadoop.git -r
25   a3b9c37a397ad4188041dd80621bdeefc46885f2
26 Compiled by ubuntu on 2021-06-15T05:13Z
27 Compiled with protoc 3.7.1
28 From source with checksum 88
29   a4ddb2299aca054416d6b7f81ca55
30 This command was run using /home/hadoop/hadoop
31   -3.3.1/share/hadoop/common/hadoop-common-3.3.1.jar

```

If any of the parts are missing or returns errors then please install the missing components. For our setup it is a further advantage if you are able to switch the spark and hadoop version to be able to run CLA baselines. If it is not possible to switch then all the experiments will not work.

From this point Code is run inside the experiments folder

2 Install

Next we install SystemDS, SystemML, and a python virtual environment to run python.

```
1 ./install-all.sh
```

To verify the install we run a few simple scripts.

```
1 ./verify-install.sh
```

Output should be like:

```
1 SYSTEMDS
2 22/09/09 16:51:48 INFO api.DMLScript: BEGIN DML run
   09/09/2022 16:51:47
3 22/09/09 16:51:48 INFO api.DMLScript: Process id:
   725211
4 7.000 4.000 4.000
5 4.000 1.000 8.000
6 7.000 7.000 9.000
7
8 SystemDS Statistics:
9 Total execution time: 0.065 sec.
10
11 22/09/09 16:51:48 INFO api.DMLScript: END DML run
   09/09/2022 16:51:48
12 SYSTEMML
13 22/09/09 16:51:49 INFO api.DMLScript: BEGIN DML run
   09/09/2022 16:51:49
14 22/09/09 16:51:49 INFO api.DMLScript: HADOOP_HOME:
   /home/hadoop/hadoop-2.7.7
15 4.000 7.000 4.000
16 4.000 4.000 1.000
17 8.000 7.000 7.000
18
19 SystemML Statistics:
20 Total execution time: 0.024 sec.
21 Number of executed MR Jobs: 0.
22
23 22/09/09 16:51:50 INFO api.DMLScript: END DML run
   09/09/2022 16:51:50
```

3 Parameters

Before starting the experiments or downloading datasets, we suggest to go through the settings to configure the execution of the experiments.

While it is possible to run everything out of the box, in one go, we suggest to go through some setting first in:

experiments/parameters.sh

Here it is possible to change what version of systemDS to install. What directory to run the experiments in. Settings for remote synchronization of results. Change parameters for JVM. And much more.

4 Data Preparation

create datasets.

5 Micro benchmarks

6 local execution

7 distributed execution

8 plotting

9 compilation of paper