# A. Artifact description

Artifact location: `https://github.com/lcchen008/irreg-simd`

## A.1 Abstract

The engineering work related to our paper mainly contains three sets of codes: *Irregular Reductions*, *Graph Applications*, and *Sparse Matrix-Matrix Multiplication (SpMM)*. More specifically, *Irregular Reductions* include *Moldyn* and *Euler*; *Graph Applications* include *SSSP* and *PageRank*; *SpMM* contains itself, but with both `float` and `double` typed implementations. We provide scripts to automate the process of data preparation and execution. Input datasets do not come with the code, but will be generated or downloaded (from a public repository) when the data preparation script is executed. The overall data preparation and execution time takes a few hours. Data preparation (conducted on the CPU) involves generation/download, and transformation of orginal data into different replicas using different configurations that are tested in our paper, and this process takes less than an hour. The binaries for the major programs (where our algorithms are implemented) will be executed on an Intel Xeon Phi. The data loading during program execution takes much longer time than that on CPU. If you are evaluating our work on a cluster, you can submit our scripts as batched jobs. If you run our scripts interactively on the console, you will see the results printed by our program highlighted in different colors. The minimum requirement for evaluators is that they should have an Intel Xeon Phi installed on a CPU machine, or have an account to a supercomputer that has Xeon Phis installed. We prefer the evaluator to have such an account to access a supercomputer (such as Stampede), which has the full stack of software and hardware that are needed.

## A.2 Description

## A.3 Check-list (artifact meta information)

### A.3.1 Data Preparation:

*data-preparation.sh:* After running this script, all the input generation/downloading and transformation will be done. All applications will be ready to run. The comments in the script are pretty self-explainary. All binaries invoked in this script runs on CPU. This script runs for about 2600 seconds.

### A.3.2 Programs:

- **Irregular Reductions:** moldyn (*run-moldyn.sh*), euler (*run-euler.sh*)
- **Graph Applications:** sssp (*run-sssp.sh*), page_rank (*run-pagerank.sh*)
- **SpMM:** spmm-float-and-double (*run-spmm.sh*, including spmm-float and spmm-double)

### A.3.3 Binary:

Binaries of the programs will be built when you run the *run-*.sh*.

### A.3.4 Data set:

- Inputs for Irregular Reductions will be generated to folder *./input*.
- Inputs for Graph Algorithms will be downloaded to *./datasets*.
- Inputs for SpMM will be downloaded to *./spmm-float-and-double/datasets*.

## A.4 Software/Hardware Requirements and How to Run

### A.4.1 Software Requirements:

- **Compiler:** You should have ***icc*** and ***g++*** installed.
- **Run-time Environment:** The operating system you are using should be a *Unix* system supporting *bash* scripts. It should also have *Python* execution environment installed, as some of our data transformation algorithms are implemented in Python. It should also have Intel Xeon Phi development tool and environment installed. *Internet access* is a requirement because several of the datasets will be downloaded from the internet.

### A.4.2 Hardware Requirements:

- **CPU:** used for data preparation, should have main memory of at least 10 GB, and the disk space of at least 60 GB.
- **Intel Xeon Phi (SE10P):** for program execution and evaluation. It would be best that the evaluator has a **TACC supercomputing center account**, and has access to the *Stampede* cluster, which was where we collected our experimental results.

### A.4.3 Execution Steps:

- Unzip our tar ball, and go to our folder.
- Run *./data-preparation.sh* for preparing data for all the following executions.
- Run *./run-moldyn.sh*, and *./run-euler.sh*, for evaluation of irregular reductions.
- Run *./run-sssp.sh*, and *./run-pagerank.sh* for evaluation of graph applications.
- Run *./run-spmm.sh* for evaluation of SpMM on both `float` and `double` types.
- Make sure that any *run-*.sh* is ran after the *./data-preparation.sh* is finished. If you are evaluating our work on TACC cluster, you can use sbatch to submit the above scripts (each script is ready for submission as the job arguments have been written at the top), also make sure that any run-*.sh is submitted after data-preparation.sh job is finished.

### A.4.4 Output:

The output are execution times. If you run the scripts in command line, you will see the results highlighted in RED on your console.

## A.5 Datasets

- **Moldyn:** generated by the data generation program distributed with original Moldyn code.

  *30-3.0r*

  *45-3.0r*
- **Euler:** from UFL sparse matrix collection.

  *gsm_106857*

  *kron_g500-logn19*
- **SSSP:** from SNAP (Stanford Large Network Dataset Collection).

  *soc-pokec*

  *higgs-twitter*
- **PageRank:**

  *soc-pokec*

  *higgs-twitter*
- **SpMM:** As we claimed in our paper, our approach works well for clustered non-zero distribution matrices, and we only include the evaluation using matrices with clustered non-zero distributions.

  *conf5_4-8x8-05*

  *crankseg_1*

*msc10848*

*shipsec5*

### A.6 Installation

Just unzip our tar ball. No installation is needed, as long as you have the right hardware and software environment to compile and run our programs.

### A.7 Evaluation and expected result

The expected results should be consistent with the data presented in our paper.

### A.8 Note

- We did not include the evaluation of all the tile sizes that are presented in our paper, as generating input and evaluating these tile sizes take too long time. We only keep selected tile sizes. If the evaluators are interested in running other tile sizes, we will provide instructions upon request.
- If you have any issues or questions related to either our proposed ideas or the implementations, please contact us, using the contact methods listed under the title of our paper.