Reproducibility Artifact

An Exact Algorithm for the Linear Tape Scheduling Problem

This document provides all the details to reproduce the performance evaluation presented in the paper entitled "An Exact Algorithm for the Linear Tape Scheduling Problem". The complete artifact can be downloaded online: https://figshare.com/s/80cee4b7497d004dbc70.

1 'input' folder

This folder contains the input data used as input for the algorithms used in the performance evaluation of the paper. This dataset is freely available at https://figshare.com/s/a77d6b2687ab69416557. The reader is invited to refer to the description document attached to the dataset for comprehensive details about the dataset used for the performance evaluation, and how it has been generated. The folder 'inputs' contains the index of the files requested on a tape, associated to the number of requests on these files. The folder 'tape' describes the position and size of the files on a tape. Both folders are used as input of the algorithm (see **code** folder).

2 'code' folder

This folder contains a Python implementation of our algorithms, alongside the ones adapted from [1] used for baseline comparison. We carefully implemented the different strategies in a Python code, that are available in the *algorithms.py* file. The *main.py* file is dedicated to the execution of all algorithms on all the instances of the **input** folder. It directly parses the different files in the **input** folder to instantiate 4 different parameters of the algorithms:

- **files_requested**: the list of requested files on the tape, comes from the `index` column in the *input/requests/TAPEXXX.txt* files.
- **request_numbers**: the number of requests of each file in the above list. Extracted from the `nb_requests` column in the *input/requests/TAPEXXX.txt* files.
- **tape**: the list of all file sizes on the tape. Extracted from the `segment_size` column of the *input/tapes/TAPEXXX.txt* files/
- **right**: a list of the right ordinate of each file in **tape**. Obtained by computing the cumulative sum of tape

We also provide in the *draw.py* file a visualization tool of the device head trajectory depending on the list of detours given by the algorithms. This tool is automatically called in 'main.py' for each input and each algorithm.

To start the performance evaluation, one should just go into the **code** folder, and start the program using the makefile:

```
cd code ; make
```

It requires to have a python3 program on the machine. It can easily be installed on any Unix-like machine using the following command

```
sudo apt-get install python3
```

The performance evaluation in the paper uses Python3 version 3.9.2. The code has been executed on a cluster using two Intel Xeon Gold 6130 CPUs per node, with 16 cores each. The execution of the algorithms have been performed sequentially on a single core of a dedicated node to avoid external disturbances.

3 'Run' folder

This folder contains the performance of the different strategies evaluated in the paper. For each algorithm, we recorded the cost induced by the list of detours in output and the simulation time to get the solution. We tested three different values of the U-turn penalty, that is a parameter:

- 0: no penalty
- 14254750000: it represents half of the average size of a tape segment according to our 169 input tapes.
- 28509500000: it represents the average size of a tape segment according to our 169 input tapes.

The file *results.csv* summarizes the cost of the list of detours induced by each algorithm, associated to the time-to-solution to get this list for each of the three penalties above presented. We also record the lower bound for each algorithm on each input.

4 'Figure' folder

This folder contains a R script to reproduce the figures presented in the paper. This script processes the *run/results.csv* to output the performance evaluations that were reported.

References

[1] Carlos Cardonha and Lucas Correia Villa Real. Theoretical and practical aspects of the linear tape scheduling problem. *CoRR*, abs/1810.09005v1, 2018. URL http://arxiv.org/abs/1810.09005v1.