Readme for reproducibility submission of SIGMOD'21 paper ID 68

1 Source code info

Repository: https://github.com/Yanqing-UTAH/ATTPCode
Programming Langauge: mainly C/C++, some of the scripts are written in Python
Required software/library: gcc/g++ >= 8 (requires support for -std=gnu++17), make, lapack and lapacke, blas and cblas, fftw3, python3, numpy, scipy, sklearn and the common shell programs (bash, grep, sed, and etc.).
Optional software/library: for plotting figures: jupyter notebook, python3 matplotlib, pandas, numpy; for recreating "configure" script if that does not work in your environment: m4, autoconf, autoheader.
Hardware requirement: we assume the architecture implements <= 48-bit virtual address space and always uses x86-64 canonical addresses. That's the case with any Intel/AMD processor and Linux kernel combination other than the recent Linux kernels that are configured to allocate memory beyond 47-bit user address space on the processors that support 5-level paging (e.g., Intel Ice Lake) as of right now (2021).

2 Test environment

Software environment:

- Ubuntu 18.04.6 LTS
- GNU make 4.1
- GCC/G++ 8.3.0
- liblapack-dev 3.7.1
- liblapacke-dev 3.7.1
- libblas-dev 3.7.1
- libatlas-base-dev 3.10.3
- libfftw3-dev 3.3.7
- python3 3.6.9
- sklearn 0.22.2
- scipy 1.4.1
- numpy 1.19.0

Note that these are the ones installed at the time we performed the experiments but it should be ok if you use newer versions.

Hardware environment:

Node type	1	2
CPU	Intel Core i7-3820	Intel Xeon E5-1650 v3
CPU Frequency	3.6 GHz	3.50 GHz
L1 Cache	32KB + 32 KB	32KB + 32 KB
L2 Cache	256 KB	256 KB
L3 Cache	10 MB (shared)	15 MB (shared)
Memory	64GB (DDR3-1600 x 8)	128GB (DDR4-2133 x 4)
Secondary storage	WD HDD 7200RPM 2TB	Seagate HDD 7200RPM 1TB
Network	not used	not used

The experiments are single-threaded and we only launch one experiement at a time on a single node so the number of cores or hyperthreading is irrelevant to our purpose. Cache size is also irrelevant for our purpose as our data structure size far exceeds even the L3 cache size (listed below just for reference). On the other hand, we have 10 type-1 nodes and 6 type-2 nodes and we may launch any of the experiments on any one of them depending on the availability. We made sure that there were no computation or I/O heavy programs running concurrently.

3 Preparing datasets

We have two scripts for creating the datasets: 1) the datasets based on the FIFA World Cup 98 website access logs for the ATTP/BITP heavy hitter experiments; 2) the synthetic datasets for the ATTP frequent direction experiments.
For the world-cup datasets (for the ATTP/BITP heavy hitters), run:

```
$ ./data_proc/world-cup/prepare_data.sh
```

It takes about xxx to generate all the datasets. In case the website hosting the raw logs is unreachable, please contact Zhuoyue (zzhao35@buffalo.edu) for our own copy.
For the matrix datasets (for the ATTP frequent directions): run:

```
$ ./data_proc/gen_mat_data.sh
```

It takes less than 12 minutes to generate all the three datasets (small, medium and large). If you only want one or some of the three datasets, specify its name as a command line argument to the script (e.g., `./data_proc/gen_mat_data.sh small`)
All the generated data are put into the data/ directory, see data/README.md for descriptions.

4 Building the code

If you're using Ubuntu, you should be able to use the following to prepare all the required prerequisites:

```
$ sudo apt install gcc g++ make liblapacke-dev libatlas-base-dev \
> libfftw3-dev python3 python3-pip
$ pip3 install sklearn scipy numpy
```

To build the code:

```
$ ./configure
$ make
```

5 Running the experiments

6 Plotting the figures

You'll need jupyter notebook, matplotlib, numpy and pandas to plot the figures. The scripts provided in plot/ work on my local WSL 2.0 installation with Ubuntu 20.04 LTS, python 3.8.10, matplotlib 3.4.3, numpy 1.21.3, pandas 1.3.4 and jupter notebook 6.0.3.

```
$ sudo apt install python3 python3-pip jupyter-notebook
$ pip3 install matplotlib numpy pandas
```

The following table lists which notebook you should run to generate specific figures as well as the expected input file. These scripts also generates pdf files for the figures in the plot/ directory.

Notebook	Figures	Input
`HH_ATTP_clientid.ipynb`	2, 3(left), 4	`filtered_logs/client_id_attp_filtered_combined.txt`
`HH_ATTP_objectid.ipynb`	3(right), 5, 6	`filtered_logs/object_id_attp_filtered_combined.txt`
`HH_BITP_clientid.ipynb`	7, 8(left), 9	`filtered_logs/client_id_bitp_filtered_combined.txt`
`HH_BITP_objectid.ipynb`	8(right), 10, 11	`filtered_logs/object_id_bitp_new_filtered_combined.txt`
`MAT_ATTP_small.ipynb`	12(a), 13(left), 14	`filtered_logs/ms_small_attp_filtered_combined.txt`
`MAT_ATTP_medium.ipynb`	12(b), 13(right), 15	`filtered_logs/ms_medium_attp_filtered_combined.txt`
`MAT_ATTP_big.ipynb`	12(c), 16	`filtered_logs/ms_big_attp_filtered_combined.txt`
`scalability_test_client_id_ATTP.ipynb`	1	`scalability_logs/scalability-test-client-id.log`