

Implementing LRU Policy for Cache Replacement in Edge Cloud

1 Introduction

In this document, the Least Recently Used(LRU) policy is used to implement service replacement in Edge Cloud Cache. The evaluations use real log files from Google[1] to evaluate the performance. The algorithm used is described below in the [implementation](#) section. There are two input files to the algorithm, one of which contains the [dataset](#) and the other contains the [drivers](#) to the algorithm.

2 Description of Data-set used

The log files from Google[1] are processed and converted to a CSV file.. The file contains the following information in the given format.

Notation	Data Type	Description
<i>JobID</i>	String	Contains the job id of the service
<i>Timestamp</i>	long	Contains the timestamp of service request
<i>TaskIndex</i>	long	Contains the task index of the service
<i>CPU</i>	double	Contains the requested amount of CPU by the service
<i>RAM</i>	double	Contains the requested amount of RAM by the service
<i>RuntimeDisk</i>	double	Contains the requested amount of runtime disk by the service

Files used for the experiment: [OutputModify\(0-0\).csv](#)¹

3 Description of Program Drivers

The program driver file² contains the following information.

Notation	Data Type	Description
<i>Services_{max}</i>	int	Contains the maximum number of service request to be processed in the experiment
<i>CPU_{max}</i>	double	Contains the maximum amount of CPU available for the experiment
<i>RAM_{max}</i>	double	Contains the maximum amount of RAM available for the experiment
<i>Runtime_disk_{max}</i>	double	Contains the maximum amount of Runtime disk available for the experiment
<i>Storage_{max}</i>	double	Contains the maximum amount of Storage available for the experiment

Files used for the experiment: [Driver.csv](#)³

¹Disk space recorded in this trace represents runtime local disk capacity usage. Disk usage required for binaries and other read-only, pre-staged runtime files is *not* included.

²The entries in this file are scaled.

³*Storage_{max}* is not considered yet in the algorithm.

To perform experiment with different maximum parameters, change the content of the program driver file.

4 Implementation of LRU Algorithm

The above input files are fed to the Java [program](#)⁴. This program implements the Edge Cloud Cache with LRU algorithm as described in the Cache Replacement Policies [document](#).

The program takes two input. One for the dataset, and the other for the program drivers. It initializes the *cache* class to the values given in the drivers file. Thereafter, the program initializes an instance of *edge_cloud* and runs the *EdgeServer* using LRU as the cache replacement policy. Change the path of output file in the Java program for different experiments.

4.1 Performance Metrics

The performance metric used in this document is **Hit Rate**. It is defined as number of requests found in the *Running* cache over the number of total services requested in the *EdgeServer*. The higher hit rates signifies better performance, as more number of services requested are found in the cache.

4.2 Evaluation Results of LRU replacement policy

The Java program produces the following information to a CSV file.

Notation	Data Type	Description
<i>DriverPath</i>	String	Path of the program driver file used in the experiment
<i>NoServices</i>	long	Number of services requested to the Edge Cloud in the experiment
<i>NoHit</i>	long	Number of cache hits in the Edge Cloud in the experiment
<i>HitRate</i>	double	Hit Rate of the Edge Cloud Cache in the experiment

Files generated in the experiment: [Results.csv](#).

This file gets updated for various runs of the experiment using different program driver files.

4.3 Generating Plots

The plot generating [program](#) uses [Matplotlib](#) to create plots. The program uses the CSV file generated in section 4.2 and the Program Driver files to generate [Hit Rate\(%\) vs Cache resources](#)⁵ graph and [Number of cache hit\(K\) vs Cache resources](#) graph.

To generate plots for different experiments, change the input CSV file of the program.

References

- [1] Charles Reiss, John Wilkes, and Joseph L. Hellerstein(2011) *Google Cluster-Usage Traces: Format + Schema*, Technical Report. Google Inc. (Posted at <https://github.com/google/cluster-data>.)
- [2] Chih-Kai Huang, and Shan-Hsiang Shen(2021) *Enabling Service Cache in Edge Clouds*, ACM Trans. Internet Things 2, 3, Article 18 (July 2021)(Posted at <https://doi.org/10.1145/3456564>)

⁴This program uses Google [GSON](#) for deep-copying hashmaps.

⁵The resources i.e. CPU, RAM and Disk are scaled.