



COLLABORATIVE DATA CACHING

¹A V Murali Krishna, ²Sumaya, ³Sai Sreeya Ramasahayam,

⁴Syed Saifullah Hussaini

¹Assistant Professor in Department of CSE Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India.

^{2,3,4}UG Scholar in Department of CSE Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India.

Abstract - Over the past ten years, the world's use of mobile equipment, such as mobile phones, wearable gadgets, tablets, smart cars, and Connectivity of Things (IoT) devices, has increased at a quicker rate. Congestion of networks is frequently caused by the high volume of online traffic. increases the latency of a network. We employed edge computing (EC) in our project to overcome this problem. In order to deploy computing resources from centralized clouds to dispersed edge servers, edge computing has evolved. It's a web-based application for our project. By storage frequently requested data on edge devices that are closer to end users, coordinated data caching with edge computing attempts to enhance the loading time of online applications. In addition to a primary cloud server that houses all of the data, edge computing servers are also installed. Edge servers save data for up to a day before instantly erasing it and reducing memory usage. The recipient, a mobile destination, receives it from an edge server or online server, which is but the only way to retrieve those information files is with a key. The highly encrypted key is produced to the target user's email and is used to safeguard and offer privacy for the data. Undoubtedly, several crucial security measures were employed in the collective cached data project to safeguard important information from unauthorized entry or surveillance.

1. INTRODUCTION

1.1 Introduction

Mobile gadgets have rapidly increased in popularity around the world. Network congestion and delay are frequently brought on by the massive amount of network traffic. Edge computing (EC), a new computer paradigm, has evolved to solve this problem. transfer processing power from a centralized cloud to a network of edge computers. Each edge server is physically powered by a few devices and connected to a base station or access point that is near to the mobile devices of app users. By using the CPU and storage capabilities of edge servers, phone and IoT application suppliers may host their apps there, ensuring low latency and excellent performance for their app consumers. To decrease computation, compute duties might be transferred from mobile devices to adjacent edge servers. expenses and energy use for certain mobile gadgets. More mobile data will be transferred over edge servers between the cloud and app user's devices as a fast-growing number of app users start using edge applications. mobile technology. From the standpoint of an app vendor, caching certain data, especially well-known ones like trending posts from Twitter and Facebook and viral videos, can considerably minimize network latency in the app user's retrieval of app data. If the data are already cached on the edge servers, app users can retrieve the data there rather than from distant cloud servers. Additionally, by significantly

reducing the quantity of data sent between the cloud and mobile devices using server-based caching, app vendors may minimize their costs.

2. Literature Survey

2.1 Optimal edge user allocation in edge computing with variable sized vector bin packing.

Y. Yang, J. Grundy, J. Hosking, F. Chen, M. Abdelrazek, P. Lai, Q. He, and J.

Globally, the variety of devices at the end connecting to mobile and cloud networks has increased recently. In this mobility study from Ericsson, it is anticipated that ensuring a dependable and low-latency connectivity to end users, which is one of the primary quality-of-service (QoS) criteria, would be a significant problem for providers of online services. The idea of edge computing was created to address this problem. When a number of middle edge servers are placed close to end devices, the resources for computing, storage, and networking are pushed out towards the border of the network.

2.2 Optimal content replication and request matching in large caching systems

A. Mukhopadhyay, N. Hegde, and M. Lelarge are the authors.

Internet traffic has increased dramatically over the past several years, mostly as a result of the transfer of multi-media items like streaming videos and movies. A result of this increase in multi-media traffic are systems that are peer-to-peer and content distribution systems (CDNs). Large CDNs often have a central server that stores the whole catalogue of materials and a large number of edge servers that each cache a small portion of the contents and respond to requests for the items they have cached. It is anticipated in such systems that access to the main server is costly. As a result, the edge servers, which are restricted by their memory, must fulfil a considerable share of the information in requests. and bandwidth limitations. They represent these servers as loss servers in this study with the goal of reducing the number of requests that are stopped at these servers. The throughput in such systems critically depends on how material is duplicated among servers and how requests for certain contents are matched to servers hosting those contents. The throughput of the cached system in the stable regime can be optimized by combining the issue of computing the optimum replication policy with the problem of compute the optimal matched policy, as shown in this work. It is demonstrated that finding the best replication strategy for a certain system is an NP-hard

task. They then provide a straightforward randomized matching system that does away with the issue of service interruptions. The caching technique's dynamics is examined using a suggested replication and matched method combined. We investigate a restricting regime in which the quantity of servers and the speeds at which the data arrive are scaled proportionately. that asymptotic optimality is attained by the suggested policies. To assess the effectiveness of various strategies and investigate the behavior of the caching system under various service time dispersion of the requests, extensive simulation results are given.

3. OVERVIEW OF THE SYSTEM

3.1 Existing System

- The cloud-based computing method of data caching uses an organized cloud architecture. where the online cache is a technique for storing data that is often requested in the cloud. It is in charge of selecting the data that will be kept in the cloud. cache, as well as how prolonged. Basically, how to effectively use cache space on cloud hosts with mobile devices is a major issue with cached data in the cloud technology ecosystem. Additionally, putting data near to end users causes responses to and requests for access to that information to take a lengthy time.

3.2 Proposed System

In our suggested system, edge computing, or edge cache, has been employed. Edge Cache stores and retrieves data using edge servers, also known as edge servers, that are placed in various places across the globe. The cloud server where the processing is initiated serves as the primary data store. These servers are scattered worldwide and feature high-speed network and storage connections. By bringing data closer to end users, they improve efficiency and safety by lowering latency. In order for the edge server to save and retrieve it, we limited storage to one day. The data is deleted from the memory later after a day, but it is still saved on the cloud.

3.3 Proposed System Design

In this project work, I used four modules and each module has own functions, such as:

1. Mobile Source
2. Edge Server
3. Cloud Server
4. Mobile Destination

3.3.1 Mobile Source

Firstly, mobile user can register in the web application and login with their valid username and password. The user can upload files (text document, images, etc.) and can secure data by encrypting the data. The user can send data to cloud through edge server. Mobile user can view uploaded data and also encrypt data and logout from application.

3.3.2 Edge Server

The edger server module can view the list of files uploaded by user and send to cloud. Basically, the files which user uploaded are stored only for one day. When user requests data on same day edge server will handle requests and send response to mobile destination. After one day, cached data is deleted from the mobile and only cloud can view those files.

3.3.3 Cloud Server

The cloud server module will login and view the files uploaded by every mobile user and can also view list of files. The cloud sever stores the uploaded data and send to the receiver address i.e. to mobile destination.

3.3.4 Mobile Destination

The mobile destination will register in the application and login with valid username and password. MD can view upload files and send request to edge server to get today's data uploaded by mobile user. After sending a request, key is sent to destination user and using this key MD can decrypt the encrypted data and download the file sent by the user.

4 Architecture

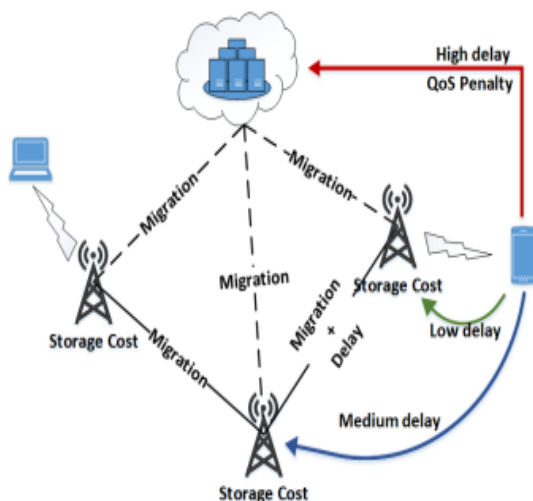
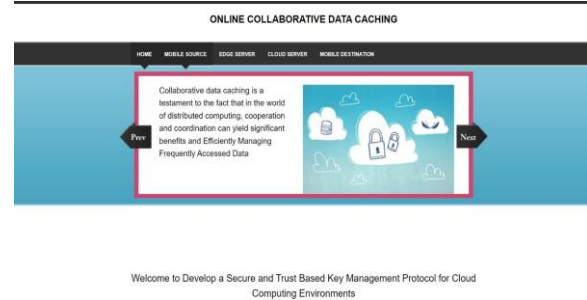


Fig 1: Frame work of Collaborative Cache

Home Page:



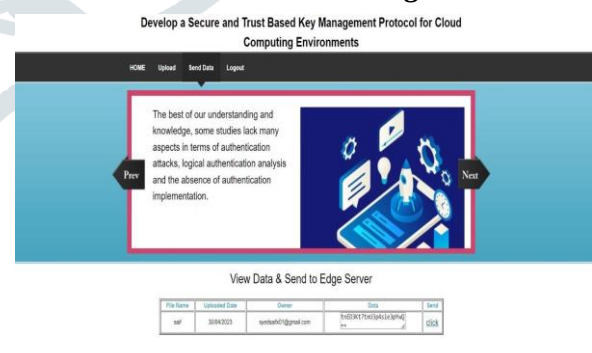
Mobile Source Home Page:



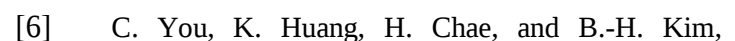
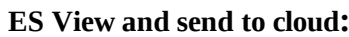
Mobile source Registration Page:



Mobile Source Send data to Edge Servers:



Edge server home page:



“Energy-efficient resource allocation for mobile-edge computation offloading,” IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1397– 1411, 2017.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2795–2808, 2016.

[8] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, and Y. Yang, “A game- Theoretical approach for user allocation in edge computing environment,” IEEE Transactions on Parallel and Distributed Systems, 2019.

