

**Research Progress Report**

Record/Log of Contact Sessions between the Student and the Guide

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Register No.: 1847246** | | **Name of Student: Nirmal Benny** | | | |
| **Class: IV MCA** | | **Paper Code: MCA481 Research Data Collection** | | | |
| **Name of the Research Guide: Dr. Shoney Sebastian** | | | | | |
| **Date** | **Contact Mode (Email / in person)** | | **Duration (Hours)** | | **List of Interaction** |
| 08/11/2019 | In person | | 2 Hrs | | Discussed about different steps in the implementation of the identified problem statement and objective |
| 12/11/2019 | In person | | 2 Hrs | | Discussed with Literature survey findings |
| 15/11/2019 | In person | | 2 Hrs | | Discussed on research gap literature survey |
| 19/11/2019 | In person | | 2 Hrs | | Discussed on existing methodology |
| 26/11/2019 | In person | | 2 Hrs | | Discussed on existing methodology |
| 29/11/2019 | In person | | 2 Hrs | | Discussed on existing methodology |
| 03/12/2019 | In person | | 2 Hrs | | Discussed on methodology |
| 06/12/2019 | In person | | 2 Hrs | | Discussed on methodology |
| 10/12/2019 | In person | | 2 Hrs | | Discussed on methodology |
| 13/12/2019 | In person | | 2 Hrs | | Feedback on survey paper |
| 16/12/2019 | In person | | 2 Hrs | | Feedback on survey paper |
| 07/01/2020 | In person | | 2 Hrs | | Feedback on survey paper |
| 14/01/2020 | In person | | 2 Hrs | | Feedback on survey paper |
| 17/01/2020 | In person | | 2 Hrs | | Discussed on cache policies |
| 21/01/2020 | In person | | 2 Hrs | | Discussed on cache policies |
| 24/01/2020 | In person | | 2 Hrs | | Discussed on cache policies |
| 28/01/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 31/01/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 07/02/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 11/02/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 17/02/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 24/02/2020 | In person | | 2 Hrs | | Discussed on possible addition to cache policies |
| 03/03/2020 | In person | | 2 Hrs | | Preparing the research progress report |
| 06/03/2020 | In person | | 2 Hrs | | Preparing the research progress presentation |
| 10/03/2020 | In person | | 2 Hrs | | Verifying the research progress and document |
| **Student Comments:**  Will continue to work on to reduce the image retrieval time from using effective cache policies | | | | | |
| **Research Guide Comments:** | | | | | |
| **Signature of Student:** | | | | **Signature of Guide:** | |



MCA481 - Research Design / Data Collection

Progress Report Submission

Submitted by

Nirmal Benny

1847243

Under the Guidance of

Dr. Shoney Sebastian

March 2020

Department of Computer Science

CHRIST (Deemed to be University)

Hosur road, Bengaluru 560 029

**Topic – Effective cache policy and prefetching to reduce the image retrieval latency on cloud**

**Reg. Number : 1847243**

**Student Name : NirmalBenny**

**Guide Name : Dr. Shoney Sebatian**

**1. Introduction**

Delivering static content at a web scale is a challenging task. It is of utmost important to deliver static contents as fast as possible because, slow website results in poor customer experience and can directly affects the companies revenue. In order to serve all the request with a minimum possible response time, techniques like Caching, CDN, file compression and encoding are used. In this paper, we are focusing on delivery of images from large scale photo storage systems like Flickr, Facebook. We will review various caching techniques that exists and propose a better approach for caching and serving the images from the storage stacks.

collection.

The objective in our static caching problem is to select images to be stored in a ﬁxed-capacity regional cache such that the average response time for future image retrieval requests is minimized. This particular caching problem is slightly di↵erent from the static caching problem in similar domains (e.g., page caching in web proxies or result caching in web search engines) because

the same image is often stored in many different resolutions.4 As a result, a cache hit does not necessarily lead to similar response time as in web search, rendering the cache hit rate not an ideal optimization target. In our case, it is possible to serve a retrieval request by generating the requested image after resizing one of its higher resolution versions previously stored in the cache . Hence, a cache hit may be possible even if the requested resolution of the image is not found in the cache. As we will discuss, this difference leads to a slightly more complex cost model when caching images.

**2. Existing scenario**

Cloud Based Medical Image Exchange give on demand Medical imaging services for visualizing and sharing data.

Medical imaging is the procedure used to make pictures of the human body for diagnosing diseases in human. Medicinal pictures are the primary methods for the medicinal services indicative systems. These Medical images originate from a board range of imaging innovations, for example, Computed tomography (CT), Magnetic Resonace imaging (MRI) etc and they produce a lot of picture information and significant therapeutic data.

Medical image sharing is the electronic exchange of medical images between hospitals, physicians and patients. Rather than using traditional media, such as a CD or DVD, and either shipping it out or having patients carry it with them, technology now allows for the sharing of these images using the cloud. The primary format for images is DICOM (Digital Imaging and Communications in Medicine). Typically, non-image data such as reports may be attached in standard formats like PDF (Portable Document Format) during the sending process.Cloud encomposes PACS within it.

The open-source distributed computing programming Hadoop is for scalable and reliability. Hadoop include Hadoop Distributed File System(HDFS) which gives high throughput get a data from cloud. MapReducer is methodology used in hadoop system which count the data using Mapper function and reduce those counted output of mapper using Reducer method[8].This paper Uses Hadoop tool to increase the retrieval speed of medical images.

**3. Applications**

Medical Images are most commonly used in healthcare centers inorder to predict and diagnose disease earlier.Medical Images in cloud helps the radiologist to easily view and share the image confidentially within their circle despite of any location.It also reduce the storage cost

**4. Problem Statement**

“To develop an algorithm to reduce the retrieval time of medical images in cloud ”

**5. Research Gap**

There are many studies concentrating in the area of retrieval of Medical Images in cloud. But the response time and accesing of Medical images are still considered as major problem.

**6. Literature Review**

In a research done by Qi Huang et al., a comprehensive study of the working of Facebook's photo cache system has been done. Facebook image management layer includes browser caches on end-user systems, Edge Caches at ~20 PoPs, an Origin Cache, and for some kinds of images, additional caching via Akamai. Image urls sent to the clients are dynamically generated by the servers to control the traffic distribution. The details encoded in the image url’s are unique photo identifier, display dimensions of the image, encode the fetch path, which specifies where a request that misses at each layer of cache should be directed next. Facebook uses 3 layers of caching arranged based on the distance from the user. Browser cache is co-located with the client, uses an in-memory hash table to test for existence in the cache, stores objects on disk, and uses the LRU eviction algorithm, If a request misses at the browser cache, the browser sends an HTTP request out to the Internet, The fetch path dictates whether that request is sent to the Akamai CDN or the Facebook Edge. Edge Level Caching To make up for the lack of speed and connectivity with cloud, processing for mission-critical applications will need to occur closer to the data source. Edge computing is a networking philosophy focused on bringing computing as close to the source of data as possible in order to reduce latency and bandwidth use. In simpler terms, edge computing means running fewer processes in the cloud and moving those processes to local places, such as on a user’s computer, an IoT device, or an edge server. Bringing computation to the network’s edge minimizes the amount of long-distance communication that has to happen between a client and server. The particular Edge Cache that a request encounters is determined by its fetch path. Each Edge Cache has an in-memory hash table that holds metadata about stored photos and large amounts of flash memory that store the actual photos . If a request hits, it is retrieved from the flash and returned to the client browser. If it misses, the photo is fetched from Facebook’s Origin Cache (step 3) and inserted into this Edge Cache. The Edge caches currently all use a FIFO cache replacement policy. Requests are routed from Edge Caches to servers in the Origin Cache using a hash mapping based on the unique id of the photo being accessed. Like the Edge Caches, each Origin Cache server has an in-memory hash table that holds metadata about stored photos and a large flash memory that Haystack, The backend, or Haystack, layer is accessed when there is a miss in the Origin cache. Because Origin servers are co-located with storage servers, the image can often be retrieved from a local Haystack server (step 4). If the local copy is held by an overloaded storage server or is unavailable due to system failures, maintenance, or some other issue, the Origin will instead fetch the information from a local replica if one is available. Should there be no locally available replica, the Origin redirects the request to a remote data centre. Haystack resides at the lowest level of the photo serving stack and uses a compact blob representation, storing images within larger segments that are kept on log structured volumes. The architecture is optimized to minimize I/O: the system keeps photo volume ids and offsets in memory, performing a single seek and a single disk read to retrieve desired data .

Xiao Bai et al.,proposed two gain-based caching policies that construct a static, ﬁxed-capacity cache to reduce the average serving time of images. The basic idea in the proposed policies is to identify the best resolution(s) of images to be cached so that the average serving time for future image . Real-life traffic log from Flickr has been used to compare various cache policies. In this paper they have defined what is a static caching problem in the context of photo caching. They have suggested frequency based heuristics and gain based haeuristics. Frequency based heuristics include Largest requested, Most requested, Most Requested per Byte. Gain based heuristics. In this line of heuristics, instead of mainly relying on the observed request frequency of images, the greedy choice property prefers to select images that can ensure low response time. Largest gain per byte (LGPB): When making caching decisions, this heuristic prefers images that are expected to bring the largest reduction in response time over all image requests. Largest gain per byte with frequency adjustment (LGPB-FA). This heuristic is a variant of . Its key difference with LGPB, as well as the frequency-based heuristics, is in the way it estimates the future frequency of each request. The motivation stems from the observation that the past frequency of requests is not always strongly correlated with their future frequency, especially in the case of infrequent requests .Reda Bellafqira, Gouenou Coatrieux et.al.,[12] proposed Content Based Image Retrieval Sytem for homomorphic encryption domain which help the radiologist to retrieve all the similar image using wavelet based image features and Content Based Image Retrieval(CBIR) is used for retrieval of encrypted medical images.

**7. Objectives of Research**

* 1. **Primary Objective**
     1. To come up with an effective caching algorithm to reduce the latency of image retrieval from cloud.
  2. **Secondary Objectives**
     1. Smoother user experience on cloud

**8. Dataset Description**

Dataset is being downloaded from NIH (National Institutes of Health)

The National Institutes of Health (NIH) is the primary agency of the [United States government](https://en.wikipedia.org/wiki/United_States_government) responsible for [biomedical](https://en.wikipedia.org/wiki/Biomedical) and [public health](https://en.wikipedia.org/wiki/Public_health) research. It was founded in the late 1870s, and is now part of the [United States Department of Health and Human Services](https://en.wikipedia.org/wiki/United_States_Department_of_Health_and_Human_Services). The NIH conducts its own scientific research through its [Intramural Research Program](https://en.wikipedia.org/wiki/NIH_Intramural_Research_Program) (IRP) and provides major biomedical research funding to non-NIH research facilities through its Extramural Research Program [15].

**8. References**

[1] C. Teng et al., "A medical image archive solution in the cloud," 2010 IEEE International Conference on Software Engineering and Service Sciences, Beijing, 2010, pp. 431-434.doi: 10.1109/ICSESS.2010.5552343

[2]C. Yang, L. Chen, W. Chou and K. Wang, "Implementation of a Medical Image File Accessing System on Cloud Computing," 2010 13th IEEE International Conference on Computational Science and Engineering, Hong Kong, 2010, pp. 321-326.

[3].C. Yang, K. Huang, W. C. Chu, K. Lai, C. Chang and C. Lu, "Implementation of Video and Medical Image Services in Cloud," 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops, Japan, 2013, pp. 451-456.

[4]Alonso-Calvo R., Crespo J., Maojo V., Muñoz A., García-Remesal M., Pérez-Rey D. (2011) Cloud Computing Service for Managing Large Medical Image Data-Sets Using Balanced Collaborative Agents. In: Demazeau Y., Pěchoucěk M., Corchado J.M., Pérez J.B. (eds) Advances on Practical Applications of Agents and Multiagent Systems. Advances in Intelligent and Soft Computing, vol 88. Springer, Berlin, Heidelberg

[5]Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun and K. Ren, "A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 11, pp. 2594-2608, Nov. 2016.

[6]Venner, J., 2009. Pro Hadoop. 1st Edn., Apress, ISBN:13: 9781430219439, pp: 440.

[7].F. M. Hani, I. V. Paputungan, M. F. Hassan, V. S. Asirvadam and M. Daharus, "Development of private cloud storage for medical image research data," 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2014, pp. 1-6.doi: 10.1109/ICCOINS.2014.6868433

[8]M.VengadapurvajaG.NishaR.AarthyN.Sasikaladevi “An Efficient Homomorphic Medical Image Encryption Algorithm For Cloud Storage Security” https://doi.org/10.1016/j.procs.2017.09.150.

[9]G. Kanagaraj and A. C. Sumathi, "Proposal of an open-source Cloud computing system for exchanging medical images of a Hospital Information System," 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011), Chennai, 2011, pp. 144-149.

[10]Ravindra Babu Bellam, M. P. Coyle, P. Krishnan and E. G. Rajan, "Issues while migrating medical imaging services on cloud based infrastructure," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 109-114.

[11]D. Patil, S. Krishnan and S. Gharge, "Medical Image Retrieval by Region Based Shape Feature For CT Images," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 155-159.

[12]Bellafqira, R., Coatrieux, G., Bouslimi, D., & Quellec, G. (2015). Content-based image retrieval in homomorphic encryption domain. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). doi:10.1109/embc.2015.7319009.

[13]Elkariem, A. F. A., Bashir, M. B., Ahmed, T. H., & Yousif, A. (2017). Distributed medical image retrieval techniques: A review. 2017 Sudan Conference on Computer Science and Information Technology (SCCSIT). doi:10.1109/sccsit.2017.8293058 .

[14]Aminzou, S., Er-Raha, B., Khamlichi, Y. I., Afdel, K., & Machkour, M. (2013). Towards a secure access to patient data in cloud computing environments. 2013 National Security Days (JNS3). doi:10.1109/jns3.2013.6595466. [15] National Institutes of health “<https://www.nih.gov/about-nih>”.

[15] National Institutes of health “https://www.nih.gov/about-nih”.