A Semantic Search Method for Large Scale Storage Systems in Cloud

Sushma C[1] , Satish B Basapur[2]

[1]PG Scholar, Department of ISE, Dr.AIT, Bangalore, Karnataka, India
[2]Assistant Professor, Department of ISE, Dr.AIT, Bangalore, Karnataka, India

***Abstract*—Processing of the large amount of data their storage and retrieval in the cloud had became a major challenge in the cloud computing environment. Hence an efficient methods need to be Used for depositing the large amounts of the data in the cloud and retrieving it back from the cloud storage systems. In this paper we explore a semantic Search method for processing the large scale data volumes in the cloud. We use the hashing algorithms and flat structured addressing schemes for the retrieval of the data by using the semantic queries. Here the data is processed by using the caching techniques and the data is retrieved by using the semantic query. This techniques reduces the time delay for the retrieval of the data from the large scale storage systems.**

1) INTRODUCTION

In the recent emerging technologies organizations and individual customers stores large amount of the data in the cloud. Processing this large amount of data and retrieval has become a major challenge in the cloud computing environment. Example in the year 2011 7% consumers had stored their volumes of data in the cloud and this approximate result had grown to 36% in 2016, according to the Gartner Inc [1] and Storage News letter [2] existing approaches mainly relay on the hierarchical Clustering technique which requires a very large amount of processing time. Since in the existing approaches processing operations are done either on the source or destination pairs which may create the bottleneck in the source and destination systems, and also the present approaches to unregulated data search and research relays on the system based lumps of files and the features related to the multimedia based images[3]. In the view to increase the performance of processing the large amount of data the following problems which is related to the data research need to be addressed.

Increased access latency: The accessing of the data may take a large amount of time due to the increased number of requests which may create the bottleneck in the cloud servers the response to the requests may take time since the present approaches to unordered search of the data and analysis mainly relays on the system based lumps of data files and the features related to the multimedia based images [3]. If we use the method which relays on the exact content it may produce the increased amounts of auxiliary data which may increase the bottleneck of the system.

High Energy Consumption: Due to the bottleneck created in the cloud servers .The response to the requests may delay due to the delay in the response time energy consumption will be high Hence the response time need to be reduced to reduce the energy consumption. The bugs in the data need to be corrected to reduce the energy consumption which may also reduce the need of virtual servers.

Data Authentication: The Cloud servers need to provide authorization to the users so that only the authorized and requested users can access the data traffic may be created in the cloud which may reduce the processing speed.

High query costs: In order to access the data in the cloud, processing of the queries are in the high demand. The research based on the data in the cloud may consume abundant systems resources such as memory space, I/O bandwidth, High performance multicore processors [4]. The main culprit for the increased amount of resource costs is the bottleneck caused by the high performance query operations.

In order to overcome the above problems the following methods can be used such as Flat Structured addressing [5] Algorithms such as the locality sensitive algorithms [6] cuckoo based hashing algorithms can be used. In order to aggregate the semantically correlated files SANE [7] approach can be used to aggregate the correlated files into flat and feasible groups to achieve increased processing of the semantic queries.

2) RELATED WORK

The real time and cost efficient scheme which is known as the SmartEye is used in the cloud supported disaster Environments. The main idea of the SmartEye is that it incorporates the increased quality of service in the network deduplication scheme for the networks which is software specified. The idea of the SmartEye is that it aggregates the network flows which contains the identical features by using the semantic hashing and provides the well known communication services for all the flows which is aggregated, here the SmartEye is not related to a single flow it mainly relays on the aggregated flows. To achieve this SmartEye uses the following optimization techniques called as the semantic based hashing and the space efficient filters. Efficient sharing of the image is useful to detect the disaster and recognition of the scene[8].

The increased use of the smart phones which is equipped with the camera and tablets had led the users to capture large amount of videos and photos. In 2011 the worldwide consumer digital storage needs had grown to 329 exabytes and in the year 2016 it had grown to 4.1 zettabytes [9]. To address the large amount of features which is extracted from the images a locality sensitive hashing algorithms is used to place

the local descriptors in the index. This approach provides the view to approximate the similarity in the queries. Which allows to examine only small fraction in the database. Even though locality sensitive hashing scheme has a better performance which is related to theoretical view, a practical implementation is very slow [10]. Local binary pattern is used for the face image recognition here the face image is divided in to several parts by applying the local binary pattern feature the divided parts are extracted and concatenated to use as face descriptor. This method is implemented to recognize the face under various challenges [11].

Further SmartStore is used which incorporates the metadata semantics of files to aggregate correlated files into semantic based groups and retrieval tools to retrieve the data. To improve the system scalability and to reduce the query latency the decentralized design techniques can be used for the complex queries which is the better technique for building the semantic related caching. Smart store limits the complexity for searching the queries for the single or the semantically aggregated groups and it limits the use of incorporating the brute force search in the system [12]. Other techniques can be used by extracting the distinct features from the images and aggregating it into a single feature. These extracted features can be matched with high probability for large database features from many images [13]. Due to the increased growth and complexity, data volumes had led the very high demand for efficient searching of the data in the cloud.

A present storage system in the cloud doesn't provide a well capability for the data analytics related to the real time. Because the correct value and the worth of the data depends heavily on how the data analytics should be carried in the real time. Since the large fractions of the data terminates with their data being lost and drastically reduced because of the data staleness.

To address the above problem a cost efficient method called as the FAST is implemented for searchable analytics of the data. Hence the basic idea of the FAST is to examine and analyze the semantic correlation among the datasets by using the correlation based hashing and feasible flat structured addressing to extremely reduce the processing Latency while incorporating small loss in the data search and correctness. The idea of the FAST is to rapidly identify the correlated files and reducing the wideness of the data to be processed [14]. In the existing approach another technique used is to hash the points from the database by confirming that the probability of the collisions is lesser for the objects that are places at a large distance other than the objects that are placed closed to each other.

This method has experimental evidence which provides an efficient improvement in the run time compared to other methods for searchable high dimensional spaces by using hierarchical tree decomposition [15]. A locality sensitive hashing scheme is used for approximating the nearest neighbor problem which is under the lp norm, based on the pstable distributions this scheme improves the running time of the algorithm, this algorithm finds the correct nearest neighbor in O(log n) time for satisfying the certain "bounded growth" condition[16].

3) PROPOSED SYSTEM

The proposed system uses a semantic search method for large scale storage systems in cloud for extracting the required file from the cloud server. The proposed system uses a correlation aware hashing, flat structured addressing and further uses cache based storage of the files it uses the bloom filter for searching files and reduces the time for searching the required file from the cloud server. The main objective of the proposed method is to reduce the time delay in searching the required file, It uses the semantic search method foe retrieving the files from the cloud server by using the keywords to search the file

Advantages of the proposed system

The proposed system is energy efficient, it consumes less energy, cost effective**,** The construction time required to build the system is less**,** Scalable for large storage systems in the cloud**,** The system proposes a query accuracy, Accurate results are retrieved through the semantic query, Performance is high

4) SYSTEM MODEL

In the view to increase the accessing capability of the data in the cloud storage systems the following techniques are used such as the hashing algorithms are used in this paper. The following Fig. 1 shows how the data is placed in the specific manner.

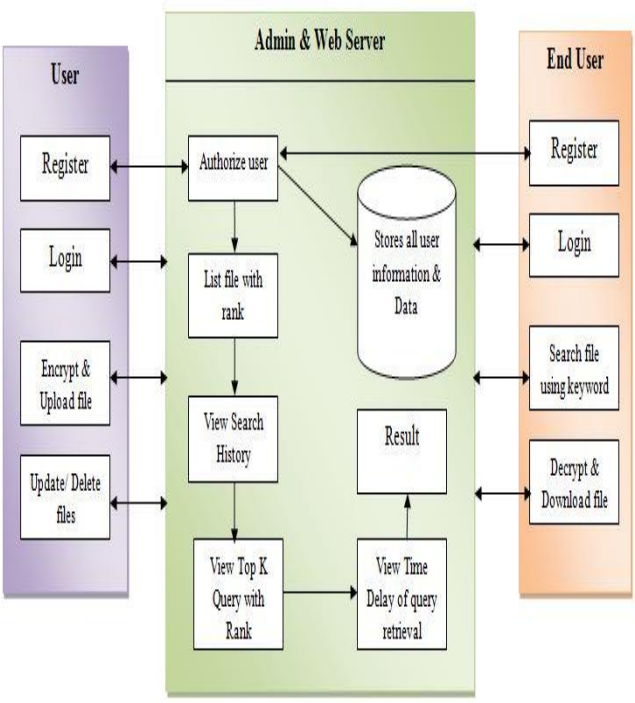

Fig. 1 System Model for Data Storage

In this approach initially the user registers to the cloud server after the registration the user login's to the cloud server later the user can upload the files in to the cloud server by encrypting the data here the user can add n number of files and can update or delete the files which is been added to the cloud server. By retrieving the files from the cloud server database which is been added by the user the admin lists all the files with rank, views the search history of the previous users and

makes the lists of top k queries in rank, views the results of the time delay and stores the updated information in the database, the end user can retrieve the file by login to the cloud server and can search the file by using the keyword, decrypts the files and the user can download the required file.

5) IMPLEMENTATION

The following Fig. 2 shows the flow chart diagram of how the end user retrieves file by using the semantic search, here the data user registers to the server, if the user is already been registered the data user needs to login to the server or else needs to register to the webServer after successful registration the user uploads the file to the server.

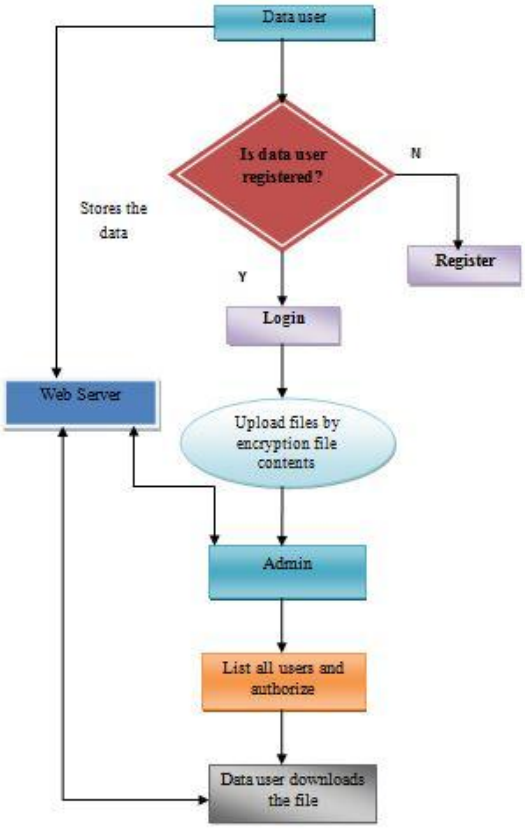

Fig. 2 Flow chart for data retrieval by the user

Here the files which is been uploaded by the user is retrieved by the admin, the admin stores all the files in the cloud server based on the file ranks, the files are placed by using Caching techniques and the locality sensitive hashing algorithms which has the complexity of O(1). Locality sensitive hashing algorithm is used to search and aggregate identical files into the correlation based groups. This provides the retrieval to be narrowed to the one or the limited number of groups by incorporating correlation awareness. Later when the user requests for the specific file, the admin uses bloom filters for the searching of the files. Bloom filters has the features of simplicity and easy to use. In bloom filters the large size vectors of files is hashed effectively to Identify similar files in the real time manner. Bloom filter uses the method based on multiple identical vectors, if two files contain identical vectors it maintains the list of the memberships of the vectors and makes the lists of the similar files. By using this bloom filters the admin searches the file requested by the user, and the user downloads the requested file. If the requested file is not available in the server database, the admin lists the correlated and similar files. Here the user searches the files by using semantic keywords. All the user transactions such as the request for the files, the files which are downloaded by the user, files has been searched by the user, files uploaded and other user information is stored in the server database.

6) TECHNIQUES USED TO IMPROVE THE ACCESSING OF THE DATA

In this paper we can use the locality sensitive hashing algorithm and the cuckoo based hashing algorithms for storing the data in the database in the cloud server and to retrieve the data from the cloud server we use the bloom filters for the users to search the files by using the semantic keywords. Further correlation based hashing and flat structured addressing schemes are used to retrieve the data by the end users.

7) ANALYSIS OF THE RESULTS

First it reduced the time taken for searching of the data and the retrieval from the cloud server, second we use the bloom filter which had simplified the complexity of searching the data because it allows more vectors to be placed in the main memory. Further we had used the flat structured addressing to obtain O(1) for increasing the performance of the query. This approach can be extended to spyglass and the smart store which uses the limited correlation properties.

8) CONCLUSION

In this paper we had explored the various techniques Used to increase the accessing capability in the existing cloud storage systems and how to access the data in the cloud servers, The disadvantages occurred due to the storage of large amount of data. Here we had explored how data need to be processed before it is used in any specific approach. And we had explored various hashing algorithms such as the Locality Sensitive hashing algorithm for hashing purpose and also had explored the bloom filters for filtering purpose to access the data through the use of semantic queries. By using these techniques we can reduce the time delay incurred for searching of the specific file and their retrieval from the large scale storage systems.

REFERENCES

[1] Gartner, Inc., “Forecast: Consumer digital storage needs, 2010-2016,” 2012.

[2] Storage Newsletter, “7% of consumer content in cloud storage in 2011, 36% in 2016,” 2012.

[3]Real-time Semantic Search using Approximate Methodology for Large-scale Storage Systems

Yu Hua, Senior Member, IEEE, Hong Jiang, Fellow, IEEE, Dan Feng, Member, IEEE

[4]D. Zhan, H. Jiang, and S. C. Seth, "CLU: Co-optimizing Locality and Utility in Thread-Aware Capacity Management for Shared Last Level Caches," IEEE Transactions on Computers, vol. 63, no. 7, pp. 1656– 1667, 2014.

[5] R. Pagh and F. Rodler, "Cuckoo hashing," Proc. ESA, pp. 121–133, 2001.

[6]P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," Proc. STOC, pp. 604–613, 1998.

[7]Y. Hua, H. Jiang, Y. Zhu, D. Feng, and L. Xu, "SANE: Semantic-Aware Namespace in Ultra-large-scale File Systems," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 25, no. 5, pp. 1328–1338, 2014.

[8]SmartEye: Real-time and Efficient Cloud Image Sharing for Disaster Environments

[9]Storage Newsletter, "7% of consumer content in cloud storage in 2011, 36% in 2016," 2012.

[10]Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," Proc. ACM Multimedia, 2004.

[11]T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037– 2041, 2006.

[12]Y. Hua, H. Jiang, Y. Zhu, D. Feng, and L. Tian, "SmartStore: A New Metadata Organization Paradigm with Semantic-Awareness for Next-Generation File Systems," Proc. SC, 2009.

[13]D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp.

[14]Y. Hua, H. Jiang, and D. Feng, "FAST: Near Real-time Searchable Data Analytics for the Cloud," Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2014. 91–110, 2004.

[15]A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," VLDB, pp. 518–529, 1999.

[16] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," Proc. Annual Sympo- sium on Computational Geometry, 2004.