A Semantic Image Search Method for Large Scale Storage Systems in Cloud

BYREDDY MOUNICA[1], H. ATEEQ AHMED[2]
[1]PG Scholar, Dept of CSE, Dr. K. V. Subba Reddy Institute of Technology, Kurnool, AP, India.
[2]Assistant Professor, Dept of CSE, Dr. K. V. Subba Reddy Institute of Technology, Kurnool, AP, 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. 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. Existing distributed storage frameworks for the most part neglect to offer a sufficient capacity for the semantic inquiries. To address this problem, we propose a near-real-time and cost-effective searchable data analytics methodology, called FAST. The idea behind FAST is to explore and exploit the semantic correlation within and among datasets via correlation-aware hashing and manageable flat-structured addressing to significantly reduce the processing latency, while incurring acceptably small loss of data-search accuracy. The near-real-time property of FAST enables rapid identification of correlated files and the significant narrowing of the scope of data to be processed. FAST supports several types of data analytics, which can be implemented in existing searchable storage systems. The close constant property of FAST empowers quick recognizable proof of connected records and the huge narrowing of the extent of information to be prepared. Here the data is processed by using the caching techniques and the data is retrieved by using the semantic query. This technique reduces the time delay for the retrieval of the data from the large scale storage systems.

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. A cloud storage environment usually amasses huge volumes of data that critically require fast and accurate data retrieval to support intelligent and adaptive cloud services. 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 images and the features related to the multimedia based images. 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. 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. Semantic Search Systems: Conventional search techniques are developed on the basis of words computation model and enhanced by the link analysis. On one hand, semantic search extends the scope of traditional information retrieval paradigm from mere document retrieval to entity and knowledge retrieval; on the other hand, it improves the conventional IR methods by looking at a different perspective: the meaning of words, which can be formalized and represented in machine processible format using ontology languages such as RDF1 and OWL2. For example, an arbitrary resource or entity can be described as an instance of a class in an ontology; having attribute values and relations with other entities. With the logical representation of resources, a semantic search system is able to retrieve meaningful results by drawing inference on the query and knowledge base. As a simple example, meaning of the query for "people in School of Computer Science" will be interpreted by a semantic search system as individuals (e.g., professors and lecturers) who have relations (e.g., work for or affiliated with) with the school. On the contrary, conventional IR systems interpret the query based on its lexical form. Web pages in which the worlds "people" and"computer science" co-occur, are probably retrieved. The cost is that users have to extract useful information from a number of pages, possibly query the search engine several times. As we will see shortly, other inference mechanisms based on logical rules and inductive approaches have also been evaluated to enable a system to interpret and understand ad-hoc queries.

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

RELATED WORK

The real time and cost efficient scheme which is known as the Smart Eye is used in the cloud supported disaster Environments. The idea of the Smart Eye 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 Smart Eye is not related to a single flow it mainly relays on the aggregated flows. To achieve this Smart Eye uses the following optimization techniques called as the semantic based hashing and the space efficient filters. 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. This approach provides the view to approximate the similarity in the queries. Which allows to examine only small fraction in the database. Real-time data analytics are very important in dealing with large-scale datasets. This is also non-trivial to cloud systems, although they contain high processing capability (hundreds of thousands of cores) and huge storage capacity (PB-level). The fundamental reason is because the analytics must be subject to hard time deadlines that usually cannot be met by brute force with an abundance of resources alone. Existing approaches often fail to meet the (near-) real-time requirements because they need to handle high-dimensional features and rely on high-complexity operations to capture the correlation. 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. A present storage system in the cloud doesn't provide a well capability for the data analytics related to the real time. To address the above problem a cost efficient method called as the FAST is implemented for searchable analytics of the data. 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.

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 the specific manner.

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 images in to the cloud server by encrypting the data here the user can add n number of images and can update or delete the images which is been added to the cloud server. By retrieving the images from the cloud server database which is been added by the user the admin lists all the images 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 images by login to the cloud server and can search the images by using the keyword, decrypts the images and the user can download the required images.

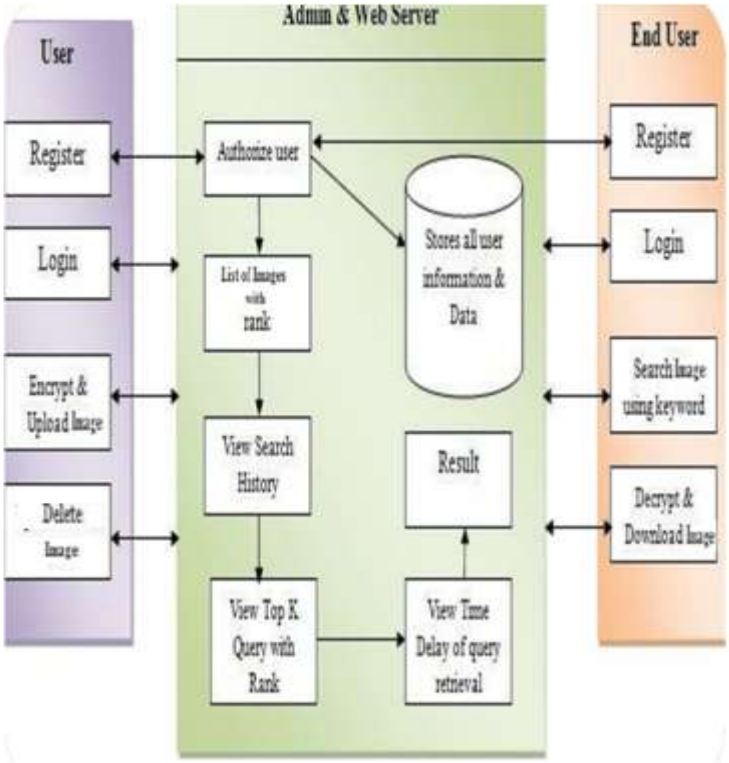

Fig:1 SYSTEM ARCHITECTURE

MODULES:

- System Construction Module
- Semantic-Aware Namespace
- Features of Images
- Flat-Structured Addressing

MODULES DESCSRIPTION:

System Construction Module

- In the first module we develop the System Construction module, to evaluate and implement a near-real-time and cost-effective semantic queries based methodology, called FAST. For this purpose we develop User and Admin entities. In User entity, a user can upload a new images, view all uploaded images and a user can search a images of other users images by using content based image retrieval.
- In the admin entity, the admin privileged access is provided and then admin monitor the user's details and users uploaded images.
- To implement FAST and examine the efficiency and efficacy of the proposed methodology, we leverage "Finding Missing Children" as a use case to elaborate the FAST design and evaluate its performance. A missing child is not only devastating to his/her family but also has negative societal consequences. Although existing surveillance systems are helpful, they often suffer from the extremely slow identification process and the heavy reliance on manual observations from overwhelming volumes of data.

Semantic-Aware Namespace

- By leveraging semantic aggregation, FAST is able to improve entire system scalability. The semantics embedded in file attributes and user access patterns can be used to reveal the potential correlation of file in a large and distributed storage system. These files are thus aggregated into the same or adjacent groups by using the semantic-aware per-file namespace.
- In order to offer smart namespace in FAST, we need to manage the file system namespace in an intelligent and automatic way. In FAST's namespace, we identify semantic correlations and data affinity via lightweight hashing schemes.
- In order to accurately represent the namespace, FAST makes use of multi-dimensional, rather than single-dimensional, attributes to identify semantic correlations. FAST hence obtains the accuracy and simplicity in namespace for large-scale file systems.
- FAST is designed to be compatible with or orthogonal to existing file systems. We hence implement FAST as a middleware between user applications and file systems. For the file system stacks, FAST is transparent, thus being flexibly used in most file systems to significantly improve system performance.

Features of Images

- To perform reliable and accurate matching between different views of an object or scene that characterize similar images, we extract distinctive invariant features from images. Feature-based management can be used to detect and represent similar images to support correlation-aware grouping and similarity search. Potential interest points are identified by scanning the image over location and scale.
- We propose to use a crowd-based aid, i.e., personal images that can be openly accessed, to identify helpful clues. People often take many similar pictures on a famous scenic spot, which actually are the snapshots of those locations in a given period of time. High-resolution cameras offer high image quality and multiple angles. Repeatedly taking pictures can further guarantee the quality of snapshots.
- Given the convenient and easy access to the cloud, these images are often uploaded and shared on the web instantaneously (e.g., by smartphones). We can therefore leverage these publicly accessible images made possible in part by the crowd sourcing activities to help find the images that are correlated with a given missing child.
- For example, if someone takes pictures in the Big Ben, the images possibly contain not only the intended men/women, but also occasionally other people, such as a missing

child in the background. If this image is uploaded and open to the public (openly accessible), we have an opportunity to find the missing child based on the input of his/her image. We can quickly obtain the clues suggesting whether the missing child had ever appeared around the Big Ben. This clue helps us locate the missing child.

- The rationale comes from the observations that instantaneously uploading and widely sharing images are becoming a habit and culture in the cloud.

Flat-Structured Addressing

- The near-real-time property of FAST enables rapid identification of correlated files and the significant narrowing of the scope of data to be processed. FAST supports several types of data analytics, which can be implemented in existing searchable storage system. FAST consists of two main functional modules, i.e., big data processing and semantic correlation analysis. FAST is able to improve entire system scalability. FAST is designed to be compatible with or orthogonal to existing file systems. We hence implement FAST as a middleware between user applications and file systems. For the file system stacks, FAST is transparent, thus being flexibly used in most file systems to significantly improve system performance.
- The namespace serves as a middleware in the file systems by offering an optional semantic-aware function. To be compliant with conventional hierarchical file systems, the user level client contains two interfaces, which can be decided by the application requirements. If working with the conventional systems, the proposed namespace bypasses the semantic middleware and directly links with the application, like existing file systems.
- Users can access file systems via the existing POSIX interfaces. Otherwise, the namespace is used via the enhanced POSIX I/O in the user space. By exploiting the semantic correlations existing in the files' metadata, FAST is able to support efficient semantic grouping and allow users to carry out the read/write operations on files via the enhanced POSIX I/O interfaces.

IMPLEMENTATION:

The following Fig.2 shows the flow chart diagram of how the end user retrieves image 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 web Server after successful registration the user uploads the image to the server.

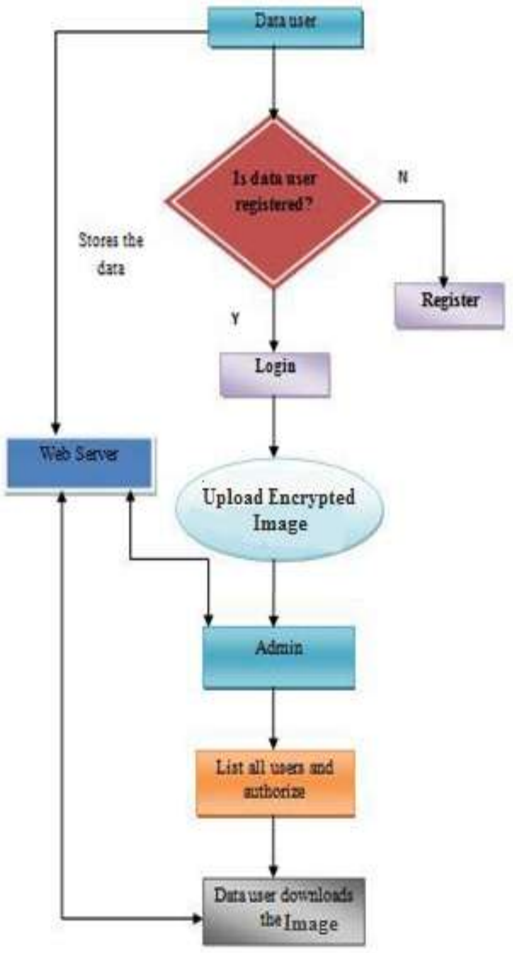

Fig. 2 Flow chart for data retrieval for user

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.

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.

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.

CONCLUSION

This paper proposes a near real-time scheme, called FAST, to support efficient and cost-effective searchable data analytics in the cloud. FAST is designed to exploit the correlation property of data by using correlation-aware hashing and manageable flat-structured addressing. 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. We discuss how the FAST methodology can be related to and used to enhance some storage systems, including Spyglass and Smart Store, as well as a use case. FAST is demonstrated to be a useful tool in supporting near real-time processing of real-world data analytics applications. 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 image 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

[4] H Alani, K O'Hara and N Shadbolt, Ontocopi: Methods and tools for identifying communities of practice. Intelligent Information Processing 2002, Vol. 221, pp. 225 236.
pp. B Aleman-Meza, C Halaschek-Wiener, I B Arpinar, C Ramakr-ishnan and A P Sheth, Ranking complex relationships on the semantic web. IEEE Internet
Computing, Vol. 9, No. 3, 2005, 37–44.

[5] B Aleman-Meza, M Nagarajan, C Ramakrishnan, L Ding,
P Kolari, A P Sheth, I B Arpinar, A Joshi and T Finin, Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection..
WWW 2006, ACM, 407–416.

[6] K Anyanwu, A Maduko and A P Sheth, Semrank: ranking complex relationship search results on the semantic web. WWW 2005, ACM, pp. 117–127.

[7] K Anyanwu and A P Sheth, -Queries: Enabling Querying for Semantic Associations on the Semantic Web. WWW 2003,
pp. 690–699.

[8] D Artz and Y Gil, A survey of trust in computer science and the semantic web. J. Web Sem., Vol. 5, No. 2, 2007, pp. 58–71.

[9]S. Lakshminarasimhan, J. Jenkins, I. Arkatkar, Z. Gong, H. Kolla, S.-H. Ku, S. Ethier, J. Chen, C. S. Chang, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, "ISABELA-QA: Query-driven analytics with ISABELA-compressed extreme-scale scientific data," in Proc. Int. Conf. High Perform. Comput, Netw., Storage Anal., 2011, pp. 1–11.

[10] M. Mihailescu, G. Soundararajan, and C. Amza, "MixApart: Decoupled analytics for shared storage systems," in Proc. 3rd USENIX Conf. File Storage Technol., 2013, pp. 133–146.

[11] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pebay, D. Thompson, H. Yu, F. Zhang, and J. Chen, "Combining in-situ and in-transit processing to enable extreme-scale scientific analysis," in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., 2012, p. 49.