International Journal of Research Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

A Novel Technique for Semantic Search Method for Large Scale Storage Systems in Cloud Using ABE

NARAGONI.DURGAPAVANI[1],DASARI.ALEKHYA[2],

KURAPATI.JYOSTHNA RANI[3],NANDAM.AKHIL SAI[4]

YEMINENI.ASHOK[5]

[1]B.Tech (CSE), Dhanekula Institute of Engineering and Technology, A.P., India.

[2]B.Tech (CSE), Dhanekula Institute of Engineering and Technology, A.P., India.

[3]B.Tech (CSE), Dhanekula Institute of Engineering and Technology, A.P., India.

[4]B.Tech (CSE), Dhanekula Institute of Engineering and Technology, A.P., India.

[5]Assistant Professor, DEPT OF CSE, Dhanekula Institute of Engineering and Technology, A.P., 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 by ABE. This process is called FAST. 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.

***Keywords*---**Large scale storage systems, Cloud, Semantic queries, FAST, Flat structured addressing, hashing.

I. 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 [3] 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[1]. 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 [1]. 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.

II. RELATED WORK

In this section, we present a brief survey of recent studies in the literature most relevant to the FAST research from the aspects of data analytics, searchable file systems and deduplication-based redundancy detection. Data analytics. Data analytics has received increasing attention from both industrial and academic communities. In order to bridge the semantic gap between the low-level data contents and the high-level user understanding of the system, a behavior-based semantic analysis framework [9] is proposed, which includes analysis engine for extracting instances of user-specified behavior models. ISABELAQA[10]is a parallel query processing engine that is designed and optimized for analyzing and processing spatiotemporal, multivariate scientific data. In this context, FAST is a useful tool that complements and improves the existing schemes to obtain correlated affinity from near duplicate images and execute semantic grouping to support fast query service. Searchable file systems. Spyglass [8] exploits the locality of file namespace

and skewed distribution of metadata to map the namespace hierarchy into a multi-dimensional K-D tree and uses multilevel versioning and partitioning to maintain consistency.

III. Existing System:

ISABELAQA [10]is a parallel query processing engine that is designed and optimized for analyzing and processing multivariate scientific data. Mix Apart uses an integrated data caching and scheduling solution to allow MapReduce computations to analyze data stored on storage systems.The frontend caching layer enables the local storage performance required by data analytics. The shared storage back-end simplifies data management.Spyglass[8] exploits the locality of file namespace and skewed distribution of metadata to map the namespace hierarchy into a multi-dimensional K-D tree and uses multilevel versioning and partitioning to maintain consistency.

IV. Proposed system:

In the context of this paper, searchable data analytics are interpreted as obtaining data value/worth via queried results, such as finding a valuable record, a correlated process ID, an important image, a rebuild system log, etc.We propose a novel near-real-time methodology for analyzing massive data, called FAST, with a design goal of efficiently processing such data in a real-time manner.The key idea behind FAST is to explore and exploit the correlation property within and among datasets via improved correlation aware hashing and flat-structured addressing to significantly reduce the processing latency of parallel queries, while incurring acceptably small loss of accuracy.The approximate scheme for real-time performance has been widely recognized in system design and high-end computing. In essence, FAST goes beyond the simple combination of existing techniques to offer efficient data analytics via significantly increased processing speed. Through the study of the FAST methodology, we aim to make the following contributions for near real-time data analytics.We propose semantic search method for large scale storage systems in cloud by using ABE.

Advantages of proposed system:

-Space-efficient summarization
-Energy efficiency via hashing
-Semantic-aware namespace
-Real system implementation

ABE (Attribute based encryption):

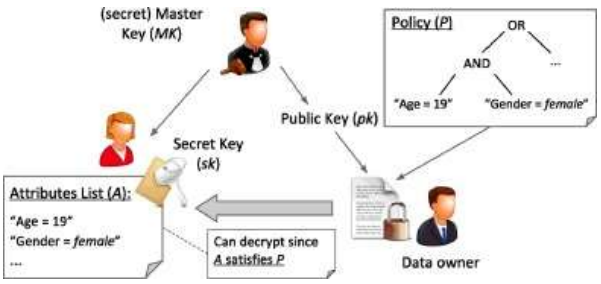
This project has a wide scope of data processing of the large amount of data for development by including some extra features that is ABE. Attribute-based encryption is a type of public key encryption in which the secrete key of a user and the cipher text.In this project we can use CP-ABE.

Ciphertext-Policy ABE:

In ciphertext-policy attribute-based encryption (CP-ABE) a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be to decrypt a ciphertext, if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions, disjunctions and (k,n)(k,n)-threshold gates, i.e., kk out of nn attributes have to be present (there may also

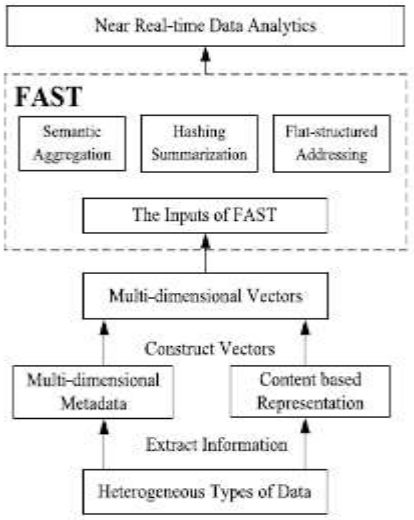
be non-monotone access policies with additional negations and meanwhile there are also constructions for policies defined as arbitrary circuits). For instance, let us assume that the universe of attributes is defined to be {A,B,C,D}{A,B,C,D} and user 1 receives a key to attributes {A,B}{A,B} and user 2 to attribute {D}{D}. If a ciphertext is encrypted with respect to the policy (A∧C)∨D(A∧C)∨D, then user 2 will be able to decrypt, while user 1 will not be able to decrypt.

CP-ABE thus allows to realize implicit authorization, i.e., authorization is included into the encrypted data and only people who satisfy the associated policy can decrypt data. Another nice features is, that users can obtain their private keys after data has been encrypted with respect to policies. So data can be encrypted without knowledge of the actual set of users that will be able to decrypt, but only specifying the policy which allows to decrypt. Any future users that will be given a key with respect to attributes such that the policy can be satisfied will then be able to decrypt the data.



CP-ABE

V. System Design:



VI. IMPLEMENTATION

System Construction Module

In the first module we develop the System Construction module. 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.

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.

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.

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.
We propose to use a crowd-based aid, i.e., personal images that can be openly accessed, to identify helpful clues. High-resolution cameras offer high image quality and multiple angles. Repeatedly taking pictures can further guarantee the quality of snapshots.

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. 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.

VII. CONCLUSION

We explore a semantic Search method for processing the large scale data volumes in the cloud by ABE. This process is 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. This enables FAST to significantly reduce processing latency of correlated file detection with acceptably small loss of accuracy. 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 sematic Search method for processing of real-world data analytics applications.

REFERENCES

[1]Real-time Semantic Search using Approximate Methodology for Largescale Storage Systems Yu Hua, Senior Member, IEEE, Hong Jiang, Fellow, IEEE, Dan Feng, Member, IEEE

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

[3] Gartner, Inc., —Forecast: Consumer digital storage needs, 2010-2016,‖ 2012.

[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] A. W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. L. Miller, "Spyglass: Fast, scalable metadata search for large-scale storage systems," in Proc. 7th USENIX Conf. File Storage Technol., 2009, pp. 153–166

[9] A. Viswanathan, A. Hussain, J. Mirkovic, S. Schwab, and J. Wroclawski, "A semantic framework for data analysis in networked systems," in Proc. 8th USENIX Conf. Netw. Syst. Design Implementation, 2011, pp. 127–140.

[10] S. Lakshmi narasimhan, 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 .HighPerform. Comput., Netw. ,StorageAnal ,2011,pp.1–11.