

Semantic Updated Metadata Paradigm With Efficient Retrieval

K.Ashok Kumar

PG Scholar,CSE,Rajalakshmi Engineering College,
Thandalam,Chennai.

S.Vinod Kumar,

Associate Professor,CSE,Rajalakshmi Engineering
College,Thandalam,Chennai.

Abstract—Semantic Container file store is a kind of storage system which can be used to store and retrieve thousands of files within a single file. It is similar to the file system. But here, before saving the file, the selected file will be encrypted and compressed to improve its security and to reduce the storage space. It can be used to secure sensitive and confidential information from unauthorized access. Based on LSI (i.e.,) Latent Semantic Indexing different types of files will be clustered and classified. The automatic indexing of files and grouping based on relativity is called “semantic” because user programmable nature of the system uses information about the semantics of updated file system objects to extract the properties for indexing. Indexing key properties of file system objects and indexing caching on the file system is one of the fantastic features of our system. File Versioning and Basic CRUD(i.e., Create, Read, Update and Delete) operations are also performed inside the file store system. Content based file retrieval also possible by using LSI technique. Semantic Container file store acts just like a single file where only the authorized users can access and view different files which are stored inside the semantic container.

Keywords: *Semantic Container File Store, LSI, CRUD, Latent Semantic Indexing.*

I. INTRODUCTION

Without File System, storing and efficient retrieving of any file from the storage is quite complex. In order to achieve fast retrieval and to control data within a specific storage system, File System was introduced. File system will instruct how and where the particular data should be placed. It is also used to manage and control the data. Semantic file store system is used to achieve efficient retrieval of data, to reduce storage size and to improve security by Rijndael encryption algorithm and Huffman encoding algorithm for compression. In order to

improve security, before uploading any file into the file store system, the selected files will be compressed and encrypted. Various CRUD operations like Create, Read, Update, and Delete are also performed inside the file store system. Latent Semantic Indexing is used to perform automatic semantic grouping and classification of data. LSI uses singular value decomposition technique for indexing and efficient retrieval. SVD is mainly used to identify patterns in the relationship between the terms and concepts contained in unstructured collection of files. Versioning gives detailed information about the different types of same content files. In Versioning, if the existing file is getting added with slight change or modification then the new file will be added with some other extensions. While storing the file into the semantic file store system, the file will be classified and stored into two separate fields. One is data and another is index field. Each and every index of the files will be stored under index field and all the information will be stored in the data field. The main goal of Semantic file store system is efficient retrieval of data even in the complex and unstructured collection of data. Top searched queries can be retrieved much faster from cache memory.

II. RELATED WORK

Providing security to a single file is very easy, whereas its quite difficult in case of providing security to the large volume of data. The tremendous growth in data volume and storing complex data creates great challenge to the file system. At the same time providing security to large file system is very difficult and impossible. So, In order to provide security to the complex data items, semantic file store system was proposed where the stored data are encrypted and compressed. [1] SANE, Semantic

Aware Namespace management is based on hierarchical directory trees. It works on the semantic relationships existing among different types of files. Normal file system often fails to provide real time response to complex data lookups. But SANE exploits semantic correlations among the different types of files which are stored in the file store system. It also provides file pre fetching and data de duplication.[2] type -ahead search data in xml allows the system to predict the correct data even if the user did any minor mistakes in the search keywords. (i.e.,) whatever the typed keyword, the system will make auto completion by supporting queries with multiple keywords in the xml data. It can also predict and find the high quality answer which are matched with the searched keyword. In this method high interactive speed can be achieved by effective algorithms and indexing .Based on Ranking the exact file can be obtained.[3] Smart Store, Here Semantic grouping algorithm is used to identify files which are correlated with each other based on physical attributes or behavioural attributes. It can also handles complex data and smart store system also acts as a high scalable system. But in all the above concepts security is lacking. In the file store system security is incorporated by performing compression and encryption. Before encrypting the files, the selected file will be split into different files. The main goal of smart store is to limit the no of searches even in case of complex data lookups.[4] Haystack, It mainly focused on facebook photo storage. Haystack will provide exact and inexpensive solutions by reducing the no of disk operations which are needed for the metadata lookups. Hence throughput will be increased and the no of disk operations will be reduced in this case. Facebook storing billions of files per day, hence in need of large storage system for efficient storing and retrieval of file from the storage. Cost effective and fault tolerant are the advantages of haystack[5] The major tasks like searching and retrieving top-k queries and processing of aggregates from the file system are not possible in hierarchical system(i.e.,) NTFS Network File System. In Just in Time Analytics after certain no of disk accesses, the system can predict the accurate answers from top-k queries without in need of any prior knowledge.(i.e.,)Which are the top five fastest century scored by sachin ? , How many photos have been uploaded into storage system since 1 week ago? .[6] Spyglass: Allows fast and complex searches over file

metadata. Spyglass allows user to manage and maintain their files Exploiting metadata search properties, fast and scalable performance can be achieved through the use of novel metadata search technique. Index and data will be frequently updated, and also occupy very less disk space and metadata collection of data and processing is much faster than others. Index partitioning method allows to perform index control.[7] Removing the duplicate files from the file system is quite complex in the large scale file system. Due to the presence of duplicates in the file system, they may influence or affect the result of the any other process in the file system. Here duplication of data in the file system will be removed while retrieving the file. Data are binded to record linkage and the performance can be measured based on different indexing techniques. Default inherent indexing is not achieved in this technique.[8] On collecting the annual snapshots of the file system metadata from the different types of file system, various changes like file size, file type, file age , total storage capacity, space used, modified file, no of users, file system users can be obtained. Distribution of directory structure and namespace distribution are the key concepts of this approach.

III. PROPOSED SYSTEM

In the proposed system, Semantic file store system is created to store huge volume of data. It will act as a file system where files can be uploaded, updated and stored for future retrieval. The main goal is to ensure security while storing the files in the file system where only authorized users can alone upload and access any files from the file store system. Security can be improved by encryption of files using certain algorithms and compression is made to reduce the space of the storage system. Based on Indexing, files can be retrieved much faster. Versioning details are also another important feature. Indexing will take place based on LSI (i.e.,) Latent Semantic Indexing. is an indexing and retrieval method that uses SVD Singular Value Decomposition technique to identify patterns in the relationship between the terms and concepts used in the unstructured collection of data. Rijndael encryption algorithm and Huffman encoding algorithm for compression are the two most common algorithms which are used to ensure encryption and compression respectively.

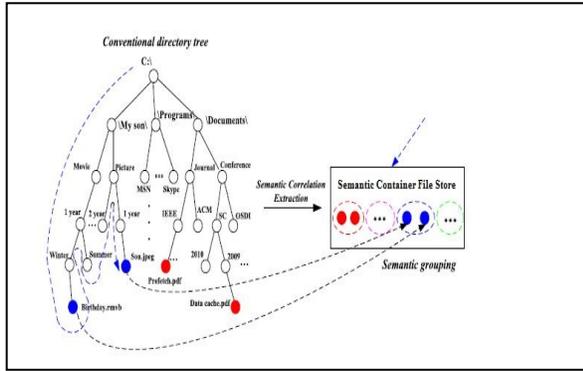


Figure 1 Semantic Grouping of files

Figure 1 clearly explains about the semantic grouping of files. The related files will be grouped together based on semantic relationship existing among them.

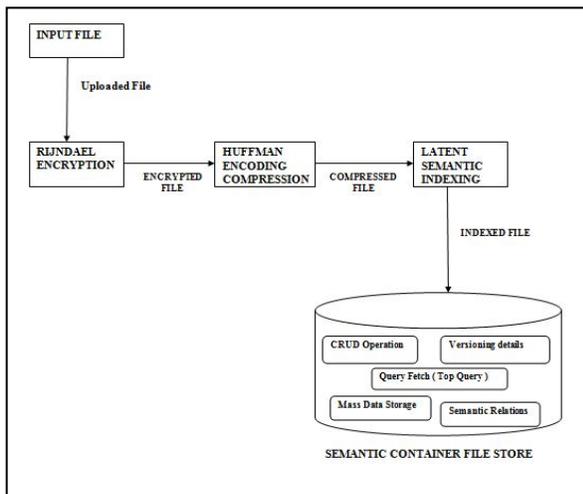


Figure 2 System Architecture

Figure 2 explains the overall architecture of the project. Here user will provide the input file into the system. The given input file will get encrypted and compressed using Rijndael and Huffman Encoding Algorithms. Indexing of file will take place based on the concept of Latent Semantic Indexing and then the file will be stored in the semantic container. File prefetching and Deduplication are the advantages.

Figure 3 explains about various different operations of our system. Semantic Container file store acts as a middle ware in the file system.

Different operations are... Metadata describes other data. It provides details about certain different information of the particular file. Various CRUD

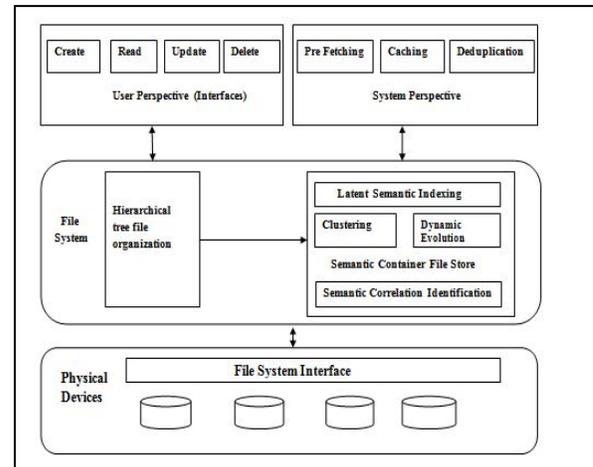


Fig.3 Semantic Container File Store as a middleware in file systems

information of the particular file. Various CRUD operations are performed in the semantic file store system. Semantic relations are relationship that existing between different files. Based on Semantic Relation and Indexing, the different files will be clustered and classified. Versioning details allows the file to be stored in some other extension if that particular file is already existing in the semantic file store system.

A. Rijndael Algorithm

Rijndael algorithm follows the following steps.

- Sub Bytes Step
- Shift Rows Step
- Mix Columns Step
- Add Round Key Step

B. Huffman Encoding Compression Algorithm

Huffman Encoding Compression algorithm follows the following steps. The steps are

- Count the frequency of each character in the file to be encoded.
- Store characters as tree nodes and put them into a priority queue.
- Begin to build the tree.
- Finish building the tree.
- Create an encoding map.
- In the output file, include the encoding map as a header.
- Encode the file

IV. MODULES

- File store creator module
- Data upload Module
- Data Visualize module
- Index Tuner Module
- Functional Operation Module

A. File Store Creator Module

This module enables user to create a file store system. A file store system should be built to wrap any kind of file formats (e.x.,) jpeg, jpg, doc. Since audio and video can be encoded with the specified algorithm the file store can provide a single file format to the user.

B. Data Upload Module

Users document will contain sensitive and confidential information. So, in order to improve the security of the data or file will be encrypted in the file store system. And to avoid maximum storage the files will be compressed and saved in the file store system. It provides set of tools to interact with the semantic documents and to form social file store system around the shared documents.

C. Data Visualize Module

This module provides the way where user can view their stored documents. Here documents will be stored in grouped manner based on the key words and the meta data about the documents. The search starts by the user providing normal query which will be translated into the semantic concept based query. The top most files can be easily searched and retrieved.

D. Index Tuner Module

To improve the performance of the document storage and to efficiently the stored files from the storage indexing is performed based on Latent Semantic Indexing. This module will cluster and classify the data and files stored in the file store system based on their semantic relationship existing among them. The documents will be hierarchically stored and grouped based on the logical metadata information about the documents.

E. Functional Operation Module

This module gives detailed view about the various operations that are happening inside the file

store system. Annotation refer to the specific part of the original data. Various operations performed are CRUD (i.e.,) Create, Read, Update and Delete. Inside the file system all the mentioned operations can happen to update the new information and delete the duplicated data. Annotation is a metadata attached to text, image or other data. Versioning details provide an feature to add the already existing file into the system with other name or extension.

V. EXPERIMENTAL RESULTS

The experimental results are shown in the following figures with explanation.

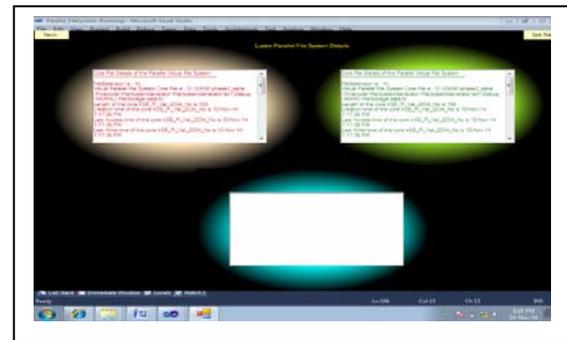


Figure 4 Semantic Container Description

Figure 4 shows the Semantic Container Description with the allocated storage space along with the index and storage field. Here 2 file system is created for the storage purpose.



Figure 5 Compressed and Encrypted files

Figure 5 shows the final outcome of the input files once the file gets compressed and encrypted.

In the below figure the encrypted and compressed files will be stored in the container and it will be used for later retrieval purpose.

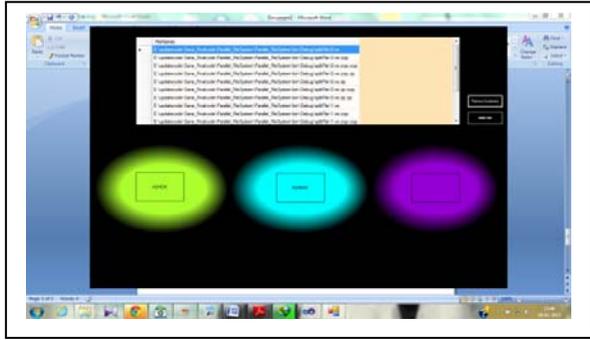


Figure 6 Insertion of files into Semantic Container

The stored files can be efficiently retrieved by the latent semantic Indexing.

VI. CONCLUSION

In this paper semantic container file store is created to store the large collection of data, Where only the authorized users can alone access the data. Security is increased by performing the encryption and storage space is reduced on compressing the document before storing into the file store system. By Indexing, data can be efficiently retrieved from the storage within short span of time.

REFERENCES

- [1].Yu Hua and Yifeng Zhu, "SANE: Semantic-Aware Namespace in Ultra-Large-Scale File System", IEEE Transactions on Parallel and Distributed System, VOL.25, NO 5, MAY 2014.
- [2] Jianhua Feng, "Efficient Fuzzy Type-Ahead Search in XML data , IEEE Transactions on Knowledge and Data Engineering, VOL 24, NO.5 MAY 2012
- [3] Y.Hua, J.Jiang, Y.Zhu, D.Feng, and L.Tian, "Smart Store:A New Meta data Organization Paradigm with Semantic-Awareness for Next-Generation File Systems, "Proc. ACM/IEEE SuperComputing, Conf. (SC), 2009
- [4] D.Beaver, S.Kumar, H.Li, J.Sobel, and P.Vajgel, "Finding a Needle in HayStack: Facebooks photo storage", Proc. Ninth USENIX conf. Operating Systems Design and Implementation.
- [5] H.Huang, N.Zhang, W.Wang, G.Das, and A.Szalay, "Just-In-Time Analytics on Large File Systems, "Proc. Ninth USENIX Conf. File and Storage Technologies (FAST),2010
- [6] A.W. Leung, M.Shao, T.Bisson, S.Pasupathy, and E.L. Miller, "Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems, " Proc. seventh USENIX Conf. File and Storage Technologies(FAST), 2009
- [7].M.Seltzer and N.Murphy, "Hierarchical File Systems are Dead", Proc. 12th Conf. Hot Topics in Operating Systems (HotOS'09), 2009
- [8] N.Agarwal, W.Bolosky, J.Douceur, and J.Lorch, "A Five- Year Study of File-System Metadata, "Proc. USENIX Conf. File and Storage Technologies (FAST), 2007

[9] A.Traeger, E.Zadok, N.Joukov, and C.Wright, "A Nine Year Study of File System and Storage Benchmarking," ACM Trans. Storage, vol.4, pp. 1-56,2008

[10] S.Patil and G.Gibson, "Scale and Concurrency of GIGA+: A File System Directories with Million of Files, "Proc. Ninth USENIX Conf. File and Storage Technologies(FAST), 2011.

AUTHORS PROFILE

[1]K.Ashok Kumar is currently pursuing Master Degree program in Computer Science and Engineering in Rajalakshmi Engineering college,Chennai,India Email id: k.ashokbecse@gmail.com

[2]S.Vinod Kumar Associate Professor in Department of Computer Science and Engineering in Rajalakshmi Engineering college,Chennai,India Email id: vinodkumar.s@rajalakshmi.edu.in