International Journal of Research
eISSN: 2348-6848 & *pISSN:* 2348-795X*Vol-4 Special Issue-6*
National Conference on Innovations in Information and Communication Technology
Held on 17-03-2017, Organized by Department of Information Technology, Meenakshi Sundararajan Engineering College,
363,Arcot Road Kodambakkam, Chennai 600024, India

EFFECTIVE DATA RETRIEVAL FOR VIRTUAL CHUNK AND DATA PARTITIONING USING ENCRYPTION TECHNIQUE

E. Dhivya[1], Ms. D. Jayanthi[2]
[1] *PG Scholar, Department of Information Technology, SVCE,*
[2]Assistant Professor, *Department of Information Technology, Sri Venkateswara College of Engineering, Chennai,Tamil Nadu.*

Abstract: -
Data partitioning is normally done for manageability, performance or availability reasons, as for load balancing. In existing system, Data compression could ameliorate the I/O pressure of data-intensive scientific applications. Virtual chunks are logical blocks pointed at by appended references without breaking the physical continuity of the file content. In the proposed system,the I/O performance of random data access in scientific applications and high-performance computing (HPC) system is strengthen to split and store the data efficiently using dynamic virtualization concept.Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the cipher text. Attribute-based encryption based keys are distributed among the corresponding admin so that access policy is applied. The proposed algorithm will reduce the time for fetching I/O performance results from the table.

Keywords:-

High Performance Computing (HPC), File Compression, Hadoop Distributed File systems, Big Data, Virtual chunk, Attribute based encryption (ABE).

I. INTRODUCTION

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime". So we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

Big data is a term utilized to refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. The nature of big data is indistinct and involves considerable processes to identify and translate the data into new insights. The term "big data" is relatively new in IT and business. However, several researchers and practitioners have utilized the term in previous literature. For instance, referred to big data as a large volume of scientific data for visualization. Several definitions of big data currently exist. Meanwhile and defined

big data as characterized by three Vs: volume, variety, and velocity. The terms volume, variety, and velocity were originally introduced by Gartner to describe the elements of big data challenges. IDC also defined big data technologies as "a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and/or analysis" specified that big data is not only characterized by the three Vs mentioned above but may also extend to four Vs, namely, volume, variety, velocity, and value.

Most of the data generated from mobile applications are in unstructured format. For example, text messages, online games, blogs, and social media generate different types of unstructured data through mobile devices and sensors. Internet users also generate an extremely diverse set of structured and unstructured data.

In the existence, Data compression could ameliorate the I/O pressure of data-intensive scientific applications. File level compression can barely support efficient random accesses to the compressed data: any retrieval request need trigger the decompression from the beginning of the compressed file. File-level compression barely supports efficient random accesses to the compressed data: any retrieval request need trigger the decompression from the beginning of the compressed file.

Block-level compression provides flexible random accesses to the compressed blocks, but introduces extra overhead when applying the compressor to each and every block that results in a degraded overall compression ratio. File systems are presented in a form of name node that registers attributes, such as access time, modification, permission, and disk space quotas. The file content is split into large blocks, and each block of the file is independently replicated across data nodes for redundancy and to periodically send a report of all existing blocks to the name node.

II.SYSTEM DESIGN

Data is splitted and stored using dynamic virtualization concept and we do not break the original file into physical chunks or blocks, but append a small number of references to the end of file. Each of these references points to a specific block that is considered as a boundary of the virtual chunk. In which we deploy three layers of connectivity named as layer 1, layer 2, layer 3 are the layers in our project. Assuming a College Global Data which is entirely stored in the Layer 1, Country wise data are classified and stored in the Layer 2and finally State / region wise data are stored in the Layer 3. ABE based keys are distributed among the corresponding admin so that access policy is applied. This system will store the entire file as such in the Layer 1 and as well as the data is splitted and stored in the other two layers. This process will surely reduce the time for the fetching any results from the table. To split the data and stored in different layers and easily retrieve the data using big data analytics.

To strengthen the I/O performance of random data accesses in scientific applications and high-performance computing (HPC) system, dynamic virtualization concept is used. Attribute-based encryption is a type of public-key

International Journal of Research
eISSN: 2348-6848 & *pISSN:* 2348-795X*Vol-4 Special Issue-6*
National Conference on Innovations in Information and Communication Technology
Held on 17-03-2017, Organized by Department of Information Technology, Meenakshi Sundararajan Engineering College,
363,Arcot Road Kodambakkam, Chennai 600024, India

encryption in which the secret key of a user and the cipher text are dependent upon attributes. In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text.

In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access. ABE uses a tree-based access structure which must be satisfied with a given set of attributes in order to decrypt the data. The tree-based access structure allows the encryptor to specify which attributes can decrypt the data. For those users who are really familiar with their data, a function that adjusts any particular range of data with an arbitrary number of references is also desirable.

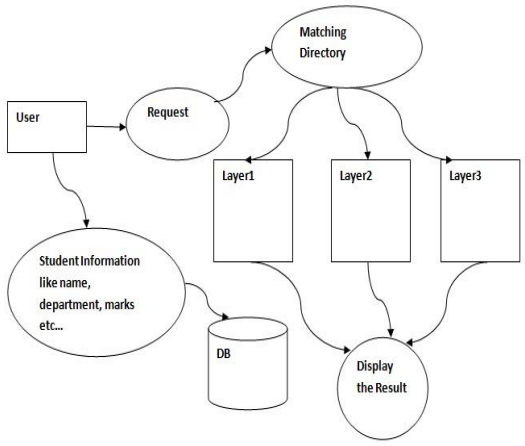

Figure 1: Dataflow Model

Above figure explains detailed about Dataflow diagram where data matches with query then display their result.

III.DESIGN AND IMPLEMENTATION

A. DATA GENERATION AND USER REGISTRATION

Authentication is a process in which the credentials provided are compared to those on file in a database of user's information. If the credentials match, the process is completed and the user is granted to access, otherwise user wants to register newly for cluster, global and region.

User Registration

At first initial stage all user must create own user name and password. After the registration the user can login to their own space. This application verify the username and password which is either matched or not with the user registration form which is already created by the user while user registration process. If the valid user did not remember the username or password correctly the user can generate their own password. Based on the User's request, the Service Provider will process the User requested Job and respond to them. All the User details will be stored in the Database of the Service Provider. In this Project, we will design the User Interface Frame to Communicate with the Server through Network Coding using the programming Languages like Java. By sending the request to Server Provider, the User can access the requested data if they authenticated by the Service Provider.

Data generation & partition

In data generation used three levels for student information that is segment

global directory, segment middle directory and segment table directory.

Partition is, first matching all directories based on student information. Finally get the result based on query, then the data are chunked and partitioned data are stored in drop box. Partitions of data have division of a global, region, cluster or its consistent element.

B.UPLOAD PAGE AND ANALYZE FRAME

Once the user successfully signed in into the server, the user is requested to search their data on Query based OR Location based. Once the user chooses the query based, the data link will displayed on the query based. If the user chooses the marks based query, the data will display in marks based like high marks or low marks. To get the data via query based.

Main server

A server is a computer program running to serve the requests of other programs, the "clients". Thus, the "server" performs some computational task on behalf of "clients". The clients either run on the same computer or connect through the network. Here the Server acts as the main resource for the client. Server is responsible for maintaining all the student information. So the server will process the user's request and get the concerned data from the database.

C.ATTRIBUTE BASED ENCRYPTION ACCESS CONTROL AND DATA RETRIEVAL

Attribute-based encryption (ABE) can be used for log encryption. Instead of encrypting each part of a log with the keys of all recipients, it is possible to encrypt the log only with attributes which match recipient's attributes. This primitive can also be used for broadcast encryption in order to decrease the number of keys used. In this module, Profile based data only access i.e. student profile.It is a public-key based one to many encryptions that allows users to encrypt. In which the secret key of a user and the cipher text are dependent upon attributes. In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text.

Attribute based encryption access control

The problem with attribute based encryption (ABE) scheme is that data owner needs to use every authorized user's public key to encrypt data. The application of this scheme is restricted in the real environment because it uses the access of monotonic attributes to control user's access in the system.

Map Reduce Technique

In this module, output of mapping part is the input of Reduce part. Map Reduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A Map Reduce program is composed of a Map() procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The Map Reduce System (also called infrastructure or framework) orchestrates the processing

International Journal of Research
eISSN: 2348-6848 & pISSN: 2348-795X Vol-4 Special Issue-6
National Conference on Innovations in Information and Communication Technology
Held on 17-03-2017, Organized by Department of Information Technology, Meenakshi Sundararajan Engineering College, 363,Arcot Road Kodambakkam, Chennai 600024, India

by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, for redundancy and fault tolerance.

Data retrieval with time frame

Once the user chooses their option to search the data, they have to enter the query that they want to search. If the user is choosing query based search mechanism, the result will be displayed as per query based.

If the user chooses the student performance based mechanism the data will displayed to the student based the performance they've chosen. Once the search has been made, the result will be displayed to the user. From that displayed result, the user is requested to select the data and Virtual Chunk at they are wanted. In this mapping part is the input of Reduce part.

D.SAMPLE SCREENSHOT

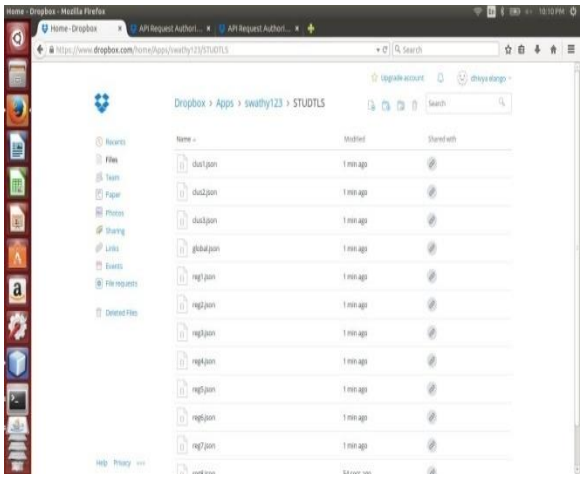

Figure 2: Data partitioning

The above snapshot includes the API request Authentication then provides the data is splitted and partition in drop box.

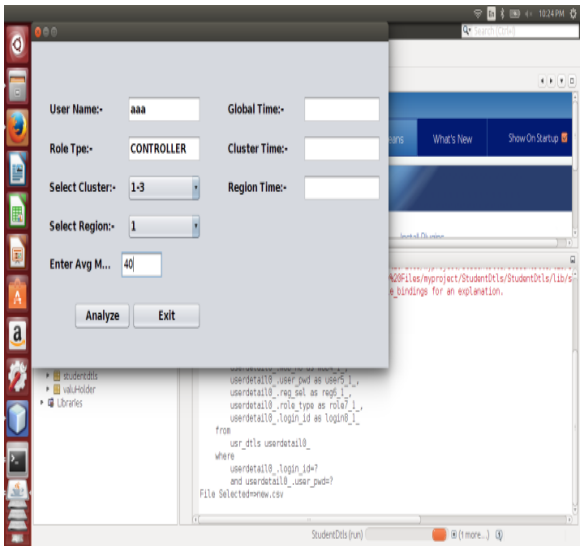

Figure 3: Attribute based encryption

The above snapshot includes the analysis of attribute based encryption algorithm. First select the role type and select their cluster and region then enter average marks. Finally analyze their frame if it matches it provides cluster and region matches.

Map Reduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

This snapshot includes the analysis of attribute based encryption algorithm. First select the role type and select their cluster and region then enter average marks. Finally analyze their frame if it does not matches it provides cluster and region are not matching

International Journal of Research
eISSN: 2348-6848 & *pISSN:* 2348-795X*Vol-4 Special Issue-6*
National Conference on Innovations in Information and Communication Technology
Held on 17-03-2017, Organized by Department of Information Technology, Meenakshi Sundararajan Engineering College,
363,Arcot Road Kodambakkam, Chennai 600024, India

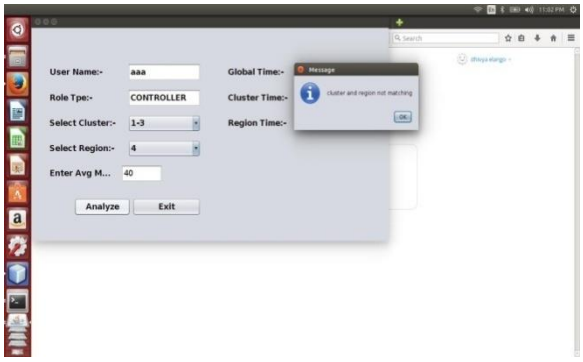

Figure 4: Analyze Frame

The above snapshot includes the analysis of attribute based encryption algorithm. First select the role type and select their cluster and region then enter average marks. Finally analyze their frame if it matches it provides entire global time, cluster time and region time. It should be fairly straightforward for the first type of users to meet their needs by extending the provided interfaces.

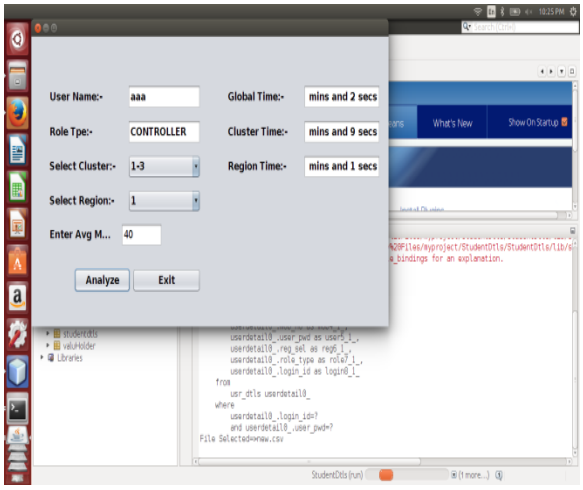

Figure 5: Display,Global,Region,Cluster Time

IV CONCLUSION AND FUTURE ENHANCEMENT

Virtual Chunk achieves the small special overhead of file-level compression as well as the small computational overhead of (physical) chunk-level decompression, the best of both worlds. I/O pattern follows a uniform distribution over the data: static Virtual Chunk and dynamic Virtual Chunk. Static Virtual Chunk assumes the references are equidistant without a prior knowledge of the application's characteristic and I/O patterns. Although static Virtual Chunk (VC) is applicable in many scenarios, it is still highly desirable to allow users to specify the reference position or distribution. We extend static Virtual Chunk (VC) to dynamic VC, a more general form of static VC to make the approach more applicable. Then formulate the procedures to use static and dynamic VC to incorporate split table compressors. An in-depth analysis is conducted for the abstract model of VC such as the optimal parameter setup.

Our future work overcomes the advantage of security using encryption technique.Comparsion of this encryption to improve their performance and its security.

REFERENCES

1. D. Laney, S. Langer, C. Weber, P.Lindstrom, A.Wegener (IEEE, 2013),Assessing the effects of data compression in simulations using physically motivated metrics
2. D. Zhao, Z. Zhang, X. Zhou, T. Li, K. Wang, D. Kimpe, P. Carns, R.Ross, and I. Raicu, "FusionFS: Toward supporting data-intensive scientific applications on extreme-

scale distributed systems," in Proc. IEEE Int. Conf. Big Data, 2014, pp. 61–70.

3. Available:http://www.hdfgroup.org/ HDF5/doc/index.html (2014).
4. TekinBicer, Jian Yin, David Chiu, and Gagan Agrawal, Karen Schuchardt (IEEE 2016), Integrating online compression to accelerate large-scale data analytics applications.
5. Dongfang Zhao, Ning Liu, Dries Kimpe, Robert Ross, Xian-He Sun, IoanRaicu(IEEE 2015),Towards exploring data-intensive scientific applications at extreme scales through systems and simulations
6. S. Lakshminarasimhan, D. A. Boyuka, S. V. Pendse, X. Zou, J. Jenkins,V. Vishwanath, M. E. Papka, and N. F. Samatova, "Scalable in situ scientific data encoding for analytical query processing," in Proc. 22nd Int. Symp. High-Perform. Parallel Distrib. Comput., 2013,pp. 1–12.
7. D. Zhao, J. Yin, and I. Raicu, "Improving the I/O throughput for data-intensive scientific applications with efficient compression mechanisms," in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., Poster Session, 2013, pp. 1–2.
8. D. Zhao, J. Yin, K. Qiao, and I. Raicu, "Virtual chunks: On supporting random accesses to scientific data in compressible storage systems," in Proc. IEEE Int. Conf. Big Data, 2014, pp. 231–240.
9. D. Zhao, K. Qiao, and I. Raicu, "HyCache+: Towards scalable high-performance caching middleware for parallel file systems," in Proc. IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., 2014,pp. 267–276.
10. D. Zhao and I. Raicu, "HyCache: A user-level caching middlewarefor distributed file systems," in Proc. IEEE 27th Int. Symp. Parallel Distrib. Process Workshops PhD Forum, 2013, pp. 1997–2006.
11. D. Zhao, K. Burlingame, C. Debains, P. Alvarez-Tabio, and I.Raicu, "Towards high-performance and cost-effective distributed storage systems with information dispersal algorithms," in Proc.IEEE Int. Conf. Cluster Comput., 2013, pp. 1–5.