© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 08 | October – December 2016
Paper is available at www.jrps.in | Email : info@jrps.in

SECURITY AND CACHE MECHANISM IN HADOOP APPLICATION

[1]Kaushal Kumar, Research Scholar, Department of CSE, IIET Kinana, Jind, kaushal.kumar03@gmail.com
[2]Abhishek Bhatnagar, Assistant Professor , Department of CSE, IIET Kinana, Jind, ap.abhi.iiet@gmail.com

ABSTRACT: In earlier time traditional tools like SQL Databases, Files etc. were used to handle data and its issues. With increase in the volume of data, traditional tools struggled a lot to store, retrieve manipulate data and hence Hadoop and Big Data evolved. Security and Performance in any application is an issue which needs to be addressed with increasing expectation of immediate availability of data and information. Security poses a major challenge which can be addressed with the help with encryption and decryption mechanisms. The main objective of encryption is to safeguard the confidentiality of data stored on computer or transmitted via Internet or other media. In Modern era encryption algorithms play a crucial role in security assurance of Computer systems andcommunications across network as these algorithms can provide confidentiality, authenticity, data integrity and Non repudiation.Another aspect of today's modern application development is that developers have a wide variety of techniques and technologies available to improve application performance and end-user experience. One of the most widely used technologies is the cache mechanism. By using cache at the client side the applications cane greatly benefited by improving response times and reducing server I/O load. One of such examples is HTTP caching techniques which are always associated with the client side cache mechanisms.

[1]Introduction

Security and Performance in Hadoop Application

In today's world security of any application is as much important as the robustness of the application and many techniques are used for securing an application system. Many algorithms are designed and developed for the achieving the security of the system. Encryption & Decryption are the oldest type of cryptographic techniques which refers to the process of scrambling data so that the recipient cannot infer the information. All these encryption mechanisms are implemented in tradition applications and achieved excellent results with 128 bit RSA algorithms. As today's era is the era of Big Data, Hadoop and Hive; securing application on these platforms is as much important as it is on any traditional system. The objective here is to achieve the security on any application which is designed and developed on Hadoop system, the security feature is provided by encrypting the data and then processing the same to the warehouse. Once the enhanced security is provided as a feature in the applications built on Hadoop platform, it is utmost important to take care of the performance of the application. Improving the performance of the overall system is another objective of the research.

Performance of the application can be increased by processing the data or information locally at the client level rather than having a process to read the server data. Performance of the overall system is increased by processing a cache file at the client side and synchronizing the same file with the Warehouse. Once data is available in the client side file it will be processed and presented to the users for processing in terms of OLTP or OLAP data set, if it is not available in client system locally it will be synchronized with server data and then presented to the users. All in all the overall objective of today's applications is addressed in the research and tried to improve the performance of the system along with the security enhancements by applying encryption decryption mechanism. In order to achieve the security and performance in the applications built on the top of Hadoop system certain methodologies are followed which are as elaborated in the forthcoming sections, also the algorithm which used are elaborated in order to achieve the desired results for security and performance.

[2] Literature review

A Review Paper on Big Data & Hadoop by Harshawardhan

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 08 | October – December 2016
Paper is available at www.jrps.in | Email : info@jrps.in

S. Bhosale1, Prof. Devendra P. Gadekar2

This paper describes techniques of Big Data along with 3 Vs; Volume, Velocity & Variety of Big Data. It also focuses on Big Data processing problems. The technical challenges discussed must be addressed for efficient & fast processing of Big Data. Challenges include not just obvious issues of scale, but also heterogeneity, privacy, timeliness, provenance, & visualization, at all stages of analysis pipeline from data in a new possession to result interpretation. These technical issues identified are common across a wide variety of functional and application domains, & therefore not cost effective to address in context of one domain alone. This Paper also describes Hadoop which is an open source software used for processing of Big Data.

S. Vikram Phaneendra & E. Madhusudhan Reddy et.al. Demonstrated in the paper that in olden days data was less & can be easily managed by RDBMS but in recent times it is difficult to handle huge volume of data through RDBMS methods. In this paper authors told that big data differs from other data in 5 dimensions such as volume, velocity, variety, value & complexity. Authors illustrated Hadoop architecture consisting of name node, HDFS to handle big data management. Hadoop architecture which can handle large data sets and scalable algorithm which does log management application of big data and could be found out in financial, retail industry, health-care, mobility, insurance industry. Authors also focused on challenges that want to be face by business or firm when handling big data: - data privacy etc. [1].

Kiran kumara Reddi & Dnvsl Indira et.al. illustrated us with knowledge that Big Data is combination of structured , semi-structured ,unstructured homogenous & heterogeneous data .The author suggested to used to a new model to handle transfer of huge amount of data over network .Under this model, these transfers are relegated to low desirer time where ample and idle bandwidth available . This bandwidth could then be repurposed for big data program without impacting other client in system. Nice model uses a store and reply approach by utilize staging servers. Model is able to accommodate differences in time zones & variations in bandwidth. They suggested that new algorithms have necessary to transfer big data & to solve many problems like security, compression, routing algorithms [2].

Jimmy Lin et.al. used Hadoop which is currently large – volume data analysis " hammer" of choice, but there exists classes of algorithms that aren't " nails" in sense that they are not particularly amenable to Map Reduce programming model . He focuses on simple solution to find alternative non-iterative algorithms that solves same problem. Standard Map Reduce is well known & described in many places .Each iteration of page rank corresponds to Map Reduce job. Author suggested repeating graph, gradient descent & EM iteration which is typically result as Hadoop job within driven set up iteration &Check for convergences. Author suggests that if all you have is a tack hammer, throw away something that's not a nail [3].

[3] Tools & Technology

BIG DATA

Big data term is used for data sets which are so large and complex that traditional data processing applications are inadequate. Big data addresses challenges include analysis, capture, data correction, data search, data sharing, data storage, transfer, data visualization, data querying, updates and information privacy. The term Big Data often refers to the use of predictive analytics and user behaviour analytics or certain other advanced data analytics methods that extract value from huge data which is not possible with traditional RDBMS.

HIVE

Hive is a data warehouse management system which is built on Hadoop platform for providing data aggregation, querying and analysis. Hive provides an SQL-like interface to query the data stored in different databases and file systems that are integrated within Hadoop platform. The traditional SQL based queries are implemented in MapReduce Java API in order to execute SQL based applications and queries over a distributed database systems. Hive gives necessary SQL abstraction logic to integrate SQL-like Queries (HiveQL) in an underlying Java API without the need to implement queries in low-level Java based API.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 08 | October – December 2016
Paper is available at www.jrps.in | Email : info@jrps.in

NetBeans

NetBeans software development platform which is open source which is designed and developed in Java. The NetBeans is a platform which allows applications to be developed from different sets of modular software components which are called as modules. Applications which are based on the NetBeans Platform, which includes the NetBeans integrated development environment (IDE), which can be extended by third party developers. The NetBeans Integrated Development Environment is majorly intended for design and development in Java, but it also supports other languages as well, in particular PHP, C/C++ and HTML5. The NetBeans is a cross-platform which runs on various platforms like Microsoft Windows, Mac, unix, Linux, Solaris and other different platforms supporting a compatible JVM.

Shell Prompt:

Unix prompt, $, is called as a command prompt, which is issued by the shell. When the prompt is displayed, any user can type a command. The shell reads user's input after user press Enter Key. Unix determines the command that user want to execute by looking at the initial letters of user input.

Shell Types:

In UNIX shells are majorly categorized in to two types:

The Bourne shell--The default prompt is the $ character, signifies it is Bourne Shell. Further Bourne shell can be sub categorized in to the following categories:

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)

The C shell-- The default prompt is the % character, signifies it is C Shell, it has following categories:

- C shell (csh)
- TENEX/TOPS C shell (tcsh)

Originally UNIX shell script was composed in mid 1970s by Stephen R. Bourne while he was working at AT&T Bell Labs in New Jersey. The Bourne shell was the first ever shell to appear on UNIX operating systems, hence it is referred as "the shell". The Bourne shell is generally installed as /bin/sh on various versions of UNIX. For the same reason, bourne shell is the shell of choice for writing scripts to use on different versions of UNIX.

[4] Proposed Implementation

Enhancing the security of Hadoop API by providing xor based data security.

Bit Wise encryption is implemented using XOR based security mechanism. The AES encryption uses XOR on individual bytes for encrypting the data. Algorithm works on the principal that key will be XORed with the intermediate result and after that permuted and substituted. The XOR operation is very common as a component in more complex ciphers. By using a constant repeating key, a simple XOR cipher can be obtained and used in frequency analysis. If the content of any message can be guessed or otherwise known then the key can be revealed. Its primary advantage is that it is very simpleto implement, and that the XOR operation is computationally not expensive.

A very simple repeating XOR (i.e. using the same key for xor operation on the whole data) cipher can be used for hidinginformation and encrypting the data. Model for encrypting the data using XoR operation is as shown in the figure and explained ahead:

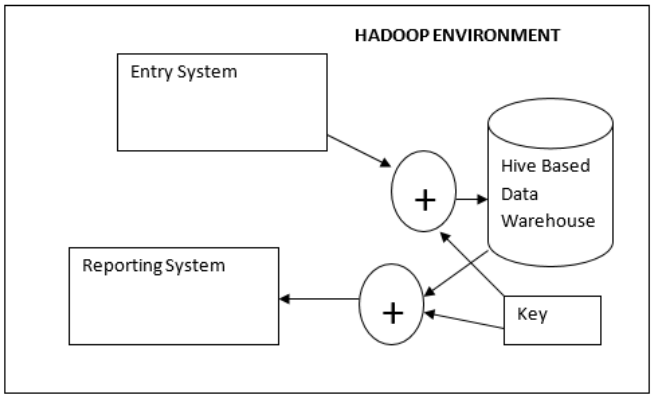

Fig 1 Hadoop enviroment

IMPROVING HADOOP PERFORMANCE BY INTRODUCING CACHE TABLES:

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 08 | October – December 2016
Paper is available at www.jrps.in | Email : info@jrps.in

When user would access data from hive based data warehouse then an instance of data would be available locally at the client side and hence preserved like a cache. Next time whenever the data is queried it would be accessed from locally available set instead of accessing data from remote node. In order to avoid the access of old data time stamping mechanism would be used. If the time stamp of local data is older than that of remote data then data would be updated from remote to local otherwise data would be captured from local copy itself. This explained mechanism would definitely improve the performance of the overall application. It is explained with the help of table structure as ahead, in following model T1 instance from hive data ware house would be copied to application layer 1. Next time if information is not updated on Hive based data warehouse then data would be captured from local T1. Same operation would be done in case of T2 with application layer 2. As we see the data is accessed locally and hence network I/O is not required improving performance.

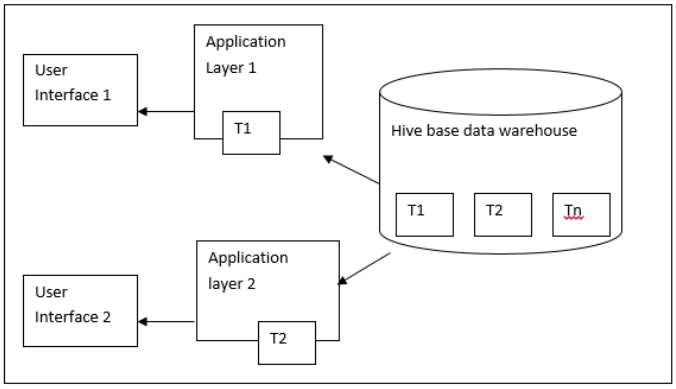

Fig 2 hive base data warehouse

[5] Implementation

Whole implementation is divided majorly in to 5 parts viz.:

- Setting up Hadoop Server and its components
- Configuring and starting Hadoop Server
- Setting up Java and NetBeans
- Designing and Developing Data Model on Hive
- Designing and Developing the application

Setting up Hadoop Server and its components includes the following steps as shown in the diagram below:

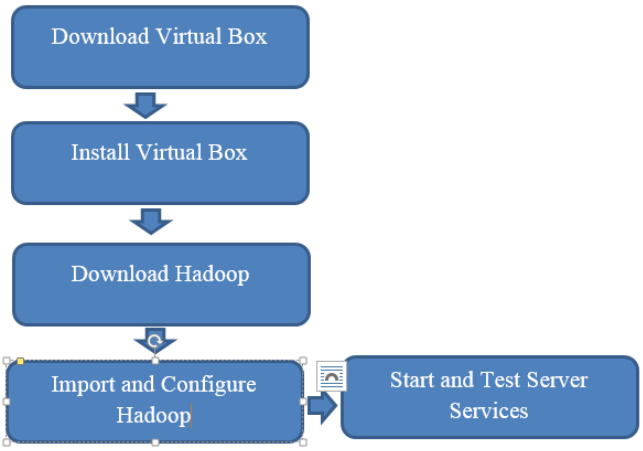

Fig 3 Downloading

After completing setup of HDP hosted on an oracle virtual machine, below screen shows how the imported Hadoop framework will look like in oracle virtual machine

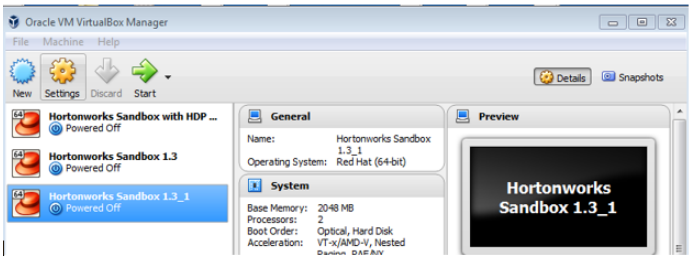

Fig 4 oracle virtual machine

Settings need to be configured on motherboard, processor acceleration and other things.

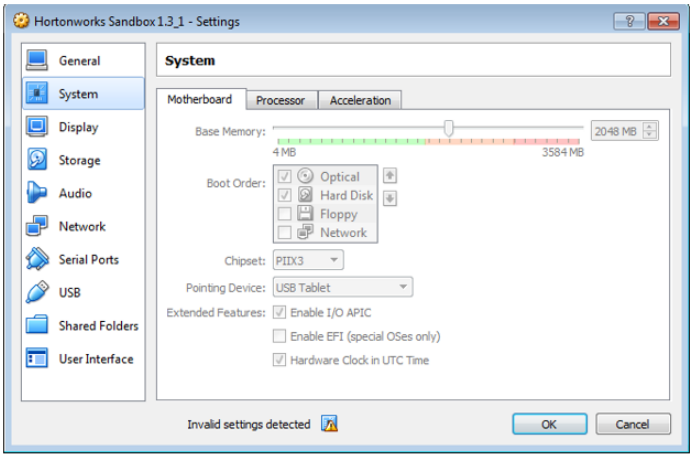

Fig 5 Motherboard, processor acceleration

After configuring and starting services Hadoop components can be accessed using a URL with IP configured

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 08 | October – December 2016
Paper is available at www.jrps.in | Email : info@jrps.in

Fig 6 URL with IP configured

Hive can be accessed with the URL as shown below:

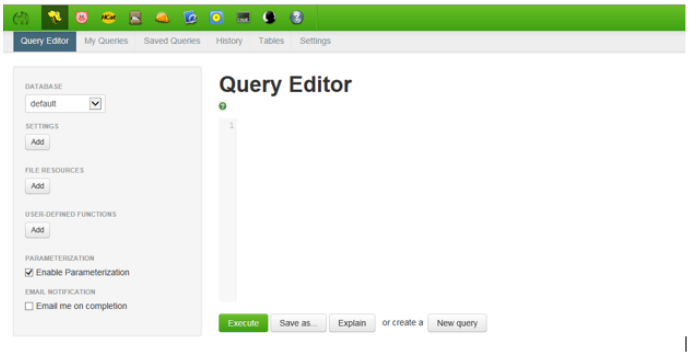

Fig 7 Hive can be accessed with the URL

A simple Web application can be designed and an encryption algorithm can be loosely coupled with application in order to produce encrypted data.

Fig 8 An encryption algorithm

Below shown files will be generated when each user enters data in the forms and press Submit Button. A file will be generate as soon as data submitted:

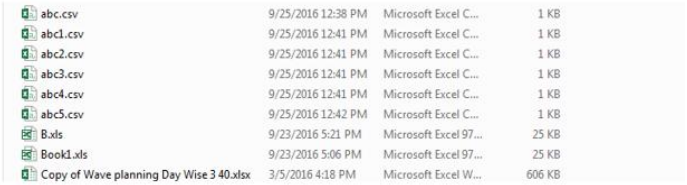

Fig 9 data submitted

Below is the content of the file in encrypted form

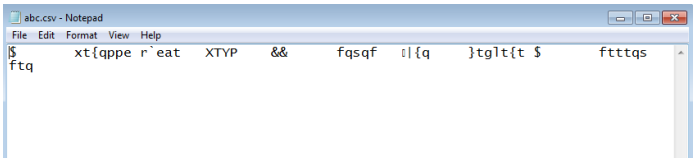

Fig 10 File in encrypted form

Once the files are generated on local machine it needs to be synchronized with Hive Data Warehouse. Shell script is written in order to synchronize the file with the hive tables.

All the files generated by the forms will be synchronized with Hive Data model using shell scripts written to synchronize the same.

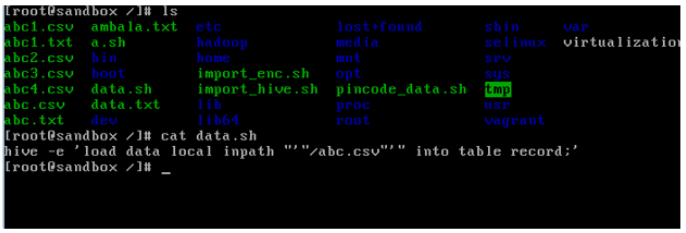

Fig 11 Synchronized with Hive Data model using shell scripts

After synchronizing data with Hive it is shown as below:

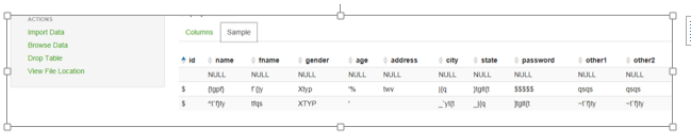

Fig 12 synchronizing data with Hive

Hence we see the data is synchronized with Hive server and it is available in file at client side which serves the purpose of encryption and cache mechanism at the client side.

[6]Conclusion

The main idea behind research is to integrate encryption into Hadoop based application and also apply client based cache concept which can handle the request from any application at client level, which will improve the performance of the application. Data entered by user in the application will be

encrypted by using XoR based encryption algorithm which will apply a key with the data to get encrypted ciphers.Encryption into a single operation makes it feasible for multiple Hadoop clusters and can be replicated with low cost across all clusters. By integrating encryption with a cache based file performance of the application can be improved as the requests will be handled at the client side only. Files generated will also be encrypted and will not be understandable to the normal users.This research typically do not change encrypted bit streams themselves but change way encryption bits are obtained. The integration of encryption and cache concept allows exploiting the client level access and increase the performance without compromising on the security of the data. As of now the research limit itself to the strings type data and it can be enhanced further on different types of data types and different type of applications across various industries.

Reference

[1] S.Vikram Phaneendra & E.Madhusudhan Reddy "Big Data-solutions for RDBMS problems- A survey" In 12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19{23 2013).

[2] Kiran kumara Reddi & Dnvsl Indira "Different Technique to Transfer Big Data : survey" IEEE Transactions on 52(8) (Aug.2013) 2348 { 2355}

[3] Jimmy Lin "MapReduce Is Good Enough?" The control project. IEEE Computer 32 (2013).

[4] Umasri.M.L, Shyamalagowri.D ,Suresh Kumar.S "Mining Big Data:- Current status and forecast to the future" Volume 4, Issue 1, January 2014 ISSN: 2277 128X [5] Albert Bifet "Mining Big Data In Real Time" Informatica 37 (2013) 15–20 DEC 2012

[6] Bernice Purcell "The emergence of "big data" technology and analytics" Journal of Technology Research 2013.

[7] Sameer Agarwal†, Barzan MozafariX, Aurojit Panda†, Henry Milner†, Samuel MaddenX, Ion Stoica "BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data" Copyright © 2013ì ACM 978-1-4503-1994 2/13/04

[8] Yingyi Bu _ Bill Howe _ Magdalena Balazinska _ Michael D. Ernst "The HaLoop Approach to Large-Scale Iterative Data Analysis" VLDB 2010 paper "HaLoop: Efficient Iterative Data Processing on Large Clusters.

[9] Shadi Ibrahim ⋆ _ Hai Jin _ Lu Lu "Handling Partitioning Skew in MapReduce using LEEN" ACM 51 (2008) 107–113

[10] Kenn Slagter · Ching-Hsien Hsu "An improved partitioning mechanism for optimizing massive data analysis using MapReduce" Published online: 11 April 2013

[11] Ahmed Eldawy, Mohamed F. Mokbel "A Demonstration of SpatialHadoop:An Efficient MapReduce Framework for Spatial Data" Proceedings of the VLDB Endowment, Vol. 6, No. 12 Copyright 2013 VLDB Endowment 21508097/13/10.

[12] Jeffrey Dean and Sanjay Ghemawat "MapReduce: Simplified Data Processing on Large Clusters" OSDI 2010

[13] Niketan Pansare1, Vinayak Borkar2, Chris Jermaine1, Tyson Condie "Online Aggregation for Large MapReduce Jobs" August 29September 3, 2011, Seattle, WA Copyright 2011 VLDB Endowment, ACM

[14] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein "Online Aggregation and Continuous Query support in MapReduce" SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-00322/10/06.

[15] Jonathan Paul Olmsted "Scaling at Scale: Ideal Point Estimation with 'Big-Data" Princeton Institute for Computational Science and Engineering 2014.

[16] Jonathan Stuart Ward and Adam Barker "Undefined By Data: A Survey of Big Data Definitions" Stamford, CT: Gartner, 2012.

[17] Balaji Palanisamy, Member, IEEE, Aameek Singh, Member, IEEE Ling Liu, Senior Member, IEEE" Cost-effective Resource Provisioning for MapReduce in a Cloud"gartner report 2010, 25

[18] Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N " Analysis of Bidgata using Apache Hadoop and Map Reduce" Volume 4, Issue 5, May 2014" 27

[19] Kyong-Ha Lee Hyunsik Choi "Parallel Data Processing with MapReduce: A Survey" SIGMOD Record, December 2011 (Vol. 40, No. 4) [20] Chen He Ying Lu David Swanson "Matchmaking: A New MapReduce Scheduling" in 10th IEEE International Conference on Computer and Information Technology (CIT'10), pp. 2736–2743, 2010