Hadoop: Bitcoin-BlockChain - A New Era Needed In Distributed Computing

Lakshmana Kumar Yeddu
Freelancer
Airoli, Navi Mumbai,
Maharashtra, India.

Shireesha Yeddu
Assistant Professor
Vivekanand Education
Society's Institute of Technology,
Chembur, Maharashtra, India.

ABSTRACT

Today, with the use of Internet, a huge volume of data been generated in the form of transactions, logs etc. As assessed, 90% of total volume of data generated since evaluation of Computers is from last 3 years only. It's because of advancements in Data storage, global connectivity with Internet high speed, mobile applications usage and IoT. BigData Technologies aims at processing the BigData for deriving trend analysis and business usage from its BigData information. This paper highlights some of the security concerns that Hadoop implemented in its current version and need for some of the enhancements along with a new methodology such as Electronic Currency (BitCoin) and BlockChain functionality. And also emphasises on why and how BitCoin and BlockChain can fit in Hadoop Eco-Systems and their possible advantages and disadvantages. Especially, in validating and authorizing business transactions with some mathematical cryptographic techniques like hashcode with the help of BlockChain Miners.

Keywords

Hadoop Security, Hadoop BitCoin, Hadoop Block Chain, Hadoop Miners, Hadoop EcoSystems, Hadoop Cryptography, Hadoop Block Size

1. INTRODUCTION

Hadoop [1] [2] is a parallel distributed computing framework by Apache for addressing BigData issues. BigData refers to huge **volume** (scale of data) of data with high **velocity** (speed of data), **veracity** (Uncertainty of data), and **variety** (analysis of streaming data). In order to query or retrieve any analysis on existing BigData, it was difficult to run the query and retrieve results in shorter duration for making quick business decisions, with the existing traditional RDBMS. The reason being, data may be unstructured, semi-structured or structured. Most of the RDBMS systems are considering structured data only. Hence Hadoop was introduced to reduce the time required to query the data of various data formats. Following are the Hadoop System Basics.

- **Block-Size:** Traditional Operating Systems or Databases uses a 4KB data size shown in Figure 1(a). For Example in Oracle, a 100MB file => 25600 x 4KB, where 25600 are number of blocks, each with 4 KB size. Whereas the same 100MB file is split in Hadoop uses only two blocks, 64 MB and 36 MB block-sizes shown in Figure 1 (b). With the less in number of blocks, helps in reduce time in many reads and write operations. Hence time is reduced.

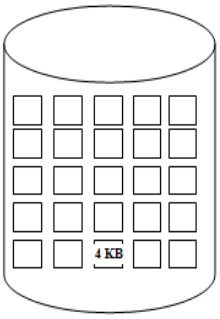

Fig 1 (a): Oracle Physical Drive – 4KB Each Block

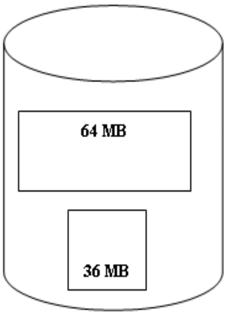

Fig 1 (b): Hadoop 1.x Block Sizes - 64 MB

The typical popular system and the block sizes are given below in Table 1.

Table 1 Typical Block Sizes of Popular Systems

Systems	Block Size
Oracle	4 KB
Teradata	64 KB
Netezza	128 KB
Hadoop	64 MB or 128 MB

- **HDFS:** Hadoop Distributed File System [1][2][3][4], used for storing huge volume of data, split in multiple data nodes in a cluster which uses physical hard disk in the nodes. It is a layer on top of existing operating system to handle data storage with configured block size either 64MB or 128MB.
- **Execution:** Hadoop initiates execution of a client request at data nodes, where the data is physically available split in blocks. The data may be in 1 to n nodes. Once the results are retrieved from each execution of DataNodes, it aggregates the final results and shares final results with end users. Whole process is done through Hadoop components called NameNode, DataNode , JobTracker and TaskTracker illustrated in Figure 1(c).
- **Hadoop Daemon Components:** The main components of Hadoop [1][2][3][4][5] are as follows:
    - ***NameNode****:* The Name Node is a master node; it is responsible for storing the Meta data of files and directories. The NameNode will have all the blocks information of a splited file in a cluster.

- ***DataNode:*** It is a slave node that has actual data. It reports to NameNode in a regular interval. Also know as *Heart Beats*.
- ***SecondaryNameNode:*** It periodically merges changes in the NameNode with the edit logs (transactions). It takes the backup of Metadata information frequently which can be used for any failure of name node to restore the data (blocks information).

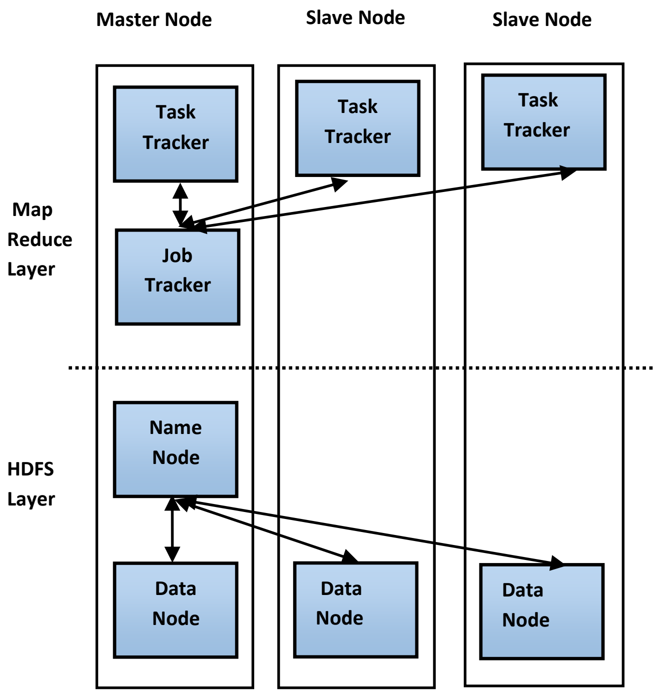

Fig 1 (c): Hadoop High-Level Architecture

- ***JobTracker:*** It is an assistant to NameNode for submitting and tracking MapReduce jobs and it assigns tasks to different task trackers on a data node where the actual data lives in.
- ***TaskTracker:*** It runs on a DataNode and manages the execution of individual tasks.
- ***ResourceManager***: It was introduced version 2.x onwards. It manages resources and schedules applications running on top of YARN.
- ***NodeManager:*** It runs on slave, DataNode machines, and is responsible for launching the application's containers, monitoring their resource usage (CPU, memory, disk, network) and reporting these to the ResourceManager.
- ***Application Master:*** Runs at slave node, responsible for tracking and management of assigned jobs. Each application master is assigned for a specific job.
- ***JobHistoryServer:*** It maintains history information about MapReduce jobs once it terminates.

2. HADOOP ECOSYSTEMS

An Eco System is independent software that runs on top of HDFS and MapReduce shown in Figure 2. Following are some of the Hadoop Eco System Components [1][2][3][4].

- ***MapReduce:*** MapReduce [6] is a batch job, it has mapper task and reduce task. Most of the Eco system use MapReduce as a backend logic for any of the queries or aggregations.
- ***Hive:*** Hive [7] is a data warehouse environment in Hadoop framework. Here the data can be stored in structured format. HQL (Hive Query Language) used to manage and process data. It can process structured data – XML, JSON and URL's (Web logs).
- ***Pig:*** Pig [7] is a data flow language (piping) in Hadoop. Data flow is a collection of pipes (Pipe is an operation). "Pig Latin" language is used to process data.
- ***Sqoop:*** It has two tools – sqoop import and sqoop export. It is to import or export data from RDBMS to Hadoop or vice versa.
- ***Flume:*** It is used to import streaming data into Hadoop.
- ***Oozie:*** It is used to define workflows and schedule the workflows.
- ***Zookeeper:*** It is used to manage collections and logs between different distributed applications.
- ***HBase:*** It is a NOSQL (Not only SQL), means it is a columnar database.
- ***Kafka:*** It is a messaging service in Hadoop used to integrated Hadoop and non Hadoop systems.

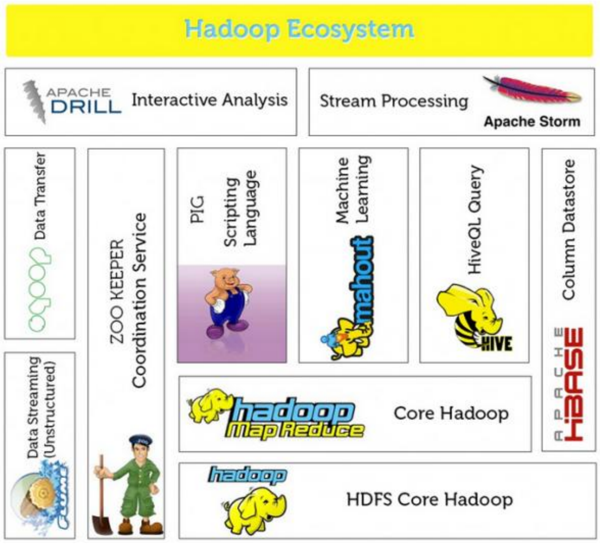

Fig 2: Hadoop EcoSystems

3. ADVANCEMENTS IN BIGDATA TECHNOLOGIES

Up recent, Apache's Spark and Flink got included into BigData systems to support structure data processing and distributed streaming. These are In-Memory processing engines where as Hadoop uses HDFS for read and write operations. Hence, these systems give much faster results.

- ***Spark:*** Apache Spark [8] is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing. MLib for machine learning. GraphX for graph processing, and Spark Streaming to perform micro batching.

- ***Flink***: Apache Flink [9] is a distributed streaming dataflow engine that aims to bridge the gap between MapReduce -like systems and shared-nothing parallel database systems.

4. SECURITY IMPLEMENTATION IN HADOOP

During initial use cases of Hadoop, it was aimed at large amounts of public web data, and its processing time using parallel distribution, confidentiality was not an issue. It was always assumed that cluster would consist of cooperating, trusted machines mostly commodity hardware with rack awareness as to distribute the data and select appropriate data nodes for execution purpose, by trusted users in a trusted environment.

4.1 Common Security Concerns

This section highlights the common security concerns with respect to Hadoop [3][5][6][10] to be addressed in near future are given below.

- To enforce authentication for users and applications on all types of clients on web consoles and processes.
- To ensure that real services not impersonated by unauthorized processes.
- To impose access control mechanism based on policies and user credentials
- Integration of existing security system with Hadoop
- To make sure standard Encryption algorithms to use for encrypting data in transit and at storage.
- Need a method to tracking and audit the events
- Need standard approaches to protect Hadoop cluster networks

4.2 Existing Security Implementation

This section describes the existing security implementations in Hadoop [3][5][10] discussed as follows:

- Hadoop uses SSH (Secured Shell) as a basic authentication for users to run on any jobs in Hadoop. Example, **#ssh-keygen** is used to generate public / private RSA (Crypto System) key pair. This applicable for all the Eco systems to interact with Hadoop.

- Apache Accumulo provides mechanisms for adding additional security when using Hadoop.

- Other open source projects, such as Knox Gateway (contributed by Horton Works) and Project Rhino (contributed by Intel) promise that big changes are coming to Hadoop itself.

- Keberos RPC Mutual Connection is used to process and authenticate users

- HTTP Web Console authentication, a pluggable authentication like HTTP SPNEGO is used in any of the WebPages.

- ACLs for HDFS file Permissions, are enforced

- Delegation Tokens are used for subsequent authorization after successful initial authorization. Keeping authorization information mostly in Context Object for any of the subsequent or intermediate transactions.

- Block Access Tokens are used to access control decision based on HDFS file permissions.

- Job Tokens are used for making access control checks simpler while initiating the Tasks.

- Network Encryption - Connections utilizing SASL can be configured to use a Quality of Protection (QoP) of confidential, enforcing encryption at the network level – this includes connections using Kerberos RPC and subsequent authentication using delegation tokens. Web consoles and MapReduce shuffle operations can be encrypted by configuring them to use SSL. Finally, HDFS File Transfer can also be configured for encryption

4.3 Vendor Specific Complements

This section focuses on vendor specific complements to Hadoop provided by third party to ensure security for big data.

- Data Encryption: Currently, data is not encrypted at HDFS. Hadoop clusters are forced to use third-party tools for implementing HDFS disk-level encryption [5][10], or security-enhanced Hadoop distributions.

- Kerberos Approach – The organizations utilizing other approaches not involving Kerberos, meaning setting up a custom authentication system in the enterprise.

- Limited Authorization Capabilities – Though Access Control Lists (ACLs) [5][10] can be configured, this may not be enough for every organization. Many organizations preferring XACML and Attribute-Based Access Control. This can be achieved using Accumulo.

- Complexity of the Security Model and Configuration. There are a number of data flows involved in Hadoop authentication – Kerberos RPC authentication for applications and Hadoop Services, HTTP SPNEGO authentication for web consoles, and the use of delegation tokens, block tokens, and job tokens. For network encryption, there are also three encryption mechanisms that must be configured – Quality of Protection for SASL mechanisms, and SSL for web consoles, HDFS Data Transfer Encryption. All of these settings need to be separately configured – and it is easy to make mistakes.

- Big Changes Coming with Project Rhino, a small help from Intel for making Data Encrypted, Token-Based Authentication & Unified Authorization Framework and Improved Security in HBase - a cell-level authorization to HBase – something that Apache Accumulo has but HBase does not.

5. BITCOIN

Bitcoin is a digital currency to facilitate the exchange of goods and services by offering a commonly accepted good. It relies on a network of volunteers that collectively implement a replicated ledger and verify transactions [11][12]. It is anticipated to have Bitcoin as common currency for trading across the globe shown Figure 5.1.

Fig 5.1: Bitcoin

6. BLOCKCHAIN

Blockchain [13] technology in Figure 6.1 (a) depicts a bag of Lego or bricks. From the bag, you can take out different bricks and put them together in different ways to create different results. The blockchain is a decentralized ledger, or list, of all transactions across a peer-to-peer network [14].

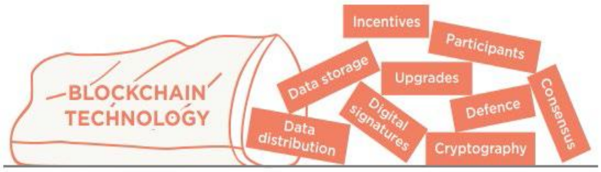

Fig 6.1(a): BlockChain

- It deals with Transactions. Transaction details are replicated in a number of systems called open ledger(distributed, replicated in multiple nodes in a cluster environment)
- It uses Digital signatures and Cryptography mathematical models for creating Hash codes for all historical transactions. It is highly impossible to break or hack any transaction as it is monitored by all peers in the system. Unless and until authorized by peers the transaction won't be committed.
- Transactions can be read or written by any peer in the cluster.

Special nodes in the Cluster acts as Miners, whose job is to authenticate, validate the transaction with mathematical models to secure. It works on publish and subscribe. The first miner which does the mathematical cryptographic calculations and authorize the transaction will get rewarded with BitCoin called "Proof of Work" illustrated in Figure 6.1 (b) [10] [15].

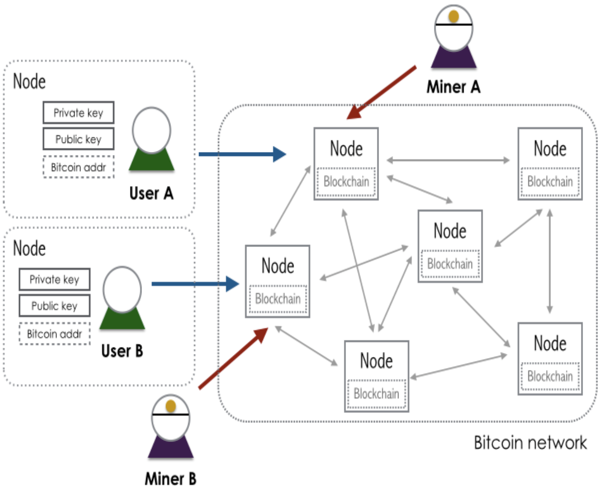

Fig 6.1 (b): BitCoin – Proof of Work

Miners: Miners are special nodes that race the transactions for validation and giving immediate response to all other peers to confirm on whether the transaction and authentication are valid or not. It uses Cryptographic mathematical models to generate the required *Hash Codes* for the transactions based on previous transactions and link the historical related transactions together. By confirming this "proof of work" it is rewarded.

6.1 BlockChain Hashing

Bitcoin uses hash code proof of work algorithm shown in Figure 6.1 (c). Here, the transactions collected as blocks. Each block contains latest balances of user's in the network. A blockchain is formed with all the transactions (blocks) from genesis block. A block header contains fields: Version, hash of the previous block (hashPrevBlock), root hash of the merkle tree of all transactions in the block (hashMerkleBlock), Timestamp, Nonce. The hash code algorithm repeatedly hashes the block header and increments the counter with nonce [16][17].

- ***Version:*** This field specifies Block version number
- ***hashPrevBlock:*** This field contains hash of the previous block header
- ***hashMerkleRoot:*** This field holds hash based on all of the transactions in the block.
- ***Timestamp:*** This field consists of time when the block was found by the miners.
- ***Nonce:*** This field value initially set to 0 and is incremented for each hash.
- ***Bits (target):*** This filed corresponds to the difficulty of finding a new block using floating point value.

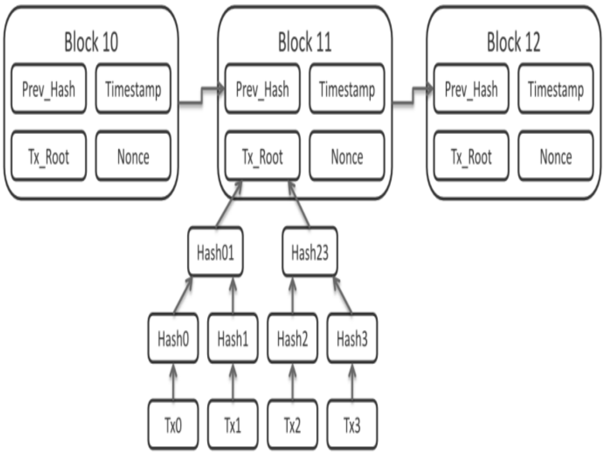

Fig 6.1 (c): BlockChain Hashing

Using hash code algorithm, all the block headers except the transaction data is hashed to forms the block hash. And this generated block hash is used as proof that the other parts of the header have not been altered, and later this block hash is used as a reference by the next block in blockchain.

7. NEED FOR BLOCKCHAIN

This part of the paper specifies the need for Block Chain in Hadoop distributed computing system.

- ***Inter Cluster Communication:*** As of now, Hadoop is mainly concentrated on processing the data and give faster response on a huge volume of the data within a cluster. When it comes to inter cluster communication, it is difficult to share the block information from one cluster to other cluster. It requires **$hadoop distcp** command to copy the files from one cluster to other and then start processing the data. It is proven that Hadoop plays a major role in processing large volumes of data. It requires a trust between clusters either in public or private open ledgers as proposing. This can be achieved with some Access control lists.

- ***Data Storage:*** As Hadoop is already having HDFS, this can be utilized for data distribution across the globe, so that many clusters co-exist. Currently one NameNode maintains Meta data about the files. If the NameNode in a cluster fails then there is no way to execute jobs without having the file information. This has to be enhanced, similar to BlockChain where there is no single point of failure. Resulting in high availability. Hadoop 2.x is exceptional, as it is having high availability with NameNode restoration process.

- ***Speculative Execution:*** As the current Hadoop uses mechanism of delegating the work to any replicated data nodes where physical data exists, in case the previously delegated Task tracker fails to execute the job as expected or not in permissible time. This can be thought of Miners functionality where Miners are racing for giving cryptographic mathematical model authentication of transaction, so that others peers can agree to the transaction and authorize it. But, this speculative execution is restricted to one Job related Task. To achieve this Kafka along with Zookeeper and Flume can be used to read live data streaming of transactions and process it with proper authentication.

- ***Open Ledger:*** Though Hadoop History server maintains logs, it is not related to transactions ACID properties (Atomicity, Consistency, Isolation and Durability). Hadoop cluster in general is private. Currently it is mostly executing the batch jobs for processing the queries in analytics. Considering future aspects of information sharing and authenticating there needs an Open Ledger where all peers will have replicated information of all transactions. Considering the sensitivity of the business data especially when there lies banking related information, it should use existing Mahout (Machine library) for effectively using the mathematical models for securing the transactions. By making use of Open Ledger, Hadoop can maintain transactional state of any object or resource. This can be thought of Enterprise Java Beans (EJBs) or POJOs.

- ***IoT Security:*** Internet of Things is a way of communicating the electronic devices(internet enabled).These devices produce enormous data. The security is vital for data sharing between centralized server and the sensors. Data can be secured with BlockChain. **Filament** is a start-up company that is already providing services, for decentralized IoT software stack with the help of unique identities in Public Ledger. As proposed Open Ledger within Hadoop would help any security issues relating to IoT either Private or Public cluster way.***For Example:*** While driving a car, for any malfunction in it, it can alert the Driver with nearest service station and also can alert the service station staff for any possible rescue to fix any issues. Though the data from IoT is pumped into Cloud, there needs a data processing programs something like Hadoop to process it and alert appropriate accordingly.

- ***Cryptographic Authentication:*** Though security is maintained in Hadoop to some extent either internally or using third party framework, it requires a strong mechanism to validate and allow any manipulation of sensitive data. Though, Hadoop is intended to write once and read many times for batch processing, there needs a sequential reading of each block in parallel requires a security. This shall avoid any manipulation of data by malware and can be declined by the peers if it seems unauthorized transaction is been initiated by any node, in case if it's compromised its resources to some hackers. If BlockChain security is maintained, there are least chances that hackers manipulate any of the data by reading through its previous transactions.

For Example: If node A transferred $4 amount to node B out of $10, then the balance that node A would have is $6. If next time node A initiates transfer of $7 to node C, then this is invalid transaction as node A does not possess required funds to transfer depicted in Figure 7.1. The peers will reject the transaction based on its previous transaction. The Table 2 shows the open ledger transactions from Node A to Node B and Node A to Node C.

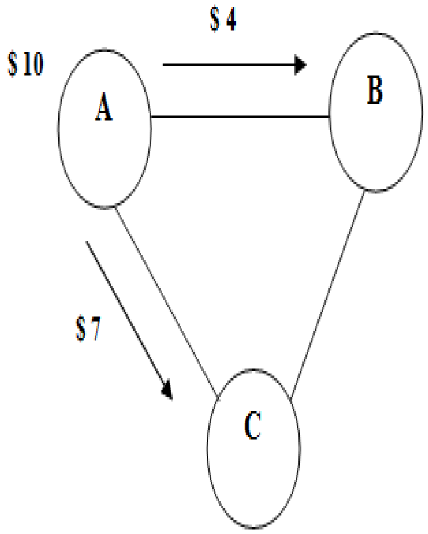

Fig 7.1: BlockChain – Cryptography Authentication

Table 2 Open Ledger

Node A - Node B ----------------> $ 4 Valid
Node A - Node C----------------->$ 7 Invalid

- ***Avoid Delays in completing the transactions:*** In present time money transfer from one country to other country is almost taking larger amounts of time using trusted third party partners, in current internet world. Other than the money any other transaction, example email, messaging, sharing any other information is almost shared in no time. But, for money there needed a third party authors to authorize and would take time to process the transaction to complete. In present world it is needed a faster response. This can be achieved with the help of parallel distributed systems and BlockChain.

8. BLOCKCHAIN IN HADOOP

The following figure 8 shows where the Block Chain fits in Hadoop Eco System. HBChain (Hadoop Block Chain) can fit in Hadoop on top of Basic Core, MapReduce, Flume, Kafka, and Zookeeper. It can leverage existing framework for both processing and securing the transactions.

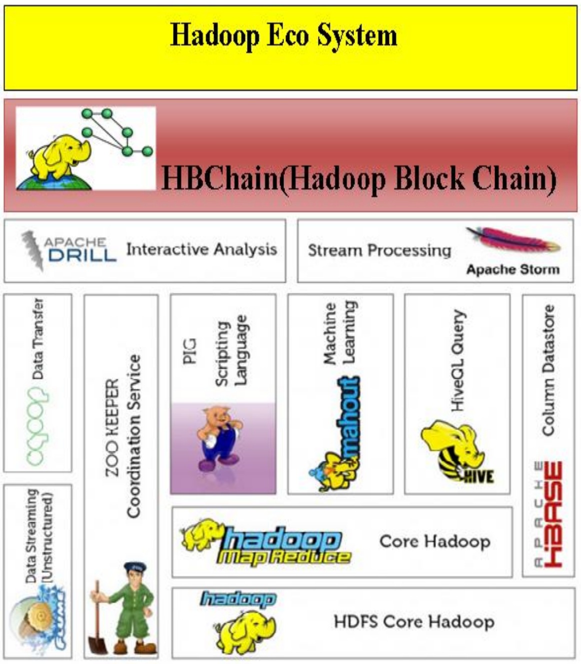

Fig 8: BlockChain in Hadoop

Advantages

- Imposing security while making any transactions
- Inter Cluster communication is possible
- Racing between Miners for validating and authorizing transactions
- Integrating global economy with electronic currency for any kind of trading- goods, materials, Money etc.
- Bringing server side components functions like EJBs or POJOs into Hadoop for storing historical session management.
- Inter Cluster data distribution and usage with the help of some ACLs
- ACLs for proper encryption of data while transit and at rest
- Scope for introducing new third party players to provide services as Miners and help community
- Reduction of third party trusted fees for any financial transactions and reduction of time.

Disadvantages:

- With the Open Ledger some of the confidential data is published.
- Revamp Hadoop basic architecture for inter cluster communication, would result in some cost .and time

9. CONCLUSION

In this paper, Hadoop Eco systems were analyzed for some of the security concerns and needed methodology of BlockChain in achieving the said goals. Current Hadoop framework does not have BlockChain functionality but hoping to implement this in future releases, for effective inter-cluster and authorization services. This would help in bringing new players (nodes) as Miners that would help in achieving faster response by paying marginal fees, compared very less to Third party trusted players. Thus increase revenue by paying less to the transactions authorized by Third Party. Two aspects covered, one security improvement and the other is BlockChain a way to authorize transaction with the help of Hashing Technique. This would help to process huge amounts of data (Peta Bytes, or Exa Bytes or even more). Processing of data would take less time in Hadoop. Since, Hadoop BlockChain (HBChain) is dealing with streaming data along with Historical data, it can be thought of using Apache's Spark and Flink for faster results as they are In-Memory processing engines along with other Hadoop Ecosystems like Kafka, Flume, Oozie with core MapReduce or Pig Latin Scripts etc This HBCoin can be used in Banking where there are many online transactions would happen and to process any historical data. This also can be extended to IoT, Military, Aero space, Logistics, Social Networking etc.

10. REFERENCES

[1] Poonam S.Patil, Rajesh. N. Phursule, Survey Paper on Big Data Processing and HadoopComponents, IJSR, Volume 3, Issue 10, pp. 585-590,October 2010, ISSN (Online): 2319-7064

[2] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, The Hadoop Distributed File System, pp. 1-10, IEEE 978-1-4244-7153-9/10, 2013.

[3] Singh Arpita Jitendrakumar, Security Issues in Big Data: In Context with Hadoop, IJIERE, Volume 2, Issue 3, 2015, pp. 127-130, ISSN: 2394 - 3343.

[4] Dr. E. Laxmi Lydia, Dr. M.Ben Swarup, Analysis of Big data through Hadoop Ecosystem Components like Flume, MapReduce, Pig and Hive, IJCSE, Vol. 5 No.01 Jan 2016, pp. 21-29, ISSN : 2319-7323.

[5] Vijaykumar Patil, Prof. Venkateshan N, Review on Big Data Security in Hadoop,IJECS, Volume 3, Issue 12, December, 2014, pp. 9507-9509, ISSN:2319-7242.

[6] Nivethitha Somu, A. Gangaa and V. S. Shankar Sriram, Authentication Service in Hadoopusing One Time Pad, IJST,Vol 7(S4), pp. 56–62, April 2014,ISSN (Print) : 0974-6846.

[7] Sanjeev Dhawan, Sanjay Rathee, Big Data Analytics using Hadoop Components like Pig and Hive, AIJRSTEM, pp.88-93, 2013, ISSN (Online): 2328-3580.

[8] Apache Spark, [Online]. Available https://en.wikipedia.org/wiki/Apache_Spark [Accessed: October 10,2016]

[9] Apache Flink, [Online] Available https://en.wikipedia.org/wiki/Apache_Flink [Accessed: October 10, 2016]

[10] Hadoop Security Model [online] Available: https://www.infoq.com/articles/HadoopSecurityModel/

[11] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, Edward W. Felten,SoK:

Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, IEEE, pp. 104-121, 2015.

[12] Christian Decker, Roger Wattenhofery, Information Propagation in the Bitcoin Network, IEEE, pp. 1-10, 978-1-4799-0521-8/13, 2013

[13] Michael Crosby, Nachiappan, Pradhan Pattanayak,Sanjeev Verma, Vignesh Kalyanaraman, BlockChain Technology Beyond Bitcoin, Sutardja Center for Entrepreneurship & Technology Technical Report, pp. 1-35, October 2015.

[14] A gentle introduction to blockchain technology [online] Available, https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/

[15] Bitcoin-part-one [online] Available, http://tech.eu/features/808/bitcoin-part-one/

[16] Block hashing algorithm [online] Available, https://en.bitcoin.it/wiki/Block_hashing_algorithm

[17] Bitcoin beta [online] Available, http://bitcoin.stackexchange.com/questions/12427/can-someone-explain-how-the-bitcoin-blockchain-works