Staleness Data Retrieval from Multi-Tiered and Replicated Clusters

G. Sekar, R. Anandhi

Abstract— The tsunami of data spans over the network is now coined with the term "Big Data". The problems faced by Big Data during data acquisition are Privacy, Security, Complexity, Heterogeneity, Storage etc. Which data to keep and which data to discard is also a big question regarding Big Data since some data are not stored in structured format. In Social networking like Facebook, twitter, Whatsapp etc, the data communication takes place only in unstructured format and so later the conversion and interpretation of those data into presentation of results is a very clear bottleneck [1]. So Big Data analysis now pulls every aspect of our usage like retail marketing, mobile services, financial services, life sciences, production and sales. So the real challenge is to get a maximum out of the data available already and predicts the nature of data to be collected as per future need. Mostly read transactions outlay write transactions over the storage of data. So this paper proposes a new methodology "Read Polling Algorithm" to perform the read operation effectively and efficiently for retrieving correct data to the intended user..

Keywords— Big Data, Data Transactions, Heterogeneity problems, Read pollin.

I. INTRODUCTION

The data that constitutes big data stores can come from sources that include desktop, social media, web sites, scientific experiments, mobile apps, sensors and other devices in the Internet Of Things (IoT) etc [2]. The concept of big data arrives with a related set of components that facilitates the organizations to solve a number of business problems and to place the data for practical use. The components of Big data includes related skill sets, technologies needed for big data projects, the analytics applied to the data, the IT infrastructure needed to support big data, and the actual use cases that make sense for big data. If no proper analysis is carried out, the data so far collected will only stored as a bunch of data with limited use. If we apply effective analytics to big data, organizations really enjoy new dimensioned benefits like good turnover, improved sales, satisfied customer service, greater efficiency, and an edge over revenue in business markets. [3]. Data analytics includes the extensive examination of data sets with necessary insights; provide conclusions about what they enclosed like trends and predictions about their future proposal. By analyzing data, organizations can make better-informed business decisions such as how, where and when to do a campaign on marketing and to introduce a new product. Analytics may be referred as basic business intelligence applications or more advanced predictive analytics used by scientific and research firms. One of the advanced types of data analytics is data mining, where specialists is exposed to large data sets to identify relationships, patterns, and trends.

Data analytics may perform either extensive data analytics to notify the data patterns and relationship among them or confirmatory data analysis by applying statistical methods to check if an assumption about a data set is true or false [4]. Another flavor is quantitative data analysis about numerical data that has quantifiable variables and can be compared statistically and qualitative data analysis focusing on non-numerical data supporting formats like video, images, and text. Data storage provides an innovative storage infrastructure that is designed especially to store, manage and retrieve big data. Today data storage supports the storage and arrangement of extensive and big data in such a way to access simply and easily (ie) used by big data applications and services [5]. Data cluster will also able to flexibly scale as required. Direct Attached Storage (DAS) pools or Network Attached Storage (NAS) supports the massive big data storage design based on object format. The storage infrastructure is connected to computing server nodes that change fast process and retrieval of massive quantities of data. Moreover, the most massive data storage infrastructures have native support for large data analytics solutions like Hadoop, Cassandra and NoSQL [6].

Data processing relates to the utilization of data sets to handle the collection and reporting of data for business aspects and for requested users. Data mining refers to the process of inquiring massive data sets to seem for related information [7]. This event is an apt example of the axiom "looking for a needle in a haystack." Decision makers would like access to narrowed domain specific pieces of data from large data sets. They use data processing to uncover the items of data that may inform leadership and facilitate chart the course for a business. Data mining involves the utilization of various forms of software packages like analytics tools. Data mining tools are often machine controlled, or labor-intensive (people-centric), where individual workers send queries for data are archived in database [8]. Data mining means the operations including a set of compact search methods that return targeted and specific results. A data mining tool may account cumulative data for more years to find the performance of the system based upon certain parameter for a specific period of time. Hence data mining acts as a "handler" of that analysis and is used to provide required

Revised Manuscript Received on September 14, 2019.

G. Sekar, PG and Research Dept. of Computer Science, Dr. Ambedkar Govt. Arts College, Chennai-39,Tamil Nadu,India. (E-mail: gsekarg@yahoo.co.in,)

R. Anandhi, PG and Research Dept. of Computer Science and Applications, D.G. Vaishnav College, Chennai-106, Tamil Nadu,India. (E-mail: anandhi78@yahoo.com)

results and outsized data cluster is the asset. The following are the technologies that support the manipulation of voluminous data over network.

- **Hadoop**: One of the main components in Big Data terminology is Apache Hadoop. Hadoop is the collection of software frameworks and libraries to control and process large data from centralized server to a number of clients. Hadoop comes under the category of open source project from apache with a significant and constant improvement in performance over years. It simplifies and easily manages the processing and controlling large amount of structured data and unstructured data at an affordable rate [9].
- **MapReduce**: Google introduced MapReduce in order to create large amount of indexes for web search. MapReduce provides the basic software framework which will help in writing application for processing either structured or unstructured data over the internet with a rate of high fault tolerance [10]. MapReduce receives a query, segments that query into parts and assign the parts to be solved by multiple machines. The answer submitted by the multiple machines will be clubbed to form the result of the query and hence it is easy to maintain large data by dividing the data among various machines through the popular distributed query processing.
- **HDFS**: HDFS stands for Hadoop Distributed File System, a Java based file system used to handle any sort of data format in clusters controlled by distributed servers. There is no strict rule over the data to be stored (ie) either fully unstructured of purely structured. Turning of inputted data to meaningful data is in the hands of developer's code that handles HDFS. HDFS guarantees a reliable fault tolerant atmosphere using low cost hardware [11].
- **Hive**: Hive was first developed by Facebook and now it is announced as open source for certain time period. Hive acts as an interface between Hadoop and SQL. It is applied to form queries in SQL over clusters maintained by Hadoop using a native interface for SQL termed as HiveQ [12]. Apache Hive performs like a data warehouse to design adhoc queries, give summary about data and analyze large data sets stored in compatible file systems of Hadoop.
- **Pig:** Yahoo introduced Pig initially which was later emerged as open-source software. It provides an interface to submit query up on Hadoop data. But in addition to that, it maintains a script documentation to set Hadoop data accessible by business persons and software developers [13]. A high level programming platform is given by Apache Pig for developers to process, control and regularizes Big Data using user-defined procedures and programming skills.

II. METHODOLOGY

Database tuning aims to maximize use of resources to perform work rapidly and efficiently. Some types of replication techniques allow the concept of dynamic ownership of data. Here the right to update the replicated data moves from one node to other node in the cluster (ie) only one node owns the right for update. [14]. Dynamic ownership would be appropriate policy since business activities move across different time zones and when the data processing follows a work flow system across business subsystems supported by heterogeneous database servers. Push strategy is the procedure where the master server leads the write transactions upon its subordinates [15]. Queries about stored data must be responded within no time, update the data by different clients must be done with consistency, and reading sensitive parts of the data must be restricted from unauthorized users. The owner of the data must secure the data from inconsistency operated by clients accessing the data simultaneously. If programs accessing the data are written to support such consistency, this will be an add-on value to their complexity. Basic read operation is shown in Figure 1.

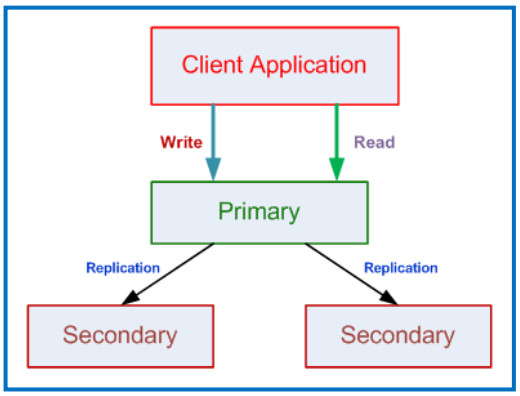

Figure-1: Basic Read Operation

The swim lane diagram in Figure-2 explains the steps carried out for performing a read operation in clusters.

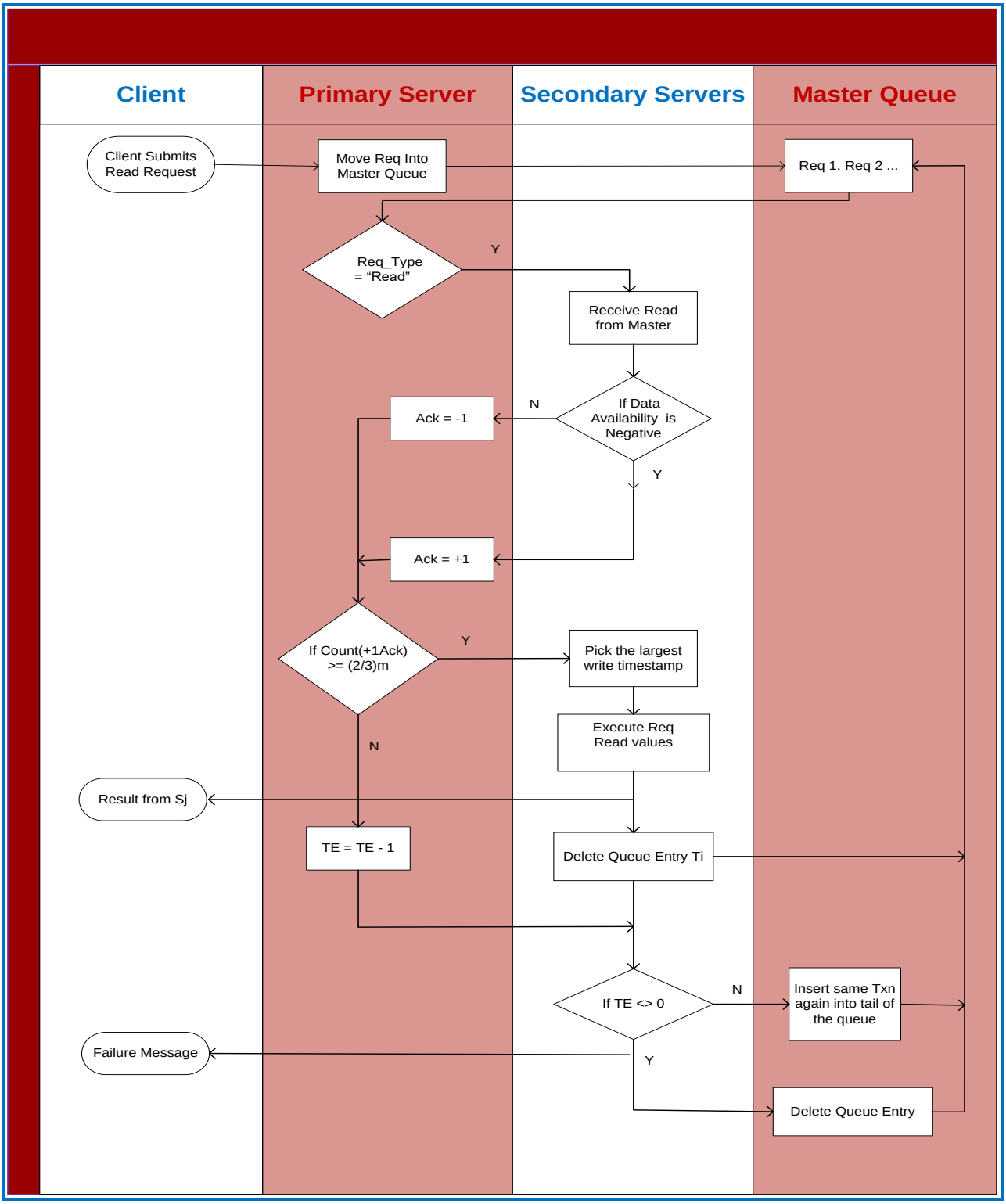

Figure-2: Read Operation in Read Polling Algorithm

In all file transactions involving huge amount of data, more care is given to write operation in order to reflect correct data on the data source. This is because the user expects that the read operation issued should retrieve accurate answer instead of approximate one and this can be guaranteed if and only if the write operation executed earlier should be done without any flaw. In all databases, usually read transactions dominate write (i.e.) queries will be more than other operations. So a database must be tuned to retrieve the requested data within no time. Hence this paper proposed a method called "Read Polling Algorithm" which will retrieve data from the source file more quickly.

It is evident that there is a head node for each cluster. The head node co-ordinates all the transactions routed to the cluster. Consider a read operation is given to a cluster of "m" nodes. The read transaction carries the data source details upon which the query has to be executed. The head node consults with the transaction synchronizer to know the statistics that in how many member nodes the write transaction is under progress upon the said database. Let "n" be the number given by the transaction synchronizer for the

statistics asked by the head node.

Step 1: Now head node comes to a conclusion to submit the read query to "m-n" nodes in order to preserve mutual exclusion property between read and write. The head node starts the acknowledgement session phase to get the consent from the member nodes by setting a fixed time duration to submit their response. After getting the read query from the head node, the member nodes will check whether the row lock bit status in its data source as "locked" or "unlocked". Locked is denoted by +1 while unlocked is by 0. If the requested row is unlocked, it denotes that the member node is eligible to participate in read operation. So it sends acknowledgement to its head node to confirm its registration. If the requested row is locked, it means the member node is not eligible to participate in read query. So it will not respond to its head node. As the acknowledgement session time outs, thereafter the head node will not accept any acknowledgement further from its member nodes and count the number of acknowledgements so far received (say "x"). The working of Step-1 is depicted diagrammatically in Figure-3.

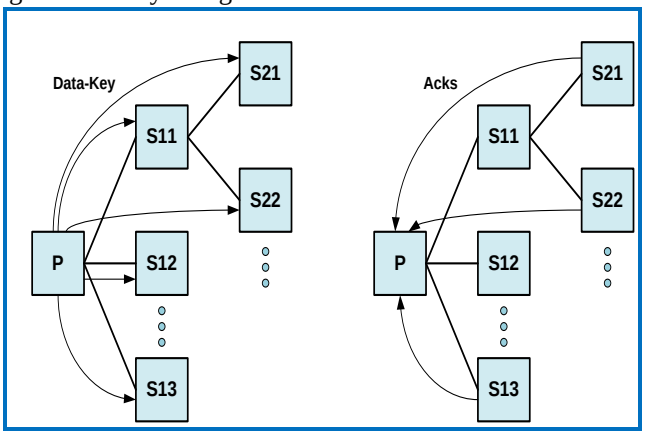

Figure-3: Read Registration Phase

Step 2: If x=0, it implies that no member node participates in the forthcoming read operation. So the head node will queue that read query in the waiting queue.

If x=1, it implies that only one member node is ready to participate in the read query and so the head node asks to give the value from that member node. Now the member node will lock the requested row(s) as +1, read the required column(s), built the read query and send to the head node as the result. If x>1, it implies that more than one member node is willing to respond to that read query. In order to tackle the competition among the ready member nodes, the head node asks them to submit the write timestamp for the requested row. The head node sorts all the write timestamp in non-increasing order and pick out the first (latest) value. The member node holding that selected value is the winner (Figure-4). Now the member node will lock the requested row(s) as +1, read the required column(s), built the query and send the result to the head node.

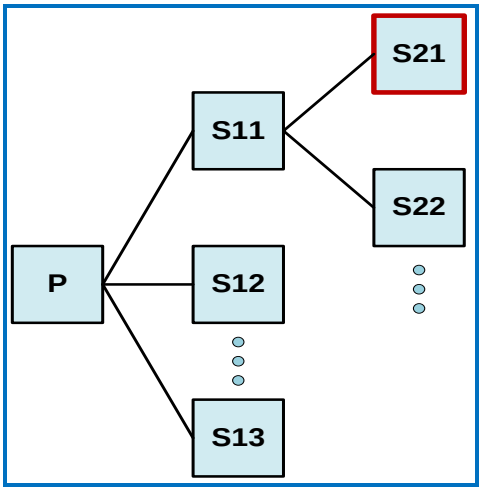

Figure-4: Server Selection Phase

Step 3: The head node acknowledges the result and the corresponding member node unlocks the row bit as 0 (Figure-5).

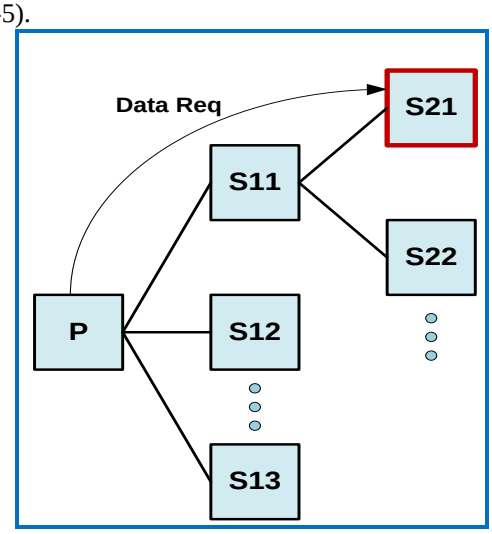

Figure-5: Read Phase

Since the read operation is executed after forming a membership among the member nodes by getting their permission for participation without forcing them, this algorithm is called as "Read Polling algorithm".

When a read request is invoked by a node i, a quorum Q of the corresponding row is selected. Nodes belonging to this quorum communicate to each other until a selection of latest value is found, forward the selected last value to the primary server, and then the acknowledgement is given by the primary to the respective replica node of the tree T of quorum Q.

$$Cost_{Read\_Qi} = (N - 1)\left(\sum Select\_cost_{j,i} + Fw\_cost_{i,j} + Ack\_cost_{j,i}\right)$$

N: Number of nodes in the quorum Q of the tree T.

$Cost_{Read\_Qi}$: Cost of the read initiated by the node i (primary server) over the quorum Q of the tree T.

$Select\_cost_{j,i}$: Selection cost of the latest value to be read from the node j to the node i.

$Fw\_cost_{i,j}$: Forward the selected value from the node i to the node j.

$Ack\_cost_{j,i}$: Acknowledgement cost from the node j to the node i.

III. RESULTS AND DISCUSSIONS

In order to study about the read performance by Read Polling algorithm, a set of transactions with 100% read transactions is given for execution at secondary servers 1, 2 and 4. The result is tabulated in table-1 and figure-6 shows the corresponding graph.

Table 1: 100% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

No. TXN	Run Time for Read Operation by Read Polling Algorithm (in seconds)			Run Time for Read Operation by Traditional Randomized (in seconds)		
	1 Secondary Server	2 Secondary Servers	4 Secondary Servers	1 Secondary Server	2 Secondary Servers	4 Secondary Servers
100	0.389	0.3901	0.4817	0.376	0.4325	0.4682
200	0.4542	0.5041	0.6017	0.4487	0.535	0.5454
500	0.6586	0.6914	0.7823	0.6796	0.697	0.7624
1000	0.9824	1.0197	1.0425	0.9903	1.0643	1.0904
2000	1.4639	1.5334	1.5919	1.5639	1.6569	1.6984
3000	2.2754	2.3114	2.3223	2.4714	2.5337	2.6106
5000	3.7871	3.8623	3.8673	3.9591	4.0498	4.082

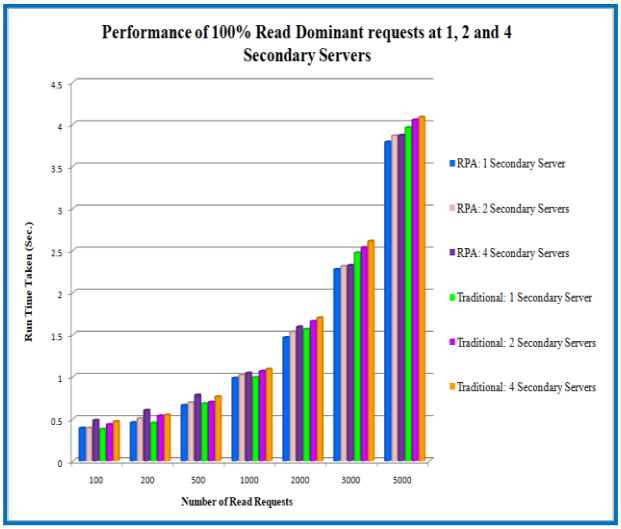

Figure-6: 100% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

Table 2: 70% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

No. TXN	Run Time for Read Operation by Read Polling Algorithm (in seconds)			Run Time for Read Operation by Traditional Randomized (in seconds)		
	1 Secondary Server	2 Secondary Servers	4 Secondary Servers	1 Secondary Server	2 Secondary Servers	4 Secondary Servers
100	0.2764	0.2835	0.3276	0.2723	0.2891	0.3346
200	0.4309	0.5306	0.6143	0.4258	0.5145	0.5442
500	0.6319	0.6816	0.7422	0.6407	0.7333	0.7909
1000	0.8503	0.9116	0.9478	0.8708	0.8933	0.9351
2000	1.3086	1.3506	1.4072	1.4153	1.4292	1.4563
3000	2.1973	2.2377	2.3324	2.2268	2.2451	2.3166
5000	3.6092	3.6913	3.7824	3.7183	3.8107	3.8737

A set of transactions with 70% read transactions is given for execution at secondary servers 1, 2 and 4. The result is tabulated in table-2 and figure-6 shows the related graph.

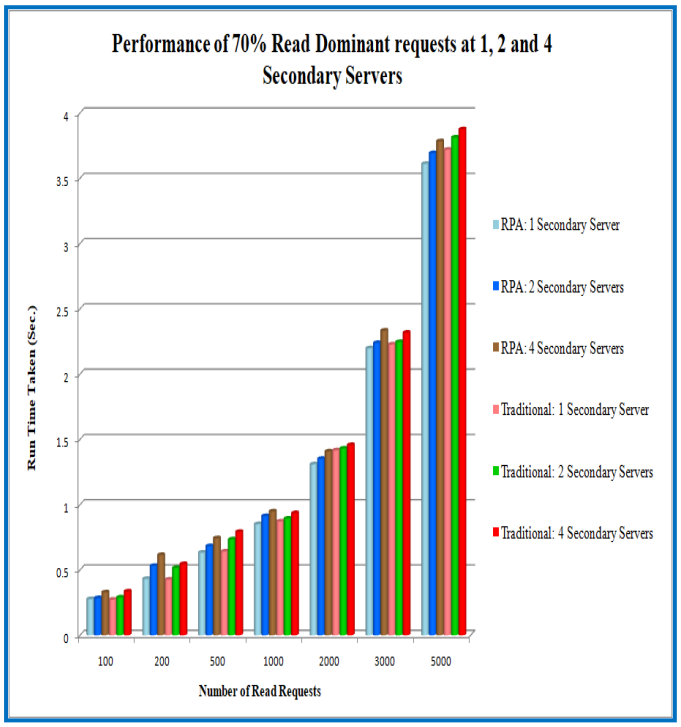

Figure-7: 70% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

A set of transactions with 50% read transactions is given for secondary servers 1, 2 and 4. The result is tabulated in table-3 and figure-8 shows the corresponding graph.

Table 3: 50% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

No. TXN	Run Time for Read Operation by Read Polling Algorithm (in seconds)			Run Time for Read Operation by Traditional Randomized (in seconds)		
	1 Secondary Server	2 Secondary Servers	4 Secondary Servers	1 Secondary Server	2 Secondary Servers	4 Secondary Servers
100	0.1478	0.1883	0.203	0.1401	0.2262	0.3061
200	0.2907	0.2944	0.3052	0.2865	0.3051	0.3416
500	0.5096	0.5362	0.6272	0.5129	0.5816	0.664
1000	0.6371	0.6427	0.7346	0.6484	0.6628	0.6883
2000	0.9972	1.0778	1.1329	1.023	1.1162	1.1398
3000	1.7845	1.8131	1.8927	1.9738	2.0362	2.0411
5000	2.7304	2.789	2.8335	2.9852	3.039	3.1377

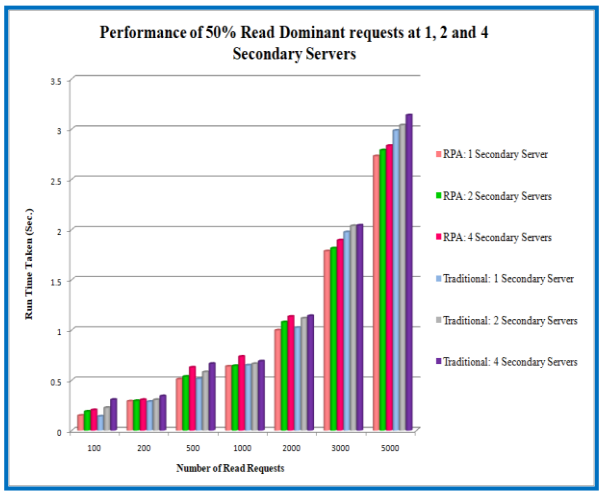

Figure-8: 50% Read Dominant Transactions at 1, 2 and 4 Secondary Servers

There are many read algorithms in practice for cloud databases to retrieve the values stored at the database like Randomized, Round-Robin etc. Randomized algorithm will choose a secondary server in random and route the read there. If read is possible, it will give the required data else the algorithm will have to try for next secondary server suitable for same read. If the first secondary server selected gives the data, the randomized will give a best performance of O (1) else it will have to try at all secondary servers giving a worst time complexity of O (n). But in the developed Read Polling algorithm, only the formation of read set will take some non-zero amount of time. But after the formation of read set servers, at least one server among them will yield a correct and updated data without fail. The comparison done between the two algorithms are given in table-4 and a graph is plotted accordingly in figure-9.

Table 4: Average time taken by two algorithms

Average Response Time	Number of Transaction	
	0 to 500	1000 to 5000
Read Polling Algorithm	0.473	1.961
Randomized Algorithm	0.4758	2.037

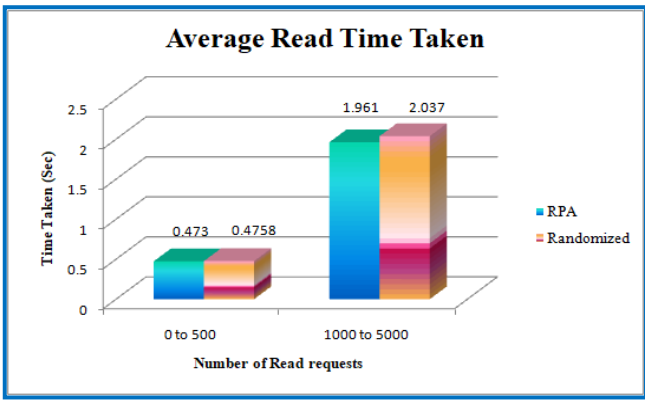

Figure-9: Randomized algorithm Vs Read Polling algorithm

IV, CONCLUSION

The concept of big data has been flourishing in near past as many organizations involved in collecting data from their business, apply analytics and capture statistics in order to improve their business. The data analytics improves speed and efficiency obviously in taking smart and adhoc decisions. Eventually the performance of the organization has been geared and tuned to run faster and remain agile than ever before with less cost of storage and thereby producing new products as per customer needs. The available data sets are so big in size and complex in nature so that traditional data processing tools are not capable of capturing, managing, manipulating and recognizing the data within a reasonable period. These data sets can include the formats like unstructured data, structured data and semi-structured data, each of which has its own significance. So this paper proposes a new algorithm to retrieve correct data within no time to satisfy user's requirements

REFERENCES

1. Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein "Online Aggregation and Continuous Query support in MapReduce" SIGMOD'10, June 6–11, 2010.
2. H.Herodotou,H.Lim,G.Luo,N.Borisov,L.Dong,F.B.Cetin, and S. Babu,"Starfish: A Selftuning System for Big Data Analytics", CIDR, pp.261–272, 2011.
3. Divyakant Agrawal Sudipto Das Amr El Abbadi,"Big Data and Cloud Computing: Current State and Future Opportunities", EDBT, pp.530-533, 2011.
4. Mukherjee, A.; Datta, J.; Jorapur, R.; Singhvi, R.; Haloi, S.Akram, "Shared disk big data analytics with Apache Hadoop",pp. 18-22, 2012.
5. Daniel E. O'Leary,"Artificial Intelligence and Big Data", IEEE Computer Society, pp. 96-99, March-April 2013.
6. Yaxiong Zhao, Jie Wu. Dache,"A Data Aware Caching for Big-Data Applications Using The MapReduce Framework",Proc. 32nd IEEE Conference on Computer Communications, INFOCOM 2013, IEEE Press, pp. 35-39, April 2013.
7. Dong, X.L.; Srivastava, D. Data Engineering (ICDE)," Big data integration", IEEE International Conference, pp.1245–1248, 2013.
8. Garlasu, D.; Sandulescu, V.; Halcu, I.; Neculoiu, G,"A Big Data implementation based on Grid Computing", Grid Computing,pp. 17-19,2013.
9. Ulla Gain1 ,VirpiHotti ,"Big Data Analytics for Professionals, Data-milling for Laypeople ",International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 4, Number 1, pp. 33-40, 2013.
10. Shweta Pandey, Vrinda Tokekar,"Prominence of MapReduce in BIG DATA Processing", Fourth International Conference on Communication Systems and Network Technologies, IEEE, pp.555-560 , 2014.
11. Sreedhar C, Dr. D. Kavitha, K. Asha Rani," Big Data and Hadoop", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 3 Issue 5, May 2014.
12. Sameera Siddiqui, Deepa Gupta. " Big Data Process Analytics: A Survey", International Journal of Emerging Research in Management &Technology ISSN: 2278-9359,Volume-3, Issue-7, July 2014.

13. Bhawna Gupta , Dr. KiranJyoti ,"Big Data Analytics with Hadoop to analyze Targeted Attacks on Enterprise Data", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) ,pp.3867-3870, 2014.
14. Jainendra Singh, "Big Data Analytic and Mining with Machine Learning Algorithm", World Journal of Computer Application and Technology 1(2),pp.51 - 57,2014.
15. Ishwarappa, Anuradha J," A Brief Introduction on Big Data 5V's Characteristics and Hadoop Technology", International Conference on Intelligent Computing, Communication & Convergence,ICCC-2015.