ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

# Map Reduce Using Big Data and Hadoop: A Review Paper

**Dr.Ramana (Naik) Banothu**
**Professor & Principal,**
**Department of Computer Science and Engineering,**
**Trinity College of Engineering & Technology,**
**Peddapalli.**

**Dr.A.Arun Kumar**
**Professor,**
**Department of Computer Science and Engineering,**
**Balaji Institute of Technology & Science.**
**Narsampet.**

**Abstract:**
The term 'Big Data' depicts creative systems and innovations to catch, store, disperse, oversee and break down petabyte-or bigger measured datasets with high-speed and diverse structures. Huge information can be organized, unstructured or semi-organized, bringing about ineptitude of routine information administration techniques. Information is produced from different diverse sources and can touch base in the framework at different rates. Keeping in mind the end goal to handle these a lot of information in a reasonable and effective way, parallelism is utilized. Huge Data is an information whose scale, differing qualities, and multifaceted nature require new design, methods, calculations, and examination to oversee it and concentrate esteem and concealed learning from it. Hadoop is the center stage for organizing Big Data, and takes care of the issue of making it valuable for examination purposes. Hadoop is an open source programming venture that empowers the conveyed handling of extensive information sets crosswise over groups of product servers. It is intended to scale up from a solitary server to a huge number of machines, with a high level of adaptation to internal failure.

**Keywords:**
Big Data, Hadoop, Map Reduce, HDFS, Hadoop Components

## 1. Introduction:

### A. Big Data: Definition:

Big data is a term that alludes to information sets or blends of information sets whose size (volume), multifaceted nature (inconstancy), and rate of development (speed) make them hard to be caught, overseen, handled or investigated by routine advancements and apparatuses, for example, social databases and desktop measurements or representation bundles, inside the time important to make them helpful. While the size used to figure out if a specific information set is viewed as large information is not solidly characterized and keeps on changing after some time, most examiners and specialists at present allude to information sets from 30-50 terabytes(10 12 or 1000 gigabytes for every terabyte) to numerous petabytes (1015 or 1000 terabytes for every petabyte) as large information. Figure No. 1.1 gives Layered Architecture of Big Data System. It can be disintegrated into three layers, including Infrastructure Layer, Computing Layer, and Application Layer through and through.

### B. 3 Vs of Big Data:

**Volume of Data:**
Volume refers to amount of data. Volume of data stored in enterprise repositories have grown from megabytes and gigabytes to petabytes.

**Variety of Data:**
Different types of data and sources of data. Data variety exploded from structured and legacy data stored in enterprise repositories to unstructured, semi structured, audio, video, XML etc.

**Velocity of Data:**
Velocity refers to the speed ofdata processing. For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

### C. Problem with Big Data Processing

### I. Heterogeneity and Incompleteness:

At the point when people expend data, a lot of heterogeneity is serenely endured. Truth be told, the subtlety and wealth of characteristic dialect can give significant profundity. Be that as it may, machine examination calculations expect homogeneous information, and can't comprehend subtlety. In outcome, information must be precisely organized as an initial phase in (or preceding) information investigation. PC frameworks work most proficiently in the event that they can store different things that are all indistinguishable in size and structure. Effective representation, get to, and examination of semi-organized information require encourage work.

### II. Scale:

Obviously, the primary thing anybody considers with Big Data is its size. All things considered, "enormous" is there in the very name. Overseeing extensive and quickly expanding volumes of information has been a testing issue for a long time. Before, this test was relieved by processors getting quicker, after Moore's law, to furnish us with the assets expected to adapt to expanding volumes of information. In any case, there is a basic move in progress now: information volume is scaling quicker than process assets, and CPU rates are static.


**Figure 1: Layered Architecture of Big Data System**

### III. Timeliness:

The other side of size is speed. The bigger the information set to be handled, the more it will take to investigate. The plan of a framework that successfully manages size is likely likewise to bring about a framework that can procedure a given size of information set quicker. In any case, it is not only this speed is typically implied when one talks about Velocity with regards to Big Data. Or maybe, there is a securing rate challenge.

### IV. Privacy:

The Privacy of information is another immense concern, and one that increments with regards to Big Data. For electronic wellbeing records, there are strict laws administering what should and can't be possible. For other information, directions, especially in the US, are less intense. Notwithstanding, there is awesome open dread in regards to the unseemly utilization of individual information, especially through connecting of information from various sources. Overseeing security is successfully both a specialized and a sociological issue, which must be tended to mutually from both points of view to understand the guarantee of enormous information.

### V. Human Collaboration:

Regardless of the gigantic advances made in computational examination, there stay numerous examples that people can without much of a stretch identify yet PC calculations experience considerable difficulties. In a perfect world, examination for Big Data won't be all computational rather it will be planned expressly to have a human on top of it. The new sub-field of visual examination is endeavoring to do this, at any rate regarding the demonstrating and investigation stage in the pipeline. In today's intricate world, it regularly takes numerous specialists from various spaces to truly comprehend what is going on. A Big Data examination framework must bolster enter from different human specialists, and shared investigation of results. These different specialists might be isolated in space and time when it is excessively costly, making it impossible to gather a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

whole group together in one room. The information framework needs to acknowledge this appropriated master information, and bolster their cooperation.

## 2. Hadoop: Solution for Big Data Processing:

Hadoop is a Programming structure used to bolster the handling of expansive information sets in a disseminated figuring environment. Hadoop was created by Google's Map Reduce that is a product structure where an application separate into different parts. The Current Appache Hadoop biological community comprises of the Hadoop Kernel, MapReduce, HDFS and quantities of different segments like Apache Hive, Base and Zookeeper. HDFS and MapReduce are clarified in taking after focuses.

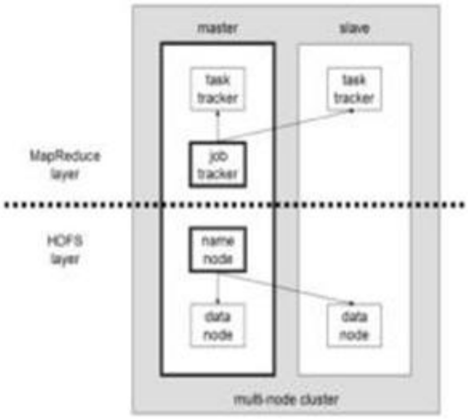


**Figure 2: Hadoop Architecture**

### A. HDFS Architecture:

Hadoop incorporates a fault-tolerant stockpiling framework called the Hadoop Distributed File System, or HDFS. HDFS can store tremendous measures of data, scale up incrementally and survive the disappointment of critical parts of the capacity framework without losing information. Hadoop makes bunches of machines and facilitates work among them. Bunches can be worked with economical PCs. In the event that one comes up short, Hadoop keeps on working the group without losing information or interfering with work, by moving work to the rest of the machines in the bunch. HDFS oversees capacity on the group by breaking approaching records into pieces, called "squares," and putting away each of the squares needlessly over the pool of servers. In the normal case, HDFS stores three finish duplicates of every document by replicating every piece to three unique servers.

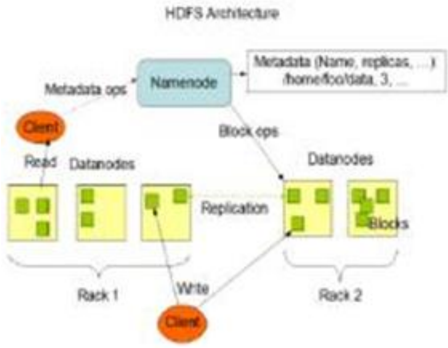


**Figure 3: HDFS Architecture**

### B. MapReduce Architecture:

The handling column in the Hadoop environment is the MapReduce structure. The structure permits the determination of an operation to be connected to an immense information set, separate the issue and information, and run it in parallel. From an investigator's perspective, this can happen on various measurements. For instance, an expansive dataset can be lessened into a littler subset where investigation can be connected. In a conventional information warehousing situation, this may involve applying an ETL operation on the information to deliver something usable by the investigator. In Hadoop, these sorts of operations are composed as MapReduce occupations in Java. There are various more elevated amount dialects like Hive and Pig that make composing these projects simpler. The yields of these occupations can be composed back to either HDFS or set in a customary information distribution center. There are two capacities in MapReduce as takes after:

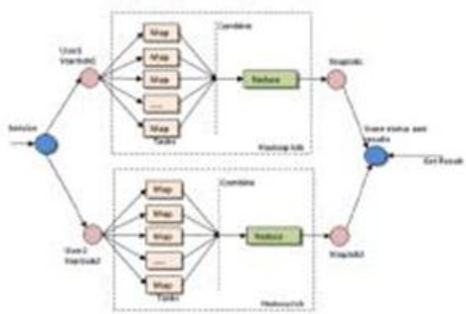

**Figure 4: MapReduce Architecture**

**Map:**
the function takes key/value pairs as input and generates an intermediate set of key/value pairs

**Reduce:**
the function which merges all the intermediate values associated with the same intermediate key

**3. Literature Review:**

**S. Vikram Phaneendra & E. Madhusudhan Reddy** et.al. Illustrated that in olden days the data was lessand easily handled by RDBMS but recently it is difficult to handle huge data through RDBMS tools, which is preferred as "big data". In this they told that big data differs from other data in 5 dimensions such as volume, velocity, variety, value and complexity. They illustrated the hadoop architecture consisting of name node, data node, edge node, HDFS to handle big data systems. Hadoop architecture handle large data sets, scalable algorithm does log management application of big data can be found out in financial, retail industry, health-care, mobility, insurance. The authors also focused on the challenges that need to be faced by enterprises when handling big data: - data privacy, search analysis, etc [1].

**Kiran kumara Reddi & Dnvsl Indira et.al.** Enhanced us with the knowledge that Big Data is combination of structured , semi-structured ,unstructured homogenous and heterogeneous data .The author suggested to use nice model to handle transfer of huge amount of data over the network .Under this model, these transfers are relegated to low demand periods where there is ample ,idle bandwidth available . This bandwidth can then be repurposed for big data transmission without impacting other users in system. The Nice model uses a store –and-forward approach by utilizing staging servers. The model is able to accommodate differences in time zones and variations in bandwidth. They suggested that new algorithms are required to transfer big data and to solve issues like security, compression, routing algorithms [2].

**Jimmy Lin et.al.** used Hadoop which is currentlythe large –scale data analysis " hammer" of choice, but there exists classes of algorithms that aren't " nails" in the sense that they are not particularly amenable to the MapReduce programming model . He focuses on the simple solution to find alternative non-iterative algorithms that solves the same problem. The standard MapReduce is well known and described in many places .Each iteration of the pagerank corresponds to the MapReduce job. The author suggested iterative graph, gradient descent & EM iteration which is typically implemented as Hadoop job with driven set up iteration &Check for convergences. The author suggests that if all you have is a hammer, throw away everything that's not a nail [3].

**Wei Fan & Albert Bifet et.al.** Introduced Big Data Mining as the capability of extracting Useful information from these large datasets or streams of data that due to its Volume, variability and velocity it was not possible before to do it. The author also started that there are certain controversy about Big Data. There certain tools for processes. Big Data as such hadoop, strom, apache S4. Specific tools for big graph mining were PEGASUS & Graph. There are certain Challenges that need to death with as such compression, visualization etc.[4].

**Albert Bifet et.al.** Stated that streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge, allowing organizations to react quickly when problem appear or detect to improve performance. Huge amount of data is created everyday termed as " big data". The tools used for mining big data are apache hadoop, apache big, cascading, scribe, storm, apache hbase, apache mahout, MOA, R, etc. Thus, he instructed that our ability to handle many exabytes of data mainly dependent on existence of rich variety dataset, technique, software framework [5].

**Bernice Purcell et.al.** Started that Big Data is comprised of large data sets that can't be handle by traditional systems.

Big data includes structured data, semi-structured and unstructured data. The data storage technique used for big data includes multiple clustered network attached storage (NAS) and object based storage. The Hadoop architecture is used to process unstructured and semi-structured using map reduce to locate all relevant data then select only the data directly answering the query. The advent of Big Data has posed opportunities as well challenges to business [6].

**Sameer Agarwal et.al.** Presents a Blink DB, a approximate query engine for running interactive SQL queries on large volume of data which is massively parallel. BlinkDB uses two key ideas: (1) an adaptive optimization framework that builds and maintains a set of multi-dimensional stratified samples from original data over time, and (2) A dynamic sample selection strategy that selects an appropriately sized sample based on a query's accuracy or response time requirements [7].

**Yingyi Bu et.al.** Used a new technique called as HaLoop which is modified version of Hadoop Map Reduce Framework, as Map Reduce lacks built-in-support for iterative programs HaLoop allows iterative applications to be assembled from existing Hadoop programs without modification, and significantly improves their efficiency by providing interoperation caching mechanisms and a loop-aware scheduler to exploit these caches. He presents the design, implementation, and evaluation of HaLoop, a novel parallel and distributed system that supports large-scale iterative data analysis applications. HaLoop is built on top of Hadoop and extends it with a new programming model and several important optimizations that include (1) a loop-aware task scheduler, (2) loop- invariant data caching, and (3) caching for efficient fix point verification [8].

**Shadi Ibrahim et.al.** Project says presence of partitioning skew1 causes a huge amount of data transfer during the shuffle phase and leads to significant unfairness on the reduce input among different data nodes.

In this paper, author develop a novel algorithm named LEEN for locality aware and fairness-aware key partitioning in Map Reduce. LEEN embraces an asynchronous map and reduce scheme. Author has integrated LEEN into Hadoop. His experiments demonstrate that LEEN can efficiently achieve higher locality and reduce the amount of shuffled data. More importantly, LEEN guarantees fair distribution of the reduce inputs. As a result, LEEN achieves a performance improvement of up to 45% on different workloads. To tackle all this he presents a present a technique for Handling Partitioning Skew in Map Reduce using LEEN [9].

**Kenn Slagter et.al.** Proposes an improved partitioning algorithm that improves load balancing and memory consumption. This is done via an improved sampling algorithm and partitioner. To evaluate the proposed algorithm, its performance was compared against a state of the art partitioning mechanism employed by Tera Sort as the performance of Map Reduce strongly depends on how evenly it distributes this workload. This can be a challenge, especially in the advent of data skew. In Map Reduce, workload distribution depends on the algorithm that partitions the data. One way to avoid problems inherent from data skew is to use data sampling. How evenly the partitioner distributes the data depends on how large and representative the sample is and on how well the samples are analyzed by the partitioning mechanism. He uses an improved partitioning mechanism for optimizing massive data analysis using Map Reduce for evenly distribution of workload [10].

**Ahmed Eldawy et.al.** presents the first full-fledged Map Reduce framework with native support for spatial data that is spatial data Spatial Hadoop pushes its spatial constructs in all layers of Hadoop, namely, language, storage, Map Reduce and operations layers. In the language layer, a simple high level language is provided to simplify spatial data analysis for non-technical users. In the storage layer, a two-layered spatial index structure is provided where the global index partitions data across nodes while the local index organizes data in each node. This structure is used to build a grid index, an R-tree or an R+-tree. Spatial-Hadoop is a comprehensive extension to Hadoop that pushes spatial data inside the core functionality of Hadoop. Spatial Hadoop runs existing Hadoop programs as is, yet, it achieves order(s) of magnitude better performance than Hadoop when dealing with spatial data. Spatial Hadoop employs a simple spatial high level language, a two-level spatial index structure, basic spatial components built inside the Map Reduce layer, and three basic spatial operations: range queries, k-NN queries, and spatial join. Author presents an efficient Map Reduce framework for Spatial Data [11].

**Jeffrey Dean et.al.** Implementation of Map Reduce runs on a large cluster of commodity machines and is highly scalable: a typical Map Reduce computation processes many terabytes of data on thousands of machines. Programmers and the system easy to use: hundreds of Map Reduce programs have been implemented and upwards of one thousand Map Reduce jobs are executed on Google's clusters every day. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine Communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system. Author proposes Simplified Data Processing on Large Clusters [12].

**Chris Jermaine et.al.** Proposes a Online Aggregation for Large-Scale Computing. Given the potential for OLA to be newly relevant, and given the current interest on very large-scale, data-oriented computing, in this paper we consider the problem of providing OLA in a shared-nothing environment. While we concentrate on implementing OLA on top of a Map Reduce engine, many of author's most basic project contributions are not specific to Map Reduce, and should apply broadly. Consider how online aggregation can be built into a Map Reduce system for large-scale data processing. Given the Map Reduce paradigm's close relationship with cloud computing (in that one might expect a large fraction of Map Reduce jobs to be run in the cloud), online aggregation is a very attractive technology. Since large-scale cloud computations are typically pay-as-you-go, a user can monitor the accuracy obtained in an online fashion, and then save money by killing the computation early once sufficient accuracy has been obtained [13].

**Tyson Condie et.al.** propose a modified Map Reduce architecture in which intermediate data is pipelined between operators, while preserving the programming interfaces and fault tolerance models of other Map Reduce frameworks. To validate this design, author developed the Hadoop Online Prototype (HOP), a pipelining version of Hadoop. Pipelining provides several important advantages to a Map Reduce framework, but also raises new design challenges. To simplify fault tolerance, the output of each Map Reduce task and job is materialized to disk before it is consumed. In this demonstration, we describe a modified Map Reduce architecture that allows data to be pipelined between operators. This extends the Map Reduce programming model beyond batch processing, and can reduce completion times and improve system utilization for batch jobs as well. We demonstrate a modified version of the Hadoop Map Reduce framework that supports online aggregation, which allows users to see "early returns" from a job as it is being computed. Our Hadoop Online Prototype (HOP) also supports continuous queries, which enable MapReduce programs to be written for applications such as event monitoring and stream processing [14].

**Jonathan Paul Olmsted et.al.** Derive the necessary results to apply variation Bayesian inference to the ideal point model. This deterministic, approximate solution is shown to produce comparable results to those from standard estimation strategies. However, unlike these other estimation approaches, solving for the (approximate) posterior distribution is rapid and easily scales to 'big data'. Inferences from the variation Bayesian approach to ideal point estimation are shown to be equivalent to standard approaches on modestly-sized roll call matrices from recent sessions of the US Congress. Then, the ability of variation inference to scale to big data is demonstrated and contrasted with the performance of standard approaches.[15]

**Jonathan Stuart Ward et.al.** did a survey of Bigdata definition, Anecdotally big data is predominantly associated with two ideas:
data storage and data analysis. Despite the sudden Interest in big data, these concepts are far from new and have long lineages. This, therefore, raises the question as to how big data is notably different from conventional data processing techniques. For rudimentary insight as to the answer to this question one need look no further than the term big data. \Big" implies significance, complexity and challenge. Unfortunately the term\big" also invites quantification and therein lies the difficulty in furnishing a definition. The lack of a consistent definition introduces ambiguity and hampers discourse relating to big data. This short paper attempts to collate the various definitions which have gained some degree of traction and to furnish a clear and concise definition of an otherwise ambiguous term [16].

**Albert Bifet et.al.** Discuss the current and future trends of mining evolving data streams, and the challenges that the field will have to overcome during the next years. Data stream real time analytics are needed to manage the data currently generated, at an ever increasing rate, from such applications as: sensor networks, measurements in network monitoring and traffic management, log records or click-streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts and others. In fact, all data generated can be considered as streaming data or as a snapshot of streaming data, since it is obtained from an interval of time. Streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge from what is happening now, allowing organizations to react quickly when problems appear or to detect new trends helping to improve their performance. Evolving data streams are contributing to the growth of data created over the last few years. We are creating the same quantity of data every two days, as we created from the dawn of time up until 2003. Evolving data streams methods are becoming a low-cost, green methodology for real time online prediction and analysis [17].

**Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N. et.al** did the analysis of big data he stated that Data is generated through many sources like business processes, transactions, social networking sites, web servers, etc. and remains in structured as well as unstructured form . Today's business applications are having enterprise features like large scale, data-intensive, web-oriented and accessed from diverse devices including mobile devices. Processing or analyzing the huge amount of data or extracting meaningful information is a challenging task. The term "Big data" is used for large data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target currently ranging from a few dozen terabytes to many peta bytes of data in a single data set. Difficulties include capture, storage, search, sharing, analytics and visualizing. Typical examples of big data found in current scenario includes web logs, RFID generated data, sensor networks, satellite and geo-spatial data, social data from social networks, Internet text and documents, Internet search indexing, call detail records, astronomy, atmospheric science, genomics, biogeochemical, biological, and other complex and/or interdisciplinary scientific project, military. Surveillance, medical records, photography archives, video archives, and large-scale ecommerce [18].

**Kyong-Ha Lee Hyunsik Choi et.al.** Proposes aprominent parallel data processing tool Map Reduce survey intends to assist the database and open source communities in understanding various technical

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

aspects of the Map Reduce framework. In this survey, we characterize the Map Reduce framework and discuss its inherent pros and cons. We then introduce its optimization strategies reported in the recent literature. author also discuss the open issues and challenges raised on parallel data analysis with Map Reduce [19].

**Chen He Ying Lu David Swanson et.al** develops anew MapReduce scheduling technique to enhance map task's data locality. He has integrated this technique into Hadoop default FIFO scheduler and Hadoop fair scheduler. To evaluate his technique, he compares not only Map Reduce scheduling algorithms with and without his technique but also with an existing data locality enhancement technique (i.e., the delay algorithm developed by Facebook).Experimental results show that his technique often leads to the highest data locality rate and the lowest response time for map tasks. Furthermore, unlike the delay algorithm, it does not require an intricate parameter tuning process [20].

## 4. Other Components of Hadoop:

The Table 1, Comparison among Components of Hadoop, gives details of different Hadoop Components which have been used now days. HBase, Hive, Mongo DB, Redis, Cassandra and Drizzle are the different components. Comparison among these components is done on the basis of Concurrency, Durability, Replication Method, Database Model and Consistency Concepts used in the components.

**Table 1: Comparison among Components of Hadoop**

| Name | HBase | Hive | MongoDB | Redis | Cassandra | Drizzle |
|---|---|---|---|---|---|---|
| **Description** | Wide-column store based on Apache Hadoop and on concepts of Big Table | Data Warehouse Software for Querying and Managing Large Distributed Datasets, built on Hadoop | One of the most popular Document Stores | In-memory Database with configurable options performance vs. persistency | Wide-column store based on ideas of Big Table and DynamoDB | My SQL fork with a pluggable micro-kernel and with an emphasis of performance over compatibility |
| **Implementation language** | Java | Java | C++ | C | Java | C++ |
| **Database Model** | Wide Column Store | Relational DBMS | Document Store | Key – Value Store | Wide Column Store | Relational DBMS |
| **Consistency Concepts** | Immediate Consistency | Eventual Consistency | Eventual Consistency, Immediate Consistency | - | Eventual Consistency, Immediate Consistency | - |
| **Concurrency** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Durability** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Replication Method** | Selected Replication factor | Selected Replication factor | Master – Slave Replication | Master – Slave Replication | Selected Replication factor | Master – Master Replication, Master – Slave Replication |

## 5. Conclusion:

We have entered an era of Big Data. The paper describes the concept of Big Data along with 3 Vs, Volume, Velocity and variety of Big Data. The paper also focuses on Big Data processing problems. These technical challenges must be addressed for efficient and fast processing of Big Data. The challenges include not just the obvious issues of scale, but also heterogeneity, lack of structure, error-handling, privacy, timeliness, provenance, and visualization, at all stages of the analysis pipeline from data acquisition to result interpretation. These technical challenges are common across a large variety of application domains, and therefore not cost-effective to address in the context of one domain alone. The paper describes Hadoop which is an open source software used for processing of Big Data.

## REFERENCES:

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 "Map Reduce 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 ACM978-1-4503-1994 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: 11April 2013.

11. Ahmed Eldawy, Mohamed F. Mokbel "A Demonstration of Spatial Hadoop: An Efficient Map Reduce 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 Map Reduce 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 Map Reduce" SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-0032-2/10/06.

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 Map Reduce 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.

21. Harshawardhan S. Bhosale and Prof. Devendra P. Gadekar " A Review Paper on Big Data and Hadoop" in International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014 1 ISSN 2250-3153.

## Author Biography:

**Dr. Ramana (Naik) Banothu**

A warded Ph.D from SC&SS, JNU, New Delhi in data mining. Working as Professor & Principal Trinity Engineering College, Pedapalli Dist., Telangana State; He is fellow member of international and national level professional bodies and he is a reviewer for peer reviewed journals. His areas of interest are Computer Graphics, Data Mining, Artificial Intelligence. He represented for Indian Delegate & presented paper on topic "Youth Role in World Peace" at 16th World Youth Festival Venezuela 2005 @ Caracus. He is a Member in Fellow FIE, Institute of Engineers (India),OUCIP – Osmania University Centre for International Programmes, (Previously called as ASRC – American Studies Research Centre ); Computer Society of India (CSI); Institute of Constitutional & Parliamentary Studies (ICPS), VP House, New Delhi; Osmania Graduates Association (OGA).