ANALYZING MANEUVER OF HADOOP FRAMEWORK AND MAPR ALGORITHM PROFICIENT IN SUPERVISING BIG DATA

Supreet Kaur[1], Dr. Gagandeep Jagdev[2]

[1]Research Scholar, M.Tech. (CE), Yadavindra College of Engineering, Talwandi Sabo (PB).

[2]Faculty, Dept. of Comp. Sc., Punjabi University Guru Kashi College, Damdama Sahib (PB).

ABSTRACT

Big data refers to the data sets that are too big to be handled using the existing database management tools and are emerging in many important applications, such as Internet search, business informatics, social networks, social media, genomics, and meteorology. In simple words it can be said that any data which challenges the currently existing techniques for handling data can be referred as big data. Big data presents a grand challenge for database and data analytics research. Gone are the days when memory was used to be measured in terms of Gigabytes, Terabytes or Petabytes. Today even larger units are being used to measure memory like Exabyte, Zettabytes and Yottabytes. In this research paper the central theme is to discuss about different technologies that are involved in handling big data efficiently. We have analyzed MPP (Massive Parallel Processing), NoSQL, Hadoop Frame work and Map Reduce technology. We will also discuss the architecture working behind these technologies in this research paper.

Keywords- Big Data, MPP, NoSQL, Hadoop, MapReduce.

I. INTRODUCTION

Big Data [2, 3] is not a single technology, technique or initiative. Rather, it is a trend across many areas of business and technology. Talking about technologies enabling the use of Big Data, there are three fundamental technological strategies for storing and providing fast access to large data sets [1, 5, 7].

- *Improved hardware performance and capacity:* Use faster CPUs, use more CPU cores (Requires parallel/threaded operations to take advantage of multi-core CPUs), increase disk capacity and data transfer throughput, increased network throughput (Massively Parallel Processing (MPP).
- *Reducing the size of data accessed:* Data compression and data structures that, by design, limit the amount of data required for queries. e.g. bitmaps, column-oriented databases, NoSQL(Not Only SQL).
- Distributing data and parallel processing: putting data on more disks to parallelize disk I/O, put slices of data on separate compute nodes that can work on these smaller slices in parallel, use massively distributed architectures with emphasis on fault tolerance and performance monitoring with higher-throughput networks to improve data transfer between nodes (Hadoop and MapReduce).

II. AMAZING FACTS ABOUT BIG DATA

- Every 2 days we create as much information as we did from the beginning of time until 2003.
- Over 90% of all the data in the world was created in the past 2 years.
- It is expected that by 2020 the amount of digital information in existence will have grown from 3.2 zettabytes today to 40 zettabytes.
- The total amount of data being captured and stored by industry doubles every 1.2 years.
- Every minute we send 204 million emails, generate 1,8 million Facebook likes, send 278 thousand Tweets, and up-load 200,000 photos to Facebook.
- Google alone processes on average over 40 thousand search queries per second, making it over 3.5 billion in a single day.
- Around 100 hours of video are uploaded to YouTube every minute and it would take you around 15 years to watch every video uploaded by users in one day.
- Around 100 hours of video are uploaded to YouTube every minute and it would take you around 15 years to watch every video uploaded by users in one day.
- If you burned all of the data created in just one day onto DVDs, you could stack them on top of each other and reach the moon – twice.
- AT&T is thought to hold the world's largest volume of data in one unique database – its phone records database is 312 terabytes in size, and contains almost 2 trillion rows.
- 571 new websites spring into existence every minute of every day.
- 1.9 million IT jobs will be created in the US by 2015 to carry out big data projects. Each of those will be supported by 3 new jobs created outside of IT – meaning a total of 6 million new jobs thanks to big data.
- Today's data centers occupy an area of land equal in size to almost 6,000 football fields.
- Twitter performs sentiment analysis on tweets posted on it and analyze 12 terabytes of tweets every day. The amount of data transferred over mobile networks increased by 81% to 1.5 exabytes (1.5 billion gigabytes) per month between 2012 and 2014. Video accounts for 53% of that total.
- The value of the Hadoop market is expected to soar from $2 billion in 2013 to $50 billion by 2020.
- The number of Bits of information stored in the digital universe is thought to have exceeded the number of stars in the physical universe in 2007.
- This year, there will be over 1.2 billion smart phones in the world (which are stuffed full of sensors and data collection features), and the growth is predicted to continue.
- The boom of the Internet of Things will mean that the amount of devices connected to the Internet will rise from about 13 billion today to 50 billion by 2020.

- 12 million RFID (Radio Frequency Identification) tags – used to capture data and track movement of objects in the physical world – had been sold in by 2011. By 2021, it is estimated that number will have risen to 209 billion as the Internet of Things takes off.
- Big data has been used to predict crimes before they happen – a "predictive policing" trial in California was able to identify areas where crime will occur three times more accurately than existing methods of forecasting. By better integrating big data analytics into healthcare, the industry could save $300bn a year – that's the equivalent of reducing the healthcare costs of every man, woman and child by $1,000 a year.
- Retailers could increase their profit margins by more than 60% through the full exploitation of big data analytics.
- The big data industry is expected to grow from US$10.2 billion in 2013 to about US$54.3 billion by 2017.

III. CHALLENGES CONCERNED WITH BIG DATA

A.Variety and Heterogeneity

In the past, data mining techniques have been used to discover unknown patterns and relationships of interest from structured, homogeneous, and small datasets. Variety, as one of the essential characteristics of big data, is resulted from the phenomenon that there exist nearly unlimited different sources that generate or contribute to big data. This phenomenon naturally leads to the great variety or heterogeneity of big data. The data from different sources inherently possesses a great many different types and representation forms, and is greatly interconnected, interrelated, and delicately and inconsistently represented. Mining such a dataset, the great challenge is perceivable and the degree of complexity is not even imaginable before we deeply get there. Heterogeneity in big data also means that it is an obligation (rather than an option) to accept and deal with structured, semi-structured, and even entirely unstructured data simultaneously. While structured data can fit well into today's database systems, semi-structured data may partially fit in, but unstructured data definitely will not. Both semi-structured and unstructured data are typically stored in files. This is especially so in data-intensive, scientific computation areas. Nevertheless, though bringing up greater technical challenges, the heterogeneity feature of big data means a new opportunity of unveiling, previously impossible, hidden patterns or knowledge dwelt at the intersections within heterogeneous big data. Like data mining, the process of big data mining shall also start with data selection (from multiple sources). Data filtering, cleaning, reduction, and transformation then follow. There emerge new challenges with each of these preprocessing steps. With data filtering, how do we make sure that the discarded data will not severely degrade the quality of the eventually mined results under the complexity of great heterogeneity of big data? The same question could be adapted and asked to all other preprocessing steps and operations of the data mining process [7, 18].

B. Scalability

The unprecedented volume/scale of big data requires commensurately high scalability of its data management and mining tools. Instead of being timid, we shall proclaim the extreme scale of big data because more data bears more potential insights and knowledge that we have no chance to discover from conventional data (of smaller scales). We are optimistic with the following approaches that, if exploited properly, may lead to remarkable scalability required for future data and mining systems to manage and mine the big data: (1) cloud computing that has already demonstrated admirable elasticity, which, combined with massively parallel computing architectures, bears the hope of realizing the needed scalability for dealing with the volume challenge of big data; (2) advanced user interaction support (either GUI- or language-based) that facilitates prompt and effective system-user interaction. Big data mining straightforwardly implies extremely time-consuming navigation in a gigantic search space, and prompt feedback/interference/guidance from users (ideally domain experts) must be beneficially exploited to help make early decisions, adjust search/mining strategies on the fly, and narrow down to smaller but promising sub-spaces [9, 18].

C. Speed/Velocity

For big data, speed/velocity really matters. The capability of fast accessing and mining big data is not just a subjective desire, it is an obligation especially for data streams (a common format of big data) – we must finish a processing/mining task within a certain period of time, otherwise, the processing/mining results becomes less valuable or even worthless. Exemplary applications with real-time requests include earthquake prediction, stock market prediction and agent-based autonomous exchange (buying/selling) systems. The speed of data mining depends on two major factors: data access time (determined mainly by the underlying data system) and, of course, the efficiency of the mining algorithms themselves [4, 18].

D. Accuracy, Trust, and Provenance

In the past, data mining systems were typically fed with relatively accurate data from well-known and quite limited sources, so the mining results tend to be accurate, too; thus accuracy and trust have never been a serious issue for concern. With the emerging big data, the data sources are of many different origins, not all well-known, and not all verifiable. Therefore, the accuracy and trust of the source data quickly become an issue, which further propagates to the mining results as well. To (at least partially) solve this problem, data validation and provenance tracing become more than a necessary step in the whole knowledge discovery process (including data mining) [18].

IV. ARCHITECTURE OF APACHE HADOOP FRAMEWORK

Much data today is not natively in structured format; for example, tweets and blogs are weakly structured pieces of text, while images and video are structured for storage and display, but not for semantic content and search: transforming such content into a structured format for later analysis is a major challenge. Fig. 1, below gives a glimpse of the Big Data analysis tools which are used for efficient and precise data handling the velocity and heterogeneity of data, tools like Hive, Pig and Mahout are used which are parts of Hadoop and HDFS framework. It is interesting to note that for all the tools used, Hadoop over HDFS is the underlying architecture. Oozie and EMR with Flume and Zookeeper are used for handling the volume and veracity of data, which are standard Big Data management tools. The layer with their specified tools forms the bedrock for Big Data management and analysis framework.

Hadoop is a scalable, open source, fault-tolerant Virtual Grid operating system architecture for data storage and processing. It runs on commodity hardware; it uses HDFS which is fault-tolerant high-bandwidth clustered storage architecture. It runs MapReduce for distributed data processing and is works with structured and unstructured data. HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time [17].

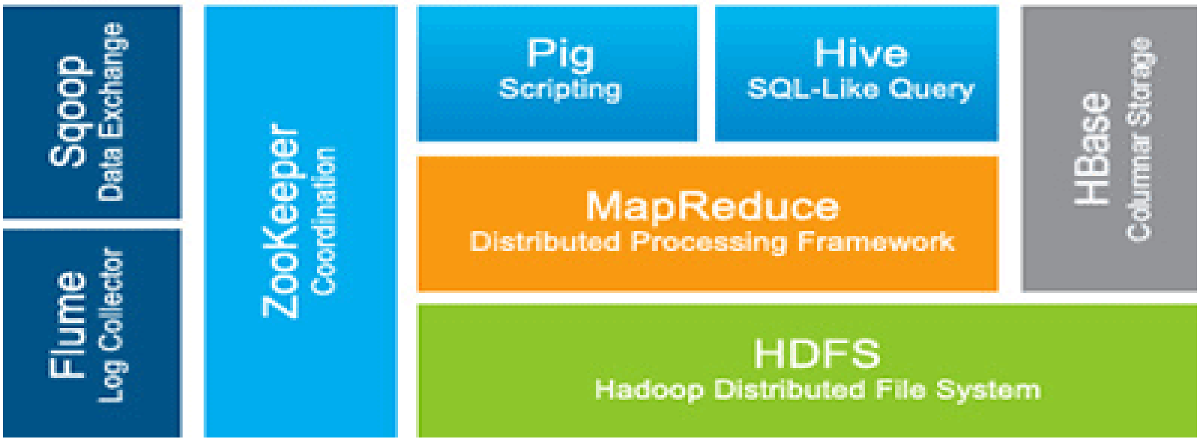

Fig. 1 Figure depicts different tools of Hadoop software

Hadoop [11, 12] is a java based framework that is efficient for processing large data sets in a distributed computing environment. Hadoop is sponsored by Apache Software Foundation. The creator of Hadoop was Doug Cutting and he named the framework after his child’s stuffed toy elephant. Applications are made run on systems with thousands of nodes making use of thousands of terabytes via Hadoop. Distributed file system in Hadoop facilitates fast data transfer among nodes and allows continuous operations of the system even if node failure occurs. This concept lowers the risk of disastrous system failure even if multiple nodes become inoperative. The inspiration behind working of Hadoop is Google’s Map reduce which is a software framework in which application under consideration is broken down into number of small parts [5, 6, 17].

Hadoop is a framework [15] which comprised of six components [4]. Every component is assigned a particular job to be performed. To understand it lets suppose entire system as a Hadoop zoo as shown in Fig. 2.

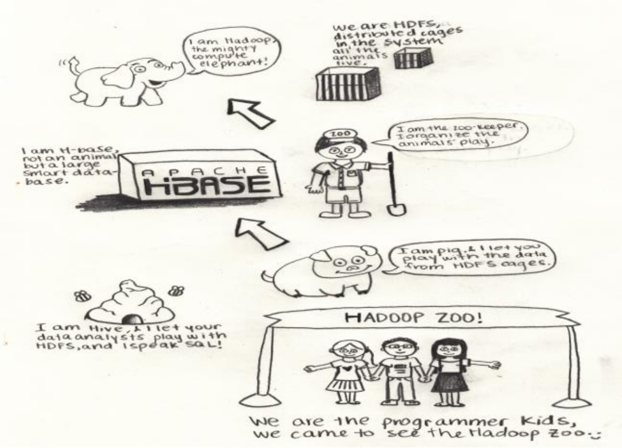
I am Hadoop, the mighty compute elephant!
we are HDFS, distributed cages in the system all the animals live.
I am the zoo-keeper, I organize the animals' play.
ZOO
I am H-base, not an animal but a large smart data-base.
APACHE HBASE
I am pig, & I let you play with the data from HDFS cages.
I am Hive, & I let your data analysts play with HDFS, and I speak SQL!
HADOOP ZOO!
We are the programmer kids, we came to see the Hadoop zoo :)

Fig. 2 Hadoop Zoo

- HDFS – HDFS are distributed cages where all animals live i.e. where data resides in a distributed format.
- Apache HBase – It is a smart and large database.
- Zookeeper- Zookeeper is the person responsible for managing animals play.
- Pig – Pig allows to play with data from HDFS cages.
- Hive- Hive allows data analysts play with HDFS and makes use of SQL.
- HCatalog helps to upload the database file and automatically create table for the user.

V. WORKING OF MAP-REDUCE ALGORITHM

MapReduce [8, 12, 13] is a framework originally developed at Google that allows for easy large scale distributed computing across a number of domains. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Hadoop MapReduce includes several stages, each with an important set of operations helping to get to your goal of getting the answers you need from big data. The process starts with a user request to run a MapReduce program and continues until the results are written back to the HDFS [14, 15].

Map Reduce by itself is capable for analyzing large distributed data sets; but due to the heterogeneity, velocity and volume of Big Data, it is a challenge for traditional data analysis and management tools. A problem with Big Data is that they use NoSQL and has no Data Description Language (DDL) and it supports transaction processing. Also, web-scale data is not universal and it is heterogeneous. For analysis of Big Data, database integration and cleaning is much harder than the traditional mining approaches. Parallel processing and distributed computing is becoming a standard procedure which are nearly non-existent in RDBMS. MapReduce is an architectural model for parallel processing of tasks on a distributed computing system. This algorithm was first described in a paper "MapReduce Simplified Data Processing on Large Clusters," by Jeffery Dean and Sanjay Ghemawat from Google [8]. This algorithm allows splitting of a single computation task to multiple nodes or computers for distributed processing. As a single task can be broken down into multiple subparts, each handled by a separate node, the number of nodes determines the processing power of the system. There are various commercial and open-source technologies that implement the MapReduce algorithm as a part of their internal architecture. A popular implementation of MapReduce is the Apache Hadoop, which is used for data processing in a distributed computing environment. As MapReduce is an algorithm, it can be written in any programming language [10, 16, 17].

The initial part of the algorithm is used to split and 'map' the sub tasks to computing nodes as shown in Fig. 3. The 'reduce' part takes the results of individual computations and combines them to get the final result. In the MapReduce algorithm [2, 3, 6], the mapping function reads the input data and generates a set of intermediate records for the computation. These intermediate records generated by the map function take the form of a (key, data) pair. As a part of mapping function, these records are distributed to different computing nodes using a hashing function. Individual nodes then perform the computing operation and return the results to the reduce function. The reduce function collects the individual results of the computation to generate a final output.

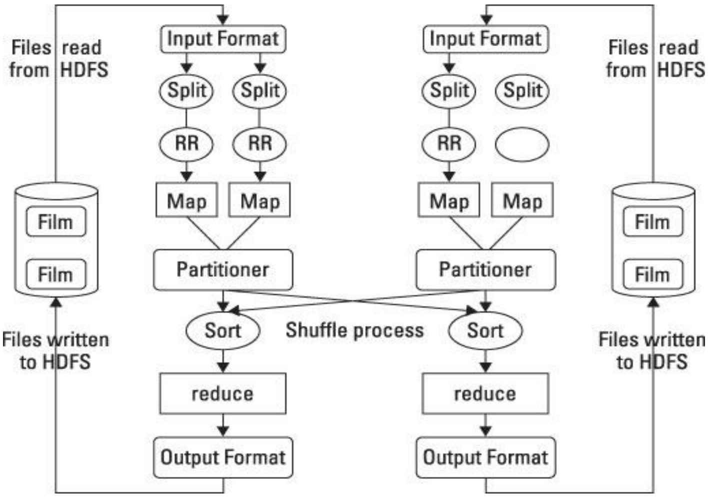

Fig. 3 Working of Map Reduce Technology

The popular social networking website Facebook also makes use of map reduce technology. We will illustrate this via an example.

Facebook has a list of friends (note that friends are a bi-directional thing on Facebook. If I'm your friend, you're mine). They also have lots of disk space and they serve hundreds of millions of requests every day. They've decided to pre-compute calculations when they can to reduce the processing time of requests. One common processing request is the "Gagan and Supreet have 230 friends in common" feature. When you visit someone's profile, you see a list of friends that you have in common. This list doesn't change frequently so it'd be wasteful to recalculate it every time you visited the profile. We're going to use map reduce so that we can calculate everyone's common friends once a day and store those results. Later on it's just a quick lookup. We've got lots of disk, it's cheap.

Assume the friends are stored as Person → [List of Friends], our friends list is then:

A -> B C D

B -> A C D E

C -> A B D E

D -> A B C E

E -> B C D

Each line will be an argument to a mapper. For every friend in the list of friends, the mapper will output a key-value pair. The key will be a friend along with the person. The value will be the list of friends. The key will be sorted so that the friends are in order, causing all pairs of friends to go to the same reducer. This is hard to explain with text, so let's just do it and see if you can see the pattern. After all the mappers are done running, you'll have a list like this:

For map (A→B C D):

(A B) →B C D

(A C) → B C D

(A D) → B C D

For map (B →A C D E) : (Note that A comes before B in the key)

(A B) → A C D E

(B C) → A C D E

(B D) → A C D E

(B E) → A C D E

For map(C → A B D E):

(A C) → A B D E

(B C) → A B D E

(C D) → A B D E

(C E) → A B D E

For map (D → A B C E):

(A D) → A B C E

(B D) → A B C E

(C D) → A B C E

(D E) → A B C E

And finally for map (E -> B C D):

(B E) → B C D

(C E) → B C D

(D E) → B C D

Before we send these key-value pairs to the reducers, we group them by their keys and get:
(A B) → (A C D E) (B C D)
(A C) → (A B D E) (B C D)
(A D) → (A B C E) (B C D)
(B C) → (A B D E) (A C D E)
(B D) → (A B C E) (A C D E)
(B E) → (A C D E) (B C D)
(C D) → (A B C E) (A B D E)
(C E) → (A B D E) (B C D)
(D E) → (A B C E) (B C D)

Each line will be passed as an argument to a reducer. The reduce function will simply intersect the lists of values and output the same key with the result of the intersection. For example, reduce ((A B) -> (A C D E) (B C D)) will output (A B): (C D) and means that friends A and B have C and D as common friends.

The result after reduction is:
(A B) -> (C D)
(A C) -> (B D)
(A D) -> (B C)
(B C) -> (A D E)
(B D) -> (A C E)
(B E) -> (C D)
(C D) -> (A B E)
(C E) -> (B D)
(D E) -> (B C)

Now when D visits B's profile, we can quickly look up (B D) and see that they have three friends in common, (A C E).

This is how Facebook analyses millions of user accounts created on it and finds out that to whom what people should be shown in there "people you may know section".

VI. CONCLUSION

The real issue is not that we are acquiring large amounts of data. It's what you do with the data that counts. The concept of Big Data is being increasingly well defined and transforming from an idea to a well-defined concept with real-world implementations. The need to process enormous quantities of data has never been greater. Not only are terabyte- and petabyte-scale datasets rapidly becoming commonplace, but there is consensus that great value lies buried in them, waiting to be unlocked by the right computational tools. Big Data analysis tools like Map Reduce over Hadoop and HDFS, promises to help organizations better understand their customers and the marketplace, hopefully leading to better business decisions and competitive advantages. For engineers building information processing tools and applications, large and heterogeneous datasets which are generating continuous flow of data, lead to more effective algorithms for a wide range of tasks, from machine translation to spam detection. The ability to analyze massive amounts of data may provide the key to unlocking the secrets of the cosmos or the mysteries of life. MapReduce can be exploited to solve a variety of problems related to text processing at scales that would have been unthinkable a few years ago [13]. New innovations in Big Data technologies are helping increase the adoption rate across various industries. Newer and more capable technologies appear in the market every day. Since Big data is an emerging technology and is at its youth, so it needs to attract organizations and youth with diverse new skill sets. The skills should extend from technical to research, analytical, interpretive and creative ones. These skills need to be developed in individuals hence requires training programs to be held by the organizations. Also universities should introduce curriculum on big data to produce skilled employees and data scientists in this expertise [10].

REFERENCES

[1] N. Marz and J. Warren. Big Data: Principles and best practices of scalable realtime data systems. Manning Publications, 2013.

[2] R. Smolan and J. Erwitt. The Human Face of Big Data.Sterling Publishing Company Incorporated, 2012.

[3] J. Gantz and D. Reinsel. IDC: The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. December 2012.

[4] http://hadoopilluminated.com/hadoop_illuminated/Intro_To_Hadoop.html#d1575e686

[5] Gagandeep Jagdev et. al.," Scrutinizing Elections Strategies by Political Parties via Mining Big Data for Ensuring Big Win in Indian Subcontinent", 4th Edition of International Conference on Wireless Networks and Embedded Systems.

[6] Gagandeep Jagdev et. al.," Applications of Big Data in medical science brings revolution in managing health care of humans",IJEEE, Vol. 2, Spl. Issue 1 (2015).

[7] Katal, A., Wazid, M., & Goudar, R. H. (2013). Big Data: Issues, Challenges, Tools and Good Practices. IEEE, 404-409.

[8] Dean, Jeffery, and Ghemawat Sanjay. 2004. "MapReduce: Simplified Data Processing on Large Clusters." Google.

[9] Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big Data: Issues and Challenges Moving Forward. International Conference on System Sciences (pp. 995-1004). Hawaii:IEEE Computer Soceity.

[10] Dr. Gagandeep Jagdev et al., "Analyzing and Filtering Big Data concerned with elections via Hadoop Framework" in International Journal of Advance Research in Science and Engineering (IJARSE), ISSN (O): 2319-8354, ISSN (P): 2319-8346, Volume No. 6, Issue No, 4, April 2017.

[11] Dr. Gagandeep Jagdev et. al., "Comparing Conventional Data Mining Algorithms with Hadoop based Map-Reduce Algorithm considering elections perspective", in International Journal of Innovative Research in Science and Engineering (IJIRSE), ISSN: 2454-9665 (O), ISSN: 2455-0663(P), Volume – 3, Issue – 3, March 2017.

[12] Dr. Gagandeep Jagdev et. al., "Big Data Proposes an Innovative concept for contesting elections in Indian Subcontinent" in International Journal of Scientific and Technical Advancements (IJSTA), ISSN-2454-1532, 2015.

[13] Dr. Gagandeep Jagdev et. al., "Implementation and Applications of Big Data in health care industry" in International Journal of Scientific and Technical Advancements (IJSTA), ISSN-2454-1532, 2015.

[14] www.forbes.com/sites/lisaarthur/2013/08/15/what-is-big-data/

[15] Mukherjee, A.; Datta, J.; Jorapur, R.; Singhvi, R.; Haloi, S.; Akram, W., (18-22 Dec.,2012) , "Shared disk big data analytics with Apache Hadoop"

[16] Aditya B. Patel, Manashvi Birla, Ushma Nair ,(6-8 Dec. 2012),"Addressing Big Data Problem Using Hadoop and Map Reduce".

[17] Shankar Ganes h Manikandan et al., "Big Data Analysis using Apache Hadoop", 978-1-4799-6541-0/14/$31.00 ©2014 IEEE.

[18] Dunren Che et al., "From Big Data to Big Data Mining: Challenges, Issues, and Opportunities", DASFAA Workshops 2013, LNCS 7827, pp. 1–15, 2013. © Springer-Verlag Berlin Heidelberg 2013.

About the Author

Dr. Gagandeep Jagdev is working in the capacity of faculty member in Dept. of Computer Science, Punjabi University Campus, Damdama Sahib (PB). His total teaching experience is above 10 years and has above 105 international and national publications in reputed journals and conferences to his credit. He is also a member of editorial board and advisory board of several reputed international peer reviewed journals and Technical Committee Member of many reputed universities. His areas of interest are Big Data, Data Mining, RFID, Biometrics, Cloud Computing.