Big Data: A Volume or Technology ?

Savita Sehgal , Yashpal Singh2
[1,2]Department of Computer Science &Engineering,
Ganga Institute of Technology and Management,
Kablana, Jhajjar, Haryana, India

***Abstract*— Big data is a buzzword, or catch-phrase, used to describe a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database and software techniques. In most enterprise scenarios the data is too big or it moves too fast or it exceeds current processing capacity. Big data has the potential to help companies improve operations and make faster, more intelligent decisions. Big data is an all-encompassing term for any collection of data sets so large or complex that it becomes difficult to process them using traditional data processing applications. The challenges include analysis, capture, curation, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on." Big data is difficult to work with using most relational database management systems and desktop statistics and visualization packages, requiring instead "massively parallel software running on tens, hundreds, or even thousands of servers". What is considered "big data" varies depending on the capabilities of the organization managing the set, and on the capabilities of the applications that are traditionally used to process and analyze the data set in its domain. Big Data is a moving target; what is considered to be "Big" today will not be so years ahead. "For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration."[1]**

I. INTRODUCTION

Definition: "Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few.

This data is "**Big Data**."

Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time. Big data "size" is a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data. Big data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive scale.

In a 2001 research report and related lectures, META Group (now Gartner) analyst Doug Laney defined data growth challenges and opportunities as being three-dimensional, i.e. increasing volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources). Gartner, and now much of the industry, continue to use this "3Vs" model for describing big data. In 2012, Gartner updated its definition as follows: "Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. Additionally, a new V "Veracity" is added by some organizations to describe it.

If Gartner's definition (the 3Vs) is still widely used, the growing maturity of the concept fosters a more sound difference between big data and Business Intelligence, regarding data and their use:

- Business Intelligence uses descriptive statistics with data with high information density to measure things, detect trends etc.;
- Big data uses inductive statistics and concepts from nonlinear system identification to infer laws (regressions, nonlinear relationships, and causal effects) from large sets of data with low information density to reveal relationships, dependencies and perform predictions of outcomes and behaviors.

Big data can also be defined as "Big data is a large volume unstructured data which cannot be handled by standard database management systems like DBMS, RDBMS or ORDBMS".

Big data is a buzzword, or catch-phrase, used to describe a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database and software techniques. In most enterprise scenarios the data is too big or it moves too fast or it exceeds current processing capacity. Big data has the potential to help companies improve operations and make faster, more intelligent decisions.[2]

II. HISTORY

Predictive analytics has its origin in the 1940s, when governments started using the first computational models. With non-linear programming and real-time analytics, data analytics and prescriptive analytics goes mainstream and becomes available to all organizations. With the rise of the big data technologies, we have now entered a new era of predictive analytics that will personalize and democratize data (analytics) for organisations, individuals and governments.

From cuneiform, the earliest form of writing, to data centers, the human race as always gathered information. The rise in technology has led to the overflow of data, which constantly requires more sophisticated data storage systems. The recognition of information overload started as early as the 1930s. The boom in the U.S. population, the issuing of social security numbers, and the general growth of knowledge (research) demanded more thorough and organized record-keeping. However, it wasn't too long before the first flag of warning was raised.

While the growth of knowledge was good for society, it was quickly leading to a storage and retrieval problem for libraries. As information continued to boom in the following decades, organizations began to design, develop, and implement centralized computing systems that would allow them to automate their inventory systems. As these systems began to mature across industries and integrate within enterprises, organizations began to use this data to provide answers and insight that would allow them to make better business decisions i.e., business intelligence.

With business intelligence piling up, the challenge of management and storage quickly surfaced yet again. In order to offer more functionality, digital storage had to become more cost-effective. This lead to the emergence of Business Intelligence (BI) platforms. As BI platforms continue to mature, the data gleaned will enable companies, scientific researchers, medical practitioners, our nation's defense and intelligence operations, and more to create revolutionary breakthroughs.[3]

III. TYPES OF DATA IN BIG DATA

***Activity Data*:** Simple activities like listening to music or reading a book are now generating data. Digital music players and eBooks collect data on our activities. Your smart phone collects data on how you use it and your web browser collects information on what you are searching for. Your credit card company collects data on where you shop and your shop collects data on what you buy. It is hard to imagine any activity that does not generate data.

Conversation Data: Our conversations are now digitally recorded. It all started with emails but nowadays most of our conversations leave a digital trail. Just think of all the conversations we have on social media sites like Facebook or Twitter. Even many of our phone conversations are now digitally recorded.

***Photo and Video Image Data*:** Just think about all the pictures we take on our smart phones or digital cameras. We upload and share 100s of thousands of them on social media sites every second. The increasing amounts of CCTV cameras take video images and every minute we up-load hundreds of hours of video images to YouTube and other sites.

Sensor Data: We are increasingly surrounded by sensors that collect and share data. Take your smart phone, it contains a global positioning sensor to track exactly where you are every second of the day, it includes an accelometer to track the speed and direction at which you are travelling. We now have sensors in many devices and products.

The Internet of Things Data: We now have smart TVs that are able to collect and process data, we have smart watches, smart fridges, and smart alarms. The Internet of Things, or Internet of Everything connects these devices so that the traffic sensors on the road send data to your alarm clock which will wake you up earlier than planned because the blocked road means you have to leave earlier to make your 9am meeting…

IV. CHARACTERSTICS OF BIGDATA

Big data can be described by the following characteristics:

Volume – The quantity of data that is generated is very important in this context.It is the size of the data which determines the value and potential of the data under consideration and whether it can actually be considered as Big Data or not.The name 'Big Data' itself contains a term which is related to size and hence the characteristic.

Variety - The next aspect of Big Data is its variety.This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts.This helps the people, who are closely analyzing the data and are associated with it, to effectively use the data to their advantage and thus upholding the importance of the Big Data.

Velocity - The term 'velocity' in the context refers to the speed of generation of data or how fast the data is generated and processed to meet the demands and the challenges which lie ahead in the path of growth and development.

Variability - This is a factor which can be a problem for those who analyse the data. This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Veracity - The quality of the data being captured can vary greatly. Accuracy of analysis depends on the veracity of the source data.

V. ARCHITECTURE

In 2000, Seisint Inc. develops C++ based distributed file sharing framework for data storage and querying. Structured, semi-structured and/or unstructured data is stored and distributed across multiple servers. Querying of data is done by modified C++ called ECL which uses apply scheme on read method to create structure of stored data

during time of query. In 2004 LexisNexis acquired Seisint Inc. and 2008 acquired ChoicePoint, Inc. and their high speed parallel processing platform. The two platforms were merged into HPCC Systems and in 2011 was open sourced under Apache v2.0 License. Currently HPCC and Quantcast File System are the only publicly available platforms capable of analyzing multiple exabytes of data.

In 2004, Google published a paper on a process called MapReduce that used such an architecture. The MapReduce framework provides a parallel processing model and associated implementation to process huge amount of data. With MapReduce, queries are split and distributed across parallel nodes and processed in parallel (the Map step). The results are then gathered and delivered (the Reduce step). The framework was very successful, so others wanted to replicate the algorithm. Therefore, an implementation of the MapReduce framework was adopted by an Apache open source project named Hadoop.

MIKE2.0 is an open approach to information management that acknowledges the need for revisions due to big data implications in an article titled "Big Data Solution Offering". The methodology addresses handling big data in terms of useful permutations of data sources, complexity in interrelationships, and difficulty in deleting (or modifying) individual records.

VI. PRACTICAL EXAMPLES

Better understand and target customers: To better understand and target customers, companies expand their traditional data sets with social media data, browser, text analytics or sensor data to get a more complete picture of their customers. The big objective, in many cases, is to create predictive models. Using big data, Telecom companies can now better predict customer churn; retailers can predict what products will sell, and car insurance companies understand how well their customers actually drive.

***Understand and Optimize Business Processes*:** Big data is also increasingly used to optimize business processes. Retailers are able to optimize their stock based on predictive models generated from social media data, web search trends and weather forecasts. Another example is supply chain or delivery route optimization using data from geographic positioning and radio frequency identification sensors.

Improving Health: The computing power of big data analytics enables us to find new cures and better understand and predict disease patterns. We can use all the data from smart watches and wearable devices to better understand links between lifestyles and diseases. Big data analytics also allow us to monitor and predict epidemics and disease outbreaks, simply by listening to what people are saying, i.e. “Feeling rubbish today - in bed with a cold” or searching for on the Internet.

Improving Security and Law Enforcement: Security services use big data analytics to foil terrorist plots and detect cyber attacks. Police forces use big data tools to catch criminals and even predict criminal activity and credit card companies use big data analytics it to detect fraudulent transactions.

Improving and Optimizing Cities and Countries: Big data is used to improve many aspects of our cities and countries. For example, it allows cities to optimize traffic flows based on real time traffic information as well as social media and weather data. A number of cities are currently using big data analytics with the aim of turning themselves into

Smart Cities, where the transport infrastructure and utility processes are all joined up. Where a bus would wait for a delayed train and where traffic signals predict traffic volumes and operate to minimize jams.

VII. MARKET

Big data has increased the demand of information management specialists in that Software AG, Oracle Corporation, IBM, FICO, Microsoft, SAP, EMC, HP and Dellhave spent more than $15 billion on software firms specializing in data management and analytics. In 2010, this industry was worth more than $100 billion and was growing at almost 10 percent a year: about twice as fast as the software business as a whole.

Developed economies make increasing use of data-intensive technologies. There are 4.6 billion mobile-phone subscriptions worldwide and between 1 billion and 2 billion people accessing the internet. Between 1990 and 2005, more than 1 billion people worldwide entered the middle class which means more and more people who gain money will become more literate which in turn leads to information growth. The world's effective capacity to exchange information through telecommunication networks was 281 petabytes in 1986, 471 petabytes in 1993, 2.2 exabytes in 2000, 65 exabytes in 2007 and it is predicted that the amount of traffic flowing over the internet will reach 667 exabytes annually by 2014. It is estimated that one third of the globally stored information is in the form of alphanumeric text and still image data, which is the format most useful for most big data applications. This also shows the potential of yet unused data (i.e. in the form of video and audio content).

While many vendors offer off-the-shelf solutions for Big Data, experts recommend the development of in-house solutions custom-tailored to solve the companies problem at hand if the company has sufficient technical capabilities.

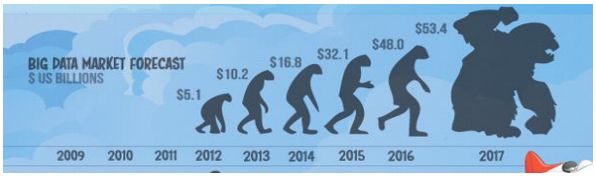

Fig.1 Big Data Market Forecast

By 2015 4.4 million IT jobs in Big Data ; 1.9 million is in US itself

INDIA-BIG DATA

- Gaining attraction
- Huge market opportunities for IT services (82.9% of revenues) and analytics firms (17.1 %)
- Current market size is $200 million. By 2015 $1 billion
- The opportunity for Indian service providers lies in offering services around Big Data implementation and analytics for global multinationals

VIII. FUTURE FORECAST

- $15 billion on software firms only specializing in data management and analytics. This industry on its own is worth more than $100 billion and growing at almost 10% a year which is roughly twice as fast as the software business as a whole.
- In February 2012, the open source analyst firm Wikibon released the first market forecast for Big Data , listing $5.1B revenue in 2012 with growth to $53.4B in 2017
- The McKinsey Global Institute estimates that data volume is growing 40% per year, and will grow 44x between 2009 and 2020.

IX. APACHE HADOOP

Apache Hadoop is an open-source software framework for distributed storage and distributed processing of Big Dataon clusters of commodity hardware. Its Hadoop Distributed File System (HDFS) splits files into large blocks (default 64MB or 128MB) and distributes the blocks amongst the nodes in the cluster. For processing the data, the HadoopMap/Reduce ships code (specifically Jar files) to the nodes that have the required data, and the nodes then process the data in parallel. This approach leverages data locality,[2] in contrast to conventional HPC architecture which usually relies on a parallel file system (compute and data separated, but connected with high-speed networking).

Since 2012, the term "Hadoop" often refers not to just the base Hadoop package but rather to the **Hadoop Ecosystem**, which includes all of the additional software packages that can be installed on top of or alongside Hadoop, such as Apache Hive, Apache Pig and Apache Spark.

The base Apache Hadoop framework is composed of the following modules:

- *Hadoop Common* – contains libraries and utilities needed by other Hadoop modules.
- *Hadoop Distributed File System (HDFS)* – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- *Hadoop YARN* – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- *Hadoop MapReduce* – a programming model for large scale data processing.

All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines, or racks of machines) are common and thus should be automatically handled in software by the framework. Apache Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers.

YARN stands for "Yet Another Resource Negotiator" and was added later as part of Hadoop 2.0. YARN takes the resource management capabilities that were in MapReduce and packages them so they can be used by new engines. This also streamlines MapReduce to do what it does best, process data. With YARN, you can now run multiple applications in Hadoop, all sharing a common resource management. As of September, 2014, YARN manages only CPU (number of cores) and memory, but management of other resources such as disk, network and GPU is planned for the future.

Beyond HDFS, YARN, and MapReduce, the entire Apache Hadoop "platform" is now commonly considered to consist of a number of related projects as well –Apache Pig, Apache Hive, Apache HBase, Apache Spark, and others.

For the end-users, though MapReduce Java code is common, any programming language can be used with "Hadoop Streaming" to implement the "map" and "reduce" parts of the user's program. Apache Pig, Apache Hive, Apache Spark among other related projects expose higher level user interfaces like Pig Latin and a SQL variant respectively. The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell-scripts.

Apache Hadoop is a registered trademark of the Apache Software Foundation.

X. HADOOP CHARACTERSTICS

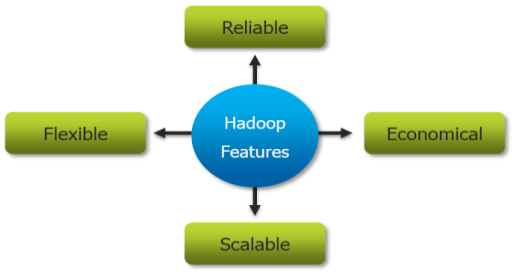

Fig.2 Major characteristics of Hadoop

XI. ARCHITECTURE OF HADOOP

Hadoop consists of the *Hadoop Common* package, which provides filesystem and OS level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2) and the Hadoop Distributed File System (HDFS). The Hadoop Common package contains the necessary Java ARchive (JAR) files and scripts needed to start Hadoop. The package also provides source code, documentation, and a contribution section that includes projects from the Hadoop Community.

For effective scheduling of work, every Hadoop-compatible file system should provide location awareness: the name of the rack (more precisely, of the network switch) where a worker node is. Hadoop applications can use this information to run work on the node where the data is, and, failing that, on the same rack/switch, reducing backbone traffic. HDFS uses this method when replicating data to try to keep different copies of the data on different racks. The goal is to reduce the impact of a rack power outage or switch failure, so that even if these events occur, the data may still be readable.

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or *worker node* acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications. Hadoop requires Java Runtime Environment (JRE) 1.6 or higher. The standard startup and shutdown scripts require that Secure Shell (ssh) be set up between nodes in the cluster.

In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the namenode's memory structures, thus preventing file-system corruption and reducing loss of data. Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the Hadoop MapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode, and DataNode architecture of HDFS are replaced by the file-system-specific equivalents.

Hadoop distributed file system

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. A Hadoop cluster has nominally a single namenode plus a cluster of datanodes, although redundancy options are available for the namenode due to its criticality. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/IP sockets for communication. Clients use remote procedure call (RPC) to communicate between each other.

HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful). With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can talk to each other to rebalance data, to move copies around, and to keep the replication of data high. HDFS is not fully POSIX-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The tradeoff of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.

HDFS added the high-availability capabilities, as announced for release 2.0 in May 2012, letting the main metadata server (the NameNode) fail over manually to a backup. The project has also started developing automatic fail-over.

The HDFS file system includes a so-called *secondary namenode*, a misleading name that some might incorrectly interpreted as a backup namenode for when the primary namenode goes offline. In fact, the secondary namenode regularly connects with the primary namenode and builds snapshots of the primary namenode's directory information, which the system then saves to local or remote directories. These checkpointed images can be used to restart a failed primary namenode without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the namenode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files. HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple namespaces served by separate namenodes.

An advantage of using HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location. For example: if node A contains data (x,y,z) and node B contains data (a,b,c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer. When Hadoop is used with other file systems, this advantage is not always available. This can have a significant impact on job-completion times, which has been demonstrated when running data-intensive jobs.

HDFS was designed for mostly immutable files and may not be suitable for systems requiring concurrent write-operations.

HDFS can be mounted directly with a Filesystem in Userspace (FUSE) virtual file system on Linux and some other Unix systems.

File access can be achieved through the native Java API, the Thrift API to generate a client in the language of the users' choosing (C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk, and OCaml),

the command-line interface, browsed through the HDFS-UI webapp over HTTP, or via 3rd-party network client libraries.

JobTracker and TaskTracker: the MapReduce engine

Above the file systems comes the MapReduce engine, which consists of one *JobTracker*, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available *TaskTracker* nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled. The TaskTracker on each node spawns off a separate Java Virtual Machine process to prevent the TaskTracker itself from failing if the running job crashes the JVM. A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status. The Job Tracker and TaskTracker status and information is exposed by Jetty and can be viewed from a web browser.

If the JobTracker failed on Hadoop 0.20 or earlier, all ongoing work was lost. Hadoop version 0.21 added some checkpointing to this process; the JobTracker records what it is up to in the file system. When a JobTracker starts up, it looks for any such data, so that it can restart work from where it left off.

Known limitations of this approach are:

- The allocation of work to TaskTrackers is very simple. Every TaskTracker has a number of available *slots* (such as "4 slots"). Every active map or reduce task takes up one slot. The Job Tracker allocates work to the tracker nearest to the data with an available slot. There is no consideration of the current system loadof the allocated machine, and hence its actual availability.
- If one TaskTracker is very slow, it can delay the entire MapReduce job – especially towards the end of a job, where everything can end up waiting for the slowest task. With speculative execution enabled, however, a single task can be executed on multiple slave nodes.

Scheduling

By default Hadoop uses FIFO, and optionally 5 scheduling priorities to schedule jobs from a work queue. In version 0.19 the job scheduler was refactored out of the JobTracker, while adding the ability to use an alternate scheduler (such as the *Fair scheduler* or the *Capacity scheduler*, described next).

FAIR SCHEDULER

The fair scheduler was developed by Facebook. The goal of the fair scheduler is to provide fast response times for small jobs and QoS for production jobs. The fair scheduler has three basic concepts.

1. Jobs are grouped into pools.
2. Each pool is assigned a guaranteed minimum share.
3. Excess capacity is split between jobs.

By default, jobs that are uncategorized go into a default pool. Pools have to specify the minimum number of map slots, reduce slots, and a limit on the number of running jobs.

CAPACITY SCHEDULER

The capacity scheduler was developed by Yahoo. The capacity scheduler supports several features that are similar to the fair scheduler.

- Jobs are submitted into queues.
- Queues are allocated a fraction of the total resource capacity.
- Free resources are allocated to queues beyond their total capacity.
- Within a queue a job with a high level of priority has access to the queue's resources.

There is no preemption once a job is running.

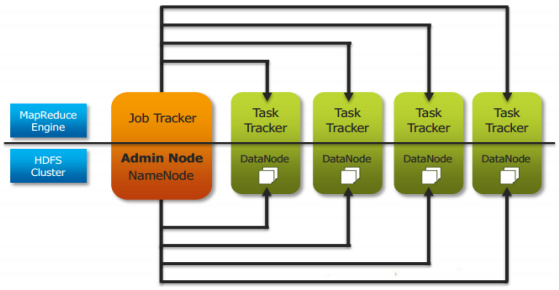

Fig.3 Architecture of Hadoop

XII. APPLICATIONS OF HADOOP

As of October 2009, commercial applications of Hadoop included:

- Log and/or clickstream analysis of various kinds
- Marketing analytics
- Machine learning and/or sophisticated data mining
- Image processing
- Processing of XML messages
- Web crawling and/or text processing

- General archiving, including of relational/tabular data, e.g. for compliance

XIII. PROMINENT USERS

Yahoo!

On February 19, 2008, Yahoo! Inc. launched what it claimed was the world's largest Hadoop production application. The Yahoo! Search Webmap is a Hadoop application that runs on a more than 10,000 core Linux cluster and produces data that was used in every Yahoo! web search query.[4]

XIV. COMPARISONS OF HADOOP WITH LEGACY TECHNOLOGIES

RDBMS EDW MPP NoSQL HADOOP

Structured	Data Types	Multi and Unstructured
Limited, No Data Processing	Processing	Processing coupled with Data
Standards & Structured	Governance	Loosely Structured
Required On Write	Schema	Required On Read
Reads are Fast	Speed	Writes are Fast
Software License	Cost	Support Only
Known Entity	Resources	Growing, Complexities, Wide
Interactive OLAP Analytics Complex ACID Transactions Operational Data Store	Best Fit Use	Data Discovery Processing Unstructured Data Massive Storage/Processing

XV. CONCLUSIONS

- It reduces traffic on capture, storage, search, sharing, analysis and visualization.
- A huge amout of data could be stored and large computations could be done in a single compound wth full safety and security at cheap cost.
- BIG DATA and BIG DATA- SOLUTIONS is one of the buring issues in the present IT Industry so, work on those will surely make you more useful to that.

REFERENCES

[1] www.webopedia.com
[2] www.wikipedia.com
[3] www.wikipedia.com
[4] www.edureka.com