SOFTWARE: PRACTICE AND EXPERIENCE
Softw. Pract. Exper. 2016; **46**:79–105
Published online 9 October 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.2374

The anatomy of big data computing

Raghavendra Kune[1,*,†], Pramod Kumar Konugurthi[1], Arun Agarwal[2],
Raghavendra Rao Chillarige[2] and Rajkumar Buyya[3]

[1] *Department of Space, Advanced Data Processing Research Institute, Hyderabad, India*
[2] *School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India*
[3] *CLOUDS Lab, Department of Computing and Information Systems, School of Engineering, University of Melbourne, Melbourne, Australia*

SUMMARY

Advances in information technology and its widespread growth in several areas of business, engineering, medical, and scientific studies are resulting in information/data explosion. Knowledge discovery and decision-making from such rapidly growing voluminous data are a challenging task in terms of data organization and processing, which is an emerging trend known as *big data computing*, a new paradigm that combines large-scale compute, new data-intensive techniques, and mathematical models to build data analytics. Big data computing demands a huge storage and computing for data curation and processing that could be delivered from on-premise or clouds infrastructures. This paper discusses the evolution of big data computing, differences between traditional data warehousing and big data, taxonomy of big data computing and underpinning technologies, integrated platform of big data and clouds known as big data clouds, layered architecture and components of big data cloud, and finally open-technical challenges and future directions. Copyright © 2015 John Wiley & Sons, Ltd.

Received 2 February 2015; Revised 7 June 2015; Accepted 2 September 2015

KEY WORDS: big data; big data computing; big data analytics as a service; big data cloud architecture

1. INTRODUCTION

Big data computing is an emerging data science paradigm of multidimensional information mining for scientific discovery and business analytics over large-scale infrastructure. The data collected/produced from several scientific explorations and business transactions often require tools to facilitate efficient data management, analysis, validation, visualization, and dissemination while preserving the intrinsic value of the data [1–5]. The IDC [6] report predicted that there could be an increase of the digital data by 40 times from 2012 to 2020. New advancements in semiconductor technologies are eventually leading to faster computing, large-scale storage, and faster and powerful networks at lower prices, enabling large volumes of data preservation and utilization at faster rate. Recent advancements in cloud computing technologies are enabling to preserve every bit of the gathered and processed data, based on subscription models, providing high availability of storage and computation at affordable price. Conventional data warehousing systems are based on predetermined analytics over the abstracted data and employ cleansing and transforming into another database known as data marts – which are periodically updated with the similar type of rolled-up data. However, big data systems work on non predetermined analytics; hence, no need of data cleansing and transformations procedures.

*Correspondence to: Raghavendra Kune, Department of Space, Advanced Data Processing Research Institute, Hyderabad, India.
†E-mail: raghav.es@gmail.com

Big data organizes and extracts the valued information from the rapidly growing, large volumes, variety forms, and frequently changing data sets collected from multiple and autonomous sources in the minimal possible time, using several statistical and machine learning techniques. Big data is characterized by five Vs such as volume, velocity, variety, veracity, and value. Big data and traditional data warehousing systems, however, have the similar goals to deliver business value through the analysis of data, but they differ in the analytics methods and the organization of the data. In practice, data warehouses organize the data in the repository, by collecting it from other several databases like enterprise's financial systems, customer marketing systems, billing systems, point-of-sale systems, and so on. Warehousing systems are poor on organizing and querying the data from the operational streaming data like click stream logs, sensor data, location data from mobile devices, customer support emails and chat transcripts, surveillance videos, and so on. Big data technologies overcome the weakness of the data warehousing systems, by harnessing new sources of data, thus facilitating enterprises analyze and extract intrinsic information through analytics. Big data technology has been gaining popularity in several domains of business, engineering, and scientific computing areas; Philip *et al.* [8] presented a survey on big data along with opportunities and challenges for data-intensive applications-stated several areas and the importance of big data. Chen *et al.* [9] presented a survey on big data and its interrelated technologies like clouds, Internet of things, online social networks, medical applications, collective intelligence, and smart grid. Chen *et al.* [10] presented the big data technologies towards data management challenges like big data diversity, big data reduction, integration and cleaning, indexing and query, and several tools for analysis and mining. Wu *et al.* [11] presented big data processing model, from the data mining perspective. Kaiser *et al.* [12] discussed several issues in big data such as storage and data transport technologies followed by methodologies for big data analytics. Buyya *et al.* [13] presented a survey on big data computing in clouds and future research directions for the development of analytics and visualization tools in several domains of science, engineering, and business.

As business domains are growing, there is a need to converge a new economic system redefining the relationships among producers, distributors, and consumers of goods and services. Obviously, it is not feasible to depend on experience or pure intuition always; however, it is also essential to use critically important data sources for decision-making. The National Institute of Standards and Technology Big Data Public Working Group described a survey of big data architectures and framework from the industry [14]. The several areas of big data computing are described in the succeeding texts.

(a) Scientific explorations: The data collected from various sensors are analyzed to extract the useful information for societal benefits. For example, physics and astronomical experiments – a large number of scientists collaborating for designing, operating, and analyzing the products of sensor networks and detectors for scientific studies. Earth observation systems – information gathering and analytical approaches about earth's physical, chemical, and biological systems via remote-sensing technologies – to improve social and economic well-being and its applications for weather forecasting, monitoring, and responding to natural disasters, climate change predictions, and so on.

(b) Health care: Healthcare organizations would like to predict the locations from where the diseases are spreading so as to prevent further spreading [15]. However, to predict exactly the origin of the disease would not be possible, until there is statistical data from several locations. In 2009, when a new flu virus similar to H1N1 was spreading, Google has predicted this and published a paper in the scientific journal *Nature* [16], by looking at what people were searching for, on the Internet.

(c) Governance: Surveillance system analyzing and classifying streaming acoustic signals, transportation departments using real-time traffic data to predict traffic patterns, and update public transportation schedules. Security departments analyzing images from aerial cameras, news feeds, and social networks or items of interest. Social program agencies gain a clearer understanding of beneficiaries and proper payments. Tax agencies identifying fraudsters and support investigation by analyzing complex identity information and tax returns. Sensor applications such stream air, water, and temperature data to support cleanup, fire prevention, and other programs.

(d) Financial and business analytics: Retaining customers and satisfying consumer expectations are among the most serious challenges facing financial institutions. Sentiment analysis and predictive analysis would play a key role in several fields like travel industry – for optimal cost estimations and retail industry – products targeted for potential customers. Forecast analysis – estimating the best price estimations and so on.

(e) Web analytics: Several websites are experiencing millions of unique visitors per day, in turn creating a large range of content. Increasingly, companies want to be able to mine this data to understand limitations of their sites, improve response time, offer more targeted ads, and so on. This requires tools to perform complicated analytics on data that far exceed the memory of a single machine or even in cluster of machines.

Service-oriented technologies also known as cloud computing are delivering compute, storage, and software applications as services over private or public networks based on pay-as-go delivery models [17, 18]. Cloud computing technologies becoming a reality, it is serving as a key enabler for big data to solve data-intensive problems over a large-scale infrastructure for information extraction. The integration of big data technologies and cloud computing read as 'big data clouds' is an emerging new generation data analytics platform for information mining, knowledge discovery, and decision-making. Hence, both the technologies put together, here, we discuss the evolution of big data technologies and compare it with traditional data warehousing technologies along with its relationship with cloud computing technologies and infrastructure. We also discuss the architecture and reference framework for big data computing on clouds. This paper is intended for researchers, technical audience of both developer and designers, and general readers who are interested in acquiring an in-depth knowledge of big data technology in information technology.

The rest of the paper is organized as follows. Section 2 describes the differences between big data and traditional data warehousing systems in, data handling, processing, storing, extracting, and so on followed by consistency, availability, and partition tolerance (CAP) theorem – the fundamental principle of database system – and illustrates the Atomicity, consistency, integrity, and durability (ACID) and basically available, soft state, and eventually consistent (BASE) properties adopted for data warehousing (relational model) and big data models, respectively. Later, we discuss the big data abstraction layers and compare it with the traditional data base model. Section 3 discusses taxonomy of big data computing and presents a detailed study of several components of the taxonomy like analytics, frameworks, technologies, programming models, schedulers, processing tools, and so on. Section 4 illustrates an integrated platform for big data and clouds followed by a layered architecture and its components. Here, we discuss elements of big data cloud, layered architecture, and reference framework. Section 5 identifies open-technical challenges, gap analysis, and future research directions in data storage/handling, specific domain areas, new programming models, and domain-specific analytics development.

2. BIG DATA CHARACTERISTICS – TRADITIONAL DATA VERSUS BIG DATA PARADIGMS

Big data refers to large-scale data architectures and facilitates tools addressing new requirements in handling data volume, velocity, and variability. Traditional databases (data warehousing) assume data are organized in rows and columns and employ data-cleansing methods on the data, while the data volumes grow over a time period and often lack on handling such large-scale data processing. Traditional data base/warehousing systems were designed to address smaller volumes of structure data, with the predictable updates and consistent data structure, which mostly operate on single server and lead to operational expenses with the increased data volume. However, big data comes in a variety of diverse formats with both batch and stream processing in several areas such as geospatial data, 3D data, audio and video, structured data, unstructured text including log files,

DOI: 10.1002/spe

sensor data, and social media. Below, we discuss the properties of traditional database (data warehousing) and big data.

2.1. Traditional database (both operational online transaction processing and warehousing online analytical processing data)

Inmo [19] described data warehousing as subject-oriented, integrated, time-variant, and nonvolatile collection of data, and helping analysts in decision-making process. Data warehouse is segregated from the organization's operational database. The operational database undergoes the per day transactions (online transaction processing) that causes the frequent changes to the data on a daily basis. Traditional databases typically address the applications for business intelligence, however, lack in providing the solutions for unstructured large volumes rapidly changing analytics in business and scientific computing. The several processing techniques under data warehouse are described in the succeeding texts.

- Analytical processing involves analyzing the data by means of basic online analytical processing operations, including slice-and-dice, drill down, drill up, and pivoting.
- Knowledge discovery through mining techniques by finding the pattern and associations, constructing analytical models, and performing classification and prediction. These mining results can be presented using visualization tools.

2.2. Big database

Big data addresses the data management and analysis issues in several areas of business intelligence, engineering, and scientific explorations. Traditional databases segregate the operational and historical data for operational and analysis reasoning, which are mostly structured. However, big data bases address the data analytics over an integrated scale out compute and data platform for unstructured data in near real time. Figure 1 depicts several issues in traditional data (data warehousing online transaction processing/online analytical processing) and big data technologies that are classified into major areas like infrastructure, data handling, and decision support software as described in the succeeding texts.

- Decision support/intelligent software tools: Big data technologies address various decision supporting tools for searching the large data volumes and construct the relations and extract the information based on several analytical methods. These tools would address several machine-learning techniques, decision support systems, and statistical modeling tools.
- Large-scale data handling: rapidly growing data distributed over several storages and compute nodes with multidimensional data formats.

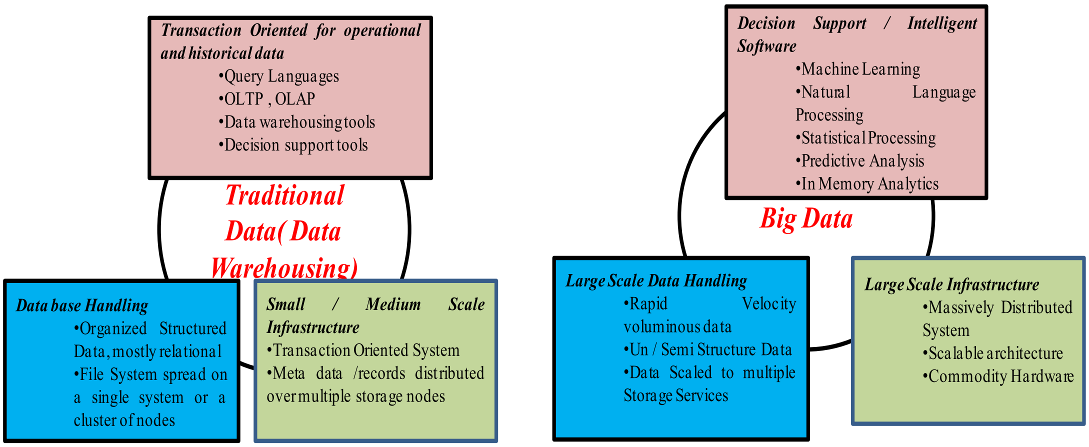
Transaction Oriented for operational and historical data
•Query Languages
•OLTP , OLAP
•Data warehousing tools
•Decision support tools
Traditional Data(Data Warehousing)
Data base Handling
•Organized Structured Data, mostly relational
•File System spread on a single system or a cluster of nodes
Small / Medium Scale Infrastructure
•Transaction Oriented System
•Meta data /records distributed over multiple storage nodes
Decision Support / Intelligent Software
•Machine Learning
•Natural Language Processing
•Statistical Processing
•Predictive Analysis
•In Memory Analytics
Big Data
Large Scale Data Handling
•Rapid Velocity voluminous data
•Un / Semi Structure Data
•Data Scaled to multiple Storage Services
Large Scale Infrastructure
•Massively Distributed System
•Scalable architecture
•Commodity Hardware

Figure 1. Big data versus traditional data (data warehousing) models. OLTP, online transactional processing; OLAP, online analytical processing.

DOI: 10.1002/spe

Table I. Traditional data warehousing versus big data issues

Serial nos.	Property	Traditional data warehousing	Big data-specific issues
1	Data volume	Data are segregated into operational and historical data. Applies extract, transformation, and load mechanisms for processing. As the data volumes are increased, the historical data are filtered from warehouse system for faster database queries.	High volume of data from several sources like web, sensor networks, social networks, and scientific experiments. Capable of handling operational and historical data together, which could be replicated on multiple storage devices for high availability and throughput.
2	Speed	Transaction-oriented and the data in turn generated from the transactions are low.	High data growth due to several sources like web and scientific sensors streaming experiments.
3	Data formats	Semi/structured data like XML and relational.	Multi-structured data handling such as relational, and un/semi-structured such as text, XML, video streaming, and so on.
4	Applicable platforms	Online transactional processing, relational database management system.	Big data analytics, text mining, video analytics, web log mining, scientific data exploration, intrinsic information extractions, graph analytics, social networking, in-memory analytics, and statistical and predictive analytics.
5	Programming methodologies/ languages	Query language like SQL.	Data-intensive computing languages for batch processing and stream computing like Map/Reduce and NoSQL programming.
6	Data backup/archival	Files/relational data need to have data backup procedures or mechanisms. Traditional data works on regular, incremental, and full backup mechanisms that are already established.	Due to large and high speeds of the data growth rates, the conventional methods are not adequate; hence, techniques such as differential backup mechanisms need to be explored.
7	DR	Data are replicated at several places to address the disaster.	DR techniques could be separated from mission critical and non critical data.
8	Relationship with clouds	Relational data bases/data warehousing tools as services over cloud infrastructures.	On-demand big data infrastructure setup, analytic services by several cloud, and big data providers.
9	Data deduplication	Applicable to transactional record deduplication while merging database records.	File and block level deduplication mechanisms need to be explored for continuous growing and stream-oriented data.
10	System users	Administrators, developers, and end users.	Data scientists and analytics end users.
11	Theorem applicable	Follows CAP theorem [20] with ACID [21] properties.	Follows CAP theorem with BASE properties [22].

DR, disaster recovery; SQL, structured query language; BASE, basically available, soft state, and eventually consistent; CAP, consistency, availability, and partition tolerance.

- Large-scale infrastructure: scale out infrastructure for efficient storage and processing.
- Batch and stream support: capability to handle both batch and stream computation.

Table I illustrates properties of big data versus traditional data warehousing computing.

2.3. *CAP theorem – ACID and basically available, soft state, and eventually consistent*

Traditional databases follow ACID [21, 23] properties, which are the primary standards for relational databases. However, distributed computing systems follow BASE [22] properties to address loss of consistency and reliability as discussed in the succeeding texts:

DOI: 10.1002/spe

- Basically available: This property states that the system guarantees the data availability; however, during the transition/changing state, the response would be either delayed or may fail in obtaining the requested data. This scenario is similar as depositing a check in your bank account, and waiting until the check goes through the clearing house, for having the funds made available.
- Soft state: The state of the system would change over time, so even during times without input, there may be changes going on due to eventual consistency; thus, state of the system is always soft.
- Eventual consistency: The system would propagate the data as it is receiving; however, it will not ensure the consistency of the data for every transaction. The data would be eventually consistent, whenever it stops receiving the input.

In 2000, Eric Brewer presented the CAP theorem, also known as Brewer's theorem [20], for the successful design, implementation, and deployment of applications in distributed computing systems. The CAP theorem states that, the partition tolerance networked shared-data system can provide either consistency or availability, but not both, as mentioned in the succeeding texts.

- Consistency: Similar to the consistency property of ACID, the data are synchronized across all cluster nodes, and all the nodes would see the similar data at the same time. It means a read sees all completed writes.
- Availability: Guaranteed that every request receives a response, means every read and writes always succeed. But, the data would get eventually consistent within the system.
- Partition tolerance: Single node failure should not cause the entire system to fail, and the system should continue to function even under circumstances of arbitrary message loss or partial failure of the system.

Big data system adopts Brewer's CAP theorem on the similar lines of BASE. The CAP theorem with ACID and BASE is depicted in Figure 2.

2.4. *Big data – abstraction layers*

Big data and the traditional/relational data layers are depicted in Figure 3.

Traditional data layers are in general classified into three layers of abstraction, physical layer, logical layer, and view/user layer. The function of each layer is described in the succeeding texts.

- Physical layer: It describes the lowest level of abstraction and uses low-level complex data structures for data storage, for example, B + −tree organization, R-Tree indexing mechanisms, and so on.
- Logical layer: It describes the type of data stored in a database, and the relationships among the data. In general, data base administrators' work at the logical level of abstraction. The several activities by this layer are data base design, tuning for better performance, tools for data base backup, and so on.

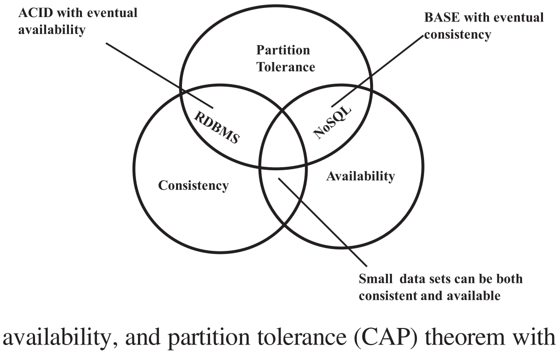
ACID with eventual availability
BASE with eventual consistency
Partition Tolerance
RDBMS
NoSQL
Consistency
Availability
Small data sets can be both consistent and available

Figure 2. Consistency, availability, and partition tolerance (CAP) theorem with Atomicity, consistency, integrity and durability and basically available, soft state, and eventually consistent (BASE) (source: National Institute of Standards and Technology [24]). RDBMS, relational database management system.

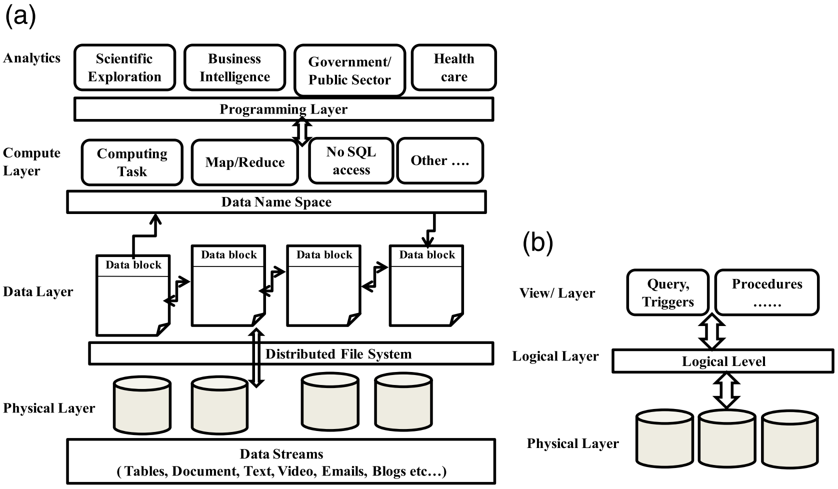

Figure 3. Data Abstraction - (a) Big Data (b)traditional data. SQL, Structured Query Language; NoSQL, Not Only SQL.

- View layer: The highest level of abstraction hides all low-level complexities, details of the data types, and offers programming tools for query and processing.

Big data abstraction adopts four-layered abstraction model; the layers from the bottom-up approach are physical layer, data layer, computing layer, and data analytics layers, respectively. Physical layer takes care of the data organization over several distributed data storage, high speed networks, and partitioned cluster of nodes. Data layer addresses the global namespace for the data access and logical expansion of the data, without knowing the underlying physical layer structure. Compute layer offers several computing methodologies, and analytic offers several technologies for analysis of the data for decision-making. The role of each layer is described in the succeeding texts.

- Physical data layer: Big data addresses several forms/types of data with a horizontally scalable infrastructure for redundancy, high performance data transfer, and efficient support for computation. This layer addresses the properties mentioned against serial numbers 1, 2, and 3 in Table I.
- Data layer: This layer provides an abstraction over the physical data layer and offers the core functionalities such as data organization, access, and retrieval of the data. This layer indexes the data and organizes them over the distributed store devices. The data are partitioned into blocks and organized onto the multiple repositories. The data to organize can be anyone of several forms; hence, several tools and techniques are employed for effective organization and retrieval of the data. The examples include key/value pair, column-oriented data, document database, relational database, semi-structured XML data, raw formats, and so on. This layer refers to the properties 3, 6, and 10 mentioned in Table I.
- Computing layer: software abstraction layer of data modeling and query and domain-specific programming application programming interfaces (APIs) to retrieve the data from its below data model layer. For example, this layer offers tools like NoSQL programming, and MapReduce for data-intensive computing, domain-specific statistical models, machine-learning techniques, and so on. This layer refers to the properties 7, 9, and 16 of Table I.
- Analytics: standards and techniques for developing the domain-specific analytics tools using the tools of software abstraction layer.

3. BIG DATA TAXONOMY

For years, several organizations are capturing the transactional-structured data using traditional–relational data bases using transactional query processing [1, 6, 25] for the information extraction.

DOI: 10.1002/spe

In recent years, technologies are evolving to perform the investigations on the whole data using distributed computing and storage technologies such as MapReduce, distributed file systems, and in-memory computing [26] with highly optimized capabilities for different business and scientific purposes. The advancements in storage capacity, data handling and processing tools, and the analysis of data can be carried out in real time or close to real time, acting on full data sets rather than on the summarized elements, leveraging tools and technologies enough to address the issue. In addition, the number of options to interpret and analyze the data has also increased, with the use of various visualization technologies. In the succeeding texts, we describe the taxonomy of big data depicted in Figure 4. The several elements of the taxonomy are described in the succeeding texts.

(i) Big data dimensions

Big data is characterized into four dimensions called four Vs, volume, velocity, variety, and veracity, as depicted in Figure 5. Aside from that, another dimension V (value/valor) also is used to characterize the quality of the data.

- Volume: Volume is concerned about the scale of data, that is, the volume of the data at which it is growing. According to IDC [6] report, the volume of data will reach to 40 Zeta bytes by 2020 and increase of 400 times by now. The volume of data is growing rapidly, because of several applications of business, social, web, and scientific explorations.

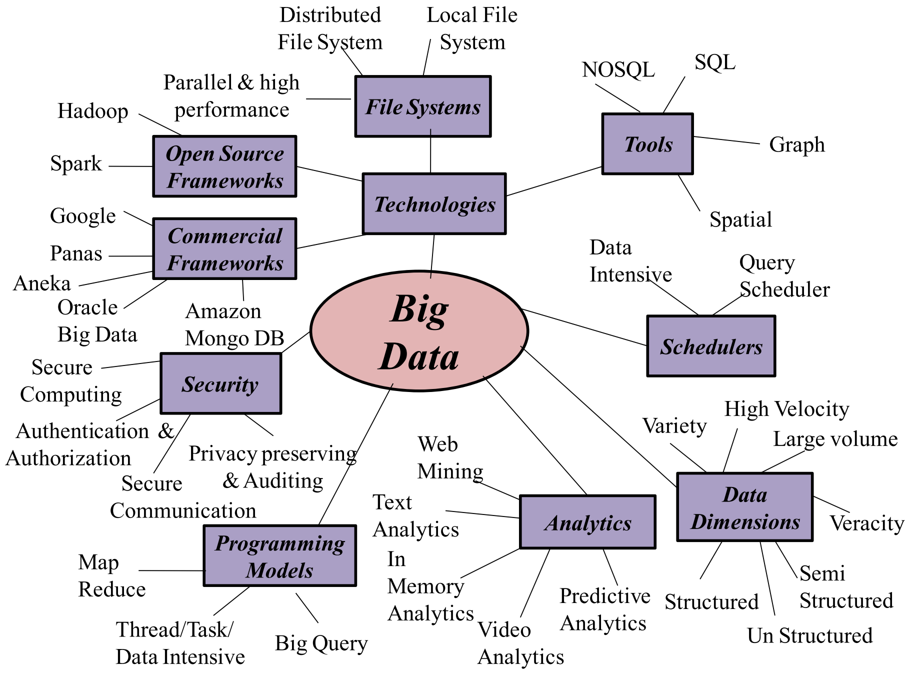

Figure 4. Big Data taxonomy. SQL, Structured query language; NoSQL, Not Only SQL.

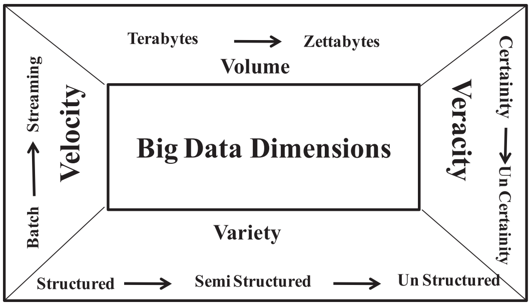

Figure 5. Data dimensions four Vs.

DOI: 10.1002/spe

- Velocity: The speed at which the data are increasing thus demanding analysis of streaming data. The velocity is due to the growing speed of business intelligence applications such as trading, transaction of telecom and banking domain, growing number of Internet connections with the increased usage of Internet, and growing number of sensor networks and wearable sensors.
- Variety: It depicts different forms of data to use for analysis such as structured-like relational databases, semi-structured-like XML, and unstructured-like video and text.
- Veracity: Veracity is concerned with uncertainty or inaccuracy of the data. In many cases, the data will be inaccurate hence filtering and selecting the data that are actually needed are really a cumbersome activity. Many statistical and analytical processes have to go for data cleansing for choosing intrinsic data for decision-making.

(ii) Analytics – big data techniques

Analytics is the process of analyzing the data using statistical models, data-mining techniques, and computing technologies. It combines the traditional analysis techniques and mathematical models to derive information. Analytics and analysis perform the same function; however, analytics is the application of science to analysis. Big data analytics refers to a set of procedures and statistical models to extract the information from a large variety of data sets. A few major big data analytics application areas are discussed in the succeeding texts.

- Text analytics: The process [45] of deriving information from text sources. The text sources forms of semi-structured data that include web materials, blogs, and social media postings (such as tweets). The technology within text analytics comes from fundamental fields including linguistics, statistics, and machine learning. In general, modern text analytics uses statistical models, coupled with linguistics theories, to capture patterns in human languages such that machines can understand the meaning of texts and perform various text analytics tasks. Text mining in the area of sentiment analysis helps organizations uncover sentiments to improve their customer relationship management.
- In-memory analytics: In-memory analytics [26] is the process that ingests the large amounts of data from a variety of sources directly into the system memory for efficient query and calculation performance. In-memory analytics is an approach for querying data when it resides in a computer's random access memory, as opposed of querying data stored in physical disks. This results in vastly shortened query response times, allowing business intelligence applications to support faster business decisions.
- Predictive analysis: Predictive analysis [46] is the process of predicting future or unknown events with the help of statistics, modeling, machine learning, and data mining by analyzing current and historical facts.
- Graph analytics: Graph analytics [44] studies the behavior analysis of various connected components, especially useful in social networking websites to find the weak or strong groups.

(iii) Technologies

Big data technologies are majorly classified into three parts, namely, (i) file system – effective way of organizing the data; (ii) computing frameworks; and (iii) tools for analytics as described in the succeeding texts.

(a) File system: File system is responsible for the organization, storage, naming, sharing, and protection of files. Big data file management is similar to distributed file system; however, the read/write performance, simultaneous data access, on-demand file system creation, and efficient techniques for file synchronizations would be major challenges for design and implementation. The goals in designing the big data file systems should include certain degree of transparency as mentioned in the succeeding texts.

- Distributed access and location transparency: Unified directory services, clients are unaware that files are distributed and can access them in the same way local files are accessed. Consistent name space encompassing local as well as remote files without any location information.

DOI: 10.1002/spe

- Failure handling: The application programs and the client should operate even with the few components failures in the system. This can be achieved with some level of replication and redundancy.
- Heterogeneity: File service should be provided across different hardware and operating system platforms.
- Support fine-grained distribution of data: To optimize performance, we may wish to locate individual objects near the processes that use them.
- Tolerance for network partitioning: The entire network or certain segments of it may be unavailable to a client during certain periods (e.g., disconnected operation of a laptop). The file system should be tolerant enough to handle the situations and applies the appropriate synchronization mechanisms.

(b) Open-source frameworks: Big data computing frameworks that are based on open-source frameworks are described in the succeeding texts.

- Apache Hadoop [7]: An open-source reliable, scalable, and distributed computing platform. It offers a software library and framework that allow distributed processing of large-scale distributed processing of large data sets across clusters of computers using simple programming models.
- Spark [27]: Apache Spark is a fast and general engine for large-scale data processing. This covers Shark structured query language, Spark Streaming, MLib machine learning, and Graphx graph analytics tools. Spark can run on Hadoop YARN [7] cluster manager and can read any existing Hadoop data.
- Storm [28]: distributed real-time stream-oriented computing for real-time analytics, online machine learning, continuous computation, distributed Remote Procedure Call (RPC), extract, transformation, and load, and so on. Storm topology consumes streams of data and processes those streams in arbitrarily complex ways, repartitioning the streams between each stage of the computation.
- S4 [29]: platform for processing continuous data streams. S4 is designed specifically for managing data streams. S4 apps are designed combining steams and processing elements in real time.

(c) Commercial frameworks

Google offers big query [30] to operate on Google big tables [31]. Amazon supports big data through Hadoop cluster and also NoSQL support of columnar database using Amazon DynamoDB [32]. Amazon elastic MapReduce [33] is a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. Windows offers HDInsight [34] service that is an implementation of Hadoop that runs on the Microsoft Azure platform. RackSpace [35] offers Horton Hadoop framework on Openstack platform, Aneka [39] offers .NET-based desktop MapReduce platform, and other enterprise frameworks based on open-source Hadoop are Horton [36] and Cloudera [37].

(d) Tools

The brief descriptions of the several big data tools are described in the succeeding texts.

- Key-value stores: Key-value pair (KVP) tables are used to provide persistence management for many NoSQL technologies. The concept is that the table has two columns – one is the key; the other is the value. The value could be a single value or a data block containing many values; the format of which is determined by program code. KVP tables may use indexing and have tables or sparse arrays to provide rapid retrieval and insertion capability, depending on the need for fast lookup, fast insertion, or efficient storage. KVP tables are best applied to simple data structures and on the Hadoop MapReduce environment. Examples of key-value data stores are Amazon's Dynamo [32] and Oracle's Berkeley DB [38].
- Document-oriented database: A document-oriented database is a database designed for storing, retrieving, and managing document-oriented or semi-structured data. The central concept of a document-oriented database is the notion of a document where the contents within the document are encapsulated or encoded in some standard format such as JavaScript object notation, binary JavaScript object notation, or XML. Examples of these databases are Apache's CouchDB [40] and 10gen's MongoDB [41].

DOI: 10.1002/spe

- Column family/big table database: Instead of storing key-values individually, they are grouped to create the composite data, each column containing the corresponding row number as key and the data as value. This type of storage is useful for streaming data such as web logs, time series data coming from several devices, sensors, and so on. The examples are HBase [42] and Google big table [31].
- Graph database: A graph database uses graph structures similar to nodes, edges, and properties for data storing and semantic query on the data. In a graph database, every entity contains direct pointers to its adjacent element, and index lookups are not required. A graph database is useful when large-scale multi-level relationship traversals are common and desirable for processing complex many-to-many connections such as social networks. A graph may be captured by a table store, which supports recursive joins such as big table and Cassandra. Examples of graph databases include infinite graph [43] from objectivity and the Neo4j open-source graph database [44].

(iv) Programming models: various programming models like data intensive, stream computing, batch processing, high performance/throughput, query processing, and column-oriented data processing are described in the succeeding texts.

- MapReduce: data-intensive programming model, with high-level programming constructs for Map and Reduce functions in the cluster of distributed compute and storage nodes. Map function performs filtering and sorting, whereas Reduce function aggregates Map output to generate the final result. MapReduce programming is a type of recursive programming model to operate the similar logic on multiple distributed data sets. The examples are Hadoop MapReduce [47], Apache Spark [27], and Aneka MapReduce [17].
- Thread/task data-intensive models: Thread programming models are used for high-performance applications; the computing logic demands more computing elements or high-end cores for processing within to meet the application deadlines, and task programming models are used for workflow programming models, for example, Aneka [17].
- Machine-learning tools: new generation of machine-learning tools for decision-making. Few tools available are Hadoop Mahout [48].
- Big query languages: New generation of query languages, examples are Google big query [30]. Web log mining is the study of the data available in the web. This involves searching for the texts, words, and their occurrences. One example for web log mining is searching for the words, and their frequencies by Google big query data analytics [30] use Google big query platform to run on the Google cloud infrastructure.

Big data computing majorly needs to address two types of scheduling mechanisms, query scheduler and data-aware scheduling. Query schedulers address several mechanisms for querying the data managed by big data systems. Data-intensive schedulers address several computing mechanisms; examples include capacity scheduler [49] and fair scheduler [50].

(v) Big data security

Big data project can uncover tremendous value for an enterprise, by revealing customer buying habits, detecting or preventing fraud, or monitoring real-time events. However, a poorly run big data project can be a security and compliance nightmare, leading to data breaches. Big data must be protected, to ensure that only the right people have appropriate access to it. Big data security addresses several mechanisms for large-scale high-volume rapidly growing varied forms of data, analytics, and large-scale compute infrastructure. As the data volumes and compute infrastructures are very large, traditional methods of computing and data security mechanisms, which are tailored for securing small-scale data and infrastructure, are inadequate. Also, the use of large-scale cloud infrastructures, with a diversity of software platforms, spreads across large networks of computers and also increases the attacks. The onion model of defense for big data security is depicted in Figure 6, and the several elements are described in the succeeding texts.

DOI: 10.1002/spe

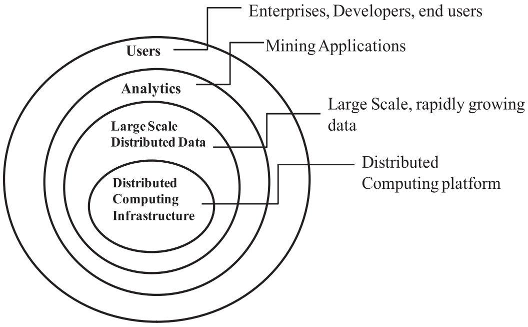
Users
Analytics
Large Scale Distributed Data
Distributed Computing Infrastructure
Enterprises, Developers, end users
Mining Applications
Large Scale, rapidly growing data
Distributed Computing platform

Figure 6. Big data security onion model of defense.

- Distributed computing infrastructure: mechanisms for providing security while data are analyzed over multiple distributed systems. Big data setup would be either confined to an enterprise or could be a large collection of several enterprises, social, and scientific collection of disparate sources distributed system. Privacy, security, and confidentiality – not revealing private and confidential information to unauthorized users. For example, in a mailing system, secrecy is concerned about preventing the users from finding out the passwords of other users.
- Large-scale distributed data: privacy-preserving mechanisms, encryption techniques for the data stored on large-scale distributed systems, role-based access and control mechanisms, and security of column, document, key-value, and graph data models to be evolved. In order to maintain fast access for the data, NoSQL databases come with little built-in security; due to their BASE properties, rather than requiring consistency after every transaction, the data base just needs to eventually reach a consistent state.
- Analytics security: developing frameworks that are secured that allow organizations for publish and use the analytics securely based on several authentication mechanisms such as one-time passwords, multi-level authentications, and role-based access mechanisms.
- Users' privacy and security: confidentiality, integrity, and authentication mechanisms to validate the users.

4. BIG DATA IN CLOUDS: AN INTEGRATED BIG DATA AND CLOUD PLATFORM

Big data in clouds is a new generation data-intensive platform for quickly building the analytics and deploying over an elastically scalable infrastructure. Based on the services rendered to the end users, these are broadly classified into four types as described in the succeeding texts.

- Public big data clouds: large-scale data organization and processing over the elastically scalable clouds infrastructure. The resources are served over Internet as pay-as-go computing models. The examples include Amazon big data computing in clouds [33], Windows Azure HDInsight [34], RackSpace Cloudera Hadoop [35, 37], and Google cloud platform of big data computing [30].
- Private big data clouds: deployment of big data platform within the enterprise over a virtualized infrastructure, with a greater control and privacy to the single organization.
- Hybrid big data clouds: federation of public and private big data clouds for scalability, disaster recovery, and high availability. In this deployment, the private tasks can be migrated to the public infrastructure during peak workloads.
- Big data access networks and computing platform: integrated platform of data, computing, and analytics delivered as services by multiple distinct providers.

Big data computing in clouds also known as 'big data clouds' is data-intensive analytics platform of large-scale, distributed compute, and storage infrastructures. The features of big data clouds are as follows: (i) large-scale distributed compute and data storages: wide range of computing facilities with seamless access to scalable storage repositories and data services; (ii) information-defined data storage: metadata-based data access instead of path and filenames; (iii) distributed virtual file system: File system could be dynamically created and mapped to the computing cluster; (iv) seamless access of computing and data: transparent access to large-scale data and compute resources; (iv) dynamic selection of data containers and compute resources: able to handle dynamic creation of virtual machines and able to access large-scale distributed data sources increasing the data location proximity; (v) high performance data and computation: Compute and data should be high-performance driven; (vi) multidimension data handling: support for several forms of data with necessary tools for processing; (vii) analytics platform services: able to develop, deploy, and use analytics over the environment; (viii) high availability of computing and data: replication mechanisms for both computing and data; and (ix) platform for data-intensive computing: support for both traditional and emerging data-intensive computing models and scalable deployment and execution of applications.

Figure 7 depicts integrated cloud and big data access networks on cloud infrastructure for analytics development. The content from several sources like social media, web logs, scientific studies, sensor networks, business transactions, and so on are growing rapidly. Deriving useful information for decision-making from such large data, fusing the information from several sources would be a challenging task. The elements of big data access network are as follows: data services, big data computing platform, data scientist, and computing cloud described in the succeeding texts.

- Data and platform services: Several providers, those who provide services for accessing both data and platform services for computing on the data, for example, Google data APIs (GData) [51], provide protocols for reading and writing data on the web for several services like content API for shopping, Google analytics, spreadsheets, and YouTube.
- Big data computing platform: platform for managing the various data sources including data management, access, programming models, schedulers, security, and so on. The platform includes various tools for accessing other data platforms using streaming, web services, and APIs. Other data platforms include data services from relational data stores, Google data, social networking, and so on.
- Data scientist: analytics developers having access to the computing platform.
- Computing cloud: computing infrastructure from private/public/hybrid clouds.

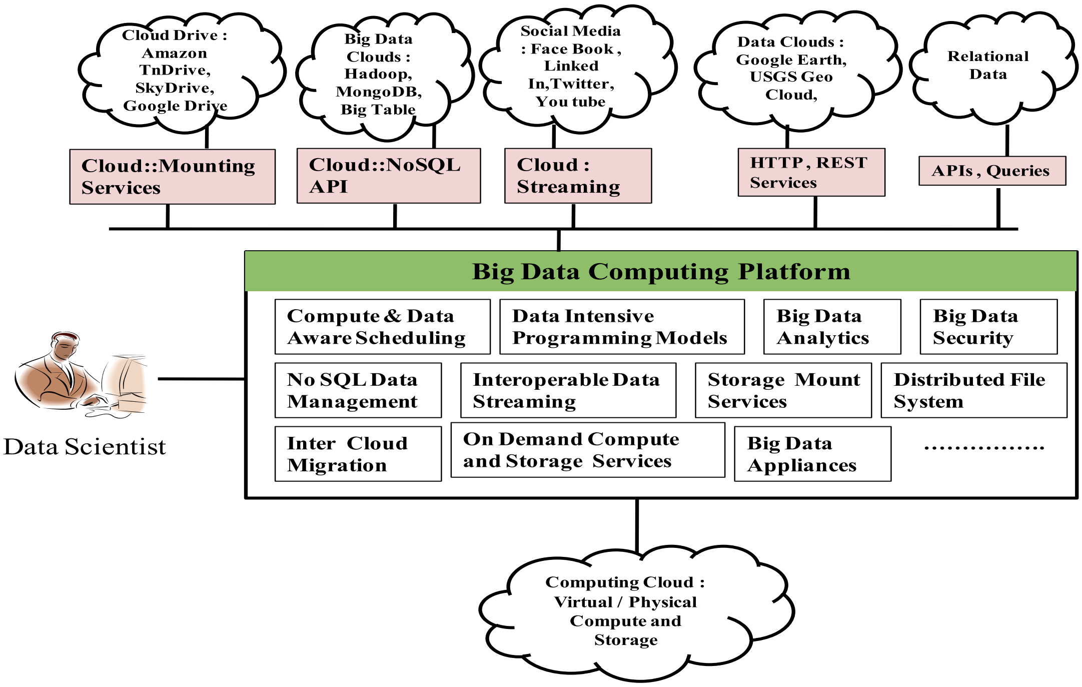
Cloud Drive : Amazon TnDrive, SkyDrive, Google Drive
Big Data Clouds : Hadoop, MongoDB, Big Table
Social Media : Face Book , Linked In,Twitter, You tube
Data Clouds : Google Earth, USGS Geo Cloud,
Relational Data
Cloud::Mounting Services
Cloud::NoSQL API
Cloud : Streaming
HTTP , REST Services
APIs , Queries
Big Data Computing Platform
Compute & Data Aware Scheduling
Data Intensive Programming Models
Big Data Analytics
Big Data Security
No SQL Data Management
Interoperable Data Streaming
Storage Mount Services
Distributed File System
Inter Cloud Migration
On Demand Compute and Storage Services
Big Data Appliances
……………
Data Scientist
Computing Cloud : Virtual / Physical Compute and Storage

Figure 7. Integrated cloud and big data compute network. USGS, United States Geological Survey; HTTP, hypertext transfer protocol; REST, representational state transfer.

DOI: 10.1002/spe

4.1. *Big data clouds for the enterprise*

Big data clouds enable enterprises to save money, grow revenue, and achieve many other business objectives in any vertical by quickly building their big data databases and writing analytics for mining the information. The benefits of big data clouds for the enterprises are mentioned in the succeeding texts.

- Build new applications: Big data clouds would allow enterprises to collect billions of real-time data points on its products, resources, or customers and then repackage that instantaneously to optimize customer experience or resource utilization.
- Improve the effectiveness and minimize the cost: Big data clouds offer services and pay-as-go consumption model similar to cloud services. This pricing model would effectively reduce both the cost of the applications development by minimizing the cost of development tools.
- Realize new sources of information and build applications to gain competitive advantage: The information could be quickly fused from several big data databases and rapidly build applications for several platforms like hand-held and mobile devices.
- Increase in customer loyalty: Increase in the amount of data sharing within the organization and the speed with which it is updated allows businesses and other organizations to more rapidly and accurately respond to customer demand.

4.2. *Elements of big data cloud*

Big data and traditional data warehousing mechanisms differ with each other in several ways like large-scale data organization, and querying followed by platforms and tools to the data scientists for analytics development. In this section, we describe elements of big data cloud as shown in Figure 8.

(i) Big data infrastructure services: This layer offers core services such as compute, storage, and data services for big data computing as described in the succeeding texts.

(a) Basic storage service: provides basis services for data delivery that is organized either on physical or virtual infrastructure and supports various operations like create, delete, modify, and update with a unified data model supporting various types of data.
(b) Data organization and access service: Data organization provides management and location of data resources for all kinds of data, and selection, query transformation, aggregation and representation of query results, and semantic querying for selecting the data of interest.
(c) Processing service: mechanism to access the data of interest, transferring to the compute node, efficient scheduling mechanism to process the data, programming methodologies, and various tools and techniques to handle the variety of data formats.

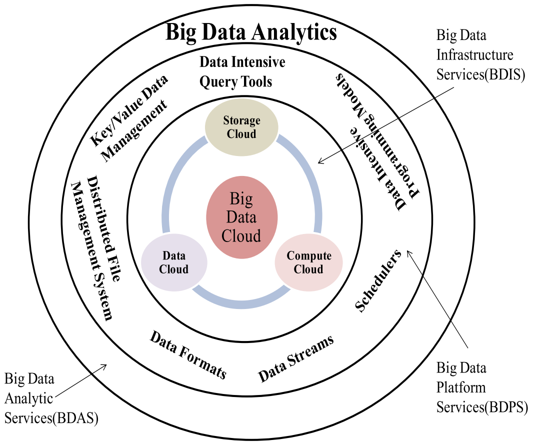
Big Data Analytics
Data Intensive Query Tools
Key/Value Data Management
Data Intensive Programming Models
Distributed File Management System
Schedulers
Data Formats
Data Streams
Storage Cloud
Big Data Cloud
Data Cloud
Compute Cloud
Big Data Infrastructure Services(BDIS)
Big Data Platform Services(BDPS)
Big Data Analytic Services(BDAS)

Figure 8. Big data cloud components.

DOI: 10.1002/spe

The elements of big data infrastructure services are described in the succeeding texts.

- Computing clouds: on-demand provisioning of compute resources, which could expand or shrink based on the analytics requirements.
- Storage cloud: large volume of storage offered over the network. The storages offered include file system, block storages, and object-based storage. Storage clouds offer to create file system of choice and also elastically scalable. Storage clouds can be accessed based on the pricing models that are usually based on data volumes and transactions/data transfer. The several services offered by storage clouds are raw, block, and object-based storages.
- Data clouds: Data clouds are similar to storage clouds; however, unlike storage space delivery, they offer data as a service. Data clouds offer tools and techniques to publish the data, tag the data, discovery the data, and process the data of interest. Data clouds operate on domain-specific data leveraging the storage clouds to serve data as a service based on the four steps of 'standard scientific model' [40] such as data collection, analysis, analyzed reports, and long-term preservation of the data.

(ii) Big data platform services: This layer offers schedulers, query mechanisms for data retrieval, and data-intensive programming models to address several big data analytic problems.

(iii) Big data analytics services: big data analytics as services over big data cloud infrastructure. The services would be offered to enterprises based on service-level agreements (SLAs) meeting QoS parameters.

4.3. Big data clouds-layered architecture

The architecture of big data computing in clouds is represented as four-layered model as shown in Figure 9. The cloud infrastructure layer handles the elastic scalable computing, storage, and networking infrastructure. The big data fabric layer addresses the several tools for data management, access, and aggregation. The third layer is the platform layer that addresses the tools and technologies for data access and processing, programming environments for designing the analytics and scheduling models for execution, and so on; the top layer is the big data analytics, focused on analytics usage, and publishing standards to offer them as services. The functional description of each of the layers is described in the succeeding texts.

(a) Cloud infrastructure: large-scale management of dynamic and elastic scalable large infrastructure of compute and storage resources as services. Virtualization technologies are used for on-demand provisioning of the resources based on SLAs and QoS parameters. The services rendered by this layer are as follows: (i) large-scale elastic infrastructure to set up big data

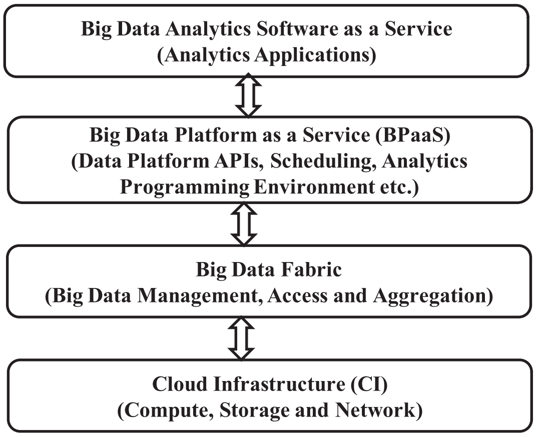
Big Data Analytics Software as a Service
(Analytics Applications)
Big Data Platform as a Service (BPaaS)
(Data Platform APIs, Scheduling, Analytics Programming Environment etc.)
Big Data Fabric
(Big Data Management, Access and Aggregation)
Cloud Infrastructure (CI)
(Compute, Storage and Network)

Figure 9. Big data cloud reference architecture.

DOI: 10.1002/spe

platform on demand; (ii) dynamic creation of virtual machines; (iii) large-scale data management for file/block/objected-based storages on demand; (iv) ability to move the data in seamless across the storage repositories; and (v) able to create the virtual machines and auto mount the file system with the compute node.

(b) Big data fabric: This layer addresses tools and APIs through which storage, compute, and application services can be accessed. This layer offers interoperable protocol APIs to connect multiple cloud infrastructures standards specified [52].

(c) Big data platform as a service: Core layer offers several platform services to work with storage/data, and computing services based on SLAs and QoS. This layer consists of middleware management tools such as schedulers, data management tools such as NoSQL tools, and data-intensive programming models for data processing. This layer would mainly focus on development of tools and software development kits (SDKs) that are essential for the design of analytics.

(d) Big data analytics: Big data analytics offered as services, where users could quickly work on analytics without investing on infrastructure and pay only for the resources consumed. This layer organizes the repository of software appliances and quickly deploys on the infrastructure and delivers the end results to the users; the pricing would be computed based on the usage, QoS provided, and so on.

4.4. Layered components

The layered architecture, sub layers, and the components in each layer are shown in Figure 10, and Table II describes the layered architecture and their corresponding mapping of the reference architecture.

(a) Infrastructure layer

This layer provides services for effective management and delivery of the computing elements, storage, data, and networking infrastructure. This layer is further classified into two sub layers, resource and interface layers. Resource layer facilitates compute, storage, and data services either on physical or virtual environments. Physical environment is similar to data centers without virtualization enforced and is similar to cluster setup in the local network. In the case of virtual environments, it could be a private/public/hybrid cloud provider who offers services based on the consumption. The functioning of resource layers over physical and virtual environments is similar; however, virtual environments offer high utilization of the resources; on-demand resource provisioning and highly scalable, however, endure performance degradations because of enforced virtualization technologies. In the succeeding texts, services offered by resource and interface layers are described in brief.

(i) Resource layer: Resource layer handles both physical and cloud resources as discussed in the succeeding texts.

(a) Physical resources: non-virtualized compute and storage resources delivered via local data centers or in-house available. The resources may be accessed via standard protocol and networking interfaces.

(b) Virtualized/cloud resources: The resources are delivered by several cloud providers like compute, storage, and application clouds. Compute clouds offer several scalable machine instances on demand; storage/data clouds offer either storage repositories or data online, and sometimes both. Software services are similar to applications offered as services over the cloud. The cloud infrastructure may be private, public, or both. However, the access mechanisms and security implementations will differ depending on the types of clouds that were chosen for the setup. In the succeeding texts, we illustrate the functions of compute, storage/data cloud, and software services.

   (i) Compute cloud: large pool of compute machine instances to serve the demands.
   Compute machines could be created at runtime, and the data needed for analytics

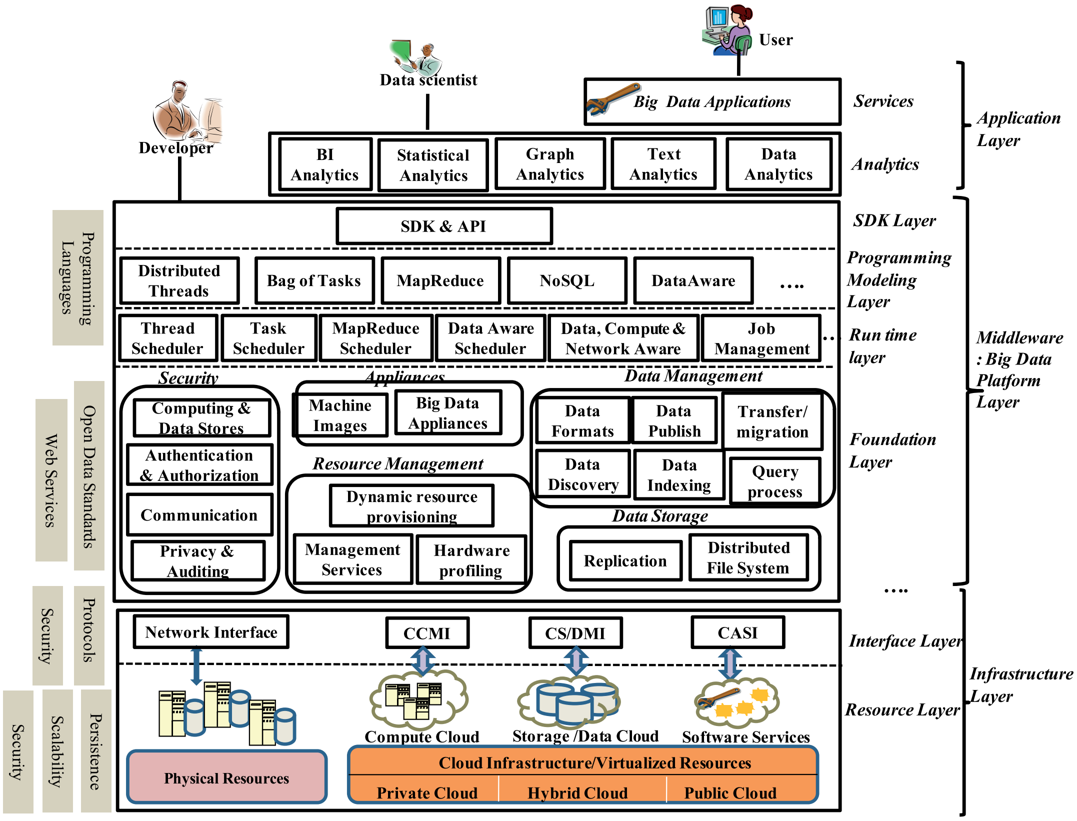

Figure 10. Big data cloud-layered components. BI, business intelligence; CCMI, cloud compute management interface; CS/DMI, cloud storage/data management interface; CASI, cloud application services interfaces; SDK, software development kit.

Table II. Layers mapped to reference architecture

Serial nos.	Layer name	Sub layers	Reference layer of architecture
1	Infrastructure	Resource and interface layers	Cloud infrastructure and big data fabric
2	Big data platform	Foundation, runtime, programming modeling layer, and SDK	Big data platform as a service
3	Applications	Analytics and big data services	Big data analytics software as a service

SDK, software development kit.

purpose may be made available dynamically.

(ii) Storage/data cloud: Storage clouds offer a pool of the storage space where in files required for analytics could be placed. However, data cloud offers the storage space along with the data necessary for compute. Such data or storage could be offered as block storage or object storage.

(iii) Service cloud: pre built analytic tools, provisioned on demand for quickly accessing the underlying data and computing resources.

(ii) Interface layer

Interface layer facilitates open-standards protocols based on web and interoperable services. The major challenges include interoperability between heterogeneous hardware and storage infrastructure, and migration/access across various cloud providers. Interface layer offer standard interfaces [52] to access compute resources, storage resources, and application services. This layer could be classified into four components based on the services rendered to the foundation

DOI: 10.1002/spe

layer, such as networking interface protocols, cloud compute management interface (CCMI), cloud storage/data management interface (CS/DMI), and cloud application services interfaces (CASI). The detailed description for each of the components is given in the succeeding texts.

- Network interface: This interface allows several physical devices access through standard networking interfaces and protocols. This includes accessing the compute instances via terminal services or web consoles. The storage devices can be mounted to the local compute machine instances or access via separate networks such as network file system protocols.
- CCMI: interoperable functional interfaces for on-demand creation and management of virtual machines of several public cloud providers.
- CS/DMI: a functional interface that applications will use to create, retrieve, update, and delete data elements from the cloud. As part of this interface, the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage container and the data that are placed in them. In addition, metadata can be set on container and their contained data elements through this interface. This interface is also used by administrative and management applications to manage container, accounts, security access, and monitoring/billing information, even for storage that is accessible by other protocols. The capabilities of the underlying storage and data services are exposed so that clients can understand the offering. Various CS/DMI interfaces are as follows:

    - Amazon S3: Amazon S3 stands for simple storage service, stores the data objects within the buckets, and composed of a file and optionally any metadata that describes that file. To store an object in Amazon S3, the file can be uploaded to the bucket, and permissions can be set on the object as well as on the metadata. Buckets are the containers, and there can be more than one bucket.
    - Open-stack swift: Object-based data storage system exposes the storage via Representational state transfer (REST) API and stores a large amount of unstructured data at low cost.

    - CASI: a set of web services that exposes the published applications through standard web protocols. This also involves application virtualization methodologies to serve only the needed applications as services from the cloud providers.

(b) Big data platform layer

This is a middleware layer that is further categorized into four sub layers based on the functionality; they are foundation layer, runtime layer, programming modeling layer, and SDK layer. The foundation layer offers mechanisms for resource management, data storage, data management, security, and virtual appliance. The runtime layer addresses several scheduling mechanisms and job management mechanisms. The programming modeling layer employs several programming standards; the SDK layer offers APIs for programming in several languages. The detailed description of the layers is given in the succeeding texts.

- Foundation layer: This is the core part of the middleware layer, which interfaces with the resource layer. This layer is mainly classified into components such as resource management, data management, appliances, data storage, and security.

- Resource management: Resource management consists of the following components.
    - Management services: the services for managing the underlying physical resources. These can be middleware services to track of the available resources. The management services include applications to monitor the resource utilizations for data and compute such as compute resources availability, storage availability, and so on.
    - Hardware profiling: the information services to retrieve the information regarding the available resources such as random access memory, network bandwidth, compute load, and so on.
    - Dynamic resource provisioning: facilitates the resources at runtime from the virtualized resources.

DOI: 10.1002/spe

- Data management: This mainly deals with data formats, discovery, and publishing mechanisms.
  - Data formats: Data formats service provides to store the data in various types of forms that include structured, unstructured, and semi-structured. Search mechanisms services offer various query mechanisms to search for the data of interest; sharing allows various access privileges.
  - Data transfer/migration: The mechanisms either pull or push the data for processing, automatic syncing of the files to the big data systems. It also contains tools that are necessary for migration of the existing structured/unstructured data to cloud big data workloads.
  - Data discovery mechanisms: several mechanisms to find the location of the data. This can be performed with the query mechanisms or searching the metadata contents. Dedicated discovery mechanisms for specific communities need to be evolved. Technologies for data discovery might include visualization, structural query mechanisms, semantic query, and so on. Data publication mechanisms: several mechanisms of domain-specific data publication and retrieval mechanisms.
  - Data indexing: Indexing mechanisms are needed to speed up the process of accessing the data. Several data indexing mechanisms need to be explored for data redundancy and replication.
  - Query process: several query processing methods for quickly analyzing the large-scale distributed data of both structure and unstructured.

- Appliances: Self-configuration and appliances eliminate the time-consuming efforts of choosing and configuring hardware, determining the proper software components, and integrating and tuning the overall configuration.
  - Machine image: the repository of machine instances for creating the systems on demand, virtual machine manager for automated management of the systems, and machine image instance, which are pre built machine templates.
  - Big data appliances: the repository and management of big data appliances that are specific for the domain-specific analytics development.
- Data storage
  - Replication procedures: several replication procedures for duplicating the data onto multiple storage repositories for data redundancy, high availability, and high-performance data transfers. Instead, repositories confined to a single location, the data would get replicated to multiple geographical locations. Some of the issues addressed replication [18] techniques specific to big data are as follows: (i) providing extremely rapid access to data from multiple sources, even in a mixed workload, (ii) reducing the drag on multi-way joins for complex queries, (iii) accelerating reporting for faster analysis, review, and decision-making; and (iv) backup and disaster recovery techniques. For example, Tervela [53] accelerates big data replication by efficiently duplicating to multiple sites with ease through one of the two methods such as changed data capture or parallel replication.
  - Distributed file system: file system that stores the data onto multiple distributed storage repositories. The file system maintains the indexing of the data and offers various logical views of the entire data available in the system.

- Big data security: privacy preserving, auditing, and role-based access mechanisms for providing security to the data for both data at rest and while in transit.

- Runtime layer: This layer is concerned about workload handling with the support of several scheduling mechanisms. Examples include thread, task, MapReduce, data and network-aware scheduling, and batch job management based on the type of computation needed. In the succeeding texts, we briefly discuss the functions and characteristics of several types of schedulers.

  - Thread scheduler: Thread scheduling exploits the available cores/processors effectively by utilizing either local system or remote system resources. Local threads execution could use shared memory; however, for remote execution, objects migration would take place. Thread scheduling is applicable for problems that are recursive, multiple data streams but applied on a single instruction. Thread scheduler determines the best resources for running the several spawn-independent threads on available resources/cores. Thread scheduling addresses high performance computing problems.

DOI: 10.1002/spe

- Task scheduler: distributed processing of tasks on several computing nodes. Task scheduling solves the high throughput problems by determining the best available resources for execution.
- Data-aware scheduler: jobs execution, knowing the best available storage locations for execution or transfer the data to compute nodes from the best available store repositories. The process could depend on computing the best replicated site that minimizes the compute time. This may apply data parallelism to pull/push the data to the compute nodes. MapReduce is an example of data-aware scheduler.
- MapReduce scheduler: type of data-aware scheduling that maps the compute process to the data nodes. After the completion of the process, the results are consolidated onto a single node for a final result.
- Data compute and bandwidth-aware scheduler: considering compute, network, and data to solve the data-intensive scheduling. The techniques applied for this type of scheduling are as follows:

i. Parallel data extractor: Parallel data extractor is the high performance data transfer module. It enables extraction of transfer of data from storage clouds to the compute node. This module pulls the data from the storage repositories by establishing multiple parallel lines between storage clouds to compute resources. Parallel data extractor identifies possible data storage resources and identifies the amount of data to be pulled from each of the storage repositories.
ii. Scheduler: the scheduler that effectively maps a set of jobs to the computing nodes, the scheduling would depend on heuristic approaches. Big data schedulers could pick up the best computing nodes or may quickly clone the virtual machines and perform the computation by applying effective data-aware scheduling techniques.
iii. Job management: management tools for monitoring the job executions.

- Programming modeling layer: several programming models to solve the big data problems. This may include coarse/fain grain programming models for thread, tasks, and data intensive. It also addresses data handling and query programming models for NoSQL databases.
- SDK layer: the programming APIs to solve big data problems. This could be Java, C, C++, and C#-based APIs.
- Application layer: Application layer provides various statistical, deterministic, probabilistic, machine-learning techniques for developing the domain-specific analytics tools. This layer offers SDKs, APIs, and tools for the analytics development and is also responsible for several management interfaces development for monitoring the big data environments. The several analytics include statistical models, graph analytics, business analytics, text analytics, and data analytics.

(c) Users: the several stakeholders of the systems like (i) developers: big data general purpose application designer; (ii) data scientist: data analysts who design the analytics applications. This could be business analytics, scientific explorations, and so on; and (iii) end users: analytics users of the system.

5. GAP ANALYSIS AND FUTURE DIRECTIONS

Big data research area is broadly classified into four major segments denoted as 4Ds, that is, depository, devise, domain, and determine, as shown in Figure 11. Depository deals about the storage technologies, devise is about working on new platforms and programming models, domain is latest trends platforms and tools specific to the various engineering domains, and finally determine is about working on the analytics for mining and information extraction. In this section, we discuss the sub elements in each segment described in the succeeding texts.

5.1. Depository

This segment deals with the long-term persistent storage and retrieval of both structured and unstructured data of geographical-dispersed locations. The several research areas include migrating from the traditional storage like storage area networks, and network-attached storage to the container-based object storage systems. This would eliminate hierarchical file structure handling

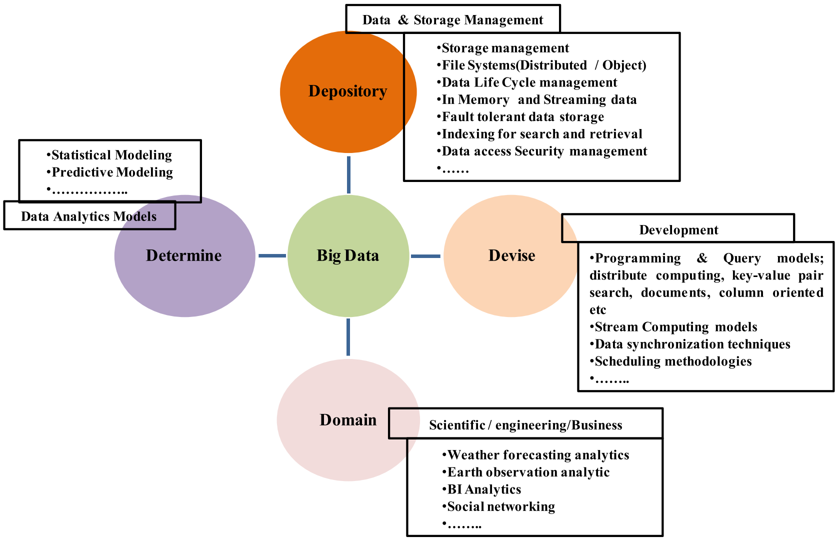
Data & Storage Management
Depository
•Storage management
•File Systems(Distributed / Object)
•Data Life Cycle management
•In Memory and Streaming data
•Fault tolerant data storage
•Indexing for search and retrieval
•Data access Security management
•……
•Statistical Modeling
•Predictive Modeling
•……………..
Data Analytics Models
Determine
Big Data
Devise
Development
•Programming & Query models; distribute computing, key-value pair search, documents, column oriented etc
•Stream Computing models
•Data synchronization techniques
•Scheduling methodologies
•……..
Domain
Scientific / engineering/Business
•Weather forecasting analytics
•Earth observation analytic
•BI Analytics
•Social networking
•……..

Figure 11. Big data research segments.

(like directories and subdirectories), eliminate the need of web servers, and load balancers as accessed via hypertext transfer protocol-based REST services, facilitate location transparency with unique object identification, high available, and fault-tolerant storages dispersed across distributed locations described in Table III.

5.2. *Devise – big data platform services*

This segment focused on the design of platforms and programming models in distributed computing, in-memory computing, stream computing, and query languages for assorted data such as key-value data stores, column-oriented data, document databases, content synchronization techniques, scheduling methodologies, and so on. Recently, several big data computing platforms are emerged such as Hadoop, Spark, Amazon elastic MapReduce, Dryad, HDInsight, Aneka, and MapReduce [47, 54] have become a *de facto* standard of big data applications over a large-scale cluster of computing nodes. However, MapReduce has limitations both from theoretical perspective [55, 56] and empirically by exploring classes of algorithms that cannot be efficiently implemented [57–60]; several limitations of the MapReduce model over Hadoop file system are described in the succeeding texts (Table IV).

5.3. *Domain – scientific, engineering, and business*

Big data analytics extract information from large data for decision-making. Examples include earth observation systems, disaster management study, weather forecasting, simulations, engineering design problems, and business intelligence applications. It is also necessary to evolve several complex applications for monitoring historical data apart from the operational data. The researchers should focus on the following activities to address the domains of data science (Table V).

5.4. *Decision – mining and determining*

Big data processing is driven by statistical and analytical models to derive information for decision-making. Big data is not just about the data, but the ability to solve business and scientific exploration problems, providing new business opportunities and thoughts. Data analytic plays a key role in information mining and derives a value out of the data. At present, several analytic systems are evolving, but the majority systems are based on the frameworks that are general purpose tools. Hence, it is essential to address several frameworks in the areas of predictive analysis, behavior

Table III. Depository: gap analysis and future directions

Key element	Gap analysis and future directions
1. Storage and network	• The unified storage systems with the combination of three layers traditional storage architectures like file system, volume manager, and data protection with wide range of industry standard protocols, including network file system, server message block, hypertext transfer protocol, file transfer protocol, REST-based object access, and so on, converged I/O protocols like InfiniBand need to be investigated for large volume I/O operations and analytics. • Write once and read many object storage technologies for both long-term preservation and mining. The indexing techniques need to be explored either attached with the file object or as separate meta file using NoSQL data structures. • New WAN-based protocols need to be investigated as the traditional WAN-based transport methods cannot move terabytes of data. These transport methods use effective bandwidth and achieve transfer speeds. • Storage protocols need to be explored for on-disk data encryption, privacy preserving, and query on encrypted data mechanisms with reasonable good speeds.
2. File system	• Conventional file systems have constraints in name space handling as they are assigned by the operating system. This would be difficult for the searching process, while the data are mostly unstructured. Hence, new storage technologies, such as object storage services, would allow the files to assign user-defined metadata separating it from the operating systems. • Information-defined data storage complements the existing distributed file system to derive the value out of the data by its content and meaning but not just with names. • File system migration tools of the traditional file systems to object-based storage systems to be evolved. • High-performance parallel data transfer with data distribution, replication, and redundant mechanisms. This could be achieved by replicating the big data file objects on multiple distributed storage systems and creating the clustered indexes. • Data migration tools need to be addressed for moving the on-premise big data files to the cloud-based big data systems, which could be relational data or unstructured like documents, texts, videos, audios, and so on.
3. Data life cycle management	• Data life cycle addresses the data management issues at several phases of data creation, usage, sharing, storing, and eventually archiving or disposing automatically based on policies defined within the management. Big data life cycle management systems need to be developed incorporating several user-defined policies for data organization and minimize the storage resource costs. For example, the policy could be data aging and addresses issues related to the data obsolete, something like, deleting the age-old objects. Another policy could be the version management, holding only the recent versions of each object in a bucket with a versioning enabled. • This N-tier storage architectures could organize the data that are mostly used in lower tiers (tier1) such as Flash/solid-state drive and migrate the data moving down to other tiers such as flat disks and capacity disks to tier N such as cloud storage for backup/archival.
4. In-memory computing systems	In-memory computing systems with object storages file systems for effective computing methods and long-term preserving need to be evolved. It also enables querying the data based on the metadata from the cloud storage pools and performs the object-based data storage, query, and object-based data analysis.
5. HA and fault-tolerant data storage	HA systems offer storage providing multiple internal components and multiple access points to storage resources. In other words, the system has a second critical component or path to data available in case something fails. This availability or single point of failure does not eliminate downtime. Instead, it minimizes it by restoring services behind the scenes, in most cases before the user notices failures.

(*Continues*)

DOI: 10.1002/spe

Table III. (Continued)

Key element	Gap analysis and future directions
6. Indexing for search and retrieval	Indexing multidimensional data and enabling object-based retrieval mechanisms instead of set-based needs to be developed for efficient query processing.
7. Data access and security	Data access and security mechanism in big data clouds need to be developed, which set policies enabling which users get access to which original data, with protection of sensitive data that maintains usable, realistic values for accurate analytics and modeling on data.

HA, high availability; I/O, input/output; REST, representational state transfer.

Table IV. Device: gap analysis and future directions

Key element	Gap analysis and future directions
1. Programming models	• MapReduce programming addressing iterative and non iterative models, with storage coupled with computing nodes and as separate services need to be investigated. Current MapReduce models are iterative in nature [61]; hence, the problems like page ranking iterative graph algorithms and gradient descent cannot be addressed. Also, the current models of MapReduce to be modified or extended to address the several problems in engineering and scientific domains like data product generation [62–64] and digital elevation model [65, 66], as the processing demands large data, hence several data aggregation and processing scheduling mechanisms to be evolved. • Debugging tools and profilers for MapReduce programming model required to be investigated for MapReduce programming. Currently, there are batch-based without user interaction. • Domain-specific languages need to evolve such data intensive, high performance, Internet of things programming, and so on to solve specific problems in several fields of final services, business sectors, scientific explorations, sensors networks, and so on.
2. Unstructured data processing	• Document, text, and graph-based data processing mechanisms with better indexing mechanisms to be explored. This could use key value pair mechanisms with schema less data bases and MapReduce functions for effectively retrieving the data by processing on the cluster of machines. • Unstructured database systems to be explored to bride gap between traditional databases and key value pair databases. These data base systems should work on object notations and perform query on multiple nodes to improve the performance.
3. Scheduling methodologies	• Evolutions of new programming models for compute-intensive big data problem: Programming models with the combination of thread, task, and MapReduce need to be devised. The current MapReduce programming model will transfer the compute to the data node. Here, compute is considered to be a small activity, as compared with data. This model will not be applicable, while compute is as large as data. Hence, new programming models are to be exploited. • QoS-based resource management scheduling methods need to be developed that would work on parameters like time, budget, accuracy, and so on. • HPC programming models such as high speed in memory computing and stream computing need to be worked for HPC big data clouds for scientific applications to address data science problems with accuracy in real time.
4. Workspace management	• Collaborative framework for analytics development to be developed. These frameworks organize the source code, data, and so on in sharing mode and allow the analysts to design and develop the applications in both on-premise and cloud-based platforms.

HPC, high-performance computing.

analysis, business intelligence, and so on. In big data mining, several open-source initiatives tools are becoming popular, as mentioned in the succeeding texts. A few research challenges are described in the succeeding texts.

Table V. Domain: gap analysis and future directions

Key element	Gap analysis and future directions
1. Data management architectures	• Data management architectures in several domains of geo spatial [67, 68], health care [69], social networking, and web log mining [70, 71] are still to be explored. The data gathered and processed in these fields are unstructured and domain-specific; hence, indexing architectures, metadata management schemes, query, and processing tools specific to the domains are still to be investigated.
2. Data visualization models	• Visualization tools have to be investigated for presenting the large-scale data and the analyzed/processed results over dashboards, reports, and charts [72, 13].
3. Domain-specific analytic models	• Domain-specific analytics tools that would pick up the appropriate NoSQL databases necessary for the analytics need to be explored. • New domain models to be investigated, for migration of the existing in-house domain-specific analytics to clouds. These models address the data management issues, extract, transformation, and load tools for the in-house data with effective indexing, processing, and tools for analysis. • Analytical models to work on the interested data regions and assigning the score based on the ranking. This could enable the analytics to pick up the most relevant data for analysis increasing the system throughput.

- Evolve new architecture for analytics to deal with both historical and real-time data at the same time. This could be achieved by organizing the unstructured data as N-tier system with effective indexing and performing the distributed queries and data-intensive programming techniques to analyze the historical data and compare with the present data to derive the intrinsic information.
- Statistical significance tools need to be developed to determine maximum likelihood (e.g., *p*-value [72]) for the evidence based on the probabilities and statistics, rather on randomness of data distribution.
- Distributed parallel data mining algorithms and frameworks for unstructured large volume of data need to be investigated, to analyze the data quickly and provide the results summary.
- Time-evolving data mining techniques need to be investigated for the evolving data sets such as words, graph analytics for social networking, behavior analytics, predictive analytics, earth observation geo intelligence solutions, weather forecasting, and so on.
- New techniques need to be evolved that could quickly identify the portion of the data that need to be mined rather as a whole to quickly deliver the analysis results.
- Big data computing in clouds and cloud-based analytics services need to be developed for domain-specific applications meeting QoS, SLAs, and budget, deadline constraints.
- Distributed real-time, predictive, and prescriptive analytics tools need to be evolved that could provide the interactivity to the running jobs, apply statistical tools to determine the information, and offer the results in real time [13].

6. SUMMARY AND CONCLUSIONS

Big data computing is an emerging platform for data analytics to address large-scale multidimensional data for knowledge discovery and decision-making. In this paper, we have studied, characterized, and categorized several aspects of big data computing systems. Big data technology is evolving and changing the present traditional data bases with effective data organization, large computing, and data workloads processing with new innovative analytics tools bundled with statistical and machine-learning techniques. With the maturity of cloud computing technologies, big data technologies are accelerating in several areas of business, science, and engineering to solve data-intensive problems. We have enumerated several case studies of big data technologies in the areas of healthcare studies, business intelligence, social networking, and scientific explorations. Further, we focus on illustrating how big data databases differ from traditional data base and discuss BASE properties supported by them.

To understand big data paradigm, we presented taxonomy of big data computing along with discussion on characteristics, technologies, tools, security mechanisms, data organization, scheduling approaches, and so on along with relevant paradigms and technologies. Later, we presented under pinning technologies for the evolution of big data and discussed how cloud computing technologies would be utilized for infrastructure services delivery for the analytics development. Later, we discussed an emerging big data computing platforms over clouds, big data clouds, an integrated technology from big data and cloud computing, and delivering big data computing as a service over large-scale clouds. The paper also discussed types of big data clouds and illustrated big data access networks, an emerging data platform services for big data analytics.

Further on, we presented layered architecture components under each of the layers followed by technologies to be addressed under each of the layers. We then compare some of the existing systems in each of the areas and categorize them based on the tools and services rendered to the users. In doing so, we have gained an insight into the architecture, strategies, and practices that are currently followed in big data computing. Also, through our characterization and detailed study, we are able to discover some of the short comings and identify gaps in the current architectures and systems. These represent some of the directions that could be followed in the future. Thus, this paper lays down a comprehensive classification framework that not only serves as a tool to understand this emerging area but also presents a reference to which future efforts can be mapped.

To conclude, big data technologies are being adopted widely for information exploitation with the help of new analytics tools and large-scale computing infrastructure to process huge variety of multidimensional data in several areas ranging from business intelligence to scientific explorations. However, more research needs to be undertaken, in several areas like data organization, decision-making, domain-specific tools, and platform tools to create next generation big data infrastructure for enabling users to extract maximum utility out of the large volumes of available information and data.

ACKNOWLEDGEMENTS

We thank Rodrigo Calheiros, Nikolay Grozev, Amir Vahid, and Harshit Gupta for their comments and suggestions on improving the quality of the paper. This work is partially supported by the Future Fellowship project of the Australian Research Council and Melbourne-Chindia Cloud Computing (MC^3) Research Network.

REFERENCES

1. Dean J, Ghemawat S. MapReduce: simplified data processing on large cluster. *Communications of the ACM* 2008; **51**(1): 107–113.
2. Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S. The data grid: towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications* 2000; **23**(3): 187–200.
3. Janaki A, KubachT, Loffer M, Schmid U. Data driven management: bringing more science into management, White Paper, McKinsey Technology Initiative Perspective, 2008. http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation [last accessed 30 November 2014].
4. CRA (Computing Research Association), Challenges and opportunities with big data, http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf [last accessed 12 November 2014].
5. Advancing discovery in science and engineering, the role of basic computing research, http://www.cra.org/ccc/files/docs/Natl_Priorities/web_data_spring.pdf [last accessed 10 August 2014].
6. IDC, The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the Far East, www.emc.com/leadership/digital-universe/index.htm [last accessed 20 November 2014].
7. Apache Hadoop, http://hadoop.apache.org [last accessed 15 October 2014].
8. Chen C.L.P, Zhang C Y. Data-intensive applications, challenges, techniques and technologies: a survey on big data, *Inform.Sci*, DOI: 10.1016/j.ins.2014.01.015.
9. Chen M, Mao S, Liu Y. Big data survey. *Mobile Networks and Applications* 2014; **19**(2): 171–209.
10. Chen J, Chen Y, Du X, Li C, Lu J, Zhao S, Zhou X. Big data challenge: a data management perspective. *Frontiers of Computer Science* 2013; **7**(2): 157–164.
11. Wu X, Zhu X, Wu GQ, Ding W. Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering* 2014; **26**(1): 97–107.

DOI: 10.1002/spe

12. Kaisler S, Armour F, Espinosa J.A and Money W. Big data: issues and challenges moving forward, in: Proceedings of the 46th IEEE Annual Hawaii international Conference on System Sciences (HICC 2013), Grand Wailea, Maui, Hawaii, January 2013, pp. 995–1004.
13. Assuncao MD, Calheiros RN, Bianchi S, Netto M, Buyya R. Big data computing and clouds: trends and future directions. *Journal of Parallel and Distributed Computing (JPDC)* 2015; **79**(5): 3–15.
14. Survey of big data architectures and framework from the industry, NIST big data public working group, 2014. http://jtc1bigdatasg.nist.gov/_workshop2/07_NBD-PWD_Big_Data_Architectures_Survey.pdf, 2014 [last accessed 30 April 2014].
15. Mayer VV, Cukier K. *Big Data: A Revolution That Will Transform How We Live, Work and Think*. John Murray Press: UK, 2013.
16. Ginsberg J. Detecting influenza epidemics using search engine query data. *Nature* 2009; **457**: 1012–1014.
17. Buyya R, Shin Yeo C, Venugopal S, Brobergand J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 2009; **25**(6): 599–616.
18. Broberg J, Venugopal S, Buyya R. Market oriented grids and utility computing: the state-of-the art and future directions. *Journal of Grid Computing* 2008; **6**(3): 255–276.
19. Inmon WH. *Building the Data Warehouse* 4th edition. Wiley publishing Inc.: Indianapolis, 2005.
20. Eric A. Brewer, Towards robust distributed systems, keynote speech in 19th ACM Symposium on Principles of Distributed Computing (PODC 2000), Portland, Oregon, July 2000.
21. Gray J. The transaction concept: virtues and limitations, in: Proceedings of the 7th International Conference on Very Large Databases (VLDB' 81) 7 (1981) 144–154.
22. Pritchett D. BASE: an ACID alternative. *Queue Object Relational Mapping* 2008; **6**(3): 48–55.
23. Harder T, Reuter A. Principles of transaction-oriented database recovery. *ACM Computing Surveys* 1983; **15**(4): 287–317.
24. Cooper M, Mell P. Tackling big data, NIST information technology laboratory computer security division, http://csrc.nist.gov/groups/SMA/forum/documents/june2012presentations/fcsm_june2012_cooper_mell.pdf [last accessed 15 September 2014].
25. A very short history of data science, http://whatsthebigdata.com/2012/04/26/a-very-short-history-of-data-science [last accessed 15 September 2014].
26. In-memory analytics, leveraging emerging technologies for business intelligence, Gartner Report, 2009.
27. Apache Spark, https://spark.incubator.apache.org [last accessed 03 April 2014].
28. Apache storm, http://storm.incubator.apache.org [last accessed 03 April 2014].
29. S4: distributed stream computing platform, http://incubator.apache.org/s4 [last accessed 20 November 2014].
30. Google big query, https://cloud.google.com/bigquery-tour [last accessed 15 January 2015].
31. Chang F, Dean J, Ghemawat S, Heish W. C., Wallach D. A, Burrows M, Chandra T, Fikes A, Gruber R. E. Big table: a distributed storage system for structured data, in: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2006), Seattle, WA, Nov 2006.
32. Decandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W. Dynamo: Amazon's highly available key-value store, in: Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP 2007), Stevenson, Washington, USA, Oct 2007.
33. Amazon elastic MapReduce, developer guide, 2015, http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-dg.pdf [last accessed 1 November 2014].
34. Chauhan A, Fontama V, Hart M, Hyong W, Woody B. *Introducing Microsoft Azure HDInsight, Technical Overview*. Microsoft press: One Microsoft Way, Redmond, Washington, 2014.
35. Rack space, www.rackspace.com [last accessed 22 August 2014].
36. Horton Hadoop, http://hortonworks.com [last accessed 22 August 2014].
37. Cloudera Hadoop, http://www.cloudera.com [last accessed 03 September 2014].
38. Oracle Berkeley DB, Oracle data sheet, http://www.oracle.com/technetwork/products/berkeleydb/berkeley-db-datasheet-132390.pdf [last accessed 03 September 2014].
39. Buyya R, Vecchiola C, Selvi T. *Mastering in Cloud Computing – Foundations and Applications Programming*. Morgan Kaufman: USA, 2013.
40. Apache couch DB, a database for the web, www.couchdb.apache.org [last accessed 10 September 2014].
41. MongoDB operations best practices, http://info.10gen.com/rs/10gen/images/10gen-MongoDB_Operations_Best_Practices.pdf.
42. Apache HBase, http://hbase.apache.org [last accessed 20 December 2014].
43. InfiniteGraph: the distributed graph database, a performance and distributed performance benchmark of InfiniteGraph and a leading open source graph database using synthetic data, infinite graph, white paper from objectivity, http://www.objectivity.com/wp-content/uploads/Objectivity_WP_IG_Distr_Benchmark.pdf, 2012 [last accessed 20 December 2014].
44. Neo4j graph database, http://www.neo4j.org [last accessed 20 December 2014].
45. Aggarwal CC, Zhai C. *Probabilistic Models for Text Mining: In Mining Text Data*. Kluwer Academic Publishers: Netherlands, 2012: 257–294.
46. Nyce C. Predictive analytics white paper, American Institute for CPCU/Insurance Institute of America, http://www.theinstitutes.org/doc/predictivemodelingwhitepaper.pdf, 2007 [last accessed 30 January 2015].
47. Apache MapReduce, http://hadoop.apache.org/docs/stable/mapred_tutorial.html [last accessed 20 February 2015].
48. Apache Mahout, scalable machine learning library, http://mahout.apache.org [last accessed 20 February 2015].
49. Capacity scheduler, http://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.pdf [last accessed 20 February 2015].
50. Fair scheduler, http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.pdf [last accessed 20 February 2015].

DOI: 10.1002/spe

51. Google-gdata, .NET library for the Google data API, http://code.google.com/p/google-gdata [last accessed 20 February 2015].
52. Cloud infrastructure management interface (CIMI) model and RESTful HTTP based protocol – an interface for managing cloud infrastructure, http://dmtf.sites/default/files/standards/documents/DSP0263_1.0.1.pdf [last accessed 13 February 2015].
53. Big data replication, www.tervela.com/big-data-replication [last accessed 20 March 2015].
54. Condie T, Conway N, Alvaro P, Hellerstein J, Elmeleegy K, Sears R. MapReduce online, in: Proceedings of the 7th USENIX on Networked systems design and implementation (NSDI 2010), San Jose, California, April 2010.
55. Karlof H, Suri S, Vassilvitskii S. A model of computation for MapReduce, in: Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), Austin, Texas, January 2010.
56. Afrati F, Sarma A, Salihoglu S, Ullman J. Vision paper: towards an understanding of the limits of Map-Reduce computation, arxiv.org/abs/1204.1754 [last accessed 09 December 2014].
57. Ekanayake J, Li H, Zhang B, Gunarathne T, Bae S. H, Qiu J, Fox G. Twister: a runtime for iterative MapReduce, in: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 2010), Chicago, Illinois, June 2010.
58. Bhatotia P, Wieder A, Akkus I, Rodrigues R, Acar U. Large-scale incremental data processing with change propagation, in: Proceedings of the 3rd USENIX Workshop on Hot topics in Cloud computing (HotCloud 2011), Portland, June 2011.
59. Bu Y, Howe B, Balazinska M, Ernst M. HaLoop: efficient iterative data processing on large cluster, in: Proceedings of the 36th International Conference on Very Large Databases (VLDB 2010), Singapore, September 2010.
60. Zhang Y, Gao Q, Gao L, Wang C. PrIter: a distributed framework for prioritized iterative computations, in: Proceedings of the 2nd ACM Symposium on Cloud Computing (SoCC 2011), Cascais, Portugal, October 2011.
61. Lin J. MapReduce is good enough? If all you have is a hammer, throw away everything that's not a nail!, http://arxiv.org/pdf/1209.2191v1.pdf [last accessed 02 March 2015].
62. Raghavendra K, Chaudhri A, Kumar K.P, Varadan G. High performance private cloud for satellite data processing – engineering in cloud, in: Proceedings of the 1st International Conference on Advances in Cloud Computing (ACC 2012), Bangalore, India, July 2012.
63. Raghavendra K, Akilan A, Ravi K, Kumar K.P, Varadan G. Satellite data product generation on Aneka .NET cloud, research product demonstration, Presented at the 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010), Melbourne, Australia, May 2010.
64. Radhadevi PV, Solanki SS. In-flight geometric calibration of different cameras of IRS-P6 using a physical sensor model. *The Photogrammetric Record* 2008; **23**(121): 69–89.
65. Deepak P, Kumar K. P, Varadan G. A service oriented utility grid for data parallel remote sensing applications, in: Proceedings of 2009 High Performance Computing & Simulation (HPCS 2009) in conjunction with International Wireless Communications and Mobile Computing Conference (IWCMC 2009), Leipzig, Germany, June 2009, pp. 131–137.
66. Deepak P, Kumar K. P, Varadan G. Service oriented utility grid for 3-dimensional topographic visualization from satellite images, in: Proceedings of the 4th International Conference on eScience, Indianapolis, Indiana, USA, Dec 2008, pp. 47–54.
67. Lee JG, Kang M. Geospatial big data: challenges and opportunities. *Big Data Research* 2015; **2**(2): 74–81.
68. Jardak C, Mahonen P, Riihiajariv J. Spatial big data and wireless networks: experiences, applications, and research challenges. *IEEE Network* 2014; **28**(4): 26–31.
69. How big data impacts health care, Harvard business review, https://hbr.org/resources/pdfs/comm/sap/18826_HBR_SAP_Healthcare_Aug_2014.pdf, 2014. [last accessed 03 March 2015].
70. Catanese S. A, Meo P. D, Ferrara E, Flumara G, Provetti A. Crawling Facebook for social network analysis purposes, in: Proceedings of 1st International Conference on Web Intelligence, Mining and Semantics (WIMS'11), Sogndal, Norway, May 2011.
71. Berger P, Hennig P, Klingbeil T, Kohnen M, Pade S, Meinel C. Mining the boundaries of social networks: crawling Facebook and Twitter for blog intelligence, in: Proceedings of 12th annual conference on Information and Knowledge Engineering (IKE' 13), Las Vegas, Nevada, USA, July 2013.
72. John A. R.A. Fisher and the making of maximum likelihood 1912–1922. *Statistical Science* 1997; **12**(3): 162–176.

DOI: 10.1002/spe