Information Sciences 275 (2014) 314–347

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Data-intensive applications, challenges, techniques and technologies: A survey on Big Data

C.L. Philip Chen *, Chun-Yang Zhang

Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China

ARTICLE INFO

Article history:
Received 28 March 2013
Received in revised form 3 January 2014
Accepted 10 January 2014
Available online 21 January 2014

Keywords:
Big Data
Data-intensive computing
e-Science
Parallel and distributed computing
Cloud computing

ABSTRACT

It is already true that Big Data has drawn huge attention from researchers in information sciences, policy and decision makers in governments and enterprises. As the speed of information growth exceeds Moore's Law at the beginning of this new century, excessive data is making great troubles to human beings. However, there are so much potential and highly useful values hidden in the huge volume of data. A new scientific paradigm is born as data-intensive scientific discovery (DISD), also known as Big Data problems. A large number of fields and sectors, ranging from economic and business activities to public administration, from national security to scientific researches in many areas, involve with Big Data problems. On the one hand, Big Data is extremely valuable to produce productivity in businesses and evolutionary breakthroughs in scientific disciplines, which give us a lot of opportunities to make great progresses in many fields. There is no doubt that the future competitions in business productivity and technologies will surely converge into the Big Data explorations. On the other hand, Big Data also arises with many challenges, such as difficulties in data capture, data storage, data analysis and data visualization. This paper is aimed to demonstrate a close-up view about Big Data, including Big Data applications, Big Data opportunities and challenges, as well as the state-of-the-art techniques and technologies we currently adopt to deal with the Big Data problems. We also discuss several underlying methodologies to handle the data deluge, for example, granular computing, cloud computing, bio-inspired computing, and quantum computing.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Big Data has been one of the current and future research frontiers. In this year, Gartner listed the "Top 10 Strategic Technology Trends For 2013" [158] and "Top 10 Critical Tech Trends For The Next Five Years" [157], and Big Data is listed in the both two. It is right to say that Big Data will revolutionize many fields, including business, the scientific research, public administration, and so on. For the definition of the Big Data, there are various different explanations from 3Vs to 4Vs. Doug Laney used *volume*, *velocity* and *variety*, known as 3Vs [96], to characterize the concept of Big Data. The term volume is the size of the data set, velocity indicates the speed of data in and out, and variety describes the range of data types and sources. Sometimes, people extend another V according to their special requirements. The fourth V can be *value*, *variability*, or *virtual* [207]. More commonly, Big Data is a collection of very huge data sets with a great diversity of types so that it becomes difficult to process by using state-of-the-art data processing approaches or traditional data processing platforms. In 2012, Gartner retrieved and gave a more detailed definition as: "Big Data are high-volume, high-velocity, and/or high-variety

* Corresponding author.
E-mail addresses: Philip.Chen@ieee.org (C.L. Philip Chen), cyzhangfst@gmail.com (C.-Y. Zhang).

0020-0255/$ - see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2014.01.015

information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization". More generally, a data set can be called Big Data if it is formidable to perform capture, curation, analysis and visualization on it at the current technologies.

With diversified data provisions, such as sensor networks, telescopes, scientific experiments, and high throughput instruments, the datasets increase at exponential rate [178,110] as demonstrated in Fig. 1 (source from [67]). The off-the-shelf techniques and technologies that we ready used to store and analyse data cannot work efficiently and satisfactorily. The challenges arise from data capture and data curation to data analysis and data visualization. In many instances, science is legging behind the real world in the capabilities of discovering the valuable knowledge from massive volume of data. Based on precious knowledge, we need to develop and create new techniques and technologies to excavate Big Data and benefit our specified purposes.

Big Data has changed the way that we adopt in doing businesses, managements and researches. Data-intensive science especially in data-intensive computing is coming into the world that aims to provide the tools that we need to handle the Big Data problems. Data-intensive science [18] is emerging as the fourth scientific paradigm in terms of the previous three, namely empirical science, theoretical science and computational science. Thousand years ago, scientists describing the natural phenomenon only based on human empirical evidences, so we call the science at that time as empirical science. It is also the beginning of science and classified as the first paradigm. Then, theoretical science emerged hundreds years ago as the second paradigm, such as Newton's Motion Laws and Kepler's Laws. However, in terms of many complex phenomenon and problems, scientists have to turn to scientific simulations, since theoretical analysis is highly complicated and sometimes unavailable and infeasible. Afterwards, the third science paradigm was born as computational branch. Simulations in large of fields generate a huge volume of data from the experimental science, at the same time, more and more large data sets are generated in many pipelines. There is no doubt that the world of science has changed just because of the increasing data-intensive applications. The techniques and technologies for this kind of data-intensive science are totally distinct with the previous three. Therefore, data-intensive science is viewed as a new and fourth science paradigm for scientific discoveries [65].

In Section 2, we will discuss several transparent Big Data applications around three fields. The opportunities and challenges aroused from Big Data problems will be introduced in Section 3. Then, we give a detailed demonstration of state-of-the-art techniques and technologies to handle data-intensive applications in Section 4, where Big Data tools discussed there will give a helpful guide for expertise users. In Section 5, a number of principles for designing effective Big Data systems are listed. One of the most important parts of this paper, which provides several underlying techniques to settle Big Data problems, is ranged in Section 6. In the last section, we draw a conclusion.

2. Big Data problems

As more and more fields involve Big Data problems, ranging from global economy to society administration, and from scientific researches to national security, we have entered the era of Big Data. Recently, a report [114] from McKinsey institute gives transformative potentials of Big Data in five domains: health care of the United States, public sector administration of European Union, retail of the United States, global manufacturing and personal location data. Their research claims that

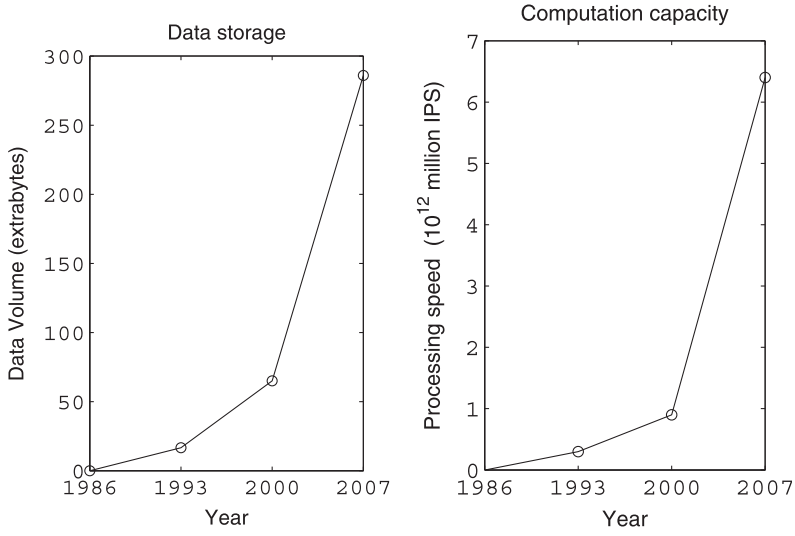

Fig. 1. Data deluge: the increase of data size has surpassed the capabilities of computation.

Big Data can make prominent growth of the world economy by enhancing the productivity and competitiveness of enterprises and also the public administrations.

Big Data has a deep relationship with *e-Science* [66], which is computationally intensive science which usually is implemented in distributed computing systems. Many issues on Big Data applications can be resolved by e-Science which require *grid computing* [80]. e-Sciences include particle physics, bio-informatics, earth sciences and social simulations. It also provides technologies that enable distributed collaboration, such as the Access Grid. Particle physics has a well-developed e-Science infrastructure in particular because of its need for adequate computing facilities for the analysis of results and storage of data originating from the European Organization for Nuclear Research (CERN) Large Hadron Collider, which started taking data in 2009. e-Science is a big concept with many sub-fields, such as *e-Social Science* which can be regarded as a higher development in e-Science. It plays a role as a part of social science to collect, process, and analyse the social and behavioral data.

Other Big Data applications lies in many scientific disciplines like astronomy, atmospheric science, medicine, genomics, biologic, biogeochemistry and other complex and interdisciplinary scientific researches. Web-based applications encounter Big Data frequently, such as recent hot spots *social computing* (including social network analysis, online communities, recommender systems, reputation systems, and prediction markets), Internet text and documents, Internet search indexing. Alternatively, There are countless sensor around us, they generate sumless sensor data that need to be utilized, for instance, intelligent transportation systems (ITS) [203] are based on the analysis of large volumes of complex sensor data. Large-scale e-commerce [183] are particularly data-intensive as it involves large number of customers and transactions. In the following subsections, we will briefly introduce several applications of the Big Data problems in commerce and business, society administration and scientific research fields.

2.1. Big Data in commerce and business

According to estimates, the volume of business data worldwide, across almost companies, doubles every 1.2 years [114]. Taking retail industry as an example, we try to give a brief demonstration for the functionalities of Big Data in commercial activities. There are around 267 million transactions per day in Wal-Mart's 6000 stores worldwide. For seeking for higher competitiveness in retail, Wal-Mart recently collaborated with Hewlett Packard to establish a data warehouse which has a capability to store 4 petabytes (see the size of data unit in Appendix A) of data, i.e., 4000 trillion bytes, tracing every purchase record from their point-of-sale terminals. Taking advantage of sophisticated machine learning techniques to exploit the knowledge hidden in this huge volume of data, they successfully improve efficiency of their pricing strategies and advertising campaigns. The management of their inventory and supply chains also significantly benefits from the large-scale warehouse.

In the era of information, almost every big company encounters Big Data problems, especially for multinational corporations. On the one hand, those companies mostly have a large number of customers around the world. On the other hand, there are very large volume and velocity of their transaction data. For instance, FICO's falcon credit card fraud detection system manages over 2.1 billion valid accounts around the world. There are above 3 billion pieces of content generated on Facebook every day. The same problem happens in every Internet companies. The list could go on and on, as we witness the future businesses battle fields focusing on Big Data.

2.2. Big Data in society administration

Public administration also involves Big Data problems [30]. On one side, the population of one country usually is very large. For another, people in each age level need different public services. For examples, kids and teenagers need more education, the elders require higher level of health care. Every person in one society generates a lot of data in each public section, so the total number of data about public administration in one nation is extremely huge. For instance, there are almost 3 terabytes of data collected by the US Library of Congress by 2011. The Obama administration announced the Big Data research and development initiative in 2012, which investigate addressing important problems facing the government by make use of Big Data. The initiative was constitutive of 84 different Big Data programs involving six departments.[1] The similar thing also happened in Europe. Governments around the world are facing adverse conditions to improve their productivity. Namely, they are required to be more effective in public administration. Particularly in the recent global recession, many governments have to provide a higher level of public services with significant budgetary constraints. Therefore, they should take Big Data as a potential budget resource and develop tools to get alternative solutions to decrease big budget deficits and reduce national debt levels.

According to McKinsey's report [114], Big Data functionalities, such as reserving informative patterns and knowledge, provide the public sector a chance to improve productivity and higher levels of efficiency and effectiveness. European's public sector could potentially reduce expenditure of administrative activities by 15–20 percent, increasing 223 billion to 446 billion values, or even more. This estimate is under efficiency gains and a reduction in the difference between actual and

[1] http://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal.

potential aggregate of tax revenue. These functionalities could speed up year productivity growth by up to 0.5 percentage points over the next decade.

2.3. Big Data in scientific research

Many scientific fields have already become highly data-driven [179,31] with the development of computer sciences. For instance, astronomy, meteorology, social computing [187], bioinformatics [100] and computational biology [117] are greatly based on data-intensive scientific discovery as large volume of data with various types generated or produced in these science fields [45]. How to probe knowledge from the data produced by large-scale scientific simulation? It is a certain Big Data problem which the answer is still unsatisfiable or unknown.

For instances, a sophisticated telescope is regarded as a very large digital camera which generate huge number of universal images. For example, the Large Synoptic Survey Telescope (LSST) will record 30 trillion bytes of image data in a single day. The size of the data equals to two entire Sloan Digital Sky Surveys daily. Astronomers will utilize computing facilities and advanced analysis methods to this data to investigate the origins of the universe. The Large Hadron Collider (LHC) is a particle accelerator that can generate 60 terabytes of data per day [29]. The patterns in those data can give us an unprecedented understanding the nature of the universe. 32 petabytes of climate observations and simulations were conserved on the discovery supercomputing cluster in the NASA Center for Climate Simulation (NCCS). The volume of human genome information is also so large that decoding them originally took a decade to process. Otherwise, a lot of other e-Science projects [66] are proposed or underway in a wide variety of other research fields, range from environmental science, oceanography and geology to biology and sociology. One common point exists in these disciplines is that they generate enormous data sets that automated analysis is highly required. Additionally, centralized repository is necessary as it is impractical to replicate copies for remote individual research groups. Therefore, centralized storage and analysis approaches drive the whole system designs.

3. Big Data opportunities and challenges

3.1. Opportunities

Recently, several US government agencies, such as the National Institutes of Health (NIH) and the National Science Foundation (NSF), ascertain that the utilities of Big Data to data-intensive decision-making have profound influences in their future developments [1]. Consequently, they are trying to developing Big Data technologies and techniques to facilitate their missions after US government passed a large-scale Big Data initiative. This initiative is very helpful for building new capabilities for exploiting informative knowledge and facilitate decision-makers.

From the Networking Information Technology Research and Development (NITRD) program which is recently recognized by President's Council of Advisors on Science and Technology (PCAST), we know that the bridges between Big Data and knowledge hidden in it are highly crucial in all areas of national priority. This initiative will also lay the groundwork for complementary Big Data activities, such as Big Data infrastructure projects, platforms development, and techniques in settling complex, data-driven problems in sciences and engineering. Finally, they will be put into practice and benefit society.

According to the report from McKinsey institute [114], the effective use of Big Data has the underlying benefits to transform economies, and delivering a new wave of productive growth. Taking advantages of valuable knowledge beyond Big Data will become the basic competition for today's enterprises and will create new competitors who are able to attract employees that have the critical skills on Big Data. Researchers, policy and decision makers have to recognize the potential of harnessing Big Data to uncover the next wave of growth in their fields. There are many advantages in business section that can be obtained through harnessing Big Data as illustrated in Fig. 2, including increasing operational efficiency, informing strategic direction, developing better customer service, identifying and developing new products and services,

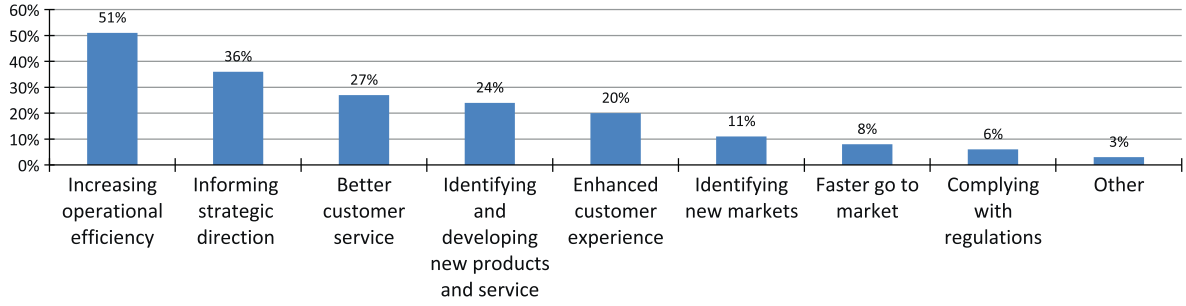

Fig. 2. Big Data Opportunities: above 50% of 560 enterprises think Big Data will help them in increasing operational efficiency, etc.

identifying new customers and markets, etc. The vertical axis denotes the percentages that how many enterprises think Big Data can help them with respect to specific purposes.

By liberal estimates [114], Big Data could produce $300 billion potential annual value to US health care, and €250 billion to European public administration. There will be $600 billion potential annual consumer surplus from using personal location data globally, and give a potential increase with 60%. Only in United States, Big Data produce 140,000 to 190,000 deep analytical talent positions and 1.5 million data-savvy managers. Undoubtedly, Big Data is usually juicy and lucrative if explored correctly.

3.2. Challenges

Opportunities are always followed by challenges. On the one hand, Big Data bring many attractive opportunities. On the other hand, we are also facing a lot of challenges [137] when handle Big Data problems, difficulties lie in data capture, storage, searching, sharing, analysis, and visualization. If we cannot surmount those challenges, Big Data will become a gold ore but we do not have the capabilities to explore it, especially when information surpass our capability to harness. One challenge is existing in computer architecture for several decades, that is, *CPU-heavy but I/O-poor* [65]. This system imbalance still restraint the development of the discovery from Big Data.

The CPU performance is doubling each 18 months following the Moore's Law, and the performance of disk drives is also doubling at the same rate. However, the disks' rotational speed has slightly improved over the last decade. The consequence of this imbalance is that random I/O speeds have improved moderately while sequential I/O speeds increase with density slowly. Moreover, information is increasing at exponential rate simultaneously, but the improvement of information processing methods is also relatively slower. In a lot of important Big Data applications, the state-of-the-art techniques and technologies cannot ideally solve the real problems, especially for real-time analysis. So partially speaking, until now, we do not have the proper tools to exploit the gold ores completely.

Typically, the analysis process is shown In Fig. 3, where the knowledge is discovered in data mining [59]. Challenges in Big Data analysis include data inconsistence and incompleteness, scalability, timeliness and data security [8,92]. As the prior step to data analysis, data must be well-constructed. However, considering variety of data sets in Big Data problems, it is still a big challenge for us to purpose efficient representation, access, and analysis of unstructured or semi-structured data in the further researches. How can the data be preprocessed in order to improve the quality data and the analysis results before we begin data analysis? As the sizes of data set are often very huge, sometimes several gigabytes or more, and their origin from heterogeneous sources, current real-world databases are severely susceptible to inconsistent, incomplete, and noisy data. Therefore, a number of data preprocessing techniques, including data cleaning, data integration, data transformation and date reduction, can be applied to remove noise and correct inconsistencies [59]. Different challenges arise in each sub-process when it comes to data-driven applications. In the following subsections, we will give a brief discussion about challenges we are facing for each sub-process.

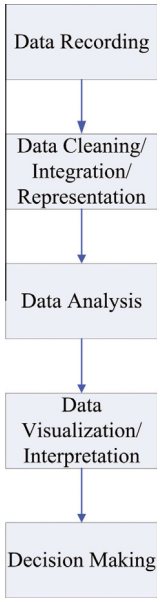

Fig. 3. Knowledge discovery process.

3.2.1. Data capture and storage

Data sets grow in size because they are increasingly being gathered by ubiquitous information-sensing mobile devices, aerial sensory technologies, remote sensing, software logs, cameras, microphones, radio-frequency identification readers, wireless sensor networks, and so on. There are 2.5 quintillion bytes of data created every day, and this number keeps increasing exponentially [67]. The world's technological capacity to store information has roughly doubled about every 3 years since the 1980s. In many fields, like financial and medical data often be deleted just because there is no enough space to store these data. These valuable data are created and captured at high cost, but ignored finally. The bulk storage requirements for experimental data bases, array storage for large-scale scientific computations, and large output files are reviewed in [194].

Big Data has changed the way we capture and store data [133], including data storage device, data storage architecture, data access mechanism. As we require more storage mediums and higher I/O speed to meet the challenges, there is no doubt that we need great innovations. Firstly, the accessibility of Big Data is on the top priority of the knowledge discovery process. Big Data should be accessed easily and promptly for further analysis, fully or partially break the restraint: CPU-heavy but I/O-poor. In addition, the under-developing storage technologies, such as solid-state drive (SSD) [73] and phase-change memory (PCM) [144], may help us alleviate the difficulties, but they are far from enough. One significant shift is also underway, that is the transformative change of the traditional I/O subsystems. In the past decades, the persistent data were stored by using hard disk drives (HDDs) [87]. As we known, HDDs had much slower random I/O performance than sequential I/O performance, and data processing engines formatted their data and designed their query processing methods to work around this limitation. But, HDDs are increasingly being replaced by SSDs today, and other technologies such as PCM are also around the corner [8]. These current storage technologies cannot possess the same high performance for both the sequential and random I/O simultaneously, which requires us to rethink how to design storage subsystems for Big Data processing systems.

Direct-attached storage (DAS), network-attached storage (NAS), and storage area network (SAN) are the enterprise storage architectures that were commonly used [99]. However, all these existing storage architectures have severe drawbacks and limitations when it comes to large-scale distributed systems. Aggressive concurrency and per server throughput are the essential requirements for the applications on highly scalable computing clusters, and today's storage systems lack the both. Optimizing data access is a popular way to improve the performance of data-intensive computing [78,77,79], these techniques include data replication, migration, distribution, and access parallelism. In [19], the performance, reliability and scalability in data-access platforms were discussed. Data-access platforms, such as CASTOR, dCache, GPFS and Scalla/Xrootd, are employed to demonstrate the large scale validation and performance measurement. Data storage and search schemes also lead to high overhead and latency [162], distributed data-centric storage is a good approach in large-scale wireless sensor networks (WSNs). Shen, Zhao and Li proposed a distributed spatial–temporal similarity data storage scheme to provide efficient spatial–temporal and similarity data searching service in WSNs. The collective behavior of individuals that cooperate in a swarm provide approach to achieve self-organization in distributed systems [124,184].

3.2.2. Data transmission

Cloud data storage is popularly used as the development of cloud technologies. We know that the network bandwidth capacity is the bottleneck in cloud and distributed systems, especially when the volume of communication is large. On the other side, cloud storage also lead to data security problems [190] as the requirements of data integrity checking. Many schemes were proposed under different systems and security models [189,134].

3.2.3. Data curation

Data curation is aimed at data discovery and retrieval, data quality assurance, value addition, reuse and preservation over time. This field specifically involves a number of sub-fields including authentication, archiving, management, preservation, retrieval, and representation. The existing database management tools are unable to process Big Data that grow so large and complex. This situation will continue as the benefits of exploiting Big Data allowing researchers to analyse business trends, prevent diseases, and combat crime. Though the size of Big Data keeps increasing exponentially, current capability to work with is only in the relatively lower levels of petabytes, exabytes and zettabytes of data. The classical approach of managing structured data includes two parts, one is a schema to storage the data set, and another is a relational database for data retrieval. For managing large-scale datasets in a structured way, *data warehouses* and *data marts* are two popular approaches. A data warehouse is a relational database system that is used to store and analyze data, also report the results to users. The data mart is based on a data warehouse and facilitate the access and analysis of the data warehouse. A data warehouse is mainly responsible to store data that is sourced from the operational systems. The preprocessing of the data is necessary before it is stored, such as data cleaning, transformation and cataloguing. After these preprocessing, the data is available for higher level online data mining functions. The data warehouse and marts are Standard Query Language (SQL) based databases systems.

NoSQL database [60], also called "Not Only SQL", is a current approach for large and distributed data management and database design. Its name easily leads to misunderstanding that NoSQL means "not SQL". On the contrary, NoSQL does not avoid SQL. While it is true that some NoSQL systems are entirely non-relational, others simply avoid selected relational functionality such as fixed table schemas and join operations. The mainstream Big Data platforms adopt NoSQL to break and transcend the rigidity of normalized RDBMS schemas. For instance, Hbase is one of the most famous used NoSQL databases

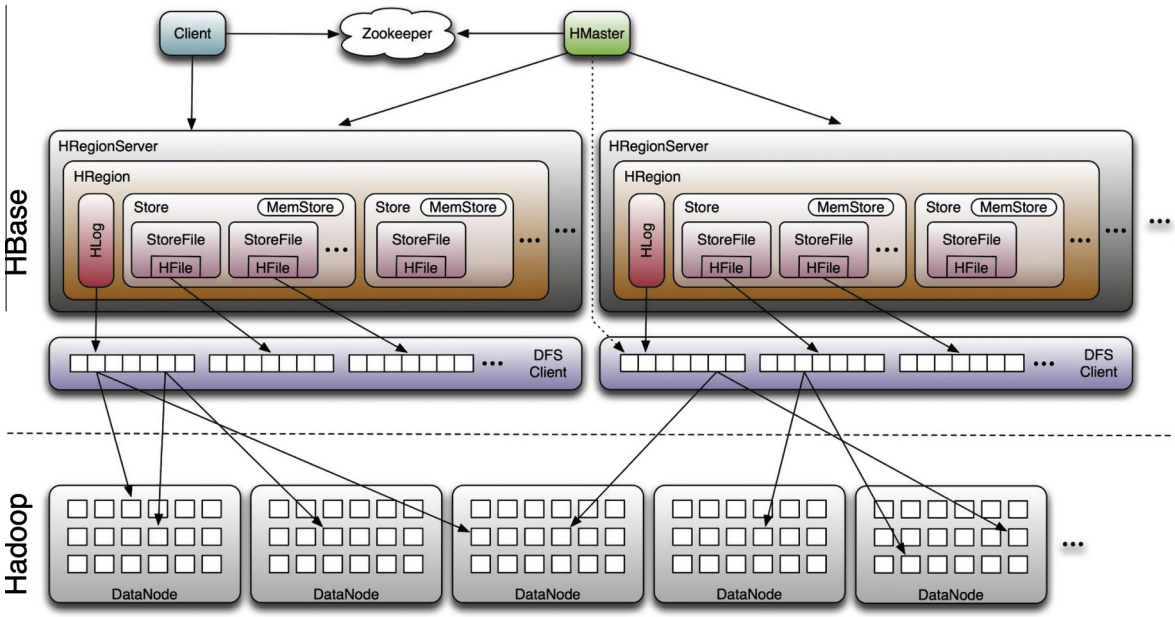

Fig. 4. Hbase NoSQL database system architecture. *Source:* from Apache Hadoop.

(see Fig. 4). However, many Big Data analytic platforms, like SQLstream and Cloudera Impala, series still use SQL in its database systems, because SQL is more reliable and simpler query language with high performance in stream Big Data real-time analytics.

To store and manage unstructured data or non-relational data, NoSQL employs a number of specific approaches. Firstly, data storage and management are separated into two independent parts. This is contrary to relational databases which try to meet the concerns in the two sides simultaneously. This design gives NoSQL databases systems a lot of advantages. In the storage part which is also called key-value storage, NoSQL focuses on the scalability of data storage with high-performance. In the management part, NoSQL provides low-level access mechanism in which data management tasks can be implemented in the application layer rather than having data management logic spread across in SQL or DB-specific stored procedure languages [37]. Therefore, NoSQL systems are very flexible for data modeling, and easy to update application developments and deployments [60].

Most NoSQL databases have an important property. Namely, they are commonly schema-free. Indeed, the biggest advantage of schema-free databases is that it enables applications to quickly modify the structure of data and does not need to rewrite tables. Additionally, it possesses greater flexibility when the structured data is heterogeneously stored. In the data management layer, the data is enforced to be integrated and valid. The most popular NoSQL database is Apache Cassandra. Cassandra, which was once Facebook proprietary database, was released as open source in 2008. Other NoSQL implementations include SimpleDB, Google BigTable, Apache Hadoop, MapReduce, MemcacheDB, and Voldemort. Companies that use NoSQL include Twitter, LinkedIn and NetFlix.

3.2.4. Data analysis

The first impression of Big Data is its volume, so the biggest and most important challenge is scalability when we deal with the Big Data analysis tasks. In the last few decades, researchers paid more attentions to accelerate analysis algorithms to cope with increasing volumes of data and speed up processors following the Moore's Law. For the former, it is necessary to develop sampling, on-line, and multiresolution analysis methods [59]. In the aspect of Big Data analytical techniques, *increment algorithms* have good scalability property, not for all machine learning algorithms. Some researchers devote into this area [180,72,62]. As the data size is scaling much faster than CPU speeds, there is a natural dramatic shift [8] in processor technology—although the clock cycle frequency of processors is doubling following Moore's Law, the clock speeds still highly lag behind. Alternatively, processors are being embedded with increasing numbers of cores. This shift in processors leads to the development of parallel computing [130,168,52].

For those real-time Big Data applications, like navigation, social networks, finance, biomedicine, astronomy, intelligent transport systems, and internet of thing, timeliness is at the top priority. How can we grantee the timeliness of response when the volume of data will be processed is very large? It is still a big challenge for stream processing involved by Big Data. It is right to say that Big Data not only have produced many challenge and changed the directions of the development of the hardware, but also in software architectures. That is the swerve to *cloud computing* [50,186,7,48], which aggregates multiple disparate workloads into a large cluster of processors. In this direction, distributed computing is being developed at high speed recently. We will give a more detail discussion about it in next section.

Data security surfaces with great attentions. Significant security problems include data security protection, intellectual property protection, personal privacy protection, commercial secrets and financial information protection [172]. Most developed and developing countries have already made related data protection laws to enhance the security. Research groups and individuals need to carefully consider the legislation of where they store and process data to make sure that they are in compliance with the regulations. For Big Data related applications, data security problems are more awkward for several reasons. Firstly, the size of Big Data is extremely large, channelling the protection approaches. Secondly, it also leads to much heavier workload of the security. Otherwise, most Big Data are stored in a distributed way, and the threats from networks also can aggravate the problems.

3.2.5. Data visualization

The main objective of data visualization [169,88] is to represent knowledge more intuitively and effectively by using different graphs. To convey information easily by providing knowledge hidden in the complex and large-scale data sets, both aesthetic form and functionality are necessary. Information that has been abstracted in some schematic forms, including attributes or variables for the units of information, is also valuable for data analysis. This way is much more intuitive [169] than sophisticated approaches. Online marketplace eBay, have hundreds of million active users and billions of goods sold each month, and they generate a lot of data. To make all that data understandable, eBay turned to Big Data visualization tool: Tableau, which has capability to transform large, complex data sets into intuitive pictures. The results are also interactive. Based on them, eBay employees can visualize search relevance and quality to monitor the latest customer feedback and conduct sentiment analysis.

For Big Data applications, it is particularly difficult to conduct data visualization because of the large size and high dimension of Big Data. However, current Big Data visualization tools mostly have poor performances in functionalities, scalability and response time. What we need to do is rethinking the way we visualize Big Data, not like the way we adopt before. For example, the history mechanisms for information visualization [64] also are data-intensive and need more efficient approaches. Uncertainty can lead to a great challenge to effective uncertainty-aware visualization and arise in any stage of a visual analytics process [195]. New framework for modeling uncertainty and characterizing the evolution of the uncertainty information are highly necessary through analytical processes.

The shortage of talent will be a significant constraint to capture values from Big Data [114]. In the United States, Big Data is expected to rapidly become a key determinant of competition across many sectors. However, this area demands for deep analytical positions on Big Data could exceed the supply being produced on current trends by $140,000$ to $190,000$ positions [114]. Furthermore, this kind of human resource is more difficult to educate. It usually takes many years to train Big Data analysts that must have intrinsic mathematical abilities and related professional knowledge. We believe that the same situation also happened in other nations, not matter developed countries or developing countries around the world. It is foreseeable that there will be another hot competition about human resources in Big Data developments.

After review a number of challenges, the optimists take a broad view challenges and hidden benefits. They have enough confidence that we have the capabilities to overcome all the obstacles as new techniques and technologies are developed. There are many critiques and negative opinions [81,167,11] from the pessimists. Some researchers think Big Data will lead to the end of theory, and doubt whether it can help us to make better decisions. Whatever, the mainstream perspectives are most positive, so a large number of Big Data techniques and technologies have been developed or under developing.

4. Big Data tools: techniques and technologies

To capture the value from Big Data, we need to develop new techniques and technologies for analyzing it. Until now, scientists have developed a wide variety of techniques and technologies to capture, curate, analyze and visualize Big Data. Even so, they are far away from meeting variety of needs. These techniques and technologies cross a number of discipline, including computer science, economics, mathematics, statistics and other expertises. Multidisciplinary methods are needed to discovery the valuable information from Big Data. We will discuss current techniques and technologies for exploiting data-intensive applications.

We need tools (platforms) to make sense of Big Data. Current tools concentrate on three classes, namely, batch processing tools, stream processing tools, and interactive analysis tools. Most batch processing tools are based on the Apache Hadoop infrastructure, such as Mahout and Dryad. The latter is more like necessary for real-time analytic for stream data applications. Storm and S4 are good examples for large scale streaming data analytic platforms. The interactive analysis processes the data in an interactive environment, allowing users to undertake their own analysis of information. The user is directly connected to the computer and hence can interact with it in real time. The data can be reviewed, compared and analyzed in tabular or graphic format or both at the same time. Google's Dremel and Apache Drill are Big Data platforms based on interactive analysis. In the following sub-sections, we'll discuss several tools for each class. More information about Big Data tools can be found in Appendix C.

4.1. Big Data techniques

Big Data needs extraordinary techniques to efficiently process large volume of data within limited run times. Reasonably, Big Data techniques are driven by specified applications. For example, Wal-Mart applies machine learning and statistical

techniques to explore patterns from their large volume of transaction data. These patterns can produce higher competitiveness in pricing strategies and advertising campaigns. Taobao (A Chinese company like eBay) adopts large stream data mining techniques on users' browse data recorded on its website, and exploits a good deal of valuable information to support their decision-making.

Big Data techniques involve a number of disciplines, including statistics, data mining, machine learning, neural networks, social network analysis, signal processing, pattern recognition, optimization methods and visualization approaches. There are many specific techniques in these disciplines, and they overlap with each other hourly (illustrated as Fig. 5).

Optimization Methods have been applied to solve quantitative problems in a lot of fields, such as physics, biology, engineering, and economics. In [153], several computational strategies for addressing global optimization problems are discussed, such as simulated annealing, adaptive simulated annealing, quantum annealing, as well as genetic algorithm which naturally lends itself to parallelism and therefore can be highly efficient. Stochastic optimization, including genetic programming, evolutionary programming, and particle swarm optimization are useful and specific optimization techniques inspired by the process of nature. However, they often have high complexity in memory and time consumption. Many research works [102,44,198] have been done to scale up the large-scale optimization by cooperative co-evolutionary algorithms. Real-time optimization [160] is also required in many Big Data application, such as WSNs and ITSs. Data reduction [197] and parallelization [173,35,199] are also alternative approaches in optimization problems.

Statistics is the science to collect, organize, and interpret data. Statistical techniques are used to exploit correlationships and causal relationships between different objectives. Numerical descriptions are also provided by statistics. However, standard statistical techniques are usually not well suited to manage Big Data, and many researchers have proposed extensions of classical techniques or completely new methods [41]. Authors in [132] proposed efficient approximate algorithm for large-scale multivariate monotonic regression, which is an approach for estimating functions that are monotonic with respect to input variables. Another trend of data-driven statistical analysis focuses on scale and parallel implementation of statistical algorithms. A survey of parallel statistics can be found in [141], and several parallel statistics algorithms are discussed in [22]. *Statistical computing* [91,193] and *statistical learning* [63] are the two hot research sub-fields.

Data mining is a set of techniques to extract valuable information (patterns) from data, including clustering analysis, classification, regression and association rule learning. It involves the methods from machine learning and statistics. Big Data mining is more challenging compared with traditional data mining algorithms. Taking clustering as an example, a natural way of clustering Big Data is to extend existing methods (such as hierarchical clustering, K-Mean, and Fuzzy C-Mean) so that they can cope with the huge workloads [205,39,24]. Most extensions usually rely on analyzing a certain amount of samples of Big Data, and vary in how the sample-based results are used to derive a partition for the overall data. This kind of clustering algorithms [90] include CLARA (Clustering LARge Applications) algorithm, CLARANS (Clustering Large Applications based upon RANdomized Search), BIRCH (Balanced Iterative Reducing using Cluster Hierarchies) algorithm, and so on. Genetic algorithms are also applied to clustering as optimization criterion to reflect the goodness.

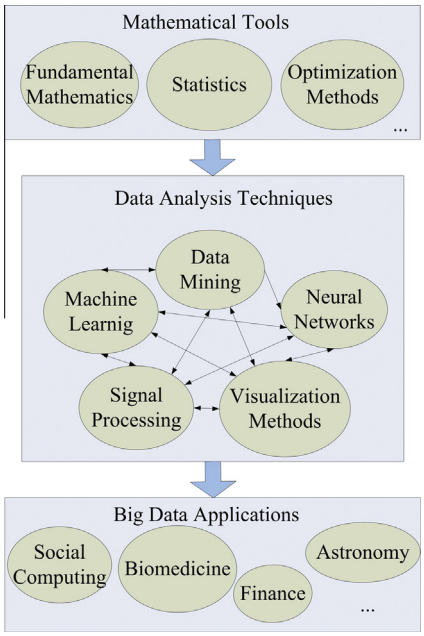

Fig. 5. Big Data techniques.

Clustering Big Data is also developing to distributed and parallel implementation [150]. Taking discriminant analysis as another example, researchers try to develop effective algorithm for large-scale discriminant analysis [33,165]. The emphasis is on the reduction of computational complexity. Taking bioinformatics as another example, it becomes increasingly data-driven that leads to paradigm change from traditional single-gene biology to the approaches that combine integrative database analysis and data mining [23]. This new paradigm enables the synthesis of large-scale portraits of genome function.

Machine learning is an important subjection of artificial intelligence which is aimed to design algorithms that allow computers to evolve behaviors based on empirical data. The most obvious characteristic of machine learning is to discovery knowledge and make intelligent decisions automatically. When Big Data is concerned, we need to scale up machine learning algorithms, both supervised learning and unsupervised learning, to cope with it. *Deep machine learning* has become a new research frontier in artificial intelligence [68,21,69,13,200]. In addition, there are several frameworks, like *Map/Reduce*, *DryadLINQ*, and *IBM parallel machine learning toolbox*, that have capabilities to scale up machine learning. For example, Support Vector Machine (SVM), which is a very fundamental algorithm used in classification and regression problems, suffers from serious scalability problem in both memory use and computation time. Parallel SVM (PSVM) [196,121] are introduced recently to reduce memory and time consumption [17]. There are many scale machine learning algorithms [152,177,106,85], but many important specific sub-fields in large-scale machine learning, such as large-scale recommender systems, natural language processing, association rule learning, ensemble learning, still face the scalability problems.

Artificial neural network (ANN) is a mature techniques and has a wide range of application coverage. Its successful applications can be found in pattern recognition, image analysis, adaptive control, and other areas. Most of the currently employed ANNs for artificial intelligence are based on statistical estimations [113], classification optimization [131] and control theory [105]. It is generally acknowledged, the more hidden layers and nodes in a neural network, the higher accuracy they can produce. However, the complexity in a neural network also increases the learning time. Therefore, the learning process in a neural networks over Big Data is severely time and memory consuming [166,206]. Neural processing of large-scale data sets often leads to very large networks. Then, there are two main challenges in this situation. One is that the conventional training algorithms perform very poorly, and the other is that the training time and memory limitations are increasingly intractable [161,49]. Naturally, two common approaches can be employed in this situation. One is to reduce the data size by some sampling methods, and the structure of the neural network maybe remains the same. The other one is to scale up neural networks in parallel and distributed ways [119,40,9,202]. For example, the combination of deep learning and parallel training implementation techniques provides potential ways to process Big Data [20,97,42].

Visualization Approaches [169] are the techniques used to create tables, images, diagrams and other intuitive display ways to understand data. Big Data visualization [88,64,53] is not that easy like traditional relative small data sets because of the complexity in 3Vs or 4Vs. The extension of traditional visualization approaches are already emerged but far away from enough. When it comes to large-scale data visualization, many researchers use feature extraction and a geometric modeling to significantly reduce the data's size before the actual data rendering [103,182]. For more closely and intuitively data interpretation, some researchers try to run batch-mode software rendering of the data at the highest possible resolution in a parallel way [112,10]. Choosing proper data representation is also very important when we try to visualize Big Data. In [182], author tried to compact data and give a good approximation to large-scale data.

Social Network Analysis (SNA) which has emerged as a key technique in modern sociology, views social relationships in terms of network theory, and it consists of nodes and ties. It has also gained a significant following in anthropology, biology, communication studies, economics, geography, history, information science, organizational studies, social psychology, development studies, and sociolinguistics and is now commonly available as a consumer tool. SNA include social system design [204], human behavior modeling [95], social network visualization [127,164], social networks evolution analysis [27], and graph query and mining [111]. Recently, online social networks and *Social media analysis* have become popular [104,82]. One of the main obstacles regarding SNA is the vastness of Big Data. Analysis of a network consisting of millions or billions of connected objects is usually computationally costly. Two hot research frontiers, *social computing* and *cloud computing*, are in favor of SNA to some degree.

Higher level Big Data technologies include distributed file systems [148,32], distributed computational systems [136], massively parallel-processing (MPP) systems [185,181], data mining based on grid computing [34], cloud-based storage and computing resources [154], as well as granular computing and biological computing. These technologies will be introduced in the following sub-sections.

Many researchers regard the curse of dimensionality as one aspect of Big Data problems. Indeed, Big Data should not be constricted in data volume, but all take the high-dimension characteristic of data into consideration. In fact, processing high-dimensional data is already a tough task in current scientific research. The state-of-the-art techniques for handling high-dimensional data intuitively fall into dimension reduction. Namely, we try to map the high-dimensional data space into lower dimensional space with less loss of information as possible. There are a large number of methods to reduce dimension [147,109,56]. Linear mapping methods, such as principal component analysis (PCA) and factor analysis, are popular linear dimension reduction techniques. Non-linear techniques include kernel PCA, manifold learning techniques such as Isomap, locally linear embedding (LLE), Hessian LLE, Laplacian eigenmaps, and LTSA [98]. Recently, a generative deep networks, called autoencoder [70], perform very well as non-linear dimensionality reduction. Random projection in dimensionality reduction also have been well-developed [25].

4.2. Big Data tools based on batch processing

One of the most famous and powerful batch process-based Big Data tools is Apache Hadoop. It provides infrastructures and platforms for other specific Big Data applications. A number of specified Big Data systems (Table 1) are built on Hadoop, and have special usages in different domains, for example, data mining and machine learning used in business and commerce.

4.2.1. Apache Hadoop and map/reduce

Apache Hadoop is one of the most well-established software platforms that support data-intensive distributed applications. It implements the computational paradigm named *Map/Reduce*. Apache Hadoop (see Fig. 6) platform consists of the Hadoop kernel, Map/Reduce and Hadoop distributed file system (HDFS), as well as a number of related projects, including Apache Hive, Apache HBase, and so on.

Map/Reduce [43], which is a programming model and an execution for processing and generating large volume of data sets, was pioneered by Google, and developed by Yahoo! and other web companies. Map/Reduce is based on the *divide and conquer method*, and works by recursively breaking down a complex problem into many sub-problems, until these sub-problems is scalable for solving directly. After that, these sub-problems are assigned to a cluster of working notes, and solved in separate and parallel ways. Finally, the solutions to the sub-problems are then combined to give a solution to the original problem. The divide and conquer method is implemented by two steps: *Map* step and *Reduce* step. In terms of Hadoop cluster, there are two kinds of nodes in Hadoop infrastructure. They are master nodes and worker nodes. The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes in Map step. Afterwards, the master node collects the answers to all the sub-problems and combines them in some way to form the output in Reduce step.

With the addition of Map/Reduce, Hadoop works as a powerful software framework [149,54] for easily writing applications which process vast quantities of data in-parallel on large clusters (perhaps thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. We give a famous and prototypical example that counts the occurrence number of each word in a set of documents for Map/Reduce framework, where the two main functions Map () and Reduce () are given in the following. More detailed Java code from Hadoop is attached in Appendix B. The Map steps are implemented on Hadoop cluster in a parallel way, and a large number of lists of intermediate data pairs with the form (key, c) are produced, where *key* represents a specified word, and the parameter *c* indicates the count of the word appearance. In Reduce steps, those lists of data pairs are integrated to the final results recursively in the main function.

As illustrated in Fig. 7, there are a master (JobTracker) and a number of slaves (TaskTracker) in the Map/Reduce framework. The master node is in charge of job scheduling and task distribution for the slaves. The slaves implement the tasks exactly as assigned by the master. As long as the systems start to run, the master node keeps monitoring all the data nodes. If there is a data nodes failed to execute the related tasks, the master node will ask the data node or another data node to re-execute the failed tasks. In practice, applications specify the input files and output locations, and submit their Map and Reduce functions via interactions of client interfaces. These parameters are important to construct a job configuration. After that, the Hadoop job client submits the job and configuration to the JobTracker. Once JobTracker receive all the necessary information, it will distribute the software/configuration to the TaskTrackers, schedule tasks and monitor them, provide status and diagnostic information to the job-client. From the foregoing, we know that coordination plays a very important role in Hadoop, it ensures the performance of a Hadoop job.

In [139], Andrew Pavlo gave an overall discussion on properties of Map/Reduce framework, as well as other approaches to large-scale data analysis. Many data mining algorithms have been designed to accommodate Map/Reduce. For example, data cube materialization and mining [126], efficient skyline computation [61] and scalable boosting methods [138].

4.2.2. Dryad

Dryad [75] is another popular programming models for implementing parallel and distributed programs that can scale up capability of processing from a very small cluster to a large cluster. It bases on dataflow graph processing [101]. The infra-

Table 1
Big Data tools based on batch processing.

Name	Specified Use	Advantage
Apache Hadoop	Infrastructure and platform	High scalability, reliability, completeness
Dryad	Infrastructure and platform	High performance distributed execution engine, good programmability
Apache Mahout	Machine learning algorithms in business	Good maturity
Jaspersoft BI Suite	Business intelligence software	Cost-effective, self-service BI at scale
Pentaho Business Analytics	Business analytics platform	Robustness, scalability, flexibility in knowledge discovery
Skytree Server	Machine learning and advanced analytics	Process massive datasets accurately at high speeds
Tableau	Data visualization, Business analytics,	Faster, smart, fit, beautiful and ease of use dashboards
Karmasphere Studio and Analyst	Big Data Workspace	Collaborative and standards-based unconstrained analytics and self service
Talend Open Studio	Data management and application integration	Easy-to-use, eclipse-based graphical environment

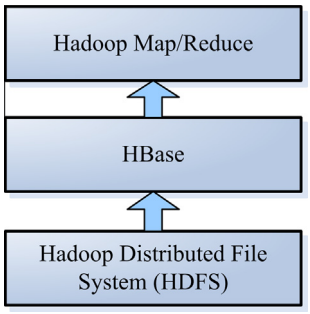

Fig. 6. Hadoop system architecture.

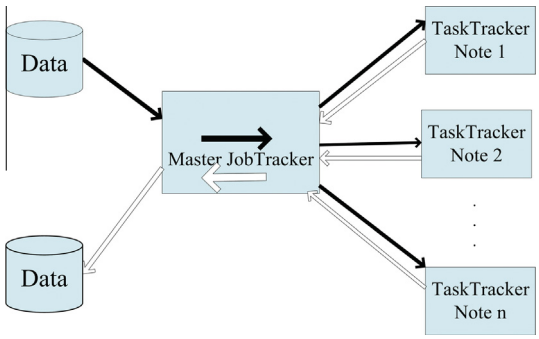

Fig. 7. Map/reduce overview: solid arrows are for Map flows, and feint arrows are for reduce flows.

structure for running Dryad consists of a cluster of computing nodes, and a programmer use the resources of a computer cluster to running their programs in a distributed way. Indeed, Dryad programmers can use thousands of machines, each of them with multiple processors or cores. One bonus is that programmers does not need to know anything about concurrent programming. A Dryad application runs a computational directed graph which is composed of computational *vertices* and communication *channels*. The computation is structured as illustrated in Fig. 8: graph vertices represent the programs, while graph edges denote the channels. A Dryad programmer writes several sequential programs and connects them using one-way channels [76]. A Dryad job is to generator a graph, and it has capability to synthesize any directed acyclic graph. These generated graphs can also be updated after execution, in order to deal with the unexpected events in the computation.

Dryad provides a large number of functionality, including generating the job graph, scheduling the processes on the available machines, handling transient failures in the cluster, collecting performance metrics, visualizing the job, invoking user-defined policies and dynamically updating the job graph in response to these policy decisions, without awareness of the semantics of the vertices [101]. Fig. 9 schematically shows the implementation schema of Dryad. There is a centralized job manager to supervise every Dryad job. It uses a small set of cluster services to control the execution of the vertices on the cluster.

Because Dryad encompasses other computational frameworks like Map/Reduce and the relational algebra, it is more complex and powerful in some degree. Otherwise, Dryad is a self-contained system with complete functions including job creation and management, resource management, job monitoring and visualization, fault tolerance, re-execution. Therefore,

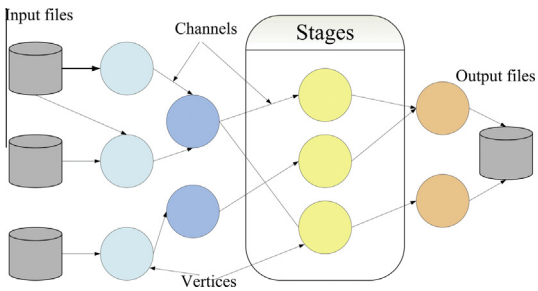

Fig. 8. The structure of dryad jobs.

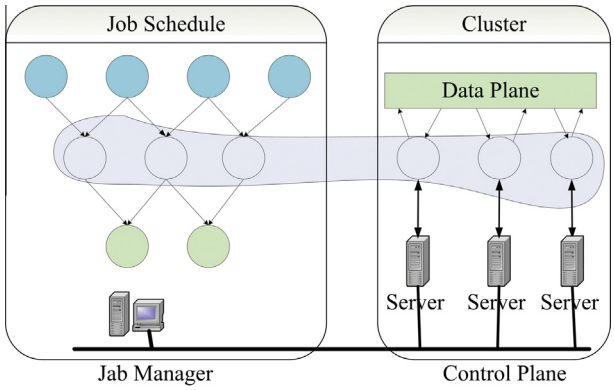

Fig. 9. Dryad architecture.

many software have been built on top Dryad, including Microsoft's Server 2005 Integration Services (SSIS) [120] and DryadLINQ [201].

4.2.3. Apache mahout

The Apache Mahout [74] aims to provide scalable and commercial machine learning techniques for large-scale and intelligent data analysis applications. Many renowned big companies, such as Google, Amazon, Yahoo!, IBM, Twitter and Facebook, have implemented scalable machine learning algorithms in their projects. Many of their projects involve with Big Data problems and Apache Mahout provides a tool to alleviate the big challenges.

Mahout's core algorithms, including clustering, classification, pattern mining, regression, dimension reduction, evolutionary algorithms and batch based collaborative filtering, run on top of Hadoop platform via the Map/reduce framework [46,47]. These algorithms in the libraries have been well-designed and optimized to have good performance and capabilities. A number of non-distributed algorithms are also contained. The goal of Mahout is to build a vibrant, responsive, diverse community to facilitate discussions not only on the project itself but also on potential use cases. The business users need to purchase Apache software license for Mahout. More detailed content can be found on the web site: http://mahout.apache.org/.

4.2.4. Jaspersoft BI suite

The Jaspersoft package is an open source software that produce reports from database columns. The software has already been installed in many business information systems. It is an scalable Big Data analyticical platform and fast to get started with no need for ETL. Jaspersoft [191] has a capability of fast data visualization on popular storage platforms, including MongoDB, Cassandra, Redis, Riak, and CouchDB. Hadoop is also represented very well by JasperReports, which provides a Hive connector to HBase. Because it is integrated with all the leading Big Data platforms, users have the flexibility to choose what works best for their own projects.

One important property of Jaspersoft is that it can quickly explore Big Data without extraction, transformation (ETL), and loading. As connected directly to mainstream Big Data stores, whether or not they have a SQL interface, it explores large-scale data using HTML5 visualizations powered by a terabyte-scale, columnar-based in-memory engine. Optimizing performance through an in-memory engine that can push down query processing to the underlying data store if necessary. Jaspersoft also have a ability to build powerful HTML5 reports and dashboards interactively and directly from Big Data store, without ETL requirement. These reports can be shared with anyone inside or outside user's organizations or embedded in user's applications.

4.2.5. Pentaho business analytics

Pentaho [4] is another software platform for Big Data. It also generate reports from both structured and unstructured large volume of data. Pentaho plays as a business analytic platform for Big Data to provide professional services for businessmen with facile access, integration, visualization and exploration of data. Therefore, Pentaho can enable business users to make data-driven decisions that have a positive effect on the performance of their organization. The techniques embedded in it have several properties, including good security, scalability, and accessibility. Similar with JasperSoft, there is a chain between Pentaho's tool and many of the most popular NoSQL databases, such as MongoDB [145] and Cassandra [36]. Once the connection to databases is established, users can drill up and drill down the columns into different information granules.

Business users can access their data though a web-based interface that Pentaho provided. With its easy way to use wizard-based approach, business users can turn their data into insight and make information-driven decisions very fast. The graphical programming interface developed by Pentaho, such as Kettle and Pentaho Data Integration, are very powerful tools to process massive data. Pentaho also develops softwares that are based on Hadoop clusters to draw HDFS file data and

HBase data, so users just need to write their code and send them out to execute on the cluster. By this way, the data analytical processes are highly escalated.

4.2.6. Skytree server

Skytree Server [156] is the first general purpose machine learning and advanced analytics system, designed to accurately process massive datasets at high speeds. It offers many sophisticated machine learning algorithms. It is easy to use, and users just need to type the right command into a command line. Skytree Server has five specific use cases, namely, recommendation systems, anomaly/outlier identification, predictive analytics, clustering and market segmentation, and similarity search.

Skytree is more focused on real-time analytics. Therefore, it is optimized to implement a number of sophisticated machine learning algorithms on Big Data via a mechanism, which the company claims can be 10,000 times faster than other congeneric platforms. It also can handle structured and unstructured data from relational databases, HDFS, flat files, common statistical packages, and machine learning libraries.

4.2.7. Tableau

Tableau [28] has three main products to process large-scale data set, including Tableau Desktop, Tableau Sever, and Tableau Public. Tableau Desktop is a visualization tool that makes it easy to visualize data and look at it in a different and intuitive way. The tool is optimized to give user all the columns for the data and let users mix them. Tableau Server is a business intelligence system that provides browser-based analytics, and Tableau Public is used to create interactive visuals.

Tableau also embed Hadoop infrastructure. It employs Hive to structure the queries, and cache the information for in-memory analytics. Caching helps to reduce the latency of a Hadoop cluster. Therefore, it can provide an interactive mechanism between users and Big Data applications.

4.2.8. Karmasphere studio and analyst

Karmasphere [3] is another Hadoop-based Big Data platform for business data analysis. It provides a new approach for self-service access and analytics to Big Data in a fast, efficient and collaborative way. Karmasphere is natively designed for Hadoop platform, it provides users an integrated and user-friendly workspace for processing their Big Data applications and presenting the workflows. From the point of its performance, it has capability to discovery business insight from huge amounts of data, including data ingestion, iterative analysis, visualization and reporting. Karmasphere Studio is a set of plug-ins built on top of Eclipse. In this well-designed integrated development environment, users can easily write and implement their Hadoop jobs on the platform.

Karmasphere Analyst is a Big Data tool which is designed by Karmasphere to escalate the analytical process on Hadoop clusters. In addition, Karmasphere Analyst also embeds Hive project for processing structured and unstructured data on Hadoop clusters. Technical analysts, SQL programmers, and database administrator can experiment with Hadoop in graphical environment. This also makes Karmasphere Analyst to be an enterprise-class Big Data platform.

4.2.9. Talend Open Studio

Talend Open Studio [28] is an open source software for Big Data applications that provides users graphical environment to conduct their analysis visually. It is developed from Apache Hadoop and involves HDFS, Pig, HCatalog, HBase, Sqoop or Hive. Users can resolve their Big Data problems in this platform without the need to write complicated Java code which cannot be avoided in Hadoop.

By using Talend Studio, users can build up their own tasks through dragging and dropping varieties of icons onto a canvas. Stringing together blocks visually can be simple after users get a feel for what the components actually do and do not do. Visual programming seems like a superordinate goal, but the icons can never represent the mechanisms with enough detail to make it possible to deeply understand. Talend Open Studio also provides Really Simple Syndication (RSS) feed, and its components maybe collect the RSS and add proxying if needed.

4.3. Stream processing Big Data tools

Hadoop does well in processing large amount of data in parallel. It provides a general partitioning mechanism to distribute aggregation workload across different machines. Nevertheless, Hadoop is designed for batch processing. It is a multi-purpose engine but not a real-time and high performance engine, since there are high throughout latency in its implementations. For certain stream data applications, such as processing log files, industry with sensor, machine-to-machine (M2M) and telematics requires real-time response for processing large amount of stream data. In those applications, stream processing for real-time analytics is mightily necessary. Stream Big Data has high volume, high velocity and complex data types. Indeed, when the high velocity and time dimension are concerned in applications that involve real-time processing, there are a number of different challenges to Map/Reduce framework. Therefore, the real-time Big Data platforms, such as SQLstream [6], Storm and StreamCloud [57], are designed specially for real-time stream data analytics.

Real-time processing means that the ongoing data processing highly requires a very low latency of response. Hence, there is not too much data accumulation at the time dimension for processing [175]. In general, Big Data may be collected and stored in a distributed environment, not in one data center. In the general Map/Reduce framework, the Reduce phase starts to work only after the Map phase finish up. But most of all, all the intermediate data generated in Map phase is saved in the

Table 2
Big Data tools based on stream processing.

Name	Specified use	Advantages
Storm	Realtime computation system	Scalable, fault-tolerant, and is easy to set up and operate
S4	Processing continuous unbounded streams of data	Proven, distributed, scalable, fault-tolerant, pluggable platform
SQLstream s-Server	Sensor, M2M, and telematics applications	SQL-based, real-time streaming Big Data platform
Splunk	Collect and harness machine data	Fast and easy to use, dynamic environments, scales from laptop to datacenter
Apache Kafka	Distributed publish-subscribe messaging system	High-throughput stream of immutable activity data
SAP Hana	Platform for real-time business	Fast in-memory computing and realtime analytic

disk before submit to the reducers for next phase. All these lead to significant latency of the processing. The high latency characteristic of Hadoop makes it almost impossible for real-time analytics.

Several Big Data tools based on stream processing have been developed or under developing. One of the most famous platforms is Storm, and others include S4 [128], SQLstream [5], Splunk [155], Apache Kafka [14], and SAP Hana [93] (see Table 2).

4.3.1. Storm

Storm [6] is a distributed and fault-tolerant real-time computation system for processing limitless streaming data. It is released as open source and free for remoulding. Storm is specifically designed for real-time processing, contrasts with Hadoop which is for batch processing. It is also very easy to set up and operate, and guarantees all the data will be processed. It is also scalable and fault-tolerant to provide competitive performances. Storm is efficient that a benchmark clocked it at over a million tuples processed per second per node. Therefore, it has many applications, such as real-time analytics, interactive operation system, on-line machine learning, continuous computation, distributed RPC, and ETL.

A Storm cluster is ostensibly similar to a Hadoop cluster. Whereas on Storm users run different topologies for different Storm tasks. However, Hadoop platform implements Map/Reduce jobs for corresponding applications. There are a number

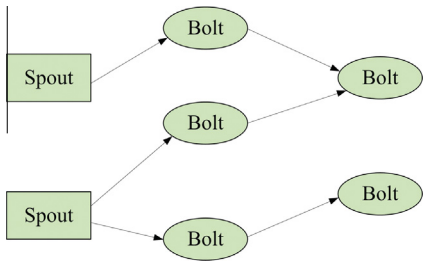

Fig. 10. A storm topology example.

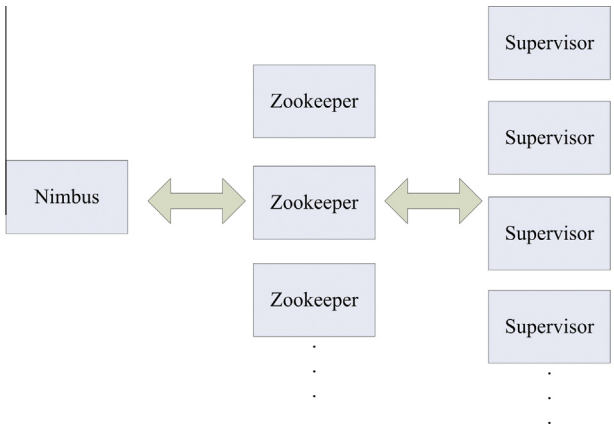

Fig. 11. A storm cluster.

of differences between Map/Reduce jobs and topologies. The key one is that a Map/Reduce job eventually finishes, whereas a topology processes messages all the time, or until users terminate it [86].

To implement real-time computation on Storm, users need to create different topologies. A topology (illustrated in Fig. 10) is a graph of computation and can be created and submitted in any programming language. There are two kinds of node in topologies, namely, spouts and bolts. A spout is one of the starting points in the graph, which denotes source of streams. A bolt processes input streams and outputs new streams. Each node in a topology contains processing logic, and links between nodes indicate how data should be processed between nodes. Therefore, a topology is a graph representing the transformations of the stream, and each node in the topology executes in parallel.

A Storm cluster consists of two kinds of working nodes. As illustrated in Fig. 11, they are only one master node and several worker nodes. The master node and worker nodes implement two kinds of daemons: Nimbus and Supervisor respectively. The two daemons have similar functions with according JobTracker and TaskTracker in Map/Reduce framework. Nimbus is in charge of distributing code across the Storm cluster, scheduling works assigning tasks to worker nodes, monitoring the whole system. If there is a failure in the cluster, the Nimbus will detect it and re-execute the corresponding task. The supervisor complies with tasks assigned by Nimbus, and starts or stops worker processes as necessary based on the instructions of Nimbus. The whole computational topology is partitioned and distributed to a number of worker processes, each worker process implements a part of the topology. How can Nimbus and the Supervisors work swimmingly and complete the job fast? Another kind of daemon called Zookeeper play an important role to coordinate the system. It records all states of the Nimbus and Supervisors on local disk.

4.3.2. S4

S4 [128] is a general-purpose, distributed, scalable, fault-tolerant, pluggable computing platform for processing continuous unbounded streams of data [94,115]. It was initially released by Yahoo! in 2010 and has become an Apache Incubator project since 2011. S4 allows programmers to easily develop applications, and possesses has several competitive properties, including robustness, decentralization, scalability, cluster management and extensibility [38].

The core platform of S4 is written in Java. The implementation of a S4 job is designed to be modular and pluggable for easily and dynamically processing large-scale stream data. S4 also employs Apache ZooKeeper to manage its cluster, like Storm does. S4 has been put to use in production systems at Yahoo! for processing thousands of search queries, and good performances show up in other applications.

4.3.3. SQLstream s-Server

SQLstream [5] is another Big Data platform that is designed for processing large-scale streaming data in real-time. It focuses on intelligent and automatic operations of streaming Big Data. SQLstream is appropriate to discovery patterns from large amounts of unstructured log file, sensor, network and other machine-generated data. The new release SQLstream s-Server 3.0 has good performances in real-time data collection, transformation and sharing, which is in favor of real-time Big Data management and analytics. The standard SQL language are still adopted in the underlying operations.

SQLstream works very fast, as it uses in-memory processing, also called "NoDatabase" technology. The data will not be stored in the disks. Instead of, the arriving data are regarded as streams and processed in-memory using streaming SQL queries. Streaming SQL is developed from strand SQL by taking advantage of multi-core computing, and achieves massively parallel streaming data processing.

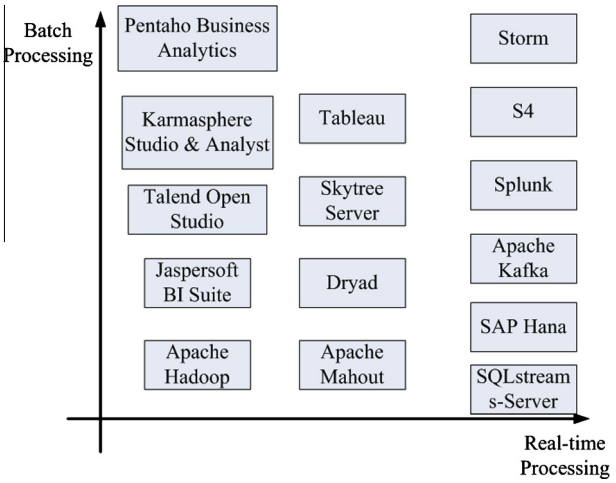

Fig. 12. Big Data platforms.

4.3.4. Splunk

Splunk is a real-time and intelligent Big Data platform for exploiting informations from machine-generated Big Data. It has been used in many famous companies, such as Amazen, Heroku and Senthub. Splunk combines the up-to-the-moment cloud technologies and Big Data to help users to search, monitor and analyze their machine-generated data via a web interface. It exhibits the results in an intuitive way, such as graphs, reports and alerts. Splunk is designed to provide metrics for many application, diagnose problems for system and IT infrastructures, and also provide intelligence for business operations. Splunk Storm [155] is a cloud version of Splunk's Big Data analytics.

Splunk is very different from the other stream processing tools. Its peculiarities include indexing structured or unstructured machine-generated data, real-time searching, reporting analytical results and dashboards. Therefore, log files are a great application for it.

4.3.5. Apache Kafka

Kafka [14] is a high-throughput messaging system that was incipiently developed at LinkedIn. It works as a tool to manage streaming and operational data via in-memory analytical techniques for obtaining real-time decision making. As a distributed publish-subscribe messaging system, Kafka has four main characteristics: persistent messaging with O (1) disk structures, high-throughput, support for distributed processing, and support for parallel data load into Hadoop. It already has wide usages in a number of different companies as data pipelines and messaging tools.

In recent years, activity and operational data play an important role to extract features of websites. Activity data is the record of various human actions on line, such as webpage content, copy content, clicklist, and searching key words. It is valuable to log these activities out into canned file and aggregate them for subsequent analysis. Operational data is data to describe the performance of servers, for instances, CPU and IO usage, request times, service logs, etc. The knowledge discovery of operational data is helpful for real-time operation management. Kafka combines off-line and on-line processing to provide real-time computation and produce ad hoc solution for these two kinds of data.

4.3.6. SAP Hana

SAP Hana [93] is an in-memory analytics platform that aims to provide real-time analysis on business processes, predictive analysis, and sentiment data processing. SAP HANA database is the core part of the real-time platform. It is a little bit different from other database systems. Operational reporting, data warehousing, and predictive and text analysis on Big Data are three HANA specific real-time analytics. SAP Hana works with very large scope of applications, whether or not they are from SAP, such as demographics and social media interactions.

4.4. Big Data tools based on interactive analysis

In recent years, Open source Big Data systems have emerged to address the need not only for scalable batch processing and stream processing, but also interactive analysis processing. The interactive analysis presents the data in an interactive environment, allowing users to undertake their own analysis of information. User are directly connected to the computer and hence can interact with it in real time. The data can be reviewed, compared and analyzed in tabular or graphic format or both at the same time.

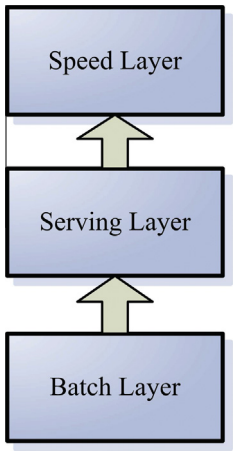

Fig. 13. Lambda architecture.

4.4.1. Google's Dremel

In 2010, Google proposed an interactive analysis system, named Dremel [118], which is scalable for processing nested data. Dremel has a very different architecture compared with well-known Apache Hadoop, and acts as a successful complement of Map/Reduce-based computations. It has capability to run aggregation queries over trillion-row tables in seconds by means of combining multi-level execution trees and columnar data layout. The system scales to thousands of CPUs and petabytes of data, and has thousands of users at Google.

4.4.2. Apache drill

Apache Drill is another distributed system for interactive analysis of Big Data [89]. It is similar to Google's Dremel. For Drill, there are more flexibility to support a various of different query languages, data formats and data sources. Like Dremel, Drill is also specifically designed to efficiently exploit nested data. It has an objective to scale up on 10,000 servers or more, and reaches the capability to process petabyes of data and trillions of records in seconds.

Drill and Dremel are experts in large-scale ad hoc querying of data. They use HDFS for storage and the Map/Reduce to perform batch analysis. By searching data either stored in columnar form or within a distributed file system, it is possible to scan over petabytes of data in seconds, to response ad hoc queries. Drill can be viewed as the open source version of Dremel. Google also provides Dremel-as-a-Service with its BigQuery offering. Other companies can design their own Big Data tools according to their special usages.

Every Big Data platform has its focus. Some of them are designed for batch processing, some are good at real-time analytic. Each Big Data platform also has specific functionality, for example, statistical analysis, machine learning, and data stream processing. We use Fig. 12 to illustrate their disadvantages and advantages, in which the capability of real-time processing increase (response time decrease) from left to right and handling capability of batch processing increase from bottom to up.

5. Principles for designing Big Data systems

Big Data analytics are doomed to be more complicated than traditional data analysis systems. How do we implement complex data-intensive tasks with satisfactory efficiency, especially in real-time? The answer is the capability to massively parallelize the analytical algorithms in such a way that all the processing happen entirely in memory and can linearly scale up and down on demand. When trying to exploit Big Data, we not only need to develop new technologies, but also new thinking ways. In designing Big Data analytics systems, we summarize seven necessary principles [116,51] to guide the development of this kind of burning issues. Big Data analytics in a highly distributed system cannot be achievable without the following principles:

Principle 1. Good architectures and frameworks are necessary and on the top priority.

Big Data cannot be solved effectively and approvingly if there are no good and proper architecture for the whole Big Data systems. In traditional information architecture, data sources that use integration techniques to transfer data into a DBMS data warehouse or operational data store, and then offer a wide variety of analytical techniques to reveal the data [174]. Then some organizations have applied oversight and standardization across projects, and perhaps have matured the information architecture capability through managing it [135]. However, Big Data systems need high-level architecture than traditional one. Many distributed and parallel processing architectures have already been proposed to address Big Data problems. There are distinct technology strategies for real-time and batch processing requirements. For real-time, key-value data stores, such as NoSQL, allow for high performance, index-based retrieval. For batch processing, Map/Reduce can be applied according to a specific data discovery strategy [174]. For different data-intensive applications, we should design different and appropriate architectures in the beginning of everything. For example, the Lambda architecture [116] solves the problem of computing arbitrary functions on arbitrary data in real-time by decomposing the problem into three layers: the batch layer, the serving layer, and the speed layer (Fig. 13). But this architecture cannot fit all the Big Data applications.

Principle 2. Support a variety of analytical methods

Big Data applications often produce complex tasks that make it impossible to be resolved by using one or a few of disciplines and analytical methods. The modern data science constantly involves a wide range of subjects and approaches. They range from data mining, statistical analysis, machine learning, distributed programming and visualization to real-time analysis, in-memory analysis and human–computer interaction. These methods are often synchronously employed in different Big Data platform [136].

Principle 3. No size fits all

When it comes to Big Data analytics, there is no one size which can fit all solutions, according to IBM's Latin America Big Data sales leader, Leonardo González [123]. Every Big Data tool has its own limitations, but if users use the proper tools for different tasks, they also can partially obtain significant benefits by using those tools. As the information keeps increasing at

exponential rate, today's Big Data problem will surely become the small data set problem in the future. Therefore, how can we deal with the big Big Data problems? Until now, we cannot answer this question, but this question will give us some directions when we try to design data-intensive systems, especially for real-time analytics.

Principle 4. Bring the analysis to data

As the size of Big Data set is extremely large, it is unadvisable and infeasible to collect and move data to only one or several centers for analysis. Data-driven analysis needs contrary analysis direction, which needs to bring the analysis tasks to data sites. In data-intensive computation problems, data is the driver not analytical human or machines. Together with systematic thinking, this principle is also resonant with the following ones.

Principle 5. Processing must be distributable for in-memory computation.

If the fourth principle holds as expected, we naturally require the processing must be distributable, since the analysis which will be carried out on different data sites must be distributed to data locations. In-memory analytic [51] which probes data stored in RAM rather than on disk, as is traditionally the case, is becoming popular because it speeds up the analysis process, even as data volumes explode. In-memory analytic is also highly necessary for real-time analytic. With the development of hard disk drives, there are almost no differences between memory and hard disk in I/O speed. Thinking about that, we believe that applications based on real-time analytic will highly benefit from in-memory analytic or in-memory-like analytic.

Principle 6. Data storage must be distributable for in-memory storage.

As a great portion of Big Data problems involve with the data and the information is generated at different addresses and different time, this principle is already met. But for the case that data generated or accumulated at data center, the data also need to be partitioned into a number of parts for in-memory analytic. The popular and potential technology cloud computing make the data storage in cloud. This is very appulsive in the Big Data problem solving. Once the data and services are stored in the cloud, users just like carry out their Big Data calculations on an unimaginative and powerful supercomputer. The real infrastructures are hidden in the cloud. Therefore, some data-driven applications can be realized.

Principle 7. Coordination is needed between processing and data units.

To improve scalability, as well as efficiency and fault-tolerance of Big Data systems, coordination between different processing units and data units on a cluster is highly necessary and essential. That is why both Storm and S4 employ independent and specialized cluster management frameworks (ZooKeeper) to control the whole data process. This principle guarantees the low latency of response which is particularly required in real-time analytics.

6. Underlying technologies and future researches

The advanced techniques and technologies for developing Big Data science is with the purpose of advancing and inventing the more sophisticated and scientific methods of managing, analyzing, visualizing, and exploiting informative knowledge from large, diverse, distributed and heterogeneous data sets. The ultimate aims are to promote the development and innovation of Big Data sciences, finally to benefit economic and social evolutions in a level that is impossible before. Big Data techniques and technologies should stimulate the development of new data analytic tools and algorithms and to facilitate scalable, accessible, and sustainable data infrastructure so as to increase understanding of human and social processes and interactions. As we discussed, the novel Big Data tools, techniques, and infrastructures will enable breakthrough discoveries and innovation in science, engineering, medicine, commerce, education, and national security – laying the foundations for competitiveness for many decades to come.

A paradigm shift in scientific investigation is on the way, as novel mathematical and statistical techniques, new data mining tools, advanced machine learning algorithms, as well as other data analytical disciplines, are well-established in the future. Consequently, a number of agencies are developing Big Data strategies to facilitate their missions. They focus on common interests in Big Data researches across the US National Institutes of Health and the US National Science Foundation. In the following subsections, we will discuss several ongoing or underlying techniques and technologies to harness Big Data, including granular computing, cloud computing, biological computing systems and quantum computing.

6.1. Granular computing

When we talk about Big Data, the first property of it is its size. As *granular computing* (GrC) [142] is a general computation theory for effectively using granules such as classes, clusters, subsets, groups and intervals to build an efficient computational model for complex applications with huge amounts of data, information and knowledge, therefore it is very natural to employ granular computing techniques to explore Big Data. Intuitively, granular computing can reduce the data size into different level of granularity. Under certain circumstances, some Big Data problems can be readily solved in such way.

GrC is a burgeoning conceptual and computing paradigm of knowledge discovery. In some degree, it has been motivated by the urgent need for efficient processing of Big Data, although the concept of Big Data have not be proposed when GrC is developing. In fact, GrC leads a significant transform from the current machine-centric to human-centric approach to information and knowledge. Theoretical foundations of granular computing are exceptionally sound and involve set theory (such as interval, box and ball), fuzzy sets [71], rough sets [83], and random sets [122] linked together in a highly comprehensive treatment of this paradigm. In [143], piecewise interval approximation and granular box regression are discussed to conduct data analysis. Su and his co-authors [176] introduced a new structure of radial basis function networks (RBFNs) that can successfully model symbolic interval-valued data. If Big Data can be transform into respective symbolic data, some algorithms in neural networks and machine learning come into play.

As well-known, GrC is called by a joint name for a variety of algorithms rather than one exact algorithm that is called granular computing. GrC is concerned with constructing and processing carried out at different level of *information granules*. The information represented by different level of granules show up distinct knowledge, features, and patterns, where the irrelevant features are hidden and valuable ones are highlighted. Taking satellite images as a example, the interests of researches within the low-resolution images may are the cloud patterns that present typhoons or other weather phenomena. However, in high-resolution satellite images, these large-scale atmospheric phenomena are ignored and small targets appear, such as a map of a city or a scene of a street. The same is generally true for all data. In different granularities of information, different features and patterns emerge. Hence, based on this fact, GrC is significantly useful to design more effective machine learning algorithms and data mining approaches.

There are a few of types of granularity that are often adopted in data mining and machine learning, including variable granulation, variable transformation, variable aggregation, system granulation (aggregation), concept granulation (component analysis), equivalence class granulation and component granulation.

As you known, the information hidden in Big Data maybe will lose partially if the data size is reduced to small ones. Not all the Big Data applications can use the GrC techniques as the process in Fig. 14. It depends the confidence and accuracy of results the system required. For example, financial data in banks and government are very sensitive and require high accuracy in some special analysis, and the sensor data generated by users in ITS need to be processed and responded one by one. In these cases, GrC dose not work well, and we need other solutions.

6.2. Cloud computing

The development of virtualization technologies have made supercomputing more accessible and affordable. Powerful computing infrastructures hidden in virtualization software make systems to be like a true physical computer, but with the flexible specification of details such as number of processors, memory and disk size, and operating system. The use of these virtual computers is known as cloud computing [50], which has been one of the most robust Big Data techniques [154,159]. The name of cloud computing comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. It entrusts remote services with a user's data, software and computation. The combination of virtual machines and large numbers of affordable processors has made it possible for internet-based companies to invest in large-scale computational clusters and advanced data-storage systems [159].

As illustrated in Fig. 15, cloud computing not only delivers applications and services over the Internet, it also has been extended to infrastructure as a service, for example, Amazon EC2, and platform as a service, such as Google AppEngine and Microsoft Azure. Infrastructure vendors provide hardware and a software stack including operating system, database, middleware and perhaps single instance of a conventional application. Therefore, it shows out illusion of infinite resources without up-front cost and fine-grained billing. It leads to the utility computing, i.e., pay-as-you-go computing.

Surprisingly, the cloud computing options available today are already well matched to the major themes of need, though some of us might not see it. Big Data forms a framework for discussing cloud computing options. Depending on special need, users can go into the marketplace and buy infrastructure services from providers like Google and Amazon, Software as a Service (SaaS) from a whole crew of companies starting at Salesforce and proceeding through NetSuite, Cloud9, Jobscience and Zuora-a list that is almost never ending. Another bonus brought by cloud environments is cloud storage which provides a possible tool for storing Big Data. Cloud storage have good extensibility and scalability in storing information as demonstrated in Fig. 16.

Cloud computing is a highly feasible technology and attract a large number of researchers to develop it and try to apply to Big Data problems. Usually, we need to combine the distributed MapReduce and cloud computing to get an effective answer

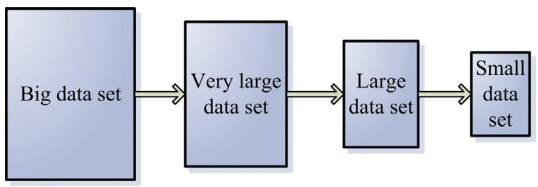

Fig. 14. GrC can be a option in some degree.

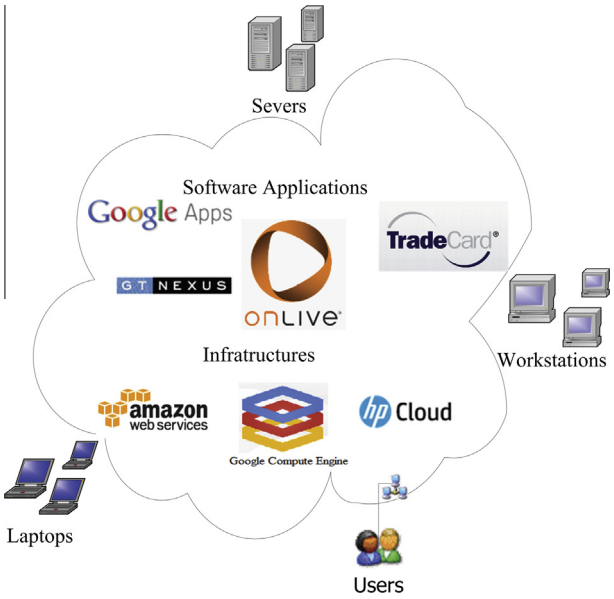

Fig. 15. Cloud computing logical diagram.

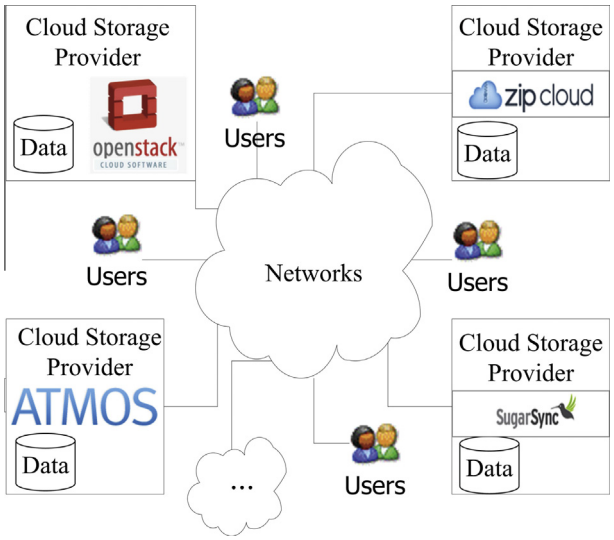

Fig. 16. Cloud storage.

for providing petabyte-scale computing [108]. CloudView [15] is a framework for storage, processing and analysis of massive machine maintenance data in a cloud computing environment, which is formulated using the Map/Reduce model and reaches real-time response. In [84], the authors extended Map/Reduce's filtering aggregation programming model in cloud environment and boosts the performance of complex analysis queries.

Apart from its flexibility, cloud computing addresses one of the challenges relating to transferring and sharing data, because data sets and analysis results held in the cloud can be shared with others [159]. There are a few disadvantages in cloud computing. The obvious one is the time and cost that are required to upload and download large quantities of data in the cloud environment. Otherwise, it becomes more difficult to control over the distribution of the computation and the underlying hardware. Furthermore, there are privacy concerns relating to the hosting of data sets on publicly accessible servers, as well as issues related to storage of data from human studies [159]. It is right to say that Big Data problems will push the cloud computing to a high level of development.

6.3. Bio-inspired computing

Human brain maybe can give a hand to help us to rethink the way we interact with Big Data. It captures and processes myriad of sensory data received moment of every day in an efficient and robust way. Human brain manages around a thousand TB of data and no neuron in the brain runs faster than 1 kHz, which is about the speed of a general PC in 1980s. However, human being does not feel heavy-headed while our brain boots up. That is because the biological computing system of our brain works in a distinct way compared with today's computer. Our brain does not need to locate and view large files with complex information sets. The information is partitioned and individually stored as simple data elements in the brain tissue. The processing for information in human brain is also executed in highly distributed and parallel ways. The multi-located storage schema and synchronous parallel processing approaches make our brain working so fast and efficiently.

Biological computing models (illustrated in Fig. 17) are better appropriate for Big Data because they have mechanisms with high-efficiency to organize, access and process data in ways that are more practical for the ranging and nearly infinite inputs we deal with every day. For today's technology, all of information is locked away in backward style data collections that are fixed and unwieldy. However, if we can store all that information in a system which is modeled more on biology rather than traditional ways, and then apply significant and increasing processing power and intelligent algorithms to analyze, rather than just move it around mechanically, then we have the possibility of generating and interacting with the world and the characters of supernatural.

Computational intelligence, which is inspired by nature, is a set of computational methodologies and approaches to address complex real-world problems. We have reason to believe that computational systems can also be illuminated by biological systems. *Biologically inspired Computing* [26] maybe provides tools to solve Big Data problems from hardware design to software design. In analogy to nature, bio-inspired hardware systems can be classified as three axes, phylogeny, ontogeny, and epigenesis [171]. In [192], authors give a review an emerging engineering discipline to program cell behaviors by embedding synthetic gene networks that perform computation, communications, and signal processing. Wang and Sun [188] proposed a bio-inspired cost minimization mechanism for data-intensive service provision. It utilizes bio-inspired mechanisms to search and find the optimal data service solution considering cost of data management and service maintenance. Tadashi [125] gave a review for biological communication (molecular communication) inspired by the cell and cell-to-cell communication. The data transformation and the communication between different computing units in Big Data systems maybe borrow some useful ideas from cells. In [140] two hardware processing architecture for modeling large networks of leaky-integrate-and-fire neurons, that integrate bio-inspired neural processing models into real-world control environments. Sergio [16] demonstrated self-synchronization mechanism, which borrowed from biological systems, as the basic tool for achieving globally optimal distributed decisions in a wireless sensor network.

Biocomputers is inspired and developed by biological molecules, such as DNA and proteins to conduct computational calculations involving storing, retrieving, and processing data. A significant feature of biocomputer is that it integrates biologically derived materials to perform computational functions and receive intelligent and efficient performance. As demonstrated in Fig. 18, a biocomputer is composed of a pathway or series of metabolic pathways involving biological materials that are engineered to behave in a certain manner based upon the conditions as input of the system. The resulting pathway of reactions that takes place constitutes an output, which is based on the engineering design of the biocomputer and can be interpreted as a form of computational analysis. There are three kinds of distinguishable biocomputers, including biochemical computers, biomechanical computers, and bioelectronic computers [151].

Once the Big Data technologies and techniques get mature enough, the following information revolution will incredibly change the way we process data. The computing systems become exponentially faster compared with current status, and novel data storage systems using biological models provide smarter interactions, inevitable data losses, and ambiguity. Genuine computational intelligence enables human-like analysis of massive quantities of data. It is true that the future con-

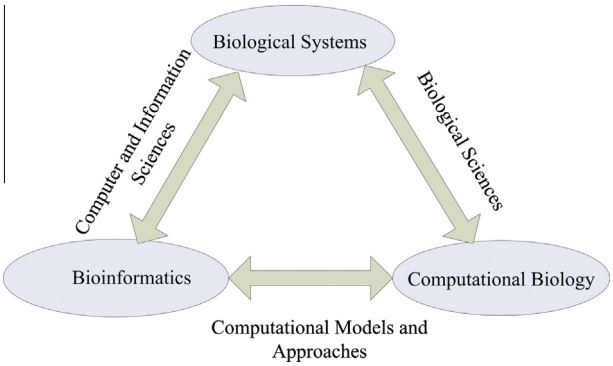

Fig. 17. Biology computing paradigm.

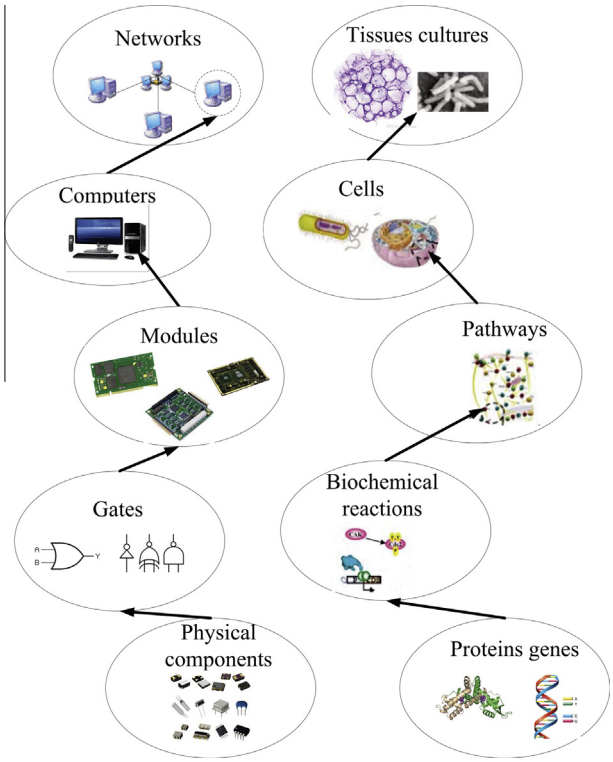

Fig. 18. Biocomputers. *Source:* from [12].

structed by bio-inspired technologies are so remarkable that a large amount of funds and human resources are poured into related research activities.

6.4. Quantum computing

A *quantum computer* has memory that is exponentially larger than its apparent physical size, and can manipulate an exponential set of inputs simultaneously. It also can compute in the twilight zone of Hilbert space. This exponential improvement in computer systems might be possible, and real powerful computer is emerging. If a real quantum computer existed now, we could solve problems that are exceptionally difficult on current computers, of course, including today's Big Data problems. Although it is very hard to develop quantum computer, the main technical difficulty in building a quantum computer could soon be the thing that makes it possible to build one. For example, D-Ware Systems Company developed their quantum computer, called "D-Wave one" with 128 qubits processor and "D-Wave two" with 512 qubits processor on 2011 and 2013 respectively.

In essence, quantum computing [107] is to harness and exploit the powerful laws of quantum mechanics to process information. In a traditional computer, information is presented by long strings of bits which encode either a zero or a one. Dif-

Table A.3
Sizes of data units.

Name	Equals to	Size in bytes
Bit	1 bit	1/8
Nibble	4 bits	1/2
Byte	8 bits	1
Kilobyte	1024 bytes	1024
Megabyte	1024 kilobytes	1,048,576
Gigabyte	1024 megabytes	1,073,741,824
Terrabyte	1024 gigabytes	1,099,511,627,776
Petabyte	1024 terrabytes	1,125,899,906,842,624
Exabyte	1024 petabytes	1,152,921,504,606,846,976
Zettabyte	1024 exabytes	1,180,591,620,717,411,303,424
Yottabyte	1024 zettabytes	1,208,925,819,614,629,174,706,176

ferently, A quantum computer uses quantum bits or qubits. The difference between qubit and bit is that, a qubit is a quantum system that encodes the zero and the one into two distinguishable quantum states. Because qubits behave quantumly, we can capitalize on the phenomena of "superposition" and "entanglement" [2]. For example, 100 qubits in quantum systems require 2^{100} complex values to be stored in classical computer systems. Nielsen and Chuang pointed out that "Trying to store all these complex numbers would not be possible on any conceivable classical computer" [129].

Many certain problems can be solved much faster by larger-scale quantum computers compared with classical computers. That is because that quantum algorithms, such as Simon's algorithm, Shor's algorithm, and other algorithms for simulating quantum systems, are more efficient and faster than traditional ones [170]. Quantum computation does not violate the Church-Turing thesis, as classical computers also can simulate an arbitrary quantum algorithm with unlimited resources.

Despite quantum computing is still in a fledging period, quantum computational operations have been executed under a small number of quantum bits in practical experiments, as the theoretical research continues to be advanced [129]. Indeed, there are a number of university institutes, national governments and military funding research groups are working on quantum computing studies to develop quantum computers for both civilian and national security purposes.

7. Conclusion

As we have entered an era of Big Data which is the next frontier for innovation, competition and productivity, a new wave of scientific revolution is about to begin. Fortunately, we will witness the coming technological leapfrogging. In this survey paper, we give a brief overview on Big Data problems, including Big Data opportunities and challenges, current techniques and technologies. We also propose several potential techniques to solve the problem, including cloud computing, quantum computing and biological computing. Although those technologies are still under development, we have confidence that in the coming future we will receive several great breakthroughs in those areas. Undoubtedly, today and future's Big Data problems will benefit from those progresses.

There is no doubt that Big Data analytics is still in the initial stage of development, since existing Big Data techniques and tools are very limited to solve the real Big Data problems completely, in which some of them even cannot be viewed as Big Data tools in the true sense. Therefore, more scientific investments from both governments and enterprises should be poured into this scientific paradigm to capture huge values from Big Data. From hardware [55] to software [163], we imminently require more advanced storage and I/O techniques, more favorable computer architectures, more efficient data-intensive techniques (cloud computing, social computing and biological computing, etc.) and more progressive technologies (Big Data platforms with sound architecture, infrastructure, approach, and properties). Big Data also means big systems, big challenges and big profits, so more research works in these sub-fields are necessary to resolve it. We are fortunately witnessing the birth and development of Big Data, and No person can settle it alone. Human resources, capital investments and creative ideas are fundamental components of development of Big Data.

Acknowledgments

This work was supported in part by the National 973 Basic Research Program of China under Grant 2011CB302801 and the Macau Science and Technology Development Fund under Grant 008/2010/A1 and University of Macau Multiyear Research Grants.

Appendix A. Sizes of data units

See Table A.3.

Appendix B. WordCount.java

```
1. package org.myorg;
2.
3. import java.io.IOException;
4. import java.util.*;
5.
6. import org.apache.hadoop.fs.Path;
7. import org.apache.hadoop.conf.*;
8. import org.apache.hadoop.io.*;
9. import org.apache.hadoop.mapred.*;
10. import org.apache.hadoop.util.*;
```

(*continued on next page*)

```
11.
12. public class WordCount {
13.
14.   public staticclass Map extends MapReduceBase implements
    Mapper < LongWritable, Text, Text, IntWritable> {
15.   private final static IntWritable one = new IntWriTable (1);
16.   private Text word = new Text ();
17.
18.   public void map (LongWritable key, Text value,
    OutputCollector <Text, IntWritabl>output,
    Reporter reporter) throws IOException {
19.   String line = value.toString ();
20.   StringTokenizer tokenizer = new StringTokenizer (line);
21.    while (tokenizer.hasMoreTokens ())
22.    word.set (tokenizer.nextToken ());
23.    output.collect (word, one);
24.     }
25.    }
26.   }
27.
28. public static class Reduce extends MapReduceBase implements
    Reducer<Text, IntWritable, Text, IntWritable> {
29.   public void reduce (Text key, Iterator<IntWritable>values,
    OutputCollector<Text, IntWritable>output, Reporter reporter) throws IOException {
30.     int sum = 0;
31.     while (values.hasNext ()) {
32.     sum += values.next ().get ();
33.     }
34.     output.collect (key, new IntWritable (sum));
35.     }
36.    }
37.
38. public static void main (String[ ] args) throws Exception {
39.   JobConf conf = new JobConf (WordCount.class);
40. conf.setJobName ("wordcount");
41.
42.   conf.setOutputKeyClass (Text.class);
43. conf.setOutputValueClass (IntWritable.class);
44.
45. conf.setMapperClass (Map.class);
46.   conf.setCombinerClass (Reduce.class);
47. conf.setReducerClass (Reduce.class);
48.
49.     conf.setInputFormat (TextInputFormat.class);
50.     conf.setOutputFormat (TextOutputFormat.class);
51.
52.     FileInputFormat.setInputPaths (conf, new Path (args[0]));
53.     FileOutputFormat.setOutputPath (conf, new Path(args [1]));
54.
55.     JobClient.runJob (conf);
57.       }
58.     }
59.
```

Source from [58].

Appendix C. Big Data Vendors

Big Data Vendors.

Vendor: 1010 data
Location: New York, NY
Website: http://www.1010data.com/index.php
Featured Big Data products: Hosted analytical platform for big data, using big table-type data structures for consolidation and analysis.

Vendor: 10gen
Location: New York, NY; Palo Alto, CA; London, Great Britain; Dublin, Ireland
Website: http://www.10gen.com/
Featured Big Data products: Commercial support and services for MongoDB.

Vendor: Acxiom
Location: Various global locations
Website: http://acxiom.com/
Featured Big Data products: Data analytics and processing, with an emphasis on marketing data and services.

Vendor: Amazon Web Services
Location: Global
Website: http://aws.amazon.com/
Featured Big Data products: Provider of cloud-based database, storage, processing, and virtual networking services.

Vendor: Aster Data
Location: San Carlos, CA
Website:http://www.asterdata.com/
Featured Big Data products: Data analytic services using Map/Reduce technology.

Vendor: Calpont
Location: Frisco, TX
Website:http://www.calpont.com/
Featured Big Data products: InfiniDB Enterprise, is a column-sorted database that also provides massively parallel processing capabilities.

Vendor: Cloudera
Location: Palo Alto and San Francisco, CA
Website:http://www.cloudera.com/
Featured Big Data products: Distributor of commercial implementation of Apache Hadoop, with services and support.

Vendor: Couchbase
Location: Mountain View, CA
Website: http://www.couchbase.com/
Featured Big Data products: Commercial sponsor of the Couchbase Server Map/Reduce-oriented database, as well as Apache CouchDB and memcached.

Vendor: Datameer
Location: San Mateo, CA
Website: http://www.datameer.com/
Featured Big Data products: Data visualization services for Apache Hadoop data stores.

Vendor: DataSift
Location: San Francisco, CA; Reading, United Kingdom
Website:http://datasift.com/
Featured Big Data products: Social media data analytical services. Licensed re-syndicator of Twitter.

Vendor: DataStax
Location: San Mateo, CA; Austin, TX
Website:http://www.datastax.com/
Featured Big Data products: Distributor of commercial implementation of Apache Cassandra, with services and support.

(*continued on next page*)

Vendor: Digital Reasoning
Location: Franklin, TN
Website:http://www.digitalreasoning.com/
Featured Big Data products: Synthesys, a hosted and local business intelligence data analysics tool.

Vendor: EMC
Location: Various global locations
Website:http://www.emc.com/
Featured Big Data products: Makers of Greenplum, a massively parallel processing data store/analytics solution.

Vendor: esri
Location: Various global locations.
Website:http://www.esri.com/
Featured Big Data products: GIS data analytical services.

Vendor: FeedZai
Location: United Kingdom
Website:http://www.feedzai.com/
Featured Big Data products: FeedZai Pulse, a real-time business intelligence appliance.

Vendor: Hadapt
Location: Cambridge, MA
Website: http://www.hadapt.com/
Featured Big Data products: Data analytic services for Apache Hadoop data stores.

Vendor: Hortonworks
Location: Sunnyvale, CA
Website: http://hortonworks.com/
Featured Big Data products: Distributor of commercial implementation of Apache Hadoop, with services and support.

Vendor: HPCC Systems
Location: Alpharetta, GA
Website: http://hpccsystems.com/
Featured Big Data products: HPCC (High Performance Computing Cluster), an open source massive parallel processing computing database.

Vendor: IBM
Location: Various global locations
Website: http://www.ibm.com/
Featured Big Data products: Hardware; data analytical services; and db2, a massive parallel processing database.

Vendor: Impetus
Location: San Jose, CA; Noida, India; Indore, India; Bangalore, India
Website: http://impetus.com/
Featured Big Data products: Data analytic and management services for Apache Hadoop data stores.

Vendor: InfoBright
Location: Toronto, ON; Dublin, Ireland; Chicago, IL
Website:http://www.infobright.com/
Featured Big Data products: InfoBright, a column store database, with services and support.

Vendor: Jaspersoft
Location: Various global locations
Website:http://www.jaspersoft.com/
Featured Big Data products: Data analytic services for Apache Hadoop data stores.

Vendor: Karmasphere
Location: Cupertino, CA
Website:http://www.karmasphere.com/
Featured Big Data products: Data analytic and development services for Apache Hadoop data stores.

Vendor: Lucid Imagination
Location: Redwood City, CA
Website:http://www.lucidimagination.com/

Featured Big Data products: Distributor of commercial implementation of Apache Lucene and Apache Solr, with services and support. Provider of LucidWorks enterprise search software.

Vendor: MapR Technologies
Location: San Jose CA; Hyderabad, India
Website: http://www.mapr.com/
Featured Big Data products: Distributor of commercial implementation of Apache Hadoop, with services and support.

Vendor: MarkLogic
Location: Various global locations
Website: http://www.marklogic.com/
Featured Big Data products: Data analyic and visualization services.

Vendor: Netezza Corp.
Location: Various global locations
Website: http://www.netezza.com/
Featured Big Data products: Massively parallel processing data appliances, analytic services.

Vendor: Oracle
Location: Various global locations
Website: http://www.oracle.com/
Featured Big Data products: Various hardware and software offerings, including Big Data Appliance, MySQL Cluster, Exadata Database Machine.

Vendor: ParAccel
Location: Campbell, CA; San Diego, CA; Wokingham, United Kingdom
Website: http://www.paraccel.com/
Featured Big Data products: Data analyics using column-store technology.

Vendor: Pentaho
Location: Various global locations
Website: http://www.pentaho.com/
Featured Big Data products: Data analytic services for Apache Hadoop data stores.

Vendor: Pervasive Software
Location: Austin, TX
Website: http://www.pervasive.com/
Featured Big Data products: Data analytic services for Apache Hadoop data stores based on Hive.

Vendor: Platform Computing
Location: Various global locations
Website:http://www.platform.com/
Featured Big Data products: Distributor of commercial implementation of Apache Hadoop, with services and support.

Vendor: RackSpace
Location: Global
Website:http://www.rackspace.com/
Featured Big Data products: Provider of cloud-based database, storage, and processing services.

Vendor: Revolution Analytics
Location: Palo Alto, CA; Seattle, WA
Website:http://www.revolutionanalytics.com/
Featured Big Data products: Data analytic and visualization services using R-based software.

Vendor: Splunk
Location: Various global locations
Website: http://www.splunk.com/
Featured Big Data products: Data analytic and visualization services using logging-oriented software.

Vendor: Tableau Software
Location: Seattle, WA; Kirkland, WA; San Mateo, CA; Surrey, United Kingdom; Paris, France
Website: http://www.tableausoftware.com/
Featured Big Data products: Business intelligence and data analytic software.

(*continued on next page*)

Vendor: Talend
Location: Various global locations
Website: http://www.talend.com/index.php
Featured Big Data products: Database management software.

Vendor: Teradata
Location: Miamisburg, OH
Website: http://www.teradata.com/
Featured Big Data products: Database management software.

Vendor: Vertica Systems
Location: Billerica, MA
Website: http://www.vertica.com/
Featured Big Data products: Data analytics using column-store based technologies.

Vendor: Apache Flume
Location: Illinois, US
Website: http://flume.apache.org/
Featured Big Data products: Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.

Vendor: Facebook Scribe
Location: California, US
Website: https://github.com/facebook/scribe/
Featured Big Data products: Aggregating log data streamed in real time.

Vendor: Google's Dremel
Location: Google, Inc.
Website: http://research.google.com/pubs/pub36632.html
Featured Big Data products: A scalable, interactive ad hoc query system for analysis of read-only nested data.

Vendor: Apache Drill
Location: Illinois, US
Website: http://incubator.apache.org/drill/
Featured Big Data products: Distributed system for interactive analysis of large-scale datasets, based on Google's Dremel.

Partial Source from [146]

References

[1] http://www.whitehouse.gov/sites/default/files/microsites/ostp/big-data-fact-sheet-final-1.pdf.
[2] http://quantumcomputers.com.
[3] Karmasphere Studio and Analyst, 2012. <http://www.karmasphere.com/>.
[4] Pentaho Business Analytics, 2012. <http://www.pentaho.com/explore/pentaho-business-analytics/>.
[5] Sqlstream, 2012. <http://www.sqlstream.com/products/server/>.
[6] Storm, 2012. <http://storm-project.net/>.
[7] Abzetdin Adamov. Distributed file system as a basis of data-intensive computing, in: 2012 6th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–3 (October).
[8] Divyakant Agrawal, Philip Bernstein, Elisa Bertino, Susan Davidson, Umeshwas Dayal, Michael Franklin, Johannes Gehrke, Laura Haas, Jiawei Han Alon Halevy, H.V. Jagadish, Alexandros Labrinidis, Sam Madden, Yannis Papakon stantinou, Jignesh Patel, Raghu Ramakrishnan, Kenneth Ross, Shahabi Cyrus, Dan Suciu, Shiv Vaithyanathan, Jennifer Widom, Challenges and Opportunities with Big Data, CYBER CENTER TECHNICAL REPORTS, Purdue University, 2011.
[9] Byungik Ahn, Neuron machine: Parallel and pipelined digital neurocomputing architecture, in: 2012 IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom), 2012, pp. 143–147.
[10] James Ahrens, Kristi Brislawn, Ken Martin, Berk Geveci, C. Charles Law, Michael Papka, Large-scale data visualization using parallel data streaming, IEEE Comput. Graph. Appl. 21 (4) (2001) 34–41.
[11] Chris Anderson, The End of Theory: The Data Deluge Makes the Scientific Method Obsolete, 2008. <http://www.wired.com/science/discoveries/magazine/16-07/pb-theory>.
[12] Ernesto Andrianantoandro, Subhayu Basu, David K Karig, Ron Weiss, Synthetic biology: new engineering rules for an emerging discipline, Mol. Syst. Biol. 2 (2006).
[13] Itamar Arel, Derek C. Rose, Thomas P. Karnowski, Deep machine learning – a new frontier in artificial intelligence research, IEEE Comput. Intell. Mag. 5 (4) (2010) 13–18.
[14] Aditya Auradkar, Chavdar Botev, Shirshanka Das, Dave DeMaagd, Alex Feinberg, Phanindra Ganti, Bhaskar Ghosh Lei Gao, Kishore Gopalakrishna, Brendan Harris, Joel Koshy, Kevin Krawez, Jay Kreps, Shi Lu, Sunil Nagaraj, Neha Narkhede, Sasha Pachev, Igor Perisic, Lin Qiao, Tom Quiggle, Jun Rao, Bob Schulman, Abraham Sebastian, Oliver Seeliger, Adam Silberstein, Boris Shkolnik, Chinmay Soman, Roshan Sumbaly, Kapil Surlaker, Sajid

Topiwala, Cuong Tran, Balaji Varadarajan, Jemiah Westerman, Zach White, David Zhang, Jason Zhang, Data infrastructure at linkedin, in: 2012 IEEE 28th International Conference on Data Engineering (ICDE), 2012, pp. 1370–1381.
[15] Arshdeep Bahga, Vijay K. Madisetti, Analyzing massive machine maintenance data in a computing cloud, IEEE Trans Parallel Distrib. Syst. 23 (10) (2012) 1831–1843.
[16] Sergio Barbarossa, Gesualdo Scutari, Bio-inspired sensor network design, IEEE Signal Process. Mag. 24 (3) (2009) 95–98.
[17] Ron Bekkerman, Mikhail Bilenko, John Langford, Scaling Up Machine Learning: Parallel and Distributed Approaches, Cambridge University Press, 2012.
[18] Gordon Bell, Tony Hey, Alex Szalay, Beyond the data deluge, Science 323 (5919) (2009) 1297–1298.
[19] M. Bencivenni, F. Bonifazi, A. Carbone, A. Chierici, A. D'Apice, D. De Girolamo, L. dell'Agnello, M. Donatelli, G. Donvito, A. Fella, F. Furano, D. Galli, A. Ghiselli, A. Italiano, G. Lo Re, U. Marconi, B. Martelli, M. Mazzucato, M. Onofri, P.P. Ricci, F. Rosso, D. Salomoni, V. Sapunenko, V. Vagnoni, R. Veraldi, M.C. Vistoli, D. Vitlacil, S. Zani, A comparison of data-access platforms for the computing of large hadron collider experiments, IEEE Trans. Nucl. Sci. 55 (3) (2008) 1621–1630.
[20] Yoshua Bengio, Learning deep architectures for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[21] Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[22] Janine Bennett, Ray Grout, Philippe Pebay, Diana Roe, David Thompson, Numerically stable, single-pass, parallel statistics algorithms, in: IEEE International Conference on Cluster Computing and Workshops, 2009, CLUSTER '09, 2009, pp. 1–8.
[23] Paul Bertone, Mark Gerstein, Integrative data mining: the new direction in bioinformatics, IEEE Eng. Med. Biol. Mag. 20 (4) (2001) 33–40.
[24] James C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, New York, NY, USA: Plenum, 1981.
[25] Ella Bingham, Heikki Mannila, Random projection in dimensionality reduction: applications to image and text data, in: Knowledge Discovery and Data Mining, ACM Press, 2001, pp. 245–250.
[26] Josh Bongard, Biologically inspired computing, Computer 42 (4) (2009) 95–98.
[27] Björn Bringmann, Michele Berlingerio, Francesco Bonchi, Aristides Gionis, Learning and predicting the evolution of social networks, IEEE Intell. Syst. 25 (4) (2010) 26–35.
[28] Jason Brooks, Review: Talend Open Studio Makes Quick etl Work of Large Data Sets, 2009. <http://www.eweek.com/c/a/Database/REVIEW-Talend-Open-Studio-Makes-Quick-ETL-Work-of-Large-Data-Sets-281473/>.
[29] Geoff Brumfiel, High-energy physics: down the petabyte highway, Nature (469) (2011) 282–283.
[30] Randal E. Bryant, Data Intensive supercomputing: The Case for Disc. Technical Report CMU-CS-07-128, 2007.
[31] Randal E. Bryant, Data-intensive scalable computing for scientific applications, Comput. Sci. Eng. 13 (6) (2011) 25–33.
[32] Pavel Bzoch, Jiri Safarik, State of the art in distributed file systems: Increasing performance, in: Engineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference on the, 2011, pp. 153–154.
[33] Deng Cai, Xiaofei He, Jiawei Han, Srda: an efficient algorithm for large-scale discriminant analysis, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 1–12.
[34] Mario Cannataro, Antonio Congiusta, Andrea Pugliese, Domenico Talia, Paolo Trunfio, Distributed data mining on grids: services, tools, and applications, IEEE Trans. Syst. Man Cyber. Part B: Cyber. 34 (6) (2004) 2451–2465.
[35] Yi Cao, Dengfeng Sun, A parallel computing framework for large-scale air traffic flow optimization, IEEE Trans. Intell. Trans. Syst. 13 (4) (2012) 1855–1864.
[36] Edward Capriolo, Cassandra High Performance Cookbook, Packt Publishing, 2011.
[37] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, Bigtable: a distributed storage system for structured data, ACM Trans. Comput. Syst. 26 (2) (2008).
[38] Jagmohan Chauhan, Shaiful Alam Chowdhury, Dwight Makaroff, Performance evaluation of yahoo! s4: a first look, in: 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2012, pp. 58–65.
[39] Long Chen, C.L.P. Chen, Mingzhu Lu, A multiple-kernel fuzzy c-means algorithm for image segmentation, IEEE Trans. Syst. Man Cyber. Part B: Cyber. 41 (5) (2011) 1263–1274.
[40] Wei-Hua Lin Shuang Shuang Li Cheng Chen, Zhong Liu, Kai Wang, Distributed modeling in a mapreduce framework for data-driven traffic flow forecasting, IEEE Trans. Intell. Trans. Syst. 14 (1) (2013) 22–33.
[41] Agostino Di Ciaccio, Mauro Coli, Angulo Ibanez, Jose Miguel, Advanced Statistical Methods for the Analysis of Large Data-Sets, Springer, 2012.
[42] Dan Cireşan, Ueli Meier, Jürgen Schmidhuber, Multi-column deep neural networks for image classification, IEEE Conf. Comput. Vision Pattern Recognit. (2012).
[43] Jeffrey Deam, Sanjay Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[44] XYamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, Ronald G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[45] Petra Fey, Takashi Gojobori, Linda Hannick, Winston Hide, David P. Hill, Renate Kania, Mary Schaeffer, Susan St Pierre, Simon Twigger, Owen White, Seung Yon Y. Rhee, Doug Howe, Maria Costanzo, Big data: the future of biocuration, Nature 455 (7209) (2008) 47–50.
[46] Rui Máximo Esteves, Chunming Rong, Using mahout for clustering wikipedia's latest articles: a comparison between k-means and fuzzy c-means in the cloud, in: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), 2011, pp. 565–569.
[47] Rui Máximo Esteves, Chunming Rong, Rui Pais, K-means clustering in the cloud – a mahout test, in: 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA), 2011, pp. 514–519.
[48] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, Cloud computing and grid computing 360-degree compared, in: Grid Computing Environments Workshop, 2008, GCE'08, 2008, pp. 1–10.
[49] Yoshiji Fujimoto, Naoyuki Fukuda, Toshio Akabane, Massively parallel architectures for large scale neural network simulations, IEEE Trans. Neural Networks 3 (6) (1992) 876–888.
[50] Borko Furht, Armando Escalante, Handbook of Cloud Computing, Springer, 2011.
[51] Lee Garber, Using in-memory analytics to quickly crunch big data, IEEE Comput. Soc. 45 (10) (2012) 16–18.
[52] A.O. García, S. Bourov, A. Hammad, V. Hartmann, T. Jejkal, J.C. Otte, S. Pfeiffer, T. Schenker, C. Schmidt, P. Neuberger, R. Stotzka, J. van Wezel, B. Neumair, A. Streit, Data-intensive analysis for scientific experiments at the large scale data facility, in: 2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2011, pp. 125–126.
[53] Bo Geng, Yangxi Li, Dacheng Tao, Meng Wang, Zheng-Jun Zha, Chao Xu, Parallel lasso for large-scale video concept detection, IEEE Trans. Multimedia 14 (1) (2012) 55–65.
[54] Dan Gillick, Arlo Faria, John DeNero, Mapreduce: Distributed Computing for Machine Learning, 2006.
[55] Maya Gokhale, Jonathan Cohen, Andy Yoo, W. Marcus Miller, Hardware technologies for high-performance data-intensive computing, Computer 41 (4) (2008) 60–68.
[56] Naiyang Guan, Dacheng Tao, Zhigang Luo, Bo Yuan, Online nonnegative matrix factorization with robust stochastic approximation, IEEE Trans. Neural Networks Learning Syst. 23 (7) (2012) 1087–1099.
[57] Vincenzo Gulisano, Ricardo Jimenez-Peris, Marta Patino-Martinez, Claudio Soriente, Patrick Valduriez, Streamcloud: an elastic and scalable data streaming system, IEEE Trans. Parallel Distrib. Syst. 23 (12) (2012) 2351–2365.
[58] Apache Hadoop, Words Count Example, 2012. <http://developer.yahoo.com/hadoop/tutorial/module4.html#wordcount>.
[59] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Diane Cerra, second ed., 2000.
[60] Jing Han, Haihong E, Guan Le, Jian Du, Survey on nosql database, in: 2011 6th International Conference on Pervasive Computing and Applications (ICPCA), 2011, pp. 363–366.

[61] Xixian Han, Jianzhong Li, Donghua Yang, Jinbao Wang, Efficient skyline computation on big data, IEEE Trans. Knowl. Data Eng. PP (99) (2012) 1–13.
[62] Kamal Hassan, Faten Mahmoud, An incremental approach for the solution of quadratic problems, Math. Modell. 8 (1987) 34–36.
[63] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical, Learning: Data Mining Inference and Prediction, second ed., Springer, 2009.
[64] Jeffrey Heer, Jock D. Mackinlay, Chris Stolte, Maneesh Agrawala, Graphical histories for visualization: supporting analysis, communication, and evaluation, IEEE Trans. Visual. Comput. Graph. 14 (6) (2008) 1189–1196.
[65] Tony Hey, Stewart Tansley, Kristin Tolle, The fourth paradigm: data-intensive scientific discovery, Microsoft Research (2009).
[66] Tony Hey, Anne E. Trefethen, The uk e-science core programme and the grid, Future Gener. Comput. Syst. 18 (8) (2002) 1017–1031.
[67] Martin Hilbert, Priscila López, The world's technological capacity to store, communicate, and compute information, Science 332 (6025) (2011) 60–65.
[68] Geoffrey E. Hinton, Learning multiple layers of representation, Trends Cogn. Sci. 11 (2007) 428–434.
[69] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[70] Geoffrey E. Hinton, Ruslan Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[71] Kaoru Hirota, Wiltold Pedrycz, Fuzzy computing for data mining, Proc. IEEE 87 (9) (1999) 575–1600.
[72] Wen-Feng Hsiao, Te-Min Chang, An incremental cluster-based approach to spam filtering, Expert Syst. Appl. 34 (3) (2008) 1599–1608.
[73] Lee Hutchinson, Solid-state revolution: in-depth on how ssds really work, Ars Technica (2012).
[74] Grant Ingersoll, Introducing apache mahout: scalable, commercial-friendly machine learning for building intelligent applications, IBM Corporation (2009).
[75] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, Dennis Fetterly, Dryad: distributed data-parallel programs from sequential building blocks, in: EuroSys '07 Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, vol. 41(3), 2007, pp. 59–72.
[76] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, Dennis Fetterly, Dryad: Distributed data-parallel programs from sequential building blocks, in: Proceedings of the 2007 Eurosys Conference, 2007.
[77] Renato Porfirio Ishii, Rodrigo Fernandes de Mello, An adaptive and historical approach to optimize data access in grid computing environments, INFOCOMP J. Comput. Sci. 10 (2) (2011) 26–43.
[78] Renato Porfirio Ishii, Rodrigo Fernandes de Mello, An online data access prediction and optimization approach for distributed systems, IEEE Trans. Parallel Distrib. Syst. 23 (6) (2012) 1017–1029.
[79] R.P. Ishii, R.F. de Mello, A history-based heuristic to optimize data access in distributed environments, in: Proc. 21st IASTED International Conf. Parallel and Distributed Computing and Systems, 2009.
[80] Bart Jacob, Michael Brown, Kentaro Fukui, Nihar. Trivedi, Introduction to Grid Computing, IBM Redbooks Publication, 2005.
[81] Adam Jacobs, The pathologies of big data, Commun. ACM 52 (8) (2009) 36–44.
[82] Mohsen Jamali, Hassan Abolhassani, Different aspects of social network analysis, in: IEEE/WIC/ACM International Conference on Web Intelligence, 2006, WI 2006, 2006, pp. 66–72.
[83] Gwanggil Jeon, Donghyung Kim, Jechang Jeong, Rough sets attributes reduction based expert system in interlaced video sequences, IEEE Trans. Consum. Electr. 52 (4) (2006) 1348–1355.
[84] Dawei Jiang, Anthony K.H. Tung, Gang Chen, Map-join-reduce: toward scalable and efficient data analysis on large clusters, IEEE Trans. Knowl. Data Eng. 23 (9) (2011) 1299–1311.
[85] Wei Jiang, Eric Zavesky, Shih-Fu Chang, Alex Loui, Cross-domain learning methods for high-level visual concept classification, in: 15th IEEE International Conference on Image Processing, 2008, ICIP 2008, 2008, pp. 161–164.
[86] M. Tim Jones, Process Real-Time Big Data with Twitter Storm, 2012. <http://www.ibm.com/developerworks/opensource/library/os-twitterstorm/index.html?ca=drs->.
[87] Vamsee Kasavajhala, Solid state drive vs. hard disk drive price and performance study, Dell PowerVault Tech. Mark. (2012).
[88] Daniel A. Keim, Christian Panse, Mike Sips, Visual data mining in large geospatial point sets, IEEE Comput. Graph. Appl. 24 (5) (2004) 36–44.
[89] Jeff Kelly, Apache drill brings sql-like, ad hoc query capabilities to big data, February 2013. <http://wikibon.org/wiki/v/Apache-Drill-Brings-SQL-Like-Ad-Hoc-Query-Capabilities-to-Big-Data>.
[90] Wooyoung Kim, Parallel clustering algorithms: survey, in: Parallel Algorithms, Spring, 2009.
[91] Ben Klemens, Modeling with Data: Tools and Techniques for Statistical Computing, Princeton University Press, 2008.
[92] Richard T. Kouzes, Gordon A. Anderson, Stephen T. Elbert, Ian Gorton, Deborah K. Gracio, The changing paradigm of data-intensive computing, Computer 42 (1) (2009) 26–34.
[93] Stephan Kraft, Giuliano Casale, Alin Jula, Peter Kilpatrick, Des Greer, Wiq: work-intensive query scheduling for in-memory database systems, in: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), 2012, pp. 33–40.
[94] K.P. Lakshmi, C.R.K. Reddy, A survey on different trends in data streams, in: 2010 International Conference on Networking and Information Technology (ICNIT), 2010, pp. 451–455.
[95] Nicholas D. Lane, Ye Xu, Hong Lu, Andrew T. Campbell, Tanzeem Choudhury, Shane B. Eisenman, Exploiting social networks for large-scale human behavior modeling, IEEE Pervasive Comput. 10 (4) (2011) 45–53.
[96] Doug Laney, 3d Data managment: controlling data volume, velocity and variety, Appl. Delivery Strategies Meta Group (949) (2001).
[97] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, Andrew Y. Ng, Building high-level features using large scale unsupervised learning, in: Proceedings of the 29th International Conference on Machine Learning, 2012.
[98] John A. Lee, Michel Verleysen, Nonlinear Dimensionality Reduction, first ed., Springer, 2007.
[99] David Leong, A new revolution in enterprise storage architecture, IEEE Potentials 28 (6) (2009) 32–33.
[100] Arthur Lesk, Introduction to Bioinformatics, third ed., Oxford University Press, 2008.
[101] Hui Li, Geoffrey Fox, Judy Qiu, Performance model for parallel matrix multiplication with dryad: dataflow graph runtime, in: 2012 Second International Conference on Cloud and Green Computing, 2012, pp. 675–683.
[102] Xiaodong Li, Xin Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (2008) 210–224.
[103] Zhong Liang, ChiTian He, Zhang Xin, Feature based visualization algorithm for large-scale flow data, in: Second International Conference on Computer Modeling and Simulation, 2010, ICCMS '10, vol. 1, 2010, pp. 194–197.
[104] Ching-Yung Lin, Lynn Wu, Zhen Wen, Hanghang Tong, Vicky Griffiths-Fisher, Lei Shi, David Lubensky, Social network analysis in enterprise, Proc. IEEE 100 (9) (2012) 2759–2776.
[105] Yan-Jun Liu, C.L.P. Chen, Guo-Xing Wen, Shaocheng Tong, Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems, IEEE Trans. Neural Networks 22 (7) (2011) 1162–1167.
[106] Yiming Liu, Dong Xu, Ivor Wai-Hung Tsang, Jiebo Luo, Textual query of personal photos facilitated by large-scale web data, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1022–1036.
[107] Stuart P. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.
[108] Steve Loughran, Jose Alcaraz Calero, Andrew Farrell, Johannes Kirschnick, Julio Guijarro, Dynamic cloud deployment of a mapreduce architecture, IEEE Internet Comput. 16 (6) (2012) 40–50.
[109] Haiping Lu, Konstantinos N. Plataniotis, Anastasios N. Venetsanopoulos, A survey of multilinear subspace learning for tensor data, Pattern Recogn. 44 (7) (2011) 1540–1551.
[110] Clifford Lynch, Big data: how do your data grow?, Nature 455 (7209) (2008) 28–29
[111] Hao Ma, Irwin King, Michael Rung-Tsong Lyu, Mining web graphs for recommendations, IEEE Trans. Knowl. Data Eng. 24 (12) (2012) 1051–1064.
[112] Kwan-Liu Ma, Steven Parker, Massively parallel software rendering for visualizing large-scale data sets, IEEE Comput. Graph. Appl. 24 (5) (2004) 36–44.

[113] Yakout Mansour, A.Y. Chang, Jeyant Tamby, Ebrahim Vaahedi, B.C. Hydro, B.R. Corns, M.A. El-Sharkawi, Large scale dynamic security screening and ranking using neural networks, IEEE Trans. Power Syst. 12 (2) (1997) 954–960, 199.
[114] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, Big data: The Next Frontier for Innovation, Competition, and Productivity, McKinsey Global Institute, 2012.
[115] Yun Mao, Feng Wang, Lili Qiu, Simon Lam, Jonathan Smith, S4: Small state and small stretch compact routing protocol for large static wireless networks, IEEE/ACM Transactions on Networking 18 (3) (2010) 761–774.
[116] Nathan Marz, James Warren, Big data: principles and best practices of scalable realtime data systems, Manning (2012).
[117] Jason McDermott, Ram Samudrala, Roger Bumgarner, Kristina. Montgomery, Computational Systems Biology, Humana Press, 2009.
[118] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, Theo Vassilakis, Dremel: interactive analysis of webscale datasets, in: Proc. of the 36th Int'l Conf. on Very Large Data Bases (2010), vol. 3(1), 2010, pp. 330–339.
[119] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, Jan Černocký, Strategies for training large scale neural network language models, in: IEEE Workshop on Automatic Speech Recognition and Understanding, 2011.
[120] Ross Mistry, Stacia Misner, Introducing microsoft SQL server 2012, Microsoft (2012).
[121] Pabitra Mitra, C.A. Murthy, Sankar K. Pal, A probabilistic active support vector learning algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 26 (3) (2004) 603–618.
[122] Ilya Molchanov, Theory of Random Sets, Springer, 2005.
[123] Christian Molinari, No One Size Fits all Strategy for Big Data, Says ibm, October 2012. <http://www.bnamericas.com/news/technology/no-one-size-fits-all-strategy-for-big-data-says-ibm>.
[124] Hannes Mühleisen, Kathrin Dentler, Large-scale storage and reasoning for semantic data using swarms, IEEE Comput. Intell. Mag. 7 (2) (2012) 32–44.
[125] Tadashi Nakano, Biological computing based on living cells and cell communication, in: 2010 13th International Conference on Network-Based Information Systems (NBiS), 2010, pp. 42–47.
[126] Arnab Nandi, Cong Yu, Philip Bohannon, Raghu Ramakrishnan, Data cube materialization and mining over mapreduce, IEEE Trans. Knowl. Data Eng. 24 (10) (2012) 1747–1759.
[127] Jean-Daniel Fekete Nathalie Henry, Michael J. McGuffin, Nodetrix: a hybrid visualization of social network, IEEE Trans. Visual. Comput. Graph. 13 (6) (2007) 1302–1309.
[128] Leonardo Neumeyer, Bruce Robbins, Anish Nair, Anand Kesari, S4: distributed stream computing platform, in: 2010 IEEE Data Mining Workshops (ICDMW), Sydney, Australia, 2010, pp. 170–177.
[129] Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2009.
[130] Christopher Oehmen, Jarek Nieplocha, Scalablast: a scalable implementation of blast for high-performance data-intensive bioinformatics analysis, IEEE Trans. Parallel Distrib. Syst. 17 (8) (2006) 740–749.
[131] Cheolhwan Oh, Staniswław H. Żak, Large-scale pattern storage and retrieval using generalized brain-state-in-a-box neural networks, IEEE Trans. Neural Networks 21 (4) (2010) 633–643.
[132] Sysoev Oleg, Oleg Burdakovb, A. Grimvalla, A segmentation-based algorithm for large-scale partially ordered monotonic regression, Comput. Stat. Data Anal. 55 (8) (2011) 2463–2476.
[133] Simone Ferlin Oliveira, Karl Fürlinger, Dieter Kranzlmüller, Trends in computation, communication and storage and the consequences for data-intensive science, in: IEEE 14th International Conference on High Performance Computing and Communications, 2012.
[134] Alina Oprea, Michael K. Reiter, Ke Yang, Space efficient block storage integrity, in: Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS 05), 2005.
[135] Oracle, Oracle information architecture: an architect's guide to big data, An Oracle White Paper in Enterprise Architecture, 2012.
[136] Tamer M. Özsu, Patrick Valduriez, Principles of Distributed Database Systems, third ed., Springer, 2011.
[137] James P. Ahrens, Bruce Hendrickson, Gabrielle Long, Steve Miller, Robert Ross, Dean Williams, Data-intensive science in the us doe: case studies and future challenges, Comput. Sci. Eng. 13 (6) (2011) 14–24.
[138] Indranil Palit, Chandan K. Reddy, Scalable and parallel boosting with mapreduce, IEEE Trans. Knowl. Data Eng. 20 (10) (2012) 1904–1916.
[139] Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker, A comparison of approaches to large-scale data analysis, in: SIGMOD '09 Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, 2009, pp. 165–178.
[140] Martin J. Pearson, Anthony G. Pipe, Benjamin Mitchinson, Kevin Gurney, Chris Melhuish, Ian Gilhespy, Mokhtar Nibouche, Implementing spiking neural networks for real-time signal-processing and control applications: a model-validated fpga approach, IEEE Trans. Neural Networks 18 (5) (2007) 1472–1487.
[141] Philippe Pébay, David Thompson, Janine Bennett, Ajith Mascarenhas, Design and performance of a scalable, parallel statistics toolkit, in: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011, pp. 1475–1484.
[142] Witold Pedrycz, Andrzej, Vladik Kreinovich, Handbook of Granular Computing, WILEY, 2008.
[143] Georg Peters, Granular box regression, IEEE Trans. Fuzzy Syst. 19 (6) (2011) 1141–1151.
[144] A. Pirovano, A.L. Lacaita, A. Benvenuti, F. Pellizzer, S. Hudgens, R. Bez, Scaling analysis of phase-change memory technology, IEEE Int. Electron Dev. Meeting (2003) 29.6.1–29.6.4.
[145] Eelco Plugge, Tim Hawkins, Peter Membrey, The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing, first ed., Apress, 2010.
[146] Brian Proffitt, Big Data Tools and Vendors, 2012. <http://www.itworld.com/big-datahadoop/251912/big-data-tools-and-vendors?page=0,0>.
[147] Miloš Radovanović, Alexandros Nanopoulos, Mirjana Ivanović, Hubs in space: popular nearest neighbors in high-dimensional data, J. Mach. Learn. Res. 11 (2010) 2487–2531.
[148] William Yurcik Larry Brumbaugh Ragib Hasan, Zahid Anwar, Roy H. Campbell, A survey of peer-to-peer storage techniques for distributed file systems, in: International Conference on Information Technology: Coding and Computing, 2005, ITCC 2005, vol. 2, 2005, pp. 205–213.
[149] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, Christos Kozyrakis, Evaluating mapreduce for multi-core and multiprocessor systems, in: IEEE 13th International Symposium on High Performance Computer Architecture, 2007, HPCA 2007, 2006, pp. 13–24.
[150] Sanjay Ranka, Sartaj Sahni, Clustering on a hypercube multicomputer, IEEE Trans. Parallel Distrib. Syst. 2 (2) (1991) 532–536.
[151] Mark Ratner, Daniel Ratner, Nanotechnology: A Gentle Introduction to the Next Big Idea, first ed., Prentice Hall Press, Upper Saddle River, NJ, USA, 2002.
[152] Vikas C. Raykar, Ramani Duraiswami, Balaji Krishnapuram, A fast algorithm for learning a ranking function from large-scale data sets, IEEE Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1158–1170, 200.
[153] Muhammad Sahimi, Hossein Hamzehpour, Efficient computational strategies for solving global optimization problems, Comput. Sci. Eng. 12 (4) (2010) 74–83.
[154] Sherif Sakr, Anna Liu, Daniel M. Batista, Mohammad Alomari, A survey of large scale data management approaches in cloud environments, IEEE Commun. Surv. Tutorials 13 (3) (2011) 311–336.
[155] Ted Samson, Splunk Storm Brings Log Management to the Cloud, 2012. <http://www.infoworld.com/t/managed-services/splunk-storm-brings-log-management-the-cloud-201098?source=footer>.
[156] Diana Samuels, Skytree: Machine Learning Meets Big Data, February 2012. <http://www.bizjournals.com/sanjose/blog/2012/02/skytree-machine-learning-meets-big-data.html?page=all>.
[157] Eric Savitz, Gartner: 10 Critical Tech Trends for the Next Five Years, October 2012. <http://www.forbes.com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five-years/>.

[158] Eric Savitz, Gartner: Top 10 Strategic Technology Trends for 2013, October 2012. <http://www.forbes.com/sites/ericsavitz/2012/10/23/gartner-top-10-strategic-technology-trends-for-2013/>.
[159] Eric E. Schadt, Michael D. Linderman, Jon Sorenson, Lawrence Lee, Garry P. Nolan, Computational solutions to large-scale data management and analysis, Nat. Rev. Genet. 11 (9) (2010) 647–657.
[160] Gayathri Seenumani, Jing Sun, Huei Peng, Real-time power management of integrated power systems in all electric ships leveraging multi time scale property, IEEE Trans. Control Syst. Technol. 20 (1) (2012) 232–240.
[161] Udo Seiffert, Training of large-scale feed-forward neural networks, in: International Joint Conference on Neural Networks, IJCNN '06, 2006, pp. 5324–5329.
[162] Haiying Shen, Lianyu Zhao, Ze Li, A distributed spatial-temporal similarity data storage scheme in wireless sensor networks, IEEE Trans. Mobile Comput. 10 (7) (2011) 982–996.
[163] Xiaohui Shen, Wei keng Liao, Alok Choudhary, Gokhan Memik, Mahmut Kandemir, A high-performance application data environment for large-scale scientific computations, IEEE Trans. Parallel Distrib. Syst. 14 (12) (2003) 1262–1274.
[164] Zeqian Shen, Kwan-Liu Ma, Tina Eliassi-Rad, Visual analysis of large heterogeneous social networks by semantic and structural abstraction, IEEE Trans. Visual. Comput. Graph. 12 (6) (2006) 1427–1439.
[165] Weiya Shi, Yue-Fei Guo, Cheng Jin, Xiangyang Xue, An improved generalized discriminant analysis for large-scale data set, in: Seventh International Conference on Machine Learning and Applications, 2008, 2008, pp. 769–772.
[166] Katsunari Shibata, Yusuke Ikeda, Effect of number of hidden neurons on learning in large-scale layered neural networks, in: ICROS-SICE International Joint Conference 2009, 2009, pp. 5008–5013.
[167] Andrew Horne Shvetank Shah, Jaime Capellá, Good Data won't Guarantee Good Decisions, 2012. <http://hbr.org/2012/04/good-data-wont-guarantee-good-decisions>.
[168] Dimitra Simeonidou, Reza Nejabati, Georgios Zervas, Dimitrios Klonidis, Anna Tzanakaki, Mike J. O Mahony, Dynamic optical-network architectures and technologies for existing and emerging grid services, J. Lightwave Technol. 23 (5) (2005) 3347–3357.
[169] Simeon Simoff, Michael H. Böhlen, Arturas. Mazeika, Visual Data Mining: Theory, Techniques and Tools for Visual Analytics, Springer, 2008.
[170] Daniel R. Simon, On the power of quantum computation, SIAM J. Comput. 26 (1994) 116–123.
[171] Moshe Sipper, Eduardo Sanchez, Daniel Mange, Marco Tomassini, Andrés Pérez-Uribe, André Stauffer, A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems, IEEE Trans. Evol. Comput. 1 (1) (1997) 83–97.
[172] Matthew Smith, Christian Szongott, Benjamin Henne, Gabriele von Voigt, Big data privacy issues in public social media, in: 2012 6th IEEE International Conference on Digital Ecosystems Technologies (DEST), 2012, pp. 1–6.
[173] Myra Spiliopoulou, Michael Hatzopoulos, Yannis Cotronis, Parallel optimization of large join queries with set operators and aggregates in a parallel environment supporting pipeline, IEEE Trans. Knowl. Data Eng. 8 (3) (1996) 429–445.
[174] Pavan Sridhar, Neha Dharmaji, A comparative study on how big data is scaling business intelligence and analytics, Int. J. Enhanced Res. Sci. Technol. Eng. 2 (8) (2013) 87–96.
[175] Michael Stonebraker, Uğur Çtintemel, Stan Zdonik, The 8 requirements of real-time stream processing, SIGMOD Rec. 34 (4) (2005) 42–47.
[176] Shun-Feng Su, Chen-Chia Chuang, C.W. Tao, Jin-Tsong Jeng, Chih-Ching Hsiao, Radial basis function networks with linear interval regression weights for symbolic interval data, IEEE Trans. Syst. Man Cyber.–Part B: Cyber. 19 (6) (2011) 1141–1151.
[177] Ping Sun, Xin Yao, Sparse approximation through boosting for learning large scale kernel machines, IEEE Trans. Neural Networks 21 (6) (2010) 883–894.
[178] Alex Szalay, Jim Gray, Science in an exponential world, Nature 440 (2006) 23–24.
[179] Alexander S. Szalay, Extreme data-intensive scientific computing, Comput. Sci. Eng. 13 (6) (2011) 34–41.
[180] Ke Tang, Minlong Lin, Fernanda L. Minku, Xin Yao, Selective negative correlation learning approach to incremental learning, Neurocomputing 72 (13-15) (2009) 2796–2805.
[181] David Taniar, High performance database processing, in: 2012 IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), 2012, pp. 5–6.
[182] David Thompson, Joshua A. Levine, Janine C. Bennett, Peer-Timo Bremer, Attila Gyulassy, Valerio Pascucci, Philippe P. Pébay, Analysis of large-scale scalar data using hixels, in: 2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2011, pp. 23–30.
[183] Ewaryst Tkacz, Adrian Kapczyński, Internet: Technical Development and Applications, Springer, 2009.
[184] P. Vettiger, G. Cross, M. Despont, U. Drechsler, U. Durig, B. Gotsmann, W. Haberle, M.A. Lantz, H.E. Rothuizen, R. Stutz, G.K. Binnig, The millipede – nanotechnology entering data storage, IEEE Trans. Nanotechnol. 1 (1) (2002) 39–55.
[185] Senthilkumar Vijayakumar, Anjani Bhargavi, Uma Praseeda, Syed Azar Ahamed, Optimizing sequence alignment in cloud using hadoop and mpp database, in: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), 2012, pp. 819–827.
[186] Mladen A. Vouk, Cloud computing – issues, research and implementations, in: 30th International Conference on Information Technology Interfaces, 2008, ITI 2008, 2008, pp. 31–40.
[187] Fei-Yue Wang, Daniel Zeng, Kathleen M. Carley, Wenji Mao, Social computing: from social informatics to social intelligence, IEEE Intell. Syst. 22 (2) (2007) 79–83.
[188] Lijuan Wang, Jun Shen, Towards bio-inspired cost minimisation for data-intensive service provision, in: 2012 IEEE First International Conference on Services Economics (SE), 2012, pp. 16–23.
[189] Qian Wang, Kui Ren, Wenjing Lou, Yanchao Zhang, Dependable and secure sensor data storage with dynamic integrity assurance, in: Proc. IEEE INFOCOM, 2009, pp. 954–962.
[190] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distrib. Syst. 22 (5) (2011) 847–859.
[191] Peter Wayner. 7 Top Tools for Taming Big Data, 2012. <http://www.networkworld.com/reviews/2012/041812-7-top-tools-for-taming-258398.html>.
[192] Ron Weiss, Subhayu Basu, Sara Hooshangi, Abigail Kalmbach, David Karig, Rishabh Mehreja, Ilka Netravali, Genetic circuit building blocks for cellular computation, communications, and signal processing, Natural Comput. 2 (2003) 47–84.
[193] Leland Wilkinson, The future of statistical computing, Technometrics 50 (4) (2008) 418–435.
[194] William J. Worlton, Bulk storage requirements in large-scale scientific calculations, IEEE Trans. Magn. 7 (4) (1971) 830–833.
[195] Yingcai Wu, Guo-Xun Yuan, Kwan-Liu Ma, Visualizing flow of uncertainty through analytical processes, IEEE Trans. Visual. Comput. Graph. 18 (12) (2012) 2526–2535.
[196] Jian xiong Dong, Adam Krzyżak, Ching Y. Suen, Fast svm training algorithm with decomposition on very large data sets, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 603–618.
[197] Jun Yan, Ning Liu, Shuicheng Yan, Qiang Yang, Weiguo Fan, Wei Wei, Zheng Chen, Trace-oriented feature analysis for large-scale text data dimension reduction, IEEE Trans. Knowl. Data Eng. 23 (7) (2011) 1103–1117.
[198] Zhenyu Yang, Ke Tang, Xin Yao, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci. 178 (15) (2008) 2985–2999.
[199] Wen Yao, Xiaoqian Chen, Yong Zhao, Michel van Tooren, Concurrent subspace width optimization method for rbf neural network modeling, IEEE Trans. Neural Networks Learn. Syst. 23 (2) (2012) 247–259.
[200] Dong Yu, Li Deng, Deep learning and its applications to signal and information processing, IEEE Signal Process. Mag. 28 (1) (2011) 145–154.
[201] Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu úlfar Erlingsson, Pradeep Kumar Gunda, Jon Currey, Dryadlinq: a system for general-purpose distributed data-parallel computing using a high-level language, in: 8th USENIX Symposium on Operating Systems Design and Implementation, 2008.
[202] Jiawei Yuan, Shucheng Yu, Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing, 2013.

[203] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, Cheng Chen, Data-driven intelligent transportation systems: a survey, IEEE Trans. Intell. Trans. Syst. 12 (4) (2011) 1624–1639.
[204] Yu Zhang, Mihaela van der Schaar, Information production and link formation in social computing systems, IEEE J. Sel. Areas Commun. 30 (1) (2012) 2136–2145.
[205] Jin Zhou, C.L. Philip Chen, Long Chen, Hong-Xing Li, Wei Zhao, A collaborative fuzzy clustering algorithm in distributed network environments, IEEE Trans. Fuzzy Syst. PP (99) (2013) 1.
[206] Qi Zhou, Peng Shi, Honghai Liu, Shengyuan Xu, Neural-network-based decentralized adaptive output-feedback control for large-scale stochastic nonlinear systems, IEEE Trans. Syst. Man Cyber Part B: Cyber 46 (6) (2012) 1608–1619.
[207] Paul Zikopoulos, Chris Eaton, Paul. Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw Hill Professional, 2011.