Big Data Storage and Management: Challenges and Opportunities

Jaroslav Pokorný(✉)

MFF UK, Malostranské nám. 25, 118 00 Praha, Czech Republic
pokorny@ksi.mff.cuni.cz

Abstract. The paper is focused on today's very popular theme – Big Data. We describe and discuss its characteristics by eleven V's (Volume, Velocity, Variety, Veracity, etc.) and Big Data quality. These characteristics represent both data and process challenges. Then we continue with problems of Big Data storage and management. Principles of NoSQL databases are explained including their categorization. We also shortly describe Hadoop and MapReduce technologies as well as their inefficiency for some interactive queries and applications within the domain of large-scale graph processing and streaming data. NoSQL databases and Hadoop M/R are designed to take advantage of cloud computing architectures and allow massive computations to be run inexpensively and efficiently. The term of Big Data 1.0 was introduced for these technologies. We continue with some new approaches called currently Big Data 2.0 processing systems. Particularly their four categories are introduced and discussed: General purpose Big Data Processing Systems, Big SQL Processing Systems, Big Graph Processing Systems, and Big Stream Processing Systems. Then, an attention is devoted to Big Analytics – the main application area for Big Data storage and processing. We argue that enterprises with complex, heterogeneous environments no longer want to adopt a BI access point just for one data source (Hadoop). More heterogeneous software platforms are needed. Even Hadoop has become a multipurpose engine for ad hoc analysis. Finally, we mention some problems with Big Data. We also remind that Big Data creates a new type of digital divide. Having access and knowledge of Big Data technologies gives companies and people a competitive edge in today's data driven world.

Keywords: Big Data · NoSQL databases · MapReduce · Hadoop · Big Data 2.0
Big Analytics

1 Introduction

One rather subjective definition by Kushal Agraval[1] says that Big Data can be as data that exceeds the processing capacity of conventional database systems. Consequently, its storage and processing require

[1] https://kushalagrawal.com/blog/big-data/.

J. Hřebíček et al. (Eds.): ISESS 2017, IFIP AICT 507, pp. 28–38, 2017.
https://doi.org/10.1007/978-3-319-89935-0_3

- new data architectures, analytic environments,
- new analytical methods,
- new tools.

In the business sphere, Big Data is data whose scale, distribution, diversity, and/or timelines require the use these new technologies to enable insights to new sources of business value.

Usually some examples from the commercial world are presented for documenting the size of Big Data. The most known example concerning the Google's database mentions gross total estimate of all data Google saved by 2016 as approximately 10EBytes[2]. A. Orlova stated in 2015[3] that Facebook generates about 10 TBytes every day, Twitter generates about 7 TBytes and some enterprises generate TBytes every single hour. In general, the digital universe is doubling in size every two years, and by 2020 – the data we create and copy annually – will reach 44 ZBytes or 44 trillion GBytes[4]. In the near future, the "Big Data" problem will begin to emerge in every enterprise.

We will consider Big Data for both *data-at-rest* as well as *data-in-motion*. For Big Data at rest we describe two kinds of systems: (1) NoSQL systems for interactive data processing; (2) systems for large scale analytics, e.g. decision support, based on MapReduce paradigm, represented by tools such as Hadoop. Hadoop-based systems enable to run long running decision support and analytical queries consuming and possible producing bulk data. Data-in-motion is the process of analyzing data on the fly without storing it. We utilize real-time processing methods in this case.

Today, users have a number of options associated with the above mentioned issues [9]. For storing and processing large datasets they can use:

- traditional parallel database systems (shared nothing architectures),
- distributed file systems and Hadoop technologies,
- key-value datastores (so-called NoSQL databases),
- new database architectures (e.g., NewSQL databases).

The Big Data landscape is dominated by two classes of technologies: systems that provide operational capabilities, i.e. *operation systems* for real-time, interactive workloads where data is primarily captured and stored; and *analytical systems* that provide analytical capabilities for retrospective, complex analysis that may use most of all the data. Usually we talk about *Big Data analytics* (shortly *Big Analytics*). These classes of Big Data technology are complementary and frequently deployed together.

Big Data storage and processing are appropriate for cloud services. This approach reinforces requirements on the availability and scalability of computational resources offered by cloud services. Authors of [5] highlight this role of cloud computing. Cloud has given enterprises the opportunity to fundamentally shift the way data is created, processed and shared.

[2] https://www.quora.com/How-big-is-Googles-database/.

[3] http://blog.azoft.com/telcos-gain-valuable-insight-with-big-data/.

[4] https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm.

The rest of the paper is organized as follows. Section 2 introduces traditionally some V's characterizing Big Data and some immediate challenges arising from them. Section 3 provides basic characteristics of NoSQL databases. Section 4 shortly introduces principles of MapReduce and Hadoop technology (Big Data 1.0). Big Data 2.0 processing systems are discussed in Sect. 5. Section 6 is devoted to the most important part of Big Data application domain - Big Analytics. Some problems with the Big Data are presented in Sect. 7. Section 8 gives the conclusion.

2 Big Data Characteristics

Big Data is most often characterized by several V's. In [11] we discussed eight such characteristics:

- *Volume*: Volume refers to the quantity of data generated and stored from various sources. Data scale in the range of TBytes to PBytes and even more. The big volume is not only a storage issue but also influences Big Analytics. Not only data samples, but often all data is captured for analysis.
- *Velocity*: Both how quickly data is being produced and how quickly the data must be processed to meet demand (e.g. streaming data). For many applications, the speed of data creation is more important than the volume. For example, a well-known source of high-velocity data is social media. Twitter users are estimated to generate nearly 100,000 tweets every 60 s.
- *Variety*: Data is of many format types – structured (e.g., call detail records in a telecom company), unstructured (product reviews on twitter), semi-structured (e.g., graph data), media, etc. Data does not come only from business transactions, but also from machines, sensors, GPS signals from cell phones, and other sources, making it much more complex to manage. There is a need to integrate this data together. From the analytics perspective, variety of data is the biggest challenge to effectively use it. It is becoming the single biggest driver of Big Data investments. Technically, some connectors are becoming crucial in integration of different data.
- *Veracity*: Managing the reliability and predictability of inherently imprecise data, e.g. to test many different hypotheses, vast training samples, etc. It means data needs to be cleaned before it can be integrated.
- *Value*: Indicates if the data is worthwhile and has value for business. Data value vision includes creating social and economic added value based on the intelligent use, management and re-uses of data sources with a view to increase business intelligence (BI). Also, an attention must be paid to the investment of storage for data. For example, storage may be cost effective and relatively cheaper at the time of purchase but it can be unreliable. Saving money can cause a risk in this case.
- *Visualization:* Concerns visual representations and insights for decision making. For example, SAS offers five Big Data challenges related to visualization and Big Data[5]:
  – meeting the need for speed,
  – understanding the data,

[5] https://www.sas.com/resources/asset/five-big-data-challenges-article.pdf.

  - addressing data quality,
  - displaying meaningful results, and
  - dealing with outliers.
- *Variability*: The different meanings/contexts associated with a given piece of data is considered. Variability even refers to data whose meaning is constantly changing. Thus, variability is different from variety.
- *Volatility*: How long the data is valid and how long should be stored, i.e. at what point is data no longer relevant to the current analysis. For example, an online e-commerce company may not want to keep a one year customer purchase history.

The first three V's have been introduced by Gartner in [6], the V associated with Veracity has been added by Snow in his blog [13]. The fifth V was introduced by Gamble and Goble in [4].

Borne adds three other V's in [2]:

- *Venue:* Considers distributed, heterogeneous data from multiple platforms, from different owners' systems, with different access and formatting requirements, private vs. public cloud.
- *Vocabulary*: Includes schema, data models, semantics, ontologies, taxonomies, and other content- and context-based metadata that describe the data's structure, syntax, content, and provenance.
- *Vagueness*: Concerns a confusion over the meaning of Big Data. Is it Hadoop? Is it something that we've always had? What's new about it? What are the tools? Which tools should I use? etc.

Not only V's characterize Big Data. Often a quality is accentuated [8].

- *Quality*: Quality characteristic measures how the data is reliable to be used for making decisions. Saying that the quality of data is high or low is basically dependent on four parameters: (a) Complete: all relevant data is available, for example all details of vendors like name, address, bank account, etc., exist (b) Accurate: data is free of misspelling, typos, wrong terms and abbreviations (c) Available: data is available when requested and easy to find (d) Timely: data is up to date and ready to support decision.

These characteristics are not independent. For example, veracity (confidence or trust in the data) drops when volume, velocity, variety and variability increase. Sometimes, a *validity* (additional V) is considered. Similar to veracity, validity refers to how accurate and correct the data is for its intended use.

The characteristics represent challenges related to data itself, i.e. *data challenges* [8]. The tasks like data acquisition, cleaning, curation, integration, storage, processing, indexing, search, sharing, transfer, mining, analysis, and visualization are called *process challenges* in [8].

We can find a lot of other characteristics of Big Data. For example, Tyrone Systems[6] company distinguishes 10 Big Data challenges in two categories: cultural and

[6] http://blog.tyronesystems.com/the-top-10-big-data-challenges/.

technological. The former tackles the legal and ethical issues related to accessing data, e.g. privacy, security, and governance. In [8] they are called *management challenges*. The latter includes, in addition to the continued development of effective dealing with Big Data, putting results of Big Data Analysis in a presentable form for making decisions, i.e. it emphasizes a visualization and visual models.

3 NoSQL Databases

Considering Big Data we do not suppose an architecture with a database stored on the large disk. We refer to a much wider technology environment, which is coined under the term of *Big Data Ecosystem* (BDE) and relates to all interconnected parts, ranging from required infrastructure to data itself. A part of the BDE is represented by the NoSQL distributed databases. They include four main categories: key-value, column-oriented, document stores, and graph databases (see, e.g., [10]). Their data models are different and they can hardly be categorized in a precise way. Some typical user characteristics of these categories can be described as follows:

- *Key-value*. A user can store and retrieve data using keys in schema-less way. A key is a unique identifier for some data item. The data items, so-called values, are stored against these keys. They may be, e.g., a scalar (string), a hash, a list, a set, a sorted set, etc. Technically, a key-value store is just a distributed persistent associative array (map). It is suitable for rapid access to unstructured data, but it is inefficient when querying or updating part of a value is necessary. Examples of these databases are Redis[7] and Memcached[8].
- *Column-oriented*. A column is a key-value pair, where the key is a qualifier and the value is related to the qualifier. A data row has a sortable row key and an arbitrary number of columns. Columns are often grouped into columns families. These databases are suitable for very rapid access to structured or semi-structured data. Their examples include well-known systems Cassandra[9] and HBase[10].
- *Document datastores*. Data is stored as documents. Documents are data structures composed of key-value pairs. Documents can contain many different key-value pairs, or key-array pairs, or even hierarchically nested document parts (usually in JSON-style). Sometimes, a document database can contain a number of document collections. Examples of document databases include MongoDB[11] and Amazon DynamoDB[12].
- *Graph databases*. Graph databases allow to store information about entities and relationships between these entities. In graph-oriented terms these databases use edges and nodes to represent and store data. These nodes are organized by some

[7] https://redis.io/.

[8] http://memcached.org/.

[9] http://cassandra.apache.org/.

[10] http://hbase.apache.org/.

[11] https://www.mongodb.com/.

[12] https://aws.amazon.com/dynamodb/.

relationships with one another, which is represented by edges between the nodes. Both the nodes and the relationships have some defined properties. The most known graph database available is Neo4j[13] and OrientDB[14].

There is DB-Engines initiative[15] to collect and present information on DBMSs. It provides a DB-engines ranking service which ranks DBMSs according to their popularity. The ranking is updated monthly. The examples of DBMSs presented above come from the first two places of ranking lists for particular categories.

4 MapReduce and Hadoop

Google introduced the MapReduce (M/R) [3] framework in 2004 for processing massive amounts of data over highly distributed clusters of nodes. M/R represents a generic framework to write massive scale data applications. It involves writing two user defined generic functions: *map* and *reduce*. In the map step, a *master node* takes the input data and the processing problem, divides it into smaller data chunks and sub-problems and distributes them to *worker nodes*. A worker node processes one or more chunks using the sub-problem assigned to it. Specifically, each map process takes a set of {`key`, `value`} pairs as input and generates one or more intermediate {`key`, `value`} pairs for each input `key`. In the reduce step, intermediate `{key, value}` pairs are processed to produce the output of the input problem. Each reduce instance takes a key and a set of values as input and produces output after processing a smaller set of values:

$$\mathtt{Map}(\mathtt{k}_1,\mathtt{v}_1) \rightarrow \mathtt{list}(\mathtt{k}_2,\mathtt{v}_2)$$
$$\mathtt{Reduce}(\mathtt{k}_2,\ \mathtt{list}(\mathtt{v}_2)) \rightarrow \mathtt{list}(\mathtt{k}_3,\mathtt{v}_3)$$

Consequently, one of the main advantages of this approach is that it isolates the application from the details of running a distributed program, such as issues on data distribution, scheduling and fault tolerance.

Many NoSQL databases are based on Apache™ *Hadoop Distributed File System*[16] (HDFS), which is a part so-called *Hadoop software stack*. HDFS is a massively distributed file system designed to run on cheap commodity hardware. Open-source software Hadoop[17] is based on the M/R implementation along with HDFS.

The stack enables to access data by three different sets of tools in particular layers which distinguishes it from the universal DBMS architecture with only SQL API in the outermost layer. The NoSQL HBase is available as a column-oriented key-value layer with `Get/Put` operations as input. Hadoop M/R system server in the middle layer enables to create M/R jobs, i.e., programs in a programming language. It is often emphasised that writing custom M/R jobs is difficult and time-consuming.

[13] https://neo4j.com/.

[14] http://www.orientechnologies.com/.

[15] http://db-engines.com/en/.

[16] http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf.

[17] http://hadoop.apache.org/.

Consequently, high-level languages HiveQL, PigLatin, and Jaql are at disposal for some users at the outermost layer. HiveQL is an SQL-like language representing a subset of SQL92, and therefore can be simply understood by SQL users. Jaql is a declarative scripting language for analysing large semi-structured datasets. Pig Latin is not declarative. Whose programs are series of assignments similar to an execution plan for relational operations in a relational DBMS.

NoSQL databases and Hadoop M/R are designed to take advantage of cloud computing architectures and allow massive computations to be run inexpensively and efficiently. This makes operational Big Data workloads much easier to manage, and cheaper and faster to implement. Some NoSQL systems provide native M/R functionality that allows for analytics to be performed on operational data in place. The term of *Big Data 1.0* was introduced for these technologies. Hadoop is its main representative.

5 Big Data 2.0 Processing Systems

For at least 10 years the M/R framework has represented the de facto standard for Big Data processing. Its fundamental principle is to move analysis to the data, i.e. to program applications in a data-centric fashion and not moving the data to an analytical system.

On the other, the research and development in the last years recognized some limitations of this approach. It is extremely complex to integrate, deploy, operate, and manage massive Hadoop environments. Hadoop cluster thinking requires special programmer skills to deploy the system and process data. Also, in processing large-scale structured data, several studies reported on the significant inefficiency of the Hadoop framework. The reason is that Hadoop is a file system built on batch processing. The Hadoop framework has also been shown to be inefficient within the domain of large-scale graph processing and streaming data [1]. It confirms a similar situation in data processing history analyzed by Stonebraker in his famous paper [14] in context of traditional relational DBMSs. He makes the argument that the relational DBMS cannot be extended ad infinitum, demonstrates how RDBMSs are inappropriate for several new applications, and argues that the DBMS market will fragment into a series of special-purpose engines. Thus a new wave of domain-specific systems for Big Data management has occurred in last years. They constitute a new generation of systems referred as *Big Data 2.0* processing systems [1, 12].

Bajaber et al. [1] distinguish four categories of Big Data 2.0 processing systems.

General Purpose Big Data Processing Systems. For example, Apache Spark[18], is an open source Big Data processing framework built around speed, ease of use, and sophisticated analytics. In a survey[19] nearly 70% of the respondents favoured Spark over dominating MapReduce, which is not appropriate to interactive applications or real-time stream processing.

[18] https://spark.apache.org/.

[19] http://www.syncsort.com/en/About/News-Center/Press-Release/New-Hadoop-Survey-Identifies-Big-Data-Trends.

Apache Spark provides programmers with an application programming interface centred on a data structure called the *resilient distributed dataset* (RDD), a read-only multiset of data items distributed over a cluster of machines that is maintained in a fault-tolerant way. It was developed in response to limitations in the M/R computing paradigm, which forces a particular linear dataflow structure on distributed programs. Spark's RDDs serve as a working set for distributed programs that offers a (deliberately) restricted form of distributed shared memory. Companies like ORACLE and SAP talk even about *Big Data Management Systems*.

Big SQL Processing Systems. SQL-on-Hadoop is a class of analytical application tools that combine established SQL-style querying with newer Hadoop data framework elements. Some examples of this technology:

- HadoopDB[20] is a hybrid combining a parallel database with Hadoop. It translates SQL queries into M/R jobs and optimizes query plans. It uses Postgres on a communication level and Hive on the translation level.
- HPE Vertica SQL on Apache Hadoop®[21] offers to perform SQL queries on Hadoop data.
- Splice Machine[22] is a Hadoop-relational DBMS. It uses HBase and HDFS as a file system. It supports real-time ACID transactions.
- BigSQL[23] is PostgresSQL implemented on Hadoop.

Most of the SQL-on-Hadoop solutions access directly HDFS, i.e. not through M/R jobs. Query accelerators based on SQL-on-Hadoop and OLAP-on-Hadoop technologies are blurring differences between traditional warehouses to the world of Big Data.

Big Graph Processing Systems. Although graph processing algorithms can be written with M/R, this approach is not appropriate for this purpose and leads to inefficient performance. Apache Giraph[24] is a graph-processing framework built on top of Hadoop. Giraph is based on the graph processing system Pregel [7] by Google.

Big Stream Processing Systems. Stream computing is a new paradigm occurring in context of scenarios like mobile devices, location services and sensor pervasiveness. Data is usually generated from multiple sources and are sent asynchronously to servers. Now, a new category of *Data Stream Management Systems* occurs. They are developed for real-time processing of data-in-motion, e.g. for analysis of data streams.

[20] http://db.cs.yale.edu/hadoopdb/hadoopdb.html.

[21] https://www.vertica.com/.

[22] https://www.splicemachine.com/.

[23] https://www.bigsql.org/.

[24] http://giraph.apache.org/.

6 Big Analytics

Big Analytics is the process of examining large data sets to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. The analytical findings can lead to more effective marketing, better customer service, new revenue opportunities, improved operational efficiency, competitive advantages over rival organizations and other business benefits. This definition of Amazon Web Services[25] emphasises clearly a purpose such analytics. Authors of [16] talk about Big Analytics as of the execution of machine learning tasks on large data sets in cloud computing environments.

In the previous chapters, we saw that there are some technologies combining analytics of Big Data with Hadoop. But enterprises with complex, heterogeneous environments no longer want to adopt a BI access point just for one data source (Hadoop). Answers to their questions are buried in a host of sources ranging from systems of records to cloud warehouses, to structured and unstructured data from both Hadoop and non-Hadoop sources, as it is emphasized in trends Big Data formulated by Tableau[26]. Incidentally, even relational DBMSs are becoming Big Data-ready. SQL Server 2016, for instance, recently added JSON support.

As regards actual Hadoop software, Tableau also emphasizes that it is no longer just a batch-processing platform for some analytical tasks. Hadoop has become a multipurpose engine for ad hoc analysis. It is even being used for operational reporting on day-to-day workloads—the kind traditionally handled by data warehouses. There is a growing trend of Hadoop becoming a core part of the enterprise IT landscape. Making Hadoop data accessible to business users is now one of the biggest challenges.

7 Problems with Big Data

Kushal Agraval[27] mentioned in his blog the following problems connected to Big Data technologies:

- Bigger data is not always better data. Quantity does not necessarily mean quality, see, e.g., data from social networks. Hence, the data filtering for useful information is a challenge in this context.
- Big Data is prone to data errors. Sometimes errors or bias are undetected owing to the size of the sample and thus produce inaccurate results [15].
- Big Analytics is often subjective. There can be multiple ways to look at the same information and to interpret it differently by different users.
- Not all the data is useful. It means, collecting data which is never used or which does not answer a particular question is relatively useless.

[25] http://searchbusinessanalytics.techtarget.com/definition/big-data-analytics.

[26] https://www.tableau.com/sites/default/files/media/Whitepapers/whitepaper_top_10_big_data_trends_2017.pdf.

[27] https://kushalagrawal.com/.

- Accessing Big Data raises ethical issues. Both in industry and in academics the issues of privacy and accountability with respect to Big Data have now raised important concerns.
- Big Data creates a new type of digital divide. Having access and knowledge of Big Data technologies gives companies and people a competitive edge in today's data driven world.

8 Conclusions

We conclude with the 10 hottest Big Data technologies based on Forrester's analysis from 2016[28]. They concern continuing development of NoSQL databases, distributed datastores, in-memory data fabric (dynamic random access memory, flash, or SSD), data preparation (sourcing, shaping, cleansing, and sharing diverse and messy data sets), and data quality. Data virtualization and data integration should contribute to delivering information from various data sources and to data orchestration across various exiting solutions (Hadoop, NoSQL, Spark, etc.). Additional technologies, i.e. predictive analysis, search and knowledge discovery, and stream analytics should support Big Analytics applications. On the other hand, the biggest challenge does not seem the technology itself. The more important problem is, how to have enough skills to make effective use of these technologies at disposal and make sense out of the data collected[29].

Acknowledgments. This work was supported by the Charles University project Q48.

References

1. Bajaber, F., Elshawi, R., Batarfi, O., Altalhi, A., Barnawi, A., Sakr, S.: Big data 2.0 processing systems: taxonomy and open challenges. J. Grid Comput. **14**, 379–405 (2016)
2. Borne, K.: Top 10 Big Data Challenges – A Serious Look at 10 Big Data V's. https://www.mapr.com/blog/top-10-big-data-challenges-serious-look-10-big-data-vs
3. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
4. Gamble, M., Goble, C.: Quality, trust and utility of scientific data on the web: toward a joint model. In: Proceedings of WebSci 2011 Conference, Koblenz, Germany, Article No. 15. ACM (2011)
5. Gupta, R., Gupta, H., Mohania, M.: Cloud computing and big data analytics: what is new from databases perspective? In: Srinivasa, S., Bhatnagar, V. (eds.) BDA 2012. LNCS, vol. 7678, pp. 42–61. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35542-4_5
6. Laney, D.: 3D data management: controlling data volume, velocity and variety. Meta Group, Gartner (2001). http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

[28] http://www.forbes.com/sites/gilpress/2016/03/14/top-10-hot-big-data-technologies/#5b66b0327f26.

[29] http://www.ebusinessbook.nl/185.

7. Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel: a system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, pp. 135–146 (2010)
8. Nasser, T., Tariq, R.S.: Big data challenges. J. Comput. Eng. Inf. Technol. **4**(3), 1–6 (2015)
9. Pokorny, J.: Database technologies in the world of big data. In: Proceedings of the 16th International Conference on Computer Systems and Technologies, CompSysTech 2015. ACM International Conference Proceeding Series, vol. 1008, pp. 1–12. ACM, New York (2015)
10. Pokorný, J.: Graph databases: their power and limitations. In: Saeed, K., Homenda, W. (eds.) CISIM 2015. LNCS, vol. 9339, pp. 58–69. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24369-6_5
11. Pokorný, J., Stantic, B.: Challenges and opportunities in big data processing (Chapter 1). In: Ma, Z. (ed.) Managing Big Data in Cloud Computing Environments. IGI Global, Advances in Data Mining and Database Management (2016)
12. Sakr, S.: Big Data 2.0 Processing Systems - A Survey. Springer Briefs in Computer Science. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38776-5
13. Snow, D.: Dwaine Snow’s Thoughts on Databases and Data Management (2012). http://dsnowondb2.blogspot.cz/2012/07/adding-4th-v-to-big-data-veracity.html
14. Stonebraker, M.: Technical perspective - one size fits all: an idea whose time has come and gone. Commun. ACM **51**(12), 76 (2008)
15. Tivari, S.: Professional NoSQL. Wiley/Wrox, Hoboken (2011)
16. Wu, C., Buyya, R., Ramamohanarao, K.: Big data analytics = machine learning + cloud computing. In: Buyya, R., Calheiros, R., Dastjerdi, A. (eds.) Big Data: Principles and Paradigms. Morgan Kaufmann, Burlington (2016)