Big Data: A Survey
The New Paradigms, Methodologies and Tools

Enrico Giacinto Caldarola[1,2] and Antonio Maria Rinaldi[1,3]
[1]*Department of Electrical Engineering and Information Technologies, University of Naples Federico II, Napoli, Italy*
[2]*Institute of Industrial Technologies and Automation, National Research Council, Bari, Italy*
[3]*IKNOS-LAB Intelligent and Knowledge Systems, University of Naples Federico II, LUPT 80134, via Toledo, 402-Napoli, Italy*

Keywords: Big Data, NoSQL, NewSQL, Big Data Analytics, Data Management, Map-reduce.

Abstract: For several years we are living in the era of information. Since any human activity is carried out by means of information technologies and tends to be digitized, it produces a humongous stack of data that becomes more and more attractive to different stakeholders such as data scientists, entrepreneurs or just privates. All of them are interested in the possibility to gain a deep understanding about people and things, by accurately and wisely analyzing the gold mine of data they produce. The reason for such interest derives from the competitive advantage and the increase in revenues expected from this deep understanding. In order to help analysts in revealing the insights hidden behind data, new paradigms, methodologies and tools have emerged in the last years. There has been a great explosion of technological solutions that arises the need for a review of the current state of the art in the Big Data technologies scenario. Thus, after a characterization of the new paradigm under study, this work aims at surveying the most spread technologies under the Big Data umbrella, throughout a qualitative analysis of their characterizing features.

1 INTRODUCTION

The ubiquitous of ICTs in all human activities and the increasing digitization of the world have led to a great availability of data coming from different sources. People moving around the world, individuals opinions and sentiments about events, facts or products — gathered by the most popular social media through handy smartphones — the data coming from the increasingly widespread sensors inside machines or worn by people are just few examples of the different sources generating an explosion of data today. Also sciences have been interested by a new paradigm coming from the data explosion. Many physicists have started to perform sophisticated algorithms over large data sets inside huge computer clusters, rather than by directly observing natural phenomena. Lots of knowledge about the universe in the last decades, for example, comes from sophisticated data analysis rather than looking through telescopes (Hey et al., 2009). The rate of data growth over years is amazing: according to ScienceDaily, a full 90% of all the data in the world has been generated over the last two years (Dragland, 2013). All of this represents a real *tsunami* requiring a paradigmatic shift respect to the past as for theories, technologies or approaches in data management and more attention to survive it (Caldarola et al., 2014). According to several authors, the data explosion we see today fall under the new term *Big Data* that is receiving a lot of buzz in the recent years (Franks, 2012). A look at Google Trends shows that, starting from 2011 until today, the term Big Data has been increasingly growing in popularity over time even though, despite the rapid spread of the term, there is not a single definition encompassing all its facets and it still remains elusive (Weinberg et al., 2013). Depending on the different perspective from which the problem of managing large data sets can be seen, we can define Big Data in several ways. From a technological perspective, Big Data represents "data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze" (Manyika et al., 2011). It may also refers to "data which exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it within a tolerable elapsed time for its user" (Merv, 2011). An important aspect of the previous definitions is that

Caldarola E. and Rinaldi A..
Big Data: A Survey - The New Paradigms, Methodologies and Tools.
DOI: 10.5220/0005580103620370
In *Proceedings of 4th International Conference on Data Management Technologies and Applications* (KomIS-2015), pages 362-370
ISBN: 978-989-758-103-8

what qualifies as Big Data will change over time with technology progress (Franks, 2012). For this reason, VP and analyst at Gartners Intelligence and Information Management Group Donald Feinberg stated that the bigness of Big Data is a moving target and what was Big Data historically or what is Big Data today wont be Big Data tomorrow. This is true to the extent that the term Big Data itself is going to disappear in the next years, to become just Data or Any Data. Taking into account the variability of the definition over the time, Adam Jacob provided the following statement: Big Data should be defined at any point in time as data whose size force us to look beyond the tried-and-true methods that are prevalent at that time (Jacobs, 2009). From a marketers point of view, Big Data is an organizational and decision problem. It is not a technology problem but a business problem (Weinberg et al., 2013). Finally, from a user point of view, Big Data can be understood as new exciting, advanced software tools which replace the existing ones. Perspectives aside, the authors define Big Data as a new time-variant paradigm in data management whose *raison d'être* comes from the enormous availability of data in every human activity that needs to be acknowledged according to different points of view: technological, economical, scientific and so on.

As argued in (Halevy et al., 2009), the more data are available, the less we need for a deep understanding of the phenomenon under study. We do not need to construct a complex model nor to describe all its rules through complex logic-based languages. What we need is to properly tune statistical analysis or machine learning techniques over large corpus of data and more insights will arise from them, and very quickly. Recently, this new approach in taming the giant wave of available data is tempting several organizations and individuals due to its real effectiveness in knowledge discovery. By knowing people's preferences and opinions, for example, modern enterprises may gain a competitive advantage over competitors, while analyzing sensor data from the workshop may helps manufacturers to improve their processes and their performances thus reducing costs and increasing revenue. A study by the Economic Times suggests that large organizations using Big Data analytics outperform competitors, who do not utilize this (Bhanu, 2013). For all these reasons, modern enterprises look with great interest to the emerging solutions in the Big Data landscape and make significant investments in these technologies. Not surprisingly, the Big Data market is growing very quickly in response to the growing demand from enterprises. According to the International Data Corporation (IDC), the market for Big Data products and services was worth 3.2 billion of dollars in 2010, and they predict the market will grow to hit 16.9 billion of dollars by 2015. That's a 39.4 percent annual growth rate, which is seven times higher than the growth rate IDC expects for the IT market as a whole.

Interestingly, many of the best known Big Data tools available are open source projects. The very best known of these is Hadoop (Shvachko et al., 2010; White, 2009), which is spawning an entire industry of related services and products. Together with Hadoop, hundreds of Big Data solutions have emerged in the last years, from NoSQL or NewSQL databases to business intelligence and analytic tools, development tools and much more, and several surveys and state-of-the-art papers have become available in turn. In (Alnafoosi and Steinbach, 2013), the authors describe an integrated framework to evaluate and assist in selecting optimum storage solution for multi-variable requirements, while (Chen et al., 2014) reviews the background and state-of-the-art of Big Data introducing the general background and related technologies, such as cloud computing, Internet of Things, data centers, and Hadoop. Also different publicly available benchmarks exist in order to evaluate the performances of such new storage technologies. The Yahoo Cloud Serving Benchmark (Cooper et al., 2010), for example, includes a set of workload scenarios to measure different system performances, such as the read and update latency, the scan latency and the scale-out latency. Some of the tested databases are: Cassandra, HBase, MySQL, etc.

However, it is not enough to adopt a Big Data technology to be competitive. It is necessary to have a deep comprehension of the data to be collected and the context where the company operates, and a wise identification of the critical variables that affect the business processes. But an appropriate choice of the technology that meets the company requirements may surely help in being competitive and effective, avoiding waste of time and money.

Taking into account all the above considerations, this work goes in the direction of helping companies in selecting the right tool to use for managing large data sets, by characterizing at a general level the Big Data problem and its technological challenges and, then, by surveying the most popular and spread Big Data storage solutions existing in the literature.

The remainder of this paper is structured as follows. The second section presents the typical model characterizing the dimensions of Big Data. The third section introduces the evaluation framework adopted for the comparison of the Big Data store solutions, whereas the fourth section illustrates the results of the comparison carried out on the most widespread exist-

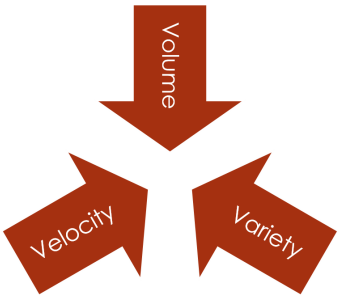

Figure 1: The Big Data Dimensions.

ing tools, based on the predefined criteria. Finally, the last section draws the conclusions, summarizing the major findings, and opens new directions for further researches in future works.

2 BIG DATA DIMENSIONS AND TECHNOLOGICAL SOLUTIONS

The concept of Big Data has different dimensions since the term Big not refer only to the quantity of data but also to the heterogeneity of data sources and to the velocity in analyzing data. A widely spread model to characterize Big Data is that of the 3Vs (Mohanty et al., 2013; Jagadish et al., 2014), which shows the three fundamental dimensions of Big Data: Volume, Velocity and Variety. Along the Volume axis, current scenarios involve technological solutions dealing with data in the order of pebibyte (2^50 bytes), exbibyte (2^60 bytes) or higher. Along the velocity dimension, it is possible to distinguish the following typology of analysis: offline analysis (without time constraints over responses), near real-time analysis (must guarantee response within tolerant time constraints), real-time analysis (must guarantee response within strict time constraints), hard-real time (must guarantee response within very strict time constraints) and streaming that refers to data stream mining (Gaber et al., 2005). Along the Variety axis, the following data formats can be mentioned: structured formats (e.g., relational DB data), semi-structured formats (XML grammars-based data, JSON-based, etc.), unstructured formats (data expressed in a no standard representation language), plain text and multiple format (which combines more data formats). Each dimensions in Figure 1 may have a greater or lesser weight than the others and in some cases may not exist at all, nevertheless we keep using the term Big Data. In addition to the dimensions previously described, some works in the literature provide other Vs: viscosity, variability, veracity and volatility (Desouza and Smith, 2014; Van Rijmenam, 2014). They measure respectively the resistance to flow of data, the unpredictable rate of flow and types, the biases, noise, abnormality, and reliability in datasets and finally how long data are available and if they should be stored. Each of the above dimensions makes traditional operations in data management more complicated. If the volume increases, for example, data storage becomes a challenge as well as data processing by means of analytics tools. Both storage systems and analytics algorithms must be scalable in this scenario. In addition, the variety dimension complicates data storage and analysis because of the integration of data with different structures. Figure 2 shows the three main operations of Big Data (storage, analytics and integration) and highlights the existing solutions for efficiently scaling Big Data dimensions. Data storage can be faced by means of modern scalable SQL (Structured Query Language)-based DBMSs (Data Base Management Systems) like Oracle Enterprise Edition or MySQL Cluster Carrier Grade Edition, in an increasing data volume scenario. When data variety increases, NoSQL (Not Only SQL) solutions such as HBase, MongoDB, Oracle NoSQL, are generally preferred respect to transactional DBMSs (Cattell, 2011). Moreover, NoSQL solutions can help in making efficient analytics over large and heterogeneous data sets. Contrary to traditional DBMSs, which guarantee efficient transactional processing but result too slow with large data sets analysis, NoSQL solutions provide fast analysis over high volume of data. Big Data analytics generally entails the adoption of programming models like Map-Reduce (Dean and Ghemawat, 2008) and their implementation like Hadoop (Shvachko et al., 2010) for processing large data sets by means of parallel-distributed algorithms on a computer cluster. The integration challenge can be faced by using traditional ETL (Extract, Transform, Load) methodology for large heterogeneous transactional databases or by adopting semantic web related technologies in order to integrate heterogeneous data models at a semantic level (Caldarola et al., 2015).

3 A FRAMEWORK FOR A QUALITATIVE EVALUATION OF BIG DATA SOLUTIONS

This section introduces the framework adopted for the qualitative evaluation of a pre-selected set of well-known Big Data tools. According to the previous section, from a technological point of view, Big Data

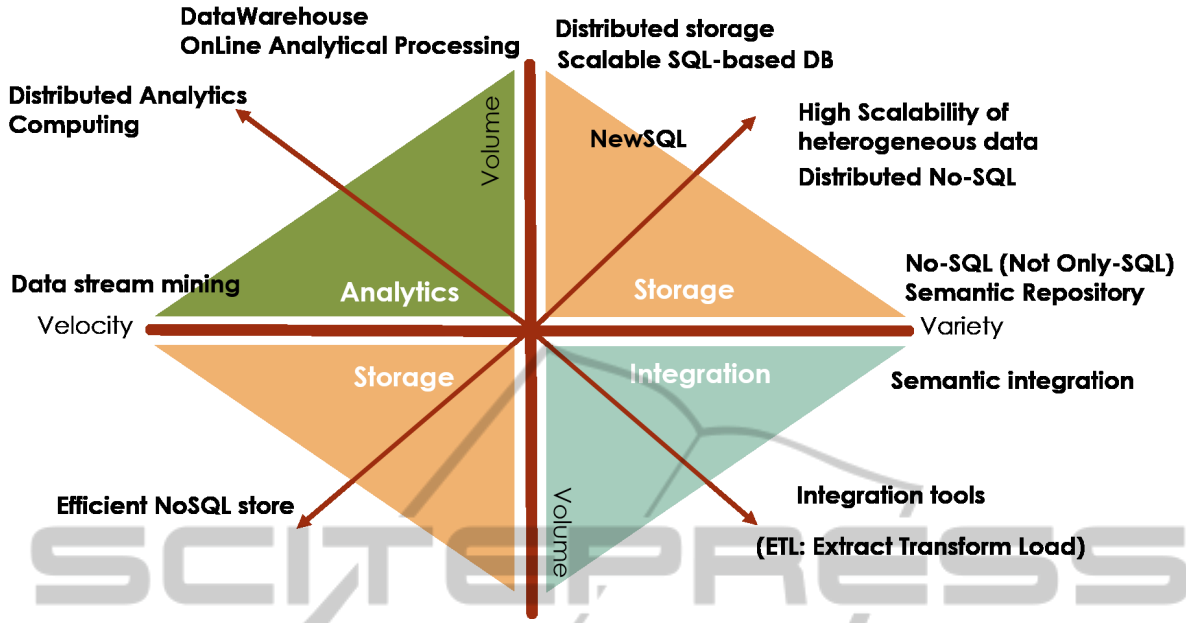

Figure 2: The Big Data Solutions.

involves solutions to different problems: data analysis, storage and integration. Furthermore, dealing with large data sets involves also other technical challenges deriving from the Big data life cycle phases, such as, managing the heterogeneity and the incompleteness of data, the vertical and horizontal scaling (scaling out and up), the timeliness, the privacy and, finally, the visualization and collaboration (Jagadish et al., 2014). For each of these challenges, a plethora of solutions and tools (with open and commercial licenses) have emerged in the last years. For example, the heterogeneity issue, due to the existence of heterogeneous information sources, has required sophisticated solutions to associate metadata to the collected data in order to correctly interpret the results. Semantically enriched metadata — formalized through ontologies using the semantic web languages — inside increasingly large collection of heterogeneous data have arisen the need for efficiently dealing with large ontologies, spawning new research fields, such as, knowledge representation and retrieval (Albanese et al., 2005; Rinaldi, 2008; Rinaldi, 2014), ontology matching and integration (Euzenat et al., 2007), partitioning (Amato et al., 2015b; Amato et al., 2015a), reuse (Modoni et al., 2015), versioning and maintenance (Flouris et al., 2006). A description of the existing solutions that deal with the other technological challenges of Big Data is out of the scope of this paper. Thus, the evaluation framework proposed in this work will focus on the store aspects of Big Data, defining some criteria to qualitatively analyze the set of preselected tools under study.

Starting from the premise that traditional DBMS solutions suffer from different pathologies with large amount of data, mostly in the modeling and analysis phase, a new trend of storage solutions have emerged in the last decade in the attempt to overcome the severe limitations of relational databases also by relaxing the ACID properties. In the following paragraph a critical comparison of the new NoSQL and NewSQL paaradigms with respect to the relational DBMSs will be proposed.

3.1 Relational DBMS and SQL Language

Relational Database Management Systems use relational tables and indexes to store and retrieve data. Some popular examples are Microsoft SQL Server, Oracle, PostgresQL, MySQL, etc. Among the benefits of using such traditional solutions are a well understood and consistent logical model that is independent from its implementation, i.e., an application than runs on MySQL can be made to run on Oracle without changing its basic assumptions. Furthermore such solutions guarantees the integrity of data over its entire life cycle and the ACID (Atomicity, Consistency, Isolation, Durability) set of properties that guarantee that database transactions are processed reliably. Other advantages are the comprehensive OLTP/transaction support, the strong OLAP/analysis tools, often built in (MS Analysis Services, Oracle OLAP). The downsides are the expensiveness of most solutions, it is not

easy to scale out, i.e., to have a lots of servers in a distributed solution and in anycase it is much expensive. Finally, the relational model does not come natural to developers, which results in translation overhead and common mistakes.

3.2 Not Only SQL

This kind of solutions use in-memory non-relational databases. These do not support the SQL language but more significantly do not support ACID or relationships between tables. Instead they are designed to query document data very quickly. Examples are: Hadoop, MongoDB, CouchDB, Riak, Redis, Cassandra, Neo4J, MemBase, HBase, etc. The most part of them are open source implementations or low cost in any case. Systems can scale out very easily, tables can be readily sharded/federated across servers. They use native well known to programmers objects, such as key-value arrays, maps, etc., so no translation to tables are required and also result very fast at finding records from massive datasets. On the other hand, the absence of a common model and the lack of ACID properties move up to the application many issues related to reliability. Transactions are at the row level only (if supported at all). Finally, they are poor at aggregation, because, where an RDMS solution would use SUM, AVG and GROUP BY, a NoSQL solution has map-reduce, which has to do the equivalent of a table-scan. NoSQL solutions are also poor at complex joins, although arguably this is something the data manager would design differently for.

3.3 New SQL

In-memory relational databases NewSQL is a class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing (read-write) workloads while still maintaining the ACID guarantees of a traditional single-node database system. These maintain ACID and relational integrity, but are in memory (like NoSQL) and readily scalable. They support SQL syntax. The most popular NewSQL systems attempt to distribute query fragments to different data nodes. These are designed to operate in a distributed cluster of shared-nothing nodes owning a subset of the data. SQL Queries are split into query fragments and sent to the nodes that own the data. These databases are able to scale linearly as additional nodes are added. Just to cite a few: Clustrix, VoltDB, etc.

3.4 The Survey of the Analyzed Solutions

Having described the main features characterizing the traditional and new storage paradigms, this subsection reports a survey of the most spread NoSQL and NewSQL solutions according to the evaluation criteria listed as follow:

1. *Category*. It represents the typology of the store solution. It may be RDBMS, NoSQL or NewSQL;
2. *Data structure*. It is related to the data structure used to memorize data, such as: column, document, key-value, graph, table.
3. *Operating system*. The operating system (e.g. Linux, Windows, Mac OS X) on which the Big Data store runs.
4. *License*. The information about the usage of the store. There are two main categories of licenses for these technologies: commercial and open source under various licenses (Apache License, GNU GPL, etc.).
5. *Query Languages*. The query language(s) supported;
6. *APIs availability*. The mechanisms provided to client application to access the storage.
7. *Latest Release version and Date*. When the latest release of the RDF store was released. If the date is not recent, the product may no longer be supported.

Taking into account the evaluation criteria above, table 1 shows the results of the survey. A brief description of each tool is also provided.

- **Apache Cassandra**[1]

  Cassandra is a NoSQL database now managed by the Apache Foundation. Cassandra's data model is column indexes based allowing log-structured updates, strong support for denormalization and materialized views, and powerful built-in caching. It's used by many organizations with large, active datasets, including Netflix, Twitter, Urban Airship, Constant Contact, Reddit, Cisco and Digg. Commercial support and services are available through third-party vendors. Operating System: OS Independent.

- **HBase**[2]

  Another Apache project, HBase is the non-relational data store for Hadoop. Features include

[1] http://cassandra.apache.org/
[2] http://hbase.apache.org/

Table 1: Evaluation synopsis of a set of technical characteristics.

Name	Category	Data structure	Operating System	License	Query Languages	API availability	Latest Release (Date)
Apache Cassandra	NoSQL	Column-based	OS Independent	Open	CQL (Cassandra Query Language)	C++, C#, Python, Java	2.1.5 (2015-04-29)
Apache HBase	NoSQL	Table, Map	OS Independent	Open	HBase Query Predicates	Java	0.94.27 (2015-03-26)
Google BigTable	NoSQL	Table, Map	OS Independent	Open		Java, Python	
MongoDB	NoSQL	Document-based	Linux, Windows, OS X	Open	Mongo DB Command line language	Many third party client library exist	3.0 (2015-03-03)
Neo4j	NoSQL	Graph	Windows, Linux	Community, Commercial	Cypher Query Language	Rest API	2.2.2 (2015-05-21)
Apache CouchDB	NoSQL	Document	Ubuntu, Windows, Mac OS X	Open	CouchDB primitives	HTTP API	1.6.1 (2015)
OrientDB	NoSQL	Graph		Open	SQL	.NET. Php, Ruby, Python	2.1 (2015-05-05)
Terrastore	NoSQL	Document	OS Independent	Open	Primitives via HTTP	HTTP API	0.8.2 (2015-09)
FlockDB	NoSQL	Graph	OS Independent	Open	Native Language	Ruby and Apache Thrift API	1.8.5 (2012-03-09)
Hibari	NoSQL	Key-Value	OS Independent	Open	OS Independent	Native Erlang, Universal Binary Format (UBF/EBF), Apache Thrift, Amazon S3, JSON-RPC	source code
Riak	NoSQL	Key-Value	Unix (several distros)	Open	CRUD Operations via HTTP requests	Java, Ruby, Python, C#, Node.js, PHP, Erlang, HTTP Api, Python, Perl, Clojure, Scala, Smalltalk, and many others.	2.1.1 (-)
Hypertable	NewSQL	Table	Linux, Mac OS X	Open	SQL-like	C++, Java, Node.js, Perl, PHP, Python, Ruby	0.9.8.7 (-)
StarDog	NoSQL	Graph	Independent	Different licenses	SPARQL	Java	3.0.2 (2015-05-12)
Apache Hive	NewSQL	Table	OS Independent	Open	HiveQL (SQL-like)	Java	1.1.0 (2015-03-08)
InfoBright Community Edition	NoSQL	Column-oriented	Windows, Linux	Community, Commercial	SQL-like	ODBC, JDBC, C API, PHP, Visual Basic, Ruby, Perl and Python	4.0.7 (-)
Infinispan	NoSQL	key-value	OS Independent	Open	Own query DSL	Java, Ruby, Python, C++, .NET (via Hot Rod Protocol)	7.2.1 (2015-04)
Redis	NoSQL	key-value	Linux	BSD	Redis commands	Many	3.0.1 (2015-05-05)
Clustrix	NewSQL	Table	Linux	Commercial	SQL	Windows	6.0
VoltDB	NewSQL	Table	OS Independent	Both	SQL	Java, JDBC, ODBC	5.2

linear and modular scalability, strictly consistent reads and writes, automatic failover support, Java API for client access, Thrift gateway and a RESTful Web service that supports XML and much more. Operating System: OS Independent.

- **Google BigTable**[3] BigTable (Chang et al., 2008) is a compressed, distributed data storage system built on Google File System, Chubby Lock Service, SSTable (log-structured storage like LevelDB) and a few other Google technologies. Recently, a public version of Bigtable was launched as Google Cloud Bigtable. BigTable also underlies Google Datastore , which is available as a part of the Google Cloud Platform.
- **MongoDB**[4]

  MongoDB was designed to support humongous databases. It is an open-source, document database designed for ease of development and scaling. It also has a full index support, replication and high availability. Commercial support is available through 10gen. Operating system: Windows, Linux, OS X, Solaris.

- **Neo4j**[5]

  Neo4j is a NoSQL, graph-based databases. It boasts performance improvements up to 1000x or more versus relational databases. Interested organizations can purchase advanced or enterprise versions from Neo Technology. Operating System: Windows, Linux.

- **Apache CouchDB**[6]

  Designed for the Web, CouchDB stores data in JSON documents that you can access via the Web or query using JavaScript. It offers distributed scaling with fault-tolerant storage. Operating system: Windows, Linux, OS X, Android.

- **OrientDB**[7]

  OrientDB is a 2nd Generation Distributed Graph Database. It can store 220,000 records per second on common hardware and can traverse parts of or entire trees and graphs of records in a few milliseconds. It combines the flexibility of document databases with the power of graph databases, while supporting features such as ACID transac-

[3]https://cloud.google.com/bigtable/docs/
[4]https://www.mongodb.org/
[5]http://neo4j.com/
[6]http://couchdb.apache.org/
[7]http://orientdb.com/orientdb/

tions, fast indexes, native and SQL queries, and JSON import and export. Operating system: OS Independent.

- **Terrastore**[8]

  Terrastore is a document store which provides advanced scalability and elasticity features without sacrificing consistency. It supports custom data partitioning, event processing, push-down predicates, range queries, map/reduce querying and processing and server-side update functions. Operating System: OS Independent.

- **FlockDB**[9]

  FlockDB is an open source distributed, fault-tolerant graph database for managing wide but shallow network graphs. It was initially used by Twitter to store social graphs (i.e., who is following whom and who is blocking whom). It offers horizontal scaling and very fast reads and writes. Operating System: OS Independent.

- **Hibari**[10]

  Hibari was originally written by Gemini Mobile Technologies to support mobile messaging and email services. It is a distributed, ordered key-value store with consistency guarantee. Hibari serves read and write requests in short and predictable latency, while batch and lock-less operations help to achieve high throughput ensuring data consistency and durability. It can store Peta Bytes of data by automatically distributing data across servers with a high fault tolerance by replicating data between servers. Operating System: OS Independent.

- **Riak**[11]

  Riak is an open source, key-value and distributed database. It also has a commercial license that add multi-Datacenter Replication, monitoring and 247 support. Many APIs in several programming languages are officially supported. Operating System: Linux, OS X.

- **Hypertable**[12]

  Hypertable is an open source, massively scalable database modeled after BigTable. Hypertable is similar to a relational database in that it represents data as tables of information, with rows and columns, but they can be thought of as massive tables of data, sorted by a single primary key, the row key. This NoSQL database offers efficiency and fast performance that result in cost savings versus similar databases. Operating System: Linux, OS X.

- **StarDog**[13]

  Stardog is a semantic graph databaseequally adept in client-server, middleware, and embedded modes. It supports the RDF graph data model, SPARQL query language, HTTP and SNARL protocols for remote access and control; OWL 2 and user-defined rules for inference and data analytics; and programmatic interaction via several languages and network interfaces. Operating System: OS Independent.

- **Hive**[14]

  The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL. Operating System: OS Independent.

- **InfoBright Community Edition**[15]

  Infobright Community Edition (ICE) is an open source database designed to deliver a scalable analytic database platform optimized for complex analytic queries on machine generated data. It is column-oriented and scales up to 50TB raw data and more than 30 concurrent queries. Operating System: Windows, Linux.

- **Infinispan**[16]

  Infinispan from JBoss describes itself as an ”extremely scalable, highly available data grid platform.” It is a key-value database, written in Java and designed for multi-core architecture. Operating System: OS Independent.

- **Redis**[17]

  Sponsored by VMware, Redis offers an in-memory key-value store that can be saved to disk for persistence. It supports many of the most popular programming languages. Operating System: Linux.

[8] https://code.google.com/p/terrastore/
[9] https://github.com/twitter/flockdb/
[10] http://hibari.github.io/hibari-doc/
[11] http://basho.com/riak/
[12] http://hypertable.org/
[13] http://stardog.com/
[14] https://hive.apache.org/
[15] https://www.infobright.com
[16] http://infinispan.org/
[17] http://redis.io/

- **Clustrix**[18]
  ClustrixDB is a distributed database. Latest version (6.0) brings many new capabilities and performance improvements, with specific optimizations for Magento-based and custom e-commerce implementations. It is able to specify that a copy of the table resides on every node, maximizing performance in certain workloads. It combines automatic data distribution to maximize parallelism, and the ability to override that for highly accessed, highly-joined tables (for example, metadata, etc.). Optimized Scheduler prioritizes critical OLTP transactions when heavy long-running analytics queries are also running.
- **VoltDB**[19] VoltDBs in-memory architecture is designed for performance. It eliminates the significant overhead of multi-threading and locking responsible for the poor performance of traditional RDBMSs that rely on disks. VoltDB was designed for High Availability from the ground up. VoltDB's supports virtualized and cloud infrastructures and combines the richness and flexibility of SQL for data interaction with a modern, distributed, fault-tolerant, cloud-deployable clustered architecture while maintaining the ACID guarantees of a traditional database system. VoltDB supports the JSON data type and several client access methods including stored procedures, JDBC and ad hoc queries. Furthermore, VoltDB supports a wide range of integrations including JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity) for data exchange. Operating System: OS Independent.

4 CONCLUSIONS

This work has provided a first evaluation of the most spread solutions existing in the Big Data landscape. As shown in the previous sections, a great number of solutions are open-source projects demonstrating the great interest that the community of developers has in such topics. At the same time, the work has highlighted the flexibility of the most part of tools that are generally multi-platform or programming language agnostic as they are provided with HTTP Restfull APIs which allow clients to easily access them. In other cases, the great availability of APIs written in the most popular programming languages (in most cases developed by third parties as depending or separate projects) contribute yet to ease the interoperability between the client tools and the back-end store database. Future works can be directed to different objectives. On the one hand, it can be improved the evaluation framework by adding other criteria not yet considered in this work, such as those related to security, scalability, and quantitative analysis performed by authoritative groups like YCSB lab. On the other hand, new but complementary study can be approached by surveying the technological solutions existing to deal with the other challenges of Big Data, such as: analytics, heterogeneity, timeliness, aggregation and transfer and finally visualization.

[18]http://www.clustrix.com/

[19]http://voltdb.com/

REFERENCES

Albanese, M., Capasso, P., Picariello, A., and Rinaldi, A. M. (2005). Information retrieval from the web: an interactive paradigm. In *Advances in Multimedia Information Systems*, pages 17–32. Springer.

Alnafoosi, A. B. and Steinbach, T. (2013). An integrated framework for evaluating big-data storage solutions-ida case study. In *Science and Information Conference (SAI), 2013*, pages 947–956. IEEE.

Amato, F., De Santo, A., Gargiulo, F., Moscato, V., Persia, F., Picariello, A., and Poccia, S. (2015a). Semindex: an index for supporting semantic retrieval of documents. In *Proceedings of the IEEE DESWeb ICDE 2015*.

Amato, F., De Santo, A., Moscato, V., Persia, F., Picariello, A., and Poccia, S. (2015b). Partitioning of ontologies driven by a structure-based approach. In *Semantic Computing (ICSC), 2015 IEEE International Conference on*, pages 320–323.

Bhanu, S. (2013). Companies adopting big data analytics to deal with challenges. *The Economic Times*.

Caldarola, E. G., Picariello, A., and Castelluccia, D. (2015). Modern enterprises in the bubble: Why big data matters. *ACM SIGSOFT Software Engineering Notes*, 40(1):1–4.

Caldarola, E. G., Sacco, M., and Terkaj, W. (2014). Big data: The current wave front of the tsunami. *ACS Applied Computer Science*, 10(4):7–18.

Cattell, R. (2011). Scalable sql and nosql data stores. *ACM SIGMOD Record*, 39(4):12–27.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):4.

Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209.

Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. (2010). Benchmarking cloud serving systems with ycsb. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pages 143–154, New York, NY, USA. ACM.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Desouza, K. C. and Smith, K. L. (2014). *Big data for social innovation*. Stanford Social Innovation Review.

Dragland, Å. (2013). Big data for better or worse. *ScienceDaily*.

Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.

Flouris, G., Plexousakis, D., and Antoniou, G. (2006). A classification of ontology change. In *SWAP*.

Franks, B. (2012). *Taming the big data tidal wave: Finding opportunities in huge data streams with advanced analytics*, volume 56. John Wiley & Sons.

Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S. (2005). Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12.

Hey, A. J., Tansley, S., Tolle, K. M., et al. (2009). *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft Research Redmond, WA.

Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM*, 52(8):36–44.

Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., and Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A. H., and Institute, M. G. (2011). Big data: The next frontier for innovation, competition, and productivity.

Merv, A. (2011). Big data. *Teradata Magazine Online*, Q1.

Modoni, G., Caldarola, E., Terkaj, W., and Sacco, M. (2015). The knowledge reuse in an industrial scenario: A case study. In *eKNOW 2015, The Seventh International Conference on Information, Process, and Knowledge Management*, pages 66–71.

Mohanty, S., Jagadeesh, M., and Srivatsa, H. (2013). *Big Data Imperatives: Enterprise Big DataWarehouse,BIImplementations and Analytics*. Apress.

Rinaldi, A. M. (2008). A content-based approach for document representation and retrieval. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 106–109. ACM.

Rinaldi, A. M. (2014). A multimedia ontology model based on linguistic properties and audio-visual features. *Information Sciences*, 277:234–246.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE.

Van Rijmenam, M. (2014). *Think Bigger: Developing a Successful Big Data Strategy for Your Business*. AMACOM Div American Mgmt Assn.

Weinberg, B. D., Davis, L., and Berger, P. D. (2013). Perspectives on big data. *Journal of Marketing Analytics*, 1(4):187–201.

White, T. (2009). *Hadoop: the definitive guide*. " O'Reilly Media, Inc.".