A Maturity Analysis of Big Data Technologies

Radu BONCEA, Ionuț PETRE , Dragoș-Marian SMADA, Alin ZAMFIROIU
ANAGRAMA
radu.boncea@anagrama.ro, ionut.petre@anagrama.ro, dragos.smada@anagrama.ro, alin.zamfiroiu@anagrama.ro

In recent years Big Data technologies have been developed at faster pace due to increase in demand from applications that generate and process vast amount of data. The Cloud Computing and the Internet of Things are the main drivers for developing enterprise solutions that support Business Intelligence which in turn, creates new opportunities and new business models. An enterprise can now collect data about its internal processes, process this data to gain new insights and business value and make better decisions. And this is the reason why Big Data is now seen as a vital component in any enterprise architecture. In this article the maturity of several Big Data technologies is put under analysis. For each technology there are several aspects considered, such as development status, market usage, licensing policies, availability for certifications, adoption, support for cloud computing and enterprise.
Keywords: *Big Data Technologies, Big Data maturity, Big Data comparative analysis*

1 Big Data Overview

Driven by the need to generate business value, the enterprise has started to adopt Big Data as a solution, migrating from the classical databases and data stores which lack the flexibility and are not optimized enough [1].

The changes in the environment make big data analytics attractive to all types of organizations, while the market conditions make it practical. The combination of simplified models for development, commoditization, a wider palette of data management tools, and low-cost utility computing has effectively lowered the barrier to entry. [2]. The concept addresses large volumes of complex data, rapid growing data sets that may come from different autonomous sources.

In recent approaches, Big Data is characterized by principles known as *the 4V – V*olume, *V*ariety, *V*elocity and *V*eracity [3]. There are opinions about accepting other principles as Big Data characteristics, such as *V*alue.

Each day more businesses realize that Big Data is relevant as the applications generate large volumes of data generated automatically, from different data sources, centralized or autonomous. As traditional databases hit limitations when the need of analyzing this data, dedicated solutions must be considered.

Important BigData Solutions:

- **Apache HBase/Hadoop** is based on Google's BigTable distributed storage system, which runs on top of Hadoop as a distributed and scalable big data store. This means that HBase can leverage the distributed processing paradigm of the Hadoop Distributed File System (HDFS) and benefit from Hadoop's MapReduce programming model. It combines the scalability of Hadoop with real-time data access as a key/value store and deep analytic capabilities of Map Reduce [4].

HBase allows to query for individual records as well as derive aggregate analytic reports across a massive amount of data. It can host large tables with billions of rows, millions of columns and run across a cluster of commodity hardware. HBase is composed of three types of servers in a master slave type of architecture. Region servers are responsible to serve data for reads and writes. When accessing data, clients communicate with HBase Region Servers directly. Region assignment, DDL (create, delete tables) operations are handled by the HBase Master process.

DOI: 10.12948/issn14531305/21.1.2017.05

- **Apache Cassandra** is a distributed database used for the administration and management of large amounts of structured data across multiple servers, while providing highly available service and no single point of failure. It provides features such as continuous availability, linear scale performance, data distribution across multiple data centers and cloud availability zones. Cassandra inherits its data architecture from Google's BigTable and it borrows its distribution mechanisms from Amazon's Dynamo.

The nodes in a Cassandra cluster are completely symmetrical, all having identical responsibilities. Cassandra also employs consistent hashing to partition and replicate data. it has the capability of handling large amounts of data and thousands of concurrent users or operations per second across multiple data centers.

Cassandra has a hierarchy of caching mechanisms and carefully orchestrated disk I/O which ensures speed and data safety. Write operations are sent first to a persistent commit log (ensuring a durable write), then to a write-back cache called a memTable; when the memTable fills, it is flushed to a sorted string table – SSTable - on disk. A Cassandra cluster is organized as a ring, and it uses a partitioning strategy to distribute data evenly.

- **Redis** represents an in-memory data structure store used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperlogs and geospatial indexes with radius queries. Redis stores all data in RAM, allowing lightning fast reads and writes. It runs extremely efficiently in memory and handles high-velocity data, needing simple standard servers to deliver millions of operations per second with sub-millisecond latency.

Redis is schema-less, but when one of its data structures (like HASH or Sorted Sets) is used, users can take advantage of the in-memory operations to accelerate the way data is processed. The Sorted Set is a structure that combines the features of a hash table with those of a sorted tree. Each entry in a Sorted Set is a combination of a string "member" and a double "score". The member acts as a key in the hash, with the score acting as the sorted value in the tree. With this combination, it can be accessed by members and scores directly by member or score value.

- **VoltDB** is a distributed SQL database intended for high-throughput transactional workloads on datasets which fit entirely in memory and where the data is automatically partitioned based on a column specified by the application developer. All data is stored in RAM memory. Disk snapshots are periodically used to backup data and provided on-disk recovery log for crash durability. Data is replicated to at least *n+1* nodes to tolerate *n* failures. Tables may be replicated to every node for fast local reads, or sharded for linear storage scalability. VoltDB determines where each record goes without the need of user's specification.

All data operations in VoltDB are single-threaded, running each operation to completion before starting the next. VoltDB is partitioning both the data and the work. For best performance the database tables and associated queries need to be partitioned so that the most common transactions can be run in parallel. Since each site operates independently, each transaction can be performed without the overhead of locking individual records that usually consumes processing time of traditional databases. VoltDB balances the requirements of maximum performance with the flexibility to accommodate less intense but equally important queries that cross partitions [5].

- **MongoDB** is an open-source document database that provides high performance, high availability, and automatic scaling. In order to provide these features, MongoDB has a document-oriented data model that permits it to split up data through multiple servers, perform data balancing, load across a cluster, re-distribute documents automatically and perform routing of user requests. In case there is a need for more capacity, MongoDB allows the addition of new machines and can

DOI: 10.12948/issn14531305/21.1.2017.05

automatically manage data to the new machines. MongoDB doesn't require stored procedures and the model and stored data have the same structure – BSON, which is similar to JSON.

MongoDB lacks a series of features that are usually common amongst the traditional relational databases. The most notable one is the lack of notably joins and complex multi-row transactions; however the decision of not implementing this feature has led to a greater scalability of MongoDB. The technology is widely used at this moment and proved in the recent years to be a fast and easy to use solution to handle Big Data.

2 The Maturity Model for BigData Solutions

Big Data management systems receive data from various sources, gathering a huge volume of data often represented by heterogeneous and diverse dimensionalities [3]. Different information collectors prefer their own schemata or protocols for data recording, and the nature of different applications also result in diverse data representations. Autonomous data sources with distributed and decentralized controls are a main characteristic of Big Data applications [6]. As a result, an analysis on some of the most popular Big Data technologies is aiming at supporting companies in selecting the proper solution for their needs.

In an adaptive organization, measurement and analysis can be valuable tools for understanding the present environment and evaluating the effectiveness of our actions. Advances in Internet and mobile technologies have dramatically expanded the scope and rate at which certain types of information can be collected. [7]

This paper proposes to analyze the maturity of some of the most used Big Data technologies. In order to achieve its goal, the analysis is performed considering the following model:

Technical Maturity

- *Development status* – describes the current developments status of the solution. The frequency in releasing major and minor releases and the frequency of the patches are noted; as well as other indicators such the number of active contributors/programmers and the average time for solving critical and blocking issues are also worth mentioning;
- *Support libraries* that can be used to integrate the solution using most popular programming languages;
- *Guidelines and documentation* – should answer the question if there is sufficient documentation for installation and administration;
- *Cross-platform support* – check if the solution can be deployed on popular operating systems (Unix/Linux/Windows). Since Internet of Things gained momentum, an important question to answer is whether the solution can be deployed on resource constrained environments such as Raspberry Pi;
- *Professional Certification* – which are the certification programs that give individuals or businesses valid certifications in the respective technology.

Business Maturity

- *Market usage* – measures the adoption by the enterprise;
- *Cloud computing support* – the Big Data solution is offered as a cloud service by largest cloud computing providers;
- *Licensing policy* – analyzes the conditions for use – free or commercial use;
- *Enterprise support* – does the solution offer enterprise support, and under which terms.

3 Technical Maturity of Big Data Solutions

This chapter presents an analysis regarding the software versions, minor and major releases, solving time of issues, number of contributors in technology development, support libraries for building client interfaces, documentation available to users and the offer of trainings and certifications.

DOI: 10.12948/issn14531305/21.1.2017.05

3.1 Development Status

Definitions of terms as considered in this paper:

Major versions supported – Represent the major releases of the software that are currently supported by the producers.

Minor versions – Represent the minor releases of the software, versions that are released as improvements to a major versions. Only stable releases are considered.

Active contributors – The number of contributors to the development of the software.

Average time for resolving critical issues (in days/issue) – Represents the average period of time needed by the developers to solve a critical issue reported. It is calculated as the average solving time for the latest 20 known reported issues. We used **Atlassian JIRA** to calculate the solving time for MongoDB, Hadoop, Cassandra and VoltDB and **GitHub** for Redis. In the case of Redis we considered 10 critical bugs, as other reported bugs were before than the year 2012.

In case of exceptions time (more than 5 times than average time) we considered to be exceptional situations and did not consider in the calculus.

Table 1. Development status of BigData solutions

	MongoDB	**Hbase/Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Number of major versions supported	2	2	2	2	2
Number of minor versions released in the last 12 months	14	2	23	8	9
Number of minor versions released in the last 24 months	22	3	52	33	17
Number of active contributors	280	104/87	200	65	220
Average time in days for resolving critical issues[days/issue]	2.4	7	6.3	27.5	18

3.2 Support Libraries

This chapter describes the support provided for developing client interfaces in order to access the solutions. A client interface allows programs that are written in various programming languages to interact with the database by using native functions of the language. The client interface handles the requests and translates them into a standardized communication protocol with the database.

MongoDB uses the term *driver* when it comes to a client library that manages the interaction with the database in a language that is also appropriate for the respective application. Beside the official supported libraries, there are also two Community Supported drivers for *Go* and *Erlang*.

Hadoop has native implementations of certain components for performance reasons and for non-availability of Java implementations. These components are available in a single, dynamically-linked native library called the native Hadoop library. The main components are HDFS, Map Reduce, Yarn. It offers web service REST APIs which is a set of URI resources that give access to the cluster, nodes and applications (MapReduce Application Master and MapReduce History Server REST API's).

Cassandra

The recommended practice for the projects is to use CQL – Cassandra Query Language. However there are client libraries that are supported by Cassandra.

VoltDB uses Java for stored procedures and for the primary client interface. Nevertheless

DOI: 10.12948/issn14531305/21.1.2017.05

client interfaces can be written in a large variety of programming languages. A part of the client interfaces are developed and packaged inside the distributed kit. There are also client interfaces which are compiled and distributed in the form of a separated kit.
Available client interfaces: C#, C++, Erlang, Go, Java (packaged with VoltDB), JDBC (packaged with VoltDB), JSON (packaged with VoltDB), Node.js, ODBC, PHP, Python

Redis

There are many client libraries supported by Redis. For certain languages, there are clients that are recommended by Redis for use. New clients can be developed added in the Redis repository. Redis documentation provides instructions on how to create a client.

Table 2. Support libraries for BigData solutions

	MongoDB	**Hbase/ Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Access methods	Proprietary protocol on BSON/JSON	Java API, RESTful HTTP API, Apache Thrift	CQL(Cassandra Query Language), Apache Thrift	Java API RESTful HTTP/JSON API JDBC	RESP - REdis Serialization Protocol
Client libraries	C, C++, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala, Casbah (Scala Driver). Go, Erlang	*Java* : Apache Yarn Client; *Python* - Hadoop YARN API, Snakebite provides a pure python HDFS client. *PHP* - PHP-Hadoop-HDFS is a wrapper for WebHDFS and CLI hadoop fs. *Ruby* – webhdfs a client for Hadoop WebHDFS	*Java* - Achilles, Astyanax, Casser, Datastax, Java driver, Kundera PlayORM *Python* -Datastax Python driver *Ruby* - Datastax Ruby driver, *C#/.NET* - Cassandra Sharp, Datastax C3 driver, Fluent Cassandra *NodeJS* - Datastax Nodejs driver, Node-Cassandra-CQL *PHP* - CQL | PHP, Datastax PHP driver, PHP-Cassandra, PHP Library for Cassandra *C++* - Datastax C++ driver, libQTCassandra *Scala* - Datastax Spark connector, Phantom, Quill *Clojure* - Alia, Cassaforte, Hayt *Erlang* - CQerl, Erlcass *Go* – CQLc, Gocassa, GoCQL Haskell – Cassy *Rust* - RustCQL	C#, C++, Erlang, Go, Java, JDBC, JSON, Node.js, ODBC, PHP, Python	ActionScript, Bash, C, C#, C++, Clojure Common Lisp, Crystal, D, Dart, Delphi, Elixir, emacs lisp, Erlang, Fancy, gawk, GNU Prolog, Go Haskell, Haxe, Io, Java, Julia, Lasso Lua, Matlab, mruby, Nim, Node.js, Objective-C OCaml, Pascal, Perl, PHP, PL/SQL, Pure Data Python, R, Racket, Rebol, Ruby, Rust Scala, Scheme, Smalltalk, Swift, Tcl, VB, VCL

3.3 Guidelines and Documentation

An analysis was performed on the documentation provided by each solution regarding the installation, administration and other features.

MongoDB – provides manuals for both installation and administration of solution. The installation guidelines are given independently for Windows, Linux and OS X, both for MongoDB Community Edition and Enterprise. There are resources available on installation using MongoDB Cloud Manager. The administration documentation addresses the ongoing operation and maintenance of MongoDB instances and deployments. It includes both high level overviews as well as tutorials that cover specific procedures and processes for operating MongoDB. [8]

DOI: 10.12948/issn14531305/21.1.2017.05

Developers can find detailed instructions on the configuration, maintenance, upgrading, monitoring or backup.
A guide providing instructions on how to get started with MongoDB fast is available in editions for mongo Shell, Python, Java, Ruby, NodeJS, C++, C#.
VoltDB – The solution comes as either pre-built distributions or as source code. The installation documentation specifies system requirements for running VoltDB, installation and upgrading, description of the resources included in the distribution kit.
The administration guidelines are detailed in providing information on how to properly design, develop and run an application On VoltDB. The manual addresses the vast majority of instructions required to properly administer the database.
There is documentation available regarding best practices in the area, Getting started tutorial, Performance and Customization, Java API or Client Wire protocol.
Hadoop/Hbase: - provides documentation about last release notes, about native libraries and security mode. It provides also documentation for the architectures, commands and tools of main components (HDFS, Map Reduce, Yarn). Also are detailed guidelines and documentation for integration with other systems, for authentication methods and other useful tools.

Table 3. Guidelines and documentation of BigData solutions

	MongoDB	**Hbase/Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Platforms	*Windows* - Server 2008R2 and later, Vista and later *OS X* -v 10.7 and later *Linux* - Amazon Linux 2013.03 and late, Debian 7 & 8, RHEL/CentOS 6.2 and later, SLES 11 & 12, Solaris 11 64-bit, Ubuntu 12.04 & 14.04 &16.04	Cross-platform	Cross-platform	Supports a variety of platforms, but the following are recommended in production: • Amazon Linux • Debian 7.1 • RHEL / CentOS 6.2+ • SLES 11+ • Ubuntu LTS 12.04 • Ubuntu LTS 14.04 • Windows Server 2012 & 2012 R2	Ubuntu 14.04, 16.04, 64 bit; CentOS / RHEL 6.5, 6.6, 6.7 64 bit; CentOS / RHEL 7.0, 7.1, 7.2, 64 bit; Oracle Linux 6.5, 64 bit.
Other Dependencies		JDK v8 (Java Development Kit)	The latest version of Java 8, either the Oracle Java Standard Edition 8 or OpenJDK 8. For using cqlsh, the latest version of Python 2.7.		-
IoT platform support (Raspberry Pi, Arduin)	Yes	Yes	Yes	No	Yes

DOI: 10.12948/issn14531305/21.1.2017.05

Cassandra - provides documentation divided in several sections such as Installation, Operation, Data Modeling, Cassandra Tools, CQL or Configuration. As a drawback for unexperienced user, the documentation is currently a work-in-progress and there are a number of sections which lack the necessary instructions.

The installation manual contains instructions regarding the correct install method, the prerequisites required to run and instructions for configuring a cluster of nodes. The administration manual still has some chapters that are in progress and do not contain information, such as *Repair*, *Read Repair* or *Hints*.

Redis – when it comes to installation it provides a Quick Start manual, with instructions on how to get it running properly. The complete Redis documentation is vast and divided in chapters such as Programming, Tutorials & FAQ, Administration, Troubleshooting, Redis Cluster.

Also a list containing all the commands implemented by Redis and documentation about each of them has proven to be a very useful document for developers.

3.4 Professional Certification

MongoDB – provides **Professional Certification Program** – includes online classes and offers both public and private training. Registered attendees can take a Certification Exam. The certifications offered are:

- *Developer Associate*: adequate to those with knowledge on the fundamentals of application design and build using MongoDB, software engineers with a good knowledge of the fundamentals and with professional experience in app development.
- *DBA Associate*: adequate to administrators that understand MongoDB concepts and mechanics, operations professionals that master the theoretical fundaments and already have the experience in managing MongoDB.

Hbase/Hadoop – There are several certifications partners of Apache, the most important being Cloudera [9] and MapR Academy [10].

Cloudera offers three certification levels - CCA Spark and Hadoop Developer, CCA Data Analyst, Cloudera Certified Administrator for Apache Hadoop (CCAH). MapR offers courses on-demand or instructor-led.

Table 4. Certifications for BigData solutions

	MongoDB	**Hbase/Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Certifications	Own certification system	Provided in partnership – Cloudera, MapR Certification Practice Questions (MCHBD	Provided in partnership – DataStax which offers DataStax Enterprise, production certified Apache Cassandra, with 24 x 7 x 365 support	Own certification system	Does not provide certifications

Cassandra – Training and certifications are offered by **DataStax**. There are also partnerships between DataStax and third-parties, such as **O'Reilly**, to provide certifications.

- *Self-pace course* – beginner courses offered on-line. It addresses the basic challenges encountered when it comes to scaling relational databases, introduction to Cassandra fundamentals - consistency, replication, anti-entropy operations, data modeling and the use of DtaStax

DOI: 10.12948/issn14531305/21.1.2017.05

Enterprise.
- *Instructor-led training* – this is considered to be an expert-level training, performed on-site, and delivers a practical approach. The course also focuses on team-work based development. [11]

VoltDB – offers courses and certification. VoltDB University Certification is designed to provide training and certifications for partner organizations, customers or individuals. Basic lessons and presentations are available on-line.

Redis – at this moment does not provide an official certification program. There are trainings offered by third-parties, but without the possibility of obtaining an official certification from Redis.

4 Bussiness Maturity of Big Data Solutions

4.1 Market Usage

In this chapter the market adoption of Big Data technologies is highlighted, along with cases of successful developments of Big Data analytics systems.

MongoDB is widely used and proved in the recent years to be a fast and easy to deploy solution to handle Big Data. It is vastly used in major log systems, mobile applications, content-centric applications, Cloud applications. Some well-known implementations are [12]:
- *LinkedIn* used MongoDB's intuitive schema design and quick document search to develop LearnIn, an internal learning platform.
- *Forbes* now offers an end-to-end publishing platform built on MongoDB as a turnkey solution to other publishers, both as SaaS and an on-premise solution, driving new revenue and expanding business.
- *Gov.uk* – the United Kingdom government website uses MongoDB to power its content API, providing data storage in the cloud. MongoDB was used for its scaling and data processing capabilities.
- *Royal Bank of Scotland* - MongoDB supports the global bank's enterprise data service which is underpinning several core trading systems.

Hbase/Hadoop
- *Facebook* uses HBase for its messaging platform;
- *Spotify* uses HBase as base for Hadoop and machine learning jobs;
- *Sophos* uses for some of their back-end systems

Cassandra
- *CERN* used Cassandra-based prototype for its ATLAS experiment to archive the online DAQ system's monitoring information;
- *Facebook* used Cassandra to power Inbox Search, with over 200 nodes deployed;
- *IBM* has done research in building a scalable email system based on Cassandra;
- *Discord* uses Cassandra to store over 120 million messages per day;
- *Wikimedia* uses Cassandra as backend storage for its public-facing REST Content API

VoltDB has a wide area of application including high-velocity applications that thrive on fast streaming data, such as telecom policy and billing applications, sensor applications like smart grid power systems, real-time digital advertising platforms, analytics for online gaming, and applications for risk and fraud detection. VoltDB is successfully used by large companies [13]:
- *Nokia Networks* - has deployed VoltDB to provide real-time data solutions that enhance mobile subscriber services through Nokia Open Telecom Application Server.
- *MAXCDN* - uses VoltDB to provide real-time analytics on their system.
- *AsiaInfo* - a leading supplier of IT solutions and services chose VoltDB as a key component of its Big Data analytics product – Veris C3 - to perform real-time analytics and decision-making on fast-moving data. It uses a real-time data feed to deliver up-to-date information in order to optimize customers' experience of mobile marketing campaigns.

Redis is adequate for highly scalable data store shared by multiple processes, multiple applications, or multiple servers.

DOI: 10.12948/issn14531305/21.1.2017.05

Communications cross-platform, cross-server, or cross-application makes it a pretty great choice for many use cases.

- *GitHub* uses Redis as a persistent key/value store for the routing information and a variety of other data;
- *Digg* just rolled out a new feature, cumulative page event counters (page views plus clicks), that is using Redis as its underlying solution;
- *StackOverflow* uses Redis as a caching layer for the entire network.

4.2 Cloud Computing Support

In this chapter we analyze the availability of the given Big Data solution on four major cloud providers.

Cloud computing provides a reliable, fault-tolerant, available and scalable environment to harbour big data distributed management systems. Storing and processing big volumes of data requires scalability, fault tolerance and availability. Cloud computing delivers all these through hardware virtualization. Thus, big data and cloud computing are two compatible concepts as cloud enables big data to be available, scalable and fault tolerant. [14]

The list of the major cloud computing vendors is based on the ranking from **Synergy Research Group**. The market share of the four cloud providers is estimated at about 63% of the total market, with Amazon being in the lead with 40% of the total market [15].

Table 5. Cloud support for BigData solutions

	MongoDB	**Hbase/Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Amazon Web Services	x	x	x	x	x
Microsoft	x	x	x	x	x
IBM	x	x	x	x	x
Google	x	x	x	x	x

Cloud computing is another paradigm which promises theoretically unlimited on-demand services to its users. The virtualization of resources allows abstracting hardware, requiring little interaction with cloud service providers and enabling users to access terabytes of storage, high processing power, and high availability in a pay-as-you-go model [16]. While all the technologies that are analyzed in this paper are offered by the major Cloud providers, the deployment methods are different.

4.3 Licensing Policy

The technologies presented in this paper are distributed under the following licenses:

The *Apache License* (ASL) is a permissive free software license written by the Apache Software Foundation (ASF). The license allows the user of the software the freedom to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the license. Apache License 2.0 was released in January 2004 and includes easier usage for non-ASF projects, improved compatibility with GPL-based software, allow the license to be included by reference instead of listed in every file.

The *BSD 3-clause* license allows user almost unlimited freedom and redistribution for any purpose with the software as long as its copyright notices and the license's disclaimers of warranty are maintained.

The *GNU Affero General Public License* is a free, copyleft license for software specifically designed to ensure cooperation with the community in the case of network server software. Version 3 allow the transfer of a work formed by linking code licensed under the one license against code licensed under the other license, despite the licenses otherwise not allowing relicensing under the terms of each other.

The *GNU General Public License* is a widely used free software license, which guarantees end users the freedom to run, study, share and modify the software. Version 3 brings

DOI: 10.12948/issn14531305/21.1.2017.05

changes in relation to software patents, free software license compatibility, the definition of "source code", and hardware restrictions on software modification.

Table 6. Licensing policy for BigData solutions

	MongoDB	**Hbase/Hadoop**	**Cassandra**	**VoltDB**	**Redis**
Open-source Licensing	Apache License 2	Apache License 2	BSD 3 clause	AGPL version 3	GPL version 3

4.4 Enterprise support

Cassandra is distributed under the The Apache License 2 and commercial distributions are provided by DataStax.

DataStax Enterprise contains the only production-certified version of Cassandra on the market along with other enterprise components like integrated analytics, search, multi-model functionality (including graph) and much more.

Impetus Technologies is a thought leader in Big Data working exclusively for ISVs and large enterprises. Impetus offers services around Cassandra and has supported multiple organizations in rolling out Cassandra based solutions.

Instaclustr provides managed Apache Cassandra hosting on Amazon Web Services, Google Cloud Platform, Microsoft Azure, and SoftLayer. Instaclustr also provides expert-level consultancy. [17]

Redis is open source software released under the terms of the three clause BSD license. The licensing model is subscription based. Redis Cloud is priced according to data capacity (both fixed and pay-as-you-go plans are available), whereas RLEC is licensed according to the number of shards (Redis processes) in a cluster. Redis Cloud offers a free-for-life tier that's limited to a single database of up to 30MB. RLEC is available as a free download that's only soft-limited by the number of shards.

A number of companies provide products which include **Apache Hadoop**, commercial support, and/or tools and utilities related to Hadoop:

- *Amazon* offers a version of Apache Hadoop on their EC2 infrastructure, sold as Amazon Elastic MapReduce;
- *Apache Bigtop* is a project for the development of packaging and tests of the Apache Hadoop ecosystem;
- *Cloudera* offers its enterprise customers a group of products and services that complement the open-source Apache Hadoop platform;
- *DataStax* provides a product which fully integrates Apache Hadoop with Apache Cassandra and Apache Solr in its DataStax Enterprise platform;
- *IBM InfoSphere BigInsights* brings the power of Apache Hadoop to the enterprise;
- *Emblocsoft* delivers enterprise Hadoop edition based on Apache Hadoop to meet the demand of enterprise data processing:
  - Big Data data visualization and advanced analytics;
  - Real time stream processing;
  - Machine learning at scale;
  - Enterprise integration.
- *DataTorrent* is certified on Apache Hadoop, and all leading distributions. The DataTorrent platform includes built-in fault tolerance and auto-scaling and can process billions of events/second. [18]

VoltDB is available in both open source and enterprise edition. The open source, or community edition, provides basic database functionality with all the transactional performance benefits of VoltDB. The enterprise edition provides additional features needed to support production environments, such as high availability, durability, and export integrations for transactionality or dynamic scaling. The scaling is done automatically, as opposed to Community edition where the scaling must be performed manually. The Enterprise edition gives the benefit of unlimited customer support. [19]

DOI: 10.12948/issn14531305/21.1.2017.05

5 Conclusions

In a world where everything tends to be connected, Big Data conquered the spotlight when it comes to innovation and competition, providing both challenges and opportunities in the IT global landscape. Collecting and analyzing huge volumes of data can empower business decisions and create competitive advantages for those who choose the appropriate Big Data solutions.

The vast majority of large companies have already adopted Big Data technologies to perform analytics. The present paper analyses five major technologies, widely adopted in the market, from two perspectives – technical maturity and business maturity.

The selection process of the appropriate technology stack can be difficult, but considering there is an entire class of performant solutions it lowers the risk of taking a bad decision. The solution should map best to the particular objectives of the company and solve core issues, which is more complex than the regular price/performance evaluation. Other important consideration should be the personnel knowledge level that will dictate what type of support is required.

This paper provides a perspective on Big Data solutions based on the customer needs; basically, depending on the customer goals the appropriate Big Data solution can be adopted. This involves a match of the business requirements to the advantages/disadvantages of each solution. It is evidently unwise and not recommended to commit to a Big Data technology without first assessing its potential value and characteristics.

Acknowledgment

This work was been carried out as part of the project ID_34_462, SMIS 2014+ from Programul Operațional Competitivitate 2014-2020 Axa prioritară 1 – Crearea de laboratoare privind cercetarea datelor de mari dimensiuni în vederea dezvoltării unor produse inovative și a unor aplicații în domeniul internetul viitorului (Creating research laboratories for BigData in order to develop innovative products and applications in the field of Future Internet).

References

[1] H. Hashem, D. Ranc, Pre-Processing And Modeling Tools For BigData, Foundations Of Computing And Decision Sciences, Vol. 41, No. 3, 2016, ISSN 0867-6356

[2] D. Loshin, Big Data Analytics, 2013, ISBN: 978-0-12-417319-4

[3] J. Zhang, M. L. Huang, Z.P. Meng, Visual Analytics for BigData Variety and Its Behaviours, Computer Science and Information Systems 12(4):1171‑1191

[4] I. Lahmer, N. Zhang, Towards a Virtual Domain Based Authentication on MapReduce, IEEE Access, vol. 4, 1658 – 1675, 2016.

[5] VoltDB documentation, https://docs.voltdb.com/

[6] X. Wu et al., 2014. Data mining with big data. IEEE Transactions on Knowledge and Data Engineering

[7] Big Data Now: 2014 Edition, Current Perspectives from O'Reilly Media, 2015, ISBN: 978-1-491-91736-7

[8] MongoDB Manual, doc.mongodb.com/manual/administration

[9] Cloudera certification, https://www.cloudera.com/more/training/certification.html

[10] MapR certification, www.mapr.com/services/mapr-academy/MapR-Certified-HBase-Developer

[11] Datastax courses, https://academy.datastax.com/courses

[12] Who uses mongodb, www.mongodb.com/who-uses-mongodb

[13] VoltDB partners, www.voltdb.com/partners

[14] P. Caldeira-Neves, B. Schmerl, J. Bernardino, J. Cámara1, Big Data in Cloud Computing: features and issues, International Conference on Internet of Things and Big Data , 2016

[15] Synergy Research group, https://www.srgresearch.com/articles/microsoft-google-and-ibm-charge-public-cloud-expense-smaller-providers

[16] J. González-Martínez et al., 2015. Cloud computing and education: A state-of-the-

DOI: 10.12948/issn14531305/21.1.2017.05

art survey. Computers & Education, 80, pp.132–151
[17] Third party support for Cassandra, https://wiki.apache.org/cassandra/ThirdPartySupport
[18] Hadoop support, https://wiki.apache.org/hadoop/Distributions%20and%20Commercial%20Support
[19] VoltDB editions, https://www.voltdb.com/edition

Radu BONCEA is a Researcher at *I.C.I. Bucharest* and Project Manager at *Anagrama*. He has been involved in several large European projects such SPOCS, eSENS, Cloud for Europe and The Once Only Principle. As a Ph.D. student at Electronics, Telecommunications and Information Technology, he's interested in IoT and Cloud Computing related technologies.

Ionuț PETRE graduated the Faculty of Electronics, Telecommunications and Information Technology in 2005. He is a PhD student at University Lucian Blaga from Sibiu, Faculty of Management. Currently he works as Researcher at *I.C.I Bucharest* and *Anagrama*. His main areas of interest are Internet of Things, e-Government, Big Data, software engineering, digital libraries, transport research. He is involved in research projects specific to the Information Society. His research was published in journal articles and proceedings of conferences.

Dragoș SMADA graduated the Faculty of Electronics, Telecommunications and Information Technology in 2005. In 2012 he graduated the Documents Information Management Master program organized by the University of Bucharest. Currently he works as a Senior Researcher at *I.C.I Bucharest* and *Anagrama*. His main areas of interest are Big Data, Internet of Things, software engineering, information security, software architecture. He participated in both national and international research projects in the IT&C. He published as author and co-author of journal articles and scientific presentations at conferences.

Alin ZAMFIROIU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2009. In 2011 he has graduated the Economic Informatics Master program organized by the Bucharest University of Economic Studies and in 2014 he finished his PhD research in Economic Informatics at the Bucharest University of Economic Studies. Currently he works like a Senior Researcher at *I.C.I Bucharest* and *Anagrama*. He has published as author and co-author of journal articles and scientific presentations at conferences.

DOI: 10.12948/issn14531305/21.1.2017.05