Big Data: Technologies and Software Products

Adriana BĂNUŢĂ, Cătălina DRAGOMIR, Mădălina GHEȚU
Bucharest University of Economic Studies, Bucharest
adrianabnta@gmail.com, catalina93.d@gmail.com, madalina.ghetu@gmail.com

The main tendency in technology leans towards huge amounts of data to be stored, analyzed and processed, in order to obtain valuable information on various topics. Regardless the domain of interest, storing data is always a must and it must be done in an efficient, secure and accessible way. Then, it can be used for statistics, studies or as training sets in the field of machine learning. The aim of this paper is to give a brief overview of the concept known as big data, as well as to present and compare the main technologies and software products used to store and manipulate this type of data.
Keywords: *Big Data, Hadoop, NoSQL, Traditional Database*

1 Introduction

Big data is a broad term, generally referring to data sets that cannot be processed in a more "traditional" manner (i.e. using relational databases), due to the voluminous data they have to store and process.

Storing capacities have evolved at a fast pace, as the technological context imposed that, needing more and more memory to save and process data. Going into the digital era, storage of data became a vital need of every organization and more advanced ways had to be found, for keeping up with both companies and people's needs. These needs include storing data of millions of users, or storing historical data to be analyzed for statistics and predictions.

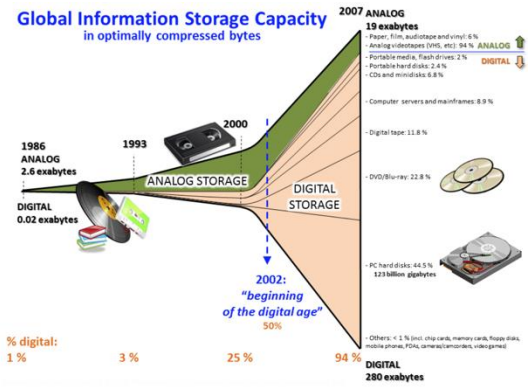


Fig. 1. Evolution of data

As shown in Fig. 1., digital storage can reach the order of Exabytes. Because of these "pretentious" specifications, the idea of big data became more and more popular and utilized.

This type of data relies on specific concepts, highlighted in Fig. 2., that come along with various challenges.

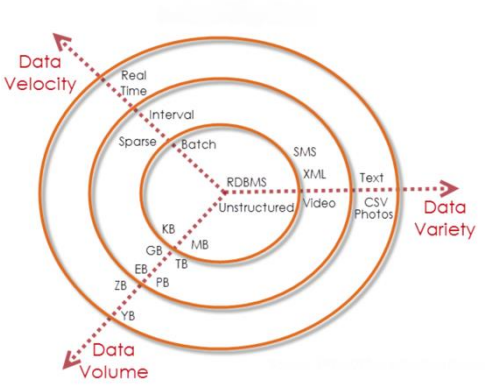


Fig. 2. Characteristics of big data

The most important one would be the volume, which means that large amounts of data can be processed at once, which brings an advantage when it comes to data analytics.

The next concept to be considered is velocity, referring to the rate at which data changed throughout an organization and being able to quickly use the available data for gaining different sorts of information can become a competitive advantage. For example, if online retailers can gather and

store a customer's history of navigation on the ecommerce site, they will be able to use that information for recommending additional purchases and this can work in their favor, inside a vying market [1].
The third most important concept refers to variety: data almost never has a homogenous form, perfectly ordered and ready to be used. This is why big data takes this raw, unprocessed form and extracts and orders its meaning, for it to be used further on, either by humans or as an input for algorithms or applications.

2 Big Data vs Traditional Databases

As shown in Table 1., there are many aspects to be considered when talking about the uses and differences of traditional databases and big data. Even if some criteria are specified, one cannot state that there is a better option when choosing the system for storing the needed data.

Table 1. Comparison between traditional databases and big data [2]

Characteristic	**Traditional Database**	**Big Data**
Data architecture	Centralized architecture, where problems are solved by a single computer, so it is costly and ineffective for large sets of data.	Distributed architecture in which the computations are done in a computer network, providing more power and improved performance.
Types of data	Structured data in fixed formats or fields that only provide insight at a small level.	Semi-structured or unstructured data, which allows the data to be gathered from a variety of sources and transformed into knowledge based information.
Volume of data	Small amount of data is stored, up to gigabytes.	Bigger amount of data, the order of petabytes.
Data schema	Fixed schema that is static and cannot be changed after it is saved.	Dynamic schema which is applied only after raw data is ready to be read.
Scaling	Very difficult to achieve, as it runs on only one server that would require a lot of power and generate high costs.	Scaled out architecture, in which the distributed computing principles employ more than one server.
Accuracy	Not all the data can be store, as this would be very expensive, so this decreases the amount of data to be analyzed, therefore the accuracy is also decreased.	Data is stored in big data systems, allowing huge amounts of data to be analyzed so the points of correlation are easily identified, providing high accurate results.

For both small and large applications that do not demand storing very big amounts of data, traditional databases are still the best option, as there are numerous things to do to optimize them and get the best out of them. It could even be considered a distributed architecture, that relies on the master-slave concept, so that data is processed fast, it is secure and can be easily recovered.
When talking about applications that store historical data, or that apply some concepts of data mining or machine learning, big data presents more advantages because of the accurate, varied data that it can store, but also because its reliable architecture and

dynamic schema.
It is obvious that both types of data storing come with their limitations and disadvantages and this is why, a thorough analysis should always be done, before choosing the best option for a specific case.

3 Big Data Technologies

When considering big data, the volumes of data that are used are way larger than the conventional ones, so powerful parallel processing in usually required. The specific architecture can be chosen regarding the needs of the application, considering which one of the three – volume, velocity, variety – is more relevant.
Cloud Dataflow is a native Google cloud data processing service integrated with simple programming model for both batch based and streaming data processing tasks.
This tool, which is a fully managed service handles the operational tasks including performance optimization and resource management. It also provides the possibility to manage the resources dynamically, in order to maintain high utilization efficiency while minimizing latency.
Cloud Dataflow provides a unified programming model method, so that programming model switching cost is no longer a issue. This method aids in batch and continuous stream processing, making it easy to express computational requirements without worrying about data source. [8]
To ease access to big stores of data, the concept of data lakes can come in handy. These data lakes represent vast data repositories that can collect data from various sources and store it in its raw, unprocessed state. Those repos differ from data warehouse, where even if data comes from different sources, it is processed and stored in a structured way
In this case, the lake and warehouse metaphors are fairly accurate. If data is like water, a data lake is natural and unfiltered like a body of water, while a data warehouse is more like a collection of water bottles stored on shelves [5].
In an enterprise, it may appear the case when they want to store data, but they are not sure yet how this data will be used, and this is when data lakes may be the best solution. For example, Internet of Things (IoT) data may be stored in data lakes, as it already has a big role in the growth and development of such storing solutions.
In the case if NoSQL databases, MongoDB is one that can be used for storing big data. Traditional relational databases cannot meet the current challenges posed by Big Data. Recently, databases of the NoSQL type are becoming more and more popular for storing large data. They have emerged from the need of companies like Google, Facebook or Twitter to manipulate huge amounts of data which traditional databases simply cannot handle.
NoSQL databases were designed to store very large volumes of data generally without a fixed scheme and partitioned on multiple servers. NoSQL databases offer flexible working modes, a simple API, and the possible consistency of a data. NoSQL databases thus become the core technology for Big Data. [9]
The main advantage of using NoSQL databases is that they allow efficient work with structured data, such as e-mail, multimedia, text processors. NoSQL databases can be seen as a new generation of databases: not relational, distributed, open source and characterized by horizontal scalability.
Another important feature of the NoSQL systems is the "shared nothing" architecture through which each server node is independent, does not shares memory or space. The architecture of a NoSQL database is shown in Fig. 4.

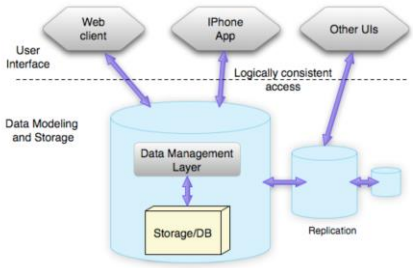


Fig. 3. NoSQL Architecture

3.1. Apache Frameworks

Hadoop

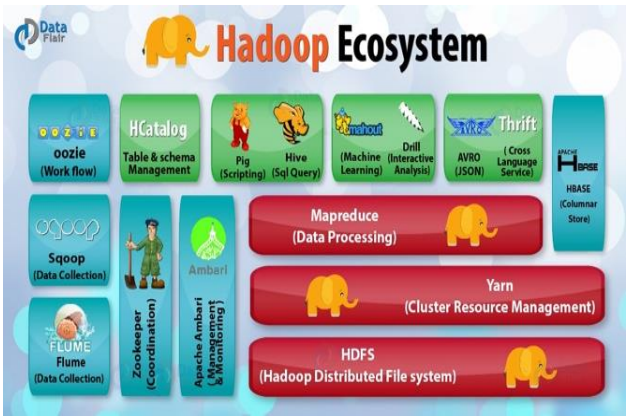


Fig. 4. Component scheme for Apache Hadoop

Apache Hadoop is an open source Java framework used for processing and querying big volume of data, on a set of large clusters. This project has been initiated by Yahoo! and its success is based on a large community of contributors for all over the world. Considering a significant technology investment done by Yahoo!, Hadoop has become a cloud computing technology ready to be used at an enterprise level and the most important framework when talking about big data.

It changes the economics and dynamics of large-scale computing, enabling scalable, fault-tolerant, flexible and cost-effective solutions [4].

Apache Hadoop consists of the following modules:

• **Hadoop Common**: contains libraries and tools needed for other Hadoop modules.

• **Hadoop YARN**: a resource management platform, responsible for cluster resource management and their use for user application planning

• **Hadoop MapReduce**: a programming model for large-scale data processing. It is named after the two basic operations that this module performs: reading data from the database, putting it in a suitable format for analysis (map-map), and performing mathematical calculations in a customer database (reduce), for example: counting men over 30 years of age.

• **Hadoop Distributed File System**: Allows you to store data in an easily accessible format in a large number of connected storage devices. A "file system" is the method used by a computer to store data so that it can be found and used. Normally, this is determined by the computer's operating system, yet a Hadoop system uses its own file system that is "above" the computer's file system itself - it can be accessed using any computer running any operating system accepted.

For data storing, Hadoop has its own distributed file system, HDFS, which makes the data available to multiple computing nodes. The typical Hadoop usage pattern involves three stages:

- Loading data into HDFS
- MapReduce operations
- Retrieving results form HDFS

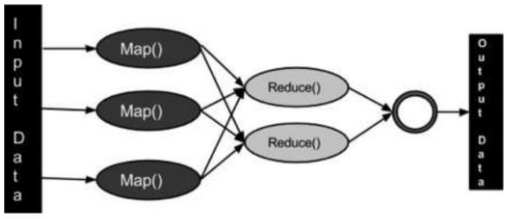


Fig. 5. MapReducer

As shown in Fig. 5., MapReducer algorithm performs two important tasks, Map and Reduce. Map is the task that takes converts an initial set of data into a new one, where elements are broken down into tuples (key/value pairs). Secondly, reduce task, takes the output from a map and uses it an

input in order to combine those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job [3].
Hadoop also offers solutions for making programming easier; working directly with Java APIs can be difficult and can expose the application to a lot of errors and restricts the usage of Hadoop to Java programmers only. Therefore, the other solutions that it offers are:

- Pig – a programming language that simplifies the common tasks of working with Hadoop: loading data, transformations, retrieving data.
- Hive – that enables Hadoop to operate as a data warehouse

Various other procedures, libraries or features have come to be considered part of the **Hadoop "framework"** in recent years, but Hadoop Distributed File System, Hadoop MapReduce, Hadoop Common, and Hadoop YARN are the most important.
The flexible nature of a Hadoop system refers to the ease of adding or modifying, for companies, the data system as their needs change, using cheap and easily accessible components from any IT service provider.
Today, this is the most widespread storage and processing system through computer groups - relatively cheap systems connected, as opposed to expensive, customized manual operations.
Some of the reasons why organizations use Hadoop is the ability to store, manage and analyze large amounts of structured and unstructured data quickly, reliably, flexibly and at low cost. The main benefits are:
• **Scalability and performance** - distributed data processing for each node in a cluster allows the company to store, manage, process, and analyze petabyte data.
• **Reliability** - Large clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resistant - when a node fails processing is redirected to the remaining functional nodes in the cluster and the data is automatically recreated for other failures that may follow.
• **Flexibility** - unlike traditional relational database management systems, structured schemes don't need to be created before data is stored. Data can be stored in any format, including semi-structured or unstructured formats.
• **Low cost** - unlike its own software, Hadoop is open source and runs on low-cost hardware groups.

Spark

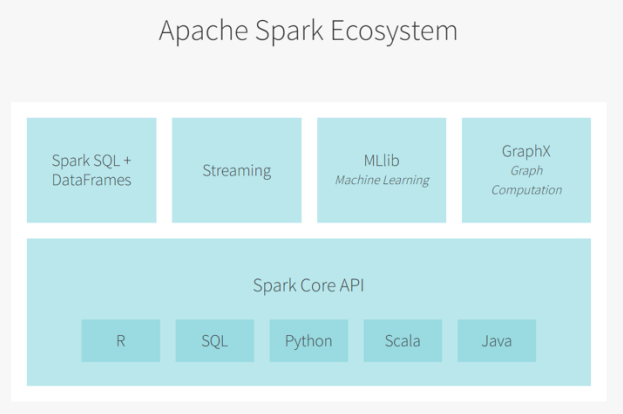


Fig. 6. Apache Spark component layout

Apache Spark is an open source cluster platform, an unified analysis engine for large-scale data processing.
Spark is seen by industry technicians as a more advanced product than Hadoop - it is newer and designed to work by processing data into pieces "in memory". This means that it transfers data from physical, magnetic hard drives to a much faster electronic memory, where processing can be done much faster - up to 100 times faster in some operations. It has proven to be very popular and is used by many large companies to store and analyze huge multi-petabyte data. This was partly because of its speed. Last year, Spark set a world record by completing a benchmark that included sorting **100 terabytes of data in 23 minutes**.

In addition, Spark has proven to be very suitable for Machine Learning applications. Machine learning is one of the growing and more exciting fields in computer science, where computers are taught to present data patterns and adapt their behaviour based on modelling and automated analysis of any task they are trying to achieve.

It is designed to be easy to install and use. To make it available to multiple companies, many vendors offer their own versions (as in the case of Hadoop) that are industry-specific or custom-tailored for individual client projects as well as associated consulting services to put them in function.

Spark uses cluster computers for its computational (analysis) power, as well as its storage. This means it can use resources from multiple computer processors connected for it analyses. With distributed storage, huge data sets gathered for Big Data analysis can be stored on multiple smaller physical disks.

It speeds up read / write operations, because the component that reads information on the disks has a smaller physical distance to migrate to the surface of the disk. Like processing power, you can add more storage space when needed, and using commonly available hardware groups (any standard hard disk for your computer) will help you maintain infrastructure costs.

The main benefits are:

• **Speed**: It can be 100 times faster than Hadoop for widespread data processing through computer memory and other optimizations. The spark is also fast when data is stored on the disk and currently holds the **world record for large-scale disk sorting**.

• **Easy to use:** has easy-to-use APIs to work on large data sets. This includes a collection of over 100 operators for data transformation and data APIs known to handle semi structured data.

• **A unified engine**: includes top-level libraries, including support for SQL queries, streaming data, automated learning and graphics processing. These standard libraries increase developer productivity and can be seamlessly combined to create complex workflows.

Ignite

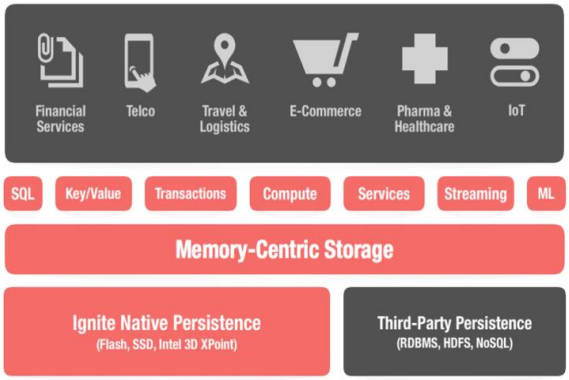


Fig. 7. Component Chart for Apache Ignite

Apache Ignite is a distributed open source, cache and processing platform designed to store and calculate large data volumes (petabyte scale) in a group of nodes. Provides a unified API that supports SQL, C ++, .NET, Java / Scala / Groovy, Node.js and more application-side access. The unified API connects cloud applications with multiple data warehouses that contain structured, semi-structured and unstructured data (SQL, NoSQL, Hadoop). It provides a high-performance data environment that allows companies to process all ACID transactions and generate valuable information from real-time and interactive queries.

Main Benefits:

• **Sustainable memory**: The durable memory component of the Ignite device treats RAM not only as a caching component but as a fully functional storage component. This means that users can turn persistence on or off as needed. If persistence is disabled, then Ignite can act as a database distributed in memory or in the data grid in memory, depending on the preference to use the SQL API or key. If persistence is enabled, then Ignite becomes a scalable distributed database that guarantees

complete data consistency and resists complete cluster errors.

• **ACID Compliance**: Data stored in Ignite is compatible with ACID both in memory and on disk, making the system consistent. Ignite operations work in the network and run on multiple servers.

• **Scalability and durability**: it's a scalable, horizontally distributed, system that accepts the addition and removal of cluster nodes on demand. Ignite also allows storage of multiple copies of data, making it resistant to partial cluster failures. If persistence is enabled, then the data stored in Ignite will not be affected by errors, they will remain consistent.

4 Software products

There are various software products that utilize the concept of big data, and in this section some of them will be presented, in order to give an overview of how anyone can find a product for addressing a specific issue or need regarding the use of big data.

Fujitsu proposes to the market a large data toolkit, generic called Big Data Software, which allows large data technology to be used efficiently in the information and mission-critical systems of companies. [10]

Practically, Big Data Software consists of four products: a parallel-distributed data processing product, a complex event processing product, a ultra-fast transaction processing product, and an in-memory data management product, all of which represent standard technologies in large data applications.

According to the company, besides the reliability and performance guaranteed by incorporating many proprietary Fujitsu technologies, which have a proven track record of good results in mission-critical systems, solutions can be easily installed and operated. They can also be combined with products from other vendors, including open source software to build ecosystems to help customers use large data solutions.

Cloudera Enterprise 4.0 Big Data is the most recent product of Cloudera and it is a management platform. [11]

This platform for managing and processing big data provides tools for deploying and managing Hadoop systems, as well as management automation of large scale clusters and an easy integration with a broader range of management tools and data sources.

The new version of Hadoop that the platform uses, offers high-availability features that eliminates the single point of failure of the Hadoop Distributed File System, increased security that allows more sensitive data to be stored in CDH, and the ability to run multiple data processing frameworks on the same Hadoop cluster.

Datameer 2.0, is a software for Big Data analytics which can combine data integration, analytics and visualization into a single package that offers a spreadsheet interface. [12] This software is offered both in enterprise edition and workgroup and desktop editions to ease the access for every user.

A Business Infographics Designer is included in this software, in order to easy creation of graphics and data visualization design control. Built on HTML5, the software provides an enhanced user interface, and also offers support for additional data sources including Facebook and Twitter. It also has a useful feature that provides improved integration with the Hive data warehouse system for Hadoop.

Another software application useful for big data LucidWorks Big Data [13], which is a cloud-based development system of open-source software for prototyping Big Data applications. This application can help businesses analyze unstructured information, the so called "dark data" - text messages, audio files, email repositories, log files and other unstructured content.

LucidWorks Big Data is good choice as it incorporates a lot of different technologies such as Hadoop, but also Apache Lucene and Solr search, which are open source applications.

Moreover, it has R programming language for developing analytical applications and supports Apache Mahout which is used for building scalable machine learning algorithms.

5. Case Study Apache Ignite vs Spark

In order to compare the two frameworks, a multi node setup was used. Each node consists of a virtual machine that runs Ubuntu Linux 16.04 LTS. These VMs were provisioned with 1 virtual CPU, 2048 MB of RAM and a network card bridged in a Local Area Network. Each node had been assigned a static IP in order to easily start and stop them. In order to monitor the cluster, Elasticsearch and Kibana were used on a laptop that collected all the data received from each node. Each node sent data to Elasticsearch through metricbeat.

To properly asses the performance of these frameworks, different scenarios were tested, but the main calculation that was performed was a matrix multiplication of square matrices. The purpose was to assess how these frameworks behave in terms of time of execution, resources used and scalability. With these benchmarks in mind, this testing plan took shape:

For each framework, a matrix multiplication algorithm would be developed in their native APIs programming language:

Scala for Spark and Java for Ignite, and these algorithms are ran across a cluster of nodes. In order to asses the way these frameworks use computing resources, how much time they need to do matrix multiplication and in order to see how scalable they are, the need to vary some parameters arised.

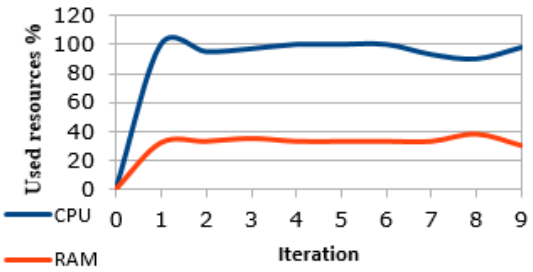
The parameters that were varied were the number of rows and columns each matrix had and the number of nodes that would take part in the computation. So, for each framework, the algorithm was ran for 9 times, varying the matrix dimension from 1000 rows and columns to 2500 and then finally to 5000, and by varying the number of nodes from 2 to 4 and a maximum of 6 nodes.

Below, we can see the way they behaved in each of these scenarios.

Spark

Table 2 – Spark Results

Iteration No.	No. of nodes	Matix dimension	Time (mm:ss)
1	2	1000x1000	00:12
2	2	2500x2500	00:23
3	2	5000x5000	00:31
4	4	1000x1000	00:08
5	4	2500x2500	00:14
6	4	5000x5000	00:20
7	6	1000x1000	00:05
8	6	2500x2500	00:12
9	6	5000x5000	00:18



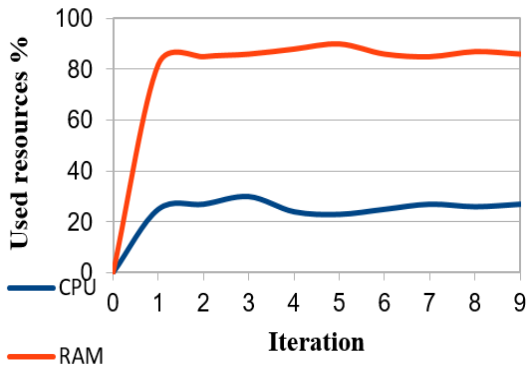
Spark used almost always 100% of CPU resources and about 33% of RAM resources

Ignite

Table 3 – Ignite Results

Iteration No.	No. of nodes	Matix dimension	Time (mm:ss)
1	2	1000x1000	00:11
2	2	2500x2500	02:56
3	2	5000x5000	28:04
4	4	1000x1000	00:05

5	4	2500x2500	01:25
6	4	5000x5000	24:09
7	6	1000x1000	00:05
8	6	2500x2500	01:52
9	6	5000x5000	27:10



Spark used almost always 100% of CPU resources and about 33% of RAM resources

6. Conclusions

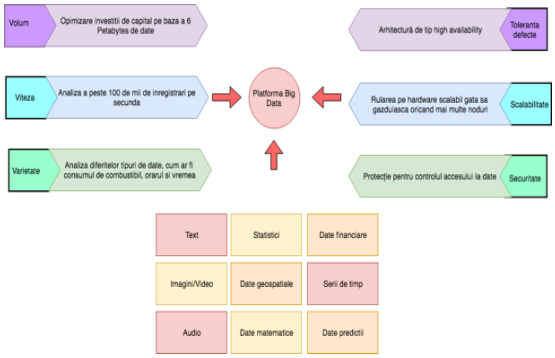


Fig.8. Summary of Big Data

As shown in Fig.8., there are various things to consider when talking about big data, no matter if it is about implementing or using such a deposit. There are the three most important requirements to be met, but also other criteria to be considered, as big data is not only about the technical part, i.e. how it is implemented, but also about the enterprise needs of such storing method.

When talking about key requirements, one should consider some main points. An application needs to handle and process huge amount of data, for example in the case of a financial application, it may need to optimize capital investments based on 6 Petabytes of information. Due to this amount, the first key requirement, volume, comes up.

Next, a vital element of many successful businesses is represented by the clients. A common need could be analyzing 100k records per second, in order to address customer satisfaction in real time, by giving suggestions or special offers. This is a good example when velocity is needed.

For the last key requirement, variety, the need to optimize shipping logistics could be considered, which demands the analysis of various types of data, such as fuel consumption, schedule, and weather patterns.

Apart from these requirements there are also some enterprise criteria that are important in every field of business. No matter if it is about finance, tourism, or statistics, a high failure tolerance is needed, which means a high availability architecture to support hardware or application failure.

Furthermore, big data should be a good option for expanding enterprises, so the hardware should be scalable and always ready to accommodate new nodes for processing more data. Security is also a must, as big data, like any other database solution, requires protection for granular data access control. The last element to be considered in that it needs to analyze data in native format and not demand a specific, standardized one, so that different types of data, such as text, images, statistics or time series can be all processed in their raw form.

Considering the presented case study, it can be stated that choosing the right technology or framework is strongly related to the needs of the specific company and to what kind of data is to be processed.

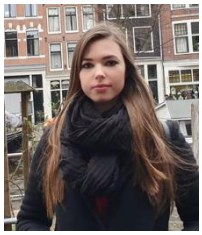
In terms of time spent computing, Spark is the clear winner by far. In terms of resources used, the two frameworks are very different. In terms of scalability, it appears that Ignite is not scalable because, as the number of nodes rises, the time does not shorten.

This issue might be because of the limitations of the LAN speed that might have slowed down the communication between the master node and the slaves.
The main conclusion to be drawn about big data is that it can be a very good solution for storing huge amounts of data, without having to put it in a specific form. There are many advantages for using it, as it provides volume, variety and velocity, and there are various technologies and software products that can address all the needs a company has when it come to storing data and processing it in order to obtain valuable information.

7. References

[1] O'Reilly Media Inc., Big Data Now: 2012 Edition, Kindle Edition
[2]https://www.projectguru.in/publications/difference-traditional-data-big-data/
[3]https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm
[4] Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing, ISBN 9763-78236-328-2
[5] https://www.datamation.com/big-data/big-data-technologies.html
[6] Mike Barlow, Real-Time Big Data Analytics: Emerging Architecture Kindle Edition
[7]http://www.imexresearch.com/newsletters/Newsletter2_22.html
[8] https://cloud.google.com/dataflow/
[9] Guy Harrison, Next Generation Databases: NoSQL, NewSQL, and Big Data, Apress Publishing, ISBN 978-1-484213-30-8
[10]http://www.clubitc.ro/2017/06/02/solutii-fujitsu-pentru-recolta-big-data/
[11]https://blog.cloudera.com/blog/tag/big-data/
[12] https://www.datameer.com/product/
[13]https://lucidworks.com/2013/01/29/getting-started-with-lucidworks-big-data/
[14] https://databricks.com/spark/about
[15]https://opensource.com/business/15/4/guide-to-apache-spark-streaming
[16]https://apacheignite.readme.io/docs
[17]https://www.baeldung.com/apache-ignite

Mădălina GHEȚU is a second year master's student at University of Economic Studies of Bucharest. She graduated from University Politehnica of Bucharest, Faculty of Engineering in Foreign Languages, holding a computer science degree. She is an instructor, teaching fundamentals of PHP programming and MySQL databases, but also a web developer, working with various technologies, including PHP, JavaScript and PostgreSQL.



Adriana BĂNUȚĂ is a second year student at University of Economic Studies of Bucharest at "Database - Business Support" Master. She is working as quality assurance engineer in an eCommerce company. She is familiar with PHP, PL/SQL and MySQL.

Cătălina DRAGOMIR is a graduate of the Faculty of Entrepreneurship, Business Engineering and Management, of the Politehnica University Bucharest in 2017. She is currently a student at the Academy of Economics Studies of Bucharest, and she is working as an IT Consultant at a small company in Bucharest.