DOI: https://doi.org/10.37075/EA.2021.2.09 Economic Alternatives, 2021, Issue 2, pp. 315-328

Big Data Analysis Architecture

Received: 03.02.2021
Available online: 28.06.2021

Daniel Delchev[*], Vanya Lazarova[**]

Abstract

Nowadays, because of the rapid changes in the environment we can observe complex developments in the companies. Furthermore, the recent technological growth renders the business world a very competitive environment. It is essential for companies to make highly accurate decisions in a short period of time in order to compete in such an environment. They have to rely not only on the amount of data that they have gathered but also on the efficiency of gathering new data from external sources. Big Data is a crucial part in the decision making of big and small enterprises. To use Big Data efficiently it is necessary for the companies to rethink their whole data infrastructure. Traditional architectures cannot enable the companies to explore a whole new range of possibilities such as: penetrating new markets, widening their existing ones, enriching their assortment, increasing their profitability and many more. Modern architectures must consist of additional data sources, new technological components and modern ways to analyse the data.[1]

Keywords: Big Data, Traditional Big Data Architecture, Big Data Analysis Architecture.
JEL: O33, O36, L20

Introduction. Data and Big Data Sources

In today's digital society Big Data is essential for both small and large companies. For a small period of time storing and correctly using data became a vital part in the success of any company. Companies can gain a number of advantages against their competitors because Big Data enables them to analyse the current market and foresee what changes might occur in the future. This knowledge allows them to make more accurate decisions about the future development of their business.

Big Data can reveal disadvantages in the already integrated ways of gathering knowledge because of the enormous amount of data that is being stored. To refine the existing strategies, the data has to be analysed in order to discover inefficiencies in the systems. By removing those flaws, systems for gaining knowledge will become more adequate in the ever-changing environment.

Big Data has been researched by numerous scientists and experts in the field but still there are a lot of improvements that

[*] PhD student, Department: Information Technologies and Communications, University of National and World Economy.
[**] Assos.prof PhD, Department: Information Technologies and Communications, University of National and World Economy.

[1] This work has been supported by the project NoBG05M2OP001-1.002-0002 "Digital Transformation of Economy in Big Data Environment", funded by the Operational Program "Science and Education for Smart Growth" 2014–2020, Procedure BG05M2OP001-1.002 "Establishment and development of competence centers".

have to be made in order to overcome the challenges that Big Data has brought. Some of them are connected to data efficiency, credibility, sensitivity, accuracy and more.

The most important aspects of Big Data Science are gathering, storing, analysing, searching, updating and visualising the data. Also it has to be well protected because sometimes it might contain sensitive information. In the past data was characterized in three different ways: in terms of speed, variety and size. Eventually two more characteristics got in use: value and credibility. Nowadays, Big Data is defined as the work with huge arrays of data which describe a user's behaviour. A broader definition is the data analysis whose objective is to obtain a meaningful result (such as a forecast for a stock market value). Such data can improve the work efficiency of a company or organisation in any field - government agencies, the field of science, business sector and financing technologies (Tsaneva, M. (2019)).

The ability to synthesise the useful data from large amounts of data is a process of data extraction. Specific techniques for extraction are becoming more and more sophisticated and nowadays they are a long way from just querying the information. Social networks, online markets, blogs, sensors, health care facilities and others produce petabytes worth of information. This quantity cannot be compared with the traditional amount that people are used to. The data has to be labeled accordingly in terms of variety, speed, size, credibility and its value. Data can be unstructured, semi-structured and structured which is reflected in the variety aspect. Speed is measured in the amount of data that is being produced by the data source per unit of time. The size of the data shows how much data there is and usually grows exponentially. The credibility of Big Data shows the truthfulness in the information but also takes in consideration its accuracy. Data value is required when we gather information from social data. Some people describe data value as a necessity when building any type of Big Data application because it allows for the extraction of the most essential information. We can observe a trend in data extraction using NoSQL databases (Radoev, M (2017; 2019)) which differ a lot from the well-known relational databases.

Everything described above proves the need of creating efficient and innovative algorithms which can find patterns in data analysis models, discover different interdependencies in data and others. Only then would we be able to extract useful information from Big Data.

Life in a digital era has rendered knowledge extraction from unstructured, semi-structured and structured data a main challenge we have to deal with. Big Data scientists have to have exceptional knowledge in the field and put a lot of hard work in their research in order to overcome the difficulties that are imposed on the companies.

One of the primary features of Big Data is the way it is structured. It is divided into three categories:

- **Unstructured data** - Email, social networks (Facebook, Twitter, LinkedIn), text files, presentations, mobile data (text messages), communications (chat), phone records, location data (GPS) of mobile devices (mobile phones, tablets) and many more. Unstructured data analysis searches for patterns in text, media, photos and files consisting of similar contents. That is a lot different from the conventional search which queries documents using a given string. Text analysis searches for repeating patterns in documents, emails, recorded conversations in order to collect knowledge. There are

several methods for unstructured data analysis. Some of them are Natural Language Processing (NLP), Master Data Management (MDM) and other statistics. Text analysis uses NoSQL databases which standardize the data in order for it to be queried using languages such as PIG and Hive.

- **Structured data** - data generated by a computer (either automatically or with the help of human intervention), metadata (time of creation, file size, author, etc.), library catalogues (date, author, location, topic, etc.), mobile numbers and phone books. Such data is stored in databases (Microsoft SQL Oracle, MySQL, etc.). Using queries one can easily extract arrays of data which can later be used for decision making. Enterprises will continue to analyse structured data using various methods such as Structured Query Language (SQL).
- **Semi-structured data** - data which does not obey the tabular structure of data models associated with databases or other forms of data tables, but nonetheless contains tags or other markers which separate semantic elements and enforce fields within the data. The main advantages of semi-structured data are:
  - Data sources are not limited to a given data model which allows for a broader range of available data sources
  - Data can be easily transferred between databases
  - The data model can be easily updated

XML is a form of semi-structured data and is used for data transferring between servers and applications. A XML analyser evaluates the document's hierarchy, extracts the document's structure and/or its contents.

Big Data sources can be divided in 3 main categories:

- **People (social networks)** - This is data gathered from posts in social networks. The contents and the quality of information depends on the platform, its purpose and the audience that the post has reached. Data gathered from news comments and even the search history of engines such as Google, Bing, etc. also falls into this category.
- **Automated systems (sensors and specialized mechanisms)** - Sensors gather images from space crafts and trace people's locations using applications on their mobile phones, IoT devices, etc. Mobile applications analyse data for geolocations and IoT devices automatically monitor activities such as tracking people's movements, rainfall information and more. Traffic from wireless and mobile devices is increasing rapidly and generating data to analyze consumer behavior.
- **IT systems in business models** - This is data generated by the Enterprise Resource Planning (ERP). ERP refers to a type of software that organizations use to manage day-to-day business activities such as accounting, procurement, project management, risk management and compliance, and supply chain operations. Knowledge extraction from the huge arrays of data is not at all an easy task. Even while you are reading this article the data sources are generating a vast quantity of data. This is a big factor which hinders a company's capacity because every time new data is generated every bit of information has to be processed and stored somewhere. That is why companies and organizations that want to take advantage of the benefits of working with Big Data must, and must have, technologies that extract knowledge from data - data mining. In addition, it is necessary to implement specialized algorithms from

the field of machine learning (Mihova, Veska (2015)). Organizations practicing in today's information society are beginning to consider not only human capital, but also the data from which they gain knowledge and information about one of the most valuable and important resources. Increasingly, information in organizations is beginning to be perceived not as plain information, but as a strategic business asset, regardless of the industry that the business is practicing in.

Big Data Processing

Extracting information (knowledge) from data is the basis of modern business intelligence. However, existing information retrieval algorithms do not work directly with Big Data, but need to be adapted to work with Big Data processing platforms such as Hadoop and MapReduce or the new open source platform Spark.

The development and implementation of algorithms for extracting information from Big Data is a serious challenge. Two main methods of Big Data processing will be considered:

- MapReduce
- MPP (Massively Parallel Processing)

MapReduce characteristics

Hadoop enables you to organize distributed processing of large volumes of data (measured in petabytes) using the map / reduce method. MapReduce is the process of dividing each individual task into smaller ones, each of which can be performed on a separate node from the cluster.

The MapReduce algorithm performs a highly efficient parallel data processing using the Hadoop MapReduce engine. This machine provides all the tools required to split Big Data into manageable pieces and process them in parallel on a distributed cluster. MapReduce is a way to perform a set of functions on a large amount of data in batch mode.

The *map* function splits the tasks in a number of smaller ones and handles their positioning in a way that balances workloads and facilitates fault recovery.

Once the distributed calculation is complete, the function *reduce* combines all the elements back and provides the final result.

In order to take advantage of MapReduce the method has to be implemented on a distributed cluster of servers which provide a way to apply the same function simultaneously by a number of machines.

MPP characteristics

Massive parallel processing (MPP) is changing the market of database management tools. This method has a performance advantage over traditional approaches, thanks to a parallel processing architecture that combines blade servers and data filtering via programmable logic devices. This combination provides fast analytics queries that support thousands of users of business analysis applications, complex analysis with tremendous speed and petabyte scalability. MPP delivers a high performance without the need of indexing or tuning.

All of the integration of hardware, software and storage is done in advance for the user because it is also a tool. This method provides a ready-to-use form of immediate data entry and allows for the execution of queries via leading ETL, BI and analytical applications via standard ODBC, JDBC and OLE DB interfaces. Massive Parallel Processing greatly simplifies the analysis by consolidating all analytical activity in one place - exactly where the data is stored.

MPP (massive parallel processing) is a coordinated processing executed by multiple processors working on separate parts of a

given program. Each processor uses its own operating system and memory.

Usually MPP processors communicate through a type of messaging interface. Some implementations allow up to 200 or more processors to run on the same application. In some implementations, up to 200+ processors can run on the same application simultaneously. Setting up MPP is a more complex task and requires consideration of how the workload and data will be distributed among the processors. MPP systems can be described as a *loosely coupled* or *shared nothing* systems. Unlike Hive QL, Impala is not based on a MapReduce algorithm. It implements a distributed architecture based on daemon processes which are responsible for all aspects of the request execution.

Good Practices for Big Data Analysis

Hadoop enables the distributed processing of a large set of data across multiple server clusters. It allows for a single server to run on up to a 1000 machines, but also introduces a high error tolerance. It also provides means to cost-effectively store and process unstructured or semi-structured data from streams of web clicks, social media, server logs, devices, sensors and more. It allows you to draw conclusions from analysis on newly added datasets. These conclusions can reveal new business opportunities and advance a given organization. Hadoop is a very good way for implementing network performance management, data storage, risk management, customer loss prevention, fraud detection, customer opinion analysis, social media analysis and others.

When employees get a better grasp on Hadoop it can be applied to many projects and bring benefits for the entire organization. Integrating new knowledge sharing practices would provide value to different business groups and improve efficiency when adopting new technologies. Best collaboration and training practices that result from Hadoop may include:

- **Private social networks.** Encouraging communication within and between teams through an internal social network. Publishing interesting articles, tutorials and presentations, which can be found through blogs via Twitter or Facebook. Such blogs can encourage employees to think outside the current technical paradigms.
- **Innovation time.** Employees can be offered the opportunity to devote a certain percentage of their time to new projects. Google is known for offering its development teams to dedicate 20% of their time for new initiatives. This has led to a number of key innovations, such as Gmail and Google Groups.
- **Topical discussions.** To offer developers the opportunity to share their knowledge and experience in a public forum. Topical discussions on Friday afternoon, for example, have become a standard for many start-ups in the Silicon Valley. One can choose the best presenters and invite a number of groups to attend a topical discussion on an arbitrarily chosen date. This is a good example of a forum for presenting projects created during an innovation period.
- **Evening consumer groups.** Periodic evening event for company employees to present new tools or products they are currently building for the consumers.
- **Mobile versions of company applications.** In many cases, operational enterprise applications are not suitable for mobile devices. The application of Hadoop allows distributed processing of a large set of data. This makes it possible to create mobile applications that provide end users with the information they need,

such as payrolls, benefits and deductions. All of this can be compacted within a sleek design.

- **Bringing back offices to the front.** Businesses often have some amount of data distributed across different systems. This data is stored in different formats, in different infrastructures and for different purposes. Therefore, it can be quite challenging to create new applications that use this data. This data is of tremendous value so businesses can use Hadoop together with the technique of "bringing back offices to the front" in order to create new applications. With Hadoop as a data center, data can be extracted from inherited, closed systems which allows for new applications to be built.
- **Single customer view.** Businesses are starting to interact more and more with their customers using new ways, such as social media, websites and mobile applications. However, Traditional Customer Relationship management systems (CRM) are insufficiently equipped to handle such interactions because the generated data is unstructured. Data pours into the databases very quickly (for example, click flows) which forces the enterprises to use non-relational databases in order to overcome these challenges.
- **Infrastructure monitoring.** Many large organizations have hundreds or even thousands of internal systems. These operating systems are often critical for maintaining a business' workflow, uptime and productivity. Monitoring is a complex process that involves processing and analyzing multiple signals from different subsystems, including storage, network, system components, and software processes.

From a traditional architecture to a Big Data architecture

In order to understand the aspects of a Big Data architecture, let's first look at a traditional information architecture for analyzing large structured data. Figure 1 shows two data sources which are typical of almost every business. Transaction data is reported by the internal operating systems such as production management, accounting and finance, logistics, warehousing, human resource management and more. The company may have a unified enterprise resource planning (ERP), or the data may come from separate subsystems. The other source of data within the company are the existing nomenclatures that the company handles (nomenclatures of employees, customers, suppliers, goods, materials, types of documents, etc.) as well as other basic data such as numbers and dates of documents, patents, state regulatory documents, etc.

Techniques for integrating and transferring data to a DBMS repository or data warehouse can be applied to these traditional sources. They offer a wide variety of analytical capabilities for extracting knowledge from data. Some of these analytical capabilities include: dashboards, reports, BI applications, summary and statistical queries, semantic interpretations for text data, and high-density data visualization tools. In addition, some organizations have applied supervision and standardization to projects and may have further developed the capacity of the information architecture by managing it at the enterprise level.

Figure 1: *Components of a traditional architecture for Big Data analysis*

The basic principles of information architecture include treating data as an asset through value, cost and risk as well as ensuring the timeliness, quality and accuracy of data. It is the responsibility of the enterprise architecture oversight to establish and maintain a balanced management approach, including the use of a center of excellence for standards management and training.

Architecture of a Bid Data Analysis System

The amount of data produced by devices, services and sensors throughout the economy and society is constantly increasing every day. This creates opportunities for innovation and improvement of existing products and services or the creation of new ones. Data diversity requires the formulation of approaches that allow data to be exploited in order to stimulate innovation and productivity, while ensuring security and respecting privacy and intellectual property rights. The type and categorization of the generated data may depend on the source, application or business model.

In order to meet the new requirements for data size, speed, diversity and value, the traditional architecture needs to be changed accordingly. The new corporate architecture for Big Data must include new data sources, new technological components and new analysis tools which differ greatly from the traditional ones.

The data sources in the architecture of a Big Data analysis system include both machine-generated data and data from word processing. The machine-generated data includes: databases, metadata, catalogs, XML files, etc. The text data includes: e-mail, social networks, text files from word processing, spreadsheets, presentations, text messages, chats, phone records, etc.

Unique distributed (multi-node) architectures for parallel processing (Distributed File Systems) have been created for analyzing large data sets. There are different technological strategies for real-time storage and batch processing. In order to store key-value data in real time one has to use Key-value Store and NoSQL (more commonly referred to as Hashtables) which allow for highly efficient data extraction based on an index. Map Reduce is used for batch processing in order to filter the data according to a specific search strategy. Once the filtered data is detected it can be analyzed directly, sent to mobile devices, loaded into other unstructured or semi-structured databases or even combined in a traditional storage environment and associated with structured data.

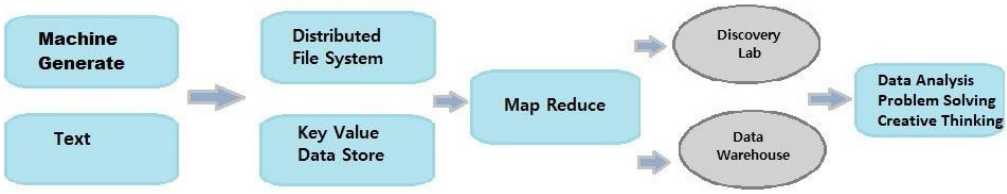

Figure 2: *Components of a Big Data Architecture*

In addition to the new components, new architectures are emerging to effectively adapt to the new requirements for storage, access, processing and analysis. Dedicated data warehouses suitable for this purpose are able to store and optimize processing for new types of data. One strategy assumes that Big Data-oriented architectures will have many types of data warehouses. Another way to process it is to parallelize MPP-based data for both speed and size. This is crucial for next-generation services and analysis, which can be scaled to any latency and size requirements. A third way to process the data is by using MPP data pipelines in order to allow the handling of data events in moving time periods with variable latency. In the long run this will change the way ETL is used for most uses.

In short this is what a corporate Big Data architecture might look like (Figure 3).

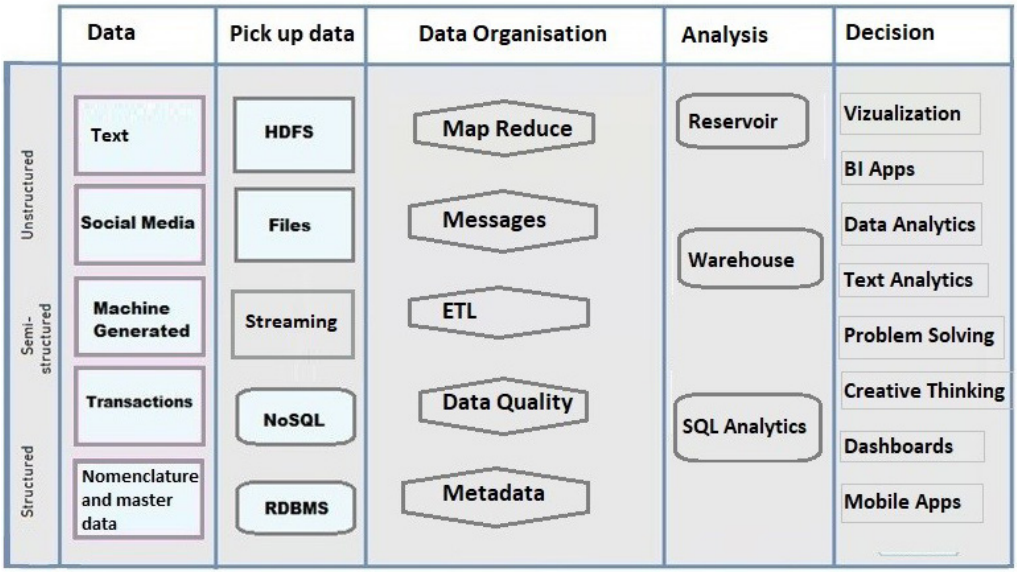

Figure 3: *Corporate Big Data Architecture*

Different types of data (structured, unstructured and semi-structured) come from the sources. Data can either be written directly (in real time) to memory processes or it can be written to disk as messages, files, or database transactions. Once the data has been collected (Pick up Data) there are a number of ways for it to be stored. It can be written to a file system, traditional RDBMS or distributed clusters such as NoSQL and Hadoop Distributed File System (HDFS). The basic techniques for organizing and quickly evaluating unstructured data are by running MapReduce (Hadoop) in a package or Spark

in memory. Additional options are available for evaluating real-time streaming.

The integration layer (Analyze) in the environment is extensive and allows the filling of a data tank, data warehouse and SQL analytical architecture. It covers all types of data and domains and manages the traditional and the new environment for data collection and processing in both directions. Different technologies can be used in the integration layer (Milev, Plamen. (2019)). Most importantly, it meets the requirements of the four Vs: exceptional volume and speed, a variety of data types and finding its veracity. All of this is self-sustained and does not depend on the analyst. In addition, the architecture provides data quality services, maintains metadata and tracks the lineage of transformation.

Technological Tools for Building a Big Data Analysis System

The means for implementing a Big Data analysis system architecture can be a combination of solutions. Here we will point out some of them and later we will consider some of the key technological innovations.

- Data entry in a Big Data platform - Apache Flume, Apache Storm, Apache Spark Streaming, Oracle Stream Explore and Apache Kafka.
- Acquisition of processed data from the primary one and its recording in relational and non-relational databases - HDFS and Cloudera Manager.
- Data organization - Apache HBase, Apache Kudu, Oracle NoSQL Database.
- Data processing - Apache Hadoop, Apache Spark, MapReduce.
- Data integration - Oracle Big Data Connectors, SQL Access
- Data preparation for analytical use and SQL access - Hadoop Hive and Hadoop Impala.

Hadoop, Hive and Impala

One of the latest technological tools of Hadoop Big Data processing and analysis is Impala - MPP system with SQL.

Impala combines the capabilities of SQL with the scalability and flexibility of Apache Hadoop using standard components such as HDFS, HBase, Metastore, YARN and Sentry. Impala users can communicate with HDFS or HBase via SQL queries. Impala can read almost all data file formats and is a pioneer in the use of the Parquet file format - a columnar storage layout that is optimized for large-scale queries.

Impala is freely available as an open source application under the Apache license. It supports data processing in memory i.e. accesses and analyzes data stored in Hadoop without data movement. Data can be accessed using Impala using SQL-like queries. It provides faster access to data in HDFS compared to other SQL mechanisms. Using Impala, data can be stored in storage systems such as HDFS, Apache HBase and Amazon s3.

Impala can be integrated with BI tools such as Tableau, Pentaho, Micro strategy and Zoom data. Impala uses metadata, ODBC driver and SQL syntax from Apache Hive.

Hive provides a mechanism for designing a Hadoop data structure and retrieving queries using a SQL-like language called HiveQL. Apache Hive makes it easy to read, write, and manage large arrays of data stored in distributed storage. The structure can be designed on data that is already stored. Nowadays, Spark is used to execute queries written in HiveQL because it offers enhanced performance.

Apache Spark

Apache Spark is a general-purpose cluster computing system. It supports many machines and uses various computational

tools for structured data processing, machine learning, graph processing and data flow processing. Soon after Apache Spark was developed, it became one of the leading cluster computing systems. The functions of Spark are manifested in different directions - it supports interfaces for Java, SQL, Scala and Python. It is used by a number of telecommunications companies, banks, game development companies and industrial giants such as Microsoft, IBM and Apple. Even some governments are using Spark for a portion of their software products.

Main features of Apache Spark are:

- Fast processing: When using Apache Spark one can achieve 100 times faster speeds when processing in RAM and 10 times the speed when doing so on a disk.
- Dynamic development: Ability to develop parallel applications because of the 80 high-level operators that Spark provides.
- In-memory calculation: In-memory processing can increase the processing speed. Here the data is cached thus, it is not necessary to retrieve data from the disk every time. Spark achieves such speeds because of a DAG execution mechanism which facilitates calculation in memory and acyclic data flow.
- Reusability: Ability to reuse the Spark code for batch processing and to execute ad-hoc requests in the flow state
- Fail safe: Spark-RDD abstraction provides resistance to failures and damage. Spark RDDs are designed to deal with the possible failure of each working node in the cluster. They ensure that data loss is reduced to zero.
- Real-time stream processing: Spark has the ability to process real-time streams with Spark Streaming.
- Multi-language support: Spark has support for multiple languages such as Java, R, Scala and Python. It has an edge over Hadoop because Hadoop allows only for Java development.
- Active and expanding community: Developers from over 50 companies have been involved in the creation of Apache Spark. This project was launched in 2009 and as of today it has about 250 developers who have contributed to its expansion. It is considered the most important project of the Apache Community.
- Complex analysis support: Spark offers special tools for data flow processing, queries and machine learning.
- Hadoop integration: Spark works independently on Hadoop YARN Cluster Manager which enables it to read existing Hadoop data.
- Spark GraphX: Spark has GraphX - a component for graph processing and parallel computing. It simplifies graph analysis tasks through algorithms and constructors.
- Cost-effectiveness: Apache Spark is a cost-effective Big Data solution. Together with Hadoop, it provides the necessary Big Data storage and replication.

There are two main components to Apache Spark. A driver which generates the user code in different tasks and distributes the tasks in the individual nodes. The second component is an executor - it performs the tasks assigned by the nodes. The cluster manager plays an important role in the connection between the two components.

The user data commands created by Apache are provided in an acyclic graph or DAG which determine which tasks have to be performed and the order they have to be executed in.

Some analysts compare Apache Spark and Apache Hadoop. However, this is not correct, as the two systems provide different functionality. Spark can be easily found in most Hadoop distributions. It is a common

belief that Hadoop has become so popular thanks to MapReduce. There are two main reasons that put Spark at the forefront and made it a desirable cluster computing system for large data set processing.

As a first reason, from a technical point of view Spark is much more convenient for developers due to the presence of Spark API (multiple interfaces). The other main reason is speed. Spark can perform tasks many times faster than MapReduce especially when it comes to multi-stage tasks. Map Reduce generates a two-step graph of execution which includes data reduction and mapping. At the same time, Apache Spark DAG includes many stages that enable a much more efficient performance. Even when data cannot be stored in memory, Apache Spark is several times faster than MapReduce.

The Apache Spark API is designed to be easier to use when compared to MapReduce. This is because much of the complexity of allocation and calculation processes remains hidden behind simple methods.

Apache Spark works with Python and R and is thus oriented towards large systems with Java and Scala. Everyone who uses Apache Spark has an accessible way to get a fast and scalable platform. We are currently flooded with vast amounts of information, which when properly analyzed, can lead to useful results. In many organizations and companies, such as Amazon, data analysis is applied after information about the products in question has been recorded by the end user. In other companies consumers receive various offers in real time as a result of the analysis of data gathered from the products in question. Every other company that generates and uses a huge amount of data can apply those actions to improve the work process and increase its customer satisfaction.

Now is the time to mention the importance of data streaming. Data streaming can be considered as a special method by which information is obtained continuously in different sequences. Information from various sources constantly floods our systems. The unstructured data that comes in a continuous sequence is called a data stream. The way such data is processed is by it being divided into logically related groups. The processing and analysis of data is called a streaming process. Further processing of the data is performed after it is divided into separate units. Low latency data processing and data analysis is called flow processing. Apache Spark can be linked to many sources of information, such as Amazon Kinesis, Apache Flume, and TCP Sockets.

Apache Spark Streaming

Spark Streaming divides the input data streams into packets. Spark Engine is used to process these packets and generates a final stream of packets. Spark Streaming can be easily integrated in many Apache Spark components such as Spark SQL and Spark MLib.

Spark Streaming is used to process data streaming. The Spark API environment plays a key role as it is adapted for streaming and error correction. Another powerful feature of Spark Streaming is real-time processing. The service provided by Spark Streaming is used by big companies such as Uber, Netflix and others. A characteristic feature of Spark Streaming is the ability to process data in real time using a standalone platform built for Spark Streaming.

Why is Apache Spark needed?

We have different types of data sources such as IoT devices, large systems and many more. Data sources generate data streams which are used by companies such as Apache Kafka. Data is grouped into clusters which process it in parallel. Continuous processes consist of several phases (operators). The

data is processed by a sequence of operators. After one operator processes the data, its output is passed to the next operator from the pipeline.

In modern computer architectures, problems occur when complex real-time data processing is required and following challenges have to be overcome:

1. Fast error recovery

This is the process of recovering information after an error occurs. The data is recalculated into parallel computing branches and allows for much faster process completion than traditional computer systems have to offer.

2. Balancing the load

Load balancing is a process of intelligent allocation of resources. What it tries to achieve is efficient use of the system resources by evenly loading all computational units of the system. It achieves that by not allowing individual resources to become idle or overloaded.

3. Unification

Unification allows for data stream analysis using queries. A stand-alone process can combine interactive streaming and batch requests.

SQL queries and analysis with Machine Learning

The development of systems with a common set of commands for database management is a necessary condition that realizes the interaction between individual analytical systems. SQL queries are widespread and accepted. Typically, systems provide libraries and machine learning modules that allow for more complex analysis.

Apache Kudu

In Apache Kudu, data is stored in tables similar to relational databases. In most cases, tables can be extremely complex, and other times they can be extremely simple. Similar to relational tables, each table contains a primary key. Using the primary key one can easily update and even delete records. An example of such a primary key in a table is an ID of a given user.

Kudu uses a simple data model which allows for easy data transfer and integration. Another great convenience is the fact that it is not necessary to use a binary format or to be familiar with JSON files. Tables have the ability to self-describe depending on the database system. This feature enables the developers to use standard tools such as Spark or SQL Engine.

We can say that Kudu is more than a file format. It has the power to store data in real time. NoSQL provides APIs for Python languages, Java and even C ++. Thus, it can be applied in data analysis and machine learning. The data model is applied in such a way that it is not necessary to even think about binary coding. This makes Kudu's API extremely easy to use. In case you have different types of data and subsets - Kudu, although not an OLTP system, is fully compatible with other systems.

Apache Kudu is open source, so you can easily link different information processing frames. Kudu can be integrated with the Hadoop ecosystem without any complications. It might not come as a surprise but Kudu also stores its data in columns instead of rows. Likewise most analytics repositories used in today's digital society store data in a similar fashion. That is because the column storage allows for more efficient compression and coding. Column storage reduces the amount of I/O data required to perform analytical queries.

A rational solution is to use a Java client for real-time data redirection. This makes Apache Impala, Map Reduce and Apache Spark a good solution for immediate

processing. What's more, you can even join a Kudu table to data stored in HDFS. The ease of use and opportunities offered by Kudu is a major choice when it comes to joining the Hadoop cluster. A distinctive feature of Kudu is that it can easily perform heavy memory operations of less than 1 GB. In addition, Kudu can execute drill-down and needle-in-a-haystack queries on millions of rows of data in less than a second.

Tables in Kudu are divided into smaller units called tablets. The partitioning can be done by configuring each table individually or by hashing. Kudu uses the Raft algorithm which ensures the preservation of information at any time, thus duplicates each operation for the respective tablets. This recording of each operation on a separate node ensures that the data will not be lost in case of machine damage. This action is performed before responding to the customer request. If, however, a machine fails, the replicated information is copied in a matter of seconds, thus keeping the system fully operational and creating a sense of security in the user that the given system is reliable. Raft Consensus makes sure that the replicated data will be consistent. This ensures that there will be the least possible delays even when some nodes are loaded with multiple parallel tasks (as in the case of Impala queries or Spark tasks).

Oracle Big Data Connectors

Oracle Big Data Connectors enables the integration of Oracle's RDBMS with data stored on large data platforms, including HDFS and NoSQL databases. Oracle Big Data Connectors provide an easy-to-use graphical environment without the need of writing a complex code. Oracle Big Data Connectors is a means of communication between Apache Hadoop and Oracle Database. Apache Hadoop can be initially used for data collection and primary processing. Afterwards, the large data can be linked to the enterprise data stored in Oracle Database in order to produce an integrated analysis.

The introduction of new technologies and techniques is always a challenge. The IT department should be expanded to include Big Data specialists familiar with the relevant technologies.

Conclusion

Nowadays, organizations have the opportunity to become faster, better and more efficient by adopting new approaches to developing their products. In order to adopt those changes, the management of the company must start to constantly adapt to the rapidly changing technologies. Some basic steps a company can take in order to successfully integrate Big Data:

- Launch simple and easy-to-implement Big Data projects.
- Expand the data sources used and explore the potential uses of internally and externally available data.
- Allow employees to receive data in real time.
- Use analytical Big Data applications and leading analytical initiatives both at the management and the operational level of the company.
- Develop strategies for the effective use of leading analytical techniques and technologies.
- Develop and maintain the staff's skill set in terms of Big Data.
- Build data visualization skills.

The skillful development and gradual implementation of Big Data analysis technologies will provide huge business opportunities. Innovation, of course, does not emerge and does not develop from scratch. Different industries and businesses design their data management architecture based on existing and future data sources they have or will have. Technologies, databases

and analysis tools are selected to serve legacy protocols and standards. The new architecture for Big Data analysis is built on the basis of the existing company architecture for data processing.

References

Edwards, Martin R. 2016. Predictive HR Analytics: Mastering the HR Metric.

Fitzenz, Jac, John R. Mattox, II. 2014. Predictive Analytics for Human Resources, Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Milev, Plamen. 2019. Integration of Software Solutions via an Intermediary Web Service. *Trakia Journal of Sciences*, issue 1, pp. 181-185.

Mihova, Veska. 2015. Methods of Using Business Intelligence Technologies for Dynamic Database Performance Administration. *Economic Alternatives,* 2015, issue 3, pp. 105-116, University of National and World Economy, Sofia, Bulgaria.

Radoev, Mitko. 2019 Using Microsoft SQL Server 2019 Big Data Clusters. In: International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE).

Radoev, M. 2017. A comparison between characteristics of NoSQL databases and traditional databases. *Comput. Sci. Info. Technol*. 5 (5), 149–153.

Tsaneva, Monika. 2019. A Practical Approach for Integrating Heterogeneous Systems. Business management, issue 2, D. A. Tsenov Academy of Economics, Svishtov, Bulgaria.

Tsaneva, Monika. 2019. General Data-driven File Export Integration Solution. In: Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE), pp. 154-159.

Apache Kudu Overview. Available URL https://kudu.apache.org/overview.html Date accessed [07.01.2021].

Apache Spark Streaming Tutorial For Beginners: Working, Architecture & Features. Available URL https://www.upgrad.com/blog/apache-spark-streaming-tutorial-beginners/ Date accessed [07.01.2021].

Overview of Apache Kudu. Available URL https://www.techntrip.com/overview-of-apache-kudu/ Date accessed [07.01.2021].

An Enterprise Architect's Guide to Big Data. Available URL (https://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf) Date accessed [03.11.2020].

Cloudera Documentation. Cloudera Enterprise 5.13.x Available URL (https://docs.cloudera.com/documentation/enterprise/5-13-x/topics/impala_file_formats.html) Date accessed [03.11.2020].

Optimized Real-time Analytics using Spark Streaming and Apache Druid. Available URL https://medium.com/gumgum-tech/optimized-real-time-analytics-using-spark-streaming-and-apache-druid-d872a86ed99d Date accessed [07.01.2021].