

Ministry of Education and Science of the Republic of Kazakhstan

A.K. Mukasheva

T.F. Umarov

I.A. Zimin

BIG DATA ANALYTICS

Almaty, 2021

UDC 004 (075.8)

LBC 32.973 я73

M92

Reviewers: D.N Shukayev, Doctor of Technical Sciences, Professor, KazNRTU named after K.I. Satpayev. D. Yedilkhan, PhD, Associate Professor, ICT program coordinator at Astana IT University.

A.K. Mukasheva., T.F. Umarov., I.A. Zimin:

M92 Big Data analytics: Textbook (for students of all specialties)/

A.K. Mukasheva, T.F. Umarov, I.A. Zimin – Almaty: Lantar Trade LLC

2021. – 116 P: 2 tab, 135 ill, bibliog. - 48 titles.

ISBN 978-601-7659-94-3

The textbook contains the relevance of the study lies in the use of modern processing approaches, the choice of modern tools and technologies for the creation of a data-processing infrastructure. At present, Big Data technologies allow for the retention of information over a long period of time, most importantly, Big Data Tools allow detailed analysis of received data and provide data for visualization by various tools.

The textbook is intended for trainees in the direction of training «061 Information and Communication Technologies».

UDC 004 (075.8)

LBC 32.973 я73

Recommended for publication by the Academic and Methodological Council for the Direction of Training «061 Information and Communication Technologies» (Protocol № 4 from 28.05.2021).

ISBN 978-601-7659-94-3

©A.K. Mukasheva., T.F. Umarov.,

I.A. Zimin., 2021

© Lantar Trade LLC, 2021

CONTENT

INTRODUCTION	5
1. RELEVANCE OF BIG DATA TECHNOLOGY IN THE WORLD	7
1.1 Definition of the term Big Data	7
1.2 Data types in the Big Data domain	9
1.3 Analytics of Big Data processing applications in all directions	11
1.4 Processes used in Big Data	12
1.5 Working with non-relational NoSQL databases in the context of Big Data	13
2. RESEARCH OF INTERACTION PROCESSES WITHIN THE HADOOP ECOSYSTEM	14
2.1 Key features of Hadoop technology	14
2.2 Application of Hadoop in various fields	16
2.3 Using Big Data to process Web server logs	17
3. SCANNING THE STAND FOR WORKING WITH BIG DATA	18
3.1 Virtualization infrastructure for Big Data	18
3.2 Proposed physical architecture for working with Big Data	24
3.3 Installing the Proxmox hypervisor on the server	28
3.4 Installing the operating system for the infrastructure	38
4. INSTALLING HORTONWORKS DATA PLATFORM AND PROCESSING WEB SERVER LOGS	44
4.1 Description of the operation of the log file processing system	44
4.2 Working with reverse proxy server logs and web servers	46
4.3 Installing and configuring the Hortonworks Data Platform for Web Server log processing	48
4.4 Shell deployment for interactive work in Apache Zeppelin	58
4.5 Analysis of Nginx Web server logs	62
5. DATA ADMINISTRATION AND PROCESSING IN A HADOOP ENVIRONMENT	73
5.1 Big Data analysis, processing and visualization	73
5.2 Data analysis using Apache Spark	86
5.3 Loading and processing data using the MapReduce model	91
CONCLUSION	99
LIST OF REFERENCES	101
APPENDIX A	

DEFINITIONS AND ABBREVIATIONS USED

BD	Big Data
SP	Software Program
RD	Relational databases
FS	File System
DFS	Distributed File System
OS	Operating system
VM	Virtual machine

INTRODUCTION

At the moment, technology is developing more and more every day, but with the advent of new technologies, peoples are also getting a huge amount of information that needs to be stored and processed in order to get up-to-date data in advertising, education, medicine. With each passing year, there are fewer and fewer areas where technology is not available. The incoming information needs to be structured, stored and processed in order to be able to further analyze the data and, on the basis of this analysis, make sound and accurate forecasts for future efficiencies. Big Data (BD) technology has become such a system to handle huge amounts of data. The BD provides technologies and techniques that enable the processing, collection and analysis of vast amounts of information. They are also large volumes that increase exponentially. The structure of sufficiently large standard tools does not allow the processing of such data. The BD allows the processing and collection of large amounts of data through the use of software that has been developed to operate and store complex sets of different data. A list of the variety of data that are used to work with the database:

1) Structured data are data that are located in many traditional databases. Structured data has a clear structure, is easy to store, does not require additional processing. Such data are easily stored using relational databases (RDB) and they use special formats such as excel, json, xml, CSV;

2) Unstructured data are data that do not have a clear format and may not be easy to store in RD. Unstructured data are found on a variety of resources, texts from different sources, information from different social networks, video and audio sources refer to unstructured data. Such data should be pre-processed and provided in a format suitable for further analysis;

3) Semi-structured data - Such data represent a symbiosis between structured data and unstructured data, and some data types contain properties that define them as semi-structured data. For example, mail messages and their content can be defined as unstructured, but at the same time postal messages contain data such as name, e-mail address, date of dispatch and receipt of electronic messages, these data can be defined as structured. Therefore, these data refer to both types of data and belong to the semi-structured data.

Web server logs are unstructured data, pre-processing of log data is required for correct work. This requires a system that effectively collects large amounts of data and processes them for further analysis. The purpose of this study is to investigate existing BD products, select the optimal software for processing web server logs, measurement data processing speed in a distributed file processing system, and obtain data visualization, in order to get a prognosis about the possibility of improving the existing infrastructure or fixing the security problems of web resources and web servers. The objectives of the study were as follows:

- search for BD Information
- search for existing solutions for BD as well as software for distributed data processing;

- installation of data processing infrastructure in a virtual environment;
- development of a program for processing web server log data in the distributed data processing system.

The novelty of the study is the development of a system for processing and storing large amounts of web server access logs for analysis and web resources statistics, and forecasting on the basis of data analysis of the need to improve the existing infrastructure or improve the security of existing web servers. A study was also conducted at the speed of data processing in relation to the volume of information received in the distributed processing system.

The theoretical importance of the study lies in the possibility of analyzing a large amount of information and storing data in a distributed file system, the possibility of processing information at a high speed due to the use of distributed information processing tools. On the basis of this study it is possible to process data from access logs of many popular web servers in order to obtain statistics on site performance and web portals.

The practical significance of the study is that it is possible to predict, on the basis of the processed data of web server logs, problems in the operation of the existing infrastructure. As well as get a full picture of the work of the organization's web servers and the possibility to get a visual result for the analysts of the work of the sites and portals of the university or other organizations that need an analyst of the work of their web decisions.

1. 1. RELEVANCE OF BIG DATA TECHNOLOGY IN THE WORLD

1.1 Definition of the term Big Data

The term BD has spread throughout society and is used in a large number of areas. Data is constantly being created and increasing with increasing speed. One of the difficulties in dealing with BD is the analysis of a large amount of information, which is due to the fact that the following factors have to be taken into account in BD analysis: speed of data processing, volume and diversity. Data must be collected, and many companies now use different tools to store their data. The BD provides new opportunities for organizations to extract the necessary information and create a competitive advantage from their most valuable information asset [1]. Large companies quickly realized that BD technology with the right amount of data allows to predict, using analytical tools, that it is profitable to put everything into mass production. To make sales forecasts and to create different marketing strategies based on the predicted results. For data storage, it was necessary to develop tools capable of storing large amounts of data and to scale up the repository quickly. In such cases, Hadoop tools with HDFS file system are commonly used. HDFS allows storing large amounts of data and easily scaling file space by quickly adding new servers to the cluster. The main solutions so far are cloud storage, which, in addition to storing data, also cleans data to better obtain further results [2].

Data are diverse, including text, unstructured data, email data, video sources, structured databases, and various data from it devices. For example, based on users' e-mails, the amount of different data, the text of the letter is structured differently, and the data of the e-mail may include the time of dispatch and receipt, as well as the addresses of the recipients and of those users, which are also contained in the copy. Another good example is the various videos. The video can be recorded on various fragments and make it possible to get the date of the network from many videos where can get a certain fragment. For example, collect the date set from the video fragments of people walking down the street. These examples illustrate the variety of information that needs to be collected and processed. For example, there is a large company that produces content every second. One example is Instagram, which generates a huge amount of content per second, which needs to be processed and systematized and done as quickly as possible. This requires traffic management systems as well as the ability to maximize the return on investment in such technologies. The main distinguishing features of the speed in the BD are the continuity and huge data flow. During the digital age, data are collected at an incredible rate of 2.5 quintillion (2.5×10^{18}) of data bytes generated every day. There are now many companies that sell and collect their data. There is now a market that provides equipment, software and various BD processing services, and this market continues to grow daily [3].

Databases are important in a variety of organizations, and the use of such a tool in various large companies proves this, such as Facebook, Google, WalMart Twitter. The listed companies use the database to predict various models on the

purchasing market, to determine the success of a particular sales model. The application of visualization also provides a broad overview of the results of the BD analysis. The use of analytical tools in the BD management technology allows for a large amount of investment [4]. For example, consider the experience of Walmart, which processes over a million customer transactions every hour and imports them into databases. It is estimated that these databases contain more than 2.5 petabytes of data. The company can combine data from a variety of sources such as: past purchases of the customers, the location of their mobile phones, internal Walmart stock control records, social media and information from external sources, such as weather, and initiate individual advertising actions. For example, if a customer was shopping for a barbecue at Walmart, and the customer happened to be within a 3-mile radius of Walmart, which has a barbecue cleaner in its warehouse, and the weather is sunny, the customer can immediately receive a voucher with money from a barbecue cleaner, delivered customer smartphone [5, 6].

In a rapidly developing world where economic sector volatility is high enough, various companies as well as government departments have begun to reflect on how the situation might be affected. One such solution is to work with BD, which requires a sufficient amount of data on the basis of which it is possible in the future to make any predictions, determine the attractiveness of particular resources, and to project best practices and strategies [7].

The problem of scalability of large amounts of data is relevant in all fields, from medicine to the humanities. In medicine, BD also takes the form of medical photos, cardiograms, and various images that occupy petabytes of memory in the system. In the humanities, it can take the form of books, various electronic journals and other heterogeneous information [8].

There are a number of studies that allow the use of BD for medical results, as well as different systems in smart houses and cities. If look at certain areas where BD is used in some way, that is, a whole layer of different areas.

When considering the BD in the sports sector, there is a widespread practice of searching for talented athletes based on the analysis of statistics and records of certain athletes, which allows to determine the fitness of an athlete of the organization. Then there is a legitimate question as to how a BD is collected in a sport industry, for example, the presence of a special tracker on an athlete to provide data on the athlete's condition, speed and various other parameters. This is a fairly large amount of data and needs to be collected and stored, and working with BD provides such opportunities. Using BD technology to calculate the results of certain sports and to predict the future success or loss of a team. The collection of information about players, athletes and the team in general makes it possible to avoid over-training of athletes, the ability to collect data on heart reductions makes it possible to prevent various injuries and chronic diseases. Similar technologies exist in many sports disciplines such as football, big tennis, and cycling [9, 10].

Also, the development of DB in sports allows ordinary people, not athletes, to use the DB for benefit, and now the market is filling up more and more smart

devices that allow to measure blood pressure, monitor the body's calorie levels, monitor calorie intake, as well as sleep control. Another equally important factor in the use of BD is that all devices can synchronize and transmit information to a single BD processing center, where the user will be provided, in a convenient form, with all the necessary results about his well-being and physical condition. Imagine that fitness bracelet synchronizes with the tracker on bicycle and provides information about the distance and the amount of calories burned. Smart scales synchronize with mobile phone and show predicted data. Thanks to these technologies can see how can lose weight if continue to ride the bike at the same speed and at the same distance, all of this is possible thanks to the database.

Smart houses also generate a lot of data. Heating systems, various automatic air ventilation systems, power control. For example, a special heating system heats only the areas where the person is located or the most necessary areas in the house. It is also possible to adjust the thermostat automatically so that when a person arrives home, the system is already heating the whole house. Of course, this smart house technology also generates huge amounts of data, and for such systems also need to process and store huge amounts of data, Hadoop allows to create such an infrastructure to collect this information and further its processing [11, 12]

1.2 Big Data data types

Most BD are unstructured data, including images, text documents and web logs, and are stored in unprocessed form and retrieved detailed information where necessary [13]. There is a large amount of information in the BD that needs to be processed. These data include: unstructured data; audio data; video data; structured data; various log file data; stream data [14].

Audio and video data are complex enough to process and find certain results, for example, to get a certain object on an image or video. This requires a record of very good quality and the system must be able not only to place the data on the server, but also to pre-process the data. Finding any text in a video stream, or on a certain image. This requires that the image itself be pre-divided into multiple blocks and that this information be further transferred to the neural network to receive text from the image [15].

In today's world, a very popular industry is the game development industry. The amount of data that network games generate is very large, it is increasing every year as users consume the content they receive very quickly, such as popular MMOGs (Massively multiplayer online game is a multiplayer online game). These players create a very large amount of content that needs to be stored and processed. Conventional RD cannot meet the storage requirements for this amount of information, and various non-RD and BD technology are very good at storing this type of data [16].

Structured data refer to the presence of data of a certain structure, usually a fixed value, and such data can be easily stored in a convenient form, for example in excel, json, xml, csv. It is also easy to store such data using, for example, RD, but there are various data that are difficult to store, let alone process in a relational data

model. Such data types include hybrid data, such as that generated by man and machine, and include:

- log data, which is generated in the process of collecting log files on the state of servers, networks, web pages, as well as access to web servers;
- financial data is also one of the biggest sectors in the generation of structured data, such as trading data on a stock exchange, data change at a high rate and are generated in seconds in the financial sector;
- data that are generated by different sensors to collect information, such as Geo-positioning tracking systems or identification tag readers. Such data are available almost continuously and in very large quantities.

Such data creates a huge stream that is difficult to process with conventional databases and uses BD tools to process them.

Unstructured data are data that do not fit a particular format and may have a completely different format and not be easy to store. Unstructured data are found almost everywhere, with 20 per cent of structured data accounting for 80 per cent of unstructured data. Unstructured data are generated by both users and equipment.

Various photos and videos refer to unstructured data, various smart city management systems, traffic management systems, and video surveillance systems. Data from various seismic equipment, imagery from space stations such as Yandex maps or google maps generate a huge amount of unstructured data. Various printed publications such as magazines, newspapers, various articles on electronic media, a very large amount of information is contained in different texts, it is one of the most convenient communication systems and it is not clearly structured, because storage or processing requires the use of recognition and classification solutions. Also, unstructured data include data from various social resources and electronic platforms. Such platforms include Instagram, YouTube, Facebook, Twitter, etc.

Unstructured data is the largest part of the data, and the amount of this type of data will only increase each year, develop and fill different applications and systems. In recognition of, for example, textual information, this is an unqualified advantage as the market for faster search applications with BD tools can be developed. Already now, various systems have been set up on the basis of the received data and their processing for consulting clients by a clever assistant. A clever system based on multiple text messages learns how to answer client questions, and this information is used in analyzing data when staff is interviewing clients' call centers, when analyzing written comments.

As the culture of content consumption has changed, users are consuming more and more information, allowing companies that work with BD to increase their development of BD as well as to develop systems capable of handling unstructured data. A very large amount of unstructured data refers to different streaming platforms, streaming Internet radio, or different content generation systems on social platforms such as Twitter, Yandex, Facebook, etc.

Log file data (journals) are usually stored in different formats and cause significant problems. Also, because most of the time a system is running, log data are becoming the biggest source of data on the server. Such data should be

transmitted and stored efficiently, but here comes the new problem of not being able to efficiently process such a volume of information. It is clear from the above that if a company receives an incident that disrupts the system or a break-in incident, it is first necessary to obtain data from the logs. All valuable information on the incident is stored there, and companies generally do not care to keep the logs properly. Especially to use the system to visualize or effectively store data. And even if find this file, will not be able to work effectively with it if the file is not segmented or weighs on the order of several terabytes. Here comes the help and the BD management system, which will help to segregate the data as well as create a convenient visualization for further investigation of the incidents, or to perform analysis to understand the problems.

Data streaming - data streaming is data that is received continuously from a large number of different equipment or systems, with no beginning or end, as the incoming data flow does not stop for a second. In general, streaming is used when the actual portions of the data do not exceed certain sizes, such as weather sensors, various readers or e-commerce systems. Data types such as streaming generate data flow, which is most often used to correlate data among themselves or to filter data in real time. It is also important to process such data in memory, and because the BD tools make it possible, using a cluster association, to build a fault-tolerant system capable of processing data in memory. This significantly improves the efficiency of such a system when processing streaming data.

1.3 Analytics of Big Data Applications in all directions

In the modern world, BD Application Analytics is a process of studying large datasets that contain different types of data. Databases allow to detect all hidden patterns, unknown correlations, market trends, customer preferences and other useful business information and more. Analytical results can then lead to more efficient marketing, new income-generating opportunities, quality customer service, improved performance, competitive advantage over competing organizations and other businesses-advantages [17].

BD has become a central theme in various endeavors and research. This is because the ability to create, collect, transmit, prepare and examine exceptional measures of difference in the information is almost universally applicable. BD, with its potential to provide valuable information for improved decision-making, has recently attracted considerable interest from both scientists and researchers. The BD analyst is increasingly a popular practice used by many organizations to obtain valuable information from the BD. The analytical process, including the deployment and use of BD analytics tools, is considered by the organizations as a tool to enhance operational effectiveness, although it has strategic potential, Generates new revenue streams and competitive advantages over competitors [18].

Analytics can be divided into 5 types:

- descriptive analyst: The simplest research class is the one that allows to combine a huge amount of information into smaller, more valuable portions of data [19];

– predictive analyst: The predictive analyst may be the most commonly used category of data analyst, as it is used to identify trends, correlations and cause-effect relationships. The category could be further divided into predictive modelling and statistical modelling [20];

– prescriptive analyst: In addition to descriptive and predictive analyst, the prescriptive analyst is one of the three main types of analytic company used for data analysis. This type of analytics is sometimes described as a form of predictive analytics, but differs slightly in its focus. The objective of the prescriptive analyst is to provide the best possible advice for a situation that is unfolding, given what the analyst can infer from the available data [21];

– diagnostic analyst: This is a form of extended analytics that analyzes data or content to answer the question «Why did this happen?» and is characterized by methods such as detail, data detection, data analysis and correlation. The diagnostic analyst examines the data in greater depth to try to understand the causes of events and behavior [22];

– cognitive analytic: Cognitive analytics combine a range of intellectual technologies, such as artificial intelligence, machine learning algorithms, deep learning, etc., to apply human intelligence to certain tasks. Basically, this type of analyst is inspired by the way the human brain processes information, infers and systematizes instincts and experiences for learning, such as understanding not only words in the text, but the full context of what is written or said. All of these intellectual technologies make cognitive applications smarter and more efficient over time, learning from their interactions with data and people [23]. Companies such as Google and Amazon are masters of BD mining and analysis. They use the knowledge gained from BD analysis to gain superiority over their competitors.

The system analyzes databases such as purchase history, purchasing habits and purchase patterns. Using BD and forecasting analyst, they created a marketing machine and created a highly successful business model. With increasing computing capacity, reliable data infrastructure, rapid algorithm development and the need to gain a better understanding of the ever-increasing amounts of data, enterprises are seeking to use a BD analyst as part of their decision-making process. Decision makers have realized that, with a better understanding, an excellent competitive position can be achieved [24]. Organizations from different domains invest in BD applications to examine large data sets to identify all hidden patterns, unknown correlations, market trends, customer preferences and other useful information. The scope of applications of database applications includes: production, health, media, Internet of Things Government, etc. [25].

1.4 Processes used in Big Data

Data science should also be considered when working with BD. This area allows to systematize, classify and give a clear answer to how the data collection process, its further processing, and its storage in a robot database environment is carried out. There are the following main processes to be followed in dealing with data:

1. Understanding Business Processes - First of all it is necessary to define what goals and tasks need to be solved with the data processing process, what exactly needs to be improved, or to create a completely new approach to data handling. It is the understanding of what processes need to be improved that makes effective use of data science in a particular area;

2. Data collection - at this stage it is necessary to obtain all the data that exist in the enterprise or in the organization. At this stage such data sources can be, for example: different databases MySQL, MSSQL, PostgreSQL; different data formats such as excel documents, csv and json files, etc. etc. Data collection also requires the use of special tools, such as different frameworks, or different API libraries for access to data. It can also be used to parse data formats csv (comma-separated) or tsv (tab-separated data).

3. Data Cleansing - to work with data it is necessary to clear them, the process of data cleaning allows filtering them. The cleaning process allows the data to be put into one particular format, and the data can already be in a certain format, such as json, but may be from different sources, in which case it is necessary to combine the data into a single database, so that further analysis of such data would be possible. Similarly, when cleaning data, the ability to replace empty values when there are missing values in a data set is an important factor and can be replaced by a standard or other predefined set. In certain cases, it may be necessary to replace data, merge or delete certain data fields or columns. Tools such as Hadoop, Spark, Yarn are well suited to handle large amounts of data.

4. Data Research - The examination of the data provides an insight into which operations will need to be carried out in the end, which types of data are used, such as textual data types or numeric data are processed differently. There is still a need to conduct statistical checks on their characteristics. For example, to determine whether data influences the performance of other data in statistics;

5. Data modeling - Modeling is a step that allows me to use prediction to get the desired result, and at this stage a large number of different algorithms are used to work with data. Such algorithms include k-means, clustering algorithms;

6. Data presentation and interpretation - this stage allows data to be presented to non-technical staff and specialists for further examination. The data provided will determine at this stage whether the organization has achieved the required result in data processing. Also at this stage, it is necessary to visualize the data so that all stakeholders correctly interpret the result and can draw conclusions in accordance with the visible visualization.

7. BD infrastructure - there are currently a wide variety of BD solutions. New implementations and products appear in large numbers on the IT market, all major components can be divided into several categories, which allow more detailed understanding of which parts of the infrastructure are responsible for which functions.

1.5 Work with non-relational NoSQL databases from Big Data

The Distributed File System (DFS) is a data warehouse on multiple servers that are combined into clusters. This file system (FS) is similar in its work to other FS except that it is online. The main advantages of such a system are the ability to store huge amounts of data, also the ability to scale the system in unlimited quantity if it is necessary to increase the FS space enough to add a new node to the cluster. The DFS provides reliable data replication systems on several cluster servers for high fault tolerance, which substantially increases the reliability of storage of such data. The second factor that is used to replicate the data is the ability to process them in parallel to achieve the maximum result. The DFS provides the best storage of data along with the others, as even if several nodes of the cluster are disabled, the files will still be available because replication is performed and failure of one of the participants does not cause loss of data.

Storing DB in a DFS is the best approach, as previously it was necessary to use vertical scaling, that is, the deployment of a more powerful server for data storage, to increase storage space. In this work, the HDFS (Hadoop File System) included in the Hadoop toolkit will be considered as such FS.

Many NoSQL companies, such as Google, Amazon and Facebook, have been involved in the development of NoSQL technologies. For a large amount of data, it is not always good to apply a relational approach to data storage. There are tasks that are difficult to implement when working with a traditional approach to solve database tasks, this requires a different approach and therefore a database structure such as NoSQL has been developed. This architecture allows to scale databases due to the cluster architecture of data storage. A relational approach is well suited to such tasks as collecting analytical data in the same format. Data that have a certain structural form and if all the data in the schema of the database will be a certain identical set, rows and columns, then RD should be used. In the same case that there is a need to work with a variety of heterogeneous information, unstructured data, or huge amounts of information that cannot work correctly with other relational models, NoSQL database is best suited, Databases using the NoSQL approach are not bound to a fixed storage scheme

Initially, RD was not designed to work with the huge amounts of data that are now being generated globally at a very high rate. There was no flexible scaling system, as RD was initially considered to be a convenient storage facility with the least computing capacity. With the emergence of rapidly growing information in today's world, a huge amount of unstructured data that is generated every day, and growing every year, needs to be processed with high speed and scalability, such qualities NoSQL databases possess, High productivity, scalability and real time.

2. STUDY OF INTERACTIONS WITHIN THE HADOOP ECOSYSTEM

2.1 Essential Features of Hadoop Technology

The Hadoop project allows to process large amounts of data with the help of various tools supplied by the foundation of Apache Software. The product was developed by Hugh Cutting and Mike Kafarrell in 2005. Many companies use Hadoop tools in various areas of business, science and education. The list of companies that use Hadoop tools include Facebook, Netflix and Amazon. Company data use Hadoop to analyze unstructured data, Hadoop equally handles structured data as well as unstructured data [26].

Hadoop allows processing of a very large amount of information, which includes images, video, audio, files, software and more. Hadoop uses many components, including Flume, HBase, Hive, Lucene, Oozie, Pig, Sqoop and Zookeeper [27].

The Hadoop ecosystem has two main components in its system:

1) The distributed file system HDFS. HDFS provides fault-tolerant operation as well as operations on various equipment. The HDFS structure allows storing data on a large number of servers. The distributed FS uses the master and slave model. The information that enters HDFS is then broken down into information blocks, by default each block is 64 megabytes [28]. Compared to other FSs: FAT, NTFS, where each block of information is between 4 and 32 KB, HDFS partitions them into larger blocks, and this parameter can be increased in configuration files if necessary. Each block created receives a special number «blk_XXXXXX, XXXXXX» - this number changes depending on the size of the block. All data placed in the cluster is stored on certain machines located in the cluster, and these nodes have the name of a data node. A metadata repository for data that has been placed in a cluster is located on a node named Name Node. An additional fault tolerance is the replication of incoming data on multiple nodes in a cluster, and these nodes can be located at different locations in the cluster. The default replicas number is 3 [29].

2) MapReduce is a distributed computing model. The MapReduce programming model that Google developed in 2004, was needed to write applications capable of handling large amounts of data through distributed computing, on a multitude of cluster servers. MapReduce is a platform for processing and managing large amounts of data in a distributed cluster that is used for applications such as search indexing, document clustering, access log analysis, and various other forms of data analysis. One of the advantages of this system is the ability to process large volumes of information by dividing them into segments(s) and transferring them to computation by different cluster servers for faster processing. MapReduce does not divide two steps when processing information:

1) Map - This function always starts first, then filters, then transforms and analyzes information. All received data is transferred further to Reduce.

2) Reduce - This function is used to add up data after processing Map.

During the operation of MapReduce with the main server (master), a map function is called, and the task is shared among the machines in the cluster. The map function then transforms the received data set, and then transfers the data to the Reduce function. Furthermore, Reduce combines the resulting data sets into smaller tuples. The Map and Reduce functions are performed as many times as the application itself [30].

Apache Hadoop Yarn is a system for distributed data storage and analysis in the Hadoop environment. Yarn allows storing data on cluster system servers, is an improved version of the MapReduce framework (MapReduce V2.). This system allows for real-time streaming as well as SQL on-line processing of data. Yarn significantly increased the speed of data processing compared to the previous version of MapReduce V1, and opened the possibility of processing data from various sources such as sensor data analysis, scientific and medical data, and social networks [31].

2.2 Application of Hadoop in various applications

One of the most sought-after areas is medicine. Hadoop tools are used very effectively in areas such as health care and medicine, allowing large amounts of data to be processed. Hadoop tools allow to improve medical service, forecast different outcomes at different course of disease [32]. Hadoop tools are also used to solve complex problems such as storing photos and images for ophthalmological research and analysis. The use of HDFS, which is part of Hadoop tools, has increased the speed of data writing and further processing [33].

Use of sensors for real-time diagnostics to provide a complete picture of diagnostics. This technology consists of 4 stages of operation: real-time patient monitoring, patient systematization, patient diagnosis, visualization of processed data, treatment recommendations. In this system, Hadoop [34] tools are the main tool for solving diagnostic problems.

Hadoop tools in bioinformatics and cancer markers are also important. Hadoop tools allow the processing of biological and biomedical data at very high speed, also allow correlation of data to find the optimal result in the study [35].

The processing of incoming images for further data storage and processing is increasingly being used for medical imaging. Medical images, various cardiograms and MRI images are very diverse and unstructured information that needs to be stored and properly processed. Medical imaging systems are usually stored in a database, and one of the main tasks is to quickly retrieve exactly the images needed for diagnostic solutions. It is only when an effective data storage system is in place that these data can be easily accessed on a large scale. Such a system is an open Hadoop platform with the ability to run in parallel mode. The MapReduce operating model created a model capable of extracting necessary characteristics from images and transmitting them to the HDFS [36] data warehouse.

There are also many examples where Hadoop technology has been applied to smart cities. Using Hadoop tools for storing incoming information from many

different sensors, smart house control systems, home power control systems. Because smart house systems generate a large volume of different structured, semi-structured, and unstructured Hadoop information, they can effectively collect and group this data. One of the many projects in the development of smart cities is the use of traffic control technology, with cameras recording various violations, generating a huge amount of video information, a system capable of storing this amount of information is necessary for its storage. Hadoop infrastructure tools allow the creation of such a technology [37]. Hadoop is also used to construct a system for collecting information in sports. One of the areas in which Hadoop tools are involved is cricket. Hadoop and Hive are used in cricket to predict and statically model the system for selecting players based on criteria such as statistics, different results depending on the team, The construction of such a system provided 91% of the result of selecting the right player for the team, depending on the necessary skills [38]. Hadoop tools are also used in the design of systems that use different sensors and temperature measuring devices. The project includes ambient temperature monitoring. Work in such a project shows how effectively Hadoop works with IoT devices. The sensors are the visible light from LEDs, which are read by various surveillance cameras and other devices. After that all data is collected in the cluster and due to flexibility and easy processing by means of the Hadoop cluster, the speed of obtaining results increases. The system reduced errors in temperature reading from 200 nodes from 5% to 3% [39].

2.3 Use Big Data to process web server logs

The use of web server logs to monitor portals and sites is an important element in the design of a reliable web resource, but there is also a need to use analytical tools to obtain statistics on the long-term performance of sites and portals.

Different sites generate huge amounts of data from log web servers every day. The problem with analyzing journals is the heterogeneous structure of the data. Web server log processing will consist of the following steps:

- loading web server logs into the HDFS file system;
- log data analysis using Apache Spark;
- data visualization in Apache Zeppelin.

Apache Spark is an open source BD platform based on speed, ease of use, and sophisticated analytics, and Spark has several advantages over other Big Data processing technologies. Apache Spark provides an integrated unified platform for managing BD processing requirements with different data sets, such as text data, graphics, and real-time batch data streams. Using Apache Spark, the web server log data will be divided by regular expressions into groups, and then the data will be processed and accessed using Spark SQL. Spark SQL is a relational data processing approach that allows the use of queries to obtain data. Apache Zeppelin allows to visualize the data that will be processed with Apache Spark. Using the relational approach in the presentation of data, an efficient data processing and visualization tool can be obtained, allowing a detailed report on what is happening

on the web server and portals located on the web server. The web server log is the following structure:

1) Web server log contains information about the IP address from which the request came. Thus, instead of an IP address, it may be the name of a network node, i.e. the web server log specifies the time when the request was made and the date is specified. The log contains the server response code and the amount of the packet sent, the name of the file that the user accessed, which the user accessed, and how the user used the browser when accessing the site;

2) Web server file contents are queries made with the HTTP protocol, one line in the web server log contains one HTTP request that was made when accessing a web server;

3) Web server log is a text format with a semi-structured data type.

Figure 2.1 shows how to store data in the NGINX Web Server Access Log File. The beginning of a new line is indicated by the following IP address from which the portal or site was connected.

```
1 2.78.223.205 - - [27/May/2020:00:00:06 +0600] "GET /css/fonts/roboto-light/Roboto-Light.ttf HTTP/1.1"
2 200 122232 "https://www.almay.edu.kz/" "Mozilla/5.0 (Linux; Android 9; SM-A505FN) AppleWebKit/537.36
3 (KHTML, like Gecko) Chrome/81.0.4044.138 Mobile Safari/537.36"
4 2.78.223.205 - - [27/May/2020:00:00:06 +0600] "GET /img/slider/distance2020.png HTTP/1.1" 206 1
5 "https://www.almay.edu.kz/" "Mozilla/5.0 (Linux; Android 9; SM-A505FN) AppleWebKit/537.36
6 (KHTML, like Gecko) Chrome/81.0.4044.138 Mobile Safari/537.36"
```

Figure 2.1 - NGINX Web Access Journal

Web server log processing will use two approaches to analyze data, in both cases data will be processed using python and the Apache Spark Distributed Data Processing Tool. In the first case, the data will be processed and placed in the spark.sql model, which will allow data processing and analysis using the query language. The Apache Zeppelin tool will use embedded data visualization models for data visualization. Completed queries in SQL will be automatically visualized and will provide detailed data. In the second approach, data will be processed and visualized using various python programming language methods and tools, and the following tools will allow the rendering and processing of pandas, numpy, matplotlib, seaborn data. Regular expressions will be used for data processing, which will allow to break up web server logs into sub-logs and then add them to the data table.

The resulting data processing model can also be used to handle data with the same structure of logs of other web servers, and if necessary, the processing code can be refined with additional regular expressions to find certain values.

3. EXTENSION OF THE BIG DATA STAND

3.1. Virtualization infrastructure for Big Data

The main purpose of this research is to adjust and develop universal infrastructure of learning to work with basic tools for working with BD. Such tools typically include Hadoop and its component tools such as HDFS, MapReduce,

Spark, Yarn, etc. One of the top-level Apache projects is Spark, which also focuses on parallel processing in the cluster, but the big difference is that it works in memory. Hadoop reads and writes files to HDFS, and Spark processes data in RAM using a concept known as Resilient Distributed Datasets (RDD)», a stable distributed data set. Spark can work either offline, with the Hadoop cluster as the source of data. Spark and Hadoop are open-source Apache projects, and Hadoop requires more disk memory, and Spark requires more RAM, which means that setting up Spark clusters can be more expensive. Spark is designed to improve, not replace, the Hadoop stacks. Spark was designed to read and write data from HDFS, as well as other data storage systems such as HBase and Amazon S3. In this way, Hadoop users can expand their processing capabilities by combining Spark with Hadoop, MapReduce, HBase and other BD frameworks [40]. There are three ways to deploy Spark in the Hadoop cluster: Standalone (Standalone), Over Yarn, and Spark in MapReduce (SIMR), as shown in Figure 3.1.

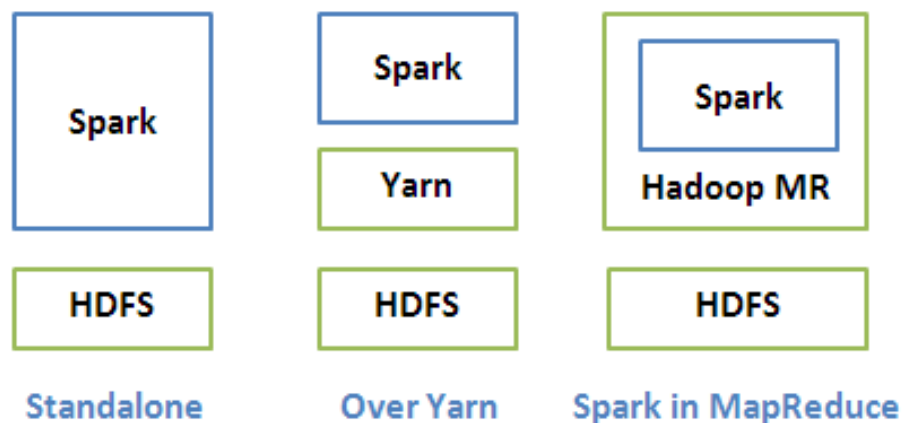


Figure 3.1 - Three Ways to Expand Spark in the Hadoop Cluster.

With autonomous deployment, can statically allocate resources to all machines in the Hadoop cluster and run Spark simultaneously with Hadoop MapReduce. The user can then perform arbitrary Spark tasks on their HDFS data. And can also run Spark on Yarn without prior installation or administrative access to develop Hadoop Yarn. This allows users to easily integrate Spark into their Hadoop stack and take advantage of Spark's full power as well as other components running over Spark. SIMR is used to run Spark tasks in addition to offline deployment. With SIMR can start working with Spark and use its shell without any administrative access [41, 42]. Apache Spark has extensive data processing tools and all of these factors make it possible to use as an efficient tool for parallel processing of a lot of data in the Hadoop cluster, and to increase processing speed.

To create a cluster management system, it was decided to create an infrastructure based on the Proxmox hypervisor. This hypervisor has a number of advantages, such as speed of operation, the ability to use it on virtually any server

equipment, as well as its ability to merge other servers on the basis of the Proxmox hypervisor into one fault-tolerant cluster.

It is necessary to understand what infrastructure is required during IS design. A clear understanding of the whole ecosystem is needed to use the BD platform. A large number of cloud platforms offer ready-made solutions for operation, but such systems are usually quite expensive. Also not a minor problem is the management of the physical machines on which the system cluster works, usually this problem is solved by the use of virtualization approach, one such tool is Hypervisor.

Review of the interaction of virtualization with BD processing techniques that both technologies offer, BD and virtualization create distributed technology that optimizes performance through flexibility in infrastructure management [43].

Virtualization can solve a number of problems when creating a system for processing large amounts of data. The possibility to deploy many machines on one server, simplicity in the control of cluster machines and creation of backups by means of virtual environment. Virtualization of servers allows splitting a physical server into several segments of small servers, and subsequently clustering them [44].

Virtualization plays a major role in the development of database infrastructure. Because of the very large amount of incoming data, finding an optimal solution in terms of cost and fault tolerance are very important factors in the implementation of the BD management system. Virtualization makes it easy to scale and manage infrastructure. The use of virtualization has a number of management advantages, due to the ability to install many machines on the server. Possibility of flexible management of virtual infrastructure, use of flexible approach for creation of backups. The BD operation requires support for various operating systems, with the ability to deploy them quickly and to produce a large stack of different operating systems (OS), as well as various additional components. The hypervisor allows flexibility to deal with such database infrastructure conditions. The hypervisor has a number of advantages, such as the ability to work with the OS as with applications, as well as the ability to instantly deploy a virtual machine (VM) to work with everything needed, with Software and OS as well as machine state imaging technology allow to go back to the past state of the machine before its failure or to another version of the configuration that was previously configured. The hypervisor also allows for easy sharing of hardware in real time between VM. If necessary, it is possible to change the amount of resources allocated to the machine if it is not necessary to use a large amount of hardware resources.

An important factor that demonstrates that the use of virtualization in the design and implementation of high-volume data technology is significantly better than the use of a server without managing a virtual environment. The virtual environment makes it possible to significantly reduce the cost of purchasing equipment, makes it possible to more efficiently manage the infrastructure due to the possibility to create new VM, without shutting down or reloading the physical server. Small infrastructure can first be deployed and further added, and the

capacity of existing infrastructure can be built up incrementally. MapReduce mechanisms also work much better in a virtual environment, as using virtualization it is possible to configure a cluster infrastructure separation where MapReduce tasks will work much more efficiently due to distributed load between nodes, Thus, overhead costs for infrastructure maintenance are significantly reduced.

A study on virtualization for a BD has shown how it is possible to build a virtualization infrastructure not only with a hypervisor, but also using systems such as containerized virtualization, as well as the use of orchestration facilities for such services with the possibility of creating cloud infrastructure to work with database [45].

Virtualization also improves speed between VM by using a virtual network instead of using a physical network. For example, when machines are installed on a single server on a hypervisor and are connected by a virtual network with one another, the network infrastructure speed is much higher and more efficient, because there are no foreign UVs on the network, or they are separated from one another. In this way, it is possible to effectively separate the necessary parts of the network from each other, as well as to create types of networks with the set of characteristics that are necessary when constructing the system. The virtualization of the network makes it possible to build flexible configurations to create distributed systems for handling BD. A hypervisor or or a VM control system allows to create, configure and control a set of operating systems isolated from each other on a physical server. The hypervisor, in turn, consists of two types, the first and the second type.

The first type of hypervisors (autonomous hypervisor) in the figure (3.2), this type of hypervisors is most often used to run directly on server equipment, and provides server resources to host the WM. Hypervisors include such hypervisors as: «Xen», «Hyper-V», 2ESXi».

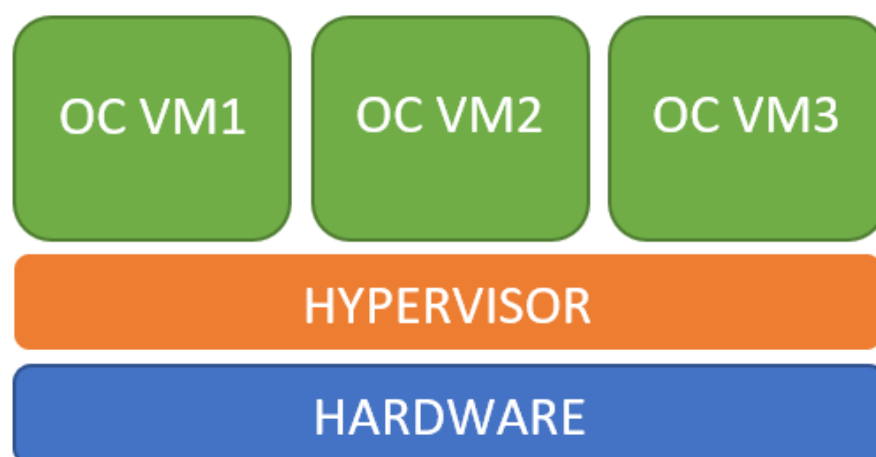


Figure 3.2-Hypervisor of the first type

The second type of hypervisor (hypervisor for guest operating system) in the figure (3.2) relates to software that is installed over the user's OS and the

hypervisor separates the user's OS from the hypervisor OS. Such hypervisors include: «VMware Workstation», «VirtualBox», «KVM», «Proxmox». Hypervisors of the second type allow to work on the surface just as these hypervisors have the same amount of functionality as hypervisors of the first type, they allow to create various VM backups. Many hypervisors of both types have built-in the ability to create a backup, as well as a snapshot of the VM state. At the same time, a large number of formats are supported for convenient migration of VM to new hypervisors, or hypervisors of other companies, which significantly reduces the delays in operation when the system migrates to new platforms and hypervisors. All hypervisors also support the ability to configure networks between machines, using virtual switches, and some hypervisors have built-in firewalls to limit and protect incoming and outgoing traffic.

Type 2 hypervisors will be used to build the infrastructure. From a virtualization perspective, it is also possible to indicate which systems were considered for work with virtualization. Such two hypervisors as «Xen» and «KVM» were considered.

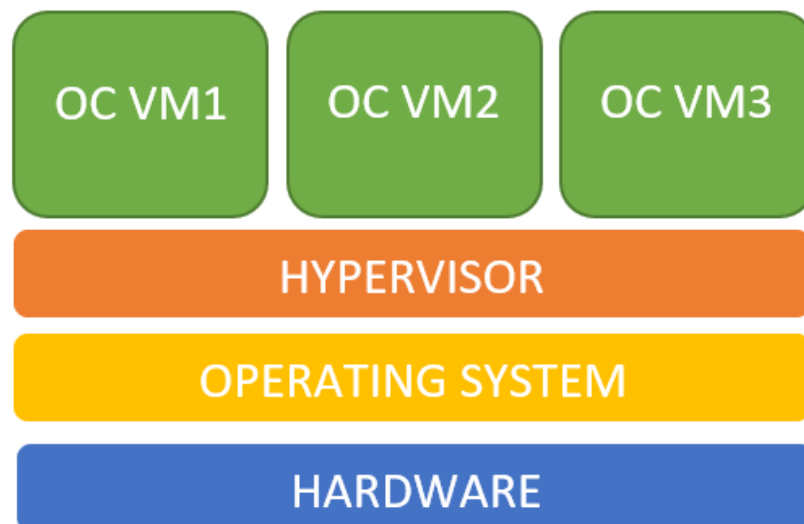


Figure 3.3-Hypervisor of the second type

«Xen» belongs to the first type of hypervisors, and was also developed in early 2000, and is a free hypervisor. This hypervisor, which allows easy creation of VM, has a number of tools to create backups and move machines to another hypervisor, not necessarily similar. Also has VM imaging technology. One of the drawbacks of this hypervisor is the lack of isolation at work when VM is created. Care should be taken to ensure that VM does not overload the server, as in case of overload, the hypervisor can stop working correctly and fail. As an example, one can consider a situation where VM takes over the entire amount of hypervisor memory, there is no function that can stop hypervisor overload, for example by reducing the amount of memory used by the VM.

«KVM» is a second-level hypervisor, the given hypervisor virtualizes the image of the OS on which in the consequence will be deployed VM. Compared to «Xen» it is easier in terms of resources, as it is part of Linux. Managed through a

scheduler of tasks and also uses memory management, due to its simplicity, allows a much more efficient allocation of resources between VM [46].

«Proxmox VM» was developed by «Proxmox Server Solutions» in Austria on conditions GNU (General Public License) because the GNU license was used in the design of this solution, it is possible to customize this solution to suit needs and the requirements of a particular project.

For the work with the hypervisor «KVM» a ready-made solution «Proxmox Virtual Environment» allowing to manage the virtualization on the basis of «KVM» was used, in the figure 3.3 there is a laugh of the work of the hypervisor «Proxmox VE». In order to manage this solution, a simple web-interface allowing to create, modify, create and manage a cluster of several servers with the developed solution «Proxmox VE» is developed. This hypervisor is a type 2 hypervisor, which runs on the Debian Linux OS. One of the advantages of being able to use cluster management via the web interface of one server. Quick debugging, does not require high qualification during installation. The web interface provides a complete overview of all installed and BM «KVM».

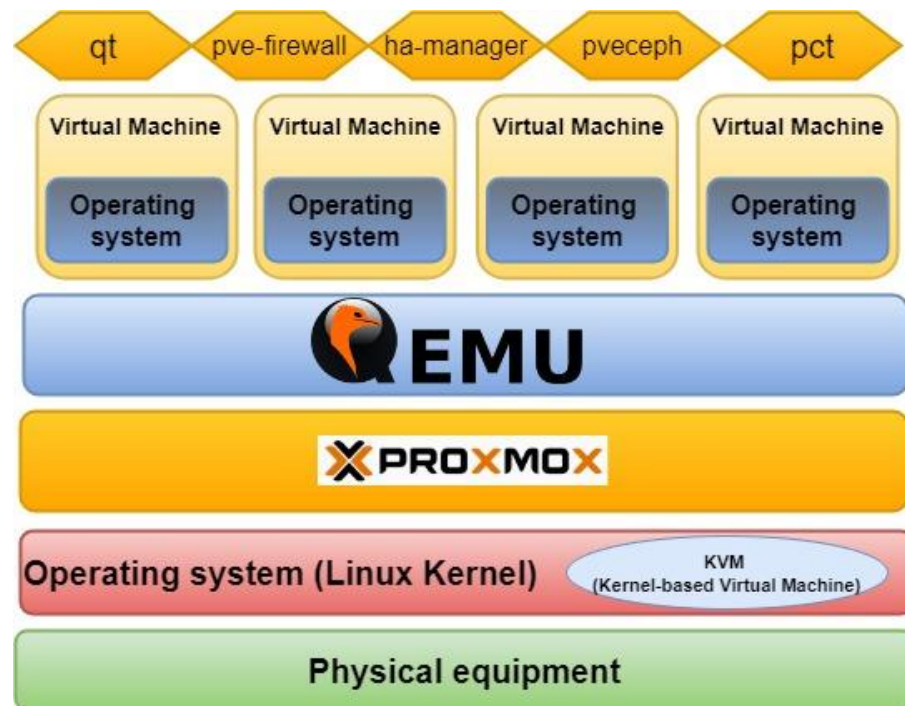


Figure 3.4-Hypervisor of the first type

«Proxmox VE» can be used on one server, and there is also the possibility of creating a cluster system. By working in the cluster system it is possible to dynamically scale server and data storage resources. Also «Proxmox VE» has a great opportunity to work with data warehouses, such as:

- Ceph RDB is an open source scalable distributed data storage system;
- GlusterFS is a scalable FS that combines the storage of disk resources from multiple servers into a single repository;
- ZFS is an FS from Sun Microsystems, which allows to store and work with DB and fail-safe.

For a hypervisor processor, the processor must support the virtual architecture. The Intel technology is called Intel VT, and the AMD virtualization-enabled processors are called AMD-V. These technologies make it possible to use hardware virtualization on company data processors, which allows the processor to work with isolated operating systems using a hypervisor as a control system. With this technology, it is possible to deploy Proxmox VE on processors. The web interface for working with «Proxmox VE» is translated into 20 languages, which allows to use this system huge number of different companies and institutions. A large community with more than 40,000 participants allows receiving practically any answer on the problem related to installation, deployment or configuration «Proxmox VE». The hypervisor «Proxmox VE» can be deployed on practically any equipment in a short period of time, regardless of equipment, where the main criterion when deployed is the presence of CPU support for virtualization technology on the server.

«Proxmox VE» uses the unique FS Proxmox Cluster file system (pmxcfs) for storing VM configurations deployed on the hypervisor, «pmxcfs» is a database-based FS, it is replicated on all members of the Proxmox cluster, for this use «syncoro». All hypervisors that are grouped together in a cluster receive information about all installed UVs, so that in case of a BM or hypervisor malfunction, a new BM could be deployed on another cluster. FS «pmxcfs» allows, when changing configuration configuration files and changing shared files, in the cluster system «Proxmox VE» to carry out a quick update of this configuration by keeping these files in a constant synchronous state. FS stores all necessary configuration data on disk, but also a copy of the data is stored in the server's RAM. Advantages of FS «pmxcfs» real-time replication of all data, checking of configuration files to avoid duplication of virtual station identifiers, and automatic updating of configurations with «corosync» on all nodes simultaneously. «Proxmox VE» possesses the technology capable to automatically detect faults in VM, and start the task on emergency automatic restart of data machines. This software stack is called ha-manager and also a stack is an automatic BM administration function where first need to specify which resources to control. Continue to process queries and problems, and in case of a failure of one of the nodes BM ha-manager makes an emergency connection to another node in the cluster, it is also possible to handle normal queries with ha-manager.

3.2 Proposed Physical Architecture for Big Data

A reliable physical system needs to be built to provide a reliable and high-performance BD system. The physical system for handling BD shall be substantially different from the systems that generate for handling traditional data. The basic paradigm for building a good physical system is the infrastructure of distributed computing. Distributed computing implies that the entire system will be physically stored on different servers or machines, which will be merged into a network infrastructure facility, and will also be connected by a distributed file system. Also for physical systems it is necessary to have built-in tools in the BD to

create a cluster and combine machines. Redundancy plays a major role in the design of the system and is important because of the huge amounts of data that must be processed and stored. There are different approaches to the way physical infrastructure is organized. If companies or organizations are small, cloud solutions can be used, they can be easily scaled up and can be phased in if additional servers are needed. Also, if an organization needs to implement locally, it is best to deploy on existing hypervisor equipment for convenient operation and deployment of necessary services. Operating an BD implies that a security system must be put in place for the processed data so that it does not end up in the hands of malicious or unscrupulous individuals. When processing, for example, data from various medical organizations, law enforcement agencies and various administrative organizations. This kind of data should be protected first. In dealing with data, the distinction between the rights to use data for the different users of the organization must be given great importance. It is necessary to create a system that allows to identify who and at what point in time has been working with certain data, that is to keep a log (log file) in which will contain all actions that users have carried out on files. Such safety requirements in the design of the system that will handle the BD shall be implemented at the earliest stages of system design. When designing architecture, it must be understood that it is likely that there will be a need to integrate an already existing system of the organization. For example, in various medical research into new approaches to treatment, equipment that does scanning or different images, various tomographies can already be stored in a medical electronic database, and it needs to be determined, how exactly to move such data to the processing system. Choose models that will optimize and increase the rate of transmission and delivery of new data to the infrastructure being built. To build such an infrastructure in a classical configuration can take from a few days to several weeks, but with new methods of designing distributed computing, this task reduces the time for building infrastructure from a few minutes to a few hours. Consideration should also be given to what types of databases should be used in the design of a new system. Depending on the type of data received, it is possible to use both a relational classical design approach and a NoSQL approach for atypical system design tasks. For example, graphical databases are nodal-based and relationship-based to detect the formation of a particular type of cancer, depending on the dietary intake of certain products.

The Hortonworks Data Platform (HDP) distribution was selected to select a platform that will allow the database system to be deployed. Table 1 of the existing solutions has been compiled to identify existing products that provide the infrastructure for operating with BD. Hortonworks company offers DB-enabled tools, as a platform the company has created HDP, this platform uses the tools of Apache company, and the platform is based on Apache Hadoop. This tool allows performing various calculations and work with BD. HDP allows to work with database using Apache's Web Interface software to manage the HDP infrastructure called Apache Ambari. Using this product to get an effective control panel for all HDP tools for BD processing. The convenience of using this software is due to the

fact that it is possible to add additional components to the working medium at a high speed, since the web interface is very well developed and even a professional in this field will be quite easy to configure HDP to work with additional tools.

The following is an analysis of the selected solutions, their technical characteristics and pros and cons when using these solutions. As a result of the research, the HDP product in Figure 3.4 was selected. This software allows to exploit the full potential of BD as it has a huge number of tools and platforms for BD analysis, processing and visualization. HDP allows the use of a large number of tools to process and manage database. HDP also allows data processing without using a relational approach in data storage, but it is possible to use the SQL (Structured query language) query language for developers and users who are used to this method of data processing and handling.

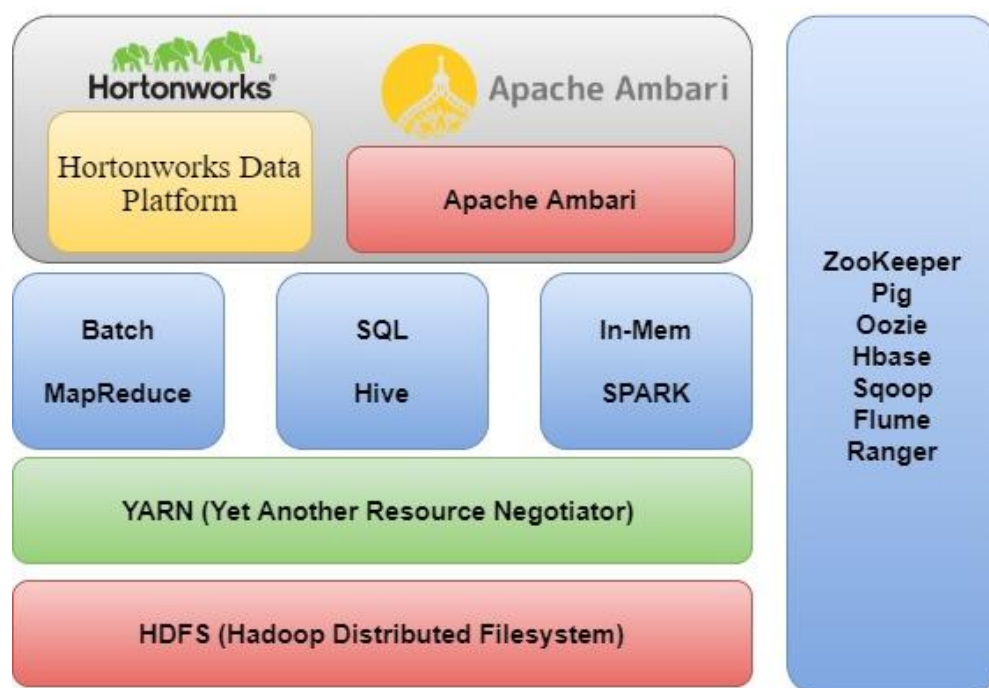


Figure 3.5 - HDP Infrastructure

HDP is an excellent solution for working with BD and also uses a data processing application library, which includes the Apache Ambari software solution. It is a software solution for management, as well as for monitoring and the ability to create a secure backups system for Apache Hadoop. Apache Ambari allows to manage a cluster using an intuitive web interface, all parameters of work and resource allocation are implemented in a single control system, which saves the system administrator or user from having to change the configuration files using the console. Apache Ambari allows to run, stop services centrally, and use this tool to work with many software solutions.

Table 1-Analysis of technical characteristics of software for implementing a cluster database processing system.

Title	Manufacturer	License	OS Support	Pay rate	The technology used
MapR 6.1	MapR	Proprietary	Red Hat, CentOS, Ubuntu, SUSE, Oracle Enterprise Linux	Only the M5 version is free/paid support, paid additional products	Apache Drill, Apache Hadoop, Apache Hive, Apache Mesos, Apache Myriad, Apache Spark, MapReduce
Cloudera Enterprise 6	Cloudera	Proprietary	Red Hat, SUSE, Oracle Linux, Ubuntu	Free for up to 50 cars\Payment for the management manager, paid support	Flume, Hbase, Hive, Hue, Impala, Kafka, Kudu, Oozie, Search, Sentry, Spark
HDP 3.0	Hortonworks	GNU GPL (Only some tools are paid)	Red Hat, CentOS, Oracle Linux, Ubuntu, Debian, Windows Server	Free\ Paid support	Hadoop, Accumulo, Atlas, DataFu, Falcon, Flume, HBase, Hive, Kafka, Knox, Mahout, Oozie, Phoenix, Pig, Ranger, Slider, Spark, Sqoop, Storm, Tez, Zeppelin, ZooKeeper

Also analyzed the functional characteristics of existing solutions for deploying a cluster system in a database environment. This analysis is presented in table 2. The table shows the characteristics of existing systems for working with BD, which gives an idea of the main characteristics of existing products.

Table 2-Functional characteristics of software for implementing a cluster BD processing system.

Title	Free use	Integration with REST IP	Possibility of implementing third-party software	Windows OS Support	OS support on Linux/Unix kernel	Using HDFS	Using Apache Products	Development Model
MapR 6.1	No	YES	YES	No	YES	Modification HDFS(NFS-MapR-FS)	Use only proprietary products	Enterprise Application Software (EAS)

Cloudera Enterprise 6	Partially	YES	YES	No	YES	YES	YES	Enterprise Application Software (EAS)
HDP 3.0	YES	YES	YES	YES	YES	YES	YES	Open-source model

Following a comparative analysis of the existing BD products, the following conclusions were drawn, which will prioritize the tools as follows:

1. It is also possible to expand and process data as quickly as possible if the paid product MapR is preferred. The system suffers from high costs and poor documentation;
2. If it is necessary to deploy a cluster system in the Windows Server environment, as well as for the least cost and further integration with more tools, it is best to use the HDP software;
3. When using different proprietary products or when planning to add them to the system in the future, Cloudera Enterprise software is the best choice.

3.3 Install Proxmox hypervisor on server

The hypervisor «Proxmox VE» allows managing the physical infrastructure with the help of the web interface, which becomes available after development of the distribution. This hypervisor was chosen because of its high fault tolerance and ability to use multiple filesystems when expanding. Also, this hypervisor has the possibility of creating a cluster between several hypervisors «Proxmox», which significantly increases the fault tolerance of the infrastructure in the future, since when creating a cluster, it is possible to configure machine replication on several nodes of the cluster. In the event of a VAM failure or if a malfunction is detected, the VAM can be automatically deployed within minutes on another hypervisor.

The hypervisor Proxmox VE 5.4 was chosen as the hypervisor to control the entire infrastructure. It is simple enough to install on new equipment, and is not demanding on system resources, when deployed it is necessary to create a boot media or to install over the network. The following are the stages of installation and use of the hypervisor «Proxmox VE 5.4». The installation steps show how easy it is to install the hypervisor using the official image of the hypervisor from the official website. Also, if necessary, it is possible to deploy on this hypervisor not only machines with Linux family OS, but it is also possible to install Windows OS. This hypervisor supports several authentication sources, such as:

- Linux PAM (standard authentication used in Linux);
- Proxmox VE (built-in authentication);
- Microsoft Active Directory.

Using these authentication methods can provide high security control when working with the hypervisor.

During the initial setup, a menu will appear where need to select Install Proxmox VE to install, as shown in Figure 3.6.



Figure 3.6-The main boot menu of the Proxmox distribution.

Next, then have to agree to a software license as shown in Fig. 3.7.



Figure 3.7 – the License agreement for use of the software.

The next step is to configure the disk space and, if necessary, partition the disk into certain partitions. During the initial installation, can set the default disk space as shown in Figure 8.



Figure 3.8 – Partitioning the disk to install the system.

Next, then need to configure the region to configure the server's time configuration correctly, as shown in figure 3.9.

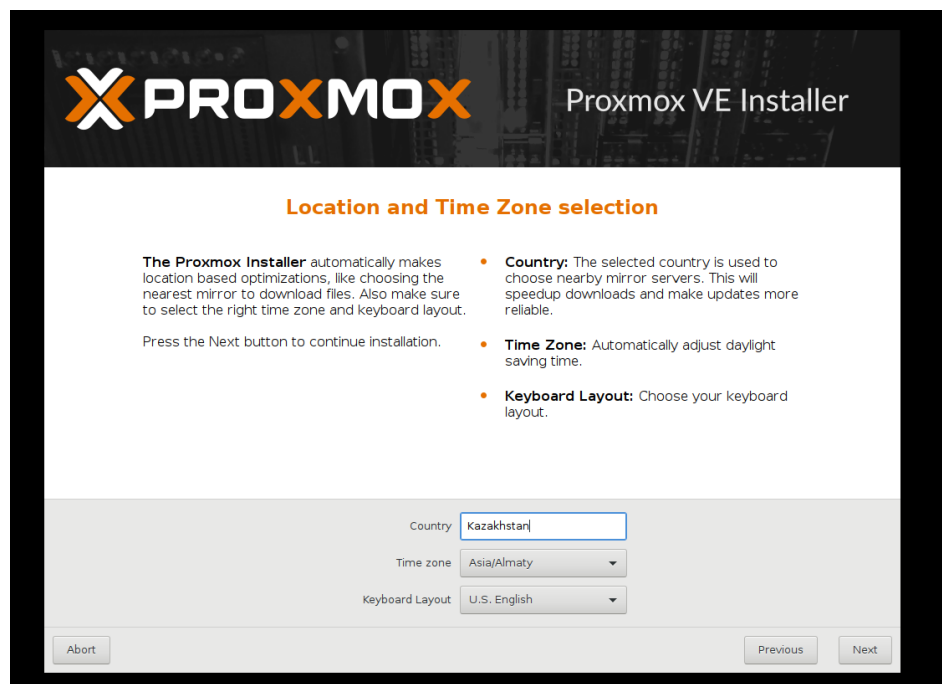


Figure 3.9 – The setting of the time zone.

After setting up the region, then need to set up a password for the superuser, as well as the mail to which system messages about the server status will be sent, as shown in Figure 3.10.



PROXMOX Proxmox VE Installer

Administration Password and E-Mail Address

Proxmox Virtual Environment is a full featured highly secure GNU/Linux system based on Debian.

Please provide the **root** password in this step.


- Password:** Please use a strong password. It should have 8 or more characters. Also combine letters, numbers, and symbols.
- E-Mail:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue installation.

Password:
 Confirm:
 E-Mail:

Figure 3.10 - Setting up a password to access the server.

Next, then need to configure the IP address for the hypervisor, if the system has a DHCP server configured, the addresses will be obtained automatically, and also need to configure the DNS server name, as shown in Figure 3.11.



PROXMOX Proxmox VE Installer

Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installation.

Afterwards press the Next button. You will be shown a list of the options that you chose during the previous steps.

- IP address:** Set the IP address for your server.
- Netmask:** Set the netmask of your network.
- Gateway:** IP address of your gateway or firewall.
- DNS Server:** IP address of your DNS server.

Management Interface:
 Hostname (FQDN):
 IP Address:
 Netmask:
 Gateway:
 DNS Server:

Figure 3.11 - Configuring the IP address and domain name.

Then, at the end, get a checklist with parameters that have been set up for further server installation, as shown in Fig. 3.12.



Figure 3.12 – All configured configuration for the server.

After installing all the hypervisor components, the installation script will ask to restart the system, where need to restart the hypervisor, as shown in Figure 3.13.

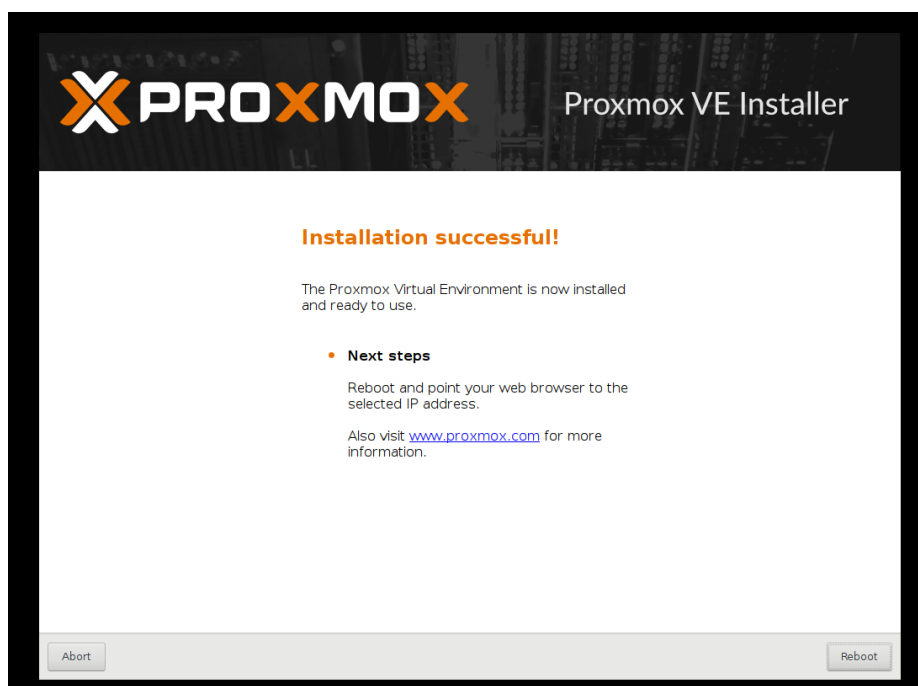


Figure 3.13 - Message about successful installation of the hypervisor.

After restarting, the server should start with instructions on what address and port can connect to the server through a web browser. The console window will show the address and port to connect to, as shown in Figure 3.14.

```
-----  
  
Welcome to the Proxmox Virtual Environment. Please use your web browser to  
configure this server - connect to:  
  
https://192.168.181.136:8006/  
  
-----  
  
HDPCL2 login: _
```

Figure 3.14 - Connecting to the hypervisor via a terminal.

To log in, go to the address indicated in the previous paragraph where want to open browser and add an address and port to login. If a dns server is configured in the system, can use the dns name to connect to the hypervisor as shown in figure.3.15. To login, then must enter the username and password configured during the hypervisor installation.

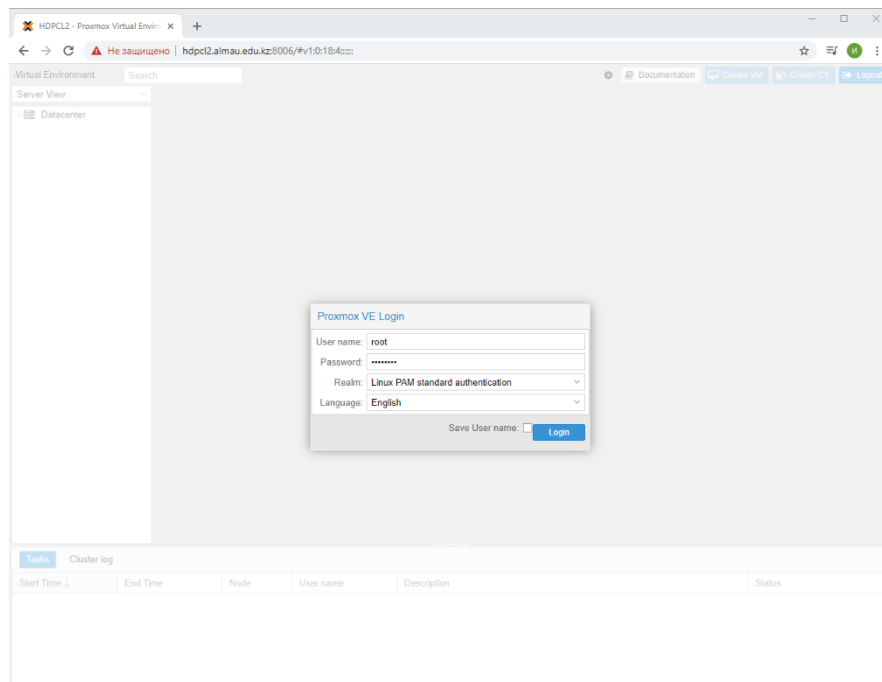


Figure 3.15 - Connecting to the hypervisor via the web interface.

After entering the superuser data, the workspace opens. Next, can proceed to managing the hypervisor as shown in Figure 3.16.

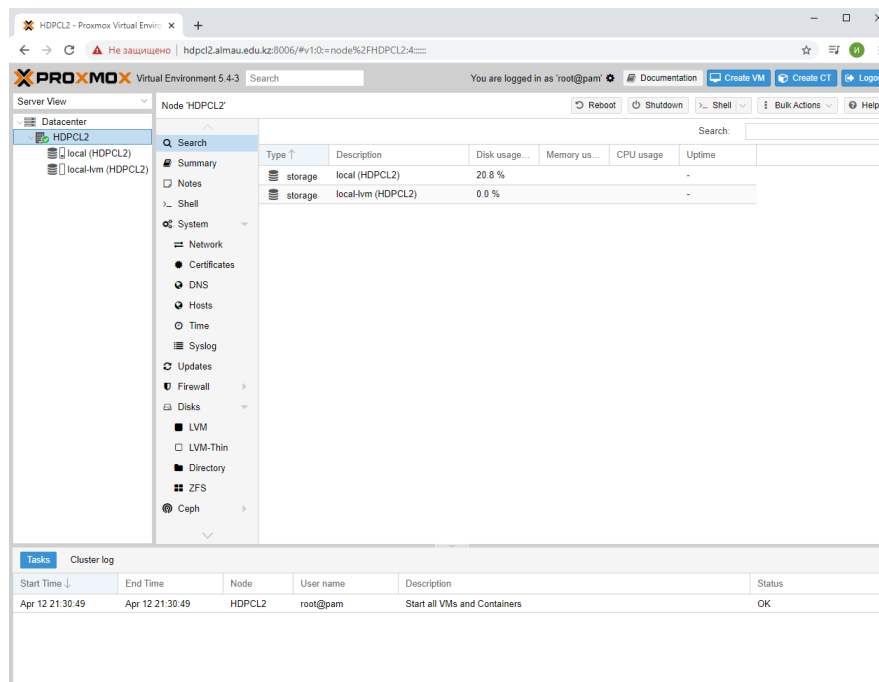


Figure 3.16 - Workspace when working with the Proxmox hypervisor.

As a result of installing the "Proxmox VE 5.4" distribution, a tool was obtained that allows to work with a virtual environment on top of physical equipment, which significantly expands the possibilities of creating new VMs as servers for working with the database. It is also possible to create a network of several hypervisors in the future in the case of expanding the fleet of server equipment. During the initial setup must specify the name and ID number of the new machine, as shown in Figure 3.17.

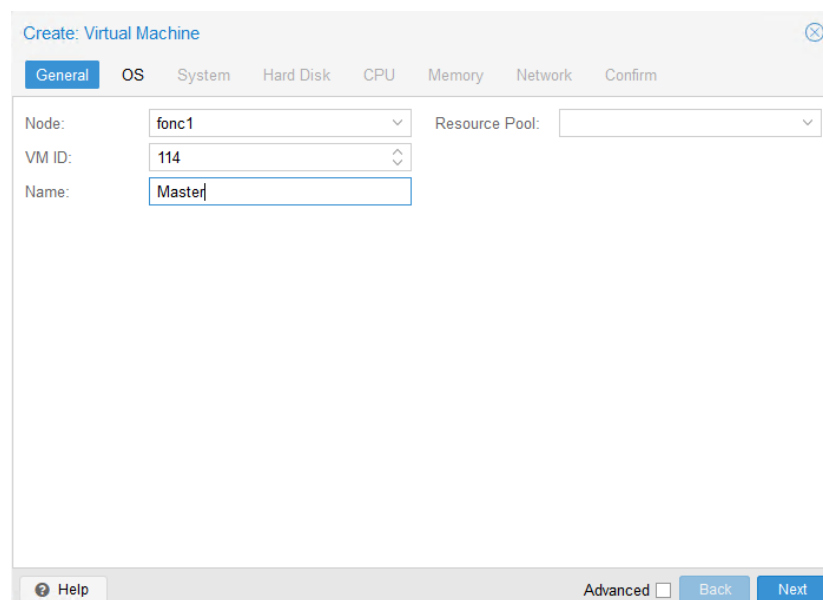


Figure 3.17 - Window for creating a new VM

Next, need to specify which image will be selected when installing the OS, as well as which version of the kernel will be used, Figure 3.18 shows the scheme.

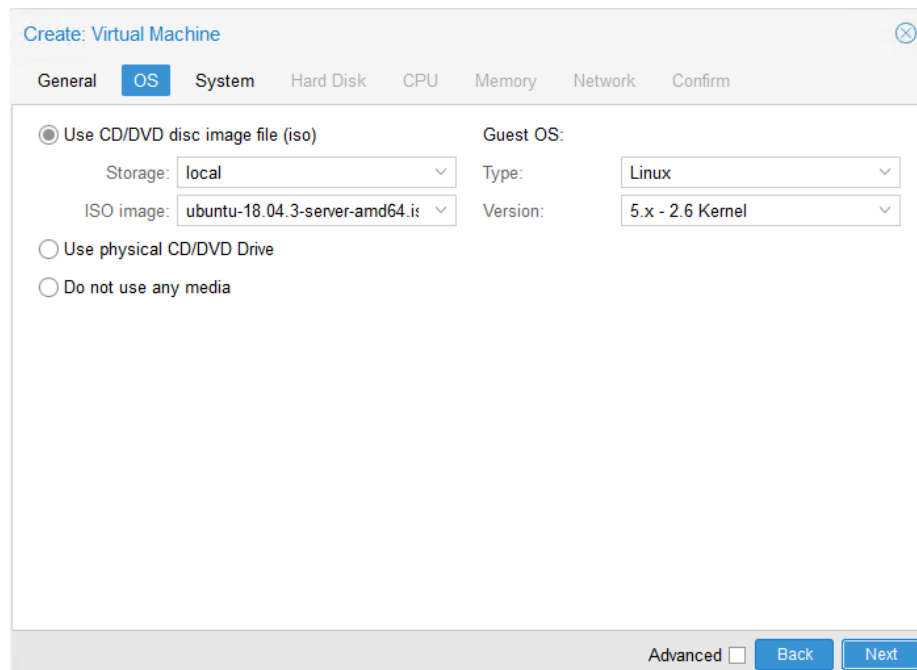


Figure 3.18 - Selecting the OS and installation samples

Next must specify the amount of disk space for the machine, as well as on which storage the machine will store the data. See Figure. The parameters selected for the laboratory bench are shown.

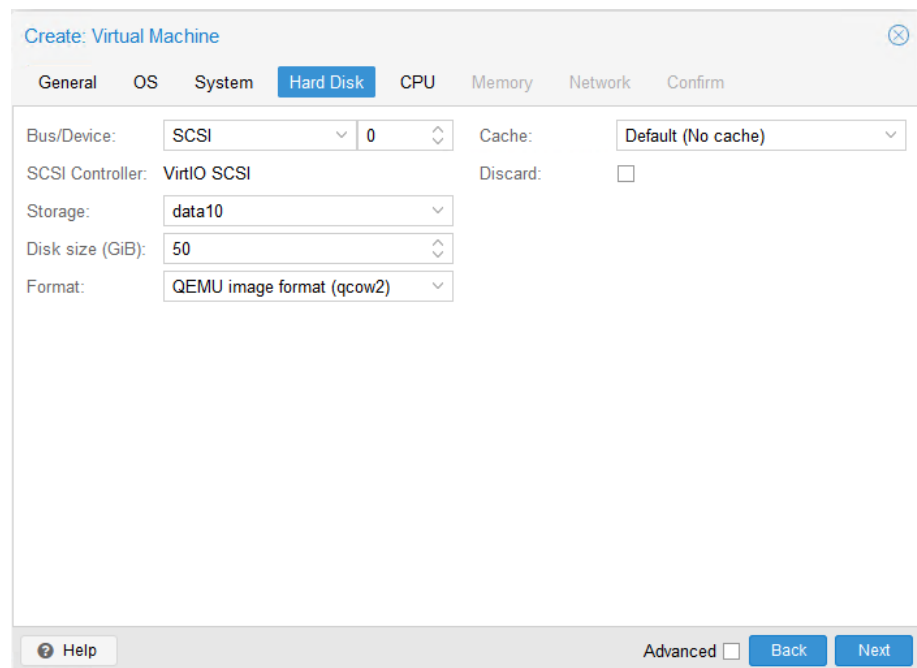


Figure 3.19 - Configuring disk Space for a new VM

The next item allows to select the number of cores as well as threads for our machine, as shown in Figure 3.20.

Create: Virtual Machine

General OS System Hard Disk **CPU** Memory Network Confirm

Sockets: 2 Type: Default (kvm64)

Cores: 2 Total cores: 4

Help Advanced ☐ Back Next

Figure 3.20 - Configuring the number of processor cores for a new VM

The next menu item allows to set how much RAM want to allocate to the machine, as shown in Figure 3.21.

Create: Virtual Machine

General OS System Hard Disk CPU **Memory** Network Confirm

Memory (MiB): 8192

Help Advanced ☐ Back Next

Figure 3.21 - Setting the required amount of memory for a new VM.

The last window displays all the settings that were selected as a result of the survey, as shown in Figure 3.22.

Create: Virtual Machine

General OS System Hard Disk CPU Memory Network Confirm

Key ↑	Value
cores	2
ide2	local:iso/ubuntu-18.04.3-server-amd64.iso,media=cdrom
memory	8192
name	Master2
net0	virtio,bridge=vbr0,firewall=1
nodename	fonc1
numa	0
ostype	l26
scsi0	data10:50,format=qcow2
scsihw	virtio-scsi-pci
sockets	2
vmid	114

☐ Start after created

Advanced ☐ Back Finish

Figure 3.22 - Background information about the settings that were applied to the new VM

To start the created VM and for further installation, need to click the "Start" button, as shown in Figure 3.23.

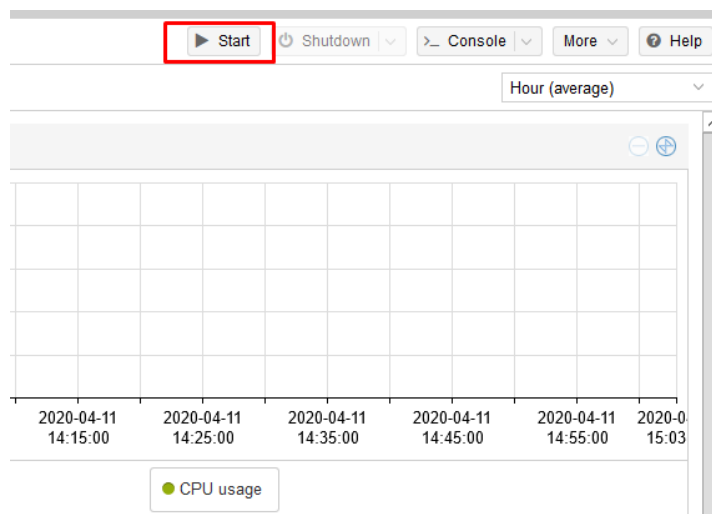


Figure 3.23 - Starting the created VM

The Linux Ubuntu Server 18.04 family OS was selected as a result of the development of the new WM. After the hypervisor was deployed, the network infrastructure was configured to allow the hypervisor to access the organization's local network. For remote access to WM over the network, the hypervisor connection to the organization's web servers was tested. This is to be able to upload to WM, where the tools to work with BD will be deployed Web server activity tracking logs.

3.4 Install operating system for infrastructure

Ubuntu Server 18.04.4 LTS was selected as the OS for installation, where the OS was chosen because it is one of the most stable currently and easy to deploy. It is also important that the OS has a huge repository with current programs. The OS has a large community and well-structured documentation. The documentation contains all the necessary information, both for the deployment of the given OS, and about the frequent problems encountered in working with the given distribution. This in turn allows for quick resolution of problems encountered when installing or deploying different components.

When initially installed, must choose the language of the interface, and the default language is English, as shown in Figure 3.23.

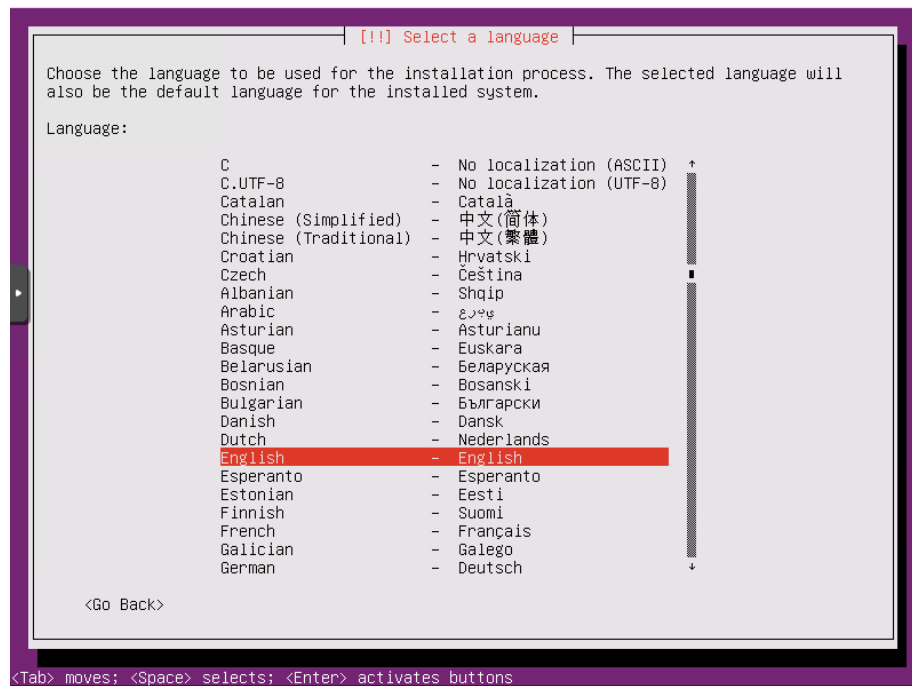


Figure 3.24 - Language selection for OS installation

During installation, the system will ask to configure IP addresses, if the system already has a DHCP server configured, then the VM will receive an automatic address, if not, then can configure the addressing manually, as shown in Figure 3.25.

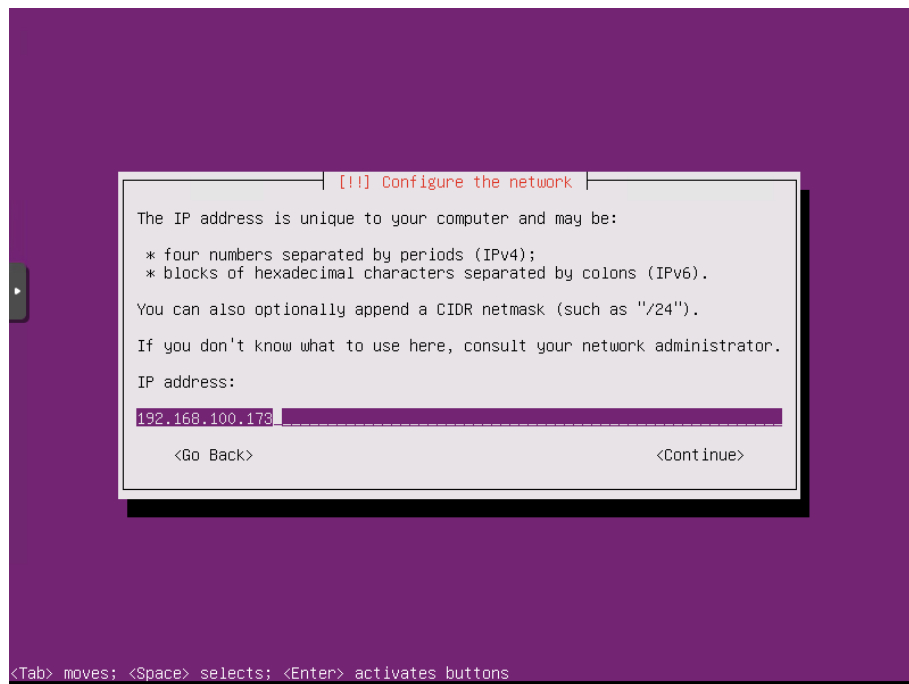


Figure 3.25 - Configuring the IP address

Next, need to configure the name for the server, as shown in Figure 3.26.

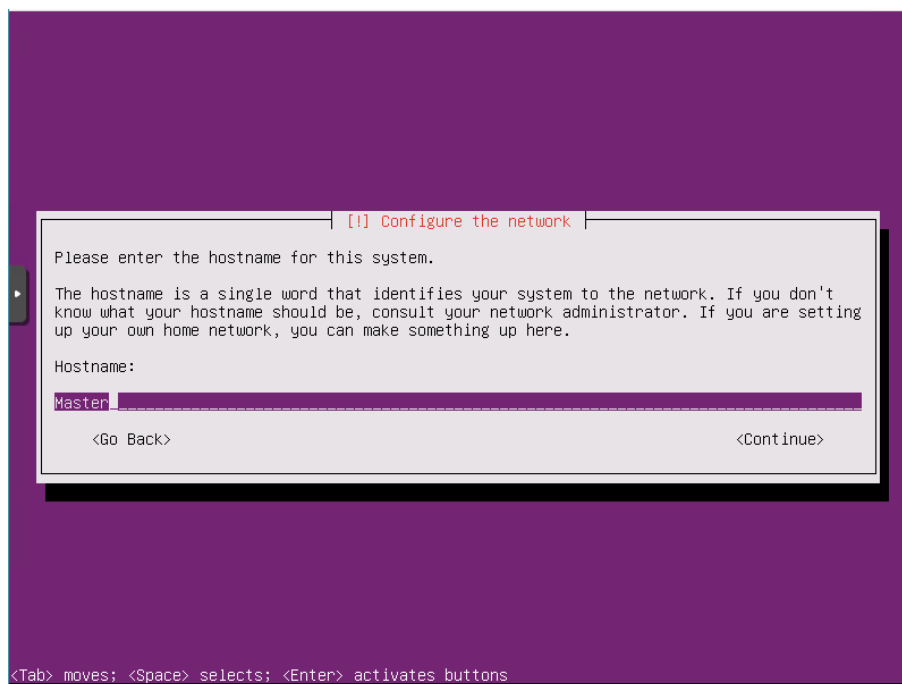


Figure 3.26 - Configuring the Server name

Next, need to configure a new user for the system, then by default, this user will have superuser capabilities, as shown in Figure 3.27.

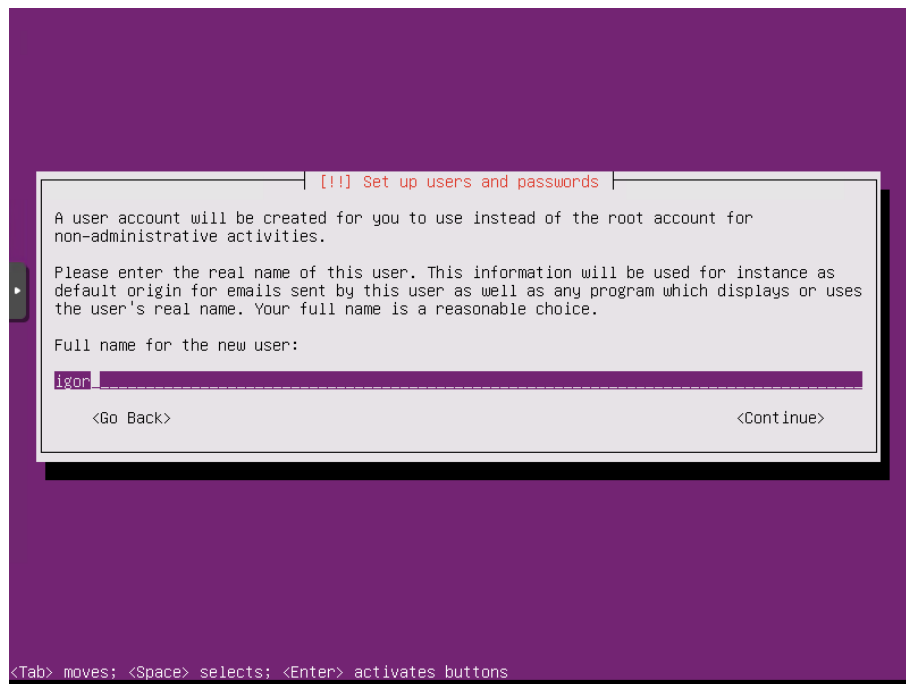


Figure 3.27 - Adding a new user

Next, need to select the region to set the correct time and date on the server as shown in Figure 3.28.

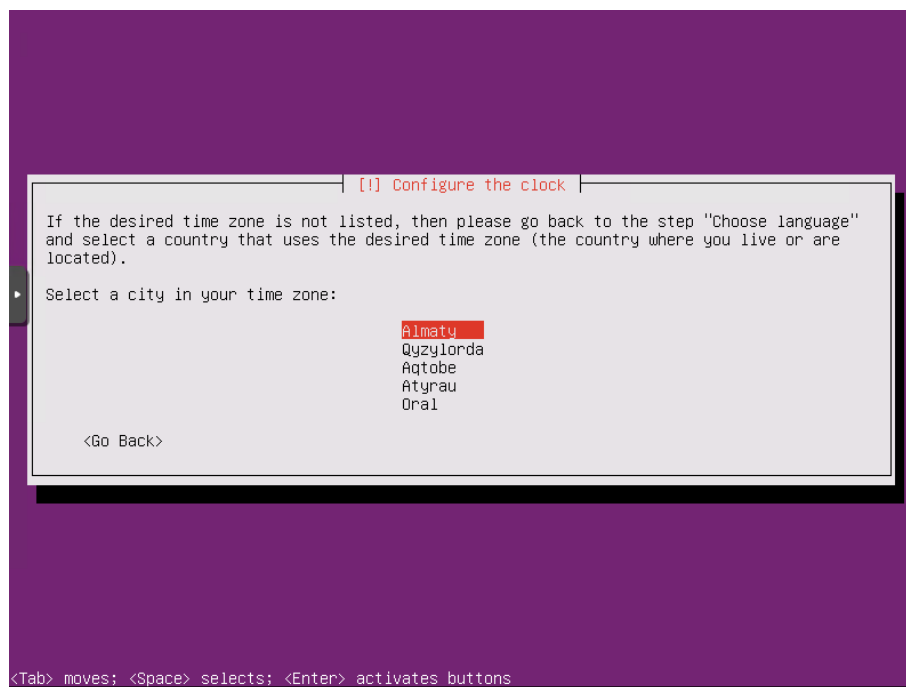


Figure 3.28 - Setting the time zone for the OS

The next step is need to choose how disk will be divided, also ho. In Figure 3.29, the first partitioning method will be selected.

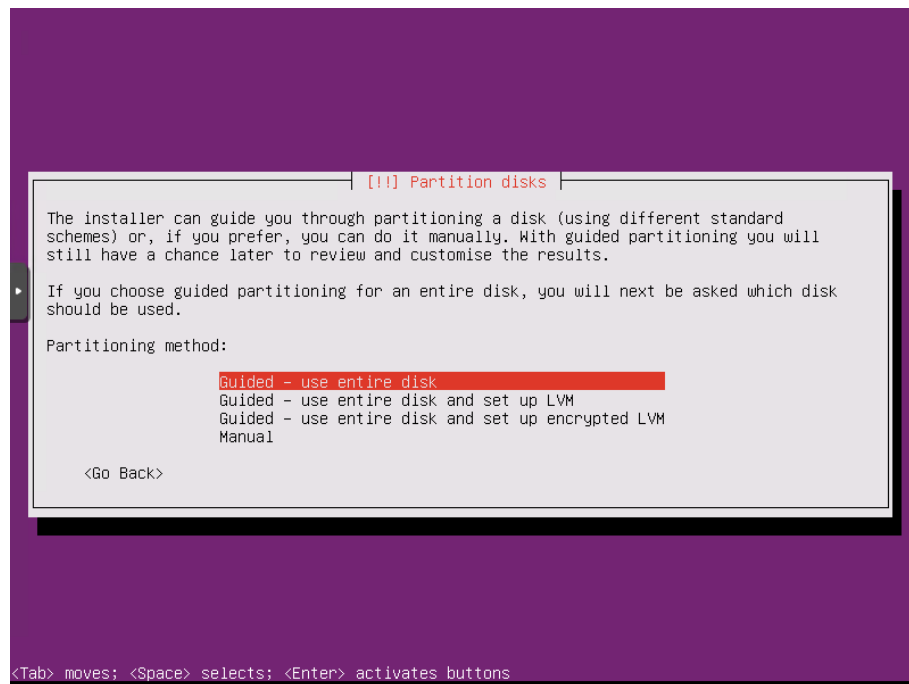


Figure 3.29-Disk layout for the OS

After that, must configure a mandatory secure update for the OS, as shown in Figure 3.30.

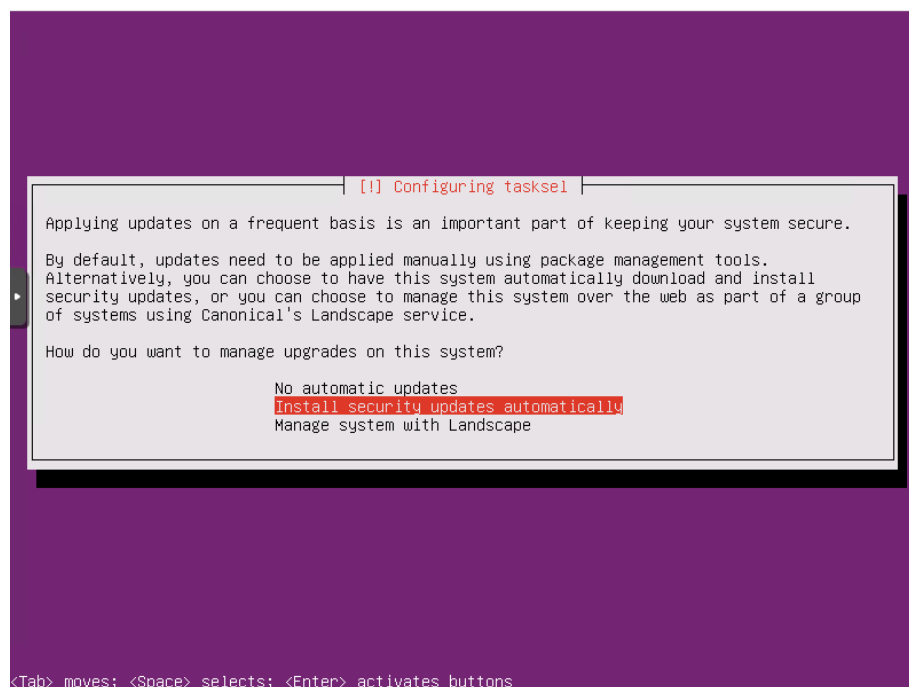


Figure 3.30-Selecting an OS Update

Next, need to select additional packages for installation, where need to install OpenSSH server in order to get remote access to the VM in the future, as shown in Figure 3.31.

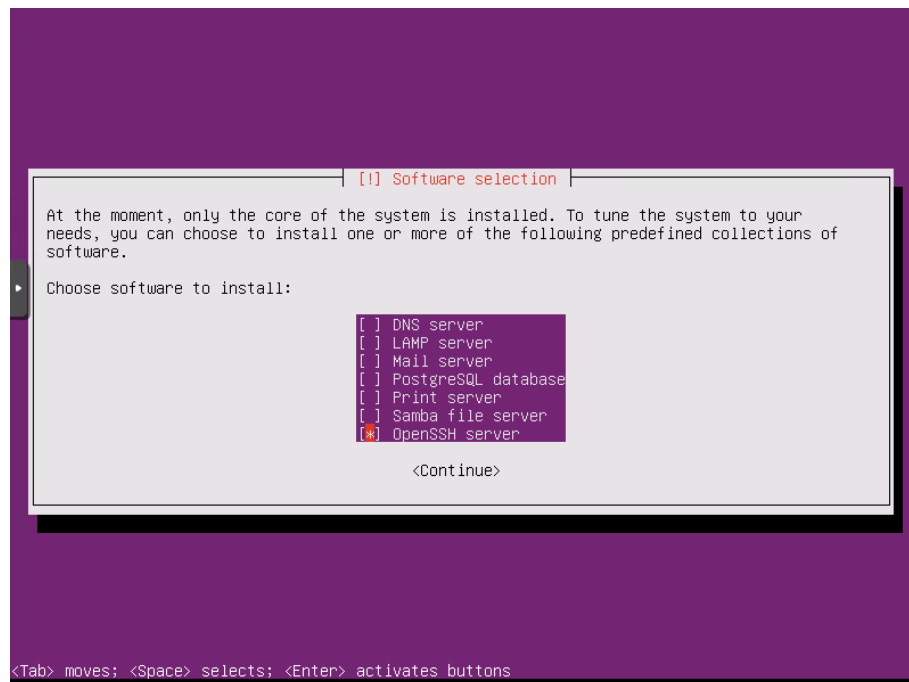


Figure 3.31-Adding additional components

After the installation is complete, will be prompted to restart the VM as shown in Figure 3.32, and after rebooting, access the installed VM with the OS.

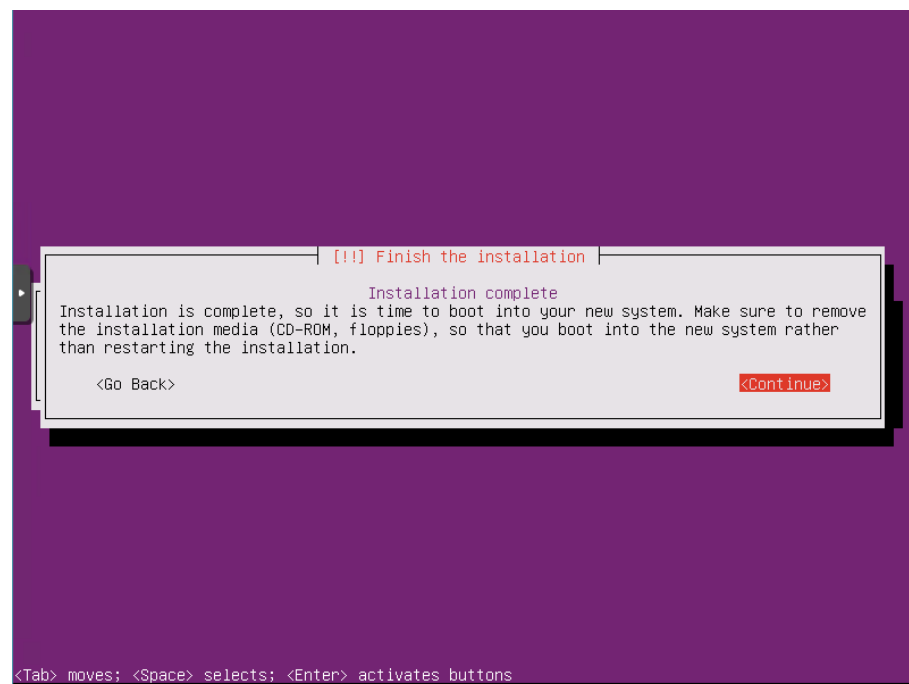


Figure 3.32 - Message about the end of OS installation

After the OS is installed, can access the VM (on which Ubuntu Server 18.04 was installed) through the console. Access to the VM can be obtained through a web browser window, this feature is provided by the Proxmox VE hypervisor, or the connection can be made over a secure SSH protocol, using any software that supports an SSH connection.

After restarting the OS, the VM will ask to enter login credentials, as shown in Figure 3.33. Then must enter the data that was configured in the previous steps when the new user was created.



Figure 3.33-Console for working with the installed OS

As a result of the installation, the remote connection to the VM via the secure SSH protocol was configured to allow authorized access to the VM via the loco network for control. The working environment on the server was then configured, where the necessary software packages were installed to further install the database tools.

The choice of this particular OS (Linux Ubuntu Server 18.04) is due to several factors, including:

- 1) Support for up-to-date and stable repositories with programs, which means that the programs that will be installed on the OS will only be stable versions, not experimental or unsuccessful;
- 2) Support of the release of this OS allows to receive the most recent updates, which significantly increases the security of this distribution;
- 3) An updated OS does not require an immediate update of the repository in the console when adding a new repository branch to the configuration file. This is done automatically, which significantly increases the speed of operation with this OS.

Once installed, WM was obtained, which can then be used to create new specimens to expand the cluster of the future system, cloning the installed OS, and creating new clones on the hypervisor.

4. INSTALLING HORTONWORKS DATA PLATFORM AND PROCESSING WEB SERVER LOGS

4.1 Log file processing system description

The aim of this study is to create a system for processing log files of web servers in the BD environment, for further analysis and visualization of the results. The basis for the experiment is the possibility of downloading and processing file logs stored on the proxy server as well as on the servers where the university sites are located directly.

Data collected by the Apache and Nginx web servers must be processed, as all incidents and interactions with the organization's websites are documented in the logs of the web servers. Such data needs to be stored and processed to obtain detailed information on how the system has operated throughout its life. Storing data from logs allows to obtain statistics of site visits during processing and visualization of logs, and can track the period in which the activity of attacks on a particular site in the system increases by studying logs of web servers.

Created system allows to collect logs from many web servers in one place and to structure them, and to process on the created platform. The main advantage for data storage is the ability to expand the data warehouse by adding new nodes to the cluster, which can be implemented using the distributed HDFS file system. This FS allows to store huge amounts of data, due to the ability to quickly add new nodes to the system. Since web servers cannot store a large amount of logs and log files are filled with information very quickly, this can lead to server overflow and disrupt its operation. To avoid this situation, a special script was created that cleans up old logs on the web server. The problem is that if after a certain period of time it is necessary to raise information on old logs (due to security problems or information leaks), this will not be possible, since these logs will be deleted from the server. In this way, need to save all the logs of web servers on a special server with a distributed file system, so that can later look for problems in the security of the web server or to analyze the operation of the web server. The created platform allows to solve this problem by using the distributed HDFS file system.

The python programming language will be used to process the log files. This problem can be solved using other programming languages. Python was chosen because it has various modules for working with databases, and for the ability to quickly learn how to create ready-made programs.

Figure 4.1 shows a diagram of the stand, which demonstrates how the IP participants interact with each other. From the web servers, the log data is sent to the HP server, to a distributed file system, then this data is analyzed using python and Apache Spark, and the result is data visualization in the Apache Zeppelin environment.

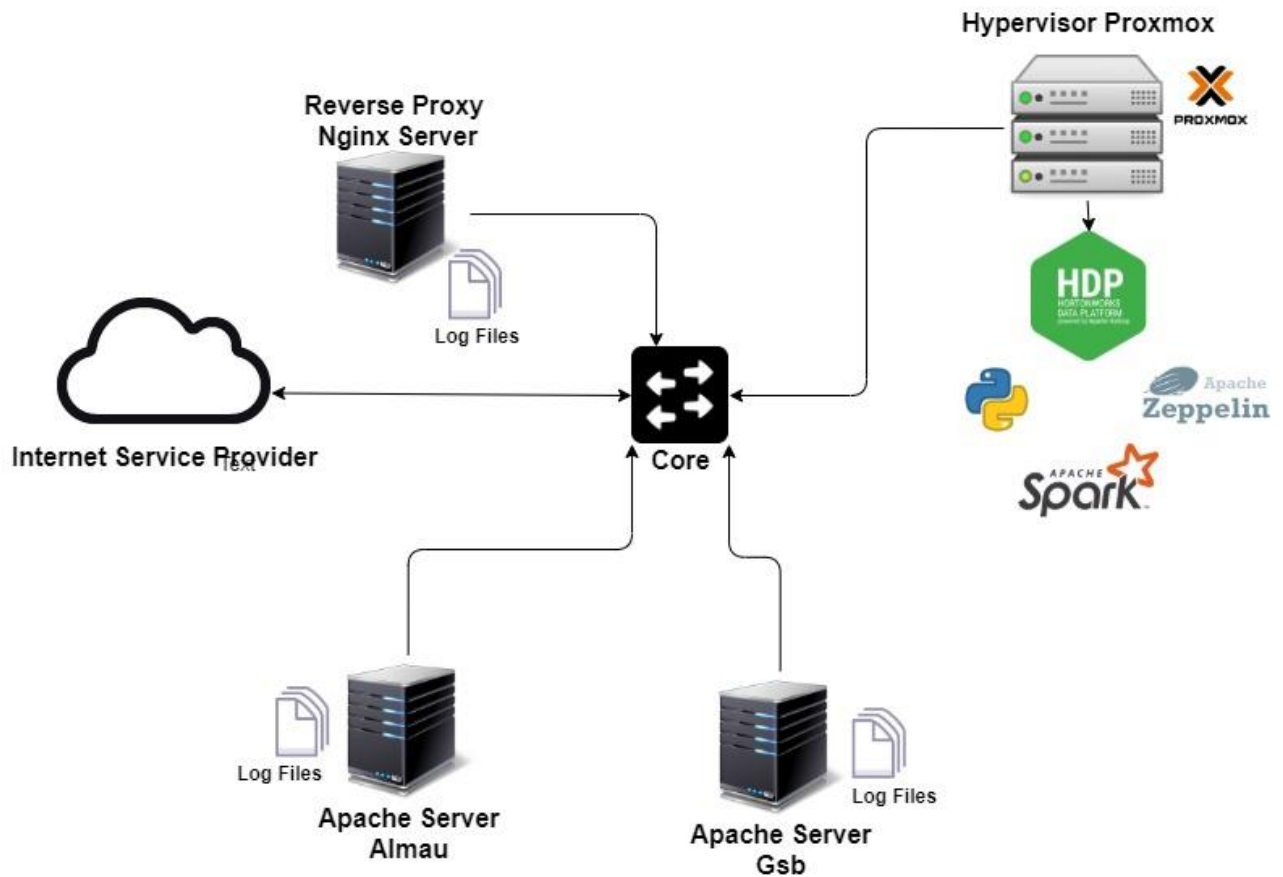


Figure 4.1-Diagram of the test bench for processing web server logs

To create an IC for processing web server logs, a server with the following characteristics was used: CPU 24 x Intel(R) Xeon (R) CPU E5-2620 v2 2.10 GHz, RAM 64 Gb.

Web servers responsible for accessing web resources will be used as data sources for processing. NGINX and Apache are used as web servers. Web server data is the most common for creating web platforms. There are also a huge number of other software products that implement web access functions to various resources. In this experiment, two of the most frequently used web servers were taken. In the future, it is possible to organize a processing system for other logs from various sources, not only from web servers, but also from various other data collection systems.

The Proxmox hypervisor was selected and installed as the data processing platform. Also, a VM was deployed, on which tools for working with the database will be installed. HDP was chosen as the data management platform tool, as shown in Figure 4.1. The specified tools that will be used in data processing. Python will be used as the programming language for writing a data processing program. Apache Spark software will be used for distributed processing of incoming files, which will make it possible to process data at a higher speed. As a wrapper for writing code in the HDP environment, Apache Zeppelin software was installed, which works as a tool for interactive interaction through the web interface for writing code and executing it. This product provides a friendly interface for writing programs, which greatly facilitates the process of developing an application.

4.2 Working with reverse proxy server logs and web servers

With the Python programming language is needed to use an interpreter to interact with various data through Spark. Apache Zeppelin will be used as an interactive system for working with the Apache Spark tool. Apache Zeppelin allows to work with tools such as Apache Spark, Scale, Hive, HBase.

During the file processing process is needed to solve the initial task of uploading data to the server. To solve the problem of uploading data to the HDP server, a script for uploading files from web servers is used, where this script is written on the Ansible software solution, which in turn allows to manage the infrastructure through configuration files. The scheme of operation in Figure 4.2 shows the loading of log files to the server.

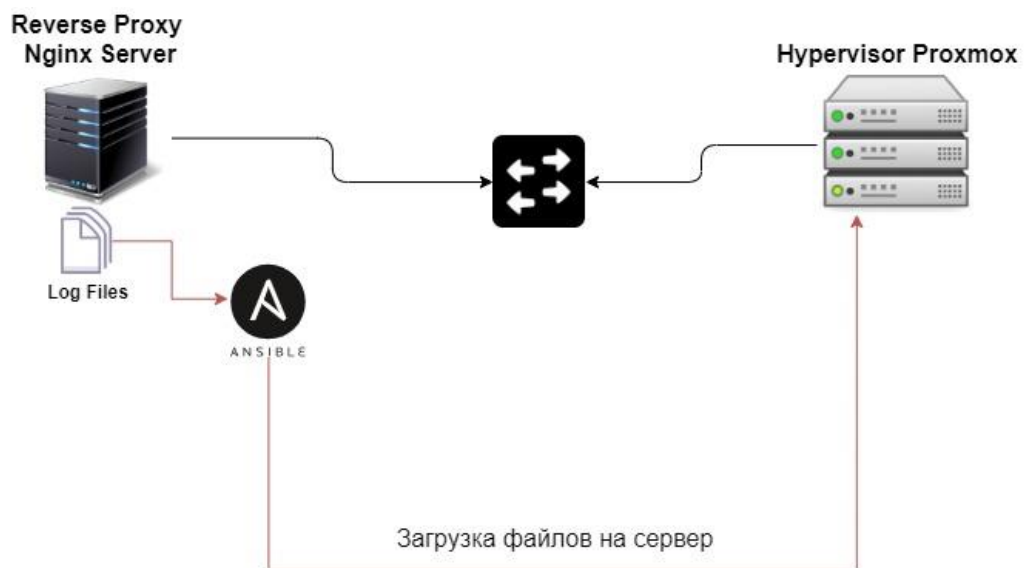


Figure 4.2-Uploading files to the HDP server

HDFS is used as a repository for all incoming log files from web servers. Distributed file storage is capable of storing a huge amount of information, which allows to download almost any number of files. Also, this FS allows to add additional nodes to the cluster to increase file space. Due to additional replication, data can be stored on multiple nodes at the same time, which increases the security of stored data. In HDFS, data is divided into fixed-size blocks and stored on different cluster nodes, as shown in Figure 4.3.

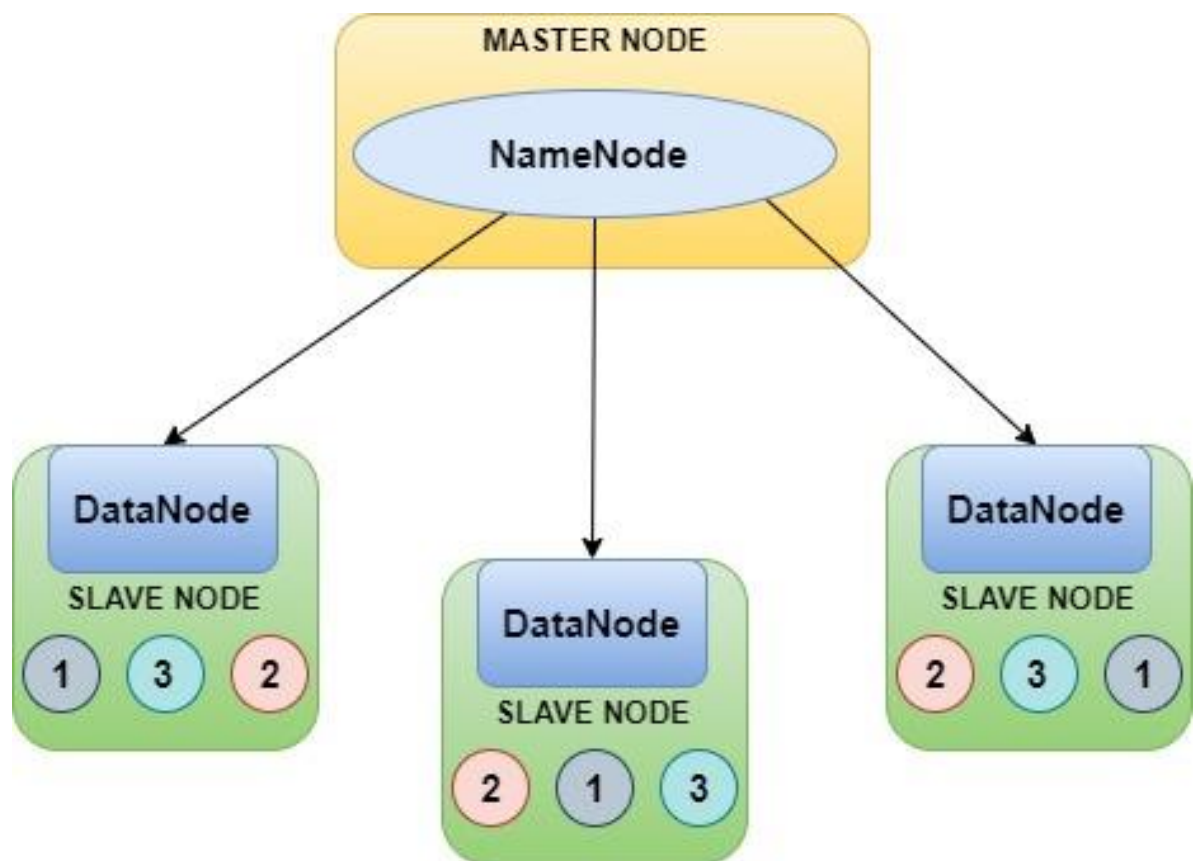


Figure 4.3 – The Architecture of HDFS

In HDFS, the main NameNode management node provides metadata storage, which in turn ensures that information about the stored data on the DataNode nodes is up-to-date. These are data nodes, where all incoming information is stored on them. User access to files is managed by the NameNode host machine.

Due to the large amount of incoming data, it is necessary to use a software solution capable of analyzing information in a distributed environment, as such a tool will be used for processing incoming data in a distributed environment Apache Spark. Apache Spark has extensive tools for data processing, such as the ability to process data using SQL libraries, the ability to process data using machine learning, the Spark program core also allows to process incoming data. Spark supports programming languages such as Python, Scala, R, Java, which makes it an effective tool for a large number of users. The key factors in choosing Apache Spark as a tool for processing web server logs is the ease of use of Spark. One of the advantages is the large support for various APIs in the Apache Spark library, as well as well-documented sources that allow to get started with the built-in Spark APIs in a short time. High speed of data processing due to the use of RAM for calculations, which increases the speed of data processing many times. The ability to integrate various products to work with Spark, such as HDFS, various NoSQL databases Apache HBase, and Apache Cassandra.

All these factors allow to use Apache Spark as an effective tool for parallel processing of a lot of data in a cluster, and increase the processing speed.

4.3 Installing and configuring the Hortonworks Data Platform for Web Server log processing

To work with the database is need to install and configure DHCP on the installed Ubuntu Server 18.03 VM, for this need to do a number of installation manipulations.

The first thing is needed to do is download the repository from the official website. This repository must be added to the main OS repository for further installation of HP on the OS. To do this, use the wget command as shown in Figure 4.4, where the command will allow to download the configuration file to the local OS.

```
root@Master2:/home/igor# wget -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.4.0/ambari.list
--2020-03-09 22:41:57-- http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.4.0/ambari.list
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)... 54.192.230.85, 54.192.230.23, 54.192.230.68, ...
Connecting to public-repo-1.hortonworks.com (public-repo-1.hortonworks.com)|54.192.230.85|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 187 [binary/octet-stream]
Saving to: '/etc/apt/sources.list.d/ambari.list'

/etc/apt/sources.list.d/ 100%[=====>]      187  --.-KB/s    in 0s

2020-03-09 22:41:57 (22.1 MB/s) - '/etc/apt/sources.list.d/ambari.list' saved [187/187]

root@Master2:/home/igor# _
```

Figure 4.4-Adding a repository for installing HDP

In the future, need to add a public key to be able to download and install programs from the repository, for this is needed to run the command as shown in Figure 4.5. The command package for installing APT (Advanced Packaging Tool), this utility has extensive tools for installing, removing, and a number of other functions that allow to add programs to an existing OS.

```
root@Master2:/home/igor# apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
Executing: /tmp/apt-key-gpghome.filaUgWqwk/gpg.1.sh --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
gpg: key B9733A7A07513CAD: public key "Jenkins (HDP Builds) <jenkins@hortonworks.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1
root@Master2:/home/igor# _
```

Figure 4.5-Adding a new repository for installing HDP

After adding the key, need to install Ambari server on the OS, for this need to run the following command as shown in Figure 4.6.

```

root@Master2:/home/igor# apt-get install ambari-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpq5 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libsensors4 postgresql
  postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common python
  python-minimal python2.7 python2.7-minimal ssl-cert sysstat
Suggested packages:
  lm-sensors postgresql-doc locales-all postgresql-doc-10 libjson-perl python-doc python-tk
  python2.7-doc binutils binfmt-support openssl-blacklist isag
The following NEW packages will be installed:
  ambari-server libpq5 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libsensors4
  postgresql postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common python
  python-minimal python2.7 python2.7-minimal ssl-cert sysstat
0 upgraded, 17 newly installed, 0 to remove and 5 not upgraded.
Need to get 398 MB of archives.
After this operation, 488 MB of additional disk space will be used.
Do you want to continue? [Y/n] _

```

Figure 4.6 – Installation of the required packages for HDP

After starting the installation, the necessary software package will be installed, which will allow to configure DHCP on the existing OS, and then deploy the necessary components. There were no errors at this stage, and all the tools were installed and deployed. In older versions, errors occurred when the virtual environment was incorrectly configured, where there was no support for the OpenJDK library. In new releases, the company has fixed this problem, and now the system automatically adds OpenJDK to the OS during installation. Also, the main components that were necessary for the correct installation and deployment of HDP were tools such as: curl, scp, wget, unzip, python.

After installing all the necessary dependencies is needed to run the Ambari server configuration, as shown in Figure 4.7. In the setup script, the first item is to install the Java platform.

```

root@Master2:/home/igor# ambari-server setup
Using python /usr/bin/python
Setup ambari-server
Checking SELinux...
WARNING: Could not run /usr/sbin/sestatus: OK
Customize user account for ambari-server daemon [y/n] (n)?
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
[1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
[2] Custom JDK
=====
Enter choice (1): 1
To download the Oracle JDK and the Java Cryptography Extension (JCE) Policy Files you must accept the
license terms found at http://www.oracle.com/technetwork/java/javase/terms/license/index.html and
t accepting will cancel the Ambari Server setup and you must install the JDK and JCE files manually
Do you accept the Oracle Binary Code License Agreement [y/n] (y)? y
Downloading JDK from http://public-repo-1.hortonworks.com/ARTIFACTS/jdk-8u112-linux-x64.tar.gz to /u
ar/lib/ambari-server/resources/jdk-8u112-linux-x64.tar.gz
jdk-8u112-linux-x64.tar.gz... 21% (37.6 MB of 174.7 MB)_

```

Figure 4.7 - Starting and Configuring Ambari Server

The next step in the script is to configure which database wants to use, by default, as shown in Figure 4.8, the Postgre Sql database will be used.

```

Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)?
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)?
Configuring database...
Default properties detected. Using built-in database.
Configuring ambari database...
Checking PostgreSQL...
Configuring local database...
Configuring PostgreSQL...
Restarting PostgreSQL
Creating schema and user...
done.
Creating tables...

```

Figure 4.8-Configuring PostgreSQL

If the Ambari server is successfully configured, a message will be sent to the console indicating that the configuration operation was completed successfully, as shown in Figure 4.9.

```

...
Ambari repo file contains latest json url http://public-repo-1.hortonworks.com/HDP/hdp_urlinfo.json,
updating stacks repoinfos with it...
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.
root@Master2:/home/igor# _

```

Figure 4.9 - Successful completion of Ambari Server startup

After successful installation of the Samba server, must install the agent to connect the VM to the server, as shown in Figure 4.10.

```

root@Master2:/home/igor# apt-get install ambari-agent
Reading package lists... Done
Building dependency tree
Reading state information... Done

```

Figure 4.10-Installing the Ambari Agent

Then need to enable the daemon, which will start the server on the VM, and will enable the server, where it will be possible to connect to the web interface, as shown in Figure 4.11.

```

root@Master2:/home/igor# ambari-server start
Using python /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
root@Master2:/home/igor# _

```

Figure 4.11 - Starting the Ambari Server

Next, must also enable the Ambari agent in order to connect the machine to the server in the future, for this need to run the startup command, as shown in Figure 4.12.

```

root@Master2:/home/igor# ambari-agent start
Verifying Python version compatibility...
Using python /usr/bin/python
Checking for previously running Ambari Agent...
Checking ambari-common dir...
Starting ambari-agent
Verifying ambari-agent process status...
Ambari Agent successfully started
Agent PID at: /run/ambari-agent/ambari-agent.pid
Agent out at: /var/log/ambari-agent/ambari-agent.out
Agent log at: /var/log/ambari-agent/ambari-agent.log
root@Master2:/home/igor#

```

Figure 4.12 - Launching the Ambari Agent

Next, to configure the DHCP server, need to connect via the web interface and need to type a username and password to log in to the server via a web browser, as shown in Figure 4.13.

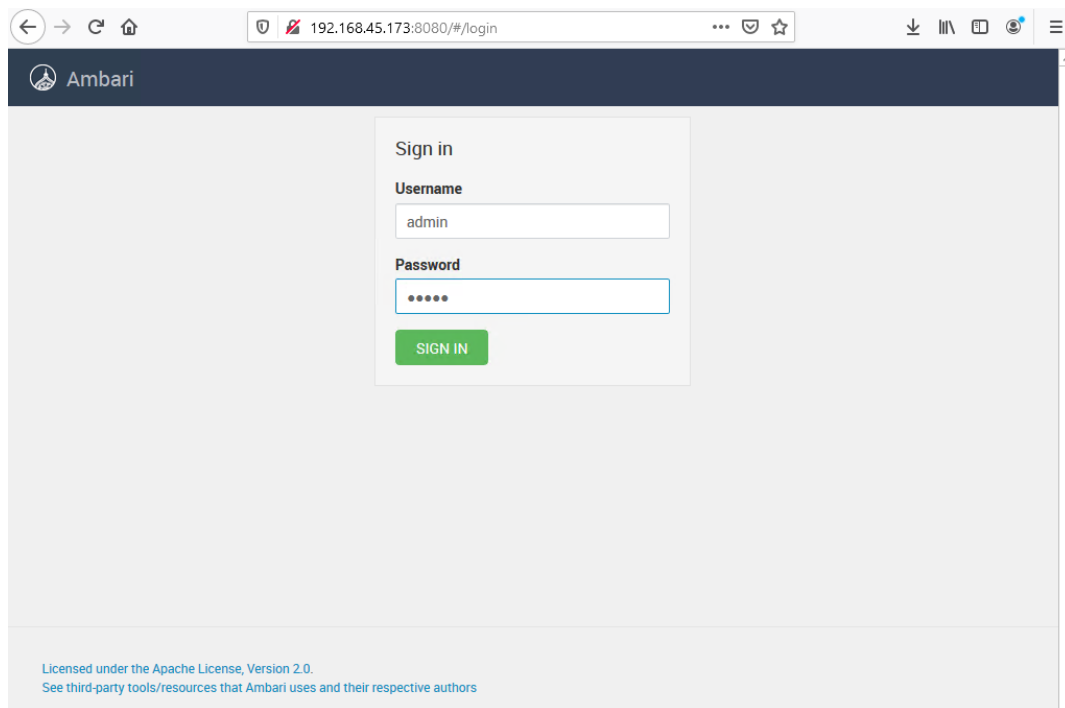


Figure 4.13 - Connecting to Ambari via the web interface

After logging in to the server, will be prompted to create a cluster. To do this, run the Launch Install Wizard as shown in Figure 4.14.

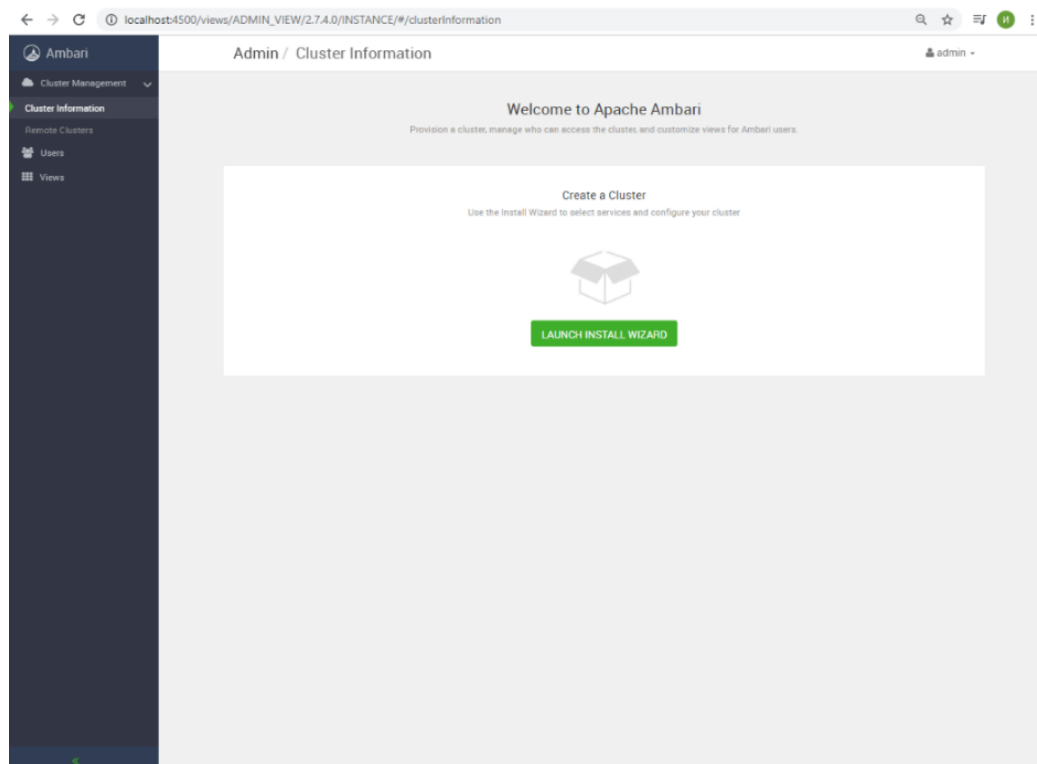


Figure 4.14 - Connecting to the Ambari server via the web interface

Next, need to specify the name of the cluster, as shown in Figure 4.15.

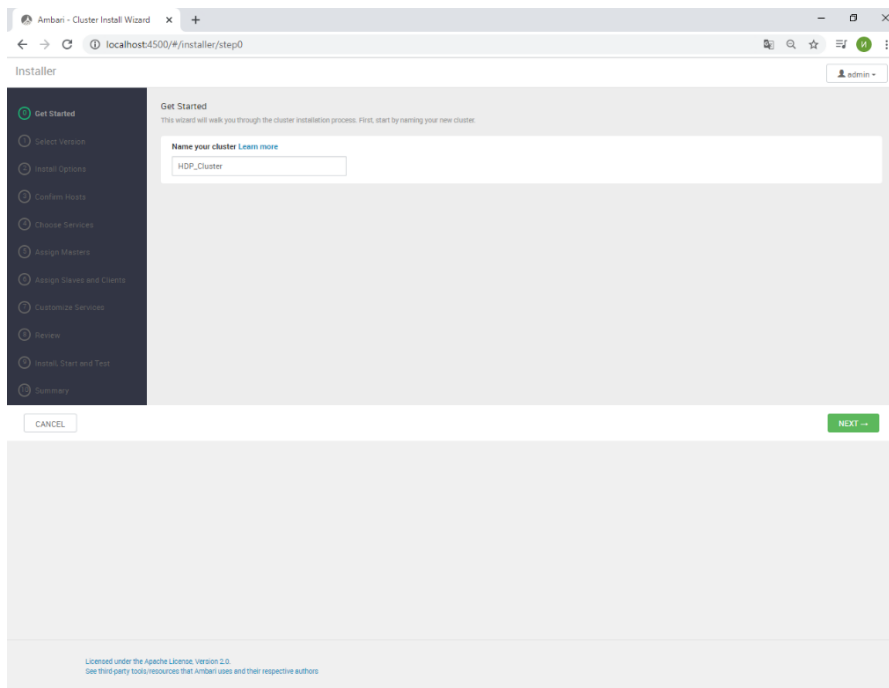


Figure 4.15 - Specifying a name for the new cluster

Then need to choose which OS and which batch file is needed, when installing, the repository for Ubuntu 18.04 was selected as shown in Figure 4.16. HDP supports many distributions to install the necessary tools, but the installation method will be different for different operating systems, because of this, Hortonworks has created many repositories for installation.

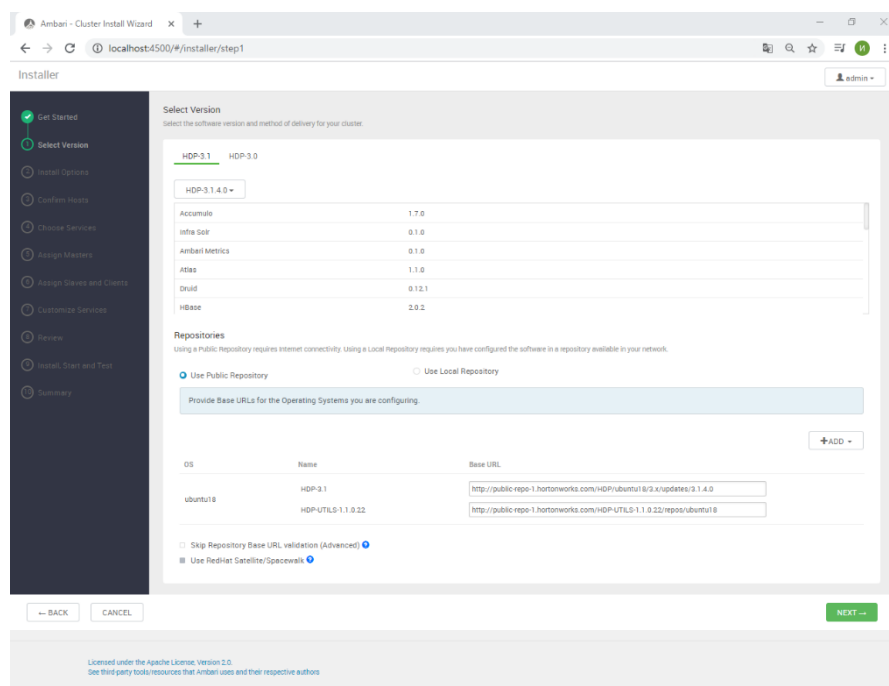


Figure 4.16 - Selecting a repository for the OS

After selecting the repository, need to connect the machines on which the agent is installed. If the system does not have a DNS server, then need to type the IP address instead of the DNS name, as shown in Figure 4.17.

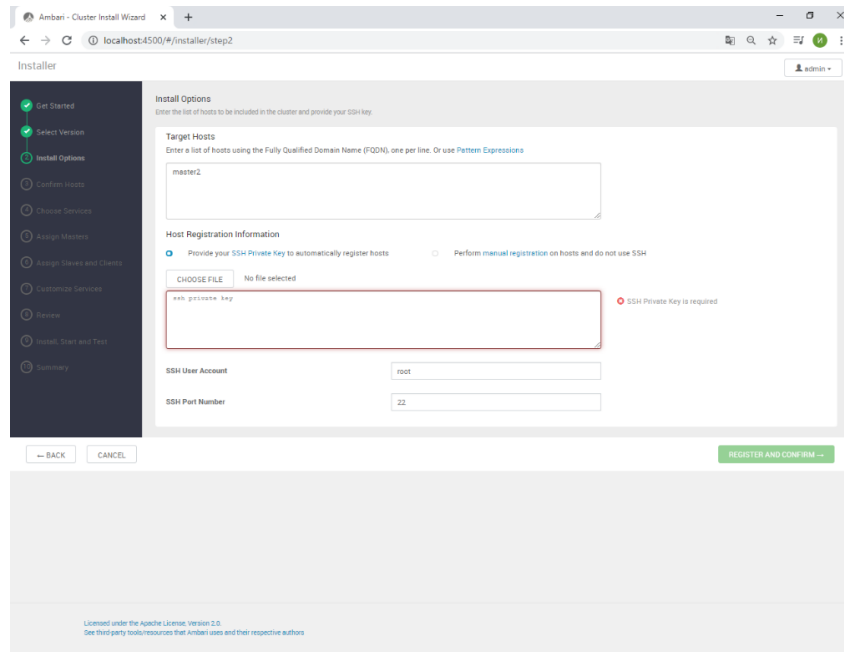


Figure 4.17-Adding nodes to a new HDP cluster

If the addition is successful, the machine will be positively added to the cluster as shown in figure 4.18.

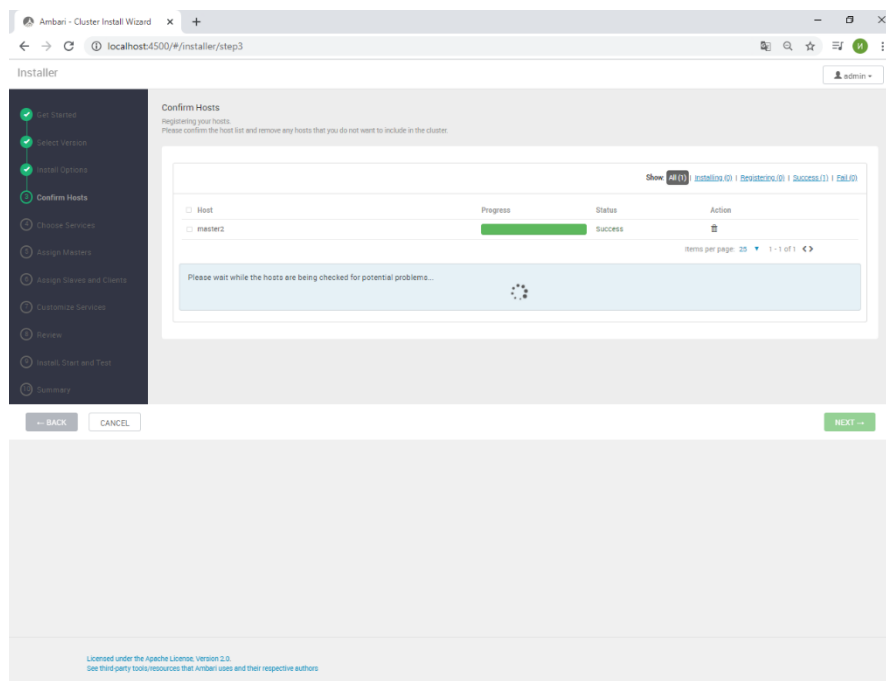


Figure 4.18 - Information about the successful addition of a new machine to the HDP cluster

The next window allows to select which core components will be used in the cluster, where can select only the core components at the beginning, and add them as needed, as shown in Figure 4.19.

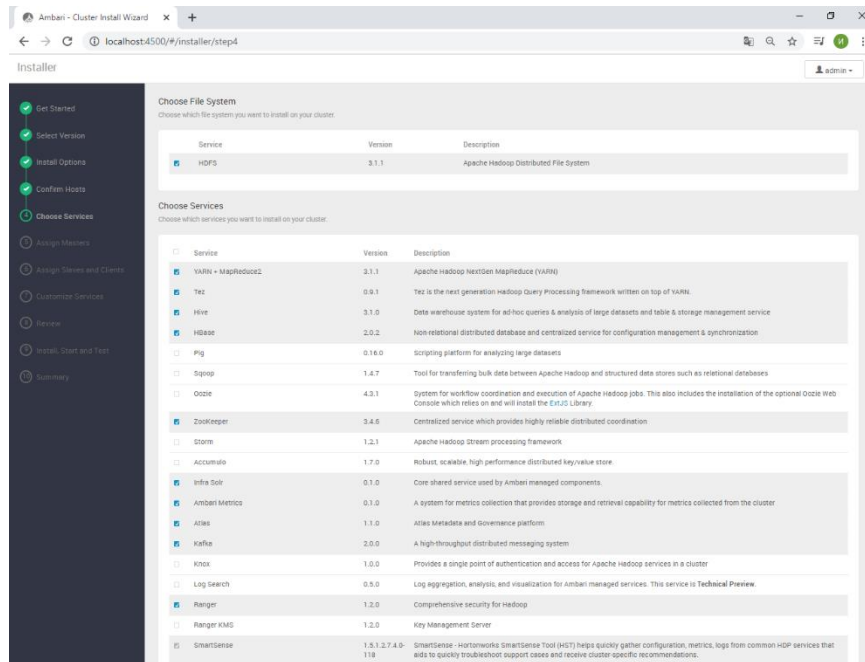


Figure 4.19-Selecting the necessary components for working with the database in an HDP environment

After all components are added, a list of the components that have been selected in Figure 4.20 as a result of the installation scenario is obtained.

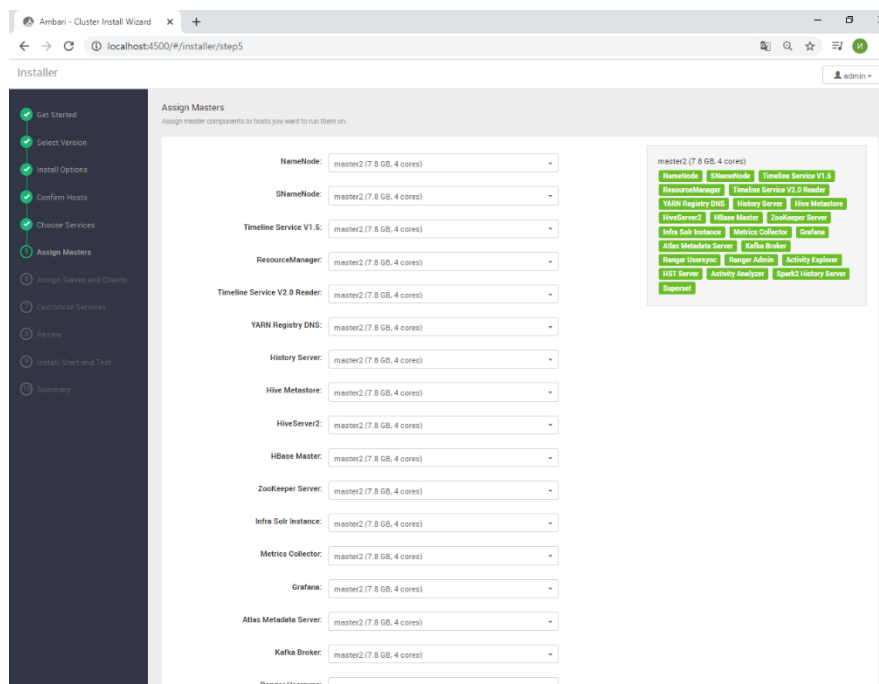


Figure 4.20-Selecting the necessary components for working with the database in an HDP environment

Next, a window appears in which need to configure the database to work with various tools, where configure access to the MySQL database to work together with Hive, as shown in Figure 4.21.

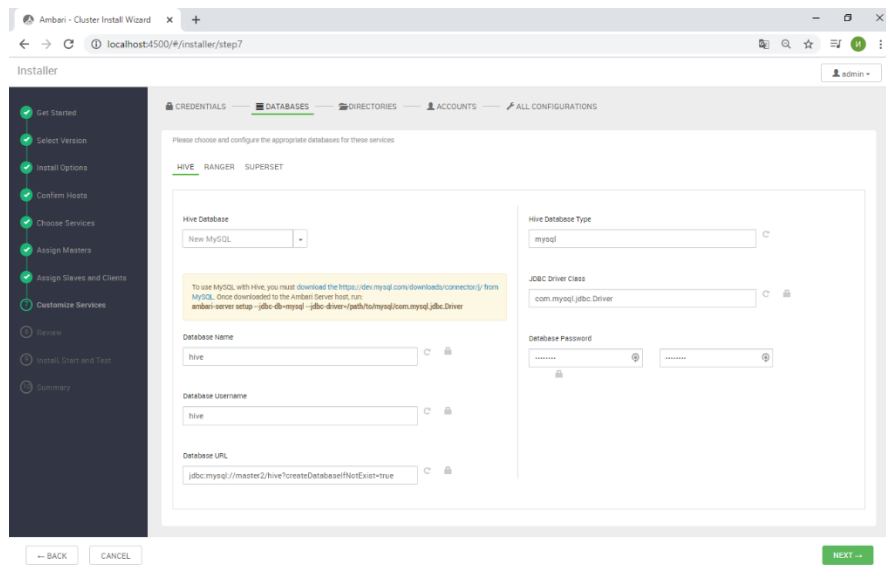


Figure 4.21-Configuring the database to work with HDP

It also needs to configure which directories will store cluster information on the server, as shown in figure 4.22. It is necessary to specify exactly where the configuration files will be stored in the local FS, and to specify these paths in the appropriate fields as shown in figure. 4. 22.

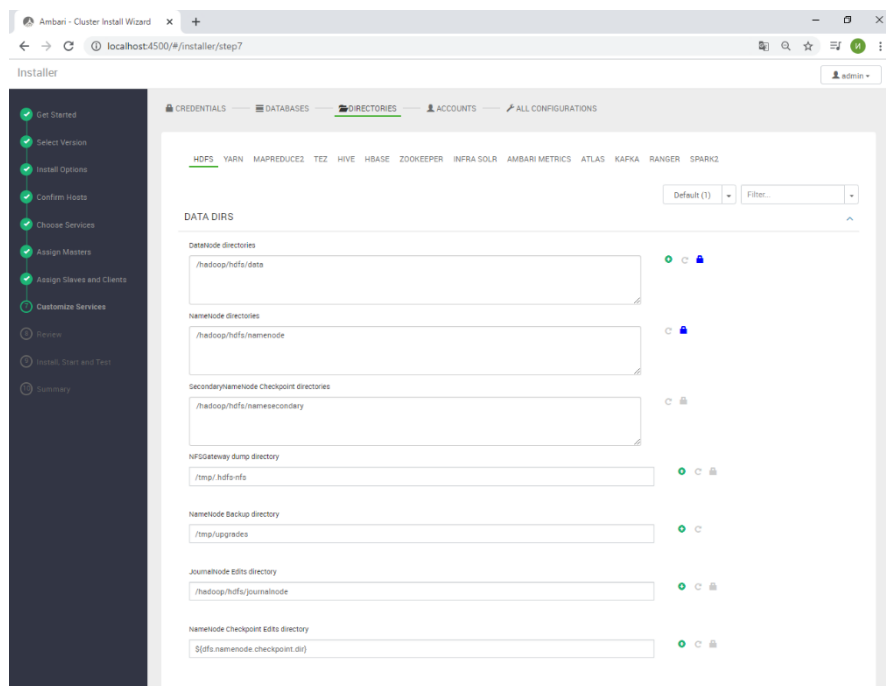


Figure 4.22 – Configuration directory for storing configuration files of the components of the HDP

Also need to check which users will manage certain services, and if necessary can change the users and user rights, as shown in Figure 4.23.

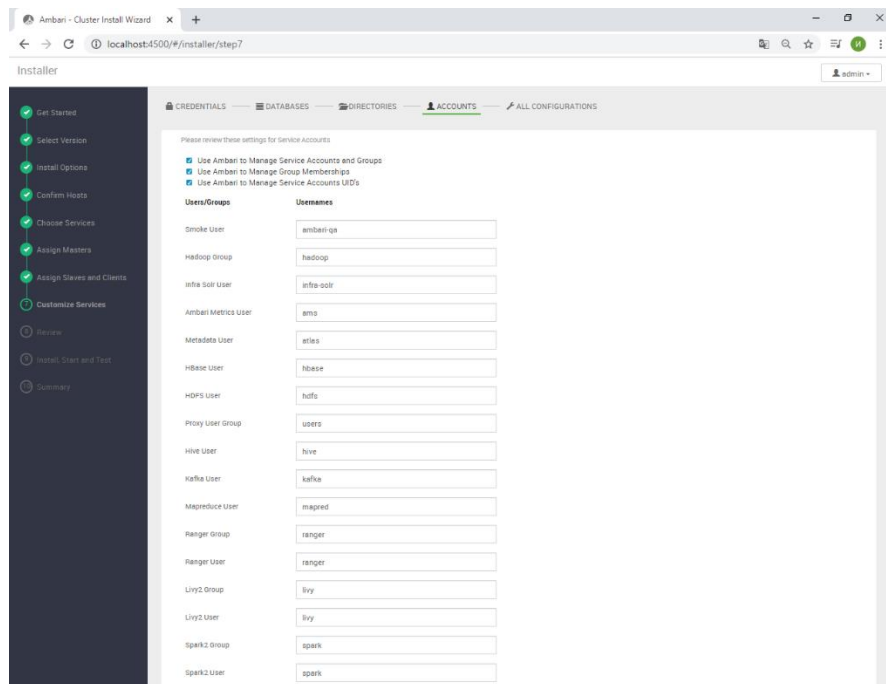


Figure 4.23-Configuring users to work with HDP tools

In Figure 4.24 is needed to specify how many resources will be allocated to the services for operation.

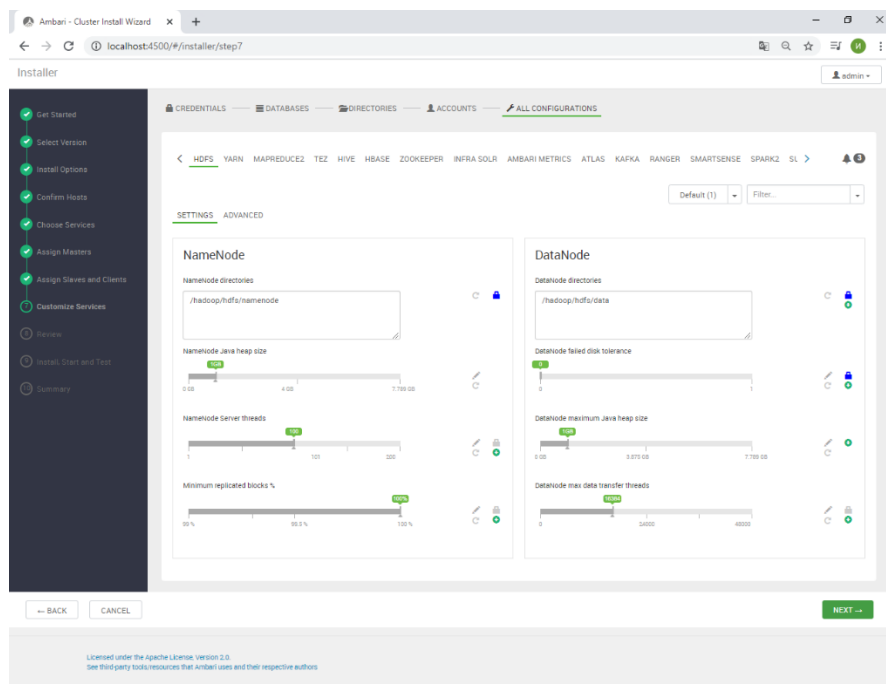


Figure 4.24-Allocating resources to HDP applications

The next step is need to run the installation of all the selected components that were configured as a result of the installation script, as shown in Figure 4.25.

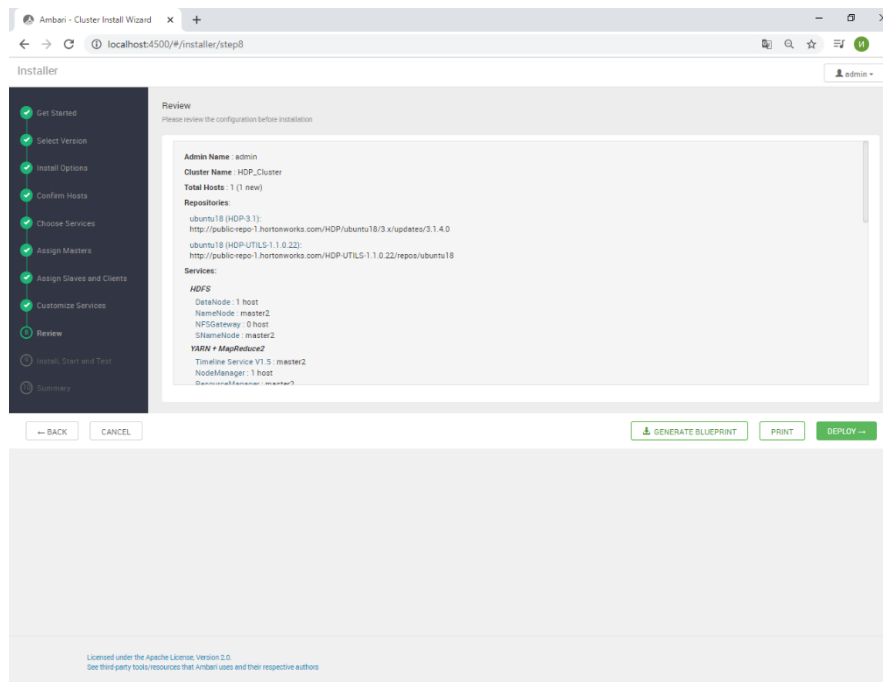


Figure 4.25-Completing the configuration of all HDP components

As a result of HDD installation and further configuration, a database management and processing system was deployed in the HDP environment. Basic tools were added, as well as a cluster manager for managing nodes for creating distributed tasks and sending them to subordinate YARN nodes. The Apache Spark distributed data processing tool has also been added. When configuring HDP, a tool was added to work with data using SQL queries of a similar language using Apache Hive. This tool will allow to manage data much more efficiently in the future due to the familiar HiveQL management system, which is an advanced query language similar to SQL.

During installation and deployment, no problems were identified, all the necessary data for deployment was contained on the company's portal, which significantly helped when configuring some services and software products included in the HDP installation package.

4.4 Shell deployment for interactive work in Apache Zeppelin

To be able to write a program for processing web server logs need a tool that allows to develop interactively. This feature is also necessary for writing programs in various programming languages, such as Scala, Java, and Python. This feature is supported by Apache Zeppelin, and the process of adding Apache Zeppelin to HDP will be described below.

To add Apache Zeppelin to the HDP repository must select a service, as shown in Figure 4.26.

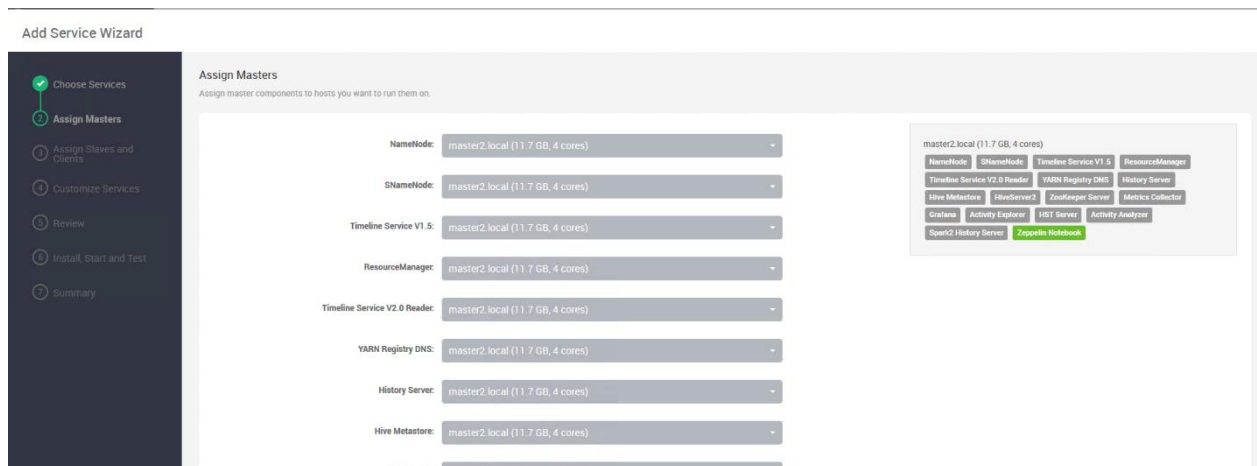


Figure 4.26-Adding Apache Zeppelin to HDP

The next step is need to check the configuration files, as shown in Figure 4.27, then click the "Next" button.

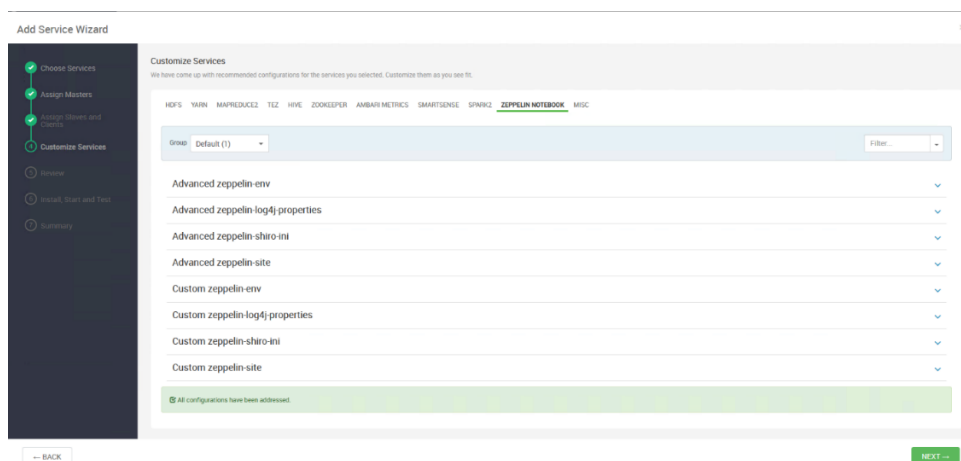


Figure 4.27-Configuring Apache Zeppelin in HDP

The following Figure 4.28 shows the final information before installing Zeppelin on the HDP platform.

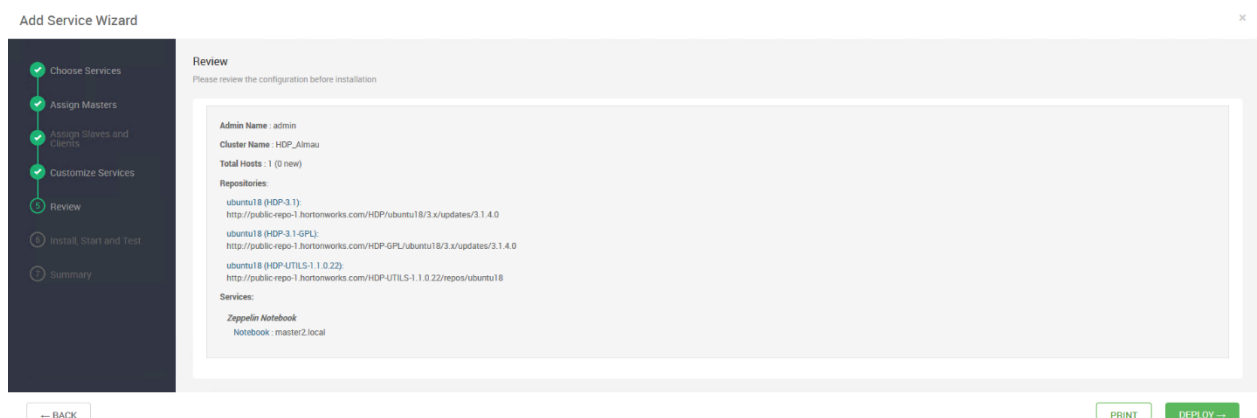


Figure 4.28-Checklist of all configured Apache Zeppelin components

After the installation starts, as shown in Figure 4.29, must wait for the Apache Zeppelin installation to complete.

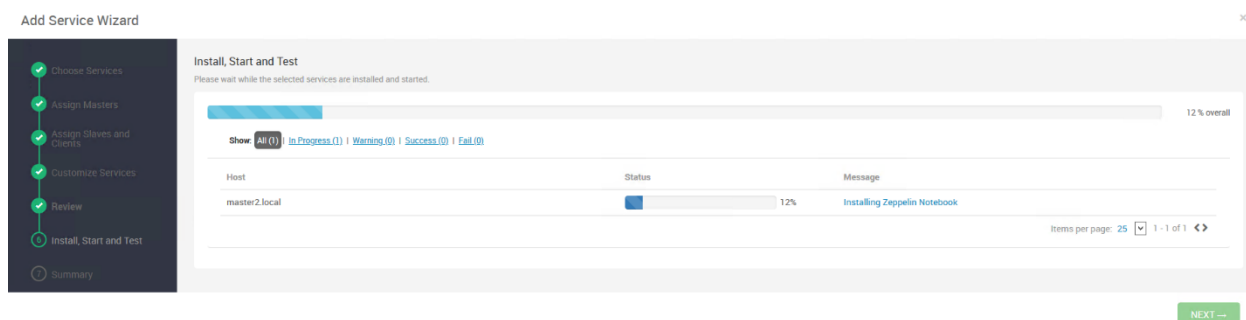


Figure 4.29 - The Process of installing Apache Zeppelin

To connect to Apache Zeppelin is need to go to the address and a specific port in the browser, then the menu will appear as shown in Figure 4.30.

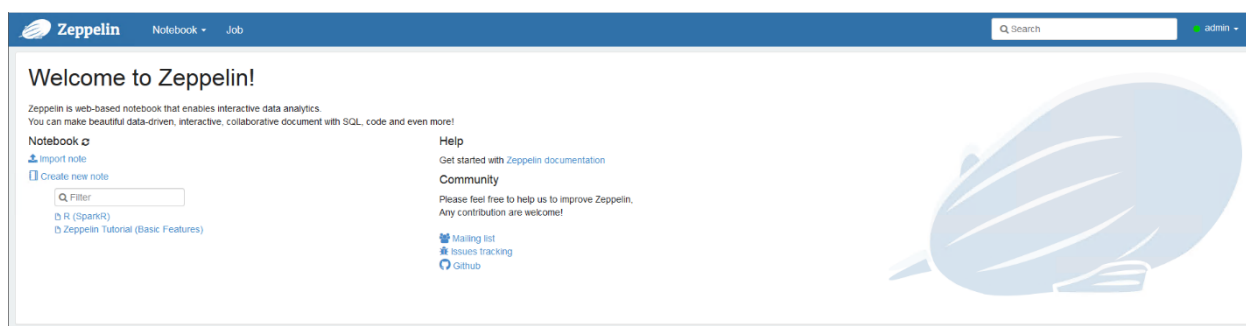


Figure 4.30-Connecting to Apache Zeppelin via the Web interface

To work with programs in Spark need to create a new diary, as shown in Figure 4.31.

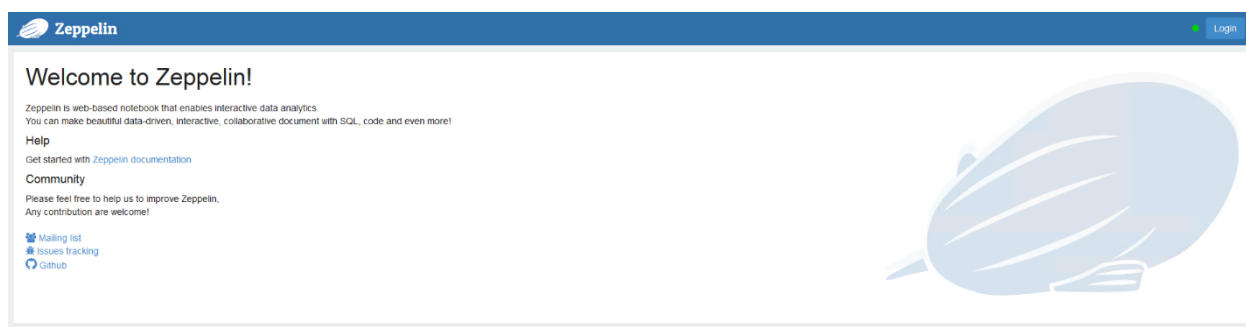


Figure 4.31 - Creating a Diary for writing code in the Apache Zeppelin environment

Next, a window will appear to create a new notepad for work, must specify the name as shown in Figure 4.32, and how exactly the incoming data will be processed.

Figure 4.32-Name of the new diary in Apache Zeppelin

After that can go to the newly created notepad to write code for data processing as in Figure 4.33.

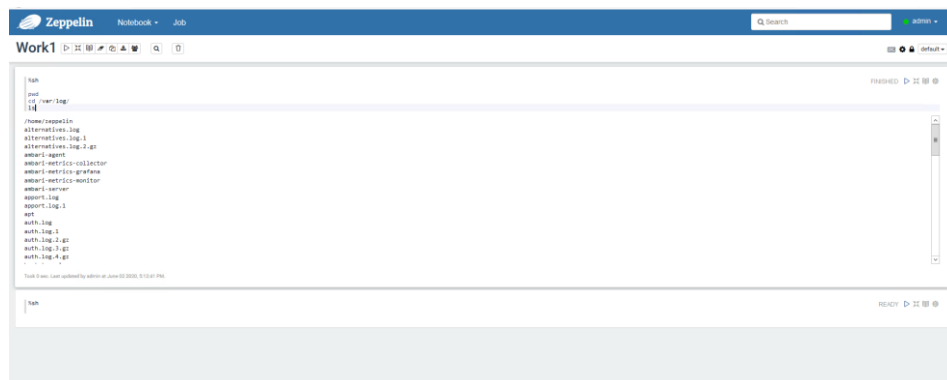


Figure 4.33-Name of the new diary in Apache Zeppelin

Apache Zeppelin was deployed on the HDP platform. It also set up plugins to work with the Python programming language, and added a plugin to work with Apache Spark. As a result of the configuration, problems were encountered in creating an environment variable in the HDP environment. This problem was solved by writing a script that allows to automatically add paths to variables for programs such as Apache Spark, and the Python programming language. This tool will help to present the data in a convenient visualization format. Apache Zeppelin has the capability of connecting an additional module to work with scripts in bash command shell language, which allows not to switch between the OS terminal and Apache Zeppelin web interfaces.

The platform is now ready for web server log analysis, and the full software package is ready for conducting and studying anomalous system traffic and scheduling web server loads.

4.5 Analysis of Nginx Web server logs

The logs of the NGINX and Apache web servers will be used as input data, as these are the two web servers that have the largest distribution in the world, the market share that occupies the NGINX web server is 19.14% of the active sites. The market share of Apache sites is 45.6% of active sites, which makes these web servers the number one in the world.

Data processing will be done using two different approaches, the first one will be processed using python, in conjunction with the Apache Spark distributed data processing tool. Visualization will be done with Apache Zeppelin built-in tools, using the SQL query language. The loaded set date will be parsed and placed in the SQL structure, using pyspark.sql library.

The next method of data processing will be carried out using python modules for visualization and data processing, such modules include the pandas module that allows data analysis, cleaning and static processing. Also used are tools like the matplotlib library, a library that allows to visualize data, graphs that can be built using matplotlib allow to accurately visualize data. The NumPy library allows to process data using various mathematical functions. The Seaborn library was developed on the basis of the python matplotlib library, and has a more convenient visualization view compared to the matplotlib library. For more complex visualizations, much less code is needed compared to other visualization tools.

Both methods will use Apache Perl and python for processing, but when providing data for analysis and visualization, the processing approach will be different. As a result of processing, a comparative analysis will be carried out by the amount of time spent on data processing in the Apache Spark distributed processing system. 3 sets of data with NGINX web server access log data will be taken for processing.

Uploading data to the server is done with the Ansible tool. The workflow of the system is shown in Figure 4.34, first the logs are loaded onto the HDP server, into the distributed HDFS file system, followed by data analysis, and visualization of the results.

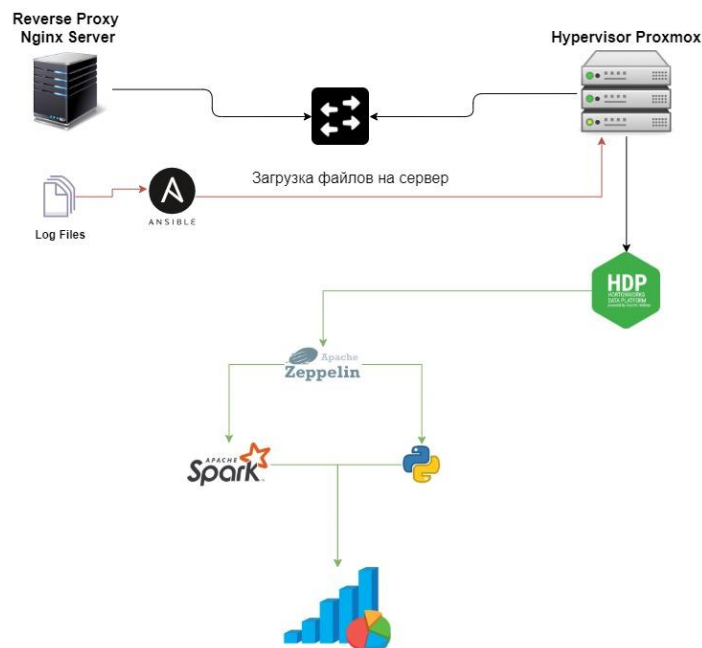


Figure 4.34-Data processing and loading scheme in HDP

There are many logs that store data about visits to certain IP addresses. The logs directly store information about the number of errors during the site access process. The amount of information received and uploaded when working with the university's websites is also stored. This data must be transmitted to the HDP server for further processing and visualization of this data.

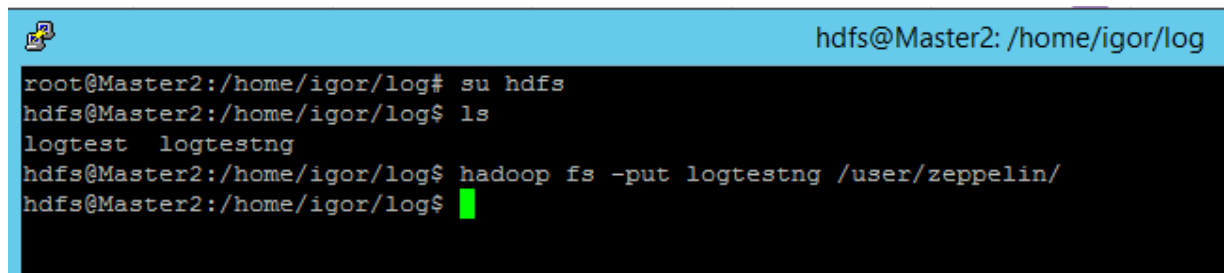
Below is the first approach to log processing using data representation via the SQL query language, and data visualization using the built-in Apache Zeppelin library. Figure 4.35 shows the log files, as well as all actions performed on the web server. There are logs that record errors when accessing the web server, which are presented separately.

access.log	151516K	мая 6 14:33
access.log.1	360690K	мая 6 00:00
access.log.10.gz	18661K	апр 27 00:00
access.log.11.gz	19782K	апр 26 00:00
access.log.12.gz	24053K	апр 25 00:00
access.log.13.gz	26968K	апр 24 00:00
access.log.14.gz	29957K	апр 23 00:00
access.log.15.gz	25270K	апр 22 00:00
access.log.16.gz	21929K	апр 21 00:00
access.log.17.gz	11629K	апр 20 00:00
access.log.18.gz	12654K	апр 19 00:00
access.log.19.gz	17417K	апр 18 00:00
access.log.2.gz	31306K	мая 5 00:00
access.log.20.gz	21950K	апр 17 00:00
access.log.21.gz	19610K	апр 16 00:00
access.log.22.gz	23352K	апр 15 00:00
access.log.23.gz	22955K	апр 14 00:00
access.log.24.gz	16044K	апр 13 00:00
access.log.25.gz	12387K	апр 12 00:00
access.log.26.gz	18362K	апр 11 00:00
access.log.27.gz	17613K	апр 10 00:00
access.log.28.gz	18555K	апр 9 00:00
access.log.29.gz	20641K	апр 8 00:00
access.log.3.gz	20185K	мая 4 00:00
access.log.30.gz	22043K	апр 7 00:00
access.log.4.gz	24905K	мая 3 00:00
access.log.5.gz	25638K	мая 2 00:00
access.log.6.gz	41556K	мая 1 00:00
access.log.7.gz	41219K	апр 30 00:00
access.log.8.gz	56214K	апр 29 00:00
access.log.9.gz	35605K	апр 28 00:00
error.log	17601	мая 6 11:41
error.log.1	17118	мая 5 20:23
error.log.10.gz	6602	апр 26 23:43

Figure 4.35-Data processing and loading scheme in HDP

Apache Zeppelin will be used as a tool for processing log files of Apache and Nginx web servers, this tool allows to analyze files using spark technology, python programming language, the ability to work using SQL language through the Hive tool for data processing. The advantage of this technology is the use of a distributed file system, as well as the use of distributed RDD technology when processing tasks.

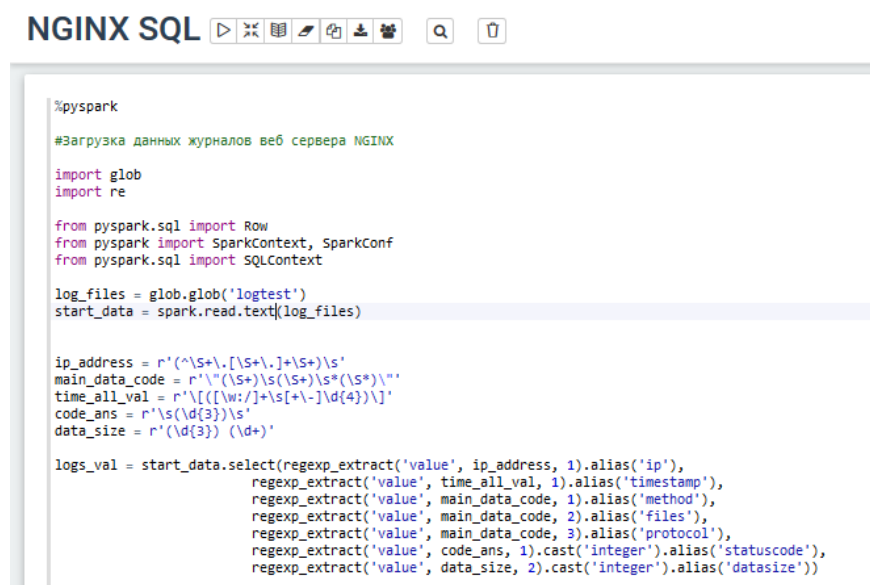
Log files are stored on the reverse proxy server. To access them, one must transfer it to HDFS. To transfer files, type the Hadoop fs-put file command, as shown in Figure 4.36.



```
hdfs@Master2: /home/igor/log
root@Master2:/home/igor/log# su hdfs
hdfs@Master2:/home/igor/log$ ls
logtest  logtestng
hdfs@Master2:/home/igor/log$ hadoop fs -put logtestng /user/zeppelin/
hdfs@Master2:/home/igor/log$
```

Figure 4.36-Loading data to HDFS

The described program allows to work with log files of web servers, such as Nginx and Apache web servers. This program allows to split files into several segments, as shown in Figure 4.37.



```
NGINX SQL
%pyspark
#Загрузка данных журналов веб сервера NGINX

import glob
import re

from pyspark.sql import Row
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext

log_files = glob.glob('logtest')
start_data = spark.read.text(log_files)

ip_address = r'^(\S+\.){1,4}\S+\s$'
main_data_code = r'^"(\S+)\s(\S+)\s*(\S*)"'
time_all_val = r'^[([w:/]+[S+[-]\d{4})\]]'
code_ans = r'^\s\d{3}\s$'
data_size = r'^(\d{3}) (\d+)'

logs_val = start_data.select(regex_extract('value', ip_address, 1).alias('ip'),
                                regex_extract('value', time_all_val, 1).alias('timestamp'),
                                regex_extract('value', main_data_code, 1).alias('method'),
                                regex_extract('value', main_data_code, 2).alias('files'),
                                regex_extract('value', main_data_code, 3).alias('protocol'),
                                regex_extract('value', code_ans, 1).cast('integer').alias('statusCode'),
                                regex_extract('value', data_size, 2).cast('integer').alias('datasize'))
```

Figure 4.37-Log Processing Program

As a result of processing the web server logs, the following results were built. Figure 4.38 shows the analysis of incoming data for a certain period of time, which allows to determine the amount of downloaded data that is received by the web server.

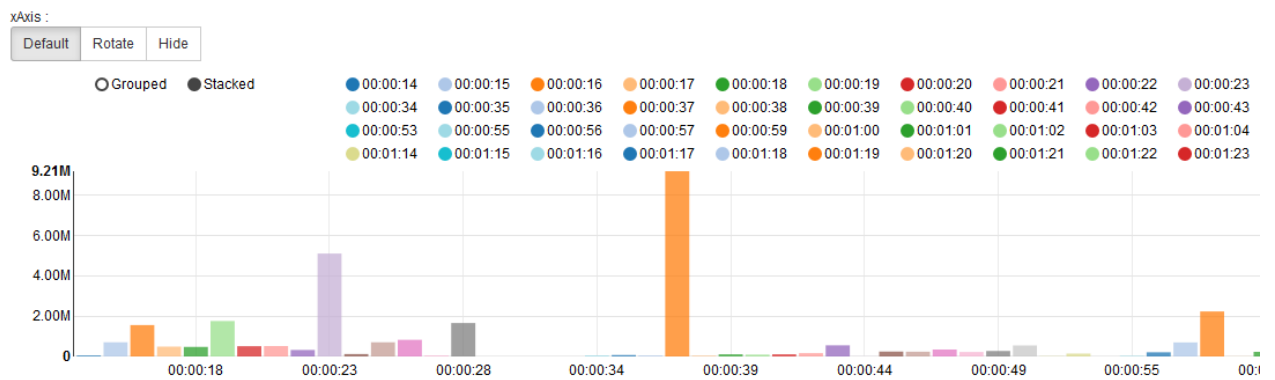


Figure 4.38 - Analysis of incoming traffic in the time interval

Analysis of files on the web server accessed by users, where this analysis allows to determine how much information has been downloaded or uploaded to these files. Thus, it is possible to track abnormal indicators as in Figure 4.39.

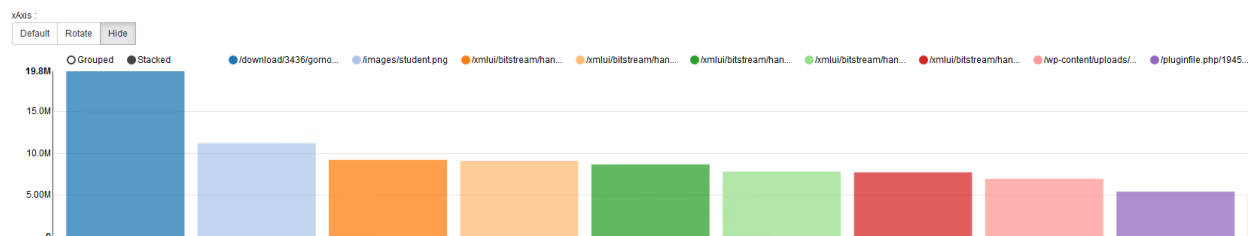


Figure 4.39-Resources most often involved in content transfers

The web server response codes are also analyzed to determine how many errors were generated by the web server when working with web resources, as shown in Figure 4.40.

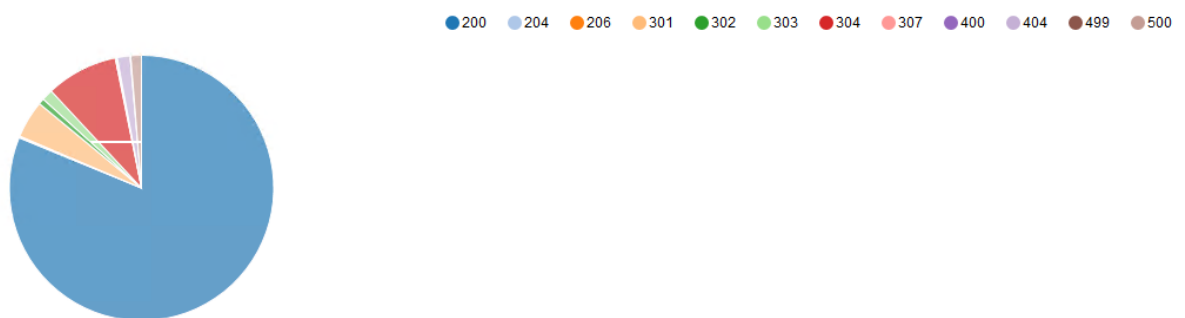


Figure 4.40-Analysis of web server response codes

For a more accurate analysis of code errors, it is possible to analyze the error codes on a time graph, where can see the number of errors issued by the web server for a certain period of time, this graph is shown in Figure 4.41.

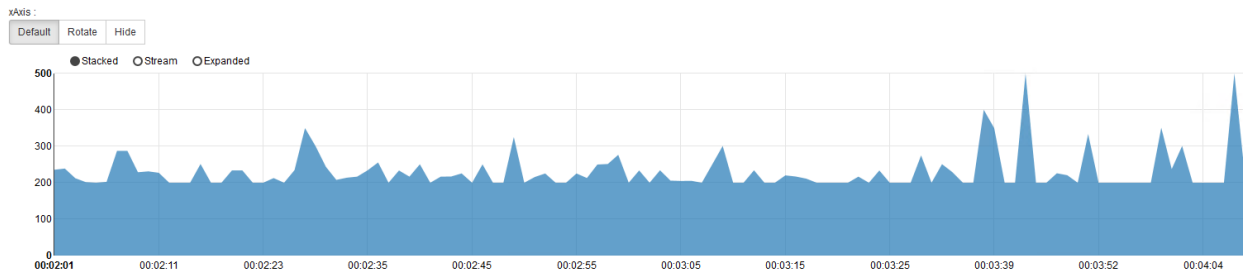


Figure 4.41-Analysis of web server error codes in the time interval

Also, an analysis was carried out on visiting web resources, to determine from which IP addresses users most often accessed web resources on the web server. Figure 4.42 shows a diagram for visualizing client IP address data.

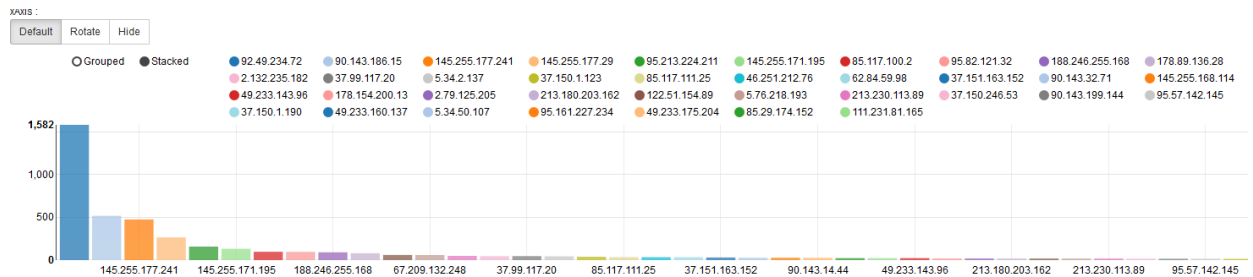


Figure 4.42 - IP addresses of users who most often visit web resources

The data was processed and placed in a tabular model, the analysis was performed using the SQL query language, and data visualization was performed using the built-in Apache Zeppelin tools to provide data to analysts or users.

The advantages of this approach in data processing, ease of data analysis for specialists who are familiar with the SQL query language, but do not have enough knowledge of database processing tools, which reduces the threshold for entering into working with databases and their analysis. The disadvantages of using this approach are the inability to process data using mathematical models, where visualization problems arise with scalability, since it is impossible to output huge values.

The next method of data processing will take place using various tools and libraries for working with the database in the python environment.

Initially, one need to load the dataset with data for processing, in the Apache Zeppelin environment as shown in Figure 4.43.

```
log_files = glob.glob('*.gz')
start_data = spark.read.text(log_files)
```

Figure 4.43 - Adding data to the Data Frame.

After placing the data on the Apache Zeppelin environment is needed to check whether the data was actually added to the variable as in Figure 4.44 for further processing of the loaded data.

```
log_files = glob.glob('*.gz')
start_data = spark.read.text(log_files)

log_files

['access.log.21.gz', 'access.log.19.gz', 'access.log.26.gz', 'access.log.29.gz', 'access.log.23.gz', 'access.log.3.gz', 'access.log.18.gz', 'access.log.9.gz', 'access.log.25.gz', 'access.log.2.gz', 'access.log.27.gz', 'access.log.12.gz', 'access.log.15.gz', 'access.log.8.gz', 'access.log.6.gz', 'access.log.17.gz']
```

Figure 4.44 - Checking the loaded dataset.

After loading the data need to clear the received it, for this, regular expressions will be used, as shown in Figure 4.45. And is needed to split the data into columns in order to separate it from each other. The next step is need to check what data were received in the new dataset.

```
%pyspark

'''Необходимо разбить данные на несколько частей, при помощи регулярных выражений
Для удобной обработки данных.'''

from pyspark.sql.functions import regexp_extract

ip_address = r'^\S+\.([\S+\.]+\S+)\s'
main_data_code = r'^([\S+)\s([\S+)\s*([\S+)\s'
time_all_val = r'^\{([W:/]+\s[+-]\d{4})\}'
code_ans = r'^\s(\d{3})\s'
data_size = r'^\s(\d{3})\s(\d+)'

logs_val = start_data.select(regexp_extract('value', ip_address, 1).alias('ip'),
                             regexp_extract('value', time_all_val, 1).alias('timestamp'),
                             regexp_extract('value', main_data_code, 1).alias('method'),
                             regexp_extract('value', main_data_code, 2).alias('files'),
                             regexp_extract('value', main_data_code, 3).alias('protocol'),
                             regexp_extract('value', code_ans, 1).cast('integer').alias('statuscode'),
                             regexp_extract('value', data_size, 2).cast('integer').alias('datasize'))

logs_val.show(10, truncate=True)
print((logs_val.count(), len(logs_val.columns)))
```

ip	timestamp	method	files	protocol	statuscode	datasize
95.58.33.101	13/May/2020:00:00...	GET	/img/slider/almau...	HTTP/1.1	200	766183
95.58.33.101	13/May/2020:00:00...	GET	/img/slider/music...	HTTP/1.1	200	455
95.58.33.101	13/May/2020:00:00...	GET	/img/slider/music...	HTTP/1.1	200	1281
95.58.33.101	13/May/2020:00:00...	GET	/images/uploads/c...	HTTP/1.1	200	168850
54.36.150.100	13/May/2020:00:00...	GET	/xmlui/search-fil...	HTTP/1.1	200	31887
95.58.33.101	13/May/2020:00:00...	GET	/img/slider/megac...	HTTP/1.1	200	1683660
95.58.33.101	13/May/2020:00:00...	GET	/images/uploads/9...	HTTP/1.1	200	62145
95.58.33.101	13/May/2020:00:00...	GET	/images/uploads/a...	HTTP/1.1	200	85734
95.58.33.101	13/May/2020:00:00...	GET	/img/slider/slide...	HTTP/1.1	200	880052
95.58.33.101	13/May/2020:00:00...	GET	/images/uploads/2...	HTTP/1.1	200	150849

Figure 4.45 - Processing data using regular expressions.

Analysis of the response codes of the web server, to obtain information about the total number of responses issued by the server during the entire operation time, as shown in Figure 4.46.

```

sns.catplot(x='statuscode', v='count', data=logs_st_code_val, kind='bar', order=logs_st_code_val['statuscode'].heis
<seaborn.axisgrid.FacetGrid object at 0x7fa272186050>

```

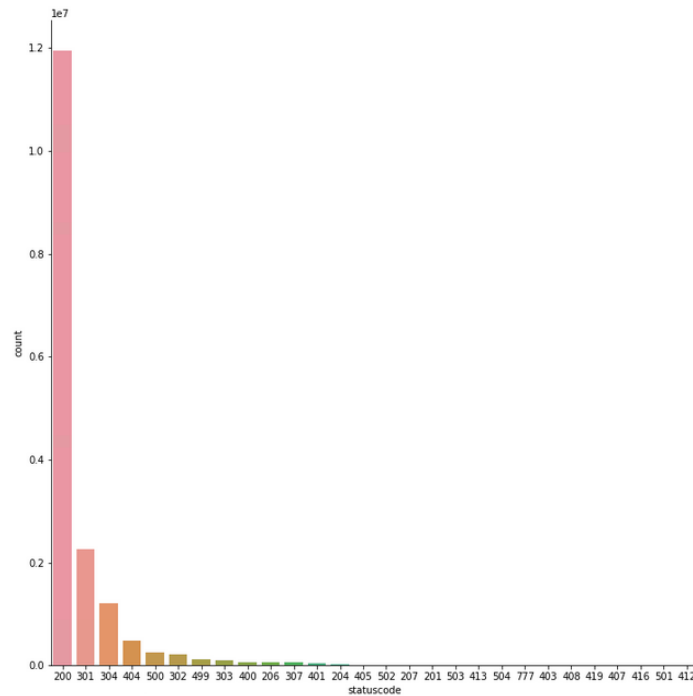


Figure 4.46 - Getting data about all server response codes.

For a more convenient representation of the data of the web server response codes, it is necessary to reduce the values to the natural log logarithm as shown in Figure 4.47, so the average values will be obtained.

```

xruspark
#Визуализация ответов веб сервера после приведения данных при помощи функции Log
plot_logs_val = (logs_st_log_val.toPandas().sort_values(by=['log(count)'],ascending=False))
sns.catplot(x='statuscode', y='log(count)', data=plot_logs_val, kind='bar', order=logs_st_code_val['statuscode'], height=10)
<seaborn.axisgrid.FacetGrid object at 0x7fa272842150>

```

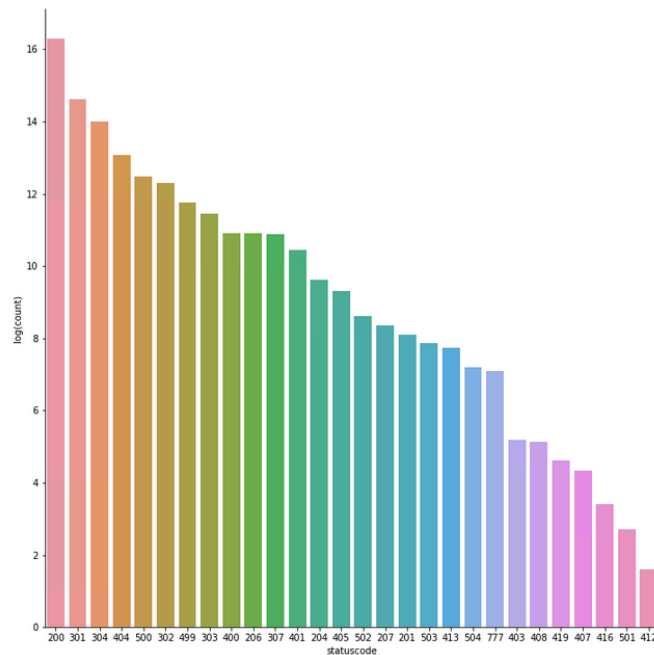


Figure 4.47 - Getting data about all server response codes in the loge section.

Analysis of the number of IP addresses that accessed the web server as shown in Figure 4.48. The graph shows the total number of unique IP addresses.

```
%pyspark
#Анализ ip адресов, которые чаще всего обращались к веб серверу
ip_val=(logs_val.groupBy('ip').count().sort('count', ascending=False).limit(20))
ip_val.show(truncate=False)
```

ip	count
84.38.181.130	1089358
95.213.224.211	448130
95.161.227.234	215533
82.148.17.75	164934
178.154.200.13	121008
213.180.203.162	107192
82.200.154.115	90506
217.196.26.18	88996
122.51.235.220	59697
111.231.81.165	57874
178.88.46.28	56956
122.51.154.89	54669
49.233.143.96	53460
91.108.6.135	52982
49.233.175.204	50154
37.150.26.178	35660
5.250.129.218	35363
188.0.131.116	32533
188.0.132.22	29803
90.143.25.106	29025

Figure 4.48 - Counting the total number of IP addresses accessing the web server.

Based on the presented data, one can make a forecast about how often the user accessed from a certain IP address, a large number of requests from a certain address can indicate that attacking requests may occur, and that it is necessary to protect the web server from attacks and isolate the attacking IP address.

Figure 4.49 shows the analysis of the error code 404, which is issued by the server, the analysis is made in the context of the month, shows the peak values for issuing errors by day.

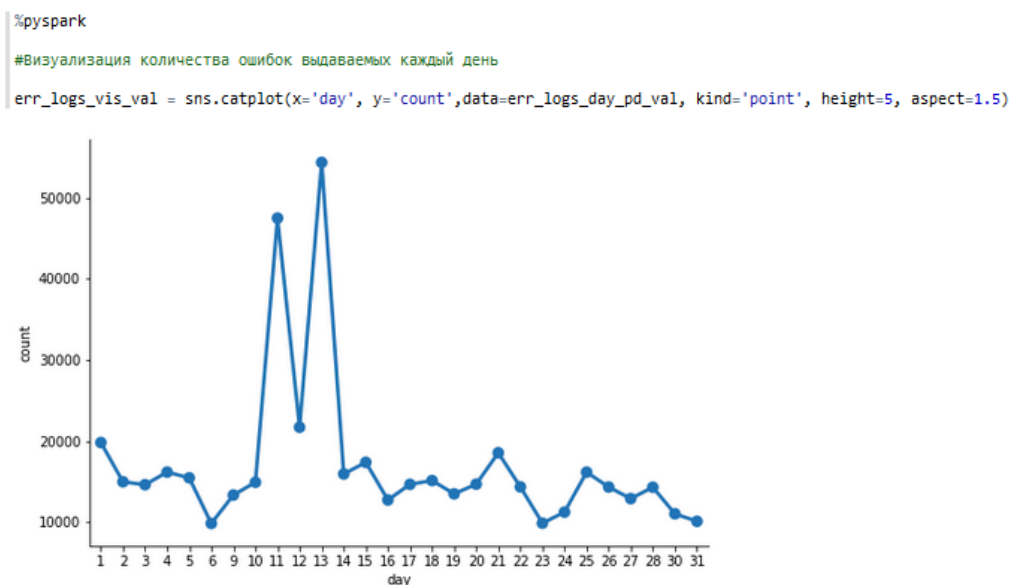


Figure 4.49 - Analysis of errors by code 404 issued by the web server.

Analyzing the 404 error codes in Figure 4.50 allows to get information about which hour had the highest value of errors issued by the web server.

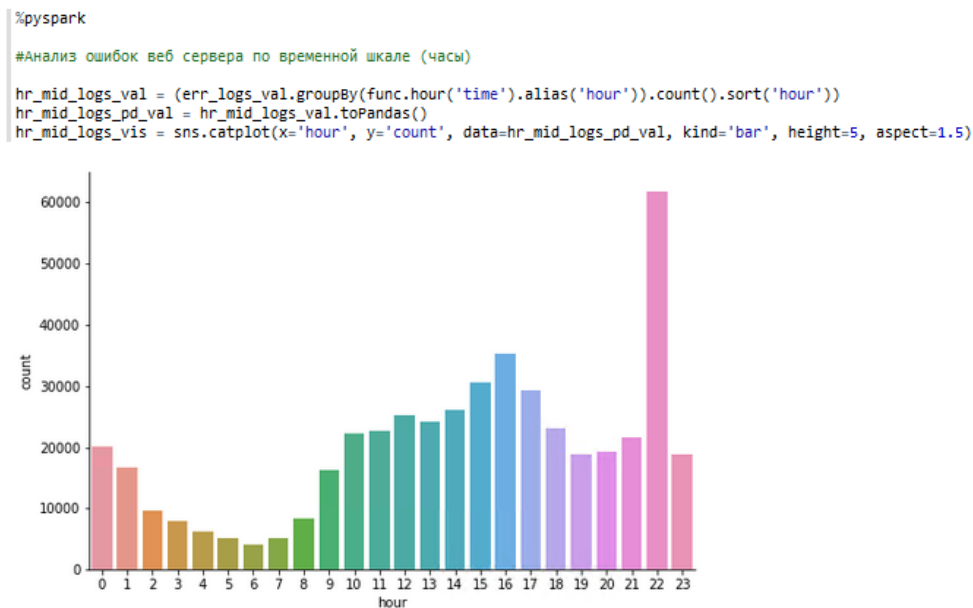


Figure 4.50 - Error analysis by code 404 for a specific hour.

When get this analysis from the 404 error code, one can determine how often errors occur for a certain period of time and make a forecast about the health and efficiency of servers and portals. Thus, it is possible to determine from the received forecast whether it is necessary to improve the performance of the site resources or the web server.

Analysis of getting information about the number of requests from clients ' IP addresses, to get statistics on traffic and visits to web resources in the context of days, as shown in Figure 4.51.

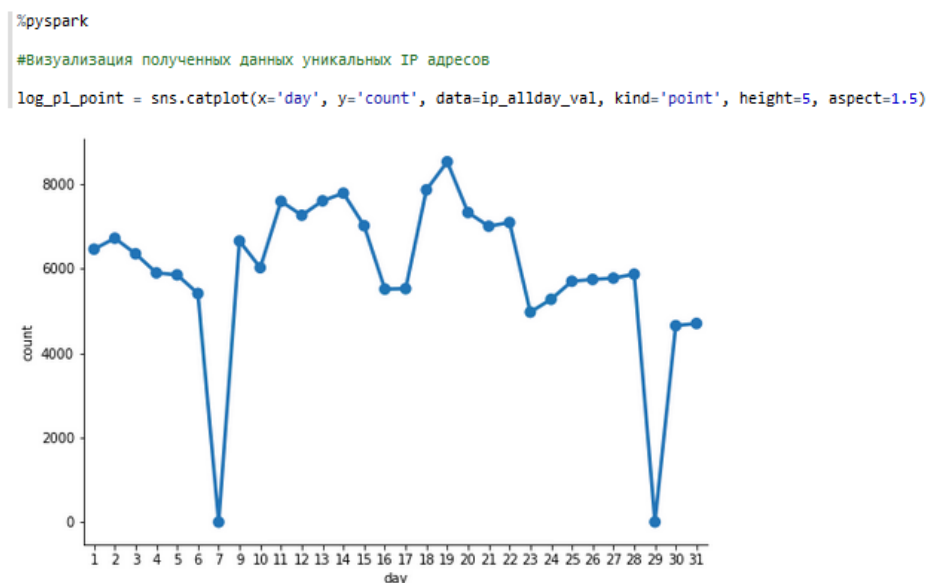


Figure 4.51 - Analysis of web server visits by day.

Figure 4.52 shows an analysis of the average load of the web server, determining the amount of downloaded content from the web server.

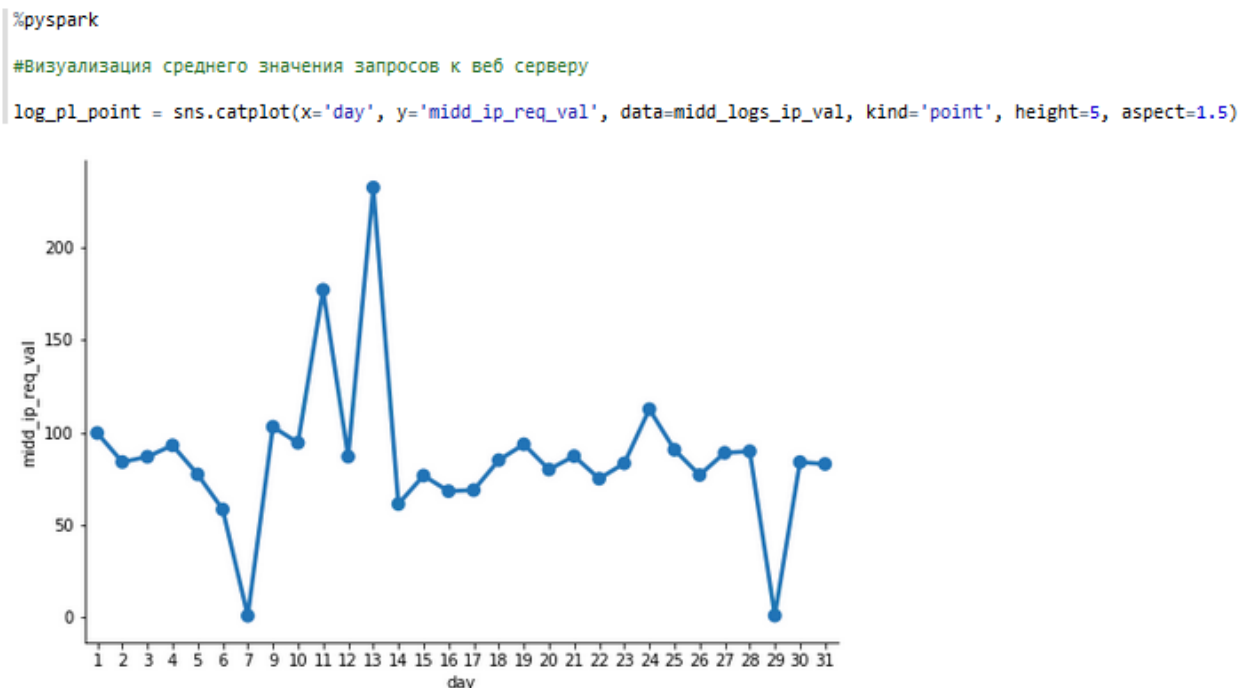


Figure 4.52 - Analysis of downloaded data from the web server.

During the analysis, the data was added to a new data set. Data processing was carried out using various libraries, such as: Numpy, Pandas, Matplotlib, Seaborn. The Apache Spark tool was used for distributed processing. The advantages of using the second type of processing are the ability to use mathematical functions to process data. Visualization tools allow to get a more detailed analysis when processing data, where can process data using various methods and libraries. For example, the ability to process data using the Numpy and Pandas libraries.

When processing data, three datasets were loaded to analyze the access logs of the NGINX web server, below is a graph of the amount of time spent on processing data sets, as shown in Figure 4.53.

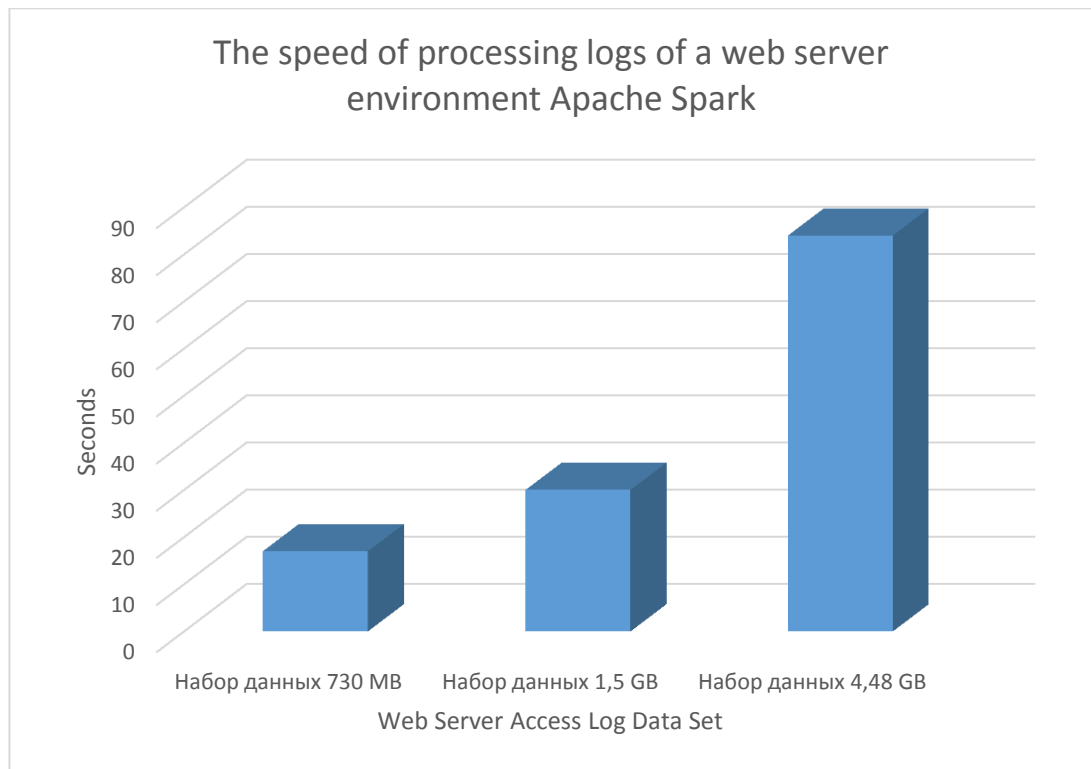


Figure 4.53 - Processing speed of data sets in the Apache Spark environment.

Figure 4.54 shows a graph of the number of processed web server access log data, the number of data specified in the lines.

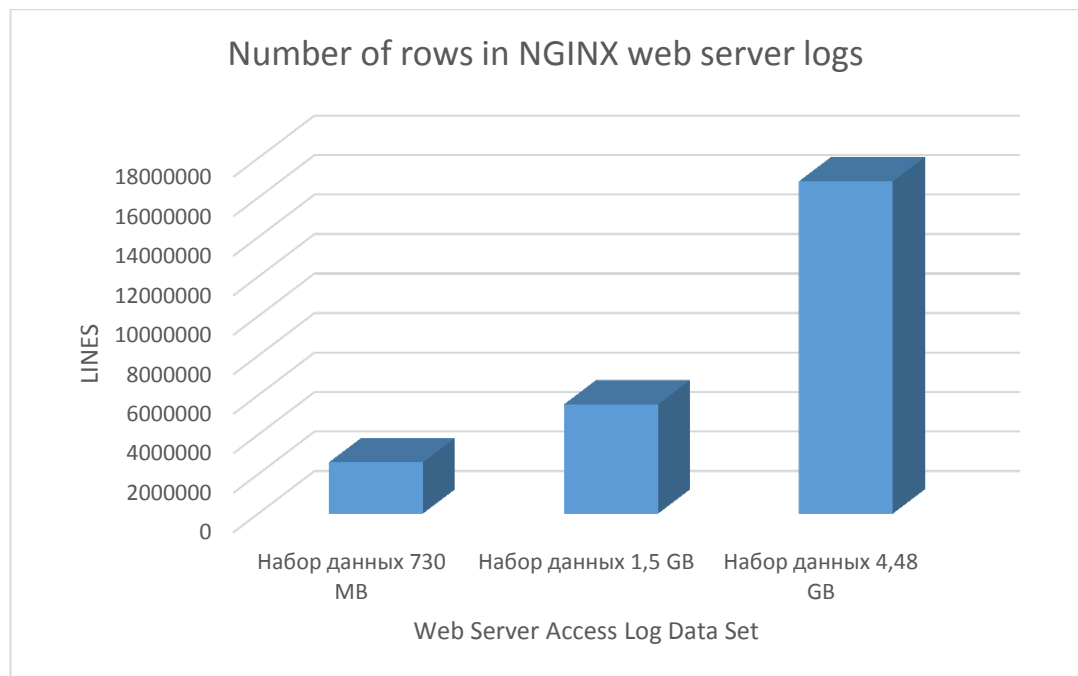


Figure 4.54 - Number of rows of web server access logs.

The above result shows that the amount of time spent processing the minimum network date values is minimal and is performed in less than a minute, while the analysis of the last network date data with 16 million rows was completed in 2 minutes. These indicators indicate that data processing in the

Apache Spark distributed data processing system can handle large amounts of data at a high speed, using server memory at all stages of data processing, and only after getting the result, the data is saved on the server disk.

After the experiment, a result was obtained to visualize the data of the processed logs of the NGINX web server in the HDP environment. Distributed processing greatly increases the speed of data processing, while other applications written in python, java, or other programming languages without support for distributed data processing run much slower, and they don't allow to process such huge amounts of data. Webserver statistics are presented on the schedules received, basic data and web server analysis are considered in detail.

Also data on processing speed of network dates in Apache Spark were presented. The received data of the access logs of the web-server allows to get detailed analysis on the work of the web-server, and allows specialists of different levels to make a decision on the need to improve the functioning of the sites. Understand what vulnerabilities exist on the web server, whether there is a need to increase security on the web server.

5 DATA ADMINISTRATION AND PROCESSING IN A HADOOP ENVIRONMENT

5.1 Big Data analysis, processing and visualization

Data research analysis refers to the critical process of conducting initial data studies to detect hypotheses, patterns, anomalies and assumptions through summary statistics and graphical representations of data.

In data mining, data research analysis is an approach to analyzing datasets to summarize their main characteristics, often using visual tools. Therefore, research analysis of data is used for the modelling task to find out what information these data are carrying. From a visual view of a column of figures or an entire spreadsheet, it is unlikely that important data characteristics can be identified. Research methods for data analysis have been developed as a tool to assist in this situation. Research analysis of data is usually classified in two ways. First, each method is either non-chartic or graphic. Second, each method is either one-dimensional or multidimensional.

The process of collecting, storing and processing medical data is a major challenge in addressing current health challenges. Use of modern information technology tools to effectively use digital health data to support decision-making in this area. One such tool is the Python programming language. This is facilitated by the simplicity of the language, as well as the wide variety of open libraries that are available. This work implements examples of research and data classification using some Python libraries.

For research, one will need to select the data set of interest (dataset). Medical data provided by the Kazakhstan Society for the Study of Diabetes within the framework of the target program "Burden of Diabetes 2019-2020 for the Republic of Kazakhstan" were used as data sets. The data set includes all

information about dispensary patients of all citizens of the Republic of Kazakhstan registered with the diagnosis of "Diabetes mellitus" from 31.12.2009 to 31.12.2019. To solve these problems, the Anaconda software solution is used, which includes all the necessary tools for solving problems of data analysis and processing. The Anaconda distribution consists of a Python interpreter and contains many different packages for data processing, including popular libraries such as: Numpy, Pandas, scipy, Matplotlib. This tool can be downloaded from the official website for installation on the OS. After installing this product, one can enable Anaconda navigator, and then choose which product want to work with. Anaconda Navigator is a graphical desktop user interface included in the Anaconda distribution. It also allows to run applications and easily manage packages, without using the command line. Figure 5.1 shows the Anaconda navigator menu.

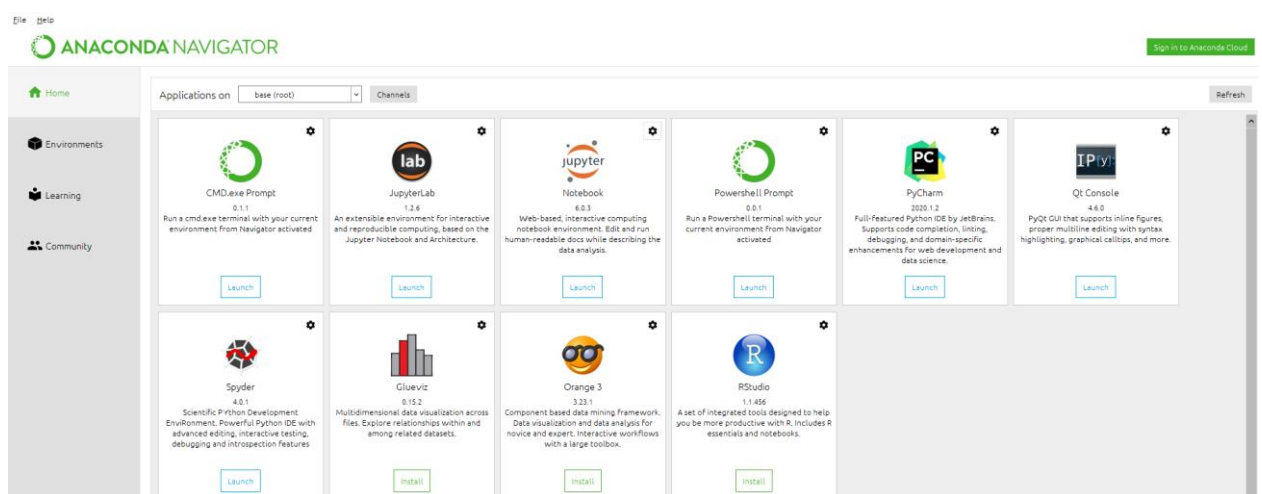


Figure 5.1 – The Menu navigator Anaconda

For data processing, the Jupiter notebook interactive data processing tool, which is included in the anaconda package, was used. After enabling Jupiter notebook, this tool will open in the browser. And then all the necessary manipulations can be carried out in the browser. After a successful launch, one can see the correct Jupiter environment configured.

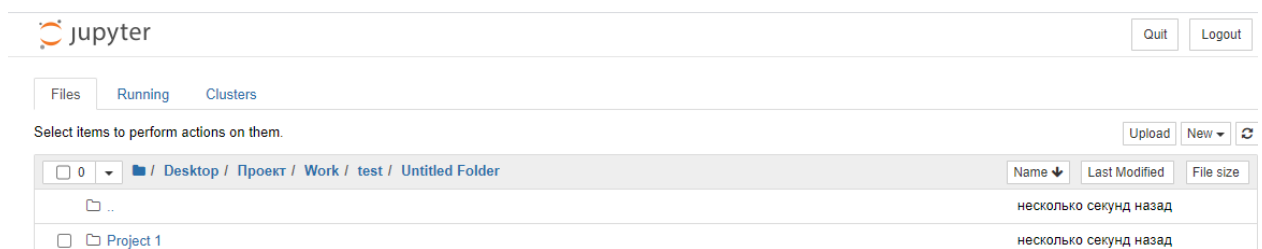


Figure 5.2-Jupyter Main Menu

The next step is to prepare the data set for loading. To do this is need to implement analysis on a subset of the data using the Pandas library. Pandas allows

to process, clean, and analyze data. In data research using Python, Pandas is also often used to work with tables. It is a library for working with data tables and is well suited for small and medium-sized datasets. Also, the most common solution is to use Pandas together with other solutions such as SQL, MongoDB, Elasticsearch. The Pandas library is one of the most popular Python tools for working with data, it supports various text, binary and sql file formats, including .xlsx, .xls and .csv. For work with Excel files with Pandas uses the xlrd and xlwt modules. To upload files to the dataset is need to import the necessary libraries for research data analysis.

```
In [1]: import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Figure 5.3 - Importing Libraries

Processing was performed on files in the format .xlsx, the files were presented in excel format. These files are supported by the pandas library, which is included in the Anaconda distribution package, and allows to process data.












 На 31.12.2009.xlsx	01.07.2020 15:35	Лист Microsoft Ex...	6 048 КБ
 На 31.12.2010.xlsx	02.07.2020 0:13	Лист Microsoft Ex...	7 238 КБ
 На 31.12.2011.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	8 453 КБ
 На 31.12.2012.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	9 418 КБ
 На 31.12.2013.xlsx	02.07.2020 0:54	Лист Microsoft Ex...	10 489 КБ
 На 31.12.2014.xlsx	02.07.2020 0:55	Лист Microsoft Ex...	11 515 КБ
 На 31.12.2015.xlsx	02.07.2020 0:55	Лист Microsoft Ex...	12 394 КБ
 На 31.12.2016.xlsx	02.07.2020 1:10	Лист Microsoft Ex...	13 692 КБ
 На 31.12.2017.xlsx	02.07.2020 0:56	Лист Microsoft Ex...	14 711 КБ
 На 31.12.2018.xlsx	02.07.2020 0:56	Лист Microsoft Ex...	15 865 КБ
 На 31.12.2019.xlsx	02.07.2020 0:57	Лист Microsoft Ex...	18 289 КБ

Figure 5.4 -List of files on dispensary patients of citizens of the Republic of Kazakhstan

To upload files to the dataset, one must register the download in the Jupiter notebook environment. Figure 5.5 shows the method of loading data into the DF dataset, which later allows to process the loaded data into the dataset.

```
In [1]: import pandas as pd
from os import listdir

filepaths = [f for f in listdir("./test/") if f.endswith('.xlsx')]
df = pd.concat(map(pd.read_excel, ["../test/На 31.12.2009.xlsx", "./test/На 31.12.2010.xlsx",
"./test/На 31.12.2011.xlsx", "./test/На 31.12.2012.xlsx", "./test/На 31.12.2013.xlsx",
"./test/На 31.12.2014.xlsx", "./test/На 31.12.2015.xlsx", "./test/На 31.12.2016.xlsx",
"./test/На 31.12.2017.xlsx", "./test/На 31.12.2018.xlsx", "./test/На 31.12.2019.xlsx"])))
```

Figure 5.5 -List of files uploaded to the dataset

The next step is to check how much data was uploaded to the dataset. Figure 5.6 shows the results of the number of rows and columns of all files where the total number of rows exceeds 2700000 records.

```
In [47]: df.shape
Out[47]: (2705693, 8)
```

Figure 5.6 -Results of counting all rows and columns

In Figure 5.7, one can get information on columns using the count function.

```
In [3]: df.count()
Out[3]: Регион          2705693
МО          2703494
Дата рождения  2705693
ПОЛ          2705693
национальность 2705693
МКБ          2705693
Дата взятия на учет 2705693
код МО        2705693
dtype: int64
```

Figure 5.7 - Counting rows in each column

Using the head function, one can see what data is loaded into the dataset. Figure 5.8 shows how to use the head function, it shows the first 10 rows of the dataset.

```
In [48]: df.head(10)
Out[48]:
```

	Регион	МО	Дата рождения	ПОЛ	национальность	МКБ	Дата взятия на учет	код МО
0	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1957-05-16 00:00:00.000	жен	Казахи	E11.7	1997-01-01 00:00:00.000	03MC
1	Акмолинская область	ГОСУДАРСТВЕННОЕ КОММУНАЛЬНОЕ КАЗЕННОЕ ПРЕДПРИЯ...	1930-08-10 00:00:00.000	муж	Украинцы	E11.8	2004-01-01 00:00:00.000	04PJ
2	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасының ж...	1948-01-05 00:00:00.000	муж	Казахи	E11.8	2003-01-01 00:00:00.000	0FA5
3	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасының ж...	1933-04-05 00:00:00.000	муж	Казахи	E11.5	2006-06-20 00:00:00.000	01UK
4	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1972-08-14 00:00:00.000	муж	Русские	E11.8	1992-01-01 00:00:00.000	014A
5	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1951-09-21 00:00:00.000	жен	Немцы	E11.9	1996-07-21 00:00:00.000	08TC
6	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1990-01-21 00:00:00.000	жен	Азербайджанцы	E10.4	2004-09-11 00:00:00.000	007L
7	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1938-05-09 00:00:00.000	муж	Казахи	E11.8	2008-05-06 00:00:00.000	007N
8	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1944-05-25 00:00:00.000	жен	Белорусы	E11.8	2008-01-01 00:00:00.000	007K
9	Акмолинская область	Ақмола облысы денсаулық сақтау басқармасы жаны...	1950-02-21 00:00:00.000	жен	Казахи	E11.7	2004-09-23 00:00:00.000	005B

Figure 5.8 – The first 10 rows of the dataset.

Visualization tools allow to get a more detailed analysis when processing data, where can process data using various methods and libraries. For example, the ability to process data using various libraries. In this study, the matplotlib library was used for data visualization, which allows visualizing incoming data after processing in Pandas. In Figure 5.9, one can see the results of processing the

calculation of the number of registered patients with a diagnosis of diabetes in the regions of the Republic of Kazakhstan. There is also a calculation of the number of patients in the official statistics for all regions of the country.

```
In [49]: import matplotlib.pyplot as plt
obl = df["Регион"].value_counts()
obl.plot(kind='barh', figsize = (12,8))
```

```
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x2bbdd93b5c8>
```

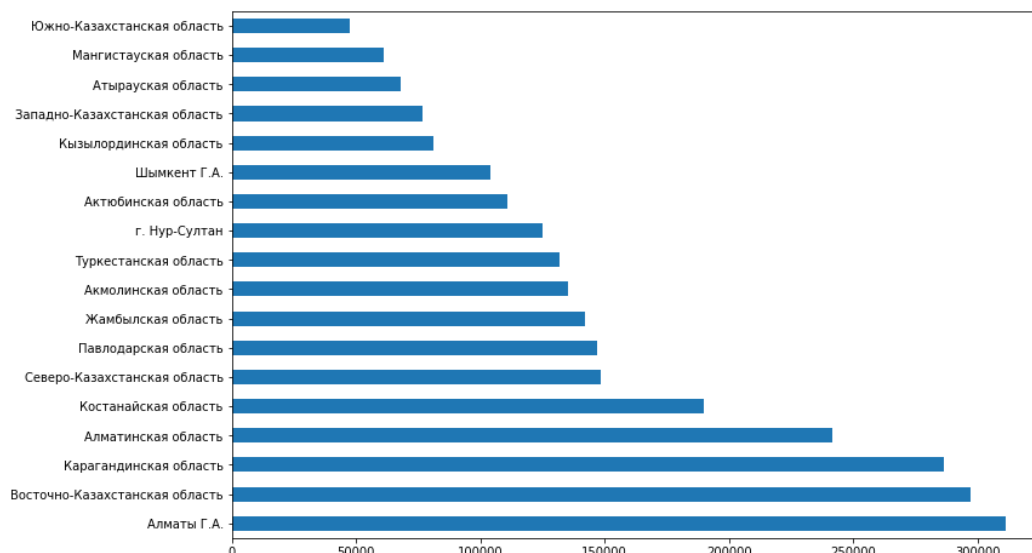


Figure 5.9 - Visualization of the number of patients by region.

Also, as a result of processing and visualization, numerical values of the number of patients diagnosed with diabetes in the regions of the Republic of Kazakhstan were obtained.

```
In [43]: obl
```

```
Out[43]: Алматы Г.А. 311212
Восточно-Казахстанская область 297488
Карагандинская область 286442
Алматинская область 241780
Костанайская область 189781
Северо-Казахстанская область 148549
Павлодарская область 146734
Жамбылская область 142321
Ақмолинская область 135296
Туркестанская область 131831
г. Нур-Султан 124775
Актыбинская область 110776
Шымкент Г.А. 104041
Кызылординская область 81184
Западно-Казахстанская область 76792
Атырауская область 68047
Мангистауская область 61018
Южно-Казахстанская область 47626
Name: Регион, dtype: int64
```

Figure 5.10 - Results of numerical values.

In the continuation of data processing, it is necessary to display a graph of patient growth in the Republic of Kazakhstan. As a result of processing figure 5.11 shows a plot of incidence rate from 2009 to 2019 in Kazakhstan.


```
In [39]: year = df["Ғод"].value_counts()
#year = year[:20]
year.plot()

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x250ba993548>
```

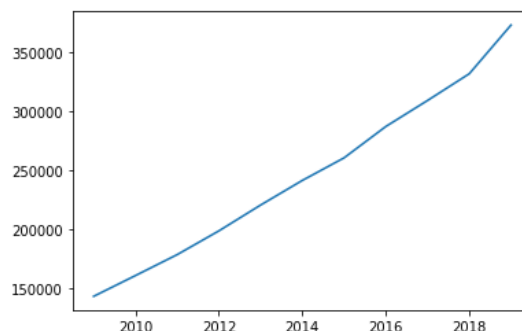


Figure 5.11 - Graph of morbidity growth in the Republic of Kazakhstan.

All files have been processed and carried out a count of all patients registered in Kazakhstan from 2009 to 2019.

```
In [40]: year
Out[40]: 2019    373183
         2018    331901
         2017    309248
         2016    287189
         2015    260627
         2014    241496
         2013    220621
         2012    198728
         2011    178594
         2010    160840
         2009    143266
         Name: Ғод, dtype: int64
```

Figure 5.12 - Numerical values of registered patients in the Republic of Kazakhstan.

As a result of processing figure 5.13 shows a plot of incidence rate from 2009 to 2019 for the city of Almaty.

```
In [44]: alm_year = akm["Ғод"].value_counts()
alm_year.plot()

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x250be4da648>
```

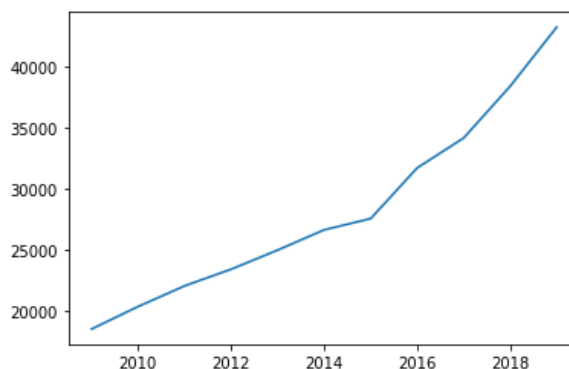


Figure 5.13 - Graph of morbidity growth in Almaty.

All files have been processed and carried out a count of all patients registered in Almaty from 2009 to 2019.

```
In [45]: alm_year
Out[45]: 2019    43221
         2018    38397
         2017    34166
         2016    31719
         2015    27576
         2014    26660
         2013    25004
         2012    23440
         2011    22081
         2010    20391
         2009    18557
         Name: год, dtype: int64
```

Figure 5.14-Numerical values of registered patients in Almaty.

For a more in-depth research analysis of data, using two files consisting of two parts for 2019, examine data by region, gender, age, etc. Next, one can upload the provided excel files to the Data Frame (data1 and data2).

```
In [2]: files = [f for f in os.listdir("./") if f.endswith(".xlsx")]
data1 = pd.read_excel(files[0], encoding = "cp1251", skiprows = range(0,7)).drop([0]).reset_index(drop=True)
data2 = pd.read_excel(files[1], encoding = "cp1251", skiprows = range(0,7)).drop([0]).reset_index(drop=True)
```

Figure 5.15-Loading data1 and data2

The last row of data1 matches the first row of data2. This allows us to combine these two separate Dataframes into a single dataframe (combined_data).

```
In [8]: combined_data = pd.concat([data1.drop([len(data1)-1]), data2]).reset_index(drop=True)
combined_data
```

Figure 5.16 -Merging two files

The result of combining two files is 1126 rows and 41 columns of data.

Out[8]:

	№ п/п	Регион, где пролежился больной	Медицинская организация, где пролежился больной	Диспансерный учет по данным ИС "ЭРДБ" с диагнозами E10-E14 (наличие в базе карты - 1, отсутствие - 0)	RpnID	Дата рождения	Дата смерти(РПН)	Возраст	Пол	Гражданство	...	Наименован операц
0	1	Акмолинская область	Государственное коммунальное предприятие на пр...	0.0	394122278	28.01.1954	29.07.2019	65	Мужской	Казахстан	...	Ni
1	2	Акмолинская область	Государственное коммунальное предприятие на пр...	1.0	394123525	21.04.1946	NaN	72	Женский	Казахстан	...	Ni
2	3	Акмолинская область	Государственное коммунальное предприятие на пр...	0.0	405395196	10.12.1938	NaN	80	Женский	Казахстан	...	Ni
3	4	Акмолинская область	Государственное коммунальное предприятие на пр...	1.0	405955008	09.03.1949	NaN	70	Женский	Казахстан	...	Ni
4	5	Акмолинская область	Государственное коммунальное предприятие на пр...	1.0	394123538	21.11.1951	NaN	67	Женский	Казахстан	...	Ni
...
112621	72487	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	394918861	16.09.1963	NaN	56	Женский	Казахстан	...	Ni
112622	72488	г.Нур-Султан	Государственное коммунальное предприятие на пр...	0.0	394381753	15.12.1984	NaN	35	Мужской	Казахстан	...	Ni
112623	72489	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	5900000000002820	13.04.1999	NaN	20	Женский	Казахстан	...	Ni
112624	72490	г.Нур-Султан	Государственное коммунальное предприятие на пр...	0.0	406596836	06.10.1982	NaN	37	Мужской	Казахстан	...	Ni
112625	72491	г.Нур-Султан	Государственное коммунальное предприятие на пр...	1.0	394788134	09.08.1960	NaN	59	Мужской	Казахстан	...	Ni

112626 rows x 41 columns

Figure 5.17 - Merged dataframe

The information stored in the downloaded and merged files allows to examine the data where patients were treated in the Republic of Kazakhstan. With the help of visualization, one can observe the results of processing in Figure 5.18.

```
In [17]: order=combined_data["Регион, где пролежился больной"].value_counts().sort_values(ascending=False).index

sns.set_context("poster")
sns.set_style("whitegrid")
fig = plt.figure(figsize=(20,15));
sns.countplot(y='Регион, где пролежился больной',
              order=order,
              hue='Пол',
              data=combined_data)

plt.xlim([0,9000])
plt.title("По половым признакам");
```

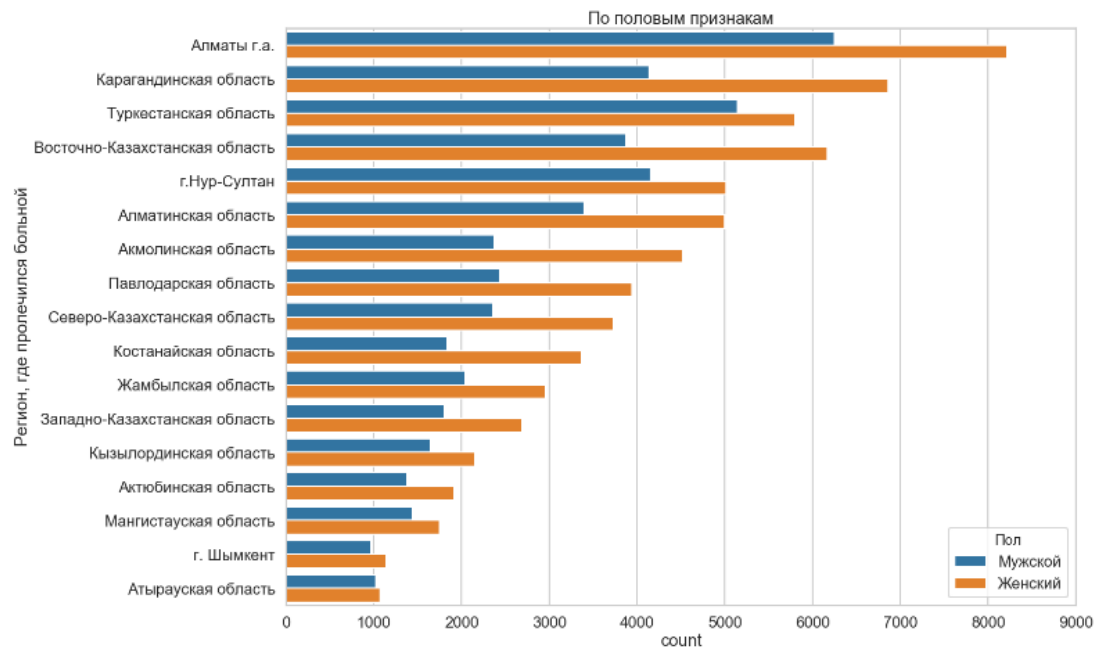


Figure 5.18 - Statistics of regions where patients were treated

As a result of processing, one can see where the patients were treated, and they were also divided by gender. It is also possible to determine the age of patients undergoing treatment.

```
In [18]: sns.set_context("poster")
sns.set_style("whitegrid")
plt.figure(figsize=(20,15))
diag3 = sns.countplot(x = "Возраст",
                      order = range(0,100,5),
                      hue = "Пол", data=combined_data)
plt.ylim([0,2500])
plt.ylabel("Количество");
```

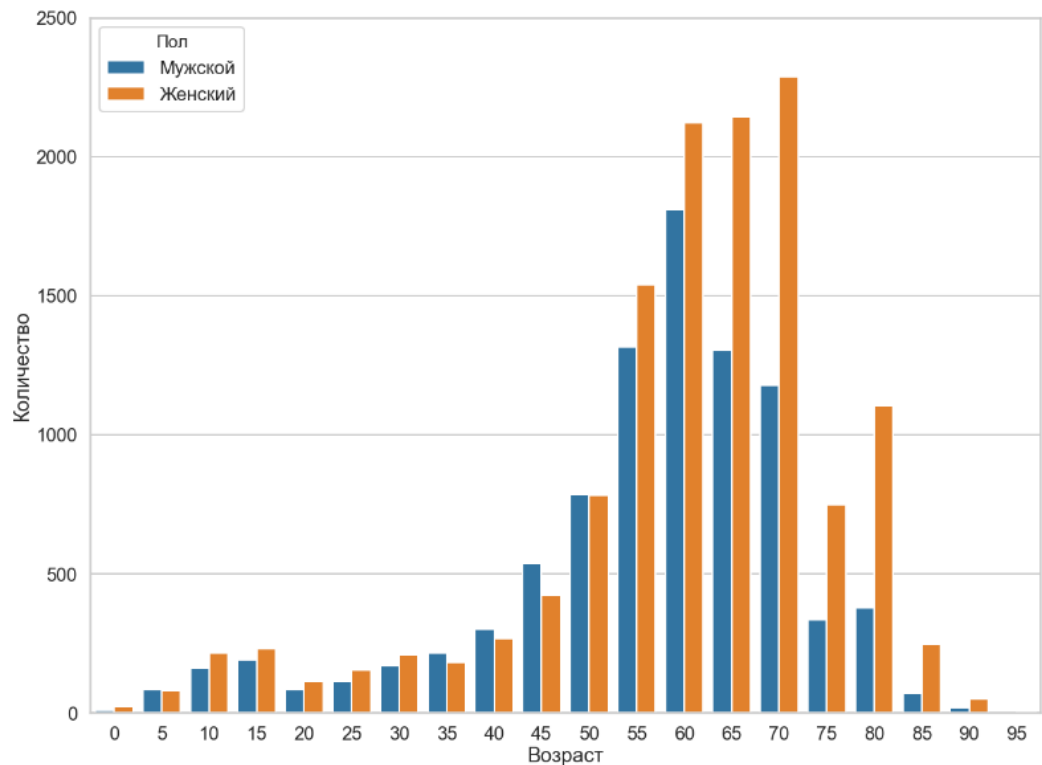


Figure 5.19 - Age of patients

Consequently, Almaty, Karaganda, Turkestan, East Kazakhstan regions, and Nur-Sultan are the leaders in the number of patients treated among the regions. It is obvious that the majority of patients are women than men, and the largest number of female patients are aged 70 years, while for men this value is less than 60. It can also be noted that the increase in the probability of the disease occurs after 40 years for both men and women. Further, during the study of patient data on citizenship, where the processing results show that the majority of patients are citizens of the Republic of Kazakhstan.

Out[23]:

Гражданство		Гражданство	
КАЗАХСТАН	111685	ИНДИЯ	3
РОССИЙСКАЯ ФЕДЕРАЦИЯ	315	СОЕДИНЁННЫЕ ШТАТЫ	3
НЕ УКАЗАНО	160	МОНГОЛИЯ	2
УЗБЕКИСТАН	128	ИЗРАИЛЬ	2
АЗЕРБАЙДЖАН	47	АМЕРИКАНСКОЕ САМОА	2
КЫРГИЗСТАН	32	ЛИТВА	1
АРМЕНИЯ	19	МАКЕДОНИЯ, БЫВШАЯ ЮГОСЛАВСКАЯ РЕСПУБЛИКА	1
КИТАЙ	19	ПАКИСТАН	1
УКРАИНА	17	ИОРДАНИЯ	1
ЛИЦО БЕЗ ГРАЖДАНСТВА	15	КАНАДА	1
Не указано	14	САУДОВСКАЯ АРАВИЯ	1
ТАДЖИКИСТАН	10	АФГАНИСТАН	1
ГРУЗИЯ	9	АВСТРАЛИЯ	1
ГЕРМАНИЯ	8	СОМАЛИ	1
ТУРЦИЯ	8	ОБЪЕДИНЁННЫЕ АРАБСКИЕ ЭМИРАТЫ	1
ТУРКМЕНИСТАН	6	МАЛАЙЗИЯ	1
МОЛДОВА, РЕСПУБЛИКА	5	ЕГИПЕТ	1
БЕЛАРУСЬ	3		

Figure 5.20 - Definition of citizenship

Consequently, 99% of all registered patients are citizens of the Republic of Kazakhstan. There is information about the method by which the patients were registered in the unified register of patients with diabetes mellitus.

```
In [24]: combined_data.loc[:, "Кем направлен"] = combined_data.loc[:, "Кем направлен"].str.upper()
```

```
sns.set_context("poster")
sns.set_style("ticks")

sns.catplot(y = "Кем направлен",
            hue = "Планово",
            kind = "count",
            data=combined_data,
            legend = False,
            height = 10,
            aspect = 2)

plt.xlabel("Количество")
plt.legend(title="Направлено", loc='best', labels=['Экстренно', 'Планово']);
plt.xlim([0,45000]);
```

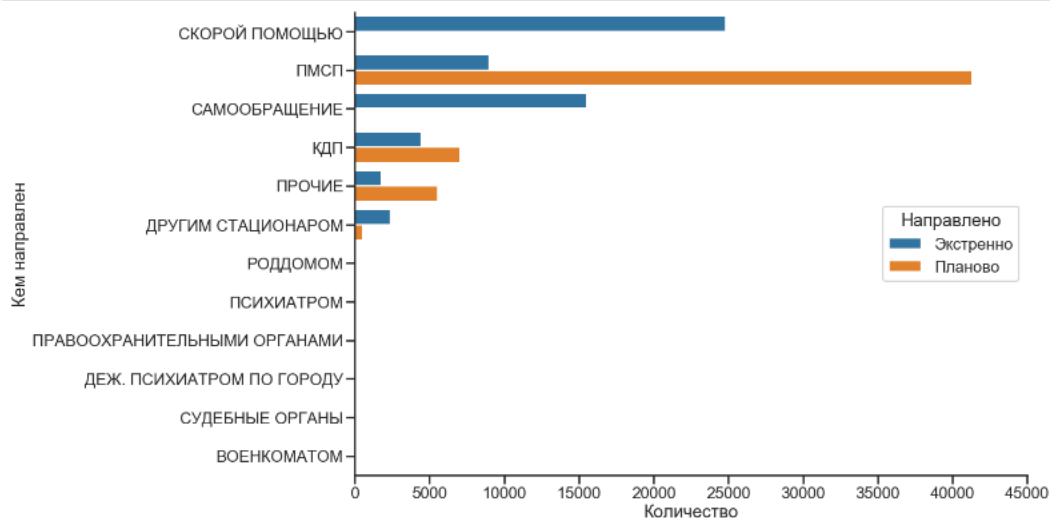


Figure 5.21 - Methods for referring patients for registration

According to Figure 5.21, the ambulance and the treatment itself are characterized by emergency delivery of the patient, while primary health care (PSMP) and consultative and diagnostic care (CDP) and others were mostly planned. Cases with a referral from a maternity hospital, a psychiatrist, law enforcement agencies, a city psychiatrist on duty, a judicial body and a military enlistment office are few in relation to the previous ones. According to this data, one can get a list of the 10 most common diagnoses. The results of the calculations are shown in Figure 5.22.

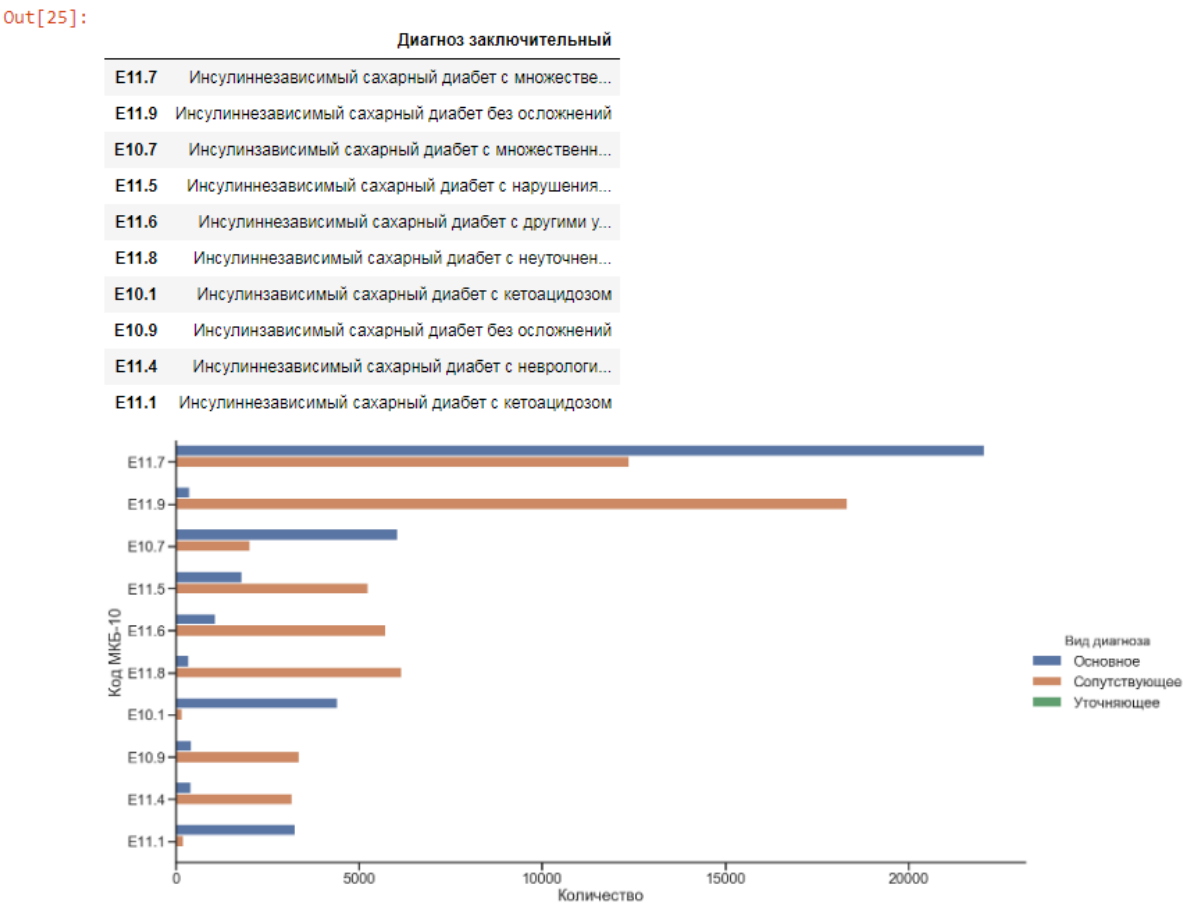


Figure 5.22 - List of common diagnoses

The two most common diagnoses are insulin-independent diabetes with multiple complications and no complications, respectively. In the first case, the diagnosis is often the main one, while for the second case, in the vast majority, it is concomitant. Other diagnoses and their types are also shown in the figure. The following example for calculations was used data on the number of bed days on average in Figure 5.23.

```
In [33]: plt.figure(figsize=(20,15))

sns.set(style = "ticks")
sns.set_context("poster")
sns.scatterplot(x="Возраст",
                y="Проведено койко-дней",
                hue="Исход лечения",
                hue_order=["Без перемен", "Улучшение", "Выздоровление", "Смерть", "Ухудшение"],
                data=combined_data)
plt.axhline(combined_data["Проведено койко-дней"].mean(), color = "k", label="Среднее значение койко-дней");
plt.legend();
```

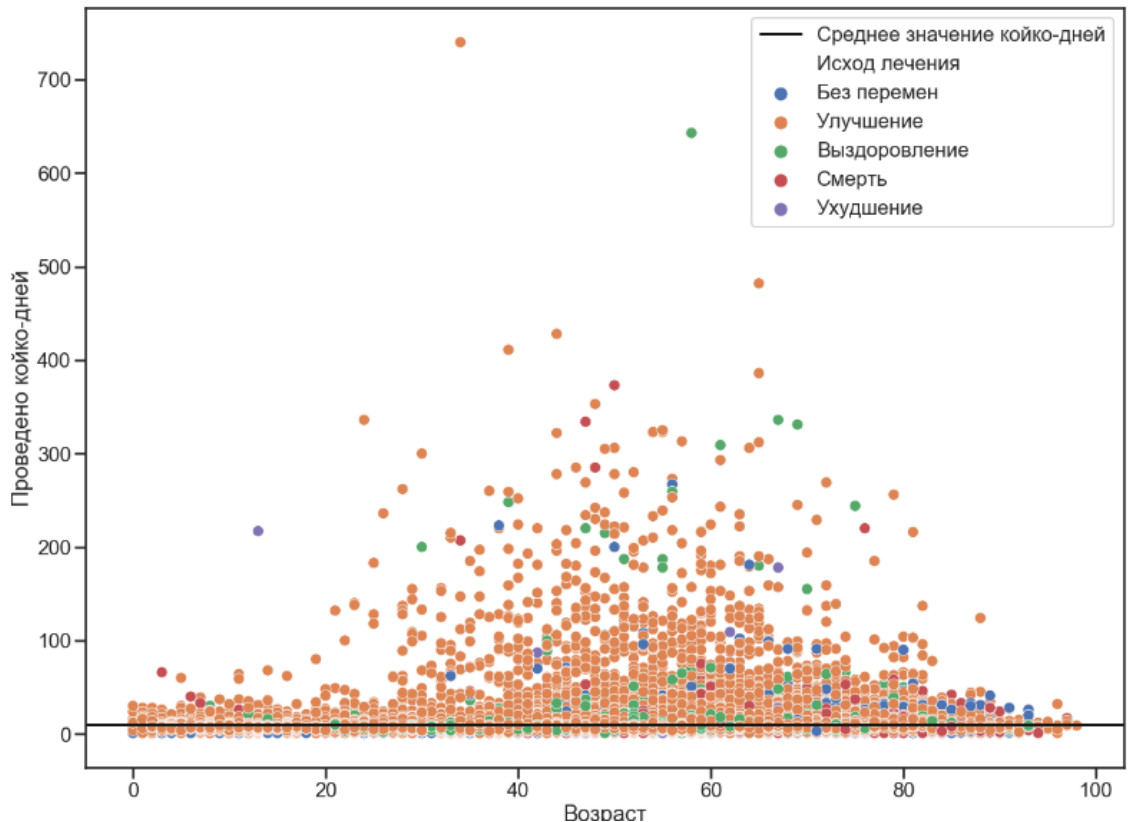


Figure 5.23 - Average value of bed days spent by patients

According to Figure 5.23, despite some values of more than 200 days, the average value of bed days is about 10 (marked with a black line). The chart shows that between the ages of 30 and 70, the probability of spending more than 50 bed days increases. The most common social status of patients is determined below.


```
In [34]: #fig = plt.figure(figsize=(20,15));
combined_data.loc[combined_data["Категории льготника"]=="-", "Категории льготника"] = "Не предоставлен о"

important_privileges = combined_data["Категории льготника"].value_counts().index[0:6]
important_status = combined_data["Социальный статус"].value_counts().index[0:10]
sns.set(style="whitegrid", context="poster")

sns.catplot(y="Социальный статус",
            order = important_status,
            hue = "Категории льготника",
            hue_order = important_privileges,
            kind = "count",
            data=combined_data,
            height = 10,
            aspect = 2);
sns.set(style = "white");
plt.xlabel("Количество");
```

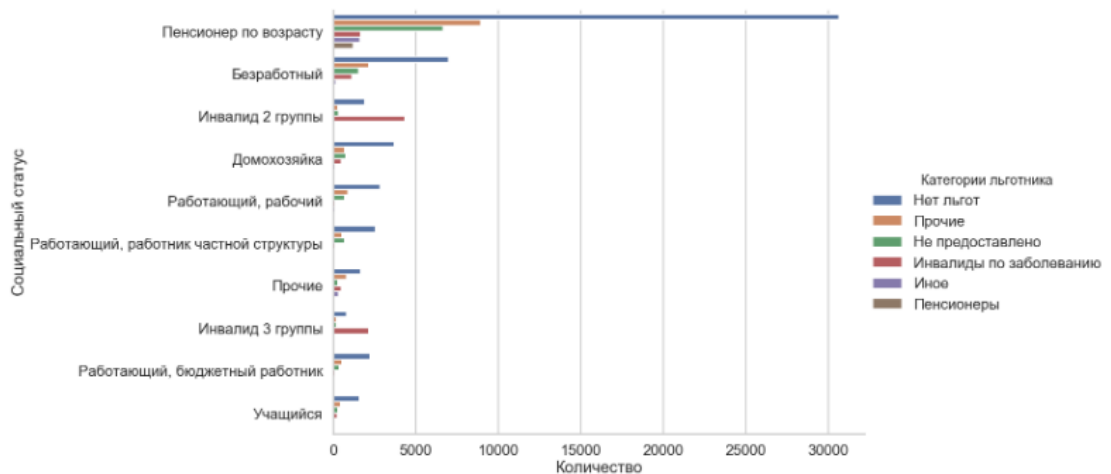
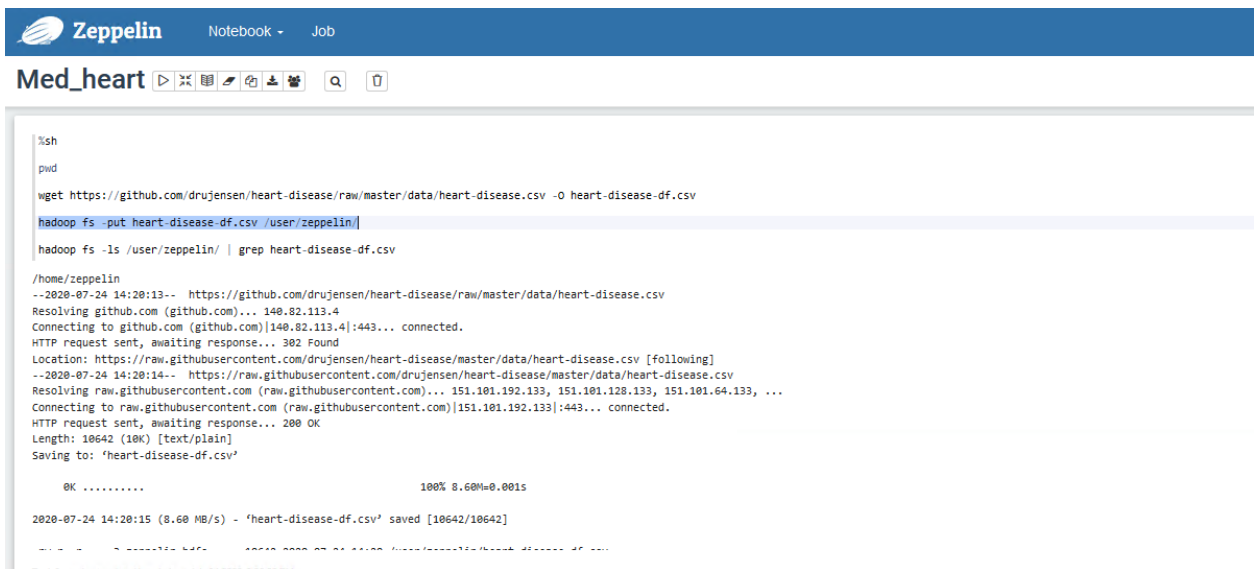


Figure 5.24 -Common social statuses of patients.

According to Figure 5.24, the 10 most common social statuses of patients, where the vast majority do not have any benefits. Of the pensioners, only about a quarter have one or another benefit. Among the disabled of the 2nd and 3rd groups, only half have disability benefits

5.2 Data analysis using Apache Spark.

To work with the data, the data sets of the Internet community "Kaggle" are used [47], which provides various data sets necessary for working with the data. For processing, a dataset on cardiovascular diseases was taken, where the main indicators of the analysis of patients were selected. The Apache Zeppelin graphical shell was used to interact with Spark, which provides data representation in a convenient visualization format. To do this, first of all, is need to upload files to HFS, and run the command `//hadoop fs -put heart-disease-df.csv /user/zeppelin/`, as shown in Figure 5.30.



```
%sh
pwd

wget https://github.com/drujensen/heart-disease/raw/master/data/heart-disease.csv -O heart-disease-df.csv
hadoop fs -put heart-disease-df.csv /user/zeppelin/

hadoop fs -ls /user/zeppelin/ | grep heart-disease-df.csv

/home/zeppelin
--2020-07-24 14:20:13-- https://github.com/drujensen/heart-disease/raw/master/data/heart-disease.csv
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/drujensen/heart-disease/master/data/heart-disease.csv [following]
--2020-07-24 14:20:14-- https://raw.githubusercontent.com/drujensen/heart-disease/master/data/heart-disease.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.192.133, 151.101.128.133, 151.101.64.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.192.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10642 (10K) [text/plain]
Saving to: 'heart-disease-df.csv'

OK ..... 100% 8.60M=0.001s

2020-07-24 14:20:15 (8.60 MB/s) - 'heart-disease-df.csv' saved [10642/10642]
```

Figure 5.25-Uploading a file to HDFS.

Next, the resulting file must be added to the dataset for further processing, as well as add libraries for data processing as shown in Figure 5.26.



```
%pyspark
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
from pyspark.sql.context import SQLContext
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from pyspark.sql import SparkSession

df = spark.read.csv('heart-disease-df.csv', header=True)

Took 1 sec. Last updated by admin at July 24 2020, 2:27:43 PM. (outdated)
```

Figure 5.26-Loading data and libraries.

The next step is to get the number of rows that were added to the dataset and view the contents, as shown in Figure 5.32.

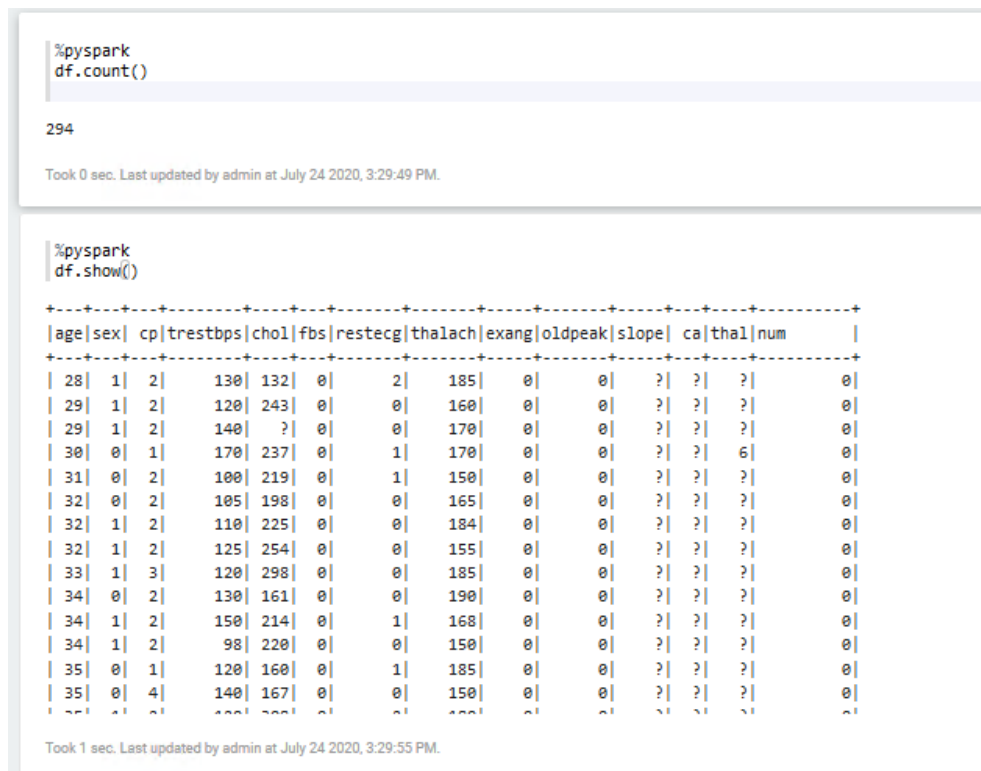


Figure 5.27 – Number of rows in the dataset.

The selected fields and their header names are shown in Figure 5.28.

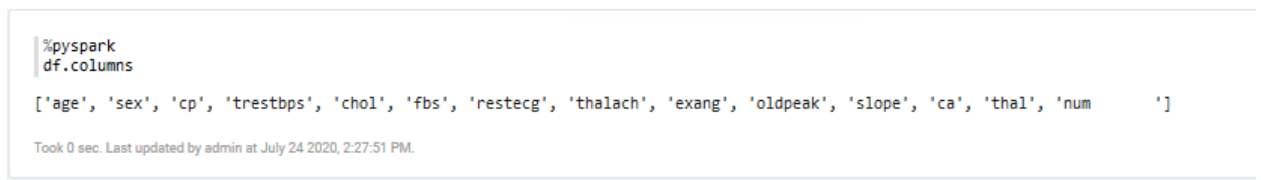


Figure 5.28 – The name of the fields

Next, it is necessary to determine which age group is most often susceptible to cardiovascular diseases. Figure 5.34 shows information about all the ages of the study participants and further these visualization results.

```
%pyspark
#Необходимо определить возрастную группу, которая максимально подвержена сердечно сосудистым заболеваниям
age_val = (df.groupBy('age').count().sort('age').cache())
age_val.count()
age_val_t = (age_val.toPandas().sort_values(by=['count'],ascending=False))
age_val_t
```

age	count
26	54
20	48
24	52
27	55
21	49
18	46
25	53
22	50
15	43
11	39
13	41
19	47
28	56
30	58
23	51
31	59
...	...

Took 4 sec. Last updated by admin at July 24 2020, 2:27:57 PM. (outdated)

Figure 5.29-Uploading a file to HDFS.

Figure 5.30 shows data visualization through the seaborn data visualization tool.

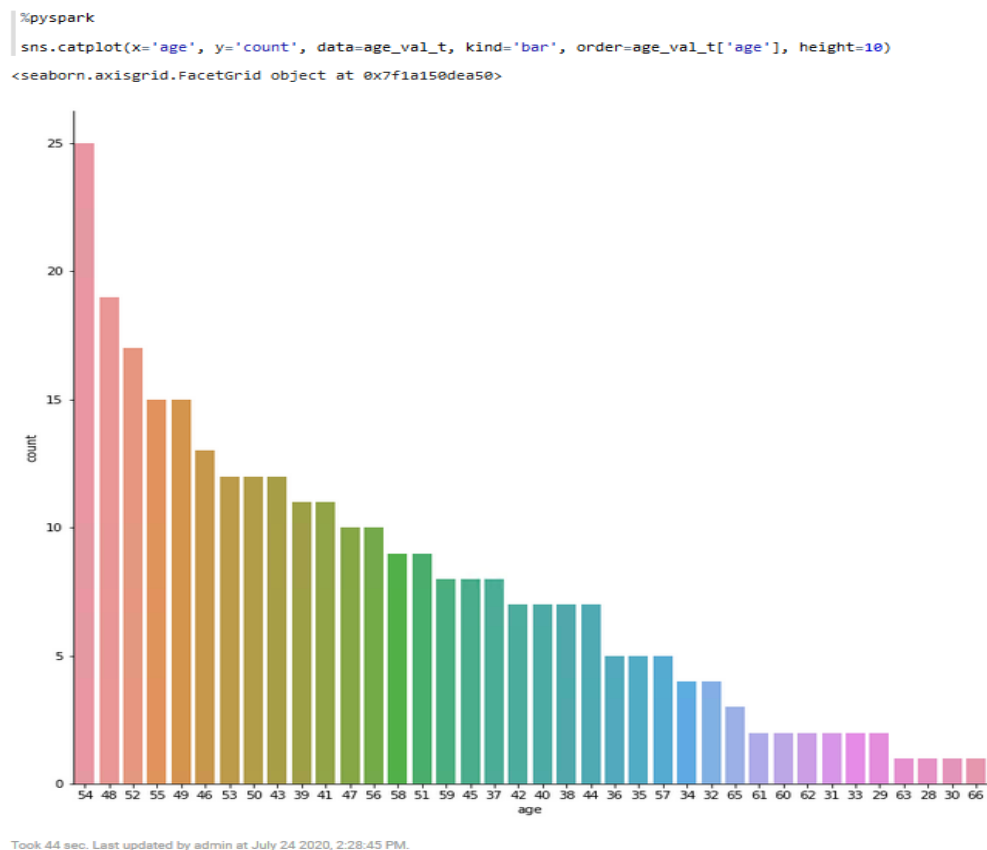


Figure 5.30-Visualization of the age group.

After the age group has been determined, it is necessary to obtain data on the maximum number of heart contractions in this group of patients. First of all is need to get data on patients whose age is 54 years, and then calculate how many patients have peak contractions.

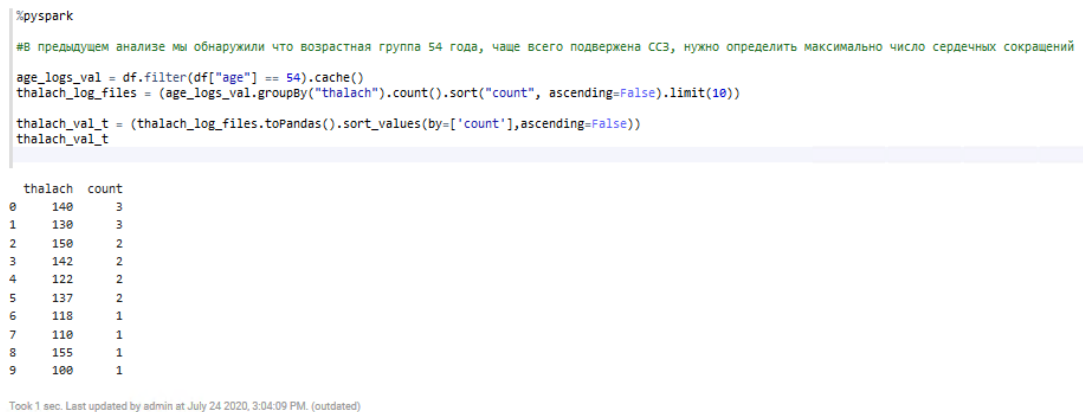


Figure 5.31 – Number of peak contractions.

The next step is to visualize this data, the image below shows the result of visualization.

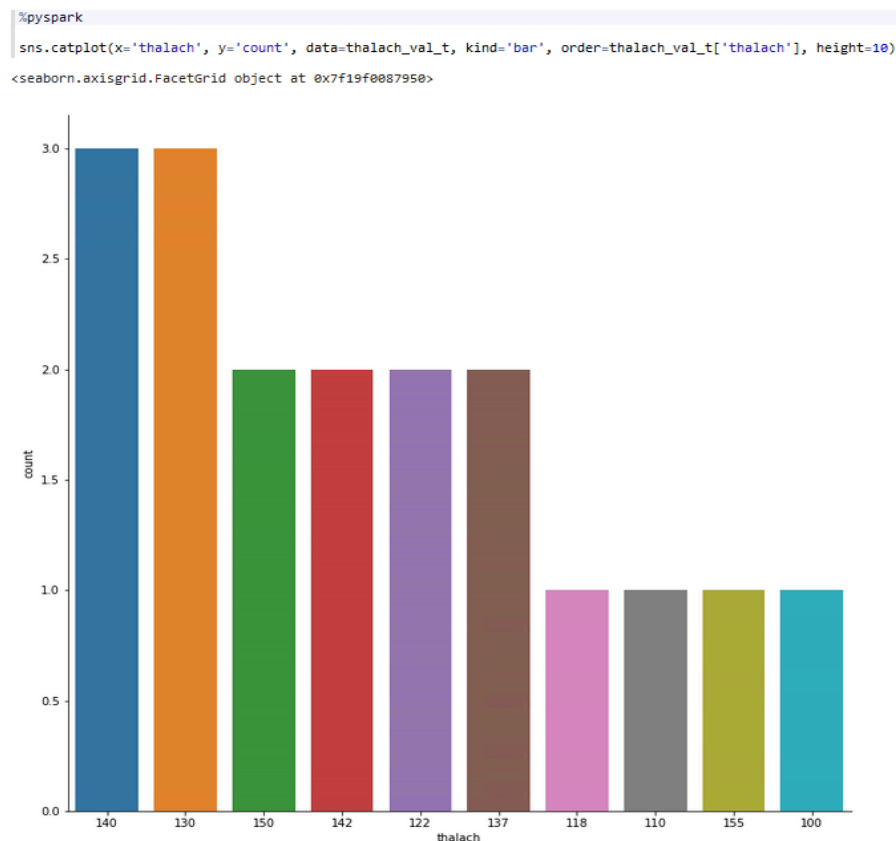


Figure 5.32-Uploading a file to HDFS.

Apache Spark has extensive data processing tools, and is used as an effective tool for parallel processing of large amounts of data in a cluster and to increase processing speed. In this experiment are used the datasets of the online community "Kaggle", about cardiovascular diseases, where the main indicators of the analysis

of patients were selected. The Apache Zeppelin graphical shell was used to interact with Spark, which provides data representation in a convenient visualization format.

5.3 Loading and processing data using the MapReduce model

In the early versions of Hadoop, the main components were MapReduce and the distributed file system Hadoop (HDFS), as well as Hadoop Common, a set of common utilities and libraries. The Java Hadoop framework will be used to implement the MapReduce distributed computing model. Hadoop is a framework that creates distributed computing in clusters to work with huge datasets. The main task of MapReduce is to use the map and reduce function to divide processing tasks into several tasks. These given tasks are performed on cluster nodes where the data are stored and then to combine that these tasks yield a consistent set of results. Hadoop is built on Java and is available through the Python programming language to write MapReduce code [48].

MapReduce typically partitions the input data set into independent blocks, which are processed in the «Map» step completely in parallel. The framework sorts the output data from this step, which then moves to the «Reduce» step, which is responsible for combining the data with the «Map» step. Usually both input and output data of the task are stored in the file system. The Framework is concerned with planning, monitoring and repeating failed tasks. Typically, the computational nodes and data storage nodes in a computational cluster are the same, that is, MapReduce and the distributed Hadoop file system (HDFS) run on the same node set. This configuration allows the platform to effectively plan tasks on nodes where data are already present, resulting in very high aggregate bandwidth across the cluster.

Hadoop MapReduce is a calculation that splits large tasks into separate tasks that can be performed in parallel on a cluster of servers. The results of the tasks can be combined to calculate the final results. As previously discussed, MapReduce breaks up the data set into independent parts that are processed in a separate node in a computing cluster. This is followed by the process of extracting data from the.xlsx file and splitting it into parts. There are about 60,000 records to be counted, counting by column, by region and by age.

Decomposition of the task. Create 3 classes necessary for the operation of our program:

1. TokenizerMapper - a class that extends the Map< KEYIN,VALUEIN,KEYOUT,VALUEOUT> superclass and maps input data of the key/value type to a set of intermediate key/value pairs;

2. IntSumReducer - a class that extends the superclass Reducer< KEYIN, VALUEIN, KEYOUT, VALUEOUT> and reduces the set of intermediate values that have a common key to a smaller set of values;

3. Main – the main class that configures the operation of the Hadoop framework.

Класс *Main*.

```

package Main;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import MapReduce.*;

public class Main {

    public static void main(String[] args) throws Exception{
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(Main.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

Figure 5.33-Main class.

Class *IntSumReducer*.

```

package MapReduce;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

```

Figure 5.34- IntSumReducer class.

Class *TokenizerMapper*.

```

package MapReduce;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}

```

Figure 5.35- TokenizerMapper Class

To run the program, one need to create two folders "age" and "region", which will store files with information from the "age" and "region" columns. Each folder has 20 files, each file has 3000 entries. In the first step, the data is from .The xlsx file must be split into separate files using a Python script.

```

import openpyxl

path = r"C:\Users\egor-\OneDrive\Рабочий стол\lab\данные\region"

workbook = openpyxl.load_workbook(path)
sheet = workbook.active

entries_counter = 0
for i in range(20):

    regions = ""
    age = ""

    for j in range(3000):
        if entries_counter >= len(sheet["B"]): break

        regions += sheet["B"][entries_counter].value + "\n"
        age += sheet["H"][entries_counter].value + "\n"

    with open("file_region_" + str(i) + ".txt", "w") as file:
        file.write(regions)

    with open("file_age_" + str(i) + ".txt", "w") as file:
        file.write(age)

```

Figure 5.36-Creating the "age" and "region" folders»

The next step is to export the missing Jar files for MapReduce as shown in Figure 5.37.

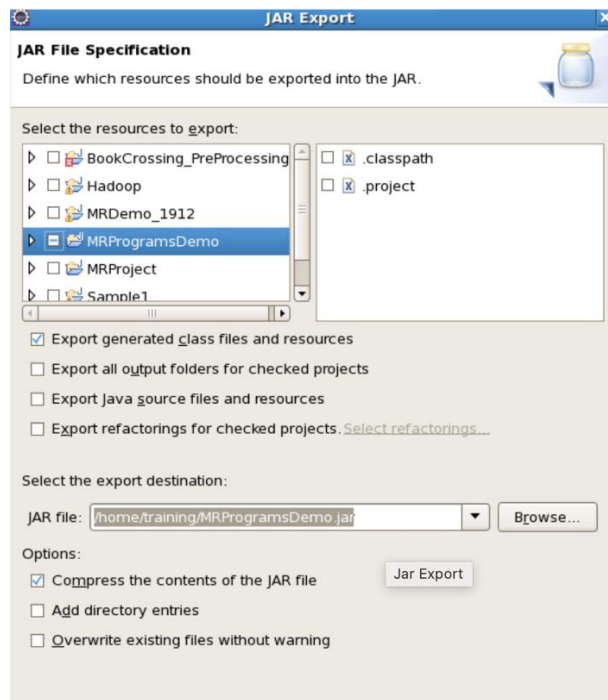


Figure 5.37-Exporting missing jar files

This is followed by compiling the specified project and putting it in a jar file. This can be done using maven builder (mvn package).

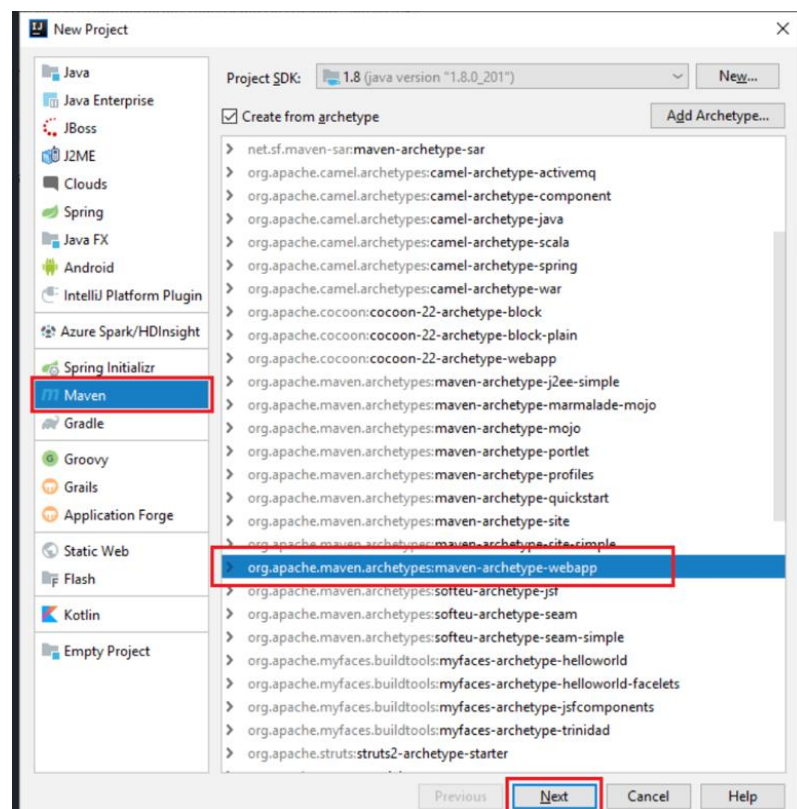


Figure 5.38 - Building the maven package

The next step is to prepare the packages (.jar) of the project.

```
PS D:\Java_Repo\MapReduceForDistinctValue> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building MapReduceForDistinctValue 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MapReduceForDistinctValue ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Java_Repo\MapReduceForDistinctValue\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MapReduceForDistinctValue ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 4 source files to D:\Java_Repo\MapReduceForDistinctValue\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ MapReduceForDistinctValue ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Java_Repo\MapReduceForDistinctValue\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ MapReduceForDistinctValue ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\Java_Repo\MapReduceForDistinctValue\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ MapReduceForDistinctValue ---
[INFO] Surefire report directory: D:\Java_Repo\MapReduceForDistinctValue\target\surefire-reports
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.bdp.mapreduce.distinct.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.045 sec
[INFO]
[INFO] Results :
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ MapReduceForDistinctValue ---
[INFO] Building jar: D:\Java_Repo\MapReduceForDistinctValue\target\MapReduceForDistinctValue-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
```

Figure 5.39 - Preparing the jar package

Then one need to resolve the dependencies in the file pom.xml using the cmd console.

```
pom
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-common</artifactId>
<version>2.7.1</version>
</dependency>
<!-- Hadoop Mapreduce Client Core -->
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-mapreduce-client-core</artifactId>
<version>2.7.1</version>
</dependency>
<dependency>
<groupId>jdk.tools</groupId>
<artifactId>jdk.tools</artifactId>
<version>${java.version}</version>
<scope>system</scope>
<systemPath>${JAVA_HOME}/lib/tools.jar</systemPath>
</dependency>
```

Figure 5.40-Adding dependencies to a file pom.xml

The next step is to add our input data to HDFS as shown in Figure 5.41.

```
[root@NN ~]# hadoop fs -copyFromLocal/home/NN/HadoopRepo/MapReduce/resources/distinctvalue/sampleddata.txt
/user/bdp/mapreduce/distinctvalue/input
```

Figure 5.41-Command to add data to HDFS

The next step is to launch the application, as shown in Figure 5. 42.

```
[root@NN ~]# hadoop jar /home/NN/HadoopRepo/MapReduce/MapReduceForDistinctValue-0.0.1-SNAPSHOT.jar com.bdp.mapreduce.distinct.driver.DistinctValueDriver /user/bdp/mapreduce/distinctvalue/input /user/bdp/mapreduce/distinctvalue/output
```

Figure 5.42 - Launching the application

The MapReduce application was successfully executed, according to Figures 5.43 and 5.44

```
INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8020
INFO input.FileInputFormat: Total input files to process: 1
INFO mapreduce.JobSubmitter: number of splits:1
INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1584546032725_0001
INFO impl.YarnClientImpl: Submitted application application_1584546032725_0001
INFO mapreduce.Job: The url to track the job: http://localhost:8020/job/job_1584546032725_0001/
INFO mapreduce.Job: Running job: job_1584546032725_0001
INFO mapreduce.Job: Job job_1584546032725_0001 running in cluster
INFO mapreduce.Job: map 0% reduce 0%
INFO mapreduce.Job: map 100% reduce 0%
```

Figure 5.43 -Executing the MapReduce command

```
Reduce shuffle bytes=3558
Reduce input records=314
Reduce output records=139
Spilled Records=628
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=126
CPU time spent (ms)=1220
Physical memory (bytes) snapshot=503894016
Virtual memory (bytes) snapshot=4236754944
Total committed heap usage (bytes)=298319872
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=1669
File Output Format Counters
Bytes Written=2823
```

Figure 5.44 - Successful execution of the program

The result is two files "file_name" and "file_region", as shown in Figure 5.45.

```
$ hadoop fs -ls /Woutputs

Found 2 items

-rw-r--r--    1 training supergroup          2020-07-28 03:36 /file_region.txt
-rw-r--r--    1 training supergroup          2020-07-28 03:53 /file_age.txt
```

Figure 5.45-Directory with output data

Figure 5.46 shows a list of files containing data files from the table.

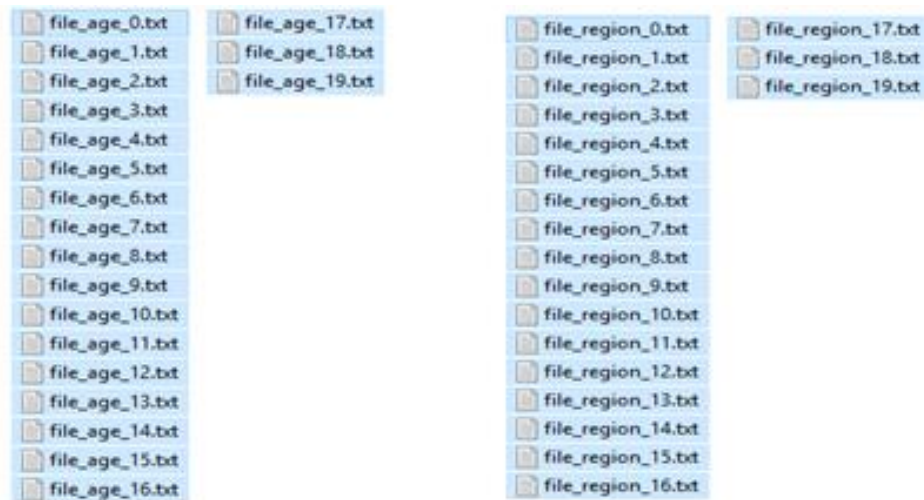


Figure 5.46-Output data files

Using regional data, one can calculate the age of patients and their number. Data was extracted from .xlsx file and divided into parts.

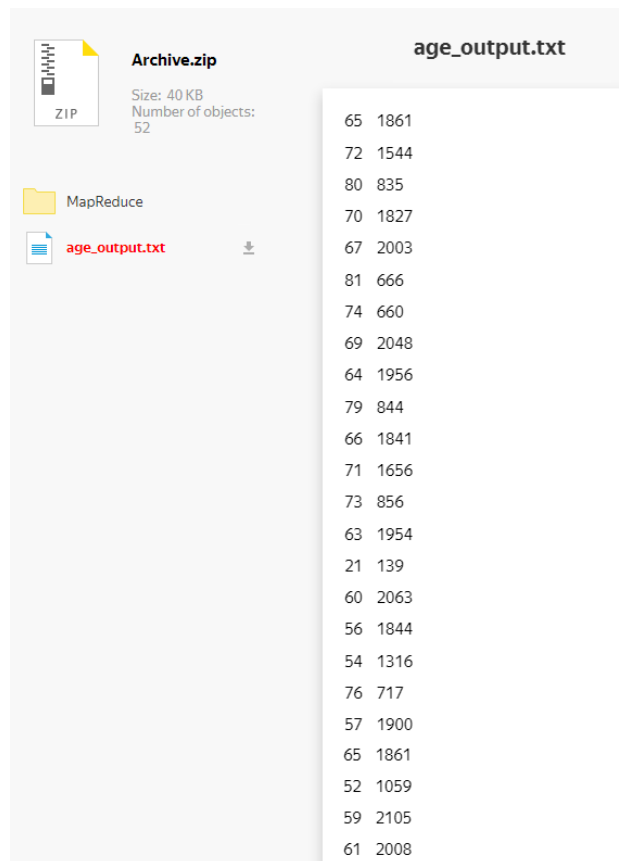


Figure 5.47-Contents of output files

As a result of processing, the contents of the output files allowed us to determine the number of patients by age. The following figure shows the results of counting the number in the "Region" column, which contains files with data from the table.

```
~$ hadoop fs -cat file_region.txt
```

1	Ақмолинская область	6920	
2	Ақмолинская область	6721	
3	Ақтүбинская область	3300	
4	Алматынская область	8400	
5	Атырауская область	2090	
6	Западно-Казахстанская область	4491	
7	Жамбылская область	5003	
8	Карагандинская область	11004	
9	Костанайская область	5203	
10	Кызылординская область	3798	
11	Мангистауская область	3197	
12	Павлодарская область	6383	
13	Северо-Казахстанская область	6088	
14	Восточно-Казахстанская область	10045	
15	Алматы г.а.	14466	
16	Туркестанская область	10958	
17	г. Шымкент	2115	
18	г.Нур-Султан	9170	

Figure 5.48 – The Contents of the file file_region.txt

As a result of processing, the contents of the output files allowed us to determine the number of patients by region.

Conclusion

The modern world produces a wealth of unstructured and structured information that provides statistics and forecasts the risks that can be faced by different companies and enterprises. The BD assumes responsibilities such as compiling statistical reports, compiling data in the database and presenting these data in a user-friendly manner. The latest developments and technologies enable information technology professionals to apply them to more complex tasks. The impact of BD technologies is increasing every year. In the IS, the use of BD allows the calculation of risks in the design of information solutions. It is also possible to collect monitoring metrics from various logs of web servers as well as network equipment metrics, as in the future, it is possible to analyze process data to obtain detailed information for statistics on web operation servers and network devices.

This book installed the necessary components to create an IS for BD. A system for managing server equipment using a hypervisor has been deployed. Web server log processing and analysis components were deployed on the established system.

As a result of the experiment, logs from the web server were processed and the resulting visualization made it possible to identify the frequently visited portions of the portal, i.e., which files are most frequently downloaded by users when working with the portal. Errors were also identified from the web server where this information allows identifying problems with the web server and starting improvements on the platform.

In selecting the software solution for creating the IS with BD, an analysis of existing solutions was made and the HDP product was used as a platform. This platform provides a huge selection of applications to work with BD. The following steps have been taken to process log data:

- web server log loading on the HDP server has been configured;
- the incoming data are analyzed using Python and the Apache Spark distributed processing tool;
- Apache Zeppelin provides data in a convenient way for analysts and users.
- data processing by python tools and libraries, which include tools such as pandas, Numpy, matplotlib, seaborn.

In order to make the website a popular and popular resource, it is necessary to optimize web portals in order to reduce the number of errors, increase the efficiency of the sites of different spheres. Information technology professionals are encouraged to improve the performance of their web portals, as web portals, sites and businesses are an important asset in the organization's advertising and are the face of the company.

The results of the research and the comparative analysis obtained in this manual allow developers and system administrators to improve the efficiency of web portals by being able to analyze critical errors in web workservers, faulty or

unused site or portal resources, finding problem safety zones in web servers and sites.

This study on data processing and visualization will help web analysts to get a more complete picture of how users use the web portal, which pages are the most frequently used. These data make it possible to better understand and analyze the behavior of users working with sites.

Apache Spark analyzed data processing rates depending on the amount of data in the network date. The results show that there is a delay in processing the data, but a small one, and processing speed in Apache Spark is high due to processing data in the server memory.

References

1. Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services. Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana. ISBN: 978-1-118-87613-8.
2. Kankanhalli, A., Hahn, J., Tan, S., & Gao, G. (2016). Big data and analytics in healthcare: Introduction to the special section. *Information Systems Frontiers*. <https://doi.org/10.1007/s10796-016-9641-2>.
3. [Monteith, S., Glenn, T., Geddes, J., & Bauer, M. (2015). Big data are coming to psychiatry: a general introduction. *International Journal of Bipolar Disorders*. <https://doi.org/10.1186/s40345-015-0038-9>.
4. Mashooque, A., Safeeullah, S., Awais, K., & Muneer, A. (2017). Big Data Analytics and Its Applications. *Annals of Emerging Technologies in Computing (AETiC)*. <https://arxiv.org/abs/1710.04135>.
5. Big Data:using smart Big Data, analytics and metrics to make better decisions and improve performance / Bernard Marr. © 2015 John Wiley & Sons Ltd. ISBN 978-1-118-96583-2.
6. Big Data in practice. How 45 successful companies used big data analytics to deliver extraordinary results. Bernard Marr. © 2016 Bernard Marr. ISBN 978-1-119-23138-7.
7. Ian, M., Safeeullah, S., Mark, L., Mark, W., & Andy, F. (2013). The White Book of... Big Data: The Definitive Guide to Big Data. <https://www.fujitsu.com/th/en/Images/WhiteBookofBigData.pdf>.
8. Lazer, D., & Radford, J. (2017). Data ex Machina: Introduction to Big Data. *Annual Review of Sociology*. <https://doi.org/10.1146/annurev-soc-060116-053457>.
9. Rein, R., & Memmert, D. (2016). Big Data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus*. <https://doi.org/10.1186/s40064-016-3108-2>.
10. Fister, Dušan & Fister jr, Iztok & Fister, Iztok.(2016). Visualization of cycling training. *StuCoSReC. Proceedings of the 2016 3rd Student Computer Science Research Conference, At Koper*.
11. Yassine, A., Singh, S., & Alamri, A. (2017). Mining Human Activity Patterns from Smart Home Big Data for Health Care Applications. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2017.2719921>.
12. Li, R. Y. M., Li, H. C. Y., Mak, C. K., & Tang, T. B. (2016). Sustainable Smart Home and Home Automation: Big Data Analytics Approach. In *International Journal of Smart Home*. <https://doi.org/10.14257/ijsh.2016.10.8.18>
13. Big Data Demystified. David Stephenson. © Pearson Education Limited 2018 (print and electronic). ISBN: 978-1-292-21810-6 (print).
14. Introducing Data Science: Big Data, Machine Learning, and more, using Python tools. Davy Cielen, Arno Meysman, Mohamed Ali. Manning Publications; ISBN-13: 978-1633430037.

15. Ben Ayed, A., Ben Halima, M., & Alimi, A. M. (2015). MapReduce based text detection in big data natural scene videos. In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2015.07.297>.
16. ALQWBANI, ZHANG ZUPING, FARES AQLAN, Abdullah. (2014). Big Data Management for MMO Games and Integrated Website Implementation. In *Global Journal of Computer Science and Technology*. <https://computerresearch.org/index.php/computer/article/view/64>.
17. Kuchipudi S., Tatireddy S., Applications of Big Data in Various Fields. Kuchipudi Sravanthi et al, (IJCSIT) *International Journal of Computer Science and Information Technologies*, – 2015, – Vol. 6 (5), – P. 4629-4632.
18. Uthayasankar S., Muhammad M., Zahir I., Vishanth W., Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*. – 2017, Vol. 70, – P. 263-286.
19. Mashooque A., Soomro S., Juman K., Kartio M., Big Data Analytics and Its Applications. *Annals of Emerging Technologies in Computing (AETiC)*, – 2017, – Vol. 1, No. 1, – P. 45-54.
20. Learn Data Science. 4 types of Data Analytics. <https://www.datascience.com/blog/4-types-of-data-analytics>.
21. Five key types of big data analytics that every business analyst should know. <https://www.captchu.edu/blog/five-types-of-big-data-business-analytics>.
22. Diagnostic analytics. <https://www.cornerstoneondemand.com/glossary/diagnostic-analytics>.
23. 5 types of analytics: prescriptive, predictive, diagnostic, descriptive, and cognitive analytics. <https://www.weirdgeek.com/2018/11/types-of-analytics/>.
24. Types of big data analytics. <https://www.rishabhsoft.com/blog/what-is-big-data-analytics-and-types>.
25. Applications for working with big data in real time in various fields. <https://www.edureka.co/blog/big-data-applications-revolutionizing-various-domains/>.
26. Bende, S., & Shedge, R. (2016). Dealing with Small Files Problem in Hadoop Distributed File System. In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2016.03.127>.
27. Saraladevi, B., Pazhaniraja, N., Paul, P. V., Basha, M. S. S., & Dhavachelvan, P. (2015). Big data and Hadoop-A study in security perspective. In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2015.04.091>.
28. Beakta, Rahul. (2015). Big Data And Hadoop: A Review Paper. <https://www.semanticscholar.org/paper/Big-Data-And-Hadoop%3A-A-Review-Paper-Beakta/2dfcdbc2cb115d6064a2570c602369677a56ffb4#extracted>.
29. Gupta, P., Kumar, P., & Gopal, G. (2015). Sentiment Analysis on Hadoop with Hadoop Streaming. In *International Journal of Computer Applications*. [semanticscholar.org/paper/Sentiment-Analysis-on-Hadoop-with-Hadoop-Streaming-Gupta-Kumar/89065bd8faf627117e42c17dc87fa1b64bb7f606](https://www.semanticscholar.org/paper/Sentiment-Analysis-on-Hadoop-with-Hadoop-Streaming-Gupta-Kumar/89065bd8faf627117e42c17dc87fa1b64bb7f606).

30. Merla, P. & Liang, Y.. (2017). Data analysis using hadoop MapReduce environment. In IEEE International Conference on Big Data (Big Data), Boston. doi: 10.1109/BigData.2017.8258541.
31. Yusuf Perwej, Bedine Kerim, Mohmed Sirelkhem Adrees, Osama E. Sheta. (2017). An Empirical Exploration of the Yarn in Big Data. In International Journal of Applied Information Systems. <http://www.ijais.org/archives/volume12/number9/1015-2017451730>.
32. Rallapalli, S., Gondkar, R. R., & Ketavarapu, U. P. K. (2016). Impact of Processing and Analyzing Healthcare Big Data on Cloud Computing Environment by Implementing Hadoop Cluster. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2016.05.171>.
33. Zhao, X., Zhang, S., & Ren, Z. (2015). Implementation based on hadoop ophthalmic imaging serialization file store. *Proceedings of Science*. <https://doi.org/10.22323/1.264.0030>.
34. Harb, H., Mroue, H., Mansour, A., Nasser, A., & Cruz, E. M. (2020). A hadoop-based platform for patient classification and disease diagnosis in healthcare applications. *Sensors (Switzerland)*. <https://doi.org/10.3390/s20071931>.
35. Luo, J., Wu, M., Gopukumar, D., & Zhao, Y. (2016). Big Data Application in Biomedical Research and Health Care: A Literature Review. *Biomedical Informatics Insights*. <https://doi.org/10.4137/bii.s31559>.
36. Lohar, N., Chavan, D., Arade, S., Jadhav, A.R., & Chikmurge, D. (2016). Content Based Image Retrieval System over Hadoop Using MapReduce. *International journal of scientific research in science, engineering and technology*. <https://www.semanticscholar.org/paper/Content-Based-Image-Retrieval-System-over-Hadoop-Lohar-Chavan/8733800d0f2eadca1df3583f971f4ed4e3127ee7>.
37. Diaconita, V., Bologa, A. R., & Bologa, R. (2018). Hadoop oriented smart cities architecture. *Sensors (Switzerland)*. <https://doi.org/10.3390/s18041181>
38. Agarwal, S., Yadav, L., & Mehta, S. (2017). Cricket Team Prediction with Hadoop: Statistical Modeling Approach. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2017.11.402>.
39. Zhou, T., Lee, X., & Chen, L. (2018). Temperature monitoring system based on Hadoop and VLC. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2018.04.325>.
40. National Library named after N. E. Bauman. Apache Hadoop. https://ru.bmstu.wiki/Apache_Hadoop.
41. Spark. Spark core programming. Tutorials Point (I) Pvt. Ltd. 2015.
42. Learning Spark: Lightning-Fast Big Data Analysis. 1st Edition by Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. ISBN-13: 978-1449358624.
43. Selina Sharmin, Asoke Datta, Md. Nurain Haider (2018). Big Data & Virtualization: Concept familiarization and relation between them. *International Journal of Engineering Development and Research (IJEDR)*.

44. Reddy, V., & Rajamani, L. (2014). Evaluation of Different Hypervisors Performance in the Private Cloud with SIGAR Framework. *International Journal of Advanced Computer Science and Applications*
45. Radchenko, G. I., Alaasam, A. B. A., & Tchernykh, A. N. (2019). Comparative analysis of virtualization methods in Big Data processing. *Supercomputing Frontiers and Innovations*. <https://doi.org/10.14529/jsfi190107>
46. Arie Taal. (2015). Resource usage in hypervisors. FNWI / Instituut voor Informatica. <https://scripties.uba.uva.nl/download?fid=626785>
47. Start with more than a blinking cursor. <https://www.kaggle.com/>. 03.06.2020.
48. White T., Hadoop: The Definitive Guide, O'Reilly & Associates Inc; 4rd Edition, 2015.

BIG DATA ANALYTICS

A.K. Mukasheva

T.F. Umarov

I.A. Zimin

This edition is signed in 01.07.2021

Form 60x84 1/16.

Volume 7 pp.

Circulation 500 pcs.

No. for order 0028238

LLC “Lantar Trade”

Tel: 87022510217

Email: lantar2018@mail.ru

Almaty city, Yegizbayeva 7B, office 704.