INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2018) Vol. 7 (2) 85−96
DOI: 10.22630/ISIM.2018.7.2.1

APPLYING E-LEARNING SYSTEMS FOR BIG DATA EDUCATION

GRZEGORZ ARKIT [a)], SILVA ROBAK [b)], ALEKSANDRA ARKIT [b)]

[a)]*Faculty of Computer Science, Electrical Engineering and Automatics*
University of Zielona Góra (UZ)

[b)]*Faculty of Mathematics, Computer Science and Econometrics*
University of Zielona Góra (UZ)

Processing massive data amounts and Big Data became nowadays one of the most significant problems in computer science. The difficulties with education on this field arise, the appropriate teaching methods and tools are needed. The processing of vast amounts of data arriving quickly requires the choice and arrangement of extended hardware platforms.

In the paper we will show an approach for teaching students in Big Data and also the choice and arrangement of an appropriate programming platform for Big Data laboratories. Usage of an e-learning platform Moodle, a dedicated platform for teaching, could allow the teaching staff and students an improved contact with by enhancing mutually communication possibilities. We will show the preparation of Hadoop platform tools and Big Data cluster based on Cloudera and Ambari. The both solutions together could enable to cope with the problems in education of students in the field of Big Data.

Keywords: Big Data, e-learning platform, Hadoop platform tools, cloud computing, Linux, virtualization

1. Introduction

One of present-day big challenges in information systems are the issues associated with coping with and utilization of the vast amounts of data and big data [23]. There are some definitions and terms concerning Big Data usage. Unfortunately, because this is a new subject, there is no one strict (official) definition of these terms.

Massive (or large) dataset, in the simplest way, is a dataset, which cannot be stored on single computer: usually this is a dataset with size at least a few TB (there is no strict limit). Such a dataset may contain many physical files; it may be homogeneous (e.g. datasets from *Large Hadron Collider* [11]) or may contain data in various formats (e.g. all kind of documents in an enterprise).

Big Data is more than massive (large) dataset – in this case we do not consider only a size of dataset, there are other important features. One definition uses 3V's characteristic [3] (it covers mainly technical parameters):

1. *Volume* – the size of stored data;
2. *Velocity* – how fast new data is generated and how fast we can access this data;
3. *Variety* – the type and nature of data (structured vs. unstructured, different kind of files: text, spreadsheets, music, pictures, movies and so on).

At present one may consider also other characteristics like [2,20,21]:

4. *Veracity* – messiness or trustworthiness of data; in some cases data is worthless if it's not accurate;
5. *Value* – how we can turn our data into value;
6. *Variability* – may describes few aspects, e.g.: number of inconsistencies in the data, changing speed at which big data is loaded into your database;
7. *Visualization* – how to present data in visual form to see some dependencies;
8. Others: *Validity*, *Vulnerability*, *Volatility*, *Viscosity*, *Virality*, etc..

In this paper we use the first 3V. Moreover, one of most important feature of Big Data (compared to RDBMs or data warehouses) is a different approach of data processing: for RDBMs (and data warehouses) data should be strictly structured before loading to database – as for Big Data, the data is stored in raw form, then the transformations are done later by the target system.

For treatment of big data we should have dedicated infrastructure (with appropriate hardware and software tools) to store and process it. Such an infrastructure should be scalable and fault tolerant.

The another important aspects associated with Big Data are data analysis methods and stakeholders cooperating in Big Data projects. In our paper we will consider only a second aspect that is stakeholder, especially how they may collaborate in teams in projects. It is presented in Fig. 1.

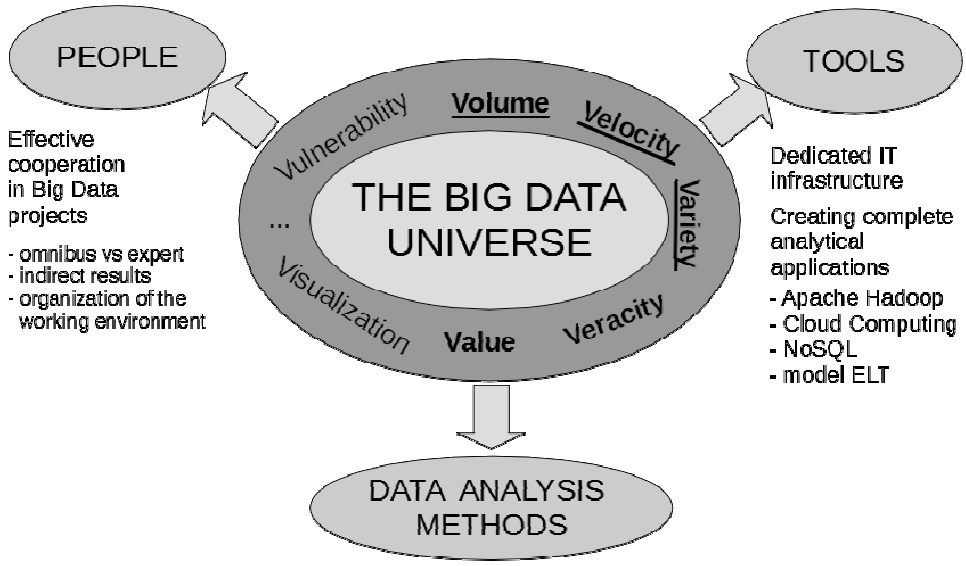

Figure 1. The Big Data universe

1.2. Goals of the work

As mentioned in above Section 1, the presented subject is relatively new, the difficulties with education on this field arise. There are two main aspects of the problem.

The first is a choice of an software platform for processing Big Data during student courses. Needed is a platform, which is free and regarded as a quasi-standard, with many tools available for diverse kinds of tasks and also accompanied with additional study materials (books, web pages, forums, etc.). In the paper we will show that such platform is Apache *Hadoop* [8]. By limited time resources we should show an appropriate subset of Hadoop tools.

The second aspect is the way of teaching in a course student course, especially with regard of group collaboration in a project. Big Data is a vast knowledge domain, difficult to comprehend by individuals collaborating in teams. We do not aim, specialists strictly concerned with one selected field, but prefer omnibuses over specialists.

Another related dilemma related to the first aspect is how to run a Big Data a platform on regular personal computer (PC). The production systems for Big Data are based on extended hardware platforms, but our available hardware resources are limited.

The aspects mentioned above lead us to two main goals of this work:

1. How to prepare a hardware platform for teaching Big Data courses, which can be launched on typical lab computer? A software solution should be enriched by additionally data samples and some prepared tasks and examples. Moreover, a hardware solution (based on free software) should be easy to launch on a student's computer. In the following Sections 2 and 4 we will consider application of some constituents of a *Hadoop* platform.
2. How to improve the teaching paths and a knowledge transfer process and at the same time collaboration in teams? We will focus on this topic in Section 3 and propose a usage of an e-learning tool.

In Section 5 we conclude our work.

2. *Hadoop* – a scalable software platform for distributed computing

An important question is what kind of hardware and software infrastructure should be used for processing large sets of data? It is obvious that it would be too much for single computer, we also know that e.g. RDBMs systems [6] can be scaled only up to a fixed limit. So, we need a system which may be scaled linear, and with a reasonable costs. We need to increase computing power by adding computers instead of replacing them.

There is such a solution - *Hadoop* which is a scalable software platform for distributed computing. *Hadoop* can store practically unlimited size of data and can process this data in distributed environments. It is an open-source, free solution, and relatively simple to scale-up. Of course we must keep in mind, that a scaling hardware generates additional [costs.

Hadoop platform contains three important modules:

- *HDFS* – Hadoop Distributed File System;
- *YARN* – a framework for job scheduling and cluster resource management;
- *MapReduce* – a *YARN*-based system for parallel processing of large data sets.

In a distributed *Hadoop* environment each part of data is stored in several copies (usually at least 3) on different computers (i.e. cluster nodes). One of the most important assumption is that data is processed locally. It means that data is processed where it is stored (on the same computer/node) which minimizes the network transfers. Moreover, the system is fault tolerant, i.e., when one of the nodes fails, the results from this node are lost. So only a repetition of the calculations on from a broken node are needed. What is more, such situations are managed by the system itself, so user do not need to undertake additional actions.

Now we will give short description of main *Hadoop* platform items: *HDFS* and *MapReduce*.

2.1. *HDFS – Hadoop* Distributed File System

As mentioned in the beginning of this Section, in *Hadoop* each file stored in a file system is divided into several parts (blocks), and each part is stored in several copies on different locations. In Fig. 2. a schema of processing model in *HDFS* architecture is presented. It is worth mentioning, that *HDFS* structure is *rack-aware*, which means that if you have a cluster build with nodes in many racks, its data will be distributed so as to minimize the effects of the failure of the whole single rack.

Commands used for managing files in *HDFS* are very similar to those used in *Linux* operating system [1, 5, 7].

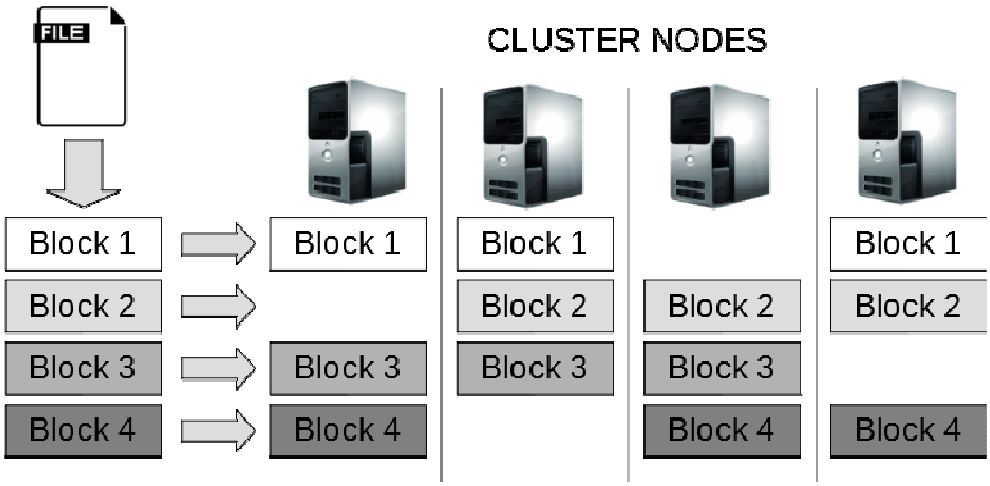

Figure 2. Processing schema in *HDFS* architecture

2.2. *MapReduce* model

MapReduce is a framework (with tools and methods) for parallel processing data in *Hadoop* environment. There are:

- a *map* operation in which for every record we calculate the key-value pairs; all pairs with the same key will be in the same group, e.g. if we process weather data and for every observations we calculate a pair year-temperature, all observations with the same year will be in the same group;
- a *reduce* operation in which for each group we calculate some features from value (aggregate), e.g. for every group of observations calculated in previous point, we may calculate for instance a maximum temperature.

This scheme is very similar to grouping and aggregating information in SQL [22].

We may create an appropriate map and reduce procedures (methods) in many programming languages (i.e. JAVA or Python), or we may use more specialized tools, like *Pig* [1, 5] – a tool with an own language similar to SQL.

2.3. *Hadoop* – choosing tools for teaching courses

We mentioned above only three tools connected with *Hadoop*: *HDFS*, *MapReduce* and *Pig*. But still, there are much more such tools available. We should keep in mind, that we should provide tools for creating complete analytical applications (paths) for solving given Big Data problems (see Fig. 3.).

There are many diverse tools for solving different problems (see Fig. 3), so we had to choose which tools should we use for our aims (there is no time available for teaching too many tools in one course). We have chosen the following applications: *HDFS* (Hadoop Distributed File System), *Avro* (data serialization system), *MapReduce* (system for parallel processing of data) and *Pig* (language and system which simplify *MapReduce* processing).

In Section 4 we will continue with further information on preparing *Hadoop* platform for a Big Data student course.

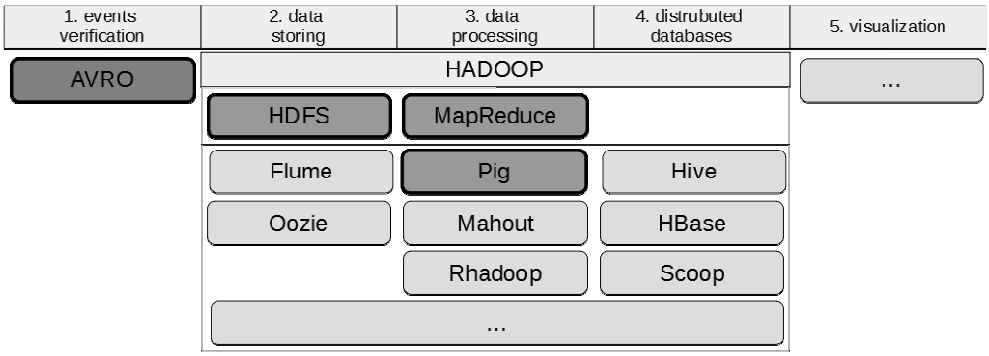

Figure 3. Tools for creating complete analytic application (*Hadoop*)

3. *Moodle* – an e-learning software platform

Even if we choose a limited set of big data tools, a range of learning materials is still large. So we must use tools dedicated for improve learning paths, like an e-learning platform to improve cooperation between a teacher and students, as well as directly between the students.

Such a tool should give us possibilities:

- to prepare and publish additional materials (tasks, exercises and examples, etc.), visible only for dedicated group(-s) of students;

- for certain tasks to request answers or solutions (within given time limits); we may observe them and react (comment or correct). Often tasks are multi-stage, and before going to the next stage students must complete the previous tasks;
- to report by the students some problems, doubts, propositions or solutions; it gives the teachers some opportunities to learn something new and/or improve a course;
- to prepare own materials and to share them with others students (and teachers);
- to present the obtained students' results; this gives opportunities for verification and comparison of code, solutions, data, indirect results, etc., with other team members and to discuss.

Apart from the primary purposes (for the teachers), a cooperation should give the students opportunities to influence on a course and to extend knowledge according to their interest. There are opportunities to learn and improve team work.

For our Big Data course decided to chose *Moodle* [4,10] - a free software tool for supporting traditional lecture and laboratory. It is a worldwide known e-learning platform, which can be used to enhance traditional forms of teaching by sharing links and various resources or making possible new additional interactions in communication (not limited to e-mail and personal contacts). This platform is very popular and appreciated in universities. It is used by several faculties in our university as well. Thanks to a wide range of functionalities like: sharing files, glossary, wiki, links, quizzes, forums, chats, blogs, workshops, *Moodle* fulfils our needs of improving cooperation between teachers and students, as well as directly between students. It also helps to stimulate students to be more active and responsible for their education. Moreover, very important to us is:

- multilingualism – usage of several languages simultaneously (configurable);
- availability of extensive documentation (including Polish language) - due to the great popularity of such software, there is a large community of users (also Polish), so it is easy to get a help;
- a mobile-compatible use interface and a cross-browser compatibility;
- a customizable interface - there are many ready-to-use themes (free of charge or paid), available on many web pages, including *Moodle* page [10].

For installing it from scratch (for training or testing purposes), one must have a valid web account with PHP handling, an access to SQL database (usually *MySQL*, especially for external web providers) and an e-mail box (for outgoing information). For installing *Moodle* the following actions are needed: download of a *Moodle* package (compressed archive) and unziping it to the web account. Next one should start the own page in web browser and follow the instructions to complete installing *Moodle*.

4. Preparing *Hadoop* platform for a Big Data student course

It will be shown how to prepare *Hadoop* platform with specific tools on a single computer. Of course a relatively modern computer, capable enough to run typical tasks at home or laboratory (see below). Such a computer usually makes use of a *Windows* operating system, and this leads to the first problem because the tools for processing Big Data are based on *Linux*. The proposed solution is a virtualization platform – below will be shown how to use *Oracle VM VirtualBox* [13] for running a separate machine with *Linux* (Fig. 4.).

For preparing such a platform basic knowledge of using *Linux* (including installation) is required. In addition, we should have a computer with an operating system and *VirtualBox* installed. Such a computer should contain at least 8GB of memory (we need memory for a host operating system and additionally for a virtual machine). Computer with at least two cores is recommended. *VirtualBox* can replaced with any other virtualization platform (e.g.: *VM Ware* or *QEmu*) [14, 15].

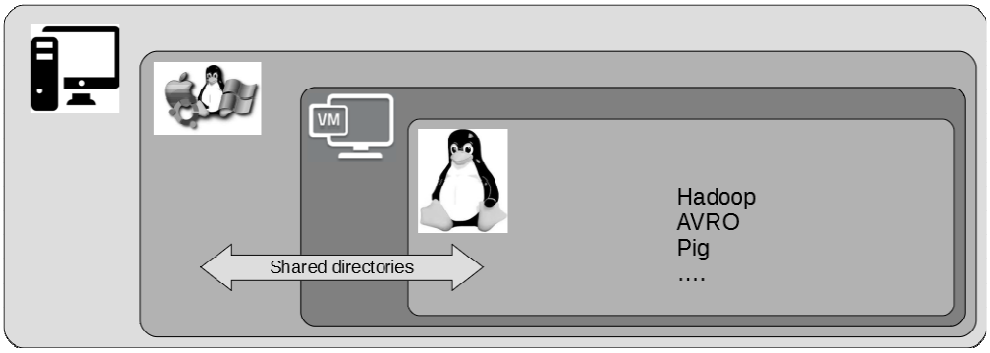

Figure 4. Preparing *Hadoop* platform on Virtual Machine VM

Only the basic steps for creating a virtual machine with operating system *Linux* and *Hadoop* platform will be shown.

Linux

- prepare your favourite distribution of *Linux* (download ISO image): we use *Ubuntu* (with *XFCE* graphical desktop – recommended, but not required);
- create a virtual machine with at least 4GB of memory (RAM) and at least 200GB of disk space (HDD) – not all this space will be required for running machine (virtual disk will be expanded on demand); if you have computer with small disk you may use an external drive (USB 3.0);
- install *Linux* from ISO image: usually it is enough to confirm default options in an installer;

- install *Guest Additions* –some additional features (like fitting screen size of *Linux* to *Windows* operating system window size or ability to exchange files with the host system) will be given;
- install Java (*Java Development Kit*);
- recommended: add an ability to run commands with admin access rights without password to current user (`sudo` command in *Ubuntu*);
- recommended: install *Midnight Commander* (*mc*) –visual file manager.

Hadoop (*HDFS*, *MapReduce*)

In the simplest scenario only downloading of *Hadoop* binaries (version 2.7) and unpacking it to any directory is needed. Next, script variables *HADOOP_HOME* and *JAVA_HOME* in `hadoop-env.sh` file must be setup. In this way a working instance of your own *Hadoop* environment is made available. It is standalone version with no dedicated file system (*HDFS*) – file system for this version of *Hadoop* is common with base *Linux* file system. It is recommended to add a path to *Hadoop* binaries directory to your system `PATH` variable.

For using such system run `hadoop-env.sh` file should be executed at first. Then execute tasks (e.g. example *WordCount* included in *Hadoop* distribution).

If a system with separate (dedicated) *HDFS* system is desirable, few files should be modified [8]. Finally, for the first usage one should format *HDFS* (`hadoop namenode -format`) and execute services (`start-all.sh`). It should be considered, that *Pseudo-Distributed Mode* is a usage of one machine, that makes files (blocks) replication impossible.

For installation also additional tools will be used, but description of installing each of them is out of scope of this paper. Some of these tools require additional tasks to install them (i.e., compiling from source), so it is recommended to installing system tools like *pip*, *git*, *snappy*, *ant*, *maven*, and so on. The tools installed in our implementation are: *Python* environment, *Avro*; *Pig* and *Eclipse IDE* (*Java Developers* edition with plug-ins for editing *Python* and *Pig* scripts).

Preparing an own *Hadoop* cluster with additional tools

In the beginning of this Section we considered preparing *Hadoop* platform on a single computer. A very valuable experience is to show students both: the process of creating a cluster for Big Data and the benefits from using it. Below it will be shown how to prepare a *Hadoop* cluster.

First of all, to build a cluster, an appropriate hardware resources are required:

- *NameNode* – computer which will act like a monitor and supervise the operation performed in our cluster; we will use computer with 64GB of RAM;

- *DataNode* – computer which stores the files and processes them; we will use 6 computers with 24GB RAM (at least 3 to be able to show the principles of HDFS);
- all computers are created as a virtual machines in virtualized environment (*Xen*), each machine has allocated 4 processor cores.

Because any block in *HDFS* is stored on at least 3 machines (default), for each 1MB of data we need 3MB of disk space (plus space for operating system, tools and space for calculating/processing data). We should remember that when calculating the required disk space. Moreover, for the same reason, to see how *HDFS* system works (data partitioning and replication) we should use at least 4 machines (see Fig. 2.).

Our cluster is based on *Linux* (*Ubuntu*, [9]), so only two versions of the machines should be prepared: *NameNode* and *DataNode* (only operating system, prepared with the same procedure as for single machine – see Section 4). In virtualized environment it is possible to copy machines.

It is possible to prepare your *Hadoop* cluster directly from binaries, but in this case the configuration files need to be edited manually. A much simpler way is a usage of an integrated distribution like *Cloudera* [16] or *Ambari* [17], which contain additional integrated tools. For installing such distribution (we have chosen *Cloudera*) its binaries should be downloaded into your *NameNode* machine, executed (install *Cloudera Manager*) and one can launch a web browser with a proper link (local machine address with a specific port). From browser the machines on which you want to install the software (*DataNode*) can be selected, and the desired tools. Installing software separately on each machine is no not need.

Hadoop cluster application example

Data size: 900MB; weather data from NOAA [12], text files, unpacked, with minor corrections.
Task: calculate minimal and maximal temperature for each year (1901 – 2016).
Results (processing time):

- for a single node Pseudo-distributed mode (one computer): about 21 hours;
- for a presented cluster: 2 hours 13 minutes.

Other options for training Big Data are for instance:

- ready-to-use virtual machines with Big Data tools: *Oracle BigData Lite Virtual Machine*, *Hortonworks Sanbox on VM*, *Cloudera QuickStart VM* (required 4-8GB of RAM for machine);
- dedicated cloud solution (free of charge): *IBM Analytics Demo Cloud* (*Ambari*): 4 machines with 32 cores each, 3x64GB RAM (*DataNode*) and 1x256GB RAM (*NameNode*);

- commercial clouds (but with free starting period or starting credit): *Amazon Web Services* [18], *Microsoft Azure* [19].

5. Conclusion

For teaching students in the domain Big Data appropriate tools and methods are needed. We have introduced an approach by using an e-learnig platform *Moodle* and *Hadoop* platform tools (processed on the single machine and at the *Hadoop* cluster).

As for the experiences with the e-learning platform *Moodle* we can say that it met the initial expectations in concerning cooperation with the students. We can state, that after implementation, the effectiveness of teaching in our course as to knowledge sharing has increased, compared to the previous education cycles. Nevertheless, the much more additional time is required for the staff to prepare teaching materials available on the platform. Even so, there is still a low awareness of students in the field of cooperation within the groups.

Considering prepared *Hadoop* platform (for one computer) we can state that it can be started on any relatively modern computer. The process of creating such a machine can be carried out independently with open, free software. In our course the virtual machine consists basically of two files, easy to upload (both the first version and then the updates). A teacher can prepare the software on his computer and then upload or replace it. A limitation may be a file size – with large files and network 100Mbit speed the upload takes a few hours.

The snapshots feature allows to save the machine state and, after class, to restore the machine to its original state. This can also be done by copying the virtual machine file, but due to its size much more time is needed. The feature of machines' separation allows each student a usage of an own machine.

Thus, the prepared solutions have greatly improved the education of in Big Data domain, the effectiveness of teaching has been increased. However low students' awareness of the need for team collaboration needs improvements and this will be our next goal by using additional *Moodle* capabilities, such as wikis.

Our plans for the future also include usage of the extended the Big Data tools set with NoSQL databases and tools for data visualization and presentation.

REFERENCES

[1] T. White (2015) *Hadoop: The Definitive Guide, 4th Edition*, O'Reilly Media, Inc (polish edition by Helion, Gliwice, 2016).

[2] M. Tabakow, J. Korczak, B. Franczyk (2014) *Big Data − definitions, challenges and information technologies*, BUSINESS INFORMATICS 1(31) (in Polish).

[3] EMC Education Services – Editor (2015) *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*, John Wiley and Sons, Inc., Indianapolis, Indiana.

[4] W. H. Rise (2008) *Moodle 1.9 e-learning course development: a complete guide to successful learning using Moodle 1.9*, Packt Publishing Ltd., Birmingham, UK (polish edition by Helion, Gliwice 2010).

[5] R. Jurney (2013) *Agile Data Science: Building Data Analytics Applications with Hadoop*, O'Reilly Media, Inc. (polish edition by Helion, Gliwice, 2015).

[6] M. Grzenda, J. Legierski (2017) *Databases, data warehouses, Big Data platforms – variety of needs and solutions*, Data Science Summit 2017.

[7] The Ubuntu Manual Team (2014) *Getting Started with Ubuntu 14.04, Second Edition*, http://ubuntu-manual.org/, 12-11-2015.

[8] The Apache Hadoop Webpage, http://hadoop.apache.org/, 20-06-2017.

[9] The Ubuntu Webpage, https://www.ubuntu.com/, 07-03-2017.

[10] The Moodle Webpage, https://moodle.com/, 27-05-2017.

[11] Large Hadron Collider, http://opendata.cern.ch/, 01-09-2017.

[12] National Centers for Environmental Information, National Oceanic and Atmospheric Administration Webpage, http://www.noaa.gov/, 20-09-2016.

[13] The Oracle VM VirtualBox Page, https://www.virtualbox.org/, 20-04-2017.

[14] The VMWare Page, https://www.vmware.com/, 20-04-2017.

[15] The QEmu Page, https://www.qemu.org/, 20-04-2017.

[16] The Cloudera Webpage, https://www.cloudera.com/, 02-02-2017.

[17] The Apache Ambari Webpage, https://ambari.apache.org/, 20-06-2017.

[18] The Amazon Web Services, https://aws.amazon.com/, 15-092017.

[19] The Microsoft Azure, https://azure.microsoft.com/, 15-09-2017.

[20] B. Marr (2015) *Why only one of the 5 Vs of big data really matters*, IBM BigData and Analitic Hub, http://www.ibmbigdatahub.com/blog/why-only-one-5-vs-big-data-really-matters, 30-04-2017

[21] T. Shafer (2017) *The 42 V's of Big Data and Data Science*, Elder Research, https://www.elderresearch.com/company/blog/42-v-of-big-data, 30-04-2017

[22] Oracle Database 12c SQL Language Reference, https://docs.oracle.com/database/122/SQLRF/toc.htm, 30-05-2017

[23] S. Robak, B. Franczyk, M. Robak (2014) *Research Problems Associated with Big Data Utilization in Logistics and Supply Chains Design and Management*, Annals of Computer Science and Information Systems, Volume 3.