Journal of Theoretical and Applied Information Technology
31st October 2017. Vol.95. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RESEARCH OF THE EFFECTIVENESS OF SQL ENGINES WORKING IN HDFS

ANDREY KOLYCHEV, KONSTANTIN ZAYTSEV
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Kashirskoe Avenue 31, Moscow, 115409, Russia

ABSTRACT

The rapid data growth at the beginning of the XXI century gave impetus to the development of big data technologies. A distributed platform Hadoop became a key element of big data technologies. Initially, it was difficult using Hadoop for tabular data processing, on which many modern industrial information systems are built. Therefore, a variety of SQL tools for Hadoop began to appear, which gave rise to the problem of choosing a specific solution. The aim of this work is to identify the most efficient SQL tools for tabular data processing in a distributed Hadoop system. For this purpose a comparative analysis of six most popular tools: Apache Hive, Cloudera Impala, Spark SQL, Presto, Apache Drill, Apache HAWQ has been done. The result of the study was the choice of the most effective means from the standpoint of completeness of the list of functions performed, tool performance and the level of SQL standards support. After summarizing of the results of a study, which has been done on all selected space coordinates comparison, Presto was the most effective tool.

Keywords: *Hadoop, SQL-Like Syntax, SQL Engine, Software Functionality, Performance, Support For The Standard SQL Language, SQL Tools, Apache HIVE, Cloudera IMPALA, SPARK SQL, Presto, Apache DRILL, Apache HAWQ.*

1. INTRODUCTION

The development of information technologies, cheaper hardware and its availability have dramatically increased the number of Internet users in the beginning of the XXI century, which led to a rapid increase for data that must be stored and processed. IDC predicts, that by 2020, the digital universe will have the amount up to 40 zettabytes, which exceeds the previous forecast for 5 zettabytes. Since the beginning of 2010, the volume of data has increased by 50 times [1]. It was the impetus for the emergence and development of Big Data technologies, designed to handle efficiently large amounts of information.

One of the central elements of technologies, working with big data is Hadoop, a software framework for the development and execution of distributed programs, running on clusters of hundreds or thousands of nodes. Using it, you can securely store and process huge volumes of various data, which are different in structure. The need for processing tabular data, of course, has not disappeared; making it possible, without significant loss of data, migrate running applications to the new platform. Initially, to process data using Hadoop, it was necessary to write programs consisting MapReduce tasks, writing of which required a lot of time and programming skills. At the same time in Data Science, one of the most popular programming languages is SQL [2]. Therefore, tools integrated with Hadoop, gradually began to appear, using which, you can write queries to data on the well-known SQL language. They are all more or less similar to each other, and are designed to solve a set of the same tasks, because of it, a problem of choice of a specific tool arises.

Various researchers use various approaches consider this problem. Processing tabular data tools are often compared by the processing speed [3, 4, 5]. Some researchers, in addition, appreciate such tools from the standpoint of the scope of their functions [6]. The researchers also compare tools, taking into account their popularity and prevalence, and pay attention to the size of the developers' community [7]. There are studies, based on distinction between tools [8]. In addition, some researchers do not try to compare tools, but especially try to highlight the most important integrated criteria, by which it would be possible to compare the tools and select the specific product [9, 10]. Some studies, in addition to the performance, draw attention to support of SQL-standards tools [11, 12].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

All these approaches have the right to life. However, they do not provide a general picture allowing to choose an efficient SQL engine in the space of the main measured features - software functionality, performance, support for the standard SQL language. This article is devoted to the solution of this problem.

2. MATERIALS AND METHOD

2.1. Tools Selection Method

The most common tools, which are used as a language for writing queries to SQL data, were selected. A set of tools, candidates for the study, was obtained from the analysis of published materials on tools, integrated with Hadoop, using SQL as query language. The selection criteria of the analysed printed documents were the following:

- the topic of a text being analysed should strictly follow the direction of "analysis/comparative overview of the tools that use SQL to work in Hadoop.

- the authority of the author or an electronic resource, that is, the material has been fetched, if the author is a specialist in this or related fields; or electronic resource material, where the material has been published, publishes materials in this subject area.

These specially selected sources of information were prepared for the analysis of a list of six tools that are mentioned in more than half of the sources. These tools were the following SQL-on-Hadoop tools.

Apache Hive. This software is for data warehousing on the base of Hadoop. Hive makes it easier to read, write, and manage large amounts of data and allows querying to data with SQL-like syntax. The Hive emerged because of the task solution management and extraction of information from huge volumes of data, produced by social network Facebook [13].

Cloudera Impala. This system is for making high-performance SQL queries for data, stored in popular formats in Hadoop. Cloudera Impala integrates with a database Apache Hive; it is possible to exchange databases and tables between these components. The compatibility of Impala with HiveQL allows you to use both Hive and Impala. Impala uses this format as Parquet (column-oriented data storage format), which is optimized for large queries, typical for data warehouses [14].

Presto. This is an open-source tool to run interactive SQL queries for data sources of all sizes, ranging from gigabytes to petabytes. Presto allows to query Hive, Cassandra relational databases and from private repositories. Supports popular file formats: Text, SequenceFile, RCFile, ORC and Parquet. To work with Hive, the access to metadata is required. Presto does not use MapReduce and it is required to interact only with HDFS. It is possible to combine in one query data from different sources [15].

Apache Spark. This is a software open source platform for the implementation of distributed processing of unstructured and semi-structured data, part of Hadoop projects ecosystem. In contrast to the classical processor from the core Hadoop, that implements a two-level concept of MapReduce disk storage, it uses specialized primitives for in-memory processing [16]. *Spark SQL* is a software platform Spark module, which has been added for more convenient and effective work with structured and weakly structured data. Spark SQL supports many data sources and work with them effectively [17].

Apache Drill. This is a tool for writing data processing programs. The work does not require a centralized metadata repository. In Drill, there is also a query optimizer, which focuses, in particular, on data storage, so it automatically restructures the query execution plan, to use storage capabilities for data processing [18].

Apache HAWQ. This is a SQL-engine for Hadoop, which interacts with HDFS directly and combines the advantages of databases with massively parallel architecture (massive parallel processing, MPP), scalability and convenience of Hadoop [19].

2.2. The Choice Of Space Comparison Coordinate

For any tool, the most important characteristics is the amount of features that a tool is able to perform and performance, that is the ratio of work performed to the time of its execution. Therefore, as tools for working with data using SQL, it is very important to know the extent, to which they support SQL standards. The following space comparison coordinates were chosen:

1) amount of tools for data processing functions;

2) performance;

3) amount SQL-standards support of the selected tools.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3. A Comparison According To The Amount Of Functions Performed

The tools were compared on the following features:

file formats which the tool supports;

data sources from which the tool supports the data request;

availability of libraries for machine learning;

ability to create user-defined functions;

ability to combine data from different sources in a single request.

These criteria were chosen as supported file formats and data sources are among the most important parameters of data processing, as on this depends directly the usefulness of this tool; the availability of machine learning libraries is needed, since many algorithms for data analysis built on machine learning; creation of user defined functions is important, because, no matter how rich the library of built-in functions is, the time will come, when it will be necessary to determine its function with unique behaviour; the possibility of a combination of data from different sources is important, because not all data are always in HDFS and it does not always make sense to transfer them there, but the situation may arise when you need simultaneously process data from different sources.

The sum of evaluations was calculated according to the formula (1).

$$S = \sum_{i=1}^{n} V_i \cdot Z_i \text{, where (1)}$$

S is a tool assessment;

N is a number of criteria comparisons;

Zi is a value of criterion fulfilment;

Vi is a weight of the criterion (0 to 1).

All criteria have the same weight: 1/15, where 15 is the number of criteria. It was done so, because it was not possible to conduct a survey among experts. The technique of estimation of the criterion was to assign a specific value:

0 if the criterion is not performed or is performed using third-party software;

1 if the criterion is met fully or with minor restrictions;

0.5 if the criterion is met with significant limitations.

2.4. Comparison According To The Performance

One of the most important criteria in the selection of any tool is its performance. It was not possible to measure independently the performance of the selected tools due to the lack of the cluster, and performance testing of distributed tools on a single computer does not make sense. However, other researchers conducted few studies in this area, but it is not worth relying on any one of them while choosing a tool, and therefore, it was decided to make the analysis of performance tests, which have already been carried out. The ranks were given according to the results of each analysis of performance test tools. Grades are assigned, so that the best tools are assigned rank 1, next rank 2 and so on, if several symbols correspond to one rank, the rank was calculated according to the formula (2)

$$r = \sum_{i=1}^{n} r_i \text{, where (3)}$$

r is the final instruments grade.

n is the number of tools that corresponds to one rank.

r' is a rank, which conforms to all the tools.

Then a synthesis of the results using the interactive method, ORESTE [20] was produced.

2.5. A Comparison According To The Amount Of Support Of SQL Standards

While choosing the criteria for the theoretical comparison of tools, another criterion should be support of SQL standards by tools, but a more detailed study of the documentation of tools revealed, that most of the tools have limited support for SQL, the degree of this limitation could not be established. Apparently, the question of compliance with the standard is still too delicate for most tools that position themselves as a SQL tool for working in the Hadoop environment.

It was decided to establish the degree of conformity of syntax of SQL tools to the standards in our own hands, as it is one of the most important criteria.

To establish the degree of support for SQL standards by tools, queries of the benchmark TPC-H from TPC company were used [21], specializing in the development of benchmarks for databases and data processing systems. The benchmark consists of twenty-two queries that contain a wide range of operators, and nested subqueries. All queries are written in SQL-92. To test the queries, a virtual machine with installed Hadoop was used, and this or that particular tool was being tested.

2.6. Summarizing Of Results

The generalization of results of comparison was performed by summing the ranks (3), assigned to the tools according to the results of all

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparisons (grades are assigned as described in section 2.4. scheme).

$$r = \sum_{i=1}^{n} r_i, \text{where} \quad (3)$$

r is the total rank of the tool;

r_i is the rank of the results of i-th coordinate comparison;

n is the number of space comparison coordinates.

An important issue when using ranking methods is the identification of the integrity of the set of estimated values. In the present work, three coordinate factors were used, for which the evaluation of the tools was carried out. The integrity of the identification was determined individually for each factor:

- The volume of functions being implemented is a logical combination of the RF of the individual SQL engines selected in the filtering step, and therefore is full of comparative instances;
- performance is a measurable factor, so it is full of comparative instances;
- the amount of support for the SQL standard is determined by the compliance of the TPC-H benchmark known in this area, therefore, for lack of the best for today, we can assume that it is full of comparative copies.

3. RESULTS

3.1. Tools Choice

Selection of tools, which are candidates for a comparison, conducted in accordance with 2.1., gave the following picture (Fig. 1) by the number of mentions in specially selected analytical articles.

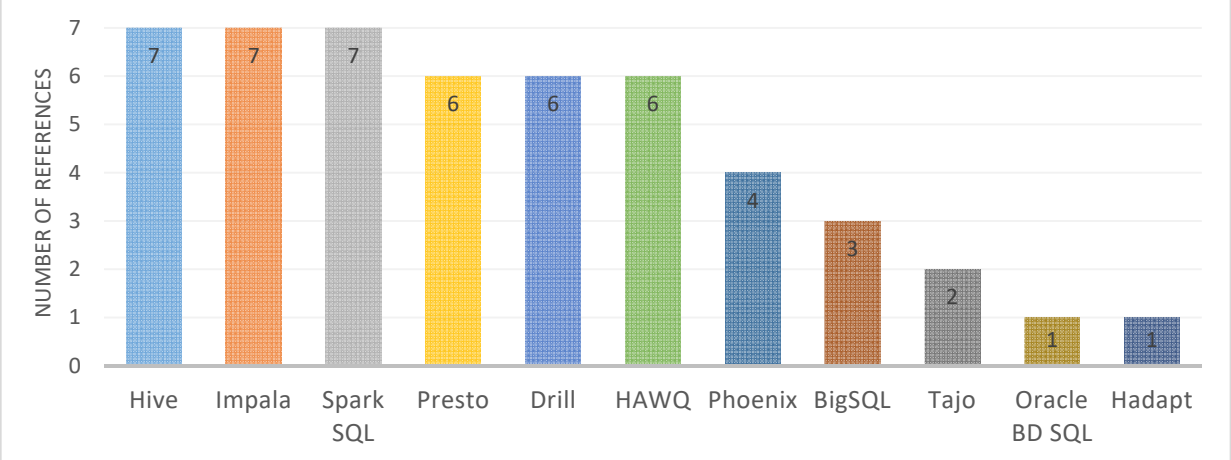

Fig. 1. Number of tools references in the articles being analyzed

The following tools were mentioned from four to seven times (that is more than in half of the materials): Hive, Impala, Spark SQL, Presto, Drill, HAWQ, Apache Phoenix. However, Apache Phoenix is mainly designed to work with HBase that is a database-class NoSQL running on top of HDFS. Thus, most often mentioned (from 6 to 7 times), are Hive, Impala, Spark SQL, Presto, Drill, HAWQ.

3.2. The Results Of The Comparison According To The Functions Performed

The results of the comparison of tools, according to the amount of functions performed, are presented in Table 1. Values in the table are the result of analysis of official websites documentation of studied tools [14-16, 18-19, 22]. As can be seen from Table 1, the comparison criteria are supported data sources, supported data formats, user-defined functions. The available libraries of machine learning and the ability to combine data from different sources in one query.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table1. Comparison Of Tools According To Their Volume Functions.

Criterion		Weight	*Hive*	*Impala*	*Spark SQL*	*Presto*	*Drill*	*HAWQ*
Supported data sources	HBASE	1/15	1	1	1	0	1	1
	RDBMS (JDBC)	1/15	0	0	1	1	1	1
	MongoDB	1/15	0	0	0	1	1	0
	Amazon S3	1/15	0	1	1	1	1	0
Supported data formats	Text File (CSV TSV PSV)	1/15	1	1	1	1	1	1
	Sequence File	1/15	1	1	1	1	1	1
	RCFile	1/15	1	1	1	1	0	0,5
	Avro Files	1/15	1	1	1	0	1	1
	ORC Files	1/15	1	0	1	1	0	0,5
	Parquet	1/15	1	1	1	1	1	0
	XML	1/15	1	0	1	0	0	1
	JSON	1/15	1	0	1	1	1	1
User-defined functions		1/15	1	1	1	1	1	1
Machine learning libraries		1/15	1	1	1	1	0	1
Combining data from different sources in a single query		1/15	0	0	1	1	1	0
Summing up			0,73	0,6	0,93	0,8	0,73	0,67

Graphically, the results of comparison are depicted in Fig. 2. As can be seen, the best tool here is Spark SQL.

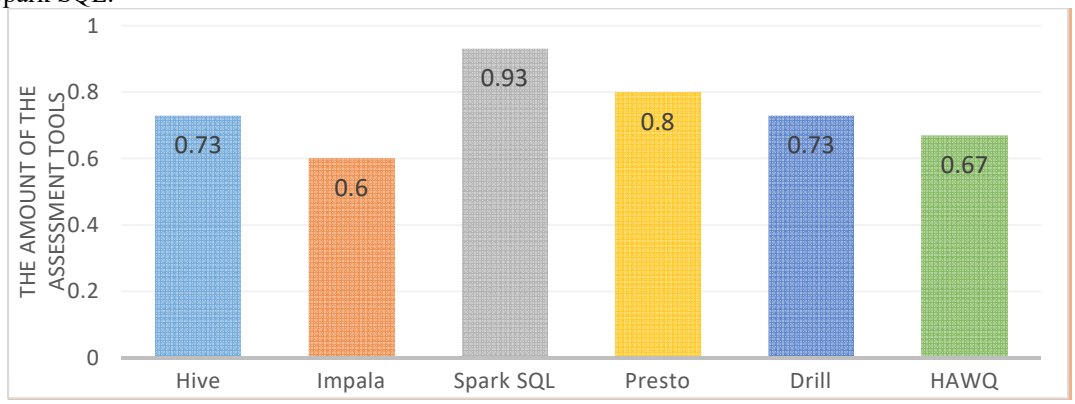

Fig. 2.The Amount Of Assessment Tools According To Their Functions.

3.3. Performance comparison results

A comparison of tools performance was conducted in accordance with the method described in 2.4. The ranking of tools after the comparison presented in figure 3.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

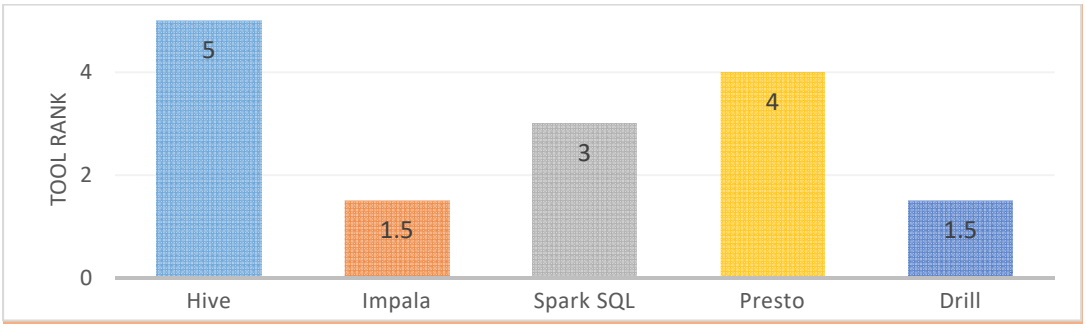

Fig. 3. Performance Tools Ranking (The Less Is The Better).

The figure shows, that the best in this comparison are tools Cloudera Impala and Apache Drill. Unfortunately, we failed to find information about testing and performance of HAWQ tools. Therefore, this tool is missing in the diagram.

3.4. The results of tools comparison according to the amount of support for SQL standards

A study of the amount of SQL standard support was carried out in accordance with the method, described in paragraph 2.5 of the present study, it gave the results, presented by the diagrams in Fig. 4 and 5.

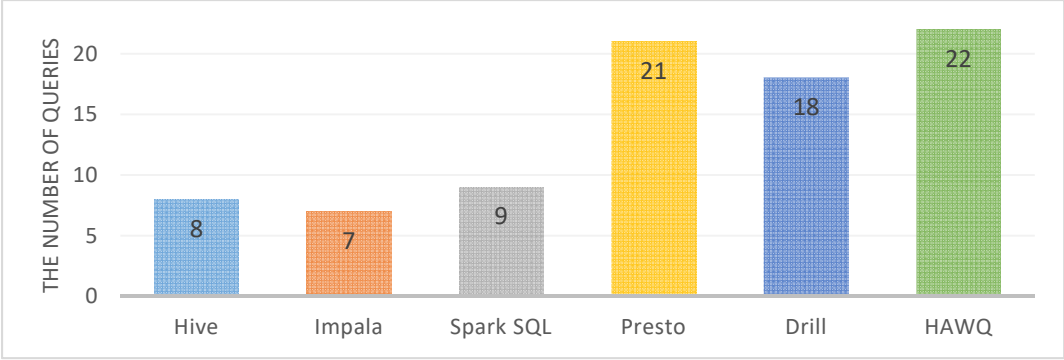

Fig. 4. The Number Of Queries Without Edits.

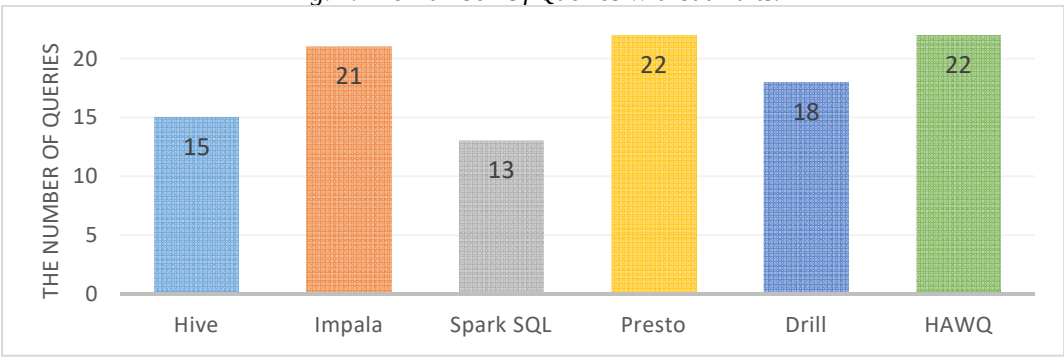

Fig. 5. The Number Of Queries With Edits.

The first bar graph shows how many queries were executed on each of the surveyed tool, without any amendments to the code of the request. HAWQ leads in this parameter, 22 of the requests were performed using it, after that comes Presto, etc. The second chart illustrates the number of queries that were performed on tools with minor code edits. Where minor revisions mean minor changes in the code of the query, for example, another name for built-in functions, that is, such changes do not change the structure of the query and not break the query into multiple subqueries or require changes, as a rule, in one line of the query. According to the results of this comparison, the best tools are Apache HAWQ and Presto.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.5. General comparison result

In accordance with the method described in 2.6, a generalization of the results of tools comparison, obtained on the selected indicators, gave the following picture (Fig. 6 and 7).

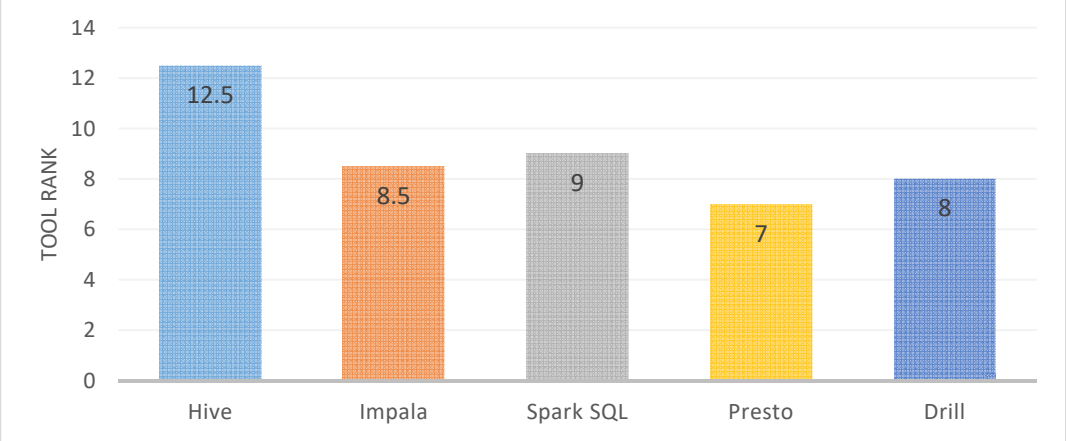

Fig. 6. Final Rank Without HAWQ (The Less Is The Better).

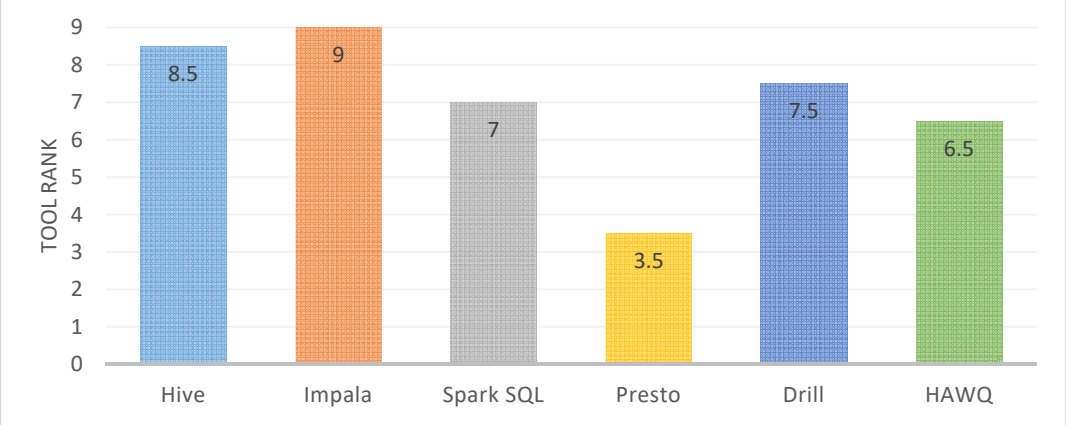

Fig. 7. Final Rank With HAWQ (The Less Is The Better).

As Apache HAWQ tool did not participate in the comparison with respect to the second coordinate, two generalisations have been done: with respect to all three coordinates (without HAWQ) and with respect to the first and the third coordinate (with HAWQ). According to the results of both generalizations, the most effective tool was Presto.

4. DISCUSSION

There are different approaches to solving the problem of choosing SQL tools for processing tabular data in Hadoop. Usually this problem is solved by comparing the tools according to any criterion, justified in the article. The most popular of these criteria is the performance [3-6]. However, increasing the number of instruments available on the market of SQL-on-Hadoop tools, there is also a task of pre-filtering of analysing means, for forming the set of tools that makes sense to compare, taking into account their distribution in the user's environment [7]. The desire to use familiar tools in various domains has led to the need to incorporate new features, such as machine learning libraries or combining data from different sources in a single request. This, in turn, gave rise to a new criterion of comparison to the scope of tools functions [6, 9, 10].

Initially, the SQL-on-Hadoop tools syntax was very limited, compared to SQL standard, but over time it is becoming closer to the SQL standards, and the question associated with the amount of SQL-standards tools support was also considered by researchers as one of the criteria [11, 12]. The choice of one of the most preferable criterion for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

tools comparison to date is questionable, and the choice has to be based on several criteria that have the biggest weight in a particular subject area.

The approach to the comparison of tools, proposed in the article, is based on a comparison of their several, most commonly used by other investigators criteria, and is a comprehensive comparison of tools, which are researched from different angles. The more preferred choice of tool can be done according to the integral criterion taking into account weights of individual criteria essential in a particular subject area.

In this paper, there are some limitations or assumptions:

- selection of source SQL engines for comparison,
- selection of the most significant criteria for comparing SQL engines,
- Use of virtual machines to evaluate the performance of SQL engines.

All restrictions and assumptions are caused only by the desire to focus attention on comparing the most significant products in the market with the use of the most significant factors.

It is worth noting, that a versatile instrument, equally successfully working in various areas of processing SQL queries, does not exist today. Therefore, it is possible that some of the described tools work together on the same cluster. However, this interaction becomes the "narrow neck" of the project, as because all the tools use their own internal data storage format, when transferring data between tools, CPU resources will be mostly spent on their serialization and deserialization. This problem has prompted tools developers, such as Calcite, Cassandra, Drill, Hadoop, HBase, Ibis, Impala, Kudu, Pandas, Parquet, Phoenix, Spark, Storm to join their efforts to develop a universal product, the Apache Arrow, which is positioned as a new high-performance interface between different systems. It also focused on a wide range of common programming languages used in these tools [23].

5. CONCLUSION

This work is devoted to identifying the most efficient SQL-on-Hadoop tool. Such common tools as Apache Hive, Cloudera Impala, Presto, Spark SQL, Apache Drill, Apache HAWQ were investigated.

For comparison of tools, the workbench with a three-dimensional space with coordinates: "the scope of functions performed by the tools", "productivity", "the amount of support the SQL standards" was built.

A comparison of scope of functions performed by the tools was conducted by the selection criteria for the comparison and study of the documentation for the selected tools. According to the results of this comparison, Spark SQL has become the best tool.

Performance tools comparison was conducted by examining the results of performance tests, conducted by other researchers. The problem of generalization of these results was also solved. According to the results of this comparison, Cloudera Impala and Apache Drill have become the best tools. The results of performance tests with the participation of the sixth tool (Apache HAWQ) have not been found.

Comparison of tools in terms of support for SQL standards included a practical review of test case execution of SQL queries on the selected tools. This was done using VM with installed tools. Apache HAWQ and Presto have become the best tools according to the results of this comparison. Analysing SQL queries, which were not performed, limitations in supporting SQL, standards tools were identified.

In conclusion, the generalization of comparison results for all coordinates of space comparison has been performed. Apache HAWQ did not participate in the comparison on the second coordinate as a tool, two generalisations have been done: on all three coordinates (without HAWQ), the first, and the third coordinate (with HAWQ). According to the results of both generalizations, Presto has become the most effective tool.

ACKNOWLEDGMENTS

This work was supported by the Competitiveness Program of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), contract with the Ministry of Education and Science of the Russian Federation No. 02.A03.21.0005, 27.08.2013.

REFERENCES:

[1].Growth in the volume of information - the realities of the digital universe. . Technologies and means of communication. (2015). [http://www.tssonline.ru/articles2/fix-corp/rost-obema-informatsii--realii-tsifrovoy-vselennoy (Retrieved: 10.06.2017).

[2].King J. (2016) Data Science Salary Survey. Sebastopol: O'Reilly Media

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3].Kornacker M. (2015). 7th Biennial Conference on Innovative Data Systems Research (CIDR'15). Impala: A Modern, Open-Source SQL Engine for Hadoop. Asilomar, California, USA. 2015.

[4].The Business Intelligence for Hadoop Benchmark 2016. . https://cdn2.hubspot.net/hubfs/488249/Asset%20PDFs/Benchmark_BI-on-Hadoop_Performance_Q4_2016.pdf (Retrieved: 10.05.2017).

[5].Awais Mehmood 2016 Sixth International Conference on Innovative Computing Technology (INTECH). Performance analysis of shared-nothing SQL-on-Hadoop frameworks based on columnar database systems. Dublin. 2016.

[6].Fast Data Hackathon (2015). http://allegro.tech/2015/06/fast-data-hackathon.html (Retrieved: 10.05.2017).

[7].THE YEAR IN SQL ENGINES. ML/DL: [site]. (2017). https://thomaswdinsmore.com/2017/02/01/year-in-sql-engines/ (Retrieved: 10.05.2017).

[8].Scott J. Apache Spark vs. Apache Drill. MAPR: [site]. (2015). https://mapr.com/blog/apache-spark-vs-apache-drill/ (Retrieved: 10.05.2017).

[9].SQL-on-Hadoop: The Paradox of Choice (2016). https://zdatainc.com/2016/12/sql-hadoop-paradox-choice/ (Retrieved: 11.05.2017).

[10]. SQL and Big Data. Ness: [site]. https://www.ness.com/sql-and-big-data/ (Retrieved: 10.05.2017).

[11]. New SQL Benchmarks: Apache Impala (incubating) Uniquely Delivers Analytic Database Performance. Cloudera Engineering Blog. https://blog.cloudera.com/blog/2016/02/new-sql-benchmarks-apache-impala-incubating-2-3-uniquely-delivers-analytic-database-performance/ (Retrieved: 10.05.2017).

[12]. Soliman M. and Antova L. (2013). Orca: A Modular Query Optimizer Architecture for Big Data. Pivotal. https://content.pivotal.io/white-papers/orca-a-modular-query-optimizer-architecture-for-big-data (Retrieved: 10.05.2017).

[13]. Du D. (2015). Apache Hive Essentials. Birmingham: Packt Publishing.

[14]. Apache Impala (incubating) - Interactive SQL. http://www.cloudera.com/documentation/enterprise/latest/topics/impala.html (Retrieved: 11.05.2017).

[15]. PRESTO DOCUMENTATION (2017). https://prestodb.io/docs/current/ (Retrieved: 16.05.2017).

[16]. Spark Programming Guide . http://spark.apache.org/docs/latest/programming-guide.html#overview (Retrieved: 10.04.2017).

[17]. Holden K. (2015). Learning Spark. Sebastopol: O'Reilly Media.

[18]. Documentation . Apache Drill: [site]. (2017). https://drill.apache.org/docs/ (Retrieved: 14.05.2017).

[19]. Apache HAWQ (incubating) Documentation. Apache HAWQ: [site]. (2017). http://hawq.incubator.apache.org/docs/userguide/2.1.0.0-incubating/overview/HAWQOverview.html (Retrieved: 14.05.2017).

[20]. Eltarenko, E.A. (1995). Estimation and selection of solutions by many criteria. Moscow, Mephi, pp. 111.

[21]. TPC-H. http://www.tpc.org/tpch/default.asp. (Retrieved: 20.05.2017).

[22]. Hive wiki. https://cwiki.apache.org/confluence/display/Hive/Home (Retrieved: 15.05.2017).

[23]. Apache Arrow Powering Columnar In-Memory Analytics. Apache Arrow: [site]. https://arrow.apache.org/ (Retrieved: 10.05.2017).