SQL Query Performance on Hadoop: An Analysis Focused on Large Databases of Brazilian Electronic Invoices

Cristiano Cortez da Rocha[1], Márcio Parise Boufleur[1], Leandro da Silva Fornasier[1], Júlio César Narciso[2], Andrea Schwertner Charão[3], Vinícius Maran[4], João Carlos D. Lima[3] and Benhur O. Stein[3]

[1]*Centro de Informática e Automação do Estado de Santa Catarina (CIASC), Florianópolis, SC, Brazil*

[2]*Diretoria de Administração Tributária, Secretaria de Estado da Fazenda de Santa Catarina, Florianópolis, SC, Brazil*

[3]*Departamento de Linguagens e Sistemas de Computação, Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria (UFSM), Santa Maria, RS, Brazil*

[4]*Coordenadoria Acadêmica, Universidade Federal de Santa Maria (UFSM), Cachoeira do Sul, RS, Brazil*

Keywords: Large Database, Query Performance, Data Management, Business-critical Data.

Abstract: Hadoop clusters have established themselves as a foundation for various applications and experiments in the field of high-performance processing of large datasets. In this context, SQL-on-Hadoop emerged as trend that combines the popularity of SQL with the performance of Hadoop. In this work, we analyze the performance of SQL queries on Hadoop, using the Impala engine, comparing it with a RDBMS-based approach. The analysis focuses on a large set of electronic invoice data, representing an important application to support fiscal audit operations. The experiments performed included frequent queries in this context, which were implemented with and without data partitioning in both RDBMS and Impala/Hadoop. The results show speedups from 2.7 to 14x with Impala/Hadoop for the queries considered, on a lower cost hardware/software platform.

1 INTRODUCTION

Large-scale analytical data processing has spread across a variety of industries and areas of interest, guided by low storage costs and the ability to collect large volumes of business-critical data. Providing such data to engineers and analysts has become increasingly mandatory and requires low response times, which represent a key factor in data mining, monitoring, prototyping strategies, simulations, and other common tasks in this field (Melnik et al., 2010).

Consequently, the processing of large data sets requires state-of-the-art solutions of high performance computing. In this context, Hadoop (White, 2012) and its related tools have established themselves as solutions for managing and performing robust analyzes of unstructured data, as well as structured and semi-structured data processing. The processing of SQL queries in particular has gained significant attention, since the data management systems of many institutions depend on or rely on SQL, in addition to the fact that many corporate users are familiar and comfortable with such language (Floratou et al., 2014). With the evolution of the Hadoop ecosystem, solutions have emerged to process SQL queries on Hadoop, seeking to extend its benefits to structured and semi-structured data (Kornacker et al., 2015).

For government institutions, the processing power and storage savings made possible by the Hadoop ecosystem offer new options for IT infrastructure in optimization and application development. In addition, governments expect such applications to be able to improve services to citizens, as well as addressing major government challenges, such as the economy, revenue collection, job creation, health and natural disasters (Kim et al., 2014).

Tax evasion is mostly performed by the taxpayers to reduce tax liability and this illegal action is usually performed to misrepresent the financial facts to government and tax authorities by providing false tax reporting, such as declaring less income, less profit and more or exaggerated costs (Rad and Shahbahrami, 2016).

According to (Allingham and Sandmo, 1972), the probability of practicing tax evasion is directly related to the probability of its detection, so governments are constantly seeking more efficient and effective ways to deal with this problem. Another challenge faced

Cortez da Rocha, C., Parise Boufleur, M., da Silva Fornasier, L., César Narciso, J., Schwertner Charão, A., Maran, V., D. Lima, J. and O. Stein, B.
SQL Query Performance on Hadoop: An Analysis Focused on Large Databases of Brazilian Electronic Invoices.
DOI: 10.5220/0006690400290037
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 29-37
ISBN: 978-989-758-298-1

by the tax authorities is the turnover of companies. According to data from the Brazilian Institute of Geography and Statistics (in portuguese, *IBGE*) (IBGE, 2012), the close rate of companies in 2 years of life is about 25%. This way, one in four companies closes before two years of existence. In many cases, the downgrade of the business register is fraudulent, leaving behind an uncollected tax liability.

Since 2007, with the institution of the Public System of Digital Bookkeeping (in portuguese, *SPED*), documents and fiscal books, previously filed manually and printed, are migrating to electronic documents, enabling: **(i)** More speed in the identification of tax offenses; **(ii)** Faster access to information; and **(iii)** Greater effectiveness in the supervision of the operations with the data crossing and electronic audit (RFB, 2010). The most commonly used electronic tax document is the electronic invoice, which documents the sales operations.

In this context, Big Data technologies can provide a quick and efficient selection of companies with fiscal irregularities, since the work of a tax auditor involves the analysis of a large amount of fiscal data and reports. Thus, it is possible to provide a rapid response to the Brazilian tax reality (Paula et al., 2016; Abrantes and Ferraz, 2016).

In this paper, in general, we try to feed the context in question with new evidence, considering both technological and business aspects. For this, a comparative analysis of two approaches was performed for the processing of large volumes of fiscal data, more specifically electronic invoices, through SQL queries executed on RDBMS and on Hadoop, using Impala (Kornacker et al., 2015) as a processing engine parallel configuration.

The rest of this paper is organized as follows: Section 2 presents the relevant concepts about the Hadoop ecosystem and the application under consideration. The related works are discussed in Section 3. The characterization of the application in more detail is presented in Section 4. In Section 5, experimental studies are reported and, finally, in Section 6, the final considerations and perspectives of future works are presented.

2 CONCEPTUAL FOUNDATION

2.1 Apache Hadoop

Apache Hadoop, is an open source framework that supports data-intensive distributed and fault-tolerant applications (White, 2012). Its aim is to make available a framework for data and processing abstractions in order to facilitate queries of large, dynamic, and rapidly growing datasets. The main modules that made up the Hadoop framework are as follows:

- **Hadoop Distributed File System (HDFS):** originally it was the Google File System. This module is a distributed file system used as distributed storage for the data; furthermore, it provides an access to the data with high throughput (Shvachko et al., 2010).
- **Hadoop YARN (MRv2):** this module is responsible for the job scheduling and for managing the cluster resources (Vavilapalli et al., 2013).
- **Hadoop MapReduce:** originally Google's MapReduce (Dean and Ghemawat, 2008), this module is a system, based on YARN, for the parallel processing of the data.

One of the main aspects that characterize Hadoop is that the HDFS has a high fault-tolerance to the hardware failure. Indeed, it is able to automatically handle and resolve these events. Therefore, it enables the deploy of lower cost clusters since it can use commodity hardware. Furthermore, HDFS is able, by the interaction among the nodes belonging to the cluster, to manage the data, for instance, to rebalance them.

The processing of the data stored on the HDFS is performed by the MapReduce framework. The MapReduce framework was designed to address the challenges of large-scale computing in the context of long-running batch jobs and allows splitting on the nodes belonging to the cluster the tasks that have to be completed.

2.2 SQL Language in Hadoop Ecosystem

Hadoop has emerged as a solution for distributed processing of large, unstructured datasets. More recently, the ecosystem of solutions around Hadoop has expanded with the emergence of solutions for structured data processing and SQL queries. Usually, such solutions translate a given SQL query into several MapReduce jobs.

Each job applies a different set of operators to different sets of data. The fast response for queries enables interactive exploration and fine-tuning of analytic queries, rather than long batch jobs traditionally associated with SQL-on-Hadoop technologies. However, performing interactive data analysis at scale demands a high degree of parallelism.

In (Chen et al., 2014), the performance of five solutions for executing SQL queries in Hadoop were analyzed. The experiments employ queries derived

from the TPC-DS benchmark and indicate that pioneering solutions, such as Hive, may not perform satisfactorily for many applications. In addition, the results encourage further research on this topic.

Impala (Kornacker et al., 2015) is part of a new generation of SQL query engines that integrate with the Hadoop ecosystem. It is a modular solution, which uses a variety of components (Metastore, HDFS, HBase, YARN and Sentry, for example). Impala circumvents MapReduce to directly access the data through a specialized distributed query engine that is very similar to those found in commercial parallel RDBMSs. Therefore, Impala's goal is to combine familiarity with the SQL language and multiuser performance of traditional databases with Hadoop scalability and flexibility. Impala does not have to translate a SQL query into another processing framework like the map/shuffle/reduce operations which Hive and Apache Pig depends today. As a result, Impala does not suffer the latencies that those operations impose.

One of the components that can be coupled to Impala is Apache Parquet (Melnik et al., 2010), which consists of a columnar storage format that is available to any project in the Hadoop ecosystem, regardless of the choice of the framework to process data, the data model or programming language. It consists of an optimized format for large blocks of data (tens, hundreds and thousands of megabytes). According to the performance analysis performed in (Kornacker et al., 2015), Parquet offers the best compression and search performance among storage formats for Hadoop ecosystem (Plain text, RC, Avro and Sequence).

2.3 Brazilian Electronic Invoice

Brazil is a country that has in the past had to deal with major bouts of tax evasion. The reasons for which are many and contain among others stringent labor laws, high interest rates and the existence of hefty taxes. Brazilian business owners are quick to point out that the fiscal burden placed by the government has led to international smuggling and rampant tax evasion.

One aspect of new Brazilians laws passed in the year 2005 are to better enable the Brazilian government to combat tax evasion by requiring a process of digital invoicing regarding the sale of goods or services. In this context, the Brazilian Electronic Invoice (in portuguese, *NF-e*) aims to implement a national electronic tax document model to gradually replace the model still in force, which is done with paper issuing (RFB, 2010). It seeks to simplify processes related to ancillary obligations, enabling real-time monitoring of commercial operations by Brazilian regulatory agencies.

The *NF-e* project has as one of its basic premises the interconnection of information systems, which provides a reduction in evasion, since it automatically documents and discloses the data of all the buying and selling operations. Specifically, for tax administrations, *NF-e* provides benefits such as: increasing the reliability of the invoice, improving the fiscal control process with exchange and sharing of information, among others. To take full advantage of such benefits, however, new challenges mediated by technologies must be addressed. In particular, the wealth of *NF-e* data generates analytical processing demands on a much larger scale than in the previous model.

3 RELATED WORK

According to (Abouzeid et al., 2009) there are two schools of thought on what technology should use for analyzing data in a high-volume data environment. Database providers argue that the strong emphasis on the performance and efficiency of parallel databases makes them suitable for performing this analysis. On the other hand, others argue that MapReduce-based systems are more appropriate because of their superior scalability, fault tolerance, and flexibility to handle unstructured data.

For (Stonebraker et al., 2010), MapReduce style systems has advantages in complex analyzes, in the extraction processes, transformation and loading of data from external sources. Therefore, these systems complement database systems, since they are designed to operate with transactions and not with massive data load.

The vision of complementary systems generates a third school for processing large volumes of data, which associates the expressiveness of the SQL language with the parallel processing power of Hadoop, which is the focus of this work. *SQL-on-Hadoop* systems enable the connection of existing SQL-based business applications with the results of large data pipelines, increasing their value and accelerating the adoption of Hadoop in commercial environments.

There is a variety of works that compare performance among SQL-on-Hadoop systems using benchmark standards (Costea et al., 2016) and even commercial databases (Kornacker et al., 2015). These comparisons highlight the particularities of the systems and even point out low-cost alternatives to implementing SQL queries on large volumes of data.

At the governmental level, the initiatives follow the American normative (Pannu et al., 2016) and begins a movement to report research papers and pro-

cessing analysis for these databases. In (Abrantes and Ferraz, 2016), a literature review is presented about big data technologies applied in the detection of tax evasion. In (Paula et al., 2016) are presented the results obtained in the implementation of the *"deep learning"* model without supervision, to classify Brazilian exporters as the possibility of committing export fraud.

The present work joins other initiatives, shifting the focus to a database marked by the wealth of attributes that make up *NF-e*, which have confidential information that must be processed several times daily. Therefore, the system needs to be adapted to the needs and repeatability of audit procedures.

According to (Earley, 2015), there are four primary benefits to adopt data analysis in auditing procedures: **(i)** Auditors can test a greater number of transactions; **(ii)** Audit quality can be improved by providing more insights in relation to the scenarios; **(iii)** Fraud can be detected more easily, since auditors can use familiar tools and technologies; and auditors can provide services and solve problems that are currently beyond their ability to use external data to support auditing activities.

4 APPLICATION CHARACTERIZATION

The tax audit activity seeks to find inconsistencies between what is practiced by taxpayers and what is established in the tax legislation. In this context, one of the carried out analyzes is to analyze the electronic invoices to verify if the rate associated with a particular product is in accordance with the law. However, there are several specificities and conditions that must be considered to determine taxation correctly.

For companies under normal taxation regime, without considering the companies under the *Brazilian National SIMPLES* regime, there are several scenarios to be analyzed, depending on the type of commercial operation. For internal transactions, that is, commercialization between companies or individuals of *Santa Catarina* (SC) state / Brazil, the ICMS (Tax on Goods Circulation and Services) tax rate depends on the type of product marketed and, therefore, this scenario presents the greatest challenge of that the taxation of marketed items is correct. The other scenarios, with interstate operations, have fixed rates that do not depend on the type of product sold, so they require less effort.

The strategy used to analyze the most complex scenario is to filter the transactions that are of internal operations and then calculate a series of metrics to check the effective rates being practiced (0%, 7%, 12%, 17% and 25%) to indicate which are the most used aliquots per product. With this processed statistic, the tax auditors have subsidies to choose the focus of the investigation.

Tax inspection in *Santa Catarina* state is structured in 15 specialties groups per product segment or economic activity, such as fuels, medicines, supermarkets, among others. Therefore, the most frequently used aliquot analysis consists of a very important consultation for the massive audit in State Secretary of Finance of Santa Catarina (SEF-SC), impacting the work of at least one tax auditor from each expert group. In addition, this analysis is carried out several times, since it is a consolidation of the behavior of the goods movements by the taxpayers and, therefore, their performance is fundamental for the investigative process of the auditors.

In a more detailed way, the analytical scenario found in the SEF-SC consists in consolidating the data of electronic invoices in two visions: **(i)** identification; and **(ii)** item. The identification of *NF-e* consists of general and summarized information of *NF-e*. The most important fields are: identification of the issuer (identification number (CPF or CNPJ), name and address), identification of the consignee, information about the freight (delivery address, quantity, weight, value and license plate) and tax information of ICMS, total value of ICMS, IPI value, total value of the note, total value of products, insurance value, discounts and other ancillary expenses). On the other hand, the item consists of the finest level of detail, containing information of each item of the NF-e, ie of each product and/or service marketed, such as: product code, product description, quantity, price unit, ICMS calculation basis and ICMS and IPI rate.

It is worth mentioning that the fiscal audit comprises a investigative work and sometimes it is an instrument in the fight against tax crimes. Therefore, there are secrets regarding certain details of the strategy of the fiscal operations, some details, steps and data considered that can not be revealed.

5 EXPERIMENTS AND RESULTS

The experiments carried out focused on queries that were relevant to auditors and were frequently executed in a corporate RDBMS, whose performance was limiting in the context in question. Such queries were chosen within the limits of secrecy, with substitution of terms when necessary, but without changes that impacts on performance. Queries were performed after data preparation in two execution environments with

time measurements to allow comparison between RDBMS and Impala/Hadoop based solutions.

5.1 Data Preparation

To make a preparation for queries, we used the partitioning of data. Partitioning consists of a technique for physically splitting the data during the loading process, based on values of one or more columns of a table, for the purpose of improving the management, performance, availability, or balancing of charge.

Typically, data partitioning is appropriate for large tables, where reading of all data requires a significant time to be performed. Another criterion that indicates an advantage in partitioning is the existence of columns having a reasonable number of distinct values, at the same time that the data volume is not too small to be able to take advantage of the cost of reading.

Thus, we sought to analyze the *NF-e* data to verify which columns would be candidates to partition the data, considering the size of the tables in question and the need for high performance in the queries. Figure 1 shows the distribution of *NF-e* items throughout the months of the year 2016. It can be seen, for the data considered, that the issuance of electronic invoices occurs in a balanced way among the 12 months. In this way, we chose to partition the data by the criterion of the month of issue of the *NF-e*.

In Table 1, the volumetry of these two origins of the *NF-e* is presented for operations in the SC state carried out in 2016.

Table 1: Volumetry of *NF-e* data sources in RDBMS and Hadoop.

Source	NF-e Identification	NF-e Item
Columns	69	26
Records	182,584,721	1,153,841,591
Volume - RDBMS	47.20 GB	112.18 GB
Volume - Hadoop	53.66 GB	128.95 GB

It is observed that the volume and quantity of records are significant. It was observed an increase in the data load in the Hadoop environment, compared to the space used in RDBMS. In addition, SEF-SC's policy is to implement a massive auditing operation, that is, to analyze all goods circulation transactions of all taxpayers, unlike the traditional method of working with samplings. This reinforces the demand for a high-performance solution.

5.2 Experimentation Platform

For the experiments presented in this section, two environments were considered: **(i)** commercial RDBMS; and **(ii)** Hadoop cluster. It is noticeable that there is a difference between these environments. However, this difference is due to organizational and licensing restrictions. Although they are different, we tried to use a configuration with virtual machines that was similar in parallelism.

In addition, it is worth noting that, in financial terms, the RDBMS environment hardware costs around 10x more than the hardware in the Hadoop environment in our experimentation platform. The hardware and software features of such environments are described below.

5.2.1 Commercial RDBMS

The RDBMS execution environment consists of an IBM Power System E850 server, dedicated to experiments. This machine has 20 PowerPC POWER8 3.36 GHz cores and 512 GB of RAM. A feature of the servers in this series is the possibility of partitioning it into smaller machines.

This way, there are 2 partitions used for the RDBMS. Each partition has 8 cores and 48 GB of memory. All of them have the AIX 7.1 TL3 operating system.

In terms of storage, an EMC VNX5500 150TB storage was used, connected by a storage network with 8GB/s Fiber Channel technology. Currently, the RDBMS consumes approximately 80TB.

The RDBMS used is a popular commercial RDBMS, called *"RDBMS-Y"* in this work, due to proprietary restrictions of the licensing agreement.

5.2.2 Hadoop Cluster

Hadoop cluster nodes are hosted in a VMWare VSphere ESX 6.0 virtual environment. For the experiments, a Dell PowerEdge R620 server with 48 Intel Xeon E5-2697 v2 2.70 GHz cores and 256 GB of RAM was designated as the dedicated host. For storage, an EMC VNX7500 storage was used, with a dedicated area of 4 TB, interconnected to the server with two 8 GB/s Fiber Channel interfaces.

Five virtual machines were created for the Hadoop cluster. The first node, named *NameNode*, was configured with 8 cores and 32 GB of RAM. The remaining 4 nodes were configured with 4 cores and 16 GB of RAM. In terms of storage, 800GB was allocated for each node, totaling 4TB. From gross storage, 1.37 TB was made available for HDFS.

In terms of software, we used Impala 2.7.0, Apache Parquet 1.5.0 (Parquet-format 2.1.0) and Apache Hive 1.1.0 with Apache Hadoop 2.6.0. These packages are provided by the Cloudera CDH 5.10.0 distribution. The HDFS replication factor was set to 3, with

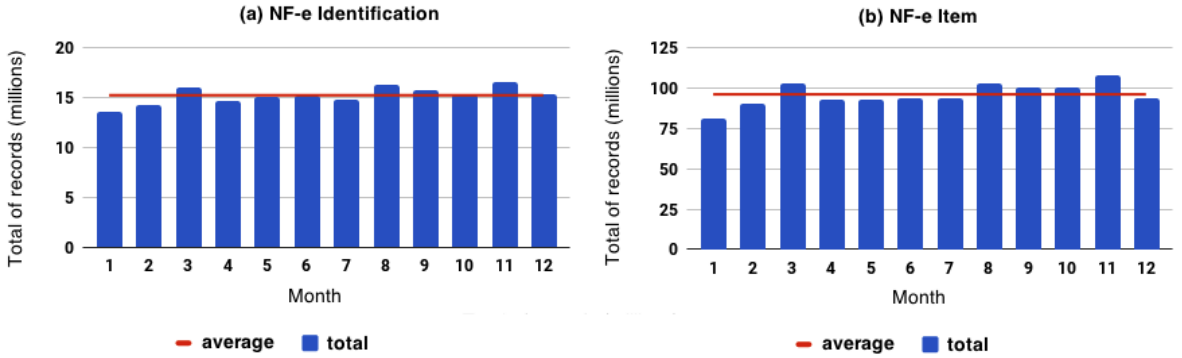

Figure 1: Monthly distribution of identification data (a) and item (b) of *NF-e*.

the Java stack set to 2 GB. The Java stack for Impala was set to 16 GB. Other parameters have been enabled for the default configuration of the distribution.

5.3 Performance Analysis

For all of the queries discussed below, the parallel query execution property was enabled in the *RDBMS-Y*, since Impala consists of a query execution engine with parallelism. In addition, as mentioned in Section 2, the Parquet format presents the best performance for storage formats. Therefore, all queries performed by Impala/Hadoop operate on tables stored in Parquet format. In the following sections we present the queries made in the comparison.

5.3.1 Query 1: Join between *NF-e Item* x *NF-e Identification*

This query aims to consolidate the data of the identification and item tables of the *NF-e* into a single table, called *nfe_complete_item*, using as a key the *nfe_id* column. This is done so that the next steps of analysis for audit can consult only one table with all the data of the *NF-e* universe, without the need to make joins, which are computationally costly. Figure 2 shows the SQL query considered in this experiment.

```
select identification.*, item.*
 from nfe_item item inner join
        nfe_identification ident
 on item.nfe_id=ident.nfe_id;
```

Figure 2: SQL code for *NF-e item* and *NF-e Identification* join.

For the execution times presented in Figure 3, without partitioning the data, we obtained a 2.7x speedup for the Impala in comparison to the *RDBMS-Y*. When considering the partitioning of the data per month, it is verified that with Impala, a 4x acceleration is obtained in relation to the *RDBMS-Y*.

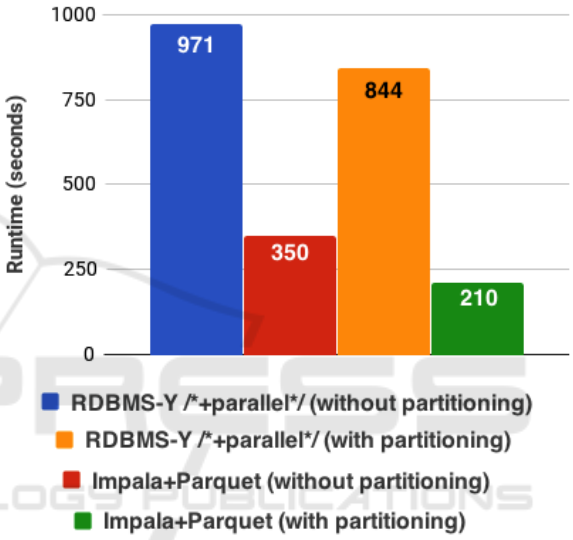

Figure 3: Comparison of query performance regarding the join between *NF-e item* and *NF-e Identification*.

5.3.2 Query 2: Most Used Aliquots

As mentioned in Section 4, the query of statistics on the aliquots associated with electronic invoice products represents an important step in the work of the *SEF-SC* tax auditors. Specifically, this query consists of calculation of a series of arithmetic operations, grouping the data by month and by textual attributes that define the products (called here *column_A* and *column_B*, for confidentiality). In addition, only transactions involving the circulation of goods within SC state defined by a set of *Tax and Operational Codes* (CFOPs) are considered. The query in question, shown in Figure 4, is executed for each different aliquot that may exist, that are, 0%, 7%, 12%, 17% and 25%.

Figure 5 shows the execution times for this query. Without data partitioning, Impala achieves a 5.2x acceleration over the RDBMS. When

```
select column_A,column_B,num_year_month,
  count(*) as number_of_occurrences,
  count(distinct if(icms_aliquota = ${ALIQUOT},
      cnpj_num, null)) as aliq_quantity_${ALIQUOT},
  sum(icms_value) as icms_value,
  sum(icms_calculation_basis_value) as icms_calculation_basis_value,
  max(unitary_value) as unitary_value_max,
  min(unitary_value) as unitary_value_min,
  avg(unitary_value) as unitary_value_avg
from nfe_complete_item
where cod_cfop_item in ('5101','5102','5103','5104',
  '5105','5106','5116','5117','5118','5119','5120','5122',
  '5123','5904','5917')
group by column_A,column_B,num_year_month;
```

Figure 4: SQL code to experiment with grouping and filtering.

considering data partitioning per month (column *num_year_month*), the Impala query displays a 7.18x speedup on the *RDBMS-Y*, with the same partitioning criteria.

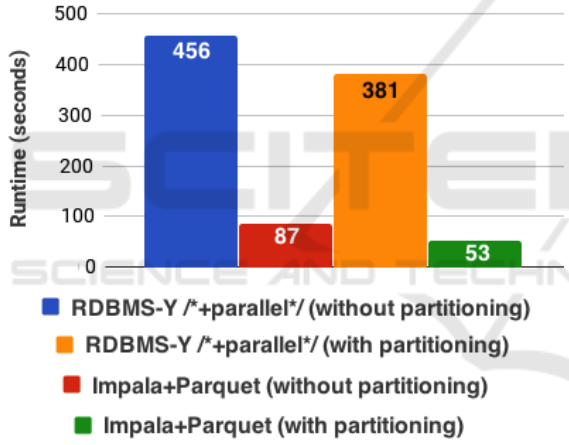

Figure 5: Comparison of query performance to search the most used aliquots.

5.3.3 Query 3: Most Used Descriptions

The tax legislation of the ICMS is extensive and complex, since the economy and commerce are dynamic, with policies and products being created and discontinued at all times. Often product classifications (defined here as textual columns *column_A* and *column_B*) are erroneously done either by register errors or by other reasons.

Thus, in many cases, only the product description becomes the means of knowing if the product classifications are correct, thus allowing a correct identification of the tax rate in which the product fits. Figure 6 presents the SQL query for the most commonly used electronic invoice description, for each combination of *column_A* and *column_B*.

```
select t.column_A, t.column_B,
       t.product_desc
from (
  select item.column_A,
         item.column_B,
         item.product_desc,
  count(*) as n_ocourences_product,
         rank() over (
         partition by item.column_A,
                      item.column_B
         order by count(*) desc
         ) as ranking
  from nfe_complete_item item
  where length(item.column_B) >= 8
  group by item.column_A,column_B,
           item.product_desc
) t
where t.ranking = 1;
```

Figure 6: SQL code for aggregation query.

Figure 7 presents the times of execution of the query. Without considering data partitioning, the SQL query via Impala presented an increase in speed of 11.25x in relation to the *RDBMS-Y*. When considering data partitioning per month, the speedup using the Impala was 14.65x, with the same partitioning criteria.

5.3.4 Discussion

We strongly believe that the modular nature of the Hadoop environment, in which Impala draws on a number of standard components that are shared across the platform, confers some advantages that cannot be replicated in a traditional, monolithic RDBMS as the one that is presented in the experiments.

In particular, the ability to mix file formats and processing frameworks means that a much broader

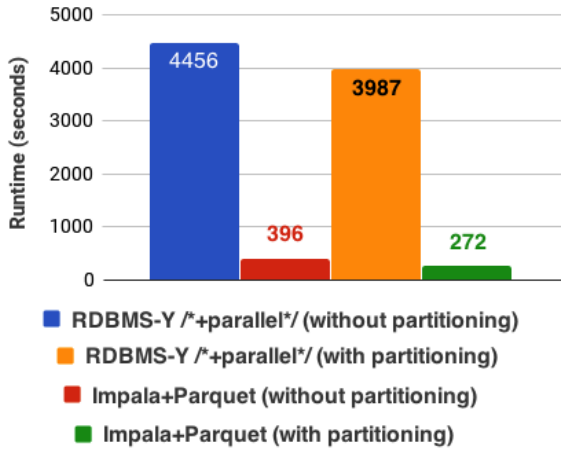

Figure 7: Comparison of query performance to search the most used descriptions.

spectrum of computational tasks can be handled by a single system without the need for data movement, which itself is typically one of the biggest impediments for an organization to do something useful with its data.

It is one of the reasons of the shown difference of performance between Impala and the RDBMS-Y. Since RDBMSs need to read data from disk to memory before start processing, there is a latency that is repeated several times when is working with large datasets. On the other hand, Impala as a MPP solution takes the processing as close possible to the data. Thus, there is less data movement and a lower latency.

6 CONCLUSION

In this work, an open source and recent technology was explored in order to perform SQL queries on Hadoop, applying it in a context in which solutions based on RDBMS are dominant. The comparison joins other research that pointed to benefits of this approach, with the differential being based not on benchmarks, but on a real application, not found in other works that investigate SQL performance on Hadoop.

Despite Hadoop's origin as a batch processing environment, the obtained results reinforce the idea that the public sector can derive great benefits from building large-scale analytical processing technologies compared to traditional RDBMS technologies. In fact, the SQL solution on Hadoop has not only shown better performance, but is also less costly in terms of hardware and software, representing a rational use of public resources.

This solution opens the way to face the challenge of tax evasion in several ways, with the possibility of a massive audit, considering the totality of transactions, companies and individuals involved. In particular, the performance results presented in this paper contributed to the creation of the Fiscal Planning and Monitoring Group in order to subsidize the control groups with information for more efficient fiscal operations.

It is also worth noting that the Impala/Hadoop approach is applicable to data from other Brazilian states, since *NF-e* is a national standard and the codes for operations and benefits are also nationally unified.

The experiments carried out opened the way for further investigations, which cross information from *NF-e* with other tax data. From the technological point of view, it is necessary to accompany new alternatives of SQL-on-Hadoop that perhaps present promising attributes for the application in question.

ACKNOWLEDGEMENTS

The authors are especially grateful to CIASC colleagues, Fábio Eduardo Thomaz, James Rosa, Robson Marcos da Cunha, Ademir João da Rosa, Luiz Carlos Brehmer Junior, Eduardo Sguario dos Reis, Dante Michels de Mattos, Sergio Luiz Borges da Silva, and Nelson Mussak Guanabara for their support and commitment to the project. In addition, the authors thank the colleagues of SEF-SC, Luiz Carlos de Lima Feitoza, Omar Afif Alemsan, and Dayna Maria Bortoluzzi for the suggestions and availability of the necessary infrastructure for the experiments presented. This research was partially supported by UFSM/-FATEC through project number 041250 - 9.07.0025 (100548).

REFERENCES

Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D., Silberschatz, A., and Rasin, A. (2009). HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads. *Proceedings of the Very Large Data Base Endowment Inc.*, 2(1):922–933.

Abrantes, P. C. and Ferraz, F. (2016). Big data applied to tax evasion detection: A systematic review. In *Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 435–440. IEEE.

Allingham, M. G. and Sandmo, A. (1972). Income tax evasion: A theoretical analysis. *Journal of public economics*, 1(3-4):323–338.

Chen, Y. et al. (2014). A study of SQL-on-Hadoop systems. In Zhan, J., Han, R., and Weng, C., editors, *Big Data Benchmarks, Performance Optimization, and Emerging Hardware: 4th and 5th Workshops, BPOE 2014,*

Salt Lake City, USA, March 1, 2014 and Hangzhou, China, September 5, 2014, Revised Selected Papers, pages 154–166. Springer International Publishing.

Costea, A. et al. (2016). VectorH: Taking SQL-on-Hadoop to the next level. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1105–1117, New York, NY, USA. ACM.

Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.

Earley, C. E. (2015). Data analytics in auditing: Opportunities and challenges. *Business Horizons*, 58(5):493–500.

Floratou, A., Minhas, U. F., and Özcan, F. (2014). SQL-on-Hadoop: Full circle back to shared-nothing database architectures. *Proceedings of the Very Large Data Base Endowment Inc.*, 7(12):1295–1306.

IBGE (2012). Demographics of companies [online]. Available: http://biblioteca.ibge.gov.br/visualizacao/livros/liv88028.pdf. [Accessed 20 October 2017].

Kim, G.-H., Trimi, S., and Chung, J.-H. (2014). Big-data applications in the government sector. *Communications of the ACM*, 57(3):78–85.

Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Choi, A., Erickson, J., Grund, M., Hecht, D., Jacobs, M., Joshi, I., Kuff, L., Kumar, D., Leblang, A., Li, N., Robinson, H., Rorke, D., Rus, S., Russell, J., Tsirogiannis, D., Wanderman-milne, S., and Yoder, M. (2015). Impala: A modern, open-source SQL engine for Hadoop. In *Proceedings of the 7h Biennial Conference on Innovative Data Systems Research (CIDR'2015)*.

Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., and Vassilakis, T. (2010). Dremel: Interactive analysis of web-scale datasets. In *Proceedings of the 36th International Conference on Very Large Data Bases*, pages 330–339.

Pannu, M., Gill, B., Tebb, W., and Yang, K. (2016). The impact of big data on government processes. In *Proceedings of the IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–5. IEEE.

Paula, E. L., Ladeira, M., Carvalho, R. N., and Marzagão, T. (2016). Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 954–960. IEEE.

Rad, M. S. and Shahbahrami, A. (2016). Detecting high risk taxpayers using data mining techniques. In *International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–5. IEEE.

RFB (2010). Sped: Public system of digital bookkeeping [online]. Available: http://sped.rfb.gov.br. [Accessed 20 October 2017].

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The hadoop distributed file system. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, pages 1–10, Washington, DC, USA. IEEE Computer Society.

Stonebraker, M., Abadi, D., DeWitt, D. J., Madden, S., Paulson, E., Pavlo, A., and Rasin, A. (2010). MapReduce and parallel DBMSs: friends or foes? *Communications of the ACM*, 53(1):64–71.

Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S., Saha, B., Curino, C., O'Malley, O., Radia, S., Reed, B., and Baldeschwieler, E. (2013). Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, pages 5:1–5:16, New York, NY, USA. ACM.

White, T. (2012). *Hadoop: The definitive guide*. " O'Reilly Media, Inc.".