# Information Management Technologies for Big Data: A Case of Oracle

Kalinka Kaloyanova
Sofia University, FMI
5, J. Bourchier Blvd.
Sofia
Bulgaria
kkaloyanova@fmi.uni-sofia.bg

Tsvetomir Hristov
Sofia University, FMI
5, J. Bourchier Blvd.
Sofia
Bulgaria
injekciona@gmail.com

Ina Naydenova
Technologica
3, Sofiysko pole Str.
Sofia
Bulgaria
naydenova@gmail.com

Zlatinka Kovacheva
Knowledge Oasis Muscat
KOM, P.B. No. 79, Al Rusayl, PC: 124
Sultanate of Oman
zkovacheva@hotmail.com

## ABSTRACT

During the last decade the volume, the rate of accumulation and the diversity of data in general have been steadily increasing, which leads to the rapid development of Big data and technological enhancements associated with it. Many leading companies in the area took the challenge and provided different solutions to manage Big data. New hardware and software technologies are introduced for information management. The paper analyzes the main Big data technologies provided by Oracle as well their implementation in several specific cases.

## CCS CONCEPTS

Å**Information systems** Ÿ **Data management systems**; *Database management system engines*

## KEYWORDS

Database, RDBMS, MapReduce, NoSQL

## 1 INTRODUCTION

The big volume of data becomes a challenge for the capabilities of the traditional data management methods and tools and forces companies and researchers to find new approaches in the case of Big data processing [1].


AWICT '17, November 13ï 14, 2017, Paris, France

ACM ISBN 978-1-4503-5310-6/17/11é $15.00
https://doi.org/10.1145/3231830.3231840

For the purpose of processing big volumes of data, various platforms have been developed in recent years. Many vendors of the classical relational database management systems stated to work in this area, too [15].

In this paper we present technologies implemented by Oracle for Big data processing and compare the ways of their implementations. Several tests are performed and the results are discussed.

## 2 ORACLE TECHNOLOGIES FOR BIG DATA

This section introduces a brief overview of basic Oracle Big data products and technologies that are subject of our study.

**Oracle Big Data Appliance** is a specific system, which delivers Big data solutions in tight integration with the well-known Relational Database Management System (RDBMS) of Oracle. For processing Big data, this framework uses the Hadoop Distributed File System (HDFS), NoSQL decision - Oracle NoSQL DB and Hive [6].

The Oracle NoSQL solution is an component of Oracle Big Data Appliance implemented as a distributed key-value database [6]. It provides relatively good performance characteristics. ñHashing and balancing algorithms are explored to ensure proper data distribution and to optimize load balancingò [6]. The installation and configuration processes are flexible [17].

To work with Big data Oracle uses **Apache Hadoop**, which is based on two main concepts - MapReduce and Hadoop Distributed File System. ñHadoop splits files into large blocks and distributes them amongst the nodes in the clusterò, Hadoop MapReduce ñtransfers packaged code for nodes to process the dataò [3]. Basically, MapReduce refers to two separate tasks. ñThe first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs)ò [16]. The second one uses the map

result and ñcombines those data tuples into a smaller set of tuplesò [4].

**Oracle Loader for Hadoop (OLH)** uses Hadoop MapReduce to create optimal sets of data for efficient loading and analysis of data, stored into Oracle DB. It differs from others by generating data in Oracle specific format for faster loading and to decrease the use of system resources [6,10].

OLH has two working modes - online and offline database mode.

In an Online database mode, the data from the Hadoop infrastructure is loaded into the database using either a Java Database Connectivity (JDBC) or an Oracle Call Interface (OCI) Direct Path interface. The OCI Direct Path interface offers a high performance direct path load of the target table. The JDBC interface uses a conventional path load. The result of the loader job is ñloaded directly into the target table by map or reduce tasks as a part of the Oracle Loader processò [14].

The loading process in Offline mode consists of two parts ï during the first part the map reducer nodes create output files based on the data stored in a Hadoop infrastructure; during the second phase the output files are loaded into an Oracle Database using different types of access drivers. Depending on the type of drivers (Oracle_datapump or Oracle_loader) the phase 1 output files could be in a binary or text format. The phase 1 and phase 2 activities could be executed separately.

**Apache Hive** is an open-source infrastructure tool. It is used to facilitate data summarization, query, and analysis. It could also be used for analysis of data stored in HDFS system. In that case HiveQL (SQL-like language) is used to generate Map Reduce code instead of writing MapReduce programs in Java [6].

In addition to Map Reduce, Oracle Big Data Appliance also provides Oracle R Support and Oracle Big Data Connectors to work with Big data.

**Oracle R Advanced Analytics for Hadoop (ORAAH)** uses a set of R packages to support processing data, which are stored in HDFS and to allow Machine Learning algorithms to be executed [11].

**Oracle SQL Connector for Hadoop Distributed File System** ñuses the Oracle external tables to provide Oracle Database with read access to Hive tables and to delimited text files stored in HDFSò [7]. It also provides access to Data Pump files, which are files in a specific for Oracle XML format. Other file formats also could be processed, using Hive tables [5,14]. The external tables are standard Oracle objects that enable the access to data in external source and identify the location of data outside of a database [10]. The data were accessed via the metadata, which were generated by the time when an external table was created. When users make queries to the external tables, they access the ñdata stored in HDFS and Hive tables as if that data were stored in tables in an Oracle Databaseò [10].

There is a difference in the data extraction and processing for the data stored as text files in HDFS or defined as tables in Hive. The tables in Hive can also be hive internal or hive external.

Using SQL Connectors, it is possible to access data in Oracle directly from Hive tables via SQL queries. External tables are defined in the database, referencing Hive tablesô location in HDFS system. Once configured, the external tables allow access to data in the same manner as querying ordinary tables [8]. It should be noted that the changes made in a Hive table structure are not reflected automatically - it is needed to republish the data or to create a new external table [9].

To work with text files**, Oracle SQL Connectors for text files in HDFS** are used. In that case, an external table is defined based on several configuration properties such as ñthe number of columns, the text delimiter, and optionally, the external table column namesò [10].

It is also possible to use external tables to read data from Oracle NoSQL DB. A specific Publish utility [12] supports the connection between external tables and the Oracle NoSQL Database.

# 3 PROCESSING AND QUERYING DATA: EXPERIMENTAL RESULTS

The comprehensive set of tools presented in the previous section provides the possibility to integrate data stored on the Big data platforms with the Oracle Database. We conducted some experiments to test their performance and to get a clearer idea on how to use these tools for different applications. The performed tests can be divided into two main groups:

- Load data comparison of the time needed to transfer data from the Big data platforms into Oracle DB using the different Oracle Big data Connectors;
- Execution time of different type queries over the data stored on the Big data platforms and Oracle DB.

## 3.1 The Environment

A dataset of 13,932,633 "log" entries is used to evaluate the performance of chosen solutions. It is a part of the heterogeneous array of data to detect human activity available at a public source [2]. The dataset was generated by a mobile application that utilizes the gyroscope sensor of a mobile phone to monitor a number of human activities - stairs down, stairs up, bike, stand, walk, sit.

Below Osample file entry is presented:

0,1424696633909, 1424696631914042029, 0.013748169, -.0006256103500000001,

-0.023376465, a, nexus4, nexus4_1, stand,

where attributes present the index, the record time, the activity time, the location, the user, the mobile model and device and the activity.

Hardware & Software specifications:

- HP Mobile Workstation: Intel i7 Q720 CPU 8 cores @1.6GHz, 8GB RAM
- Windows 7 64bit, fnOracle VM VirtualBox 5.1.14
- Oracle Big Data Lite 4.6.0
- Oracle NoSQL Database (KVLite)
- SQL*Plus and Oracle SQL Developer

Data preparation

As a first step for preparing the data environment for the test experiments, the dataset should be loaded in HDFS, Hive and Oracle NoSQL Database and an access to that data should be provided to the Oracle DB.

Several options for data integration between Oracle DB and the Big data platforms could be used:

Åntegration via Oracle SQL Connector for HDFS

Åntegration via Oracle Loader for Hadoop in Online/Offline Mode

Åntegration via NoSQL Database Publish Utility

## 3.2 Query Performance Tests

After data loading, different experiments could be conducted. Below we present several tests to discuss the performance of various SQL queries (analytical and operative) over the entire data set or over a part of it.

The tests involved an Oracle Database internal table holding the test dataset and several types of external tables to access the dataset from HDFS, Hive and Oracle NoSQL Database. The external tables were created using different Big data Connectors configurations.

**Test1 - Analytical query over full set of data**

The test (Fig. 1) compares the execution time of an analytical query against the Oracle DB internal table and external tables accessing data stored on the Big data platforms. The query search for the number and the type of activities for every model and device.

The results show that Oracle Database internal table access has the best execution time. The execution times of the query over the Hive internal and external tables are identical which is expected given that the only difference between the two tables is the permissions for access of the files in HDFS, and thus there is no gap in the actual performance. For that reason, our next tests use only one of the Hive tables.

The most interesting part of the results here is the performance of a query on Oracle Database external table that accesses data in Oracle NoSQL Database. While the time for loading data from Oracle NoSQL DB into Oracle Database was almost identical to that for loading data from HDFS text and Oracle Data Pump files, in this case performing a query on Oracle NoSQL DB is about 10 times slower.

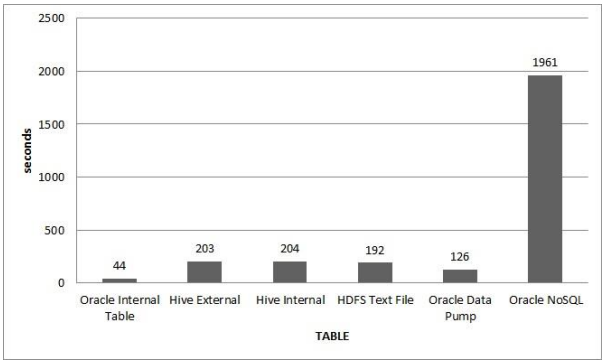


**Figure 1: Analytical query.**

**Test 2 - Querying specific data (one attribute filtering)**

This test compares the times required to access data when we want to filter it based on one attribute and the filtering criteria have a good selectivity (42 rows after the filtering).

The query is searching for a specific value within the first attribute. The results are presented in the figure below.

As evident in the data shown in Fig.2, when the data set is filtered to a small subset, both in a Hive table and an HDFS text file the time is almost equal. Again we have a significant difference between the access time to an HDFS text file and Oracle Data Pump files. Access to data in Oracle NoSQL DB using Oracle Database Publish utilities is confirmed as the slowest of all the methods that were under consideration.

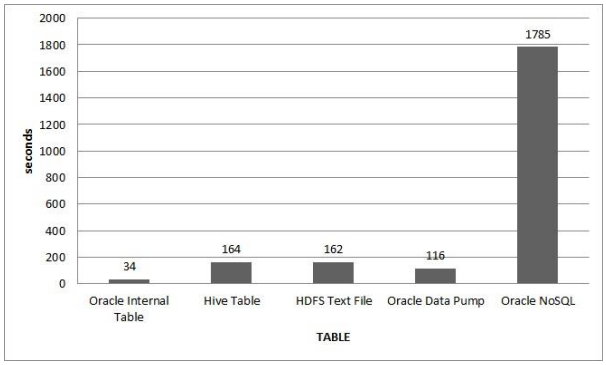


**Figure 2: One attribute query.**

**Test 3 - Join Query including Oracle internal table**

In this case the test set consist of a part of all user data - records of users, whose names are in a specific range. An additional Oracle internal table has been created to hold some personal data for the selected users. The test is based on a query that joins this table to the other test tables.

The results from the test (Fig.3) indicate an overall increase in time due to the join operation, but the correlation between the queries times remains the same as in the above two tests.

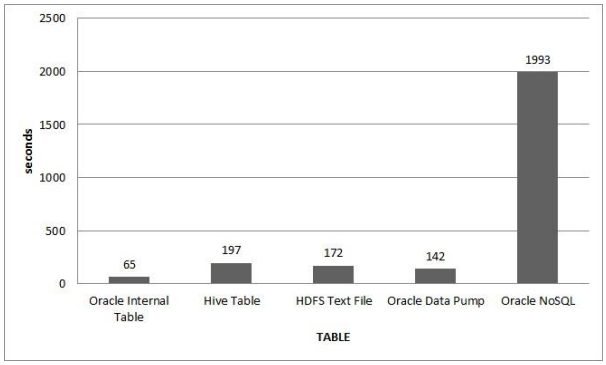


**Figure 3: Query on join tables.**

**Test 4 - Querying Oracle NoSQL Database**

As noted in the previous three tests, the access to data in Oracle NoSQL DB is much slower than the alternatives. In the previous three tests the data has not been queried for specific values in the columns forming their key in Oracle NoSQL DB KVStore.

To rule this out as a possible reason for the poor results, a query from Oracle Database to Oracle NoSQL based on all key attributes will has been executed.

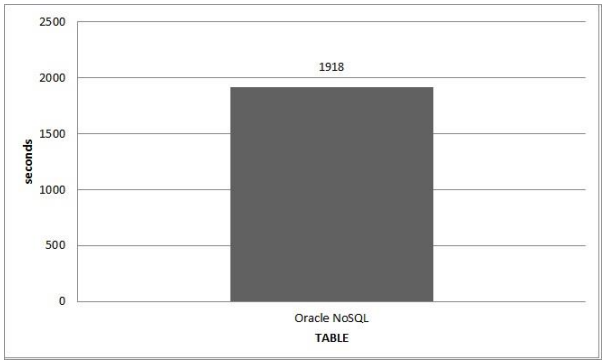


**Figure 4: Oracle NoSQL query.**

Fig. 4 illustrates that the access time in this case is about the same as the time in the first three examples. As a result it can be concluded that even with a query based on all attributes forming the key in KVStore, NoSQL Database Publish Utility does not succeed in making a push-down to the NoSQL database and crawls all data.

## 4 CONCLUSIONS

This paper addresses a set of Big data solutions that are part of the fundamental technologies offered by Oracle for data storage and analysis.

In several examples, a number of opportunities for the deployment of Big data solutions with Hadoop framework and Oracle NoSQL Database have been demonstrated. The main methods for integration of data between Big data platforms and Oracle RDBMS have been presented and a comparison of the performance of the different options has been provided.

The study reveals that Oracle has a large range of solutions to integrate data from Hadoop framework in Oracle Database. Each of the discussed solutions has some positive aspects and drawbacks. Hadoop framework inherently provides solutions for simultaneous processing of large amounts of data that are generally used to data warehouse tasks and build data warehouse of next generation. This partly explains the many different solutions to integrate data from Hadoop framework in an Oracle Database.

Unlike the wide range of solutions for Hadoop framework Oracle does not offer many opportunities for NoSQL Database integration with Oracle Database. The only solution that is available for now is Oracle NoSQL Publish Database Utility. The test results show that it is not optimized for operation over a part of data.

There is no general rule that defines the best technology for the integration of Big data. The users could choose a solution that is most closely to the business needs. In this paper we discuss four data presentations, supported by Oracle and give practical direction for using Oracle decisions.

Our future research plans include exploration of new types of implementations, including data analytics options, based on Oracle R options as well as exploring the potential of Big data for decision support data management [16].

## ACKNOWLEDGMENTS

This work was sponsored by the University of Sofia ñSt. Kliment Ohridskiò SRF under the contract 80-10-119/ 2017.

## REFERENCES

[1] D. Laney. 2012. The Importance of óBig dataô A Definition, Gartner.
[2] Heterogeneity Activity Recognition Data Set https://archive.ics.uci.edu/ml/datasets/Heterogeneity+Activity+Recognition
[3] Apache Hadoop, Wikipedia. https://en.wikipedia.org/wiki/Apache_Hadoop
[4] What is MapReduce https://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/
[5] Yegulalp, S.10 ways to query Hadoop with SQL, http://www.infoworld.com/article/2683729/hadoop/10-ways-to-query-hadoop-with-sql.html
[6] Oracle® Big data Appliance https://docs.oracle.com/cd/E37231_01/doc.20/e36963.pdf
[7] Big data Connectors Userôs guide, https://docs.oracle.com/cd/E37231_01/doc.20/e36961/sqlch.htm
[8] Creating external tables from Hive tables, https://docs.oracle.com/cd/E37231_01/doc.20/e36961/sqlch.htm#BDCUG361
[9] Oracle Database Gateways, http://www.oracle.com/technetwork/database/gateways/gateways-fov-133149.pdf
[10] Oracle Big data Connectors User's Guide https://docs.oracle.com/cd/E53435_01/doc/doc.30/e53067/osch.htm#BDCUG125
[11] Oracle R Advanced Analytics for Hadoop 2.7.0 Release Notes http://www.oracle.com/technetwork/database/database-technologies/bdc/r-advanalytics-for-hadoop/documentation/oraah-2-7-0-release-notes-3386220.pdf
[12] Publish (Oracle NoSQL Database API) - Oracle Help Center https://docs.oracle.com/cd/E35584_01/.../oracle/.../Publish.htm /
[13] Oracle NoSQL Database Concepts Manual http://docs.oracle.com/cd/NOSQL/html/ConceptsManual/Oracle-NoSQLDB-Concepts.pdf
[14] Big Data Connectors User's Guide https://docs.oracle.com/cd/E27101_01/doc.10/e27365/olh.htm
[15] Watson, Hugh J. 2014. "Tutorial: Big Data Analytics: Concepts, Technologies, and Applications," Communications of the Association for Information Systems: Vol. 34 , Article 65, http://aisel.aisnet.org/cais/vol34/iss1/65
[16] Z. Kovacheva , I. Naydenova, K. Kaloyanova, K. Markov, "Big Data Mining: In-Database Oracle Data Mining Over Hadoop", AIP Conference Proceedings 1863, 040003 (2017); http://doi.org/10.1063/1.4992195, 2017
[17] Oracle NoSQL Database, http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html