Hadoop Integrating with Oracle Data Warehouse and Data Mining

Nodar MOMTSELIDZE*
Alex KUKSIN**

Abstract

Various industry verticals are seeing vast amounts of data that are stored on file systems. This article describes an integration of Hadoop cluster HDFS files with Oracle database in order to create Data Warehouse and implement Data Mining solutions. We will provide examples of predefined cluster ETL map/reduce algorithms for loading Oracle database and describe different Hadoop architectures for ETL processing.

Keywords: hadoop, oracle

I.Summary

- Distributed processing of large data sets
- Distributed File System (HDFS), MapReduce
- Hive – A Big Scale Data Warehouse
- ORACLE's Big Data Appliance
- Data Mining
- Analytics with R language
- Using this technologies in "Telco" project

II. Distributed File Systems

Demand to distributed File Systems [1]
1. Storing files of huge size (hundred Tb. Or Petabites);
2. Transmissivity-open/close,read/write distributed files
3. Soft dimensional squeezing (add clusters)
4. Reliability

III. Before MapReduce:

Before Apache Hadoop technology, it was difficult to
- Dimensional squeezing scale data processing
- Managing hundreds or thousands of processors
- Managing parallelization and distribution
- I/O Scheduling
- Status and monitoring
- Fault/crash tolerance

MapReduce technology provides all of these, easily!

IV. Map / Reduce

How does it solve our previously mentioned problems?

Hadoop [2] is a popular open-source map-reduce implementation, which is being used as an alternative to store and process extremely large data sets on commodity hardware.

MapReduce is highly scalable and can be used across many computers.

Many small machines can be used to process jobs that normally could not be processed by a large machine.
Map: input record ⟹ (key, value)
Reduce: (key, {v1, ..., vn}) ⟹ output record

Figure 1: Hadoop a big scale data structure.

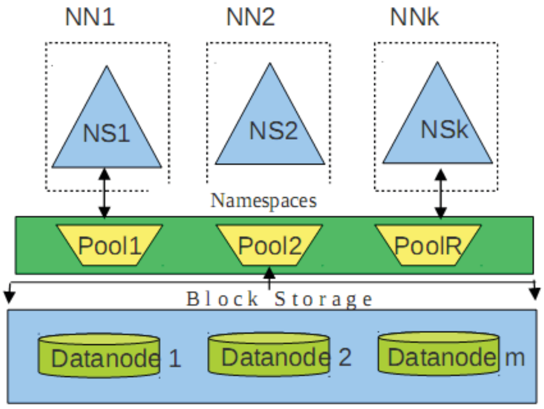
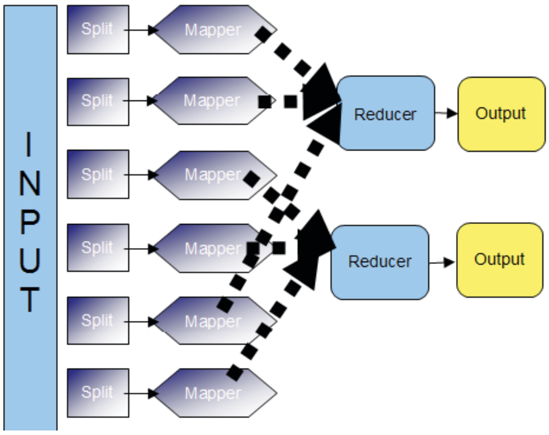


Figure2: Map/Reduce Process.



V. Hive – A Big Scale Data Warehouse

Data Warehousing Tool on Top of Hadoop:

Hive [3], an open-source data warehousing solution built on top of Hadoop.

Hive supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into map-reduce jobs *executed on Hadoop.*

* Prof., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia. E-mail: nmomtselidze@ibsu.edu.ge
** Optima Soft, United States. E-mail: akuksin@gmail.com

In addition, HiveQL supports custom map-reduce scripts to be plugged into queries. The language includes a type system with support for tables containing primitive types, collections like arrays and maps, and nested compositions of the same.

Hive consists of three parts:

- Metastore over Hadoop
- Libraries for (De)Serialization
- Query Engine(HQL)

The Metastore acts as the system catalog for Hive. It stores all the information about the tables, their partitions, the schemas, the columns and their types, the table locations etc.

Hive can take an implementation of the SerDe java interface provided by the user and associate it to a table or partition. As a result, custom data formats can easily be interpreted and queried from.

Each dependent task is only executed if all of its prerequisites have been executed. A map/reduce task first serializes its part of the plan into a plan.xml file. This file is then added to the job cache for the task and instances of ExecMapper and ExecReducers are spawned using Hadoop.

Similar to traditional databases, Hive stores data in tables, where each table consists of a number of rows, and each row consists of a specified number of columns. Each column has an associated type. The type is either a primitive type or a complex type. Currently, the following primitive types are supported:

- Integers – bigint(8 bytes), int(4 bytes), smallint(2 bytes), tinyint(1 byte). All integer types are signed.
- Floating point numbers – float(single precision),
- double(double precision)
- String

Hive also natively supports the following complex types:

- Associative arrays – map<key-type, value-type>
- Lists – list<element-type>
- Structs – struct<file-name: field-type, ... >

Example:

```
CREATE TABLE t1(st string, fl float,
li list<map<string, struct<p1:int, p2:int>>);

SELECT t1.a1 as c1, t2.b1 as c2
FROM t1 JOIN t2 ON (t1.st = t2.st2);
```

HiveQL has extensions to support analysis expressed as map-reduce programs by users and in the programming language of their choice.

```
FROM (
MAP doctext USING ‘python wc_mapper.py’
AS (word, cnt)
FROM docs
CLUSTER BY word
) a
REDUCE word, cnt USING ‘python wc_reduce.py’;
```

You can use java, python and other lang. for map/reduce

VI. Hive – HQL Approach

GROUP BY: as key in Map

WHERE: calculates in Map. For main fields recommended create partitioning. For historical values generally partitioning by time_key.

SUM/AVG as value in Map

SUM/AVG: final value is calculating in Reduce

JOIN: Reduce or Map

HAVING: is filtering in the final step in Reduce

VII. Why HDFS → Oracle [4]

Hadoop provides a massively parallel architecture to process information from huge volumes of unstructured and semi-structured content.

Data have to be analyzed from BI tools usually using relational databases powered by analytical functions and Data Mining options.

The critical data should be accessible between different systems and easy enabled for any analytical needs.

VIII. Different range of requirements

Scheduled bulk data loading from hadoop to the data warehouse

Streaming hot hadoop data from databases

Accessing HDFS data for map/reduce directly from database, BI systems or OLTP applications

Using RDBMS based repository for HADOOP metadata

IX. Native HDFS Oracle Connectors

Oracle Loader for Hadoop

OLH loads data from Hadoop to Oracle Database. It runs as a MapReduce job on Hadoop to partition, sort, and convert the data into an Oracle-ready format, offloading to Hadoop the processing that is typically done using database CPUs. The data is then loaded to the database by the Oracle Loader for Hadoop job (online mode) or written to HDFS as Oracle Data Pump files for load or access later (offline mode) with Oracle Direct Connector for HDFS. Oracle Loader for Hadoop evenly distributes the load across Hadoop reducer tasks, handling skew in input data that could otherwise cause bottlenecks.

Oracle Direct Connector for Hadoop Distributed File System (HDFS)

ODC for HDFS is a connector for high speed access to data on HDFS from Oracle Database. With this connector SQL in the database can be used to directly query data on HDFS. The data can be text files or Oracle Data Pump files generated by Oracle Loader for Hadoop. The connector can also be used to load the data into the database with SQL, if the application requires.

The connectors can be used together or separately.

X. Oracle Loadr for Hadoop

Oracle Loader for Hadoop performs the following preprocessing on the data:

Convert the data into Oracle data types

Partition the data according to the partitioning scheme of the target table, and

Optionally sort the data within each partition

Oracle Loader for Hadoop uses the efficient direct path load mechanism, with minimal additional processing in the database.

When input data is not evenly distributed among partitions of the target table, Oracle Loader for Hadoop evenly distributes the load across Hadoop reducer tasks so that a node does not become a bottleneck because it is loading more data than the others are.

Oracle Loader for Hadoop has online and offline load options. In the online load option, the data is both pre-processed and loaded into the database as part of the Oracle Loader for Hadoop job. Each reduce task makes a connection to Oracle Database, loading into the database in parallel. The database has to be available during the execution of Oracle Loader for Hadoop.

XI. Oracle Direct connectors for HDFS

Oracle Direct Connector for HDFS allows Oracle Database to access files on HDFS via external tables. External tables allow data in files that are outside the database to be queried and, with Oracle Direct Connector for HDFS, these can be files on HDFS. The files can be text files or Oracle Data Pump formatted files created by Oracle Loader for Hadoop. External tables can be queried like any other table, so the data becomes accessible through SQL in Oracle Database. The data can be queried and joined with other tables in the database, and used for in-database analysis.

Text files on HDFS can be directly read by Oracle Direct Connector for HDFS. Data formats other than text files can be pre-processed by Oracle Loader for Hadoop into Oracle Data Pump files so that Oracle Direct Connector for HDFS can read them.

The data can also be loaded into the data base using SQL. Loading into the database is helpful if the data will be accessed frequently, and is required if the data has to be updated by the database.

XII. Oracle Data Integrator Application Adapter for Hadoop

The Oracle Data Integrator Application Adapter for Hadoop enables data integration developers to integrate and transform data easily within Hadoop using Oracle Data Integrator. Employing familiar and easy-to-use tools and preconfigured knowledge modules, the adapter provides the following capabilities:

Loading data into Hadoop from the local file system and HDFS.

Performing validation and transformation of data within Hadoop.

Loading processed data from Hadoop to Oracle Database for further processing and generating reports.

Typical processing in Hadoop includes data validation and transformations that are programmed as MapReduce jobs. Designing and implementing a MapReduce job requires expert programming knowledge. However, using Oracle Data Integrator and the Oracle Data Integrator Application Adapter for Hadoop, it's not needed to write MapReduce jobs. Oracle Data Integrator uses Hive and the Hive Query Language (HiveQL), a SQL-like language for implementing MapReduce jobs. The Oracle Data Integrator graphical user interface enhancing the developer's experience and productivity while enabling them to create Hadoop integrations.

XIII. Oracle R connector for Hadoop

Oracle R Connector for Hadoop is an R package that provides transparent access to Hadoop and to data stored in HDFS.

R Connector for Hadoop provides users of the open-source statistical environment R with the ability to analyze data stored in HDFS, and to scalably run R models against large volumes of data.

R Connector for Hadoop enables users to write mapper and reducer functions in R and execute them on the Hadoop Cluster through R. In addition, users can easily transition R scripts from test environments to production. Hadoop-based R programs can be deployed on a Hadoop cluster without needing to know Hadoop internals, the Hadoop or HDFS command line interfaces, or IT infrastructure. R Connector for Hadoop can optionally be used together with the Oracle Advanced Analytics Option for Oracle Database. The Oracle Advanced Analytics Option enables R users to transparently work with database resident data without having to learn SQL or database concepts but with R computations executing directly in-database.

XIV. Using FUSE

HDFS files are not directly accessible through the normal operating system, but there is a way to access HDFS data using File system in Userspace.

Using one of the FUSE drivers and mounting HDFS on the database instance, even on every instance in case of using a RAC database, HDFS files can be easily accessed using the External Table infrastructure.

External tables present data stored in a file system in a table format and can be used in SQL queries transparently. External tables could be used to access data stored in HDFS (the Hadoop File System) from inside the Oracle database through FUSE and execute all desired operations in parallel directly from the external tables.

Sqoop is a tool designed to transfer data between Hadoop and relational databases. You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.

XV. Apache Sqoop Software

Sqoop automates most of this process, relying on the database to describe the schema for the data to be imported. Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance.

Sqoop reads the table row-by-row into HDFS. The output of this import process is a set of files containing a copy of the imported table. The import process is performed in parallel. By this, the output will be in multiple files. These files may be delimited text files (for example, with commas or tabs separating each field), or binary Avro or SequenceFiles containing serialized record data.

Sqoop's export process will read a set of delimited text files from HDFS in parallel, parse them into records, and insert them as new rows in a target database table, for consumption by external applications or users.

XVI. Quest Data connector for Oracle and Hadoop

Quest Data Connector for Oracle and Hadoop is an optional plugin to Sqoop.

Quest Data Connector for Oracle and Hadoop accepts responsibility for the following Sqoop Job types:

] Import jobs that are Non-Incremental.

] Export jobs

] Quest Data Connector for Oracle and Hadoop does not accept responsibility for other Sqoop job types. For example, Quest Data Connector for Oracle and Hadoop does not accept eval jobs etc.

Quest Data Connector for Oracle and Hadoop accepts responsibility for those Sqoop Jobs with the following attributes:

] Oracle-relatedl Table-Based — Jobs where the table argument is used and the specified object is a table.

XVII. Data Mining

Generally, data mining (sometimes called data or knowledge discovery) is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

XVIII. Analytics with R Language

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy, and
- a well-developed, simple and effective programming language (called `S') which includes conditionals, loops, user defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)

XIX. Telco Use Case

Is there influence of Service Quality (SQ) (voice, SMS, MMS, MBB, …) on churn? For which segments of subscribers, for which services, on which SQ problem levels? What is the high-level estimation of potential of protected revenue through SQ optimization?

Our solution:

- Drill down to identify areas for SQ optimization with high potential:
    - o For what subscribers? For what Services?
    - o What SQ problems from what current level to what target level should be improved? Where?
    - o We should proceed to consider SQ input to business cases only for those network optimization initiatives, which influence the areas for SQ optimization with high potential identified at this step (***Figure3****)*.
- Estimate churn reduction effect caused by expected SQ change while making decisions on concrete network optimizations which influence the identified areas for SQ optimization with high potential network optimizations: modernization / capacity expansion / capacity distribution policies change
- Evaluate churn reduction effect caused by actual SQ change after performing concrete network optimizations
- Estimation of the analysis outcomes is illustrated on the following diagram:

References

[1]http://lib.custis.ru/Apache_Hadoop_(Владимир Климонтович_на_ADD-2010)

[2] Apache Hadoop. Available at http://wiki.apache.org/hadoop

[3] Apache Hive. Available at http://hive.apache.org/

[4] http://www.oracle.com/us/corporate/press/1453721

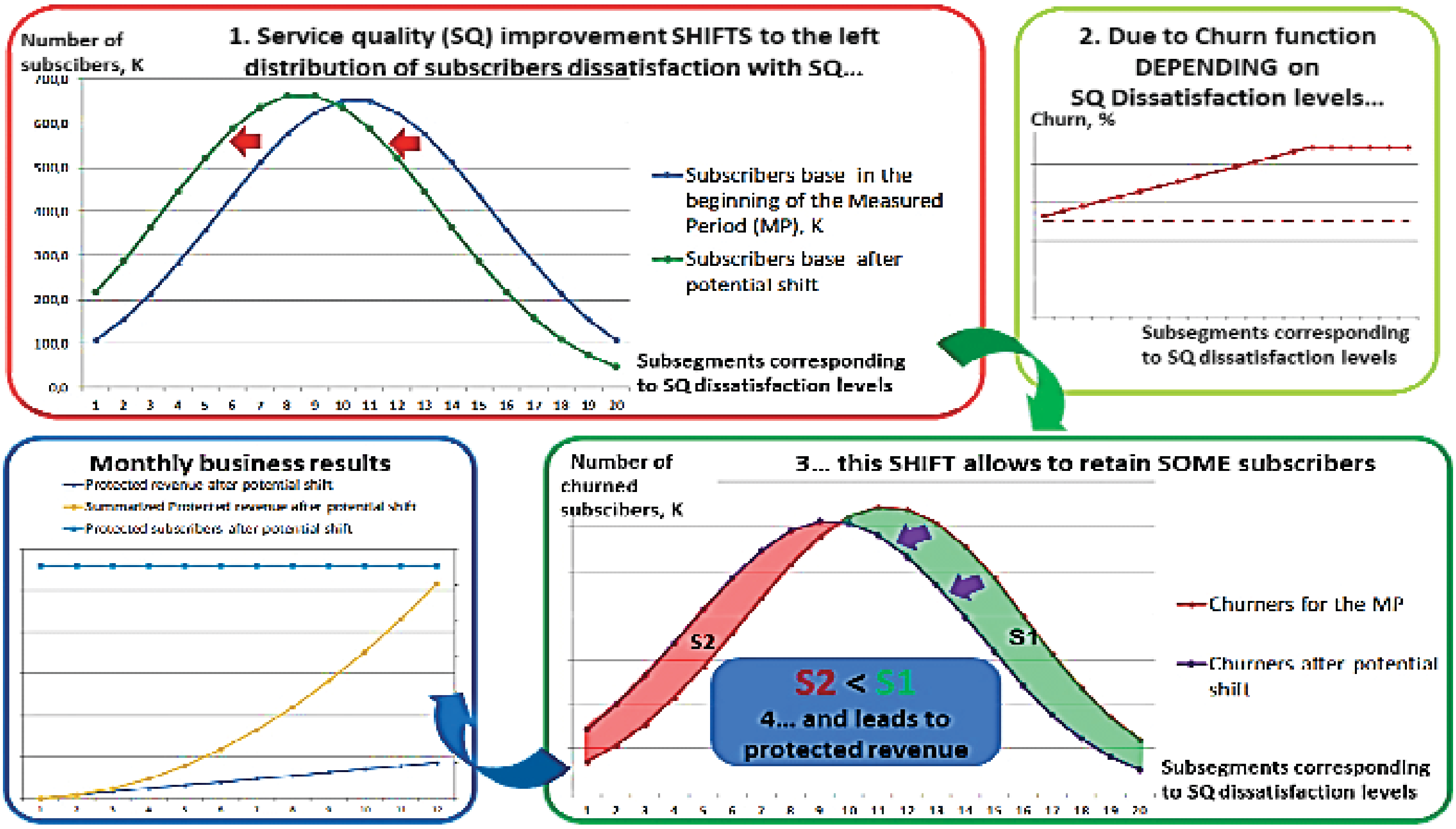


Figure3: *Service quality diagrams.*