ORACLE
BIG DATA

An Oracle White Paper
June 2012

High Performance Connectors for Load and Access of Data from Hadoop to Oracle Database

ORACLE

Executive Overview

Oracle Loader for Hadoop and Oracle Direct Connector for Hadoop Distributed File System (HDFS) enable high performance load and access of data from a Hadoop platform to Oracle Database. These efficient connectors, optimized for Hadoop and Oracle Database, make it easy to acquire and organize unstructured data on Hadoop and bring it together with data in Oracle Database, so that applications can analyze all data in the enterprise.

A load speed of 12 TB an hour from Oracle Big Data Appliance to Oracle Exadata was achieved in our performance testing.

The fastest load option with Oracle's connectors is 5 to 20 times faster than third party products that were measured.

This paper will describe the use cases for Oracle Loader for Hadoop and Oracle Direct Connector for HDFS, and how, using these technologies, it was possible to load up to 12TB/hour from Oracle Big Data Appliance to Oracle Exadata.

Introduction

The potential treasure trove of less structured data such as weblogs, social media, email, sensors, and location data can provide a wealth of useful information for business applications. Hadoop provides a massively parallel architecture to distill desired information from huge volumes of unstructured and semi-structured content. Frequently, this data needs to be analyzed with existing data in relational databases, the platform for most commercial applications. Relational databases contain critical data that enables users to derive greater insights from the less structured data that is processed and stored on Hadoop clusters.

A set of technologies and utilities referred to as "connectors" are necessary to make the data on Hadoop available to the database for analysis. Oracle Loader for Hadoop and Oracle

Direct Connector for HDFS are two high performance connectors to load and access large volumes of data on Hadoop. These connectors support a range of requirements from customer use cases:

- Periodic bulk load for inclusion in data warehouses. Example:
    - A bank plans a daily upload of 100 million records when the database is not busy.
- Streaming in data from Hadoop for immediate analysis in the database. Examples:
    - 24x7 monitoring and analysis of log feeds: This load is regular and continuous.
    - Monitoring of sensor data, triggering loading into the database when an error is detected. This load occurs with irregular frequency.
- Accessing and analyzing data in place on HDFS by database applications.
- Minimize intermediate staging while making data available for analysis in the database for simpler storage management.

Developed and supported by Oracle, these connectors take advantage of our knowledge of Oracle internals to enable robust and high performance load and access of data on a Hadoop cluster.

Oracle Loader for Hadoop

Oracle Loader for Hadoop 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)

Oracle Direct Connector 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.

In the following sections of this paper, we will look at the performance and some technical features of these two connectors.

Load and Access of Data from Hadoop to Oracle Database

The key technical benefits of these connectors are:

- Fast performance
- Reduced database server CPU usage
- Flexibility to automatically handle any skew in the data
- Automatic, optimal, load balancing to maximize performance

Performance goals of these connectors are

1. Load the data into a table as fast as possible.
2. Minimize impact of the load on deployed database systems, so that they can meet their performance SLAs without interruption.

These two connectors work together to achieve these performance goals. In addition, they provide load options to support a range of use cases.

Technical Features

These two connectors work with any Hadoop deployment based on Apache Hadoop 0.20.2 such as Cloudera's Distribution Including Apache Hadoop (CDH3) and any Oracle Database (versions 10.2.0.5 and 11.2.0.2 or greater). However, they are optimized to work with Oracle Big Data Appliance and Oracle Exadata Database Machine. Both connectors load scalar data into Oracle Database.

Oracle Loader for Hadoop

Oracle Loader for Hadoop uses the parallel processing framework of Hadoop to perform the data pre-processing operations that are typically done on the database server as part of the load process. Offloading this step to Hadoop reduces CPU time used on the database server.

Oracle Loader for Hadoop is a MapReduce program and is submitted as a Hadoop job to a cluster, typically at the end of other MapReduce processing of the data. It can read data using Oracle supported input formats (delimited text, Hive, and Avro input formats) and any user-defined input format, thus supporting a wide range of input formats.

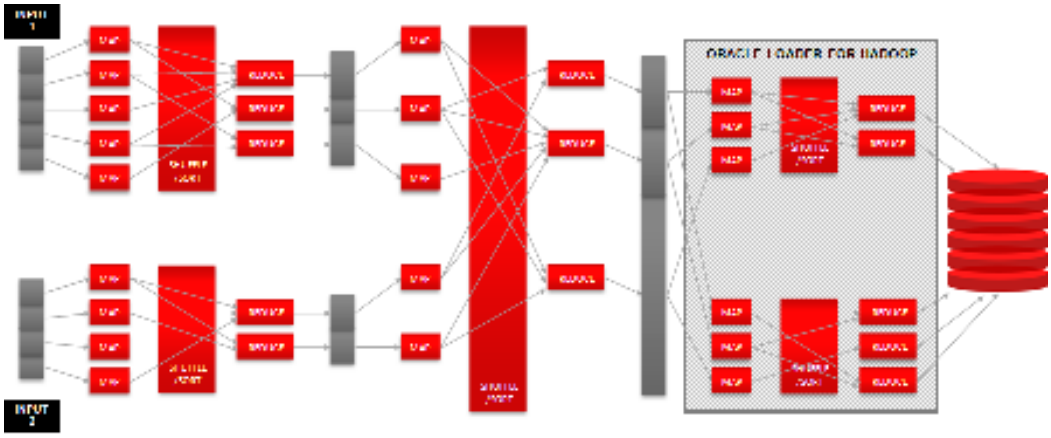

Figure 1. Oracle Loader for Hadoop is a MapReduce job that runs on Hadoop.

Target table is the table in the database into which data is loaded. This is defined by the user and can have any table definition and any partitioning scheme (except reference or virtual column based partitioning). All scalar data types are supported. The table can contain columns with unsupported data types, but these columns must be nullable or otherwise set to a value.

Oracle Loader for Hadoop performs the following pre-processing 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.

Load Options

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.

In the offline load option, each reduce task creates an Oracle Data Pump file that is written to HDFS. The database can be offline during this step. The data is later loaded into the database using Oracle Direct Connector for HDFS.

Oracle Direct Connector 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. So Oracle Direct Connector for HDFS, when used with Oracle Loader for Hadoop, can support a wide range of input data formats.

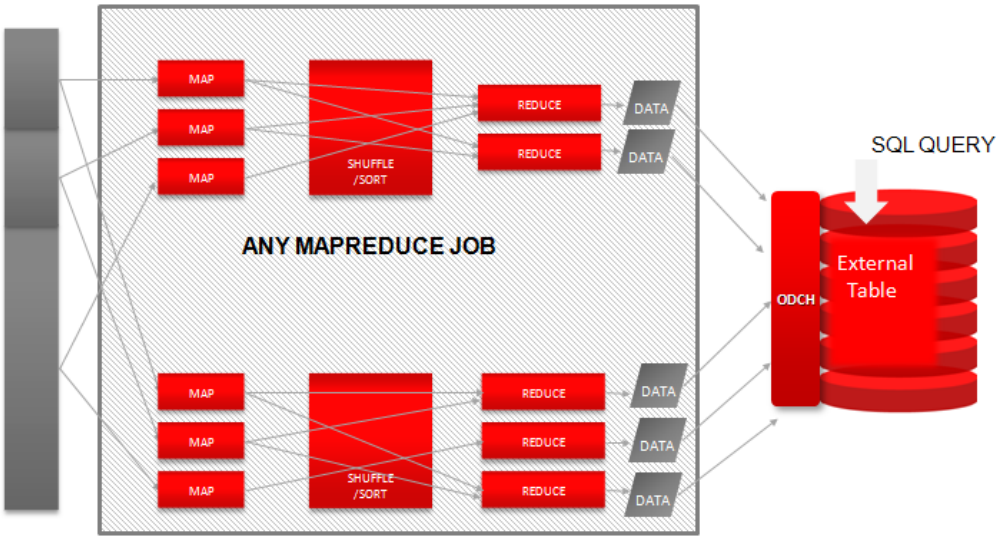

Figure 2. Oracle Direct Connector for HDFS

Oracle Direct Connector for HDFS (referred to as ODCH in Figure 2) uses a Hadoop client that runs on the machine that deploys the database. The connector first executes a *publish* step, generating files that point to URIs of the data files on HDFS. After this step the data is available for query through external tables. Data files are read in parallel by the database.

Loading Data with Oracle Direct Connector for HDFS

The data can also be loaded into the database 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.

Load and Access Options of the Connectors

The connectors can be used together or separately. Table 1 summarizes the load and access options using Oracle Loader for Hadoop and Oracle Direct Connector for HDFS.

TABLE 1. LOAD AND ACCESS OPTIONS

LOAD OPTION	DESCRIPTION	CHARACTERISTICS
Offline load or data access of Oracle Data Pump files	Oracle Loader for Hadoop creates Oracle Data Pump files. These are accessed or loaded from HDFS with Oracle Direct Connector for HDFS	Fastest load option Reduced CPU time used on database server Loads can be scheduled during off-peak times Data can reside on HDFS and be queried with SQL in the database (without loading) User-defined input formats allow a wide range of input data formats to be loaded
Load or access of text files	Database reads the text files directly from HDFS with Oracle Direct Connector for HDFS	Fastest load option, but uses more CPU time on the database server than the above option Useful when there is a lack of time or Hadoop resources to transform the data before loading Text data can reside on HDFS and be queried with SQL in the database (without loading) Input data is restricted to text files (other types of data need to be first transformed into Oracle Data Pump files using Oracle Loader for Hadoop as in the above option)
Online load with OCI direct path	Oracle Loader for Hadoop creates an OCI direct path connection from each reduce task	Fastest online load option Lowest CPU time used on the database server Data is loaded as it is pre-processed Eliminates the work needed to manage files on disk during intermediate staging steps

Online load with JDBC	Oracle Loader for Hadoop creates a JDBC connection from each map or reduce task	Useful for prototype development and testing, small loads

Performance

The performance tests measured load times and database CPU used for the different load options. They also compared Oracle's connectors with third party products.

Configuration Overview

All the tests used Oracle Big Data Appliance and Oracle Exadata, connected by 40 GB/second InfiniBand fabric. Oracle Big Data Appliance has been designed and optimized for Hadoop from the ground up. It has 18 servers, each with 2 CPUs. Each CPU has 6 cores for a total of 216 cores. Each server is configured with 48 GB of memory for a total of 864 GB of memory. The total storage capacity is 500 TB. Hadoop can be configured to have up to 330 Mapper tasks on this hardware. Oracle Exadata is an engineered system for Oracle Database and the X2-8 model used in the tests comes with two 8-socket database servers, InfiniBand switches, and various technologies to support high throughput.

Performance Testing Goals

(1) Measure the performance of the different load options.
(2) Identify and describe the performance characteristics of the different load options to understand the applicability to various use cases.
(3) Measure the impact of the load on the database server by measuring the CPU time for the different load options.

Performance Tests

Data sizes ranging from 100GB (555 million rows) to 4 TB (22 billion rows) were used. The target table had 7 columns of different data types (NUMBER, VARCHAR2, DATE and TIMESTAMP) and was hash partitioned with a total of 128 partitions. Data load times were measured and load speeds (amount of data loaded per unit of time) were calculated. The load options tested were:

- Offline load option using Oracle Data Pump files
- Offline load option using text files
- Online load option with OCI direct path
- Online load option with JDBC

Note on Measuring Load Time

The *database load time* is the time taken to load data from Hadoop reducer nodes (for online load options) or from HDFS (for offline load options) into an Oracle Database table. An alternative measure is the *end-to-end load time*, which is the data pre-processing time on Hadoop plus the database load time. The *database load time* is an accurate representation of the data transfer rate from the Hadoop system to the database. The *end-to-end load time* factors in the time taken to perform data pre-processing on Hadoop to reduce the CPU time used on the database server.

- In the online load options the data pre-processing and the load step both occur as part of the same job, so the end-to-end load time is also pertinent.
- In the case of offline load options, the data pre-processing step is a separate job, and could be done at a different time.
- The end-to-end load time is higher than the database load time, and the trade off is that data pre-processed on Hadoop uses less CPU time on the database server.

The details of how each load option works are described below, to better understand the performance results. The data processing step is done by Oracle Loader for Hadoop. The load step is done either by Oracle Loader for Hadoop (for online load options) or Oracle Direct Connector for HDFS (for offline load options).

Load of text files

Text files are loaded (or accessed) using Oracle Direct Connector for HDFS. A SQL statement is used to load the data from an external table to the target table.

Offline Load Option with Oracle Data Pump files

Oracle Data Pump files are created using Oracle Loader for Hadoop, and then loaded using Oracle Direct Connector for HDFS. A SQL statement is used to load the data from an external table to the target table.

Online Direct Path Load Option

This load option uses the OCI direct path load interface, bypassing SQL. Data is pre-processed on Hadoop, and then data from each partition is loaded using the direct path load interface to the database. Each reducer task makes a connection to the database and loads data directly into the appropriate partition.

Online JDBC Load Option

Like other open source and third party products that are based on JDBC, this load option uses SQL insert statements to load the data. The data is pre-processed on Hadoop, and then data from each partition is loaded using SQL insert statements. Each reducer task makes a JDBC connection to the database and loads data into the appropriate partition.

Test 1: Load Times

Tables 2 shows the load times for the offline load option using text and Oracle Data Pump files. Table 3 shows the load times for the online direct path load option.

Offline load option

Test Data Size: 4 TB

Test Configuration: Oracle Big Data Appliance and two node RAC database on Oracle Exadata X2-8

Table 2 shows the *database load time* for 4 TB, with a time of 1124 seconds. This is the fastest load option with a **load rate of over 12 TB/hour**. Data is loaded from HDFS on Oracle Big Data Appliance into a full rack Oracle Exadata using Oracle Direct Connector for HDFS. The load speeds for text files and Oracle Data Pump files are similar, but loading Oracle Data Pump files uses less CPU time (see Test 2).

TABLE 2. LOAD RESULTS FOR OFFLINE LOAD OPTION (4 TB)

LOAD OPTION	ELAPSED TIME (SEC)	LOAD RATE (GB/SEC)	LOAD RATE (TB/HOUR)
Offline load of Oracle Data Pump files or text files	1124	3.6	**12**

Online load option

Test Data Size: 2 TB

Test Configuration: Oracle Big Data Appliance and a single database instance on Oracle Exadata X2-8

Table 3 shows the *database load time* and *end-to-end load time* for the online direct path load option. As discussed above, both are relevant measures for online load options since the data pre-processing step is part of the data loading job. *End-to-end load time* includes the pre-processing time on Hadoop and the database load time.

In this test data is loaded from Oracle Big Data Appliance into Oracle Exadata running one database instance. Since much of the data pre-processing is offloaded to Hadoop, a single database instance was sufficient for testing. Other available database instances can be fully used for applications running on the database in a deployment scenario.

TABLE 3. LOAD RESULTS FOR ONLINE LOAD OPTION WITH DIRECT PATH LOAD (2 TB)

LOAD OPTION	ELAPSED TIME (SEC)	LOAD RATE (GB/SEC)	LOAD RATE (TB/HOUR)
Online direct path load	1239	1.6	**5.8**
Database load			
End-to-end load	3536	0.6	**2**

The data load rate of 2 TB might appear low compared to the other load rates, but note that this is the *end-to-end load* rate, so it includes the time spent on Hadoop to pre-process the data before the load.

Test 2: Measuring the Impact on the Database

Test Configuration: Oracle Big Data Appliance and a single database instance on Oracle Exadata X2-8

One of the benefits of these connectors is that they offload the data pre-processing step to Hadoop to reduce CPU time used on the database server. This allows other applications running on the database to continue using database CPU even during the load of large volumes of data that are typical in big data use cases.

The CPU time used on the database server varies by load option. This is compared in Figure 3. 'CPU seconds used per GB' (on the Y axis) is the number of seconds the database server CPU is used while loading divided by data size in GB. The number of CPU seconds used while loading is measured by the CPU BUSY_TIME parameter in the Automatic Workload Repository (AWR) report. This number includes CPU seconds used for database processes and operating system processes.

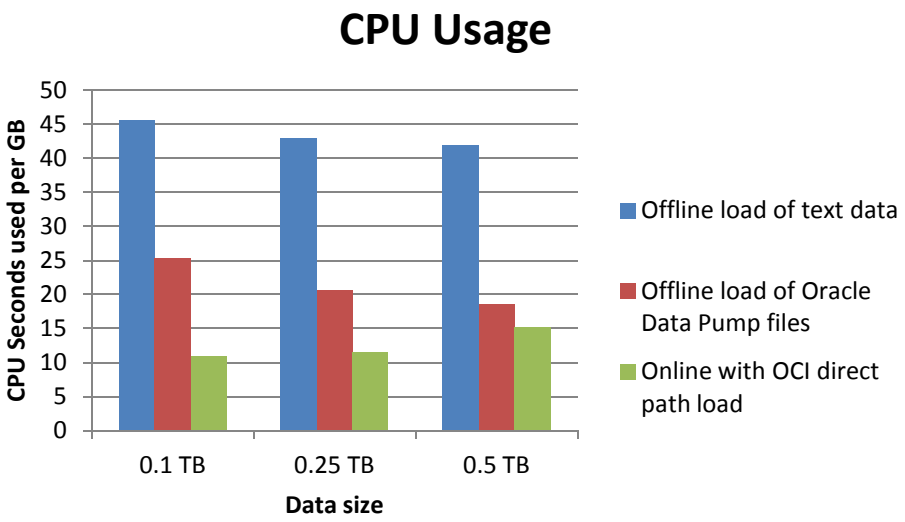

Figure 3. CPU usage on the database server

Loading Oracle Data Pump files uses 50% less CPU time compared to loading text data. Online direct path load uses only 20-30% of CPU time compared to loading text data. The online direct path load option is more efficient than the offline load option with Oracle Data Pump files since it bypasses the SQL of the database server. These two load options use less CPU time compared to loading text data because the data is pre-processed (partitioned and transformed into Oracle ready data types) on Hadoop before the load begins.

Discussion

The online direct path option uses the least CPU time on the database server when compared to any other load option. This is a particularly important feature for online load options since applications can continue to run without being impacted by the load. The offline load options are faster, but they use more CPU time on the database server, since the database is doing more work. Choosing a load option depends on whether raw load speed is important or minimizing database CPU time is important. Creating Oracle Data Pump files on Hadoop has the benefit of both fast load times and offloading processing to Hadoop.

Test 3: Data Loading Scalability

Test Configuration: Oracle Big Data Appliance and a single database instance on Oracle Exadata X2-8

For each load option we compare the load rates for three different load sizes. Again, for the online load option we include the end-to-end load rate. As Tables 4 and 5 illustrate, the load rates are sustained as the data size grows. The elapsed time will increase with data size, but the load rate (amount of data loaded per second) remains constant. The load rate improves slightly as the data sizes get larger because the setup overhead as a fraction of the total time is minimized. Note that in these tests data is loaded from Oracle Big Data Appliance to a single database instance on Oracle Exadata.

TABLE 4. SCALABILITY OF LOAD RATES (OFFLINE)

LOAD OPTION	DATA SIZE (TB)	DATABASE LOAD RATE (GB/SEC)	ELAPSED TIME (SEC)
Offline load of Oracle Data Pump files or text files	0.1	1.3	76
	0.25	1.5	164
	0.5	1.6	316
	4	1.8	2246

TABLE 5. SCALABILITY OF LOAD RATES (ONLINE)

LOAD OPTION	DATA SIZE (TB)	END-TO-END LOAD RATE (GB/SEC)	ELAPSED TIME (SEC)
Online load with Direct Path	0.1	0.5	207
	0.25	0.54	472
	0.5	0.54	1239
	2	0.57	3536

Test 4: Comparison with Third Party Products

Test Configuration: Oracle Big Data Appliance and a single database instance on Oracle Exadata X2-8

The connectors were compared with third party products. The tests with third party products were also run on Oracle Big Data Appliance connected by InfiniBand to Oracle Exadata, and were highly tuned on this infrastructure. End-to-end load rates and CPU usage on the database server were compared for a data size of 0.25 TB.

Many third party load products are JDBC based so they were compared with the online load option of Oracle Loader for Hadoop with JDBC, so that similar products are compared. Oracle Loader for Hadoop with JDBC is more than twice as fast as JDBC based third party products (see Table 6) and uses 85% less database CPU time (see Figure 4).

TABLE 6: COMPARING LOAD RATES WITH THIRD PARTY PRODUCTS (0.25 TB)

LOAD OPTION	END-TO-END LOAD RATE (GB/SEC)	ELAPSED TIME (SEC)
Oracle Loader for Hadoop Online Load with JDBC	0.11	2226
Comparable third party product	0.042	5946

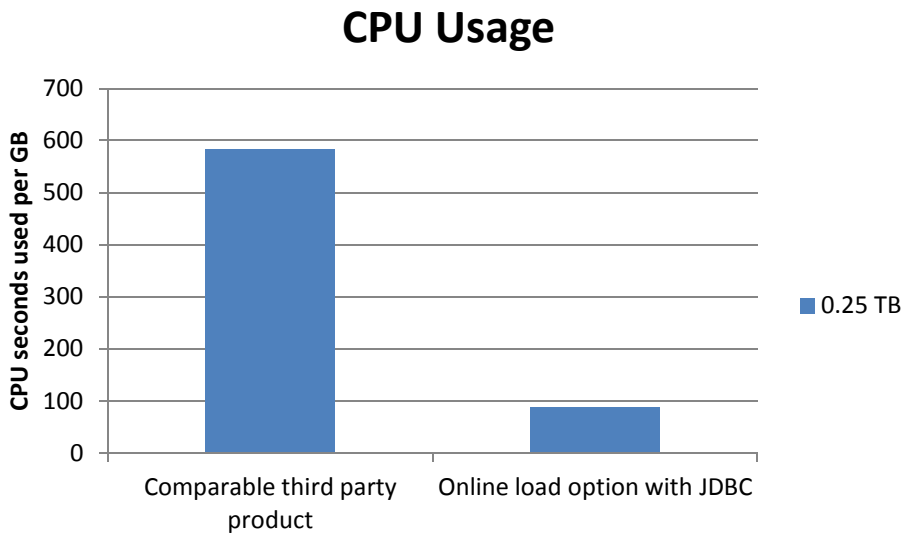

Figure 4. Comparing CPU usage on the database server: Third party product and Online load option using JDBC

Some products have implemented a direct path load option that loads into a copy of the user's table with a partitioning scheme defined by the loader to optimize the load process. This means the data will have to be copied into the target table after the load, and partitioned according to the user's desired partitioning scheme. (Partitioning is a generally a best practice for large volumes of data.) This adds to the time taken to load and consumes additional database CPU. Oracle Loader for Hadoop's online direct path load option, like its other load options, loads directly into the user's target table using the user's partitioning scheme. So an equivalent comparison was not possible.

Functionally, there are similarities between Oracle Direct Connector for HDFS and Fuse-DFS. Both can be used to access files on HDFS from the database. Oracle Direct Connector for HDFS with Oracle Data Pump files is 5 times faster than using Fuse-DFS (see Table 7) while using nearly 75% less CPU time on the database server (Figure 5).

TABLE 7: COMPARING LOAD RATES OF ORACLE DIRECT CONNECTOR FOR HDFS WITH FUSE-DFS (0.25 TB)

LOAD OPTION	LOAD RATE (GB/SEC)	ELAPSED TIME (SEC)
Oracle Direct Connector for HDFS	1.5	164
Comparable third party product (FUSE-DFS)	0.3	777

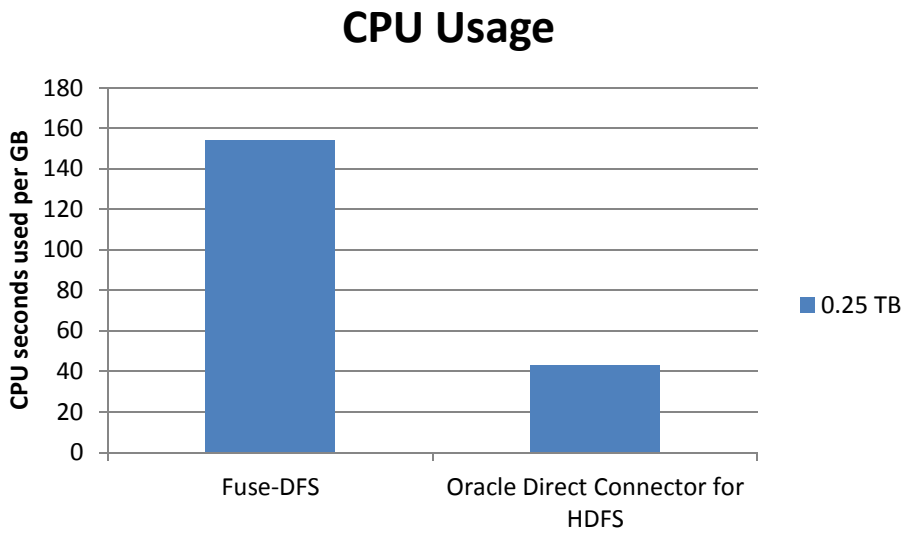

Figure 5. Comparing Oracle Direct Connector for HDFS and Fuse-DFS: CPU usage on the database server

Discussion

Comparing products that work similarly is technically sound. When comparing the Oracle Loader for Hadoop online load option with JDBC to third party products that use JDBC, the Oracle Loader for Hadoop option is more than twice as fast as the third party products and uses 85% less CPU time. Oracle Direct Connector for HDFS is 5 times faster than Fuse-DFS, and for loading Oracle Data Pump files uses about 75% less CPU time.

When performance is compared across load options, the fastest load option with Oracle's connectors (loading 12 TB an hour) is 5 to 20 times faster than third party products that were measured.

Loading of skewed data

When data is not evenly distributed among the partitions of the target table, the sampler option distributes the workload evenly across all reducers. This prevents any reducer node from becoming a bottleneck. Thus the load rates remain the same whether the data is skewed or uniformly distributed.

Summary

Files on HDFS can be loaded at a rate of 12 TB an hour with Oracle Big Data Appliance and Oracle Exadata. Database CPU time can be reduced by 50% if the data is first transformed into Oracle Data Pump files using Oracle Loader for Hadoop.

With the online load options, the database does less work and the Hadoop nodes do more work. So the online direct path load option uses only 20-30% of database CPU time compared to some other load options, but is slower than the fastest load time. This option is useful for continuous data feeds. It also has the advantage of simpler manageability because there is no intermediate staging of files.

The connectors were compared with third party products by matching them with load options that functioned similarly. Oracle Direct Connector for HDFS is five times faster than Fuse-DFS while using 75% less CPU time (when used with Oracle Data Pump files), and Oracle Loader for Hadoop online JDBC load option is twice as fast as the JDBC based third party products and uses 85% less CPU time. The fastest load option with Oracle's connectors (loading 12 TB an hour) is five to 20 times faster than these other products.

If input data corresponding to partitions in the target table is skewed, some nodes will do more work than others and slow down the load job. With Oracle Loader for Hadoop the load can be balanced evenly across the nodes to avoid such bottlenecks.

With the support of user-defined input formats, Oracle Loader for Hadoop can read data in virtually any format. When it transforms data into Oracle Data Pump files it enables Oracle Direct Connector for HDFS to access data that originated in a wide range of formats.

Conclusion

Oracle Loader for Hadoop and Oracle Direct Connector for Hadoop Distributed File System (HDFS) are high performance connectors to load and access data from a Hadoop platform to Oracle Database. They achieve a load speed of up to 12 TB an hour from Oracle Big Data Appliance to Oracle Exadata, and use less database CPU time than other products. Developed and supported by Oracle, these fast and efficient connectors support a range of use cases.

High Performance Connectors for Load and Access of Data from Hadoop to Oracle Database
June 2012
Contact: Melliyal Annamalai

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Hardware and Software, Engineered to Work Together