Oracle® Big Data Appliance

Software User's Guide

Release 1 (1.1)

E36162-04

October 2012

Provides an introduction to Oracle Big Data Appliance end-user software and to the administrative tools and procedures.

Oracle Big Data Appliance Software User's Guide, Release 1 (1.1)

E36162-04

Contents

Preface

The *Oracle Big Data Appliance Software User's Guide* describes how to manage and use the installed software.

Audience

This guide is intended for users of Oracle Big Data Appliance including:

- Application developers
- Data analysts
- Data scientists
- Database administrators
- System administrators

The *Oracle Big Data Appliance Software User's Guide* introduces the terminology and concepts necessary to discuss Oracle Big Data Appliance. However, you must acquire the necessary information about administering Hadoop clusters and writing MapReduce programs from other sources.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Oracle Big Data Appliance Owner's Guide*
- *Oracle Big Data Connectors User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introducing Oracle Big Data Appliance

This chapter presents an overview of Oracle Big Data Appliance and describes the software installed on the system. This chapter contains the following sections:

What Is Big Data?

Using transactional data as the source of business intelligence has been commonplace for many years. As digital technology and the World Wide Web spread into every aspect of modern life, other sources of data can make important contributions to business decision making. Many businesses are looking to these new data sources. They are finding opportunities in analyzing vast amounts of data that until recently was discarded.

Big data is characterized by:

- High Variety
- High Complexity
- High Volume
- High Velocity

These characteristics pinpoint the challenges in deriving value from big data, and the differences between big data and traditional data sources that primarily provide highly structured, transactional data.

High Variety

Big data is derived from a variety of sources, such as:

- Equipment sensors: Medical, manufacturing, transportation, and other machine sensor transmissions
- Machines: Call detail records, web logs, smart meter readings, Global Positioning System (GPS) transmissions, and trading systems records

- Social media: Data streams from social media sites such as Facebook and blogging sites such as Twitter

Analysts can mine this data repeatedly as they devise new ways of extracting meaningful insights. What seems irrelevant today might prove to be highly pertinent to your business tomorrow.

Challenge: Delivering flexible systems to handle this high variety

High Complexity

As the variety of data types increases, the complexity of the system increases. The complexity of data types also increases in big data because of its low structure.

Challenge: Finding solutions that apply across a broad range of data types.

High Volume

Social media can generate terabytes of daily data. Equipment sensors and other machines can generate that much data in less than an hour.

Even traditional data sources for data warehouses, such as customer profiles from customer relationship management (CRM) systems, transactional enterprise resource planning (ERP) data, store transactions, and general ledger data, have increased tenfold in volume over the past decade.

Challenge: Providing scalability and ease in growing the system

High Velocity

Huge numbers of sensors, web logs, and other machine sources generate data continuously and at a much higher speed than traditional sources, such as individuals entering orders into a transactional database.

Challenge: Handling the data at high speed without stressing the structured systems

The Oracle Big Data Solution

Oracle Big Data Appliance is an engineered system comprising both hardware and software components. The hardware is optimized to run the enhanced big data software components.

Oracle Big Data Appliance delivers:

- A complete and optimized solution for big data
- Single-vendor support for both hardware and software
- An easy-to-deploy solution
- Tight integration with Oracle Database and Oracle Exadata Database Machine

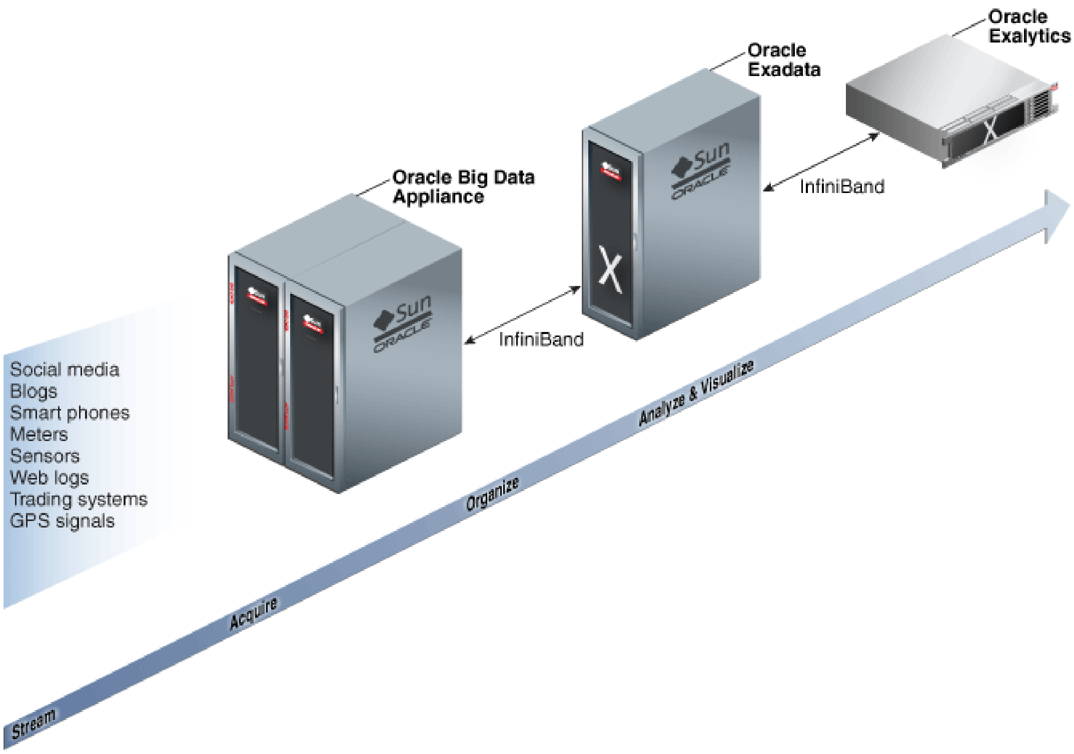
Oracle provides a big data platform that captures, organizes, and supports deep analytics on extremely large, complex data streams flowing into your enterprise from a large number of data sources. You can choose the best storage and processing location for your data depending on its structure, workload characteristics, and end-user requirements.

Oracle Database enables all data to be accessed and analyzed by a large user community using identical methods. By adding Oracle Big Data Appliance in front of Oracle Database, you can bring new sources of information to an existing data warehouse. Oracle Big Data Appliance is the platform for acquiring and organizing big data so that the relevant portions with true business value can be analyzed in Oracle Database.

For maximum speed and efficiency, Oracle Big Data Appliance can be connected to Oracle Exadata Database Machine running Oracle Database. Oracle Exadata Database Machine provides outstanding performance in hosting data warehouses and transaction processing databases. Moreover, Oracle Exadata Database Machine can be connected to Oracle Exalytics In-Memory Machine for the best performance of business intelligence and planning applications. The InfiniBand connections between these engineered systems provide high parallelism, which enables high-speed data transfer for batch or query workloads.

Figure 1–1 shows the relationships among these engineered systems.

Figure 1–1 Oracle Engineered Systems for Big Data



Software for Big Data

The **Oracle Linux** operating system and Cloudera's Distribution including Apache Hadoop (CDH) underlie all other software components installed on Oracle Big Data Appliance. **CDH** is an integrated stack of components that have been tested and packaged to work together.

CDH has a batch processing infrastructure that can store files and distribute work across a set of computers. Data is calculated on the same computer where it is stored. In a single Oracle Big Data Appliance rack, CDH distributes the files and workload across 18 servers, which compose a **cluster**. Each server is a node in the cluster.

The software framework consists of these primary components:

- File system: The **Hadoop Distributed File System (HDFS)** is a highly scalable file system that stores large files across multiple servers. It achieves reliability by replicating data across multiple servers without RAID technology. It runs on top of the Linux file system on Oracle Big Data Appliance.

- MapReduce engine: The MapReduce engine provides a platform for the massively parallel execution of algorithms written in Java.
- Administrative framework: Cloudera Manager is a comprehensive administrative tool for CDH.

CDH is written in Java, and Java is the language for applications development. However, several CDH utilities and other software available on Oracle Big Data Appliance provide graphical, web-based, and other language interfaces for ease of use.

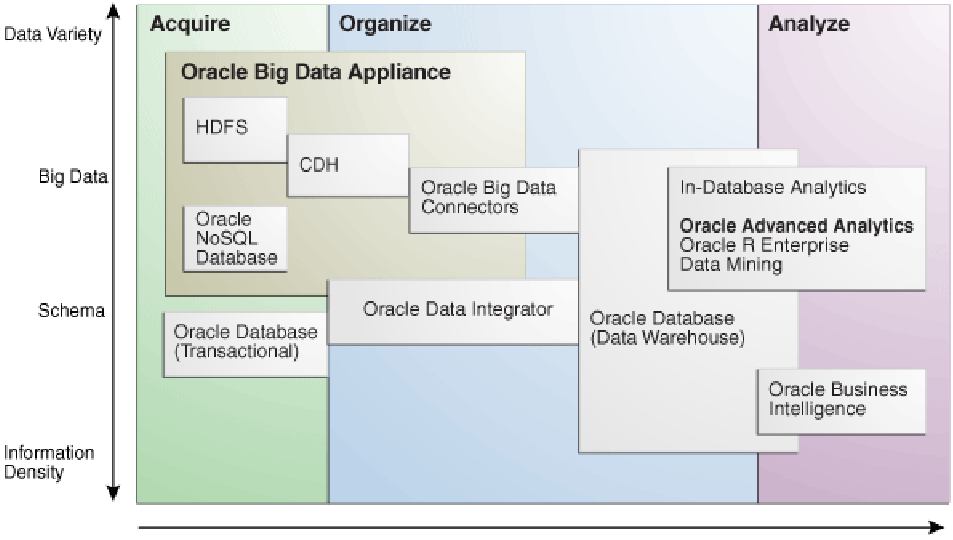
Software Component Overview

The major software components perform three basic tasks:

- Acquire
- Organize
- Analyze and visualize

The best tool for each task depends on the density of the information and the degree of structure. Figure 1–2 shows the relationships among the tools and identifies the tasks that they perform.

Figure 1–2 Oracle Big Data Appliance Software Overview



Acquiring Data for Analysis

Oracle Big Data Appliance provides these facilities for capturing and storing big data:

- Hadoop Distributed File System (HDFS)
- Oracle NoSQL Database
- **Hive**

Databases used for online transaction processing (OLTP) are the traditional data sources for data warehouses. The Oracle solution enables you to analyze traditional data stores with big data in the same Oracle data warehouse. Relational data continues to be an important source of business intelligence, although it runs on separate hardware from Oracle Big Data Appliance.

Hadoop Distributed File System

Cloudera's Distribution including Apache Hadoop (CDH) on Oracle Big Data Appliance uses the Hadoop Distributed File System (HDFS). HDFS stores extremely large files containing record-oriented data. On Oracle Big Data Appliance, HDFS splits large data files into chunks of 256 megabytes (MB), and replicates each chunk across three different nodes in the cluster. The size of the chunks and the number of replications are configurable.

Chunking enables HDFS to store files that are larger than the physical storage of one server. It also allows the data to be processed in parallel across multiple computers with multiple processors, all working on data that is stored locally. Replication ensures the high availability of the data: if a server fails, the other servers automatically take over its work load.

HDFS is typically used to store all types of big data.

See Also:

- For conceptual information about Hadoop technologies, refer to this third-party publication:

  Hadoop: The Definitive Guide, Third Edition by Tom White (O'Reilly Media Inc., 2012., ISBN: 978-1449311520).

- For documentation about Cloudera's Distribution including Apache Hadoop, see the Cloudera library at

  http://oracle.cloudera.com/

Oracle NoSQL Database

Oracle NoSQL Database is a distributed key-value database built on the proven storage technology of Berkeley DB Java Edition. Whereas HDFS stores unstructured data in very large files, Oracle NoSQL Database indexes the data and supports transactions. But unlike Oracle Database, which stores highly structured data, Oracle NoSQL Database has relaxed consistency rules, no schema structure, and only modest support for joins, particularly across storage nodes.

NoSQL databases, or "Not Only SQL" databases, have developed over the past decade specifically for storing big data. However, they vary widely in implementation. Oracle NoSQL Database has these characteristics:

- Uses a system-defined, consistent hash index for data distribution
- Supports high availability through replication
- Provides single-record, single-operation transactions with relaxed consistency guarantees
- Provides a Java API

Oracle NoSQL Database is designed to provide highly reliable, scalable, predictable, and available data storage. The key-value pairs are stored in shards or partitions (that is, subsets of data) based on a primary key. Data on each shard is replicated across multiple storage nodes to ensure high availability. Oracle NoSQL Database supports fast querying of the data, typically by key lookup.

An intelligent driver links the NoSQL database with client applications and provides access to the requested key-value on the storage node with the lowest latency.

Oracle NoSQL Database includes hashing and balancing algorithms to ensure proper data distribution and optimal load balancing, replication management components to handle storage node failure and recovery, and an easy-to-use administrative interface to monitor the state of the database.

Oracle NoSQL Database is typically used to store customer profiles and similar data for identifying and analyzing big data. For example, you might log in to a website and see advertisements based on your stored customer profile (a record in Oracle NoSQL Database) and your recent activity on the site (web logs currently streaming into HDFS).

Oracle NoSQL Database is an optional component of Oracle Big Data Appliance. It is always installed, but might not be activated during installation of the software.

See Also:

- *Oracle NoSQL Database Getting Started Guide* at

  http://docs.oracle.com/cd/NOSQL/html/index.html

- *Oracle Big Data Appliance Licensing Information*

Hive

Hive is an open-source data warehouse that supports data summarization, ad hoc querying, and data analysis of data stored in HDFS. It uses a SQL-like language called **HiveQL**. An interpreter generates MapReduce code from the HiveQL queries. By storing data in Hive, you can avoid writing MapReduce programs in Java.

Hive is a component of CDH and is always installed on Oracle Big Data Appliance. Oracle Data Integrator Application Adapter for Hadoop uses Hive extensively.

Organizing Big Data

Oracle Big Data Appliance provides several ways of organizing, transforming, and reducing big data for analysis:

- MapReduce
- Oracle R Support for Big Data
- Oracle Big Data Connectors

MapReduce

The MapReduce engine provides a platform for the massively parallel execution of algorithms written in Java. MapReduce uses a parallel programming model for processing data on a distributed system. It can process vast amounts of data quickly and can scale linearly. It is particularly effective as a mechanism for batch processing of unstructured and semistructured data. MapReduce abstracts lower-level operations into computations over a set of keys and values.

Although big data is often described as unstructured, incoming data always has some structure. However, it does not have a fixed, predefined structure when written to HDFS. Instead, MapReduce creates the desired structure as it reads the data for a particular job. The same data can have many different structures imposed by different MapReduce jobs.

A simplified description of a MapReduce job is the successive alternation of two phases: the Map phase and the Reduce phase. Each Map phase applies a transform function over each record in the input data to produce a set of records expressed as key-value pairs. The output from the Map phase is input to the Reduce phase. In the Reduce phase, the Map output records are sorted into key-value sets, so that all records in a set have the same key value. A reducer function is applied to all the records in a set, and a set of output records is produced as key-value pairs. The Map phase is logically run in parallel over each record, whereas the Reduce phase is run in parallel over all key values.

Oracle R Support for Big Data

R is an open-source language and environment for statistical analysis and graphing It provides linear and nonlinear modeling, standard statistical methods, time-series analysis, classification, clustering, and graphical data displays. Thousands of open-source packages are available in the Comprehensive R Archive Network (CRAN) for a spectrum of applications, such as bioinformatics, spatial statistics, and financial and marketing analysis. The popularity of R has increased as its functionality matured to rival that of costly proprietary statistical packages.

Analysts typically use R on a PC, which limits the amount of data and the processing power available for analysis. Oracle eliminates this restriction by extending the R platform to directly leverage Oracle Big Data Appliance. Analysts continue to work on their PCs using the familiar R user interface while manipulating huge amounts of data stored in HDFS using massively parallel processing.

The standard R distribution is installed on all nodes of Oracle Big Data Appliance, enabling R programs to run as MapReduce jobs on vast amounts of data. Users can transfer existing R scripts and packages from their PCs to use on Oracle Big Data Appliance.

Oracle R Connector for Hadoop provides R users with high-performance, native access to HDFS and the **MapReduce** programming framework. Oracle R Connector for Hadoop is included in the Oracle Big Data Connectors. See "Oracle R Connector for Hadoop" on page 1-9.

Oracle R Enterprise is a separate package that provides real-time access to Oracle Database. It enables you to store the results of your analysis of big data in an Oracle database, where it can be analyzed further.

These two Oracle R packages make Oracle Database and the Hadoop computational infrastructure available to statistical users without requiring them to learn the native programming languages of either one.

See Also:

- For information about R, go to

  `http://www.r-project.org/`

- For information about Oracle R Enterprise, go to

  `http://docs.oracle.com/cd/E27988_01/welcome.html`

Oracle Big Data Connectors

Oracle Big Data Connectors facilitate data access between data stored in CDH and Oracle Database. The connectors are licensed separately from Oracle Big Data Appliance and include:

- Oracle Direct Connector for Hadoop Distributed File System

- Oracle Loader for Hadoop
- Oracle Data Integrator Application Adapter for Hadoop
- Oracle R Connector for Hadoop

> **See Also:** *Oracle Big Data Connectors User's Guide*

Oracle Direct Connector for Hadoop Distributed File System

Oracle Direct Connector for Hadoop Distributed File System (Oracle Direct Connector) provides read access to HDFS from an Oracle database using **external tables**.

An external table is an Oracle Database object that identifies the location of data outside of the database. Oracle Database accesses the data by using the metadata provided when the external table was created. By querying the external tables, users can access data stored in HDFS as if that data were stored in tables in the database. External tables are often used to stage data to be transformed during a database load.

You can use Oracle Direct Connector to:

- Access any data stored in HDFS files
- Access comma-separated value (CSV) files generated by Oracle Loader for Hadoop
- Load data extracted and transformed by Oracle Data Integrator

Oracle Loader for Hadoop

Oracle Loader for Hadoop is an efficient and high-performance loader for fast movement of data from CDH into a table in an Oracle database. Oracle Loader for Hadoop partitions the data and transforms it into a database-ready format on CDH. It optionally sorts records by primary key before loading the data or creating output files.

You can use Oracle Loader for Hadoop as either a Java program or a command-line utility. The load runs as a MapReduce job on the CDH cluster.

Oracle Loader for Hadoop also reads from and writes to Oracle Data Pump files.

Oracle Data Integrator Application Adapter for Hadoop

Oracle Data Integrator (ODI) extracts, transforms, and loads data into Oracle Database from a wide range of sources.

In ODI, a knowledge module (KM) is a code template dedicated to a specific task in the data integration process. You use Oracle Data Integrator Studio to load, select, and configure the KMs for your particular application. More than 150 KMs are available to help you acquire data from a wide range of third-party databases and other data repositories. You only need to load a few KMs for any particular job.

Oracle Data Integrator Application Adapter for Hadoop contains the KMs specifically for use with big data. Table 1–1 describes these knowledge modules. They stage the data in Hive, a data warehouse built on Hadoop, for the best performance.

Table 1–1 Oracle Data Integrator Application Adapter for Hadoop Knowledge Modules

KM	Description
IKM File to Hive (Load Data)	Loads data from local and HDFS files into Hive tables. It provides options for better performance through Hive partitioning and fewer data movements.

Table 1–1 (Cont.) Oracle Data Integrator Application Adapter for Hadoop Knowledge

KM	Description
IKM Hive Control Append	Integrates data into a Hive target table in truncate/insert (append) mode. Data can be controlled (validated). Invalid data is isolated in the error table and can be recycled.
IKM Hive Transform	Integrates data into a Hive target table after the data has been transformed by a customized script such as Perl or Python.
IKM File-Hive to Oracle (OLH)	Integrates data from an HDFS file or Hive source into an Oracle Database target using Oracle Loader for Hadoop.
CKM Hive	Validates data against constraints.
RKM Hive	Reverse engineers Hive tables.

See Also: *Oracle Fusion Middleware Application Adapters Guide for Oracle Data Integrator* Version 1.1.1.6 or later

Oracle R Connector for Hadoop

Oracle R Connector for Hadoop is an R package that provides an interface between the local R environment, Oracle Database, and CDH on Oracle Big Data Appliance. Using simple R functions, you can sample data in HDFS, copy data between Oracle Database and HDFS, and schedule R programs to execute as MapReduce jobs. You can return the results to Oracle Database or to a file on your laptop.

Analyzing and Visualizing Big Data

After big data is transformed and loaded in Oracle Database, you can use the full panoply of Oracle business intelligence solutions and decision support products to further analyze and visualize all your data.

See Also:

- Oracle Business Intelligence website at

  `http://www.oracle.com/us/solutions/ent-performance-bi/business-intelligence/index.html`

- Data Warehousing and Business Intelligence in the Oracle Database Documentation Library at

  `http://www.oracle.com/pls/db112/portal.portal_db?selected=6&frame=`

2

Administering Oracle Big Data Appliance

This chapter provides information about the software and services installed on Oracle Big Data Appliance. It contains these sections:

Managing CDH Operations

Cloudera Manager is installed on Oracle Big Data Appliance to help you with Cloudera's Distribution including Apache Hadoop (CDH) operations. Cloudera Manager provides a single administrative interface to all Oracle Big Data Appliance servers configured as part of the Hadoop cluster.

Cloudera Manager simplifies the performance of these administrative tasks:

- Monitor jobs and services
- Start and stop services
- Manage security and Kerberos credentials
- Monitor user activity
- Monitor the health of the system
- Monitor performance metrics
- Track hardware use (disk, CPU, and RAM)

Cloudera Manager runs on the Cloudera Manager node (node02) and is available on port 7180.

To use Cloudera Manager:

1. Open a browser and enter a URL like the following:

   ```
   http://bda1node02.example.com:7180
   ```

In this example, `bda1` is the name of the appliance, `node02` is the name of the server, `example.com` is the domain, and `7180` is the default port number for Cloudera Manager.

2. Log in with a user name and password for Cloudera Manager. Only a user with administrative privileges can change the settings. Other Cloudera Manager users can view the status of Oracle Big Data Appliance.

> **See Also:** *Cloudera Manager User Guide* at
>
> http://ccp.cloudera.com/display/ENT/Cloudera+Manager+User+Guide
>
> or click **Help** on the Cloudera Manager Help menu

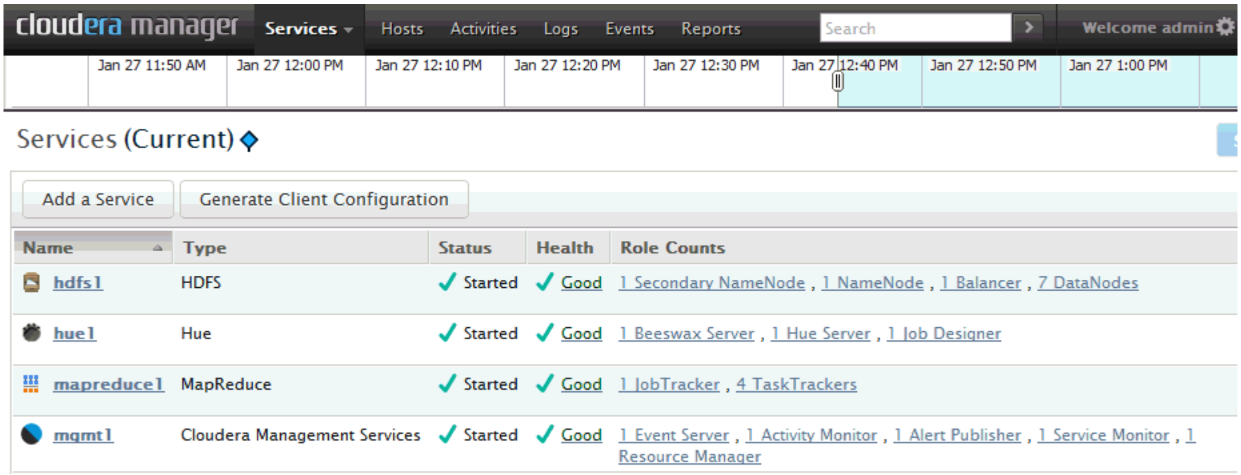
Monitoring the Status of Oracle Big Data Appliance

In Cloudera Manager, you can choose any of the following pages from the menu bar across the top of the display:

- **Services**: Monitors the status and health of services running on Oracle Big Data Appliance. Click the name of a service to drill down to additional information.
- **Hosts**: Monitors the health, disk usage, load, physical memory, swap space, and other statistics for all servers.
- **Activities**: Monitors all MapReduce jobs running in the selected time period.
- **Logs**: Collects historical information about the systems and services. You can search for a particular phrase for a selected server, service, and time period. You can also select the minimum severity level of the logged messages included in the search: TRACE, DEBUG, INFO, WARN, ERROR, or FATAL.
- **Events**: Records a change in state and other noteworthy occurrences. You can search for one or more keywords for a selected server, service, and time period. You can also select the event type: Audit Event, Activity Event, Health Check, or Log Message.
- **Reports**: Generates reports on demand for disk and MapReduce use.

Figure 2–1 shows the opening display of Cloudera Manager, which is the Services page.

Figure 2–1 Cloudera Manager Services Page



Name	Type	Status	Health	Role Counts
hdfs1	HDFS	✓ Started	✓ Good	1 Secondary NameNode , 1 NameNode , 1 Balancer , 7 DataNodes
hue1	Hue	✓ Started	✓ Good	1 Beeswax Server , 1 Hue Server , 1 Job Designer
mapreduce1	MapReduce	✓ Started	✓ Good	1 JobTracker , 4 TaskTrackers
mgmt1	Cloudera Management Services	✓ Started	✓ Good	1 Event Server , 1 Activity Monitor , 1 Alert Publisher , 1 Service Monitor , 1 Resource Manager

Performing Administrative Tasks

As a Cloudera Manager administrator, you can change various properties for monitoring the health and use of Oracle Big Data Appliance, add users, and set up Kerberos security.

To access Cloudera Manager Administration:

1. Log in to Cloudera Manager with administrative privileges.
2. Click **Welcome admin** at the top right of the page.

Collecting Diagnostic Information

If you need help from Oracle Support to troubleshoot CDH issues, then you should first collect diagnostic information using Cloudera Manager.

To collect diagnostic information about CDH:

1. Log in to Cloudera Manager with administrative privileges.
2. From the Help menu, click **Send Diagnostic Data**.
3. Verify that **Send Diagnostic Data to Cloudera Automatically** is not selected. Keep the other default settings.
4. Click **Collect Host Statistics Globally**.
5. Wait while all statistics are collected on all nodes.
6. Click **Download Result Data** and save the ZIP file with the default name. It identifies your CDH license.
7. Go to My Oracle Support at `http://support.oracle.com`.
8. Open a Service Request (SR) if you have not already done so.
9. Upload the ZIP file into the SR. If the file is too large, then upload it to `ftp.oracle.com`, as described in the next procedure.

To upload the diagnostics to ftp.oracle.com:

1. Open an FTP client and connect to `ftp.oracle.com`.

   You can use an FTP client such as WinSCP to upload the ZIP file. See Example 2–1 if you are using a command-line FTP client.

2. Log in as user `anonymous` and leave the password field blank.
3. In the bda/incoming directory, create a directory using the SR number for the name, in the format SR*number*. The resulting directory structure looks like this:

```
bda
   incoming
      SRnumber
```

4. Set the binary option to prevent corruption of binary data.
5. Upload the diagnostics ZIP file to the bin directory.
6. Update the SR with the full path and file name.

Example 2–1 shows the commands to upload the diagnostics using the Windows FTP command interface.

Example 2–1 Uploading Diagnostics Using Windows FTP

```
ftp> open ftp.oracle.com
Connected to bigip-ftp.oracle.com.
220-***********************************************************************
220-Oracle FTP Server
         .
         .
         .
220-****************************************************************************

220
User (bigip-ftp.oracle.com:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> cd bda/incoming
250 Directory successfully changed.
ftp> mkdir SR12345
257 "/bda/incoming/SR12345" created
ftp> cd SR12345
250 Directory successfully changed.
ftp> bin
200 Switching to Binary mode.
ftp> put D:\Downloads\3609df...c1.default.20122505-15-27.host-statistics.zip
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.
ftp: 706755 bytes sent in 1.97Seconds 358.58Kbytes/sec.
```

Using Hadoop Monitoring Utilities

Users can monitor MapReduce jobs without providing a Cloudera Manager user name and password.

Monitoring the JobTracker

Hadoop Map/Reduce Administration monitors the **JobTracker**, which runs on port 50030 of the JobTracker node (node03) on Oracle Big Data Appliance.

To monitor the JobTracker:

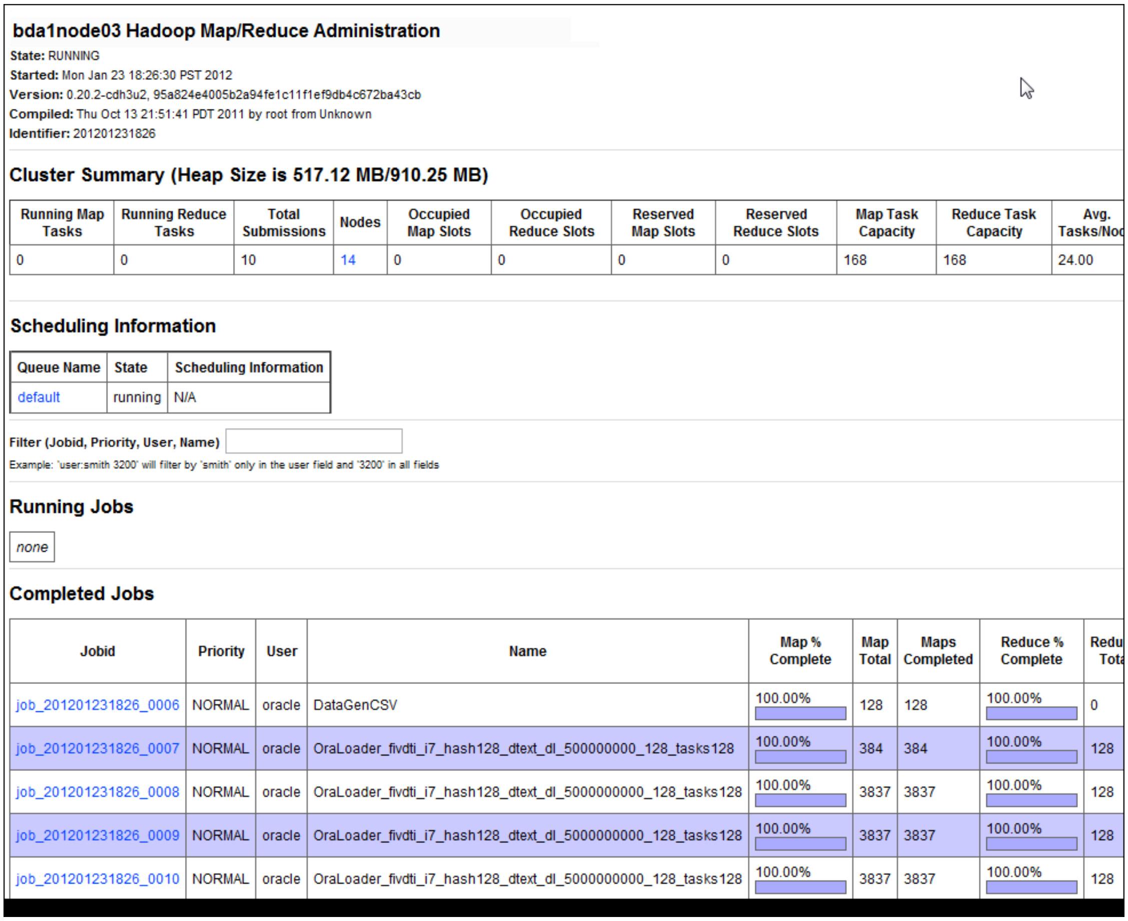
- Open a browser and enter a URL like the following:

```
http://bda1node03.example.com:50030
```

In this example, `bda1` is the name of the appliance, `node03` is the name of the server, and `50030` is the default port number for Hadoop Map/Reduce Administration.

Figure 2–2 shows part of a Hadoop Map/Reduce Administration display.

Figure 2–2 Hadoop Map/Reduce Administration



bda1node03 Hadoop Map/Reduce Administration

State: RUNNING
Started: Mon Jan 23 18:26:30 PST 2012
Version: 0.20.2-cdh3u2, 95a824e4005b2a94fe1c11f1ef9db4c672ba43cb
Compiled: Thu Oct 13 21:51:41 PDT 2011 by root from Unknown
Identifier: 201201231826

Cluster Summary (Heap Size is 517.12 MB/910.25 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Nod
0	0	10	14	0	0	0	0	168	168	24.00

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Redu Tota
job_201201231826_0006	NORMAL	oracle	DataGenCSV	100.00%	128	128	100.00%	0
job_201201231826_0007	NORMAL	oracle	OraLoader_fivdti_i7_hash128_dtext_dl_500000000_128_tasks128	100.00%	384	384	100.00%	128
job_201201231826_0008	NORMAL	oracle	OraLoader_fivdti_i7_hash128_dtext_dl_5000000000_128_tasks128	100.00%	3837	3837	100.00%	128
job_201201231826_0009	NORMAL	oracle	OraLoader_fivdti_i7_hash128_dtext_dl_5000000000_128_tasks128	100.00%	3837	3837	100.00%	128
job_201201231826_0010	NORMAL	oracle	OraLoader_fivdti_i7_hash128_dtext_dl_5000000000_128_tasks128	100.00%	3837	3837	100.00%	128

Monitoring the TaskTracker

The Task Tracker Status interface monitors the **TaskTracker** on a single node. It is available on port 50060 of all noncritical nodes (node04 to node18) in Oracle Big Data Appliance.

To monitor a TaskTracker:

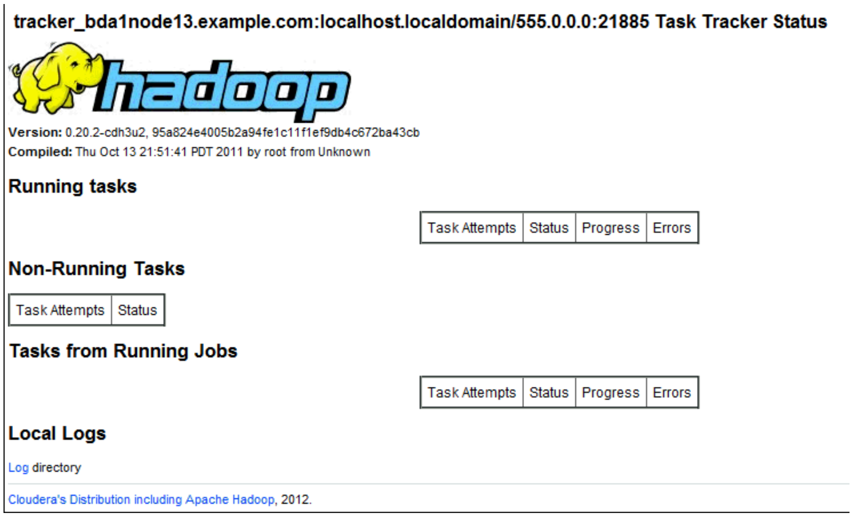
- Open a browser and enter the URL for a particular node like the following:

  ```
  http://bda1node13.example.com:50060
  ```

  In this example, `bda1` is the name of the rack, `node13` is the name of the server, and `50060` is the default port number for the Task Tracker Status interface.

Figure 2–3 shows the Task Tracker Status interface.

Figure 2–3 Task Tracker Status Interface



tracker_bda1node13.example.com:localhost.localdomain/555.0.0.0:21885 Task Tracker Status

Version: 0.20.2-cdh3u2, 95a824e4005b2a94fe1c11f1ef9db4c672ba43cb
Compiled: Thu Oct 13 21:51:41 PDT 2011 by root from Unknown

Running tasks

Task Attempts	Status	Progress	Errors

Non-Running Tasks

Task Attempts	Status

Tasks from Running Jobs

Task Attempts	Status	Progress	Errors

Local Logs

Log directory

Cloudera's Distribution including Apache Hadoop, 2012.

Providing Remote Client Access to CDH

Oracle Big Data Appliance supports full local access to all commands and utilities in Cloudera's Distribution including Apache Hadoop (CDH).

You can use a browser on any computer that has access to the client network of Oracle Big Data Appliance to access Cloudera Manager, Hadoop Map/Reduce Administration, the Hadoop Task Tracker interface, and other browser-based Hadoop tools.

To issue Hadoop commands remotely, however, you must connect from a system configured as a CDH client with access to the Oracle Big Data Appliance client network. This section explains how to set up a computer so that you can access HDFS and submit MapReduce jobs on Oracle Big Data Appliance.

To follow these procedures, you must have these access privileges:

- Root access to the client system
- Login access to Cloudera Manager

If you do not have these access privileges, then contact your system administrator for help.

Installing CDH on the Client System

The system that you use to access Oracle Big Data Appliance must run an operating system that Cloudera supports for CDH3. For the list of supported operating systems, see "Before You Install CDH3 on a Cluster" in the *Cloudera CDH3 Installation Guide* at

`https://ccp.cloudera.com/display/CDHDOC/Before+You+Install+CDH3+on+a+Cluster`

To install the CDH client software:

1. Follow the installation instructions for your operating system provided in the *Cloudera CDH3 Installation Guide* at

   `https://ccp.cloudera.com/display/CDHDOC/CDH3+Installation`

   When you are done installing the Hadoop core and native packages, the system can act as a basic CDH client.

Note: Be sure to install CDH3 Update 4 (CDH3u4) or a later version.

2. To provide support for other components, such as **Hive**, **Pig**, or **Oozie**, see the component installation instructions.

Configuring CDH

After installing CDH, you must configure it for use with Oracle Big Data Appliance.

To configure the Hadoop client:

1. Open a browser on your client system and connect to Cloudera Manager. It runs on the Cloudera Manager node (node02) and listens on port 7180, as shown in this example:

   ```
   http://bda1node02.example.com:7180
   ```

2. Log in as `admin`.
3. Cloudera Manager opens on the Services tab. Click the **Generate Client Configuration** button.
4. On the Command Details page (shown in Figure 2–4), click **Download Result Data** to download `global-clientconfig.zip`.
5. Unzip `global-clientconfig.zip` into the `/tmp` directory on the client system. It creates a `hadoop-conf` directory containing these files:

   ```
   core-site.xml
   hadoop-env.sh
   hdfs-site.xml
   log4j.properties
   mapred-site.xml
   README.txt
   ssl-client.xml.example
   ```

6. Open `hadoop-env.sh` in a text editor and change `JAVA_HOME` to the correct location on your system:

   ```
   export JAVA_HOME=full_directory_path
   ```

7. Delete the number sign (#) to uncomment the line, and then save the file.
8. Copy the configuration files to the Hadoop conf directory:

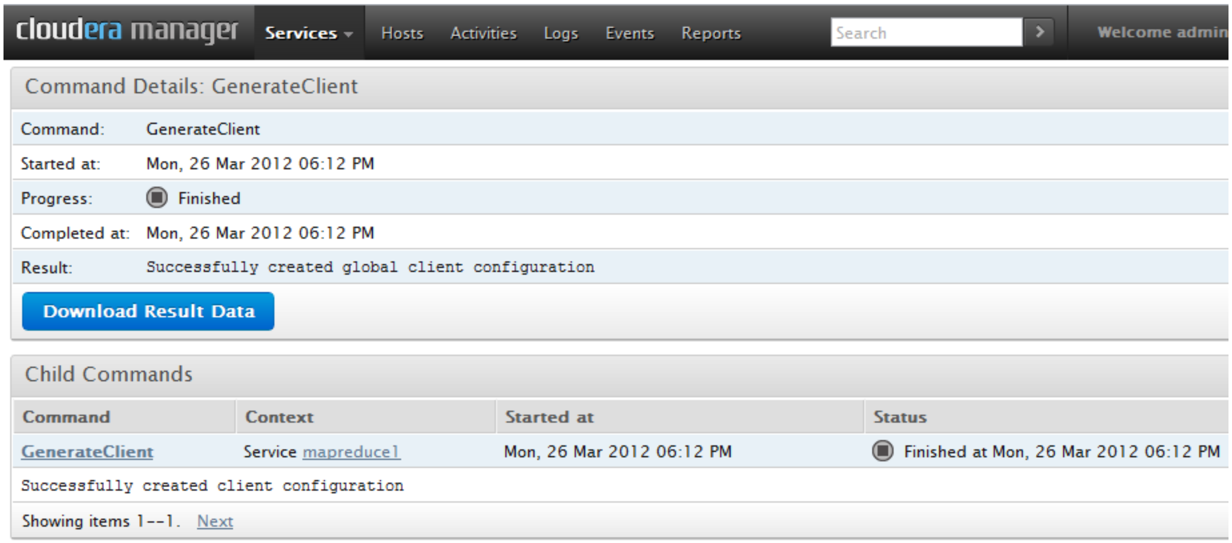
   ```
   cd /tmp/hadoop-conf
   cp * /usr/lib/hadoop/conf/
   ```

9. Validate the installation by changing to the `mapred` user and submitting a MapReduce job, such as the one shown here:

   ```
   su mapred
   hadoop jar /usr/lib/hadoop/hadoop-examples.jar pi 10 1000000
   ```

Figure 2–4 shows the download page for the client configuration.

Figure 2–4 Cloudera Manager Command Details: GenerateClient Page



Command Details: GenerateClient

Command:	GenerateClient
Started at:	Mon, 26 Mar 2012 06:12 PM
Progress:	Finished
Completed at:	Mon, 26 Mar 2012 06:12 PM
Result:	Successfully created global client configuration

Child Commands

Command	Context	Started at	Status
GenerateClient	Service mapreduce1	Mon, 26 Mar 2012 06:12 PM	Finished at Mon, 26 Mar 2012 06:12 PM

Successfully created client configuration

Showing items 1--1. Next

Managing User Accounts

Every open-source package installed on Oracle Big Data Appliance creates one or more users and groups. Most of these users do not have login privileges, shells, or home directories. They are used by daemons and are not intended as an interface for individual users. For example, Hadoop operates as the `hdfs` user, MapReduce operates as `mapred`, and Hive operates as `hive`. Table 2–1 identifies the operating system users and groups that are created automatically during installation of Oracle Big Data Appliance software for use by CDH components and other software packages.

You can use the `oracle` identity to run Hadoop and Hive jobs immediately after the Oracle Big Data Appliance software is installed. This user account has login privileges, a shell, and a home directory. Oracle NoSQL Database and Oracle Data Integrator run as the `oracle` user. Its primary group is `oinstall`.

> **Note:** Do not delete or modify the users created during installation, because they are required for the software to operate.

When creating additional user accounts, define them as follows:

- To run MapReduce jobs, users must be in the `hadoop` group.
- To create and modify tables in Hive, users must be in the `hive` group.

Table 2–1 Operating System Users and Groups

User Name	Group	Used By	Login Rights
flume	flume	**Flume** parent and nodes	No
hbase	hbase	**HBase** processes	No
hdfs	hadoop	**NameNode**, **DataNode**	No
hive	hive	Hive metastore and server processes	No
hue	hue	**Hue** processes	No

Table 2–1 (Cont.) Operating System Users and Groups

User Name	Group	Used By	Login Rights
mapred	hadoop	JobTracker, TaskTracker, **Hive Thrift** daemon	Yes
mysql	mysql	MySQL server	Yes
oozie	oozie	**Oozie** server	No
oracle	dba, oinstall	Oracle NoSQL Database, Oracle Loader for Hadoop, Oracle Data Integrator, and the Oracle DBA	Yes
puppet	puppet	Puppet parent (puppet nodes run as root)	No
sqoop	sqoop	**Sqoop** metastore	No
svctag	--	Auto Service Request	No
zookeeper	zookeeper	**ZooKeeper** processes	No

Recovering Deleted Files

CDH provides an optional trash facility, so that when a user deletes a file, it is moved to a trash directory for a set period of time instead of being deleted immediately from the system.

Restoring Files from the Trash

When the trash facility is enabled, you can easily restore files that were previously deleted with the Hadoop rm file-system command. Files deleted by other programs are not copied to the trash directory.

To restore a file from the trash directory:

1. Check that the deleted file is in the trash. The following example checks for files deleted by the oracle user:

```
$ hadoop fs -ls .Trash/Current/user/oracle
Found 1 items
-rw-r--r--  3 oracle hadoop  242510990 2012-08-31 11:20
/user/oracle/.Trash/Current/user/oracle/ontime_s.dat
```

2. Move or copy the file to its previous location. The following example moves ontime_s.dat from the trash to the HDFS /user/oracle directory.

```
$ hadoop fs -mv .Trash/Current/user/oracle/ontime_s.dat /user/oracle/ontime_
s.dat
```

Setting Up the Trash Facility

In this release of Oracle Big Data Appliance, the trash facility is disabled by default. Complete the following procedure to enable it.

To enable the trash facility:

1. On each node where you want to enable the trash facility, add the following property description to /etc/hadoop/conf/hdfs-site.xml:

```
<property>
     <name>fs.trash.interval</name>
     <value>1</value>
</property>
```

> **Note:** You can edit the hdfs-site.xml file once and then use the `dcli` utility to copy the file to the other nodes. See the *Oracle Big Data Appliance Owner's Guide*.

2. Change the trash interval as desired (optional). See "Changing the Trash Interval" on page 2-10.
3. Restart the hdfs1 service:
   a. Open Cloudera Manager. See "Managing CDH Operations" on page 2-1.
   b. Locate hdfs1 on the Cloudera Manager Services page.
   c. Expand the hdfs1 Actions menu and choose **Restart**.
4. Verify that trash collection is working properly:
   a. Copy a file from the local file system to HDFS. This example copies a data file named ontime_s.dat to the HDFS /user/oracle directory:

   ```
   $ hadoop fs -put ontime_s.dat /user/oracle
   ```

   b. Delete the file from HDFS:

   ```
   $ hadoop fs -rm ontime_s.dat
   Moved to trash: hdfs://bda1node02.example.com/user/oracle/ontime_s.dat
   ```

   c. Locate the trash directory in your home Hadoop directory, such as /user/oracle/.Trash. The directory is created when you delete a file for the first time after trash is enabled.

   ```
   $ hadoop fs -ls .Trash
   Found 1 items
   drwxr-xr-x  - oracle hadoop  0 2012-08-31 11:20 /user/oracle/.Trash/Current
   ```

   d. Check that the deleted file is in the trash. The following command lists files deleted by the `oracle` user:

   ```
   hadoop fs -ls .Trash/Current/user/oracle
   Found 1 items
   -rw-r--r--  3 oracle hadoop  242510990 2012-08-31 11:20
   /user/oracle/.Trash/Current/user/oracle/ontime_s.dat
   ```

5. If trash collection is not working on a particular node, then verify that `fs.trash.interval` is set in the /etc/hadoop/conf/hdfs-site.xml file on that node, and then restart the hdfs1 service.

Changing the Trash Interval

The **trash interval** is the minimum number of minutes that a file remains in the trash directory before being deleted permanently from the system. The default value is 1440 minutes (24 hours).

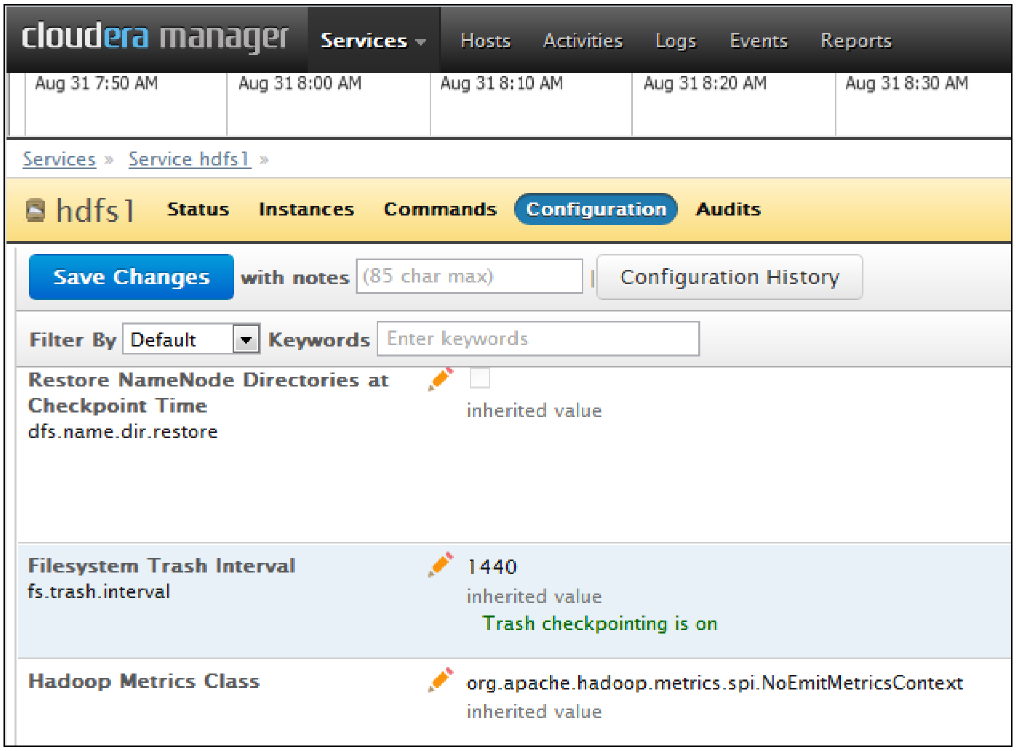
To change the trash interval:

1. Open Cloudera Manager. See "Managing CDH Operations" on page 2-1.
2. On the Services page under Name, click **hdfs1**.
3. On the hdfs1 page, click the **Configuration** subtab.

4. Search for or scroll down to the Filesystem Trash Interval property under NameNode Settings. See Figure 2–5.
5. Enter a new interval (in minutes) in the value field. A value of `0` disables the trash facility.
6. Click **Save Changes**.
7. Expand the Actions menu at the top of the page and choose **Restart**.

Figure 2–5 shows the Filesystem Trash Interval property in Cloudera Manager.

Figure 2–5 HDFS Property Settings in Cloudera Manager



Restore NameNode Directories at Checkpoint Time
dfs.name.dir.restore
□
inherited value

Filesystem Trash Interval
fs.trash.interval
1440
inherited value
Trash checkpointing is on

Hadoop Metrics Class
org.apache.hadoop.metrics.spi.NoEmitMetricsContext
inherited value

Software Layout

The following sections identify the software installed on Oracle Big Data Appliance and where it runs in the rack. Some components operate with Oracle Database 11.2.0.2 and later releases.

Software Components

These software components are installed on all 18 servers in Oracle Big Data Appliance Rack. Oracle Linux, required drivers, firmware, and hardware verification utilities are factory installed. All other software is installed on site using the Mammoth Utility.

> **Note:** You do not need to install software on Oracle Big Data Appliance. Doing so may result in a loss of warranty and support. See the *Oracle Big Data Appliance Owner's Guide*.

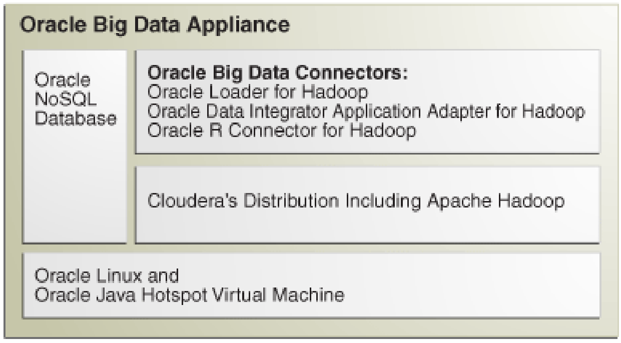
Installed software:

- Oracle Linux 5.6
- Java **HotSpot** Virtual Machine 6 Update 29
- Cloudera's Distribution including Apache Hadoop Release 3 Update 4 (CDH)
- Cloudera Manager 3.7
- Oracle Loader for Hadoop 1.1
- Oracle NoSQL Database Community Edition 11*g* Release 1.2.125
- Oracle Data Integrator Agent 11.1.1.6.0
- Oracle R Connector for Hadoop 1.1
- **Oracle R Distribution** 2.13.2
- **Oracle Database Instant Client** 11.2.0.3
- **MySQL Database** 5.5.17 Advanced Edition

> **See Also:** *Oracle Big Data Appliance Owner's Guide* for information about the Mammoth Utility

Figure 2–6 shows the relationships among the major components.

Figure 2–6 Major Software Components of Oracle Big Data Appliance


Oracle Big Data Appliance
Oracle NoSQL Database
Oracle Big Data Connectors:
Oracle Loader for Hadoop
Oracle Data Integrator Application Adapter for Hadoop
Oracle R Connector for Hadoop
Cloudera's Distribution Including Apache Hadoop
Oracle Linux and
Oracle Java Hotspot Virtual Machine

Logical Disk Layout

Each server has 12 disks. The critical information is stored on disks 1 and 2.

Table 2–2 describes how the disks are partitioned.

Table 2–2 Logical Disk Layout

Disk	Description
1 to 2	150 gigabytes (GB) mirrored, physical and logical partition with the Linux operating system, all installed software, NameNode data, and MySQL Database data, for a total of four copies 2.8 terabytes (TB) HDFS data partition
3	Single Oracle NoSQL Database partition, if activated during software installation; otherwise, a single HDFS data partition
4 to 12	Single HDFS data partition

Software Services

This section contains the following topics:

- Monitoring the Services
- About the Parent Services
- About the Child Services
- Where Do the Services Run?

Monitoring the Services

You can use Cloudera Manager to monitor the services on Oracle Big Data Appliance.

To monitor the services:

1. In Cloudera Manager, click the **Services** tab at the top of the page to display the Services page.
2. Click the name of a service to see its detail pages. The service opens on the Status page.
3. Click the link to the page that you want to view: Status, Instances, Commands, Configuration, or Audits.

About the Parent Services

Table 2–3 describes the parent services and those that run without child services. A **parent service** controls one or more child services.

Services that are always on are required for normal operation. Services that you can switch on and off are optional.

Table 2–3 Parent Services

Service	Role	Description	Default Status
hbase	--	HBase database	OFF
hdfs1	NameNode	Tracks all files stored in the cluster.	Always ON
hdfs1	Secondary NameNode	Tracks information for the NameNode	Always ON
hdfs1	Balancer	Periodically issues the balancer command; although the balancer service is enabled, it does not run all the time	Always ON
hive	--	Hive data warehouse for Hadoop	Always ON
hue1	**Hue** Server	GUI for HDFS, MapReduce, and Hive, with shells for Pig, Flume, and HBase	Always ON
mapreduce1	JobTracker	Used by MapReduce	Always ON
mgmt1	all	Cloudera Manager	Always ON
MySQL	--	MySQL Master Database	ON
ODI Agent	--	Oracle Data Integrator agent, installed on same node as MySQL Database	ON
oozie	--	Workflow and coordination service for Hadoop	OFF
ZooKeeper	--	ZooKeeper coordination service	OFF

About the Child Services

Table 2–4 describes the child services. A **child service** is controlled by a parent service.

Table 2–4 Child Services

Service	Role	Description	Default Status
HBase Region Server	--	Hosts data and processes requests for HBase	OFF
hdfs1	DataNode	Stores data in HDFS	Always ON
mapreduce1	TaskTracker	Accepts tasks from the JobTracker	Always ON
NoSQL DB Storage Node	--	Supports Oracle NoSQL Database	ON
nosqldb	--	Supports a web console or command-line interface for administering Oracle NoSQL Database	ON

Where Do the Services Run?

All services are installed on all servers, but individual services run only on designated nodes in the Hadoop cluster.

Service Locations

Table 2–5 identifies the nodes where the services run on the primary rack. Services that run on all nodes run on all racks of a multirack installation.

Table 2–5 Software Service Locations

Service	Node Name	Initial Node Position
Balancer	HDFS node	Node01
Beeswax Server	JobTracker node	Node03
Cloudera Manager Agents	All nodes	All nodes
Cloudera Manager SCM Server	Cloudera Manager node	Node02
DataNode	All nodes	All nodes
Hive Server	JobTracker node	Node03
Hue Server	JobTracker node	Node03
JobTracker	JobTracker node	Node03
MySQL Backup[1]	JobTracker node	Node03
MySQL Primary Server[1]	Cloudera Manager node	Node02
NameNode	HDFS node	Node01
Oracle Data Integrator Agent[2]	JobTracker node	Node03
Oracle NoSQL Database Administration[2]	Cloudera Manager node	Node02
Oracle NoSQL Database Server Processes[2]	All nodes	All nodes
Puppet Agents	All nodes	All nodes
Puppet Master	HDFS Node	Node01
Secondary NameNode	Cloudera Manager node	Node02
TaskTracker	All noncritical nodes	Node04 to Node18

[1] If the software was upgraded from version 1.0, then MySQL Backup remains on node02 and MySQL Primary Server remains on node03.

[2] Started only if requested in the *Oracle Big Data Appliance Configuration Worksheets*

NameNode

The NameNode is the most critical process because it keeps track of the location of all data. Without a healthy NameNode, the entire cluster fails. This vulnerability is intrinsic to Apache Hadoop (v0.20.2 and earlier).

Oracle Big Data Appliance protects against catastrophic failure by maintaining four copies of the NameNode logs:

- Node01: The working copy of the NameNode snapshot and update logs is stored in `/opt/hadoop/dfs/` and is automatically mirrored in a local Linux partition.
- Node02: A backup copy of the logs is stored in `/opt/shareddir/` and is also automatically mirrored in a local Linux partition.

A backup copy outside of Oracle Big Data Appliance can be configured during the software installation.

> **Note:** The Secondary NameNode is not a backup of the primary NameNode and does not provide failover. The Secondary NameNode performs memory-intensive functions for the primary NameNode.

Unconfigured Software

The following tools are installed but not configured. Before using them, you must configure them for your use.

- Flume
- HBase
- Mahout
- Oozie
- Sqoop
- Whirr
- ZooKeeper

> **See Also:** *CDH3 Installation and Configuration Guide* for configuration procedures at
>
> `http://oracle.cloudera.com`

Effects of Hardware on Software Availability

The effects of a server failure vary depending on the server's function within the CDH cluster. Sun Fire servers are more robust than commodity hardware, so you should experience fewer hardware failures. This section highlights the most important services that run on the various servers of the primary rack. For a full list, see "Service Locations" on page 2-14.

Critical and Noncritical Nodes

Critical nodes are required for the cluster to operate normally and provide all services to users. In contrast, the cluster continues to operate with no loss of service when a noncritical node fails.

The critical services are installed initially on the first four nodes of the primary rack. Table 2–6 identifies the critical services that run on these nodes. The remaining nodes (initially node05 to node18) only run DataNode and TaskTracker services. If a hardware failure occurs on one of the critical nodes, then the services can be moved to another, noncritical server. For example, if node02 fails, its critical services might be moved to node05. For this reason, Table 2–6 provides names to identify the nodes providing critical services.

Table 2–6 Critical Nodes

Node Name	Initial Node Position	Critical Functions
HDFS Node	Node01	NameNode, balancer, puppet master; the cluster is unusable without the NameNode service.
Cloudera Manager Node	Node02	Cloudera Manager, Secondary NameNode, MySQL Server, Oracle NoSQL Database KV Administration
JobTracker Node	Node03	JobTracker, Hue, MySQL Server Backup
Backup Node	Node04	NameNode and Secondary NameNode data backups, unless an external NFS directory is configured for the cluster, making node04 a noncritical node

HDFS Node

The HDFS node (node01) is critically important because it is where the NameNode runs. If this server fails, the effect is downtime for the entire cluster, because the NameNode keeps track of the data locations. However, there are always four copies of the NameNode metadata.

The current state and update logs are written to these locations:

- **HDFS node (node01)**: /opt/hadoop/dfs/ on Disk 1 is the working copy with a Linux mirrored partition on Disk 2 providing a second copy.
- **Backup node (node04)**: /opt/shareddir/ on Disk 1 is the third copy, which is also duplicated on a mirrored partition on Disk 2. These copies can be written to an external NFS directory instead of the backup node.

Cloudera Manager Node

The Secondary NameNode runs on the Cloudera Manager node (node02). There are backups for this data like those for the NameNode. If the node fails, then these services are also disrupted:

- **Cloudera Manager**: This tool provides central management for the entire CDH cluster. Without this tool, you can still monitor activities using the utilities described in "Using Hadoop Monitoring Utilities" on page 2-4.
- **MySQL Master Database**: Cloudera Manager, Oracle Data Integrator, Hive, and Oozie use MySQL Database. The data is replicated automatically, but you cannot access it when the master database server is down.

- **Oracle NoSQL Database KV Administration**: Oracle NoSQL Database database is an optional component of Oracle Big Data Appliance, so the extent of a disruption due to a node failure depends on whether you are using it and how critical it is to your applications.

JobTracker Node

The JobTracker assigns MapReduce tasks to specific nodes in the CDH cluster. Without the JobTracker node (node03), this critical function is not performed. If the node fails, then these services are also disrupted:

- **Oracle Data Integrator**: This service supports Oracle Data Integrator Application Adapter for Hadoop. You cannot use this connector when the JobTracker node is down.
- **Hue**: Cloudera Manager uses Hadoop User Experience (Hue), and so Cloudera Manager is unavailable when Hue is unavailable.
- **MySQL Backup Database**: MySQL Database continues to run, although there is no backup of the master database.

Backup Node

The backup node (node04) backs up the NameNode and Secondary NameNode for most installations. The backups are important for ensuring the smooth functioning of the cluster, but there is no loss of user services if the backup node fails.

Some installations are configured to use an external NFS directory instead of a backup node. This is a configuration option decided during installation of the appliance. When the backup is stored outside the appliance, the node is noncritical.

Noncritical Nodes

The noncritical nodes (node05 to node18) are optional in that Oracle Big Data Appliance continues to operate with no loss of service if a failure occurs. The NameNode automatically replicates the lost data to maintain three copies at all times. MapReduce jobs execute on copies of the data stored elsewhere in the cluster. The only loss is in computational power, because there are fewer servers on which to distribute the work.

Security on Oracle Big Data Appliance

You can take precautions to prevent unauthorized use of the software and data on Oracle Big Data Appliance.

This section contains these topics:

- CDH Security
- Port Numbers Used on Oracle Big Data Appliance
- About Puppet Security

CDH Security

Apache Hadoop is not an inherently secure system. It is protected only by network security. After a connection is established, a client has full access to the system.

Cloudera's Distribution including Apache Hadoop (CDH) supports Kerberos network authentication protocol to prevent malicious impersonation. You must install and configure Kerberos and set up a Kerberos Key Distribution Center and realm. Then you configure various components of CDH to use Kerberos.

CDH provides these securities when configured to use Kerberos:

- The CDH master nodes, NameNode, and JobTracker resolve the group name so that users cannot manipulate their group memberships.
- Map tasks run under the identity of the user who submitted the job.
- Authorization mechanisms in HDFS and MapReduce help control user access to data.

> **See Also:** `http://oracle.cloudera.com` for these manuals:
>
> - *CDH3 Security Guide*
> - *Configuring Hadoop Security with Cloudera Manager*
> - *Configuring TLS Security for Cloudera Manager*

Port Numbers Used on Oracle Big Data Appliance

Table 2–7 identifies the port numbers that might be used in addition to those used by CDH. For the full list of CDH port numbers, go to the Cloudera website at

`http://ccp.cloudera.com/display/CDHDOC/Configuring+Ports+for+CDH3`

To view the ports used on a particular server:

1. In Cloudera Manager, click the **Hosts** tab at the top of the page to display the Hosts page.
2. In the Name column, click a server link to see its detail page.
3. Scroll down to the Ports section.

> **See Also:** The Cloudera website for CDH port numbers:
>
> - *Hadoop Default Ports Quick Reference* at
>
>   `http://www.cloudera.com/blog/2009/08/hadoop-default-ports-quick-reference/`
> - *Configuring Ports for CDH3* at
>
>   `https://ccp.cloudera.com/display/CDHDOC/Configuring+Ports+for+CDH3`

Table 2–7 Oracle Big Data Appliance Port Numbers

Service	Port
Automated Service Monitor (OASM)	30920
MySQL Database	3306
Oracle Data Integrator Agent	20910
Oracle NoSQL Database administration	5001
Oracle NoSQL Database processes	5010 to 5020
Oracle NoSQL Database registration	5000
Port map	111
Puppet master service	8140

Table 2–7 (Cont.) Oracle Big Data Appliance Port Numbers

Service	Port
Puppet node service	8139
rpc.statd	668
ssh	22
xinetd (service tag)	6481

About Puppet Security

The puppet node service (`puppetd`) runs continuously as root on all servers. It listens on port 8139 for "kick" requests, which trigger it to request updates from the puppet master. It does not receive updates on this port.

The puppet master service (`puppetmasterd`) runs continuously as the puppet user on the first server of the primary Oracle Big Data Appliance rack. It listens on port 8140 for requests to push updates to puppet nodes.

The puppet nodes generate and send certificates to the puppet master to register initially during installation of the software. For updates to the software, the puppet master signals ("kicks") the puppet nodes, which then request all configuration changes from the puppet master node that they are registered with.

The puppet master sends updates only to puppet nodes that have known, valid certificates. Puppet nodes only accept updates from the puppet master host name they initially registered with. Because Oracle Big Data Appliance uses an internal network for communication within the rack, the puppet master host name resolves using `/etc/hosts` to an internal, private IP address.

Glossary

ASR

Oracle Auto Service Request, a software tool that monitors the health of the hardware and automatically generates a service request if it detects a problem.

See also **OASM**.

balancer

A service that ensures that all nodes in the cluster store about the same amount of data, within a set range. Data is balanced over the nodes in the cluster, not over the disks in a node.

CDH

Cloudera's Distribution including Apache Hadoop, the version of Apache Hadoop and related components installed on Oracle Big Data Appliance.

Cloudera's Distribution including Apache Hadoop (CDH)

See **CDH**.

cluster

A group of servers on a network that are configured to work together. A server is either a master node or a worker node.

All servers in an Oracle Big Data Appliance rack form a cluster. Servers 1, 2, and 3 are master nodes. Servers 4 to 18 are worker nodes.

See **Hadoop**.

DataNode

A server in a CDH cluster that stores data in HDFS. A DataNode performs file system operations assigned by the NameNode.

See also **HDFS**; **NameNode**.

Flume

A distributed service in CDH for collecting and aggregating data from almost any source into a data store such as HDFS or HBase.

See also **HBase**; **HDFS**.

Hadoop

A batch processing infrastructure that stores files and distributes work across a group of servers. Oracle Big Data Appliance uses Cloudera's Distribution including Apache Hadoop (CDH).

Hadoop Distributed File System (HDFS)

See **HDFS**.

Hadoop User Experience (Hue)

See **Hue**.

HBase

An open-source, column-oriented database that provides random, read/write access to large amounts of sparse data stored in a CDH cluster. It provides fast lookup of values by key and can perform thousands of insert, update, and delete operations per second.

See also **cluster**.

HDFS

Hadoop Distributed File System, an open-source file system designed to store extremely large data files (megabytes to petabytes) with streaming data access patterns. HDFS splits these files into data blocks and distributes the blocks across a CDH cluster.

When a data set is larger than the storage capacity of a single computer, then it must be partitioned across several computers. A distributed file system can manage the storage of a data set across a network of computers.

See also **cluster**.

Hive

An open-source data warehouse in CDH that supports data summarization, ad hoc querying, and data analysis of data stored in HDFS. It uses a SQL-like language called HiveQL. An interpreter generates MapReduce code from the HiveQL queries.

By using Hive, you can avoid writing MapReduce programs in Java.

See also **Hive Thrift**; **HiveQL**; **MapReduce**.

Hive Thrift

A remote procedure call (RPC) interface for remote access to CDH for Hive queries.

See also **CDH**; **Hive**.

HiveQL

A SQL-like query language used by Hive.

See also **Hive**.

HotSpot

A Java Virtual Machine (JVM) that is maintained and distributed by Oracle. It automatically optimizes code that executes frequently, leading to high performance. HotSpot is the standard JVM for the other components of the Oracle Big Data Appliance stack.

Hue

Hadoop User Experience, a web user interface in CDH that includes several applications, including a file browser for HDFS, a job browser, an account management tool, a MapReduce job designer, and Hive wizards. Cloudera Manager runs on Hue.

See also **HDFS**; **Hive**.

Java HotSpot Virtual Machine

See **HotSpot**.

JobTracker

A service that assigns MapReduce tasks to specific nodes in the CDH cluster, preferably those nodes storing the data.

See also **Hadoop**; **MapReduce**.

MapReduce

A parallel programming model for processing data on a distributed system.

A MapReduce program contains these functions:

- Mappers: Process the records of the data set.
- Reducers: Merge the output from several mappers.
- Combiners: Optimizes the result sets from the mappers before sending them to the reducers (optional).

MySQL Database

A SQL-based relational database management system. Cloudera Manager, Oracle Data Integrator, Hive, and Oozie use MySQL Database as a metadata repository on Oracle Big Data Appliance.

NameNode

A service that maintains a directory of all files in HDFS and tracks where data is stored in the CDH cluster.

See also **HDFS**.

node

A server in a CDH cluster.

See also **cluster**.

OASM

Oracle Automated Service Manager, a service for monitoring the health of Oracle Sun hardware systems. Formerly named Sun Automatic Service Manager (SASM).

Oozie

An open-source workflow and coordination service for managing data processing jobs in CDH.

Oracle Database Instant Client

A small-footprint client that enables Oracle applications to run without a standard Oracle Database client.

Oracle Linux

An open-source operating system. Oracle Linux 5.6 is the same version used by Exalogic 1.1. It features the Oracle Unbreakable Enterprise Kernel.

Oracle R Distribution

An Oracle-supported distribution of the R open-source language and environment for statistical analysis and graphing.

Oracle R Enterprise

A component of the Oracle Advanced Analytics Option. It enables R users to run R commands and scripts for statistical and graphical analyses on data stored in an Oracle database.

Pig

An open-source platform for analyzing large data sets that consists of the following:

- Pig Latin scripting language
- Pig interpreter that converts Pig Latin scripts into MapReduce jobs

Pig runs as a client application.

See also **MapReduce**.

Puppet

A configuration management tool for deploying and configuring software components across a cluster. The Oracle Big Data Appliance initial software installation uses Puppet.

The Puppet tool consists of these components: puppet agents, typically just called puppets; the puppet master server; a console; and a cloud provisioner.

See also **puppet agent**; **puppet master**.

puppet agent

A service that primarily pulls configurations from the puppet master and applies them. Puppet agents run on every server in Oracle Big Data Appliance.

See also **Puppet**; **puppet master**

puppet master

A service that primarily serves configurations to the puppet agents.

See also **Puppet**; **puppet agent**.

Sqoop

A command-line tool that imports and exports data between HDFS or Hive and structured databases. The name Sqoop comes from "SQL to Hadoop." Oracle R Connector for Hadoop uses the Sqoop executable to move data between HDFS and Oracle Database.

table

In Hive, all files in a directory stored in HDFS.

See also **HDFS**.

TaskTracker

A service that runs on each node and executes the tasks assigned to it by the JobTracker service.

See also **JobTracker**.

ZooKeeper

A centralized coordination service for CDH distributed processes that maintains configuration information and naming, and provides distributed synchronization and group services.

Index