Cloudera Runtime 7.0.1

Apache Hive Overview

Date published: 2019-09-23
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Apache Hive key features

Major changes to Apache Hive 2.x improve Apache Hive 3.x transactions and security. Knowing the major differences between these versions is critical for SQL users, including those who use Apache Spark and Apache Impala.

Hive is a data warehouse system for summarizing, querying, and analyzing huge, disparate data sets. Cloudera Runtime (CR) services include Hive on Tez and Hive Metastore. Hive on Tez is based on Apache Hive 3.x, a SQL-based data warehouse system. The enhancements in Hive 3.x over previous versions can improve SQL query performance, security, and auditing capabilities. The Hive metastore (HMS) is a separate service, not part of Hive, not even necessarily on the same cluster. HMS stores the metadata on the backend for Hive, Impala, Spark, and other components.

ACID transaction processing

Hive 3 tables are ACID (Atomicity, Consistency, Isolation, and Durability)-compliant, which is critical to observing the right to be forgotten requirement of the GDPR (General Data Protection Regulation).

Shared metastore

Hive metastore (HMS) interoperates with multiple engines, Impala and Spark for example, simplifying interoperation between engines and user data access.

Low-latency analytical processing (CDP Public Cloud)

Hive processes transactions using low-latency analytical processing (LLAP) or the Apache Tez execution engine. The Hive LLAP service is not available in CDP Private Cloud Base.

Spark integration with Hive

Spark and Hive ACID tables interoperate using the Hive Warehouse Connector. You can access external tables from Spark directly using SparkSQL. You do not need HWC to read or write Hive external tables. Spark users just read from or write to Hive directly. You can write Hive external tables in ORC format only.

Security improvements

Apache Ranger secures Hive data by default. To meet demands for concurrency improvements, ACID support for GDPR, render security, and other features, Hive tightly controls the location of the warehouse on a file system, or object store, and memory resources.

Workload management at the query level

You can configure who uses query resources, how much can be used, and how fast Hive responds to resource requests. Workload management can improve parallel query execution, cluster sharing for queries, and query performance.

Materialized views

Because multiple queries frequently need the same intermediate roll up or joined table, you can avoid costly, repetitious query portion sharing, by precomputing and caching intermediate tables into views.

Query result cache

Hive filters and caches similar or identical queries. Hive does not recompute the data that has not changed. Caching repetitive queries can reduce the load substantially when hundreds or thousands of users of BI tools and web services query Hive.

Information schema database

When launched, Hive creates two databases from JDBC data sources: information_schema and sys. All Metastore tables are mapped into your tablespace and available in sys. The information_schema data reveals the state of the system, similar to sys database data. You can query information_schema using SQL standard queries.

Unavailable or unsupported interfaces

- Hive CLI (replaced by Beeline)
- WebHCat
- Hcat CLI
- SQL Standard Authorization
- MapReduce execution engine (replaced by Tez)
- S3 and LLAP (CDP Private Cloud Base 7.0 only)
- Hive CLI (replaced by Beeline)
- WebHCat
- Hcat CLI
- SQL Standard Authorization
- MapReduce execution engine (replaced by Tez)
- Spark execution engine (replaced by Tez)
- Hive Indexes

Changes after upgrading

To locate and use your Apache Hive 3.x tables after an upgrade, you need to understand the changes that occur during the upgrade process. Changes to the location of tables, permissions to HDFS directories, table types, ACID-compliance occur, and other changes occur.

Hive changes to table references using dot notation

Upgrading to CDP includes the Hive-16907 bug fix, which rejects `db.table` in SQL queries. The dot (.) is not allowed in table names. To reference the database and table in a table name, both must be enclosed in backticks as follows: `db`.`table`.

Hive changes to ACID properties

Hive 2.x and 3.x can have transactional and non-transactional tables. Transactional tables have atomic, consistent, isolation, and durable (ACID) properties. In Hive 2.x, the initial version of ACID transaction processing is ACID v1. In Hive 3.x, the mature version of ACID is ACID v2.

Native and non-native storage formats

Storage formats are a factor in upgrade changes to table types. Hive 2.x and 3.x supports the following Hadoop native and non-native storage formats:

- Native: Tables with built-in support in Hive, such as those in the following file formats:
  - Text
  - Sequence File
  - RC File
  - AVRO File
  - ORC File
  - Parquet File
- Non-native: Tables that use a storage handler, such as the DruidStorageHandler or HBaseStorageHandler

Upgrade changes to table types

The following table compares Hive table types and ACID operations before an upgrade from HDP 2.x and after an upgrade to CDP. The ownership of the Hive table file is a factor in determining table types and ACID operations after the upgrade.

Table 1: HDP 2.x and 3.x Table Type Comparison

HDP 2.x				HDP 3.x	
Table Type	**ACID v1**	**Format**	**Owner (user) of Hive Table File**	**Table Type**	**ACID v2**
External	No	Native or non-native	hive or non-hive	External	No
Managed	Yes	ORC	hive or non-hive	Managed, updatable	Yes
Managed	No	ORC	hive	Managed, updatable	Yes
			non-hive	External, with data delete	No
Managed	No	Native (but non-ORC)	hive	Managed, insert only	Yes
			non-hive	External, with data delete	No
Managed	No	Non-native	hive or non-hive	External, with data delete	No

Removal of Hive View and Tez View

CDP does not include Hive View or Tez View. In lieu of these capabilities, users who upgrade to CDP can use Data Analytics Studio.

Unsupported interfaces

CDP does not support the following interfaces that were available in HDP and CDH platforms.

CDP does not support following interfaces that were available in HDP and CDH platforms.

- S3 for storing tables and LLAP (available in CDP Public Cloud only)
- Hive CLI (replaced by Beeline)
- WebHCat
- Hcat CLI
- SQL Standard Authorization
- MapReduce execution engine (replaced by Tez)
- Spark execution engine (replaced by Tez)
- Hive Indexes
- Hive View and Tez View

You can use Data Analytics Studio in lieu of Hive View.

Partially unsupported interfaces

Apache Hadoop Distributed Copy (DistCP) is not supported for copying Hive ACID tables. See link below.

Unsupported Features

CDP does not support the following features that were available in HDP and CDH platforms:

- CREATE TABLE that specifies a managed table location

  Do not use the LOCATION clause to create a managed table. Hive assigns a default location in the warehouse to managed tables.
- CREATE INDEX

  Hive builds and stores indexes in ORC or Parquet within the main table, instead of a different table, automatically. Set hive.optimize.index.filter to enable use (not recommended--use materialized views instead). Existing indexes are preserved and migrated in Parquet or ORC to CDP during upgrade.
- Hive metastore (HMS) high availablility (HA) load balancing

  You need to set up HMS HA as described in the documentation (see link below).
-

Unsupported Connector Use

CDP does not support the Sqoop exports using the Hadoop jar command (the Java API) that Teradata documents. For more information, see link below.

Related Information

Configuring HMS for high availability

DispCp causes Hive ACID jobs to fail

Apache Hive 3 architectural overview

Understanding Apache Hive 3 major design features, such as default ACID transaction processing, can help you use Hive to address the growing needs of enterprise data warehouse systems.

Apache Tez

Apache Tez is the Hive execution engine for the Hive-on-Tez service in Cloudera Manager. MapReduce is not supported. In a Cloudera cluster, if a legacy script or application specifies MapReduce for execution, an exception occurs. Most user-defined functions (UDFs) require no change to execute on Tez instead of MapReduce.

With expressions of directed acyclic graphs (DAGs) and data transfer primitives, execution of Hive queries on Tez instead of MapReduce improves query performance. In Cloudera Data Platform (CDP), Tez is usually used only by Hive, and HiveServer launches and manages Tez AM automatically when HiveServer2 starts. SQL queries you submit to Hive are executed as follows:

- Hive compiles the query.
- Tez executes the query.
- Resources are allocated for applications across the cluster.
- Hive updates the data in the data source and returns query results.

Hive on Tez runs tasks on ephemeral containers and uses the standard YARN shuffle service.

Data storage and access control

One of the major architectural changes to support Hive 3 design gives Hive much more control over metadata memory resources and the vfile system, or object store. The following architectural changes from Hive 2 to Hive 3 provide improved security:

- Tightly controlled file system and computer memory resources, replacing flexible boundaries: Definitive boundaries increase predictability. Greater file system control improves security.
- Optimized workloads in shared files and YARN containers

CDP Private Cloud Base stores Hive data on HDFS by default. CDP Public Cloud stores Hive data on S3 by default. In the cloud, Hive uses HDFS merely for storing temporary files. Hive 3 is optimized for object stores such as S3 in the following ways:

- Hive uses ACID to determine which files to read rather than relying on the storage system.
- In Hive 3, file movement is reduced from that in Hive 2.
- Hive caches metadata and data agressively to reduce file system operations

The major authorization model for Hive is Ranger. Hive enforces access controls specified in Ranger. This model offers stronger security than other security schemes and more flexibility in managing policies.

This model permits only Hive to access the data warehouse. If you do not enable the Ranger security service, or other security, CDP Data Center by default Hive uses storage-based authorization (SBA) based on user impersonation.

HDFS permission changes

In CDP Private Cloud Base, SBA relies heavily on HDFS access control lists (ACLs). ACLs are an extension to the permissions system in HDFS. CDP Data Center turns on ACLs in HDFS by default, providing you with the following advantages:

- Increased flexibility when giving multiple groups and users specific permissions
- Convenient application of permissions to a directory tree rather than by individual files

Transaction processing

You can deploy new Hive application types by taking advantage of the following transaction processing characteristics:

- Mature versions of ACID transaction processing:

  ACID tables are the default table type.

  ACID enabled by default causes no performance or operational overload.
- Simplified application development, operations with strong transactional guarantees, and simple semantics for SQL commands

  You do not need to bucket ACID tables.
- Materialized view rewrites
- Automatic query cache
- Advanced optimizations

Hive client changes

CDP Private Cloud Base supports the thin client Beeline for working on the command line. You can run Hive administrative commands from the command line. Beeline uses a JDBC connection to Hive on Tez to execute commands. Parsing, compiling, and executing operations occur in Hive on Tez. Beeline supports many of the command-line options that Hive CLI supported. Beeline does not support hive -e set key=value to configure the Hive Metastore.

You enter supported Hive CLI commands by invoking Beeline using the hive keyword, command option, and command. For example, hive -e set. Using Beeline instead of the thick client Hive CLI, which is no longer supported, has several advantages, including low overhead. Beeline does not use the entire Hive code base. A small number of daemons required to run queries simplifies monitoring and debugging.

Hive enforces allowlist and denylist settings that you can change using SET commands. Using the denylist, you can restrict memory configuration changes to prevent instability. Different Hive instances with different allowlists and denylists to establish different levels of stability.

You use the grunt command line to work with Apache Pig.

Apache Hive Metastore sharing

HiveServer, Impala, and other components can share a remote Hive metastore. In CDP Public Cloud, HMS uses a pre-installed MySQL database. You perform little, or no, configuration of HMS in the cloud.

Spark integration

Spark and Hive tables interoperate using the Hive Warehouse Connector in some cases.

You can access ACID and external tables from Spark using the Hive Warehouse Connector. You do not need the Hive Warehouse Connector to read Hive external tables from Spark and write Hive external tables from Spark.

Query execution of batch and interactive workloads

You can connect to Hive using Data Analytics Studio (DAS), a JDBC command-line tool, such as Beeline, or using an JDBC/ODBC driver with a BI tool, such as Tableau. Clients communicate with an instance of the same HiveServer version. You configure the settings file for each instance to perform either batch or interactive processing.

Apache Hive content roadmap

The content roadmap provides links to the available content resources for Apache Hive.

Table 2: Apache Hive Content roadmap

Task	Resources	Source	Description
Understanding	Presentations and Papers about Hive	Apache wiki	Contains meeting notes, presentations, and whitepapers from the Apache community.
Getting Started	Hive Tutorial	Apache wiki	Provides a basic overview of Apache Hive and contains some examples on working with tables, loading data, and querying and inserting data.
Administering	Setting Up the Metastore	Apache wiki	Describes the metastore parameters.
	Setting Up Hive Server	Apache wiki	Describes how to set up the server. How to use a client with this server is described in the HiveServer2 Clients document.
Developing	Materialized Views	Apache wiki	Covers accelerating query processing in data warehouses by pre-computing summaries using materialized views.
	JdbcStorageHandler	Apache wiki	Describes how to read from a JDBC data source in Hive.
	Hive transactions	Apache wiki	Describes ACID operations in Hive.
	Hive Streaming API	Apache wiki	Explains how to use an API for pumping data continuously into Hive using clients such as NiFi and Flume.
	Hive Operators and Functions	Apache wiki	Describes the Language Manual UDF.
	Beeline: HiveServer2 Client	Apache wiki	Describes how to use the Beeline client.
Reference	SQL Language Manual	Apache wiki	Language reference documentation available in the Apache wiki.
Contributing	Hive Developer FAQ	Apache wiki	Resources available if you want to contribute to the Apache community.
	How to Contribute	Apache wiki	
	Hive Developer Guide	Apache wiki	
	Plug-in Developer Kit	Apache wiki	

Task	Resources	Source	Description
	Unit Test Parallel Execution	Apache wiki	
	Hive Architecture Overview	Apache wiki	
	Hive Design Docs	Apache wiki	
	Project Bylaws	Apache wiki	