CDH Post-Upgrade 1.0.0

Migrating Impala Data to CDP Private Cloud

Date published: 2021-2-22
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Migrating Impala Data Overview

Before migrating Impala data from the CDH platform to CDP, you must be aware of the semantic and behavioral differences between CDH and CDP Impala and the activities that need to be performed prior to the data migration.

To successfully migrate your critical Impala data to the Cloud environment you must learn about the capacity requirements in the target environment and also understand the performance difference between your current environment and the targeted environment.

Impala Changes between CDH and CDP

There are some differences between Impala in CDH and Impala in CDP. These changes affect Impala after you migrate your workload from CDH 5.13-5.16 or CDH 6.1 or later to Cloudera Private Cloud Base or Public Cloud. Some of these differences require you to change your Impala scripts or workflow.

The version of Impala you used in CDH version 5.11 - 5.16 or 6.1 or later changes to Impala 3.4 when you migrate the workload to Cloudera Private Cloud Base or Public Cloud.

Change location of Datafiles

If Impala managed tables are located on the HDFS in /user/hive/warehouse before the migration, the tables, converted to external, remain there. The migration process sets the hive.metastore.warehouse.dir property to this location, designating it the Hive warehouse location. You can change the location of the warehouse using Cloudera Manager.

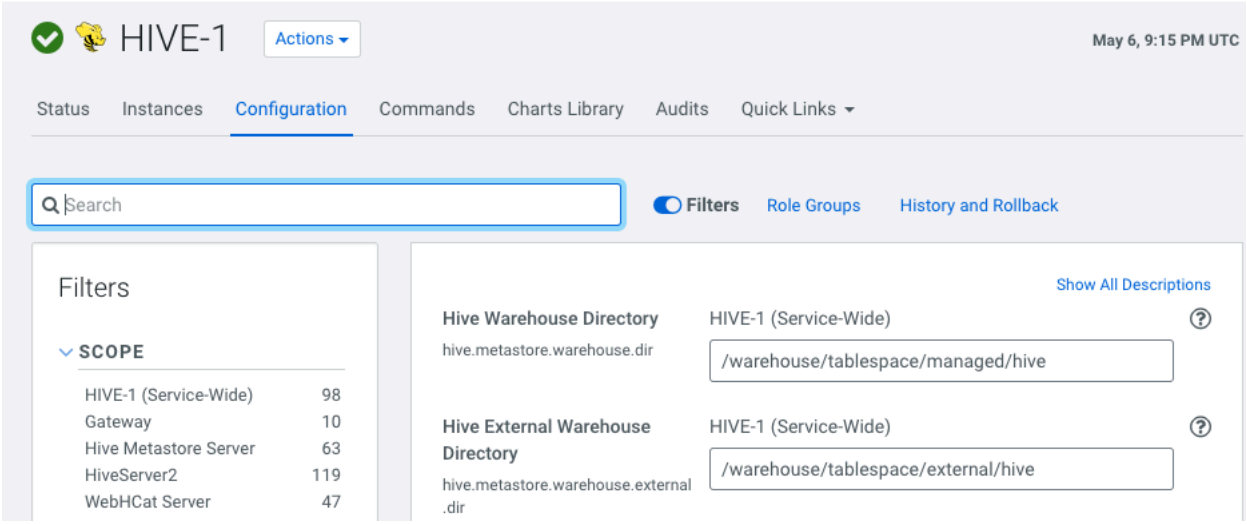
About this task

The location of existing tables does not change after a CDH to CDP migration. In CDP, there are separate HDFS directories for managed and external tables.

- The data files for managed tables are available in the warehouse location specified by the Cloudera Manager configuration setting, Hive Warehouse Directory.
- The data files for external tables are available in the warehouse location specified by the Cloudera Manager configuration setting, Hive Warehouse External Directory.

After migration, the (hive.metastore.warehouse.dir) is set to /user/hive/warehouse where the Impala managed tables are located.

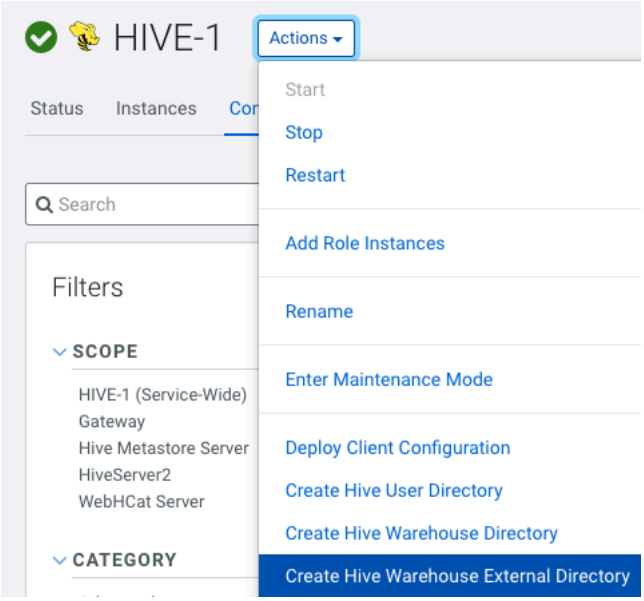
You can change the location of the warehouse using the Hive Metastore Action menu in Cloudera Manager.



Procedure

Create Hive Directories using the Hive Configuration page

a) Hive Action Menu Create Hive User Directory
b) Hive Action Menu Create Hive Warehouse Directory
c) Hive Action Menu Create Hive Warehouse External Directory



Set Storage Engine ACLs

You must be aware of the steps to set ACLs for Impala to allow Impala to write to the Hive Warehouse Directory.

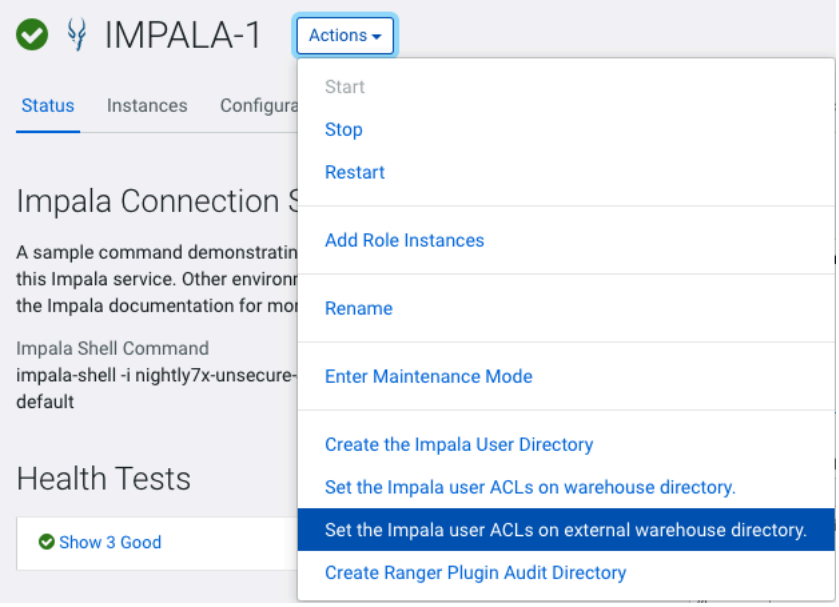
About this task

After migration, the (hive.metastore.warehouse.dir) is set to /user/hive/warehouse where the Impala managed tables are located. When the Impala workload is migrated from CDH to CDP, the ACL settings are automatically set for the default warehouse directories. If you changed the default location of the warehouse directories after migrating to CDP then follow the steps to allow Impala to write to the Hive Warehouse Directory.

Complete the initial configurations in the free-form fields on the Hive/Impala Configuration pages in Cloudera Manager to allow Impala to write to the Hive Warehouse Directory.

Procedure

1. Set Up Impala User ACLs using the Impala Configuration page
   a) Impala Action Menu Set the Impala user ACLs on warehouse directory
   b) Impala Action Menu Set the Impala user ACLs on external warehouse directory



2. Cloudera Manager sets the ACL for the user "Impala". However before starting the Impala service, verify permissions and ACLs set on the individual database directories using the sub-commands getfacl and setfacl.
   a) Verify the ACLs of HDFS directories for managed and external tables using getfacl.
   Example:

```
$ hdfs dfs -getfacl hdfs:///warehouse/tablespace/managed/hive
# file: hdfs:///warehouse/tablespace/managed/hive
# owner: hive
# group: hive
user::rwx
group::rwx
other::---
default:user::rwx
default:user:impala:rwx
default:group::rwx
default:mask::rwx
default:other::---
```

```
$ hdfs dfs -getfacl hdfs:///warehouse/tablespace/external/hive
# file: hdfs:///warehouse/tablespace/external/hive
# owner: hive
# group: hive
# flags: --t
user::rwx
group::rwx
other::rwx
default:user::rwx
default:user:impala:rwx
default:group::rwx
default:mask::rwx
```

```
default:other::rwx
```

b) If necessary, set the ACLs of HDFS directories using setfacl

Example:

```
$ hdfs dfs -setfacl hdfs:///warehouse/tablespace/managed/hive
$ hdfs dfs -setfacl hdfs:///warehouse/tablespace/external/hive
```

For more information on using the sub-commands getfacl and setfacl, see Using CLI commands to create and list ACLs.

c) The above examples show the user Impala as part of the Hive group. If in your setup, the user Impala does not belong to the group Hive then ensure that the Group user Impala belongs to has WRITE privileges assigned on the directory.

To view the Group user Impala belongs to:

```
$ id -Gn impala
uid=973(impala) gid=971(impala) groups=971(impala),972(hive)
```

Automatic Invalidation/Refresh of Metadata

To pick up new information when raw data is ingested into Tables you can use the hms_event_polling_interval_s flag.

New Default Behavior

When raw data is ingested into Tables, new HMS metadata and filesystem metadata are generated. In CDH, to pick up this new information, you must manually issue an Invalidate or Refresh command. However in CDP, this feature is controlled by the hms_event_polling_interval_s flag. This flag is set to 2 seconds by default. This option automatically refreshes the tables as changes are detected in HMS. When automatic invalidate/refresh of metadata is enabled, the Catalog Server polls Hive Metastore (HMS) notification events at a configurable interval and automatically applies the changes to Impala catalog. If specific tables that are not supported by event polling need to be refreshed, you must run a table level Invalidate or Refresh command.

For more information on Automatic Invalidation of metadata, see Automatic Invalidation

Metadata Improvements

In CDP, all catalog metadata improvements are enabled by default. You may use these few knobs to control how Impala manages its metadata to improve performance and scalability.

use_local_catalog

In CDP, the on-demand use_local_catalog mode is set to True by default on all the Impala coordinators so that the Impala coordinators pull metadata as needed from catalogd and cache it locally. This results in many performance and scalability improvements, such as reduced memory footprint on coordinators and automatic cache eviction.

catalog_topic_mode

The granularity of on-demand metadata fetches is at the partition level between the coordinator and catalogd. Common use cases like add/drop partitions do not trigger unnecessary serialization/deserialization of large metadata.

The feature can be used in either of the following modes.

Metadata on-demand mode

In this mode, all coordinators use the metadata on-demand.

Set the following on catalogd:

```
--catalog_topic_mode=minimal
```

Set the following on all impalad coordinators:

```
--use_local_catalog=true
```

Mixed mode

In this mode, only some coordinators are enabled to use the metadata on-demand.

Cloudera recommends that you use the mixed mode only for testing local catalog's impact on heap usage.

Set the following on catalogd:

```
--catalog_topic_mode=mixed
```

Set the following on impalad coordinators with metdadata on-demand:

```
--use_local_catalog=true
```

Limitation:

HDFS caching is not supported in On-demand metadata mode coordinators.

Reference:

See Impala Metadata Management for the details about catalog improvements.

Default Managed Tables

In CDP, managed tables are transactional tables with the insert_only property by default. You must be aware of the new default behavior of modifying file systems on a managed table in CDP and the methods to switch to the old behavior.

New Default Behavior

- You can no longer perform file system modifications (add/remove files) on a managed table in CDP. The directory structure for transactional tables is different than non-transactional tables, and any out-of-band files which are added may or may not be picked up by Hive and Impala.
- The insert_only transactional tables cannot be currently altered in Impala. The ALTER TABLE statement on a transactional table currently displays an error.
- Impala does not currently support compaction on transaction tables. You should use Hive to compact the tables.
- The SELECT, INSERT, INSERT OVERWRITE, and TRUNCATE statements are supported on the insert-only transactional tables.

Steps to switch to the CDH behavior:

- If you do not want transactional tables, set the DEFAULT_TRANSACTIONAL_TYPE query option to NONE so that any newly created managed tables are not transactional by default.
- External tables do not drop the data files when the table is dropped. To purge the data along with the table when the table is dropped, add external.table.purge = true in the table properties. When external.table.purge is set to true, the data is removed when the DROP TABLE statement is executed.

Automatic Refresh of Tables on Impala Clusters

The property enable_insert_events is used in CDP to refresh the tables or partitions automatically on other Impala clusters when Impala inserts into a table.

enable_insert_events

If Impala inserts into a table it refreshes the underlying table or partition. When this configuration enable_insert_events is set to True (default) Impala generates INSERT event types which when received by other Impala clusters automatically refreshes the tables or partitions.

Note: Event processing must be ON, for this property to work.

Interoperability between Hive and Impala

This topic describes the changes made in CDP for the optimal interoperability between Hive and Impala for the improved user experience.

Statistics Interoperability Between Hive and Impala

New default behavior:

Statistics for tables are engine specific, namely, Hive or Impala, so that each engine could use its own statistics and not overwrite the statistics generated by the other engine.

When you issue the COMPUTE STATS statement on Impala, you need to issue the corresponding statement on Hive to ensure both Hive and Impala statistics are accurate.

Impala COMPUTE STATS command does not overwrite the Hive stats for the same table.

Steps to switch to the CDH behavior:

There is no workaround.

Hive Default File Format Interoperability

New default behavior:

The managed tables created by Hive are of ORC file format, by default, and support full transactional capabilities. If you create a table without specifying the STORED AS clause and load data from Hive, then such tables are not readable or writable by Impala. But Impala can continue to read non-transactional and insert-only transactional ORC tables.

Steps to switch to the CDH behavior:

- You must use the STORED AS PARQUET clause when you create tables in Hive if you want interoperability with Impala on those tables.
- If you want to change this default file format at the system level, in the Hive_on_Tez service configuration in Cloudera Manager, set the hive_default_fileformat_managed field to parquet.

Impala supports a number of file formats used in Apache Hadoop. It can also load and query data files produced by other Hadoop components such as hive. After upgrading from any CDH 5.x version to 7.1, if you create a RC file in Hive using the default LazyBinaryColumnarSerDe, Impala will not be able to read the RC file. However you can set the configuration option of hive.default.rcfile.serde to ColumnarSerDe to maintain the interoperability between hive and impala.

Managed and External Tablespace Directories

New default behavior:

In CDP, there are separate HDFS directories for managed and external tables.

- The data files for managed tables are located in warehouse location specified by the Cloudera Manager configuration setting, hive_warehouse_directory.
- The data files for external tables are located in warehouse location specified by the Cloudera Manager configuration setting, hive_warehouse_external_directory.

If you perform file system level operations for adding/removing files on the table, you need to consider if its an external table or managed table to find the location of the table directory.

Steps to switch to the CDH behavior:

Check the output of the DESCRIBE FORMATTED command to find the table location.

ORC Support Disabled for Full-Transactional Tables

In CDP 7.1.0 and earlier versions, ORC table support is disabled for Impala queries. However, you have an option to switch to the CDH behavior by using the command line argument ENABLE_ORC_SCANNER.

New Default Behavior

In CDP 7.1.0 and earlier versions, if you use Impala to query ORC tables you will see it fail. To mitigate this situation, you must add explicit STORED AS clause to code creating Hive tables and use a format Impala can read. Another option is to set global configuration in Cloudera Manager to change hive_default_fileformat_managed.

Steps to switch to the CDH behavior:

Set the query option ENABLE_ORC_SCANNER to TRUE to re-enable ORC table support.

This option does not work on a full transactional ORC table, and the queries return an error.

Note: If you are using CDP 7.1.1 or later versions, the argument ENABLE_ORC_SCANNER is enabled by default and you can use Impala to query ORC tables without any manual interventions.

ORC vs Parquet in CDP

The differences between Optimized Row Columnar (ORC) file format for storing Hive data and Parquet for storing Impala data are important to understand. Query performance improves when you use the appropriate format for your application. The following table compares Hive and Impala support for ORC and Parquet in CDP Public Cloud and Cloudera Private Cloud Base. ORC vs Parquet

Authorization Provider for Impala

Using the BDR service available in CDH you can migrate the permissions in CDP because Ranger is the authorization provider instead of Sentry. You must be aware how Ranger enforces a policy in CDP which may be different from using Sentry.

New behavior:

- The CREATE ROLE, GRANT ROLE, SHOW ROLE statements are not supported as Ranger currently does not support roles.
- When a particular resource is renamed, currently, the policy is not automatically transferred to the newly renamed resource.
- SHOW GRANT with an invalid user/group does not return an error.

The following table lists the different access type requirements to run SQL statements in Impala.

SQL Statement	Impala Access Requirement
DESCRIBE *VIEW*	VIEW_METADATA on the underlying tables

SQL Statement	Impala Access Requirement
ALTER TABLE RENAME ALTER VIEW RENAME	ALL on the target table / view ALTER on the source table / view
SHOW DATABASES SHOW TABLES	VIEW_METADATA

where:

- VIEW_METADATA privilege denotes the SELECT, INSERT, or REFRESH privileges.
- ALL privilege denotes the SELECT, INSERT, CREATE, ALTER, DROP, and REFRESH privileges.

For more information on the minimum level of privileges and the scope required to run SQL statements in Impala, see Impala Authorization.

Migrating Sentry Policies

As CDP leverages Ranger as its authorization service, you must migrate permissions from Sentry to Ranger. You can use the BDR service available in CDH to migrate the permissions. This service migrates Sentry authorization policies into Ranger as a part of the replication policy job. When you create the replication policy, choose the resources that you want to migrate and the Sentry policies are migrated for those resources. You can migrate all permissions or permissions on a set of objects to Ranger.

The Sentry Permissions section of the Create Replication Policy wizard contains the following options:

- Include Sentry Permissions with Metadata - Select this to migrate Sentry permissions during the replication job.
- Exclude Sentry Permissions from Metadata - Select this if you do not want to migrate Sentry permissions during the replication job.

The Replication Option section of the Create Replication Policy wizard contains the following options:

- Include Metadata and Data
- Include Metadata Only

Stages of Migration

Sentry and Ranger have different permission models. Sentry permissions are granted to roles and users. These are translated to permissions for groups and users since Ranger currently does not support roles. This is followed by grouping by resource because Ranger policies are grouped by resource. All the permissions that are granted to a resource are considered a single Ranger policy.

The migration of Sentry policies into Ranger is performed in the following operations:

- Export - The export operation runs in the source cluster. During this operation, the Sentry permissions are fetched and exported to a JSON file. This file might be in a local file system or HDFS or S3, based on the configuration that you provided.
- Translate and Ingest - These operations take place on the target cluster. In the translate operation, Sentry permissions are translated into a format that can be read by Ranger. The permissions are then imported into Ranger. When the permissions are imported, they are tagged with the source cluster name and the time that the ingest took place. After the import, the file containing the permissions is deleted.

Because there is no one-to-one mapping between Sentry privileges and Ranger service policies, the Sentry privileges are translated into their equivalents within Ranger service policies. For more information on how Sentry actions is applied to the corresponding action in Ranger, see Sentry to Ranger Permissions.

Note: Because the authorization model in Ranger is different from Sentry's model, not all policies can be migrated using BDR. For certain resources you must manually create the permissions after migrating the workload from CDH to CDP.

Data Governance Support by Atlas

Both CDH and CDP environments support governance functionality for Impala operations. When migrating your workload from CDH to CDP, you must migrate your Navigator metadata to Atlas manually because there is no automatic migration of Navigator metadata from CDH to CDP.

The two environments collect similar information to describe Impala activities, including:

- Audits for Impala access requests
- Metadata describing Impala queries
- Metadata describing any new data assets created or updated by Impala operations

The services that support these operations are different in the two environments. Functionality is distributed across services as follows:

Feature	CDH	CDP
Auditing		
• Access requests	Audit tab in Navigator console	Audit page in Ranger console
• Service operations that create or update metadata catalog entries	Audit tab in Navigator console	Audit tab for each entity in Atlas dashboard
• Service operations in general	Audit tab in Navigator console	No other audits collected.
Metadata Catalog		

• Impala operations:
  • CREATE TABLE AS SELECT
  • CREATE VIEW
  • ALTER VIEW AS SELECT
  • INSERT INTO
  • INSERT
  • OVERWRITE
	Process and Process Execution entities
	Column- and table-level lineage
		Process and Process Execution entities
		Column- and table-level lineage

Migrating Navigator content to Atlas

As part of migrating your workload from CDH to CDP, you must use Atlas as the Cloudera Navigator Data Management for your cluster in CDP. You can choose to migrate your Navigator metadata to Atlas manually because there is no automatic migration of Navigator metadata from CDH to CDP. Atlas 'rebuilds' the metadata for existing cluster assets and lineage using new operations. However Navigator Managed metadata tags and any metadata you manually entered in CDH must be manually ported to Atlas Business Metadata Tags. If you have applications that are using the Navigator, you must port them to use Atlas APIs.

Note: Navigator Audit information is not ported. To keep legacy audit information you can keep a "read-only" Navigator instance until it is no longer needed. You may need to upgrade CM/Navigator on your legacy cluster to a newer version to avoid EOL.

Migrating content from Navigator to Atlas involves 3 steps:

- extracting the content from Navigator
- transforming that content into a form Atlas can consume
- importing the content into Atlas

Impala configuration differences in CDH and CDP

There are some configuration differences related to Impala in CDH and CDP. These differences are due to the changes made in CDP for the optimal interoperability between Hive and Impala for improved user experience. Review the changes before you migrate your Impala workload from CDH to CDP.

Default Value Changes in Configuration Options

Configuration Option	Scope	Default in CDH 6.x	Defaul
DEFAULT_FILE_FORMAT	Query	TEXT	PARQ
hms_event_polling_interval_s	Catalogd	0	2
ENABLE_ORC_SCANNER	Query	TRUE	FALSE
use_local_catalog	Coordinator / Catalogd	false	true
catalog_topic_mode	Coordinator	full	minima

New Configuration Options

Configuration Option	Scope	Default Value
default_transactional_type	Coordinator	insert_only
DEFAULT_TRANSACTIONAL_TYPE	Query	insert_only
disable_hdfs_num_rows_estimate	Impalad	false
disconnected_session_timeout	Coordinator	900
PARQUET_OBJECT_STORE_SPLIT_SIZE	Query	256 MB
SPOOL_QUERY_RESULTS	Query	FALSE
MAX_RESULT_SPOOLING_MEM	Query	100 MB
MAX_SPILLED_RESULT_SPOOLING_MEM	Query	1 GB
FETCH_ROWS_TIMEOUT_MS	Query	N/A
DISABLE_HBASE_NUM_ROWS_ESTIMATE	Query	FALSE
enable_insert_events		TRUE

Default File Formats

To improve useability and functionality, Impala significantly changed table creation. In CDP, the default file format of the tables is Parquet.

New Default Behavior

When you issue the CREATE TABLE statement without the STORED AS clause, Impala creates a Parquet table instead of a Text table as in CDH.

For example, if you create an external table based on a text file without providing the STORED AS clause and then issue a select query, the query fails in CDP, because Impala expects the file to be in the Parquet file format.

Steps to switch to the CDH behavior:

1. Add the explicitly stored as clause to in the CREATE TABLE statements if the file format is not Parquet.
2. Start Coordinator with the default_transactional_type flag set to text for all tables.

3. Set the default_file_format query option to TEXT to revert to the default Text format for one or more CREATE TABLE statements.

For more information on transactions supported by Impala, see SQL transactions in Impala

Reconnect to HS2 Session

Clients can disconnect from Impala while keeping the HiveSever2 (HS2) session running and can also reconnect to the same session by presenting the session_token.

New Default Behavior

By default, disconnected sessions are terminated after 15 min.

- Clients will not notice a difference because of this behavioral change.
- If clients are disconnected without the driver explicitly closing the session (for example, because of a network fault), disconnected sessions and the queries associated with them may remain open and continue consuming resources until the disconnected session is timed out. Administrators may notice these disconnected sessions and/or the associated resource consumption.

Steps to switch to the CDH behavior:

You can adjust the --disconnected_session_timeout flag to a lower value so that disconnected sessions are cleaned up quickly.

Automatic Row Count Estimation

To optimize complex or multi-table queries Impala has access to statistics about the volume of data and how the values are distributed. Impala uses this information to help parallelize and distribute the work for a query.

New Default Behavior

The Impala query planner can make use of statistics about entire tables and partitions. This information includes physical characteristics such as the number of rows, number of data files, the total size of the data files, and the file format. For partitioned tables, the numbers are calculated per partition, and as totals for the whole table. This metadata is stored in the Metastore database, and can be updated by either Impala or Hive.

If there are no statistics available on a table, Impala estimates the cardinality by estimating the size of table based on the number of rows in the table. This behavior is switched on by default and should result in better plans for most cases when statistics are not available.

For some edge cases, it is possible that Impala generates a bad plan (when compared to the same query in CDH) when the statistics are not present on that table and could negatively affect the query performance.

Steps to switch to the CDH behavior:

Set the DISABLE_HDFS_NUM_ROWS_ESTIMATE query option to TRUE to disable this optimization.

Using Reserved Words in SQL Queries

For ANSI SQL compliance, Impala rejects reserved words in SQL queries in CDP. A reserved word is one that cannot be used directly as an identifier. If you need to use it as an identifier, you must quote it with backticks.

About this task

New reserved words are added in CDH 6. To port SQL statements from CDH 5 that has different sets of reserved words, you must change queries that use references to such Tables or Databases using reserved words in the SQL syntax.

Procedure

1. Find a table having the problematic table reference, such as a create table statement that uses a reserved word such as select in the CREATE statement.
2. Enclose the table name in backticks.

```
CREATE TABLE select (x INT): fails
CREATE TABLE `select` (x INT): succeeds
```

For more information, see Impala identifiers and Impala reserved words.

Other Miscellaneous Changes in Impala

Review the changes to Impala syntax or service that might affect Impala after migrating your workload from CDH version 5.13-5.16 or CDH version 6.1 or later to Cloudera Private Cloud Base or CDP Public Cloud. The version of Impala you used in CDH version 5.11 - 5.16 or 6.1 or later changes to Impala 3.4 in Cloudera Private Cloud Base.

Decimal V2 Default

In CDP, Impala uses DECIMAL V2 by default.

To continue using the first version of the DECIMAL type for the backward compatibility of your queries, set the DECIMAL_V2 query option to FALSE:

```
SET DECIMAL_V2=FALSE;
```

Column Aliases Substitution

To conform to the SQL standard, Impala no longer performs alias substitution in the subexpressions of GROUP BY, HAVING, and ORDER BY.

The example below references to the actual column sum(ss_quantity) in the ORDER BY clause instead of the alias Total_Quantity_Purchased and also references to the actual column ss_item_sk in the GROUP BY clause instead of the alias Item because aliases are no longer supported in the subexpressions.

```
select
    ss_item_sk as Item,
    count(ss_item_sk) as Times_Purchased,
    sum(ss_quantity) as Total_Quantity_Purchased
    from store_sales
    group by ss_item_sk
    order by sum(ss_quantity) desc
    limit 5;
    +-------+-----------------+--------------------------+
    | item  | times_purchased | total_quantity_purchased |
    +-------+-----------------+--------------------------+
    | 9325  | 372             | 19072                    |
    | 4279  | 357             | 18501                    |
    | 7507  | 371             | 18475                    |
    | 5953  | 369             | 18451                    |
    | 16753 | 375             | 18446                    |
    +-------+-----------------+--------------------------+
```

Default PARQUET_ARRAY_RESOLUTION

The PARQUET_ARRAY_RESOLUTION query option controls the behavior of the indexed-based resolution for nested arrays in Parquet. In Parquet, you can represent an array using a 2-level or 3-level representation. The

default value for the PARQUET_ARRAY_RESOLUTION is THREE_LEVEL to match the Parquet standard 3-level encoding. See Parquet_Array_Resolution Query Option for more information.

Clustered Hint Default

The clustered hint is enabled by default, which adds a local sort by the partitioning columns in HDFS and Kudu tables to a query plan. The noclustered hint, which prevents clustering in tables having ordering columns, is ignored with a warning.

Query Options Removed

The following query options have been removed:

- DEFAULT_ORDER_BY_LIMIT
- ABORT_ON_DEFAULT_LIMIT_EXCEEDED
- V_CPU_CORES
- RESERVATION_REQUEST_TIMEOUT
- RM_INITIAL_MEM
- SCAN_NODE_CODEGEN_THRESHOLD
- MAX_IO_BUFFERS
- RM_INITIAL_MEM
- DISABLE_CACHED_READS

Shell Option refresh_after_connect

The refresh_after_connect option for starting the Impala Shell is removed.

EXTRACT and DATE_PART Functions

The EXTRACT and DATE_PART functions changed in the following way:

- The output type of the EXTRACT and DATE_PART functions changed to BIGINT.
- Extracting the millisecond part from a TIMESTAMP returns the seconds component and the milliseconds component. For example, EXTRACT (CAST('2006-05-12 18:27:28.123456789' AS TIMESTAMP), 'MILLIS ECOND') returns 28123.

Port for SHUTDOWN Command

If you upgraded from CDH 6.1 or later and specified a port as a part of the SHUTDOWN command, change the port number parameter to use the Kudu7: RPC (KRPC) port for communication between the Impala brokers.

Change in Client Connection Timeout

The default behavior of client connection timeout changes after upgrading.

In CDH 6.2 and lower, the client waited indefinitely to open the new session if the maximum number of threads specified by --fe_service_threads has been allocated.

After upgrading, the server requires a new startup flag, --accepted_client_cnxn_timeout to control treatment of new connection requests. The configured number of server threads is insufficient for the workload.

If --accepted_client_cnxn_timeout > 0, new connection requests are rejected after the specified timeout.

If --accepted_client_cnxn_timeout=0, client waits indefinitely to connect to Impala. This sets pre-upgrade behavior.

The default timeout is 5 minutes.

Interoperability between Hive and Impala

Impala supports a number of file formats used in Apache Hadoop. It can also load and query data files produced by other Hadoop components such as Hive. After upgrading from any CDH 5.x version to Cloudera Private Cloud Base 7.1, if you create an RC file in Hive using the default LazyBinaryColumnarSerDe, Impala can not read the RC file. However you can set the configuration option of hive.default.rcfile.serde to ColumnarSerDe to maintain the interoperability between hive and impala.

Improvements in Metadata

After upgrading from CDH to CDP, the on-demand use_local_catalog mode is set to True by default on all the Impala coordinators so that the Impala coordinators pull metadata from catalogd and cache it locally. This reduces memory footprint on coordinators and automates the cache eviction.

In CDP, the catalog_topic_mode is set to minimal by default to enable on-demand metadata for all coordinators.

Recompute the Statistics

After migrating the workload from any CDH 5.x version to Cloudera Private Cloud Base 7.1, recompute the statistics for Impala. Even though CDH 5.x statistics are available after the upgrade, the queries do not benefit from the new features until the statistics are recomputed.

Mitigating Excess Network Traffic

The catalog metadata can become large and lead to excessive network traffic due to dissemination through the statestore. The --compact_catalog_topic flag was introduced to mitigate this issue by compressing the catalog topic entries to reduce their serialized size. This saves network bandwidth at the cost of a small quantity of CPU time. This flag is enabled by default.

Factors to Consider for Capacity Planning

Choosing the right size of your cloud environment before migrating your workload from CDH to CDP Public Cloud is critical for preserving performance characteristics. There are several factors from your query workload to consider when choosing the CDP capacity for your environment.

- Query memory requirements
- CPU utilizations
- Disk bandwidth
- Working set size
- Concurrent query execution

Before getting into sizing specifics, it is important to understand the core hardware differences between PC and on-prem hosts:

	CDH Host Recommendation	AWS R5D.4xlarge Instance
CPU cores	20-80	16
Memory	128GB min. 256GB+ recommended	128
Network	10Gbps min. 40Gbps recommended	Up to 10Gbps
Ephemeral Storage	12 x 2TB drives (1000MBps sequential)	2 x 300GB NVMe SSD (1100MBps sequential)
Network Storage	N/A	gp2 250 MB/s per volume

An R5D.4xlarge instance closely matches the CDH recommended CPU, memory, and bandwidth specs and so it is recommended as the instance type for CDP. However, AWS ephemeral storage cannot be used as primary

database storage since it is transient and lacks sufficient capacity. This core difference requires a different strategy for achieving good scan performance.

CDP Sizing and Scaling

Before migration, scaling and concurrency must be planned. In a public cloud environment, the ability to acquire better scaling and concurrency elastically in response to workload demand enables the system to operate at a lower cost than the maximum limits you plan for. If you configure your target environment to accommodate your peak workload as a constant default configuration, you might have cost overruns when system demand falls below that level.

In CDP, the T-Shirt size defines the number of executor instances for an individual cluster and hence determines memory limits and performance of individual queries. Conversely, the warehouse sizing parameter and auto-scaling determine how many clusters are allocated to support concurrent query execution.

Sizes	Number of Executors
X-Small	2
Small	10
Medium	20
Large	40

The T-shirt size must be at least large enough to support the highest memory usage by a single query. In most cases, the size will not need to be larger but may provide better data caching if there is commonality in working sets between queries. Increasing the T-Shirt size can directly increase single-user capacity and also increase multi-user capacity. This is because the additional memory and resources from the larger cluster allows larger datasets to be processed and can also support concurrent query execution by sharing resources. Choosing a size that is too small can lead to poor data caching, spilling of intermediate results, or memory paging. Choosing a size this is too large can incur excessive PC run cost due to idle executors.

One caveat to consider when choosing a T-shirt size based on existing hardware is what other processes are running on the same host in your on-prem environment. In particular HDFS or other locally hosted filesystems may be consuming significant resources. You may be able to choose a smaller size for CDP since these processes will be isolated in their own pod in the CDP environment. It may be helpful to look at CM per-process metrics to isolate impalad and Impala frontend java processes that CDP will put on executor instances and to aggregate these metrics across your cluster. The java processes can accumulate significant memory usage due to metadata caching.

Concurrency

The size of your target environment corresponds to the peak concurrency the system can handle. Concurrency is the number of queries that can be run at the same time.

Each executor group can run 12 queries concurrently and occasional peaks can be handled transparently using the auto scaling feature. An autoscaling leading to add one more executor group doubles the query conncurrency to 24. Scaling the warehouse by adding more clusters allows for additional concurrent queries to run but will not improve single-user capacity or performance. This is because executors from the additional clusters are private to that cluster. Concurrently executed queries will be routed to the different clusters and run independently. The number of clusters can be changed to match concurrent usage by changing the autoscaling parameters.

Caching Hot Dataset

Currently CDH supports caching mechanisms on the compute nodes to cache the working set that is read from remote filesystems, such as, remote HDFS data node, S3, ABFS, and ADLS. This offsets the input/output performance difference.

In CDP Public Cloud, frequently accessed data is cached in a storage layer on SSD so that it can be quickly retrieved for subsequent queries, which boosts the performance. Each executor can have upto 200 GB of cache. So a medium-size can keep 200 * 20 = 4 TB of data in cache. For columnar formats (for example, ORC) data in cache is decompressed but not decoded. If the expected size of the hot dataset is 6 TB, which requires approximately 30

executors, you can choose to overprovision (choose a large) to ensure full cache coverage; or underprovision (choose a medium) to reduce cost at the expense of a lower rate of finding in the cache (cache hit rate).

Note: Impala data cache uses LIRS based algorithm.

Planning Capacity Using WXM

You can generate a capacity plan for your target environment using WXM if you have it deployed in your environment. To build a custom cloud environment that meets your capacity requirements you must analyze your existing CDH architecture, understand your business needs, and generate a capacity plan.

There might be differences in how you should size your impala compute clusters (either in Datahub or in CDW service) because the compute node sizes (CPU and RAM) are different than what you are currently using in CDH. If you are currently using a 20 node CDH cluster, it does not necessarily mean that you will need a 20 node Datahub cluster or a 20 node Impala virtual warehouse in CDW.

Using WXM Functionality to Generate a Capacity Plan

Benefits of using WXM

- You can explore your cluster and analyze your workload before migrating the data. You can also identify Impala workloads that are good candidates for cloud migration.
- You also have the provision to optimize your workload before migrating them to CDP Public Cloud. This mitigates the risk of run-away costs in the cloud due to suboptimal workloads.
- You can generate a cloud-friendliness score for your workload to be migrated.
- You have an option to auto-generate capacity for your target environment.
- WXM in conjunction with the Replication manager automates the replication plan.

Prerequisites to Use WXM

Before you set up Cloudera Manager Telemetry Publisher service to send diagnostic data to Workload XM, you must ensure you have the correct versions of Cloudera Manager and CDH.

To use Workload XM with CDH clusters managed by Cloudera Manager, you must have the following versions:

- For CDH 5.x clusters:
  - Cloudera Manager version 5.15.1 or later
  - CDH version 5.8 or later
- For CDH 6.x clusters:
  - Cloudera Manager version 6.1 or later
  - CDH version 6.1 or later

Note: Workload XM is not available on Cloudera Manager 6.0 whether you are managing CDH 5.x or CDH 6.x clusters.

After you have verified that you have the correct versions of Cloudera Manager and CDH, you must configure data redaction and your firewall.

For information on configuring firewall, see Configuring a Firewall for Workload XM.

For information on redacting data before sending to WXM, see Redaction Capabilities for Diagnostic Data

Steps to Auto-generate Capacity Plan

If you have a Cloudera workload manager deployed in your environment, follow the high level steps to generate the capacity plan and migrate the Impala workload to the cloud.

1. On the Cloudera workload manager page, choose a cluster to analyze your data warehouse workloads. The Summary page for your workload view contains several graphs and tabs you can view to analyze. Using the Workloads View feature you can analyze workloads with much finer granularity. For example, you can analyze how queries that access a particular database or that use a specific resource pool are performing against SLAs. Or you can examine how all the queries are performing that a specific user sends to your cluster.
2. On the Data Warehouse Workloads View page you can choose an Auto-generated Workload Views by clicking Define New and choosing Select recommended views from the drop-down menu. Review the Criteria that are used to create the workload views, select the one from the auto-generated workload views that aligns with your requirements.
3. You can custom build the workload to be migrated by clicking on Define New and choosing Manually define view from the drop-down menu. You have the option to define a set of criteria that enables you to analyze a specific set of your workloads.
4. If you chose to custom build, once the custom build workload is generated, you are returned to the Data Warehouse Workloads page, where your workload appears in the list. Use the search bar to search for your workload and click on the workload to view the workload details.
5. The detail page for your workload view contains several graphs and tabs you can view to analyze. Review the workload and make sure that this is the workload you want to migrate to the cloud.
6. After you are satisfied with the workload you want to burst, click the Burst to Cloud option and select View Performance Rating Details.
7. Review the cloud performance rating details and make the call to proceed with the migration to cloud by clicking Start Burst to Cloud Wizard.
8. Burst to Cloud wizard walks you through the steps to generate the capacity plan and to replicate the workload you selected to the destination on the cloud.

Performance Differences between CDH and CDP

Assess the performance changes this migration can bring. If you are planning to migrate the current Impala workload to Public Cloud, conduct a performance impact analysis to evaluate how this migration will affect you.

IO Performance Considerations

On-prem CDH hosts often have substantial IO hardware attached to support large scan operations on HDFS, potentially providing 10s of GB per seconds of bandwidth with many SSD devices and dedicated interconnects. Due to the transient nature and cost structure of cloud instances, such a model is not practical for primary storage in CDP.

Like many AWS database offerings, HFS in CDP uses EBS volumes for persistence. The EBS gp2 has a bandwidth limit of 250MB/sec/volume. In addition, EBS may be throttled to zero throughput for extended durations if bandwidth exceeds thresholds. Because of these limitations, it is not practical in many cases to rely on direct IO to EBS for performance. EBS is also routed over shared network hardware and may have additional performance limitations due to redundancy.

To mitigate the PC IO bandwidth discrepancy, ephemeral storage is relied on heavily for caching working sets. While this is existing Impala behavior carried over from CDH, the penalty for going to primary storage is much higher so more data must be cached locally to maintain equivalent performance. Since ephemeral storage is also used for spilling of intermediate results, it is import to avoid excess spilling which could compete for bandwidth.