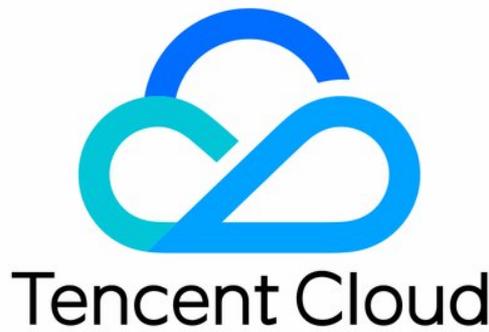


Elastic MapReduce

Best Practice



Copyright Notice

©2013–2024 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Trademark Notice

 Tencent Cloud

This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Best Practice

Operational Practices of EMR on CVM

Practical Investigation into the Reasons for Non-Execution of Automatic Scaling Rules

Data Migration

HDFS Data Migration Using COS

HDFS Data Migration Using DistCp

Practical HBase Data Migration

Practical Hive Data Migration

Customized Scaling Best Practices

Principles for triggering execution of scaling rules

Optimal Practices for Setting Scaling Policies

Best Practices for Setting Time-based Scaling Rules

Best Practices for Setting Load Scaling Rules

Best Practice

Operational Practices of EMR on CVM

Practical Investigation into the Reasons for Non-Execution of Automatic Scaling Rules

Last updated: 2024-01-12 11:22:45

1. Elastic resources have reached the minimum instance count. If further reduction is required, consider adjusting the minimum instance count.

Reason for occurrence: A reduction rule has been triggered, but the current number of elastic nodes is less than the minimum node count.

Solution: If further reduction is necessary, reset the minimum node count.
2. Elastic resources have exceeded the maximum instance count. If further expansion is required, consider adjusting the maximum instance count.

Reason for occurrence: An expansion rule has been triggered, but the current number of elastic nodes equals the maximum node count.

Solution: If further expansion is necessary, reset the maximum node count.
3. Scaling specifications have not been set, thus expansion is not possible. Consider adding specifications and retrying.

Reason for occurrence: As shown above, an automatic scaling rule has been triggered, but no node specifications have been added in the **Console > Auto Scaling > Scaling Specification Management** section.

Solution: Click on **Add Specification** in the upper right corner and select the desired node specification.
4. Resources are insufficient. Consider switching to a specification with ample resources or [submit a ticket](#) to contact us.

Reason for occurrence: An expansion rule has been triggered, but the resources of the selected machine type in the current availability zone are insufficient.

Solution: Reset to a node specification with sufficient resources.
5. The current retry time setting is too short. It is recommended to extend the expiration retry time.

Reason for occurrence: From the trigger time to the expiration retry time of the time scaling rule, there are other automatic scaling processes in the cluster, causing the

current time scaling rule not to be executed.

Solution: You can edit the rule to appropriately extend the expiration retry time, ensuring the rule can be executed.

6. The account balance is insufficient, preventing expansion.

Reason for occurrence: An expansion rule has been triggered, but there is insufficient balance when placing the order.

Solution: Go to the [Cost Center](#) to recharge the account.

7. Currently, there are no elastic resources that meet the conditions for scaling down.

Reason for occurrence: A scaling down rule has been triggered, but there are currently no elastic node resources or all nodes are set to be destroyed on schedule.

Solution: If you need to continue scaling down nodes scheduled for destruction, you can choose to manually scale down.

8. The cluster status is not in a state that allows expansion, thus expansion is not possible.

Reason for occurrence: An expansion rule has been triggered, but the current cluster is in a non-running state such as installing components or expanding, and cannot be expanded.

Solution: You can manually expand or edit the rule to appropriately extend the expiration retry time, ensuring the rule can be executed.

9. The cluster is in the expansion cooling-off period and temporarily cannot trigger expansion. It is recommended to adjust the cooling-off time of the scaling rule.

Reason for occurrence: An expansion rule has been triggered, but the cluster is currently in the cooling-off period of other scaling operations and cannot be executed.

Solution: You can appropriately shorten the cooling-off time of other rules, or extend the expiration retry time of this expansion rule.

Data Migration

HDFS Data Migration Using COS

Last updated: 2024-01-02 14:21:08

If you need to migrate your HDFS raw data to EMR, you can achieve using either of the following: migrate data with Tencent Cloud Object Storage (COS) service as a transfer stop; migrate data with DistCp, a built-in tool of Hadoop for large inter/intra-cluster copying. This document describes how to migrate data with the first method.

Non-HDFS Original Data

If your original data is not an HDFS file but rather from another formatted source, you can upload it to COS using either the COS web console or the API provided by COS. Afterward, the uploaded data can be analyzed in the EMR cluster. Please refer to [Local Data Migration to COS](#) for more information about data transfer via COS.

Migrating an HDFS file

1. Acquire the COS migration tool.

[Obtain the migration tool](#), for more migration tools, please refer to the [Tool Overview](#).

2. Tool configuration.

The configuration files are all placed in the conf directory of the tool directory, copy the core-site.xml of the HDFS cluster that needs to be synchronized to the conf, which includes the configuration information of the NameNode, edit the configuration file cos_info.conf, including appid, bucket, region, and key information.

Caution

- It's advised that users use the sub-account key, abide by the [principle of minimal authority explanation](#), to prevent the leakage of resources outside of the intended storage bucket or object.
- If you absolutely must use a permanent key, it's recommended to abide by the [principle of minimal authority explanation](#) to limit the scope of permissions for the permanent key.

Explanation of Command Parameters:

```
-ak <ak> the cos secret id // User's SecretId. It is recommended to use a sub-account key, adhering to the principle of least privilege
```

to minimize usage risks. For obtaining a sub-account key, refer to:
<https://cloud.tencent.com/document/product/598/37140>

```

-appid,--appid <appid>          the cos appid
-bucket,--bucket <bucket_name>  the cos bucket name
-cos_info_file,--cos_info_file <arg>  the cos user info config default is
./conf/cos_info.conf
-cos_path,--cos_path <cos_path>    the absolute cos folder path
-h,--help                        print help message
-hdfs_conf_file,--hdfs_conf_file <arg>  the hdfs info config default is ./conf/core-
site.xml
-hdfs_path,--hdfs_path <hdfs_path>  the hdfs path
-region,--region <region>        the cos region. legal value cn-south, cn-east,
cn-north, sg
-sk <sk>                          the cos secret key // User's SecretKey. It is
recommended to use a sub-account key, adhering to the principle of least privilege
to minimize usage risks. For obtaining a sub-account key, refer to:
https://cloud.tencent.com/document/product/598/37140
-skip_if_len_match,--skip_if_len_match skip upload if hadoop file length match cos

```

3. Executing Migration:

```

# All operations must be performed within the tool directory. If both the
configuration file and command line parameters are set, the command line
parameters will take precedence.
./hdfs_to_cos_cmd -h
# Copying from HDFS to COS (If a file already exists on COS, it will be overwritten)
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
# Copying from HDFS to COS, if the length of the file to be copied and COS is
consistent, the upload is ignored (applicable for re-copying after the initial copy).
# Here, only the length is evaluated, as calculating the file summary on Hadoop
would be too costly.
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -
skip_if_len_match
# Setting parameters entirely through the command line.
./hdfs_to_cos_cmd -appid 1252xxxxxx -ak
AKIDVt55xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -sk
KS08jDVbVEIxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -bucket test -cos_path /hdfs
-hdfs_path /data/data -region cn-south -hdfs_conf_file
/home/hadoop/hadoop-2.8.1/etc/hadoop/core-site.xml

```

4. Upon executing the verification command, the following logs are outputted:

```

[Folder Operation Result : [ 53(sum)/ 53(ok) / 0(fail)]]
[File Operation Result: [22(sum)/ 22(ok) / 0(fail) / 0(skip)]]

```

[Used Time: 3 s]

- Sum denotes the total number of files that need to be migrated.
- Ok signifies the number of files successfully migrated.
- Fail represents the number of files for which the migration failed.
- Skip denotes the count of files skipped due to the uploaded file and the identically named file having the same length, after adding the skip_if_len_match parameter.

You may also sign in to the COS console to determine if the data has been rightly migrated. For the guidelines on using COS, see [Getting Started](#).

FAQs

- Please ensure that the configuration information you provide, including the appID, key information, bucket, and region, is accurate, and that the machine's time aligns with Beijing time (a discrepancy of about 1 minute is normal). If the difference is substantial, please adjust the machine's time.
- Please ensure that the machine where the copy program is located can also connect to the DateNode. Although the NameNode can be connected via an external IP, the DateNode machine where the block is located has an internal IP and cannot be connected. Therefore, it is recommended to execute the synchronization program on a node of Hadoop to ensure access to both the NameNode and DateNode.
- Regarding permission issues, use the current account to download files using Hadoop commands to check for normalcy, then use the synchronization tool to sync data on Hadoop.
- For files that already exist on COS, they are overwritten by default unless the user explicitly specifies `-skip_if_len_match`, in which case the upload is skipped if the file lengths match.
- The COS path is considered a directory, and all files copied from HDFS will ultimately be stored in this directory.

HDFS Data Migration Using DistCp

Last updated: 2024-01-02 14:21:19

Should you require the migration of raw data from your proprietary HDFS to Tencent Cloud EMR, there are two methods available for data migration. The first involves the use of Tencent Cloud Object Storage (COS) as a data transfer intermediary, while the second employs the Hadoop's inherent file migration tool, DistCp, for data migration. This document primarily focuses on data migration via DistCp.

DistCp (distributed copy) is a file migration tool inherent to Hadoop. It utilizes MapReduce to execute its distribution, error handling and recovery, and reporting functions. It expands the list of files and directories as input for mapping tasks, with each task duplicating the partitions of files specified in the source list. The use of DistCp necessitates the establishment of network intercommunication between the self-built cluster and the EMR cluster.

The steps for data migration using DistCp are as follows:

Step 1: Establish Network Connectivity

Migration of Locally Self-Built HDFS Files to EMR

The migration of locally self-built HDFS files to the EMR cluster necessitates the establishment of a dedicated network line. Assistance can be sought from the development team to facilitate this process.

Migration of Self-Built HDFS Files on CVM to EMR

- If the network to which the CVM belongs and the network to which the EMR cluster belongs are under the same VPC, files can be freely transferred.
- If the network to which the CVM belongs and the network to which the EMR cluster belongs are under different VPCs, it is necessary to establish network connectivity using a peering connection.

Utilizing Peering Connection

Subnet 1: Subnet A 192.168.1.0/24 in VPC1 of Guangzhou.

Subnet 2: Subnet B 10.0.1.0/24 in VPC2 of Beijing.

1. Log in to the [Private Network Console – Peering Connection](#), select Guangzhou as the region at the top of the list, choose the private network VPC1, and then click on **+Create**.

Peering connections South China (Guangzhou) All VPCs Help of peering connection

To get notified about abnormal peer connection behaviors instantly, please [Configure alarms](#).

+ Create Search by peering connect

ID/Name	Mo...	Status	Local region	Local VPC	Peer region	Peer account	Peer VPC	Band...	Servi...	Billing mode	Operation
		Connected	South China (Guangzhou)		South China (Guangzhou)	My account		Unlimited	Gold	Free	Delete

2. Navigate to the page for establishing a peering connection.

- **Name:** The name of the peering connection, for instance, PeerConn.
- **Local Region:** Fill in the local region, for example, Guangzhou.
- **Local Network:** Fill in the local network, for instance, VPC1.
- **Peer Account Type:** Fill in the account to which the peer network belongs. If the Guangzhou and Beijing networks are under the same account, select **My Account**. If they are not under the same account, choose **Other Account**.

Note

If both the local network and the peer network are in the same region, for example, Guangzhou, communication is free and there is no need to select **Bandwidth Cap**. If they are not in the same region, charges will apply, and the bandwidth cap is optional.

- **Peer Region:** Fill in the peer region, for example, Beijing.
- **Peer Network:** Fill in the peer network, for example, VPC2.

Create peering connection ×

Name

Local region

Local network

Peer account type My account Other accounts

Network type Public Cloud VPC BM Virtual Private Cloud

Peer region

Peer network

Bandwidth cap Unlimited

Billing method A 5 Gbps free tier is applied for intra-region bandwidth. And for cross-region bandwidth, [No free tier](#) , [Billing description](#)

- When connecting private networks within the same account, the peer connection takes effect immediately after creation. When creating a peer connection with a private network of another account, it takes effect after the peer accepts this connection. Refer to [Creating Peer Connection Communication within the Same Account](#) and [Creating Peer Connection Communication across Accounts](#) .
- Configure the local and peer route tables for the peer connection.
 - Log in to the [Private Network Console](#) , click on **Subnet** in the left directory to enter the management page. Click on the associated route table ID of the subnet specified by the peer connection's local end (for example, the subnet VPC1 in Guangzhou) to enter the route table details page.

Subnet Guangzhou 68 All VPCs Help of Subnet

Create Please enter the Subnet Q ⚙ ⌵

ID/Name	Network	CIDR	IPv6 CIDR	Availability zo...	Associated route t...	CVM	Available IPs	Default subnet	Creation time	Tags	Operation
				Guangzhou Zone 7		12	234	No	2023-11-28 11:11:44		Delete More

- Click on **Add Routing Policy**.

Add routing policy Export Enable Disable Search by the destination Q

<input type="checkbox"/> Destination	Next hop type	Next hop	Remark	Enable routing	Route status in CCN	Operation
	LOCAL	Local	Delivered by default, indicates that CVMs in the VPC are interconnected.	<input checked="" type="checkbox"/>	-	Publish to CCN
	LOCAL	Local	Delivered by default, indicates that the container IP ranges interconnect with CVMs in the VPC.	<input checked="" type="checkbox"/>	-	Publish to CCN

Total items: 2 20 / page 1 / 1 page

- In the destination end, fill in the peer CIDR (for example, the CIDR of VPC2 in Beijing is 10.0.1.0/24), choose **Peer Connection** for the next hop type, and select the established peer connection (PeerConn) for the next hop.

Add a route ✕

[Routing policies control the traffic flow in the subnet. For details, please see Configuring Routing Policies.](#)

Destination	Next hop type	Next hop	Remark	Operation
<input type="text" value="such as 10.0.0.0/16"/>	<input type="text" value="Public IP of CVM"/>	<input type="text" value="Public IP of CVM"/>	<input type="text"/>	<input type="button" value="x"/>

[+ New line](#)

- The above steps are for configuring the route table from Guangzhou VPC1 to Beijing VPC2. It is also necessary to configure the route from Beijing VPC2 to Guangzhou VPC1, the configuration process is the same.
- Upon completion of the route table configuration, different private network segments can communicate with each other.

Step 2: Execute Copy

```
# Inter-cluster Copying, copying a folder to another cluster
hadoop distcp hdfs://nn1:9820/foo/bar hdfs://nn2:9820/bar/foo
```

```
# Specifying File Copy
```

```
hadoop distcp hdfs://nn1:9820/foo/a hdfs://nn1:9820/foo/b hdfs://nn2:9820/bar/foo
```

```
# If there are too many specified files, the -f parameter can be used.
```

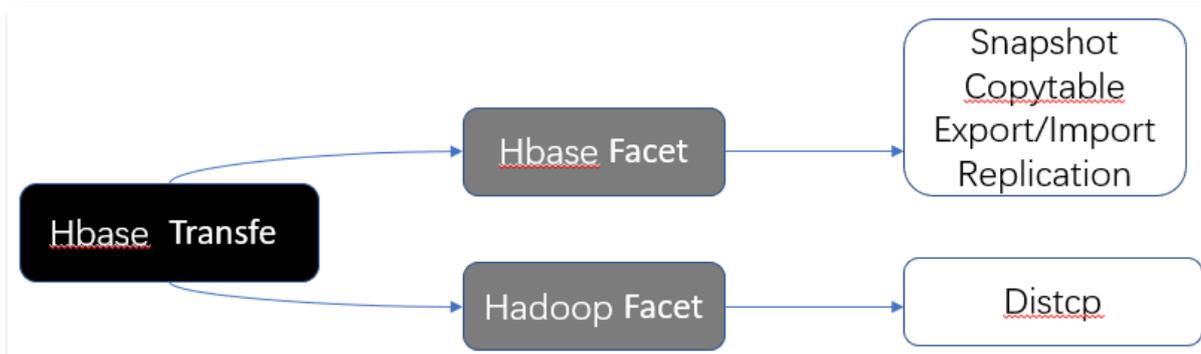
Note

- For the aforementioned command, it is imperative that the source and destination versions are identical.
- Should another client still be writing to the source file, the copy may fail; if a file is being copied to the destination, attempts to overwrite that file will fail; if the source file is moved prior to being copied, the copy will fail, resulting in a `FileNotFoundException` error message.

Practical HBase Data Migration

Last updated: 2024-01-02 14:21:29

HBase tables, constructed atop Hadoop HDFS, can be migrated from two perspectives: utilizing the migration method of Hadoop HDFS's distcp and employing the migration tools provided at the HBase table structure level.



As depicted above, there are various strategies for HBase migration, among which the Snapshot-based migration method is the recommended approach.

Migration of HBase Based on Snapshot

Note:

The following steps are all performed as a Hadoop user.

1. Establish a table on the original target cluster that mirrors the structure of the existing table.

```
$ hbase shell
hbase> create 'myTable', 'cf1', 'cf2'
```

2. Initialize the table data on the original cluster.

```
$ hbase shell
hbase> put 'myTable', 'row1', 'cf1:a', 'value1'
hbase> put 'myTable', 'row2', 'cf2:b', 'value2'

hbase> scan 'myTable'
ROW                                COLUMN+CELL
row1                                column=cf1:a, timestamp=2023-08-09T16:43:10.024, value=value1      row2
column=cf2:b, timestamp=2023-08-09T16:43:20.036, value=value2
```

3. Create a snapshot in the original cluster using `hbase shell` .

```
$ hbase shell
hbase> snapshot 'myTable', 'myTableSnapshot'
```

Here, `'myTable'` is the table name in HBase, and `'myTableSnapshot'` is the name of the snapshot. After creation, you can use `list_snapshots` to confirm success, or use `delete_snapshot` to remove the snapshot.

```
hbase> delete_snapshot 'myTableSnapshot'
```

4. Export the snapshot from the source cluster to the target cluster.

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot
myTableSnapshot -copy-to hdfs://10.0.0.38:4007/hbase/snapshot/myTableSnapshot
```

Here, `10.0.0.38:4007` is the `$activeip:$rpcport` of the target cluster. When exporting a snapshot, the system level will initiate a mapreduce task. You can add `-mappers 16 -bandwidth 200` afterwards to specify the mapper and bandwidth. The 200 here refers to 200MB/sec.

5. Restore the snapshot to the target HDFS in the target cluster by executing the following command.

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot
myTableSnapshot -copy-from /hbase/snapshot/myTableSnapshot -copy-to /hbase/
```

6. Restore the corresponding HBase table and data from HDFS in the target cluster.

```
hbase> disable "myTable"
hbase> restore_snapshot 'myTableSnapshot'
hbase> enable 'myTable'
```

7. Finally, you can conduct tests through simple HBase table operations.

Practical Hive Data Migration

Last updated: 2024-01-02 14:21:41

The migration of Hive involves two components: data migration and metadata migration. The primary storage for Hive table data is HDFS, hence the data migration predominantly occurs at the HDFS layer. Hive's metadata is primarily stored in relational databases, which can be seamlessly migrated to TencentDB in the cloud, ensuring high availability.

Migration of Hive Metadata

1. Dump the source Hive metadata database.

```
mysqldump -hX.X.X.X -uroot -pXXXX --single-transaction --set-gtid-purged=OFF
hivemetastore > hivemetastore-src.sql
# If MySQL data has not enabled GTID, please remove --set-gtid-purged=OFF from
the command line.
# X.X.X.X represents the database server address.
# XXXX represents the database password.
# If the database user is not root, please use the correct username.
# 'hivemetastore' is the name of the Hive metadata database.
```

2. Confirm the default storage path of Hive table data in HDFS in the target cluster.

The default storage path of Hive table data in HDFS is specified by the

`hive.metastore.warehouse.dir` configuration item in `hive-site.xml`. If the storage path of the Hive table in HDFS in the target cluster needs to be consistent with the path of the Hive table in the source cluster, you can refer to the following example to modify the configuration file. For instance, `hive.metastore.warehouse.dir` in the source cluster's `hive-site.xml` has the following value.

```
<property>
<name>hive.metastore.warehouse.dir</name>
<value>/apps/hive/warehouse</value>
</property>
```

The value of `hive.metastore.warehouse.dir` in the target cluster's `hive-site.xml` is as follows.

```
<property>
<name>hive.metastore.warehouse.dir</name>
<value>/usr/hive/warehouse</value>
</property>
```

If the storage location of the Hive table in HDFS in the target cluster remains consistent with the source cluster Hive, then modify `hive.metastore.warehouse.dir` in the target's `hive-site.xml` to be:

```
<property>
  <name>hive.metastore.warehouse.dir</name>
  <value>/apps/hive/warehouse</value>
</property>
```

3. Confirm the target Hive metadata `SDS.LOCATION` and `DBS.DB_LOCATION_URI` fields. Obtain the current `SDS.LOCATION` and `DBS.DB_LOCATION_URI` fields through the following query.

```
SELECT DB_LOCATION_URI from DBS;
SELECT LOCATION from SDS;
```

The query results would resemble the following:

```
mysql> SELECT LOCATION from SDS;
+-----+
| LOCATION |
+-----+
| hdfs://HDFS2648/usr/hive/warehouse/hitest.db/t1 |
| hdfs://HDFS2648/usr/hive/warehouse/wyp |
+-----+
mysql> SELECT DB_LOCATION_URI from DBS;
+-----+
| DB_LOCATION_URI |
+-----+
| hdfs://HDFS2648/usr/hive/warehouse |
| hdfs://HDFS2648/usr/hive/warehouse/hitest.db |
+-----+
```

Wherein, `hdfs://HDFS2648` is the default file system name for HDFS, specified by `fs.defaultFS` in `core-site.xml`.

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://HDFS2648</value>
</property>
```

`/usr/hive/warehouse` is the default storage path for Hive tables in HDFS, and is also the value specified by `hive.metastore.warehouse.dir` in `hive-site.xml`. Therefore, we need to modify the `SDS.LOCATION` and `DBS.DB_LOCATION_URI` fields in the source hive metadata SQL file. Ensure that these two fields in the imported Hive metadata database use the correct path. The following `sed` command can be used to modify the SQL file in bulk.

```
Replace IP: sed -i 's/oldcluster-ip:4007/newcluster-ip:4007/g' hivemetastore-src.sql
Replace defaultFS: sed -i 's/old-defaultFS/new-defaultFS/g' hivemetastore-src.sql
```

Note

If components such as Kudu, Hbase are used, and Metastore is used as the metadata service, it is also necessary to change the corresponding location field in the target Hive metadata.

4. Cease the target Hive services MetaStore, HiveServer2, WebHcatalog.
5. Backup the target Hive metadata database.

```
mysqldump -hX.X.X.X -uroot -pXXXX --single-transaction --set-gtid-purged=OFF
hivemetastore > hivemetastore-target.sql
# If MySQL data has not enabled GTID, please remove --set-gtid-purged=OFF from
the command line.
# X.X.X.X represents the database server address.
# XXXX represents the database password.
# If the database user is not root, please use the correct username.
# hivemetastor is the name of the Hive metadata database.
```

6. Drop/Create the target Hive metadata.

```
mysql> drop database hivemetastore;
mysql> create database hivemetastore;
```

7. Import the source Hive metadata database into the target database.

```
mysql -hX.X.X.X -uroot -pXXXX hivemetastore < hivemetastore-src.sql
# X.X.X.X represents the database server address.
# XXXX represents the database password.
# If the database user is not root, please use the correct username.
# hivemetastor is the name of the Hive metadata database.
```

8. Upgrade Hive metadata.

If the target and source Hive versions are identical, this step can be bypassed; otherwise, query the Hive version in the source cluster and target cluster respectively.

```
hive --service version
```

The upgrade scripts for Hive are stored in the

```
/usr/local/service/hive/scripts/metastore/upgrade/mysql/
```

 directory.

Hive does not support cross-version upgrades, for instance, upgrading Hive from 1.2 to 2.3.0 requires sequential execution:

```
upgrade-1.2.0-to-2.0.0.mysql.sql -> upgrade-2.0.0-to-2.1.0.mysql.sql -> upgrade-2.1.0-to-2.2.0.mysql.sql -> upgrade-2.2.0-to-2.3.0.mysql.sql
```

The primary operations of the upgrade script include creating tables, adding fields, and modifying content. If the table or field already exists, exceptions regarding existing fields during the upgrade process can be ignored. For instance, when upgrading Hive from 2.3.3 to 3.1.1.

```
mysql> source upgrade-2.3.0-to-3.0.0.mysql.sql;
mysql> source upgrade-3.0.0-to-3.1.0.mysql.sql;
```

9. If there are Phoenix tables in the source Hive, modify the Zookeeper address of the Phoenix tables in the target Hive metadata.

Obtain the `phoenix.zookeeper.quorum` configuration of the Phoenix tables through the following query.

```
mysql> SELECT PARAM_VALUE from TABLE_PARAMS where PARAM_KEY =
'phoenix.zookeeper.quorum';
+-----+
| PARAM_VALUE |
+-----+
| 172.17.64.57,172.17.64.78,172.17.64.54 |
+-----+
```

View the Zookeeper address of the target cluster, that is, the value specified by `hbase.zookeeper.quorum` in the `hive-site.xml` configuration file.

```
<property>
  <name>hbase.zookeeper.quorum</name>
```

```
<value>172.17.64.98:2181,172.17.64.112:2181,172.17.64.223:2181</value>  
</property>
```

Change the Zookeeper address of the Phoenix tables in the target Hive metadata to the Zookeeper address of the target cluster.

```
mysql> UPDATE TABLE_PARAMS set PARAM_VALUE =  
'172.17.64.98,172.17.64.112,172.17.64.223' where PARAM_KEY =  
'phoenix.zookeeper.quorum';
```

10. Inspect the case format of the table names in the target Hive metadata. Refer to the following example to change all lowercase table names to uppercase:

```
alter table metastore_db_properties rename to METASTORE_DB_PROPERTIES;
```

11. Initiate the target Hive services MetaStore, HiveServer2, and WebHcatalog.

12. Finally, verification can be conducted through a simple Hive SQL query.

Customized Scaling Best Practices

Principles for triggering execution of scaling rules

Last updated: 2024-01-02 14:21:58

Upon execution of the expansion rule, the principle of preset resource addition is applied.

Each cluster can be configured with up to five scaling specifications. When the expansion rule is triggered, the expansion will be carried out according to the priority of the specifications. If the quantity of high-priority specifications is insufficient, the expansion will be supplemented by a combination of secondary priority resource specifications and high-priority specifications to supplement computing resources. The execution order for pay-as-you-go and bidding instances remains the same.

- **When resources are abundant, the order of priority is as follows: 1 > 2 > 3 > 4 > 5.**

📌 For instance:

Given five preset specifications and sufficient resources, when the expansion rule is triggered and requires the expansion of 10 nodes, the expansion will proceed with 10 nodes according to the first specification, while the remaining preset specifications will not be selected.

- **In the event of resource scarcity, the following sequence is adhered to: 1+2 > 1+2+3 > 1+2+3+4 > 1+2+3+4+5.**

📌 For instance:

Assuming there are 8 nodes for preset specification 1, 4 nodes for specification 2, and 3 nodes for specification 3, when the expansion rule is triggered and requires the expansion of 13 nodes, the expansion will proceed in the following order: 8 nodes for specification 1, 4 nodes for specification 2, and 1 node for specification 3.

- **In the event of a resource specification shortage, for instance, if specification 2 is out of stock, the following sequence is adhered to: 1+3 > 1+3+4 > 1+3+4+5.**

📌 For instance:

- Assuming there are 8 nodes for preset specification 1, no nodes for specification 2 due to unavailability, and 3 nodes for specification 3, when the expansion rule is

triggered and requires the expansion of 10 nodes, the expansion will proceed in the following order: 8 nodes for specification 1, none for specification 2, and 2 nodes for specification 3.

- Assuming there are 8 nodes for preset specification 1, with no nodes available for the remaining preset specifications, when the expansion rule is triggered and requires the expansion of 10 nodes, the expansion rule will be activated, expanding 8 nodes under specification 1, resulting in a partial success of the expansion.

Expansion methods include the selection of nodes, memory, and core count. These three methods only support the input of non-zero integer values. When the method of core count or memory is selected, the expansion ensures the maximum computing power for the conversion of the expansion node quantity.

ⓘ For instance:

- When expanding by core count and the expansion is set to 10 cores, but the specification expands in priority order to 8 cores, the rule will be triggered to expand to two 8-core nodes.
- When expanding by memory and the expansion is set to 20G, but the specification expands in priority order to 16G, the rule will be triggered to expand to two 16G nodes.

Principles of elastic node reduction during the execution of reduction rules.

Elastic nodes expanded by the auto-scaling feature will, when the reduction rule is triggered, execute according to the "expand first, reduce later; expand later, reduce first" principle. Non-elastic nodes will not trigger reduction actions when the reduction rule is triggered, as non-elastic nodes only support manual reduction.

⚠ Note:

Scheduled node destruction will not be constrained by the "expand first, reduce later; expand later, reduce first" principle or the cluster's "minimum node count" rule. Reduction can be executed as soon as the time arrives, and by default, graceful reduction is within a 2-hour range.

Reduction methods: Supports the selection of nodes, memory, and core count; these three methods only support the input of non-zero integer values. When the method of core count and memory is selected, reduction ensures normal business operations by reducing the

minimum number of nodes. The number of nodes is converted according to time in reverse order, ensuring at least one node is reduced.

ⓘ For instance:

- When reducing by core count, if the reduction is set to 20 cores, and the reduction rule is triggered, the cluster will have elastic nodes in reverse chronological order, specifically 3 nodes with 8 cores and 16G each, and 2 nodes with 4 cores and 8G each. The reduction will successfully reduce 2 nodes with 8 cores and 16G each.
- When reducing by memory, if the reduction is set to 30G, and the reduction rule is triggered, the cluster will have elastic nodes in reverse chronological order, specifically 3 nodes with 8 cores and 16G each, and 2 nodes with 4 cores and 8G each. The reduction will successfully reduce 1 node with 8 cores and 16G.

Principle of executing scaling rules in sequence

Supports the setting of mixed elastic rules for time scaling and load scaling. The rule trigger follows the principle of "first triggered, first executed; simultaneously triggered, executed according to rule priority order". The rule status is used to mark whether the rule is enabled, with the default being enabled. When the rule operation is not needed but the rule configuration is still desired to be retained, the rule status can be set to closed.

- Settings for scaling based solely on load
 - 1.1 Adheres to the principle of "first triggered, first executed; simultaneously triggered, executed according to rule priority order", such as: **1>2>3>4>5**.
 - 1.2 A single load scaling rule supports the setting of multiple metrics. The rule is triggered when all metrics meet the conditions.
 - 1.3 Supports monitoring cluster load changes within a specified time period and setting load scaling to take effect.
- Settings for scaling based solely on time
 - 1.1 Adheres to the principle of "first triggered, first executed; simultaneously triggered, executed according to rule priority order", such as: **1>2>3>4>5**.
 - 1.2 The rule is executed repeatedly. If the rule expires, its status will become invalid and it will be in a closed state. There will be an alarm notification before the expiration. For details, please refer to [Alarm Configuration](#).
- Settings for scaling based on a combination of load and time

Adheres to the principle of "first triggered, first executed; simultaneously triggered, executed according to rule priority order", such as: **1>2>3>4>5**.

Correspondence of queue load metrics

Category	Aspect	EMR Automatic Scaling Metrics	Description
AvailableV Cores	root	AvailableVCores#root	Quantity of available virtual cores in the Root queue
	root.default	AvailableVCores#root.default	Quantity of available virtual cores in the root.default queue
	Custom Sub-Queue	For instance: AvailableVCores#root.test	Quantity of available virtual cores in the root.test queue
PendingV Cores	root	PendingVCores#root	Virtual cores awaiting availability in the Root queue
	root.default	PendingVCores#root.default	Virtual cores awaiting availability in the root.default queue
	Custom Sub-Queue	For instance: PendingVCores#root.test	Virtual cores awaiting availability in the root.test queue
Available MB	root	AvailableMB#root	Quantity of available memory (MB) in the Root queue
	root.default	AvailableMB#root.default	Quantity of available memory (MB) in the root.default queue
	Custom Sub-Queue	For instance: AvailableMB#root.test	Quantity of available memory (MB) in the root.test queue
PendingM B	root	PendingMB#root	Quantity of memory (MB) awaiting availability in the Root queue
	root.default	PendingMB#root.default	Quantity of memory (MB) awaiting availability in the root.default queue
	Custom Sub-Queue	For instance: PendingMB#root.test	Quantity of memory (MB) awaiting availability in the root.test queue
Available	Cluste	AvailableMemPer	Percentage of remaining memory

MemPercentage	rs	centage	
ContainerPendingRatio	Clusters	ContainerPendingRatio	Ratio of the number of containers awaiting allocation to the number of allocated containers
AppsRunning	root	AppsRunning#root	Number of tasks running in the root queue
	root.default	AppsRunning#root.default	Number of tasks running in the root.default queue
	Custom Sub-Queue	For instance: AppsRunning#root.test	Number of tasks running in the root.test queue
AppsPending	root	AppsPending#root	Number of tasks pending in the root queue
	root.default	AppsPending#root.default	Number of tasks pending in the root.default queue
	Custom Sub-Queue	For example: AppsPending#root.test	Number of tasks pending in the root.test queue
PendingContainers	root	PendingContainers#root	Number of containers awaiting allocation in the root queue
	root.default	PendingContainers#root.default	Number of containers awaiting allocation in the root.default queue
	Custom Sub-Queue	For example: PendingContainers#root.test	Number of containers awaiting allocation in the root.test queue
AllocatedMB	root	AllocatedMB#root	Amount of memory allocated in the root queue
	root.default	AllocatedMB#root.default	Amount of memory allocated in the root.default queue
	Custom	For instance: AllocatedMB#root	Amount of memory allocated in the root.test queue

	Sub-Queue	.test	
Allocated MB	root	AllocatedVCores#root	Number of virtual cores allocated in the root queue
	root.default	AllocatedVCores#root.default	Number of virtual cores allocated in the root.default queue
	Custom Sub-Queue	For instance: AllocatedVCores#root.test	Number of virtual cores allocated in the root.test queue
Reserved VCores	root	ReservedVCores#root	Number of virtual cores reserved in the root queue
	root.default	ReservedVCores#root.default	Number of virtual cores reserved in the root.default queue
	Custom Sub-Queue	For instance: ReservedVCores#root.test	Number of virtual cores reserved in the root.test queue
Allocated Containers	root	AllocatedContainers#root	Number of containers allocated in the root queue
	root.default	AllocatedContainers#root.default	Number of containers allocated in the root.default queue
	Custom Sub-Queue	For instance: AllocatedContainers#root.test	Number of containers allocated in the root.test queue
Reserved MB	root	ReservedMB#root	Amount of memory reserved in the root queue
	root.default	ReservedMB#root.default	Amount of memory reserved in the root.default queue
	Custom Sub-Queue	For example: ReservedMB#root.test	Amount of memory reserved in the root.test queue
AppsKilled	root	AppsKilled#root	Number of tasks terminated in the root

			queue
	root.default	AppsKilled#root.default	Number of tasks terminated in the root.default queue
	Custom Sub-Queue	For instance: AppsKilled#root.test	Number of tasks terminated in the root.test queue
AppsFailed	root	AppsFailed#root	Number of failed tasks in the root queue
	root.default	AppsFailed#root.default	Number of failed tasks in the root.default queue
	Custom Sub-Queue	For instance: AppsFailed#root.test	Number of failed tasks in the root.test queue
AppsCompleted	root	AppsCompleted#root	Number of tasks completed in the root queue
	root.default	AppsCompleted#root.default	Number of tasks completed in the root.default queue
	Custom Sub-Queue	For instance: AppsCompleted#root.test	Number of tasks completed in the root.test queue
AppsSubmitted	root	AppsSubmitted#root	Number of tasks submitted in the root queue
	root.default	AppsSubmitted#root.default	Number of tasks submitted in the root.default queue
	Custom Sub-Queue	For example: AppsSubmitted#root.test	Number of tasks submitted in the root.test queue
AvailableVCoresPercentage	Clusters	AvailableVCoresPercentage	Percentage of available virtual cores within the cluster
MemPendingRatio	root	MemPendingRatio#root	Percentage of memory awaiting availability in the root queue

	root.default	MemPendingRatio#root.default	Percentage of memory awaiting availability in the root.default queue
	Custom Sub-Queue	For instance: MemPendingRatio#root.test	Percentage of memory awaiting availability in the root.test queue

Optimal Practices for Setting Scaling Policies

Best Practices for Setting Time-based Scaling Rules

Last updated: 2024-01-02 14:22:16

In accordance with the discernible peaks and troughs within a certain business cycle, select the execution frequency of either repeated execution or one-time execution, and configure the rules for scaling up and scaling down. When opting for repeated execution, the rule's validity period can be configured, setting a deadline for when the rule takes effect. Beyond this validity period, the scaling rule will no longer be triggered.

For instance:

Should your business experience an increase starting at 10 PM each day, and a decrease beginning at 6 AM, with an expected duration of one month, you have the option to configure the time-based strategy type. This allows for the setting of two scaling rules (one for expansion and one for contraction) or a single rule for scaling up (scheduled termination).

- **Expansion Rule:** Execute daily, configure the scaling up rule at 10 PM each day, for a duration of one month.

Rule type Scale out Scale in

Policy type

Rule name ⓘ
1-64 characters; supports Chinese characters, letters, digits, -, and _

Execution type Once Recurring

Every day(s)

Execution time

Valid to ⓘ

Scale-out mode ⓘ

More ▾

- **Contraction Rule:** Execute daily, configure the scaling down rule at 6 AM each day, for a duration of one month.

Rule type Scale out Scale in

Policy type

Rule name ⓘ
1-64 characters; supports Chinese characters, letters, digits, -, and _

Execution type Once Recurring

Every day(s)

Execution time

Valid to ⓘ

Scale-out mode ⓘ

More ▾

- **Expansion Rule + Scheduled Termination:** Execute daily, configure the scaling up rule at 10 PM each day, with the batch of resources being used for 8 hours (translating to 6 AM each day), and continue this for a duration of one month. The repetition supports: "Daily", "Weekly", "Monthly". Please set according to the actual situation. For other rule configuration items and usage introductions, please refer to: [Setting Time-based Scaling](#).

Execution type Once Recurring

Daily day(s)

Execution time

Valid to ⓘ

Scale-out mode ⓘ

More ^

Scale-out service

[Specify configuration](#) The component inherits the cluster-level configuration by default. To adjust the configuration, you can s configuration group.

Deployment process **NodeManager**

A deployment process refers to the process information of node deployment.

Node Label ⓘ

Resource supplement retry ⓘ Disable Enable

Retry time after expiration ⓘ

The retry time after expiration should be between 0 and 21600 seconds

Cooldown period ⓘ

Cool-down period range: 0-43200 seconds

Scheduled termination ⓘ hour

Note:

1. The aforementioned times for supplementing resources to the cluster represent an ideal scenario. The actual time consumed for resource expansion is related to the quantity per instance. It is recommended to set the time rule in advance by more than 5 minutes, in accordance with the demand situation.

2. During peak order times, expansion may not reach the elastic target number due to resource contention. It is recommended that you enable the "Resource Supplement Retry Strategy" in your expansion rules.
3. When a contraction action is triggered, a node may be executing a task. To prevent the node from being immediately released, it is recommended that you enable graceful contraction. For more details, please refer to Graceful Contraction.

Best Practices for Setting Load Scaling Rules

Last updated: 2024-01-02 14:22:24

In accordance with the fluctuations in the YARN cluster metrics, select past indicators that align with business changes, configure specific thresholds, then save and apply. Upon business alterations, the corresponding rules will be triggered. The selection of metrics should be inversely proportional to capacity changes. Following a scaling activity, the alteration in the number of instances can reduce the corresponding metrics.

ⓘ For instance:

Configure the expansion rule such that if the average value of AppsPending#root is greater than or equal to 1 within 300 seconds, and this pattern repeats consecutively twice, then a scaling action is triggered. This can effectively reduce the number of tasks pending in the queue.

• Expansion Rule:

The screenshot displays the configuration interface for an Expansion Rule. The 'Rule type' is set to 'Scale out'. The 'Policy type' is 'By load'. The 'Rule name' field is empty, with a note indicating it should be 1-64 characters. The 'Validity' is set to 'Unlimited'. The 'Statistical rule' is configured with 'Rule' set to 'root', 'AvailableVCores#root', a greater-than sign '>', and the value '1'. Below this, there is an 'Add metric' button. The 'Statistical period' is set to '300Second'. The 'Repeat count' is set to '1'. The 'Scale-out mode' is set to 'Node'.

- The contraction rule follows the same principle: Please set according to actual conditions.

For detailed introductions and usage of other rule configuration items, please refer to: [Setting Load Scaling](#).

- 1.1 Within each rule, multiple metric conditions can be configured. When these conditions are simultaneously met, scaling is triggered.
- 1.2 To prevent frequent scaling that could lead to resource wastage, a certain cooling period can be configured for the rule. During this cooling period, even if the scaling conditions are met, no scaling activity will occur.
- 1.3 Configure the effective time (the current rule is effective within a custom time range). Different scaling rules can be combined, and scaling conditions with different content can be configured for different time periods.

The screenshot displays the configuration interface for a scaling rule. Key settings include:

- Rule type:** Scale out (selected), Scale in.
- Policy type:** By load (selected), By time.
- Validity:** Daily, 00:00:00 ~ 23:59:59.
- Statistical rule:** A dropdown menu is open showing options: Saturday, Sunday, Daily (selected), and Custom. The rule expression is `root > AvailableVCores#root` with a value of `1`.
- Statistical period:** Statistical period 300Second.
- Repeat count:** 1.
- Scale-out mode:** Node.

Note:

1. During peak order times, expansion may not reach the elastic target number due to resource contention. It is recommended that you enable the "Resource Supplement Retry Strategy" in your expansion rules.
2. When a scaling-in action is triggered, a node may be executing a task. To prevent the node from being immediately released, it is recommended to enable graceful scaling-in. For more details, see [Graceful Scaling-In](#).