E-MapReduce

FAQ

FAQ

EMR FAQs

Q: What is the difference between a job and an execution plan

A: Descriptions of a job and an execution plan are as follows:

- Job

In E-MapReduce, to create a job is to create a configuration about how to run the job. A job cannot be run directly. The configuration of a job must contain the jar package to be run for the job, the input and output addresses of data, and some running parameters. After such a job is created, you can name it (that is, define a job). When you want to debug the running job, an execution plan is required.

- Execution plan

The execution plan is a bond that associates the job and the cluster. Through the execution plan, multiple jobs can be combined into a job sequence and prepare a running cluster for the job (or automatically create a temporary cluster or associate an existing cluster). The execution plan also helps to set a periodical execution plan for the job sequence and automatically releases the cluster after the task is accomplished. The execution record list displays successful execution plans and logs.

Q: How can I view a job log

A: The E-MapReduce system uploads running job logs to OSS according to the jobid plan (that is, the path that is set by users when they create the cluster). You can view the job logs directly on the webpage. If you log on to the master node for job submission, and you are running the script, the logs are determined by your script according to your plan.

Q: How can I view logs on OSS

A: You can search directly through OSS for all log files, and download them. However, since OSS is unavailable for direct viewing of log files, this procedure may cause issues. The following describes how to use OSS to view log files.

1. Go to the execution plan page.

2. Find the corresponding execution plan and click “Running Log” to enter the running log page.
3. Find the specific execution log on the running log page, such as the last execution log.
4. Click the corresponding “Execution Cluster” to view the ID of the execution cluster.
5. Search for OSS://mybucket/emr/spark/cluster ID directory under the OSS://mybucket/emr/spark directory.
6. Multiple directories are displayed under OSS://mybucket/emr/spark/cluster ID/jobs according to the execution ID of the job, and each directory stores the running log file of the job.

Q: What is the timing policy of the cluster, execution plan, and running job

A: Three timing policies are as follows:

The timing policy of the cluster

In the cluster list, the running time of every cluster is displayed. Calculation of the running time is: Running time = Time when the cluster is released - Time when the cluster is established. Once a cluster starts to be established, the timing starts until the end of the lifecycle of the cluster.

The timing policy of the execution plan

In the running log list of the execution plan, the running time of every execution plan is displayed. The timing policy can be summarized in two categories:

If the execution plan is executed on-demand, the running process of every execution log involves the cluster creation, job submission for running, and cluster release. The calculation policy of an on-demand execution plan is: Running time = The time when the cluster is created + The total time used for completing running all the jobs in the execution plan + The time when the cluster is released.

If the execution plan is associated with an existing cluster, the entire execution cycle does not involve the cluster establishment and releasing. In this case, Running time = The total time used for completing running all the jobs in the execution plan.

The timing policy of the job:

The job here refers to the jobs assigned to the execution plan. Click the View Job List on the right of the running log of every execution plan to see the job. Here the calculation of the

running time of every job is: Running time = The actual time when the job running ends - The actual time when the job starts to run. The actual time when the job running starts (ends) refers to the time points when the job is actually scheduled for running or stops running by the Spark or Hadoop cluster.

Q: During ODPS reading/writing, why does the java.lang.RuntimeException.Parse response failed: '<!DOCTYPE html>...' display?

A: Check whether the ODPS tunnel endpoint is correct.

Q: Why is the TPS inconsistent when multiple Consumer IDs consume the same Topic

A: The topic may have been created in a beta or other testing environment, therefore creating inconsistent consumption data of some consumer groups. Please submit a ticket indicating the corresponding topic and Consumer ID to MQ for troubleshooting.

Q: Can I view job logs on the worker nodes in E-MapReduce

A: Yes, however, the "Save Log" option must be enabled when the cluster is created. The log location is: Execution Plan List > Running Log > Execution Log > View Job List > Job List > View Job Worker Instance.

Q: Why is no data in the external table created in Hive

A: Take the following example:

```
CREATE EXTERNAL TABLE storage_log(content STRING) PARTITIONED BY (ds STRING)

ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION 'oss://xxx:xxxx@log-124531712.oss-cn-hangzhou-internal.aliyuncs.com/biz-logs/airtake/pro/storage';

hive> select * from storage_log;

OK

Time taken: 0.3 seconds

No data is in the created external table.
```

In the preceding example, Hive does not automatically associate the partitions directory of the

specified directory. You must associate it manually. For example:

```
    alter table storage_log add partition(ds=123);
OK
Time taken: 0.137 seconds
hive> select * from storage_log;
OK
abcd 123
efgh 123
```

Q: Why does the Spark Streaming job stop after running for a period of time

A: First, check whether the Spark version is earlier than Version 1.6. Spark Version 1.6 repaired a memory leak bug. Earlier versions of Spark may retain this bug, which causes container memory overuse, meaning the job is not executed. Additionally, check whether your code has been optimized for memory usage.

Q: Why is a job still in “Running” status in E-MapReduce Console even though the Spark Streaming job has ended

A: Check whether the running mode of the Spark Streaming job is “yarn-client.” If yes, we recommend that you change it to the “yarn-cluster” mode. E-MapReduce is not currently optimized for monitoring the status of Spark Streaming jobs in the “yarn-client” mode.

Q: Why does “Exception in thread “main” java.lang.RuntimeException: java.lang.ClassNotFoundException: Class com.aliyun.fs.oss.nat.NativeOssFileSystem not found” display

A: When reading/writing OSS data in Spark jobs, you must package the SDK provided by E-MapReduce into the Jar package. The specific operations can be found at: Development Manual > Spark > Development Preparation.

Q: How can I transmit AccessKeyId and AccessKeySecret parameters for jobs to read/write OSS data

A: One simple method is to use the complete OSS URI. For more information, see: Development Manual > Development Preparation

Q: Why does “Error: Could not find or load main class” display

A: Check whether the path protocol header of the job jar package is “ossref” in the job

configuration. If the protocol header is different, change it to “ossref”.

Q: How can I use the cluster machine division

A: The E-MapReduce contains a master node and multiple slave (or worker) nodes. The master node does not participate in data storage and computing tasks, and the slave nodes are used for data storage and computing. For example, in a cluster with three 4-core 8G machines, one of the machines serves as the master node and the other two serve as the slave nodes. In this case, the available computing resources of the cluster are two 4-core 8G machines.

Q: Why memory overuse happens when Spark is connected to Flume

A: Check whether the data receiving mode is Push-based. If it is a different mode, switch it to the Push-based mode for receiving data. Reference

Q: Why does java.io.IOException: Input stream not be reset when only 5242880 bytes have been written, and does not exceed the available buffer size of 524288 display

A: If insufficient cache is detected, and the preceding error is displayed, we recommend that you use EMR-SDK Version 1.1.0 or later to avoid insufficient cache during OSS network reconnection tries.

Q: Why does “Failed to access metastore. This class should not accessed in runtime.org.apache.hadoop.hive.ql.metadata.HiveException: java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient” display

A: The job execution mode must be yarn-client (or local) for Spark to process Hives data. Yarn-cluster is not supported. Otherwise, the preceding exception appears. Some third-party packages in the job jar file may also trigger the exception while running Spark.

Q: How can I use local sharing library in MR jobs

A: A simple method is to modify the mapred-site.xml file. For example:

```
  <property>
<name>mapred.child.java.opts</name>
<value>-Xmx1024m -Djava.library.path=/usr/local/share/</value>
```

```
</property>

<property>
<name>mapreduce.admin.user.env</name>
<value>LD_LIBRARY_PATH=$HADOOP_COMMON_HOME/lib/native:/usr/local/lib</value>
</property>
```

You then only must add the library file you need.

Q: How can I specify the OSS data source file path in the MR/Spark job

A: See the following.OSS URL: oss://[accessKeyId:accessKeySecret@]bucket[.endpoint]/object/path

This URI is used for specifying input/output data sources in the job, and is similar to hdfs://. In OSS data operations, you can configuration accessKeyId, accessKeySecret, and endpoint to Configuration, or you can specify accessKeyId, accessKeySecret, and endpoint in URI. For more information, see Development Preparation.

Q: Why does the Spark SQL display “Exception in thread “main” java.sql.SQLException: No suitable driver found for jdbc:mysql:xxx” error

A:

1. The mysql-connector-java of an earlier version may have similar issues. Update it to the latest version.
2. In the job parameters, use “—driver-class-path ossref://bucket/.../mysql-connector-java-[version].jar” to load mysql-connector-java package. The preceding issue also occurs when mysql-connector-java is directly packaged into the job jar package.

Q: When Spark SQL is connected to RDS, why does ConnectionException appear

A: Check whether the RDS database address is an intranet address. If it is not, go to the RDS console to switch the database address to an intranet address.

Q: When Spark SQL is connected to RDS, why does the “Invalid authorization specification, message from server: ip not in whitelist” appear

A: Check the RDS whitelist settings and add the intranet addresses of the cluster machines to the RDS whitelist.

Q: During the use of OSS SDK in the Spark program, the following message is displayed “ java.lang.NoSuchMethodError:org.apache.http.conn.ssl.SSLConnetionSocketFactory.init(Ljavax/net/ssl/SSLContext;Ljavax/net/ssl/HostnameVerifier)” What does it mean

A: The http-core and http-client packages that OSS SDK is dependent on have version conflicts with the Spark and Hadoop running environments. We recommend that you do not use OSS SDK in the code, as it requires you to manually solve the dependency conflicts. However, if you want to perform some basic operations, such as list on OSS files, see Simple operations on OSS files.

Cluster port configuration

Hadoop HDFS

Service	Limits	Port	Access Requirements	Configuration	Description
NameNode	-	9000	External	fs.default.name or fs.defaultFS	fs.default.name has expired but is still usable.
NameNode	-	50070	External	dfs.http.address or dfs.namenode.http-address	dfs.http.address has expired but is still usable.

Hadoop YARN (MRv2)

Service	Limits	Port	Access Requirements	Configuration	Description
JobHistory Server	-	10020	Internal	mapreduce.jobhistory.address	-
JobHistory Server	-	19888	External	mapreduce.jobhistory.webapp.address	-
ResourceManager	-	8025	Internal	yarn.resourcemanager.resource-	-

				tracker.address	
ResourceManager	-	8032	Internal	yarn.resourcemanager.address	-
ResourceManager	-	8030	Internal	yarn.resourcemanager.scheduler.address	-
ResourceManager	-	8088	Internal	yarn.resourcemanager.webapp.address	-

Hadoop MapReduce (MRv1)

Service	Limits	Port	Access Requirements	Configuration	Description
JobTracker	-	8021	External	mapreduce.jobtracker.address	-