链滴

# hive 内存设置

作者：bian

Configuring Heapsize for Mappers and Reducers in Hadoop 2

If a YARN container grows beyond its heap size setting, the map or reduce task will fail with a error similar to the one below:

**Container [pid=14639,containerID=container1400188786457000601001609] is running bey nd physical memory limits. Current usage: 2.5 GB of 2.5 GB physical memory used; 3.1 GB of 2.5 GB virtual memory used. Killing container.**

The default heapsize for mappers is 1.5GB and for reducers is 2.5GB on the Altiscale platform.

You can solve this by increasing the heap size for the container for mappers or reducers, dep nding on which one is having the problem when you look at the job history UI or container lo s.

|

```
mapreduce.{map|reduce}.memory.mb vs. mapreduce.{map|reduce}.java.opts
```

|

In Hadoop 2, tasks are run within containers launched by YARN. mapreduce.{map|reduce}.m mory.mb is used by YARN to set the memory size of the container being used to run the map or reduce task. If the task grows beyond this limit, YARN will kill the container.

To execute the actual map or reduce task, YARN will run a JVM within the container. The Had op property mapreduce.{map|reduce}.java.opts is intended to pass options to this JVM. This c n include -Xmx to set max heap size of the JVM. However, the subsequent growth in the me ory footprint of the JVM due to the settings in mapreduce.{map|reduce}.java.opts is limited by the actual size of the container as set by mapreduce.{map|reduce}.memory.mb.

Consequently, you should ensure that the heap you specify in mapreduce.{map|reduce}.java. pts is set to be less than the memory specified by mapreduce.{map|reduce}.memory.mb. If, for example, you see the following fatal error reported by your mapper or reducer:

2014-10-10 00:19:39,693 FATAL [main] org.apache.hadoop.mapred.YarnChild: Error running c ild : java.lang.OutOfMemoryError: Java heap space

This is a good indication that you need to make adjustments to mapreduce.{map|reduce}.java opts and commensurate changes to mapreduce.{map|reduce}.memory.mb.

For example:

hadoop jar -Dmapreduce.map.memory.mb=4096 -Dmapreduce.map.java.opts=-Xmx3686m

and from the Hive CLI, you would run:

|

```
hive> ``set mapreduce.map.memory.mb=4096;

hive> ``set mapreduce.map.java.opts=-Xmx3686m;
```

|

Note: The two properties yarn.nodemanager.resource.memory-mb and yarn.scheduler.maxim

原文链接：hive 内存设置

m-allocation-mb cannot be set by customers.

Setting the heap size for Mappers or Reducers

You can solve the memory error by increasing the heap size for the container for mappers or educers, depending on which one is having the problem when you look at the job history UI r container logs.

SET mapreduce.{map|reduce}.memory.mb=;

set warning

Icon

Important: You should also raise the java heap specified by mapreduce.{map|reduce}.java.opts.

However, you should ensure that the heap you specify in mapreduce.{map|reduce}.java.opts is set to be **less** than the container memory specified by mapreduce.{map|reduce}.memory.mb.

Specifically, a good rule of thumb is to set the jave heap size to be 10% less than the containe size:

**mapreduce.{map|reduce}.java.opts = mapreduce.{map|reduce}.memory.mb x 0.9**

See the section above for further explanation of these two settings.

For example (in Hive), to configure **Reducer** memory allocation:

|

SET mapreduce.reduce.memory.mb=3809; SET mapreduce.reduce.java.opts=-Xmx3428m;

|

or to configure **Mapper** memory allocation:

|

SET mapreduce.map.memory.mb=3809; SET mapreduce.map.java.opts=-Xmx3428m;

|

As a Hadoop job option, for example:

hadoop jar -Dmapreduce.reduce.memory.mb=5120 -Dmapreduce.reduce.java.opts=-Xmx460 m

Increasing the memory size of mappers or reducers comes at the expense of reduced paralleli m of your cluster since it can now launch fewer containers simultaneously, so do feel free to e periment with the memory settings to find the lowest heapsize that will allow you to complete your jobs comfortably.

We would suggest that you at least bump up the values 20% higher according to the virtual emory used from the logs if you have ran into similar execptions. For example, given the foll wing error:

Container [pid=14639,containerID=container1400188786457000601001609] is running beyo

d physical memory limits. Current usage: 2.5 GB of 2.5 GB physical memory used; 3.1 GB of 12
5 GB virtual memory used. Killing container.

you should try at least 3809 (3174 x 1.2) as the new heapsize (mapreduce.{map|reduce}.java.o
ts) value and bump up the mapreduce.{map|reduce}.memory.mb accordingly.

Setting the container heapsize in Hive

Most tools that operate on top of the Hadoop MapReduce framework provide ways to tune t
ese Hadoop level settings for its jobs. For example, in Hive there are multiple ways to do this.
hree of these are shown here:

1. Pass directly via the Hive command line:

hive -hiveconf mapreduce.map.memory.mb=5120 -hiveconf mapreduce.reduce.memory.mb=
120 -hiveconf mapreduce.map.java.opts=-Xmx4608m -hiveconf mapreduce.reduce.java.opts=
Xmx4608m -e select count(*) from test_table;

2) Set the ENV variable before invoking Hive:

export HIVE_OPTS=-hiveconf mapreduce.map.memory.mb=5120 -hiveconf mapreduce.reduc
.memory.mb=5120 -hiveconf mapreduce.map.java.opts=-Xmx4608m -hiveconf mapreduce.re
uce.java.opts=-Xmx4608m

3. Use the set command within the Hive CLI.

|

```
set mapreduce.map.memory.mb=5120;

set mapreduce.map.java.opts=-Xmx4608m;

set mapreduce.reduce.memory.mb=5120;

set mapreduce.reduce.java.opts=-Xmx4608m;

select count(*) from test_table;
```

|

The above 3 examples use a theoritical value that has no assumption. In order to identify whe
her to bump up the mapper's or reducer's memory settings, you should be able to

tell from the Job History UI that will indicate whether it is failing in the Mapper phase or the
educer phase. This varies from application to application that runs on MapReduce and

also varies based on input data and algorithm.

Settings the container heapsize for HiveServer2 sessions

HiveServer2 provides a different channel than HiveCLI and the Hive command line tool. If you are submitting queries via HiveServer2 with JDBC or ODBC driver, or a python module such as pyhs2, the following examples show you how to customize the values.

1. Beeline / JDBC URL

The JDBC URL string will look like this:

```
jdbc:hive2://localhost:10000/default?mapreduce.map.memory.mb=3809;mapreduce.map.java
opts=-Xmx3428m;mapreduce.reduce.memory.mb=2560;mapreduce.reduce.java.opts=-Xmx2
04m;
```

You use a semi-colon to specify multiple key-value pairs to customize this session with HiveS rver2. The default in the URL before the question mark is pointing to the default databas
.

2. pyhs2 module example

You will need to perform the SET statements without the semi-colon in the cursor.

```
import pyhs2

conn = pyhs2.connect(host='hostnametoyourhiveserver2',port=10000,user='alti-test',authMe
hanism=PLAIN,database='default')

cur = conn.cursor()

cur.execute(SET mapreduce.map.memory.mb=3809)

cur.execute(SET mapreduce.map.java.opts=-Xmx3428m)

cur.execute(SET mapreduce.reduce.memory.mb=2560)

cur.execute(SET mapreduce.reduce.java.opts=-Xmx2304m)

cur.execute("SELECT COUNT(*) FROM yourtableexample")
```

Note: The authMechanism depends on what is enabled in your HiveServer2 settings.