Demystify Apache Tez Memory Tuning - Step by Step

(/users/369/amcbarnett.html)

Ancil McBarnett (/users/369/amcbarnett.html) created · Feb 04, 2016 at 04:48 AM · **edited (/revisions/14309.html)** · Feb 04, 2016 at 05:42 AM

Short Description:

How Tez should be configured with step by step instructions and explanation to give insight on the rationale behind settings

Article

Introduction

Apache Tez has become a very important framework and API to support batch and interactive over terabytes and petabytes of data for many engines within HDP such as Pig, Hive, Java. Cascading and others, with performance advantages at scale over Map Reduce and even Spark at certain volumes of data.

For more on Apache Tez see http://hortonworks.com/hadoop/tez/ (http://hortonworks.com/hadoop/tez/)

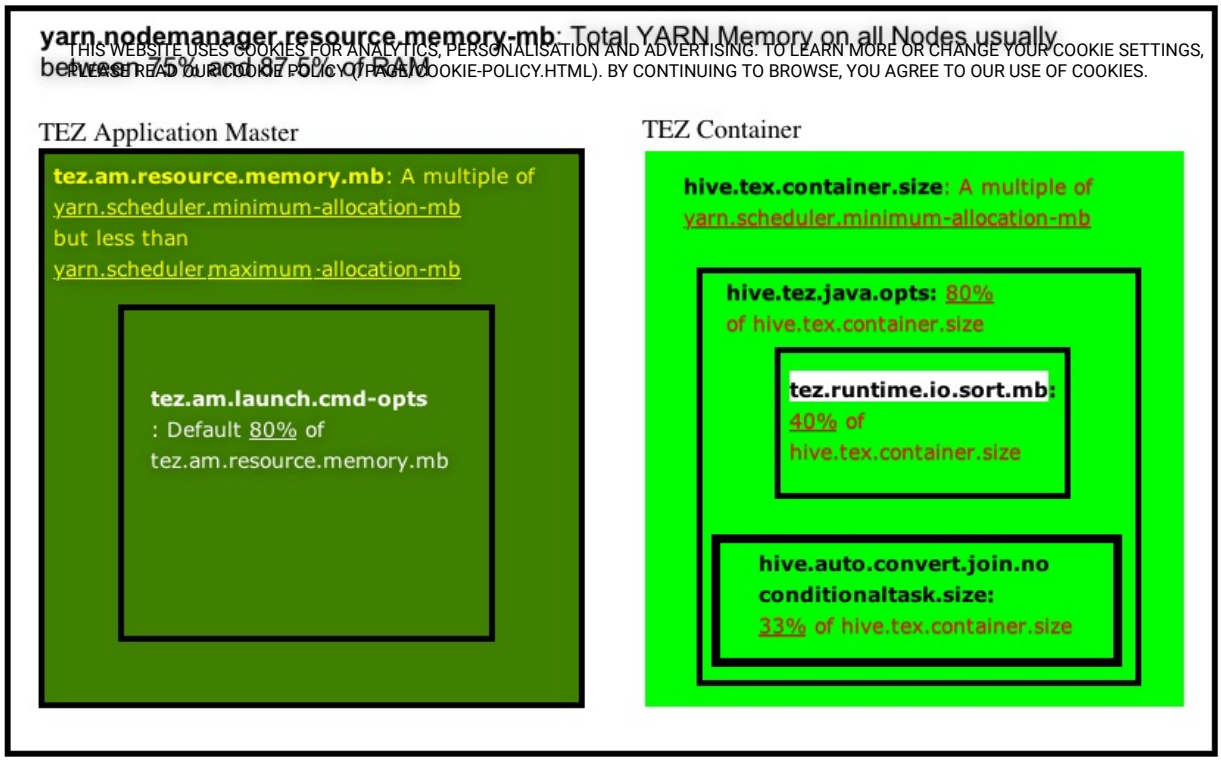
This article is meant to outline the best practice in configuring and tuning Tez, and why you would set certain values in certain properties to get performance at scale, with step by step instructions.

With this in place you would hopefully prevent out of memory errors when you execute your Hive Queries or Pig Scripts as seen in https://community.hortonworks.com/questions/5780/hive-on-tez-query-map-output-outofmemoryerror-java.html (https://community.hortonworks.com/questions/5780/hive-on-tez-query-map-output-outofmemoryerror-java.html) https://community.hortonworks.com/questions/12067/what-is-the-workaround-when-getting-hive-outofmemo.html (https://community.hortonworks.com/questions/12067/what-is-the-workaround-when-getting-hive-outofmemo.html)

Tez Memory Demystified

I find a diagram usually helps to understand why you would set certain properties.

This is a quick summary of the main memory settings for Tez for both the Application Master and Container. Please refer to it as you read below.



A list of some of the main Tez properties can be found here:

http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_installing_manually_book/content/ref-ffec9e6b-41f4-47de-b5cd-1403b4c4a7c8.1.html (http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_installing_manually_book/content/ref-ffec9e6b-41f4-47de-b5cd-1403b4c4a7c8.1.html)

I also highly recommend reading the Hive Tuning Guide for HDP (http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_hive_architectural_overview.html)

(http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_hive_architectural_overview.html)

Steps to Configure

Step 0 - If you are a Hortonworks Support Subscription Customer, begin to utilize the **SamrtSense** tool. **Hortonworks SmartSense** is a cluster diagnostic and recommendation tool that is critical for efficient support case resolution, pre-emptive issue detection and performance tuning. Your recommended Tez configurations would be provided to you as a customer. This is the value Hortonworks brings.

You can access the white paper here http://hortonworks.com/info/hortonworks-smartsense/ (http://hortonworks.com/info/hortonworks-smartsense/)

Upload your bundles, apply the recommendations, and you have no need to go any further in this article.

But if you must....

Step 1 - Determine your YARN Node manager Resource Memory (***yarn.nodemanager.resource.memory-mb***) and your YARN minimum container size (***yarn.scheduler.minimum-allocation-mb***). Your ***yarn.scheduler.maximum-allocation-mb*** is the same as ***yarn.nodemanager.resource.memory-mb.***

yarn.nodemanager.resource.memory-mb is the Total memory of RAM allocated for all the nodes of the cluster for YARN. Based on the number of containers, the minimum YARN memory allocation for a container is ***yarn.scheduler.minimum-allocation-mb.*** ***yarn.scheduler.minimum-allocation-mb*** will be a very important setting for our Tez Application Master and Container sizes.

So how do we determine this with just the number of cores, disks, and RAM on each node? The Hortonworks easy button approach. Follow the instructions at this link, Determine HDP Memory Config (http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_installing_manually_book/content/determine-hdp-memory-config.html).

For example, if you are on HD Insight running a D12 node with 8 CPUs and 28GBs of memory, with no HBase, you run:

```
Run python yarn-utils.py -c 8 -m 28 -d 2 -k False
```

Your output would look like this.

```
[root@sandbox scripts]# python yarn-utils.py -c 4 -m 28 -d 2 -k False
 Using cores=4 memory=28GB disks=2 hbase=False
 Profile: cores=4 memory=27648MB reserved=1GB usableMem=27GB disks=2
 Num Container=4
 Container Ram=6656MB
 Used Ram=26GB
 Unused Ram=1GB
 yarn.scheduler.minimum-allocation-mb=6656
 yarn.scheduler.maximum-allocation-mb=26624
 yarn.nodemanager.resource.memory-mb=26624
 mapreduce.map.memory.mb=6656
 mapreduce.map.java.opts=-Xmx5324m
 mapreduce.reduce.memory.mb=6656
 mapreduce.reduce.java.opts=-Xmx5324m
 yarn.app.mapreduce.am.resource.mb=6656
 yarn.app.mapreduce.am.command-opts=-Xmx5324m
 mapreduce.task.io.sort.mb=2662
```

In Ambari, configure the appropriate settings for YARN and MapReduce or in a non-Ambari managed cluster, manually add the first three settings in yarn-site.xml and the rest in mapred-site.xml on all nodes.

Step 2 - Determine your Tez Application Master and Container Size, that is ***tez.am.resource.memory.mb*** and ***hive.tez.container.size***.

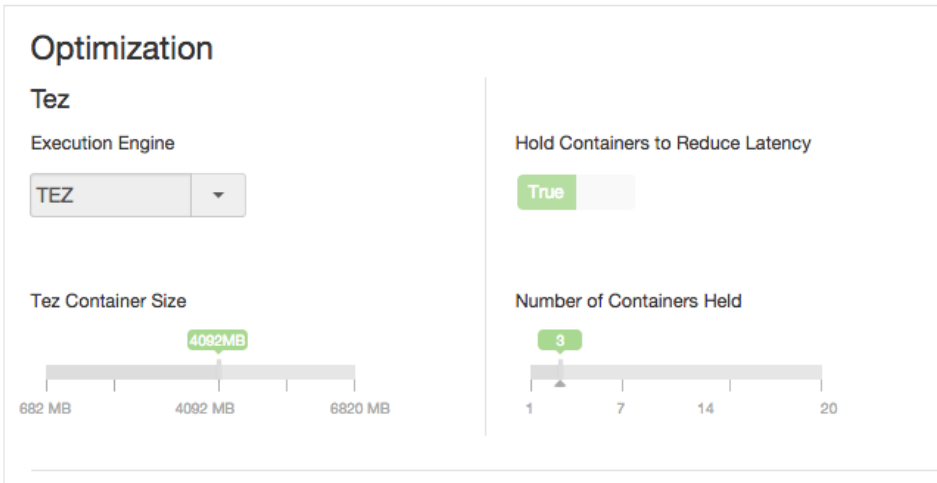
Set ***tez.am.resource.memory.mb*** to be the same as ***yarn.scheduler.minimum-allocation-mb*** the YARN minimum container size.

Set ***hive.tez.container.size*** to be the same as or a small multiple (1 or 2 times that) of YARN container size ***yarn.scheduler.minimum-allocation-mb*** but NEVER more than ***yarn.scheduler.maximum-allocation-mb***. You want to have headroom for multiple containers to be spun up.

A general guidance: Don't exceed Memory per processors as you want one processor per container. So if you have for example, 256GB and 16 cores, you don't want to have your container bigger than 16GB.

Bonus:

- Container Reuse set to True: tez.am.container.reuse.enabled (Default is true)
- Prewarm Containers when HiveSever2 Starts, under Hive Configurations in Ambari.

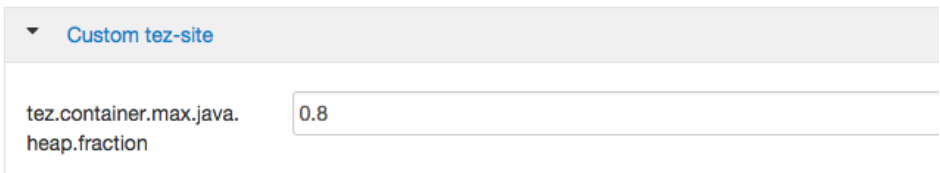


Step 3 - Application Master and Container Java Heap sizes (***tez.am.launch.cmd-opts*** and ***hive.tez.java.ops*** respectively)

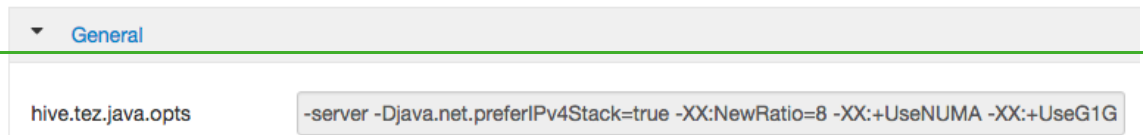
By default these are BOTH 80% of the container sizes, ***tez.am.resource.memory.mb*** and ***hive.tez.container.size***respectfully.

NOTE: ***tez.am.launch.cmd-opts*** is automatically set, so no need to change this.

In HDP 2.3 and above, no need to also set ***hive.tez.java.op***s as it can be automatically set controlled by a new property ***tez.container.max.java.heap.fraction*** which is defaulted to 0.8 in ***tez-site.xml.*** This property is not by default in Ambari. If you wish you can add it to the Custom tez-site.sml.



As you can see from Ambari, in Hive -> Advance configurations, there are no manual memory configurations set for ***hive.tez.java.opts***



if you wish to make the heap 75% of the container, then set the Tez Container Java Heap Fraction to 0.75

If you wish this set manually, you can add to ***hive.tez.java.op***s for example -Xmx7500m -Xms 7500m, as longs as it is a fraction of ***hive.tez.container.size***

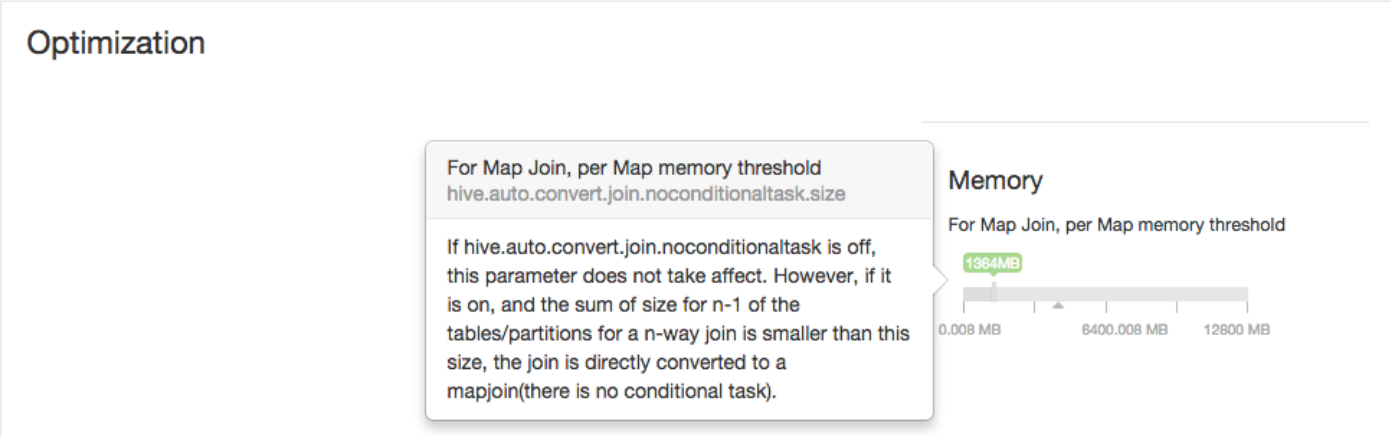
Step 4: Now to determine Hive Memory Map Join Settings parameters.

tez.runtime.io.sort.mb is the memory when the output needs to be sorted.

tez.runtime.unordered.output.buffer.size-mb is the memory when the output does not need to be sorted.

hive.auto.convert.join.noconditionaltask.size is a very important parameter to size memory to perform Map Joins. You want to perform Map joins as much as possible.

In Ambari this is under the Hive Confguration



For more on this see http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_setting_memory_usage_for_hive_perf.html (http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_setting_memory_usage_for_hive_perf.html)

(http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_setting_memory_usage_for_hive_perf.html)

```
SET tez.runtime.io.sort.mb to be 40% of hive.tez.container.size.  You should rarely have more than2GB set.

By default hive.auto.convert.join.noconditionaltask = true

SET hive.auto.convert.join.noconditionaltask.size  to 1/3 of hive.tez.container.size

SET tez.runtime.unordered.output.buffer.size-mb to 10% of hive.tez.container.size
```

FOR MORE ADVANCED SETTINGS CONCERNING QUERY OPTIMIZATION

Step 5 - For for Query optimization and Mapper Parallelism see http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_query_optimization_hive.html (http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.4/bk_performance_tuning/content/ch_query_optimization_hive.html)

Step 6 - Determining Number of Mappers

The following parameters control the number of mappers for splittable formats with Tez:

```
set tez.grouping.min-size=16777216; -- 16 MB min split

set tez.grouping.max-size=1073741824; -- 1 GB max split

Increase min and max split size to reduce the number of mappers.
```

See also

How Initial task parallelism works (https://cwiki.apache.org/confluence/display/TEZ/How+initial+task+parallelism+works)

https://community.hortonworks.com/questions/905/how-are-number-of-mappers-determined-for-a-query-w.html (https://community.hortonworks.com/questions/905/how-are-number-of-mappers-determined-for-a-query-w.html)

References For Microsoft Azure and HDInsight

- https://azure.microsoft.com/en-us/documentation/articles/hdinsight-hadoop-hive-out-of-memory-error-oom/ (https://azure.microsoft.com/en-us/documentation/articles/hdinsight-hadoop-hive-out-of-memory-error-oom/)