ANSWERED

Allocating more memory to YARN?

Question asked by **reedv** on Jan 22, 2018
Latest reply on Jan 23, 2018 by deborah

Like • 0 Comment • 1

Reading this blog post to find the parameter to change to dedicate more memory of each node in the cluster to YARN, instructs to edit the parameter: **`yarn.scheduler.minimum-allocation-mb`**
Looking at the configuration page in the YARN web UI, the config parameter for setting the total memory allocation to yarn from each node is listed as

```
....
<property>
   <name>yarn.nodemanager.resource.memory-mb</name>
   <value>${nodemanager.resource.memory-mb}</value>
   <source>yarn-default.xml</source>
</property>
....
```

so I take this to mean would need to edit the yarn-default.xml file in $HADOOP_HOME (**and restart YARN/the cluster?**), but where is this file? Can I just add this property to the $HADOOP_HOME/etc/hadoop/yarn-site.xml files for all the nodes?
Looking in the mapr $HADOOP_HOME/etc/hadoop, could only find

```
[root@mapr002 hadoop-2.7.0]# ls etc/hadoop/yarn*
etc/hadoop/yarn-env.sh etc/hadoop/yarn-site.xml.template
etc/hadoop/yarn-site-2017-12-14.17-02.xml etc/hadoop/yarn-timelineserver-properties.xml
etc/hadoop/yarn-site-2017-12-14.17-56.xml etc/hadoop/yarn-timelineserver-security-properties.xml
etc/hadoop/yarn-site.xml
```

and none of these files contained any mention of the resource.memory-mb property. Further searching (after reading this blog post), lead me to look in the location /opt/mapr/conf/conf.d (**should I have used /opt/mapr/conf/conf.d.new?**) which has the file warden.resourcemanager.conf where I'd expect to get to set
`YARN_NODEMANAGER_OPTS= -Dnodemanager.resource.memory-mb=10817`
but there seems to be no such variable there.
Looking in /opt/mapr/hadoop/hadoop-2.7.0/etc/hadoop/yarn-env.sh, there are the lines

```
....
YARN_RESOURCEMANAGER_OPTS="$YARN_RESOURCEMANAGER_OPTS -Dfs.cache.lru.enable=true"
export YARN_RESOURCEMANAGER_OPTS="${YARN_RESOURCEMANAGER_OPTS} ${MAPR_LOGIN_OPTS}"
export YARN_NODEMANAGER_OPTS="${YARN_NODEMANAGER_OPTS} ${MAPR_LOGIN_OPTS}"
export YARN_HISTORYSERVER_OPTS="${YARN_HISTORYSERVER_OPTS} ${MAPR_LOGIN_OPTS}"
export YARN_TIMELINESERVER_OPTS="${YARN_TIMELINESERVER_OPTS} ${MAPR_LOGIN_OPTS}"
....
```

Yet the command "echo ${YARN_RESOURCEMANAGER_OPTS} " returns nothing. So, what should be done here? Should I just edit the line in yarn-env.sh is to look like

```
YARN_RESOURCEMANAGER_OPTS="$YARN_RESOURCEMANAGER_OPTS -Dfs.cache.lru.enable=true -Dnodemanager.resource.memory-mb=<desired per-node mem. allocation amount>"
```

As an aside, is there a way to have certain node contribute more memory that others? I ask because ultimately, in our cluster, some nodes have more memory allocated to them than others that they can spare, so it would be very convenient it we could just use resources from those nodes specifically.
version: mapr 6.0
installed via installer script web GUI.

deborah
Jan 23, 2018 2:06 PM

★ Correct Answer

Here is a very general picture of how resource allocation works:

When you run `configure.sh` on a node, Warden allocates the node's memory in this order:

- First it allocates memory to the OS
- Then it allocates memory required by the mfs services
- Then it allocates memory required by the various services (nfs, webserver, etc.)
- 85% of what is left is allocated to YARN

Allocating Memory to Services

In the `/opt/mapr/conf/warden.conf` file, you can see what has been set for each service. Some services, such as resourcemanager and nodemanager, are set in `/opt/mapr/conf/conf.d/warden.<service>.conf`.

There are three parameters to look for, ending in `heapsize.min`, `heapsize.max`, and `heapsize.percent`. Warden will allocate `heapsize.percent`, but it must be at least `heapsize.min` and cannot exceed `heapsize.max`.

For example, suppose you have these settings (max and min are in MB):

```
...heapsize.percent=10
...heapsize.min=5000
...heapsize.max=10000
```

- If 10% of physical memory is 8000 MB, that much will be allocated
- If 10% of physical memory is 12000 MB, only 10000 will be allocated (upper bound is heapsize.max)
- If 10% of physical memory is 4000 MB, 5000 MB will be allocated (lower bound is heapsize.min)

Keep in mind, this is the memory allocated for the service to run, not the memory that can be allocated to YARN tasks. If you change `warden.conf` or `warden.<service>.conf`, restart the Warden for them to take effect.

Note: if you are NOT using MapR-DB, make sure the value of `isDB` in warden.conf is set to `false`. Otherwise, memory will be allocated for MapR-DB.

Allocation of memory for YARN resource containers

You can configure how YARN allocates memory to resource containers in the `yarn-site.xml` file (located at `/opt/mapr/hadoop/hadoop-<version>/etc/hadoop/yarn-site.xml`). For example:

- `yarn.scheduler.minimum-allocation-mb` is the smallest amount of memory that will be allocated to each resource container (resources are allocated in increments of this amount)
- `yarn.scheduler.maximum-allocation-mb` is the largest amount of memory that will be allocated to each resource container
- `yarn.nodemanager.resource.memory-mb` is the amount of physical memory, in MB, that can be allocated for all containers (obviously, you don't want this to be more than the amount of memory that is available for YARN)

Example (these numbers are not necessarily reasonable for any given system, just using for the example):

```
yarn.scheduler.minimum-allocation-mb = 1000
yarn.scheduler.maximum-allocation-mb = 5000
yarn.nodemanager.resource.memory-mb = 20000
```

- A task requests 800 MB, it will get 1000 (per the minimum allocation)
- Five tasks want to run concurrently with 5000 MB each - four of them can run, but then the `resource.memory-mb` is hit and the fifth task cannot run until one of the others completes
- A task requests 25000 MB, it will generate an error (more than the total memory available)

Finally - the settings in these files can differ from node to node. If you modify `yarn-site.xml`, you need to restart the node manager.

See the reply in context

No one else had this question

OUTCOMES

Helpful(1)

Visibility: Answers • **84 Views**

Last modified on Jan 22, 2018 6:13 PM

Tags: yarn | resource management | yarn-site.xml | mapr 6.0 | allocate

Categories: Core Hadoop and Resource Management | Provisioning, Configuration, Monitoring, Inst…

0

1 Reply

deborah

Jan 23, 2018 2:06 PM

Here is a very general picture of how resource allocation works:

When you run `configure.sh` on a node, Warden allocates the node's memory in this order:

- First it allocates memory to the OS
- Then it allocates memory required by the mfs services
- Then it allocates memory required by the various services (nfs, webserver, etc.)
- 85% of what is left is allocated to YARN

Allocating Memory to Services

In the `/opt/mapr/conf/warden.conf` file, you can see what has been set for each service. Some services, such as resourcemanager and nodemanager, are set in `/opt/mapr/conf/conf.d/warden.<service>.conf`.

There are three parameters to look for, ending in `heapsize.min`, `heapsize.max`, and `heapsize.percent`. Warden will allocate `heapsize.percent`, but it must be at least `heapsize.min` and cannot exceed `heapsize.max`.

For example, suppose you have these settings (max and min are in MB):

```
...heapsize.percent=10
...heapsize.min=5000
...heapsize.max=10000
```

- If 10% of physical memory is 8000 MB, that much will be allocated
- If 10% of physical memory is 12000 MB, only 10000 will be allocated (upper bound is heapsize.max)
- If 10% of physical memory is 4000 MB, 5000 MB will be allocated (lower bound is heapsize.min)

Keep in mind, this is the memory allocated for the service to run, not the memory that can be allocated to YARN tasks. If you change `warden.conf` or `warden.<service>.conf`, restart the Warden for them to take effect.

Note: if you are NOT using MapR-DB, make sure the value of `isDB` in warden.conf is set to `false`. Otherwise, memory will be allocated for MapR-DB.

Allocation of memory for YARN resource containers

You can configure how YARN allocates memory to resource containers in the `yarn-site.xml` file (located at `/opt/mapr/hadoop/hadoop-<version>/etc/hadoop/yarn-site.xml`). For example:

- `yarn.scheduler.minimum-allocation-mb` is the smallest amount of memory that will be allocated to each resource container (resources are allocated in increments of this amount)
- `yarn.scheduler.maximum-allocation-mb` is the largest amount of memory that will be allocated to each resource container
- `yarn.nodemanager.resource.memory-mb` is the amount of physical memory, in MB, that can be allocated for all containers (obviously, you don't want this to be more than the amount of memory that is available for YARN)

Example (these numbers are not necessarily reasonable for any given system, just using for the example):

```
yarn.scheduler.minimum-allocation-mb = 1000
yarn.scheduler.maximum-allocation-mb = 5000
yarn.nodemanager.resource.memory-mb = 20000
```

- A task requests 800 MB, it will get 1000 (per the minimum allocation)
- Five tasks want to run concurrently with 5000 MB each - four of them can run, but then the `resource.memory-mb` is hit and the fifth task cannot run until one of the others completes
- A task requests 25000 MB, it will generate an error (more than the total memory available)

Finally - the settings in these files can differ from node to node. If you modify `yarn-site.xml`, you need to restart the node manager.

1 person found this helpful

Actions

Like • 1