HBase Metrics

Table of contents

1. Introduction

HBase emits Hadoop metrics.

2. HOWTO

First read up on Hadoop metrics. If you are using ganglia, the GangliaMetrics wiki page is useful read.

To have HBase emit metrics, edit `$HBASE_HOME/conf/hadoop-metrics.properties` and enable metric 'contexts' per plugin. As of this writing, hadoop supports **file** and **ganglia** plugins. Yes, the hbase metrics files is named hadoop-metrics rather than *hbase-metrics* because currently at least the hadoop metrics system has the properties filename hardcoded. Per metrics *context*, comment out the NullContext and enable one or more plugins instead.

If you enable the *hbase* context, on regionservers you'll see total requests since last metric emission, count of regions and storefiles as well as a count of memstore size. On the master, you'll see a count of the cluster's requests.

Enabling the *rpc* context is good if you are interested in seeing metrics on each hbase rpc method invocation (counts and time taken).

The *jvm* context is useful for long-term stats on running hbase jvms -- memory used, thread counts, etc. As of this writing, if more than one jvm is running emitting metrics, at least in ganglia, the stats are aggregated rather than reported per instance.

3. Using with JMX

In addition to the standard output contexts supported by the Hadoop metrics package, you can also export HBase metrics via Java Management Extensions (JMX). This will allow viewing HBase stats in JConsole or any other JMX client.

3.1. Enable HBase stats collection

To enable JMX support in HBase, first edit `$HBASE_HOME/conf/hadoop-metrics.properties` to support metrics refreshing. (If you've already configured `hadoop-metrics.properties` for another output context, you can skip this step).

```
# Configuration of the "hbase" context for null
hbase.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
```

```
hbase.period=60

# Configuration of the "jvm" context for null
jvm.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
jvm.period=60

# Configuration of the "rpc" context for null
rpc.class=org.apache.hadoop.metrics.spi.NullContextWithUpdateThread
rpc.period=60
```

3.2. Setup JMX remote access

For remote access, you will need to configure JMX remote passwords and access profiles. Create the files:

$HBASE_HOME/conf/jmxremote.passwd (set permissions to 600)

```
monitorRole monitorpass
controlRole controlpass
```

$HBASE_HOME/conf/jmxremote.access

```
monitorRole readonly
controlRole readwrite
```

3.3. Configure JMX in HBase startup

Finally, edit the $HBASE_HOME/conf/hbase-env.sh script to add JMX support:

$HBASE_HOME/conf/hbase-env.sh

Add the lines:

```
HBASE_JMX_OPTS="-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.ssl=false"
HBASE_JMX_OPTS="$HBASE_JMX_OPTS
-Dcom.sun.management.jmxremote.password.file=$HBASE_HOME/conf/jmxremote.passwd"
HBASE_JMX_OPTS="$HBASE_JMX_OPTS
-Dcom.sun.management.jmxremote.access.file=$HBASE_HOME/conf/jmxremote.access"

export HBASE_MASTER_OPTS="$HBASE_JMX_OPTS
-Dcom.sun.management.jmxremote.port=10101"
export HBASE_REGIONSERVER_OPTS="$HBASE_JMX_OPTS
-Dcom.sun.management.jmxremote.port=10102"
```

After restarting the processes you want to monitor, you should now be able to run JConsole (included with the JDK since JDK 5.0) to view the statistics via JMX. HBase MBeans are exported under the **hadoop** domain in JMX.