



HBASE LAB

Overview: HDFS typically only allows you to append to files and then batch read all of the files with brute force using MapReduce to find what you need. HBase is a layer over HDFS that gives you a loose columnar schema using Google's BigTable design. Random access in Hadoop has never been easier!

- 1) In Cloudera Manager, start the HBase service. This service could turn red later if, for instance, your VM falls asleep. If you are experiencing issues with HBase during the lab be sure to return to Cloudera Manager to verify that the HBase service is still running.
- 2) Go to the HBase Browser in Hue. Select to create a new table. Call the table "codemash" and give it a single column family of "f".
- 3) Now select the table you just created. You should see a yellow bar that says "No rows to display".
- 4) Let's make a new row. There's a blue button in the bottom right for this that might be hard to see if you are using a browser inside your VM.
- 5) Call the new row "hot_tub" and add a couple fields of "f:loc → waterpark" and "f:att → swimsuit" to describe where to describe where to find it (loc = location) and what to wear there (att = attire). We use short column family and key names to help performance.
- 6) Make another row the same way called "precompiler" with the fields "f:loc → convention center", "f:att → Hadoop swag" and one more field this time of "f:t → laptop" (t = tool).
- 7) Oops. We should have specified that the tool cell should be "64 bit laptop". Edit the cell in HBase browser by clicking on the cell value of "laptop" and change it to "64 bit laptop". Click away from the cell to save. It now

appears that we've updated the value.

- 8) Now, let's explore using an HBase shell. You can do this in Hue by selecting Hue Shell > HBase Shell or by opening a new Terminal window in the VM and executing the command "hbase shell".
- 9) Your first command to run in HBase shell is "list". This will list your tables.
- 10) Now let's query the codemash table. Try: scan 'codemash' to see the contents of the table.
- 11) Let's **get** a specific row from the table: get 'codemash', 'hot_tub'
- 12) We now have a new row we want to put to codemash. Put the new row by putting its first cell: put 'codemash', 'game_room', 'f:loc', 'lobby'
- 13) See what you did with another: scan 'codemash'
- 14) Delete a cell because you finished your "foofoo drink":
delete 'codemash', 'hot_tub', 'f:t'
- 15) See your delete's result with another: scan 'codemash'
- 16) Let's see what really happened when we updated our "laptop" cell to a "64 bit laptop cell with:
get 'codemash', 'precompiler', {COLUMN => 'f:t', VERSIONS => 3}
^^ that command asks for the last 3 versions of the "f:t" value!

Play around: HBase shell is a JRuby shell. Know a little Ruby? Try it out!

Bonus #1: Tables can be altered in HBase but you must disable them to do this. Let's add a second column family (f2) to our codemash table:

```
> disable 'codemash'  
> alter 'codemash', 'f', 'f2'  
> enable 'codemash'  
> put 'codemash', 'precompiler', 'f2:bonus1', 'altering tables'
```

View your work in Hue HBase browser.

Bonus #2: HCatalog/Metadata can be created against an HBase table by creating an “external” table. This allows you to use SQL or Pig against HBase. Let's make such a table with the following query in Hive UI:

```
create external table codemash(key string, place string, attire string, tool string)
stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with SERDEPROPERTIES("hbase.columns.mapping" = ":key, f:loc, f:att, f:t")
TBLPROPERTIES("hbase.table.name" = "codemash");
```

Now go to Hue Metastore Manager to view a sample of your data. Try a Hive or Impala query against this new table.

Bonus #3 ZooKeeper - You've probably seen the name ZooKeeper tossed about. For instance, it's one of the services listed in Cloudera Manager. With a run of "sudo jps" in a Terminal you'll actually see it called "QuorumPeerMain". ZooKeeper helps distributed things stay in sync and is used extensively by HBase. ZooKeeper is a fairly transparent tool to most Hadoop users - meaning you could use it for years without really interacting with it directly. Here, will play with it a bit anyhow because, well, why not?

You may invoke a ZooKeeper shell directly with:

```
/usr/lib/zookeeper/bin/zkCli.sh -server 127.0.0.1:2181
```

From a ZooKeeper shell, try:

```
ls /                # this lists the ZNodes
create /mash bacon  # create a new ZNode "mash" with initial value "bacon"
ls /                # lists the ZNodes including "mash"
get /mash           # shows the "bacon" value along with other attributes
set /mash eggs      # "eggs" now replaces "bacon"
get /mash           # see "eggs" and the "dataVersion" was incremented
delete /mash        # say goodbye
```

OK, so we're on a single node that doesn't need distribution help and this example for ZooKeeper is rather simple one. But now you're slightly more familiar than you once were with ZooKeeper.