

coreservlets.com – Hadoop Course

HBase Installation and Shell

In this exercise, you have a chance to examine development installation of HBase. You will get a chance to interact with the HBase via HBase Shell as well as create a table, insert multiple records, retrieve data and finally, drop the table.

Approx. Time: 60 minutes

Answer

1. What is one good reason to use HBase?
2. What is one bad reason to use HBase?
3. Can you create new columns at run-time or do they need to be pre-defined ahead of time?
4. Where will HBase store its data (HFiles and HLog/WAL)? How is it configured?
5. Why does replication need to be set to 1 in Pseudo-Distributed mode?

Perform

Using HBase shell commands perform the following tasks:

1. Start HBase
2. Start HBase shell
3. Create table called 'Book' whose schema will be able to house book's title, description, author's first and last names. Book's title and description should be grouped as they will be saved and retrieved together. Author's first and last name should also be grouped.
4. Add the following information to the 'Book' table:

ID	Title	Description	Author's first name	Author's last name
1	Faster than the speed love	Long book about love.	Brian	Dog
2	Long day	Story about Monday.	Emily	Blue
3	Flying Car	Novel about airplanes.	Phil	High

5. Count the number of rows. Make sure that every row is printed to the screen as it being counted.
6. Retrieve an entire record with id '1'
7. Only retrieve title and description for record with id '3'
8. Change the last name of an author for the record with title 'Long Day' to 'Happy'.
 - a. Display the record on the screen to verify the change.
 - b. Display both new and old value. You should be able to see both 'Blue' and 'Happy'. Why is that??
9. Display all the records to the screen
10. Display title and author's last name for all the records
11. Display title and description for the first 2 records

12. Explore HBase Web-based management console, try and learn as much as you can about your new table.
13. Check the 'detailed' status of your cluster via HBase shell
14. Delete a record whose title is 'Flying Car'.
 - a. Validate the record was deleted by scanning all the records or by attempting to select the record
15. Drop the table 'Book'

Answer Solution

1. HBase is quite excellent at handling a lot of data and scaling horizontally to thousands of nodes. It can host billions of rows, and petabytes of data . HBase is also good at supporting a large amount of client requests. In my experience, it can handle 1000s of gets per second per machine.
2. If you need a rich access interface such as SQL or text-based search (like Lucene), then HBase is not for you. You only have access to your data by row-id with a very limited set of operations (get and scan).
3. Columns are created at the time that data is being inserted. They don't need to be defined up front. HBase is designed to scale to millions of columns per column family. On the other hand, table name and its family need to be defined before the data is inserted.
4. For a Pseudo-Distributed mode and Fully-Distributed mode HFiles and HLog are stored in HDFS and for Local mode they are just stored on the local file system. The location is configured in via property **hbase.rootdir** in the `<hbase_install>/conf/hbase-site.xml`
5. There is only 1 node... nowhere to replicate :)

Perform Solution

```
1. $ cd $HBASE_HOME/bin
```

```
$ ./start-hbase.sh
```

```
2. $ hbase shell
```

3. Since title and description need to be grouped together and so do author's first and last name, it would be wise to place them into 2 families such as 'info' and 'author':

```
hbase> create 'Book', {NAME=>'info'}, {NAME=>'author'}
```

Then 'title' and 'description' will become columns of 'info' family and 'first' and 'last' columns of 'author' family.

4. Create put statements to insert records specified in the table:

```
put 'Book', '1', 'info:title', 'Faster than the speed love'
```

```
put 'Book', '1', 'info:description', 'Long book about love.'
```

```
put 'Book', '1', 'author:first', 'Brian'
```

```
put 'Book', '1', 'author:last', 'Dog'
```

```
put 'Book', '2', 'info:title', 'Long day'
```

```
put 'Book', '2', 'info:description', 'Story about Monday.'
```

```
put 'Book', '2', 'author:first', 'Emily'
```

```
put 'Book', '2', 'author:last', 'Blue'
```

```
put 'Book', '3', 'info:title', 'Flying Car'
```

```
put 'Book', '3', 'info:description', 'Novel about airplanes.'
```

```
put 'Book', '3', 'author:first', 'Phil'
```

```
put 'Book', '3', 'author:last', 'High'
```

5. hbase> count 'Book', INTERVAL => 1
6. hbase> get 'Book', '1'
7. hbase> get 'Book', '1', {COLUMNS=>['info:title', 'info:description']}
8. The record with title 'Long Day' has an ID of '1', to update a value simply put a record with the same ID:
hbase> put 'Book', '2', 'author:last', 'Happy'
To verify select the record
hbase> get 'Book', '2', {COLUMNS=>'author:last'}
To display both version simply
hbase> get 'Book', '2', {COLUMNS=>'author:last', VERSIONS=>3}
The reason we can see a previous value is because cells are versions and HBase preserves 3 versions by default.
9. hbase> scan 'Book'
10. hbase> scan 'Book', {COLUMNS=>['info:title', 'author:last']}
11. hbase> scan 'Book', {COLUMNS=>['info:title', 'info:description'], LIMIT=>2}
or
hbase> scan 'Book', {COLUMNS=>['info:title', 'info:description'], STOPROW=>'3'}
12. Book table is hosted via 1 Region Server and there is only 1 Region. There are no start or end keys for that region because there is only 1 region. It has 2 families 'info' and 'author'. There is no compression set for both families, and replication is set to 3.
13. hbase> status 'detailed'
14. The record whose title is 'Flying Car' has row id of '3'. So simply delete all the columns with the record whose id '3':
delete 'Book', '3', 'info:title'
delete 'Book', '3', 'info:description'
delete 'Book', '3', 'author:first'
delete 'Book', '3', 'author:last'
15. hbase> disable 'Book'
hbase> drop 'Book'