CSCI4180 Tutorial Week 10
Assignment 2
HBase Setup and Implementation

7 November 2013

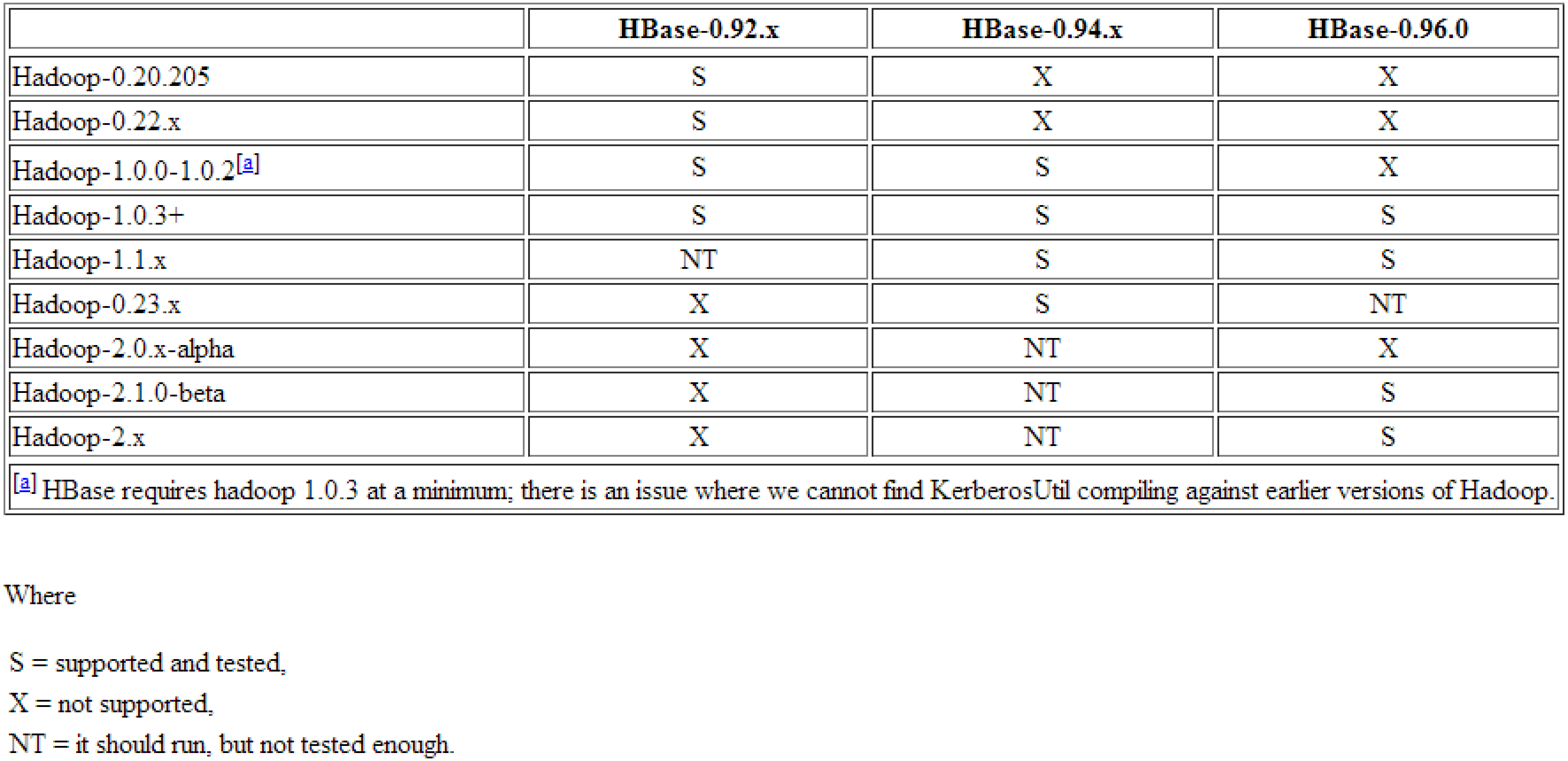
Jeremy Chan
SHB118

cwchan@cse.cuhk.edu.hk

Prerequisite

- A functional Hadoop configuration
  - Follow **Introduction to Cloud Platform 02**
  - Double check the following configurations
    - `/etc/hosts`
    - Environment Variables: `HADOOP_HOME, PATH`
    - `hadoop/conf/hadoop-env.sh`
    - `hadoop/conf/core-site.xml`
    - `hadoop/conf/mapred-site.xml`
    - `hadoop/conf/hdfs-site.xml`
    - `hadoop/conf/masters`
    - `hadoop/conf/slaves`
  - Test with `hadoop dfsadmin -report`

HBase Version



	HBase-0.92.x	HBase-0.94.x	HBase-0.96.0
Hadoop-0.20.205	S	X	X
Hadoop-0.22.x	S	X	X
Hadoop-1.0.0-1.0.2[a]	S	S	X
Hadoop-1.0.3+	S	S	S
Hadoop-1.1.x	NT	S	S
Hadoop-0.23.x	X	S	NT
Hadoop-2.0.x-alpha	X	NT	X
Hadoop-2.1.0-beta	X	NT	S
Hadoop-2.x	X	NT	S

[a] HBase requires hadoop 1.0.3 at a minimum; there is an issue where we cannot find KerberosUtil compiling against earlier versions of Hadoop.

Where

S = supported and tested,
X = not supported,
NT = it should run, but not tested enough.

http://hbase.apache.org/book/configuration.html

Step 1: Downloading HBase

- We use HBase 0.92.2

http://archive.apache.org/dist/hbase/hbase-0.92.2/hbase-0.92.2.tar.gz

- Untar it into ~/hbase
- Set environment variables (e.g. in .bashrc)
  - HBASE_HOME=~/hbase
  - PATH=$HBASE_HOME/bin:$PATH
  - HADOOP_CLASSPATH=`hbase classpath`

- **The following slides assume**
  - Namenode: test1
  - Datanodes: test2, test3, test4

Step 2: HBase Environment

- Make the following changes in **`~/hbase/conf/hbase-env.sh`**

- export JAVA_HOME=/usr/lib/jvm/[JAVA PATH]
- export HBASE_MANAGES_ZK=true
  - Let HBase to manage Zookeeper

Step 3: HBase Connection to HDFS

- Make the following changes in **~/hbase/conf/hbase-site.xml**

```
<property>
    <name>hbase.master</name>
    <value>test1:60000</value>
</property>
<property>
    <name>hbase.rootdir</name>
    <value>hdfs://test1:54310/hbase</value>
</property>
<property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
</property>
<property>
    <name>hbase.zookeeper.quorum</name>
    <value>test2,test3,test4</value>
</property>
```

Use the namenode

Use the same address as *fs.default.name* in core-site.xml

Run HBase in distributed mode

We use datanotes for zookeeper quorum

Step 4: HBase regionservers

- Add the list of datanodes to **`~/hbase/conf/regionservers`**

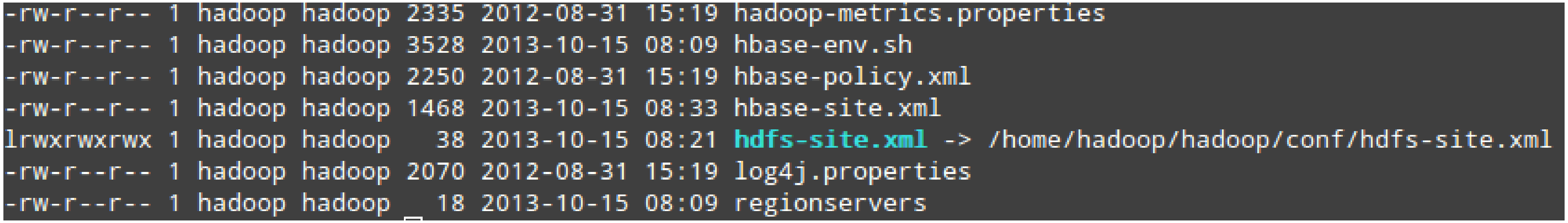
- One host per line
  - Follow format in `~/hadoop/conf/slaves`

Step 5: Copy Hadoop Core

- Remove `hadoop-core-1.0.3.jar` in `~/hbase/lib`
- Copy `hadoop-core-0.20.203.0.jar` from ~/hadoop to `~/hbase/lib`

Step 6: Setup HBase Client

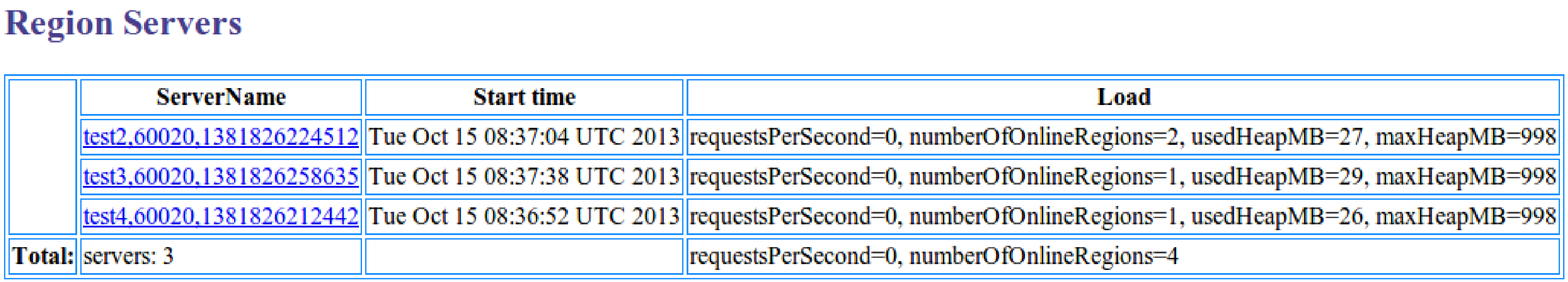
- Add a symbolic link for `hdfs-site.xml` in `~/hbase/conf`
  - `ln -s /home/hadoop/hadoop/conf/hdfs-site.xml /home/hadoop/hbase/conf/hdfs-site.xml`



```
-rw-r--r-- 1 hadoop hadoop 2335 2012-08-31 15:19 hadoop-metrics.properties
-rw-r--r-- 1 hadoop hadoop 3528 2013-10-15 08:09 hbase-env.sh
-rw-r--r-- 1 hadoop hadoop 2250 2012-08-31 15:19 hbase-policy.xml
-rw-r--r-- 1 hadoop hadoop 1468 2013-10-15 08:33 hbase-site.xml
lrwxrwxrwx 1 hadoop hadoop   38 2013-10-15 08:21 hdfs-site.xml -> /home/hadoop/hadoop/conf/hdfs-site.xml
-rw-r--r-- 1 hadoop hadoop 2070 2012-08-31 15:19 log4j.properties
-rw-r--r-- 1 hadoop hadoop   18 2013-10-15 08:09 regionservers
```

Start HBase

- ~/hbase/bin/start-hbase.sh
- Check status in web management page
  - http://test1:60010/master-status
- Ignore this warning (Our Hadoop version does not have HDFS append support)
  - You are currently running the HMaster without HDFS append support enabled



Region Servers

	ServerName	Start time	Load
	test2,60020,1381826224512	Tue Oct 15 08:37:04 UTC 2013	requestsPerSecond=0, numberOfOnlineRegions=2, usedHeapMB=27, maxHeapMB=998
	test3,60020,1381826258635	Tue Oct 15 08:37:38 UTC 2013	requestsPerSecond=0, numberOfOnlineRegions=1, usedHeapMB=29, maxHeapMB=998
	test4,60020,1381826212442	Tue Oct 15 08:36:52 UTC 2013	requestsPerSecond=0, numberOfOnlineRegions=1, usedHeapMB=26, maxHeapMB=998
Total:	servers: 3		requestsPerSecond=0, numberOfOnlineRegions=4

Test HBase in Shell

```
[hduser@localhost ~]$ hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported
commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.90.5, r1212209, Fri Dec 9 05:40:36 UTC 2011
hbase(main):001:0> create 'test', 'data'
0 row(s) in 1.9300 seconds
hbase(main):002:0> list
TABLE
test
1 row(s) in 0.0250 seconds
hbase(main):003:0> put 'test', 'row1', 'data:1', 'value1'
0 row(s) in 0.1970 seconds
```

- You may submit an HBase script to create the tables

Compile and Run HBase Program

- Similar to assignment 1
  - `mkdir wordcount`
  - `` javac -cp `hbase classpath` WordCount.java -d wordcount ``
  - `jar -cvf wordcount.jar -C ./wordcount .`
  - `hadoop jar wordcount.jar org.myorg.WordCount`

Writing Data to HBase

1. Setting up Configuration

```
HBaseConfiguration hbaseConfig = new HBaseConfiguration();
HTable htable = new HTable(hbaseConfig, "bigram_in");
```

Table Name

2. Writing a row using `int` as key and `String` as value

```
int key = 1234;
String val = “abc”;
byte[] rowkey = Bytes.toBytes(key);
Put put = new Put(rowkey);
put.add(Bytes.toBytes("cf"), Bytes.toBytes("line"),
val.getBytes());
htable.put(put);
```

Column Family

Column Family Member

3. Flush HBase

```
htable.flushCommits();
htable.close();
```

Setting up MapReduce on HBase

1. Setting up Configuration

```
Configuration config = HBaseConfiguration.create();
Job job = new Job(config, "HBaseBigram");
Scan scan = new Scan();
TableMapReduceUtil.initTableMapperJob("bigram_in", scan,
       MyMapper.class, Text.class, IntWritable.class, job);
TableMapReduceUtil.initTableReducerJob("bigram_out",
       MyTableReducer.class, job);
job.setNumReduceTasks(1);
```

2. Mapper Prototype

```
public static class MyMapper extends TableMapper<Text, IntWritable>
public void map(ImmutableBytesWritable row, Result value,
       Context context) throws IOException, InterruptedException
```

3. Reducer Prototype

```
public static class MyTableReducer extends TableReducer<Text,
       IntWritable, ImmutableBytesWritable>
public void reduce(Text key, Iterable<IntWritable> values, Context
       context) throws IOException, InterruptedException
```

Reading Data from HBase

1. Scanning all rows in HBase

```
Scan scan = new Scan();
ResultScanner scanner = htable.getScanner(scan);
Result r;
while (((r = scanner.next()) != null)) {
    String key = new String (r.getRow());
    byte[] val = r.getValue(Bytes.toBytes("cf"),
Bytes.toBytes("line"));
    String valString = new String(val);
    System.out.println(key + " " + valString);
}
```

2. Close scanner and connectionHBase

```
scanner.close();
htable.close();
```

Hints

- Feel free to reuse most of your code from Assignment 1

- There are many performance configurations (e.g. buffer) in HBase that you can tune

- Make sure the output table schema is correct
  - Table name: “`bigram_result`”
  - Column family: `result`
    - Member: `count`

- Make sure the filenames are correct
  - `HBaseImport.java`
  - `HBaseBigram.java`
  - `HBaseExport.java`

Questions?

Thank You