BigDataBench-JStorm User Manual

1. Introduction

BigDataBench-JStorm based on the stream computing system JStorm, it contains three benchmark programs:

(1) BigDataBench-JStorm-CfByUser: User-based collaborative recommendation algorithm;

(2) BigDataBench-JStorm-RollingTopNWords: RollingTopNWord algorithm which used to recommend hot topic.

(2) BigDataBench-JStorm-Search: Real-time search based on lucene.

2. Deploy JStorm

Inspired by Apache Storm (Storm is a distributed and fault-tolerant realtime computation system), JStorm is completely implemented from scratch in Java, and provides many features which are much more enhanced. JStorm has been widely used in many enterprise environments and proved robust and stable.

Step 0: Prerequisites

Zookeeper:3.4.6
Java JDK: version 1.6 or later
Python: version 2.6
Zeromq:2.1.7
Jzmq:

Step 1: Download JStorm released package

1. Download the most recent stable release of JStorm from https://github.com/alibaba/jstorm/wiki/Downloads .We recommend version 0.9.6.3 (http://42.121.19.155/jstorm/jstorm-0.9.6.3.zip), which was used and tested in our environment.
2. Unpack the tarball.

```
$ unzip jstorm-0.9.6.3.zip
```

3. Set environment variable *JSTORM_HOME (/path/to/jstorm-0.9.6.3)*, add *$JSTORM_HOME/bin* to your PATH.

```
$vim ~/.bash_profile
```

In the *~/.bash_profile*, add:

```
export JSTORM_HOME=/path/to/jstorm-0.9.6.3
export PATH=$JSTORM_HOME/bin:$PATH
```

```
$ source ~/.bash_profile
```

Step 2: Deploy Zookeeper cluster

1. Download zookeeper package from http://zookeeper.apache.org/releases.html#download. For example, downloading the latest version zookeeper-3.4.7 (http://apache.fayea.com/zookeeper/zookeeper-3.4.7/zookeeper-3.4.7.tar.gz).
2. Unpack the tarball.

```
$tar zxvf zookeeper-3.4.7.tar.gz
```

3. Set environment variable *ZOOKEEPER_HOME (/path/to/zookeeper-3.4.7)*, add *$ZOOKEEPER_HOME/bin* to your PATH.

```
$vim ~/.bash_profile
```

In the *~/.bash_profile*, add:

```
export ZOOKEEPER_HOME=/path/to/zookeeper-3.4.7
export PATH=$ ZOOKEEPER_HOME /bin:$PATH
```

```
$source ~/.bash_profile
```

4. Configure *$ZOOKEEPER_HOME/conf/zoo.cfg. It can be configured as local mode or cluster mode, please refer* ZooKeeper Getting Started Guide.

```
$cd $ZOOKEEPER_HOME/conf

$cp zoo_sample.cfg zoo.cfg
```

In *zoo.cfg*, modify:

```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/home/zookeeper-3.4.3/data
# the port at which the clients will connect
clientPort=2181
```

5. Start Zookeeper.

```
$ZOOKEEPER_HOME/bin/zkServer.sh start
```

6. Stop Zookeeper.

```
$ZOOKEEPER_HOME/bin/zkServer.sh stop
```

Step 3: Install Python

If the python version in current system is 2.4 or higher, please skip this section. You can also use the following steps to install Python. We take Python 2.7.11 as example.

1. Download Python 2.7.11 from https://www.python.org/ftp/python/2.7.11/Python-2.7.11.tgz .
2. Unpack the tarball.

```
$tar zxvf Python-2.7.11.tgz
```

3. Compile and install Python.

```
$cd Python-2.7.2
$./configure
$make
$make install
```

4. Add Python's lib library.

```
$sudo vi /etc/ld.so.conf
```

In *ld.so.conf*, add:

/usr/local/lib/

Then, execute command as follows:

```
$sudo /sbin/ldconfig –v
```

Step 4: Install Java

Note that, if the current OS is 64-bit, please install 64-bit jdk; and if OS is a 32-bit system, then download one jdk for 32 bit.

Step 5: Install zeromq

From JStorm 0.9.0, the default RPC framework is Netty, if JStorm only use netty, zeromq doesn't need to be installed.

1. Download zeromq.

```
$wget http://download.zeromq.org/zeromq-2.1.7.tar.gz
```

2. Unpack the tarball.

```
$tar zxf zeromq-2.1.7.tar.gz
```

3. Compile and install zeromq.

```
$cd zeromq-2.1.7
$./configure
$make
$sudo make install
$sudo ldconfig
```

Note: If you do not have root privileges, or the current user does not have sudo privileges, please execute "./configure --prefix=/home/xxxxx" replace "./configure", /home/xxxx is the installation target directory.

Step 6: Install Jzmq

1. Download Jzmq.

```
$git clone git://github.com/nathanmarz/jzmq.git
```

2. Compile and install jzmq.

```
$cd jzmq
$./autogen.sh
$./configure
$make
$make install
```

Note: If you do not have root privileges, or the current user does not have sudo privileges, execute "./configure --prefix=/home/xxxxx" to replace "./configure", /home/xxxx is the installation target directory.

Step 2: JStorm Configuration

1. Configure *$JSTORM_HOME/conf/storm.yaml*.

```
$vi $JSTORM_HOME/conf/storm.yaml
```

In *storm.yaml*, modify:

```
########### These MUST be filled in for a storm configuration
 storm.zookeeper.servers:
     - "localhost"

 storm.zookeeper.root: "/jstorm"
```

```
# %JSTORM_HOME% is the jstorm home directory
 storm.local.dir: "%JSTORM_HOME%/data"

 java.library.path: "/usr/local/lib:/opt/local/lib:/usr/lib"

 supervisor.slots.ports:
     - 6800
     - 6801
     - 6802
     - 6803
```

Note: Please refer to JStorm Configuration for more details.

2. Run following commands on the node where the jar packages will be submitted.

```
$mkdir ~/.jstorm

$cp –f $JSTORM_HOME/conf/storm.yaml ~/.jstorm
```

3. Make sure Zookeeper have been started successfully, create /jstorm in Zookeeper.

```
$cd $ZOOKEEPER_HOME/bin

$./zkCli.sh

[zk: 127.0.0.1:2181(CONNECTED) 0 ] create /jstorm ""
```

3. Running the Workloads

3.1 Application Domain- E-commerce

3.1.1 Workload – CfByUser

1. Required Software Stacks

JStorm: 0.9.3.6

Zookeeper: 3.4.6

2. Download workloads

Download the Benchmark form this link:

http://prof.ict.ac.cn/bdb_uploads/bdb_streaming/E-commerce-JStorm.tar.gz

Unpack the downloaded tar file:

```
$tar xzvf E-commerce-JStorm.tar.gz
```

3. Prepare the input

Download the input file from this link:

http://grouplens.org/datasets/movielens/

Put these files in your path, and modify /BigDataBench-JStorm-CfByUser/resource/CfByUser.properties

cfbyuser.testfile.dir=/path/cf-data/test.txt

cfbyuser.trainingfile.dir=/path/cf-data/test_training.txt

Note: these input files also can be divided into small parts and distributed placed on multiple machines.

4. Run the workload

Make sure Zookeeper and JStorm have been successfully started, and then commit the JStorm topology:

```
$cd E-commerce-JStorm/BigDataBench_JStorm_CfByUser/
$vim sumbit.sh
```

Modify 'JStorm_home' as JSTORM_HOME, for example:

JStorm_home=/home/jstorm-0.9.6.3

Run commit.sh:

```
$ ./sumbit.sh
```

The information of selecting workload will be printed on the screen:

…….

[INFO 2015-12-08 11:06:32 StormSubmitter:151 main] Finished submitting topology: CfByUser

5. Collect the running results

The outputs are saved in $JSTORM_HOME/logs/ CfByUser-worker-xxxx.log.out.

The results format is as follows：

bolttime:285,2889,4;223

bolttime:285,2647,3;195

bolttime:285,2889,4;231

bolttime:285,2889,4;238

bolttime:285,2647,3;209

bolttime:285,2611,4;193

bolttime:285,2889,4;233

Blue indicates the processing time of a record on each bolt.

3.2 Application Domain- Social Networks

3.2.1 Workload – RollingTopWords

1. Required Software Stacks

JStorm: 0.9.6.3

Zookeeper: 3.4.6

2. Download workloads

Download the Benchmark form this link:

http://prof.ict.ac.cn/bdb_uploads/bdb_streaming/SocialNetwork-JStorm.tar.gz

Unpack the downloaded tar file:

```
$tar xzvf SocialNetwork-JStorm.tar.gz
```

3. Prepare the input

Spout generate input randomly.

4. Run the workload

Make sure Zookeeper and JStorm have been successfully started, and then commit the JStorm topology:

```
$cd SocialNetwork-JStorm/BigDataBench_JStorm_RollingTopWords/
$vim sumbit.sh
```

Modify 'JStorm_home' as JSTORM_HOME, for example:

JStorm_home=/home/jstorm-0.9.6.3

Run commit.sh:

```
$ ./sumbit.sh
```

The information of selecting workload will be printed on the screen:

…….

[INFO 2015-12-08 11:30:38 StormSubmitter:151 main] Finished submitting topology: RollingTopWords

5. Collect the running results

The outputs are saved in $JSTORM_HOME/logs/RollingTopWords-worker-xxxx.log.out.

The results format is as follows：

[["nathan",1]]

[["mike",28],["nathan",18]]

[["mike",44],["nathan",35]]

[["nathan",62],["mike",57]]

[["nathan",82],["mike",81]]

3.3 Application Domain- SearchEngine

3.3.1 Workload – Search

1. Required Software Stacks

JStorm: 0.9.6.3

Zookeeper: 3.4.6

2. Download workloads

Download the Benchmark form this link:

http://prof.ict.ac.cn/bdb_uploads/bdb_streaming/SearchEngine-JStorm.tar.gz

Unpack the downloaded tar file:

```
$tar xzvf SearchEngine-JStorm.tar.gz
```

3. Prepare the input

Download the input file from this link:

http://prof.ict.ac.cn/bdb_uploads/bdb_streaming/Search-Data.tar.gz

Put these files in your path, and modify /BigDataBench-JStorm-Search/resource/Search.properties

searchwords.dir=/home/search-data/words.txt

indexid.dir=/home/search-data/indexid

index.dir=/home/search-data/index/data-

Note: these input files also can be divided into small parts and distributed placed on multiple machines.

4. Run the workload

Make sure Zookeeper and JStorm have been successfully started, and then commit the JStorm topology:

```
$cd SearchEngine-JStorm/BigDataBench_JStorm_Search/
$vim sumbit.sh
```

Modify 'JStorm_home' as JSTORM_HOME, for example:

JStorm_home=/home/jstorm-0.9.6.3

Run commit.sh:

```
$ ./sumbit.sh
```

The information of selecting workload will be printed on the screen:

…….

[INFO 2015-12-08 11:44:30 StormSubmitter:151 main] Finished submitting topology: Search

5. Collect the running results

The outputs are saved in $JSTORM_HOME/logs/Search-worker-xxxx.log.out.

The results format is as follows：

善 哉 ,550,171+[doc=5598 score=0.8072934 shardIndex=-1, doc=9655 score=0.67402977 shardIndex=-1, doc=6864 score=0.62421775 shardIndex=-1, doc=1342 score=0.6242021 shardIndex=-1, doc=601 score=0.62066185 shardIndex=-1, doc=6646 score=0.61896014 shardIndex=-1, doc=8418 score=0.6062665 shardIndex=-1, doc=8598 score=0.5864191 shardIndex=-1, doc=5071 score=0.060073324 shardIndex=-1, doc=8052 score=0.056206994 shardIndex=-1],1449546337469,0

Red indicates the search word; Green indicates the total time of search; Blue indicates the search results, "doc" indicates the id of related pages, "score" indicates the score of related pages, the higher score represents the greater relevance.

4. Appendix

Format specification

Bold for emphasis.

Italic for fold and file names

`$command` for command lines

Contents for contents in configuration files

Some exception explanations should be put in footnote