Written by Haejoon Lee in KAIST, 2014.10.14

How to test NoSQL & HBase by YCSB

Step1. Installing

JDK Version : java version "1.8.0_20

Hadoop Version : hadoop 1.2.1""

Cassandra : Cassandra 1.2.18

- **wget http://apache.mirror.cdnetworks.com/cassandra/1.2.18/apache-cassandra-1.2.18-src.tar.gz**

YCSB : YCSB 0.1.4

- **wget** https://github.com/downloads/brianfrankcooper/YCSB/ycsb-0.1.4.tar.gz

Installed location : for YCSB /usr/lib/ycsb for Cassandra /usr/lib/Cassandra

OS : Ubuntu 14.04 amd64

Machine: a single Node (localhost)

After installing Cassandra and YCSB in the folders, you should set up the path in environment setting file. (~/.profile or /etc/profile: $CASSANDRA_HOME or $YCSB_HOME).

If you can command 'ycsb' and 'cassandra' in home directory,

/home/jjoon>>ycsb

/home/jjoon>>cassandra

Everything is okay, so far/

Step2. Configuration

Before testing, you should make properties folders in your cassandra directory.

/usr/lib/cassandra>>mkdir {commitlog,log,saved_caches,data}

Change configuration files (/cassandra/conf/log4j-server.properties, /cassandra /conf/Cassandra.yaml)

In log4j-server.properties

Log4j.rootLogger = INFO -> DEBUG

Log4j.appender.R.File = ~/log/system.log

In Cassandra.yaml

Data_file_directories: ~/data

Commitlog_directory: ~/commitlog

Saved_caches_directory: ~/saved_caches

Written by Haejoon Lee in KAIST, 2014.10.14

Step3. Testing

Before testing it by YCSB, you should start Cassandra Server, format tables in Cassandra.

```
>>canssandra –f                    // Starting server
>>canssandra-cli –host localhost // Starting Client to connect with Server
: create keyspace usertable;
: use usertable;
: create column family with comparator=UTF8Type and default_validation_class=UTF8Type;
```

While Cassandra runs,

YCSB load (Execute the load phase) & run (Execute the transaction phase)

```
>>./bin/ycsb load cassandra-10 -P workloads/workloada –p hosts=localhost >> load.log
>>./bin/ycsb run cassandra-10 -P workloads/workloada –p hosts=localhost >>run.log
```

YCSB includes 6 workload (https://github.com/brianfrankcooper/YCSB/wiki/Core-Workloads)

- **workloada**
- **workloadb**
- **workloadc**
- **workloadd**
- **workloade**
- **workloadf**

You can set up each property by using '-p' optional value

-p recordcount=10000 ₩ -p operationcount=10000 ₩ -p workload=com.yahoo.ycsb.workloads.CoreWorkload ₩ -p readallfields=true ₩ -p readproportion=0.5 ₩ -p updateproportion=0 ₩ -p scanproportion=0 ₩ -p insertproportion=0.5 ₩	-p readmodifywriteproportion=0 ₩ -p requestdistribution=zipfian ₩ -p hosts=172.21.81.139,172.21.81.127 ₩ -p cassandra.connectionretries=1 ₩ -p cassandra.operationretries=1 ₩ -p cassandra.readconsistencylevel=ALL ₩ -p cassandra.writeconsistencylevel=ALL ₩ -p cassandra.deleteconsistencylevel=ALL ₩ -threads 10

Written by Haejoon Lee in KAIST, 2014.10.14

Step4. How to test HBase by YCSB

HBase ?

- **HBase is an open source, non-relational, distributed database modeled after Google's BigTable and written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed Filesystem), providing BigTable-like capabilities for Hadoop**

HBase : 0.98.5
- wget http://apache.mirror.cdnetworks.com/hbase/hbase-0.98.5/hbase-0.98.5-hadoop1-bin.tar.gz

After installing HBase in the folders, you should set up the path in environment setting file. (~/.profile or /etc/profile: $HBASE_HOME) >>hbase

<<ZooKeeper 설치>>

If you use HBase, you should install zookeeper in hbase folder.

http://mirror.apache-kr.org/zookeeper/zookeeper-3.4.6/

[hadoop@master ~]$ tar xvzf zookeeper-3.4.6
[hadoop@master ~]$ ln -s /home/hadoop/zookeeper-3.4.6 zookeeper // Symbolic link
[hadoop@master ~]$ cd zookeeper
[hadoop@master zookeeper]$ cd conf
[hadoop@master conf]$ ls
configuration.xsl log4j.properties zoo_sample.cfg
[hadoop@master conf]$ cp zoo_sample.cfg zoo.cfg
[hadoop@master conf]$ vi zoo.cfg

Modify this line in zookeeper/conf/zoo.cfg

- **dataDir=/tmp/zookeeper**
- **Server.1=localhost:2888:3888**

Written by Haejoon Lee in KAIST, 2014.10.14

<<Change Hbase Configuration Files>> /hbase/conf/

< hbase-env.h>

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
#export HBASE_CLASSPATH=/usr/lib/hbase/conf
#export HBASE_REGIONSERVERS=/usr/lib/hbase/conf/regionservers
export HBASE_MANAGES_ZK=true
```

<hbase-site.xml>

```
<configuration>
    <property>
        <name>hbase.rootdir</name>
        <value>hdfs://localhost:9000/hbase</value>
    </property>
    <property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
    </property>
    <property>
        <name>hbase.zookeeper.quorum</name>
        <value>localhost</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>hbase.zookeeper.property.dataDir</name>
        <value>/tmp/zookeeper</value>
    </property>
</configuration>
```

>>./start-hbase.sh // starting HBase jps

Before you run Hbase Shell, the view should be like this picture.

```
[hadoop@master ~]$ jps
3338 NodeManager
2731 NameNode
9484 HMaster
2884 DataNode
3206 ResourceManager
9606 HRegionServer

9421 HQuorumPeer
3061 SecondaryNameNode
```

Written by Haejoon Lee in KAIST, 2014.10.14

>>hbase shell // to create a table on HBase

```
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.99.0, r757a0952ce4674704d91a858b42ea94be8199b81, Tue Sep 16 12:58:15 PDT 2014

hbase(main):001:0> create 'testdata', 'cf'
```

Go inside this newly created YCSB directory and move inside the HBase directory. You will find an xml file here named as pom.xml. Open this pom.xml file and edit it.

```
<dependencies>
    <dependency>
      <groupId>org.apache.hbase</groupId>
      <artifactId>hbase</artifactId>
      <!--<version>${hbase.version}</version>-->
      <version>0.98.5</version>
    </dependency>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-core</artifactId>
      <!--<version>1.2.1</version>-->
      <version>1.2.1</version>
    </dependency>
    <dependency>
      <groupId>com.yahoo.ycsb</groupId>
      <artifactId>core</artifactId>
      <version>${project.version}</version>
    </dependency>
  </dependencies>
```

Command for YCSB after creating Table on HBase

```
>>./bin/ycsb load hbase -P workloads/workloada -p columnfamily=cf -p recordcount=1000000 -p threadcount=4 >> load.log

>>./bin/ycsb run hbase -P workloads/workloada -p columnfamily=cf -p recordcount=1000000 -p threadcount=4 >> load.log
```

Written by Haejoon Lee in KAIST, 2014.10.14

YCSB Information

YCSB에서는 성능(Performance), 확장성(Scalability), 가용성(Availability), 복제(Replication)등 4가지 측면에서 벤치마크를 수행할 수 있도록 해준다. 1차적으로 성능 중심의 테스트를 수행하려고 한다. 성능 벤치마크의 경우 아래와 같이 이미 만들어진 다양한 유형의 작업 부하를 사용할 수 있으며, 필요한 경우 새로운 유형의 작업 부하를 만들 수 있다.

- 성능 작업 부하 유형
■Workload A: 업데이트 중심의 작업 (읽기 50%, 업데이트 50%)
사례) 최근의 액션을 저장하는 세션 정보 어플리케이션
■Workload B: 읽기 중심의 작업 (읽기 95%, 업데이트 5%)
사례) 포토 태그 ? 태그는 한번만 작성하고 주로 읽기 작업이 실행된다.
■Workload C: 읽기 전용 작업 (읽기 100%)
사례) 사용자 프로파일 캐시 ? 외부 저장소에 저장되어 있는 사용자 정보를 조회
■Workload D: 최근 레코드 중심의 읽기 작업 (읽기 95%, 쓰기 5%)
최근 저장된 레코드를 중심으로 읽기 수행
사례) SNS 사용자 상태 업데이트 ? 가장 최근의 상태와 뉴스만이 중요함
■Workload E: 영역 스캔 (읽기 95%, 쓰기 5%)
각 작업마다 100개 내외의 레코드 영역을 한번에 쿼리 한다
사례) 쓰레드 형식의 게시판
■Workload F: 읽기-쓰기-수정
읽기, 수정, 쓰기 작업을 순서대로 수행
사례) 사용자 데이터베이스 ? 각 사용자가 액션 수행 시 레코드를 읽고 업데이트

Reference) http://damul21c.tistory.com/7

Why we moved Cassandra to HBase?

http://ria101.wordpress.com/2010/02/24/hbase-vs-cassandra-why-we-moved/