Data Serving

The data serving benchmark relies on the Yahoo! Cloud Serving Benchmark (YCSB). YCSB is a framework to benchmark data store systems. This framework comes with the interfaces to populate and stress many popular data serving systems. Here we provide the instructions and pointers to download and install YCSB and use it with the Cassandra data store.

Prerequisite Software Packages

1. Cassandra 0.7.3
2. YCSB 0.1.3 Clone from git hub, use:
   git clone git://github.com/brianfrankcooper/YCSB.git
3. Apache Ant and Java JDK (we tested with 1.6.0.23)

Download the Data Serving benchmark.

Important: The distributed YCSB package might not compile with Cassandra 0.7.x. This is why the code should be obtained from the git repository.

Preparing YCSB

The following YCSB website provides detailed instructions to install YCSB. Here we summarize all the instructions that are necessary to prepare YCSB to benchmark the Cassandra data store. YCSB consists of two main components:

- The client interface is used to benchmark a specific data store, (Cassandra 0.7.3 in our case).
- The workload specifies the distribution of requests (read/write ratio, key) that clients send to the data store.

1. After downloading and unpacking the YCSB framework either from the data serving benchmark package or from github as mentioned above:
   - *cd YCSB*
   - *ant* (Make sure you set the *JAVA_HOME* parameter correctly and install the Apache Ant tool before this step).
2. Build the client interface (database layers)

   Because YCSB can be used with various database systems, we need to build the specific interface for Cassandra. To do so, we need to copy the jar files from the Cassandra package to YCSB.

   - Unpack the downloaded Cassandra file. You may want to download (and later install) Cassandra on a different machine than the one that runs the client.
   - *cd $CASSANDRA_PATH/lib/* (in our case this should be *apache-cassandra-0.7.3/lib*)
     Make sure that all the jar files exist in this directory. The Cassandra-0.7.3

source distribution didn't have the apache-cassandra-0.7.3.jar file. This file is included in the bin distribution we are using in this tutorial.

- Copy all the jar files into the *YCSB/db/cassandra-0.7/lib/* directory on the client machine. (You can use the *scp* command to copy the files between different machines). If the client is running on the same machine as Cassandra then the command will be

  *cp *.jar ~/YCSB/db/cassandra-0.7/lib/*

Then:

- On the client machine *cd ~/YCSB/*
- *ant*
- *ant dbcompile-cassandra-0.7*

This will create the *ycsb.jar* file that contains all the classes needed to run the YCSB benchmark in the *build/* directory.

Finally copy the files to load and run the database to YCSB folder.

- *cd ~/YCSB*
- *cp ../run* ./*
- *cp settings* ./*

Installing Cassandra

The Cassandra project website provides detailed instructions for installing and setting up various Cassandra distributions. Here, we provide the main instructions required to have an operational Cassandra 0.7:

1. Running Cassandra on a single node:
   - cd to the unpacked Cassandra directory
   - *ant*
   - Check the configuration parameters: *conf/cassandra.yaml* contains default values for the Cassandra parameters. First, ensure that the paths for the following parameters point to the directories where you have the write permissions.
     data_file_directories, *commitlog_directory*, and *saved_caches_directory*.
   - In *conf/log4j-server.properties*, make sure that the parameter: *log4.appender.R.File* is set to the directories of your choice that you have write permissions to.
   - Set your *JAVA_HOME* environment variable properly.
   - Run Cassandra by invoking: *bin/cassandra -f*
     (if you do not see error messages, then you have a good chance of successful installation.)
   - Optionally you can follow the instructions in the README file in the installation folder for testing your installation.

2. Running Cassandra on a cluster of nodes: The instructions to run Cassandra are almost the same as installing multiple Cassandra data stores on multiple nodes. However, you need to configure each Cassandra instance properly to communicate with each other. The way that a Cassandra node is designed to communicate with other nodes is through the Gossip protocol. Each Cassandra node should know at least one reliable Cassandra node called the seed. More details at the link above.
    - Configure the seed for each node in the *conf/cassandra.yaml* file under the *seeds* directory.
    - Configure the *listen_address* and *rpc_address* in *conf/cassandra.yaml* to the hostname (or the IP address of the node).

Generating the dataset

The YCSB client has a data generator. After starting Cassandra, YCSB can load data. First, you need to create a keyspace named *usertable* and a column family for YCSB. This is a must for YCSB to load data and run. In order to create a keyspace and a column family, you can use the following commands after connecting to the server with *cassandra-cli* utility under *$CASSANDRA_PATH/bin*.

1. *create keyspace usertable with replication_factor=1;*
   Note: the semicolon is important in the all the commands.
2. *use usertable;*
3. *create column family data with column_type = 'Standard' and comparator = 'UTF8Type';*
4. *exit;*

   Then

5. *cd $YCSB_PATH*
6. To generate the standard data set, we provide two files: *settings_load.dat* and *run_load.command*.
   The first file, *settings_load.dat*, specifies several parameters related to the generated data, mainly:
    - hosts: specifies the IP address of the machine running Cassandra.
    - recordcount: specifies the number of records to be loaded in the data store.

   The second file, *run_load.command*, starts loading the database with the parameters specified in settings_load.dat file :
   java -cp build/ycsb.jar:db/cassandra-0.7/lib/ com.yahoo.ycsb.Client -load -s -db com.yahoo.ycsb.db.CassandraClient7 -P workloads/workloada -P settings_load.dat*

   Note that the above command will load the data necessary for “workloada”. To specify other workload mixes, you only need to change the name in the *run.command* file.

More detailed instructions on generating the data set can be found in Step 5 at this link. Although Step 5 in the link describes the data loading procedure other steps (e.g., 1, 2, 3, 4) are very useful to understand the YCSB settings.

A rule of thumb on the data set size:

To emulate a realistic setup, you can generate more data than your main memory size if you have a low-latency, high-bandwidth I/O subsystem. For example, for a machine with 24GB memory, you can generate 30 million records corresponding to a 30GB data set size

Note: The data set resides in Cassandra's data folder(s).The actual data takes up more space than the total size of the records because data files have metadata structures (e.g., index). Make sure you have enough disk space.

Tuning the server performance

1. In general the server settings are under the *$CASSANDRA_PATH/conf* folder. The main file is *cassandra.yaml*. The file has comments about all parameters. They can also be found here: http://wiki.apache.org/cassandra/StorageConfiguration
2. You can modify the *target* and *threadcount* variables to tune the benchmark and utilize the server. The throughput depends on the number of hard drives on the server. If there are enough disks, the cores can be utilized after running the benchmark for 10 minutes. Make sure that half of the main memory is free for the operating system file buffers and caching.
3. Additionally, the following are useful pointers for performance tuning:
    - http://spyced.blogspot.com/2010/01/linux-performance-basics.html
    - http://wiki.apache.org/cassandra/MemtableThresholds

Running the benchmark

After you install and run the server, install the YCSB framework files and populate Cassandra, you are one step away from running the benchmark.
To specify the run time parameters for the client, a good practice is to create a settings file. You can keep the important parameters (e.g., *target*, *threadcount*, *hosts*, *operationcount*, *recordcount*) in this file, similar to what we did in the data generation phase.

In the package, we provide two files to facilitate the run phase. The first is *settings.dat* and the second is *run.command*.

The settings.dat file defines the IP address(es) of the node(s) running Cassandra, in addition to the *recordcount* parameter (which should be less than or equal to the number specified in the data generation step to avoid potential errors). The *operationcount* parameter sets the number of operations to be executed on the data store.

On the other hand, *run.command* takes the *settings.dat* file as an input and runs the following command:

java -cp build/ycsb.jar:db/cassandra-0.7/lib/ com.yahoo.ycsb.Client -t -s -db com.yahoo.ycsb.db.CassandraClient7 -P workloads/workloada -P settings.dat*

To keep the benchmark running for a long time, you can override the *operationcount* variable.

Step 6 at this link provides detailed instructions on running the benchmark.