YCSB – Yahoo! Cloud Serving Benchmark

YCSB is a framework for bench-marking systems. The primary goal of YCSB is to develop framework and common set of workloads to evaluate the different 'Key – value' and 'Could' serving stores.

There is a core workloads executed by the generator which gives complete picture of a system performance. Also we can extend the client to perform the extra scenario checks which is not covered by the core workloads and client can support different databases bench-marking as well. It is used to benchmark and compare multiple systems (Install same hardware configuration and run same workloads).

How to add new database?

We can add a new database by *implementing a database interface layer*. This layer helps us to hide details of the database also helps client generate operations without understanding database specific API. The database interface layer is simple abstract class which consist of Read, insert, update, delete and scan operations of the database.

How to create a new Layer?

1. Create a project module for the database

The database interface modules are directories at the top level YCSB project. Name the module after your database and each module in the project is considered to be a self contained java project.

Create top level directory which includes a pom.xml to specify any dependencies your interface needs to compile and README.md to specify your interface details and Maven project structure will include all the java classes.

2. Extend com.yahoo.ycsb.DB

Base class of all database interface layer implementation is com.yahoo.ycsb.DB class. This is an abstract class, create a new class which extends DB class. The created new class should be public no-argument constructor, because the instances inside the factory which requires no-argument constructor.

The client will create once instance for the database per worker thread. If multiple worker threads generated for the workloads, multiple instances will be created for our DB class.

3. *Implement init()*

The init() method is used to connect to database and any other initialization needed. The method will be called per DB instance. Also properties to configure the layer can be changed using the parameter file.

public void init() throws DBException

The properties changed will pass to the DB instance after the constructor. Properties should be retrieve by init() not by the constructor.

4. *Implement the database query and update methods*

The methods used to implement the database query are

Read() - Reads a single record
Scan() -Perform a range scan
Update() - Update single record
Insert() - Insert single record
Delete() - Delete single record

The mentioned methods takes table name and record key. Read() and Scan() provide a structure Hashmap to store returned data. Insert() and Update() uses hashmap to map field name to value.

5. *Compile database interface layer*

Compile the code separately from YCSB client and framework. Change the DB class and recompile without recompiling the YCSB client.

6. *Test database layer*

The simple way to test the layer is to create the DB instance and allows to interact directly to the database without starting any workload.

7. *Use it with the YCSB client*

Implemented classes and necessary libraries are available in the classpath. Run the YCSB client, specify the DB argument and provide fully qualified class name of the database.

References:

1. https://github.com/brianfrankcooper/YCSB/wiki/Getting-Started
2.https://github.com/brianfrankcooper/YCSB/wiki
3.https://github.com/brianfrankcooper/YCSB/wiki/Adding-a-Database