Documentation

Tajo Benchmark on Cloudera CDH

Andreas Grimm

[redacted]

February 2016

Goethe-Universität Frankfurt am Main

Fachbereich 12: Informatik und Mathematik

Institut für Informatik

Working time:
Oktober 2015 - February 2016

Supervisor:
Todor Ivanov

Contents

1 Introduction

About Tajo:

- compatible with ANSI/ISO SQL - SQL like syntax
- advanced query optimization and lot's of optimization techniques possible
- used for low-latency and scalable ad-hoc queries
- HDFS or external sources for data possible
- user defined functions, interactive shell

2 Installation

2.1 Prerequisites

i) Hadoop 2.3.0 or higher (up to 2.6.0)

ii) Java 1.7 or higher

iii) Protocol buffer 2.5.0

The first two work out of the box on CDH 5.4.2 but the protocol buffer 2.5.0 has to be installed manually. The steps are:

```
>> wget http://protobuf.googlecode.com/files/protobuf-2.5.0.tar.gz
>> tar -zxvf protobuf-2.5.0.tar.gz
>> cd protobuf-2.5.0
>> ./configure
>> make
>> sudo make install
>> export PATH=$PATH:/usr/local/bin
```

It is recommended to install the commons-logging-1.2 package manually. Go to http://commons.apache.org/proper/commons-logging/download_logging.cgi and download the source package (commons-logging-1.2-src.tar.gz). Go to the target directory of the download and do the following:

```
>> cd commons-logging-1.2-src
>> mvn package
```

If the build fails, try to add the following plugin to the pom file:

```
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>exec-maven-plugin</artifactId>
    <version>1.2</version>
</plugin>
```

2.2 Download tajo

Try to get tajo using

```
>> git clone https://git-wip-us.apache.org/repos/asf/tajo.git
```

This failed in my case. The other way, which is also recommended, is to download it directly from http://www.apache.org/dyn/closer.cgi/tajo. I used tajo version 0.10.0. This makes sence, since the tutorials that can be found out in the net are mostly for 0.9.0 or 0.8.0. Using a even higher version than 0.10.0 might cause unnecessary troubles due to incompatibilities. Navigate to the download directory:

```
>> tar xzvf tajo-0.10.0-src.tar.gz
```

2.3 Configure the project pom file with the CDH5 maven repository

Do some modifikations on several pom.xml files as described at https://cwiki.apache.org/confluence/display/TAJO/How+to+Build+Apache+Tajo+for+CDH5. The modifications for cloudera CDH 5.4.2 can be found in Appendix B. Note that some of these mods are not necessary for the following. After the modification go to tajo-0.10.0-src (in my case /home/cloudera/Desktop/tajo-0.10.0-src/) and execute

```
>> mvn clean install package -DskipTests -Pdist
   -Dhadoop.version=2.6.0-cdh5.4.2
```

This should create a local build of apache tajo on CDH 5.4.2 on CentOs 6.4 using Red Hat as operating system.

3 Configuration

Now do the required modifications on the config file. Since I downloaded the tajo distribution to the desktop the path is:
/home/cloudera/Desktop/tajo-0.10.0-src/tajo-dist/target/tajo-0.10.0/ Open /conf/tajo-env.sh and add the following lines:

```
export /usr/lib/hadoop
export JAVA_HOME=/usr/java/jdk1.7.0_67-cloudera
export TAJO_CLASSPATH=$HADOOP_HOME/client/*:$HADOOP_HOME/lib/*
```

Some sources say that the hadoop home is /opt/cloudera/parcels/CDH/lib/hadoop but that is only the case if you upgraded using parcels. The tutorial on the official tajo homepage says that the last path is not needed, but in my case it was to run tajo. After that the local tajo cluster should be ready to start:

```
sudo ./bin/start-tajo.sh
```

Finally you can start the tajo-shell:

```
sudo ./bin/tsql
```

After each session one should probably stop the cluster using:

```
sudo ./bin/stop-tajo.sh
```

4 TPCH Benchmark

First create a new database:

```
CREATE DATABASE tpch;

\c tpch;
```

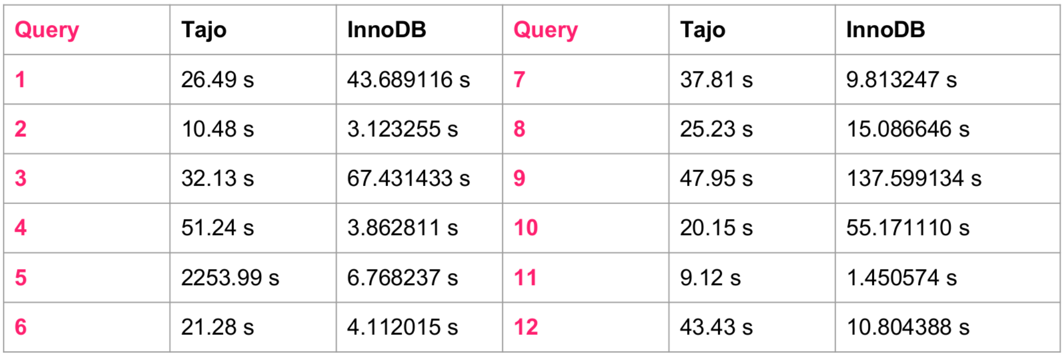
Import tables and data using the statements listed in App. C. The external data was created using the data generator from https://github.com/xinglin/tpch and the step-by-step guide from https://sites.google.com/site/halitsch88/Implementation-TPC-H-schema-into-MySQL-DBMS.
Most of the TPCH queries have to be modified. The following sources werde used:
https://cwiki.apache.org/confluence/display/TAJO/TPC-H+Benchmark+Set
https://s3-us-west-2.amazonaws.com/tajo/taas/documents/Analyze_Data_with_Tajo.pdf
Since some queries have to be split into several commands it's useful to measure the total execution time via

```
 /usr/bin/time -f Time:%e ./tsql -f querieX.sql databaseY > resultZ.txt
```

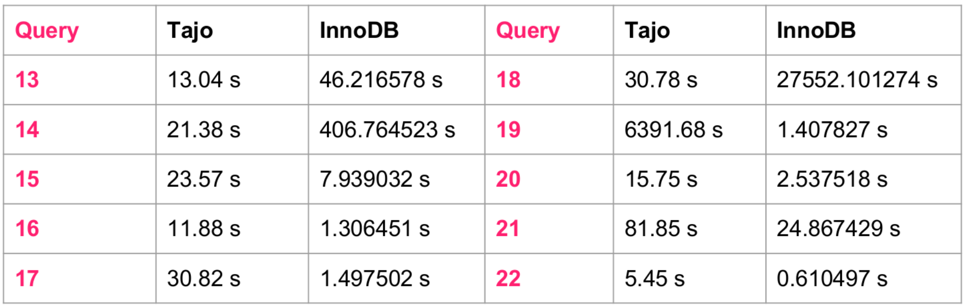
where querieX.sql is the executed querie, databaseY the database withe the stored data, and the results are stored in resultZ.txt.

5 Results

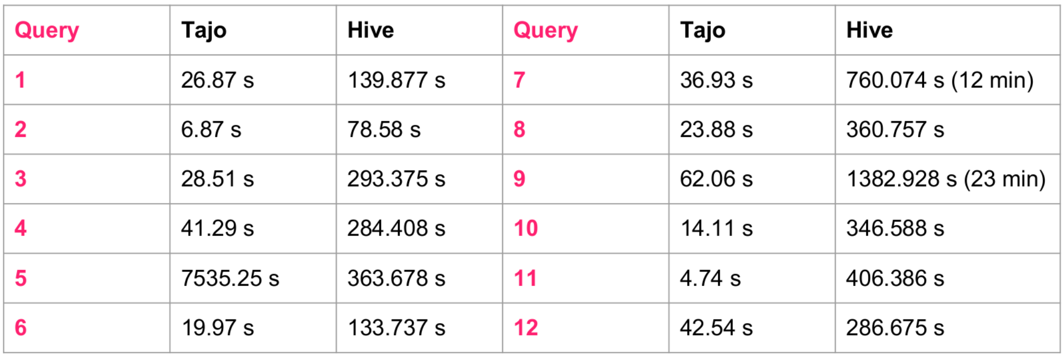
Results for the benchmarks using 1GB in the case innodb vs. tajo and 2GB in the case hive vs tajo.



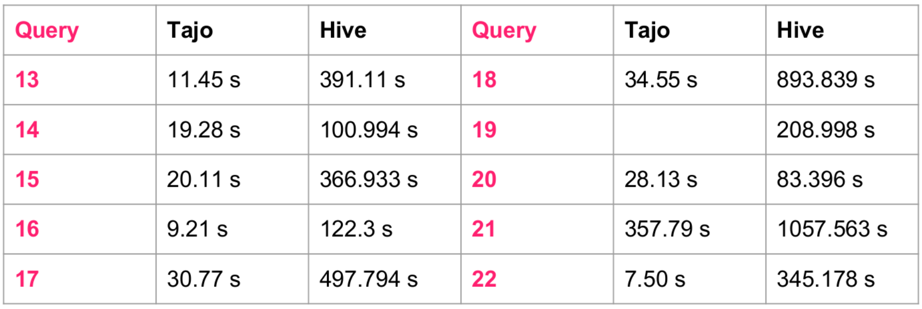
Query	Tajo	InnoDB	Query	Tajo	InnoDB
1	26.49 s	43.689116 s	7	37.81 s	9.813247 s
2	10.48 s	3.123255 s	8	25.23 s	15.086646 s
3	32.13 s	67.431433 s	9	47.95 s	137.599134 s
4	51.24 s	3.862811 s	10	20.15 s	55.171110 s
5	2253.99 s	6.768237 s	11	9.12 s	1.450574 s
6	21.28 s	4.112015 s	12	43.43 s	10.804388 s



Query	Tajo	InnoDB	Query	Tajo	InnoDB
13	13.04 s	46.216578 s	18	30.78 s	27552.101274 s
14	21.38 s	406.764523 s	19	6391.68 s	1.407827 s
15	23.57 s	7.939032 s	20	15.75 s	2.537518 s
16	11.88 s	1.306451 s	21	81.85 s	24.867429 s
17	30.82 s	1.497502 s	22	5.45 s	0.610497 s



Query	Tajo	Hive	Query	Tajo	Hive
1	26.87 s	139.877 s	7	36.93 s	760.074 s (12 min)
2	6.87 s	78.58 s	8	23.88 s	360.757 s
3	28.51 s	293.375 s	9	62.06 s	1382.928 s (23 min)
4	41.29 s	284.408 s	10	14.11 s	346.588 s
5	7535.25 s	363.678 s	11	4.74 s	406.386 s
6	19.97 s	133.737 s	12	42.54 s	286.675 s



Query	Tajo	Hive	Query	Tajo	Hive
13	11.45 s	391.11 s	18	34.55 s	893.839 s
14	19.28 s	100.994 s	19		208.998 s
15	20.11 s	366.933 s	20	28.13 s	83.396 s
16	9.21 s	122.3 s	21	357.79 s	1057.563 s
17	30.77 s	497.794 s	22	7.50 s	345.178 s

A Summary

- no primary keys supported by tajo
- modified sql syntax: ‘10-10-2010’::date
- no scalar subqueries
- no views
- case-insensitive
- “not null“ not allowed
- use text in stead of varchar

- buggy error messages: Error: no error message
- some queries hang without reason
- drop table if exists sometimes does not seem to work, temporary tables had to be dropped manually

B Modifications on pom files

All paths in this section refer to the directory you unpacked the tajo distribution.
For example in my case /tajo-project/pom.xml referred to the directory /home/cloudera/Desktop/tajo-0.10.0-src/tajo-project/pom.xml assuming you downloaded the tajo distribution to
the desktop.

B.1 tajo-project

If one of the following is missing in /tajo-project/pom.xml, add it.

```
<repositories>
    <repository>
      <id>apache.snapshots</id>
      <url>http://repository.apache.org/snapshots</url>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </repository>

    <repository>
      <id>eclipse-jetty</id>
      <url>
      http://repo2.maven.org/maven2/org/eclipse/jetty/jetty-distribution/
      </url>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>

    <repository>
      <id>cloudera</id>
      <url>
      https://repository.cloudera.com/artifactory/cloudera-repos/
      </url>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
</repositories>
```

B.2 tajo-catalog

Add the following profile to tajo's catalog pom file at /tajo-catalog/tajo-catalog-drivers/pom.xml:

```
<profile>
        <id>hcatalog-cdh5.4.2</id>
        <activation>
          <activeByDefault>false</activeByDefault>
        </activation>
        <modules>
          <module>tajo-hcatalog</module>
        </modules>
</profile>
```

Adding the following to /tajo-dist/pom.xml did not work for me:

```
if [ -f $ROOT/tajo-catalog/tajo-catalog-drivers/tajo-hcatalog/target/lib/
                                                  hive-hcatalog-core-*-cdh*.jar ]
then
run cp -r $ROOT/tajo-catalog/tajo-catalog-drivers/tajo-hcatalog/target/
                                        lib/hive-hcatalog-core-*-cdh*.jar lib/
fi
```

B.3 tajo-hcatalog

/tajo-catalog/tajo-catalog-drivers/tajo-hcatalog/pom.xml:

```
 <profile>
        <repositories>
          <repository>
            <id>cloudera</id>
            <url>
            https://repository.cloudera.com/artifactory/cloudera-repos/
            </url>
            <snapshots>
              <enabled>true</enabled>
            </snapshots>
          </repository>
        </repositories>
        <id>hcatalog-cdh5.4.2</id>
        <activation>
          <activeByDefault>false</activeByDefault>
        </activation>
        <properties>
          <hive.version>1.1.0-cdh5.4.2</hive.version>
        </properties>
        <dependencies>
          <dependency>
            <groupId>javax.jdo</groupId>
```

```
    <artifactId>jdo2-api</artifactId>
    <version>2.3-eb</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.hive</groupId>
    <artifactId>hive-exec</artifactId>
    <version>${hive.version}</version>
    <scope>provided</scope>
    <exclusions>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-common</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-contrib</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-hbase-handler</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-metastore</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-serde</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-shims</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-testutils</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.thrift</groupId>
        <artifactId>libfb303</artifactId>
      </exclusion>
      <exclusion>
        <groupId>org.apache.thrift</groupId>
        <artifactId>libthrift</artifactId>
      </exclusion>
      <exclusion>
        <groupId>javax.jdo</groupId>
        <artifactId>jdo2-api</artifactId>
      </exclusion>
    </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.hive</groupId>
  <artifactId>hive-metastore</artifactId>
  <version>${hive.version}</version>
  <scope>provided</scope>
  <exclusions>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-common</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-serde</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-shimss</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.thrift</groupId>
      <artifactId>libfb303</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.thrift</groupId>
      <artifactId>libthrift</artifactId>
    </exclusion>
    <exclusion>
      <groupId>javax.jdo</groupId>
      <artifactId>jdo2-api</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.hive</groupId>
  <artifactId>hive-cli</artifactId>
  <version>${hive.version}</version>
  <scope>provided</scope>
  <exclusions>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-common</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-exec</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-metastore</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-serde</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-service</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-shims</artifactId>
    </exclusion>
    <exclusion>
      <groupId>javax.jdo</groupId>
      <artifactId>jdo2-api</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.hive.hcatalog</groupId>
  <artifactId>hive-hcatalog-core</artifactId>
  <version>${hive.version}</version>
  <scope>provided</scope>
  <exclusions>
    <exclusion>
      <groupId>org.apache.hive</groupId>
      <artifactId>hive-cli</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.hive</groupId>
```

```
          <artifactId>hive-common</artifactId>
        </exclusion>
        <exclusion>
          <groupId>org.apache.hive</groupId>
          <artifactId>hive-exec</artifactId>
        </exclusion>
        <exclusion>
          <groupId>org.apache.hive</groupId>
          <artifactId>hive-metastore</artifactId>
        </exclusion>
        <exclusion>
          <groupId>org.apache.hive</groupId>
          <artifactId>hive-serde</artifactId>
        </exclusion>
        <exclusion>
          <groupId>org.apache.hive</groupId>
          <artifactId>hive-service</artifactId>
        </exclusion>
        <exclusion>
          <groupId>org.apache.hive</groupId>
          <artifactId>hive-shims</artifactId>
        </exclusion>
        <exclusion>
          <groupId>javax.jdo</groupId>
          <artifactId>jdo2-api</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-mapreduce-client-core</artifactId>
      <version>${hadoop.version}</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
</profile>
```

C Import statements

C.1 Supplier

```
create external table SUPPLIER (
    S_SUPPKEY bigint,
    S_NAME text,
    S_ADDRESS text,
    S_NATIONKEY bigint,
    S_PHONE text,
    S_ACCTBAL double,
    S_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/supplier.tbl';
```

C.2 Lineitem

```
create external table LINEITEM (
    L_ORDERKEY bigint,
    L_PARTKEY bigint,
    L_SUPPKEY bigint,
    L_LINENUMBER bigint,
    L_QUANTITY double,
    L_EXTENDEDPRICE double,
    L_DISCOUNT double,
    L_TAX double,
    L_RETURNFLAG text,
    L_LINESTATUS text,
    L_SHIPDATE date,
    L_COMMITDATE date,
    L_RECEIPTDATE date,
    L_SHIPINSTRUCT text,
    L_SHIPMODE text,
    L_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/lineitem.tbl';
```

C.3 Part

```
create external table PART (
    P_PARTKEY bigint,
    P_NAME text,
    P_MFGR text,
    P_BRAND text,
    P_TYPE text,
    P_SIZE integer,
    P_CONTAINER text,
    P_RETAILPRICE double,
    P_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/part.tbl';
```

C.4 Partsupp

```
create external table PARTSUPP (
    PS_PARTKEY bigint,
    PS_SUPPKEY bigint,
    PS_AVAILQTY int,
    PS_SUPPLYCOST double,
    PS_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/partsupp.tbl';
```

C.5 Customer

```
create external table CUSTOMER (
    C_CUSTKEY bigint,
    C_NAME text,
    C_ADDRESS text,
    C_NATIONKEY bigint,
    C_PHONE text,
    C_ACCTBAL double,
    C_MKTSEGMENT text,
    C_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/customer.tbl';
```

C.6 Orders

```
create external table ORDERS (
    O_ORDERKEY bigint,
    O_CUSTKEY bigint,
    O_ORDERSTATUS text,
    O_TOTALPRICE double,
    O_ORDERDATE date,
    O_ORDERPRIORITY text,
    O_CLERK text,
    O_SHIPPRIORITY int,
    O_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/orders.tbl';
```

C.7 Nation

```
create external table NATION (
    N_NATIONKEY bigint,
    N_NAME text,
    N_REGIONKEY bigint,
    N_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/nation.tbl';
```

C.8 Region

```
create external table REGION (
    R_REGIONKEY bigint,
    R_NAME text,
    R_COMMENT text)
using text with ('text.delimiter'='|')
location 'file:/path/to/table/region.tbl';
```