Threeglav.com

StreamHorizon – Test Demo deployment Guide

Version 3.0.2


STREAM
HORIZON
DataProcessingPlatform

TABLE OF CONTENTS

Introduction 3
Installation requirements 3
Demo directory structure 3
Configuring StreamHorizon demo 4
Oracle demo 4
To test Oracle JDBC deployment 4
To test Oracle in EXTERNAL TABLE (bulk load) mode 4
MySQL demo 5
To test MySQL JDBC deployment 5
To test MySQL in bulk load mode 5
Running StreamHorizon demo 6
What is desired treadpool size? 6
Starting streamHorizon instance 6
Checking throughput of your server for Oracle database 6
Checking throughput of your server for MySQL database 7
Realistic Testing 7
Throughput guidelines 7
Realistic data 8

INTRODUCTION

INSTALLATION REQUIREMENTS

Thank you for downloading StreamHorizon test demo.

We will refer to directory where you have installed the demo as $ENGINE_HOME.

There are few requirements that need to be met in order to run StreamHorizon.

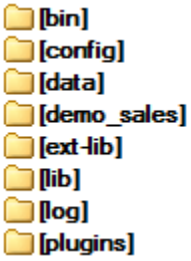
1. Mainstream operating system (Linux, Windows, Solaris)
2. JDK 1.7+ (we recommend Oracle HotSpot)
3. Database must support JDBC interface.

NOTE: Even if you intend to run multiple (clustered) StreamHorizon instances there is no need to install binaries more than one per single (physical) server.

NOTE: 64-bit Java deployment is desireable (but not mandatory) for performance reasons

DEMO DIRECTORY STRUCTURE

Demo you have downloaded comes in two flavours, Oracle and MySQL. Demo is located in $ENGINE_HOME/demo_sales/ directory as shown below.



Directory name	Purpose
$ENGINE_HOME/bin/	All startup scripts should be placed in this directory. There are two default scripts (start.bat and start.sh) that can be utilised for starting single instance (default). If you are customizing new startup scripts always place them in this directory.
$ENGINE_HOME/config/	All configuration files are placed in this directory: • engine-config.xml is single most important configuration file which desciribes database connectivity, directory structues and your database (demo database model).
$ENGINE_HOME/data/	This is engine private directory used for housekeeping. You should not add, delete or modify content of this folder.
$ENGINE_HOME/demo_sales/	Demo feature demonstrating capabilities of StreamHorizon engine (performance and functionality).

$ENGINE_HOME/ext-lib/	Additional dependencies required by your custom plugins.
$ENGINE_HOME/lib/	External dependencies needed by engine are placed here. You should not add, delete or modify anything within this directory.
$ENGINE_HOME/log/	Engine log files can be found here.
$ENGINE_HOME/plugins/	Java plugins. Plain java (*.java) files can be placed in this directory. Engine will compile and load them (when configured to do so IN ENGINE-CONFIG.XML FILE).

CONFIGURING STREAMHORIZON DEMO

ORACLE DEMO

Oracle deployment comes in two flavours, JDBC and external tables. JDBC performs slower than external tables generally, however setup script for external tables requires you to create Oracle directories (as instructed in execution script).

TO TEST ORACLE JDBC DEPLOYMENT

1. Move $ENGINE_HOME/demo_sales/oracle/jdbc_deploy/engine-config.xml file into $ENGINE_HOME/config/ folder.
2. Open engine-config.xml file and change parameters commented out with string "SET ME!"
3. Execute Oracle script StreamHorizon/demo_sales/oracle/jdbc_deploy/oracle_create_schema.sql. Please read NOTE's at the beginning of the script prior to execution
4. Copy sample file $ENGINE_HOME /demo_sales/sales_20140107_data.csv and file multiplier script file_multiplier.sh (or .bat) to your <sourceDirectory> as configured in engine-config.xml. Execute file_multiplier script in order to create 500 files.
5. NOTE: by default data in fact table (sales_fact) will be stored in your default tablespace. This tablespace (if not big enough) may be filled up during processing and if so will cause errors in StreamHorizon logs (StreamHorizon/log/sh-engine*log). To avoid this either edit file multiplier script file_multiplier to create less than 500 copies of the feed files or extend your default tablespace. Alternatively, change create table statement for sales_fact table to point to tablespace of your choice.
6. Please skip to section RUNING STREAM HORIZION DEMO

TO TEST ORACLE IN EXTERNAL TABLE (BULK LOAD) MODE

1. Move $ENGINE_HOME/demo_sales/oracle/ext_table_deploy/engine-config.xml into StreamHorizon/config/ folder.
2. Open engine-config.xml and change parameters commented out with string "SET ME!"
3. If you have already executed $ENGINE_HOME/demo_sales/oracle/jdbc_deploy/oracle_create_schema.sql as part of install of JDBC setup please skip next step.
4. Execute Oracle script $ENGINE_HOME/demo_sales/oracle/ext_table_deploy/oracle_create_schema.sql. Please read NOTE's at the beginning of the script prior to execution.

5. Execute Oracle script $ENGINE_HOME/demo_sales/oracle/ext_table_deploy/externalTableCreate.sql. Please read NOTE's at the beginning of the script prior to execution
6. Copy sample file $ENGINE_HOME/demo_sales/sales_20140107_data.csv and file multiplier script file_multiplier.sh (or .bat) to your <sourceDirectory> as configured in engine-config.xml. Execute file multiplier script in order to create 500 files.
7. NOTE: by default data in fact table (sales_fact) will be stored in your default tablespace. This tablespace (if not big enough) may be filled up during processing and if so will cause errors in StreamHorizon logs (StreamHorizon/log/sh-engine*log). To avoid this either edit file multiplier script file_multiplier to create less than 500 copies of the feed files or extend your default tablespace. Alternatively, change create table statement for sales_fact table to point to tablespace of your choice.
8. Please skip to section RUNING STREAM HORIZION DEMO

MYSQL DEMO

MySQL deployment comes in two flavours, JDBC and reading rows from a text file (LOAD DATA INFILE).

It is very important to understand how to tune MySQL database engine for high throughput and bulk loading. Please refer to official MySQL documentation.

TO TEST MYSQL JDBC DEPLOYMENT

1. Move $ENGINE_HOME/demo_sales/mysql/engine-config.xml file into $ENGINE_HOME/config/ folder.
2. Open engine-config.xml file and change parameters commented out with string "SET ME!"
3. Execute script $ENGINE_HOME/demo_sales/mysql_create_schema.sql.
4. Make sure that element of engine-config.xml is set to <bulkLoadDefinition outputType="jdbc">
5. Make sure that <bulkLoadInsert> element has adequate <command> tag uncomented. Value starting with 'insert into sales_fact….' Should be uncommented if you run in JDBC mode.
6. Copy sample file $ENGINE_HOME/demo_sales/sales_20140107_data.csv and file multiplier script file_multiplier.sh (or .bat) to your <sourceDirectory> as configured in engine-config.xml. Execute file multiplier script in order to create 500 files.
7. Please skip to section RUNNING STREAM HORIZON DEMO

TO TEST MYSQL IN BULK LOAD MODE

1. Execute first four steps needed for running MySQL JDBC demo (see above)
2. In $ENGINE_HOME/config/engine-config.xml change <bulkLoadInsert> so that it uses first command (with LOAD DATA INFILE syntax)
3. Make sure that element of engine-config.xml is set to <bulkLoadDefinition outputType="file">
4. Make sure that <bulkLoadInsert> element has adequate <command> tag uncomented. Value starting with 'LOAD DATA INFILE….' Should be uncommented if you run in JDBC mode.
5. Copy sample file $ENGINE_HOME/demo_sales/sales_20140107_data.csv and file multiplier script file_multiplier.sh (or .bat) to your <sourceDirectory> as configured in engine-config.xml. Execute file multiplier script in order to create 500 files.
6. Please skip to section RUNNING STREAM HORIZON DEMO

RUNNING STREAMHORIZON DEMO

Before we run StreamHorizon we need to set StreamHorizon threadpools. StreamHorizon has two threadpools, ETL and EB threadpools.

- If you run test via JDBC (both Oracle and MYSQL) your ETL threadpool need be greater than zero while DB Threadpol need be set to zero.
- If you run in BULK MODE (BULK for MYSQL and EXTERNAL TABLES for Oracle) both, ETL and DB threadpools whould be set to same numbers.
- ETL threadpool (number of threads) is set by assigning number to element <etlProcessingThreadCount> of $ENGINE_HOME/config/engine-config.xml
- DB threadpool (number of threads) is set by assigning number to element <databaseProcessingThreadCount> of $ENGINE_HOME/config/engine-config.xml NOTE: for JDBC deployment this number should be set to zero.

WHAT IS DESIRED TREADPOOL SIZE?

Rough estimate is that (assuming that your server is not under heavy load) you should set threadpool size to number of cores of your server. So, for 24 core server for JDBC following settings should be used:

- <etlProcessingThreadCount>24</etlProcessingThreadCount>
- <databaseProcessingThreadCount>0</databaseProcessingThreadCount>

For BULK LOAD deployment following should be used:

- <etlProcessingThreadCount>24</etlProcessingThreadCount>
- <databaseProcessingThreadCount>24</databaseProcessingThreadCount>

Please modify your threadpool sizes and save changes in engine-config.xml file.

STARTING STREAMHORIZON INSTANCE

Please start StreamHorizon instance by executing $ENGINE_HOME/bin/start.sh (or .bat)

Please read "REALISTIC TESTING" section below

CHECKING THROUGHPUT OF YOUR SERVER FOR ORACLE DATABASE

Use *select * from sh_dashboard_jdbc* or *select * from sh_dashboard_ext_tables* (depending on the mode you run) to see throughput of StreamHorizon instance. This view returns single record and gives server level statistics of total number of data records loaded, time window and achieved throughput per second of StreamHorizon instance. You will notice that througput of StreamHorizon increases as server runs.

Note that views sh_dashboard* will only show throughput metrics for data loaded since last start of StreamHorizon engine instance.

To see detail performance & activity logs look at sh_metrics table. This table contains one record for every file processed if you run in JDBC mode and two records (one created by ETL thread and one created by DB thread) if you run in Oracle external table mode.

To troubleshoot please look at $ENGINE_HOME/log/sh-engine*log

CHECKING THROUGHPUT OF YOUR SERVER FOR MYSQL DATABASE

Use *select * from sh_metrics* to see throughput of StreamHorizon instance when using MySQL database.

REALISTIC TESTING

As first run of StreamHorizon will form all cardinalities of the all dimensions from scratch average throughput will be ~3 times slower than actual.

After first successful run, after dimensions are repopulated kill instance of StreamHorizon. Copy files back from <archiveDirectory> to <sourceDirectory>. Start instance of StreamHorizon again and observe the throughput. To avoid repeating this every time you run StreamHorizon write a script which moves files from archive to source directory and change engine-config.xml to invoke it on every startup of StreamHorizon engine instance:

```
<onStartupCommands>

      <command type="shell">/yourMOveScriptComesHere.sh (or .bat)</command>

      <command type="sql">truncate table sales_fact</command>

      <command type="sql">truncate table sh_metrics</command>

</onStartupCommands>
```

NOTE: if you run one set of files, than pause let's say an hour (don't kill StreamHorizon instance) and then run another batch of files view sh_dashboard will show very low throughput. This is because view calculates time window of first file processed by instance after startup and last file processed since instance startup. If batch was loaded in 1 minute, if idle time of server server was 98 minutes, and if second batch was running for 1 minute, throughput shown by sh_dashboard will be 2% of actual throughput of the StreamHorizon.

To avoid this simply observe throughput during the batch immediately after all files are loaded and make sure you restart StreamHorizon before next test.

StreamHorizon increases data throughput as batch progresses.

THROUGHPUT GUIDELINES

In our testing environments we have used commodity Windows, Linux and Solaris servers.

StreamHorizon performs better if deployed on the same server as database server. It is resource lightweight and can efficiently coexist/operate with database instance.

We have used 24 core servers and have achieved througput above 800K per second for Oracle External Tables (running with 24 ETL and 24 DB threads) and throughput above 600K for JDBC Oracle setup running with 26 ETL and zero DB threads.

If you deploy StreamHorizon on the same server as database server if your server has 32 cores number of ETL and DB threads should be around 26. These, of course, are rough guidelines; you should perform multiple tests by adjusting ETL and DB thread count in order to determine optimal settings for your hardware configuration.

REALISTIC DATA

Our tests utilize data typical for Time Series Analysis, Credit Risk, Market Risk and PnL datasets.

File sizes are large (but realistic), data is partially 'sorted' because of the way Tick and Risk data is produced. Such nature of data increases throughput of the StreamHorizon (or any other data processing & ETL platform).

If you are happy with StreamHorizon next step would be to configure it to work with your data model and data feeds.

We are happy to help you in setting up, deploying and delivering your StreamHorizon Instance.