

Table of Contents

Installation.....	1
Requirements.....	1
Source code.....	1
Web site configuration.....	1
Install JGraph.....	1
Access rights.....	1
Monitoring schema on the target database.....	1
Framework configuration.....	2
Database access.....	3
Set the environment variables.....	3

Installation

Requirements

The framework is coupled to several tools available at CERN. You will need to check if everything is set up correctly.

- Every client machine should have an AFS access. It allows us to distribute the code to the clients and gather the results in a single folder. NFS can be used as a substitute but it will cause problems for the web based report.
- One of the client machines should be able to access all the other nodes via ssh and without password. This one will be used to start the manager script.

Source code

You can get the source code from the Persistency CVS :

```
cd ~/
cvs -d :pserver:anonymous@coral.cvs.cern.ch:/cvs/coral co -d PerfTests -P coral/Tests/Performance
```

Web site configuration

A web site should be configured in order to publish the web based report. An AFS based is easily created using this page : <https://www.cern.ch/WebServices/Services/CreateNewSite/> . Choose *AFS folder* for the *Site type* and put the log folder path (ie: `/afs/cern.ch/user/j/jsmith/PerfTests/logs/`).

Install JPGraph

Get it from there : <http://www.aditus.nu/jpgraph/jpdownload.php> (use "PHP4: JpGraph 1.x - series") and put it in the log directory (as jpgraph) :

```
cd ~/PerfTests/logs/
wget http://hem.bredband.net/jpgraph/jpgraph-1.22.tar.gz -O /tmp/jpgraph.tar.gz
tar xzf /tmp/jpgraph.tar.gz
mv jpgraph* jpgraph
```

Access rights

Change the access privileges to the log directory to enable access from web :

```
fs setacl -dir ~/PerfTests/logs/ ~/PerfTests/logs/jpgraph/src/ ~/PerfTests/logs/jpgraph/src/lang/
```

Monitoring schema on the target database

You should create a specific schema on the monitored database to use the monitoring feature. A script called [install.sql](#) is provided in order to do so. You should execute it with a DBA account. You will be asked for a username and a password which will be needed by the framework afterward (in authentication.xml).

Framework configuration

Modify testConf.xml[?] to configure the test environment.

- Indicate the client machine and RAC node hostnames

```
<?xml version="1.0" ?>
<config>
  <machines>
    <client>node1</client>
    <client>node2</client>
    <client>node3</client>

    <server>server1</server>
    <server>server2</server>
  </machines>
```

- you can enable or disable the different monitoring graphs (0 to disable)

```
<statistic cpu="1" waitEvents="1"/>
```

- you can set the "ramping-up" of the connections

```
<execution
  numCliNode="1"    -- number of machines used for this test run
  numSessPerCliNode="2"  -- total number of client programs which will be started on each client
  numSessIni="1"    -- number of client programs which will be started on each client machine at s
  numSessInc="1"    -- number of client programs which will be started on each client machine after
  stepDuration="20"  -- waiting time between each "ramping-up step"
  maxTestDuration="3600"  -- maximum duration of the test, after this period the clients are killed
/>
```

- then you should indicate the folders used by the program

```
<locations
  baseDir="/afs/cern.ch/user/j/jsmith/PerfTests/"  -- directory where you downloaded the framework
  clientLogDir="/afs/cern.ch/user/j/jsmith/PerfTests/logs/">  -- directory where you plan to store logs
/>
```

- you should specify the metric names you'll be using

```
<metrics>
  <dynamic>Object inserted</dynamic>  -- metrics which will be divided by the time taken to achieve
  <static>Memory usage in MB</static>  -- metrics which do not need calculation, meaningful as is
</metrics>
```

- here you should indicate the information related to the client program

```
<clientConf
  dbUser="ORACLE_USER"  -- name of the Oracle user which will be used as client for the test
  clientCmd="client.py"  -- command used to start the test program on the different client machines
  clientExe="python"    -- test program real name as seen by the operating system (eg: python)
  reportPeriod="10"    -- waiting time between each monitoring output (also used in the client program)
  reportDbConnect="/dbdev/OracleStress"  -- connection string used by the monitoring script
/>
```

- and finally, you can save meaningful files in the log folder

```
<fileToSave>
  <file name="Client execution script">client.py</file>
  -- the client program should be saved in order to reproduce the test easily
  <file name="Configuration file">testConf.xml</file>
```

```
-- a saved version of the configuration file is also needed
to generate the report
</fileToSave>
```

Database access

You'll have to modify the configuration files authentication.xml and dblookup.xml. They are needed by the monitoring extension and perhaps for the test client.

- authentication.xml [↗](#), gathering the Oracle usernames & passwords :

First the account for the monitoring

```
<?xml version="1.0" ?>
<connectionslist>
  <connection name="oracle://oracle_service_name/oracle_monitoring_username">
    <parameter name="user" value="oracle_monitoring_username" />
    <parameter name="password" value="oracle_monitoring_password" />
  </connection>
```

Then the account for the test client for each node (if using COOL or CORAL)

```
<connection name="oracle://oracle_service_name_node1/oracle_client_username">
  <parameter name="user" value="oracle_client_username" />
  <parameter name="password" value="oracle_client_password" />
</connection>

<connection name="oracle://oracle_service_name_node2/oracle_client_username">
  <parameter name="user" value="oracle_client_username" />
  <parameter name="password" value="oracle_client_password" />
</connection>
</connectionslist>
```

- dblookup.xml [↗](#), define a service name for the monitoring :

Persistency uses also this file to declare the monitoring service name

```
<?xml version="1.0" ?>
<servicelist>
  <logicalservice name="/OracleStressReport">
    <service name="oracle://oracle_serice_name/oracle_monitoring_username" accessMode="readonly" />
  </logicalservice>
</servicelist>
```

Set the environment variables

There are two files provided depending on your shell. For bash, sh or zsh use [bashrc](#) [↗](#) and for csh or tcsh use [cshrc](#) [↗](#).

You can copy them in your `.bashrc` (or `.cshrc`) or source them from there :

```
source ~/PerfTests/bin/bashrc
```

Don't forget to modify the following variable if needed :

```
export CMTCONFIG=slc4_amd64_gcc34 -- eg can be changed to slc4_ia32_gcc34
```

This topic: PSSGroup > OracleRACLoadTestingInstall

Topic revision: r4 - 2010-06-11 - PeterJones



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)