Measuring SQL response time

# Introduction

Customers often report that saves and loads from a CAD integration client are slower than expected. The first thing to check is that your DBMS is performing within acceptable parameters. This is a quick and easy way to either rule out the DBMS response, or identify it as a performance bottleneck.

It is much easier to perform this test on a 2 tier client than 4 tier. 4 tier is possible, but it requires either system downtime or an advanced configuration procedure, which is documented here

# Procedure

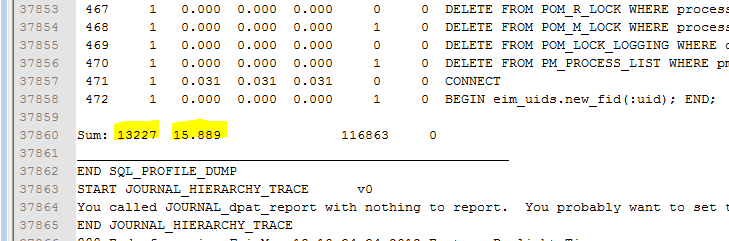
## 2 tier clients

Add the following variables to your tcserver startup environment. There are many ways to do this, but the easiest is to edit the %TC\_ROOT%\iiopservers\start\_TcServer1.bat file, and add these lines to the top:

set TC\_KEEP\_SYSTEM\_LOG=Y

set TC\_SQL\_DEBUG=BJPT

Stop any RAC session that is running, and kill any existing TAO windows. Run the test case and shut the CAD integration session down completely, including killing the associated TAO window. Find the tcserver syslog that corresponds to your session – usually this will be the %TEMP% directory. Scroll directly to the bottom of the tcserver log file to find the summary table. At the bottom of the summary table you will see (a) the total number of SQL statements executed during that session, and (b) the total elapsed time consumed by those SQL statements. Divide the time by the SQL count to get the average response time, in seconds. For example, this syslog shows that 13,227 statements were execute d in 15.889 seconds:



Dividing 15.889 by 13,227 gives 0.0012 seconds, or 1.2 milliseconds average response time. Anything below 2.0 milliseconds is generally considered to be acceptable performance, although obviously the faster the better. If your result is greater than 2 ms, you should ask your DBA to run statistics, monitor statspack, or whatever else should be done to improve the DBMS performance.

## 4 tier clients

If you’re working in a test environment it may be feasible to apply the 2 tier instructions to your tcserver pool startup file. But in a production system this means bringing the server pool down twice: Once to add the debug settings, and a second time to back them out. A better option is to configure a private tcserver for the CAD client who is running the performance test. This is done as follows:

### On the 4 tier server:

Start a Teamcenter command line, and run the following commands:

cd /d %IMAN\_TMP\_DIR%

set TC\_KEEP\_SYSTEM\_LOG=Y

set TC\_SQL\_DEBUG=BJPT

tcserver id=ipemserver –OrbEndpoint iiop://cii3w265:9999 > tcserver.console 2>&1 (substitute your server hostname for cii3w265)

### On the integration client

(This example uses the Pro/E integration, modify as needed for your client):

Modify the client\_specific properties file, adding the following lines:

iiopServerCount=1

IIOP\_SERVER\_1.MARKER=ipemserver

IIOP\_SERVER\_1.NAME=ipemserver

IIOP\_SERVER\_1.HOST=cii3w265

IIOP\_SERVER\_1.PORT=9999

Modify the integration’s ipemrunnersoa.bat file, commenting out the highlighted lines:

REM set TXD\_TIER=4

set TXD\_SERVER\_DIR=C:\apps\teamcenter\tc83\_4t\iiopservers

set TC\_TMP\_DIR=%IPR\_TEMP\_DIR%

set TPR=%IPR\_DIR%

set JACORB\_HOME=%IPEM\_DIR%

REM if not exist "%TXD\_SERVER\_DIR%\tcserver.xml" goto no\_2\_tier

set TXD\_TIER=2

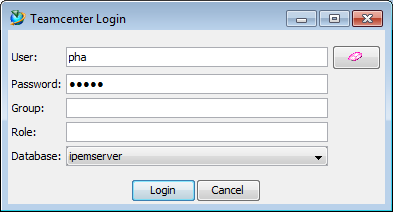
REM pushd %IPR\_TEMP\_DIR%

REM start /WAIT "TAO ImR" /min cmd /c "%TXD\_SERVER\_DIR%\start\_imr.bat"

REM popd

REM :no\_2\_tier

Start the integration client and log into Teamcenter. You will see the value “ipemserver” appear in the list of databases – select it:



Run the test case, then shut down the 4 tier integration client. Locate the tcserver syslog on the server (the file name is given in the integration’s txd log file – search for the string “syslog”), then perform the same calculation as shown above, under the 2 tier instructions.