Oracle 12c Database Performance Application Tuning example

By David Itshak

shaked19@gmail.com

http://www.ildba.co.il/author/cimid/
http://www.sqlserver.co.il/?cat=940

Global Hebrew Virtual PASS Chapter :
https://www.youtube.com/watch?v=x4hGjYGBfkc
https://www.youtube.com/watch?v=eJO8G9if3EY

Sqlsatrday Israel 2016 :
When ODAC, MATLAB, Oracle Sever 11GR2, and Win HPC Server 2008 Come together...

David Itshak

sahked19@gmail.com
AGENDA

• Basic Term : Scale in/out, **Hyperthreading**, Network BW
• A background on HPC architecture
• Integration with compute nodes with ODAC CLIENT and ORACLE DB
• ODT & ODAC-Deployment on Compute Nodes
• Guidelines to ODP.NET Performance & improving latency.
• MATLAB & HPC & ORACLE PL\SQL
• BONUS
  – Performance Tuning with ODT
  – USING CIFS PROTOCOL FOR RMAN BACKUPS ON WINDOWS
ענין ייעוץ מachsob

- גדל ביכולים המachsob באמצעת ביזיו משאבי המachsob.
- מערכות כאלו מורכבות מאשכולות של מחשבים או מעבדים
  שמוקשרים ביניהם באמצעות תקשורת מהירה.
 总裁ה הישראלית 예מה ביצועים מחשבים אמצעי משאבי המשימה
  ביצוע מחשבים ואחסון ביצועים הינו משימה בעדיפות לאומית
  גלובס, 2.1.2011 (גלוס, 2.1.2011)

- בן התפקידים של מליאונוקס היישארית למשל מאפרה
  אינטגרציה בין הרכיבים שעימים לכל חכמה חכמה לכל
  עמוסים ברשת.

- מחשבים ייעוץ ביצועים מחשבים בעיתונן ייעוץ ביצועים וHint
  ביצועים ליאומי (גלוס, 2011)
Scale In

- More CPUs & Cores
- More memory (DRAM, Flash Memory)
- Faster & Larger Disks
- Faster network (interconnect)
- Limited number Of nodes (RAC, MSSQL failover Cluster)
- More Money!
- Limited - physical limitations !!!
Scale In Example-Oracle *Exadata*

- A prebuilt 8-Node RAC cluster with Super SAN
- All the CPU power you need (Min 64 cores)
- Mega DRAM Server Memory (Min 576G)
- Super-Mega Flash Memory (Min 5.3T)
- Super fast interconnect (Min 40Gb/s)
- 100T of SAS disk (Min 28T useable)
Scale In Example-Oracle **Exadata**

RAC - Shared Data Model

*Exadata puts it back into One Machine*

---

**Instance 1**

- Shared Memory/Global Area
- Shared SQL
- Log Buffer

**Instance 2**

- Shared Memory/Global Area
- Shared SQL
- Log Buffer

...  

**Instance N-1**

- Shared Memory/Global Area
- Shared SQL
- Log Buffer

**Instance N**

- Shared Memory/Global Area
- Shared SQL
- Log Buffer

---

**Shared Disk Database**
Scale Out- HPC

• Grid Computing.

• Advantages of HPC 2008 R2:
  – Compute Nodes can uses also for other computations, thus fully utilized HPC Cluster.
  – Using available win 2008 R2 infrastructure
  – Using WIN 7 stations as compute nodes
  – Replacing other HPC expensive software.
  – Scalable computing power cannot be reach with APP server.

• Theoretically no physical limitation.

• But Network can be a limit
Hyperthreading

- Circuitry added to Intel CPUs resulting in single CPU functioning as 2 CPUs.
- All versions of Oracle/SQL Server are supported in Hyperthreaded environments.
Network Bandwidth

- Bandwidth is not speed, it is capacity- (Mbits/Sec)
- Number of bits that can be sent the same time
- Highest BW is not always the fastest route because of Latency
- Check Latency/Round Trip Time (RTT)
- Latency depends on the size of the package
- Ping with various packages size

Usage: ping  [-n count] [-l size]

Options:
  -n count Number of echo requests to send.
  -l size Send buffer size.
Network Bandwidth

ping dvd8csqldb -l 32 -n 4

Pinging server1 [2.2.11.12] with 32 bytes of data:

Reply from 2.2.11.12: bytes=32 time=68ms TTL=127

Ping statistics for 2.2.11.12:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 1ms, Maximum = 68ms, Average = 18ms

ping dvd8csqldb -l 6400 -n 4

Pinging server1 [2.2.11.12] with 6400 bytes of data:

Reply from 2.2.11.12: bytes=6400 time=106ms TTL=127

Ping statistics for 2.2.11.12:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 1ms, Maximum = 106ms, Average = 27ms
HPC Background

- High Performance Computing gives engineers the computation resources they need to speed research and development.

- Examples:
  - Test simulations
  - Modeling solutions
  - Highly complex problems

- Computes nodes
  - Dozens nodes on HPC, Win 2008R2.
  - In future: Several thousands on Win 7 works stations
  - Run .Net APP process VS Oracle DB.
  - .Net APP use Oracle Data Access Components (ODAC) 11.2.0.1
HPC Background-1

• Oracle DB
  – 5 Tera of Data.
  – Enterprise Edition 11.2.0.2, 64 bit, Win 2008 R2

• No of Processes running VS Oracle DB
  – Job can assigned per CPU Core.
  – 12 cores per compute Nodes.
  – Assume we have 100 Compute Nodes
  – In Full utilization
    • Job per core
    • 100 X 12 =1200 processes of .NET APP using ODAC vs ORACLE DB.
What Runs ON HPC

- MATLAB
- Mechanical CAD:
  - ABAQUS
  - CST (Computer Simulation Technology)
- Monte Carlo Simulation
Windows HPC Server 2008 R2

- Complete, integrated platform for HPC Clustering
- Built on top Windows Server 2008 R2 64-bit Operating System
- Addresses the needs of traditional and emerging HPC

Windows Server 2008 R2 HPC Edition
- Secure, Reliable, Tested
- Support for high performance hardware (x64, high-speed interconnects)

Microsoft HPC Pack 2008 R2
- Job Scheduler
- Resource Manager
- Cluster Management
- Message Passing Interface

Microsoft Windows HPC Server 2008 R2
- Integrated Solution out-of-the-box
- Leverages investment in Windows administration and tools
- Makes cluster operation easy and secure as a single system

Latest CTP available from http://www.microsoft.com/hpc
Basic Architecture of an HPC Cluster

- **Head Node**
  - The single point of management and job scheduling for the cluster.
  - Provides failover and controls and mediates access to the cluster resources.

- **Compute Node**
  - Carries out the computational tasks assigned to it by the job scheduler
Basic Architecture of an HPC Cluster-2

• **Job Scheduler**
  - Queues jobs and their associated tasks.
  - Allocates resources to these jobs, initiates the tasks on the compute nodes;
  - Monitors the status of jobs, tasks, and compute nodes.

• **Broker Node**
  - Act as intermediaries between the application and the services. The broker load-balances the service requests to the services, and finally return results to the application.
Basic Cluster

• The minimal set of components to run an HPC application on a cluster.
What is a cluster?

• A cluster is the top-level unit of Windows HPC Server. Contains:

• **Node**
  
  • Single physical or logical computer
  
  • May have one or more processors.
  
  • Types of nodes: head node, compute nodes, or WCF Broker nodes.

• **Queue**
  
  • Contains pending jobs
  
  • Provides job scheduling.
  
  • One queue per cluster
  
  • Completed jobs purged periodically
What is a cluster? (continued)

• **Job**
  
  • Collection of tasks (1 or many)
  • Initiated by user as interactive or batch processes.
  • Jobs reserve resources for subsequent use by one or more tasks.

• **Tasks**
  
  • execution of a program on given compute nodes.
  • Types of tasks:
    – serial program (single process), or
    – a parallel program (using multi-threading, OpenMP, or MPI).
The Heat Map view gives instant feedback on the health of the cluster—upwards of 1,000 nodes, without scrolling.
What is the life of a job?

- Admission
- Scheduling
- Execution
Interfaces - Job scheduling
Windows HPC Server 2008 With High Availability

1. User submits job.

2. Assigns nodes for client job

3. Compute Nodes run vs ORACLE Server

4. Job finished, response return to client

Dozens Compute nodes:
(ODAC) 11.2.0.1.2 32 bit, Win 2008 R2

Oracle (~4T Data)
11.2.0.81, 64 bit
Win 2008 R2

SAN
NAS

Oracle Data file on HDS storage

RMAN Writes to Fast Recovery area on shared Drives using CIFS Protocol to SATA disks.
ODAC -1 & ODT

• We use
  – Oracle Data Access Components (ODAC) 11.2.0.1
  – Oracle Developer Tools (ODT) for Visual Studio 11.2.0.1
  – 11g Release 2 ODAC 11.2.0.1.2 with Xcopy Deployment
• ODT is tightly integrated “Add-in” for Visual Studio .NET 2010/2008/2005
• Available for free download today
  – http://otn.oracle.com/dotnet
ODT & ODAC- 2

• ODP.NET
  – ADO.NET compliant data provider
  – Native access to Oracle database
  – Utilize advanced Oracle Database features
    • RAC, performance, security, data types, XML, etc.
ODAC - 3 & ODT

- Current version: ODAC 11.2.0.3.0 with Oracle Developer Tools for Visual Studio
- Released December 28, 2011
- Includes support for Entity Framework and LINQ
Oracle .NET Data Provider

• In addition to basic Oracle client connectivity software
• .NET apps require the use of managed data provider (By .NET framework).
• The layer between the .NET application code and Oracle client connectivity software.
• Best performance
1 Oracle NET Data Provider

• ODP.NET provides standard ADO.NET data access, while exposing Oracle database-specific features:
  – XML DB
  – data access performance optimizations
  – Real Application Clusters load balancing & fast connection failover.
2 Oracle .NET Data Provider

- ODP.NET version, 11.2, supports connecting to Oracle Database 9i and higher versions.
- Db server can be on Windows, Linux, UNIX, or any other operating system platform Oracle database supports.
Oracle NET Data Provider 3

• Recommended Book

Pro ODP .NET for Oracle Database 11g

Edmund Zehoo
ODT & ODAC-Deployment on Compute Node
Connecting via EZConnect
• Host name, port number, and service name all in one.
• Specific to TCP/IP connections.
• No need to manage tsnames.ora on compute nodes.

```
OracleConnection conn = new OracleConnection();
conn.ConnectionString = "DataSource=dbhost:1521/GISDB; User ID=SYSTEM;Password=PASS!@";  
try
{
    // Setting a cache size of 0 automatically disables the statement cache.
    String _connString = "Data Source=TESTDBHOST/GISDB;User Id=HR;Password=PASS124;Statement Cache Size=0;Self Tuning=false;";
    OracleConnection _connObj = new OracleConnection(_connString);
    _connObj.Open();
......
```

ODT & ODAC-Deployment on Compute Node

- Automatic/silent installation in the following order:
  - Silent installation 11g Release 2 (11.2.0.2) Patch Set 1 32 bit Client
  - 2-bit Oracle Data Access Components (ODAC) 11.2.0.1.2 for Windows or the higher on the same ORACLE Home.

- Check assembly GAC for correct version Of Oracle Assemblies
ODP.NET Performance- Why?

• **N** Compute nodes with **ODAC 11.2.0.1.2** 32 bit, **M** cores per node, **NXM Process**
  - Performance gain is always **NXM**
  - Example: 100 nodes, 12 cores, X1200 Performance gain

• Highest BW is not always the fastest route because of Latency.

• Reduce number of round-trips to server by executing multiple statements & returning multiple datasets in a single round-trip to Oracle.
  - Performance gain becomes visibly higher.
When Network, ODAC, HPC Compute Node, Oracle Sever come together ...

• The scenario
  – User complains about bad response time from client (Compute Node)
  – .NET App with 11.2.0.2 32 bit Client & 32-bit Oracle Data Access Components (ODAC) 11.2.0.1.2 for Windows

• First Steps, tune ORACLE DB :
  – Run Active session history (ASH) reports
  – Run Automatic Database Diagnostic Monitor (ADDM) reports
  – Run SQL Access Adviser from Database control
    • No special recommendations
    • DB was already tuned when was upgrade to 11GR2.
    • SQL were very fast.
Trace the sessions

• Find the sessions and enable SQL trace for a session

SELECT sid, serial#, username, machine FROM v$session;
dbms_monitor.session_trace_enable(session_id => 127, serial_num => 29, 
  waits => TRUE, binds => FALSE);

• Run some activity.

dbms_monitor.session_trace_disable(session_id => 127, serial_num => 29);

• Disable trace in session:

Tkprof input_file output_file sys=no

• Format the trace files without system.
Analyze the trace Files

- DB waits for SQL*Net message to client while instructing the network layer to send data to the client network.
- DB waits for SQL*Net message from client while waiting for data from the client.
- For each round-trip by SQL*Net layer, you should see pair of those wait events.

<table>
<thead>
<tr>
<th>Elapsed times include waiting on following events:</th>
<th>Times</th>
<th>Max. Wait</th>
<th>Total Waited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event waited on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL*Net message to client</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SQL*Net message from client</td>
<td>1000</td>
<td>0.08</td>
<td>5.94</td>
</tr>
</tbody>
</table>

- SQL*Net message from client is high!
- Tune ODP.NET Performance & Network
Tuning the network - The easy way

• Install the free utility WIRESHARK
  – A network packet analyzer.
  – Captures network packets and tries to display that packet data as detailed as possible.
Tuning the network - The easy way-1

- Set WIRESHARK filter on listener port 1521 of Oracle server and clients IPs.
Tuning the network - The easy way-2

- Run WIRESHARK to monitor network, packages size & quantity
Tuning the network - The easy way-3

- Summary reports of WIRSHARK
- Display a list of conversations (traffic between two endpoints),
  Packages between clients and DB server
ODP.NET Performance

• Measuring Performance
  – Enabling the Performances Counters
  – Measuring Performance Programatically

• Speeding Up Connections with Connection Pooling

• Performing Faster Floating Point Arithmetic

• Executing Statements Faster
  – Batching Your SQL Statements Together For Execution
  – Using Statement Caching
  – REF Cursors and Multiple Active Resultsets (MARs)

• Passing Parameters More Efficiently
  – Using Bind Arrays to Pass Parameters in Bulk
  – Using PL/SQL Associative Arrays
ODP.NET Performance 2

• Managing LOBs More Efficiently
  – Enabling the LOB Cache
  – Setting the InitialLOBFetchSize Property

• Retrieving Data More Efficiently
  – Changing the FetchSize Property
  – Using the Client Result Cache

• Importing Data More Efficiently

• Applying Optimization Best Practices
  – Using Stored Procedures Whenever Possible
  – Using the Right Data Access Object
ODP.NET Performance Demonstrations

- Speeding Up Connections with Connection Pooling
- Performing Faster Floating Point Arithmetic
- Retrieving Data More Efficiently
  - Using the Client Result Cache
- Executing Statements Faster
  - REF Cursors and Multiple Active Resultsets (MARs)
ODP.NET Performance - sample App

Please run the SQL statements in the SQLScript.txt file first (using SQL*Plus), before running any of the samples below:

- Using Connection pooling
- Using BINARY FLOAT/BINARY DOUBLE
- Batching SQL statements
- Using Statement caching
- Using Bind Arrays
- Using PL/SQL Associative Arrays
- Using LOB Caching
- Changing Fetchsize property
- Using the client result cache
- Using OracleBulkCopy
Measuring Performance Programmatically

StopWatch class

- Use a hardware-based high-resolution timer or system timer otherwise.
- `kernel32.dll's QueryPerformanceCounter Windows API call to retrieve the time`

```csharp
Stopwatch _stopwatch = new Stopwatch();
String _Results;
String _connstring = "Data Source=localhost/NEWDB;User Id=TEST;Password=PASS;Pooling=false"
try
{
    //Open and close connections 10 times without connection pooling enabled
    OracleConnection _connObj = new OracleConnection(_connstring);
    _stopwatch.Start();
    for (int i = 1; i <= 10; i++)
    {
        _connObj.Open();
        _connObj.Close();
    }
    _stopwatch.Stop();
```
Speeding Up Connections with Connection Pooling

```csharp
//Open and close connections 10 times without connection pooling

OracleConnection _connObj = new OracleConnection(_connstring);
stopwatch.Start();
for (int i = 1; i <= 10; i++)
{
    _connObj.Open();
    _connObj.Close();
}
stopwatch.Stop();
_Results = "Without connection pooling:\t" +
    stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";

//Open and close connections 10 times with connection pooling

_connstring = "Data Source=testdb/GISDB;User Id=HR;Password=PASS123;Pooling=true";
_connObj = new OracleConnection(_connstring);
stopwatch.Reset();
stopwatch.Start();
for (int i = 1; i <= 10; i++)
{
    _connObj.Open();
    _connObj.Close();
}
stopwatch.Stop();
_Results = _Results + "With connection pooling:\t" +
    stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";
```
BINARY_FLOAT and BINARY_DOUBLE performance of NUMBER data type

```java
OracleConnection _connObj = new OracleConnection(_connstring);
_connObj.Open();
OracleCommand _cmdObj = _connObj.CreateCommand();

//Adding NUMBERS
_cmdObj.CommandText = "DECLARE" +
" Number1 NUMBER:=1;" +
" Number2 NUMBER:=1;" +
"BEGIN" +
" FOR i IN 1 .. 100000 LOOP" +
" Number1:=Number1 + Number2;" +
" END LOOP;" +
"END;";
_stopwatch.Start();
_cmdObj.ExecuteNonQuery();
_stopwatch.Stop();
_Results = _Results + _stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";

//Adding BINARY_FLOAT numbers
_cmdObj.CommandText = "DECLARE" +
" BinaryFloat1 BINARY_FLOAT:=1;" +
" BinaryFloat2 BINARY_FLOAT:=1;" +
"BEGIN" +
" FOR i IN 1 .. 100000 LOOP" +
" BinaryFloat1:=BinaryFloat1 + BinaryFloat2;" +
" END LOOP;" +
"END;";
_stopwatch.Reset();
_stopwatch.Start();
_cmdObj.ExecuteNonQuery();
_stopwatch.Stop();
_Results = _Results + "Adding BINARY FLOATs:\t" +
```

BINARY_FLOAT and BINARY_DOUBLE

Use machine arithmetic computation works is passed to the operating system, extremely efficient.
BINARY_FLOAT and BINARY_DOUBLE performance of NUMBER data type-2

```csharp
// Adding BINARY_DOUBLE numbers
_cmdObj.CommandText = "DECLARE" +
" BinaryDouble1 BINARY_DOUBLE:=1;" +
" BinaryDouble2 BINARY_DOUBLE:=1;" +
"BEGIN" +
" FOR i IN 1 .. 1000000 LOOP" +
" BinaryDouble1:=BinaryDouble1 + " +
" BinaryDouble2;" +
" END LOOP;" +
"END;";
_stopwatch.Reset();
_stopwatch.Start();
_cmdObj.ExecuteNonQuery();
_stopwatch.Stop();
_Results = _Results + "Adding BINARY_DOUBLEs:\t" +
_stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";
MessageBox.Show(_Results);
_connObj.Close();
```
Using the Client Result Cache

```sql
SQL> ALTER SYSTEM SET client_result_cache_size=64000 scope=spfile;
System altered.
SQL> shutdown immediate;
...
SQL> SHOW PARAMETER result_cache_mode;
NAME                TYPE       VALUE
----------------------------------------------- ------------------
result_cache_mode    string     MANUAL

SQL> SHOW PARAMETER client_result_cache_size;
NAME                TYPE        VALUE
----------------------------------------------- ------------------
client_result_cache_size big integer 64000

Specify /*+ result_cache */ hint in SQL

SELECT /*+ result_cache */ ID,Name FROM Products
```
Without Client Result Cache

```csharp
//Retrieve 1,000 rows without using the client result cache
_cmdObj.CommandText = "SELECT * FROM Products";
_stopwatch.Start();

for (int i = 1; i <= 1000; i++)
{
  OracleDataReader _rdrObj = _cmdObj.ExecuteReader();
  while (_rdrObj.Read()) { }
}

_rdrObj.Close();
```
Without Client Result Cache

Stopwatch _stopwatch = new Stopwatch();
String _Results;
try
{
    //Retrieve 10,000 products with statement caching disabled
    //Setting a cache size of 0 automatically disables the statement cache
    String _connstring = "Data Source=TESTDBHOST/GISDB;User Id=HR;Password=PASS124;Statement Cache Size=0;Self Tuning=false;"
    OracleConnection _connObj = new OracleConnection(_connstring);
    _connObj.Open();
    OracleCommand _cmdObj = _connObj.CreateCommand();
    _stopwatch.Start();
    _cmdObj.CommandText = "SELECT * FROM Products WHERE ID=:IDValue";
    OracleParameter _paramObj =
        _cmdObj.Parameters.Add("IDValue", OracleDbType.VarChar2);
    for (int i = 1; i <= 10000; i++)
    {
        _paramObj.Value = "E" + Convert.ToString(i);
        OracleDataReader _rdrObj = _cmdObj.ExecuteReader();
        _rdrObj.Dispose();
    }
    _stopwatch.Stop();
    _Results = "Without Statement Caching:\n" +
        _stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";
    _cmdObj.Dispose();
    _connObj.Close();
Using the Client Result Cache

//Retrieve 1,000 rows with client result cache
_cmdObj.CommandText = "SELECT /*+ result_cache */ * FROM Products";
_stopwatch.Reset();
_stopwatch.Start();
for (int i = 1; i <= 1000; i++)
{
    OracleDataReader _rdrObj = _cmdObj.ExecuteReader();
    while (_rdrObj.Read()) { }
    _rdrObj.Close();
}
_stopwatch.Stop();
Using the Client Result Cache

//Retrieve 10,000 products with statement caching enabled
_connstring = "Data Source=TESTDBHOST/GISDB;User Id=HR;Password=PASS124;Statement Cache Size=5;Self Tuning=false;";
_connObj.ConnectionString = _connstring;
_connObj.Open();
_cmdObj = _connObj.CreateCommand();
_stopwatch.Reset();
_stopwatch.Start();
_cmdObj.CommandText = "SELECT * FROM Products WHERE ID=:IDValue";
_paramObj = _cmdObj.Parameters.Add("IDValue", OracleDbType.VarChar2);
for (int i = 1; i <= 10000; i++)
{
    _paramObj.Value = "E" + Convert.ToString(i);
    OracleDataReader _rdrObj = _cmdObj.ExecuteReader();
    _rdrObj.Dispose();
}
_stopwatch.Stop();
_Results = _Results + "With Statement Caching:\t" +
_stopwatch.Elapsed.TotalSeconds.ToString() + " seconds\n";
_cmdObj.Dispose();
_connObj.Close();
MessageBox.Show(_Results);
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
REF Cursors and Multiple Active Resultsets (MARs)
• Allow you to reduce the number of round-trips to the server
  – executing multiple statements and returning multiple datasets in a single round-trip to Oracle.
• When accessing the database over the network, the performance gain becomes visibly higher.
• Return multiple REF cursors from your SPs and OracleDataAdapter.Fill() method to obtain a DataSet
  – various result sets are populated into separate DataTable objects in the same dataset.
REF Cursors and MARs - 2

• Retrieve multiple active result sets single stored procedure using ref cursor.

```sql
CREATE OR REPLACE PACKAGE ProductsPackage IS
    TYPE refCursor IS REF Cursor;
    PROCEDURE proc_GetProductsInfo(cheapProducts OUT refCursor, expensiveProducts OUT refCursor);
END ProductsPackage;

CREATE OR REPLACE PACKAGE BODY ProductsPackage IS
    PROCEDURE proc_GetProductsInfo (cheapProducts OUT refCursor, expensiveProducts OUT refCursor)
    IS
    BEGIN
        OPEN cheapProducts FOR
        SELECT * FROM Products WHERE Price<500;
        OPEN expensiveProducts FOR
        SELECT * FROM Products WHERE Price>500;
    END;
END ProductsPackage;
```
try
{
    OracleConnection _connObj = new OracleConnection(_connstring);
    _connObj.Open();
    OracleCommand _cmdObj = _connObj.CreateCommand();
    _cmdObj.CommandText = "ProductsPackage.proc_GetProductsInfo";
    _cmdObj.CommandType = CommandType.StoredProcedure;
    //Create the REF cursor parameter for the products that are < $500
    OracleParameter _chpProdParam = new OracleParameter();
    _chpProdParam.ParameterName = "cheapProducts";
    _chpProdParam.OracleDbType = OracleDbType.RefCursor;
    _chpProdParam.Direction = ParameterDirection.Output;
    _cmdObj.Parameters.Add(_chpProdParam);
    //Create the REF cursor parameter for the products that are > $500
    OracleParameter _expProdParam = new OracleParameter();
    _expProdParam.ParameterName = "expensiveProducts";
    _expProdParam.OracleDbType = OracleDbType.RefCursor;
    _expProdParam.Direction = ParameterDirection.Output;
    _cmdObj.Parameters.Add(_expProdParam);
    OracleDataAdapter _adapterObj = new OracleDataAdapter(_cmdObj);
    DataSet _datasetObj = new DataSet();
    _adapterObj.Fill(_datasetObj);
    //The result sets are stored in separate DataTables in the same dataset
    dataGridView1.DataSource = _datasetObj.Tables[0];
    dataGridView2.DataSource = _datasetObj.Tables[1];
    _connObj.Close();
    _connObj.Dispose();
    _connObj = null;
}
Performance Tuning with ODT

• SQL Tuning Advisor
  – Tune ad-hoc queries in Query Window
  – Tune inefficient application SQL detected by Oracle Performance Analyzer
  – Explain Plan feature in Oracle Query Window

• Oracle Performance Analyzer
  – Tune your running application’s use of the database
ODT & ODAC 11.2.0.1 – Examples from our DEV environment

• Implementing new features ODT & ODAC.
  • Manage Users, Roles and Privileges with Server Explorer.

• Tune Ad-hoc SQL statements in Query Window with SQL Tuning Advisor.
Performance Tuning in Visual Studio – SQL Tuning Advisor

• Requirements
  – ADVISOR Privilege
  – Oracle Database license for Oracle Diagnostic Pack
  – Oracle Database license for the Oracle Tuning Pack

• How to run:
  – Oracle Query Window “Tune SQL” button
  – Oracle Performance Monitor – Tune SQL button

• Implement Findings Button
  – Automatically fix the problem for certain finding types

• View Report Button
  – View more details about how to fix a problem
SQL tuning Adviser in Visual Studio

- Application user can Manage Users, Roles and Privileges with Server Explorer.
- OEM_ADVISOR privilege to use SQL Tuning Adviser
  - SQL: GRANT OEM_ADVISOR to user
  - GUI: Connect as sysdba to dev DB and give to user ADVISOR privilege