Oracle and .NET: Best Practices for Performance and Deployment

Alex Keh
Senior Principal Product Manager
Server Technologies
Program Agenda

1. Optimization Process
2. Optimizing ODP.NET Performance (Connections, Data Retrieval and Updates, ODP.NET Data Types)
3. Caching
4. ODP.NET Deployment
Optimization Process
Oracle .NET Application Performance Optimization Steps

• .NET data access tuning
  – Use ODP.NET best practices

• SQL tuning
  – Use SQL Tuning Advisor in Visual Studio

• Database tuning under real world conditions
  – Oracle Performance Analyzer in Visual Studio detects issues you have missed
  – May need to modify application based on findings
  – Can be used during testing phase or production
Optimizing ODP.NET Performance
General – All ODP.NET Objects

• Close/Dispose all ODP.NET objects explicitly
  – Garbage collector cannot reliably implicitly dispose objects under heavy load
    • May see increasing memory usage

• Can use “Using” statement instead

• Recommended for all ODP.NET objects
Connections

• Use connection pooling
  – Min Pool Size = # connections at steady state or average load
  – Max Pool Size = # connections at maximum capacity
    • Min and Max Pool Size always obeyed over other CP attributes

• ODP.NET performance counters
  – Monitor with Windows Performance Monitor
  – More granularity in monitoring – New in ODAC 12c
    • Monitor at app domain, pool, or DB instance level
    • DB instance level monitoring applies if load balancing or FCF enabled
Connection Management
RAC, Data Guard, and GoldenGate

• Run-time connection load balancing
  – Automated load balanced at connection dispense
  – Set “Load Balancing = true”

• Fast Connection Failover
  – Automatic severed connection removal
  – Set “HA Events = true”
Commands
Statement Caching

• Retains previously parsed statement in shared pool
  – Cursor stays open on the client side for faster re-use
    • No additional lookup needed on server
  – Metadata remains on the client

• Caches most recently used statements
  – Works with SQL and PL/SQL statements
  – Best with bind variables

• Bind variables recommended for literal values

• Self-tuned cache size – on by default
  – No code changes needed
Commands
Data Retrieval

• Control how much data is retrieved per DB roundtrip
  – Too much data retrieved – excessive client-side memory used
  – Too little data retrieved – additional round trips

• Use OracleCommand.RowSize and OracleDataReader.FetchSize to control result size
  – RowSize populated after statement execution
    • Set dynamically at run-time
  – FetchSize can be set as multiple of RowSize
Fetch Size and Row Size
Commands

Statement Batching

• Use OracleDataAdapter.UpdateBatchSize to batch updates from DataSet

• Execute multiple commands in one DB roundtrip
  – Use anonymous PL/SQL
    • Useful for disparate or similar statements
Commands
Mass Data Movement with Arrays

• PL/SQL associative arrays
  – Pass large amounts of data between .NET and DB of the same data type

• Use parameter array binding
  – Useful if executing the same statement multiple times
  – Bind variables are the same, variable values can be different
  – One execution for each element in the bound array

• Remember: PL/SQL associative arrays and parameter array binding are two different concepts
Promotable Transactions

• Promote local transactions to distributed at run time
  – Better application performance
  – Lower resource usage

• On by default

• Requirements
  – Requires Oracle Database 11g (11.1.0.7+) and ODAC 11.1.0.7.20+
  – Subsequent connections to any other version or database
REF Cursors

- Using SQL provides best performance for straightforward retrieval
- If using PL/SQL and manipulating result set, REF Cursors may be better
- Retrieve data as needed
  - Control data retrieved via FetchSize
  - Fill a DataSet with just a portion of the REF Cursor result
  - Explicit control over what REF Cursor data is retrieved and how
- Defers result set retrieval until needed
- Pass REF Cursors back as input stored procedure parameters
- OracleRefCursor class
SecureFiles and LOBs

• Data retrieval options
  – Defer retrieval (default) with LOB locator
  – Retrieve data immediately with OracleCommand.InitialLOBFetchSize
  – Retrieve a chunk using Read method
    • Use Search method to find data to be retrieved

• Update/Insert/Delete
  – Modify LOB without retrieving the data to the client side
    • Uses LOB locator

• OracleClob and OracleBlob classes

• Use SecureFile data type in Oracle Database 11g and higher
Oracle .NET Caching Solutions

- Oracle .NET client-side DB caches
  - Client Result Cache
  - Continuous Query Notification (CQN) – customizable cache
  - TimesTen In-Memory Database

- Automatically updates/alerts client cache upon server changes
- Each serves separate caching requirements
- Server-side caches can be used with .NET
  - DB In-Memory option
  - Server result cache
  - Etc.
Oracle Client Result Cache

• Automatically updating query result set cache

• Benefits
  – Easy to use
    • No code changes required
  – Snapshot consistent
    • Cache refreshes without user intervention
  – More scalability and performance
    • Data retrieval from client, rather than server
    • No additional round trips
Oracle Client Result Cache Updates

1. Upon data change, client receives change notification on subsequent round trip (or max lag)
   - Invalidation notifications piggyback on existing client round trips
   - No changes to the cache results yet

2. Cache waits for next execution to refresh results
   - Does not initiate an independent round trip
     - No unnecessary DB traffic

Cache entries do not timeout
   - Uses Least Recent Used algorithm
Client Result
Cache
Oracle Continuous Query Notification (CQN)

• Programmatic control over cache notifications and updates
• Also known as Database Change Notification
• Benefits over Client Result Cache
  – More control over how cache behavior
    • What if multiple DB usernames access the same results?
    • What if only a subset of the cached data is required?
    • How long should a query be cached?
    • Do I want additional logic executed when the cache is refreshed?
• Requires significant customization
  – CQN provides cache infrastructure
OracleCommand

Notification Request

Add Dependency

OracleDependency

Application

Listener

OnChange

Execute()

Invalidate()

Data Change

TABLE

Data Dictionary

Notification Queue
Oracle TimesTen In-Memory Database
Memory-Optimized Relational Database

• Fully featured relational database
• Oracle compatible SQL and PL/SQL
• Persistent and durable
  – Transactions with ACID properties
  – Flexible options for durability
• Exceptional performance
  – Instantaneous response time, high throughput, embeddable
• ODP.NET TimesTen provider for TimesTen DB
  – Same ODP.NET APIs
ODP.NET Deployment
Unmanaged – Deployment Platform Target

• Select either “x86” or “x64”
  – Depends on platform target
  – Ensure ODP.NET version supports that platform

• Do not use “AnyCPU” (default)
  – “AnyCPU” instructs assembly to run natively on current CPU
    • x86 or x64
  – May not be intended ODP.NET platform version installed
Unmanaged – Instant Client (IC) Benefits

• Control over install process
  – Xcopy
    • Fine grain control over installation process
    • Great for large scale deployments or ISV packaging
  – OUI – great for small scale or individual deployments

• Smaller install size compared to standard install
  – Xcopy install – smallest footprint
  – OUI install – small footprint

• Fastest client deployment – Xcopy install
Unmanaged – Managing Multiple Oracle Homes

• ODP.NET reads .NET config files for unmanaged Oracle DLL location
  – DllPath setting

• Each app can use different Oracle client DLLs
  – Even if same ODP.NET version is used

• DLL search order
  – 1. Current application directory
  – 2. DllPath – Application config or web.config
  – 3. DllPath – Machine.config
  – 4. DllPath – Windows Registry – HKLM\Software\Oracle\ODP.NET\<version>\DllPath
  – 5. Windows System Path setting
Managed – Managing Multiple Oracle Homes

• Option 1: single configuration file deployment
  – .NET config file can contain TNS, SQL*Net, and LDAP settings
    • i.e. Machine.config, web.config, app.config
    • No *.ora files deployed

• Option 2: multi-file configuration
  – .NET config file plus...
    – Tnsnames.ora, sqlnet.ora, and/or ldap.ora

• Option 3: no configuration files
  – Store connect info in Data Source attribute
Additional Oracle .NET Resources

OTN
otn.oracle.com/dotnet

Twitter
twitter.com/OracleDOTNET

YouTube
youtube.com/OracleDOTNETTeam

Email
alex.keh@oracle.com
Oracle .NET Customer Advisory Board

• Focus group that provides Oracle input and help to prioritize new features
  – Led by Oracle VP

• Work directly with Oracle Development and PM

• Best for organizations in which Oracle .NET is strategic

• Contact me for details and how to apply
Questions and Answers
Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Hardware and Software
Engineered to Work Together