Oracle and .NET: Best Practices for Performance and Deployment

Alex Keh
Senior Principal Product Manager
Program Agenda

- Optimization Process
- Optimizing ODP.NET Performance
  - Connections
  - Data Retrieval and Updates
  - ODP.NET Data Types
  - Caching
Program Agenda

- ODP.NET Deployment
- High Availability – New in Oracle Database 12c
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
Connections

- Use connection pooling
  - Better to have too many connections than too few
- Keep number of connections at steady state
  - Should never destroy or create large numbers of connections
  - Min Pool Size = # connections at steady state or average load
  - Max Pool Size = # connections at maximum capacity
- Close/Dispose connections (and all objects) explicitly
  - Do not necessarily rely on the garbage collector
  - Recommendation applies to all ODP.NET objects
Connections

- ODP.NET performance counters
  - Monitor with Windows Performance Monitor

- New in ODAC 12c
  - More granularity in monitoring
    - Monitor at app domain, pool, or DB instance level
    - DB instance level monitoring applies if load balancing or FCF enabled
Connection Management

RAC and Data Guard

- RAC automatic connection load balancing
  - Load Balancing = true

- Fast Connection Failover
  - RAC and Data Guard automatic “bad” connection removal
  - 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

- 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
  - First connection to Oracle Database 11g (11.1) or higher
  - Subsequent connections to any other version or database
ODP.NET Data Types

- Avoid unnecessary data type conversions
- .NET vs. ODP.NET Types
  - OracleDataReader Type Accessors
  - OracleParameter.DbType vs. OracleParameter.OracleDbType
- Option to store Oracle data types in DataSet
  - OracleDataAdapter.ReturnProviderSpecificTypes = true
  - Available in ADO.NET 2.0
REF Cursors

- 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
SecureFiles and LOBs

- Update/Insert/Delete
  - Modify LOB without retrieving the data to the client side
    - Uses LOB locator
- Use SecureFile data type in Oracle Database 11g
Oracle .NET Caching Solutions

- Oracle .NET database caches
  - Client Result Cache
  - Continuous Query Notification (CQN)
  - TimesTen In-Memory Database

- Automatically updates/alerts client cache upon server changes
- Each serves separate caching requirements
Oracle Client Result Cache

- Automatically updating cache of query result sets
- 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 users 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?
OracleCommand

Add Dependency

OracleDependency

Application

Notification Request

OnChange

Listener

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 with ODP.NET
- Persistent and durable
  - Transactions with ACID properties
  - Flexible options for durability
- Exceptional performance
  - Instantaneous response time, high throughput, embeddable
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

- Solution for ODP.NET 10.2.0.4 and higher
  - ODP.NET reads .NET config files for location of unmanaged Oracle DLLs
  - Each app can use different Oracle client DLLs even if same ODP.NET managed version is used
Unmanaged – Managing Multiple Oracle Homes

- DLL search order
  - 1. Current application directory
  - 2. Application config or web.config
  - 3. Machine.config
  - 4. 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 contains TNS, SQL*Net, and LDAP settings
    - i.e. Machine.config, web.config, app.config

- Option 2: multi-file configuration
  - .NET config file
    - Tnsnames.ora, sqlnet.ora, and ldap.ora

- Option 3: no configuration files
  - Store connect info in Data Source attribute
High Availability – New in Oracle Database 12c
Global Data Services

- Extend RAC services to a global basis
  - Access to FCF, load balancing, and affinity capabilities
  - RAC, Active Data Guard, and GoldenGate can participate

- Benefit
  - Optimizes utilization, HA, and performance

- No code changes

- Requires Oracle Database 12c and ODP.NET 12c
Global Data Services

Concepts

Without GDS

Sales Service

GoldenGate

Sales Service

With GDS

Sales Global Service

GoldenGate
Faster and More Graceful Planned Outage

- Offline DB alerts ODP.NET of impending downtime
- ODP.NET stops allocating and closes idle connections
  - Connections returned to the pool are closed
- Benefit
  - DB brought offline as quickly as possible without end user disruptions
- Set ODP.NET attribute “HA Events = true”
- Requires Oracle Database 12c and ODP.NET 12c
Transaction Guard

- ODP.NET can determine whether a transaction committed even upon a DB failure
- Benefit
  - Ensures transaction commits at most once
- App can query transaction outcome
  - OracleConnection properties return transaction ID and status
  - OracleLogicalTransactionStatus class
- Requires Oracle Database 12c and ODP.NET 12c
Recoverable Error Detection and Recovery

- After DB error, ODP.NET determines if failed transaction is recoverable or not
  - Returns error message if operation cannot be retried
- `OracleException.IsRecoverable` property indicates if transaction recoverable and can be retried
- Benefit
  - Determine with certainty whether to rollback or resubmit
- Requires Oracle Database 12c and ODP.NET 12c
Transaction Guard Scenario

1. ODP.NET receives FAN down event or error.

2. IsRecoverable=false → roll back.
   IsRecoverable=true → re-submit.

3. To re-submit, retrieve LogicalTransactionId.

4. Obtain new session and retrieve txn status.

5. If committed and completed, done.
   If not committed nor completed, re-submit.
Conclusion and Q & A
Additional Oracle .NET Resources

OTN
otn.oracle.com/dotnet

Twitter
twitter.com/OracleDOTNET

YouTube
youtube.com/OracleDOTNETTeam

Email
alex.keh@oracle.com
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