

Accelerate Application Performance

Tips for Faster Oracle Database .NET Programs

Alex Keh and Christian Shay
Senior Principal Product Managers
Server Technologies
October 24, 2018



**Live for
the Code**

Program Agenda

- 1 Optimization Process
- 2 ODP.NET – Connections, Data Retrieval/Updates, Data Types
- 3 Caching
- 4 Oracle Performance Analyzer and SQL Tuning Advisor

Optimization Process

Oracle .NET

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

ODP.NET

Connections, Data Retrieval and Updates, Data Types

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
 - See documentation for connection pooling parameter options
- ODP.NET performance counters
 - Monitor with Windows Performance Monitor or programmatically
 - Granular monitoring at app domain, pool, or DB instance level
 - See [documentation](#) for details on how to enable

Connection Management

RAC, Data Guard, GoldenGate, and GDS

- Fast Application Notification (FAN) infrastructure
 - ODP.NET 12c and 18c (mostly) uses Oracle Notification Service (ONS)
 - ONS delivers FAN events faster than previous AQ infrastructure
- **Run-time** connection load balancing
 - Automated load balancing at connection dispense time
 - Better performance because dispense based on real-time load
 - Set “Load Balancing = true” in connection string
 - Now the default in ODP.NET 12.2

Commands

Bind Variables

- Prevents re-parsing of frequently executed statements
 - Works with SQL and PL/SQL statements
- Improves subsequent command executions
 - Literal value changes forces a re-parse and re-optimization
 - Literal values should become bind variables
- Executed statements stored in Oracle shared pool
 - Re-parsing and re-optimization uses CPU and requires shared pool locks

Commands

Statement Caching

- Retains previously parsed statement in shared pool
 - Prevents repeated parsing in server
- 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
 - Can be set dynamically at run-time
 - `FetchSize` can be set as multiple of `RowSize`

ORACLE®

D E M O N S T R A T I O N

Fetch Size and Row Size

Commands

Data Updates

- Statement batching
 - Use `OracleDataAdapter.UpdateBatchSize` to batch updates from `DataSet`
 - Execute multiple commands in one DB roundtrip
 - By using anonymous PL/SQL

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

REF Cursors

- OracleRefCursor class
- Defers result set retrieval until needed
- Retrieve data as needed
 - Control data retrieved via FetchSize
 - Fill a DataSet with just a portion of the REF Cursor result
- Usage
 - Can create REF Cursors as part of an anonymous PL/SQL block
 - Can return REF Cursors from stored procedures
 - Can pass REF Cursors to database as input stored procedure parameters

SecureFiles and LOBs

- Data retrieval options
 - Control amount of data returned with `OracleCommand.InitialLOBFetchSize`
 - Retrieve a chunk using `OracleClob` and `OracleBlob` classes `Read` method
 - Use `Search` method to find data to be retrieved
- Update/Insert/Delete SQL statements acting on LOBs
 - Modify LOB without retrieving the data to the client side
 - Uses LOB locator
- `VARCHAR2`, `NVARCHAR2`, and `RAW` now support up to 32 KB
 - ODP.NET 12c and Oracle DB 12c enhancement

Caching

Oracle .NET

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 meets separate caching requirements
- Server-side caches can be used with .NET
 - DB In-Memory option, server result cache, etc.
- Oracle Coherence – application server in-memory grid cache

Oracle Performance Analyzer and SQL Tuning Advisor

Oracle and Visual Studio

SQL Tuning Advisor

- Use when designing new SQL statements
- Tune ad-hoc SQL statements in Query Window
- Tune bad SQL found by Oracle Performance Analyzer
 - (Use if SQL is performing poorly under load)

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

ORACLE[®]

D E M O N S T R A T I O N

SQL Tuning Advisor

Oracle Performance Analyzer

- Detects performance issues in an application's use of the database under load
- Requirements
 - SYSDBA
 - Oracle Database license for Oracle Diagnostic Pack
- Can be use during testing
- Can be also used on production applications

Oracle Performance Analyzer

- Simple to use
 - Connect in Server Explorer as SYSDBA (to container if using Oracle Multitenant)
 - Run your application
 - Start Oracle Performance Analyzer
 - Enter amount of time to analyze
 - Press Start to start timer
 - Sufficient “database time” required to get results
 - View findings and actions
 - Implement recommended actions
 - Run again for additional recommendations

ORACLE®

D E M O N S T R A T I O N

Performance Analyzer

Additional Oracle .NET Resources



OTN

otn.oracle.com/dotnet



GitHub

github.com/oracle/dotnet-db-samples/



Twitter

twitter.com/OracleDOTNET



YouTube

youtube.com/OracleDOTNETTeam



Email

alex.keh@oracle.com and christian.shay@oracle.com

Upcoming .NET Sessions – Thursday

Online schedule: <https://bit.ly/2yXG8oQ>

- Eliminate Application Downtime with Oracle Database and .NET
 - Thursday – 9:00 AM – 9:45 AM Moscone West – 3010
- Hands on Lab: Building .NET Applications with Oracle
 - Thursday – 12:00 PM – 1:00 PM Marriott Marquis (Yerba Buena Level) – Salon 3 / 4

Questions and Answers