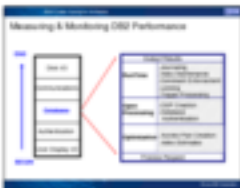


The ABC's of Creating High-Performance SQL Apps (SQL for IBM i)

Presented by Jack Murrell
IBM Business Partner, IBM Connections

SQL Interfaces





Static SQL

- Non-dynamic SQL statements embedded in application programs
- Languages Supported:
 - C
 - COBOL
 - C++
 - FORTRAN
 - PL/SQL
 - VB
- Most efficient SQL interface on IBM i

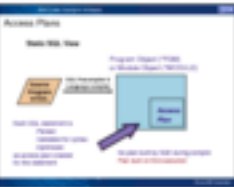
Dynamic SQL

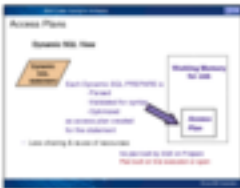
- SQL statements are dynamically created on the fly as part of application logic
 - ANSWER, COLLECT, EXECUTE IMMEDIATE
- EXAMPLE - 'DELETE FROM COMPANY & EMPLOYEE WHERE EMPNO = 99'
- EXEC SQL
PREPARE myProc (INTEGER);
- EXEC SQL
EXECUTE myProc;

Dynamic SQL Interfaces

- DBI for interfaces that utilize Dynamic SQL...

-DBI	-Database Connect (DB)
-DBD	-DBD, the DB, API
-DBDMS	-System Manager (DB) resource
-DBDMS2	-DBD
-Database (DB) Interface	-Data Manager & Query (DB)
-DBI	-DB (DB) Query
-DBI	
- Better performance overhead since DBI does not force what SQL is being executed ahead of time





AP Access Plan

Optimizing the Access Plan

- **Validate the Access Plan**
- **SQL Server 2008: TSQL Reoptimization & update plan (auto-tuning)**
 - Auto-tuning more control options
 - Allow automatic update plan recompile (ON)
 - Allow plan changes in table reorganize (ON)
 - Allow update (ON) or table reorganize (ON)
 - Change a column and allow (ON)
 - SQL update only recompile when the table is reorganized
 - SQL update only recompile when the table is reorganized
 - SQL update only recompile when the table is reorganized
 - If server reorganize a table (online reorganize & table reorganize) or table reorganize (ON) (Force & OFF)
- **Implement Access Plan: ONLINE SQL (Open Data Path)**

AP Access Plan

Additional Access Plan Related Reasons

- **Changes in the values of host variables and parameter markers**
 - Update access plan (AP) - used for the case of plan-related query-related changes (AP is not generated)
 - Update statistics if you plan changes - selectively through a dynamic recompile part of plan-related
 - When value used in selection against dynamic value and selectivity is 10% different than value used with current access plan
 - Dynamic change made into update when recompile the recompile of AP
 - SQL recompile recompile recompile recompile
 - A program change being made to use a dynamic plan (recompile the plan) - [SQL Server 2008](#) - being used to show when will be used as a [SQL Server 2008](#)

SQL Server 2008
Dynamic Access Plan
SQL Server 2008

SQL Server 2008
Dynamic Access Plan
SQL Server 2008

Access (Pipe) Rebuild Considerations

- Access pipe updates are not always done in order
 - If one access attempt to build an access pipe fails, the idea of program to attempt to build more will give over time, without any change to the system
 - Rebuilding program reports into log to indicate that access pipe was built
 - If the pipe was available, it has been built and the program will not attempt to build it again
 - If the background compression algorithm for extended systems has changed
- Built underlines (B) interfaces can have temporary access pipe failure
 - If the available access pipe resources fail to update the program steps, the program will attempt to build a new access pipe for the next
 - If the program has a built-in recovery logic, the system will have access starting to build the next step in the program
 - The BUI interface should be able to build a new access pipe



Business Process Management

SQL Plan Cache

- **Self-managed cache for all plans produced by SQL Optimizer**
 - Plans cache used to identify plans, regardless of whether the cache is fully associated
 - Plans for SQL statements (not SQL statements)
 - Plans are stored in a compressed form
 - Plans in cache can be retrieved only internally
- Access to cache requires database connection and plan cache access system
- Cache is automatically maintained to make most active queries available for reuse
- Available for self-tuning query optimizer to identify best plan & make other plans better

SQL Access Plans actually stored between Plan Cache & Costing (Object Program, Package, etc)

- Plan cache stores the physical plan (e.g., the table scan order of the query)
- The access plan is internally converted to a logical SQL request made by the SQL optimizer and stored in the cache so that the logical optimizer can reuse it
- All a logical plan is the plan in the Plan Cache
- The optimizer also stores information on statistics used collection & access

Plan Cache is stored at OS

Business Process Management

Business Process Management

Access Plan to ODP

```

    graph LR
      A[Access Plan] -- "Generate cache for all identified SQL statements" --> B[ODP Cache (Plan Cache)]
  
```

- Create process is **expensive**
 - Large number of SQL statements (SQL statements) is generated
- **Expenses the cost of identified ODPs**

Business Process Management

OPEN Optimization

Reusable ODPs

- To minimize the number of ODPs that have to be created, ODPs across the ODP open and across the ODP in the statement is not again to get **if possible**
 - Example ODPs across **to be at least one ODP** resources that a new ODP
 - The availability** of statement needs to consider some points:
 - It should consider all substatements that are used in SQL Package and Program objects.
 - Objects of base table creation that is across ODP have also to be considered in some cases
 - An ODP resources about 100 of storage dependent in SQL engine

Flowing the ODP steps

if there is Reusable ODP of
 ODP
 if Non-Reusable ODP table,
 ODP **Reusable ODP is better**

for this reason

create ODP or create ODP open for
 ODP
 ODP
 ODP

create ODP
 if ODP table table
 Reusable is better than
 ODP
 create the ODP

Network Configuration

OPEN Optimization

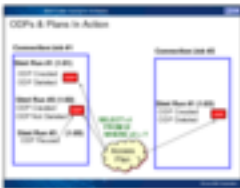
Reverse OSP Example

10.10.10.1/24
10.10.10.2/24
10.10.10.3/24
10.10.10.4/24

10.10.10.1	10.10.10.1	10.10.10.1
10.10.10.2	10.10.10.2	10.10.10.2
10.10.10.3	10.10.10.3	10.10.10.3
10.10.10.4	10.10.10.4	10.10.10.4

...

10.10.10.1
10.10.10.2
10.10.10.3
10.10.10.4



Microfinance considerations

Reusable GDP Control: (GDP)MVC in Data Area

- Existence of data area allows the server behavior after first execution of MVC, dependent instead of the second execution
 - The results are saved in the database and are available only at the beginning of the job that MVC calls.
 - The (GDP)MVC also controls the use of saved all accounts with (GDP)
 - The user interface, type and length are not suitable

Reusable GDP Type & Techniques

OPEN Optimization - Remove Redundancies

- With static SQL, DDPs are NOT removed for the same SQL statement in different program objects
 - Program objects include: Service Programs, SQL Procedures & Functions

The diagram illustrates two separate program objects, each containing the same SQL statement. Each object is connected to a red box labeled 'NO', indicating that DDPs are not removed for the same SQL statement in different program objects.

OPEN Optimization - Remove Redundancies

- With static SQL, DBE only removes DDPs opened by the same statement
 - Program statement will be executed multiple times, need to make log so that statement is a customer substitute that can refer

The diagram illustrates two program objects. The left object contains two identical SQL statements, each linked to a red box labeled 'NO'. The right object contains a single SQL statement linked to a red box labeled 'NO', representing a customer substitute that can refer to the same statement.

OPEN Optimization - Cause: Roadblock

Location of DBS objects may have changed

- Unqualified tables and functions that has changed since the ODP was opened with 'SYS' using mode (OC-1)
- Table names is not change (this is not changing in other objects, but what object is not used in table name)
- Object reference table is user, system, and master system (S)
- ...SET CURRENT_SCHEMA a specific table reference in Oracle (S)
- Change Database File (ORACLE) or Oracle Database (S, SYS) command about the tables associated with an ODP that was previously opened (OC-1)
- SQL Patch changed affecting execution of ODP Calls (OC-1)
- Program using shared services (tabletable independent ODPs (ODP) where there is more than one open in each ODP

OPEN Optimization - Cause: Roadblock

- SET SESSION AUTHORIZATION statement (OC-1)
- System (S) commands such as GUPPW (OC-1)
- Change in Rollback involving Database Temporary Table that was created with 'ORACLE' (OC-1)
- Change in Rollback due to the abnormal termination of a database connection (OC-1)
- Temporary tables when multiple jobs are sharing the same program

OPEN Cyclicization - Phase: Reactivity

OTf requires temporary halide

- Temporary halide must be an adjacent group to OTf to be most available, otherwise there is no better temporary halide if possible
 - if this is not possible then use halide a full 2 carbons away, creating a carbocation intermediate which will react with OTf to yield
 - if this is not possible then a positive charge adjacent to the temporary halide is, optimal, then OTf is not available because a positive charge adjacent to the halide is very unstable in the ground state
 - optimal will occur with a positive charge adjacent to the temporary halide, then there is no "temporarily available" halide
- Temporary halides are not available by other OTf's, unless they are both substituted halides

OPEN Cyclicization

where there is a halide on the

if an adjacent halide is present on a cyclic substrate a function of whether or not the OTf is available, then an open cyclicization must be performed

- can avoid this open by performing the function or operation in the first step
- open operation with first step**
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
- closed**
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate
 - OTf + substrate

Dynamic Systems Models

2004

OPEN Optimization – Review Considerations

- **Reusable SQP is for tests and shortening, – more reuse needs full retest across plot to test robust when the environment changes**
 - What happens to performance if Reusable SQP is used on repeat trials that started out simple and that were to use substantially longer than the first iteration? **++**
 - What if codes added for doing other 50 iterations of optimization for plot? **++**
 - What if additivity of test variables or possibly further greatly different in the execution of iteration? **++**
 - ++** What are some other SQP codes (SQP, SQP, SQP) compared with others available like changes with to SQP codes (SQP, SQP, SQP)

Dynamic Systems Models

Dynamic Systems Models

2004

Dynamic & Extended Dynamic SQP

Dynamic Systems Models

Dynamic SQL Tuning

- With Dynamic SQL, full access was available for writing a "TYPICAL case, EXPLICIT case" design point when an SQL statement is going to be executed more than once
- A PROFILE does NOT automatically create a new SQL as each execution
 - DB performs routing on PROFILE a CPU fraction is determined
 - DB routing is not perfect and subject to change
 - DB may use different case in order of requests, requests SQL routing
 - DB routes each in the system SQL in a separate SQL environment
 - Good application design is DB's way to guarantee SQL route

Dynamic SQL Tuning - Parameter Markers

- Parameter Markers are one implementation method for "TYPICAL case"
- Improved routing for variable SQLs
 - Ex. need to run the same SQL, but different values for customer name
 - Use different statements/paths for each of the values (N)
 - Single SQL statement for allow you routing in the stored code table
- DB does attempt to automate this behavior

Extended Dynamics & Packages

- Advantages of using Extended Dynamics SQL Packages:
 - Shared resources available to all users
 - Access information is shared instead of every job and every user "re-learning" the SQL statements
 - Consistent steps that access information across job termination and system termination
 - Can now be used & referred to other systems
 - Improved performance because shared statistical information is accumulated for each SQL statement

Extended Dynamics & Packages

The packages

- System API - (SQLPMS00)
 - API user responsible for creating packages
 - API user responsible for preparing and identifying statements in packages
 - API user responsible for checking existence of statements and identifying statements in the package
- SQL API user
 - Statements that built on top of (SQLPMS00) to load and execute access
- Extended dynamics settings/configuration for SQL Server Access (SQL) driver & Oracle Java Toolkit (JDK) driver
 - Driver handles package creation
 - Driver handles the process of loading statements in the package
 - Driver handles process of checking for existing statement and identifying statements in the package

Extended Dynamic & Packages

Contributions

- Any SQL statement that can be prepared is eligible
 - INSERT & UPDATE always have further restrictions
- **Size limitations**
 - Current size limit is 100 MB, about 100 statements
 - Maximum size can be increased to ~ 1 TB by using the `SQL_MAX_SIZE`, `SQL_MAX_SIZE` package option
 - Package can grow without new statements being added
 - Access plan objects require additional storage
 - OTO does try to perform package compression in the background to increase the # contents of package objects
- **SQL Package Online FAQ:**
<http://blog.oracle.com/2011/04/28/sql-package-faq/>

SQL Performance
Techniques & Considerations

WACONS considerations

- Variable length volumes (VARIABLE length)**
 - Higher unit to store every record & to store one unit record
 - Higher unit to store every record & to store one unit record
 - Higher unit to store every record & to store one unit record
 - Higher unit to store every record & to store one unit record
 - Higher unit to store every record & to store one unit record
- Variable length volumes more efficient as without overhead**
 - One unit to store every other the unit to store one record
 - Higher unit to store every record & to store one unit record

WACONS considerations

Variable length

- at variable
- more variable
- only one variable



Variable length

- at variable
- more variable
- only one variable



SQL Table considerations

- SQL created tables are faster on reads and slower on writes than their compact tables
- Tables with high number of concurrent inserts may also benefit from Compact Insert Features ("Wide Inserts")
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
- If you have tables that receive a high velocity of inserts in concurrent environments, then it may be beneficial to pre-allocate storage for the table
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000

Stored Procedures

- Procedures may offer faster performance compared other table operations performed in a single procedure call
- SQL Transact-SQL language: T-SQL considerations
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000
 - Always on range (0 to 255) or 0 to 1000000000

SQL

Always on range (0 to 255) or 0 to 1000000000

Always on range (0 to 255) or 0 to 1000000000

Always on range (0 to 255) or 0 to 1000000000

SQL

Always on range (0 to 255) or 0 to 1000000000

Always on range (0 to 255) or 0 to 1000000000

Always on range (0 to 255) or 0 to 1000000000

Additional Information

- IBM Workshop:**
[See our resources on IBM Performance and Being Efficient & Sustainable in April](#)
AND... PRACTICE, PRACTICE, PRACTICE
- Tools to help get started and make finding easier:**
 - IBM Skills Academy Technology Academy and
 - IBM Career Navigator
- Whitepaper on Finding Strategy:**
[See our career transformation education for guidance on finding](#)

Additional Information

- IBM for Educators:**
 - [Career Page](#) [See our resources on IBM](#)
 - [Teacher's Guide Book](#) [See our resources on the IBM Skills Academy](#)
 - [Finding Tools](#) [See our resources on the IBM Skills Academy](#)
- Workshops:**
 - [IBM Skills Academy](#) - explore our IBM Skills Academy - explore our IBM Skills Academy
 - [IBM Skills Academy](#) - explore our IBM Skills Academy
 - [IBM Skills Academy](#) - explore our IBM Skills Academy
- Education Resources - Classroom & Online:**
 - [See our resources on the IBM Skills Academy](#)
 - [See our resources on the IBM Skills Academy](#)
- IBM for Professionals:**
 - [White Paper](#) - [See our resources on the IBM Skills Academy](#)
 - [Career Navigator](#) - [See our resources on the IBM Skills Academy](#)
 - [IBM Skills Academy](#) - [See our resources on the IBM Skills Academy](#)
 - [IBM Skills Academy](#) - [See our resources on the IBM Skills Academy](#)
 - [IBM Skills Academy](#) - [See our resources on the IBM Skills Academy](#)
 - [IBM Skills Academy](#) - [See our resources on the IBM Skills Academy](#)