DB2 10 for z/OS

Optimization Hints

Prepared by Angelo Sironi
Sironi IT Consulting s.n.c.

Milan, 9 April 2013
Rome, 10 April 2013

Table of Contents

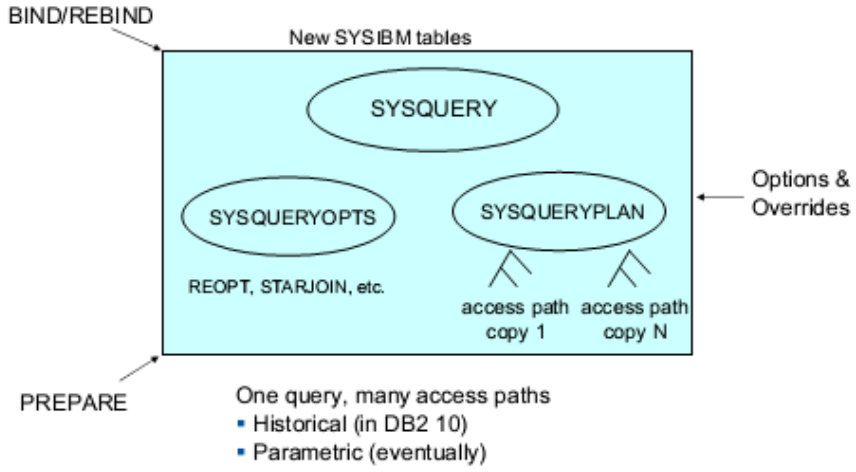
Optimization Hints: Hint Matching

- **Pre-DB210**
  - User-level
  - Based on QUERYNO
  - Usually, not suited for Dynamic SQL
- **New with DB2 10**
  - Based on SQL statement text
  - Hints can be enforced for the entire DB2 subsystem, irrespective of static vs. dynamic, etc.
  - Hints integrated into the new Access Path Repository
    - SYSIBM.SYSQUERY
    - SYSIBM.SYSQUERYOPTS
    - SYSIBM.SYSQUERYPLAN

Optimization Hint Types

- **PLAN_TABLE hints** (a.k.a. user level hints)
  - Available pre-DB2 10
  - Tries to enforce a particular access path for an SQL statement that is issued by a specific single authorization ID
- **Statement-level optimization parameter hints**
  - Specifies that DB2 uses certain optimization parameters to process all statements that match the hint
- **Statement-level access path hints**
  - Specifies that DB2 tries to use specified PLAN_TABLE rows to determine the access path for matching SQL statements

Access Path Repository



Source: Terry Purcell, What's new from the optimizer in DB2 10 for z/OS?, IDUG 2010 North America

Optimization Hints: Recommended Maintenance

- **PM63698**
  - MSGDSNT291I ISSUED INVALIDLY ON BIND QUERY COMMAND
- **PM71137**
  - EXPLAININPUTSCHEMA SUPPORT FOR BIND QUERY

Enabling optimization hints

- **Set OPTHINTS=YES in DSNZPARM to enable**
  - SET CURRENT OPTIMIZATION HINT statements
  - The OPTHINT bind option
  - Matching for statement-level hints that are stored in the SYSIBM.SYSQUERY, SYSIBM.SYSQUERYPLAN, and SYSIBM.SYSQUERYOPTS catalog tables
- **Create *userid*.PLAN_TABLE and index *userid*.PLAN_TABLE_HINT_IX on columns**
  - QUERYNO
  - APPLNAME
  - PROGNAME
  - VERSION
  - COLLID
  - OPTHINT

New Statement-Level Optimization Hints

- **This type of hint applies to all instances of a (static or dynamic) statement**
  - System-wide, or ..
  - From a particular package
- **The hint can specify**
  - Specific bind options to be applied (Parameter Hint)
  - A particular access path (Access Path Hint)
- **Matching of statement text used to apply the hint**
- **Hints of this type are stored in the catalog tables that make up the Access Path Repository**
- **Statement-level hints are not supported for statements in packages created by**
  - CREATE FUNCTION
  - CREATE TRIGGER
  - CREATE PROCEDURE

Stmt-Level Optimization Hints: Pre-reqs.

- User has one of the following authorities:
  - SQLADM
  - SYSOPR
  - SYSCTRL
  - SYSADM
- An instance of the DSN_USERQUERY_TABLE and the PLAN_TABLE tables have been created under the same user schema
  - See DDL in members DSNTESC and DSNTESH of the SDSNSAMP library
- All object names are UPPER CASE in the query text of the SQL statements
- The package that contains the statement was created by a BIND PACKAGE statement

Statement-Level Opt. Hints: Scope

- Specified by the value of column HINT_SCOPE in DSN_USERQUERY_TABLE
  - System-wide
    - HINT_SCOPE = 0
    - SCHEMA = *blank or ''*
  - Under a particular Schema
    - HINT_SCOPE = 0
    - SCHEMA = some value
  - A particular Collection and Package and VERSION (or any VERSION)
    - HINT_SCOPE = 1
    - COLLECTION = some value
    - PACKAGE = some value
    - VERSION = '' ','*' or some value
- Hint applied based on statement text matching within the specified scope

Statement-Level Opt. Hints: Query Text

- **Matching SQL query text supplied by user**
  - Column QUERY_TEXT of table DSN_USERQUERY_TABLE
- **Recommended source of SQL query text for populating DSN_USERQUERY_TABLE**
  - For Static SQL
    - SYSIBM.SYSPACKSTMT catalog table
    - DBRM
  - For Dynamic SQL
    - Dynamic Statement Cache (DSC)
  - Ensure that object names and SQL keywords in the statement text are specified by **upper case** characters
    - especially for dynamic SQL statements

Statement-level Optimization Parameter Hints

- **Available Parameters**
  - REOPT bind option
    - 1 – REOPT(ONCE)
    - A – REPOT(AUTO)
    - N – REOPT(NONE)
    - Y – REOPT(ALWAYS)
  - STARJOIN subsystem parameter
    - Y – Star join is enabled
    - N – Star join is disabled
  - PARAMDEG subsystem parameter (0 – 254) (MAX_PAR_DEGREE column)
  - CDSSRDEF subsystem parameter (DEF_CURR_DEGREE column)
    - ONE – Query parallelism is disabled
    - ANY – Query parallelism is enabled
  - SJTABLES subsystem parameter

Statement-level Optimization Parameter Hints

- **Reasons for interest**
  - Use REOPT for «generic» SQL statements like the following

```
SELECT …
  FROM MY_TABLE
WHERE C1 BETWEEN ? AND ?
  AND C2 BETWEEN ? AND ?
  AND …. BETWEEN …
```

```
SELECT ….
  FROM MY_TABLE
 WHERE SURNAME LIKE ?
   AND SSN LIKE ?
   AND FISCAL_CODE LIKE ?
   AND … LIKE …
```

```
SELECT …
  FROM MY_TABLE
 WHERE ( :h = 1 OR C1 = :h )
   AND ( :h = 2 OR C2 = :h)
   AND  (:h = 3 OR C3 BETWEEN … )
   AND etc…
```


DSN_USERQUERY_TABLE: Parm. Hints

QUERYNO	HINT SCOPE	QUERY TEXT	SCHEMA	COLLECTION	PACKAGE	VERSION
Specified value <> PLAN_TABLE QUERYNO value and any other DSN_USERQUERY_TABLE QUERYNO value! QUERYNO used only for primary key of DSN_USERQUERY_TABLE	**0**		A value or *blank*	Any value (not used)	Any value (not used)	Any value (not used)
	1		A value or *blank*	A value	A value	A value or '*' or blank

(.. additonal columns on next slide…)

DSN_USERQUERY_TABLE: Parm. Hints

REOPT	STARJOIN	MAX PAR DEGREE	DEF CURR DEGREE	SJTABLES	USERFILTER	QUERYID
'A' = AUTO '1' = ONCE 'N' = NONE 'Y' = ALWAYS blank = Not specified	'Y' = enabled 'N' = disabled	A value (-1, if unspecified)	'ONE' Parallelism disabled. 'ANY' Parallelism enabled. *blank* Not specified.	Minimum number of tables to qualify for the star join processing, or -1 when not specified	A filter name that groups a set of queries together, or *blank*	Set by DB2 at BIND QUERY. Matches PK in Access Path Repository tables

Stmt-Level Parameter Opt. Hints: How-to

- **INSERT rows into DSN_USERQUERY_TABLE**
  - **Specify scope**, with values for columns
    - HINT_SCOPE = 0 or 1
    - QUERY_TEXT
      - Insert the text of the SQL statement (in DB2 internal format…)
    - SCHEMA (blank if all table references qualified; else, should match the proper schema qualifier)
    - COLLECTION (optional if HINT_SCOPE = 0)
    - PACKAGE (optional if HINT_SCOPE = 0)
    - VERSION (optional if HINT_SCOPE = 0)

Stmt-Level Parameter Opt. Hints: How-to (cont.)

- **Specify parameter(s) of interest**, with a value for columns
  - REOPT
  - STARJOIN
  - MAX_PAR_DEGREE
  - DEF_CURR_DEGREE
  - SJTABLES
- **Identify the hint** with a value for column QUERYNO
  - To be a valid Parameter Hint
    - The specified value must NOT exist in a PLAN_TABLE row
    - Must be Unique with respect to the current content for the DSN_USERQUERY_TABLE in use

Stmt-Level Parameter Opt. Hints: How-to (cont.)

- **Issue a BIND QUERY command**
  - Omit the LOOKUP option, or specify LOOKUP(NO)
  - Specify EXPLAININPUTSCHEMA(*schema-name*) to qualify the EXPLAIN tables to be used as input
    - DB2 creates hints for every DSN_USERQUERY_TABLE row, by inserting data into the following catalog tables
      - SYSIBM.SYSQUERY
      - SYSIBM.SYSQUERYOPTS
    - The QUERYID column correlates rows in these tables
- **Bind or rebind the interested Packages**
  - You do not need to specify a value for OPTHINT (unless you also need to enable Access Path Optimization Hints for the Package)

Stmt-Level Parm. Opt. Hints: Example (1 of 2)

```
INSERT INTO DSN_USERQUERY_TABLE
( QUERYNO, SCHEMA, HINT_SCOPE, QUERY_TEXT,
  USERFILTER, OTHER_OPTIONS, COLLECTION,
  PACKAGE, VERSION, REOPT, STARJOIN, MAX_PAR_DEGREE,
  DEF_CURR_DEGREE, SJTABLES, OTHER_PARMS )
SELECT 1111111 AS QUERYNO, 'ANGELOS' AS SCHEMA
     , 1 AS HINT_SCOPE, STATEMENT, '' AS USERFILTER
     , '' AS OTHER_OPTIONS, COLLID, NAME, VERSION
     , 'Y' AS REOPT, '' AS STARJOIN
     , -1 AS MAX_PAR_DEGREE
     , '' AS DEF_CURR_DEGREE, -1 AS SJTABLES
     , '' AS OTHER_PARMS
  FROM SYSIBM.SYSPACKSTMT
 WHERE COLLID = 'CLSIR01'
   AND NAME = 'SIRSAMP1'
   AND STMTNO  = 88   ;
```


19

Stmt-Level Parm. Opt. Hints: Example (2 of 2)

- **BIND QUERY**

```
DSN SYSTEM(DSNX)
BIND QUERY
DSNT280I  +DSNX BIND QUERY FOR QUERYNO = 1111111
           SUCCESSFUL
DSNT290I  +DSNX BIND QUERY COMMAND COMPLETED
END
```

  - DB2 updates the following Catalog tables
    - SYSIBM.SYSQUERY
    - SYSIBM.SYSQUERYOPTS
  - Unique value assigned by DB2 to QUERYID column in DSN_USERQUERY_TABLE
- **Rebind interested Package(s)**
  - No SYSLOG information provided on applied hints
  - For option REOPT = 'Y', check SYSIBM.SYSPACKSTMT
    - STATUS = 'G'

20

Stmt. Level Access Path Hints: Pre-reqs.

- Same pre-req. as for general Statement Level Optimization hints
  - See previous pages
- An instance of the PLAN_TABLE created under user schema
- An index created on the following PLAN_TABLE columns
  - QUERYNO
  - APPLNAME
  - PROGNAME
  - VERSION
  - COLLID
  - OPTHINT

Stmt. Level Access Path Hints: Pre-reqs.

- User has one of the following authorities:
  - SQLADM
  - SYSOPR
  - SYSCTRL
  - SYSADM
- An instance of the following tables is created under user schema
  - DSN_USERQUERY_TABLE
  - PLAN_TABLE
- An index is created on the following PLAN_TABLE columns
  - QUERYNO, APPLNAME, PROGNAME, VERSION, COLLID, OPTHINT

Stmt. Level Access Path Hints: Pre-reqs.

- All object names are UPPER CASE in the query text of the SQL statements
- The package that contains the statement was created by a BIND PACKAGE statement

Stmt-Level Access Path Hints: How-to

- INSERT rows into DSN_USERQUERY_TABLE
  - **Specify scope**, with values for columns
    - HINT_SCOPE = 0 or 1
    - QUERY_TEXT
      - Insert the text of the SQL statement (in DB2 internal format…)
    - SCHEMA (optional)
    - COLLECTION (optional if HINT_SCOPE = 0)
    - PACKAGE (optional if HINT_SCOPE = 0)
    - VERSION (optional if HINT_SCOPE = 0)
  - **Identify the hint** with a value for column QUERYNO
    - Insert the value of the QUERYNO column of existing PLAN_TABLE rows that describe the access path that you want to enforce

Statement-Level Opt. Hints: How to (cont.)

- **Issue a BIND QUERY command**
  - Omit the LOOKUP option, or specify LOOKUP(NO)
    - DB2 creates hints for every DSN_USERQUERY_TABLE row, by inserting data into the following catalog tables
      - SYSIBM.SYSQUERY
      - SYSIBM.SYSQUERYOPTS
    - The QUERYID column correlates rows in these tables
- **Bind or rebind the interested Packages**
  - Do not specify a value for OPTHINT

Precedence of Optimization Hints

- **When several optimization hints of different types might apply to a particular statement, DB2 applies a single hint, according to the following precedence order:**
  1. PLAN_TABLE hints
  2. Statement-level access path hints
  3. Statement-level parameters
  4. Hints created by DB2 for access path reuse

BIND QUERY LOOKUP(NO)

- **BIND QUERY LOOKUP(NO)**
  - Reads information from DSN_USERQUERY_TABLE and inserts the data into catalog tables
    - SYSIBM.SYSQUERY
    - SYSIBM.SYSQUERYPLAN
    - SYSIBM.SYSQUERYOPTS
- **Messages**
  - DSNT280I, if row inserted successfully
  - DSNT281I, if row not inserted successfully

BIND QUERY LOOKUP(YES)

- **BIND QUERY LOOKUP(YES)**
  - Reads information from DSN_USERQUERY_TABLE and looks for matching rows in the catalog tables
    - SYSIBM.SYSQUERY
    - SYSIBM.SYSQUERYPLAN
- **Messages**
  - DSNT280I, for each valid matching row in SYSIBM.SYSQUERY or SYSIBM.SYSQUERYPLAN
  - DSNT281I, if no valid matching row exists in SYSIBM.SYSQUERY or SYSIBM.SYSQUERYPLAN
  - DSNT291I, if no rows have valid matching row in SYSIBM.SYSQUERY or SYSIBM.SYSQUERYPLAN

FREE QUERY

- **FREE QUERY filter-block**
  - Used to delete hints from DB2 Catalog tables
- **Filter-block**
  - QUERYID (value)
    - Deletes a specific hint based on the value of QUERYID column value
  - QUERYID(ALL)
    - Deletes all hints
  - FILTER('user filter')
    - where 'user filter' is the value of column USERFILTER in DSN_USERQUERY_TABLE and SYSIBM.SYSQUERY tables
  - PACKAGE(package-block)
    - See details on Command Reference Manual

Optimization Hints: Additional Considerations

- **Limitations**
  - Apparently, no way to know if & which Parameter Hint is being applied when binding a Package
    - Impact analysis becomes difficult
  - When a statement gets modified, the hint will not apply, unless specific actions taken
    - Update the SQL statement for the hint, bind query & package, if hint applied to a single instance
    - Define a new hint with the modified SQL statement text, bind query, if existing hint applied to multiple instances

Issues & Comments/Suggestions

- **BIND QUERY**
  - Issue
    - ✓ Hints are created or replaced for every row in DSN_USERQUERY_TABLE when you issue a BIND_QUERY command
  - IBM Suggestion
    - ✓ You might delete rows from DSN_USERQUERY_TABLE to prevent existing hints from being replaced when you issue subsequent BIND QUERY commands
  - Comment
    - ✓ *Why should we be concerned if an hint gets replaced?!?*

Governance: Where/Which Hints used?

- **Statement-level access path hints**
  - SYSPACKSTMT.ACCESSPATH = 'H'
  - SYSPACKAGE.OPTHINT = 'hintname'
- **Statement-level parameter hints**

PARAMETER	REFRENCED BY		
	TABLE	COLUMN	VALUES
REOPT(ALWAYS)	SYSPACKSTMT	STATUS	B, F, G*, J
REOPT(<>ALWAYS)	?	?	?
STARJOIN	?	?	?
PARAMDEG	?	?	?
SJTABLES	?	?	?

Notes: 1. STATUS also set when REOPT(**ALWAYS**) specified at BIND
2. It is set only for REOPT(ALWAYS)

Governance: Where/Which Hints used?

- **REOPT option**
  - ✓ For Static SQL & REOPT(ALWAYS) Hints, check STATUS = 'G' in SYSPACKSTMT
    - ✓ Would be STATUS = 'B' or 'F' or 'J' in some case (e.g. VALIDATE(RUN) and statement not validated at Bind)
  - ✓ But STATUS = 'G' also for stmts of Packages bound with option REOPT(ALWAYS) specified
  - ✓ No information if REOPT(ONCE), REOPT(AUTO)
  - ✓ If Package bound with REOPT(ALWAYS) and hint specifies a different value <> ALWAYS for REOPT option for some specific statement(s), then those statement(s) will have a STATUS NOT IN ('B', 'F', 'G', 'J')
  - ✓ Apparently NO WAY to know which specific hint (by unique identifier QUERYID) is applied to the statement
- ✓ **Other options (STARJOIN, etc.)**
  - ✓ Apparently, non information whatsoever

Governance & Change Management

- **Query modification(s)**
  - What happens if/when the text of a query referenced by a hint gets modified by application maintenance?
    - ✓ At BIND, the hint won't be applied anymore …
    - ✓ … unless the modified statement text is already bound in the Query Repository or a new hint is created and implemented
    - ✓ But how do you know which hint you should modify or just free?
  - Comment
    - ✓ This is a DB design issue… (already pointed out, implicitely)

Implemented Approach – 1

- **Created a «System» Hints table**
  - Identifies relevant queries by their coordinates
    - (COLLECTION), PACKAGE, STMTNO, STATUS
    - COLLECTION name assumed, if not supplied
    - VERSION = most recent only
    - REOPT(ALWAYS) only (currently)
- **Developed a REXX Script to**
  - Populate a «System» copy of DSN_USERQUEY_TABLE
    - Used as a temporary repository of hints to be handled
  - Issue BIND QUERY
  - Build and RUN REBIND of relevant Package(s)
  - Archives DSN_USERQUERY_TABLE enriched info
  - Checks SYSPACKSTMT & Print report / diagnostics

Implemented Approach – 2

- **Additional implementations**
  - REXX Script to "reset" hint(s)
    - Issue FREE QUERY with proper identifier
    - Build and RUN REBIND of relevant Package(s)
  - Some additional queries for reporting on hint(s) application to additional Package(s) as a consequence of weekly mass Rebind
- **NOTE:**
  - Everything done so far only applied to Static SQL

Concluding remarks

- No single solution, external from SQL code, can support all situations
- New approach and new hints valuable…
- But … implementation cumbersome
- Governance difficult, when not impossible
- Documentation not always clear & complete
- But … yes, new hints work and can be used
- We look forward and hope there will be future enahncements