Performance Enhancements for DB2 UDB for z/OS Version 8

Part Three
Tom Moulder
December 2, 2004
Topics for Today

- Star Join Enhancements
- REOPT (Once)
- Drop Global Temporary Tables
- Lock Avoidance
- Non-Correlated “Exists” Subquery
- “In List” Changes
- Faster “Short” Prepares
- Plan_Table Changes
- Visual Explain Changes
Star Join Overview

- Components
  - Fact Table
  - Dimensions
- Dimension Tables Highly Normalized
- Built Around a Central Fact Table
Application Characteristics

• Fact and Dimension Tables Joined to Decode a Fact Table Code
  – These are equi-join predicates

• Local Predicates on Dimension Tables
  – Selectivity is applied to the Dimension, not Fact

• Usually Large Number of Tables Joined

• No Join Predicates Cross Dimensions
DB2 Sample Schema

**Region (Dimension)**
1000 Rows

**Product (Dimension)**
60,000 Rows

**Time (Dimension)**
39 Rows

**Sales (Fact)**
100 Billion Rows

```
SELECT *
FROM SALES S, TIME T,
REGION L, PRODUCT P
WHERE S.TIME = T.ID
AND S.REGION = L.ID
AND S.PRODUCT = P.ID
AND T.YEAR = 2002
AND T.QTR = 1
AND L.CITY IN ('Boston','Seattle')
AND P.ITEM = 'stereo';
```
Access Path Objectives

• Matching Index Scan of Fact Table
  – Using as many Join Predicates as Possible

• Access Dimensions First
  – Further Reduce Fact Table Rows Accessed

• Selective Join Dimensions
  – Further Reduce Fact Table Rows Accessed
  – No Join Predicates = Cartesian Product
  – Therefore Used Selectively based on Cardinality
Performance Challenges

• Join of Dimensions
  – Called Snowflakes – Cartesian Product

• Sort of Dimensions
  – Tablespace Scan as Access Method Primarily
  – Inside-Out Processing use Sparse Index (V7)

• V8 Changes
  – In Memory Sorts Based on Size and Memory
  – Outside-In Processing use Sparse Index
Processing Phases

Outside-In Processing

- Dimension Time: 39 Rows
- Dimension Region: 1000 Rows
- Dimension Product: 60,000 Rows
- Fact Table Index Column Order
- Scans That Impact Performance
- Use Sparse Index
- NLJ

Inside-Out Processing

- Dimension: 100B Rows
- NLJ
- Result Set
- In-Memory Workfiles can Convert to NLJ

Scans That Impact Performance Now Use Sparse Index

5% Filtering Yields 5B Row Workfile

Use Sparse Index

Sales 100B Rows

Workfile
Some Details

• In-Memory Workfiles
  – Above the Bar (2GB Line)
  – Binary Search, Not Scan or B-Tree Index
  – I/O Avoidance to Workfile
  – Sensitive to Real Storage (More is Better)
  – Paging would be a Concern
  – SJMXPOOL Sets Maximum Size up to 1GB
  – STARJOIN Turns on New Features
Some Details

• Sparse Indexes
  – In-Memory Index for Equal Join Predicates
  – Primarily for Join of Large Composite Table
  – Used When Workfile is too Large for In-Memory
  – Makes Nested Loop Join Possible
  – Avoids Merge Scan Join Sort of Fact Table Data
Controlled Materialization

- **Optimizer Changes for Star Join**
  - Workfile Avoidance when Cost is Excessive
  - Uses Index on Dimension Table
  - Optimizer now evaluates more options

- **Back to Correlation Statistics**
  - Important for more Accurate Filter Factors
  - Helps Optimizer make Better Choices
  - Good Idea for Dimension Tables
REOPT Changes

• V7 Options
  – REOPT (VARS)
  – NOREOPT (VARS)

• V8 Options
  – REOPT (NONE)
  – REOPT (ALWAYS)
  – REOPT (ONCE)
REOPT Changes

• V7 Options still work in V8
  – REOPT(VARS) = REOPT(ALWAYS)
  – NOREOPT (VARS) = REOPT (NONE)

• REOPT (ONCE)
  – Access Path Chosen at First Execution
  – Dynamic Statement Caching Enabled
  – Potentially Better Access Path because of Host Variables
New DTT Option

- Declared Global Temporary Tables
  - V7
  - Required an Explicit Drop for Remote Connections so the Thread would go inactive and return to Pool
  - V8
  - New Option on the Declare
  - “Drop Table on Commit”
  - Implicit Drop if no Cursors with Hold are Open
  - Allows Threads to go Inactive More Easily
Lock Avoidance

• Applies to Indirect References in Data
  – Variable Length Row Updates May Cause This
    • Won’t fit because of Expansion
    • Pointer Replaces the Original Row
    • Row Placed in Another Page
  – V7 Locks Both Data Pages
  – V8 Locks only the Original Data Page

• Online Schema Change is Affected by This
Existence Check Changes

• Before V8
  – Existence Check Materialized ALL Rows for Select

• V8
  – Existence Check Stops after ONE Row is Materialized

```sql
SELECT EMPNO, LASTNAME
FROM DSN8810.EMP
WHERE EXISTS
  (SELECT * FROM DSN8810.PROJ
   WHERE PRSTDATE > '2005-01-01');
```
In List Enhancements

• V7 Enhancements Through APARS
  – Predicate Pushdown with PQ73454
  – Correlated Subquery Transformation with PQ73749

• V8 Changes
  – INLISTP DSNZPARM Default is now 50
  – Enhancements now Enabled by Default
  – No Action Required to get the Benefit
“Short” Prepare Changes

• “Short” Prepare?
  – When Dynamic Statement is Found in Cache

• Before V8
  – Copy into Local Storage for the Thread
  – Getmain/Freemain Largest Portion of Process

• V8
  – System-Wide Pools of Storage for Statements
  – Copy to a System Pool using Best-Fit Algorithm
New Columns for Plan_Table

• Table_Encode CHAR(1)
  – Encoding Scheme of Table
    • E = EBCDIC
    • A = ASCII
    • U = UNICODE
    • M = Mixed
New Columns for Plan_Table

- **Table_SCCSID** Fixed (16)
  - SBCS Identifier for Table
  - Zero, if Table_Encode = M
- **Table_MCCSID** Fixed (16)
  - Mixed CCSID for Table
  - Zero, if Table_Encode = M
- **Table_DCCSID** Fixed (16)
  - DBCS Identifier for Table
  - Zero, if Table_Encode = M
New Columns for Plan_Table

• ROUTINE_ID Integer
  – Points to the Table Function Record in SYSIBM.SYSROUTINES

• TNAME
  – Now may contain the Name of an MQT

• TABLE_TYPE
  – M = Materialized Query Table in TNAME
Alias Now Allowed

- All the Explain Tables can be an Alias
- Alias Creation must use the “Owner.Table_Name” format
- Allows a Common set of Tables for Explains
- Fewer Objects to Manage
- Regular Maintenance Required
- Manage Extents and Space based on Usage
Visual Explain Re-Write

- Java Client using DB2 Connect to z/OS
- All Capabilities of V7 with new UI
- Still Free
- Management Client Package Component
- Download from the Web
New Inputs

• PLAN_TABLE
• DSN_STATEMNT_TABLE
• DSN_FUNCTION_TABLE
• Catalog Statistics
• Additional Optimizer Tables
  – These are not Externalized
  – Not Documented
New Data in Reports

- Single Predicate Filter Factor Estimates
- Identify Predicates as Stage 1 or Stage 2
- Row Estimates at all Stages of Execution
- When Predicate is Applied
- Partition List if Partial Scanning
- Index Filter Factor Estimates
- Details on Parallel Execution
New Data in Reports

• Sort Information
  – Key Length
  – Data Length
  – Estimated Row Counts
New Output

- Reports
  - In HTML Format
  - In XML Format
- Share With Others
- IBM Support
  - Service SQL Option
  - One-Click generation
  - Everything IBM needs in a Single Package
Ease of Use

• Necessary Tables
  – Easily Created Through the UI
  – Just Specify Database and Tablespace Names
  – Delete Rows Within the UI
  – Select Rows from the Tables for Viewing
Graphical Display

1. Query
2. QB1 30029.727
3. Fetch 30029.727
4. SORTRID 30030
5. IXSCAN 30030
6. SNI#PKS#KQEP#DSON 179998372
7. LINEITEM 179998372

Table access (# fetches)
Rid sort (# rids)
Index scan (qualified rows)

Index (FULLKEYCARDF)
Table (CARDF)
Click on IXSCAN

- Displays
  - Input RIDS (going into the Scan)
  - Leaf Page Count
  - Filter Factor
    - Matching Predicates
    - Screening Predicates
    - Total Filter Factor as a Result of Predicates
  - Scanned Leaf Pages
  - Output RIDS
  - Matching Columns
Service SQL

- Uses Statement or Plan_Table as Input
- Collects and Sends to IBM Support
  - SQL Statement
  - Object DDL
  - Catalog Statistics
  - ZPARMS (if DSNWZP SP is Available)
  - Environment Specifics
    - CPU Speed
    - Pools Sizes (Buffer, Sort, RID)
    - Number of Processors
Summary

- Wow! A Lot of Information to Digest
- Points to Remember
  - NCCR – No Coding Changes Required
  - Significant Performance Improvements
  - Many changes are easy to implement in V8
- Questions Anyone
Next Steps with DB2 V8 and Lightyear

- Our series of detailed presentations on various DB2 V8 topics:

<table>
<thead>
<tr>
<th>Pre-requisites</th>
<th>V7 to V8 Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicode</td>
<td>Utilities</td>
</tr>
<tr>
<td>Access Path Review</td>
<td>Highlights of New Functionality</td>
</tr>
<tr>
<td>Schema Evolution</td>
<td>In Depth Review of the “Top” 5 New Functions</td>
</tr>
<tr>
<td>Catalog Changes</td>
<td>Enhancements to SQL</td>
</tr>
</tbody>
</table>
Next Steps with DB2 V8 and Lightyear

- FREE presentations available NOW.
- These presentations will be given to just one customer at a time and will be tailored to that customer.
- Presentations will vary in length, dependent upon topic and tailoring.
- A 15 minute “prep” call will be required.
Next Steps with DB2 V8 and Lightyear

• Access Path Review is a performance factor
• We offer a free on-site two-day review
  – Your systems
  – Your SQL
  – Predictive Report of potential problem SQL

• Watch our website for details on both of these exciting and FREE offers and register your interest.
Next Steps with DB2 V8 and Lightyear

• Migration Workshop
• Created for Systems Staff Prior to Install
• One Day Joint Venture With IBM and Lightyear
  – Dec. 14 – Madison, Wisconsin
  – Dec. 15 – Schaumburg, Illinois
  – Dec. 16 – San Francisco, California
  – Dec. 17 – San Ramon, California
• Go to our website for details on these FREE offers and register.
Services available from:

Palo Alto – Fort Worth - Calgary - Laguna Beach
Scottsdale – Chicago – St. Louis – Boston

www.lightyr.com

- Database migration to DB2
- VS/2 and DL/2 software sales and related services (sole North American distributor)
- CICS, DB2, IMS, & z/OS software and tools, sales and upgrades
- Customized on-site technical seminars & education classes
- WebSphere MQ and application integration services
- CICS & IMS Web enabling design and implementation
- Database and online system performance analysis and tuning
Questions …