



David Skinner
Software Developer
March 2013



Product Implementation Training (PIT)

IBM FileNet Content Manager 5.2 CBR Query Optimization

Introduction

- Course Overview
 - Current CBR Search Design and Limitations
 - CBR Query Optimization and Configuration
 - Best Practices, Troubleshooting and Documentation
- Target Audience:
 - Customer Support and Field Technical Resources
- Prerequisites:
 - Familiarity with Content Platform Engine, CBR and CSS
- Release Date March 2013

© Copyright International Business Machines Corporation 2013. All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.


Course Objectives



After this course you will be able to:

- Understand CBR Query Optimization and Dynamic Switching
- Enable Dynamic Switching and set the Dynamic Threshold
- Know the Limitations, Best Practices and Troubleshoot

Course Roadmap

- 
- ➔ Current Behavior
 - Use Case
 - Demonstration
 - Current Design
 - CBR Query Optimization
 - Configuration
 - Search SQL Options
 - Limitations
 - Demonstration
 - Best Practices and Troubleshooting
 - Documentation

Use Case

- Many customers have complained of slow CBR searches in these circumstances
 - There were many full text search hits
 - There were few relational database hits
 - *“Why does CBR take so long to find a handful of hits?”*

Demonstration – Current Behavior

- *Test System*

- 365,000 content indexed documents
 - Each document's content includes word 'copyright'
- DocTitle property is indexed in the database
 - Only one DocTitle starts with 'Test'

- Search for one 'Test' document:

*Select D.This FROM Document D INNER JOIN ContentSearch CS
ON D.This = CS.QueriedObject
WHERE D.DocumentTitle LIKE 'Test%'
AND CONTAINS (*, 'copyright')*

- *Go to demo...*

CBR Search – Current Design

- *Current Design*

- 1) The full text portion of the search is issued to retrieve a batch
- 2) The batch of full text hits are inserted into a DB Temp table
- 3) A join query is issued between the Temp and DocVersion tables
- 4) Steps 1-3) are repeated to fill the client page

- *Current Limitation*

- Can only perform the full text portion of the search first
 - Works well if there are not a large number of full text hits
 - Work poorly if there are many full text hits and many iterations of the join query are needed to fill a client page request

Select ... FROM Document D INNER JOIN ContentSearch CS ...

WHERE D.DocumentTitle LIKE 'Test%' AND CONTAINS (, 'copyright')*

Course Roadmap

- Current Behavior
 - Use Case
 - Demonstration
 - Current Design
- ➔ CBR Query Optimization
 - Configuration
 - Search SQL Options
 - Limitations
- Demonstration
- Best Practices and Troubleshooting
- Documentation

CBR Query Optimization

- *New Design:*
 - Object Stores can be configured to “Dynamically Switch”
 - For every CBR Search, an estimate is first made of the full text hit count based on the full text search criteria
 - If the estimate is larger than a configured threshold, the relational database portion of the search is executed first (DB First)
 - Otherwise the full text search is issued first (Content First)
 - The DB hits are then searched against CSS as a “scalar query”
 - Does not work with Legacy Content Search Engine
- *Benefits:*
 - Allows much faster searches when there are not a lot of relational DB hits: All the hits can be found quickly in the database.

Configuration

- *Object Store Enablement*
 - Set the CBR Query Optimization property on each Object Store
 - Default value is Null (not set)
 - Searches are done Content First
 - Set to value of 1
 - Searches will Dynamically Switch
 - Set CBR Query Dynamic Threshold property
 - For example: 10000
 - Only used if CBR Query Optimization property is set to 1

Search SQL Options

- *Explicit Search Enablement*

- Set CE SQL Option
 - OPTIONS (CBR_CONTENT_FIRST)
 - OPTIONS (CBR_DB_FIRST)
 - OPTIONS (CBR_DYNAMIC_THRESHOLD 10000)
- Overrides Object Store Dynamic Switching
- Example:

*Select ... FROM Document D INNER JOIN ContentSearch CS ...
WHERE D.DocumentTitle LIKE 'Test%' AND CONTAINS (*, 'copyright')
OPTIONS (CBR_DB_FIRST)*

Limitations

- *Searches cannot execute DB First when:*
 - Using “Order By Rank”, unless...
 - Object Store property: CBR Query Rank Override
 - Disabled (default): Order By Rank is obeyed
 - » Search is always performed Content First
 - Enabled: Order By Rank is ignored for DB First
 - » Allows dynamic switching
 - Required: Results are never returned in rank order
 - » Setting applies with or w/o Dynamic Switching
 - Using a merged scope search:
 - CBR Query Rank Override property must be set to Required on one or more object stores

Limitations (cont'd)

- *Searches cannot execute DB First when:*
 - Using a Property condition OR'd with Contains
 - Using content property conditions
 - Example: WHERE rank > 38
- *Disallowed searches will:*
 - Execute Content First when configured dynamically on object store
 - Throw Error when DB First / Dynamic specified in Search Option

Limitations (cont'd)

- *Additional Limitations*

- CBR Count Estimation time
 - Multiple collections may take longer depending on hits
 - Better with common words
- Count Estimation Accuracy
 - Better with large collections (> 200K)
 - Better with common words
- Relational search needs
 - Good indexed property conditions
 - Not a large relational result set

Summary

- Dynamic Switching is off by default
- *Enable when...*
 - End-users are searching common words like “wheel” or “nuclear”
 - CBR hit count is in the tens of thousands
 - Relational conditions restrict the number of database hits
 - At least one search property is indexed in the database
 - Search does not require rank order

Course Roadmap



- Current Behavior
 - Use Case
 - Demonstration
 - Current Design
- CBR Query Optimization
 - Configuration
 - Search SQL Options
 - Limitations
- ➔ Demonstration
- Best Practices and Troubleshooting
- Documentation

Best Practices

- DB first searches require a database index on at least one property in the WHERE clause
 - Specify limiting conditions to achieve relatively small hit counts
- Ensure that the dynamic threshold is large enough
 - If the CBR hits are spread across many collections, this requires an increasingly larger threshold
 - See P8 documentation for a table of minimum threshold
- Dynamic count estimation works best for searches on common words

Troubleshooting

- Dynamic switching search is still slow
 - Find out if the search was executed DB or Content First
 - Enable P8 Search Tracing with detail, CBR tracing
 - Monitor PCH Counters for CBR Queries Database First
 - Check the estimation time from P8 search tracing
 - Check that some property conditions are indexed (query plan)
- Dynamic switching did not choose DB First
 - Check the accuracy of the count estimate
 - Search words may have been uncommon
 - Check the number of collections and minimum threshold needed
 - Contact development for JVMs to control default behavior

Documentation



- SQL Syntax Reference - CBR Query Optimization
 - Internal Information Center (available now):
 - http://cmfogbert.usca.ibm.com:7777/p8ic520/topic/com.ibm.p8.ce.dev.ce.doc/query_sql_syntax_cbr.htm
 - External Information Center (available at eGA):
 - http://pic.dhe.ibm.com/infocenter/p8docs/v5r2m0/topic/com.ibm.p8.ce.dev.ce.doc/query_sql_syntax_cbr.htm