

DB2 for Linux, UNIX, and Windows Performance Tuning and Monitoring Workshop Information

Length: 4.0 Days
Ref: CL412G
Delivery method: Classroom
Price: EUR

Overview

Learn how to tune for optimum performance the IBM DB2 9 for Linux, UNIX, and Windows relational database management system and associated applications written for this environment. Learn about DB2 9 for Linux, UNIX, and Windows in a single partition database environment. Explore performance issues affecting the design of the database and applications using the database, the major database performance parameters, and the different tools that assist in performance monitoring and tuning.

Use tools that are common across the Linux, UNIX, and Windows environments. During labs running on DB2 9.7, develop your ability to use monitoring tools, Explain tools and DB2 utilities like RUNSTATS, REORG and db2batch to tune a database running on your local LINUX workstation.

Course Materials

The course materials cover DB2 9.7 for Linux, UNIX, and Windows.

Hands-On Labs

The twelve labs use the DB2 9.5 Linux, UNIX, and Windows tools and commands to monitor, configure and analyze performance. The labs are performed using one DB2 9.5 for Linux system per student.

Public

This is an advanced course for database designers, database administrators, and application developers working with DB2 for Linux, UNIX, and Windows who are concerned about performance.

This course is appropriate for those using DB2 in a z/Linux environment.

Prerequisites

You should complete:

- *DB2 9 Database Administration Workshop for Linux, UNIX, and Windows (CL2X2)* or
- *DB2 9 for Linux, UNIX, and Windows Quickstart for Experienced Relational DBAs (CL482)*

- **or** have equivalent experience.

Objective

- Define the impact of database design (tables, indexes, and data placement) on database performance
- Describe database application programming considerations and how they affect performance
- Identify and describe the parameters (database and non-database) that affect performance
- Tune parameters to achieve optimum performance
- Identify and use the tools that assist in monitoring and tuning of a database

Topics

Database Monitoring

- Describe the basic principles in monitoring a DB2 database
- List the tools for monitoring database and application activity
- Use GET SNAPSHOT commands to produce reports for analysis of database performance
- Utilize the administrative routines and views provided by to DB2 to simplify application access to database performance statistics
- Use the db2pd to perform performance analysis or problem determination for a DB2 database
- Utilize the enhanced table functions with SQL in reporting and monitoring of the database system, data objects, and the package cache to help you quickly identify issues that might be causing problems
- Configure the DB2 Database configuration options that control the collection of request, activity and object level metrics on the entire database
- Compare the enhanced table functions with the snapshot monitoring facilities provided by previous DB2 releases

Database Input/Output (I/O) Management

- Describe processing for reading database pages into buffer pools
- Describe processing for writing database pages from buffer pools
- Monitor database read and write activity using GET SNAPSHOT commands or Administrative Routines and Views
- Monitor database logging activity and select appropriate values for SOFTMAX and MINCOMMIT
- Describe how scan sharing can reduce the I/O workload for accessing large tables
- Explain the alternate page cleaning processing associated with th
DB2_USE_ALTERNATE_PAGE_CLEANING DB2 Registry variable

Tablespace and Table Design for Performance

- Select appropriate values for table space page size and extent size to support application

performance requirements

- Describe the calculation of prefetch size when PREFETCHSIZE is set to AUTOMATIC
- List the advantages of selecting DMS or SMS table space management as well as using Automatic Storage-managed table spaces
- Set file system caching options for table spaces to optimize table space performance
- Describe the various row insertion algorithms for tables that are based on using the APPEND option or defining a clustering index
- Plan and implement Row Compression to reduce disk and memory requirements and improve application performance
- Describe how DB2's Index compression option can reduce disk storage requirements for indexes
- Explain how DB2 can compress temporary data to improve application performance
- Utilize the DB2 provided tools and functions to estimate compression results for Indexes and data

DB2 memory management

- Describe memory heap usage for instance memory, database shared memory and application memory
- Explain the management of database shared memory based on setting the configuration option DATABASE_MEMORY to AUTOMATIC, COMPUTED or a specific number of pages
- Select the mode for managing data sort memory using SHEAPTHRES, SORTHEAP, and SHEAPTHRES_SHR
- Monitor DB2 memory usage using the db2mtrk commands and SQL statements
- Utilize the db2pd command for monitoring current database memory usage

Automated Memory Management

- Describe how STMM can be used to automatically manage database shared memory heaps
- Explain the differences in STMM processing based on the setting of DATABASE_MEMORY
- Plan and configure a database for self tuning memory
- Activate or deactivate STMM for selected memory heaps
- Describe the management of sortheap memory based on the configuration of sheapthres and sheapthres_shr
- Explain how DB2 can automatically increase or decrease database memory for multiple DB2 databases running on the same server

Application Performance Considerations

- List examples of application coding techniques that can effect performance
- Describe the performance advantages of using stored procedure
- Design applications to maximize record blocking, minimize sort operations and minimize lock contention
- Monitor application performance and lock waits using GET SNAPSHOT commands and SQL queries

- Set the DB2 registry variables DB2_EVALUNCOMMITTED, DB2_SKIPINSERTED and DB2_SKIPDELETED to reduce lock contention based on application requirements
- Utilize the currently committed form of locking to reduce lock contention between read-only and update applications

Using Explain Tools

- Describe the advantages of using Visual Explain
- Describe the advantages of using db2exfmt
- Create special tables used by Visual Explain and db2exfmt
- Identify how to set the Explain snapshot and Explain mode registers to capture the information of interest
- Differentiate between the different methods of viewing Explain information

The DB2 Optimizer

- Describe the stages of the SQL compiler
- Choose the appropriate optimization class
- Describe the catalog statistics and database configuration options impact on access plan selection
- Implement a statistical view to provide better cardinality estimates for complex queries
- Enable the statement concentrator using the STMT_CONC database manager configuration parameter to reduce SQL compilation overhead for dynamic SQL statements
- Utilize the db2look utility to extract catalog statistics from existing tables to mimic an existing database environment
- Create an optimizer profile

Using Indexes for Performance

- Describe the Indexing options that can be used to improve performance: Index Only Access, Clustered Index, Reverse Scans, Include Columns, and Index Freespace
- Describe the Block Indexing capability for MDC tables
- Monitor index usage using the MON_GET_INDEX function and db2pd commands
- Explain how multiple indexes can be combined using Index ORing and Dynamic Bitmap Index ANDing
- Use the Design Advisor to predict performance gains from adding new indexes

Complex SQL Performance

- Review Explain reports for costly sort operations
- Describe the differences between Nested Loop, Merge Scan and Hash Joins
- Plan the implementation of Refresh Immediate or Refresh Deferred Materialized Query Tables to improve query performance

- Utilize the Design Advisor to analyze SQL statements and recommend new MQTs
- Describe the features of range-partitioned tables to support large DB2 tables using multiple table spaces, including the roll-in and roll-out of data ranges
- Explain the difference between partitioned and non-partitioned indexes for a range-partitioned table
- Implement partitioned indexes to improve performance when you roll data out or roll data into a range-partitioned table
- Use the DB2 Explain tools to determine if partition elimination is being used to improve access performance to large range-partitioned tables

Tools and Utilities for Performance

- Use the RUNSTATS utility to collect table and index statistics to enable the DB2 Optimizer to generate efficient access strategies
- Select appropriate RUNSTATS options to collect Distribution Statistics or Column Group Statistics to improve cardinality estimates during SQL compilation
- Use the table and indexes statistics to plan for table and index reorganization using the REORG utility
- Set the policies and options for automation of catalog statistics collection
- Monitor the activity associated with implementing real-time statistics collection
- Use the db2batch utility to run SQL workloads and collect performance statistics that can be used to benchmark database and application changes

Event Monitoring

- Create Event Monitors to collect performance statistics at the database, application or SQL statement level
- Implement simple Workload Management definitions to utilize the Workload Manager-based event monitoring including activities, statistics and threshold violations
- Evaluate Event Monitor data using the Event Analyzer tool, the db2evmon text-based tool the db2evmonfmt application or using SQL queries
- Define a LOCKING Event Monitor to capture deadlocks, lock timeout or lock waits
- Configure a DB2 database to control information captured for deadlocks, lock timeouts or extended lock waits
- Implement Event Monitors for units of work or lock-related events that store information in unformatted Event Monitor tables
- Capture SQL section information using an ACTIVITIES Event Monitor and use the data to generate Explain reports

Agenda

Day 1

- Welcome
- Unit 1: Database monitoring
- Exercise 1: Database Monitoring
- Unit 2: Database I/O Management
- Unit 3: Table Space and Table Design for Performance
- Exercise 2: I/O Management and Table Space Design

Day 2

- Unit 4: DB2 Memory Management
- Exercise 3: Memory Management
- Exercise 4: Section 1 - Setup for STMM lab
- Unit 5: Automatic Memory Management
- Exercise 4: Sections 2, 3 - Using STMM to Manage Memory
- Unit 6: DB2 Application Performance Considerations
- Exercise 5: Application Performance

Day 3

- Unit 7: Using Explain Tools
- Exercise 6: Using Explain Tools
- Unit 8: The DB2 Optimizer
- Exercise 7: DB2 Optimizer
- Unit 9: Using Indexes for Performance
- Exercise 8: Index Performance

Day 4

- Unit 10: Complex SQL Performance
- Exercise 9: Complex SQL Performance
- Unit 11: Tools and Utilities for Performance
- Exercise 10: Tools and Utilities
- Unit 12: Event Monitoring
- Exercise 11: Event Monitors

□