IBM DB2 LUW Performance Tuning and Monitoring for Single and Multiple Partition DBs

Duration: 5 Days  Course Code: CL442G

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 support of single partition and multiple partition (DPF) database environments. 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 9.7 Linux, UNIX, and Windows tools and commands to monitor, configure and analyze performance. The labs are performed using one DB2 9.7 for Linux system per student.

Target Audience:

This is an advanced course for database designers, database administrators, and application developers working with DB2 9 for Linux, UNIX, and Windows who are concerned about performance in both single and multiple partition databases.

Objectives:

- 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 for Online Transaction processing (OLTP) or Data Warehouse environments
- Identify and use the tools that assist in monitoring and tuning of single partition and multiple partition (DPF) databases
- Analyze Explain reports to identify the access strategies selected by the DB2 Optimizer for execution of SQL statements including the selection of indexes, join techniques, sorts and table queues.

Prerequisites:

You should complete:

- DB2 9 for LUW Multiple Partition DBA Workshop (CF24) or
- DB2 9 for LUW Multiple Partition Environment for Single Partition DBAs (CG24) or have equivalent experience
## Content:

### 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 relational monitoring interfaces 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 monitoring features with the snapshot monitoring facilities provided by previous DB2 releases

### Automatic memory management
- Describe how Self-Tuning Memory Manager (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 sort heap 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

### 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 tablespaces, 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

### 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
- 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
- Implement and monitor use of a block based buffer pool to improve table scan performance
- Describe the alternate page cleaning processing associated with the DB2 Registry variable DB2_USE_ALTERNATE_PAGE_CLEANING
- Describe how scan sharing can reduce the I/O workload for accessing large tables
- 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

### Application Performance Considerations
- List examples of application coding techniques that can effect performance
- Describe the performance advantages of using stored procedures
- Design applications to maximize record blocking, minimize sort operations and minimize lock contention
- Evaluate setting the CUR_COMMIT database configuration option to reduce lock waits, lock timeouts or deadlocks
- Explain how DB2 can access the committed version of a data row that is currently being updated by another
<table>
<thead>
<tr>
<th>Set file system caching options for table spaces to optimize table space performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe the various row insertion algorithms for tables that are based on using the APPEND option or defining a clustering index</td>
</tr>
<tr>
<td>Plan and implement Row Compression to reduce disk and memory requirements and improve application performance</td>
</tr>
<tr>
<td>Describe how DB2's Index compression option can reduce disk storage requirements for indexes</td>
</tr>
<tr>
<td>Explain how DB2 can compress temporary data to improve application performance</td>
</tr>
<tr>
<td>Utilize the DB2 provided tools and functions to estimate compression results for Indexes and XML data</td>
</tr>
</tbody>
</table>

**Database memory management**

- Describe memory heap usage for instance memory, database shared memory, and agent private 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 command or the graphical application Memory Visualizer.
- Utilize the db2pd for monitoring current database memory usage
- Use AUTOCONFIGURE to set database configuration defaults when a new database is created or after a database workload changes

**Further Information:**

For More information, or to book your course, please call us on Head Office 01189 123456 / Northern Office 0113 242 5931

info@globalknowledge.co.uk
www.globalknowledge.co.uk

Global Knowledge, Mulberry Business Park, Fishponds Road, Wokingham Berkshire RG41 2GY UK