

Chapter 12: Monitoring and Tuning

IBM DB2 Universal Database V8.1

Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

Objectives

- In this section, we will cover:
 - ▶ DB2 Architecture Review
 - ▶ Database Manager Configuration Parameters Tuning
 - ▶ Database Configuration Parameters Tuning
 - ▶ Query Parallelism
 - ▶ Use of the EXPLAIN Facility
 - ▶ Use of the SNAPSHOT Monitors
 - ▶ Use of the EVENT Monitors
 - ▶ Use of the Health Monitor and Health Center
 - ▶ Understand performance issues and diagnostics

Chapter 12: Monitoring and Tuning

DB2 Architecture Review

Database Manager and Database Configuration Parameters

Query Parallelism

EXPLAIN Facility

SNAPSHOT Monitors

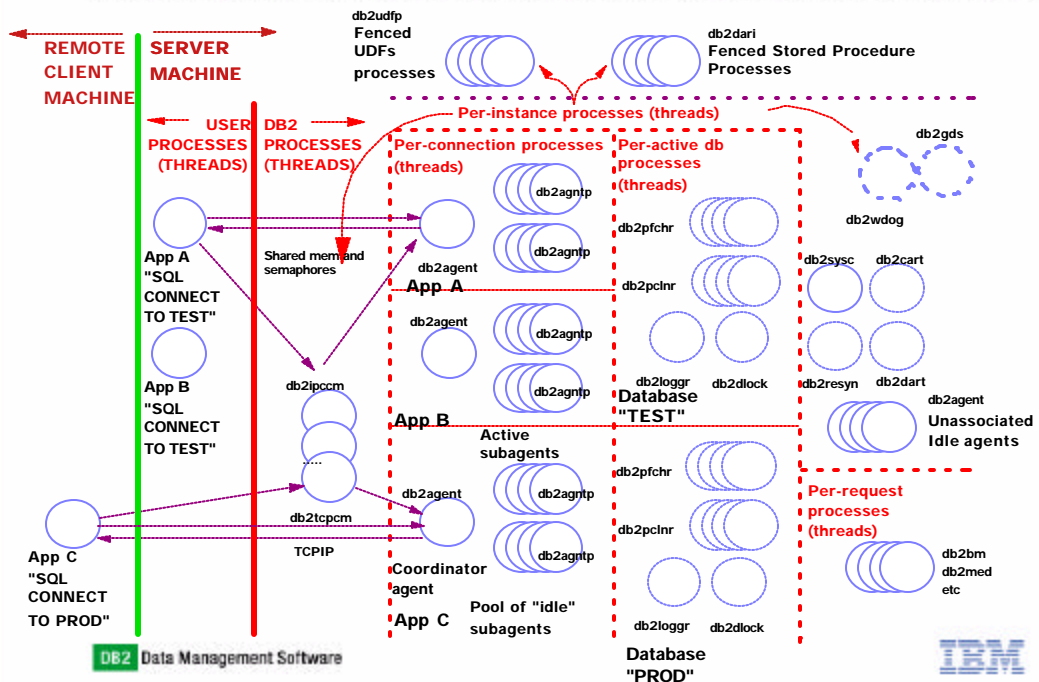
EVENT Monitors

Health Monitors and Health Center

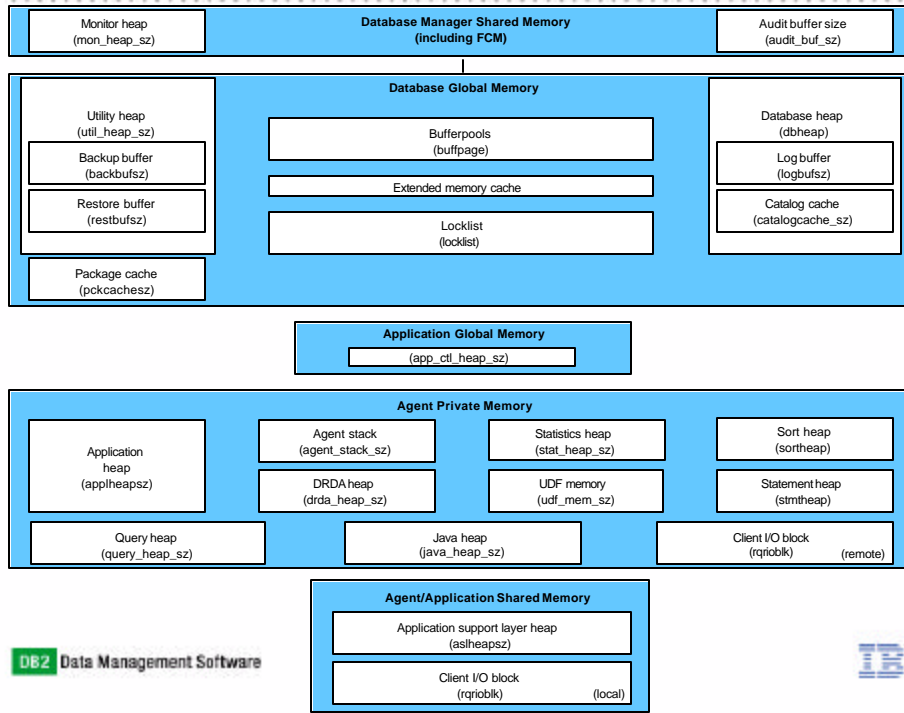
DB2 Diagnostics Logs

IBM Software Group

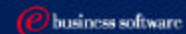
DB2 Process Model



DB2 Memory Model



DB2 Data Management Software



Chapter 12: Monitoring and Tuning

DB2 Architecture Review

Database Manager and Database Configuration Parameters

Query Parallelism

EXPLAIN Facility

SNAPSHOT Monitors

EVENT Monitors

Health Monitors and Health Center

DB2 Diagnostics Logs

IBM Software Group

Database Manager Configuration Parameters

- Database Manager Configuration Parameters are established when the instance is created
- Most of them affect the amount of system resources that will be allocated to a single instance of the database manager
- To display the database manager configuration parameters for an instance using the Command Line:
 - ▶ GET DATABASE MANAGER CONFIGURATION
 - ▶ GET DBM CFG

```

Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level          = 0x0a00

Maximum total of files open                          (MAXTOTFILOP) = 16000
CPU speed (millisec/instruction)                    (CPUSPEED)   = 2.078315e-006
Communications bandwidth (MB/sec)                   (COMM_BANDWIDTH) = 1.000000e+000

Max number of concurrently active databases          (NUMDB) = 8
Data Links support                                  (DATA LINKS) = NO
Federated Database System Support                   (FEDERATED) = YES
Transaction processor monitor name                  (TP_MON_NAME) =
. . . . .
    
```

Display DBM Configuration Parameters

- Use the SHOW DETAIL clause to list the current and defer values
- The SHOW DETAIL clause requires an explicit instance attachment
- Example:
 - ▶ ATTACH TO db2
 - ▶ GET DBM CFG SHOW DETAIL
 - ▶ DETACH

```

Database Manager Configuration

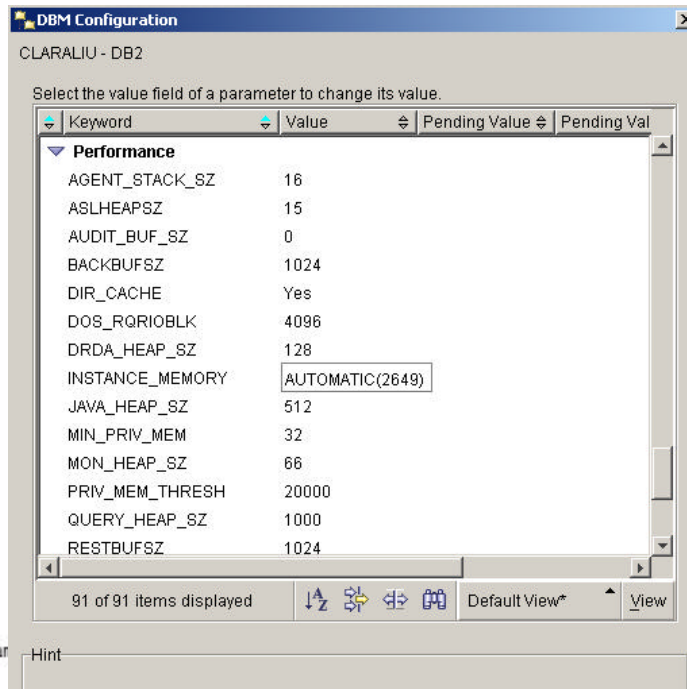
Node type = Enterprise Server Edition with local and remote clients

Description                                     Parameter      Current Value   Delayed Value
-----
Database manager configuration release level          = 0x0a00

Maximum total of files open                          (MAXTOTFILOP) = 16000           16000
CPU speed (millisec/instruction)                    (CPUSPEED)   = 2.078315e-006   2.078315e-006
Communications bandwidth (MB/sec)                   (COMM_BANDWIDTH) = 1.000000e+000   1.000000e+000

Max number of concurrently active databases          (NUMDB) = 8           8
Data Links support                                  (DATA LINKS) = NO           NO
Federated Database System Support                   (FEDERATED) = YES           YES
Transaction processor monitor name                  (TP_MON_NAME) =
. . . . .
Diagnostic error capture level                      (DIAGLEVEL) = 3           4
Notify Level                                        (NOTIFYLEVEL) = 3           4
    
```

Display DBM Configuration Using the Control Center



DB2 Data Mar

Hint



Update DBM Configuration Parameters

- Use the IMMEDIATE option (default behavior) to update DBM configuration parameters and make the changes immediately while the instance is running
- For some DBM configuration parameters, the database manager must be stopped (db2stop) and then restarted (db2start) for the new parameter values to take effect
- Other parameters can be changed online, they are called configurable online configuration parameters
- To change a DBM parameter online, explicitly attach to the instance is required, otherwise changes are deferred
 - ▶ ATTACH TO db2inst1
 - ▶ UPDATE DBM CFG USING maxagents 500 max_querydegree 3 IMMEDIATE
 - ▶ DETACH
- Update DBM configuration parameters and defer the changes to take effect when the instance is restarted
 - ▶ UPDATE DBM CFG USING maxagents 500 max_querydegree 3 DEFERRED

DB2 Data Management Software



Database Configuration Parameters

- The Database Configuration Parameters limit the system resources that a database can use
- To display the database configuration parameters using the Command Line:
 - ▶ GET DATABASE CONFIGURATION FOR <dbname>
 - ▶ GET DB CFG FOR <dbname>

```
Database Configuration for Database sample

Database configuration release level          = 0x0a00
Database release level                      = 0x0a00

Database territory                          = US
Database code page                          = 1252
Database code set                           = IBM-1252
Database country/region code                = 1

Dynamic SQL Query management                (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database          (DISCOVER_DB) = ENABLE

Default query optimization class            (DFT_QUERYOPT) = 5
Degree of parallelism                       (DFT_DEGREE) = 1
Continue upon arithmetic exceptions         (DFT_SQLMATHWARN) = NO
Default refresh age                         (DFT_REFRESH_AGE) = 0
Number of frequent values retained         (NUM_FREQVALUES) = 10
Number of quantiles retained               (NUM_QUANTILES) = 20
```

DB2 Data Management Software



Display Database Configuration Parameters

- Use the SHOW DETAIL clause to list the current and defer values
- The SHOW DETAIL clause requires a connection to the database
- Example:
 - ▶ CONNECT TO sample
 - ▶ GET DB CFG FOR sample SHOW DETAIL

```
Database Configuration for Database sample

Description                                Parameter    Current Value    Delayed Value
-----
Database configuration release level          = 0x0a00
Database release level                      = 0x0a00

Database territory                          = US
Database code page                          = 1252
Database code set                           = IBM-1252
Database country/region code                = 1

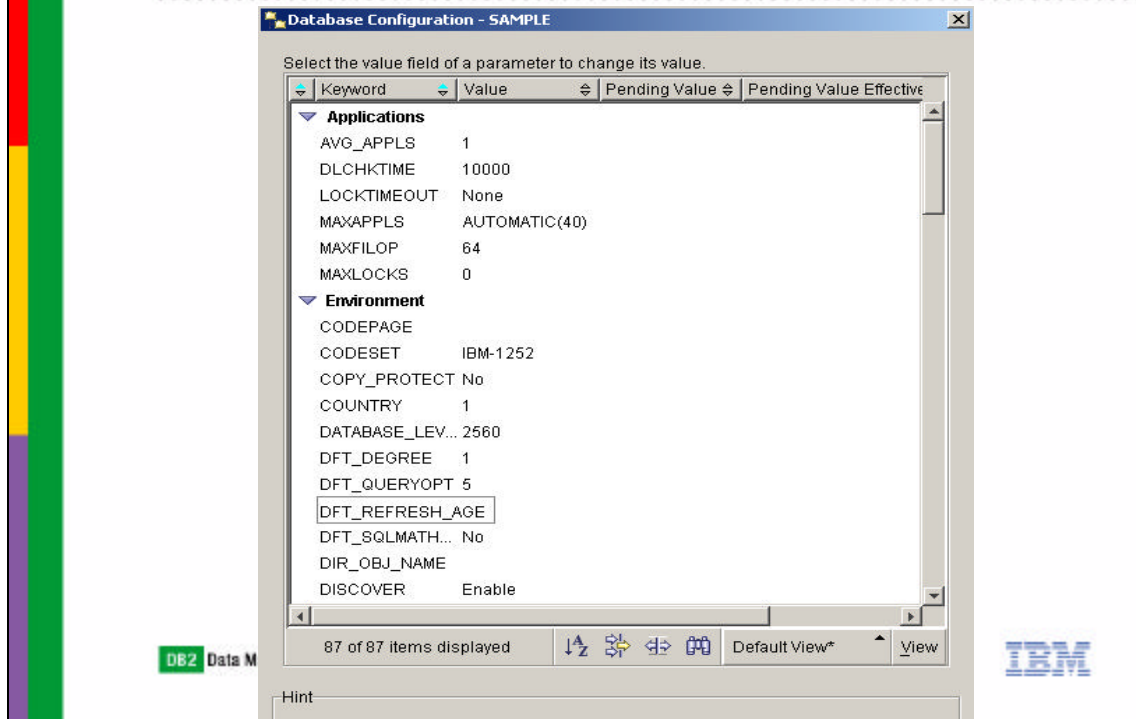
Dynamic SQL Query management                (DYN_QUERY_MGMT) = DISABLE    DISABLE
Discovery support for this database          (DISCOVER_DB) = ENABLE    ENABLE

Default query optimization class            (DFT_QUERYOPT) = 5          3
Degree of parallelism                       (DFT_DEGREE) = 1          3
```

DB2 Data Management Software



Display DB Configuration Using the Control Center



Update Database Configuration Parameters

- Use the IMMEDIATE option (default behavior) to update DB configuration parameters and make the changes immediately while other applications are still connected to the database
- For some DB configuration parameters, all applications must disconnect from the database for the new parameter values to take effect
- Other parameters can be changed online, they are called configurable online configuration parameters
- To change a DB parameter online, explicitly connect to the database
 - ▶ CONNECT TO sample
 - ▶ UPDATE DB CFG FOR sample USING dbheap 800 buffpage 1000 IMMEDIATE
- Update DB configuration parameters and defer the changes to take effect when the first user connects to the database or at database activation time
 - ▶ UPDATE DBM CFG USING dbheap 800 buffpage 1000 DEFERRED

Set DBM and DB Configuration Parameters to AUTOMATIC

- Some DBM and DB configuration parameters supports the value AUTOMATIC
- If set to automatic, DB2 will automatically adjust the parameter to reflect current resource requirements
- DBM configuration parameter that supports AUTOMATIC:
 - ▶ instance_memory - Amount of memory reserved for the instance
- DB configuration parameter that supports AUTOMATIC:
 - ▶ database_memory - Size of Database Shared Memory
 - ▶ maxappls - Maximum Number of Active Applications

Tuning Database Manager Configuration Parameters

- NUMDB
 - ▶ It limits the maximum number of concurrently active databases
 - ▶ Value should reflect reality as closely as possible
 - ▶ Each database takes up storage and an active database uses a new shared memory segment, system resource can be reduced by limiting the number of separate databases on a server
- MAXAGENTS
 - ▶ Maximum number of database manager agents available at any given time to accept application requests. This value limits the memory usage of the instance, because each additional agent requires additional memory
- NUM_POOLAGENTS
 - ▶ A guideline for how large the agent pool should grow. If more agents are created than this value, they will be destroyed when execution is completed
- SHEAPTHRES
 - ▶ Controls the total amount of memory for sorting available across the entire instance for all sorts
 - ▶ Should be roughly 10x SORTHEAP (DB CFG)

Tuning Database Configuration Parameters

- MAXAPPLS
 - ▶ Specifies the maximum number of concurrent applications (local and remote) allowed to be connected to the database
- AVGAPPLS
 - ▶ This parameter is used by the SQL optimizer to help estimate how much buffer pool will be available at run-time for the access plan chosen.
 - ▶ The SQL optimizer should know the number of concurrent(complex) queries that will be running in the system so that it can be more conservative in assumptions of buffer pool availability.
- SORTHEAP
 - ▶ This parameter defines the maximum number of private memory pages to be used for private sorts, or the maximum number of shared memory pages to be used for shared sorts.
 - ▶ If the sort is a private sort, then this parameter affects agent private memory.
 - ▶ If the sort is a shared sort, then this parameter affects the database shared memory.
- SHEAPTHRES_SHR - sortheap threshold for shared sorts
 - ▶ This parameter represents a hard limit on the total amount of database shared memory that can be used for sorting at any one time. When the total amount of shared memory for active shared sorts reaches this limit, subsequent sorts will fail (SQL0955C).
 - ▶ If the value of sheapthres_shr is 0, the threshold for shared sort memory is equal to DBM configuration parameter sheapthres.

Tuning Database Configuration Parameters

- CHNGPGS_THRESH
 - ▶ Specifies the level (percentage) of changed pages at which asynchronous page cleaners will be started
 - ▶ Set this parameter to a lower value if insert/update/delete activity is heavy
- NUM_IOCLEANERS
 - ▶ Specifies the number of asynchronous page cleaners
 - ▶ To avoid I/O wait, set this parameter to a higher value if insert/update/delete activity is heavy
- NUM_IOSERVERS
 - ▶ Specifies the number of I/O servers for a database
 - ▶ Set this value to be one or two more than the number of physical devices present on the server to maximize I/O parallelism

Tuning Database Configuration Parameters

- LOCKLIST
 - ▶ One lock list per database, it contains locks held by all connected apps
 - ▶ Increase the value if insert/update/delete activity is heavy
- MAXLOCKS
 - ▶ Defines a percentage of the lock list held by an application that must be filled before the database manager performs lock escalation
- MINCOMMIT
 - ▶ Delay writing of log records to disk until min # commits performed
 - ▶ Increase count if high update activity from many concurrent users

Chapter 12: Monitoring and Tuning

DB2 Architecture Review
Database Manager and Database Configuration Parameters

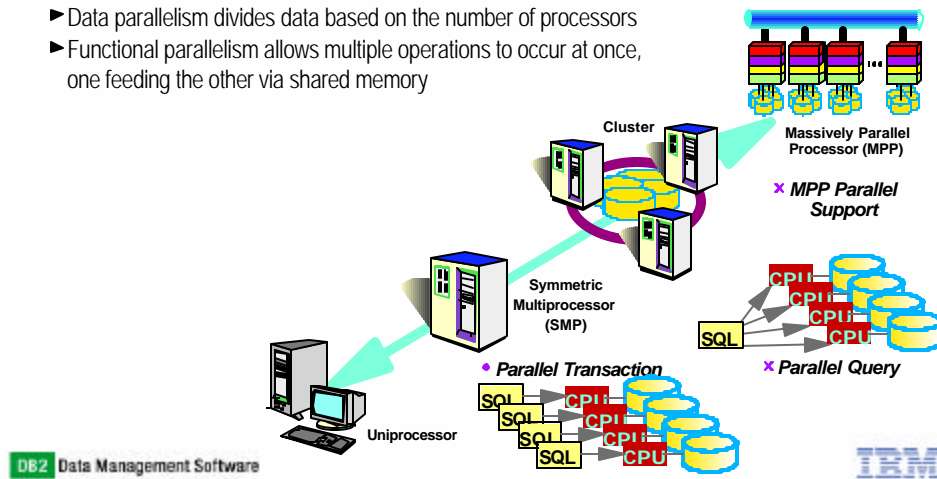
Query Parallelism

EXPLAIN Facility
SNAPSHOT Monitors
EVENT Monitors
Health Monitors and Health Center
DB2 Diagnostics Logs

IBM Software Group

Query Parallelism

- Inter-partition parallelism - Parallel Query
- Intra-partition parallelism - Parallel Transaction
 - ▶ DB2 is able to take advantage of multiple processors in a SMP machine to parallelize non update operations
 - ▶ Enabled by INTRA_PARALLEL parameter
 - ▶ Data parallelism divides data based on the number of processors
 - ▶ Functional parallelism allows multiple operations to occur at once, one feeding the other via shared memory



Query Parallelism

- Instance level (DBM CFG)
 - ▶ INTRA_PARALLEL
 - Enables and disables parallelism for the instance
 - ▶ MAX_QUERYDEGREE
 - Maximum degree of parallelism for the instance
- Database level (DB CFG)
 - ▶ DFT_DEGREE
 - Default degree of parallelism for the database
- Statement Level
 - ▶ CURRENT DEGREE
 - Sets degree of parallelism for dynamic statements
 - ▶ DEGREE
 - Sets degree of parallelism for statements in a package at bind time
 - ▶ RUNTIME DEGREE
 - Changes or sets degree of parallelism for a running application
 - Can only REDUCE degree of parallelism
- Client Level (db2cli.ini)
 - ▶ DB2DEGREE
 - Sets degree of parallelism for CLI/ODBC applications

Which Degree of Parallelism is used?

- Degree of parallelism will be the lowest of:
 - MAX_QUERYDEGREE
 - RUNTIME DEGREE
 - Degree when the statement is precompiled or executed

- Assuming INTRA_PARALLEL is set to YES !

Chapter 12: Monitoring and Tuning

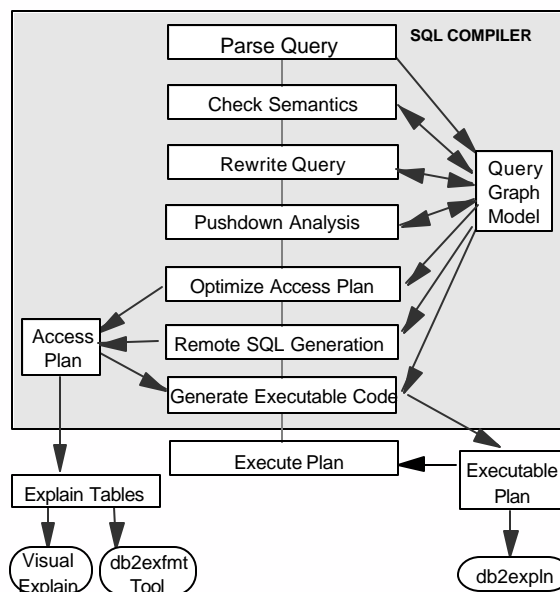
DB2 Architecture Review
Database Manager and Database Configuration Parameters
Query Parallelism
EXPLAIN Facility
SNAPSHOT Monitors
EVENT Monitors
Health Monitors and Health Center
DB2 Diagnostics Logs

IBM Software Group

DB2 Optimizer Characteristics

- Cost based optimizer, uses current statistics to determine how to retrieve data (i.e. access plan)
- Very important to keep statistics up to date
 - ▶ Use RUNSTATS command will update statistics
 - ▶ Use the REORGCHK command with UPDATE STATISTICS to also invoke RUNSTATS
 - ▶ Do RUNSTATS or REORGCHK ON TABLE SYSTEM on system catalog tables.
- Optimizer has a "throttle" to control how much optimization is done during access plan generation
 - ▶ Parameter is DFT_QUERYOPT, range is 0 - 9
 - ▶ Default optimization level is 5, generally good for most queries
 - ▶ Consider reducing optimization level for simple statements and/or require short query compilation time
 - ▶ Complex queries should use at least the default query optimization level
- Optimizer has ability to rewrite SQL statement to improve performance
- Use Explain tool to see rewritten SQL statement

Query Compilation



Monitoring and Tuning a Database

- Monitoring and tuning a DB should be done as follows:
 - ▶ Define the objectives
 - ▶ What is wrong or what do you want to accomplish
 - ▶ Determine information to be analyzed
 - ▶ Determine the monitor(s) to be used
 - ▶ Obtain monitor data
 - ▶ Analyze the information
 - ▶ Determine the changes required
 - ▶ Implement changes (one at a time)

Monitoring Facilities

- DB2 provides different facilities for monitoring the your SQL, databases, and activity
- The main tools available are the:
 - ▶ Explain Facility
 - Visual explain(data stored in EXPLAIN tables)
 - Text explain (db2expln, dynexpln, db2exfmt)
 - ▶ Snapshot Monitor
 - ▶ Event Monitor
 - ▶ Health Center

What is an "Explain" ?

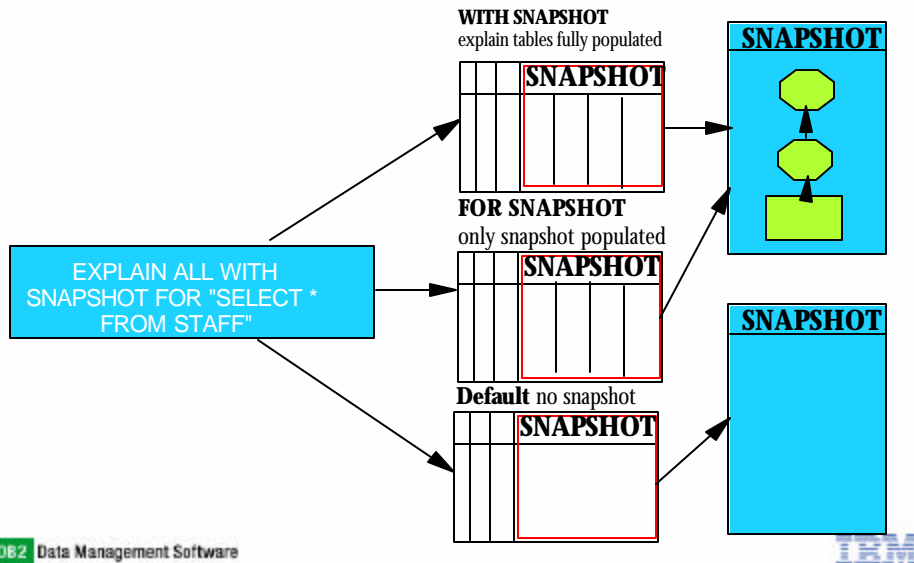
- Facility to capture detailed information about the access plan chosen by the SQL compiler to resolve an SQL statement
- Both static and dynamic SQL are supported
- All elements of SQL processing are captured - table access, index access, joins, unions, scans, etc
- Timing information called "Timerons" is captured for each processing step
- Timerons are a unit of measurement used to give a rough relative estimate of the resources, CPU and I/O costs
- Access plan information captured in DB2 explain tables, created automatically or by executing EXPLAIN.DDL script
- Information can be analyzed through text or GUI based tools
- GUI: Visual Explain
- Text:
 - ▶ db2expln - Capture access plan for static SQL statements in packages that are stored in the DB2, need to provide database name, package name, package creator, and section number
 - ▶ dynexpln - Capture access plan for dynamic SQL statement
 - ▶ db2exfmt - Formats contents of the explain tables

Gathering Explain Data

- There are 4 general methods of populating the Explain tables:
 - ▶ EXPLAIN Statement
 - Captures information about the access plan chosen for the supplied explainable statement and places this information into the Explain tables
 - ▶ CURRENT EXPLAIN MODE Special Register
 - Gather information on dynamic SQL
 - ▶ CURRENT EXPLAIN SNAPSHOT Special Register
 - Gather explain snapshot data for dynamic SQL
 - ▶ BIND options
 - Gather information during the bind process

The EXPLAIN Statement

- Good if need information on one SQL statement
- Will end up populating all or subset of Explain tables



Explain Special Registers

- Can set interactively or can be embedded in dynamic SQL program.
- There are 2 special Registers:
 - ▶ EXPLAIN MODE (only explain data populated)
 - ▶ EXPLAIN SNAPSHOT (full explain plus snapshot)

```
SET CURRENT EXPLAIN MODE [ NO | YES | EXPLAIN ]
SET CURRENT EXPLAIN SNAPSHOT [ NO | YES | EXPLAIN ]
```

- Options:
 - ▶ NO - Information is not gathered
 - ▶ YES - tables populated during execution of all statements until register set to NO
 - ▶ EXPLAIN - tables populated without execution
- Options For EXPLAIN MODE only:
 - ▶ RECOMMEND INDEXES and EVALUATE INDEXES
 - ▶ Used by the Designer Advisor

Explain BIND Options

- There are 2 BIND options:
 - ▶ EXPLAIN (only explain info)
 - ▶ EXPLSNAP (full explain plus snapshot)

```
BIND package EXPLAIN [ NO | YES | ALL ]  
BIND package EXPLSNAP [ NO | YES | ALL ]
```

- NO - Information is not gathered
- YES - static SQL information gathered
- ALL - static & dynamic SQL information gathered

Examining Explain Data - Text Tools

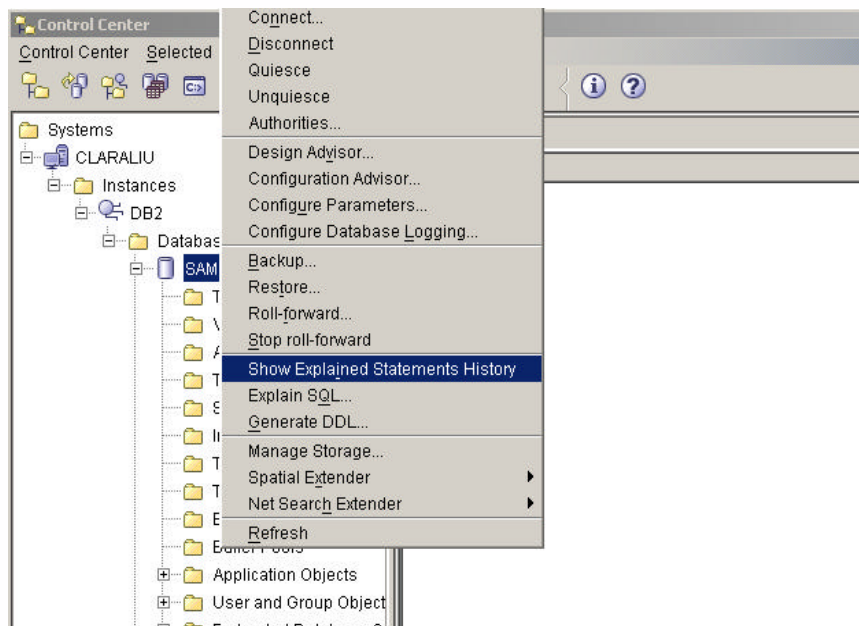
- There are three text-based utilities for EXPLAIN information:
 - ▶ dynexpln - Dynamic SQL access plan information
 - ▶ db2expln - Static SQL access plan information
 - ▶ db2exfmt - Static and Dynamic SQL access plan information
 - Recommended text tool

Examining Explain Data - GUI Tools

- GUI tool is called Visual Explain
- Presents access plan in a tree like structure
- Visual Explain can only display access plans "explained" (i.e. generated) with the "snapshot" option
- There are two ways to invoke Visual Explain:
 - ▶ Control Center
 - Explained statements history
 - All explained statements will be shown but only those that created with SNAPSHOT can be examined with VISUAL EXPLAIN
 - EXPLAIN SQL
 - Good for a single SQL statement
 - ▶ Command Center
 - Choose "Statement" and select "EXPLAIN"
 - Select "Generate Access Plan" button



Show Explained Statements History



EXPLAIN SQL From the Control Center

Explain SQL Statement - SAMPLE
CLARALIU - DB2 - SAMPLE

SQL text

```
SELECT empno, firstname, lastname  
FROM department, employee  
WHERE workdept=deptno  
ORDER BY lastname
```

Query number: 1

Query tag:

Optimization class: 5

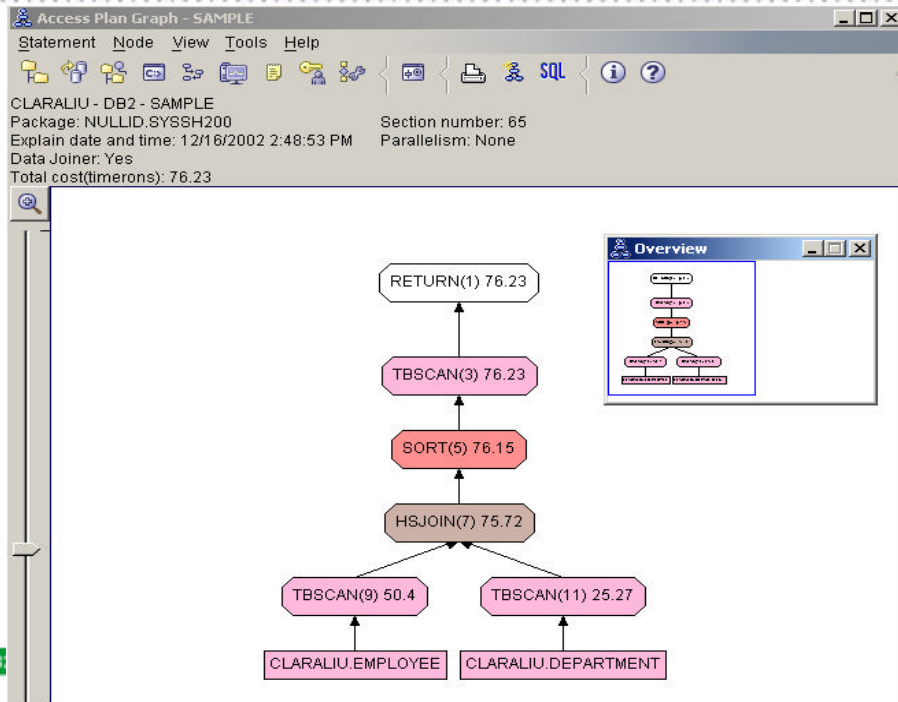
Populate all columns in Explain tables

OK Cancel Help

DB2 Data Management Software



Visual Explain



Chapter 12: Monitoring and Tuning

DB2 Architecture Review
Database Manager and Database Configuration Parameters
Query Parallelism
EXPLAIN Facility
SNAPSHOT Monitors
EVENT Monitors
Health Monitors and Health Center
DB2 Diagnostics Logs
IBM Software Group

Snapshot Monitors

- Provides cumulative information in the form of resettable counters
- Monitors set by switches at the instance or application level with:
 - ▶ UPDATE DBM CONFIGURATION command (**instance level**)
 - ▶ UPDATE MONITOR SWITCHES command (**application level**)
- Instance level switches affect all databases in the instance
- Can get a snapshot from:
 - ▶ CLP using the GET SNAPSHOT command
 - ▶ SQL table functions
 - ▶ Snapshot monitor APIs in a C or C++ application
- Amount of data returned depends on level
 - ▶ Database manager
 - ▶ Database
 - ▶ Application
 - ▶ Bufferpools
 - ▶ Tablespace
 - ▶ Table
 - ▶ Lock
 - ▶ Dynamic SQL

Snapshot Monitors

Group	Info provided	Monitor Switches
Sorts	number of heaps used, overflows, sorts, performance	SORT
Locks	number of locks held, number of deadlocks	LOCK
Tables	measure activity on table (rows read, rows written)	TABLE
Buffer Pools	number reads and writes, time taken	BUFFERPOOL
UOW	start/end times and completion status	UOW
SQL	start/stop time statement identification	STATEMENT
Time Stamp	time stamps	TIMESTAMP

DB2 Data Management Software



Display, Set, Reset Monitor Switches

- To check the status of the monitor switches:

- ▶ GET MONITOR SWITCHES

```

Monitor Recording Switches

Switch list for db partition number 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
Get timestamp information (TIMESTAMP) = OFF
    
```

- To activate monitor switches:

- UPDATE MONITORE SWITCHES USING

- BUFFERPOOL on LOCK on SORT on STATEMENT on TIMESTAMP on TABLE on UOW on

- The switches will remain active until the application (CLP) terminates, or until they are deactivated with another UPDATE MONITOR SWITCHES command

- UPDATE MONITORE SWITCHES USING

- BUFFERPOOL off LOCK off SORT off STATEMENT off TIMESTAMP off TABLE off UOW off

- To reset counter of monitor switches:

- RESET MONITOR ALL

DB2 Data Management Software



GET SNAPSHOT Command

- Obtain snapshot information by entering the GET SNAPSHOT command with the desired parameters

```
>>-GET SNAPSHOT FOR----->
>---+--DATABASE MANAGER---+--WRITE TO FILE 1----->
| +--DB MANAGER-----+
| | '-DBM-----'
| +--ALL---+--DCS---+--DATABASES-----+
| | '-DCS-'
| +--ALL---+--DCS---+--APPLICATIONS-----+
| | '-DCS-'
| +--ALL BUFFERPOOLS-----+
| +-----+--APPLICATION---+--APPLID---appl-id---+
| | '-DCS-' | '-AGENTID---appl-handle-' |
| +--FCM FOR ALL DBPARTITIONNUMS-----+
| +--LOCKS FOR APPLICATION---+--APPLID---appl-id---+
| | | '-AGENTID---appl-handle-' |
| +--ALL REMOTE_DATABASES-----+
| +--ALL REMOTE_APPLICATIONS-----+
| +--ALL-----+--ON---database-alias---'
| +-----+--DATABASE---+
| | '-DCS-' | '-DB-----' |
| +-----+--APPLICATIONS---+
| | '-DCS-' |
| +--TABLES-----+
| +--TABLESPACES-----+
| +--LOCKS-----+
| +--BUFFERPOOLS-----+
| +--REMOTE_DATABASES-----+
| +--REMOTE_APPLICATIONS-----+
| '-DYNAMIC SQL-----'
```

GET SNAPSHOT Command - Examples

- GET SNAPSHOT FOR DBM

```
Database Manager Snapshot

Node name =
Node type = DB Server with local and remote clients
Instance name = DB2
Number of database partitions in DB2 instance = 1
Database manager status = Active

Product name = DB2 v8.1.0
Product identification =
Service level = n020211

Sort heap allocated = 0
Post threshold sorts = Not Collected
Piped sorts requested = 0
Piped sorts accepted = 0
```

- GET SNAPSHOT FOR BUFFERPOOLS ON sample

```
Bufferpool Snapshot

Bufferpool name = IBMDEFAULTBP
Database name = SAMPLE
Database path = /home/andrewkm/andrewkm/NODE0000/
SQL00001/
Input database alias = SAMPLE
Buffer pool data logical reads = 34
Buffer pool data physical reads = 15
Buffer pool data writes = 0
Buffer pool index logical reads = 59
Buffer pool index physical reads = 33
```

Capturing Snapshots Using SQL - Direct Access

- Issue queries with snapshot table functions, and receive result sets containing monitor data
- Users must have SYSADM, SYSCTRL, or SYSMAINT authority to capture database snapshots
- Snapshot table functions:
 - ▶ SNAPSHOT_AGENT
 - ▶ SNAPSHOT_APPL, SNAPSHOT_APPL_INFO
 - ▶ SNAPSHOT_BP
 - ▶ SNAPSHOT_CONTAINER
 - ▶ SNAPSHOT_DATABASE
 - ▶ SNAPSHOT_DBM
 - ▶ SNAPSHOT_DYN_SQL
 - ▶ SNAPSHOT_FCM, SNAPSHOT_FCMNODE
 - ▶ SNAPSHOT_LOCK, SNAPSHOT_LOCKWAIT
 - ▶ SNAPSHOT QUIESCERS
 - ▶ SNAPSHOT_RANGERS
 - ▶ SNAPSHOT_STATEMENT
 - ▶ SNAPSHOT_SUBJECT
 - ▶ SNAPSHOT_SWITCHES
 - ▶ SNAPSHOT_TABLE
 - ▶ SNAPSHOT_TBREORG
 - ▶ SNAPSHOT_TBS, SNAPSHOT_TBS_CFG

Capturing Snapshots Using SQL - Direct Access

- The monitor table functions have two input parameters:
 - ▶ Database name
 - If NULL is used, the database currently connected will be used
 - ▶ Partition number
 - Enter an integer between 0 and 999 corresponding to partition number wished to monitor
 - To capture a snapshot for the current connected partition, enter a value of -1 or NULL
 - To capture a global snapshot, enter a value of -2
- Example:
 - ▶ To capture a snapshot of the ranges for a table space map:

```
SELECT * FROM TABLE( SNAPSHOT_RANGES( 'SAMPLE', -1 ))
AS SNAPSHOT_RANGES
```
 - ▶ To capture a snapshot of dynamic SQL statement cache information:

```
SELECT * FROM TABLE( SNAPSHOT_DYN_SQL( 'SAMPLE', -1 ))
AS SNAPSHOT_DYN_SQL
```
 - ▶ To capture a snapshot of lock information:

```
SELECT * FROM TABLE( SNAPSHOT_LOCK( 'SAMPLE', -1 ))
AS SNAPSHOT_LOCK
```
 - ▶ To capture a snapshot of application information:

```
SELECT agent_id, locks_held FROM TABLE( SNAPSHOT_APPL( 'SAMPLE', -1 ))
AS SNAPSHOT_APPL
```

Capturing Snapshots Using SQL - File Access

- Use the `SNAPSHOT_FILEW` stored procedure to capture snapshots data and save them to files
- Users must have `SYSADM`, `SYSCTRL`, or `SYSMAINT` authority to capture database snapshots
- Once an authorized users have called the `SNAPSHOT_FILEW` stored procedure, any user can issue queries with the corresponding snapshot table functions to retrieve data generated

Capturing Snapshots Using SQL - File Access

- In addition to identifying the database and partition to be monitored, the snapshot request type that determines the scope of monitor data is also required
- For complete list of snapshot request type and its associated number, refer to the DB2 Administration Guide
- Some snapshot request types are:

Snapshot Request Types	Numbers
<code>SNAPSHOT_DATABASE</code>	2
<code>SNAPSHOT_APPL</code>	6
<code>SNAPSHOT_LOCKWAIT</code>	6
<code>SNAPSHOT_TABLE</code>	5
<code>SNAPSHOT_CONTAINER</code>	13
<code>SNAPSHOT_BP</code>	22
<code>SNAPSHOT_DYN_SQL</code>	36

Capturing Snapshots Using SQL - File Access

- The SNAPSHOT_FILEW stored procedure has three input parameters:

- ▶ Snapshot request type
- ▶ Database name
 - ─ If NULL is used, the database currently connected will be used
- ▶ Partition number
 - ─ Enter an integer between 0 and 999 corresponding to partition number wished to monitor
 - ─ To capture a snapshot for the current connected partition, enter a value of -1 or NULL
 - ─ To capture a global snapshot, enter a value of -2

- Example:

- ▶ To capture a snapshot of application information about the SAMPLE database for the current connected partition:

```
CALL SNAPSHOT_FILEW ( 6,'SAMPLE',-1 )
```

- Any user can access snapshot data from SNAPSHOT_FILEW files by using snapshot table functions in SQL queries

- Example:

- ▶ A snapshot of general application information for the SAMPLE database is made:

```
SELECT * FROM TABLE( SNAPSHOT_APPL(CAST(NULL AS VARCHAR(1)),  
CAST (NULL AS INTEGER)))
```

as SNAPSHOT_APPL

DB2 Data Management Software



DB2 Data Management Software



business software

Chapter 12: Monitoring and Tuning

DB2 Architecture Review
Database Manager and Database Configuration Parameters
Query Parallelism
EXPLAIN Facility
SNAPSHOT Monitors
EVENT Monitors
Health Monitors and Health Center
DB2 Diagnostics Logs

IBM Software Group

Event Monitors

- Event monitors are used to collect information about the database and any connected applications when specified events occur
- Events represent transitions in database activity
 - ▶ e.g. connections, deadlocks, statements, and transactions
- Event monitors are database objects, they are created and manipulated using SQL data definition language statements
- No switches, define event monitors by the type of event wish to monitor
- Event types:
 - ▶ DEADLOCKS - Applications involved, and locks in contention
 - ▶ DEADLOCKS WITH DETAILS - Detailed locking information
 - ▶ STATEMENTS - Statement start/stop time, CPU used, text of dynamic SQL
 - ▶ TRANSACTIONS - UOW work start/stop time, previous UOW time
 - ▶ CONNECTIONS - All application level counters.
 - ▶ DATABASE - All database level counters.
 - ▶ BUFFERPOOLS - Counters for buffer pool, prefetchers, page cleaners and direct I/O for each buffer pool
 - ▶ TABLESPACES - Counters for buffer pool, prefetchers, page cleaners and direct I/O for each table space
 - ▶ TABLES - Rows read or written for each table

Creating a Table Event Monitor

- A table event monitor streams event records to SQL tables, it makes capture, parse, and management of event monitoring data more easily
- For every event type an event monitor collects, target tables are created for each of the associated logical data groups
- Need SYSADM or DBADM to create a table event monitor
- Indicate that event monitor data is to be collected in a table or sets of table:

```
CREATE EVENT MONITOR stmtmon
  FOR STATEMENTS WITH DETAILS
  WRITE TO TABLE
CREATE EVENT MONITOR dlmon
  FOR CONNECTIONS, DEADLOCKS WITH DETAILS
  WRITE TO TABLE BUFFERSIZE 8 BLOCKED
```

 - ▶ BUFFERSIZE
 - ─ Specify the size of the table event monitor buffers
 - ▶ BLOCKED or NONBLOCKED
 - ─ For blocked event monitors, each agent that generates an event will wait for the event buffers to be written to table if they are full. This can degrade database performance, as the suspended agent and any dependent agents cannot run until the buffers are clear. Use the BLOCKED clause to ensure no losses of event data

Creating a Table Event Monitor

■ Example #1:

```
CREATE EVENT MONITOR stmtmon FOR STATEMENTS WITH DETAILS WRITE TO TABLE
```

▶ The STATEMENTS event type collect data from the event_connheader, event_stmt, and event_subsection logical data groups, following event tables are created in the creator's schema:

- user1.connheader_stmtmon
- user1.stmt_stmtmon
- user1.subsection_stmtmon
- user1.control_stmtmon

▶ The user1.control_stmtmon table is created for every write-to-table event monitor

▶ The control table contains event monitor metadata containing event_start, event_db_header, and event_overflow data

■ Example #2:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE CONN,
```

```
DLCONN (EXCLUDES(agent_id, lock_wait_start_time)),
```

```
DLLOCK (INCLUDES(lock_mode, table_name))
```

- ▶ All the monitor elements for CONN are captured
- ▶ For DLCONN, all monitor elements except agent_id and lock_wait_start_time are captured
- ▶ For DLLOCK, lock_mode, table_name are the only monitor elements captured

Creating a Table Event Monitor

■ Example #3:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE CONN,
```

```
DLCONN (TABLE mydept.dlconnections EXCLUDES(agent_id, lock_wait_start_time)),
```

```
DLLOCK (TABLE dllocks IN mytablespace INCLUDES(lock_mode, table_name))
```

▶ Target tables and the associated table spaces are:

- CONN: user1.conn_dlmon (on the default table space)
- DLCONN: mydept.dlconnections (on the default table space)
- DLLOCK: user1.dllocks (on the MYTABLESPACE table space)

■ Example #4:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE BUFFERSIZE 8 NONBLOCKED MANUALSTART
```

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE BUFFERSIZE 8 NONBLOCKED AUTOSTART NONBLOCKED
```

- ▶ Specify if the event monitor is to be activated automatically each time the database starts
- ▶ By default, event monitors are not activated automatically when the database starts

Creating a File or Pipe Event Monitor

- Information collected by event monitors can be stored in files or named pipes

- File Event Monitor

- Specify the directory where event files are to be stored
- Streams event records to numbered files with the extension ".evt" (e.g. 00000000.evt, 00000001.evt, and 00000002.evt)

- Example #1:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8 BLOCKED AUTOSTART
```

- Example #2:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8 NONBLOCKED  
MAXFILES NONE MAXFILESIZE NONE MANUALSTART
```

- Pipe Event Monitor

- Streams event records directly from the event monitor to a named pipe
- If the event monitor is unable to write data to the pipe (for instance, if it is full), monitor data will be lost

- Example:

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO PIPE '/home/rrihi/dlevents' MANUALSTART
```

Managing Event Monitors

- SYSADM or DBADM authorities are required to create, drop, activate, deactivate an event monitor

- Event Monitor Definitions Stored in Catalogs

- SYSCAT.EVENTMONITORS
- SYSCAT.EVENTS

- To get status of an event monitor:

- Select monitor from SYSCAT.EVENTMONITORS table OR
- Use the SQL Function EVENT_MON_STATE (event-monitor-name)
 - 0 means inactive
 - 1 means active

- To activate or deactivate an event monitor:

- SET EVENT MONITOR event-monitor name STATE = 0|1

- To drop an event monitor:

- DROP EVENT MONITOR event-monitor-name

Examine Event Monitor Output

- How do I examine event monitor output?
 - ▶ Query the event monitor tables
 - ▶ Use db2evmon productivity tool
 - ▶ Use db2eva from Control Center
- Examples:
 - ▶ db2evmon db <dbname> evm <evmon-name> path <evmon-path>
 - ▶ db2eva db <dbname> evm <evmon-name>

Chapter 12: Monitoring and Tuning

DB2 Architecture Review
Database Manager and Database Configuration Parameters
Query Parallelism
EXPLAIN Facility
SNAPSHOT Monitors
EVENT Monitors
Health Monitors and Health Center
DB2 Diagnostics Logs
IBM Software Group

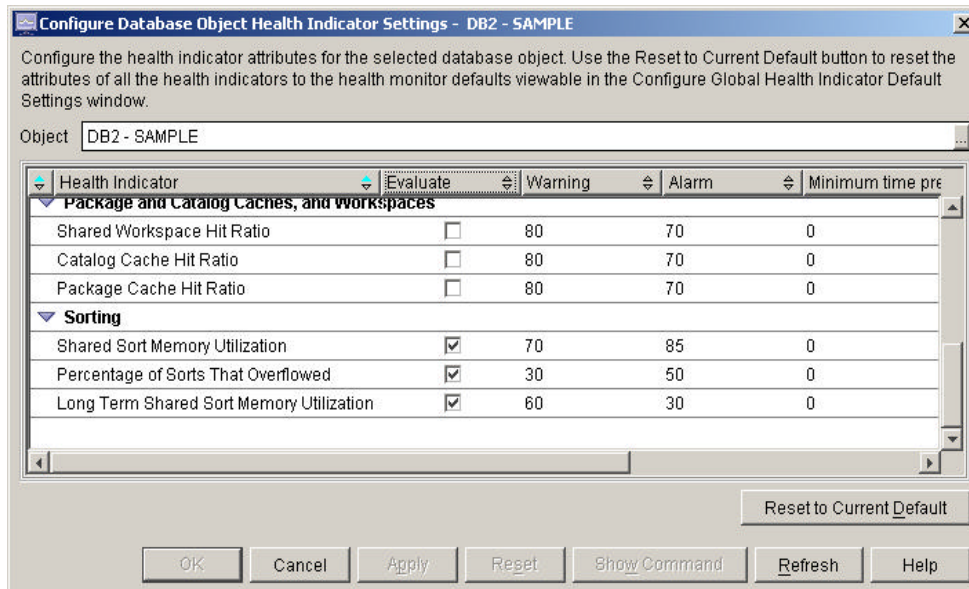
Monitoring Health of DB2 Systems

- Two new features introduced in DB2 V8 to monitor the health of DB2 systems:
 - ▶ Health Monitor
 - ▶ Health Center
- The management by exception capability generates alert to potential system health issues
- The Health Monitor is a server-side tool that constantly monitors the health of the instance, even without user interaction
- An alert is raised if a defined threshold has been exceeded (e.g. the available log space is not sufficient) or an object is in an abnormal state (e.g. an instance is down)
- When an alert is raised two things can occur:
 - ▶ Alert notifications can be sent by e-mail or to a pager address
 - ▶ Preconfigured actions can be taken (e.g. a script or a task)
- The Health Monitor checks the state of your system against the health-indicator thresholds to determine when to issue an alert
- Use the Health Center, commands, or APIs to customize the threshold settings of the health indicators, and define who should be notified and what script or task should be run if an alert is issued

Health Alerts and Reactions

- A. Set up "health" monitoring
 - ▶ Determine the database system parameters that need to be monitored
 - ▶ Determine which database objects will be monitored
 - ▶ Identify / create snapshot or event monitors needed to collect data for the health of the database system
 - ▶ Determine threshold values of the monitored parameters that indicate "health" problems
 - ▶ Implement notification mechanisms and start the monitors
- B. React to a "health" issue
 - ▶ Identify an issue from the performance or event monitor data
 - ▶ Find more details and review them
 - ▶ Determine one or more alternative actions to fix the problem (use help and manuals to do this), and how other parameters will be affected
 - ▶ Write scripts (if necessary) to execute the actions or invoke the needed tools
 - ▶ Verify that the problem has been resolved (by running script or checking the affected objects)

Configure Health Indicators

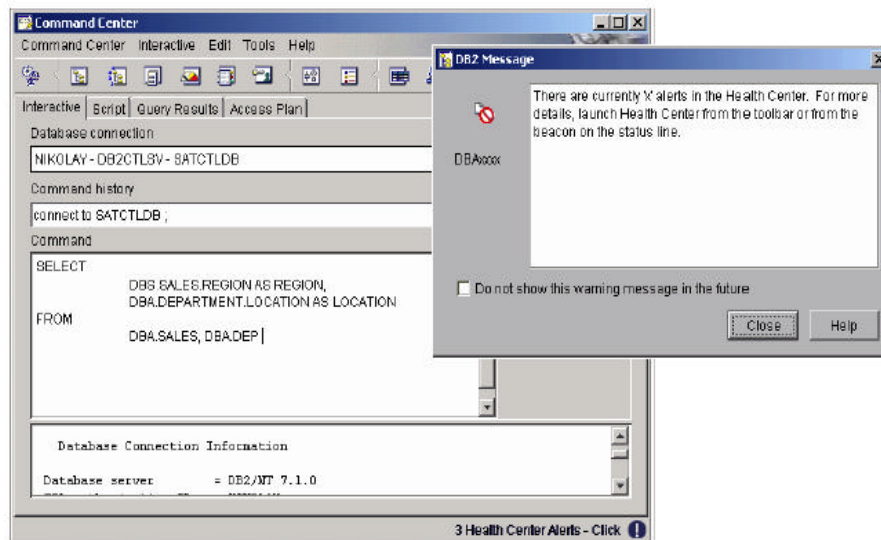


DB2 Data Management Software



Health Alerts

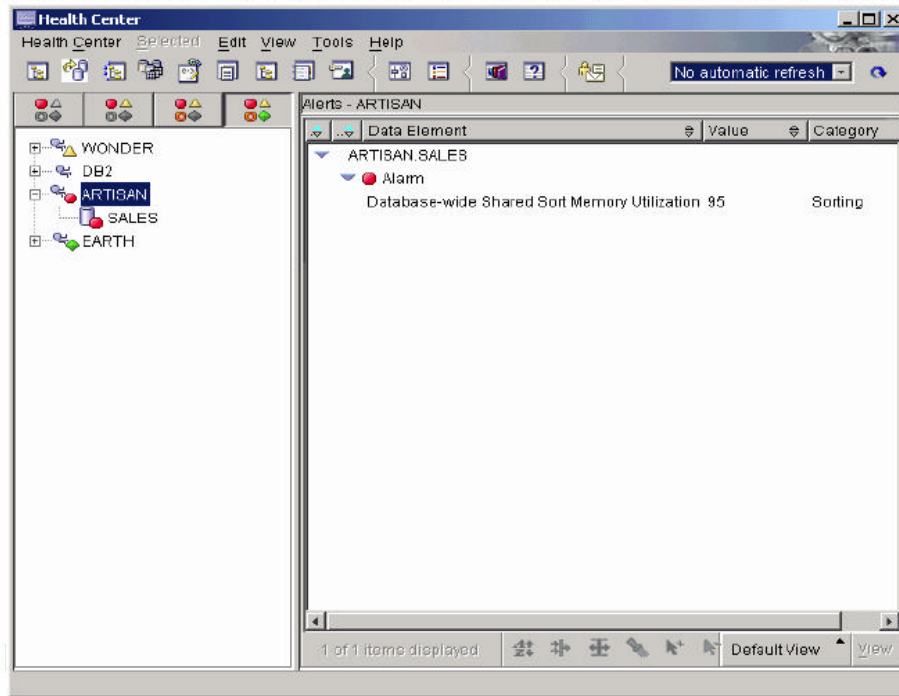
- User receives an alert (E- mail, DB2 Message or Health Center Beacon)



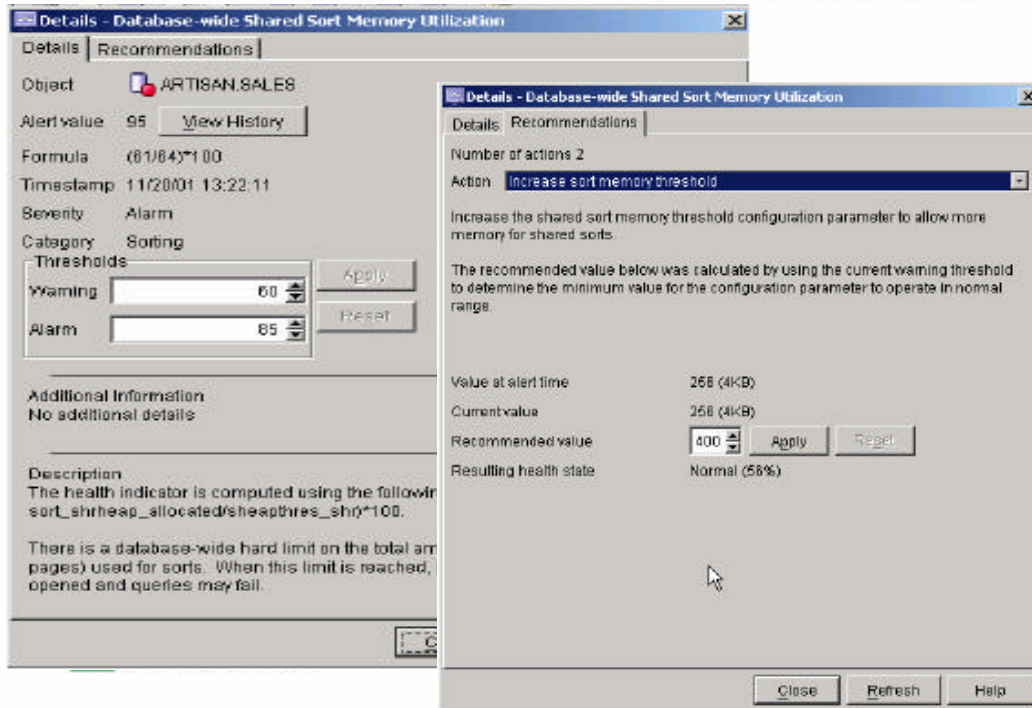
DB2 Data Management Software



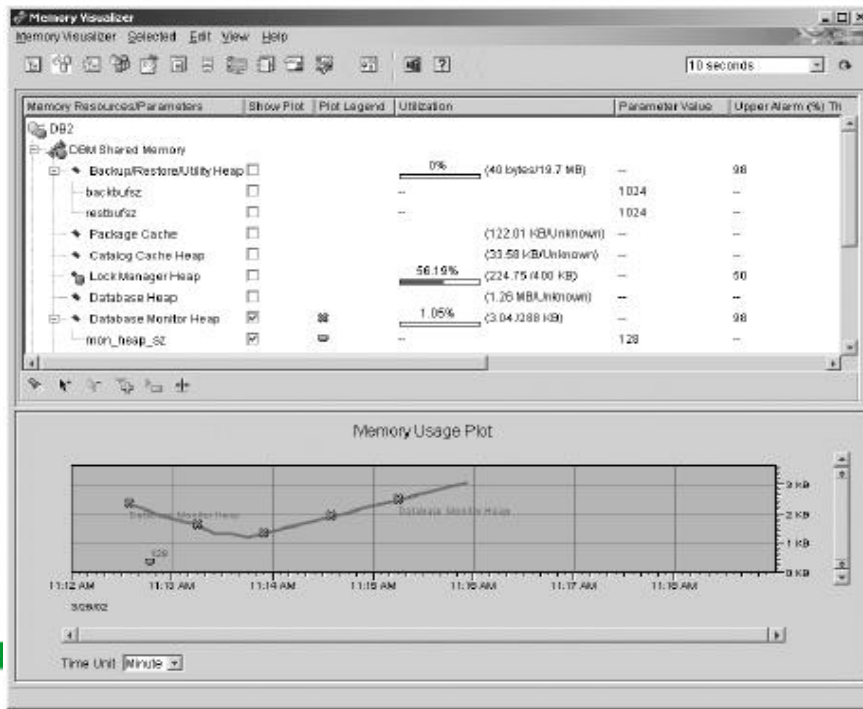
Health Center



Health Center



Health Center



DB2

DB2 Data Management Software

IBM

business software

Chapter 12: Monitoring and Tuning

- DB2 Architecture Review
- Database Manager and Database Configuration Parameters
- Query Parallelism
- EXPLAIN Facility
- SNAPSHOT Monitors
- EVENT Monitors
- Health Monitors and Health Center
- DB2 Diagnostics Logs**

IBM Software Group

DB2 Administration Notify Logs

- DB2 provides diagnosis information at the point failure to the administration notification log
- On UNIX platforms, the administration notification log is a text file called instance.nfy
- On Windows, all administration notification messages are written to the Event Log
- The DBM configuration parameter `notifylevel` specifies the level of information to be recorded
- There are 5 levels of information possible:
 - ▶ 0 -- No administration notification messages captured (not recommended)
 - ▶ 1 -- Fatal or unrecoverable errors
 - ▶ 2 -- Immediate action required
 - ▶ 3 -- Important information, no immediate action required (default)
 - ▶ 4 -- Informational messages