Agenda

- Introduction to DB2 Monitoring Internals
- Introduction to monitoring via SQL
- Monitoring Status and Performance with SQL
- Monitoring Health and Diagnosing problems with SQL
- Using the Database Health Monitor
- Using Optim Performance Manager included with DB2 AESE
Introduction to DB2 Monitoring Internals
DB2 Monitoring Internals

- **What is Snapshot monitoring?**
  - A “picture” of the state of the DB2 system at a point in time
  - A report on a set of counters (mostly) stored inside DB2
  - Just like a camera, a snapshot is initiated by a human

- **What is an Event monitor?**
  - A similar set of information (counters mostly) triggered by a defined event
  - For example, information about what an application did when it disconnects from the database
  - We won’t discuss Event Monitoring in this session
Types of Monitor Elements

- **Counters**
  - Measures the number of times an activity occurs (always increases) – Can be reset
  - E.g.: Rows read from a table, number of physical page reads, etc.

- **Gauges**
  - Indicates the current value of an item (may increase or decrease over time) – not reset (value are current state)
  - E.g.: Number of currently active sorts, amount of log space currently allocated, etc.

- **Information**
  - Reference type information about a monitor element – not reset
  - E.g.: Server Platform, Authentication ID of connected user, etc.

- **Timestamp**
  - Indicates the date and time an activity took place. – not reset. Number of seconds and microseconds since Jan 1, 1970
  - E.g.: Last time a database was backed up, snapshot time, etc.

- **Time**
  - Returns the number of seconds and microseconds spent on an activity – Can be reset
  - E.g.: Time spent reading data pages, elapsed time of a unit of work, etc.
How Does It Work?

```
db2 get snapshot for database …
```

**DB2 Memory**

- **Bufferpool**
- **Pkg Cache**
- **Sortheap**

**I/O**

**Tablespaces**

**Snapshot as of 2004-10-01 9:04am**
- Pkg Cache Lookup = 2347
- Sortheap Allocated = 100Meg
- Async Read Time = 5000sec
- Logical Read Time = 100ms
Command Line Syntax

- **GET SNAPSHOT FOR**
  - DATABASE MANAGER
  - DATABASE ON <dbname>
  - TABLESPACES ON <dbname>
  - TABLES ON <dbname>
  - BUFFERPOOLS ON <dbname>
  - LOCKS ON <dbname>
  - APPLICATIONS ON <dbname>
  - DYNAMIC SQL ON <dbname>

- You must have SYSADM, SYSCTRL, SYSMAINT or SYSMON authority
Introduction to Monitoring via SQL Functions
What’s a Table UDF

- **UDF = User Defined Function**
  - Shipped with DB2 – not user defined
- **A function that takes a structured set of information and makes appear to be a table**

<table>
<thead>
<tr>
<th>Instance name</th>
<th>Status</th>
<th>Serv_level</th>
<th>Priv_sort_alloc</th>
<th>Priv_sort_high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database manager status</td>
<td>Active</td>
<td>S040219</td>
<td>0</td>
<td>277</td>
</tr>
<tr>
<td>Service level</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private Sort heap allocated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private Sort heap high water mark</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example:

- Snapshot_dbm
How Does It Work?

SELECT * FROM TABLE(SNAP_GET_DBM)

<table>
<thead>
<tr>
<th>Snapshot_time</th>
<th>Pkg_cache_lookup</th>
<th>Sorheap_alloc</th>
<th>Async_read_time</th>
<th>Logical_read_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-10-01 9:04am</td>
<td>2347</td>
<td>100</td>
<td>5000</td>
<td>100</td>
</tr>
</tbody>
</table>
The Syntax of a Select Statement

select * from `table(snap_get_dbm(-1))` as sntable

<table>
<thead>
<tr>
<th>Table Function</th>
<th>Name of the table function</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>table</code></td>
<td><code>snap_get_dbm(-1)</code></td>
<td><code>-1 = current partition number</code></td>
</tr>
</tbody>
</table>

select * from `table(snap_get_db_v91('',-1))` as sntable

Arguments

```
"" = current database
-1 = current partition number
```
DB2 9 Makes Your Life Simpler
– Administrative Views

- Table Functions still exist but now you have VIEWS
- All views are in the SYSIBMADM schema
- Convert coded values to text strings
- Can be a control point to allow people with lower authority to view monitor information
  - Grant select on view and execute on table function
SNAPSHOT Views

- Database Manager
  - SNAPDBM
  - SNAPDBM_MEMORY_POOL

- Database Level
  - SNAPDB
  - SNAPDB_MEMORY_POOL
  - SNAPBP
  - SNAPBP_PART
  - SNAPHADR

- Application Level
  - SNAPAPPL
  - SNAPAPPL_INFO
  - SNAPLOCKWAIT *
  - SNAPSTMT
  - SNAPAGENT
  - SNAPSHOT_SUBSECTION
  - SNAPAGENT_MEMORY_POOL
  - SNAPDYN_SQL
  - SNAPLOCK *

- Object Level
  - SNAPTAB
  - SNAP_TAB_REORG
  - SNAPTBSP
  - SNAPTBSP_PART
  - SNAPTBSP_QUIESCER
  - SNAPCONTAINER
  - SNAPTBSP_RANGE
  - SNAPUTIL
  - SNAPUTIL_PROGRESS
  - SNAPDETAILLOG
  - SNAPSTORAGE_PATHS

- Database Partitioning Feature (DPF)
  - SNAPFCM
  - SNAPFCM_PART

* Deprecated in 9.7 FP1
“Convenience” Monitor Views

- APPLICATIONS
- APPL_PERFORMANCE
- BP_HITRATIO
- BP_READ_IO
- BP_WRITE_IO
- CONTAINER_UTILIZATION
- LOCKS_HELD *
- LOCKWAIT *
- LOG_UTILIZATION
- LONG_RUNNING_SQL
- QUERY_PREP_COST
- TBSP_UTILIZATION
- TOP_DYNAMIC_SQL

* Deprecated in 9.7 FP1
Administrative Views

- ADMINTABINFO
- ADMINTABCOMPRESSINFO
- ADMIN_GET_INDEX_INFO
- ADMIN_GET_INDEX_COMPRESS_INFO
- ADMIN_EST_INLINE_LENGTH
- ADMIN_IS_INLINED
- ADMIN_GET_DBP_MEM_USAGE
- DBCFG
- DBMCFG
- REG_VARIABLES
- DB_PARTITIONS
- DB_HISTORY
New 9.7 Monitor Functions

New Time Spent and Time Waiting Metrics – find bottlenecks

- **Application Information**
  - `MON_GET_CONNECTION`
  - `MON_GET_CONNECTION_DETAILS`
  - `MON_GET_PKG_CACHE_STMT`
  - `MON_GET_UNIT_OF_WORK`
  - `MON_GET_UNIT_OF_WORK_DETAILS`

- **Object**
  - `MON_GET_TABLE`
  - `MON_GET_INDEX`
  - `MON_GET_TABLESPACE`
  - `MON_GET_CONTAINER`
  - `MON_GET_BUFFERPOOL`
  - `MON_GET_EXTENT_MOVEMENT_STATUS`

- **Workload Management**
  - `MON_GET_WORKLOAD`
  - `MON_GET_WORKLOAD_DETAILS`
  - `MON_GET_SERVICE_SUBCLASS`
  - `MON_GET_SERVICE_SUBCLASS_DETAILS`
Monitoring Performance
With SQL Select Statements
Long Running SQL

SELECT ELAPSED_TIME_MIN,  
    SUBSTR(AUTHID,1,10) AS AUTH_ID,  
    AGENT_ID,  
    APPL_STATUS,  
    SUBSTR(STMT_TEXT,1,20) AS SQL_TEXT  
FROM SYSIBMADM.LONG_RUNNING_SQL  
WHERE ELAPSED_TIME_MIN > 0  
ORDER BY ELAPSED_TIME_MIN DESC

<table>
<thead>
<tr>
<th>ELAPSED_TIME_MIN</th>
<th>AUTH_ID</th>
<th>AGENT_ID</th>
<th>APPL_STATUS</th>
<th>SQL_TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>EATON</td>
<td>878</td>
<td>LOCKWAIT</td>
<td>update org set deptn</td>
</tr>
</tbody>
</table>
Buffer Pool Query

Display buffer pool hit ratios (data, index and XML)

```
SELECT SUBSTR(BP_NAME,1,20) as BP_NAME,
       TOTAL_HIT_RATIO_PERCENT as ALL_HR,
       DATA_HIT_RATIO_PERCENT as DATA_HR,
       INDEX_HIT_RATIO_PERCENT as INX_HR,
       XDA_HIT_RATIO_PERCENT as XML_HR
FROM SYSIBMADM.BP_HITRATIO;
```

<table>
<thead>
<tr>
<th>BP_NAME</th>
<th>ALL_HR</th>
<th>DATA_HR</th>
<th>INX_HR</th>
<th>XML_HR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMDEFAULTBP</td>
<td>98</td>
<td>80</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>LARGE_BP</td>
<td>99</td>
<td>99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMALL_BP</td>
<td>25</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Package Cache Query

- Look at all the queries in the package cache
  - Both Dynamic and Static
  - See execution time, wait time (by component), and much more

```sql
SELECT
  SUBSTR(STMT_TEXT,1,20) AS STMT,
  SECTION_TYPE AS TYPE,
  NUM_EXECUTIONS,
  TOTAL_ACT_TIME AS TOTAL_TIME,
  TOTAL_ACT_WAIT_TIME AS WAIT_TIME
FROM TABLE(MON_GET_PKG_CACHE_STMT('', '', '', -1))
```

<table>
<thead>
<tr>
<th>STMT</th>
<th>TYPE</th>
<th>NUM_EXECUTIONS</th>
<th>TOTAL_TIME (ms)</th>
<th>WAIT_TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select * from emp</td>
<td>D</td>
<td>10</td>
<td>123</td>
<td>7</td>
</tr>
<tr>
<td>with aa as (select *</td>
<td>D</td>
<td>100</td>
<td>2845</td>
<td>860</td>
</tr>
</tbody>
</table>
Package Cache Query

- Other useful bits of information in MON_GET_PKGCACHE_STMT function:
  - NUM_EXECUTIONS
  - PREP_TIME
  - TOTAL_ACT_TIME
  - TOTAL_ACT_WAIT_TIME
  - TOTAL_CPU_TIME
  - LOCK_WAIT_TIME
  - TOTAL_SECTION_SORT_TIME
  - TOTAL_SECTION_SORTS
  - LOCK_ESCALS
  - LOCK_WAITS
  - ROWS_MODIFIED
  - ROWS_READ
  - TOTAL_SORTS
  - SORT_OVERFLOWS
  - DEADLOCKS
  - LOCK_TIMEOUTS
  - LOG_BUFFER_WAIT_TIME
  - LOG_DISK_WAIT_TIME
  - STMT_TEXT CLOB(2MB)
Lock Wait Query

select substr(ai_h.appl_name,1,10) as "Hold App",
    substr(ai_h.primary_auth_id,1,10) as "Holder",
substr(ai_w.appl_name,1,10) as "Wait App",
substr(ai_w.primary_auth_id,1,10) as "Waiter",
lw.lock_mode as "Hold Mode",
lw.lock_object_type as "Obj Type",
substr(lw.tabname,1,10) as "TabName",
substr(lw.tabschema,1,10) as "Schema",
timestampdiff(2,char(lw.snapshot_timestamp - lw.lock_wait_start_time))
    as "waiting (s)"
from  sysibmadm.snapappl_info ai_h,
      sysibmadm.snapappl_info ai_w, sysibmadm.snaplockwait lw
where lw.agent_id = ai_w.agent_id
  and  lw.agent_id_holding_lk = ai_h.agent_id

<table>
<thead>
<tr>
<th>Hold App</th>
<th>Holder</th>
<th>Wait App</th>
<th>Waiter</th>
<th>Hold Mode</th>
<th>Obj Typ</th>
<th>TabName</th>
<th>Schema</th>
<th>waiting</th>
</tr>
</thead>
<tbody>
<tr>
<td>db2bp.exe</td>
<td>CEATON</td>
<td>db2bp.exe</td>
<td>USER2</td>
<td>X</td>
<td>Row</td>
<td>T1</td>
<td>CEATON</td>
<td>15</td>
</tr>
<tr>
<td>db2bp.exe</td>
<td>CEATON</td>
<td>db2bp.exe</td>
<td>USER1</td>
<td>X</td>
<td>Row</td>
<td>T1</td>
<td>CEATON</td>
<td>6</td>
</tr>
</tbody>
</table>
Excessive Sorting

Show the sort time, and wait time for all sorts by connection

```
SELECT
  APPLICATION_HANDLE AS APP_HDL,
  SUBSTR(CLIENT_USERID,1,10) AS USERID,
  TOTAL_SECTION_SORTS AS NUM_SORTS,
  TOTAL_SECTION_SORT_TIME AS TOTAL_TIME,
  TOTAL_SECTION_SORT_PROC_TIME AS SORT_TIME,
  TOTAL_SECTION_SORT_TIME - TOTAL_SECTION_SORT_PROC_TIME AS WAIT_TIME
FROM TABLE(MON_GET_CONNECTION(NULL,-1))
```

<table>
<thead>
<tr>
<th>APP_HDL</th>
<th>USERID</th>
<th>NUM_SORTS</th>
<th>TOTAL_TIME</th>
<th>SORT_TIME</th>
<th>WAIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>CEATON</td>
<td>36</td>
<td>7579</td>
<td>7495</td>
<td>84</td>
</tr>
</tbody>
</table>
Top Consuming Transactions

Show the transactions with the most CPU and most Wait Time

SELECT
  APPLICATION_HANDLE AS APP_HDL,
  SUBSTR(CLIENT_USERID,1,10) AS USERID,
  TOTAL_RQST_TIME,
  TOTAL_CPU_TIME,
  TOTAL_WAIT_TIME,
  CLIENT_IDLE_WAIT_TIME
FROM TABLE(MON_GET_UNIT_OF_WORK(NULL,-1))
## New in FP1

- Unit of Work monitor also includes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL_COMPILE_TIME</td>
<td>BIGINT</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_APP_ROLLBACKS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>INT_ROLLBACKS</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_RUNSTATS_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_RUNSTATS_PROC_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_RUNSTATS</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_REORG_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_REORG_PROC_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_COMPILE_PROC_TIME</td>
<td>TOTAL_REORGS</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_ACT_TIME</td>
<td>TOTAL_LOAD_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ACT_TIME</td>
<td>TOTAL_LOAD_TIME</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_ACT_TIME</td>
<td>TOTAL_LOAD_PROC_TIME</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ACT_TIME</td>
<td>TOTAL_LOADS</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>CAT_CACHE_INSERTS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>CAT_CACHE_LOOKUPS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>PKG_CACHE_INSERTS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>PKG_CACHE_LOOKUPS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>THRESH_VIOLATIONS</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>NUM_LW_THRESH_EXCEEDED</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>UOW_LOG_SPACE_USED</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>ADDITIONAL_DETAILS</td>
<td>BLOB(100K)</td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>TOTAL_ROUTE_TIME</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
</tbody>
</table>
Monitoring Health And Status With SQL Select Statements
Monitoring Table Access

Show the most active tables

```sql
SELECT SUBSTR(TABSCHEMA,1,10) AS SCHEMA,
       SUBSTR(TABNAME,1,20) AS NAME,
       TABLE_SCANS,
       ROWS_READ,
       ROWS_INSERTED,
       ROWS_DELETED
FROM TABLE(MON_GET_TABLE('', '', -1))
ORDER BY ROWS_READ DESC
FETCH FIRST 5 ROWS ONLY
```

<table>
<thead>
<tr>
<th>SCHEMA</th>
<th>NAME</th>
<th>TABLE_SCANS</th>
<th>ROWS_READ</th>
<th>ROWS_INSERTED</th>
<th>ROWS_DELETED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEATON</td>
<td>WIKI_ACTIONS</td>
<td>14</td>
<td>6608</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSTABLES</td>
<td>16</td>
<td>6161</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CEATON</td>
<td>WIKI_VISITORS</td>
<td>12</td>
<td>5664</td>
<td>0</td>
<td>70</td>
</tr>
<tr>
<td>SYSTOOLS</td>
<td>HMON_ATM_INFO</td>
<td>19</td>
<td>3627</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSINDEXES</td>
<td>0</td>
<td>348</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Monitoring Index Access

- **Show me the indexes that have been most active**
  - Metrics will only be returned for indexes on tables that have been accessed since the database was activated.

```sql
SELECT
  SUBSTR(TABSHEMA,1,10) AS SCHEMA,
  SUBSTR(TABNAME,1,20) AS NAME,
  IID, NLEAF, NLEVELS,
  INDEX_SCANS,
  KEY_UPDATES,
  BOUNDARY_LEAF_NODE_SPLITS +
  NONBOUNDARY_LEAF_NODE_SPLITS AS PAGE_SPLITS
FROM TABLE(MON_GET_INDEX("","",-1))
ORDER BY INDEX_SCANS DESC
FETCH FIRST 5 ROWS ONLY
```

<table>
<thead>
<tr>
<th>SCHEMA</th>
<th>NAME</th>
<th>IID</th>
<th>NLEAF</th>
<th>NLEVELS</th>
<th>INDEX_SCANS</th>
<th>UPDATES</th>
<th>SPLITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTOOLS</td>
<td>HMON_ATM_INFO</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>754</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSUSERAUTH</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>425</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSPLANAUTH</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>192</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSTABLES</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>186</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSINDEXES</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>145</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
SQL to View Notification Log

- Show me all the Critical and Error messages in the last 24 hours

```sql
SELECT TIMESTAMP, SUBSTR(MSG,1,400) AS MSG
FROM SYSIBMADM.PDLOGMSGS_LAST24HOURS
WHERE MSGSEVERITY IN ('C','E')
ORDER BY TIMESTAMP DESC
```

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>MSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-03-16-09.41.47.673002</td>
<td>ADM6044E  The DMS table space &quot;SMALLTBSP&quot; (ID &quot;2&quot;) is full. If this is an autoresize or automatic storage DMS tablespace, the maximum table space size may have been reached or the existing containers or storage paths cannot grow any more. Additional space can be added to the table space by either adding new containers or extending existing ones using the ALTER TABLESPACE SQL statement.</td>
</tr>
</tbody>
</table>
SQL to View Database History

- Show the average and maximum time taken to perform full backups

```
SELECT AVG(TIMESTAMPDIFF(4,CHAR(TIMESTAMP(END_TIME) - TIMESTAMP(START_TIME)))) AS AVG_BTIME,
       MAX(TIMESTAMPDIFF(4,CHAR(TIMESTAMP(END_TIME) - TIMESTAMP(START_TIME)))) AS MAX_BTIME
FROM SYSIBMADM.DB_HISTORY
WHERE OPERATION = 'B'
AND OPERATIONTYPE = 'F'
```

<table>
<thead>
<tr>
<th>START_TIME</th>
<th>SQLCODE</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>20061114093635</td>
<td>-204</td>
<td>DROP TABLESPACE IBMDB2SAMPLEXML</td>
</tr>
<tr>
<td>20061218125352</td>
<td>-1422</td>
<td>CREATE REGULAR TABLESPACE SMALLTSP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AVG_BTIME</th>
<th>MAX_BTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
</tr>
</tbody>
</table>
Finding the Log Hog

Display information about the application that currently has the oldest uncommitted unit of work

```
SELECT AI.APPL_STATUS as Status, 
    SUBSTR(AI.PRIMARY_AUTH_ID,1,10) AS "Authid", 
    SUBSTR(AI.APPL_NAME,1,15) AS "Appl Name", 
    INT(AP.UOW_LOG_SPACE_USED/1024/1024) 
    AS "Log Used (M)", 
    INT(AP.APPL_IDLE_TIME/60) AS "Idle for (min)", 
    AP.APPL_CON_TIME AS "Connected Since"
FROM SYSIBMADM.SNAPDB DB, 
    SYSIBMADM.SNAPAPPL AP, 
    SYSIBMADM.SNAPAPPL_INFO AI 
WHERE AI.AGENT_ID = DB.APPL_ID_OLDEST_XACT 
AND AI.AGENT_ID = AP.AGENT_ID;
```
What’s New in 9.7 FP1

- **CREATE EVENT MONITOR FOR PACKAGE CACHE**
  - records events from both dynamic and static SQL when they are flushed from package cache
  - Information collected same as `MON_GET_PKG_CACHE_STMT`

- **Can view the information from event monitor as**
  - An XML document created by the new `EVMON_FORMAT_UE_TO_XML` table function
  - Relational tables populated by the new `EVMON_FORMAT_UE_TO_TABLES` procedure

Must run `db2updv97`
New Lightweight Lock Monitors

- MON_GET_APPL_LOCKWAITS table function
  - Returns information about the locks that all applications are waiting to acquire

- MON_GET_LOCKS table function
  - Returns a list of all locks held

- MON_FORMAT_LOCK_NAME table function
  - Formats the internal lock name and returns details about the lock in a row-based format. Each row consists of a key-value pair pertaining to a particular lock.

- MON_LOCKWAITS View
  - Returns information about agents working on behalf of applications that are waiting to obtain locks in the currently connected database.

- Deprecated:
  - SNAPLOCK, SNAPLOCKWAIT, LOCKS_HELD, LOCKWAITS views
Lock Wait Query

```sql
select substr(ai_h.appl_name,1,10) as "Hold App",
    substr(ai_h.primary_auth_id,1,10) as "Holder",
    substr(ai_w.appl_name,1,10) as "Wait App",
    substr(ai_w.primary_auth_id,1,10) as "Waiter",
    lw.lock_mode as "Hold Mode",
    lw.lock_object_type as "Obj Type",
    substr(lw.tabname,1,10) as "TabName",
    substr(lw.tabschema,1,10) as "Schema",
    timestampdiff(2, char(lw.snapshot_timestamp - lw.lock_wait_start_time))
    as "waiting (s)"
from sysibmadm.snapappl_info ai_h,
    sysibmadm.snapappl_info ai_w, sysibmadm.snaplockwait lw
where lw.agent_id = ai_w.agent_id
and   lw.agent_id_holding_lk = ai_h.agent_id
```

Who is holding the lock:
- db2bp.exe CEATON
- db2bp.exe USER2

Who is waiting on the lock:
- db2bp.exe CEATON
- db2bp.exe USER1

How long is the wait:
- db2bp.exe CEATON 15
- db2bp.exe CEATON 6
New Lightweight Version

```
select substr(HLD_APPLICATION_NAME,1,10) as "Hold App",
        substr(HLD_USERID,1,10) as "Holder",
        substr(REQ_APPLICATION_NAME,1,10) as "Wait App",
        substr(REQ_USERID,1,10) as "Waiter",
        LOCK_MODE as "Hold Mode",
        LOCK_OBJ_TYPE as "Obj Type",
        TABNAME,1,10) as "TabName",
        TABSCHEMA,1,10) as "Schema",
        LOCK_WAIT_ELAPSED_TIME as "waiting (s)"
from SYSIBMADM.MON_LOCKWAITS;
```

Also available:
```
REQ_STMT_TEXT
HLD_CURRENT_STMT_TEXT
LOCKNAME
```
## MON_FORMAT_LOCK_NAME

```
SELECT SUBSTR(NAME,1,20) AS NAME,
       SUBSTR(VALUE,1,50) AS VALUE
FROM TABLE(
           MON_FORMAT_LOCK_NAME(
                         '0000000E000000000000B00C152'))

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCK_OBJECT_TYPE</td>
<td>ROW</td>
</tr>
<tr>
<td>TBSP_NAME</td>
<td>PRODTBSPACE1</td>
</tr>
<tr>
<td>TABSCHEMA</td>
<td>CEATON</td>
</tr>
<tr>
<td>TABNAME</td>
<td>PRODUCTS</td>
</tr>
<tr>
<td>ROWID</td>
<td>00 00 00 0C 00 C1</td>
</tr>
</tbody>
</table>
```
Other New Admin Views in FP1

- MON_BP_UTILIZATION
- MON_TBSP_UTILIZATION
- MON_LOCKWAITS
- MON_PKG_CACHE_SUMMARY
- MON_CURRENT_SQL
- MON_CURRENT_UOW
- MON_SERVICE_SUBCLASS_SUMMARY
- MON_WORKLOAD_SUMMARY
- MON_CONNECTION_SUMMARY
- MON_DB_SUMMARY
Data Studio Health Monitor
Data Studio Health Monitor

- **Included with DB2**

- **View system health at a glance.**
  - Visualize warnings and problem areas instantly
  - Configure alert thresholds for health indicators, such as data server status and space utilization

- **Browse alert history**
  - Collect and retain alert history for up to seven days.
  - Display alert statistics by time period, database, or alert type.

- **Manage current application connections.**
  - Track information such as rows read and idle time for currently connected applications.
  - Verify that applications can access the database.
  - Force applications to enhance system performance.

- **View the current state of the table spaces of your database.**
  - View information such as state, total size, and current utilization for the table spaces of your databases.

- **View the status of utilities operating on your database.**
Quickly Visualize High Level Database Status

![Data Studio Health Monitor](image)

### Alert Severity

- All
- Critical Alerts (0)
- Warning Alerts (0)
- Critical and Warning Alerts (0)

### Data Source

<table>
<thead>
<tr>
<th>All</th>
<th>Critical</th>
<th>Warning</th>
<th>Data Server Status</th>
<th>Connections</th>
<th>Storage</th>
<th>Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE</td>
<td>0</td>
<td>0</td>
<td>Green</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

2 total items

10 Items per page

Page 1 of 1

© 2011 IBM Corporation
Manage Current Application Connections

- See the state of each application connected to the database

The application connections that are listed are currently active for the selected database. To improve performance, you can disconnect applications.

<table>
<thead>
<tr>
<th>Agent ID</th>
<th>Name</th>
<th>Application ID</th>
<th>Authorization ID</th>
<th>Status</th>
<th>Client Product</th>
<th>Client ID</th>
<th>Idle Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5432</td>
<td>db2bp.exe</td>
<td>*LOCAL.DB2.11...</td>
<td>CHRIS</td>
<td>Waiting for a Lock</td>
<td>SQL09070</td>
<td>2432</td>
<td>0</td>
</tr>
<tr>
<td>5458</td>
<td>db2bp.exe</td>
<td>*LOCAL.DB2.11...</td>
<td>CHRIS</td>
<td>Unit of Work Waiting</td>
<td>SQL09070</td>
<td>7360</td>
<td>184</td>
</tr>
<tr>
<td>5410</td>
<td>db2jcc_application</td>
<td>9.23.36.75.443...</td>
<td>CHRIS</td>
<td>Unit of Work Waiting</td>
<td>JCC03580</td>
<td>0</td>
<td>590</td>
</tr>
<tr>
<td>5416</td>
<td>db2fw1</td>
<td>*LOCAL.DB2.11...</td>
<td>CHRIS</td>
<td>Database Connection Comple...</td>
<td>JCC03580</td>
<td>0</td>
<td>593</td>
</tr>
<tr>
<td>5415</td>
<td>db2fw0</td>
<td>*LOCAL.DB2.11...</td>
<td>CHRIS</td>
<td>Database Connection Comple...</td>
<td>JCC03580</td>
<td>0</td>
<td>593</td>
</tr>
</tbody>
</table>
Sort Applications by Rows Read, Written, Idle Time

- Easy to see if applications are stuck waiting and for how long
- Are there applications doing table scans?
View Tablespace Utilization and Container locations

- Show free space and space consumed for each tablespace
- Drill down to see the containers for each tablespace

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>Content type</th>
<th>State</th>
<th>Utilization</th>
<th>Free Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SYSCATSPACE</td>
<td>DMS</td>
<td>ANY</td>
<td>NORMAL</td>
<td>81.47%</td>
<td>18208</td>
</tr>
<tr>
<td>2</td>
<td>USERSPACE1</td>
<td>DMS</td>
<td>LARGE</td>
<td>NORMAL</td>
<td>44.88%</td>
<td>17920</td>
</tr>
<tr>
<td>4</td>
<td>IBMDB2SAMPLE...</td>
<td>DMS</td>
<td>LARGE</td>
<td>NORMAL</td>
<td>35.43%</td>
<td>20992</td>
</tr>
<tr>
<td>3</td>
<td>IBMDB2SAMPLE...</td>
<td>DMS</td>
<td>LARGE</td>
<td>NORMAL</td>
<td>21.25%</td>
<td>25600</td>
</tr>
<tr>
<td>5</td>
<td>SYSTOOLSPACE</td>
<td>DMS</td>
<td>LARGE</td>
<td>NORMAL</td>
<td>2.73%</td>
<td>31840</td>
</tr>
<tr>
<td>1</td>
<td>TEMPSPACE1</td>
<td>SMS</td>
<td>SYSTEMP</td>
<td>NORMAL</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>6</td>
<td>SYSTOOLSTMPS...</td>
<td>SMS</td>
<td>USRTEMP</td>
<td>NORMAL</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Optim Performance Manager
See the snapshot history
Optim Performance Manager - Overview
Dashboard Time Slider and Time Controls

Recent Button
Displays the latest collected data. Content is refreshed when a new sample is taken.

Clock Button
Indicates the time remaining until the dashboard is refreshed. Click to start or stop refresh.

Zoom In
Shows less of the timeline to be able to position the time slider to specific data.

Back and Forward to Data Point
These two arrows move the time slider to the previous or next data point.

End Time
Specifies the latest time on the time slider clock.

History Button
Displays data based on the position of the time slider on the timeline.

Aggregation Level
Indicates the level of granularity of the data.

Missing Data Points
Missing lines indicate that no monitored data points are available.

Data Point Line
Blue lines indicate points in time for which monitoring data is available.

Hide Button
Click to hide time controls.

Time slider & line
Indicates the range of time shown in the dashboard.

Duration
Specifies how much data is shown at one time in the dashboard.
“Diagnose” using OPM
Extended Insight Analysis Dashboard

- Alert by the connection attribute grouping
- Graphically represent your trouble spot data
- View any and all information through time slider controls
- Set OPM to run in your Tivoli Enterprise Portal Console
- Drill down into end-to-end details
- Show comparisons of the client, data server and network times to see where the bulk of the time is being spent
Hover KPI info

Buffer Pool Size: 314
Time: 04/16 18:46:00

Hover mouse over to see point in time values for KPIs
Easy collaboration

Email alerting from OPM when a metric has been exceeded

From: Thuan Bui/Santa Teresa/IBM@IBMUS
To: Thuan Bui/Santa Teresa/IBM@IBMUS
Date: 04/02/2010 01:18:53 PM
Subject: IBM Optim Performance Manager.

IBM Optim Performance Manager.

DB2 threshold violation was detected at [Apr 2, 2010, 1:52:14 PM PDT] on DB2 system 'LOCALHOST_60001_SDB'.

Details:

Counter : AVG_NUMBER_ROWS_SELECTED_ROW, Average number of rows read per selected row (rows)
Severity : PROBLEM
Owner : DB2ADMIN

Current value : 26.83
Warning level : > 5.0
Problem level : > 10.0

Violator:
Database name : G5DB
Database path : C:\DB2_01\NODE0000\SQL00002
Total connections to database : 20
Database status : Database is active
Database connection time : 2010-04-01 18:39:10.253002
Partition name : PART0
Partition number : 0
OPM quickly identifies an issue

OPM alerts on the glass and email
Get end-to-end application insight

Contributors but not causal to the slowdown

Data Server Time Percentage very high
OPM Dashboards Provide Direction

Alerts with details

Switch to Workload Dashboard
Drilldown Workload To Diagnose Further

Number of transactions dramatically increased, causing KPIs to also increase.
Deeper Diagnosis: Bufferpool and I/O Drilldowns

Hit ratios take a dive

Low buffer pool size

Logical and Physical I/O are 1:1 indicating every page read requires an I/O
Iterative changes

- 20 pages to 100 pages and then to 200 pages
- Data BP Hit Ratio increasing
- Logical and Physical I/O starting to separate
# Performance Manager Packaging

<table>
<thead>
<tr>
<th>Feature</th>
<th>Data Studio Health Monitor (included in DB2)</th>
<th>Optim Performance Manager (included in DB2 AESE)</th>
<th>Optim Performance Manager Extended Edition</th>
<th>DB2 Performance Optimization Feature or AESE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alerts and notifications</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Overview health summary</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Diagnostic dashboards</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Standard reporting</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OPM privileges, OQT integration</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Extended Insight</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ITCAM, pureQuery integration</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DB2 WLM administration tooling</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DB2 WLM feature</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Summary

- Monitoring in DB2 is changing rapidly
  - Moving to time spent and time waiting metrics
  - Each release and fixpack typically adds more monitor elements you can leverage

- Much of the support is targeted at helping tool vendors
  - However, you can use SQL to get at the same info