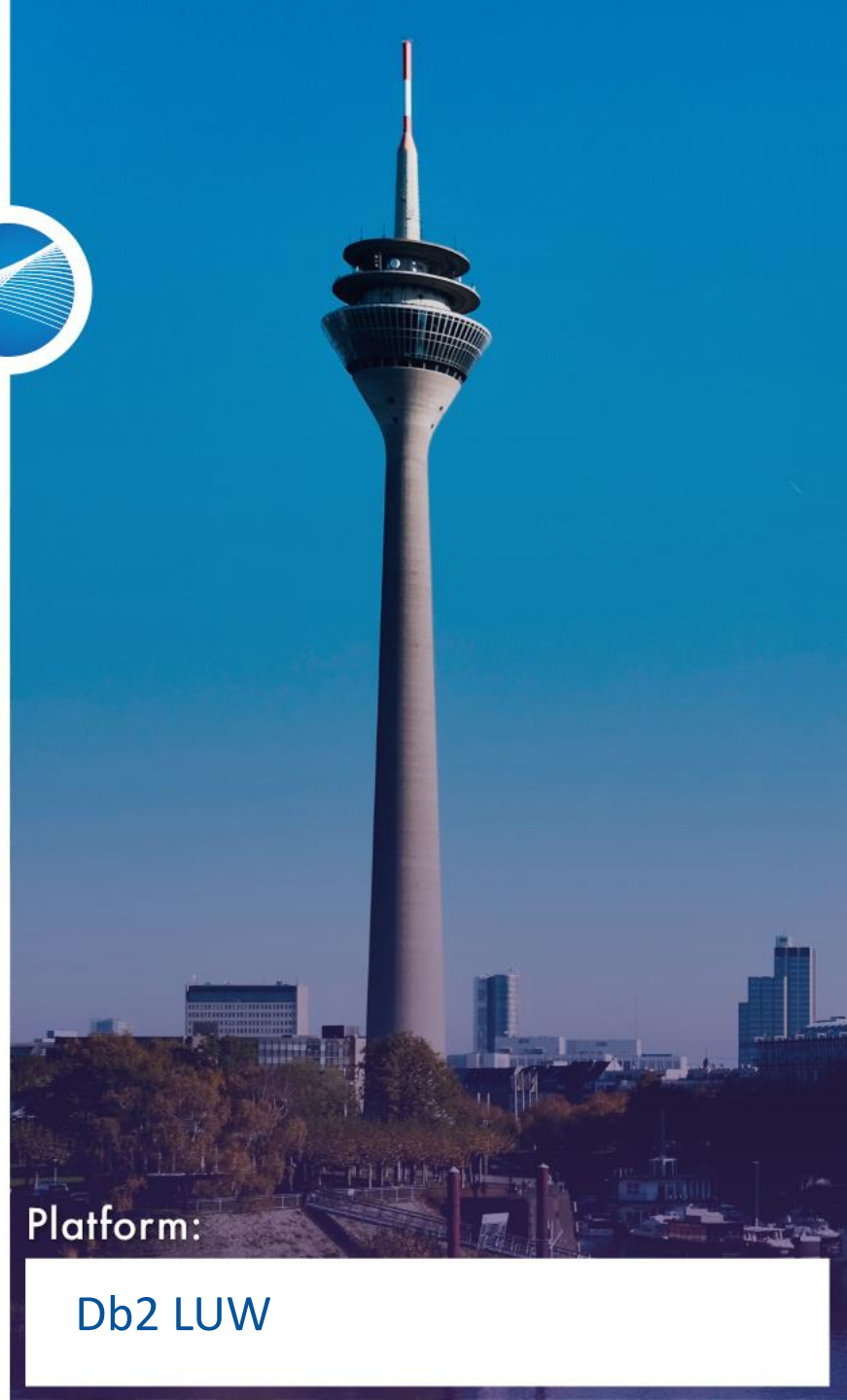
IDUG

2025

EMEA Db2 TECH CONFERENCE

Dusseldorf | October 26 - 30



Db2 Performance and Tuning

Phil Gunning, IBM Gold Consultant, MBA, CISSP

Gunning Technology Solutions, LLC

Session Code: C9

Abstract

- This presentation will cover some recommended automatic settings for DBM and DB CFG parameters
- STMM and AUTOMATIC settings have matured over various releases
- Help new Db2 customers get up to speed fast with Db2 initial setup and with adequate performance from the start
- Still need to know which parameters should be AUTOMATIC and which should not be, if any and the knowledge to determine what's adequate
- I still find customers with old settings from prior database upgrades
- More and more companies are trying to run Db2 without a trained Db2 DBA
  - No monitoring being done
  - No knowledge of settings
- Many companies still get good Db2 performance even in these situations

Abstract

- Understanding all the included Db2 Monitoring tools
- Snapshot, Request and Activity based parameters
- db2pd
- dmctop
  - MON_GET table functions
    - db2top
    - dsmtop
- MON_GET table functions
- DB2MON
- Identifying and capturing suboptimal SQL using MONREPORT, DB2MON,dmctop
- Capture suboptimal SQL and use EXPLAIN_FROM_SECTION and DB2 Design Advisor
- Quick Db2 Health Checks to use

Db2 Provided Monitoring Tools

- DB2MON
- MONREPORT reporting modules
- MON_GET table function facilities
  - Request, Activity, Time based
  - Intended to replace snapshot monitoring
- db2top and dsmtop
  - – Deprecated
- dmctop -- Comes with Db2, replaces db2top and dsmtop
  - Db2 11.5.6
  - Download if prior to DB2 11.5.6
  - Text based
  - Enhanced functionality
    - Throughput
    - Lock Contention reporting (holders and waiters)
- db2pd
  - Lightweight overhead
  - Command driven
  - Many options

MONREPORT

- The MONREPORT reporting module has been around since Db2 version 9.7
- Prior to that one could obtain similar data for top dynamic SQL via an SQL Query with a ranking function
- MONREPORT has several reporting options
- Database Level
  - DBSUMMARY
  - LOCKWAIT
- SQL Related
  - CURRENTSQL
  - PKGCACHE
- APPLICATION
  - CURRENTAPPS
  - CONNECTION

Troubleshooting and Tuning Methodology with MONREPORT

- Identify CURRENTSQL and PKGCACHE entries with metric ranking
- Identify high cost SQL from metrics
  - CPU
  - IO
    - Rows read/rows written
  - Lockwait
  - Number of Executions
- Select executable ID of item of interest based on ranking
- Feed the executable ID to the EXPLAIN_FROM_SECTION stored procedure
- Explain the plan using DB2EXFMT
- Look for table scans, sorts, index scans
- Rewrite or use Design Advisor to review possible index improvements
- Iterate

Monitoring and Tuning Approach

- The next few slides will show the monitoring and tuning approach I use on a daily basis to support the various clients that outsource to us
- Call MONREPORT.CURRENTSQL
- Review the Output for the metrics that are indicate high cost

MONREPORT CURRENTSQL Report


```
Result set 1
--------------

TEXT
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Monitoring report - current SQL
------------------------------------------------------------------------------
Database:               HSPRD
Generated:              09/26/2025 12:54:27
-- Command options --
MEMBER:                 All

==============================================================================
Part 1 - Summaries by 'top' metrics

Top 10 current activities by TOTAL_CPU_TIME
------------------------------------------------------------------------------
ACTIVITY  UOW_ID  APPLICATION  TOTAL_CPU  STMT_TEXT
_ID               _HANDLE      _TIME
--------  ------  -----------  ---------  ------------------------------------
1         3       13075        45692786   SELECT "CLAIM_ID",     "CUR_STATUS",
1         40      12975        24425471   select (b.clm_id) as "Claim ID" , crf.
1         1       13295        0          CALL monreport.currentsql()
63        1       13295        0          SELECT ARRAY_AGG(A.ACTIVITY_ID ORDER B

Top 10 current activities by ROWS_READ
------------------------------------------------------------------------------
ACTIVITY  UOW_ID  APPLICATION  ROWS_READ      STMT_TEXT
_ID               _HANDLE
--------  ------  -----------  -------------  ----------------------------------
1         3       13075        27265357       SELECT "CLAIM_ID",     "CUR_STATUS
1         40      12975        22816408       select (b.clm_id) as "Claim ID" ,
1         1       13295        0              CALL monreport.currentsql()
63        1       13295        0              SELECT ARRAY_AGG(A.ACTIVITY_ID ORD
Standard input
```

CURRENTSQL Continued


```
================================================================================
Part 2 - Overall ranking of activities

ACTIVITY  UOW_ID  APPLICATION  TOTAL_CPU  ROWS_READ      DIRECT_READS +
_ID               _HANDLE      _TIME                     DIRECT_WRITES
--------  ------  -----------  ---------  -------------  ----------------------
1         3       13075        1          1              1
1         5       13295        2          2              1
63        5       13295        2          2              1

================================================================================
Part 3 - Complete statement text for activities

--------------------------------------------------------------------------------
ACTIVITY_ID/UOW_ID/APPLICATION_HANDLE = 1/3/13075

SELECT "CLAIM_ID",     "CUR_STATUS",     "STATUS_DT",     "PROCESSOR_ID" FROM (S
ELECT DISTINCT CEV.CEV_CLM_ID AS CLAIM_ID, CASE     WHEN TRIM(CEV.CEV_TYPE) = 'P
D' AND CSEV.CSEV_CSV_CLM_ID IS NULL THEN 'Paid In Full'  WHEN TRIM(CEV.CEV_TYPE)
 = 'PD' AND CSEV.CSEV_CSV_CLM_ID IS NOT NULL THEN 'Partial Denial'     WHEN CEV.
CEV_RUN_ID IS NULL AND TRIM(CEV.CEV_TYPE) = 'DN' THEN 'Full Denial-INV'     WHEN
 CEV.CEV_RUN_ID IS NOT NULL AND TRIM(CEV.CEV_TYPE) = 'DN' THEN 'Full Denial'
 ELSE INITCAP(TCL.TCL_TYPE_DESC) END AS CUR_STATUS, CEV.CEV_EV_DT AS STATUS_DT,
CEV.CEV_USER_ID   AS PROCESSOR_ID  FROM HS.CEV_CLM_EV AS CEV JOIN HS.CDT_CLM_DT
      AS CDT ON CEV.CEV_CLM_ID = CDT.CDT_CLM_ID AND CDT.CDT_DT_QUAL = '050' AND
CDT.CDT_DT >= CURRENT_DATE - 2 YEAR JOIN HS.TCL_TYPE_CD_LOOKUP AS TCL  ON TRIM(C
EV.CEV_TYPE) = TCL.TCL_TYPE_CODE AND TCL.TCL_ID = 84 LEFT JOIN HS.CSEV_CSV_EV AS
 CSEV ON CEV.CEV_CLM_ID = CSEV.CSEV_CSV_CLM_ID
    AND CSEV.CSEV_DEL_TS IS NULL                                    AND TRIM(C
SEV.CSEV_TYPE) = 'DN'                                        AND CSEV.CSEV_ADD_TS
= (SELECT CSEV_ADD_TS FROM HS.CSEV_CSV_EV WHERE CSEV_DEL_TS IS NULL AND CSEV_CSV
_CLM_ID = CSEV.CSEV_CSV_CLM_ID AND CSEV_CSV_LN_NUM = CSEV.CSEV_CSV_LN_NUM ORDER
BY CSEV_ADD_TS DESC FETCH FIRST 1 ROWS ONLY WITH UR)  WHERE CEV.CEV_DEL_TS IS NU
LL AND CEV.CEV_ADD_TS = (SELECT cstat1.CEV_ADD_TS FROM HS.CEV_CLM_EV AS cstat1 W
HERE cstat1.CEV_CLM_ID = CEV.CEV_CLM_ID AND cstat1.CEV_DEL_TS IS NULL ORDER BY c
stat1.CEV_ADD_TS DESC FETCH FIRST 1 ROWS ONLY WITH UR) WITH UR) AS "tq_66FRGB1hy
"
Standard input
```

MONREPORT PKGCACHE


```
Result set 1
------------

TEXT
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Monitoring report - package cache
--------------------------------------------------------------------------------
Database:                               HSPRD
Generated:                              07/15/2024 14:42:48
-- Command options --
CACHE_INTERVAL:                         All statements
SECTION_TYPE:                           Dynamic and static SQL
MEMBER:                                 All

================================================================================
Part 1 - Summaries by 'top' metrics

Top 10 statements by TOTAL_CPU_TIME
--------------------------------------------------------------------------------
#    TOTAL_          STMT_TEXT
     CPU_TIME
--   -----------     ------------------------------------------------------------
1    537462816       SELECT "CLAIM_ID",     "LINE_NUM",     "LINE_ID",     "LINE_CUR
2    103516002       SELECT "Provider ID" AS "Provider_ID",     "Claim ID" AS "Claim
3    87041692        SELECT "ER_ID",     "CMS Submission Date" AS "CMS_Submission_Da
4    21239625        SELECT "Provider ID" AS "Provider_ID",     "Fee Schedule ID" AS
5    4679624         SELECT "Provider ID" AS "Provider_ID",     "Provider Contract N
6    3106885         SELECT "Provider ID" AS "Provider_ID",     "Contract Status" AS
7    2618724         SELECT "DTRR_Unique ID" AS "DTRR_Unique_ID",     "SLA Record ID
8    2427585         SELECT "SR_ID",     "First Service Request Comment" AS "First_S
9    2131747         SELECT "Provider ID" AS "Provider_ID",     "Affiliated Provider
10   2028886         SELECT     SOURCE,    CUS_REF_ID,    CUS_NAME,    ALT_REQ_ID,    RE
```

MONREPORT PKGCACHE


```
Top 10 statements by TOTAL_ACT_WAIT_TIME per exec
------------------------------------------------------------------------------
#   TOTAL_ACT    LOCK_WAIT    STMT_TEXT
    _WAIT_TIME   _TIME
--  -----------  -----------  ------------------------------------------------
1   7871         0            SELECT "CLAIM_ID",     "LINE_NUM",     "LINE_ID",
2   4697         0            SELECT "Provider ID" AS "Provider_ID",     "Claim
4   3003         0            SELECT "Provider ID" AS "Provider_ID",     "Fee Sc
12  1334         1334         SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT
8   1051         0            SELECT "SR_ID",     "First Service Request Comment
13  587          587          SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT
3   404          0            SELECT "ER_ID",     "CMS Submission Date" AS "CMS_
19  280          0            SELECT * FROM hs.SRAT_SR_ATTACHMENT WHERE SRAT_SR_
5   231          0            SELECT "Provider ID" AS "Provider_ID",     "Provid
20  153          0            INSERT INTO hs.GAAT_GA_ATTACHMENT  (GAAT_GA_ID, GA
```

MONREPORT PKGCACHE


```
Top 10 statements by I/O wait time
------------------------------------------------------------------------------
#    I/O                  STMT_TEXT
     wait time
--   -------------------  ------------------------------------------------------
1    3477                 SELECT "CLAIM_ID",     "LINE_NUM",     "LINE_ID",
14   2672                 DECLARE CSV_INFO CURSOR FOR SELECT CSV_LN_NUM ,CSV_SVP
2    2554                 SELECT "Provider ID" AS "Provider_ID",     "Claim ID"
4    2321                 SELECT "Provider ID" AS "Provider_ID",     "Fee Schedu
15   2148                 SELECT CSEV_TYPE INTO :H00349    FROM CSEV_CSV_EV A WH
16   1790                 SELECT A.CCRL_RULE_VAL_C, A.CCRL_RULE_VAL_D, A.CCRL_EF
41   1517                 SELECT INE_TYPE ,INE_EVR_CD ,INE_EV_DT ,INE_ADD_TS ,IN
18   1497                 DECLARE CSR_FOR_CUS_DT CURSOR FOR SELECT A.CLM_ID, C.C
42   1483                 SELECT CEV_ADD_TS, CEV_USER_ID INTO :H00166    , :H001
43   1455                 INSERT INTO hs.ATHA_ATH_ATTACHMENT (ATHA_ATH_ID, ATHA_
```

MONREPORT PKGCACHE


```
Top 10 statements by I/O wait time per exec
------------------------------------------------------------------------------
#    I/O                    STMT_TEXT
     wait time
--   -------------------    ------------------------------------------------
1    3477                   SELECT "CLAIM_ID",     "LINE_NUM",     "LINE_ID",
2    2554                   SELECT "Provider ID" AS "Provider_ID",     "Claim ID"
4    2321                   SELECT "Provider ID" AS "Provider_ID",     "Fee Schedu
8    1018                   SELECT "SR_ID",     "First Service Request Comment" AS
3    345                    SELECT "ER_ID",     "CMS Submission Date" AS "CMS_Subm
20   152                    INSERT INTO hs.GAAT_GA_ATTACHMENT  (GAAT_GA_ID, GAAT_T
5    137                    SELECT "Provider ID" AS "Provider_ID",     "Provider C
44   114                    SELECT "Authorization ID" AS "Authorization_ID",     "
45   107                    SELECT CLM_ID, CDT_DT, CDT_TO_DT, INS_INR_CD FROM hs.C
6    102                    SELECT "Provider ID" AS "Provider_ID",     "Contract S

==============================================================================
```

MONREPORT PKGCACHE


```
Part 2 - EXECUTABLE_IDs for statements in Part 1

#   EXECUTABLE_ID
--  --------------------------------------------------------------------------
1   x'00000001000000000000000000000184E4F000000000002202407151403551019 78'
2   x'00000001000000000000000000018405D000000000002202407151400256728 75'
3   x'00000001000000000000000000000186D410000000000022024071514200518879 7'
4   x'00000001000000000000000000000185C3B000000000002202407151409538480 43'
5   x'00000001000000000000000000000185FEB000000000002202407151413574747 71'
6   x'00000001000000000000000000001860BE000000000002202407151414427397 37'
7   x'00000001000000000000000000018649B000000000002202407151417368696 80'
8   x'00000001000000000000000000018405C000000000002202407151400235547 96'
9   x'00000001000000000000000000018614B000000000002202407151415193210 54'
10  x'00000001000000000000000000001664440000000000022024071511245030150 5'
11  x'00000001000000000000000000000184DA8000000000002202407151402565141 24'
12  x'00000001000000000000000000018548E000000000002202407151407084118 34'
```

EXPLAIN_FROM_SECTION

- The EXPLAIN_FROM_SECTION procedure can be used to explain SQL from the package cache or from a package cache event monitor
- Capture the SQL using the MONREPORT PKGCACHE module
- Use the Executable ID as input to EXPLAIN_FROM_SECTION
  - Uses actual access plan used during statement execution
  - Output stored in explain tables and can be formatted using db2exfmt

❖My favorite way

MONREPORT to EXPLAIN FROM SECTION STORED PROCEDURE

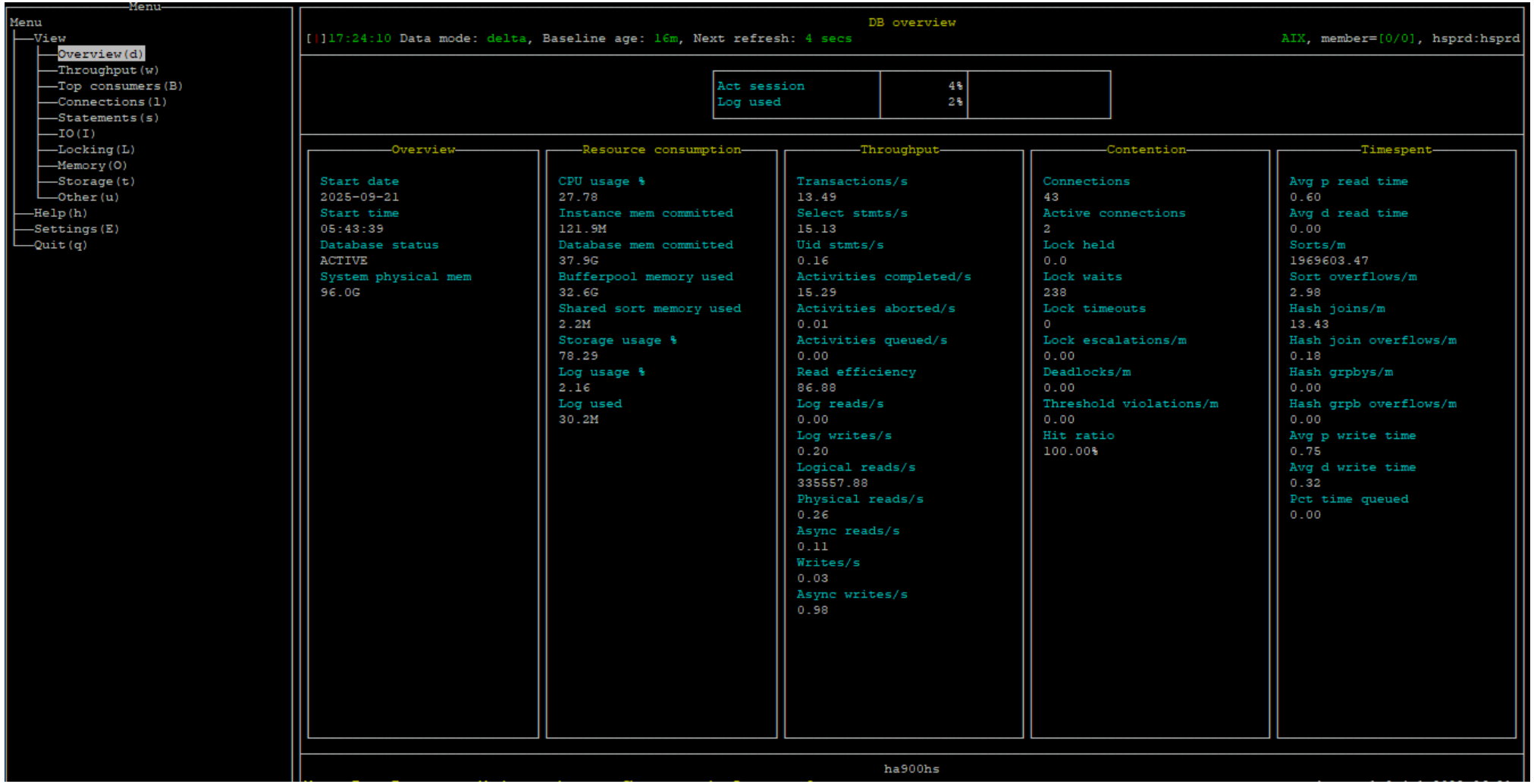

```
CALL EXPLAIN_FROM_SECTION
     ( x'00000001000000000000000000000184E4F000000000002202407151403551019 78',
      'M', NULL, 0, 'HSPRD', ?, ?, ?, ?, ? );
```

EXPLAIN FROM SECTION DB2EXFMT

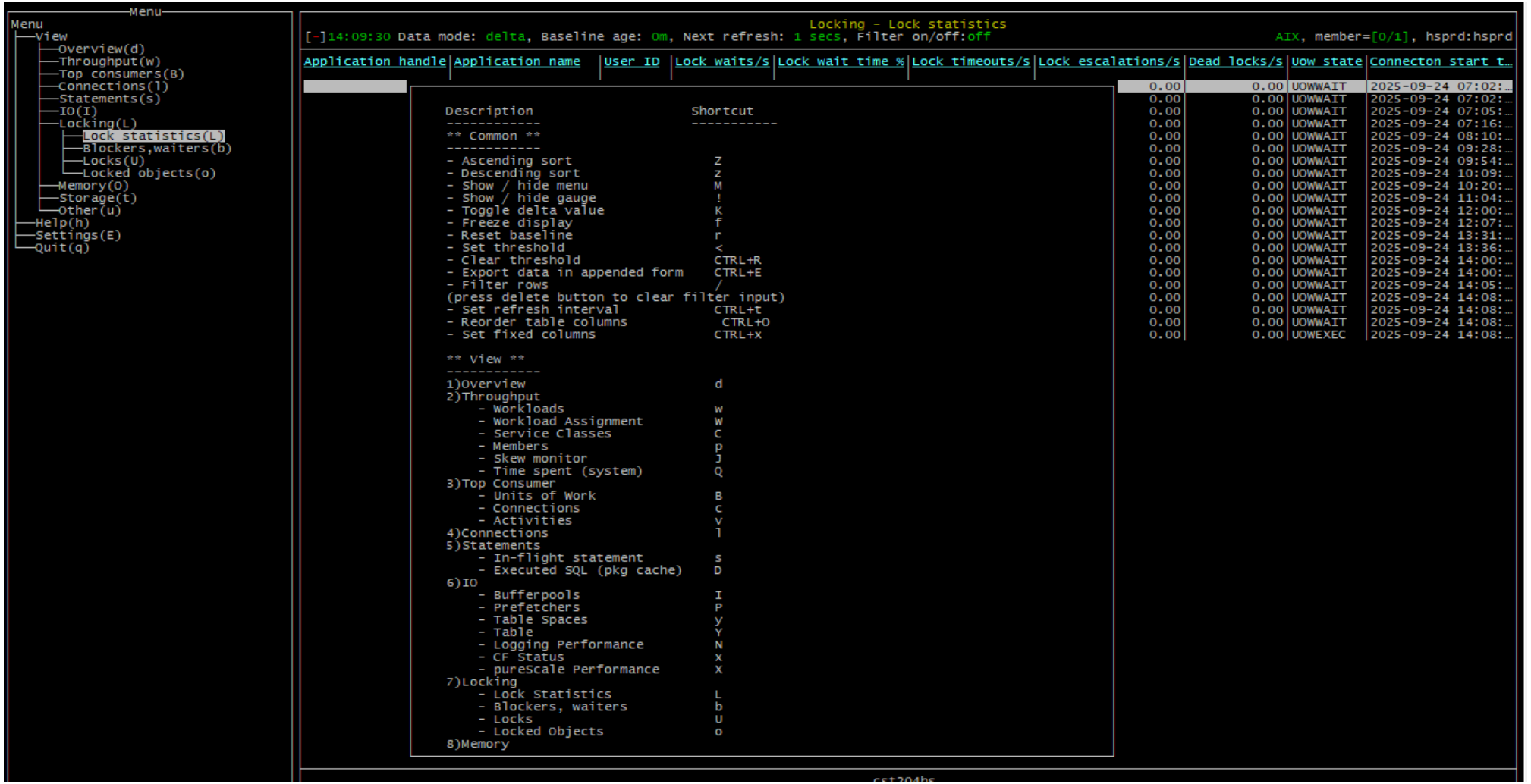

```
pgunning@[ahp202hs]:
(/home/pgunning)> db2exfmt
DB2 Universal Database Version 11.5, 5622-044 (c) Copyright IBM
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

Enter Database Name ==> hsprd
Connecting to the Database.
Connect to Database Successful.
Using SYSTOOLS schema for Explain tables.
Enter up to 26 character Explain timestamp (Default -1) ==>
Enter up to 128 character source name (SOURCE_NAME, Default %%)
Enter source schema (SOURCE_SCHEMA, Default %%) ==>
Enter section number (0 for all, Default 0) ==>
Enter outfile name. Default is to terminal ==> t134xfmt.out
Output is in t134xfmt.out.
Executing Connect Reset -- Connect Reset was Successful.
pgunning@[ahp202hs]:
```

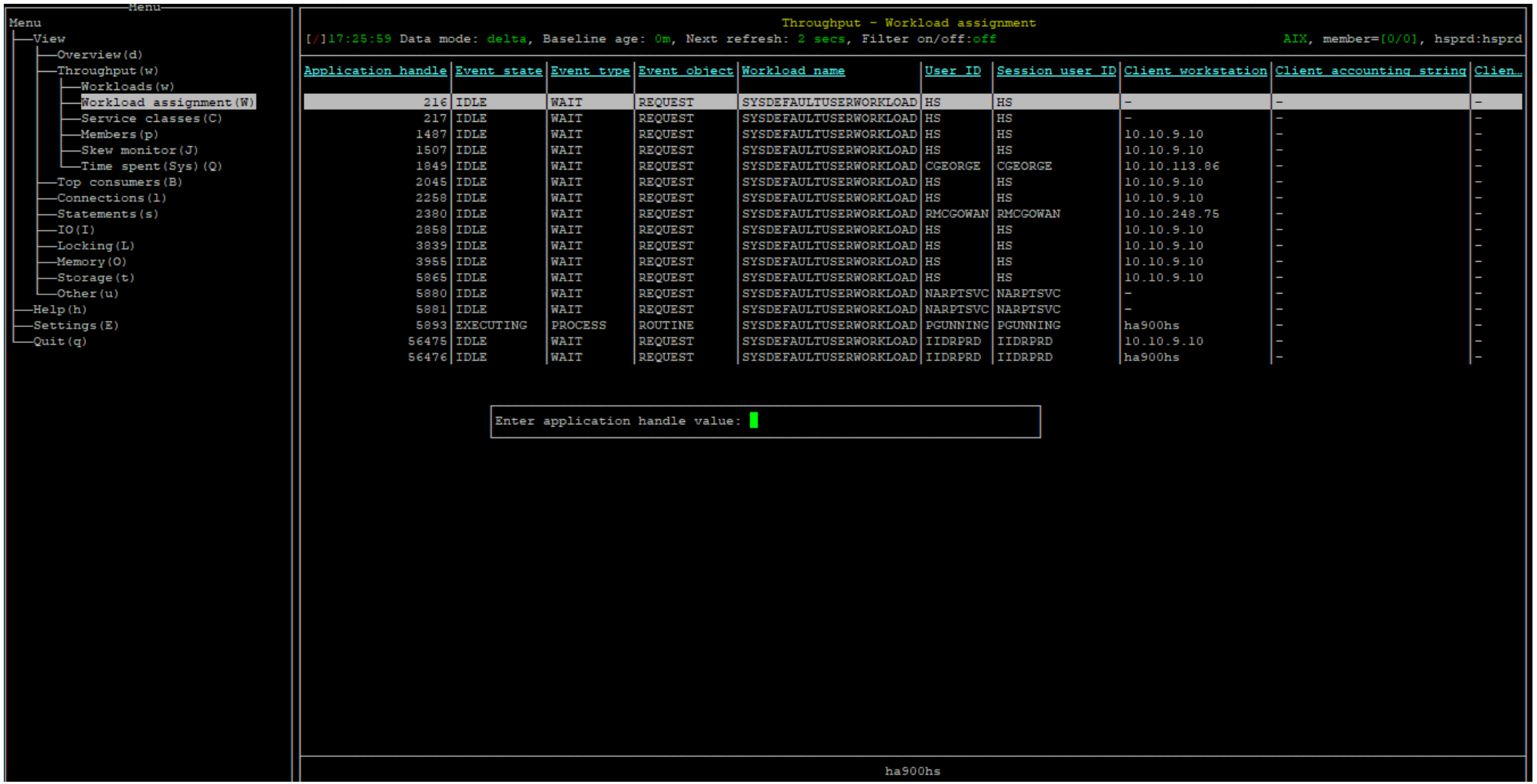
dmctop – Main Menu



dmctop shortcuts

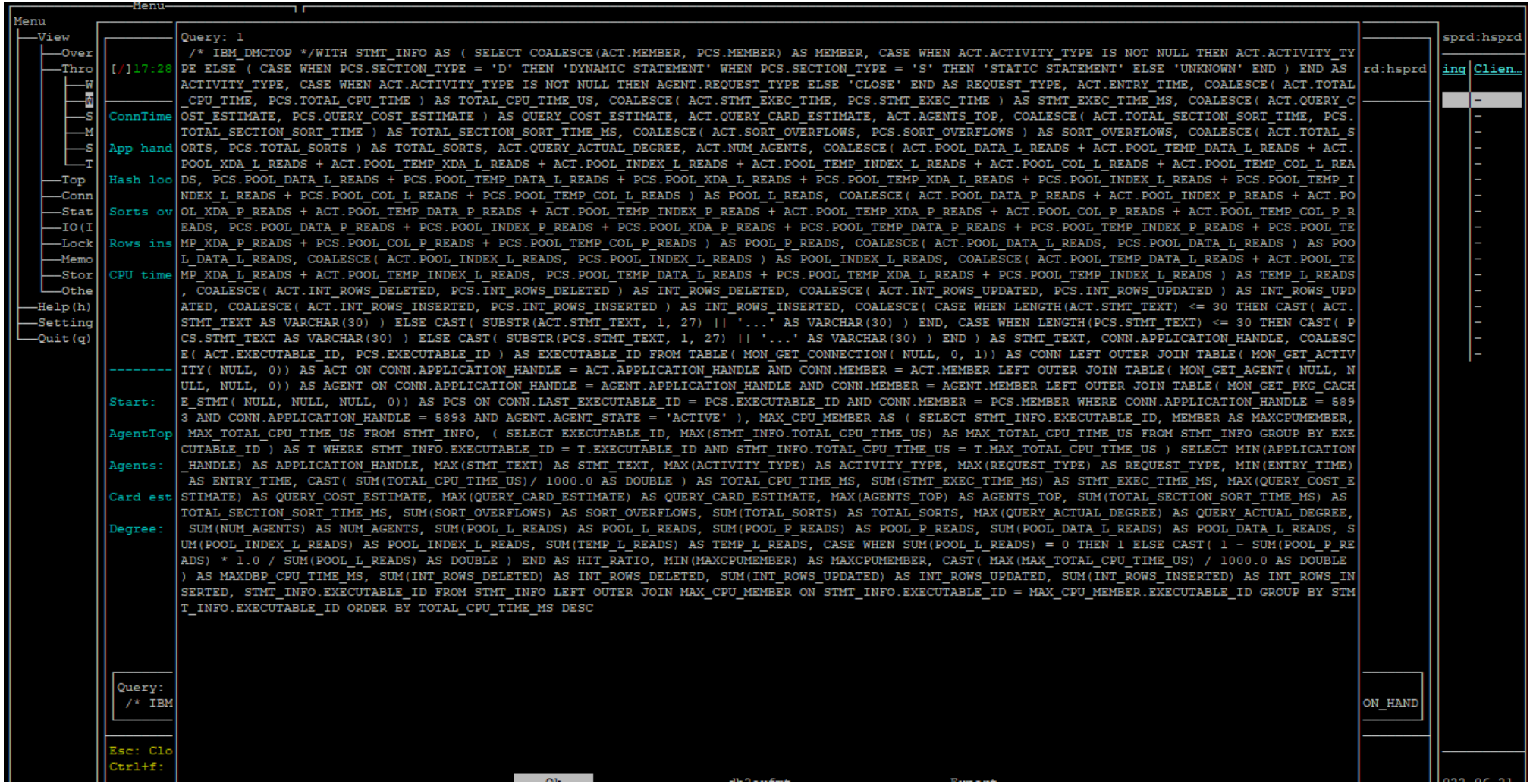


dmctop -- Throughput



dmctop – Application Drilldown

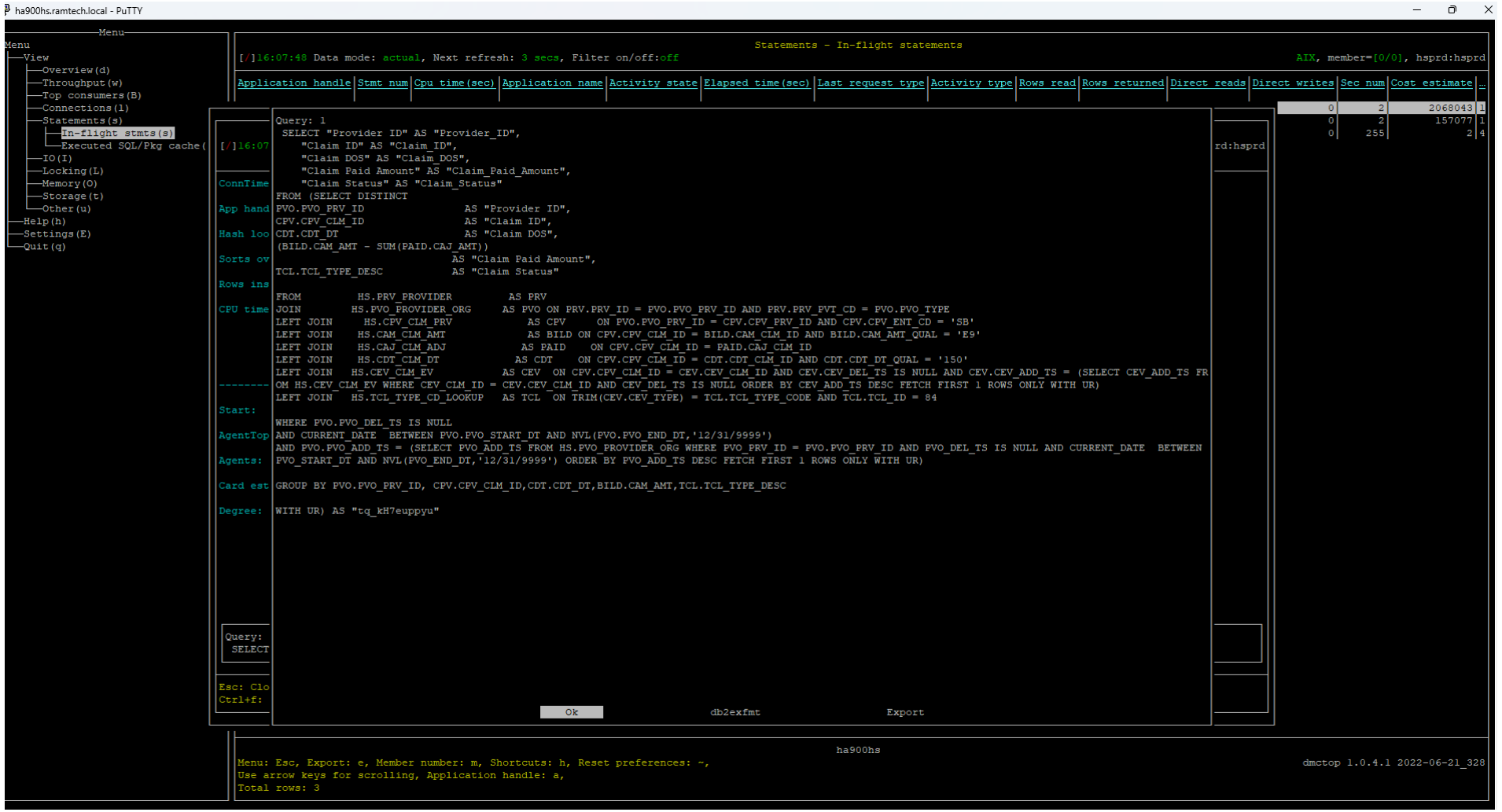
dmctop – SQL Associated with Application Handle



SQL Behind DMCTOP


```
/* IBM_DMCTOP */WITH STMT_INFO AS ( SELECT COALESCE(ACT.MEMBER, PCS.MEMBER) AS MEMBER, CASE WHEN ACT.ACTIVITY_TYPE IS NOT NULL THEN ACT.ACTIVITY_TYPE ELSE ( CASE WHEN PCS.SECTION_TYPE = 'D' THEN 'DYNAMIC STATEMENT' WHEN
PCS.SECTION_TYPE = 'S' THEN 'STATIC STATEMENT' ELSE 'UNKNOWN' END ) END AS ACTIVITY_TYPE, CASE WHEN ACT.ACTIVITY_TYPE IS NOT NULL THEN AGENT.REQUEST_TYPE ELSE 'CLOSE' END AS REQUEST_TYPE, ACT.ENTRY_TIME,
COALESCE( ACT.TOTAL_CPU_TIME, PCS.TOTAL_CPU_TIME ) AS TOTAL_CPU_TIME_US, COALESCE( ACT.STMT_EXEC_TIME, PCS.STMT_EXEC_TIME ) AS STMT_EXEC_TIME_MS, COALESCE( ACT.QUERY_COST_ESTIMATE, PCS.QUERY_COST_ESTIMATE ) AS
QUERY_COST_ESTIMATE, ACT.QUERY_CARD_ESTIMATE, ACT.AGENTS_TOP, COALESCE( ACT.TOTAL_SECTION_SORT_TIME, PCS.TOTAL_SECTION_SORT_TIME ) AS TOTAL_SECTION_SORT_TIME_MS, COALESCE( ACT.SORT_OVERFLOWS, PCS.SORT_OVERFLOWS ) AS
SORT_OVERFLOWS, COALESCE( ACT.TOTAL_SORTS, PCS.TOTAL_SORTS ) AS TOTAL_SORTS, ACT.QUERY_ACTUAL_DEGREE, ACT.NUM_AGENTS, COALESCE( ACT.POOL_DATA_L_READS + ACT.POOL_TEMP_DATA_L_READS + ACT.POOL_XDA_L_READS +
ACT.POOL_TEMP_XDA_L_READS + ACT.POOL_INDEX_L_READS + ACT.POOL_TEMP_INDEX_L_READS + ACT.POOL_COL_L_READS + ACT.POOL_TEMP_COL_L_READS, PCS.POOL_DATA_L_READS + PCS.POOL_TEMP_DATA_L_READS + PCS.POOL_XDA_L_READS +
PCS.POOL_TEMP_XDA_L_READS + PCS.POOL_INDEX_L_READS + PCS.POOL_TEMP_INDEX_L_READS + PCS.POOL_COL_L_READS + PCS.POOL_TEMP_COL_L_READS ) AS POOL_L_READS, COALESCE( ACT.POOL_DATA_P_READS + ACT.POOL_INDEX_P_READS +
ACT.POOL_XDA_P_READS + ACT.POOL_TEMP_DATA_P_READS + ACT.POOL_TEMP_INDEX_P_READS + ACT.POOL_TEMP_XDA_P_READS + ACT.POOL_COL_P_READS + ACT.POOL_TEMP_COL_P_READS, PCS.POOL_DATA_P_READS + PCS.POOL_INDEX_P_READS +
PCS.POOL_XDA_P_READS + PCS.POOL_TEMP_DATA_P_READS + PCS.POOL_TEMP_INDEX_P_READS + PCS.POOL_TEMP_XDA_P_READS + PCS.POOL_COL_P_READS + PCS.POOL_TEMP_COL_P_READS ) AS POOL_P_READS, COALESCE( ACT.POOL_DATA_L_READS,
PCS.POOL_DATA_L_READS ) AS POOL_DATA_L_READS, COALESCE( ACT.POOL_INDEX_L_READS, PCS.POOL_INDEX_L_READS ) AS POOL_INDEX_L_READS, COALESCE( ACT.POOL_TEMP_DATA_L_READS + ACT.POOL_TEMP_XDA_L_READS + ACT.POOL_TEMP_INDEX_L_READS,
PCS.POOL_TEMP_DATA_L_READS + PCS.POOL_TEMP_XDA_L_READS + PCS.POOL_TEMP_INDEX_L_READS ) AS TEMP_L_READS, COALESCE( ACT.INT_ROWS_DELETED, PCS.INT_ROWS_DELETED ) AS INT_ROWS_DELETED, COALESCE( ACT.INT_ROWS_UPDATED,
PCS.INT_ROWS_UPDATED ) AS INT_ROWS_UPDATED, COALESCE( ACT.INT_ROWS_INSERTED, PCS.INT_ROWS_INSERTED ) AS INT_ROWS_INSERTED, COALESCE( CASE WHEN LENGTH(ACT.STMT_TEXT) <= 30 THEN CAST( ACT.STMT_TEXT AS VARCHAR(30) ) ELSE
CAST( SUBSTR(ACT.STMT_TEXT, 1, 27) || '...' AS VARCHAR(30) ) END, CASE WHEN LENGTH(PCS.STMT_TEXT) <= 30 THEN CAST( PCS.STMT_TEXT AS VARCHAR(30) ) ELSE CAST( SUBSTR(PCS.STMT_TEXT, 1, 27) || '...' AS VARCHAR(30) ) END ) AS
STMT_TEXT, CONN.APPLICATION_HANDLE, COALESCE( ACT.EXECUTABLE_ID, PCS.EXECUTABLE_ID ) AS EXECUTABLE_ID FROM TABLE( MON_GET_CONNECTION( NULL, 0, 1)) AS CONN LEFT OUTER JOIN TABLE( MON_GET_ACTIVITY( NULL, 0)) AS ACT ON
CONN.APPLICATION_HANDLE = ACT.APPLICATION_HANDLE AND CONN.MEMBER = ACT.MEMBER LEFT OUTER JOIN TABLE( MON_GET_AGENT( NULL, NULL, NULL, 0)) AS AGENT ON CONN.APPLICATION_HANDLE = AGENT.APPLICATION_HANDLE AND CONN.MEMBER =
AGENT.MEMBER LEFT OUTER JOIN TABLE( MON_GET_PKG_CACHE_STMT( NULL, NULL, NULL, 0)) AS PCS ON CONN.LAST_EXECUTABLE_ID = PCS.EXECUTABLE_ID AND CONN.MEMBER = PCS.MEMBER WHERE CONN.APPLICATION_HANDLE = 2919 AND
CONN.APPLICATION_HANDLE = 2919 AND AGENT.AGENT_STATE = 'ACTIVE' ), MAX_CPU_MEMBER AS ( SELECT STMT_INFO.EXECUTABLE_ID, MEMBER AS MAXCPUMEMBER, MAX_TOTAL_CPU_TIME_US FROM STMT_INFO, ( SELECT EXECUTABLE_ID,
MAX(STMT_INFO.TOTAL_CPU_TIME_US) AS MAX_TOTAL_CPU_TIME_US FROM STMT_INFO GROUP BY EXECUTABLE_ID ) AS T WHERE STMT_INFO.EXECUTABLE_ID = T.EXECUTABLE_ID AND STMT_INFO.TOTAL_CPU_TIME_US = T.MAX_TOTAL_CPU_TIME_US ) SELECT
MIN(APPLICATION_HANDLE) AS APPLICATION_HANDLE, MAX(STMT_TEXT) AS STMT_TEXT, MAX(ACTIVITY_TYPE) AS ACTIVITY_TYPE, MAX(REQUEST_TYPE) AS REQUEST_TYPE, MIN(ENTRY_TIME) AS ENTRY_TIME, CAST( SUM(TOTAL_CPU_TIME_US)/ 1000.0 AS
DOUBLE ) AS TOTAL_CPU_TIME_MS, SUM(STMT_EXEC_TIME_MS) AS STMT_EXEC_TIME_MS, MAX(QUERY_COST_ESTIMATE) AS QUERY_COST_ESTIMATE, MAX(QUERY_CARD_ESTIMATE) AS QUERY_CARD_ESTIMATE, MAX(AGENTS_TOP) AS AGENTS_TOP,
SUM(TOTAL_SECTION_SORT_TIME_MS) AS TOTAL_SECTION_SORT_TIME_MS, SUM(SORT_OVERFLOWS) AS SORT_OVERFLOWS, SUM(TOTAL_SORTS) AS TOTAL_SORTS, MAX(QUERY_ACTUAL_DEGREE) AS QUERY_ACTUAL_DEGREE, SUM(NUM_AGENTS) AS NUM_AGENTS,
SUM(POOL_L_READS) AS POOL_L_READS, SUM(POOL_P_READS) AS POOL_P_READS, SUM(POOL_DATA_L_READS) AS POOL_DATA_L_READS, SUM(POOL_INDEX_L_READS) AS POOL_INDEX_L_READS, SUM(TEMP_L_READS) AS TEMP_L_READS, CASE WHEN SUM(POOL_L_READS)
= 0 THEN 1 ELSE CAST( 1 - SUM(POOL_P_READS) * 1.0 / SUM(POOL_L_READS) AS DOUBLE ) END AS HIT_RATIO, MIN(MAXCPUMEMBER) AS MAXCPUMEMBER, CAST( MAX(MAX_TOTAL_CPU_TIME_US) / 1000.0 AS DOUBLE ) AS MAXDBP_CPU_TIME_MS,
SUM(INT_ROWS_DELETED) AS INT_ROWS_DELETED, SUM(INT_ROWS_UPDATED) AS INT_ROWS_UPDATED, SUM(INT_ROWS_INSERTED) AS INT_ROWS_INSERTED, STMT_INFO.EXECUTABLE_ID FROM STMT_INFO LEFT OUTER JOIN MAX_CPU_MEMBER ON
STMT_INFO.EXECUTABLE_ID = MAX_CPU_MEMBER.EXECUTABLE_ID GROUP BY STMT_INFO.EXECUTABLE_ID ORDER BY TOTAL_CPU_TIME_MS DESC;
```

dmctop – SQL Associated with Application Handle



dmctop – db2exfmt


```
DB2 Universal Database Version 11.5, 5622-044 (c) Copyright IBM Corp. 1991, 2019
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool



******************** EXPLAIN INSTANCE ********************

DB2_VERSION:       11.05.8
FORMATTED ON DB:   HSPRD
SOURCE_NAME:       SYSLH100
SOURCE_SCHEMA:     NULLID
SOURCE_VERSION:
EXPLAIN_TIME:      2025-09-27-12.10.44.067533
EXPLAIN_REQUESTER: HSPRD

Database Context:
----------------
        Parallelism:          None
        CPU Speed:            2.361721e-07
        Comm Speed:           100
        Buffer Pool size:     7923369
        Sort Heap size:       50000
        Database Heap size:   1200
        Lock List size:       1104017
        Maximum Lock List:    97
        Average Applications: 1
        Locks Available:      34268688

Package Context:
---------------
        SQL Type:           Dynamic
        Optimization Level: 5
        Blocking:           Block All Cursors
        Isolation Level:    Uncommitted Read



---------------- STATEMENT 1  SECTION 2 ----------------
        QUERYNO:       1
        QUERYTAG:
        Statement Type:   Select
        Updatable:        No
        Deletable:        No
        Query Degree:     1

Original Statement:
------------------
SELECT "Provider ID" AS "Provider_ID",
    "Claim ID" AS "Claim_ID",
    "Claim DOS" AS "Claim_DOS",
    "Claim Paid Amount" AS "Claim_Paid_Amount",
    "Claim Status" AS "Claim_Status"
FROM (SELECT DISTINCT
PVO.PVO_PRV_ID                          AS "Provider ID",
CPV.CPV_CLM_ID                          AS "Claim ID",
CDT.CDT_DT                                    AS "Claim DOS",
(BILD.CAM_AMT - SUM(PAID.CAJ_AMT))
                                        AS "Claim Paid Amount",
TCL.TCL_TYPE_DESC                       AS "Claim Status"

"explain.92050000" 2367 lines, 63505 characters
```

dmctop -- Locking


```
Menu
Menu
├──View
│   ├──Overview(d)
│   ├──Throughput(w)
│   ├──Top consumers(B)
│   ├──Connections(l)
│   ├──Statements(s)
│   │   ├──In-flight stmts(s)
│   │   └──Executed SQL/Pkg cache(D)
│   ├──IO(I)
│   ├──Locking(L)
│   │   ├──Lock statistics(L)
│   │   ├──Blockers,waiters(b)
│   │   ├──Locks(U)
│   │   └──Locked objects(o)
│   ├──Memory(O)
│   ├──Storage(t)
│   └──Other(u)
├──Help(h)
├──Settings(E)
└──Quit(q)
```

Locking - Locks

[|]16:17:52 Data mode: actual, Next refresh: 8 secs, Filter on/off:off — AIX, member=[0/0], hsprd:hsprd

Locks held: 22
Agents waiting: 0
Lock list storage %: 0.01%
Applications connected: 13

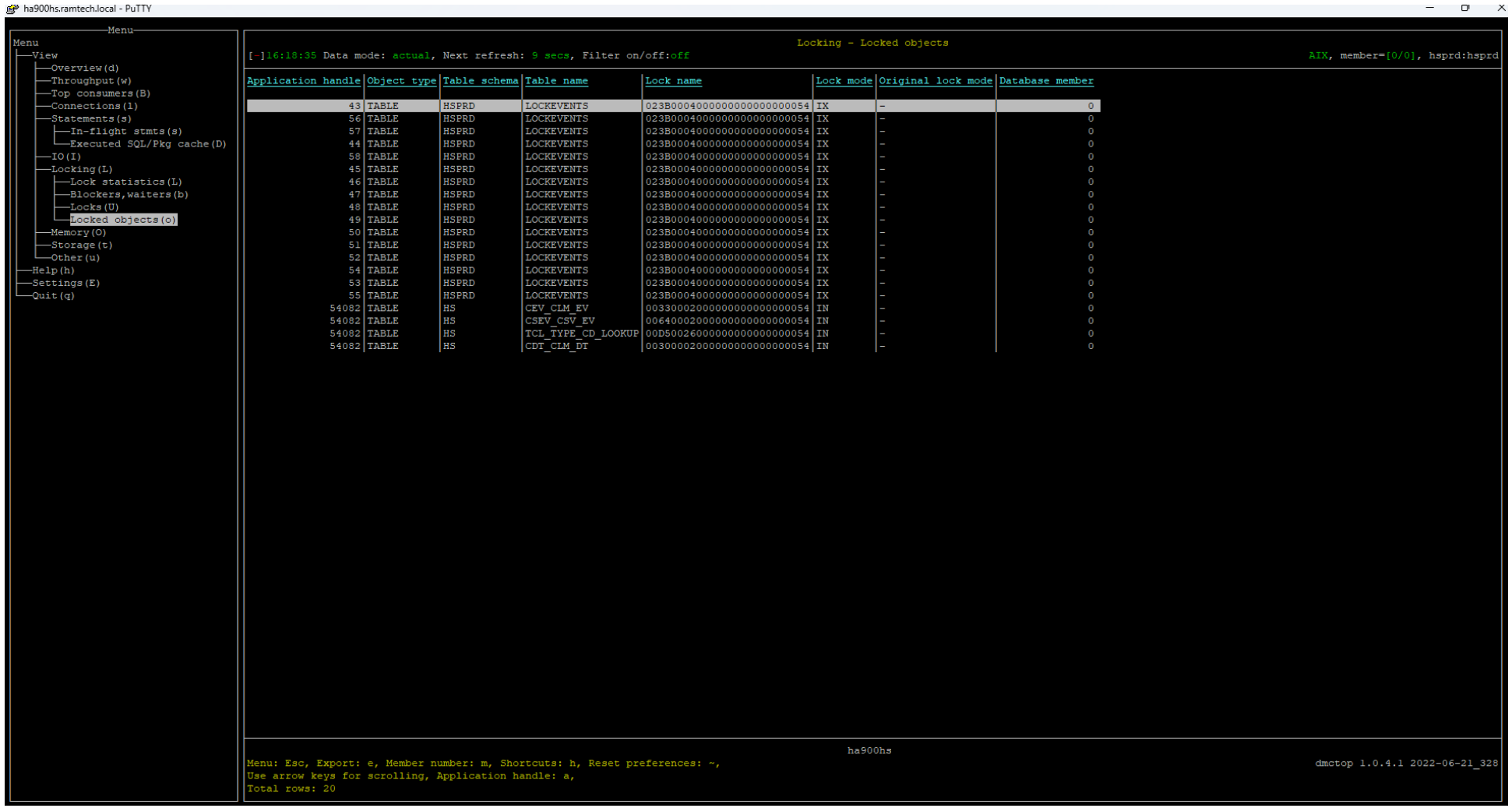
Application handle	Application name	Event state	Event type	Event object	Object name	Lock mode	Object type	Lock status	Lock count	Is blocker	Locked by	Tablespace name
43	db2fw0	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
44	db2fw1	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
45	db2fw2	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
46	db2fw3	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
47	db2fw4	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
48	db2fw5	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
49	db2fw6	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
50	db2fw7	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
51	db2fw8	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
52	db2fw9	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
53	db2fw10	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
54	db2fw11	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
55	db2fw12	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
56	db2fw13	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
57	db2fw14	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
58	db2fw15	EXECUTING	PROCESS	REQUEST	HSPRD.LOCKEVENTS	IX	TABLE	Granted	1	No	-	GTSLOCK
54082	CData/DRDA	EXECUTING	PROCESS	REQUEST	HS.CDT_CLM_DT	IN	TABLE	Granted	1	No	-	MCSCDT
54082	CData/DRDA	EXECUTING	PROCESS	REQUEST	HS.CEV_CLM_EV	IN	TABLE	Granted	2	No	-	MCSCEV
54082	CData/DRDA	EXECUTING	PROCESS	REQUEST	HS.CSEV_CSV_EV	IN	TABLE	Granted	2	No	-	MCSCSEV
54082	CData/DRDA	EXECUTING	PROCESS	REQUEST	HS.TCL_TYPE_CD_LOOKUP	IN	TABLE	Granted	1	No	-	MCSREF
54082	CData/DRDA	EXECUTING	PROCESS	REQUEST	VARIATION	S	VARIATION	Granted	1	No	-	-
54090	dmctop	EXECUTING	PROCESS	ROUTINE	VARIATION	S	VARIATION	Granted	1	No	-	-

ha900hs

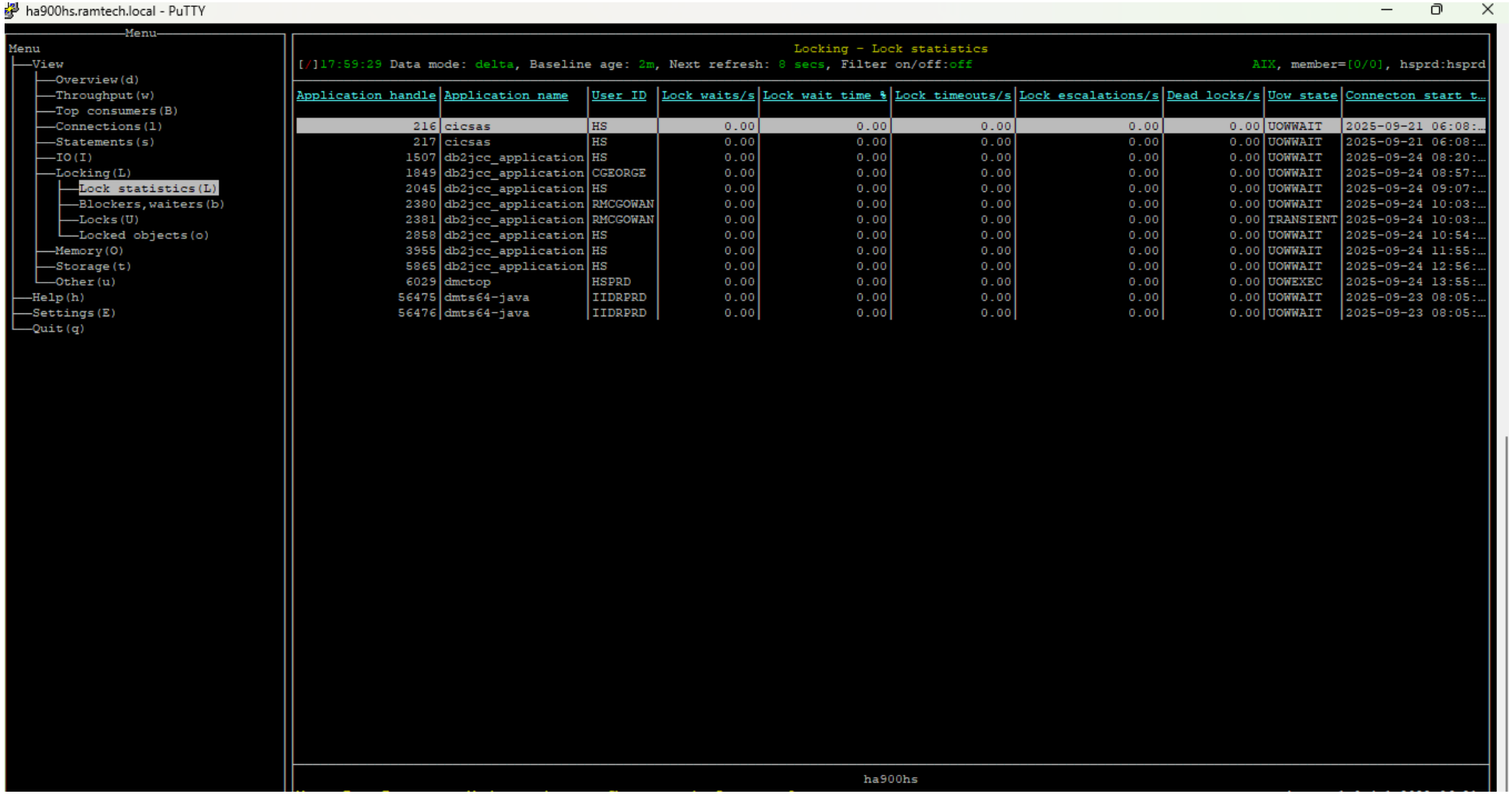
Menu: Esc, Export: e, Member number: m, Shortcuts: h, Reset preferences: ~,
Use arrow keys for scrolling, Application handle: a, Lock Chain: H,
Total rows: 22

dmctop 1.0.4.1 2022-06-21_328

dmctop – Lock Objects

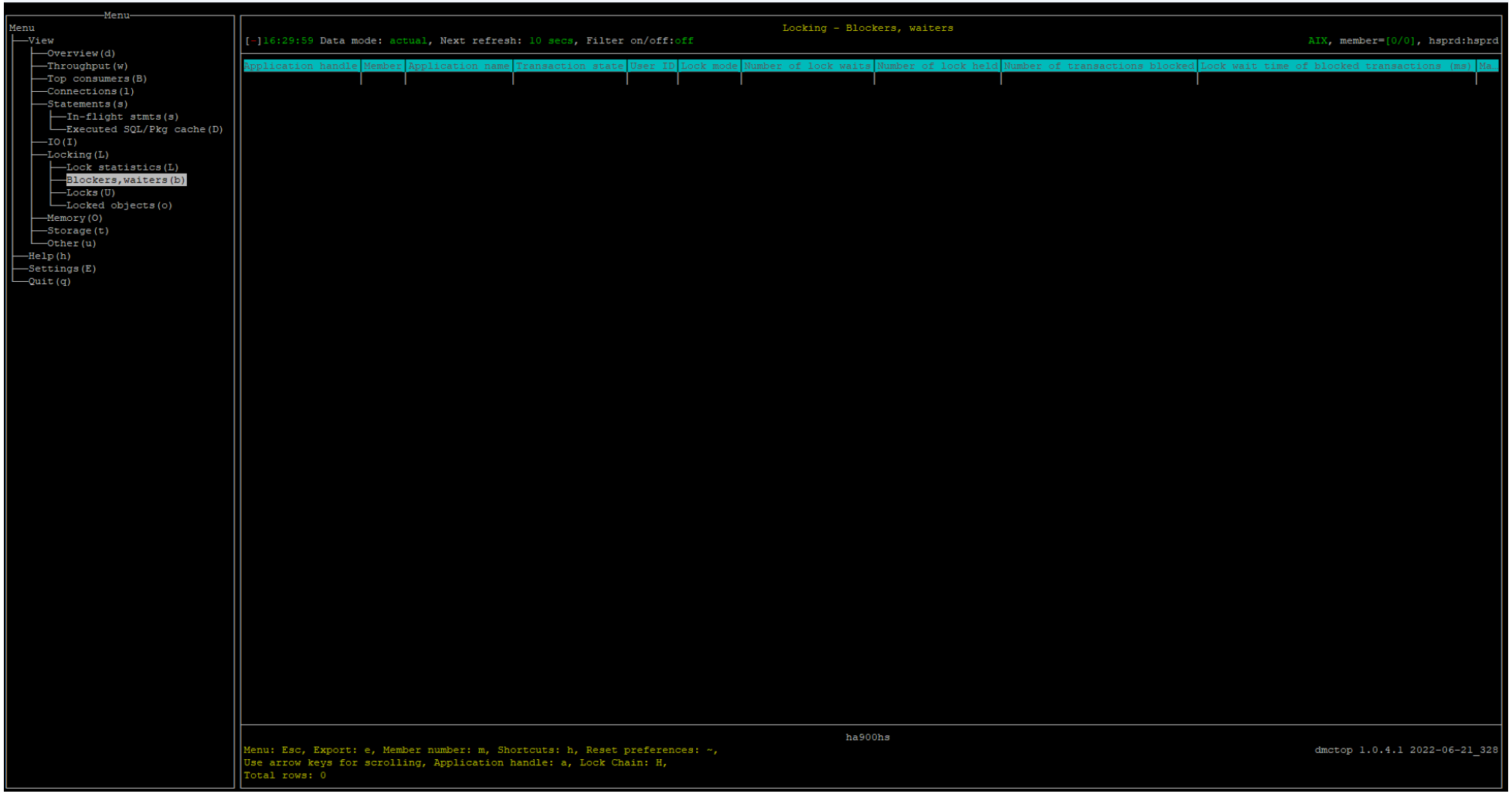


dmctop – Lock Statistics

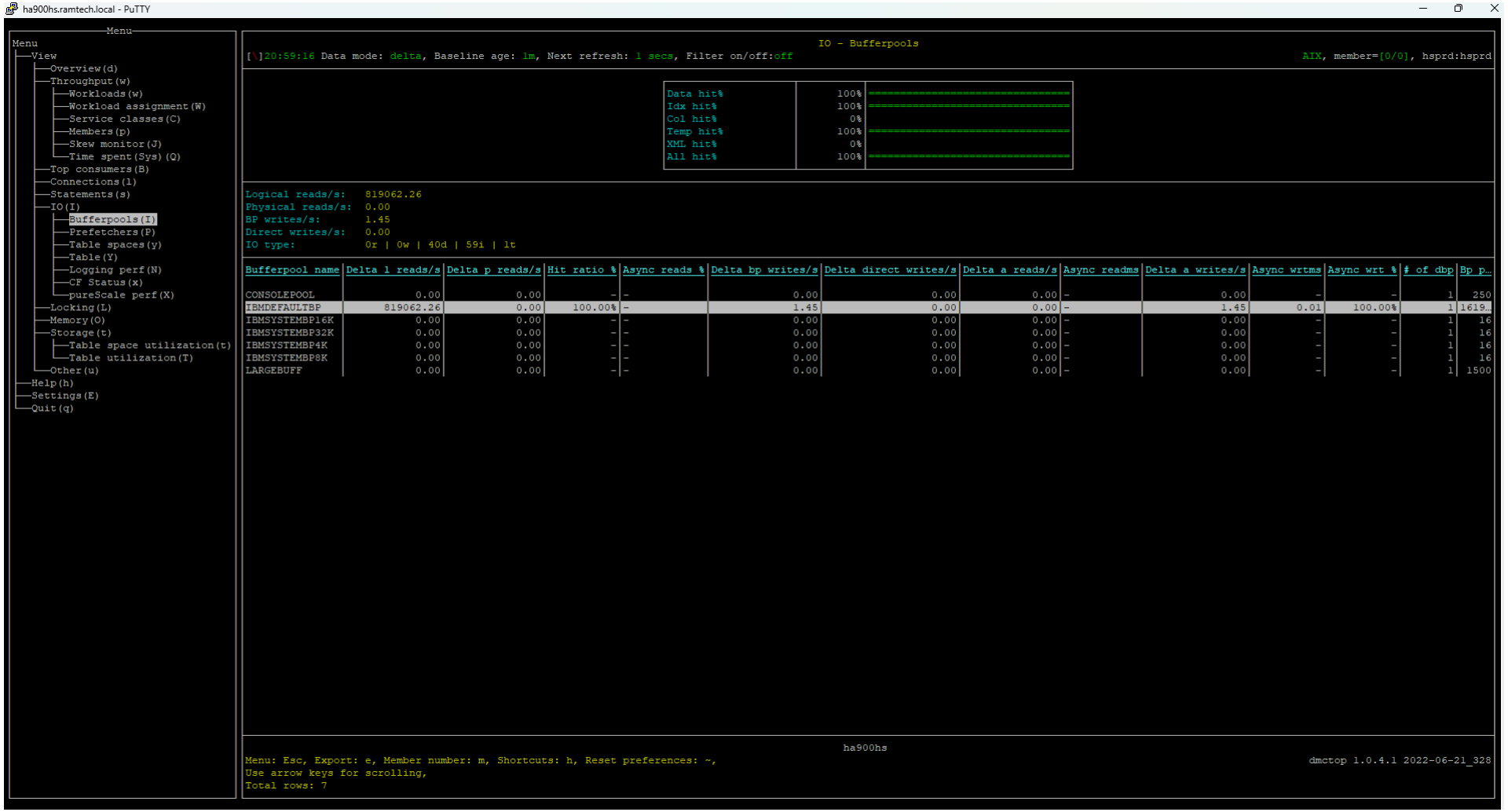


dmctop – Launch Appl Drilldown

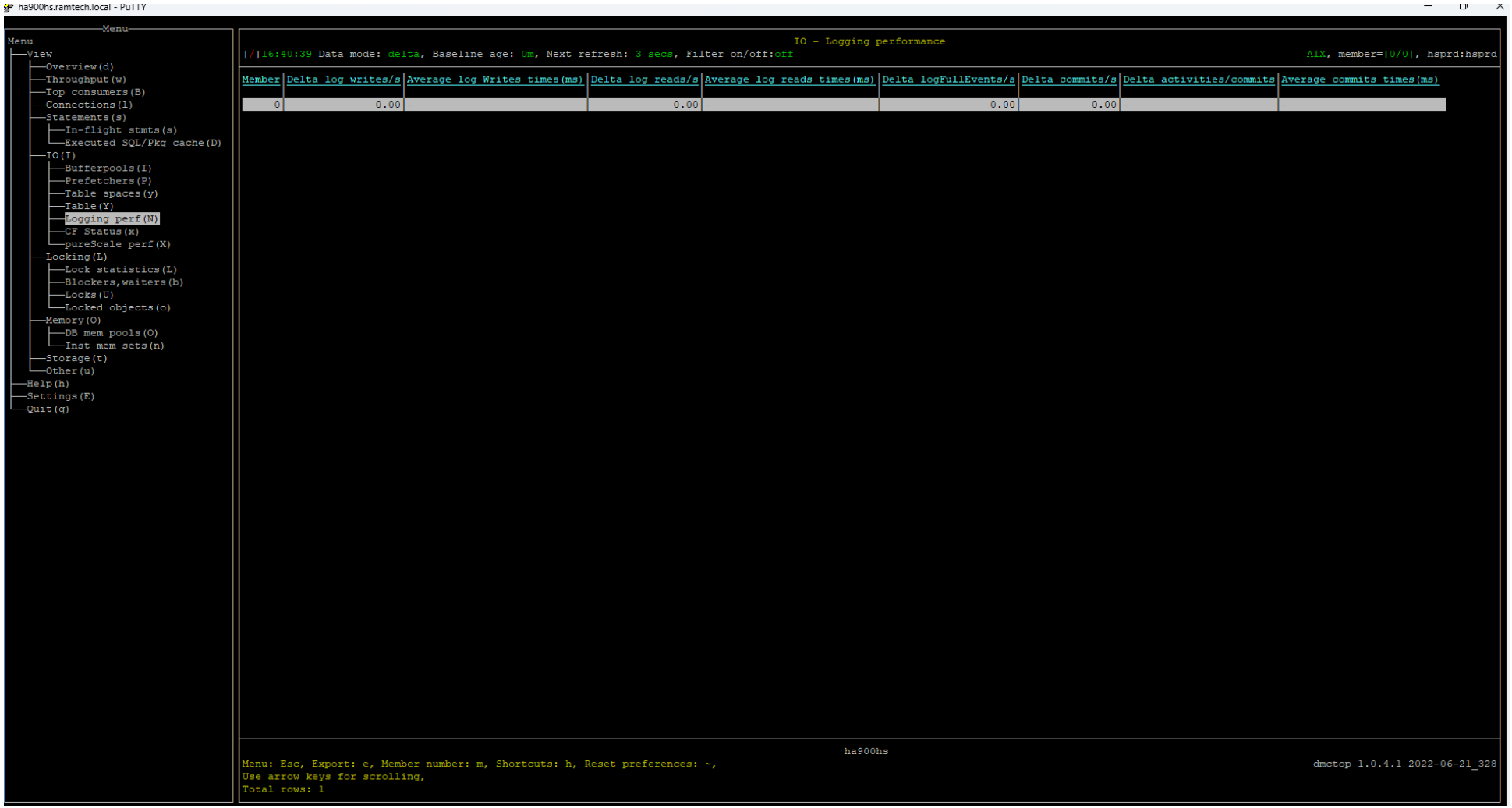
dmctop – Blockers and Waiters



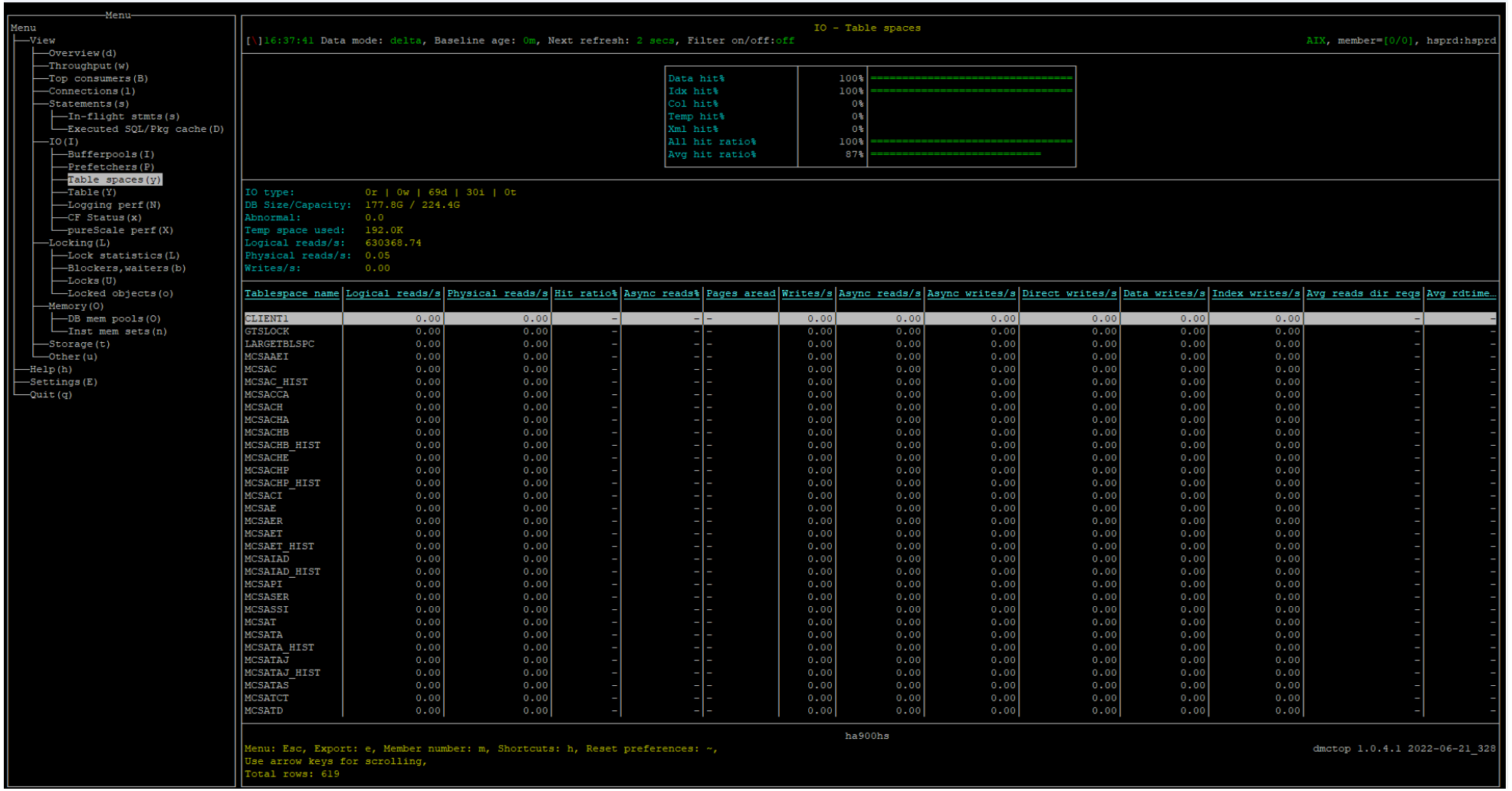
dmctop – Bufferpools



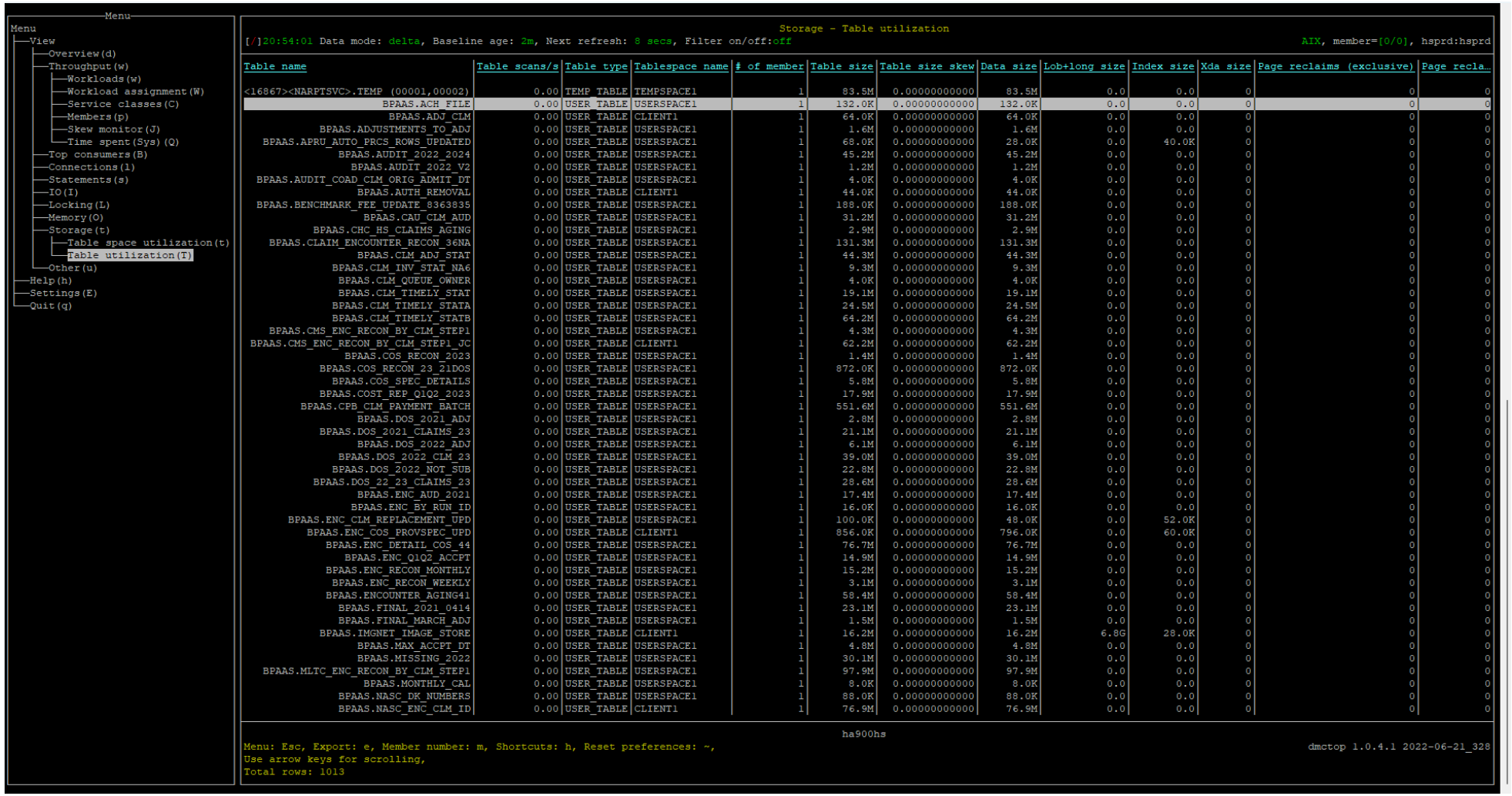
dmctop – Logging Performance



dmctop – Tablespace Details



dmctop – Table Statistics



db2mon

- db2mon
  - Consists of Db2 supplied scripts provided with each Db2 installation on LINUX and Unix
  - Some basic monitoring provided
  - Uses deltas between monitoring intervals to report activities during monitored intervals
  - Additional setup and configuration required for historical reporting capability (longer monitoring interval)
  - Included in the Samples/Perf subdirectory
    - Prior to Db2 version 11.1 available as a download
- Uses lightweight in-memory monitoring interfaces (MON_GET functions)
- Online and Offline mode
  - Default interval of 30 seconds

db2mon

- Requires that below DB CFG Parameters be enabled
  - MON_ACT_METRICS (at least to BASE, default)
  - MON_REQ_METRICS (at least to BASE, EXTENDED is ideal for db2mon)
  - MON_OBJ_METRICS (at least BASE)
- To install and configure a db2mon.cmd file on Windows refer to the following links
  - https://jazz.net/wiki/bin/view/Deployment/Db2MustGather
  - https://www.ibm.com/docs/en/db2/12.1.0?topic=tuning-collecting-reporting-performance-monitor-data
- Offline mode produces output in IXF format from start to end of monitoring interval for loading and analysis in another system
  - Offloads any overhead from monitored database

dmctop – Database Throughput Metrics


```
================================================
 DB#THRUP: Throughput metrics at database level
================================================

/* IBM_DB2MON */ select min(ts_delta) ts_delta, member, decimal((sum(act_completed_total) / float(min(ts_delta))), 10, 1) as act_per_s, decimal((sum(total_app_commits) / float(min(ts_delta))
), 10, 1) as cmt_per_s, decimal((sum(total_app_rollbacks) / float(min(ts_delta))), 10, 1) as rb_per_s, decimal((sum(deadlocks) / float(min(ts_delta))), 10, 1) as ddlck_per_s, decimal((sum(se
lect_sql_stmts) / float(min(ts_delta))), 10, 1) as sel_p_s, decimal((sum(uid_sql_stmts) / float(min(ts_delta))), 10, 1) as uid_p_s, decimal((sum(rows_inserted) / float(min(ts_delta))), 10, 1
) as rows_ins_p_s, decimal((sum(rows_updated) / float(min(ts_delta))), 10, 1) as rows_upd_p_s, decimal((sum(rows_returned) / float(min(ts_delta))), 10, 1) as rows_ret_p_s, decimal((sum(rows_
modified) / float(min(ts_delta))), 10, 1) as rows_mod_p_s, decimal((sum(pkg_cache_inserts) / float(min(ts_delta))), 10, 1) as pkg_cache_ins_p_s, decimal((sum((pool_data_p_reads + pool_temp_d
ata_p_reads + pool_index_p_reads + pool_temp_index_p_reads + pool_xda_p_reads + pool_temp_xda_p_reads + pool_col_p_reads + pool_temp_col_p_reads)) / float(min(ts_delta))) , 10, 1) as p_rd_pe
r_s from mon_get_workload_diff where ts_delta > 0 group by member order by member asc with UR

TS_DELTA    MEMBER ACT_PER_S    CMT_PER_S    RB_PER_S     DDLCK_PER_S  SEL_P_S      UID_P_S      ROWS_INS_P_S ROWS_UPD_P_S ROWS_RET_P_S ROWS_MOD_P_S PKG_CACHE_INS_P_S P_RD_PER_S
----------- ------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------------ ------------
         34      0        869.5         57.6          0.4          0.0        868.0         20.6       1713.4          5.2       1000.9      11007.3                4.5       1425.6

  1 record(s) selected.


================================================================================
```

dmctop – Waits and Processing Time Database Level


```
============================================================
 DB#TIMEB: Time breakdown at database level (wait + processing)
============================================================

/* IBM_DB2MON */ select member, integer(sum(total_rqst_time)) as total_rqst_tm, decimal(sum(total_compile_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_compile, decimal(sum(total_section_time) / float(sum(total_rqst_time)) * 10
0, 5, 2) as pct_section, decimal(sum(total_section_sort_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_sort, decimal(sum(total_col_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_col, decimal(sum(total_col_synopsis_time)
 / float(sum(total_rqst_time)) * 100, 5, 2) as pct_col_synop, decimal(sum(total_commit_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_commit, decimal(sum(total_rollback_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_rba
ck, decimal(sum(total_connect_request_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_conn, decimal(sum(total_routine_user_code_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_rtn_usr_code, decimal(sum(total_backup_time)
/ float(sum(total_rqst_time)) * 100, 5, 2) as pct_backup, decimal(sum(total_index_build_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_idx_bld, decimal(sum(total_runstats_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_r
unstats, decimal(sum(total_reorg_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_reorg, decimal(sum(total_load_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_load from mon_get_workload_diff group by member order by membe
r asc with UR

MEMBER TOTAL_RQST_TM PCT_COMPILE PCT_SECTION PCT_SORT PCT_COL PCT_COL_SYNOP PCT_COMMIT PCT_RBACK PCT_CONN PCT_RTN_USR_CODE PCT_BACKUP PCT_IDX_BLD PCT_RUNSTATS PCT_REORG PCT_LOAD
------ ------------ ----------- ----------- -------- ------- ------------ ---------- --------- -------- ---------------- ---------- ----------- ------------ --------- --------
     0        72281        0.10       50.55     1.88    0.00         0.00       0.02      0.00     0.00            41.50       0.00        0.00         0.00      0.00     0.00

  1 record(s) selected.


====================================
 DB#WAITT: Wait times at database level
====================================

/* IBM_DB2MON */ select w.member, integer(sum(total_rqst_time)) as total_rqst_tm, integer(sum(total_wait_time)) as total_wait_tm, decimal((sum(total_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_rqst_wait, decimal((sum(lo
ck_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lock, decimal((sum(lock_wait_time_global) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_glb_lock, decimal((sum(total_extended_latch_wait_time) / float(sum(total_rqst_t
ime))) * 100, 5, 2) as pct_ltch, decimal((sum(log_disk_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lg_dsk, decimal((sum(log_buffer_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lg_buf, decimal((sum(recla
im_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_rclm, decimal((sum(cf_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_cf, decimal((sum(prefetch_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_
pftch, decimal((sum(diaglog_write_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_diag, decimal((sum(audit_file_write_wait_time) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_aud_w, decimal((sum(audit_subsystem_wait_t
ime) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_aud_ss, decimal((sum(evmon_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_evmon, decimal((sum(comm_exit_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct
_commexit, decimal((sum(lob_prefetch_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lob_pftch, decimal((sum(ext_table_recv_wait_time + ext_table_send_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_extbl, dec
imal((sum(fed_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_fed, decimal((sum(pool_read_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_pool_r, decimal((sum(direct_read_time) / float(sum(total_rqst_time))) * 100,
 5, 2) as pct_dir_r, decimal((sum(direct_write_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_dir_w, decimal((sum(fcm_recv_wait_time + fcm_send_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_fcm, decimal((sum(tcp
ip_send_wait_time + tcpip_recv_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_tcpip, decimal((sum(ida_send_wait_time + ida_recv_wait_time) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_ida from mon_get_workload_diff
w group by w.member order by w.member asc with UR

MEMBER TOTAL_RQST_TM TOTAL_WAIT_TM PCT_RQST_WAIT PCT_LOCK PCT_GLB_LOCK PCT_LTCH PCT_LG_DSK PCT_LG_BUF PCT_RCLM PCT_CF  PCT_PFTCH PCT_DIAG PCT_AUD_W PCT_AUD_SS PCT_EVMON PCT_COMMEXIT PCT_LOB_PFTCH PCT_EXTBL PCT_FED PCT_POOL_R PCT_DIR_R PC
T_DIR_W PCT_FCM PCT_TCPIP PCT_IDA
------ ------------ ------------ ------------ -------- ------------ -------- ---------- ---------- -------- ------- --------- -------- --------- ---------- ---------- ------------ ------------ --------- ------- ---------- --------- --
------- ------- ---------- -------
     0        72281          599         0.82     0.00         0.00     0.00       0.01       0.00     0.00    0.00      0.00     0.00      0.00       0.00       0.00         0.00         0.00      0.00    0.00       0.00      0.00
   0.48     0.00       0.00    0.00

  1 record(s) selected.


========================================
 DB#PROCT: Processing times at database level
========================================

/* IBM_DB2MON */ select member, integer(sum(total_rqst_time)) as total_rqst_tm, integer(sum(total_rqst_time) - sum(total_wait_time)) as total_proc_time, decimal(sum(total_compile_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as p
ct_comp_proc, decimal(sum(total_section_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_sect_proc, decimal(sum(total_section_sort_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_sect_sort_proc, decimal(sum(total
_commit_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_commit, decimal(sum(total_rollback_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_rback , decimal(sum(total_col_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2
) as pct_col_proc, decimal(sum(total_connect_request_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_conn_proc from mon_get_workload_diff group by member order by member asc with UR

MEMBER TOTAL_RQST_TM TOTAL_PROC_TIME PCT_COMP_PROC PCT_SECT_PROC PCT_SECT_SORT_PROC PCT_COMMIT PCT_RBACK PCT_COL_PROC PCT_CONN_PROC
------ ------------ --------------- ------------- ------------- ------------------ ---------- --------- ------------ -------------
     0        72281           71682          0.10         50.06               1.88       0.02      0.00         0.00          0.00

  1 record(s) selected.


Standard input
```

dmctop – Top SQL by Execution Time


```
================================================
 SQL#TOPEXECT: Top SQL statements by execution time
================================================

/* IBM_DB2MON */ select member, integer(num_exec_with_metrics) as num_exec, m.coord_stmt_exec_time, decimal(m.coord_stmt_exec_time / double(num_exec_with_metrics), 10, 2) as avg_coord_exec_t
ime, decimal( (m.coord_stmt_exec_time / double(total_coord_stmt_exec_time)) * 100, 5, 2 ) as pct_coord_stmt_exec_time, m.total_act_time, total_cpu_time, total_cpu_time / num_exec_with_metric
s as avg_cpu_time, case when total_act_time > 0 then decimal( (total_act_wait_time / double(total_act_time)) * 100, 5, 2 ) else 0 end as pct_wait_time, decimal(total_section_time / double(nu
m_exec_with_metrics), 20, 2) as avg_sect_time, decimal(total_col_time / double(num_exec_with_metrics), 20, 2) as avg_col_time, effective_isolation as iso, replace(replace(cast(substr(stmt_te
xt,1,200) as varchar(200)), chr(10), ' '), chr(13), ' ') as stmt_text from mon_get_pkg_cache_stmt_diff m, (select sum(coord_stmt_exec_time) as total_coord_stmt_exec_time from mon_get_pkg_cac
he_stmt_diff where coord_stmt_exec_time > 0), (select executable_id, coord_stmt_exec_time from mon_get_pkg_cache_stmt_diff where coord_stmt_exec_time <> 0 order by coord_stmt_exec_time desc
fetch first 100 rows only) c where (total_act_time <> 0 or m.coord_stmt_exec_time <> 0) and num_exec_with_metrics <> 0 and c.executable_id = m.executable_id order by c.coord_stmt_exec_time d
esc, total_act_time desc, member asc with UR

MEMBER NUM_EXEC    COORD_STMT_EXEC_TIME AVG_COORD_EXEC_TIME PCT_COORD_STMT_EXEC_TIME TOTAL_ACT_TIME       TOTAL_CPU_TIME       AVG_CPU_TIME         PCT_WAIT_TIME   AVG_SECT_TIME          AVG
_COL_TIME           ISO STMT_TEXT

------ ----------- -------------------- -------------------- ---------------------- -------------------- -------------------- -------------------- --------------- ---------------------- ---
-------------------- --- ------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------
     0           1                  449               449.00                   8.36                  449               115335               115335                 0.00                  449.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0           1                  422               422.00                   7.85                  422               112623               112623                 0.00                  422.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0           1                  418               418.00                   7.78                  418               124017               124017                 0.00                  418.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0         159                  260                 1.63                   4.84                  260                13866                   87                90.38                    1.63
                0.00 CS  DECLARE CSV_INFO CURSOR FOR SELECT CSV_LN_NUM ,CSV_SVP_CD ,CSV_PRC_ID_QUAL ,CSV_PRC_CD ,CSV_SRV_QTY ,CSV_DGN_1_CD ,CSV_DGN_2_CD ,CSV_DGN_3_CD ,CSV_DGN_4_CD ,CSV_SVT_C
D ,CSV_REV_CD ,CSDT_DT ,CSDT_TO_DT
     0           1                  222               222.00                   4.13                  222                60624                60624                 0.00                  222.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0           1                  201               201.00                   3.74                  201                59986                59986                 0.00                  201.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0           2                  192                96.00                   3.57                  192                  928                  464               100.00                   96.00
                0.00 CS  SELECT * FROM hs.CRS_CLM_RESULT a WHERE a.CRS_CLM_ID = 908449182 AND a.CRS_TYPE = '103' AND a.CRS_DEL_TS IS NULL  AND a.CRS_ADD_TS = ( SELECT MAX(CRS_ADD_TS) FROM hs.
CRS_CLM_RESULT WHERE CRS_CLM_ID =
     0         380                  159                0.41                   2.96                  159                 6247                   16                94.33                    0.41
                0.00 CS  SELECT SUM(CSAJ_AMT) INTO :H00270   :H00088       FROM CSAJ_CSV_ADJ WHERE CSAJ_CSV_CLM_ID = :H00271         AND CSAJ_CSV_LN_NUM = :H00272

     0           1                  149               149.00                   2.77                  149                37621                37621                 0.00                  149.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0           1                  147               147.00                   2.73                  147                37705                37705                 0.00                  147.00
                0.00 CS  SELECT     SOURCE,   CUS_REF_ID,   CUS_NAME,   ALT_REQ_ID,   REQ_ID,   REQ_ID_DESC,   REQ_ADDL_INFO,   REC_DT,   REQ_TS,   REQ_PARENT_TS,   REQ_TO_ID,   PREV_QUEUE_ID
,   REQ_DT,   REQ_TM,   REQ_USER_I
     0          35                  144                4.11                   2.68                  144                23123                  660                 0.00                    4.11
                0.00 CS  SELECT MAX(PVCR_RULE_VAL_D) INTO :H00160         :H00030       FROM PPT_PRD_PLAN_TIER, PVCR_PVC_RULE A, PVC_PRV_CNTRCT B, CSV_CLM_SRV WHERE CSV_CLM_ID = :H00005
AND PPT_ID = CSV_PPT_ID AND B.PVC_
     0         509                  136                0.26                   2.53                  136                 9993                   19                82.35                    0.26
                0.00 CS  SELECT CSEV_TYPE INTO :H00353    FROM CSEV_CSV_EV A WHERE CSEV_CSV_CLM_ID = :H00350         AND CSEV_CSV_LN_NUM = :H00351         AND CSEV_DEL_TS IS NULL AND CSEV_A
Standard input
```

dmctop – Top SQL Time Spent Waiting


```
=====================================================
 SQL#TOPWAITW: Top SQL statements by time spent waiting
=====================================================

/* IBM_DB2MON */ select member, decimal((total_act_wait_time / double(total_act_time)) * 100, 5, 2) as pct_wait, decimal((log_disk_wait_time / double(total_act_time)) * 100, 5, 2) as pct_lg_
dsk, decimal((log_buffer_wait_time / double(total_act_time)) * 100, 5, 2) as pct_lg_buf, decimal((lock_wait_time / double(total_act_time)) * 100, 5, 2) as pct_lock, decimal((lock_wait_time_g
lobal / double(total_act_time)) * 100, 5, 2) as pct_glb_lock, decimal((total_extended_latch_wait_time / double(total_act_time)) * 100, 5, 2) as pct_ltch, decimal((reclaim_wait_time / double(
total_act_time)) * 100, 5, 2) as pct_rclm, decimal((cf_wait_time / double(total_act_time)) * 100, 5, 2) as pct_cf, decimal((prefetch_wait_time / double(total_act_time)) * 100, 5, 2) as pct_p
ftch, decimal((diaglog_write_wait_time / double(total_act_time)) * 100, 5, 2) as pct_diag, decimal((pool_read_time / double(total_act_time)) * 100, 5, 2) as pct_pool_r, decimal((direct_read_
time / double(total_act_time)) * 100, 5, 2) as pct_dir_r, decimal((direct_write_time / double(total_act_time)) * 100, 5, 2) as pct_dir_w, decimal(((fcm_recv_wait_time+fcm_send_wait_time) / d
ouble(total_act_time)) * 100, 5, 2) as pct_fcm, decimal((audit_file_write_wait_time / double(total_act_time)) * 100, 5, 2) as pct_aud_w, decimal((audit_subsystem_wait_time / double(total_act
_time)) * 100, 5, 2) as pct_aud_ss, decimal((evmon_wait_time / double(total_act_time)) * 100, 5, 2) as pct_evmon, decimal((comm_exit_wait_time / double(total_act_time)) * 100, 5, 2) as pct_c
ommexit, decimal((lob_prefetch_wait_time / double(total_act_time)) * 100, 5, 2) as pct_lob_pftch, decimal(((ext_table_recv_wait_time + ext_table_send_wait_time) / double(total_act_time)) * 1
00, 5, 2) as pct_extbl, decimal((fed_wait_time / double(total_act_time)) * 100, 5, 2) as pct_fed, decimal(((ida_send_wait_time + ida_recv_wait_time) / double(total_act_time)) * 100, 5, 2) as
 pct_ida, replace(replace(cast(substr(stmt_text,1,200) as varchar(200)), chr(10), ' '), chr(13), ' ') as stmt_text from mon_get_pkg_cache_stmt_diff m, (select executable_id, sum(total_act_wa
it_time) sum_members_total_act_wait_time from mon_get_pkg_cache_stmt_diff group by executable_id order by sum(total_act_wait_time) desc fetch first 100 rows only) c where total_act_wait_time
 <> 0 and num_exec_with_metrics <> 0 and c.executable_id = m.executable_id order by sum_members_total_act_wait_time desc, total_act_wait_time desc, member asc with UR

MEMBER PCT_WAIT PCT_LG_DSK PCT_LG_BUF PCT_LOCK PCT_GLB_LOCK PCT_LTCH PCT_RCLM PCT_CF  PCT_PFTCH PCT_DIAG PCT_POOL_R PCT_DIR_R PCT_DIR_W PCT_FCM PCT_AUD_W PCT_AUD_SS PCT_EVMON PCT_COMMEXIT PC
T_LOB_PFTCH PCT_EXTBL PCT_FED PCT_IDA STMT_TEXT

------ -------- ---------- ---------- -------- ------------ -------- -------- ------- --------- -------- ---------- --------- --------- ------- --------- ---------- --------- ------------ --
---------- --------- ------- ------- ------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------

     0    90.38       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      90.38      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 DECLARE CSV_INFO CURSOR FOR SELECT CSV_LN_NUM ,CSV_SVP_CD ,CSV_PRC_ID_QUAL ,CSV_PRC_CD ,CSV_SRV_QTY ,CSV_DGN_1_CD ,CSV_DGN_2_CD ,CSV_DGN_3_CD ,CSV_DGN_4
_CD ,CSV_SVT_CD ,CSV_REV_CD ,CSDT_DT ,CSDT_TO_DT
     0   100.00       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00     100.00      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT * FROM hs.CRS_CLM_RESULT a WHERE a.CRS_CLM_ID = 908449182 AND a.CRS_TYPE = '103' AND a.CRS_DEL_TS IS NULL  AND a.CRS_ADD_TS = ( SELECT MAX(CRS_AD
D_TS) FROM hs.CRS_CLM_RESULT WHERE CRS_CLM_ID =
     0    94.33       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      94.33      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT SUM(CSAJ_AMT) INTO :H00270   :H00088       FROM CSAJ_CSV_ADJ WHERE CSAJ_CSV_CLM_ID = :H00271          AND CSAJ_CSV_LN_NUM = :H00272


     0    82.35       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.73     0.00      81.61      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT CSEV_TYPE INTO :H00353    FROM CSEV_CSV_EV A WHERE CSEV_CSV_CLM_ID = :H00350          AND CSEV_CSV_LN_NUM = :H00351          AND CSEV_DEL_TS IS N
ULL AND CSEV_ADD_TS = (SELECT MAX(CSEV_ADD_TS) F
     0    86.36       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      86.36      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT CHI_CD, CHI_CD_QUAL INTO :H00173, :H00007      FROM CHI_CLM_HLTH_CD WHERE CHI_CLM_ID = :H00006     AND CHI_SEQ_NUM = :H00266      AND CHI_CD_QUAL
 IN ('ABF', 'ABK', 'ASD', 'ATD', 'BF', 'BK', 'SD
     0    91.37       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      91.37      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT CRF_REF_NUM INTO :H00179      FROM CRF_CLM_REF WHERE CRF_CLM_ID = :H00180     AND CRF_REF_QUAL = :H00181


     0    88.23       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      88.23      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 SELECT CPV_PRV_ID, CPV_TXC_CD INTO :H00196    , :H00198     :H00088       FROM CPV_CLM_PRV WHERE CPV_CLM_ID = :H00005     AND CPV_ENT_CD = :H00197


     0    96.77       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      96.77      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 DECLARE MENU_CSR CURSOR FOR SELECT SMO_TYPE_IND, SMO_AUTH_IND, SMO_SOPT_ID, SMO_ALT_SOPT_ID FROM SMO_SEC_MENU_OPTION A WHERE SMO_SGP_ID = :H00001     AN
D SMO_SMU_ID = :H00002     AND SMO_DEL_TS IS NUL
     0    95.00       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      95.00      0.00      0.00    0.00      0.00       0.00      0.00         0.00
       0.00      0.00    0.00    0.00 UPDATE CGID_CGI_DX_INFO SET CGID_DEL_USER_ID = :H00097      , CGID_DEL_TS = CURRENT TIMESTAMP WHERE CGID_CLM_ID = :H00098     AND CGID_DEL_TS IS NULL


     0    85.71       0.00       0.00     0.00         0.00     0.00     0.00    0.00      0.00     0.00      85.71      0.00      0.00    0.00      0.00       0.00      0.00         0.00
```

dmctop – Time Spent by Package


```
==========================================
 PKG#EXECT: Time spent executing by package
==========================================

/* IBM_DB2MON */ select member, cast(substr(package_name,1,20) as varchar(20)) as package_name, sum(num_exec_with_metrics) as num_stmts_exec, sum(coord_stmt_exec_time) as coord_stmt_exec_tim
e from mon_get_pkg_cache_stmt_diff where coord_stmt_exec_time > 0 group by member, package_name order by coord_stmt_exec_time desc fetch first 100 rows only with UR

MEMBER PACKAGE_NAME         NUM_STMTS_EXEC       COORD_STMT_EXEC_TIME
------ -------------------- -------------------- --------------------
     0 -                                   530                 3142
     0 MCSU042A                          11274                 1180
     0 MCSU022                            9933                  376
     0 MCSO304                             361                  203
     0 MCSO306                             354                   60
     0 MCSO302                             482                   49
     0 MCSO277                             130                   46
     0 MCSO458                              89                   44
     0 MCSO106                              81                   41
     0 MCSU049B                            145                   40
     0 MCSO380                             355                   34
     0 MCSO305                             446                   26
     0 MCSO276                              23                   21
     0 MCSO379                             136                   20
     0 MCSU019O                             39                   14
     0 MCSO300                             134                   11
     0 MCSO459                             115                    9
     0 MCSO307                               3                    8
     0 MCSU026                             418                    8
     0 MCSU043                               4                    6
     0 MCSO480                              50                    5
     0 MCSU014                              24                    5
     0 MCSO303                               6                    3
     0 MCSU050                              38                    3
     0 MCSU028                               3                    3
     0 MCSU027                               5                    3
     0 MCSO272                               4                    2
     0 MCSU021                              34                    2
     0 MCSO261                              14                    1
     0 MCSU039                               2                    1
     0 MCSU008                              12                    1
     0 MCSO301                               1                    1
     0 MCSO289                               1                    1

  33 record(s) selected.


================================================================================
 SQL#TOPWAITT: Wait time breakdown for top SQL statements by execution time
--------------------------------------------------------------------------------
```

dmctop – Sort Metrics


```
========================================
 DB#SORT: Sort metrics at database level
========================================

/* IBM_DB2MON */ select member, integer(sum(total_sorts)) total_sorts, integer(sum(sort_overflows)) sort_overflows, sum(total_section_sort_time) tot_sect_sort_tm, sum(total_section_sort_proc_time) tot_sect_sort_proc_tm , sum(sort_shrheap
_allocated) sort_shrheap_allocated, integer(sum(total_hash_joins)) total_hsjn, integer(sum(hash_join_overflows)) hsjn_ovfl, integer(sum(post_threshold_hash_joins)) pst_thr_hsjn, integer(sum(post_shrthreshold_hash_joins)) pst_shrthr_hsjn
from mon_get_workload_diff group by member order by member asc with UR

MEMBER TOTAL_SORTS SORT_OVERFLOWS TOT_SECT_SORT_TM     TOT_SECT_SORT_PROC_TM SORT_SHRHEAP_ALLOCATED TOTAL_HSJN  HSJN_OVFL   PST_THR_HSJN PST_SHRTHR_HSJN
------ ----------- -------------- -------------------- --------------------- ---------------------- ----------- ----------- ------------ ---------------
     0           4              0                 1360                  1360                    285           4           0            0               0

  1 record(s) selected.
```

dmctop – System level Information


```
==================================================
 DB#SYSRE: Database system resource usage information
==================================================

/* IBM_DB2MON */ select member, cast(substr(os_name,1,8) as varchar(8)) as os, cast(substr(host_name,1,16) as varchar(16)) host_name, cast(substr(os_version,1,8) as varchar(8)) os_ver, cast(substr(os_release,1,8) as varchar(8)) os_rel, s
mallint(cpu_total) cpu_tot, smallint(cpu_online) cpu_onl, smallint(cpu_configured) cpu_cfg, integer(cpu_speed) cpu_speed, smallint(cpu_hmt_degree) cpu_hmt, integer(memory_total) memory_total, integer(memory_free) memory_free, decimal(cpu
_load_short,6,1) cpu_load_shrt, decimal(cpu_load_medium,6,1) cpu_load_med, decimal(cpu_load_long,6,1) cpu_load_lng, decimal(cpu_usage_total,6,1) cpu_usage_tot, integer(swap_pages_in) swap_pages_in, integer(swap_pages_out) swap_pages_out
from env_get_system_resources_diff order by member with UR

MEMBER OS       HOST_NAME        OS_VER   OS_REL   CPU_TOT CPU_ONL CPU_CFG CPU_SPEED   CPU_HMT MEMORY_TOTAL MEMORY_FREE CPU_LOAD_SHRT CPU_LOAD_MED CPU_LOAD_LNG CPU_USAGE_TOT SWAP_PAGES_IN SWAP_PAGES_OUT
------ -------- ---------------- -------- -------- ------- ------- ------- ----------- ------- ------------ ----------- ------------- ------------- ------------- ------------- ------------- --------------
     0 AIX      ha900hs          7        2             32      16      32        3100       8        98304         897           2.7           3.3           3.5          32.0             0              0

  1 record(s) selected.
```

dmctop – Processing Time at Database Level


```
================================================
 DB#PROCT: Processing times at database level
================================================

/* IBM_DB2MON */ select member, integer(sum(total_rqst_time)) as total_rqst_tm, integer(sum(total_rqst_time) - sum(total_wait_time)) as total_proc_time
ct_comp_proc, decimal(sum(total_section_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_sect_proc, decimal(sum(total_section_sort_proc_tir
_commit_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_commit, decimal(sum(total_rollback_time) / float(sum(total_rqst_time)) * 100, 5, 2) as
) as pct_col_proc, decimal(sum(total_connect_request_proc_time) / float(sum(total_rqst_time)) * 100, 5, 2) as pct_conn_proc from mon_get_workload_diff

MEMBER TOTAL_RQST_TM TOTAL_PROC_TIME PCT_COMP_PROC PCT_SECT_PROC PCT_SECT_SORT_PROC PCT_COMMIT PCT_RBACK PCT_COL_PROC PCT_CONN_PROC
------ ------------- --------------- ------------- ------------- ------------------ ---------- --------- ------------ -------------
     0         72281           71682          0.10         50.06               1.88       0.02      0.00         0.00          0.00

  1 record(s) selected.
```

dmctop – Wait times at Database level


```
====================================
DB#WAIT: Wait times at database level
====================================

/* IBM_DB2MON */ select w.member, integer(sum(total_rqst_time)) as total_rqst_tm, integer(sum(total_wait_time)) as total_wait_tm, decimal((sum(total_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_rqst_wait, decimal((sum(lo
ck_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lock, decimal((sum(lock_wait_time_global) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_glb_lock, decimal((sum(total_extended_latch_wait_time) / float(sum(total_rqst_t
ime))) * 100, 5, 2) as pct_ltch, decimal((sum(log_disk_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lg_dsk, decimal((sum(log_buffer_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lg_buf, decimal((sum(recla
im_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_rclm, decimal((sum(cf_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_cf, decimal((sum(prefetch_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_
pftch, decimal((sum(diaglog_write_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_diag, decimal((sum(audit_file_write_wait_time) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_aud_w, decimal((sum(audit_subsystem_wait_t
ime) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_aud_ss, decimal((sum(evmon_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_evmon, decimal((sum(comm_exit_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct
_commexit, decimal((sum(lob_prefetch_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_lob_pftch, decimal((sum(ext_table_recv_wait_time + ext_table_send_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_extbl, dec
imal((sum(fed_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_fed, decimal((sum(pool_read_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_pool_r, decimal((sum(direct_read_time) / float(sum(total_rqst_time))) * 100,
 5, 2) as pct_dir_r, decimal((sum(direct_write_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_dir_w, decimal((sum(fcm_recv_wait_time + fcm_send_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_fcm, decimal((sum(tcp
ip_send_wait_time + tcpip_recv_wait_time) / float(sum(total_rqst_time))) * 100, 5, 2) as pct_tcpip, decimal((sum(ida_send_wait_time + ida_recv_wait_time) / double(sum(total_rqst_time))) * 100, 5, 2) as pct_ida from mon_get_workload_diff
w group by w.member order by w.member asc with UR

MEMBER TOTAL_RQST_TM TOTAL_WAIT_TM PCT_RQST_WAIT PCT_LOCK PCT_GLB_LOCK PCT_LTCH PCT_LG_DSK PCT_LG_BUF PCT_RCLM PCT_CF  PCT_PFTCH PCT_DIAG PCT_AUD_W PCT_AUD_SS PCT_EVMON PCT_COMMEXIT PCT_LOB_PFTCH PCT_EXTBL PCT_FED PCT_POOL_R PCT_DIR_R PC
T_DIR_W PCT_FCM PCT_TCPIP PCT_IDA
------ ------------ ------------ ------------ -------- ------------ -------- ---------- ---------- -------- ------- --------- -------- --------- ---------- ---------- ------------ ------------ --------- ------- ---------- --------- --
------- ------- --------- -------
     0        72281          599         0.82     0.00         0.00     0.00       0.01       0.00     0.00    0.00      0.00     0.00      0.00       0.00       0.00         0.00         0.00      0.00    0.00       0.00      0.00
   0.48     0.00      0.00     0.00

  1 record(s) selected.
```

dmctop – Bufferpool Information


```
============================
 BPL#SIZES: Bufferpool sizes
============================

/* IBM_DB2MON */ select member, cast(substr(bp_name,1,20) as varchar(20)) as bp_name, b.pagesize, mgb.bp_cur_buffsz as num_pages, decimal(double(b.pagesize) * mgb.bp_cur_buffsz / 1024 / 1024, 10, 2) as size_mb, automatic from syscat.buff
erpools b, mon_get_bufferpool_diff mgb where b.bpname = mgb.bp_name order by member with UR

MEMBER BP_NAME             PAGESIZE    NUM_PAGES           SIZE_MB      AUTOMATIC
------ ------------------- ----------- ------------------- ------------ ---------
     0 IBMDEFAULTBP               4096             7921619     30943.82         1
     0 LARGEBUFF                 32768                1500        46.87         1
     0 CONSOLEPOOL               32768                 250         7.81         0
     0 DB2MONBP                   4096                1000         3.90         1

  4 record(s) selected.


================================================
 BPL#HITRA: Bufferpool data and index hit ratios
================================================

/* IBM_DB2MON */ select member, cast(substr(bp_name,1,20) as varchar(20)) as bp_name, case when (pool_data_l_reads) > 0 then decimal( ( double(pool_data_lbp_pages_found - pool_async_data_lbp_pages_found ) / (pool_data_l_reads + pool_temp
_data_l_reads) ) * 100, 5, 2 ) else 0 end as row_data_lbp_hitratio, case when (pool_col_l_reads) > 0 then decimal( ( double(pool_col_lbp_pages_found - pool_async_col_lbp_pages_found ) / (pool_col_l_reads + pool_temp_col_l_reads) ) * 100,
 5, 2 ) else 0 end as col_data_lbp_hitratio, case when (pool_index_l_reads) > 0 then decimal( ( double(pool_index_lbp_pages_found - pool_async_index_lbp_pages_found ) / (pool_index_l_reads + pool_temp_index_l_reads) ) * 100, 5, 2 ) else
0 end as index_lbp_hitratio from mon_get_bufferpool_diff where (pool_data_l_reads + pool_temp_data_l_reads + pool_index_l_reads + pool_temp_index_l_reads + pool_xda_l_reads + pool_temp_xda_l_reads + pool_col_l_reads + pool_temp_col_l_rea
ds) > 0 order by (pool_data_l_reads + pool_temp_data_l_reads + pool_index_l_reads + pool_temp_index_l_reads + pool_xda_l_reads + pool_temp_xda_l_reads + pool_col_l_reads + pool_temp_col_l_reads) desc with UR

MEMBER BP_NAME             ROW_DATA_LBP_HITRATIO COL_DATA_LBP_HITRATIO INDEX_LBP_HITRATIO
------ ------------------- --------------------- --------------------- ------------------
     0 IBMDEFAULTBP                        99.99                  0.00             100.00

  1 record(s) selected.


================================================
 BPL#READS: Bufferpool read statistics (overall)
================================================

/* IBM_DB2MON */ select member, cast(substr(bp_name,1,20) as varchar(20)) as bp_name, pool_data_l_reads, pool_data_p_reads, pool_index_l_reads, pool_index_p_reads, pool_col_l_reads, pool_col_p_reads, pool_read_time, case when ((pool_data
_p_reads + pool_temp_data_p_reads + pool_index_p_reads + pool_temp_index_p_reads + pool_xda_p_reads + pool_temp_xda_p_reads + pool_col_p_reads + pool_temp_col_p_reads)) > 0 then decimal( pool_read_time / double((pool_data_p_reads + pool_
temp_data_p_reads + pool_index_p_reads + pool_temp_index_p_reads + pool_xda_p_reads + pool_temp_xda_p_reads + pool_col_p_reads + pool_temp_col_p_reads)), 5, 2 ) else null end as avg_read_time from mon_get_bufferpool_diff where (pool_data
_l_reads + pool_temp_data_l_reads + pool_index_l_reads + pool_temp_index_l_reads + pool_xda_l_reads + pool_temp_xda_l_reads + pool_col_l_reads + pool_temp_col_l_reads) > 0 order by (pool_data_l_reads + pool_temp_data_l_reads + pool_index
_l_reads + pool_temp_index_l_reads + pool_xda_l_reads + pool_temp_xda_l_reads + pool_col_l_reads + pool_temp_col_l_reads) desc with UR

MEMBER BP_NAME             POOL_DATA_L_READS    POOL_DATA_P_READS    POOL_INDEX_L_READS   POOL_INDEX_P_READS   POOL_COL_L_READS     POOL_COL_P_READS     POOL_READ_TIME       AVG_READ_TIME
------ ------------------- -------------------- -------------------- -------------------- -------------------- -------------------- -------------------- -------------------- --------------
     0 IBMDEFAULTBP                  16617472                    2             15798777                    0                    0                    0                    0           0.00

  1 record(s) selected.
```

HELD LOG CONDITION


```
SELECT
con.application_handle,
con.application_id,
con.application_name,
con.client_pid,
uow.uow_start_time,
uow.uow_log_space_used
FROM
table(mon_get_connection(cast(null as bigint), -1)) as con,
table(mon_get_unit_of_work(null, -1)) as uow
WHERE
con.application_handle = uow.application_handle and
uow.uow_log_space_used != 0
ORDER BY uow.uow_start_time ;
```

SQL

- In my experience suboptimal SQL is the cause of 90% of performance problems
- STMM does a pretty good job of allocating and controlling shared memory resources
  - Bufferpools
  - Other memory areas
- When the DBM, DB CFG, Bufferpools configured correctly, either set to AUTOMATIC or a value, and RUNSTATS, REORGS and REBINDS done on a regular basis, most performance problems are SQL related
- Lock contention, high number of locktimeouts, deadlocks are second most problems seen based on experience
- SQL Tuning would be at least a whole other presentation or more and I will highlight some best practices

SQL Best Practices

- Developers EXPLAIN SQL before it goes into production
- Applications are tested in a test or QA environment prior to move to production
- DBAs assist developers in test or QA as needed in reviewing and monitoring SQL in development
- Developers, architects, data modelers work closely with DBAs in implementation of the logical model, with a key activity of INDEX DESIGN
- Big problem we find is little involvement of DBAs in initial Index Design

SQL Best Practices

- Use the Bonnie Baker method
  - Retrieve only the data required
  - Restrict before you join
  - Create indexes on LOCAL, ORDER BY and JOIN Predicates
  - AND create indexes to support columns most frequently accessed
    - Reports
- Use EXPLAIN
- Use Design Advisor with knowledge of the application and business requirements
  - Preclude creating indexes that only benefit low cost SQL
  - DO create indexes based on recommendations with high cost reduction but with knowledge of how it will or will not impact the workload
  - Run workload instead of individual SQL statements if possible
  - Work with the developers and business process owner to target important SQL

SQL Best Practices

- Use range predicates where possible
- Use Expression based indexing
- Review EXPLAIN sections with highest cost to target for tuning via rewrite or indexing or use of statistical views
- Review EXPLAIN for sorts and temp table usage and determine if index can improve
- Become aware of AI optimization improvements and see if you can take advantage of them

Lock Monitoring

- Drop the old db2detaildeadlock event monitor which is still installed with new Db2 installation
- Use the LOCKING event monitor
  - Should be created and enabled for all production databases and test or dev databases where locking issues are of interest
- Compile the db2evmonfmt Java program and copy stylesheet to path
    - C:\Program Files\IBM\SQLLIB\samples\java\jdbc
    - /home/<instname>/sqllib/samples/java/jdbc
- Run db2evmonfmt with desired options to format the LOCKING event monitor data, Sample command:
  - java db2evmonfmt -d hsprd -ue LOCKEVENTS -hours 24 -ftext > lockf27.txt

Sample Formatted Locking Report

Attributes	Requester	Owner
Participant No	1	2
Application Handle	021574	022868
Application ID	10.90.6.71.55912.241026043441	10.90.6.70.51802.241026020023
Application Name	db2jcc_application	db2jcc_application
Authentication ID	TQPOSAPP	TQGAMEUSER
Requesting AgentID	37004	45040
Coordinating AgentID	37004	45040
Agent Status	UOW Executing	UOW Executing
Application Action	No action	No action
Lock timeout value	15	0
Lock wait value	4500	0
Workload ID	1	1
Workload Name	SYSDEFAULTUSERWORKLOAD	SYSDEFAULTUSERWORKLOAD

Sample Formatted LOCKING Report

Current Activities of Participant No 1

Activity ID : 1
Uow ID : 2
Package Name : SYSSH200
Package Schema : NULLID
Package Version :
Package Token : SYSLVL01
Package Sectno : 2
Reopt value : none
Incremental Bind : no
Eff isolation : CS
Eff degree : 0
Actual degree : 1
Eff locktimeout : 15
Stmt first use : 2024-10-26-00.34.42.010023
Stmt last use : 2024-10-26-00.34.42.010023

Past Activities of Participant No 1

Activities not available

Current Activities of Participant No 2

Activity ID : 8
Uow ID : 45805
Package Name : SYSSH100
Package Schema : NULLID
Package Version :
Package Token : SYSLVL01
Package Sectno : 6
Reopt value : none
Incremental Bind : no
Eff isolation : UR
Eff degree : 0
Actual degree : 1
Eff locktimeout : 15
Stmt first use : 2024-10-26-00.34.25.901284

db2pd Background

- db2pd returns information without acquiring any

  locks or use of DB2 engine resources
- Since no locks are obtained, data returned by dbp2d may not be completely current or accurate
  - Zero use of database engine resources however is a good trade-off
- It is important to become familiar with db2pd as there is much information provided by db2pd that cannot be obtained through snapshot or event monitoring
- db2pd contains many options
- We will cover these new monitoring and problem determination information elements in this presentation

Using db2pd

- Command line tool
- Requires one of the following authorization
  - SYSADM authority level.
  - SYSCTRL authority level.
  - SYSMAINT authority level.
  - SYSMON authority level.
- No required connection or instance attachment
- For database level information to be retrieved, database must be active
- Standard options are:
  - -c command, read commands from input file
  - -r repeat, num sec count
  - -i interactive
  - -file, specifies output file

-transactions option

- Command: db2pd –db db2mon –trans

Database Partition 0 -- Database DB2MON -- Active -- Up 1 days 04:27:09

Transactions:

Address AppHandl [nod-index] TranHdl Locks State Tflag Tflag2 Firstlsn Lastlsn LogSpace SpaceReserved TID AxRegCnt GXID

0x027D1000 599 [000-00599] 99 0 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x000000009016 1 0

0x027D1A80 591 [000-00591] 100 0 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x000000008FF6 1 0

0x027D2500 589 [000-00589] 101 0 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x000000008FFD 1 0

0x027D2F80 588 [000-00588] 102 0 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x000000009012 1 0

0x027D3A00 9 [000-00009] 103 2 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x0000000003FA 1 0

0x027D4480 8 [000-00008] 104 0 READ 0x00000000 0x00000000 0x000000000000 0x000000000000 0 0 0x000000009014 1 0

-locks option

- Command: db2pd –db db2mon –locks

Database Partition 0 -- Database DB2MON -- Active -- Up 1 days 04:44:50

Locks:

Address	TranHdl	Lockname	Type	Mode	Sts	Owner	Dur	HldCnt	Att	ReleaseFlg
0x032DE790	103	4442324143434553BBCB10EC41	Internal P	..S	G	103	1	0	0x0000	0x40000000
0x032DE740	103	53514C4445464C5428DD630641	Internal P	..S	G	103	1	0	0x0000	0x40000000

Snippet from previous –trans output:

0x027D3A00	9	[000-00009]	103	2	READ	0x00000000	0x00000000	0x000000000000	0x000000000000	0	0	0x0000000003FA 1 0
0x027D4480	8	[000-00008]	104	0	READ	0x00000000	0x00000000	0x000000000000	0x000000000000	0	0	0x000000009014 1 0

Resolving Lock Contention with db2pd

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 16:39:33

db2pd –db GTSTST1 –locks –file /tmp/lockc.txt

Locks:

Address	TranHdl	Lockname	Type	Mode	Sts	Owner	Dur	HldCnt	Att	ReleaseFlg
0x0459C510	2	53514C4332453036BD4A32C841	Internal P	..	S	G	2	1	0	0x0000	0x40000000
0x0459CA10	3	53514C4332453036BD4A32C841	Internal P	..	S	G	3	1	0	0x0000	0x40000000
0x0459CA60	3	01000000010000000001007B0056	Internal V	.	S	G	3	1	0	0x0000	0x40000000
0x0459C9E8	3	53514C4445464C5428DD630641	Internal P	.	S	G	3	1	0	0x0000	0x40000000
0x0459EF90	2	02000300270000000000000052	Row	.	X	G	2	1	0	0x0008	0x40000002
0x0459CAB0	3	02000300270000000000000052	Row	.N	S	W	2	1	0	0x0000	0x00000001
0x0459C8F8	2	02000300000000000000000054	Table	.I	X	G	2	1	0	0x0000	0x40000002
0x0459CA88	3	02000300000000000000000054	Table	.I	S	G	3	1	0	0x0000	0x00000001

TranHdl 2 has an X lock on this row

Type of lock

Lock mode

TranHdl 3 is waiting on a lock held by TranHdl 2

-locks showlocks option

Locks:

Address TranHdl Lockname Type Mode Sts Owner Dur HldCnt Att ReleaseFlg

0x0459C510 2 53514C4332453036BD4A32C841 Internal P ..S G 2 1 0 0x0000 0x40000000 Pkg UniqueID 434c5153 36304532 Name c8324abd Loading = 0

0x0459CA10 3 53514C4332453036BD4A32C841 Internal P ..S G 3 1 0 0x0000 0x40000000 Pkg UniqueID 434c5153 36304532 Name c8324abd Loading = 0

0x0459CA60 3 0100000001000000001007B0056 Internal V ..S G 3 1 0 0x0000 0x40000000 Anchor 123 Stmt 1 Env 1 Var 1 Loading 0

0x0459C9E8 3 53514C4445464C5428DD630641 Internal P ..S G 3 1 0 0x0000 0x40000000 Pkg UniqueID 444c5153 544c4645 Name 0663dd28 Loading = 0

0x0459EF90 2 02000300270000000000000052 Row ..X G 2 1 0 0x0008 0x40000002 TbspaceID 2 TableID 3 RecordID 0x27

0x0459CAB0 3 02000300270000000000000052 Row .NS W 2 1 0 0x0000 0x00000001 TbspaceID 2 TableID 3 RecordID 0x27

0x0459C8F8 2 02000300000000000000000054 Table .IX G 2 1 0 0x0000 0x40000002 TbspaceID 2 TableID 3

0x0459CA88 3 02000300000000000000000054 Table .IS G 3 1 0 0x0000 0x00000001 TbspaceID 2 TableID 3

Bufferpool Monitoring with db2pd


```
Database Member 0 -- Database HSPRD -- Active -- Up 5 days 07:59:30 -- Date 2025-09-26-13.43.09.007199

Bufferpools:
First Active Pool ID      1
Max Bufferpool ID         3
Max Bufferpool ID on Disk 3
Num Bufferpools           7

Address            Id   Name               PageSz     PA-NumPgs  BA-NumPgs  BlkSize    NumTbsp    PgsToRemov CurrentSz  PostAlter  SuspndTSCt Automatic
0x0A00030031DCF280 1    IBMDEFAULTBP       4096       7921619    0          0          615        0          7921619    7921619    0          True
0x0A00030031DDC320 2    LARGEBUFF          32768      1500       0          0          3          0          1500       1500       0          True
0x0A00030031DE93C0 3    CONSOLEPOOL        32768      250        0          0          1          0          250        250        0          False
0x0A00030031D9B000 4096 IBMSYSTEMBP4K      4096       16         0          0          0          0          16         16         0          False
0x0A00030031DA80A0 4097 IBMSYSTEMBP8K      8192       16         0          0          0          0          16         16         0          False
0x0A00030031DB5140 4098 IBMSYSTEMBP16K     16384      16         0          0          0          0          16         16         0          False
0x0A00030031DC21E0 4099 IBMSYSTEMBP32K     32768      16         0          0          0          0          16         16         0          False

Bufferpool Statistics for all bufferpools (when BUFFERPOOL monitor switch is ON):

BPID DatLRds    DatPRds     HitRatio TmpDatLRds TmpDatPRds  HitRatio IdxLRds    IdxPRds     HitRatio TmpIdxLRds TmpIdxPRds  HitRatio
1    2140278010*52081822    100.00%  713883546  458         100.00%  8406920152*10618689    100.00%  2776       0           100.00%
2    531841     64822       97.63%   118        0           100.00%  194027     18079       92.97%   0          0           00.00%
3    1494       95          93.98%   0          0           00.00%   399        132         66.92%   0          0           00.00%
4096 0          0           00.00%   0          0           00.00%   0          0           00.00%   0          0           00.00%
4097 0          0           00.00%   0          0           00.00%   0          0           00.00%   0          0           00.00%
4098 0          0           00.00%   0          0           00.00%   0          0           00.00%   0          0           00.00%
4099 0          0           00.00%   0          0           00.00%   0          0           00.00%   0          0           00.00%

BPID DataWrts   IdxWrts     DirRds     DirRdReqs  DirRdTime  DirWrts    DirWrtReqs DirWrtTime
1    452501     14755261    1907340081 8390719    6557566    1050146    153642     95321
2    1104       11807       9324416    5536       11813      256        4          1
3    0          0           1525760    942        1447       0          0          0
4096 0          0           0          0          0          0          0          0
4097 0          0           0          0          0          0          0          0
4098 0          0           0          0          0          0          0          0
4099 0          0           0          0          0          0          0          0

BPID AsDatRds   AsDatRdReq AsIdxRds   AsIdxRdReq AsRdTime   AsDatWrts  AsIdxWrts  AsWrtTime
1    48508104   7942432    7216142    1088970    4749955    329204     8572346    575079
2    52233      14056      4442       720        10107      967        7426       133
3    5          2          0          0          1          0          0          0
Standard input
```

db2pd –d dbname -bufferpools

BPID	TotRdTime	TotWrtTime	VectIORds	VectIOReq	BlockIORds	BlockIOReq	FilesClose	NoVictAvl	UnRdPFetch
1	6664259	6650392	55724246	9031403	0	0	0	321697	0
2	28416	4153	56675	14776	0	0	0	171	10
3	144	0	5	2	0	0	0	0	0
4096	0	0	0	0	0	0	0	0	0
4097	0	0	0	0	0	0	0	0	0
4098	0	0	0	0	0	0	0	0	0
4099	0	0	0	0	0	0	0	0	0

Standard input

db2pd –d hsprd tcbstats


```
TCB Table Stats:
Address            TableName          SchemaNm Scans      UDI        RTSUDI             PgReorgs   NoChgUpdts Reads      FscrUpdates Inserts    Updates    Deletes    OvFlReads  OvFlCrtes
PgDictsCrt CCLogReads CCRemoteReqs CCLockWaits CCRemRetryLckWs StoreBytes BytesSaved
0x0A00030184568F88 H_UPN_UNIVERSAL_PR HS       0          0          0                  0          0          3          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003013B9BAA08 IEAR_IEA_REMARK    HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A000301115AEA08 SYSBOOT            SYSIBM   1          0          0                  0          0          1          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030198D20288 SYSVIEWS           SYSIBM   0          0          0                  0          0          130        0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030184565E88 H_ZIP_ZIP_CODE     HS       0          0          0                  0          0          16280      0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003012D479D88 IES_INS_EOI_STATUS HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003011482A308 SYSVIEWDEP         SYSIBM   0          0          0                  0          0          2394       0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003012D417108 AC_ACCT            HS       3          0          0                  0          0          68612      0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030114E2F188 CGID_CGI_DX_INFO   HS       0          68585      68585              0          0          1370350    1014        36510      32379      0          0          0
0          0          0           0          0             -          -
0x0A000301116ACE08 SYSPLAN            SYSIBM   1          1868       1868               25         1068       17443      92          55         2114       54         2          0
0          0          0           0          0             -          -
0x0A0003012D467188 CCBP_CUS_CNT_BILL_ HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003012D418F08 SYSSEQUENCES       SYSIBM   0          0          0                  0          0          108        0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A000301932A3C88 SYSROUTINEAUTH     SYSIBM   0          0          0                  0          0          327        0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003012D4D3C88 API_ADDL_PRV_INFO  HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030127036388 CSWR_CSV_WIZARD_RE HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003013B9AAB08 DIA_DIALOG         HS       0          0          0                  0          0          6          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A000301147EA608 SYSROUTINES        SYSIBM   45         0          0                  0          0          75326      0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A0003012D554E08 ATAS_AUTH_SRV_AUD  HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030114A55008 E999_EMT_999_ERR   HS       0          0          0                  0          0          0          0           0          0          0          0          0
0          0          0           0          0             -          -
0x0A00030114826E08 SYSPLANDEP         SYSIBM   0          68252      68252              1673       0          82247      65687       37509      5586       37309      1880       364
Standard input
```

db2pd –d hsprd –tcbstats index


```
TCB Index Stats:
Address             TableName          IID   PartID EmpPgDel   RootSplits BndrySplts PseuEmptPg EmPgMkdUsd Scans      IxOnlyScns KeyUpdates InclUpdats NonBndSpts PgAllocs   Merges     PseuDel
s   DelClean   IntNodSpl
0x0A0003018456A9E8 H_UPN_UNIVERSAL_PR 1     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A0003012D592468 IEAR_IEA_REMARK    2     n/a    0          0          0          0          0          0          0          0          0          0          1          0          0
    0          0
0x0A0003012D592468 IEAR_IEA_REMARK    1     n/a    0          0          0          0          0          0          0          0          0          0          1          0          0
    0          0
0x0A00030198D22068 SYSVIEWS           1     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301845678E8 H_ZIP_ZIP_CODE     1     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A0003012D54E168 IES_INS_EOI_STATUS 1     n/a    0          0          0          0          0          0          0          0          0          0          1          0          0
    0          0
0x0A0003011482C7E8 SYSVIEWDEP         3     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A0003011482C7E8 SYSVIEWDEP         2     n/a    0          0          0          0          0          1524       0          0          0          0          0          0          0
    0          0
0x0A0003011482C7E8 SYSVIEWDEP         1     n/a    0          0          0          0          0          9          0          0          0          0          0          0          0
    0          0
0x0A00030126FF8DE8 AC_ACCT            1     n/a    0          0          0          0          0          2309       904        0          0          0          0          0          0
    0          0
0x0A00030114E32BE8 CGID_CGI_DX_INFO   1     n/a    0          0          0          0          0          9150       0          0          0          825        847        0          0
    0          22
0x0A000301117AC8E8 SYSPLAN            4     n/a    0          0          0          0          0          754        753        0          0          0          0          0          54
    58         0
0x0A000301117AC8E8 SYSPLAN            3     n/a    0          0          0          0          0          5169       1          0          0          0          0          0          52
    57         0
0x0A000301117AC8E8 SYSPLAN            2     n/a    0          0          0          0          0          1          1          0          0          0          0          0          33
    28         0
0x0A000301117AC8E8 SYSPLAN            1     n/a    0          0          0          0          0          10813      2486       0          0          0          0          0          54
    42         0
0x0A0003012D468E68 CCBP_CUS_CNT_BILL_ 1     n/a    0          0          0          0          0          1          0          0          0          0          0          0          0
    0          0
0x0A0003012D3D6BE8 SYSSEQUENCES       2     n/a    0          0          0          0          0          1          0          0          0          0          0          0          0
    0          0
0x0A0003012D3D6BE8 SYSSEQUENCES       1     n/a    0          0          0          0          0          52         0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     7     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     6     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     5     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     4     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     3     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     2     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
    0          0
0x0A000301932A7968 SYSROUTINEAUTH     1     n/a    0          0          0          0          0          0          0          0          0          0          0          0          0
Standard input
```

Registry Settings

- Standard Registry settings we use in an OLTP Environment

```
(/home/hsprd)> db2set -all
[i] DB2_CAPTURE_LOCKTIMEOUT=ON
[i] DB2_USE_ALTERNATE_PAGE_CLEANING=ON
[i] DB2_EVALUNCOMMITTED=YES
[i] DB2_SKIPDELETED=YES
[i] DB2OPTIONS=-t
[i] DB2DBDFT=hsprd
[i] DB2COMM=TCPIP,SSL
[i] DB2_PARALLEL_IO=*
[i] DB2AUTOSTART=YES
[g] DB2SYSTEM=ha900hs
hsprd@[ha900hs]:
```

Recent Client Production Problem

- **STMM=OFF**

```
Self tuning memory                   (SELF_TUNING_MEM) = OFF
Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC(400112)
```

```
Address            Id   Name                PageSz     PA-NumPgs  BA-NumPgs  BlkSize    NumTbsp    PgsToRemov CurrentSz
PostAlter  SuspndTSCt Automatic
0x0A00020011DB6280 1    IBMDEFAULTBP        4096       256000     0          0          610        0          256000
256000     0          False
```

- 600GB Database
- 254 GB of RAM available on the LPAR
- IBMDEFAULTBP set to **256,000 4k pages**
  - **1GB bufferpool for 600 GB database**
  - **Slow claim adjudication reported**
  - Large Medicare Provider/Management Software
- **INSTANCE_MEMORY set to AUTOMATIC**

```
Global instance memory (% or 4KB)     (INSTANCE_MEMORY) = AUTOMATIC(33554432)
Member instance memory (% or 4KB)                        = GLOBAL
```

Recent Client Production Problem Summary

- INSTANCE_MEMORY set to AUTOMATIC
- STMM set to OFF
- IBMDEFAULTBP set to 256,000 4k pages
  - Not AUTOMATICALLY MANAGED
- STMM not ON and even if ON couldn’t manage the IBMDEFAULTBP
- INSTANCE_MEMORY reserving large amount of memory that can’t be used by the database
- 600GB database IO bound due to small bufferpool and inadequate settings
- No database monitoring whatsoever

Client Production Problem Solution

- ✓Enable STMM for the production database deferred to next restart
- ✓Alter IBMDEFAULTBP to SIZE AUTOMATIC deferred
- ✓No change needed at instance level except possible reduction in instance_memory since Websphere and the application run on the LPAR as well
- ✓Made changes and monitored

DB2DIAG.LOG/

- db2diag -g db:= -gi level=severe
- db2diag -g db:= -gi level=warning

- Administration notification log .nfy file

```
base sys utilities  sqleWatchDog Probe:20

ADM0503C  An unexpected internal processing error has occurred. All database
manager processes associated with this instance have been shutdown. Diagnostic
information has been recorded. Contact IBM Support for further assistance.
^^
2024-07-21-05.41.59.071212   Instance:hsprd   Node:000
PID:9961812(db2star2)   TID:1   Appid:none
base sys utilities  DB2StartMain Probe:911

ADM7513W  Database manager has started.
^^
2024-07-21-17.07.09.011375   Instance:hsprd   Node:000
PID:13435332(db2logmgr (HSPRD) 0)   TID:3988   Appid:none
data protection services  sqlpgArchiveLogFile Probe:3109   Database:HSPRD

ADM1844I  Started archive for log file "S0029848.LOG"
^^
```

References

- ✓ db2mon -- https://www.idug.org/news/an-introduction-to-db2mon
- db2mon documentation -- https://www.ibm.com/docs/en/db2/12.1.0?topic=tuning-collecting-reporting-performance-monitor-data
- https://public.dhe.ibm.com/ps/products/db2/info/vr115/pdf/en_US/db2_sys_mon_guide_115.pdf
- https://public.dhe.ibm.com/ps/products/db2/info/vr121/pdf/en_US/db2_sys_mon_guide_1212.pdf
- https://www.ibm.com/docs/en/db2/11.5.x?topic=commands-dsmtop-db2-text-based-monitoring-tool-command
- https://ibm.github.io/dmctop-wiki/Getting_started/run_dmctop/
- https://ibm.github.io/dmctop-wiki/

IDUG

2025

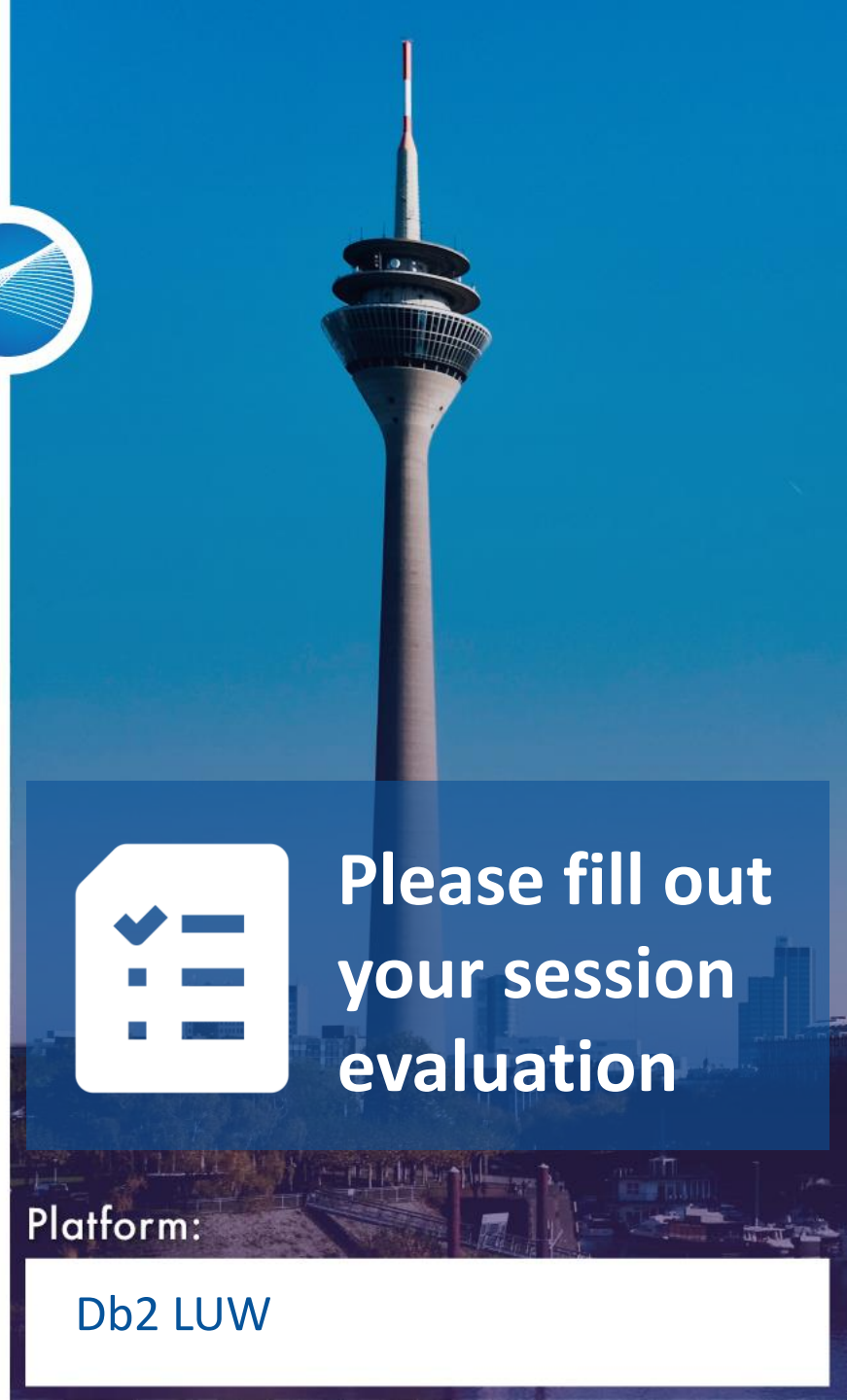
EMEA Db2 TECH CONFERENCE

Dusseldorf | October 26 - 30

Db2 Performance and Tuning

Phil Gunning, IBM Gold Consultant, MBA, CISSP, Gunning Technology Solutions, LLC

Session Code: C9



Please fill out your session evaluation

Platform:

Db2 LUW


Dusseldorf | October 26 - 30
IDUG
2025 EMEA Db2 Tech Conference