DB2 Performance & Tuning
(it’s all about the SQL!)

MICHAEL WINGFIELD
STROBE SOLUTION CONSULTANT

January 27, 2015
Mainframe Relevance

220 Billion Lines of COBOL Code (5 Billion Lines/Year)
Over 1 Million CICS Transaction per second per day
30 Billion Business Transactions per day on Mainframe
6 Trillion $ in Credit Card Payments per year on Mainframe
80% of Worldwide Corporate Data is on the Mainframe
50 Billion Devices generating transactions by 2020

The Mainframe is ideally suited to handle these transactions!
Moore’s Law

In 1965 one of the co-founders of Intel – Gordon Moore – described a trend that still holds true today – namely that processor speed doubles about every 18 months. However, that trend is nearing it’s end.

The Mainframe System z processor – zEC12 – is currently the fastest microprocessor in the industry at 5.5 GHz – but CMOS technology is not expected to ever exceed 7 GHz.

Application Performance Management is becoming more and more imperative to keep Operating Costs at a minimum since we can’t keep throwing hardware at performance issues.
CICS/Batch/DDF
DB2 Versions 9 & 10
CICS
Measurement Session Data - CICS

Profile Name:  
Job Name:  
Initiated On: 2011-12-05 10:30:43

Percent Time
- 20.21% Wait time
- 79.79% Execution time

Runtime margin of error: 0.99%
CPU margin of error: 2.07%

Clock Time
- Execution time: 298.14 sec
- Wait time: 205.61 sec
- System time: 1.89 sec
Session duration: 5 min 0.82 sec
Multiprocessor ratio: 1.1

Processor Time
- Service units: 3,551,664

Job Environment
- Job Number: JOB15156
- Address Space ID: 007F
- Program: ODFSIP
- Session Start Date: 201111205
- Session Start Time: 10:30:43
- System: z/OS 61.11.0
- Data Facility: DFSMS 1.11.0
- CPU Model: 2617-707
- SMF/System ID: SY1/SY1
- LPAR: SY1
- 64-bit Architecture: enabled
- Number of z/OS IP: 6

Parameters
- Estimated Session Time: 5 min
- Target Sample Size: 10,000
- Request Number: 64
- Request Type: Active
- Profile Run Date: 2011-12-05
- Profile Run Time: 10:40:23
- System(s) Requested: Selected Systems: SY1
- DDF Parameters: DDFQSTR=REQLOG,AUTHID

Names
- DB2 9.1.0 NFM
- CICS TS 4.1
- UNIX SERVICES

DB2 sysid
- DB2

Coupling Facilities
- CF1
- CF2

Statistics
- Total Samples Taken: 10,000
- Samples Processed: 10,000
**Program CPU Usage**

**Total CPU**

<table>
<thead>
<tr>
<th>Pseudo-Section</th>
<th>Description</th>
<th>Attributed</th>
<th>Combined</th>
<th>Solo</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>DB2 SYSTEM SERVICES</td>
<td>0.00</td>
<td>31.52</td>
<td>29.21</td>
<td>31.52</td>
</tr>
<tr>
<td>COMMON</td>
<td>COMMON AREA</td>
<td>0.00</td>
<td>11.74</td>
<td>11.12</td>
<td>11.74</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS SYSTEM SERVICES</td>
<td>0.00</td>
<td>0.79</td>
<td>0.73</td>
<td>0.79</td>
</tr>
<tr>
<td>IMSRRES</td>
<td>LE/370 LIBRARY SUBROUTINE</td>
<td>0.00</td>
<td>0.73</td>
<td>0.67</td>
<td>0.73</td>
</tr>
<tr>
<td>LELR</td>
<td>MV3 NUCLEUS</td>
<td>0.00</td>
<td>0.73</td>
<td>0.62</td>
<td>0.73</td>
</tr>
<tr>
<td>NUCLEUS</td>
<td>SUPERVISOR CONTROL</td>
<td>0.00</td>
<td>0.22</td>
<td>0.11</td>
<td>0.22</td>
</tr>
</tbody>
</table>

**.SYSTEM CPU**

<table>
<thead>
<tr>
<th>Module</th>
<th>RMODE</th>
<th>Description</th>
<th>Interval Length</th>
<th>Attributed</th>
<th>Combined</th>
<th>Solo</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSACCEC9</td>
<td>ANY</td>
<td></td>
<td>12288</td>
<td>0.00</td>
<td>38.43</td>
<td>38.18</td>
<td>38.43</td>
</tr>
<tr>
<td>GSVCEXIT</td>
<td>ANY</td>
<td></td>
<td>1416</td>
<td>0.00</td>
<td>4.66</td>
<td>4.27</td>
<td>4.66</td>
</tr>
<tr>
<td>PSTDRO9</td>
<td>ANY</td>
<td></td>
<td>40960</td>
<td>0.00</td>
<td>1.24</td>
<td>0.96</td>
<td>1.24</td>
</tr>
<tr>
<td>QZ715010</td>
<td>ANY</td>
<td></td>
<td>120512</td>
<td>0.33</td>
<td>1.17</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>QZ71EE14</td>
<td>ANY</td>
<td></td>
<td></td>
<td>0.51</td>
<td>0.51</td>
<td>0.51</td>
<td>0.51</td>
</tr>
<tr>
<td>ITCP1026</td>
<td>ANY</td>
<td></td>
<td></td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>GP14</td>
<td>ANY</td>
<td></td>
<td></td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>GSVCEXIT</td>
<td>ANY</td>
<td></td>
<td></td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>PSTDRO9</td>
<td>ANY</td>
<td></td>
<td></td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
</tbody>
</table>

**.USER CPU**

**CA Detector**

>200 Hours of CPU/Yr Per Region!!!
DB2 Activity by Query

>120 Hours of CPU/Yr Per Region!!!
Access Path – Non-Matching Index Scan W/Page Reference – Sort for Uniqueness and Order By

Note – All Leaf Pages in the Index are Scanned as well as Table Pages of Rows that Qualify.
Note – The First Two Columns of Index XIE1RULE have a Cardinality of only 1.

Index XIE1RULE is 2148 Leaf Pages
DECLARE PBC-LOCAL-CSR CURSOR FOR
SELECT DISTINCT CHAR(CR.RCPT_INCUR_STRT_DT,ISO) AS LOCAL_EFF_DT,
CHAR(CR.RCPT_INCUR_END_DT,ISO) AS LOCAL_EFF_TO_DT,
CHAR((CR.RCPT_INCUR_END_DT - 1 DAY),ISO) AS LOCAL_EFF_THRU_DT
FROM RULE R
INNER JOIN CLM_RULE CR
ON R.PLAN_PROFL_UID = CR.PLAN_PROFL_UID
WHERE R.PRFX_CD = :H
AND R.LOC_PLAN_CD ≠ 'STX'
AND R.STS_CD = '3'
AND R.STS_QUAL_CD = '02'
AND CR.PGM_CD = 'A'
AND CR.DLVY_METH_CD = '2'
AND CR.RCPT_INCUR_STRT_DT ≠ VALUE(CR.RCPT_INCUR_END_DT, DATE(:H))
AND VALUE(CR.RCPT_INCUR_END_DT,DATE(:H)) > DATE (:H)
ORDER BY LOCAL_EFF_DT ASC,
LOCAL_EFF_TO_DT ASC

Consider adding an index to Table ITS.RULE to change the Non-Matching Index Scan to a Matching Index Only Scan as follows:

Column 1 - PRFX_CD
2 - STS_QUAL_CD
3 - STS_CD
4 - LOC_PLAN_CD
5 - PLAN_PROFL_UID
DB2 Activity by Query

>90 Hours of CPU/Yr Per Region!!!
### Access Path – Non-Matching Index Scan W/Page Reference

Note – All Leaf Pages in the Index are Scanned as well as Table Pages of Rows that Qualify.
Index EDIPRNP2 is 3063 Leaf Pages
SQL Statement Text and Performance Recommendations

SELECT 'Y'
    INTO :H
FROM EDI_PROV_NPI
    EDIPRNP2(0)
WHERE (( PROV_NPI = :H
    OR PYEE_NPI = :H)
    OR ( PROV_PLAN_CD = :H
        AND PROV_NR = :H))
    AND SUBM_ID = :H
FETCH FIRST ROW ONLY
WITH UR

Consider adding an index to Table PROV.EDI_PROV_NPI to change the Non-Matching Index Scan to a Matching Index Only Scan as follows:

Column 1 - SUBM_ID
2 - PROV_NR
3 - PROV_PLAN_CD
4 - PROV_NPI
5 - PYEE_NPI

The Size of this New Index would be a little more than Double the Size of Index EDIPRNP2
Measurement Session Data - CICS
### DB2 Stored Procedure SQL Activity

**Profile Name:***  
**Job Name:**  
**Initiated On:** 2011-12-05 10:30:43

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Package</th>
<th>Schema</th>
<th>Collection</th>
<th>Bind Timestamp</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP2A6136</td>
<td>SYSPROC</td>
<td>DB2</td>
<td></td>
<td>12/05/2011 04:30:46 PM</td>
<td>43,896</td>
<td>0.000043, 0.143352, 19.426867</td>
<td>0, 0, 0.000243, 0.002697, 10.635605</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Target Statement**
- Statement Number: 1420  
- Statement Type: N/A  
- Statement Count: 6,056  
- Elapsed Time (sec): 0.002382, 0.143352, 14.425792  
- Parallel Task CPU Distribution (sec): 0, 0, 0.001275, 0.029287, 7.720092  
- CPU Time (sec): N/A

**SQL Statement Text**
```
SELECT JSTX.REQ_SEL_ID, JSTX.EFF_DT INTO @H, ... | SQLAF
```

**Run Time Statistics For Scans Executed By The SQL Statement**

<table>
<thead>
<tr>
<th>Type</th>
<th>Rows Processed</th>
<th>Rows Examined</th>
<th>Rows-Q Stage1</th>
<th>Rows-Q Stage2</th>
<th>Rows Inserted</th>
<th>Rows Deleted</th>
<th>Rows Updated</th>
<th>Get Page Requests</th>
<th>RI Page Scans</th>
<th>RI Page Deletes</th>
<th>LOB Page Scans</th>
<th>LOB Page Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX</td>
<td>10,005,065</td>
<td>30,530,303</td>
<td>176,065</td>
<td>17,409</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>774,545</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SEQ</td>
<td>6,376,110</td>
<td>6,376,110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>203,029</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Requesters**

<table>
<thead>
<tr>
<th>Requester Location</th>
<th>Authorization ID</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>1524</td>
<td>N/A, SNC</td>
<td>6,491</td>
<td>0.000424, 0.040089, 2.749489</td>
<td>0</td>
</tr>
<tr>
<td>SELECT</td>
<td>4291</td>
<td>N/A, SNC</td>
<td>1,461</td>
<td>0.000328, 0.011866, 0.47891</td>
<td>0</td>
</tr>
<tr>
<td>SELECT</td>
<td>1471</td>
<td>N/A, SNC</td>
<td>6,651</td>
<td>0.000054, 0.02158, 0.360755</td>
<td>0</td>
</tr>
</tbody>
</table>

>100 Hours of CPU/Yr Per Region!!!
Access Path Consists of Joins and SubSelects and No Index Only Access
SQL Analysis - DB2 Catalog Statistics

Consider Turning DB2 Compression On for any Table with more than 40 to 50 Pages

Note – Use the DSN1COMP Utility to determine Compression Ratio – if less than 25%, don’t turn on
SELECT JSTX.BEN_SEL_ID,  
   JSTX.EFF_DT  
 INTO :H,:H  
FROM BEN_SEL_TXT_XREF JSTX,  SELTXTX1(2)  
   BEN_SEL_VAL JVAL   BSELVAL1(2)  
WHERE (JVAL.GRP_NR = :H) 
   AND (JVAL.DIV_FROM <= :H) 
   AND (JVAL.DIV_THRU >= :H) 
   AND (JVAL.INCL_VAL_SW = 'Y') 
   AND (NOT EXISTS (SELECT 'X' 
                   FROM BEN_SEL_VAL JVAL2   BSELVAL1(2)  
                   WHERE (JVAL2.BEN_SEL_ID = JVAL.BEN_SEL_ID) 
                   AND (JVAL2.GRP_NR = :H) 
                   AND (JVAL2.DIV_FROM <= :H) 
                   AND (JVAL2.DIV_THRU >= :H) 
                   AND (JVAL2.INCL_VAL_SW = 'N'))) 
   AND (JSTX.BEN_SEL_ID = JVAL.BEN_SEL_ID) 
   AND (JSTX.PROD_TST_CD = 'P') 
   AND (JSTX.EFF_DT <= :H) 
   AND ((JSTX.EFF_THRU_DT IS NULL) 
      OR (JSTX.EFF_THRU_DT >= :H)) 
   AND (EXISTS (SELECT 'X' 
                 FROM BEN_TXT_PCE JPCE   BTXTPCE1(1)  
                 WHERE (JSTX.BEN_TXT_PCE_ID = JPCE.BEN_TXT_PCE_ID) 
                 AND (JPCE.HLTH_DENT_CD = :H)))  
FETCH FIRST 1 ROWS ONLY

Turn DB2 Compression on for Tables ACSIS.BEN_SEL_VAL and ACSIS.BEN_TXT_PCE. Also, Table ACSIS.BEN_SEL_TXT_XREF is only 83% Compressed – run a Reorg followed by Runstats. INCLUDE Column HLTH_DENT_CD in Index BTXTPCE1 for Index Only Access. Consider adding an index to Table ACSIS BEN_SEL_VAL to change the access of the Subselect to Index Only as follows:

Column 1 - GRP_NR  
2 - DIV_FROM  
3 - DIV_THRU  
4 - INCL_VAL_SW  
5 - BEN_SEL_ID

Note – INCLUDE (BSELVAL1) could be used for Index Only Access
DB2 SQL Activity

>10 Hours of CPU per Year per Region
SQL Analysis

DB2 Explain

DB2 Catalog Statistics
SQL Statement Text and Performance Recommendations

- CICS DBRM - 2740 Select (Stmt Count 76)

SELECT INT(CDE_DSC) INTO :H
FROM CAD_CDE_VAL CIP0XA71(0) ← 6,379 Rows/63 Pg
WHERE SUBSTR(CDE_SET_ID,1,5) = 'CIO7H'
   AND CDE_VAL = 'VALUE1' ← Colcard 3228

Change the SUBSTR to WHERE CDE_SET_ID LIKE 'CIO7H%' to change the access from a Non-Matching Index Scan to a Matching Index Scan with a MATCHCOLS of 1.

Another option would be to also create an index to increase the MATCHCOLS as follows:

Column 1 - CDE_VAL
2 - CDE_SET_ID

In DB2 V11, SUBSTR (from position 1) becomes indexable – similar to LIKE
DB2 SQL Activity

>100 Hours of CPU/Yr Per Region!!!
SQL Statement Text and Performance Recommendations

DECLARE CVRG_CURSOR CURSOR FOR
SELECT A.ROW_EXPR_DATE,
    B.CVG_DSPLY_CODE,
    B.CVG_DSPLY_SEQ_NUM,
    B.DESC_CODE
FROM RISK_CVG A,
    XPKRISK_CVG(1)
CVG_DESC B
WHERE A.PLCY_SEQ_NUM = :H
    AND :H BETWEEN A.ROW_EFF_DATE AND A.ROW_EXPR_DATE
    AND A.VER_SEQ_NUM = 0
    AND A.RISK_SEQ_NUM = :H
    AND A.ROW_EXPR_DATE > :H
    AND A.CLASS_CODE <> '009460'
    AND A.CVG_CODE = B.CVG_CODE
    AND A.CVG_TYPE_CODE = B.CVG_TYPE_CODE
    AND A.CVG_SUB_TYPE_CODE = B.CVG_SUB_TYPE_CODE
ORDER BY B.CVG_DSPLY_SEQ_NUM
WITH UR

Consider re-writing the BETWEEN predicate to change from a Stage 2 Predicate to a Stage 1 Predicate as follows:

    AND A.ROW_EFF_DATE <= :H
    AND A.ROW_EXPR_DATE >= :H

In DB2 V11, this is re-written to be indexable
Another consideration would be to add an index to Table RISK_CVG to increase the MATCHCOLS from 1 to 4 and change the access to Index Only as follows:

Column 1 - PLCY_SEQ_NUM
2 - RISK_SEQ_NUM
3 - VER_SEQ_NUM
4 - ROW_EXPR_DATE
5 - ROW_EFF_DATE
6 - CLASS_CODE
7 - CVG_CODE
8 - CVG_TYPE_CODE
9 - CVG_SUB_TYPE_CODE

Run Time Statistics after Index was added

<table>
<thead>
<tr>
<th>Type</th>
<th>Rows Processed</th>
<th>Rows Examined</th>
<th>Rows-Q Stage1</th>
<th>Rows-Q Stage2</th>
<th>Rows Inserted</th>
<th>Rows Deleted</th>
<th>Rows Updated</th>
<th>Get Pages Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDX</td>
<td>127</td>
<td>10,969</td>
<td>19,716</td>
<td>9,857</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8,129</td>
</tr>
<tr>
<td>SEQD</td>
<td>9,859</td>
<td>9,859</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4,970</td>
</tr>
</tbody>
</table>

Over 90% Reduction in Getpages

Higher Throughput & Faster Response Time
Batch
Daily Batch Job Consuming Over 80 Hours of CPU per Year; Waits for CPU Over 140 Hours per Year; DB2 Wait Over 50 Hours per Year.
>12 Hours of CPU per Year Associated with this one SQL Statement
Table REF.LOC_REL is a Candidate for DB2 Compression – 812 Pages
Turn DB2 Compression on for Table REF.LGC_REL. There are over 12 million Selects against a 28,000 row table - many duplicate Selects. Consider checking programmatically to see if the program is about to access the same row that was previously accessed, then don’t issue the Select.

Probably 10 of the 12 Hours of CPU Yearly Could Be Saved
DB2 Activity by Query

Profile Name: [Redacted]  Job Name: [Redacted]  Initiated On: 2011-12-05 23:15:35

<table>
<thead>
<tr>
<th>DBRM/Package</th>
<th>Bind Timestamp</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
<th>CPU %</th>
<th>Wait %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average Max Total</td>
<td>SP CPU Total CPU Average Max Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2478600D</td>
<td>11 15 2011 04:25:59 PM</td>
<td>21,169,700</td>
<td>0.000023 N/A 490.316097</td>
<td>0 0</td>
<td>0.000001 N/A 202.072561 N/A</td>
<td>17.36</td>
<td>0.51</td>
</tr>
<tr>
<td>F247023D</td>
<td>11 15 2011 04:20:23 PM</td>
<td>12,523,446</td>
<td>0.000032 N/A 401.362054</td>
<td>0 0</td>
<td>0.000014 N/A 174.35708 N/A</td>
<td>14.98</td>
<td>0.05</td>
</tr>
<tr>
<td>F247055D</td>
<td>11 29 2011 04:42:52 PM</td>
<td>7,105,119</td>
<td>0.00212 N/A 1508.302006</td>
<td>0 0</td>
<td>0.000223 N/A 162.560291 N/A</td>
<td>13.97</td>
<td>35.05</td>
</tr>
</tbody>
</table>

Target Statement
- Statement Number
- Statement Type
- Statement Count
- Elapsed Time (sec)
- Parallel Task CPU Distribution (sec)
- CPU Time (sec)
- CPU %
- Wait %

SQL Statement Text
- DECLARE MMY_HIST_CURSOR CURSOR FOR SELECT 'S' [...]

Run Time Statistics For Scans Executed By The SQL Statement
- Type
- Rows Processed
- Rows Examined
- Rows-Q Stage1
- Rows-Q Stage2
- Rows Inserted
- Rows Deleted
- Rows Updated
- Cnt Page Requests
- RI Page Scans
- RI Page Deletes
- LOB Page Scans
- LOB Page Updates

<table>
<thead>
<tr>
<th>Type</th>
<th>Rows Processed</th>
<th>Rows Examined</th>
<th>Rows-Q Stage1</th>
<th>Rows-Q Stage2</th>
<th>Rows Inserted</th>
<th>Rows Deleted</th>
<th>Rows Updated</th>
<th>Cnt Page Requests</th>
<th>RI Page Scans</th>
<th>RI Page Deletes</th>
<th>LOB Page Scans</th>
<th>LOB Page Updates</th>
</tr>
</thead>
</table>
### SQL Analysis

#### DB2 Catalog Statistics

<table>
<thead>
<tr>
<th>Table</th>
<th>DBNAME</th>
<th>TSNAME</th>
<th>CARDF</th>
<th>NUM DEP MQTS</th>
<th>SPLIT ROWS</th>
<th>NPAGES</th>
<th>PCT PAGES</th>
<th>PCT ROW COMP</th>
<th>PARTKEY COL NO</th>
<th>AVG ROW LEN</th>
<th>STATSTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACSS.SHPY</td>
<td>ACSSHDB</td>
<td>ACSSHFP</td>
<td>1862706593</td>
<td>0</td>
<td>0</td>
<td>25415737</td>
<td>99</td>
<td>100</td>
<td>0</td>
<td>55</td>
<td>2011-12-03 15:06:53</td>
</tr>
<tr>
<td>ACSS.SHLN</td>
<td>ACSSHDB</td>
<td>ACSSHLP</td>
<td>9473302148</td>
<td>0</td>
<td>0</td>
<td>52138781</td>
<td>99</td>
<td>100</td>
<td>0</td>
<td>200</td>
<td>2011-11-10 14:57:32</td>
</tr>
<tr>
<td>ACSS.CRPY</td>
<td>ACSSCRDB</td>
<td>ACSSCRFP</td>
<td>824756822</td>
<td>0</td>
<td>0</td>
<td>8884583</td>
<td>99</td>
<td>100</td>
<td>0</td>
<td>52</td>
<td>2011-04-17 07:30:43</td>
</tr>
<tr>
<td>ACSS.CRLN</td>
<td>ACSSCRDB</td>
<td>ACSSCRLP</td>
<td>619812983</td>
<td>0</td>
<td>0</td>
<td>13041145</td>
<td>99</td>
<td>100</td>
<td>0</td>
<td>78</td>
<td>2011-06-18 02:46:45</td>
</tr>
<tr>
<td>ACSS.CRCL</td>
<td>ACSSCRDB</td>
<td>ACSSCRCLP</td>
<td>103093919</td>
<td>0</td>
<td>0</td>
<td>6023358</td>
<td>99</td>
<td>100</td>
<td>0</td>
<td>212</td>
<td>2011-12-03 14:04:18</td>
</tr>
</tbody>
</table>
DECLARE MM_HIST_CURSOR CURSOR FOR
SELECT 'S', JSHLN.PATIENT_SOC_SEC_NR, JSHLN.SVC_FROM_DT,
    JSHPY.CLM_NR, JSHPY.SRC_OF_CLM, JSHPY.CLM_SPLIT_IND,
    JSHPY.LINE_ITEM, JSHPY.VERSION, JSHPY.ADJUD_DT,
    JSHPY.DISPB, JSHPY.PMT_SEQ_NR, JSHLN.PATIENT_FIRST_INIT,
    ,
    JSHPY.REDUCT_AMT2, JSHPY.REDUCT_CD3, JSHPY.REDUCT_AMT3,
    JSHPY.REDUCT_CD4, JSHPY.REDUCT_AMT4, JSHPY.REDUCT_CD5,
    JSHPY.REDUCT_AMT5
FROM SHLN JSHLN, IXSHLN01(9) ← 947M Rows/52M Pg
    IXSHPY01(2) ← 1.7B Rows/25M Pg
WHERE JSHLN.CONTRACT_NR = :H
    AND JSHLN.CONTRACT_PREFIX = :H
    AND JSHLN.CLM_NR = JSHPY.CLM_NR
    AND JSHLN.SRC_OF_CLM = JSHPY.SRC_OF_CLM
    AND JSHLN.CLM_SPLIT_IND = JSHPY.CLM_SPLIT_IND
    AND JSHLN.LINE_ITEM = JSHPY.LINE_ITEM
    AND JSHLN.VERSION = JSHPY.VERSION
    AND JSHLN.ADJUD_DT = JSHPY.ADJUD_DT
    AND JSHLN.DISPB = JSHPY.DISPB
    AND JSHLN.ADJUD_DT < :H
    AND JSHPY.ADJUD_EXPLN IN ('960', '9X2')
    AND ( JSHLN.ADJUSTMENT_DT IS NULL
            OR JSHLN.ADJUSTMENT_DT >= :H)
    AND ((JSHLN.SRC_OF_CLM NOT IN ('NA', 'NB', 'NC', 'ND', 'NE', 'NG',
        'NH', 'NN', 'NP', 'NS', 'NV', 'NW')
    AND JSHLN.CONTRACT_PREFIX IN ('BLS', 'BSO', 'FXB'))
            OR JSHLN.CONTRACT_PREFIX NOT IN ('BLS', 'BSO', 'FXB'))
    UNION ALL
SELECT 'C', JCRCL.PATIENT_SOC_SEC_NR, JCRCL.SVC_FROM_DT,
      JCRPY.CLM_NR, JCRPY.SRC_OF_CLM, JCRPY.CLM_SPLIT_IND,
      JCRPY.LINE_ITEM, JCRPY.VERSION, JCRPY.ADJUD_DT,
      JCRPY.DISP, JCRPY.PMT_SEQ_NR, JCRCL.PATIENT_FIRST_INIT, 
      
      JCRPY.REDUCT_AMT2, JCRPY.REDUCT_CD3, JCRPY.REDUCT_AMT3,
      JCRPY.REDUCT_CD4, JCRPY.REDUCT_AMT4,
      JCRPY.REDUCT_CD5, JCRPY.REDUCT_AMT5
FROM CRCL JCRCL,
     CRPY JCRPY
WHERE (JCRLN.CONTRACT_NR = :H)
      AND (JCRLN.CONTRACTPREFIX = :H)
      AND (JCRCL.CONTRACT_NR = JCRCL.CONTRACT_NR)
      AND (JCRCL.CONTRACT_PREFIX = JCRCL.CONTRACT_PREFIX)
      AND (JCRCL.CLM_NR = JCRCL.CLM_NR)
      AND (JCRLN.SRC_OF_CLM = JCRCL.SRC_OF_CLM)
      AND (JCRCL.CLM_SPLIT_IND = JCRCL.CLM_SPLIT_IND)
      AND (JCRCL.VERSION = JCRCL.VERSION)
      AND (JCRCL.CONTRACT_NR = JCRPY.CONTRACT_NR)
      AND (JCRCL.CONTRACT_PREFIX = JCRPY.CONTRACT_PREFIX)
      AND (JCRCL.CLM_NR = JCRPY.CLM_NR)
      AND (JCRLN.SRC_OF_CLM = JCRPY.SRC_OF_CLM)
      AND (JCRPY.CLM_SPLIT_IND = JCRPY.CLM_SPLIT_IND)
      AND (JCRPY.LINE_ITEM = JCRPY.LINE_ITEM)
      AND (JCRPY.VERSION = JCRPY.VERSION)
      AND (JCRCL.ADJUSTMENT_DT = JCRPY.ADJUSTMENT_DT)
      AND (JCRPY.DISP = JCRPY.DISP)
      AND (JCRPY.ADJUSTMENT_DT < :H)
      AND (JCRLN.ADJUSTMENT_DT IS NULL 
           OR JCRLN.ADJUSTMENT_DT >= :H)
      AND ((JCRPY.ADJUD_SYS_CD = 'C'
            AND JCRPY.LINE_ITEM = 0001)
           OR (JCRPY.ADJUD_SYS_CD IN ('M', '4')
               AND JCRPY.ADJUD_EXPLN IN ('960', '9X2')))
      AND (((JCRCL.SRC_OF_CLM NOT IN ('NA', 'NB', 'NC', 'ND', 'NE', 'NG',
                                    'NH', 'NN', 'NP', 'NS', 'NV', 'NW'))
             AND (JCRCL.CONTRACT_PREFIX IN ('BLS', 'BSO', 'FXB')))
           OR (JCRCL.CONTRACT_PREFIX NOT IN ('BLS', 'BSO', 'FXB')))
ORDER BY 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

SQL Statement Text and Performance Recommendations
Is there an Archive Strategy for these extremely large tables? Consider using Solid State Devices to improve the DB2 I/O Wait Time. DB2 Version 10 may provide some increased efficiency for the Order By Sort which on average is sorting around 800 rows. Consider Multi-Row Fetch – 100 Rows at a time. Ensure access to these five tables is in CONTRACT_NR order to invoke Dynamic Prefetch.

Using Columns From Multiple Tables in the Order By Clause Automatically Requires a Sort

Implementing Multi-Row Fetch Requires the Use of an Array within the Program but can save 50% of the CPU.
Measurement Session Data - Batch

Just Your Typical 36 Hour Batch Job!!!

Over 1800 Hours of Runtime Per Year; Over 350 Hours of CPU Per Year; Over 500 Hours Waiting For CPU Per Year; Over 1000 Hours of DB2 Wait
Over 20 Million Getpages Associated with this one SQL Statement
Index Only Access on Two of the Three Tables – Nested Loop Joins – Overall Not Too Bad
Table TP6L000 with Over 19,000 Pages is a Candidate for DB2 Compression
```
SELECT A.PBUSGR_NO
INTO :H
FROM PBUSGRKY A,
    PBUSGRTY B,
    PBUSGRD C
WHERE RMBRCP_TXIDFRMT_CD = :H
AND RMBRCP_TXIDNO = :H
AND A.PBUSGR_NO <> :H
AND ( PRVDR_ID = :H
    OR ADDR_NO = :H
    OR NTWK_ID = 0
    OR PRVCLS_CD > ' ' 
    OR PBUSGRKY_ROLE_NM > ' ' 
    OR PBUSGRKY_PRDGRP_CD > ' ' 
    OR PBUSGRKY_CTGRY_CD > ' ' 
    OR PROD_CD > ' ' 
    OR SPCLT_MAJCLS_CD > ' ' 
    OR PRVTY_CD = :H
    OR GEOAREA_CD > ' ' 
    OR PRVORG_NO = 0)
AND PBUSGRKY_EFF_DT <= :H
AND PBUSGRKY_EXPRTN_DT > :H
AND PBUSGRKY_SPSD_DT = '0001-01-01'
AND A.PBUSGR_NO = B.PBUSGR_NO
AND A.PBUSGR_NO = C.PBUSGR_NO
AND PBUSGRTY_EFF_DT <= :H
AND PBUSGRTY_EXPRTN_DT > :H
AND PBUSGRTY_SPSD_DT = '0001-01-01'
AND ( BUSGRTY_CD = :H
    OR 0 = 1)
AND PBUSGRD_EFF_DT <= :H
AND PBUSGRD_EXPRTN_DT > :H
AND PBUSGRD_SPSD_DT = '0001-01-01'
AND BUSGRST_CD NOT IN ('NLV','CLSD','IPAR','FRZN','PEND')
```

Using OR 0 = 1 Discourages DB2 From Using Index XP6L0002

Stage 2 Predicate
Performance Recommendations

Turn DB2 Compression on for Table TP6L000. Using OR 0=1 (which changes the predicate to a Stage 2 predicate) is done to keep DB2 from choosing a particular index (in this case probably XPL60002 - did this index not perform well). Consider using concatenate as follows: BUSGRTY_CD = :H||’’ which keeps it a Stage 1 predicate. Also, consider changing predicate AND BUSGRST_CD NOT IN ('NLV','CLSD','IPAR','FRZN','PEND') from NOT IN to IN and name the other four values for that column.

Using the Concatenate Instead of the OR 0 = 1, Keeps the Predicate Stage 1

Change From NOT IN to IN and Use the Other Four Values
50 Hours of CPU per Year
330 Hours of DB2 Wait

Wait Time by Module
This Single SQL Statement is Responsible For Nearly All of the CPU Usage and DB2 Wait Time
Low Number of Matching Columns for Index OTH2_05_CPOS, List Prefetch is On, and No Index Only Access
Both Index OTH2_05_CPOS and FK1_03_CPOS have Columns with 0 Cardinality Indicating that Runstats was Run at Some Time When the Tables were Empty.
--- Batch - DBRM --- 17 Open (Stmt Count 3,624) 
50 Fetch - 6,457

DECLARE AGENT_CURSOR CURSOR FOR

SELECT CPG_ID, CPG_CRT_DTE, CPG_CRT_TME, CP_ID, 
   CP_AGNT_ID, CP_PYMT_MTHD_CDE, CP_AMT 
FROM CPO_PAYMENT_GROUP, OTH2_05_CPOS(1)L 
   CPO_PAYMENT 
WHERE CPG_ID = CP_CPG_ID 
   AND CP_AGNT_ID = :H 
   AND CP_STATUS = 'U' 
   AND CP_PYMT_MTHD_CDE = :H 
   AND CP_AUTH_NUM = '' 
   AND CPG_FLT_NUM IS NULL 
   AND CPG_CRT_DTE = :H 
   AND :H - CPG_CRT_TME <= :H 
   AND CPG_CRT_TME - :H <= :H 
ORDER BY CPG_CRT_TME

After a Reorg, run a Full Runstats on all Columns and Indexes for both of these Tables - some of the columns on these tables have 0 for Column Cardinality which means that Runstats was run when the table was empty. Consider doing the arithmetic in the program rather than in the SQL - perhaps a Range Predicate could then be used and would be a Stage 1 Predicate as follows:

AND CPG_CRT_TME BETWEEN :H AND :H

Consider Doing the Arithmetic in the Program Rather Than in the SQL
Consider adding indexes to these two Tables to increase the MATCHCOLS and change the access to Index Only for both Tables as follows:

Column 1 - AGNT_ID
2 - CP_STATUS
3 - CP_PYMT_MTHD_CDE
4 - CP_AUTH_NUM
5 - CP_CPG_ID
6 - CP_ID
7 - CP_AMT

Column 1 - CPG_CRT_DTE
2 - CPG_FLT_NUM
3 - CPG_CRT_TME
4 - CPG_ID

Changed from 90 Minutes Down to 5 Minutes by Increasing the Number of Matching Columns and Changing the Access Path to Index Only
DDF
DDF – Distributed Data Facility

Mainframe
DB2 Subsystem A

DB2 Database

SNA - TCP/IP

Mainframe
DB2 Subsystem B

DB2 Database

Workstation

Early 1990s

Neon
Shadow

DB2 Connect

Oracle Gateway

DB2 Connect
Typical Transaction Flow from Browser to Mainframe

4 out of 5 Consumers use Smartphones

Today!
According to IBM, DDF is the #1 DB2 for z/OS Interface for Client-Server Applications – or - represents the fastest growing access into DB2.

These Client-Server Applications may be using CICS Transaction Gateway or Websphere MQ or Websphere Application Server utilizing JDBC or ODBC.

And, most of this traffic is Dynamic SQL!!

Finding and Tuning this Dynamic SQL Traffic from DDF is critical to reduce MIPS associated with DDF and iStrobe provides everything needed to do Performance & Tuning in the DDF Address Space.
Most of the CPU shows up in Pre-Emptible SRBs called Enclaves under Control of Work Load Manager
## DB2 DDF SQL Activity

### Profile Name: [Redacted]  Job Name: [Redacted]  Initiated On: 2011-08-29 13:56:04

<table>
<thead>
<tr>
<th>DBRM/Package</th>
<th>Bind Timestamp</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Max</td>
<td>Total</td>
</tr>
<tr>
<td>SYSLN300</td>
<td>08/29/2011 06:56:05 PM</td>
<td>1,463,150</td>
<td>0.000196e-3</td>
<td>10.666927</td>
<td>7601.988611</td>
</tr>
<tr>
<td>SYSLN200</td>
<td>08/29/2011 06:56:05 PM</td>
<td>54,043</td>
<td>0.010598</td>
<td>14.249172</td>
<td>576.648482</td>
</tr>
<tr>
<td>SYSSTAT</td>
<td>08/29/2011 06:56:10 PM</td>
<td>81</td>
<td>0.001692</td>
<td>22.963359</td>
<td>40.674441</td>
</tr>
<tr>
<td>CG2211A</td>
<td>08/29/2011 06:56:34 PM</td>
<td>380</td>
<td>0.006904</td>
<td>0.271071</td>
<td>1.565576</td>
</tr>
<tr>
<td>SYSLN200</td>
<td>08/29/2011 07:00:37 PM</td>
<td>199</td>
<td>0.03452</td>
<td>3.88138</td>
<td>6.869423</td>
</tr>
<tr>
<td>TR21011A</td>
<td>08/29/2011 07:00:19 PM</td>
<td>4,005</td>
<td>0.001267</td>
<td>0.097439</td>
<td>5.137725</td>
</tr>
<tr>
<td>ABJSONET</td>
<td>08/29/2011 06:56:35 PM</td>
<td>101</td>
<td>0.003164</td>
<td>0.78926</td>
<td>0.318688</td>
</tr>
<tr>
<td>ASNMUFDT</td>
<td>08/29/2011 06:56:35 PM</td>
<td>92</td>
<td>0.007283</td>
<td>0.337113</td>
<td>0.670081</td>
</tr>
<tr>
<td>CAG8C11A</td>
<td>08/29/2011 06:56:35 PM</td>
<td>551</td>
<td>0.001239</td>
<td>0.188692</td>
<td>0.626588</td>
</tr>
<tr>
<td>TR2N11A</td>
<td>08/29/2011 07:00:19 PM</td>
<td>462</td>
<td>0.005681</td>
<td>0.070243</td>
<td>2.738265</td>
</tr>
<tr>
<td>ABSPROC</td>
<td>08/29/2011 06:56:36 PM</td>
<td>4</td>
<td>0.058555</td>
<td>0.073291</td>
<td>0.234219</td>
</tr>
<tr>
<td>SYSSH100</td>
<td>08/29/2011 07:00:00 PM</td>
<td>30</td>
<td>0.227348</td>
<td>6.573538</td>
<td>6.520425</td>
</tr>
<tr>
<td>DSNREX0</td>
<td>08/29/2011 07:00:41 PM</td>
<td>20</td>
<td>0.001689</td>
<td>0.01432</td>
<td>0.30784</td>
</tr>
<tr>
<td>RAARDB1</td>
<td>08/29/2011 06:57:36 PM</td>
<td>3</td>
<td>0.030835</td>
<td>0.015379</td>
<td>0.032493</td>
</tr>
<tr>
<td>RAARDB2</td>
<td>08/29/2011 06:56:38 PM</td>
<td>15</td>
<td>0.012958</td>
<td>0.150047</td>
<td>0.194363</td>
</tr>
<tr>
<td>RAARDB3</td>
<td>08/29/2011 06:56:39 PM</td>
<td>6</td>
<td>0.02029</td>
<td>0.006265</td>
<td>0.121737</td>
</tr>
<tr>
<td>TR91011A</td>
<td>08/29/2011 07:00:20 PM</td>
<td>15</td>
<td>0.000264</td>
<td>0.001855</td>
<td>0.004255</td>
</tr>
<tr>
<td>DSNAPC08</td>
<td>08/29/2011 07:00:09 PM</td>
<td>2</td>
<td>0.011773</td>
<td>0.023845</td>
<td>0.023845</td>
</tr>
<tr>
<td>BIUT111A</td>
<td>08/29/2011 07:03:21 PM</td>
<td>3</td>
<td>0.004296</td>
<td>0.00279</td>
<td>0.048287</td>
</tr>
<tr>
<td>CO19111A</td>
<td>08/29/2011 06:56:09 PM</td>
<td>4</td>
<td>0.000185</td>
<td>0.000236</td>
<td>0.00074</td>
</tr>
</tbody>
</table>
## DB2 DDF SQL Activity

### Statement Details

<table>
<thead>
<tr>
<th>Statement Type</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (usec)</th>
<th>CPU Time (usec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>1.772</td>
<td>0.086424</td>
<td>0.419070</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL Statement Text

**SELECT SQLSTATEM1 WHERE ...**

### Execution Details

<table>
<thead>
<tr>
<th>Executing Statement</th>
<th>Elapsed Time (usec)</th>
<th>Parallel Task CPU Distribution (usec)</th>
<th>CPU Time (usec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FETCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREPARE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Run Time Statistics

- **INDEX:**
  - Rows Processed: 26,164,078
  - Rows Examined: 31,400
  - Rows Stage 2: 49
  - Rows Updated: 0
  - Rows Deleted: 0
  - Requests: 235,546

- **SEDO:**
  - Rows Processed: 504
  - Rows Examined: 45
  - Rows Stage 2: 49
  - Rows Updated: 0
  - Rows Deleted: 0
  - Requests: 0

- **SELECT:**
  - Rows Processed: 518
  - Rows Examined: 49
  - Rows Stage 2: 49
  - Rows Updated: 0
  - Requests: 0

### Additional Notes

- The image shows a detailed view of a DB2 DDF SQL activity report, including execution details and run time statistics for various SQL statements.
- Specific queries and their performance metrics are highlighted, such as the SELECT statement and its associated rows processed and examiners.
- The report provides insights into the efficiency and performance of SQL queries in a DB2 environment.
### SQL Analysis

**DB2 Catalog Statistics**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>CARDF</th>
<th>NUM DCP MOTS</th>
<th>SPLIT ROWS</th>
<th>NPAGES</th>
<th>POT PAGES</th>
<th>POT ROW COMP</th>
<th>PARTKEY COL ND</th>
<th>AVG ROW LEN</th>
<th>STATETIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXX13</td>
<td>TXT13X1</td>
<td>200530</td>
<td>0</td>
<td>30857</td>
<td>00</td>
<td>02</td>
<td>0</td>
<td>60</td>
<td>56</td>
<td>2011-05-28 19:30:31</td>
</tr>
<tr>
<td>TXX13</td>
<td>TXT13X1</td>
<td>6841404</td>
<td>0</td>
<td>823794</td>
<td>90</td>
<td>100</td>
<td>0</td>
<td>383</td>
<td>56</td>
<td>2011-05-28 19:30:31</td>
</tr>
</tbody>
</table>

### DB2 Explain

<table>
<thead>
<tr>
<th>QBLOCK NO</th>
<th>PLAN NO</th>
<th>METHOD</th>
<th>INDEX ONLY</th>
<th>CREATOR</th>
<th>TNAME</th>
<th>ADDRESS CREATOR</th>
<th>ADDRESS NAME</th>
<th>ADDRESS TYPE</th>
<th>MATCH COLS</th>
<th>SORTB</th>
<th>CORTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>O</td>
<td></td>
<td>DBC4TXD1</td>
<td>TXT13G_CORRESPOND</td>
<td>DBC4TXD1</td>
<td>TXX13G</td>
<td>1</td>
<td>0</td>
<td>ANN</td>
<td>NNNN</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>O</td>
<td></td>
<td>DBC4TXD1</td>
<td>TXT13G_CLAIM</td>
<td>DBC4TXD1</td>
<td>TXX13U</td>
<td>2</td>
<td>2</td>
<td>NNNN</td>
<td>NNNN</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>O</td>
<td></td>
<td>DBC4TXD1</td>
<td>TXT13G_CORRESPOND</td>
<td>DBC4TXD1</td>
<td>TXX13G</td>
<td>1</td>
<td>0</td>
<td>ANN</td>
<td>NNNN</td>
</tr>
</tbody>
</table>

### Indexes (SYINDEXES)

<table>
<thead>
<tr>
<th>NAME</th>
<th>CREATOR</th>
<th>CLUSTER ACT</th>
<th>CLUSTER SPAC</th>
<th>FIRST KEYCARD</th>
<th>FULL KEYCARD</th>
<th>NLEAF</th>
<th>NLEVELS</th>
<th>CLUSTER RATIO</th>
<th>BPool</th>
<th>P Sz</th>
<th>UNIQUE</th>
<th>IDX TYPE</th>
<th>AVG KEYLEN</th>
<th>PAD</th>
<th>CPM PRES</th>
<th>IDX EXT TYP</th>
<th>STATETIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXX13X1</td>
<td>DBC4TXD1</td>
<td>Y</td>
<td>N</td>
<td>TXX13X1</td>
<td>5</td>
<td>863154</td>
<td>3</td>
<td>98.62%</td>
<td>SP2</td>
<td>4000</td>
<td>D</td>
<td>2</td>
<td>2011-05-27 19:30:31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXX13X1</td>
<td>DBC4TXD1</td>
<td>Y</td>
<td>N</td>
<td>TXX13X1</td>
<td>5</td>
<td>207153</td>
<td>3</td>
<td>98.62%</td>
<td>SP2</td>
<td>4000</td>
<td>D</td>
<td>2</td>
<td>2011-05-27 19:30:31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXX13X1</td>
<td>DBC4TXD1</td>
<td>Y</td>
<td>N</td>
<td>TXX13X1</td>
<td>5</td>
<td>162213</td>
<td>3</td>
<td>98.62%</td>
<td>SP2</td>
<td>4000</td>
<td>D</td>
<td>2</td>
<td>2011-05-27 19:30:31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXX13X1</td>
<td>DBC4TXD1</td>
<td>Y</td>
<td>N</td>
<td>TXX13X1</td>
<td>5</td>
<td>1338</td>
<td>3</td>
<td>98.62%</td>
<td>SP2</td>
<td>4000</td>
<td>D</td>
<td>2</td>
<td>2011-05-27 19:30:31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Key Columns (SYKEYS)

<table>
<thead>
<tr>
<th>COLSEQ</th>
<th>COLNAME</th>
<th>COLNO</th>
<th>ORDERING</th>
<th>COLCARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FILE_ID</td>
<td>57</td>
<td>A</td>
<td>55585</td>
</tr>
<tr>
<td>2</td>
<td>SS_ROW_NUM</td>
<td>42</td>
<td>A</td>
<td>27275</td>
</tr>
</tbody>
</table>

### Index File Dependencies (SYSPLANDEP)

### Index Page Access Dependencies (SYSPAC=CPBP)
**SQL Statement Text and Performance Recommendations**

**DIST - DDF - Package SYSLH200 - 60 Select (Stmt Count 69)**

```sql
SELECT REQUEST_DT_TM, CORR_BY_IND
FROM DBC4TXD1.TXT13G_CORRESPOND2
WHERE CORR_TYPE = ???
    AND REQUEST_DT_TM = (SELECT MAX(A.REQUEST_DT_TM)
                          FROM DBC4TXD1.TXT13G_CORRESPOND2 A, TXI13G0(2) Y
                          DBC4TXD1.TXV132_OLD_CLAIM B TXI132U(2)
                          WHERE B.FILE_ID = ???
                                   AND B.SS_ROW_NUM = ???
                                   AND A.CORR_TYPE = ???
                                   AND A.CLAIM_NUM = B.CLAIM_NUM)
WITH UR
```

Consider adding Column CLAIM_NUM to the end of Index TXI132U to change the access to Index Only. Also, consider adding an index to Table CORRESPOND2 to change the Non-Matching Index Scan to Matching Index Only as follows:

- Column 1 - CORR_TYPE
- Column 2 - REQUEST_DT_TM
- Column 3 - CORR_BY_IND

**Change the Non-Matching Index Scan of Nearly 16,000 Leaf Pages to a Matching Index Only Access – and Change the Access to Index Only of Table OLD_CLAIM in the SubSelect**
The Access is a Non-Matching Index Scan of Over 14,000 Leaf Pages and only 1% of the Rows on Table MASS_EDIT are Compressed (needs a Reorg to Compress all Rows)
SQL Statement Text and Performance Recommendations

- DDF - Package SYSLH200 - 608 Select (Stmt Count 54)

```
SELECT * FROM DBC4TXD1.TXV137_MASS_EDIT
WHERE EDIT_IND = ????
    AND UPD_DT_TM = ????
    AND UPD_ID = ????
WITH UR
```

Table TXT137_MASS_EDIT is only 1% compressed - Reorg followed by Runstats. Select only the columns needed rather than Select *. Consider adding an index to Table TXT137_MASS_EDIT to change the Non-Matching Index Scan to a Matching Index Scan as follows:

```
Column 1 - UPD_ID
2 - UPD_DT_TM
3 - EDIT_IND
```

Change the Non-Matching Index Scan of Over 14,000 Leaf Pages to a Matching Index Access – and Avoid Using SELECT * (of all Columns) and just Select the Columns Required
SQL Analysis

DB2 Explain

DB2 Catalog Statistics

A Tablespace Scan of Over 16,000 Data Pages
SQL Statement Text and Performance Recommendations

- DDF - Package SYSLH200 - 455 Select (Stmt Count 54)

```
SELECT MAX(FILE_ID) AS FILE_ID
FROM DBC4TXD1.TXT134_SS_LOG
WHERE ORIG_FILE_ID = ????
  AND SINGLE_CLAIM_IND = ????
WITH UR

608.6K Rows/16K Pg
```

Consider adding an index to Table TXT134_SS_LOG to change the Tablespace Scan to a Matching Index Only Scan as follows:

Column 1 - ORIG_FILE_ID
2 - SINGLE_CLAIM_IND
3 - FILE_ID

Change the Tablespace Scan to a Matching Index Only Access – in Fact, An I1 Fetch which is Wicked-Fast
DB2 DDF SQL Activity

<table>
<thead>
<tr>
<th>DBRM/Pack</th>
<th>Bind Timestamp</th>
<th>Statement Type</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSH201</td>
<td>04/23/2014 03:33:09 PM</td>
<td>DB2</td>
<td>56,006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSLH200</td>
<td>04/23/2014 03:33:09 PM</td>
<td>DB2</td>
<td>4,566</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSLN200</td>
<td>04/23/2014 03:33:14 PM</td>
<td>DB2</td>
<td>253</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SQL Statement Text:

```
SELECT OINS_SID_NO AS SID, OINS_TDCID_NO AS TDCJ... |
```

Run Time Statistics For Scans Executed By The SQL Statement:

```
Type Processed | Rows Examined | Rows-Stage1 | Rows-Stage2 | Rows-Inserted | Rows-Deleted | Rows-Updated | Got Page Requests | RI Page Scans | RI Page Deletes | LOB Page Scans | LOB Page Updates |
SEQD            | 3,556,620     | 3,556,620   | 3,556,620   | 2             | 0            | 0            | 0                | 0            | 0              | 0              | 0              |
SEQW            | 0             | 0            | 0            | 0             | 0            | 0            | 0                | 0            | 0              | 0              | 0              |
FETCH           | 83            | 2            | 0.000744     | 0.001393      | 0.001488     | 0            | 0                | 0            | 0              | 0              | 0              |
PREPARE         | 81            | 2            | 0.000282     | 0.000443      | 0.000524     | 0            | 0                | 0            | 0              | 0              | 0              |
```
A Tablespace Scan of Nearly 280,000 Data Pages and DB2 Compression is Not Turned On
SQL Statement Text and Performance Recommendations

DIST – DDF Package SYSLN200 – 80 Select (Stmt Count 6)

```
SELECT OIMS_SID_NO AS SID,
     OIMS_TDCJID_NO AS TDCJ,
     OIMS_NAME AS NAME,
     OIMS_DOB AS DOB,
     OIMS_SEX AS GENDER,
     OIMS_RACE AS RACE,
     OIMS_SETUP_DTE AS SETUPDATE,
     OIMS_LEG_RESI_INC AS LCOR,
     OID_SSN_NO AS SSN
FROM TAMASTER R
LEFT OUTER JOIN
     TAOID_DOCUMENTS X1CU_SAOIDDOC(1)
ON OID_SID_NO = OIMS_SID_NO
WHERE UPPER(OIMS_NAME) LIKE ?
```

Consider turning DB2 Compression on for Table TAMASTER. Using the UPPER Function on the predicate prevents DB2 from using the index that is on Column OIMS_NAME (Index). Consider removing the UPPER Function.

Change the Tablespace Scan to a Matching Index Access by Removing the UPPER Function – or – Create an Index on Expression

INCLUDING Column OID_SSN_NO with Index X1CU_SAOIDDOC will change that Access to Index Only
### DB2 DDF SQL Activity

#### Profile Name: [Redacted]

<table>
<thead>
<tr>
<th>DBRM/Package</th>
<th>Bind Timestamp</th>
<th>Statement Type</th>
<th>Statement Count</th>
<th>Elapsed Time (sec)</th>
<th>Parallel Task CPU Distribution (sec)</th>
<th>CPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SELECT</td>
<td>5</td>
<td>34.176</td>
<td>0.000104/0.010205/3.5568</td>
<td>0.000145/0.001261/4.945796</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELECT</td>
<td>1943</td>
<td>15</td>
<td>0.212622/0.951545/3.189331</td>
<td>0.271173/0.330075/4.06767</td>
</tr>
</tbody>
</table>

#### SQL Statement Text

```
SELECT EX_ID, EX_UNIT_TYPE FROM TPRM_SISP WHERE 1=1
```

#### Run Time Statistics for Rows Executed by the SQL Statement

<table>
<thead>
<tr>
<th>Type</th>
<th>Rows Processed</th>
<th>Rows Examined</th>
<th>Rows-Q Stage1</th>
<th>Rows-Q Stage2</th>
<th>Rows Inserted</th>
<th>Rows Deleted</th>
<th>Rows Updated</th>
<th>Get Page Requests</th>
<th>RI Page Requests</th>
<th>RI Page Transfers</th>
<th>LOB Page Transfers</th>
<th>LOB Page Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQD</td>
<td>184,041</td>
<td>184,041</td>
<td>172,020</td>
<td>172,020</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5,754</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SEQW</td>
<td>172,020</td>
<td>172,020</td>
<td>172,020</td>
<td>172,020</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,303</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PREPARE</td>
<td>1844</td>
<td>1844</td>
<td>0.000249</td>
<td>0.000053</td>
<td>0.001245</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000325</td>
<td>0.000074</td>
</tr>
<tr>
<td>FETCH</td>
<td>1870</td>
<td>1870</td>
<td>0.000447</td>
<td>0.001242</td>
<td>0.002233</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000315</td>
<td>0.000162</td>
</tr>
</tbody>
</table>
A Tablespace Scan of Over 5,000 Data Pages and DB2 Compression is Not Turned On
SQL Statement Text and Performance Recommendations

**DIST** – DDF Package SYSSH201 – 1843 Select (Stmt Count 15)

SELECT EM_ID,
    EM_UNIT_TYPE
FROM TPEM_SISP
    X4_TPEM_SISP(1)
WHERE (  EM_UNIT_TYPE = ?
    OR EM_UNIT_TYPE = ?)
AND EM_SID_NO = ?
    AND EM_UNIT_STATUS = ?
AND EM_ID NOT IN(SELECT EMSIS.EM_ID
    FROM TPEM_SISP EMSIS
    WHERE
        EMSIS.EM_UNIT_STATUS = ?
    OR EMSIS.EM_UNIT_STATUS = ?
    OR (  EMSIS.EM_UNIT_STATUS = ?
        AND NOT EMSIS.EM_DISCNNT_DATE = ?))
WITH UR
FOR FETCH ONLY

Consider adding Columns EM_UNIT_STATUS and EM_UNIT_TYPE to the end of Index X4_TPEM_SISP to increase the MATCHCOLS. Could this be re-written to exclude the Subselect - only a few rows qualify for the main Select but then a large number of rows are accessed in the Subselect to see if Column EM_ID has certain statuses. Are the statuses being checked in the Subselect different than the status being checked in the main Select?

Adding Columns EM_UNIT_STATUS, EM_UNIT_TYPE, and EM_ID to Index X4_TPEM_SISP Would Change the Main Access to Index Only
Bufferpool Tuning – It’s Only Memory

Consider doing this command weekly/monthly saving the output in a text file and use the Getpage Requests and I/O counts to determine the bufferpool hit ratios; increasing the bufferpool sizes and parameters to try and achieve hit ratios greater than 90%. These counts are reset after each time this Command is issued.

\[
\text{(GetPage Req} - (\text{Sync I/O} + \text{Async Pgs Rd})) / \text{GetPage Req} \times 100
\]

Async Pgs Rd = Sequential Prefetch Pg + List Prefetch Pg + Dynamic Prefetch Pg
Remember!

Business Functions don’t exist to reduce costs; they exist to generate value!

So reducing Operating Costs through Application Performance Management helps the Business generate value
Identifying Tables Without Compression

```sql
--CATALOG QUERY TO LIST TABLES WITHOUT COMPRESSION

SELECT SUBSTR(B.TSNAME,1,10) AS TSNAME,
       SUBSTR(B.CREATOR,1,10) AS CREATOR,
       SUBSTR(B.NAME,1,15) AS NAME,
       INT(B.NPAGESF) AS NPAGESF, INT(B.CARDF) AS CARDF,
       A.COMPRESS AS CMP, C.BPOOL,
       A.PCTFREE, A.FREEPAGE, C.PGSIZE,
       C.SEGSIZE, C.STATSTIME
FROM SYSIBM.SYSTABLEPART A,
     SYSIBM.SYSTABLES B,
     SYSIBM.SYSTABLESPACE C
WHERE A.COMPRESS  = ' '
     -- 'Y' WILL LIST TABLES WITH COMPRESSION
     AND B.CREATOR <> 'SYSIBM'  -- NO CATALOG TABLES
     AND A.TSNAME = B.TSNAME   -- JOIN PREDICATES
     AND B.TSNAME = C.NAME
     AND A.DBNAME = B.DBNAME
     AND B.DBNAME = C.DBNAME
     AND A.PARTITION IN (0,1)   -- DON'T COUNT PARTITIONS MORE THAN ONCE
     --AND B.PCTROWCOMP < 98    -- THOSE TABLES NOT FULLY COMPRESSED
     AND B.CARD <> -1           -- ENSURE RUNSTATS
     AND A.CARD <> -1           -- ENSURE RUNSTATS
     AND B.TYPE = 'T'           -- ONLY TABLES
     --AND A.FREEPAGE < 10       -- ADDITIONAL QUALIFIERS
     --AND A.CARD < 50000       --
     AND B.NPAGES > 50          -- NUMBER OF PAGES THRESHOLD
ORDER BY B.TSNAME, B.NPAGES DESC WITH UR;
```
<table>
<thead>
<tr>
<th>TSNAME</th>
<th>CREATOR</th>
<th>NAME</th>
<th>NPAGESF</th>
<th>CARDF</th>
<th>CMP</th>
<th>BPOOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASC</td>
<td>MFHCWW0</td>
<td>DASC</td>
<td>447</td>
<td>39261</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DBMXSTS</td>
<td>DFHMXS0</td>
<td>DM66MODLIST</td>
<td>223</td>
<td>2548</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DBMXSTS</td>
<td>DFHMXS0</td>
<td>DM66COLUMNS</td>
<td>176</td>
<td>2795</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DBMXSTS</td>
<td>DFHMXS0</td>
<td>DM66MCTRL</td>
<td>116</td>
<td>1017</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DBTJSTS</td>
<td>BUSTJS0</td>
<td>DBCOLDATA</td>
<td>5429</td>
<td>227113</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DHOBJ</td>
<td>MFHCWW0</td>
<td>DHOBJ</td>
<td>210</td>
<td>2727</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DPRP</td>
<td>MFHCWW0</td>
<td>DPRP</td>
<td>435</td>
<td>25617</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>DSUBEX</td>
<td>MFHCWW0</td>
<td>DSUBEX</td>
<td>195</td>
<td>21367</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>FDCI630</td>
<td>BFHRWA0</td>
<td>DBCOLDATA_NFM</td>
<td>16129</td>
<td>712579</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>SASC</td>
<td>MFHCWW0</td>
<td>SASC</td>
<td>577</td>
<td>25944</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>SPRP</td>
<td>MFHCWW0</td>
<td>SPRP</td>
<td>1255</td>
<td>41391</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>SUMABW1</td>
<td>FADB2</td>
<td>DBCOLDATA_N22</td>
<td>123</td>
<td>5332</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>SUMCOLN</td>
<td>FDDBA</td>
<td>DBCOLDATA</td>
<td>21446</td>
<td>949361</td>
<td>BP0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCTFREE</th>
<th>FREEPAGE</th>
<th>PGSIZE</th>
<th>SEGSIZE</th>
<th>STATSTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>64</td>
<td>2011-05-16-12.15.26.642782</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>4</td>
<td>16</td>
<td>2013-01-17-10.24.54.264222</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>4</td>
<td>16</td>
<td>2013-01-17-10.24.54.264222</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>4</td>
<td>16</td>
<td>2011-06-09-07.44.48.666261</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>64</td>
<td>2011-05-16-12.15.28.650474</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>64</td>
<td>2011-05-16-12.15.26.992491</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>64</td>
<td>2011-05-16-12.15.27.606492</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>64</td>
<td>2011-05-16-12.15.28.134038</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2013-03-20-15.16.34.202069</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2011-05-16-12.15.23.389421</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2011-05-16-12.15.25.821831</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2011-10-19-15.59.33.172783</td>
</tr>
</tbody>
</table>
Identifying Unused Non-Unique Indexes

-- CATALOG QUERY TO IDENTIFY UNUSED NON-UNIQUE INDEXES

SELECT DISTINCT(SUBSTR(A.NAME,1,16)) AS INDEXNAME,
       SUBSTR(B.TSNAME,1,10) AS TSNAME,
       SUBSTR(B.CREATOR,1,10) AS CREATOR,
       SUBSTR(B.NAME,1,12) AS TABLENAME,
       C.NLEAF, LASTUSED
FROM SYSIBM.SYSINDEXES A,
     SYSIBM.SYSTABLES B,
     SYSIBM.SYSINDEXSPACESTATS C
WHERE C.LASTUSED IS NULL  -- SET TO NULL IF NOT USED
  AND  B.CREATOR <> 'SYSIBM'  -- IGNORE CATALOG
  AND  A.TBNAME = B.NAME  -- JOIN PREDICATES
  AND  A.NAME = C.NAME
  AND  C.NLEAF > 100  -- SIZE THRESHOLD
  AND  A.UNIQUERULE = 'D'  -- NON-UNIQUE INDEXES
ORDER BY C.NLEAF DESC  -- SHOW LARGEST FIRST
WITH UR;

<table>
<thead>
<tr>
<th>INDEXNAME</th>
<th>TSNAME</th>
<th>CREATOR</th>
<th>TABLENAME</th>
<th>NLEAF</th>
<th>LASTUSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERF6INDEX_4</td>
<td>FRJGDF6</td>
<td>PFHJGD0</td>
<td>FRPERF6_EMP</td>
<td>7954</td>
<td>----------</td>
</tr>
<tr>
<td>PERF1INDEX_5</td>
<td>FRJGDF1</td>
<td>PFHJGD0</td>
<td>FRPERF1_EMP</td>
<td>7172</td>
<td>----------</td>
</tr>
<tr>
<td>INDEX_4</td>
<td>FDHA14SF</td>
<td>FDDBA2</td>
<td>FDHASH_EMP</td>
<td>5336</td>
<td>----------</td>
</tr>
<tr>
<td>PERF8INDEX_6A</td>
<td>FRNEWF8</td>
<td>FRNEW</td>
<td>FRPERF8_EMP</td>
<td>3220</td>
<td>----------</td>
</tr>
<tr>
<td>PERF8INDEX_6</td>
<td>FRPERF8</td>
<td>FRDBA</td>
<td>FRPERF8_EMP</td>
<td>3208</td>
<td>----------</td>
</tr>
<tr>
<td>FACC21CT_IX1_NFM</td>
<td>SUMCOLN</td>
<td>MFHCWW0</td>
<td>FACC21CT_NFM</td>
<td>1893</td>
<td>----------</td>
</tr>
<tr>
<td>DBCOLDATA_IX1_NF</td>
<td>FDC1630</td>
<td>BFHRWA0</td>
<td>DBCOLDATA_NF</td>
<td>1353</td>
<td>----------</td>
</tr>
<tr>
<td>DBCOLDATA_IX_NFM</td>
<td>SUMCOLN</td>
<td>PMIASH0</td>
<td>DBCOLDATA_NF</td>
<td>1125</td>
<td>----------</td>
</tr>
<tr>
<td>FD621CO_IX1_NFM</td>
<td>SUMCOLN</td>
<td>MFHCWW0</td>
<td>FD621CO_NFM</td>
<td>707</td>
<td>----------</td>
</tr>
<tr>
<td>FD621CO_IX_NFM</td>
<td>SUMCOLN</td>
<td>MFHCWW0</td>
<td>FD621CO_NFM</td>
<td>682</td>
<td>----------</td>
</tr>
<tr>
<td>CWAF_EVENT_UTC_I</td>
<td>AF123TS</td>
<td>AF123AS</td>
<td>CWAF_EVENT</td>
<td>404</td>
<td>----------</td>
</tr>
</tbody>
</table>

Null
-- Number of Pages and Tables defined to each Bufferpool
--
select bpool, sum(b.npages) as NPAGES, sum(ntables) as NTABLES
--
select substr(a.name,1,8) as TSNAME,
--  substr(a.creator,1,8) as CREATOR,
--  substr(a.dbname,1,8) as DBNAME,
--  char(b.npages) as NPAGES, bpool,
--  pgsize as PGSZ, segsize as SGSZ,
--  ntables as NTBLS, a.statstime
  from sysibm.systablespace a,
  sysibm.systables b
--where bpool = 'BP0'
  where a.creator not like 'SYS%'
    -- Skip the Catalog
    and npages > 1
    -- Size Threshold
    and a.name = b.tsname
    -- Join Predicates
    and a.dbname = b.dbname
    and a.creator = b.creator
  group by bpool with ur;
-- Group by Bufferpool
--order by a.creator, a.name with ur;

<table>
<thead>
<tr>
<th>BPOOL</th>
<th>NPAGES</th>
<th>NTABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0</td>
<td>246610</td>
<td>19138</td>
</tr>
<tr>
<td>BP1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>BP16K0</td>
<td>1460</td>
<td>20</td>
</tr>
<tr>
<td>BP32K</td>
<td>8651</td>
<td>1</td>
</tr>
</tbody>
</table>
Number of Leafs and Indexes per Bufferpool

---Number of Leafs and Indexes defined to each Bufferpool---

```sql
select bpool, sum(a.nleaf) as NLEAFS, count(name) as INDEXES
--lect substr(a.name,1,18) as IXNAME,
--    substr(a.creator,1,8) as CREATOR,
--    substr(a.tbname,1,12) as TBNAME,
--    a.nleaf as LEAFS, a.bpool,
--    reset(a.nlevels) as LVLS,
--    char(a.pgsize) as PGSZ, a.statstime
  from sysibm.sysindexes a
--where bpool = 'BP0'
  where a.creator not like 'SYS%'  -- Skip the Catalog
    and a.nleaf > 1  -- Size Threshold
  group by bpool with ur;
--order by bpool with ur;

---+-----+-----+-----+
<table>
<thead>
<tr>
<th>BPOOL</th>
<th>NLEAFS</th>
<th>INDEXES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0</td>
<td>395726</td>
<td>177</td>
</tr>
<tr>
<td>BP1</td>
<td>27</td>
<td>11</td>
</tr>
<tr>
<td>BP8K0</td>
<td>23481</td>
<td>28</td>
</tr>
<tr>
<td>BP16K0</td>
<td>1539</td>
<td>21</td>
</tr>
</tbody>
</table>
---+-----+-----+-----+
Must Go Faster!!!

LAS VEGAS MOTOR SPEEDWAY 2009

112°F
Thank You