Driving issues for BOP

Program code must do more than merely produce the correct result...

It must be scalable

It must perform under system stress

It must be economical
The members of the BOP have the following experience:

- Utilities
- Database
- SQL
- SAS, System Monitors & Measurements
- Program coding
- System Z/OS Architecture
The members of the BOP team focus on reducing mainframe CPU cost.

Most of our time is spent tuning applications code but we also help with any work that occupies an address space on the MF such as: CICS, IDMS, MQ series, Monitors and DB2.
Humana MF environment as of June, 2014.

- 2 EC12
- Z/OS version 2.1, July 2014
- 10 production LPARS
- 13 production DB2 regions v9 with one V10 CM (all v10 by August)
- 4 production DB2 retrieval regions (where ad-hoc goes to suffer)
- 14 test/QA DB2 regions all V10 CM
- Data sharing with two small DB2 production regions on separate LPARS
Main Monitors and tools we use to find savings

- TMON MVS/DB2/CICS
- CA Database Adm Suite for DB2 and Detector
- COGITO EZ-DB2
- UBS Hainer BPA4DB2
- IBM Data Studio
- IBM RMF PM
- SAS against RMF and SMF data
- TSO and EMAIL alerts
BOP team requirements for DB2

- SELECT authority against all tables in test and production for explains, unloads, data analysis.

- DB2 sandbox with DBADM, we execute the DDL to support before/after solutions along with testing new indexes, MQT, UDF, etc. We usually use production data to load into the sandbox to test solutions and avoid unknown (to us) test data.

- Access to all tools and monitors at HUMANAs disposal.

- Ability to produce “testing code” and execute it.

- SMF data access via SAS performance database

- Compuware PurePath “end to end monitor”
Approach

- Look for waste – Waste typically stands out
  - High CPU time
  - Long response time / run time usually due to wait times
  - High I/O counts
  - High SQL counts
  - High DB2 getpages counts
  - High CICS calls
  - Unusual MQ message behavior

- Create reporting to expose the largest consumers of resources from multiple angles

- Take spot measurements when you notice something big – watch your system using real-time monitors
Multiple perspectives are required
- Total resource by individual program execution
- Total resource by program in aggregate
- Peak hours, peak days – Month-end
- Use multiple descriptive statistics
  • Averages hide problems
- Utilities are not exempt
Listen...

- Your subsystems will cry foul
  - CICS abends
  - DB2 SQL errors

- Abend processing is costly

- How many abends are you happy with?

- How many SQL errors make sense?

- Recurrent batch failures that get “fixed” constantly
99% of the time, the problem is in the code (program, SQL, index, etc.)

Reporting tells you what, but not where or why

We use Compuware Strobe for our microscope into the code

Detector shows you recurrent problems and can drill down into trace data

TMON/Omegamon real-time observation and reporting

CICS Interdependency Analyzer

IBM Data Studio

Create your own tools
What’s In It For You?

- Reduced batch clock time
- Reduced transaction response time
- Improved scalability
- Consistent performance under stress
- Reduced cost of application
- Reduced abends and support calls
- Reduced failed transactions
- Improved user experience
- Improved transaction accuracy
“A GETPAGE by any other name is MONEY!”
some english fella.

- A getpage implies a trip to one or more DB2 address spaces.
- Found in buffer is ok but not free and if it is found more times than it exist in the table/indexes for one unit of work, then it is redundant.
- Batch processing on large transactional tables for non transactional reporting can sometimes be negated (Fastunload) and turn days into hours or hours into minutes of execution. Time=Work=Money.
- In general we are not involved in database/applications design. We recommend DDL changes when other measures are exhausted (Altered SQL, increased predicate use, sorting keys prior to executing sql, program arrays, etc.).
- We utilize memory wherever/whenever possible.
Main Monitors and tools we use to find savings at Humana

• TMON MVS/DB2/CICS
• CA Database Adm Suite for DB2 and Detector
• COGITO EZ-DB2
• UBS Hainer BPA4DB2
• IBM data studio
• IBM RMF PM
• SAS against RMF and SMF data
• TSO and EMAIL alerts
We start where are the savings are most valuable but are weary of the things that creep in or push this window around (usually batch)
CA Detector exception analyzer was a good starting point for us to work on the most obvious problems.

### DETECTOR Exception SQL User Summary

<table>
<thead>
<tr>
<th>Command =&gt;</th>
<th>05-29-14 10:58</th>
</tr>
</thead>
</table>

#### DB2 SSID => DB2

#### View Type => M - Activity E - Exception O - Object

#### View By => U - User P - Plan F - Prof L - Collid

| Interval Date => 05-01-14 | Interval Time => 10:00:00 | Elapsed Time => 01:00 |

#### OPID | EXCEPTIONS | SQL | TIMEPCT | CPUPCT | INDB2_TIME | INDB2_CPU |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EFR0001</td>
<td>1</td>
<td>428741</td>
<td>0.09%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>SMA9774</td>
<td>2</td>
<td>73</td>
<td>5.71%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>EBP0001</td>
<td>9</td>
<td>73</td>
<td>5.71%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>BXX0289</td>
<td>1</td>
<td>213</td>
<td>8.72%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>MB0046</td>
<td>1</td>
<td>3519</td>
<td>0.03%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>DB2011G1</td>
<td>3</td>
<td>184</td>
<td>0.24%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>PROD6E</td>
<td>17</td>
<td>682</td>
<td>3.05%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>PXX0102</td>
<td>2</td>
<td>594</td>
<td>3.05%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>CRD001</td>
<td>16</td>
<td>186</td>
<td>5.52%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>CCP2901</td>
<td>52</td>
<td>24230</td>
<td>18.77%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>PROD10</td>
<td>1</td>
<td>427</td>
<td>0.18%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
<tr>
<td>CICS</td>
<td>273</td>
<td>620890</td>
<td>50.21%</td>
<td>4.06%</td>
<td>03:25:76</td>
<td>00:01:59</td>
</tr>
</tbody>
</table>

---

Humana 3270 Mod4 Display Session - A - @WL00035

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Over the years we have lowered the thresholds

We continue to lower the Getpage trigger as we work thru and fix the exceptions generated. In general 99% of the exceptions are triggered by the getpage threshold but there are occasions where CPU work creates an exception.

Profile ID ==> COLLPROF  Profile SSID ==> DBP2

Last Update Userid ==> xyzzx
Last Update Date ==> 11-16-12
Last Update Time ==> 13:04:00

Dynamic SQL Exception Thresholds
--------------------------------
CPU Time ==> 00:00:00.500
Getpage Reqs ==> 00040000
Fetch Count ==> 00300000

Static SQL Exception Thresholds
--------------------------------
CPU Time ==> 00:00:10.000
Getpage Reqs ==> 00040000
Fetch Count ==> 00300000

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
The following example is used to highlight some of the tools we use to explore MSU consumption stemming from a DB2 region.

In this case there are alerts/reports to the DBA staff that show a growth in table data that may need attention but this one slipped by and wasn’t caught.

Basically a COBOL stored procedure is performing SQL against a table that grew from a few rows to a few thousand and the original access path proved poor (Tablespace scan).
When looking at CA Subsystem Analyzer the table TDOLTSS is historically not on the top of this list.

16.0  >  -------- SS Analyzer Table Activity Display -------- 05-29-14 14:06
Command ==>                                                  Scroll ==> CSR
            LINE 1 OF 621

DB2 SSID ==> DBP5
View Type ==> O * -Object,  V -Volume,  B -BP,  A -Extends,  S -Subsys,  Q -SQL
View By   ==> T  D -Dbase,  S -Space,  * -Table                Filter ==> N Yes,No,Set
View Optn ==> A * -Activity,  B -Access,  C -Access Ratios

Interval Date => 05-28-14    Interval Time => 13:00:00    Elaps Time => 01-00

S -Index Activity,  A -SQL Activity,  D -Table Detail,  I -Index Analysis

<table>
<thead>
<tr>
<th>TABLENAME</th>
<th>DBNAME</th>
<th>TSNAME</th>
<th>PCTGP</th>
<th>GETPAGE</th>
<th>TB_GETPAGE</th>
<th>IS_GETPAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ SYSLGRNX</td>
<td>DSNDB01</td>
<td>SYSLGRNX</td>
<td>41.6%</td>
<td>847035642</td>
<td>842592956</td>
<td>4442686</td>
</tr>
<tr>
<td>_ TDOLTSS</td>
<td>DDOMBR</td>
<td>SDOLTSS</td>
<td>12.5%</td>
<td>253732703</td>
<td>253726656</td>
<td>6047</td>
</tr>
<tr>
<td>_ TDOMEMH</td>
<td>DDOMBR</td>
<td>SDOMEMH</td>
<td>11.4%</td>
<td>231442525</td>
<td>40236141</td>
<td>191206384</td>
</tr>
<tr>
<td>_ TDOCSTC</td>
<td>DDOCSTC</td>
<td>SDOCSTC</td>
<td>9.4%</td>
<td>190880825</td>
<td>64380595</td>
<td>126500230</td>
</tr>
<tr>
<td>_ TDOPERS</td>
<td>DDOMBR</td>
<td>SDOPERS</td>
<td>9.0%</td>
<td>183822468</td>
<td>54973147</td>
<td>128849321</td>
</tr>
<tr>
<td>_ TDOKCCOV</td>
<td>DDOKYCOV</td>
<td>SDOKCCOV</td>
<td>7.9%</td>
<td>161750888</td>
<td>10426646</td>
<td>151324242</td>
</tr>
</tbody>
</table>
Further exploring shows the sql and program

16.0  ----------------- DETECTOR SQL Call Text Display ----------------- 05-29-14 14:21
Command ==>                                                  Scroll ==> CSR

DB2 SSID ==> DBP5         Planname ==>                    Program ==> MES047
Type    ==> PACK          Collid    ==> MEHUMSPROC
Version ==> 2013-06-22-02.17.02.219408

-----------------------------------------------------------------------------------
E  -Explain SQL, T -Explain Text, I -ISQL ==> _

DECLARE INPUT_CSR CURSOR FOR
   SELECT LTSS.PERS_GEN_KEY , LTSS.STATE_CODE , LTSS.PROGRAM_CODE ,
          LTSS.PROD_PLAN_ID , LTSS.LTSS_EFF_DATE , LTSS.LTSS_END_DATE ,
          LTSS.SRC_PLATFORM_CD , LTSS.SRC_APPL_CD
   FROM A_DOLTSS LTSS INNER JOIN A_ME047I TEMP ON LTSS.PERS_GEN_KEY =
           TEMP.MBR_PERS_GEN_KEY WITH UR
-----------------------------------------------------------------------------------
Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
In house GUI tool using SMF, Detector and SA data
Besides SMF data we feed the tool with CA Detector and Subsystem Analyzer data.
Getpage counts for the program are sourced from Detector.
Getpage counts for the Table are sourced from Subsystem Analyzer.
Table usage comparison data pulled from SA. One day compare for 4/16 vs 5/14 is 162+ million getpages. Notice the lack of indexed getpages.
Table growth in GUI built from DB2 catalog Real time statistics

Growth of HUM.TDOLTSS

2014-03-22
2014-05-10

Date

0
500
1000
1500
2000

CARDF

HUM.TDOLTSS
EZ-DB2 for an hour shows the SQL as well
Before Rebind, 1 hour in Detector Averages

16.0  < > ------------ DETECTOR Package/DBRM SQL Display ------------ 06-04-14 11:27
Command ==> Scroll ==> CSR
LINE 1 OF 6

DB2 SSID ==> DBP5          Type ==> PROC          Program ==> MES047
Collid  ==> MEHUMSPROC
Version  ==> 2013-06-22-02.17.02.219408
Total/Avg  ==> A

Interval Date  ==> 06-03-14  Interval Time  ==> 10:00:00  Elapsed Time  ==> 01:00

Q  -View SQL text, T  -View Tables/Indexes, E  -Explain, D  -View Detail

<table>
<thead>
<tr>
<th>SQL_CALL</th>
<th>INDB2_CPU</th>
<th>GETPAGE</th>
<th>GETPFAIL</th>
<th>SYNCREAD</th>
<th>SPFETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ FETCH</td>
<td>00:00.000824</td>
<td>42.97</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>_ OPEN</td>
<td>00:00.000078</td>
<td>8.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>_ INSERT</td>
<td>00:00.000054</td>
<td>2.22</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>_ OPEN</td>
<td>00:00.000008</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>_ CLOSE</td>
<td>00:00.000005</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>_ INSERT</td>
<td>00:00.000038</td>
<td>2.98</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

******************************************************************************

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
After Rebind 1 hour in detector, Approx 30 MSU reduction for this hour at 10am

16.0  < > -------- DETECTOR Package/DBRM SQL Display -------- 06-04-14 11:25
Command ==> Scroll ==> CSR

DB2 SSID ==> DBP5            Type ==> PROC                  Program ==> MES047
Collid   ==> MEHUMSPROC
Version  ==> 2013-06-22-02.17.02.219408
Total/Avg => A

Interval Date => 06-04-14    Interval Time => 10:00:00    Elapsed Time => 01:00

Q -View SQL text, T -View Tables/Indexes, E -Explain, D -View Detail

SQL_CALL        INDB2_CPU   GETPAGE       GETPFAIL    SYNCREAD    SPFETCH
              -------        -----------     -----------     -----------     -----------
_ INSERT      00:00.000053  2.25         0.00        0.00        0.00
_ FETCH       00:00.000032  4.61         0.00        0.00        0.00
_ OPEN        00:00.000010  0.00        0.00        0.00        0.00
_ OPEN        00:00.000006  0.00        0.00        0.00        0.00
_ CLOSE       00:00.000003  0.00        0.00        0.00        0.00
_ INSERT      00:00.000036  3.00        0.00        0.00        0.00

******************************************************************************

1 hour

INDB2_CPU    GETPAGE
    -------        -----------
BEFORE  09:32.699365  31285653
AFTER   01:04.778710  4078480
View of getpage against the table after Bind

Getpage Consumption for Table TDOLTSS (DB DDOMBR)
View of Index activity since bind sourced SA

Getpage Consumption for Index: XDOLTSS0 (DB DDOMBR)
Buffer activity by index of interest sourced SA
Perform the SQL only one time per key set

- When possible a sort step can be introduced to order the input file according to the predicates. I don’t remember the cost of the sort ever negating savings in DB2 work.

- When the predicate set returned from the sort is distinct then this sort will only provide clustering order relief to the program.

- When the predicates are repeated (most cases) then working storage can be interrogated to prevent the same sql execution more than once.
Perform the sql one time only - BATCH

• When you see many more of the same sql calls against a table or join than there are results then preform the sql one time and place the results into a program array.

• In some cases this may be more memory than available so in some cases we have used VSAM files to place the data and remove the DB2 work saving CPU and I/O wait time.
SQL and Index change
Cobol Program

SELECT A.TYPE_CD_ID, A.TYPE_CD, A.SERV_RATE_AMT, A.SERV_CNT, B.SERV_RATE_AMT, B.SERV_CNT
FROM TFNPPPX A
INNER JOIN TFNPPPY B
ON A.TYPE_CD_ID = B.TYPE_CD_ID
AND A.TYPE_CD = B.TYPE_CD
WHERE (A.SERV_CCYYMM_DT  = :WS-CCYYMM1
AND B.SERV_CCYYMM_DT  = :WS-CCYYMM2)
AND ROUND((A.SERV_RATE_AMT * :WS-PCT),2)
BETWEEN B.SERV_RATE_AMT - .50
AND B.SERV_RATE_AMT + .50;

CREATE INDEX HUM.XFNPPPY1
ON HUM.TFNPPPY
(SERV_CCYYMM_DT ASC
,TYPE_CD ASC
,TYPE_CD_ID ASC)

SELECT A.TYPE_CD_ID, A.TYPE_CD, A.SERV_RATE_AMT, A.SERV_CNT, B.SERV_RATE_AMT, B.SERV_CNT
FROM TFNPPPX A
INNER JOIN TFNPPPY B
ON A.TYPE_CD_ID = B.TYPE_CD_ID
AND A.TYPE_CD = B.TYPE_CD
WHERE (A.SERV_CCYYMM_DT  = :WS-CCYYMM1
AND B.SERV_CCYYMM_DT  = :WS-CCYYMM2)
AND B.SERV_RATE_AMT <= ROUND((A.SERV_RATE_AMT * :H),2) + .50
AND B.SERV_RATE_AMT >= ROUND((A.SERV_RATE_AMT * :H),2) - .50

CREATE INDEX HUM.XFNPPPY1
ON HUM.TFNPPPY
(SERV_CCYYMM_DT ASC
,TYPE_CD ASC
,TYPE_CD_ID ASC
,SERV_RATE_AMT ASC)
Index change and SQL change example for BATCH

RV045 MSU and Run Time

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
GETPAGE = MONEY

RV045 Getpage Consumption

Month (Monthly Run)

Getpages

June 4,276,479,299
May 6,470,052,819
August 527,346,253

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Creating a flat Master file with FASTUNLOAD and Match/merge for multiple batch jobs as input

MSU Consumption by ME Jobs

Tuning effects

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Approximately 160 MSU saved per day

ME Batch: Avg MSU/Day Consumption

Year Month

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>2012 - Avg MSU/Day</th>
<th>2013 - Avg MSU/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>Feb</td>
<td>1,017.57</td>
<td>834.57</td>
</tr>
<tr>
<td></td>
<td>Mar</td>
<td>917.86</td>
<td>745.69</td>
</tr>
<tr>
<td></td>
<td>Apr</td>
<td>999.19</td>
<td>741.29</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>890.89</td>
<td>698.68</td>
</tr>
<tr>
<td></td>
<td>Jun</td>
<td>899.86</td>
<td>899.19</td>
</tr>
<tr>
<td></td>
<td>Jul</td>
<td>890.89</td>
<td>890.89</td>
</tr>
<tr>
<td></td>
<td>Aug</td>
<td>895.26</td>
<td>891.34</td>
</tr>
<tr>
<td></td>
<td>Sep</td>
<td>914.70</td>
<td>895.26</td>
</tr>
<tr>
<td></td>
<td>Oct</td>
<td>912.92</td>
<td>914.70</td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>920.22</td>
<td>912.92</td>
</tr>
<tr>
<td></td>
<td>Dec</td>
<td>957.74</td>
<td>920.22</td>
</tr>
</tbody>
</table>

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Best case, don’t run the same work twice!
SQL join broken into desperate fastunloads and match merged

<table>
<thead>
<tr>
<th>Background</th>
<th>Goal</th>
<th>Approach</th>
<th>Where do we Look?</th>
<th>What to do?</th>
<th>Results</th>
<th>Contact</th>
</tr>
</thead>
</table>

AE0795* MSU Consumption

![Graph showing MSU consumption over time with a specific date range highlighted.](image)
Tie Detector results to Dynatrace Pure path example

Select DISTINCT RTRIM(GRP.GROUP_NBR) GroupNumber,
RTRIM(GRP.PLATFORM_CD) Platform,
RTRIM (GRP.BLK_BUS_CD) BlockOfBusiness,
RTRIM (GRP.GROUP_NM) GroupName,
GRP.EFF_DT EffectiveDate,
GRP.TERM_DT TermDate
From Hum.TOUGRP GRP
INNER JOIN Hum.TMAPIDC PIDC
  ON PIDC.GROUP_NBR = GRP.GROUP_NBR
INNER JOIN Hum.TSOPLAT PLAT
  ON PLAT.PLATFORM_CD = GRP.PLATFORM_CD
INNER JOIN Hum.TSOBLK BLK
  ON BLK.BLK_BUS_CD = GRP.BLK_BUS_CD
WHERE PIDC.MBR_UMID_NBR = ?
AND GRP.PLATFORM_CD = ?
AND (GRP.BLK_BUS_CD = ? OR GRP.BLK_BUS_CD = ?)
ORDER BY GroupNumber ASC FOR FETCH ONLY
Searching by database in the Dynatrace tool reveals the same SQL with over 6 second response time
Resolution in this case was request for a new index.
6 Seconds to less than 0.04 response time and ~270K to
Less than 4K Getpages per execution
Clipping Coupons

- Newer database releases
- Page fixing buffers
Immediate relief with DB2 Version 10

DBP8 Started Task TCB,SRB,IO MSU

DB2 Version 10
Conv. Mode

TCMSU
SRMSU
OMSU

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
We page fixed two of the busiest buffer pools in one of our busiest DB2 region. These two buffers are for large randomly accessed tablespaces and large randomly accessed indexes. These two BP represent 29% 4K pages allocated to the DB2 region and about 20% getpage activity in the pools.

The savings are for the DBM1 address space only for asynchronous (i.e., prefetch) read I/O and database writes and do not show savings in the allied address spaces like DDF (DIST), CICS or batch programs that perform SQL and incur synchronous read I/O.

Before page fixing the buffers we check the LPAR memory history and looked at the number page-ins required for the buffers over a period of weeks and determined we would not be.
Check the LPAR page rates, this one day example is typical before and after we page fixed. No problems.
Check the high private memory, most notably Zone 1 workload (onlines) 8am-5pm.
Check the System memory, most notably Zone 1 workload (onlines) 8am-5pm.

System Memory Usage
Wednesday, February 12, 2014
System Name=SYSD

Source: SMF/MCS DETAIL2,SCFPAGnn Data

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Check the bufferpool to see how many times BP pages were paged out/in by MVS since startup.

```
-DISPLAY BUFFERPOOL (BP4) DETAIL

********************************** TOP OF DATA **********************************
DSNB401I -DBP5 BUFFERPOOL NAME BP4, BUFFERPOOL ID 4, USE COUNT 77
DSNB402I -DBP5 BUFFER POOL SIZE = 180000 BUFFERS  AUTOSIZE = NO
   ALLOCATED = 180000  TO BE DELETED = 0
   IN-USE/UPDATED = 108  BUFFERS ACTIVE = 180000
DSNB406I -DBP5 PGFIX ATTRIBUTE -
   CURRENT = NO
   PENDING = NO
   PAGE STEALING METHOD = LRU
DSNB404I -DBP5 THRESHOLDS -
   VP SEQUENTIAL = 30
   DEFERRED WRITE = 5  VERTICAL DEFERRED WRT = 5, 0
   PARALLEL SEQUENTIAL =50  ASSISTING PARALLEL SEQT= 0
DSNB409I -DBP5 INCREMENTAL STATISTICS SINCE 07:01:07 FEB 12, 2014
DSNB411I -DBP5 RANDOM GETPAGE = 29356257 SYNC READ I/O (R) = 4924899
   SEQ. GETPAGE = 18043535 SYNC READ I/O (S) = 7685
   DMTH HIT = 0  PAGE-INS REQUIRED = 0
```
Savings in DBM1 with I/o as work performed

DBP5DBM1 I/O MSU

Page fixed two buffer pools

Axis Title

Background | Goal | Approach | Where do we Look? | What to do? | Results | Contact
Rename your SYSDUMMY table to SYSPROBABLYDONTNEEDTOUSETHISSTABLEDUMMY

An example showing use of DSNDB06 SYSDUMMY1

<table>
<thead>
<tr>
<th>PurePath</th>
<th>Response Time [ms]</th>
<th>Breakdown</th>
<th>Size</th>
<th>Agent</th>
<th>Application</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>307.24</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>128.33</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>84.47</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>57.73</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>57.08</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>54.25</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>53.68</td>
<td>cpu (96%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>53.03</td>
<td>cpu (69%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>52.15</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>46.49</td>
<td>io (92%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>45.96</td>
<td>cpu (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>41.99</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>40.21</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>39.35</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>38.66</td>
<td>io (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
<tr>
<td>/gcpws/Inquiries.asmx</td>
<td>37.91</td>
<td>cpu (100%)</td>
<td>4</td>
<td>CCPGCP_Web[Local...</td>
<td>CCP/GCP</td>
<td>2014-04-02 13:</td>
</tr>
</tbody>
</table>

**PurePath Tree (showing only relevant nodes)**

```
<table>
<thead>
<tr>
<th>Method</th>
<th>Argument</th>
<th>Exec Total [ms]</th>
<th>Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>isFirstTimeCaller(long, String)</td>
<td></td>
<td>306.61</td>
<td></td>
</tr>
<tr>
<td>Database Summary</td>
<td></td>
<td>215.72</td>
<td></td>
</tr>
<tr>
<td>1 x SELECT 1 FROM SYSIBM.SYSDUMMY1 WHERE EXISTS</td>
<td>(SELECT CINQ_CONTACT_ID FROM)</td>
<td>210.28</td>
<td></td>
</tr>
<tr>
<td>1 x SELECT 1 FROM SYSIBM.SYSDUMMY1 WHERE EXISTS</td>
<td>(SELECT INQ_CONTACT_ID FROM)</td>
<td>5.43</td>
<td></td>
</tr>
</tbody>
</table>
```
Subsystem Analyzer  a day activity against SYSDUMMY1  * there is relief for bound singleton selects against sysdummy1 in V10.

16.0    >    -------- DETECTOR Table SQL Activity Display        -------- 06-04-14 10:58
COMMAND ==> Scroll ==> CSR

LINE 1 OF 42

DB2 SSID ==> DBP2    Database ==> DSNDDB06    Tablespace ==> SYSEBCDC
Table    ==> SYSDUMMY1    Tablecreator ==> SYSIBM

View Type ==> A  * -Table SQL, B -Table SQL/Multi-Table, C -Table SQL/Workfile
Total/Avg ==> T

Interval Date => 06-03-14  Interval Time => 09:00:00  Elapsed Time => 01:01

S  -View Indexes, A -View All Tables, P -View Plans, Q -View SQL Text E -Explain

<table>
<thead>
<tr>
<th>SQLCALL</th>
<th>PLCNT</th>
<th>PROGRAM</th>
<th>STMT#</th>
<th>SECT#</th>
<th>PCTTGP</th>
<th>SQL</th>
<th>TB_GETPAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>1</td>
<td>EK012</td>
<td>0001450</td>
<td>00005</td>
<td>94.0%</td>
<td>5961742</td>
<td>11923484</td>
</tr>
<tr>
<td>SELECT</td>
<td>4</td>
<td>EK61EK61</td>
<td>0000263</td>
<td>00002</td>
<td>2.7%</td>
<td>169872</td>
<td>339744</td>
</tr>
<tr>
<td>FETCH</td>
<td>1</td>
<td>SYSSH200</td>
<td>0000004</td>
<td>00004</td>
<td>2.3%</td>
<td>29696440</td>
<td>297736</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RVSTMAN</td>
<td>0000731</td>
<td>00006</td>
<td>.4%</td>
<td>27954</td>
<td>55908</td>
</tr>
<tr>
<td>FETCH</td>
<td>1</td>
<td>SYSTAT</td>
<td>0000004</td>
<td>00004</td>
<td>.2%</td>
<td>2898170</td>
<td>31186</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>IQ908</td>
<td>0003398</td>
<td>00017</td>
<td>.2%</td>
<td>12531</td>
<td>25062</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RV082</td>
<td>0003051</td>
<td>00002</td>
<td>.0%</td>
<td>1260</td>
<td>2520</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RV082</td>
<td>0003728</td>
<td>00005</td>
<td>.0%</td>
<td>454</td>
<td>908</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RV082</td>
<td>0003734</td>
<td>00006</td>
<td>.0%</td>
<td>454</td>
<td>908</td>
</tr>
<tr>
<td>FETCH</td>
<td>1</td>
<td>DQPS024</td>
<td>0002474</td>
<td>00001</td>
<td>.0%</td>
<td>3084</td>
<td>286</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>IOK2IOK2</td>
<td>0001513</td>
<td>00004</td>
<td>.0%</td>
<td>83</td>
<td>166</td>
</tr>
<tr>
<td>FETCH RSN</td>
<td>1</td>
<td>UTAFUSQ9</td>
<td>0000750</td>
<td>00002</td>
<td>.0%</td>
<td>1523358</td>
<td>80</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RV084B</td>
<td>0005581</td>
<td>00006</td>
<td>.0%</td>
<td>19</td>
<td>38</td>
</tr>
<tr>
<td>SELECT</td>
<td>1</td>
<td>RV099B</td>
<td>0002938</td>
<td>00003</td>
<td>.0%</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>FETCH</td>
<td>1</td>
<td>DSNESM68</td>
<td>0000391</td>
<td>00001</td>
<td>.0%</td>
<td>2404</td>
<td>18</td>
</tr>
</tbody>
</table>
• Use an assembler program for date, time, timestamp where possible and use the DB2 SET if that cannot be done.

• Use FETCH FIRST ROW ONLY for existence check vs SYSDUMMY

SELECT 1 FROM SYSIBM.SYSDUMMY1
WHERE EXISTS (SELECT CONTACT_ID
FROM HUM.TIQCINQ
WHERE INQA_GEN_KEY=?
AND STATUS_CD = 'CL'
AND INQUIRY_TYPE='GI'
AND DAYS(CREATED_TS)>=Days(CURRENT_DATE) - ? )
FOR FETCH ONLY WITH UR

SELECT 1
FROM HUM.TIQCINQ
WHERE INQA_GEN_KEY=?
AND STATUS_CD = 'CL'
AND INQUIRY_TYPE='GI'
AND DAYS(CREATED_TS)>=Days(CURRENT_DATE) - ?
FETCH FIRST ROW ONLY WITH UR
SYSIBM.SYSDUMMY1
Batch rewrite SQL BOP Ticket example and results
Show the problem and a unit tested possible solution
In this case changing the sql to use a join versus an existence check and forcing the sql to use a different index.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Approach</th>
<th>Where do we Look?</th>
<th>What to do?</th>
<th>Results</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Final SQL in 8700-LOOK-FOR-A-TRUMP**

```
<table>
<thead>
<tr>
<th>PROVIDER_ID</th>
<th>FK_PROV_ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>CATEGORY</th>
<th>BILLING_OFFICE_FG</th>
<th>PROV_TYPE</th>
<th>PROV_TYPE_IND</th>
<th>PREFERRED_PHONE_ID</th>
<th>END_DATE</th>
<th>EFFECTIVE_DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>U</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>iWS-ID-2-LOOK-FOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**New SQL for 8700-LOOK-FOR-A-TRUMP**

```
FROM EDS.SECONDARY_PROV_ID A , EDS.PROVIDER B , EDS.SECONDARY_PROV_ID C 
WHERE ((A.PROVIDER_ID = iWS-ID-2-LOOK-FOR) 
   OR A.PROVIDER_ID = iWS-EDI-NAME-FIRST-LAST-ALL) 
   OR A.PROVIDER_ID = iWS-EDI-NAME-LAST-FIRST-ALL 
   OR A.PROVIDER_ID = iWS-MEMM-NAME-FIRST-LAST-ALL 
   OR A.PROVIDER_ID = iWS-MEMM-NAME-LAST-FIRST-ALL 
   OR A.PROVIDER_ID = iWS-EDIM-NAME-FIRST-LAST-ALL 
   OR A.PROVIDER_ID = iWS-EDIM-NAME-LAST-FIRST-ALL 
   OR A.PROVIDER_ID = iWS-MEMMNAME-FIRST-LAST-ALL 
   OR A.PROVIDER_ID = iWS-MEMMNAME-LAST-FIRST-ALL 
   OR A.PROVIDER_ID = iWS-MEMMNAME-LAST-FIRST-ALL) 
   OR (A.PROVIDER_ID = iWS-EDI-NAME-LAST-FIRST-ALL) 
   OR (A.PROVIDER_ID = iWS-EDI-NAME-FIRST-LAST-ALL) 
   OR (A.PROVIDER_ID = iWS-MEMM-NAME-LAST-FIRST-ALL) 
   OR (A.PROVIDER_ID = iWS-MEMM-NAME-FIRST-LAST-ALL) 
   OR (A.PROVIDER_ID = iWS-EDIM-NAME-LAST-FIRST-ALL) 
   OR (A.PROVIDER_ID = iWS-EDIM-NAME-FIRST-LAST-ALL) 
   OR (A.PROVIDER_ID = iWS-MEMMNAME-LAST-FIRST-ALL) 
   OR (A.PROVIDER_ID = iWS-MEMMNAME-FIRST-LAST-ALL))
AND (A.PROV_TYPE_IND = iWS-OLD-LIC OR 
   A.PROV_TYPE_IND LIKE iWS-TYPE-2-LOOK-FOR )
```
Remove the separate existence select and incorporate it into the join.
Savings from this ticket, show the world and let applications take the credit if it means they will be more willing in the future to make changes.

### EF400 Impact

<table>
<thead>
<tr>
<th>Getpage Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thursday</td>
</tr>
<tr>
<td>905,968,844</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
<tr>
<td>817,934,684</td>
</tr>
</tbody>
</table>

**Legend:**
- Week 1 (5/15 - 5/21)
- Week 2 (5/22 - 5/28)
- Week 3 (5/29 - 6/4)
- Week 4 (6/5 - 6/11) - After Change
GETPAGE = MONEY, and I am thru
Contact Information

Mark Youngs
Technical Architect

Louisville KY
myoungs@humana.com