

Humana Application Performance

B O P

Batch Online Performance

Mark Youngs – Technical Architect

Driving issues for BOP

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

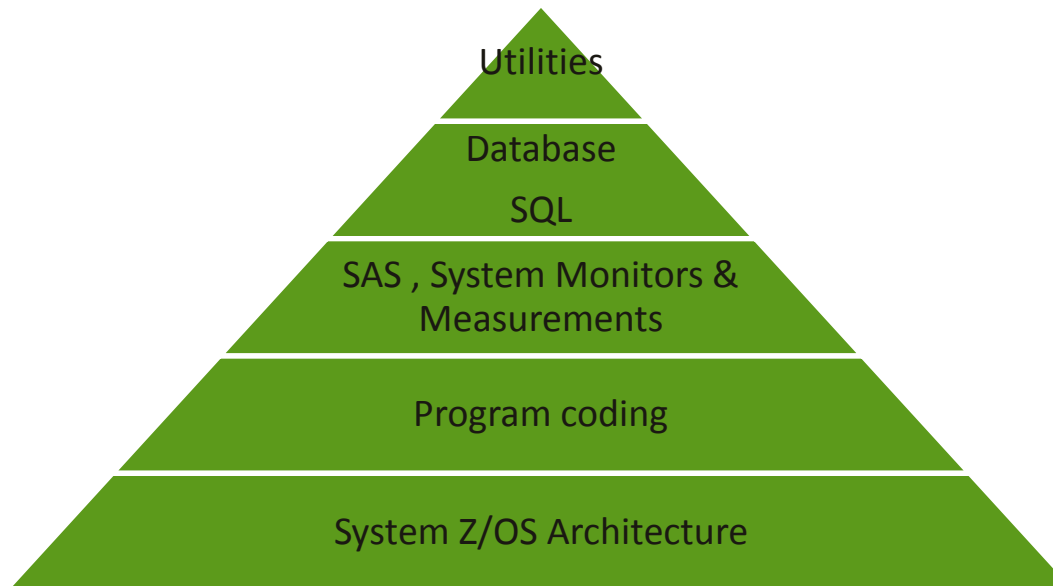
It must be scalable

It must perform under system stress

It must be economical

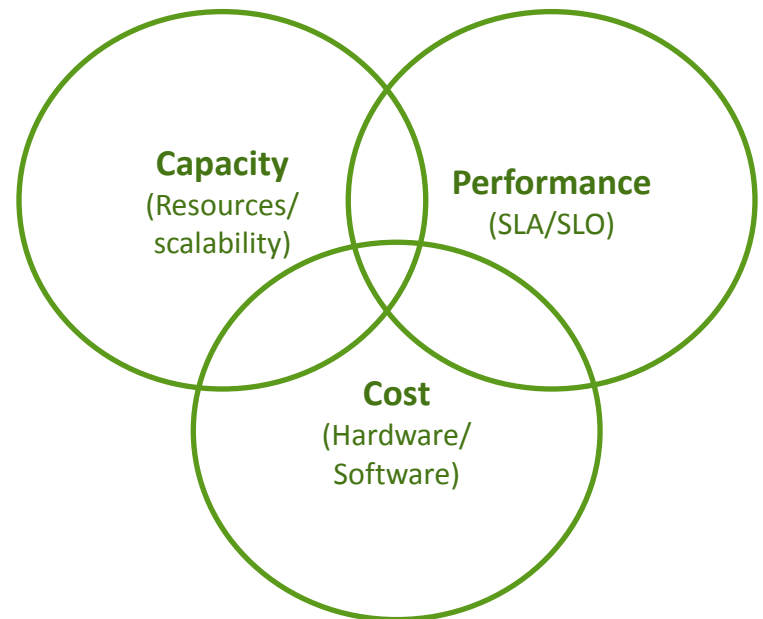
BOP Team resume

The members of the BOP have the following experience



BOP Team Scope

- ✓ 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, ect. 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 HUMANA's 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

Perspective

- 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

Tools

- 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.

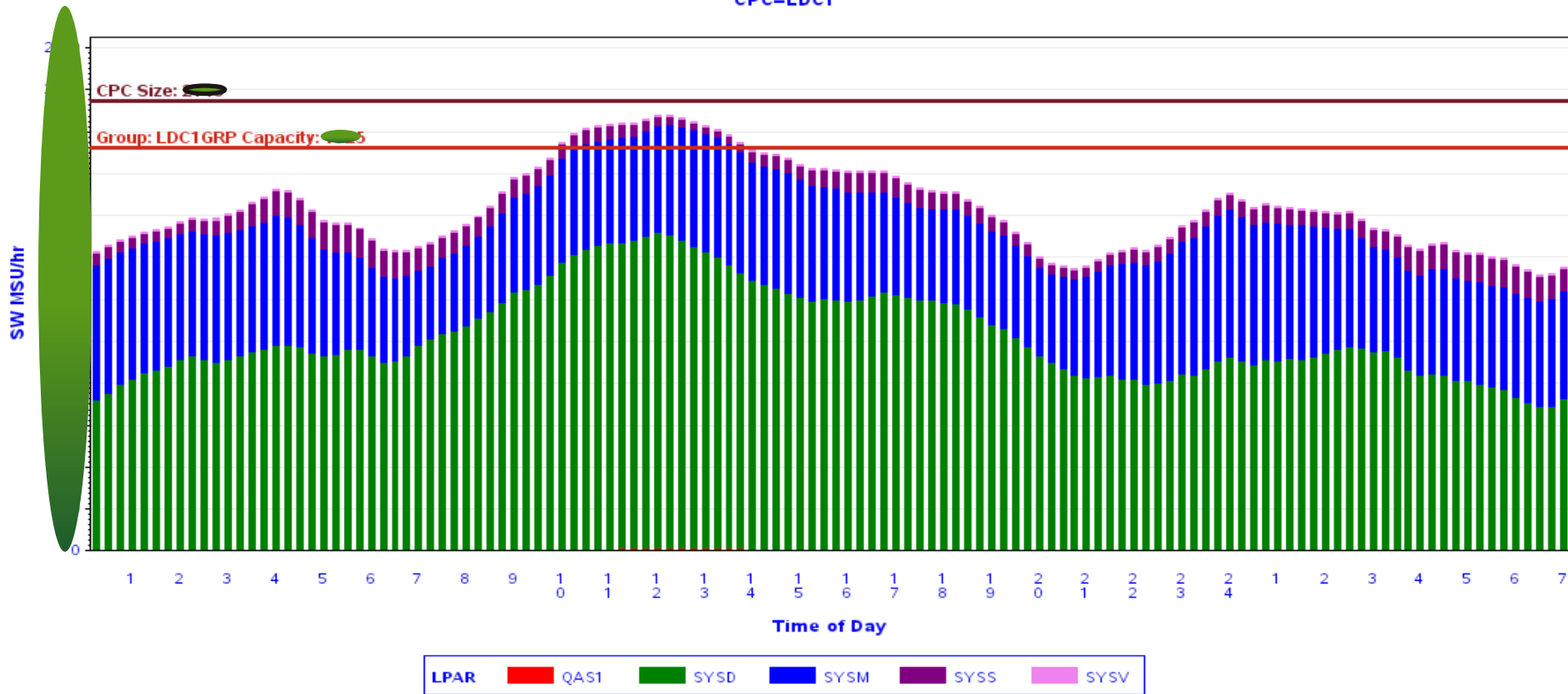
- A getpage implies a trip to one or more DB2 address spaces
- Found in buffer is ok but not free and if it is found more times than it exist in the table/indexes for one unit of work, then it is redundant.
- Batch processing on large transactional tables for non transactional reporting can sometimes 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 the savings are most valuable but are weary of the things that creep in or push this window around (usually batch)

4 Hour Rolling Average vs. Group Capacity
 Thursday, May 1, 2014
 CPC=LDC1



Source: SMF/MCS DETAIL2.HARCPUnn data

CA Detector exception analyzer was a good starting point for us to work on the most obvious problems

```
Humana 3270 Mod4 Display Session - A - @WL00035
File Edit View Communication Actions Help

16.0 > ----- DETECTOR Exception SQL User Summary ----- 05-29-14 10:58
Command ==> Scroll ==> CSR
DB2 SSID ==> DBP2 LINE 1 OF 14
View Type ==> X A -Activity * -Exception E -Error O -Object Total/Avg ==> T
View By ==> U * -User G -Prog P -Plan F -Prof C -Corrid L -Collid

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

S -View Exception Reqs, D -View Detail

OPID EXCEPTIONS SQL TIMEPCT CPUPCT INDB2_TIME INDB2_CPU
-----
-- EFR0001 1 3 .23% .84% 00:08.617548 00:04.154646
-- SMA9774 1 428741 .18% .54% 00:06.744837 00:02.668090
-- FLD0001 2 6 .09% .21% 00:03.470323 00:01.051173
-- EBP0001 9 73 5.71% 4.86% 03:25.767695 00:23.987516
-- BXJ0289 1 213 8.72% 1.03% 05:13.781639 00:05.101725
-- HXB0246 1 3519 .03% .23% 00:01.432640 00:01.168510
-- DB2GC1P 3 184 .24% .48% 00:08.966073 00:02.404392
-- PRODAE 17 868 5.80% 4.18% 03:28.951003 00:20.646243
-- P XK0102 2 284 3.08% 4.85% 01:51.130176 00:23.948726
-- CRD0001 16 108 5.52% 3.75% 03:18.830142 00:18.535509
-- CCP2001 52 24230 18.77% 23.88% 11:15.602507 01:57.776961
-- PRODIO 1 427 .18% .12% 00:06.484752 00:00.620649
-- PRODIO 1 7 1.15% .22% 00:41.422374 00:01.132248
-- CICS 273 620890 50.21% 54.73% 30:07.014796 04:29.862283
***** BOTTOM OF DATA *****

MA + a 03/008
lousna.humana.com:8998
```


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
```

When GOOD things go BAD

- 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
```

TABLENAME	DBNAME	TSNAME	PCTGP	GETPAGE	TB_GETPAGE	IS_GETPAGE
SYSLGRNX	DSNDB01	SYSLGRNX	41.6%	847035642	842592956	4442686
TDOLTSS	DDOMBR	SDOLTSS	12.5%	253732703	253726656	6047
TDOMEMH	DDOMBR	SDOMEMH	11.4%	231442525	40236141	191206384
TDOCSTC	DDOCSTC	SDOCSTC	9.4%	190880825	64380595	126500230
TDOPERS	DDOMBR	SDOPERS	9.0%	183822468	54973147	128849321
TDOKCCOV	DDOKYCOV	SDOKCCOV	7.9%	161750888	10426646	151324242

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
***** BOTTOM OF DATA *****
```

In house GUI tool using SMF, Detector and SA data

http://loupcmwps02/BOPWeb/Home.aspx

BOP Reporting ... x

File Edit View Favorites Tools Help

BOP Reporting Tool

Batch

- Program Reports
- Top 10

CICS

- Program Reports
- Abends
- Top 10
- CICS-MQ

DB2

- High Level
- SQL Errors
- ID Reports
- Table usage
- DB2 Memory
- Table usage Comarision

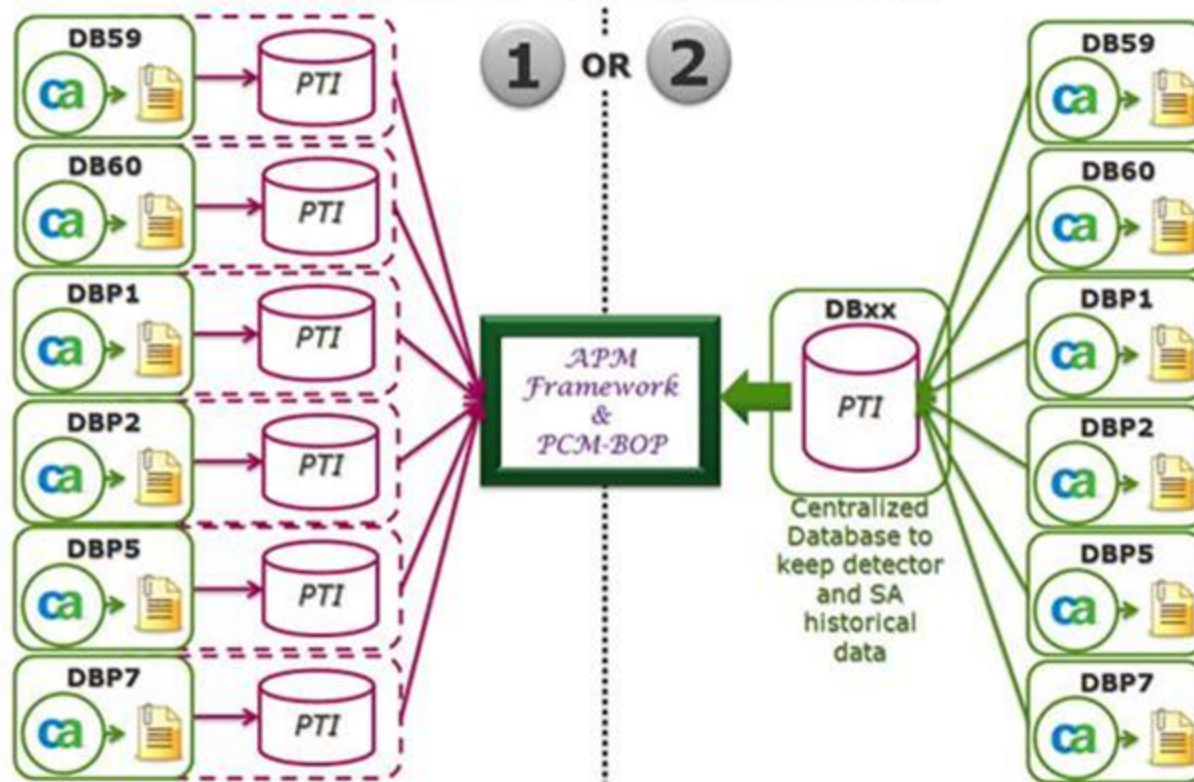
Other

- Mix & Match
- SysCode

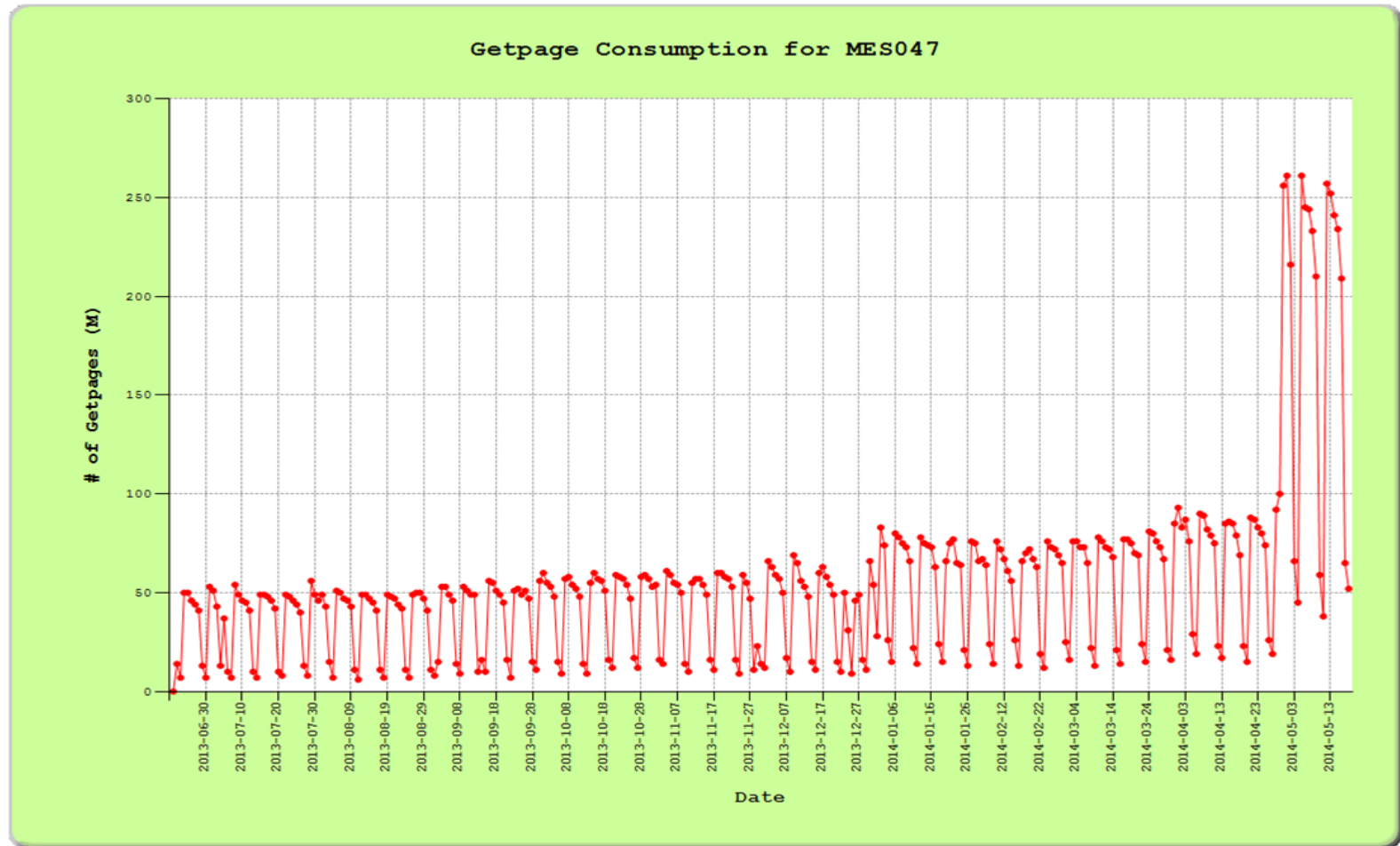
Background | Goal | Approach | [Where do we Look?](#) | What to do? | Results | Contact

Besides SMF data we feed the tool with CA Detector and Subsystem Analyzer data.

Detector and Subsystem analyzer data collection externalization



Getpage counts for the program are sourced from Detector.



Getpage counts for the Table are sourced from Subsystem Analyzer.

Getpage Consumption for Table TDOLTSS (DB DDOMBR)

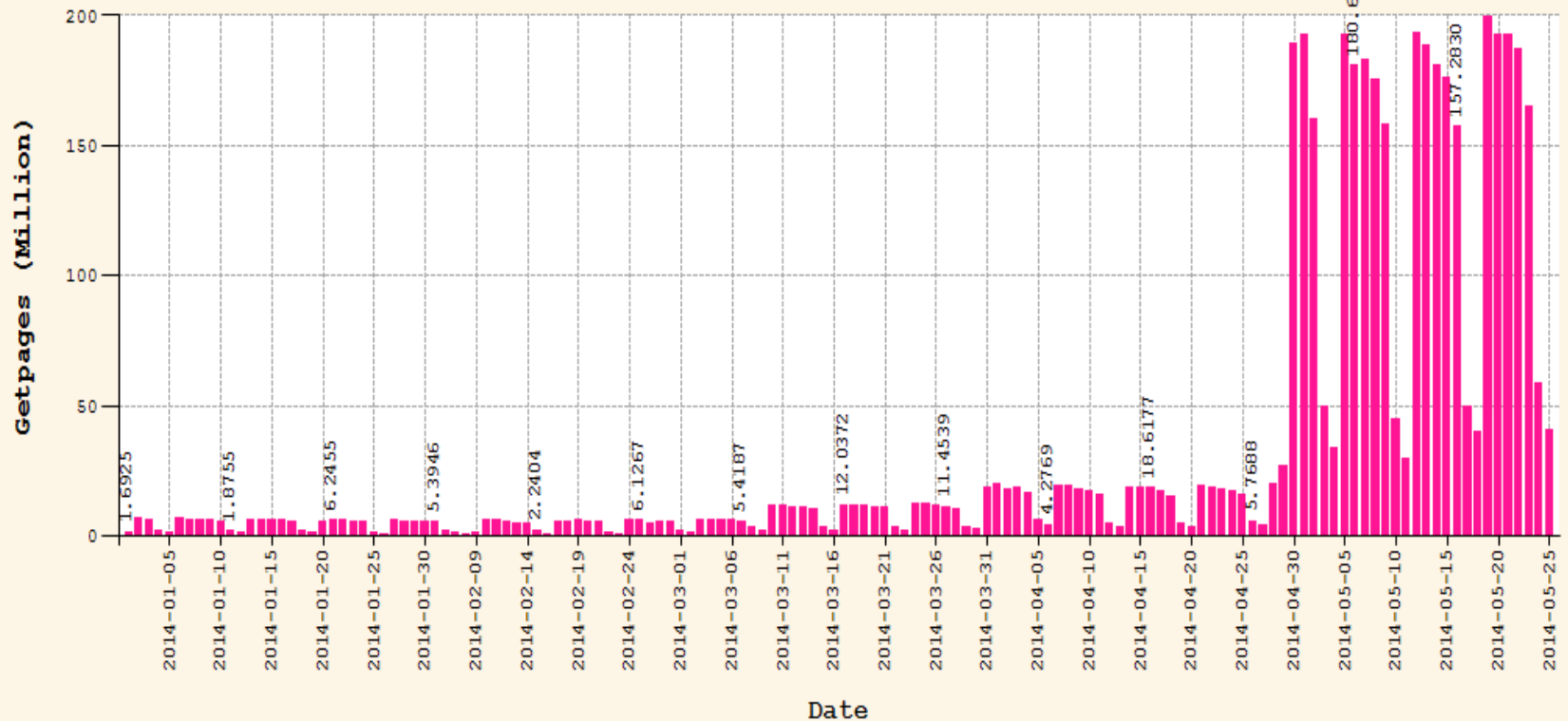


Table usage comparison data pulled from SA. One day compare for 4/16 vs 5/14 is 162+ million getpages . Notice the lack of indexed getpages.

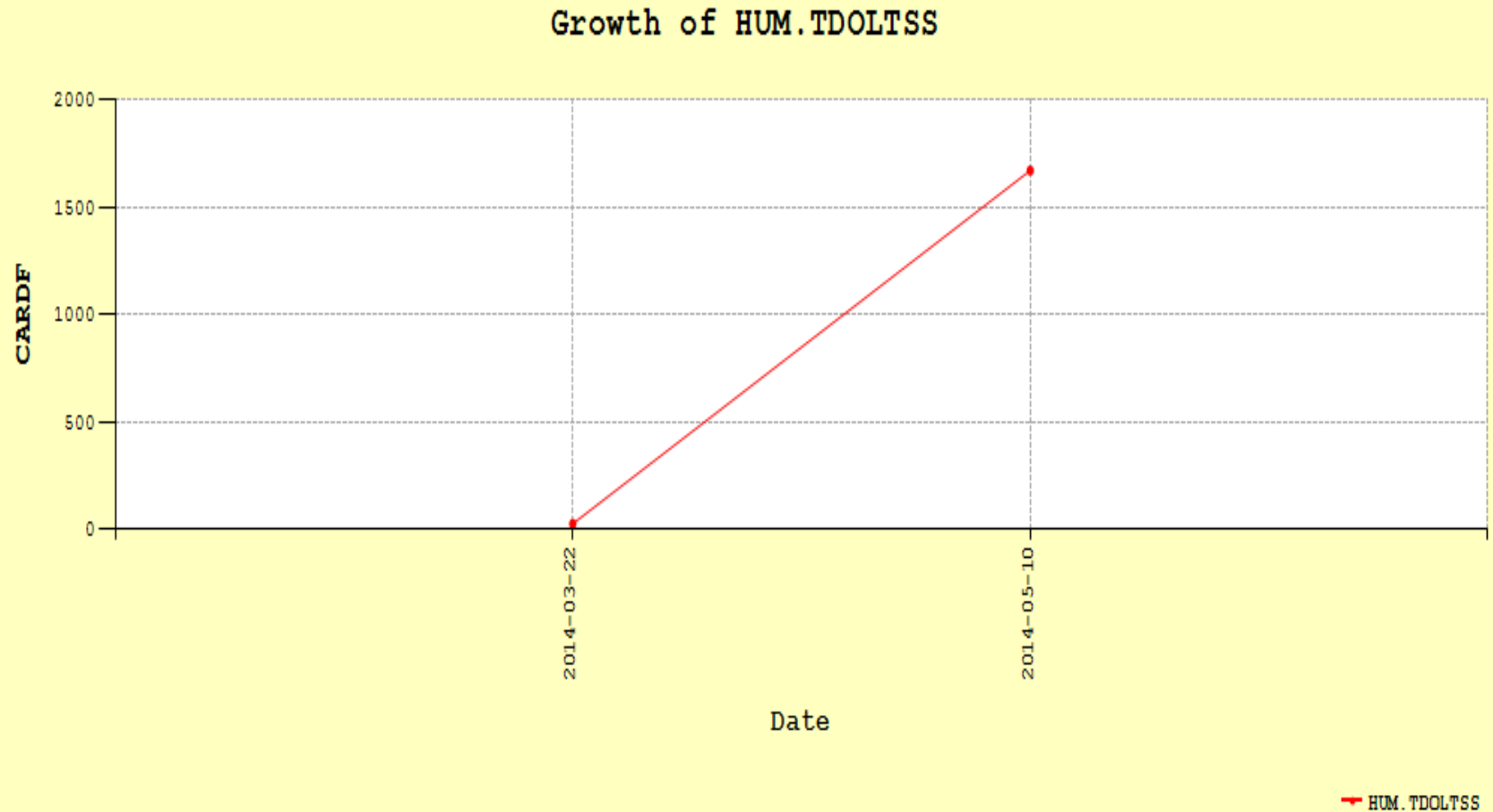
Table Usage Comparison

DB2 Region: DBP5 | Date (Base): 2014-04-16 | Date (Compare with): 2014-05-14

Pull Report

Region	Creator	Database	Table	Compare with Date	After Total GP	After Seq. GP	After Idx. GP	Base Date	Before Total GP	Before Seq GP	Before Idx GP	Difference (Total GP)	Difference (Seq GP)	Difference (Idx GP)
DBP5	HUM	DDOMBR	TDOLISS	2014-05-14	181019983	181019756	227	2014-04-16	18616930	18616854	76	162403053	162402902	151
DBP5	HUM	DMPPRD1	TMPFFV	2014-05-14	158867626	158532162	335464	2014-04-16	4434011	4098222	335789	154433615	154433940	-325
DBP5	HUM	DDOMEM	IDOPLNR	2014-05-14	43127162	43121689	5473	2014-04-16	20866783	20864387	2396	22260379	22257302	3077
DBP5	HUM	DDOCLMH	TDOAUTH	2014-05-14	52739519	42080454	10659065	2014-04-16	41925564	31233481	10692083	10813955	10846973	-33018
DBP5	HUM	DDOPLN	IDOPLAN	2014-05-14	30921712	42872	30878840	2014-04-16	22200362	56166	22144196	8721350	-13294	8734644
DBP5	SYSIBM	DSNDB01	SYSLGRNX	2014-05-14	841783409	36184	841747225	2014-04-16	834746456	32297	834714159	7036953	3887	7033066
DBP5	HUM	DDOMBR	TDOMEMC	2014-05-14	25095226	1369043	23726183	2014-04-16	19396728	669646	18727082	5698498	699397	4999101
DBP5	HUM	DDOCODE	IDOCODL	2014-05-14	31290042	158047	31131995	2014-04-16	26887967	894305	25993662	4402075	-736258	5138333
DBP5	HUM	DDOCLMH	TDOYPR	2014-05-14	5028146	0	5028146	2014-	862964	0	862964	4166182	0	4166182

Table growth in GUI built from DB2 catalog Real time statistics



EZ-DB2 for an hour shows the SQL as well

Humana 3270 Mod4 Display Session - A - @WC00212

File Edit View Communication Actions Help

Consolidated SQL In Workload Row 1 to 35 of 5859
 Enter Command: Trace Workload: DBP5ALL DB2 System: DBP5
 Scroll ==> CSR Show: A (A/S/D) More: >

Workload SQL No	Program Name	Total Stmt No	SQL Stmt Type	Total Duplicates: SQL Execs	552 SQL Consold	Total CPU	Total Clock	Total GetPages
158	MES027	796	S-CURSOR	711955	0	04:38:48	13:11:12	629063015
4329	MES021	1492	S-CURSOR	2730925	0	57:06:297	06:39:02	316548632
5738	MES021	1504	S-CURSOR	1913774	0	38:55:631	02:52:52	226301053
3252	MES021	1466	S-CURSOR	1834843	0	37:43:886	05:34:02	213692705
3285	GC17GC17	4191	S-SELECT	462	0	22:42:759	50:11:175	145202837
9	I064LNK	902	S-CURSOR	12515054	0	21:07:128	01:49:04	98507328
347	I035I035	5568	S-SELECT	326445	0	21:01:142	01:15:39	19871633
107	MES023	646	S-CURSOR	3709920	0	20:30:536	03:57:10	81538132
44	MES030	210	S-CURSOR	6912238	0	19:00:344	05:38:34	75249217
31	MES047	233	S-CURSOR	7164326	0	18:57:111	01:52:53	72628748
79	MES030	238	S-CURSOR	6909484	0	18:54:080	01:57:46	130370950
445	DDF	445	D-CURSOR	105536	18	18:19:352	10:34:32	68710746
1332	Duplicat #0013	1332	S-SELECT	71295345	11	17:52:836	35:57:012	142590696
73	MES021	1461	S-CURSOR	433396	0	13:32:715	43:17:335	67086666
842	DDF	842	D-CURSOR	13814	7450	13:26:129	25:32:461	308538
4335	MES022	462	S-CURSOR	3134472	0	13:14:943	01:38:51	87630930
53	MES042	333	S-CURSOR	4940471	0	12:28:601	03:09:23	50274302
104	MES026	376	S-CURSOR	1922739	0	12:03:440	02:23:41	55559906
89	DDF	89	D-CURSOR	455992	32236	10:45:419	25:32:191	3291809
163	MES027	941	S-CURSOR	708160	0	10:38:729	20:18:178	6000818
7	DDF	7	D-CURSOR	6265379	0	09:31:228	25:23:676	22628711
77	MES029	579	S-CURSOR	3387310	0	09:20:062	48:34:476	57485641
351	I035I035	1433	S-CURSOR	224967	0	09:18:258	35:21:841	6829838
157	MES027	1896	S-INSERT	856746	0	08:43:950	22:20:436	6019141
3269	MES045	470	S-CURSOR	25377	0	08:33:597	16:48:245	45454134
137	DDF	137	D-CURSOR	358075	0	08:25:994	07:16:31	14798557
2	DDF	2	D-CURSOR	2442378	0	08:25:904	01:27:28	16453815
23	MES046	285	S-CURSOR	7164211	0	08:15:254	01:46:48	45301070
1260	DDF	1260	D-CURSOR	6426	3303	08:10:052	15:54:823	179509
4338	MES022F	463	S-CURSOR	1681014	0	08:05:597	01:26:49	48016453
918	I035I035	5620	S-SELECT	81317	0	08:00:132	34:37:547	5716515
34	MES048	328	S-CURSOR	6911796	0	07:56:729	01:16:28	49415384
45	DDF	45	D-CURSOR	2056076	0	07:55:355	24:53:429	24898696
148	DDF	148	D-CURSOR	1344777	872	07:37:314	06:30:43	33837147
349	I035I035	1998	S-CURSOR	5130412	0	07:35:022	45:15:330	37974708

MA + a 29/035

lousna.humana.com:8998

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

SQL_CALL	INDB2_CPU	GETPAGE	GETPFFAIL	SYNCREAD	SPFETCH
— FETCH	00:00.000824	42.97	0.00	0.00	0.00
— OPEN	00:00.000078	8.00	0.00	0.00	0.00
— INSERT	00:00.000054	2.22	0.00	0.00	0.00
— OPEN	00:00.000008	0.00	0.00	0.00	0.00
— CLOSE	00:00.000005	0.00	0.00	0.00	0.00
— INSERT	00:00.000038	2.98	0.00	0.00	0.00

***** BOTTOM OF DATA *****

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

```

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

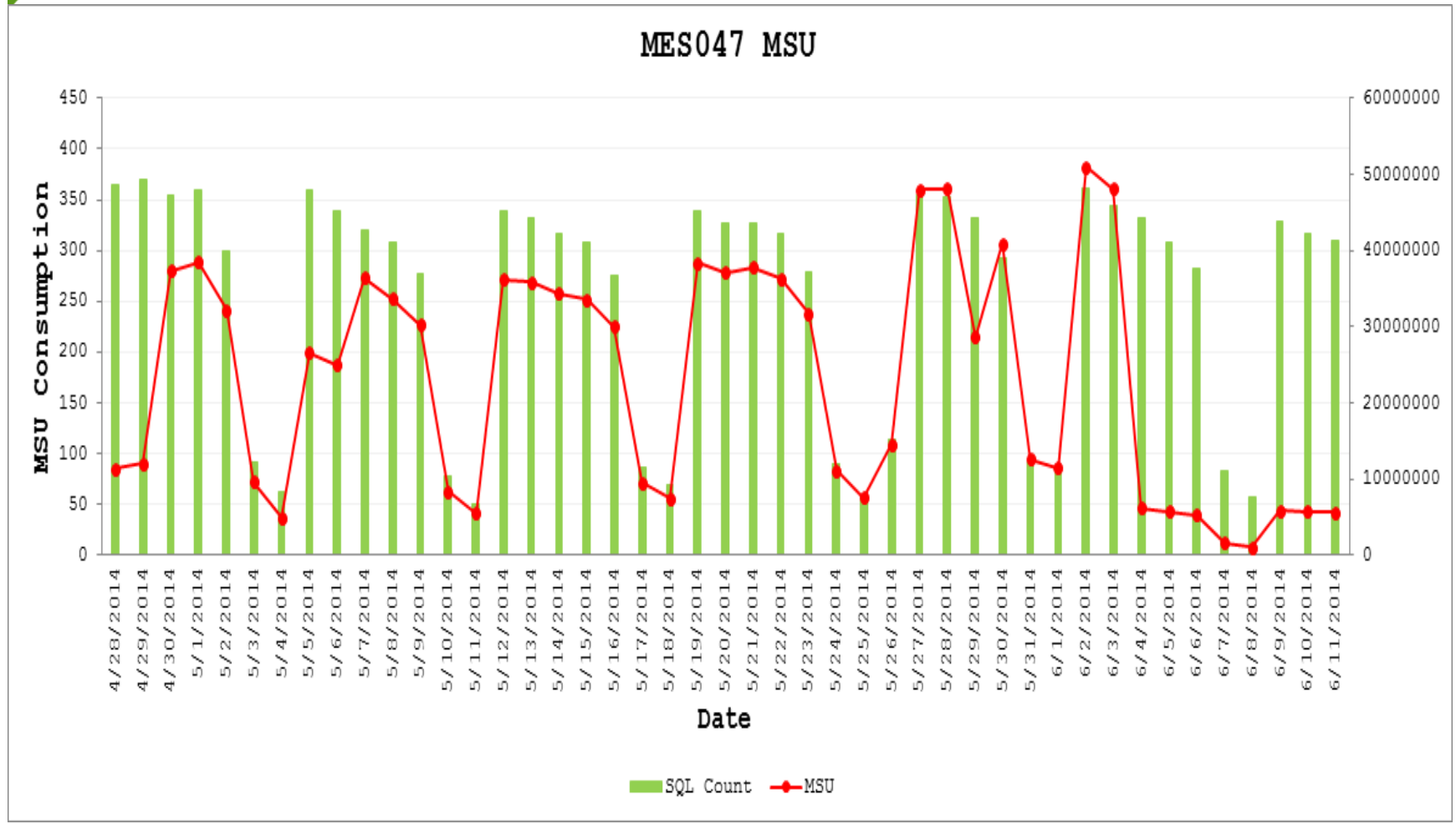
SQL_CALL	INDB2_CPU	GETPAGE	GETPFAIL	SYNCREAD	SPFETCH
INSERT	00:00.000053	2.25	0.00	0.00	0.00
FETCH	00:00.000032	4.61	0.00	0.00	0.00
OPEN	00:00.000010	0.00	0.00	0.00	0.00
OPEN	00:00.000006	0.00	0.00	0.00	0.00
CLOSE	00:00.000003	0.00	0.00	0.00	0.00
INSERT	00:00.000036	3.00	0.00	0.00	0.00

***** BOTTOM OF DATA *****

1 hour	INDB2_CPU	GETPAGE
BEFORE	09:32.699365	31285653
AFTER	01:04.778710	4078480

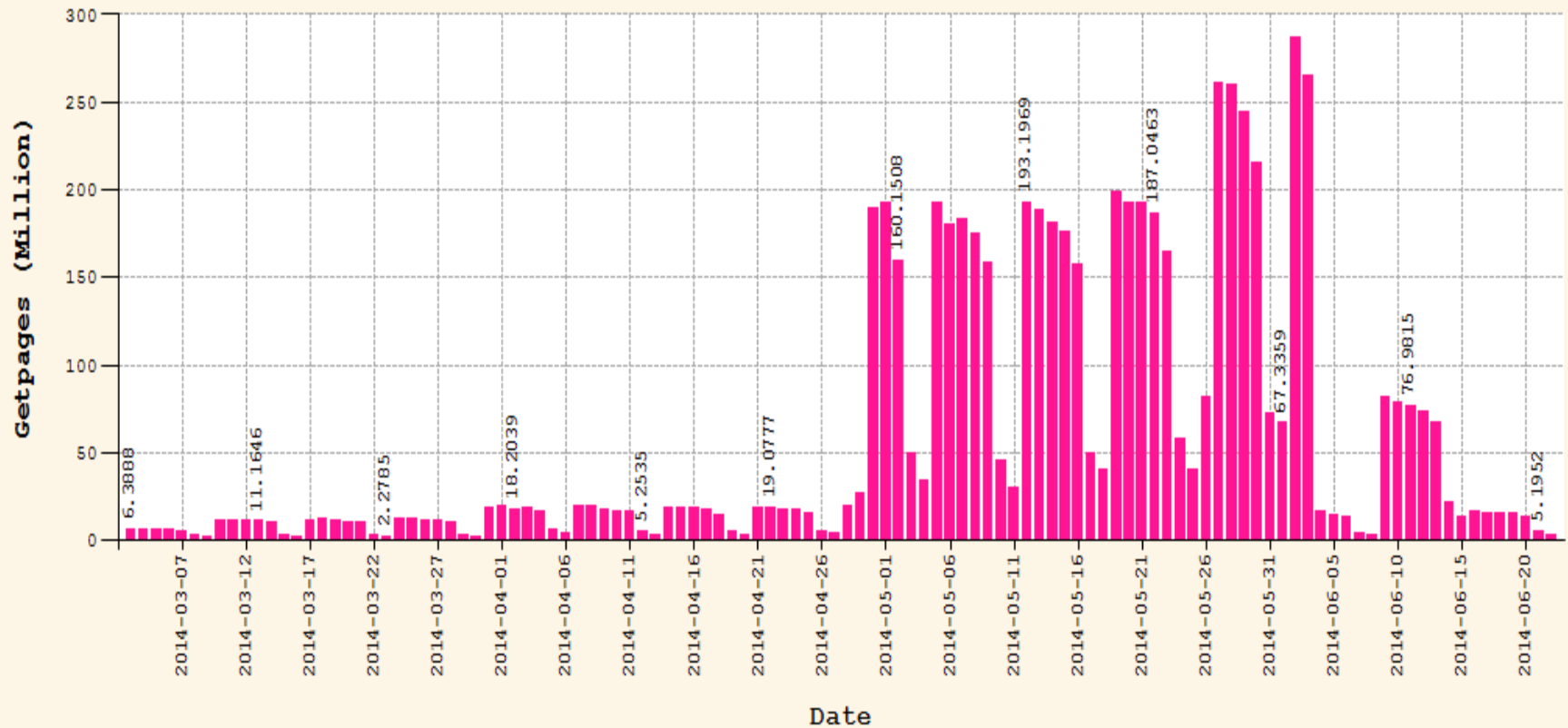
View of Program MSU after Bind

MES047 MSU



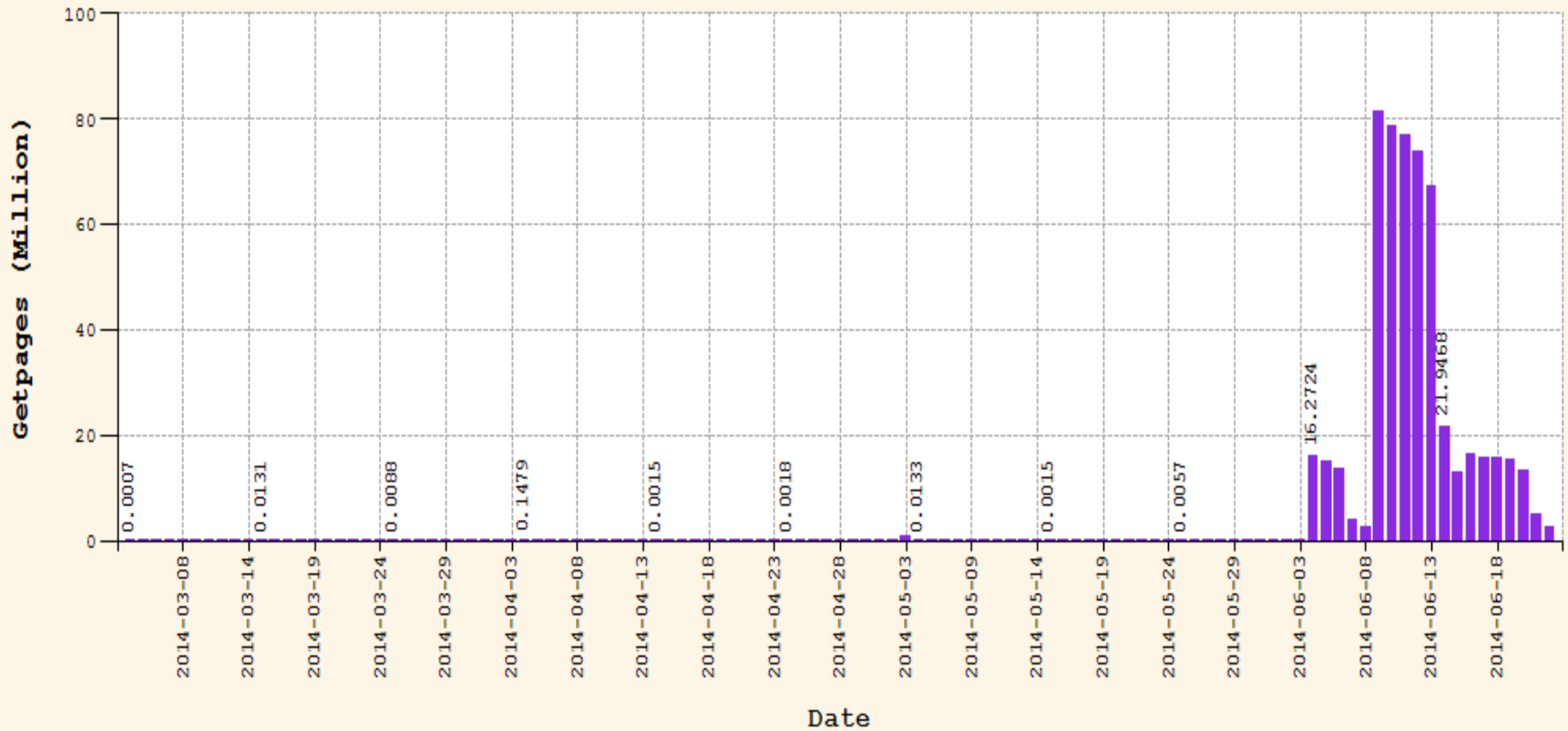
View of getpage against the table after Bind

Getpage Consumption for Table TDOLTSS (DB DDOMBR)

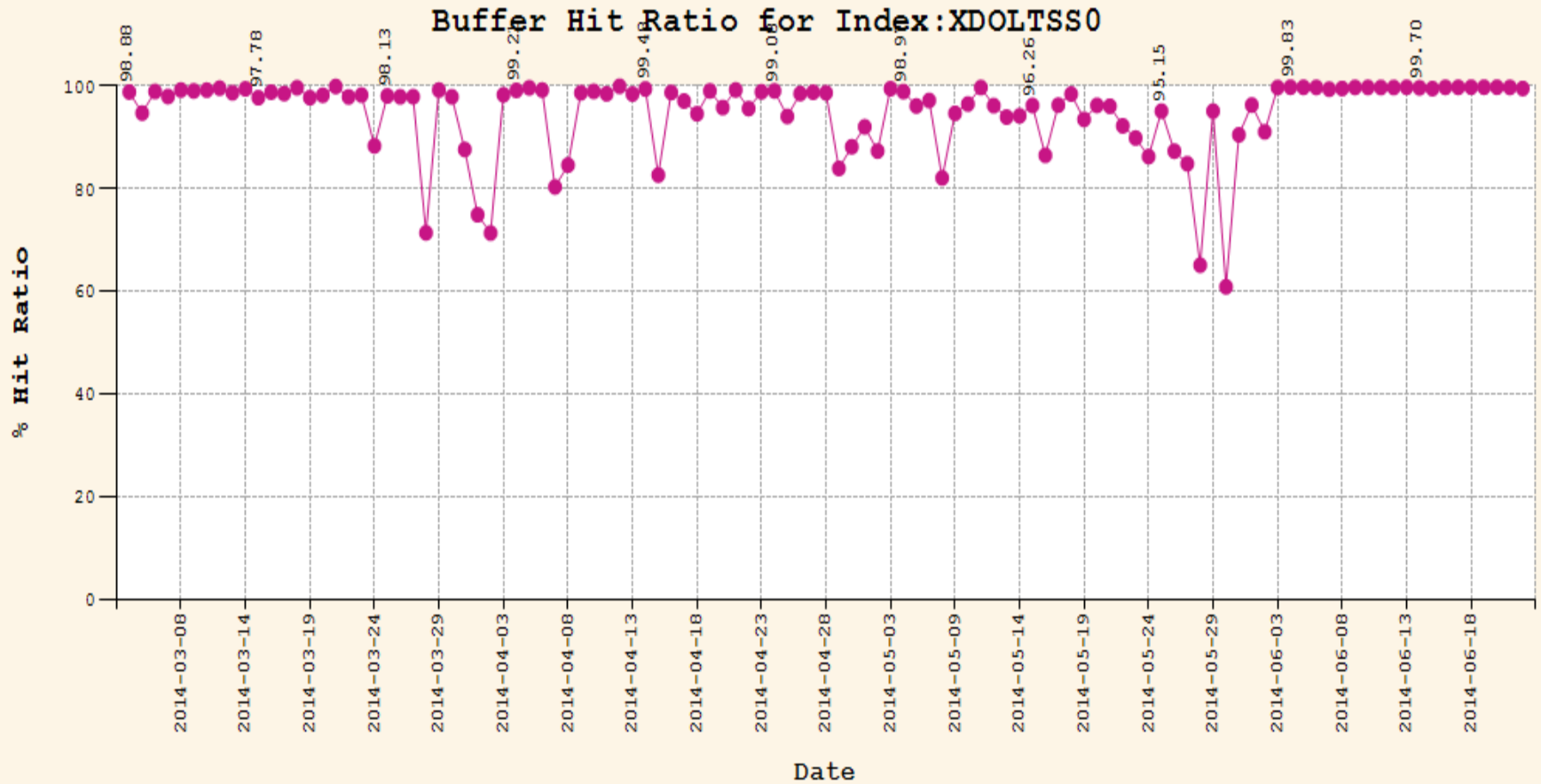


View of Index activity since bind Sourced SA

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 perform 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

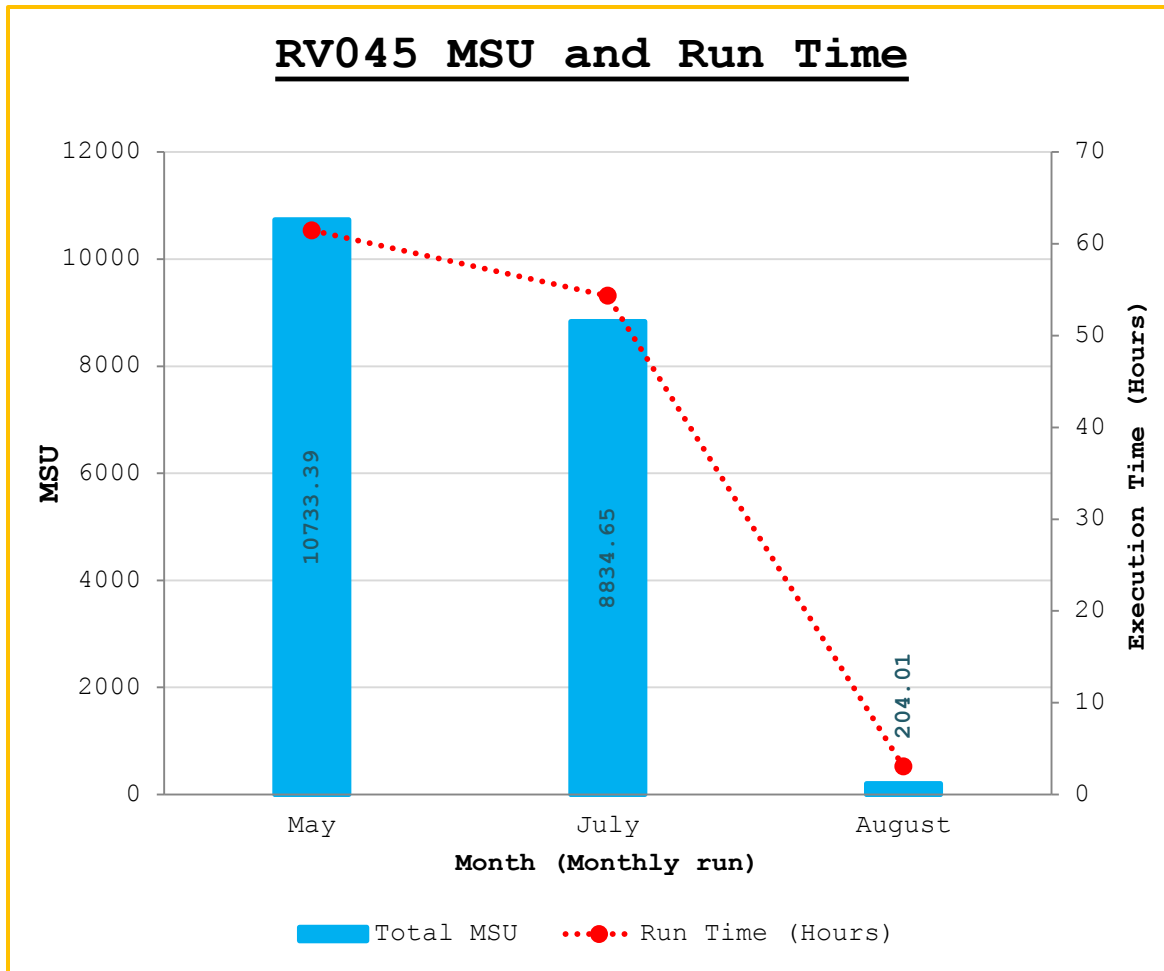
```
SELECT A.TYPE_CD_ID, A.TYPE_CD, A.SERV_RATE_AMT,  
       A.SERV_CNT, B.SERV_RATE_AMT, B.SERV_CNT  
FROM   TFNPPPX A  
       INNER JOIN TFNPPPY B  
         ON A.TYPE_CD_ID = B.TYPE_CD_ID  
         AND A.TYPE_CD   = B.TYPE_CD  
WHERE  (A.SERV_CCYMM_DT = :WS-CCYMM1  
        AND B.SERV_CCYMM_DT = :WS-CCYMM2)  
        AND ROUND( (A.SERV_RATE_AMT * :WS-PCT), 2)  
          BETWEEN B.SERV_RATE_AMT - .50  
          AND B.SERV_RATE_AMT + .50;
```

```
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_CCYMM_DT = :WS-CCYMM1  
        AND B.SERV_CCYMM_DT = :WS-CCYMM2)  
        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_CCYMM_DT ASC  
, TYPE_CD      ASC  
, TYPE_CD_ID   ASC)
```

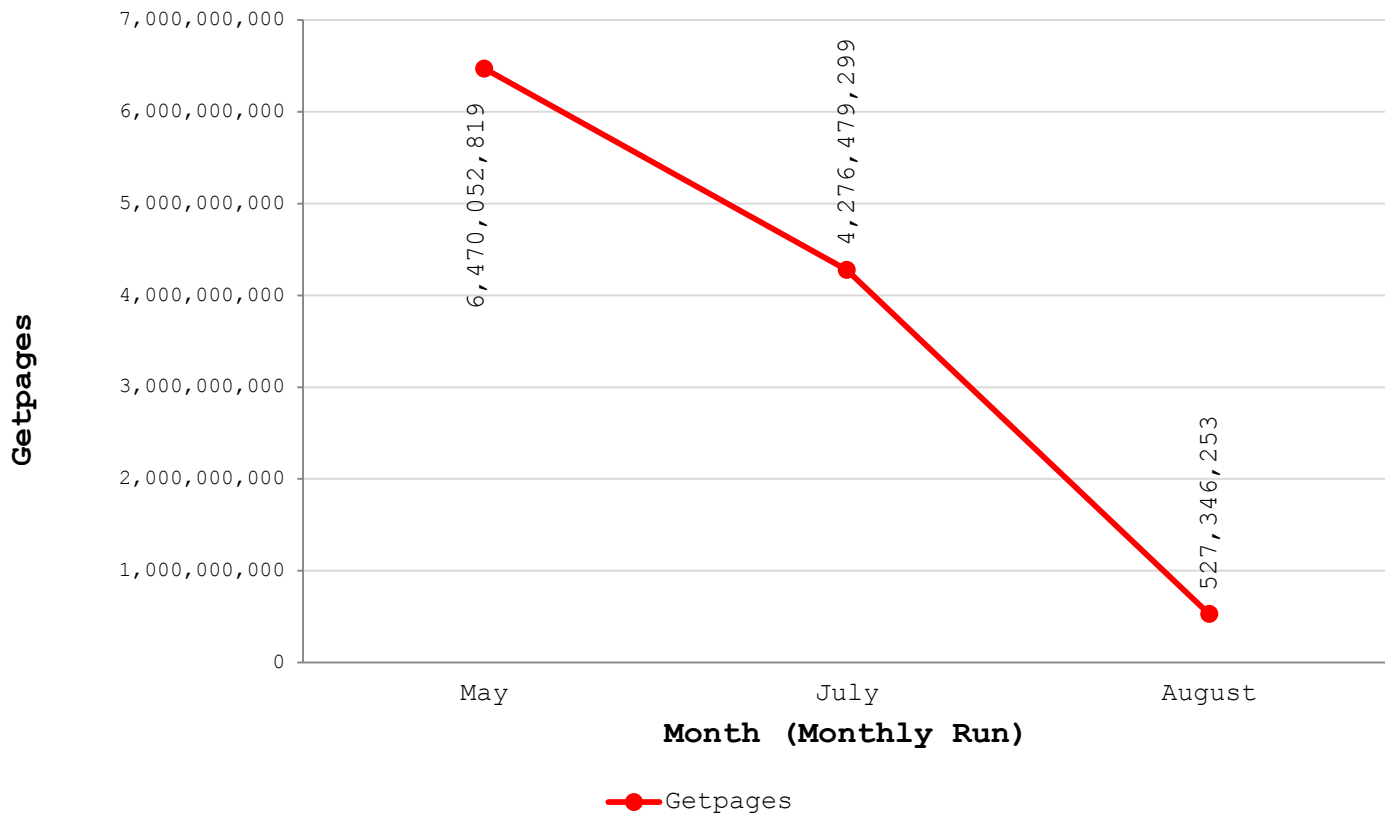
```
CREATE INDEX HUM.XFNPPPY1  
ON HUM.TFNPPPY  
(SERV_CCYMM_DT ASC  
, TYPE_CD      ASC  
, TYPE_CD_ID   ASC  
, SERV_RATE_AMT ASC)
```

Index change and SQL change example for BATCH

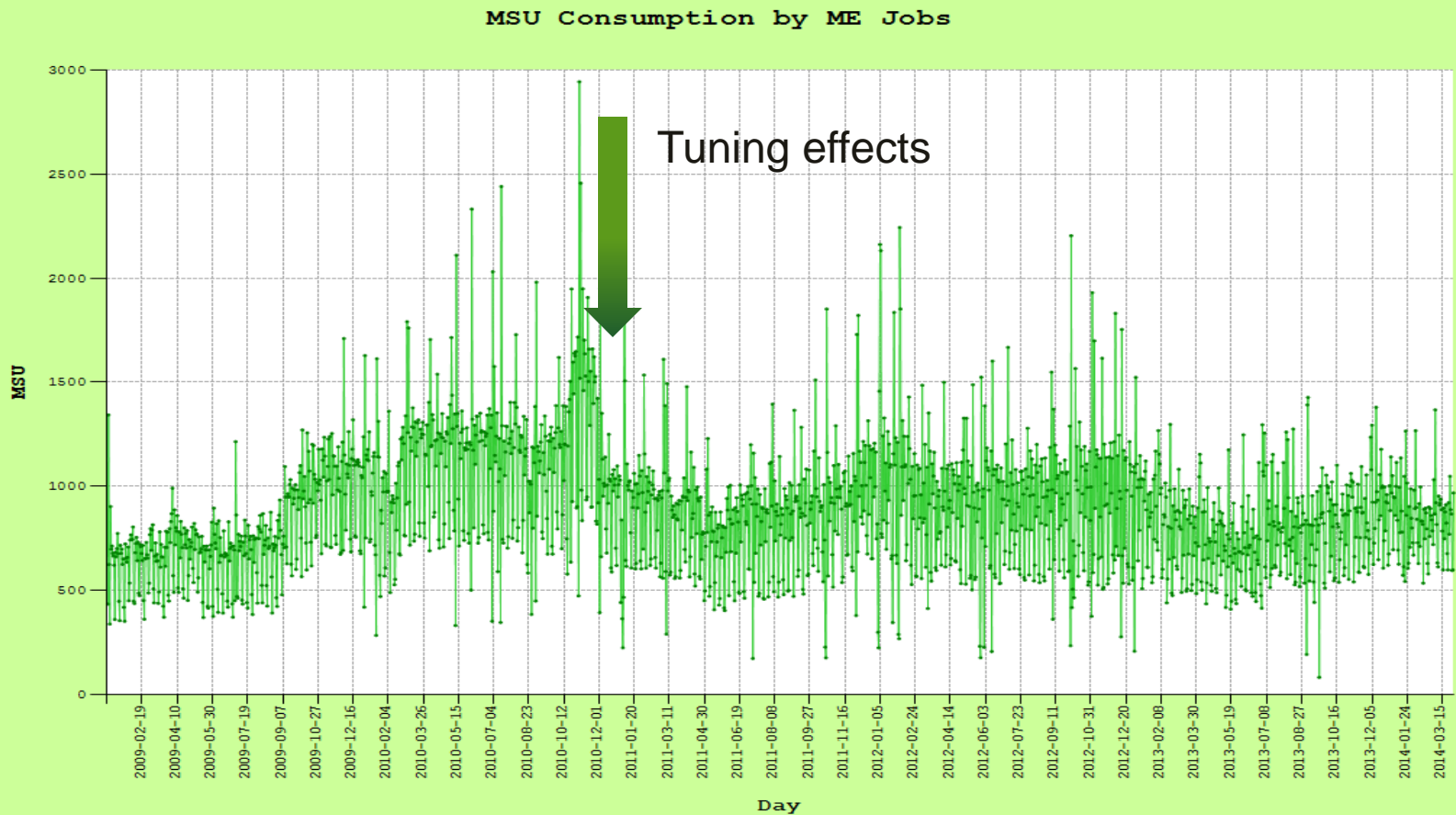


GETPAGE = MONEY

RV045 Getpage Consumption

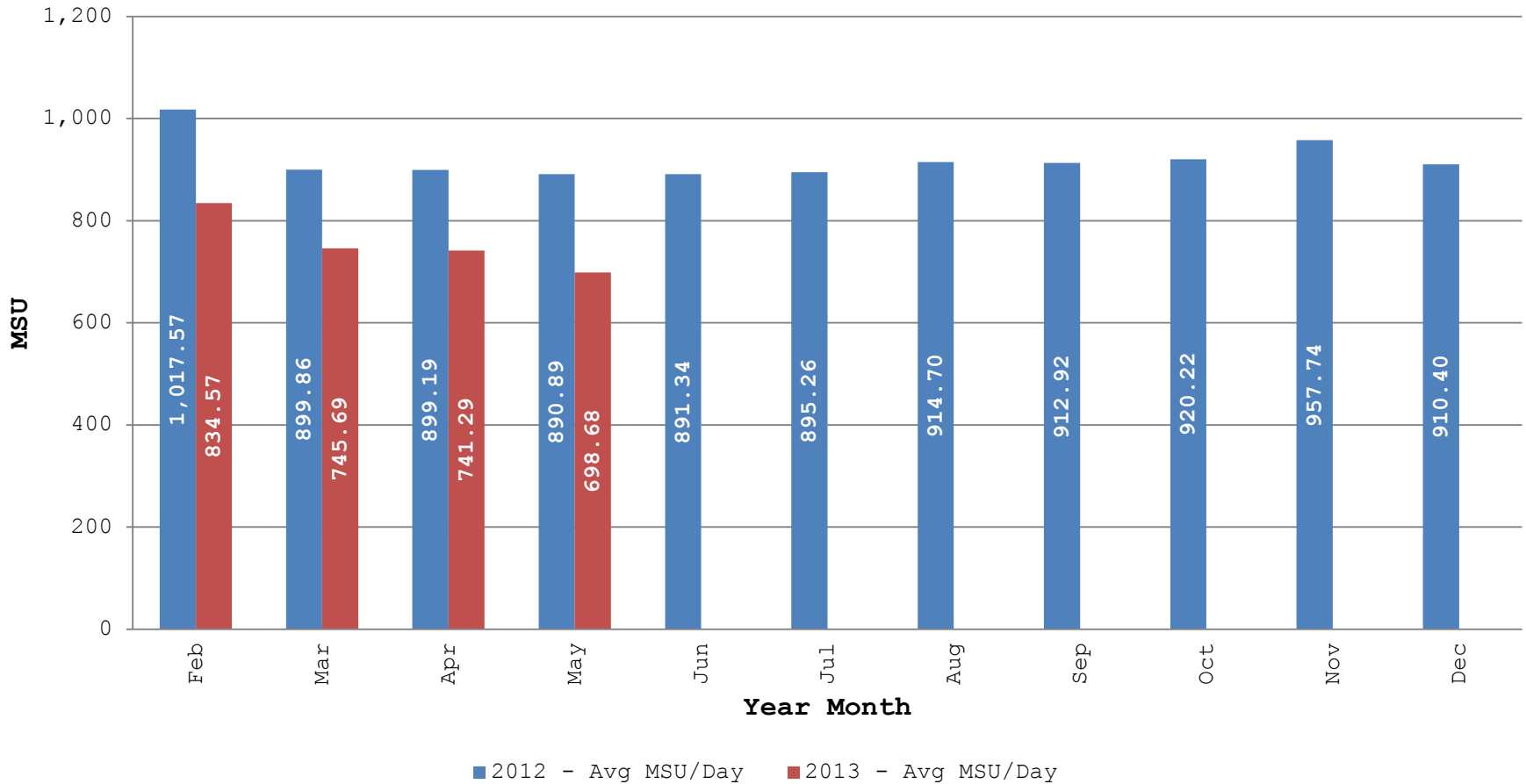


Creating a flat Master file with FASTUNLOAD and Match/merge for multiple batch jobs as input



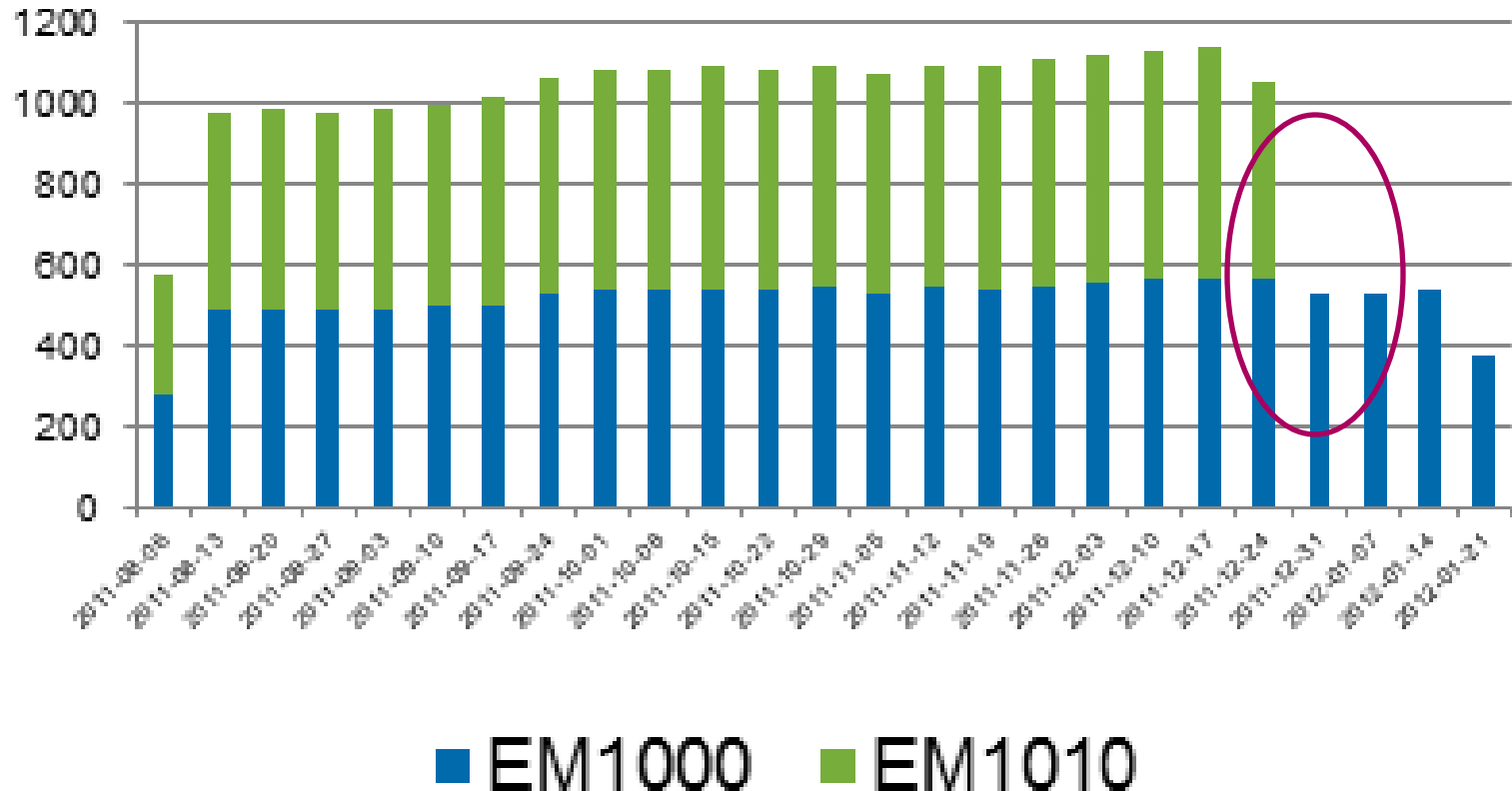
Approximately 160 MSU saved per day

ME Batch : Avg MSU/Day Consumption



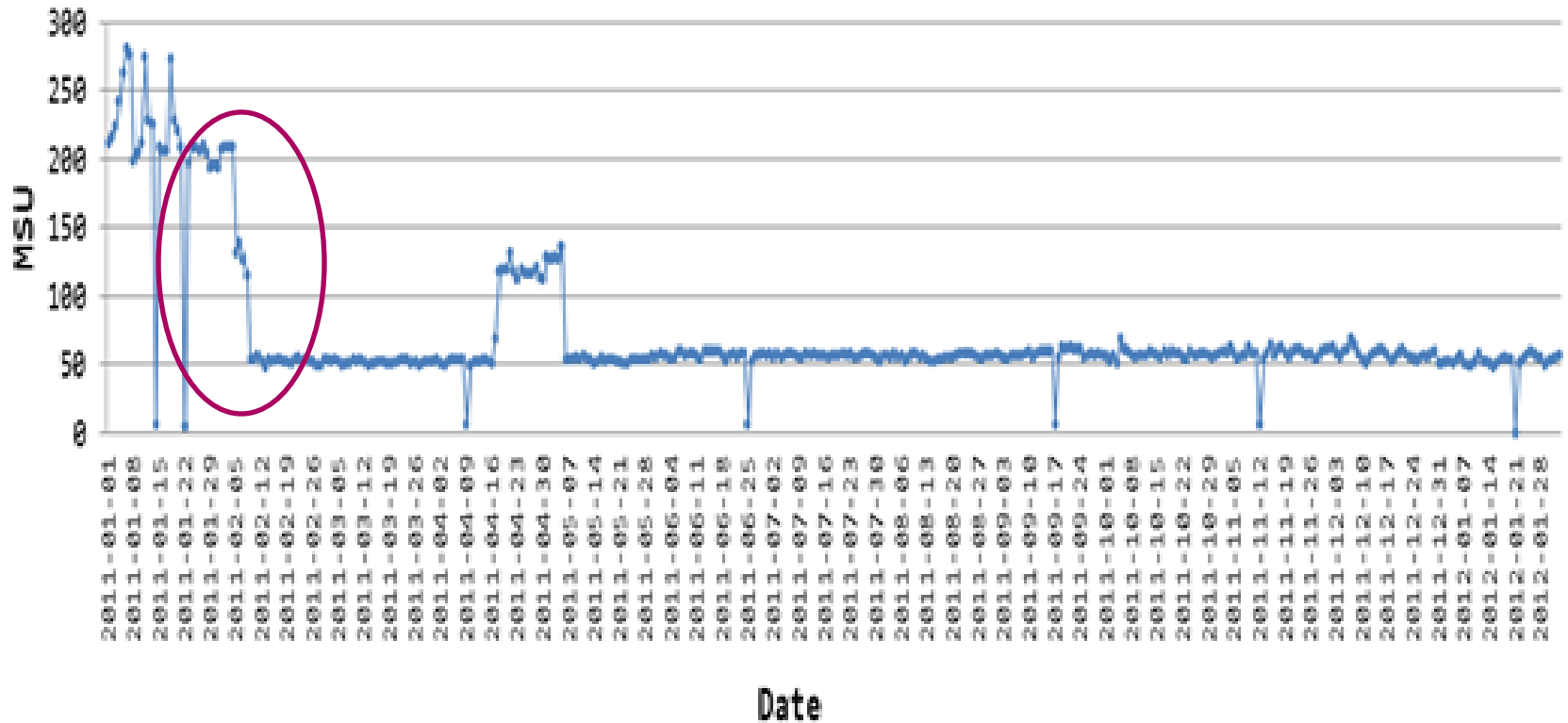
Best case, don't run the same work twice!

EM1000 and EM1010 MSU's



SQL join broken into desperate fastunloads and match merged

AE0795* MSU Consumption



Tie Detector results to Dynatrace Pure path example

```
OPID ==> S10WEB DB2 SSID ==> DB59
Total/Avg ==> T
Interval Date => 04-04-14 Interval Time => 14:00:00 Elapsed Time => 01:00
-----
S -View SQL stats, Q -View SQL text, E -Explain, D -View Detail
-----
START_TIME PLANNAME SQL INDB2_TIME INDB2_CPU GETPAGE
-----
14:36:11.380 DISTSERV 3 00:07.755971 00:04.351168 272787
14:36:19.585 DISTSERV 3 00:06.096578 00:04.079058 272787
```

```
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.TOVGRP 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

PurePath	Response Time [ms]	Breakdown	S...	Agent	Application	Start Time	Duration [ms]
/SolarWebServicesWS/GroupAndCoverageRelatedSe...	6147.21	io (100%)	3	SOAP_Web[Local_Net...	SolarWebServicesWS	2014-04-04 14:40:07.261	6147.21

PurePaths Contributors Errors

PurePath Tree (showing only relevant nodes)

Method	Argument	Exec Total [ms]	Breakdown	Class
System.Web.HttpApplication.IExecutionStep.Execute()	/SolarWebServices...	6147.21	io (100%)	HttpApplication-
SearchGroup(GroupSearchDto)		6146.61	io (100%)	GroupAndCover-
Database Summary		6139.94		-
1 x Select DISTINCT RTRIM(GRP.GROUP_NBR) GroupNumber,RTRIM(GRP.PLATFORM_CD) Platform, RTRIM(GRP.BLI	SolarSecurity	6139.94		-

Details

Node Details

Method:

```
1 x 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.TOVGRP 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
```

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

PurePath	Response Time [ms]	Breakdown	Size	Agent	Application	Star
/SolarWebServicesWS/GroupAndCoverageRelatedSe...	43.48	io (100%)	3	SOAP_Web[Local_Net...	SolarWebServi...	2014-04-24 11:59:!

PurePaths Contributors Errors

PurePath Tree (showing only relevant nodes)

Method	Argument	Exec Total [ms]	Breakdown
System.Web.HttpApplication.IExecutionStep.Execute()	/SolarWebServices...	43.48	io (100%)
SearchGroup(GroupSearchDto)		42.75	io (100%)
Database Summary		35.70	
1 x Select DISTINCT RTRIM(GRP.GROUP_NBR) GroupNumber,RTRIM(GRP.PLATFORM_CD) Platfc	SolarSecurity	35.70	

Details

Node Details

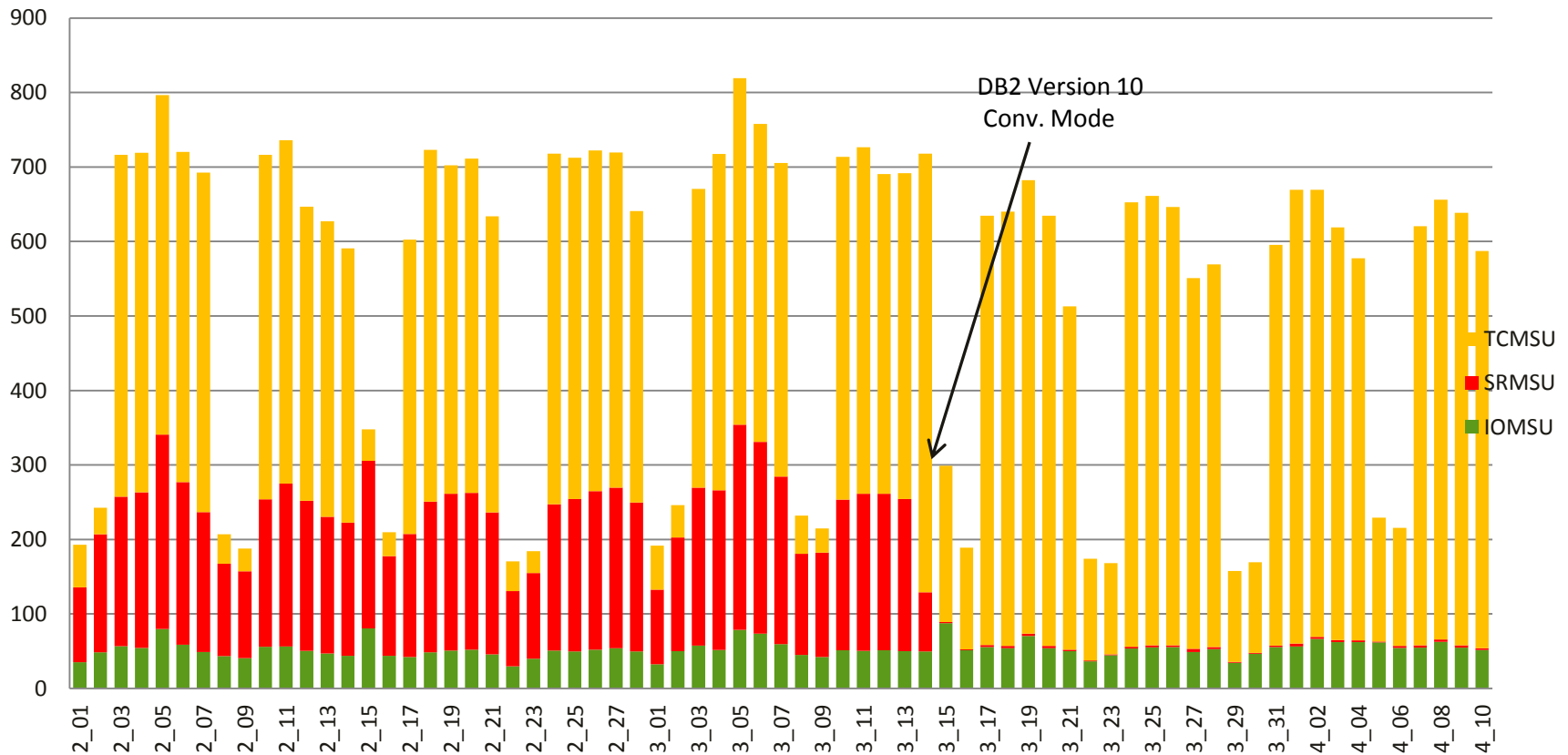
Method: 1 x Select DISTINCT RTRIM(GRP.GROUP_NBR) GroupNumber,RTRIM(GRP.PLATFORM_CD) Platform, RTRIM(GRP.BLK_BUS_CD) BlockOfBusiness, RTRIM(GRP.GROUP_NM) GroupName,GRP.EFF_DT EffectiveDate,GRP.TERM_DT TermDate From Hum.TOVGRP GRP INNER JOIN Hum.TMAPIDC PIDC ON PIDC.GROUP_NBR = GRP.GROUP_NBR INNER JOIN 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

Clipping Coupons

- Newer database releases
- Page fixing buffers

Immediate relief with DB2 Version 10

DBP8 Started Task TCB,SRB,IO MSU



Page fix

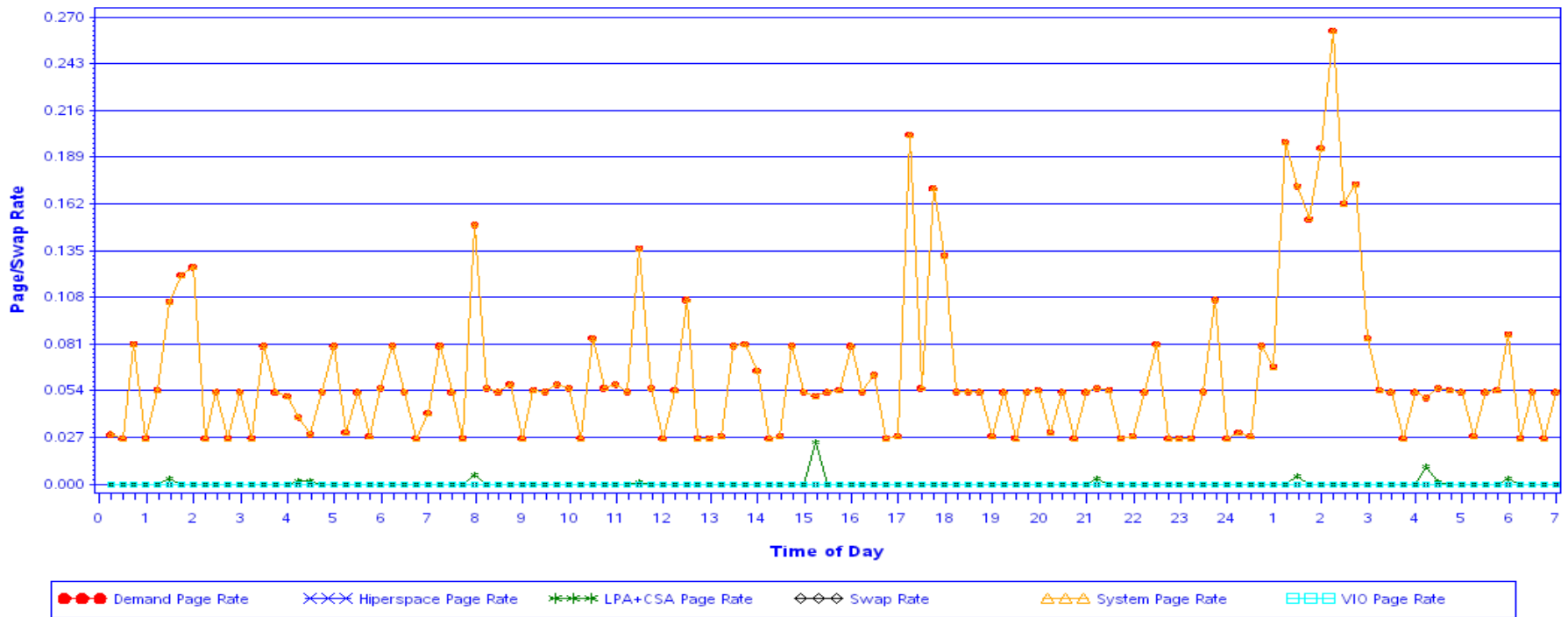
We page fixed two of the busiest buffer pools in one of our busiest DB2 region. These two buffers are for large randomly accessed tablespaces and large randomly accessed indexes. These two BP represent 29% 4K pages allocated to the DB2 region and about 20% getpage activity in the pools.

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.

LPAR Paging Rates
Wednesday, February 12, 2014
System Name=SYS0

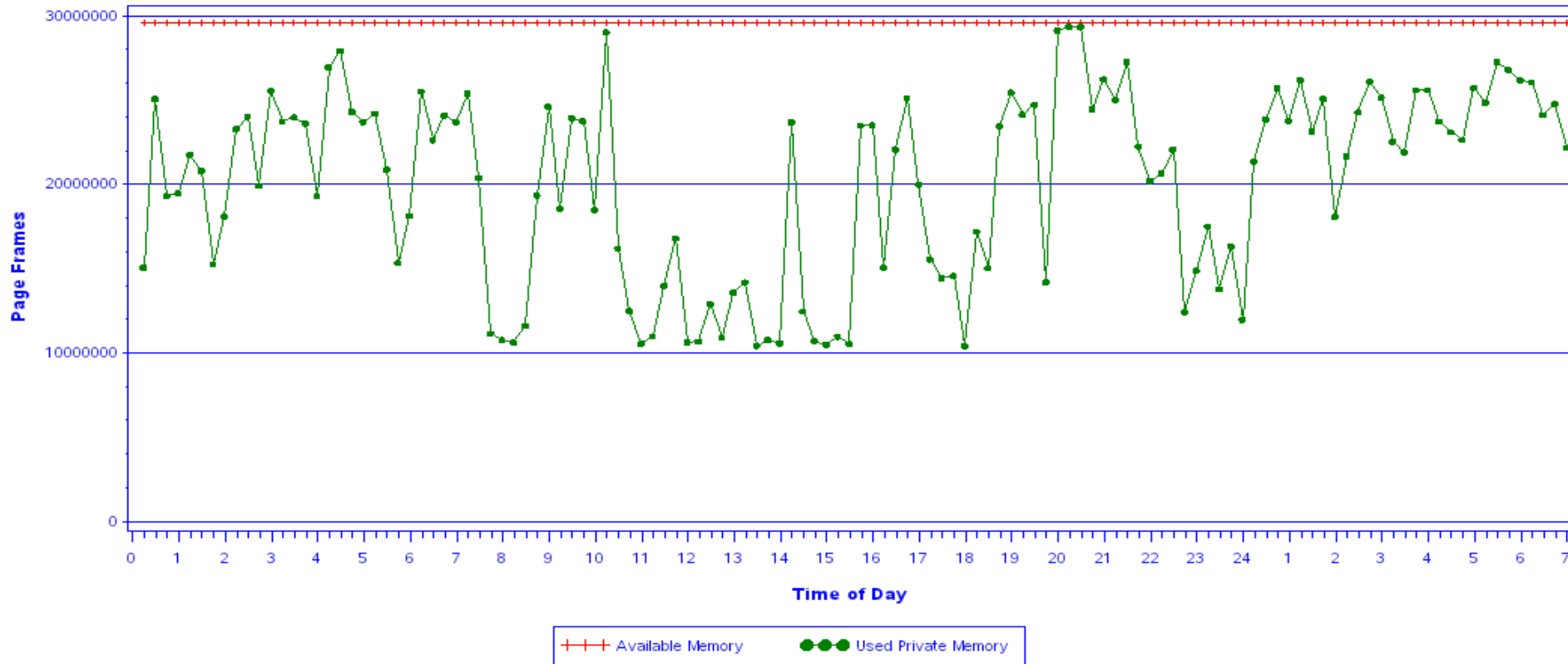


Source: SMF/MCS DETAIL2.SCPPAGnn data

F

Check the high private memory, most notably Zone 1 workload (onlines) 8am-5pm.

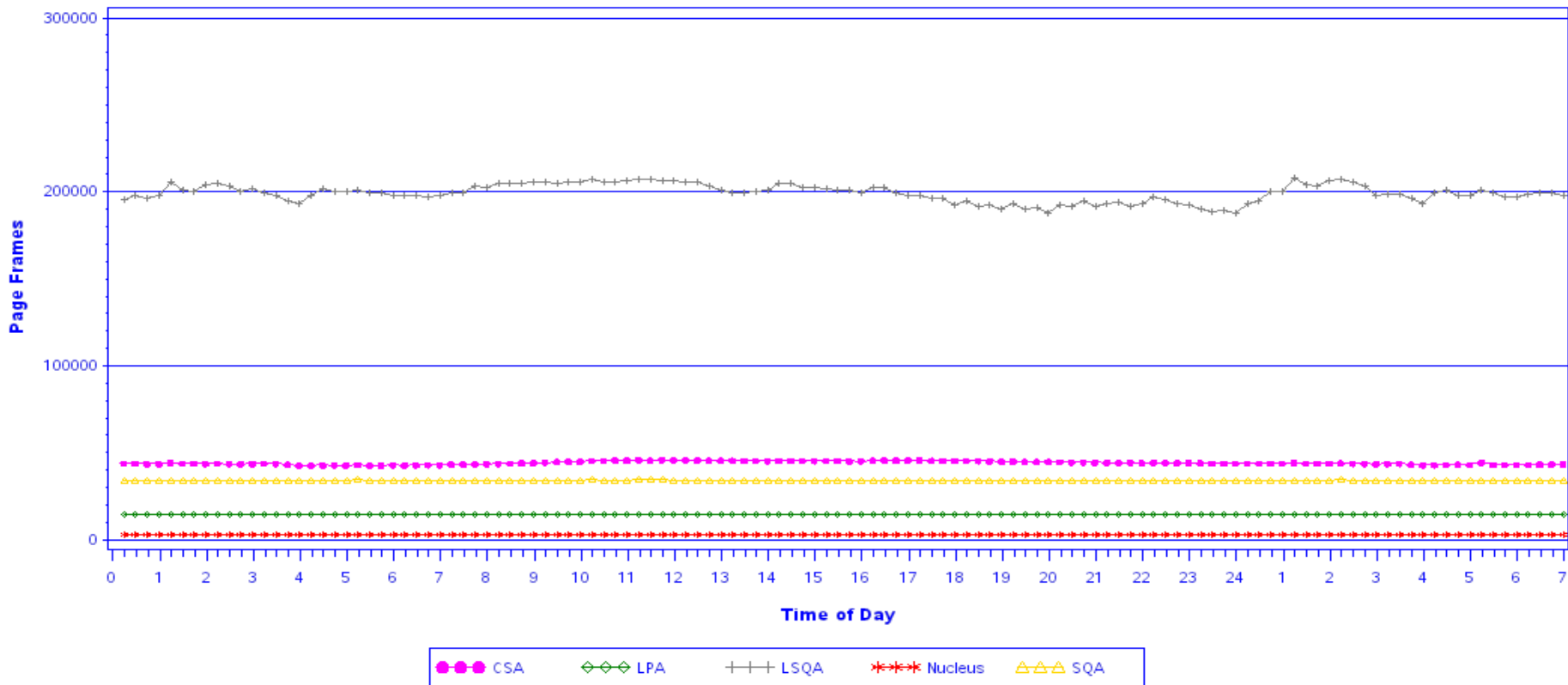
Private Memory Usage
Wednesday, February 12, 2014
System Name=SYS0



Source: SMF/MCS DETAIL2.SCPPAGnn Data

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.SCPPA.Gnn Data

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
```

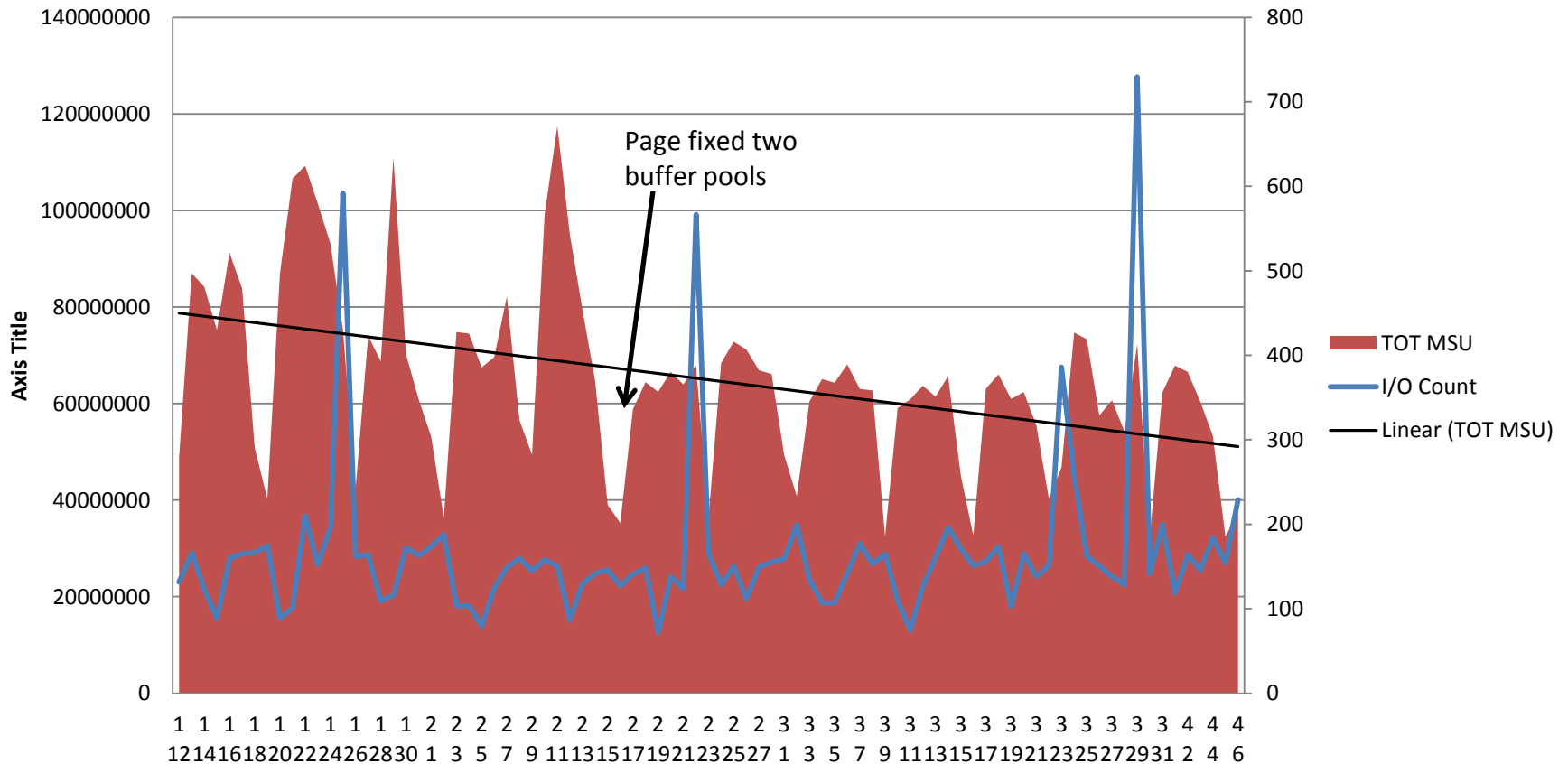
```
DSNB411I -DBP5 RANDOM GETPAGE    = 29356257 SYNC READ I/O (R) =  
4924899
```

```
    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



Rename your SYSDUMMY table to SYSPROBABLYDONTNEEDTOUSETHISTABLEDUMMY

An example showing use of DSNDB06 SYSDUMMY1

PurePath	Response Time [ms]	Breakdown	Size	Agent	Application	
/gcpws/Inquiries.aspx	307.24	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	128.33	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	84.47	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	67.73	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	67.08	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	54.25	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	53.68	cpu (96%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	53.03	cpu (69%) io (31%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	52.15	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	46.49	io (93%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	45.96	cpu (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	41.99	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	40.21	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	39.35	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	38.66	io (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:
/gcpws/Inquiries.aspx	37.91	cpu (100%)	4	CCPGCP_Web[Local_...	CCP/GCP	2014-04-02 13:

PurePaths Contributors Errors

PurePath Tree (showing only relevant nodes)

Method	Argument	Exec Total [ms]	Break
System.Web.HttpApplication.IExecutionStep.Execute()	/gcpws/Inquiries....	307.24	
@ IsFirstTimeCaller(long, String)		306.61	
Database Summary		215.72	
1 × SELECT 1 FROM SYSIBM.SYSDUMMY1 WHERE EXISTS (SELECT CINQ.CONTACT_ID FROM ...)		210.28	
1 × SELECT 1 FROM SYSIBM.SYSDUMMY1 WHERE EXISTS (SELECT INQ.CONTACT_ID FROM ...)		5.43	

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 ==> DSNDB06           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

SQLCALL	PLCNT	PROGRAM	STMT#	SECT#	PCTTGP	SQL	TB_GETPAGE
SELECT	1	EK012	0001450	00005	94.0%	5961742	11923484
SELECT	4	EK61EK61	0000263	00002	2.7%	169872	339744
FETCH	1	SYSSH200	0000004	00004	2.3%	29696440	297736
SELECT	1	RVSTMAN	0000731	00006	.4%	27954	55908
FETCH	1	SYSSTAT	0000004	00004	.2%	2898170	31186
SELECT	1	IQ908	0003398	00017	.2%	12531	25062
SELECT	1	RV082	0003051	00002	.0%	1260	2520
SELECT	1	RV082	0003728	00005	.0%	454	908
SELECT	1	RV082	0003734	00006	.0%	454	908
FETCH	1	DQPS024	0002474	00001	.0%	3084	286
SELECT	1	IOK2IOK2	0001513	00004	.0%	83	166
FTCH RSN	1	UTAFUSQ9	0000750	00002	.0%	1523358	80
SELECT	1	RV084B	0005581	00006	.0%	19	38
SELECT	1	RV099B	0002938	00003	.0%	13	26
FETCH	1	DSNESM68	0000391	00001	.0%	2404	18

Little things add up, sweat the small stuff.

- 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 **SYSDUMMY1**

```
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

B23 Program EF400 is consuming over a Billion Getpages every day. Looks like it is been called from EF865 which consumes over 3000 MSU/Week. Below are some observation and suggestions to improve it's performance.

BOP : Excessive Resource Consumption Detail

BOP Thresholds

* BOP Type: Batch MSUs if Others -> []

* Job: EF8630* * Written By: Kapil Gupta

* Program: EF400 (from EF865) * Contact Person (Dev Team): []

* System Code (LPAR): EF (SYSM, DBP7) * Observation Date/Time: 1/15/2014

MSUs

Priority: 1

~Estimated Savings (%) :	MSU	Getpages	IOs	SQLErrors

SD Ticket# []

Screen Prints History Other Details

Form have

- Detector Stats
- SQL
- Explain

* Short description of the existing scenario and problem

Program EF400 is consuming over a Billion Getpages every day. Looks like it is been called from EF865 which consumes over 3000 MSU/Week. Below are some observation and suggestions to improve it's performance.

Details | BOP Thresholds | Screen Prints | Performance History | Other Details

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 .

The screenshot shows a Microsoft Excel spreadsheet with a yellow background. The spreadsheet is titled 'BOP Program EF400 [Compatibility Mode] - Microsoft Excel'. The active sheet is 'J21'. The spreadsheet content is as follows:

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
4				Other Details															
5																			
6																			
7				Original SQL in 8700-LOOK-FOR-A-TRUMP						New SQL for 8700-LOOK-FOR-A-TRUMP									
8				FK_PROV_ID						SELECT DISTINCT A.FK_PROV_ID									
9				PROV_TYPE						,B.PROV_TYPE									
10				FIRST_NAME						,B.FIRST_NAME									
11				LAST_NAME						,B.LAST_NAME									
12				CATEGORY						,B.CATEGORY									
13				BILLING_OFFICE_FG						,B.BILLING_OFFICE_FG									
14				PROV_TYPE_IND						,A.PROV_TYPE_IND									
15				PREFERRED_PHONE_ID						,B.PREFERRED_PHONE_ID									
16				EFFECTIVE_DATE						,A.EFFECTIVE_DATE									
17				END_DATE						,A.END_DATE									
18				EDS.SECONDARY_PROV_ID A						FROM EDS.SECONDARY_PROV_ID A									
19				EDS.PROVIDER B						,EDS.PROVIDER B									
20				A.PROVIDER_ID = :WS-ID-2-LOOK-FOR				OR		,EDS.SECONDARY_PROV_ID C									
21				A.PROVIDER_ID = :WS-EDI-NAME-FIRST-LAST-ALL				OR		WHERE ((A.PROVIDER_ID = :WS-ID-2-LOOK-FOR									
22				A.PROVIDER_ID = :WS-EDI-NAME-LAST-FIRST-ALL				OR		A.PROVIDER_ID = :WS-EDI-NAME-FIRST-LAST-ALL									
23				A.PROVIDER_ID = :WS-MEMH-NAME-LAST-FIRST-ALL				OR		A.PROVIDER_ID = :WS-EDI-NAME-LAST-FIRST-ALL									
24				A.PROVIDER_ID = :WS-MEMH-NAME-FIRST-LAST-ALL				OR		A.PROVIDER_ID = :WS-MEMH-NAME-LAST-FIRST-ALL									
25				A.PROVIDER_ID = :WS-EDIM-NAME-FIRST-LAST-ALL				OR		A.PROVIDER_ID = :WS-MEMH-NAME-FIRST-LAST-ALL									
26				A.PROVIDER_ID = :WS-EDIM-NAME-LAST-FIRST-ALL				OR		A.PROVIDER_ID = :WS-EDIM-NAME-FIRST-LAST-ALL									
27				A.PROVIDER_ID = :WS-MEMHM-NAME-LAST-FIRST-ALL				OR		A.PROVIDER_ID = :WS-EDIM-NAME-LAST-FIRST-ALL									
28				A.PROVIDER_ID = :WS-MEMHM-NAME-FIRST-LAST-ALL)						A.PROVIDER_ID = :WS-MEMHM-NAME-LAST-FIRST-ALL									
29										A.PROVIDER_ID = :WS-MEMHM-NAME-FIRST-LAST-ALL) OR 0 = 1)									
30				A.PROV_TYPE_IND = :WS-OLD-LIC OR						AND (A.PROV_TYPE_IND = :WS-OLD-LIC OR									
31				A.PROV_TYPE_IND LIKE :WS-TYPE-2-LOOK-FOR)						A.PROV_TYPE_IND LIKE :WS-TYPE-2-LOOK-FOR)									
32																			

The spreadsheet also shows a 'Page' label in cell N4 and a status bar at the bottom with '100%' zoom level.

Remove the separate existence select and incorporate it into the join.

The screenshot shows a Microsoft Excel spreadsheet with a SQL query. The query is split into two columns, J and K. Column J contains a SELECT statement with various filters. Column K contains a WITH UR; statement with similar filters. Below the query, there are two sections: 'Path Analysis' and 'Access Path Analysis' with performance metrics.

```

35      AND :WS-CLAIM-DATE
36 END_DATE BETWEEN :WS-CLAIM-DATE
37      AND '9999-12-31'
38 IDENTIFIER = A.FK_PROV_ID
39 INACTIVE_FLAG = 'N'
40 (B.CATEGORY = :WS-CATEGORY1 OR
41 B.CATEGORY = :WS-CATEGORY2 OR
42 B.CATEGORY = :WS-CATEGORY3)
43 OR
44 (B.CATEGORY = :WS-CATEGORY4
45 B.PROV_TYPE = :WS-PROVIDER-TYPE))
46
47 XISTS
48 SELECT C.FK_PROV_ID FROM EDS.SECONDARY_PROV_ID C
49 WHERE C.PROVIDER_ID = :WS-LOOKUP-PROVIDER-ID
50      AND C.PROV_TYPE_IND = 'TX'
51      AND C.VOID_FLAG = 'N'
52      AND C.EFFECTIVE_DATE BETWEEN '0001-01-01'
53      AND :WS-CLAIM-DATE
54      AND C.END_DATE BETWEEN :WS-CLAIM-DATE
55      AND '9999-12-31'
56      AND C.FK_PROV_ID = A.FK_PROV_ID)
57 R;
58
59 Path Analysis:
60
61 (ms) 1 (su) 3 (tc) +.81597 E+01
62
63 0 PPlnNo: 0 QblkNo: 1 PlanNo: 1 MxOpSq: 0

```

```

AND A.EFFECTIVE_DATE BETWEEN '0001-01-01'
AND :WS-CLAIM-DATE
AND A.END_DATE BETWEEN :WS-CLAIM-DATE
AND '9999-12-31'
AND B.IDENTIFIER = A.FK_PROV_ID
AND B.INACTIVE_FLAG = 'N'
AND ((B.CATEGORY = :WS-CATEGORY1 OR
B.CATEGORY = :WS-CATEGORY2 OR
B.CATEGORY = :WS-CATEGORY3)
OR
(B.CATEGORY = :WS-CATEGORY4
AND B.PROV_TYPE = :WS-PROVIDER-TYPE))
AND C.PROVIDER_ID = :WS-LOOKUP-PROVIDER-ID
AND C.PROV_TYPE_IND = 'TX'
AND C.VOID_FLAG = 'N'
AND C.EFFECTIVE_DATE BETWEEN '0001-01-01'
AND :WS-CLAIM-DATE
AND C.END_DATE BETWEEN :WS-CLAIM-DATE
AND '9999-12-31'
AND C.FK_PROV_ID = A.FK_PROV_ID
WITH UR;

```

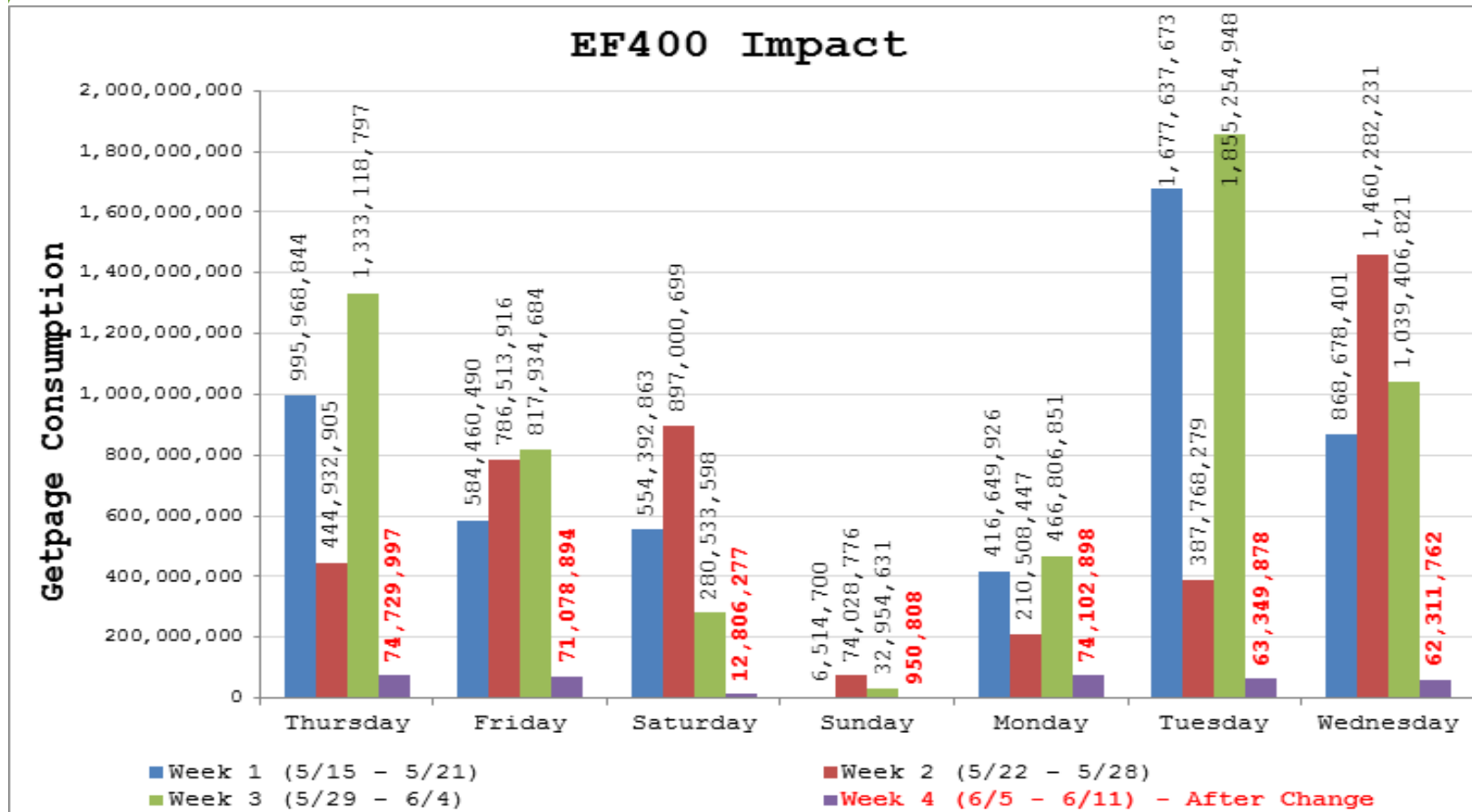
Access Path Analysis:

```

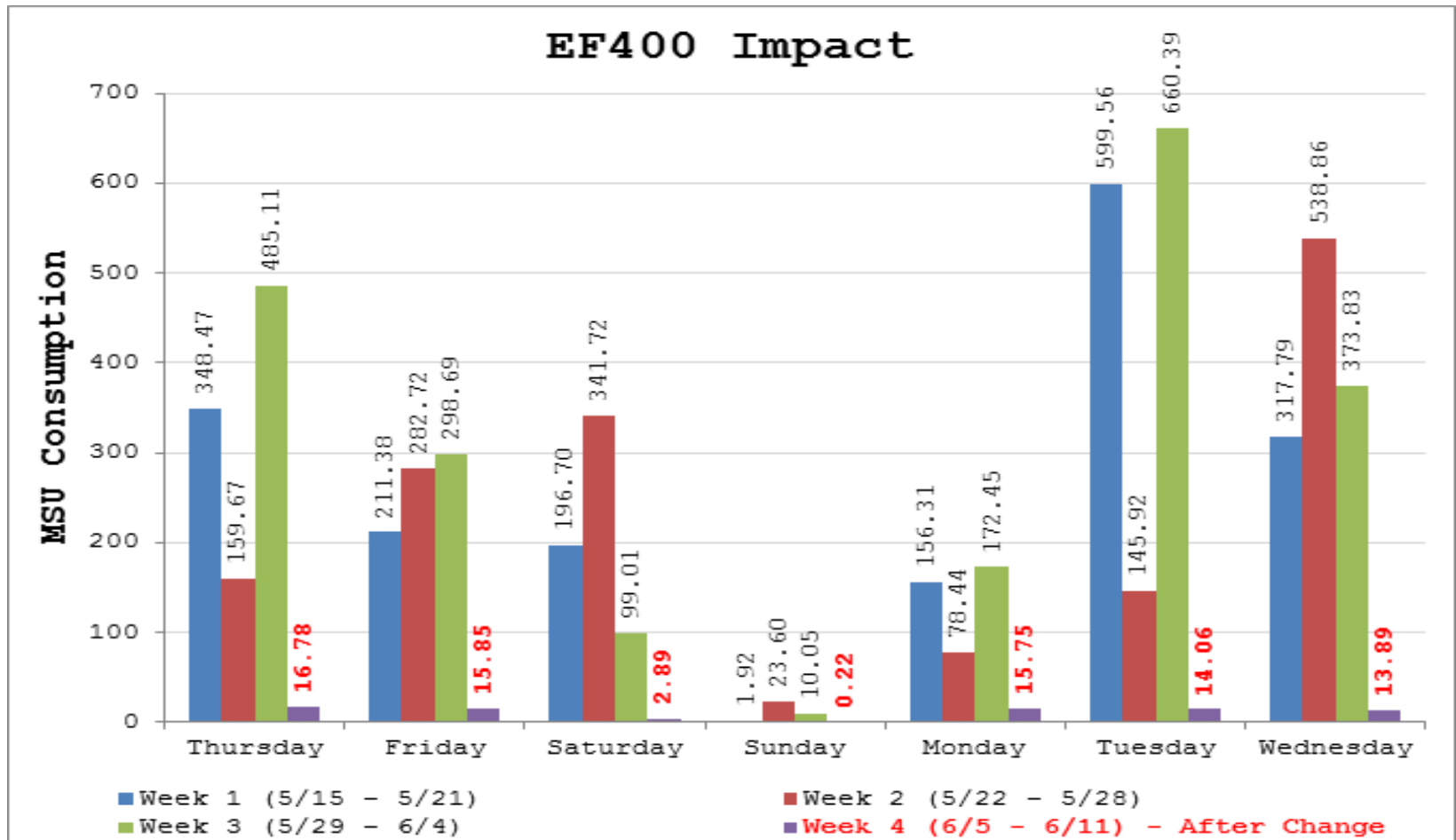
Cost: (ms) 1 (su) 9 (tc) +.20748 E+02
PQbkNo: 0 PPlnNo: 0 QblkNo: 1 PlanNo: 1 MxOpSq: 0

```

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.



GETPAGE = MONEY, and I am thru



Contact Information

Mark Youngs
Technical Architect

Louisville KY
myoungs@humana.com