

# Creative Application Tuning Techniques

Gary Cherlet  
gary@cherlet.com



CA IDMS™ Technical Conference

Framingham MA  
December 2-5, 2014

## Abstract

The purpose of this session is to introduce application designers and coders to a wide variety of tools and techniques which can be used to tune poorly performing applications. Using a case study approach, we will examine a number of common design problems, and our solutions. For each solution we will look at implementation techniques, and the effect that the solution had when applied to a live problem situation.

Specific code examples will be reviewed, and the results of the tuning exercise will be presented. This paper will be of particular interest to Analysts and programmers, but there will be much food for thought for DBAs and others who are interested in technical topics. The examples will be of particular interest to any sites who have implemented CAS modules.



CA IDMS™ Technical Conference



2

## Biography – Gary Cherlet

- Retired after 42 years in computing
- 30 years specializing in CA IDMS— pre/post sales support for Cullinet (four and a half years) and Allen Systems Group (one year in Singapore)
- Roles— DBA, application developer, security specialist, performance and tuning specialist, IDD customisation and reporting, IDMS/SQL functions and procedures
- Training in all aspects of using IDMS – DBA's and developers – wrote courseware for developers
- Where— Canada, Singapore, England, Malaysia, Thailand, Philippines and Australia

3



CA IDMS™ Technical Conference



## Agenda



### Intro to System Monitoring and Tuning

- **Monitoring, Tuning, Objectives**
- **Identifying Problem Areas**
- **Techniques Used**
  - **Use of the RETURN Verb**
  - **User-written CA ADS BIFs**
  - **CA ADS Arithmetic vs BIFs**
  - **Look Aside Message Buffer**
  - **Next Number Server Task**
  - **Database Procedures**
  - **Subschema Tailoring**
  - **Job Submission**
- **Results**
- **Conclusions**

4



CA IDMS™ Technical Conference

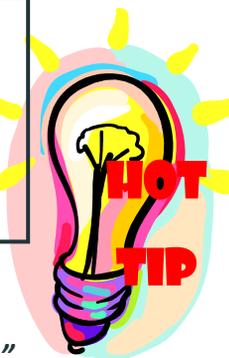


## Intro to System Monitoring + Tuning

- System Monitoring
  - Data Collection
  - Data Analysis
- System Tuning
  - *“If it ain’t broke - don’t fix it!”*.
  - The difference between:

*“being on the outside looking in”*  
and  
*“working from the inside out”*

Turn “stats  
collection  
OFF” in  
sysgen and  
reduce CPU  
by 18-20%



5



CA IDMS™ Technical Conference



## Application Tuning Objectives

- Reduce the number of deadlocks
- Reduce the number of I/Os
- Reduce execution time
- Minimise impact on applications

6



CA IDMS™ Technical Conference



## Agenda

- **Intro to System Monitoring and Tuning**
  - ✓ **Monitoring, Tuning, Objectives**
    - **Identifying Problem Areas**
  - **Techniques Used**
- **Use of the RETURN Verb**
- **User-written CA ADS BIFs**
- **CA ADS Arithmetic vs BIFs**
- **Look Aside Message Buffer**
- **Next Number Server Task**
- **Database Procedures**
- **Subschema Tailoring**
- **Job Submission**
- **Results**
- **Conclusions**

## Identifying the Problem Areas (1)

- **Deadlock Summary Report**
  - Name of Aborted Program
  - Name of program with which deadlock occurred
  - IDMS Status (store/modify/erase?)
  - DBKEY that caused the deadlock
- **Page level locks**
  - Too many / too large clusters
- **Specific DB keys**
  - OOAK or NOOAK problems

## Identifying the Problem Areas (2)

- Perf Mon Interval Monitor (PMIM)
  - Watch for DBKey WAIT time and high levels of IO activity at peak times
- Perf Mon Real Time Monitor (PMRM)
  - Ad-hoc observations – particularly bottlenecks
- Capacity Planning Reports
  - The Capacity Planner's "hit list" of rogue programs

9



CA IDMS™ Technical Conference



## Agenda

- ✓ **Intro to System Monitoring and Tuning**
  - ✓ **Monitoring, Tuning, Objectives**
  - ✓ **Identifying Problem Areas**

### **Techniques Used**

- **Use of the RETURN Verb** 
- **User-written CA ADS BIFs**
- **CA ADS Arithmetic vs BIFs**
- **Look Aside Message Buffer**
- **Next Number Server Task**
- **Database Procedures**
- **Subschema Tailoring**
- **Job Submission**
- **Results**
- **Conclusions**

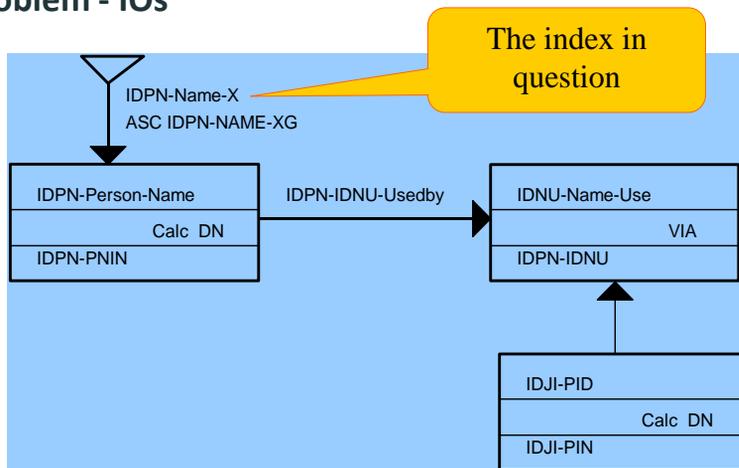
10



CA IDMS™ Technical Conference



## Using the RETURN Verb in Index Processing: Problem - IOs



11

## Using the RETURN Verb in Index Processing: Problem - IOs

- Original code used OBTAIN ... USING ... sort-key
- Original had RETRIEVAL LOCKS IS **YES**
  
- New code uses RETURN .. Using ... sort-key
- Set RETRIEVAL LOCKS IS **NO** for CPU savings as well as IO reduction

12

Using the RETURN Verb in Index Processing

Query	OLD				NEW			
	Ph/IO	P/Rqs	Locks	DBRqs	Ph/IO	P/Rqs	Locks	DBRqs
SMITH//	194	976	609	683	128	748	212	756
S/X/	634	1874	1258	622	43	786	126	764
S//	306	1418	795	845	191	1007	205	1013
SM//	235	1136	650	716	154	943	227	952
S//K	639	2031	1337	707	44	821	154	802
S/J/R	640	2302	1512	884	46	827	154	808
S//R	642	2436	1555	940	42	862	164	845
Sum:	3290	12173	7716	5397	648	5994	1242	5940
				Diff:	-2642	-6179	-6474	543
				% Diff:	-80.3	-50.76	-83.9	10.06

**DB requests up because we now have to issue OBTAIN as well as RETURN for “candidate” records – BUT look at the SAVINGS !**



## Agenda

- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

### Techniques Used

- ✓ Use of the RETURN Verb
  - User-written CA ADS BI
  - CA ADS Arithmetic vs BIFs
  - Look Aside Message Buffer
    - Results
    - Conclusions
- Next Number Server Task
- Database Procedures
- Subschema Tailoring
- Job Submission

User written CA ADS BIFs:

**Problem - CPU**

- Required text processing for user applications:
  - Remove Redundant Spaces
  - Capitalise the First Letter of Each Word
  - Word Capitalisation **and also** Remove Spaces
  - Remove Non-Alphabetic Characters

User written CA ADS BIFs:

**Problem - CPU**

**Removing Redundant Spaces (REMSPACE)**

This function returns the string that results when all occurrences of two or more SPACES in the specified source string are reduced to a single SPACE character in the result string.

Initial value (where **b** = space):

"O' HEARN**bbb**AND**bb**MCHEARN' S**bbbbbbbbbbbbbb**"

Returned string:

"O' HEARN**b**AND**b**MCHEARN' S**bbbbbbbbbbbbbb**"

User written CA ADS BIFs:

**Problem - CPU**

**Capitalise the First Letter of Each Word (JWORDCAP)**

This function returns the string that results when the first letter of each word in the specified source string is capitalised and all other characters in the string are converted to lower case. This is similar to the CA supplied WORDCAP, except that the following special cases are recognised:

d'Angelo (**D** remains lower case)

McAndrews (**A** also upper case)

Apostrophe's (the **S** remains lower case following apostrophe's at the end of words)

Initial value (where **b** = space):

"O' HEARNbbbANDbbbMCHEARN' Sbbbbbbbbbbbbbbb"

Returned string:

"O' HearnbbbAndbbbMcHearn' sbbbbbbbbbbbbbbb"

User written CA ADS BIFs:

**Problem - CPU**

**Word Capitalisation and Space Reduction (JWCAPREM)**

This function returns the string that results when the first letter of each word in the specified source string is capitalised and all other characters in the string are converted to lower case (JWORDCAP), and also reduces multiple spaces to a single space (REMSPACE). As before, the the following special cases are recognised:

d'Angelo (**D** remains lower case)

McAndrews (**A** also upper case)

Apostrophe's (**S** remains lower case after apostrophe's at the end of words)

Initial value (where **b** = space):

"O' HEARNbbbANDbbbMCHEARN' Sbbbbbbbbbbbbbbb"

Returned string:

"O' HearnbAndbMcHearn' sbbbbbbbbbbbbbbb"

User written CA ADS BIFs:  
**Problem – CPU**

```

DIALOG X
MOVE XX000-NAME-TEXT TO
GUT0025-STRING.

MOVE 'Y' TO
GUT0025-REMOVE-SPACES-FLAG.

MOVE 'N' TO
GUT0025-UPPER-LOWER-FLAG.

LINK 'GUT0025D'.

MOVE GUT0025-STRING TO
XX000-NAME-TEXT
    
```

**Old Technique**  
 Time / 1000 Iterations:  
13 secs

- ```

GUT0025D
Text processing
Xer by Xer scans
Nested BIFs
Heavy Subscripting
    
```

User written CA ADS BIFs:  
**Problem – CPU - Solution**

```

DIALOG X
MOVE XX000-NAME-TEXT TO
GUT0025-STRING.

LINK 'GUT0025D'.

MOVE GUT0025-STRING TO
XX000-NAME-TEXT
    
```

```

GUT0025D
MOVE
REMSPACE (GUT0025-STRING)
TO GUT0025-STRING.
    
```

Interim measure to get benefits wherever GUT0025D used

```

DIALOG X
MOVE REMSPACE (XX000-NAME-TEXT)
TO XX000-NAME-TEXT.
    
```

**New Technique**  
 Time / 1000 Iterations:  
 1 sec

## User written CA ADS BIFs: Problem - CPU

- User-written BIFs
  - Able to replace LINK to utility routine with the user-written BIF
  - Fewer GETSTG's/FREESTG's
  - Reduced system mode time
  - Reduced DC mode WAITS
  - Immediate (partial) benefit by using the new BIF in the LINKed to dialog
  - Full benefit by using BIF at the higher level

21



CA IDMS™ Technical Conference



## Agenda

- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- CA ADS Arithmetic vs BIFs 
- Look Aside Message Buffer
  - Results
  - Conclusions
- Next Number Server Task
- Database Procedures
- Subschema Tailoring
- Job Submission

22



CA IDMS™ Technical Conference



## CA ADS Arithmetic instead of CA ADS BIFs: Problem - CPU

- We had a date difference+offset calculation routine that was used “heaps” (Y2K compliant DATEDIF and DATEOFF not ready)
  - Converted CA ADS BIFs to Arithmetic
  - Converted Arithmetic to User Written BIFs
- Some Basics for arithmetic data items
  - COBOL design rules
  - Use signs
  - Use COMP or COMP-3 instead of DISPLAY
  - Use COMP for items used as subscripts

23



CA IDMS™ Technical Conference



## CA ADS Arithmetic instead of CA ADS BIFs: Problem - CPU

### Code Comparison

Let's not argue the relative merits of readability or maintainability – this code is in a “black box” routine – so once it works nobody has to look at it again. Here's just one line of a much larger routine (see appendix A of the written paper) !

#### Using a BIF - original code

```
COMPUTE GUT0016-WORK2-MM =
MOD (GUT0016-WORK2-MM + 9, 12) .
```

#### Without a BIF - intermediate code

```
COMPUTE GUT0016-TEMP-MM = GUT0016-WORK2-MM + 9.
DIVIDE 12 INTO GUT0016-TEMP-MM
GIVING GUT0016-TEMP-NUM
REMAINDER GUT0016-TEMP-MM.
```

24



CA IDMS™ Technical Conference



CA ADS Arithmetic instead of CA ADS BIFs:  
**Problem - CPU**

| 2k Iterations   | Elapse | SYS    | USER   | Tot CPU | DC Rqs    | Get Stg | Free Stg |
|-----------------|--------|--------|--------|---------|-----------|---------|----------|
| Old GUT0016     | 32.57  | 13.8   | 16.33  | 30.13   | 86033     | 34026   | 34023    |
| Tmp GUT0016     | 25.3   | 7.76   | 13.59  | 21.35   | 46033     | 14026   | 14023    |
| Difference %    | 22.321 | 43.768 | 16.779 | 29.14   | 46.49379  | 58.779  | 58.784   |
| Saving for 100k |        | 302    | 137    | 439     | 2,000,000 |         |          |
| CPU Mins Saved  |        |        |        | 7.3167  |           |         |          |

**Anticipated Savings**

At this stage we have an “intermediate” version of our date handling routine – we thought if we can save 7 minutes on 100,000 uses by changing to basic arithmetic – can we do better yet by writing built-in-functions?

CA ADS Arithmetic instead of CA ADS BIFs:  
**Problem - CPU**

Using a User Written BIF - final code

```

MOVE GUT0016-WORK1-DATE TO GUT0016-WORK2-DATE.
MOVE DBTOSERL(GUT0016-WORK1-DATE) TO GUT0016-WORK2.
MOVE GUT0016-NUM-AREA TO GUT0016-DATE-AS-NUM.
CALL NUMBTOD8.
RETURN.
DEFINE NUMBTOD8.
MOVE SERLTODB(GUT0016-DATE-AS-NUM) TO GUT0016-WORK2.
MOVE GUT0016-DATE-AREA TO GUT0016-WORK2-DATE.
GOBACK.
    
```

Please see Appendix A of written paper for the full code + intermediate code replaced by this!

## CA ADS Arithmetic instead of CA ADS BIFs: Problem - CPU

| 2k Iterations   | Elapse | SYS    | USER   | Tot CPU | DC Rqs    | Get Stg | Free Stg |
|-----------------|--------|--------|--------|---------|-----------|---------|----------|
| Old GUT0016     | 32.57  | 13.8   | 16.33  | 30.13   | 86033     | 34026   | 34023    |
| New GUT0016     | 16     | 7      | 6.37   | 13.37   | 42033     | 12026   | 12023    |
| Difference %    | 50.875 | 49.275 | 60.992 | 55.626  | 51.14317  | 64.656  | 64.662   |
| Saving for 100k |        | 340    | 498    | 838     | 2,200,000 |         |          |
| CPU Mins Saved  |        |        |        | 13.967  |           |         |          |

### Anticipated Savings

With a built-in-function we look like saving 14 minutes of CPU on 100,000 uses of this routine (our volumes are now much higher than this and 100,000 was a conservative estimate anyway) – so we're pretty happy that we bit the bullet and did this way back when!

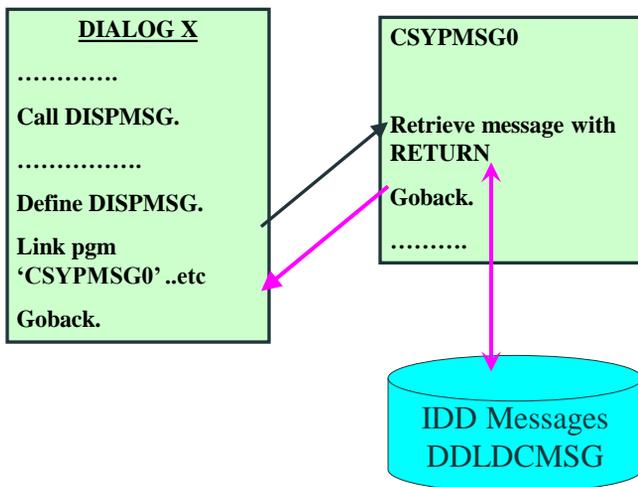
## Agenda

- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

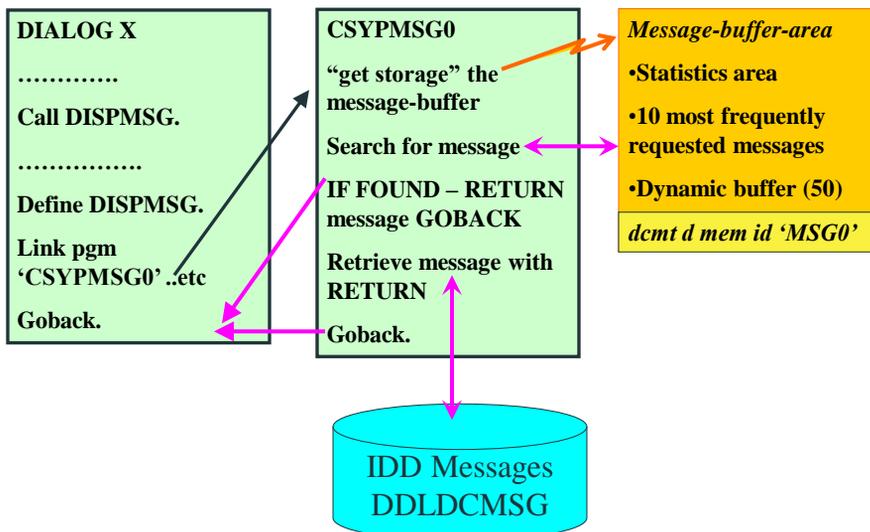
### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- ✓ CA ADS Arithmetic vs BIFs
- Look Aside Message Buffer
- Next Number Server Task
- Database Procedures
- Subschema Tailoring
- Job Submission
- Results
- Conclusions

### Look Aside Message Buffers: Message Processor – Problem - High IO on DDLDCMSG



### Look Aside Message Buffers: Revised Message Processor – Problem – IOs - fixed



### Agenda

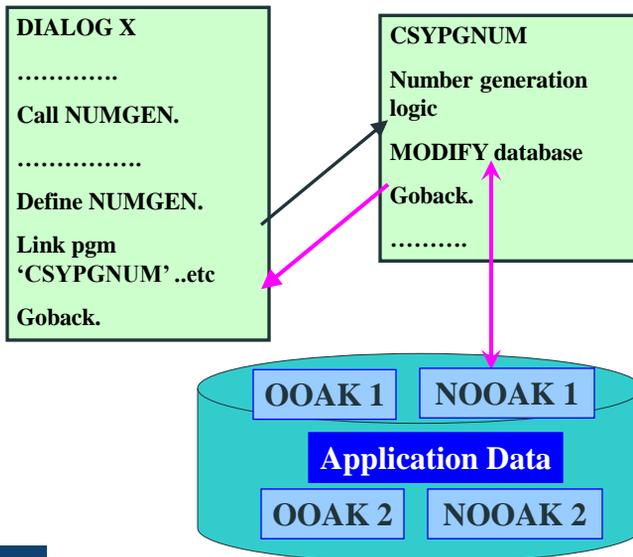
- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

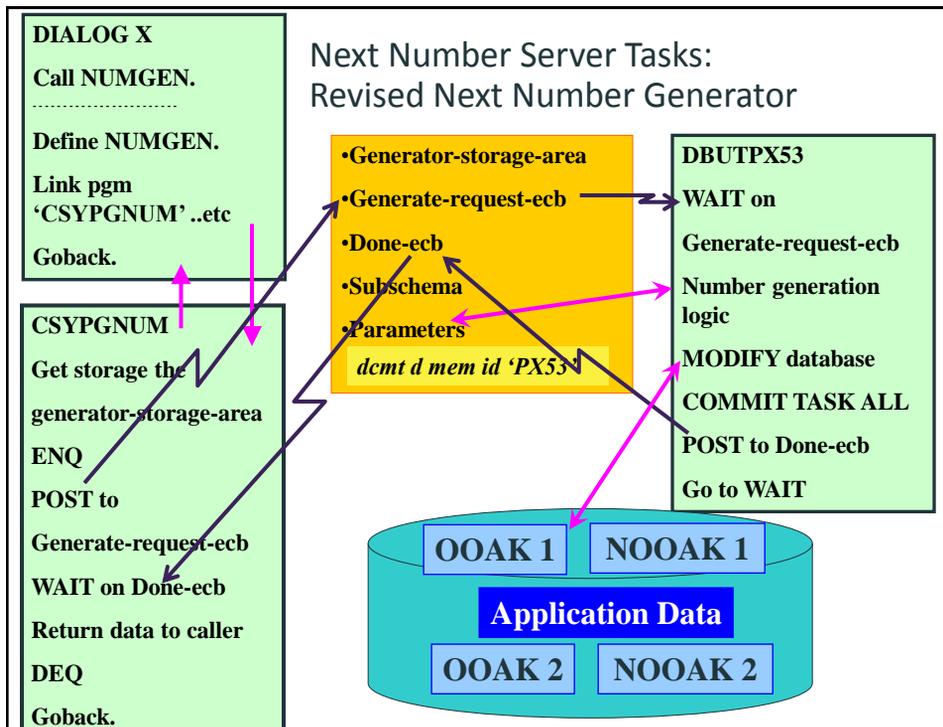
### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- ✓ CA ADS Arithmetic vs BIFs
- ✓ Look Aside Message Buffer
- Results
  - Conclusions
- Next Number Server Task
- Database Procedures
- Subschema Tailoring
- Job Submission

### Next Number Server tasks:

### Next Number Generator – Problem – DBKey Deadlocks





## Agenda

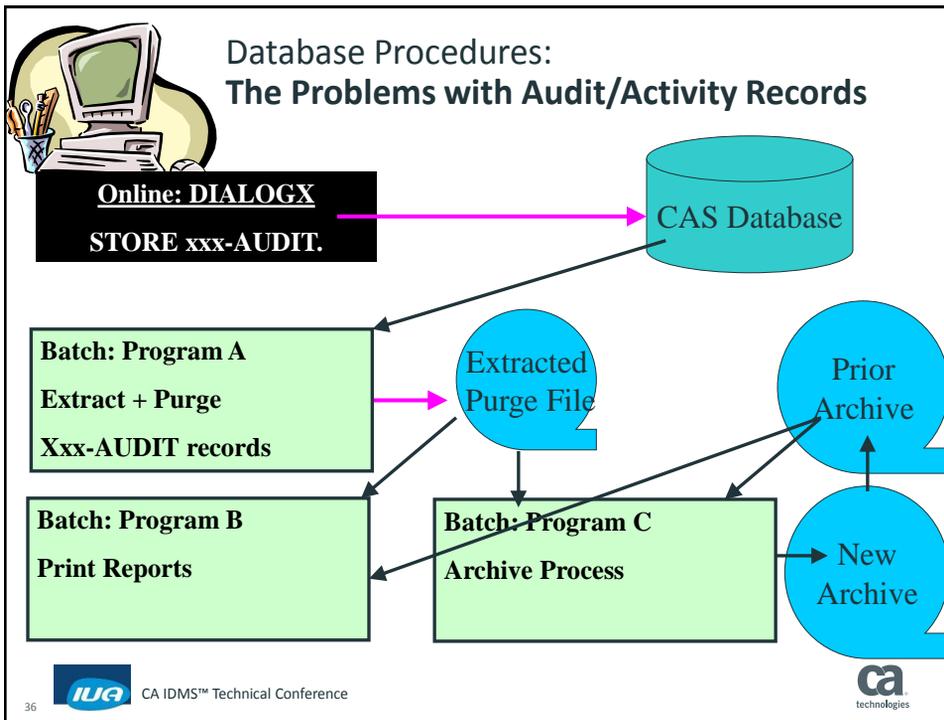
- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

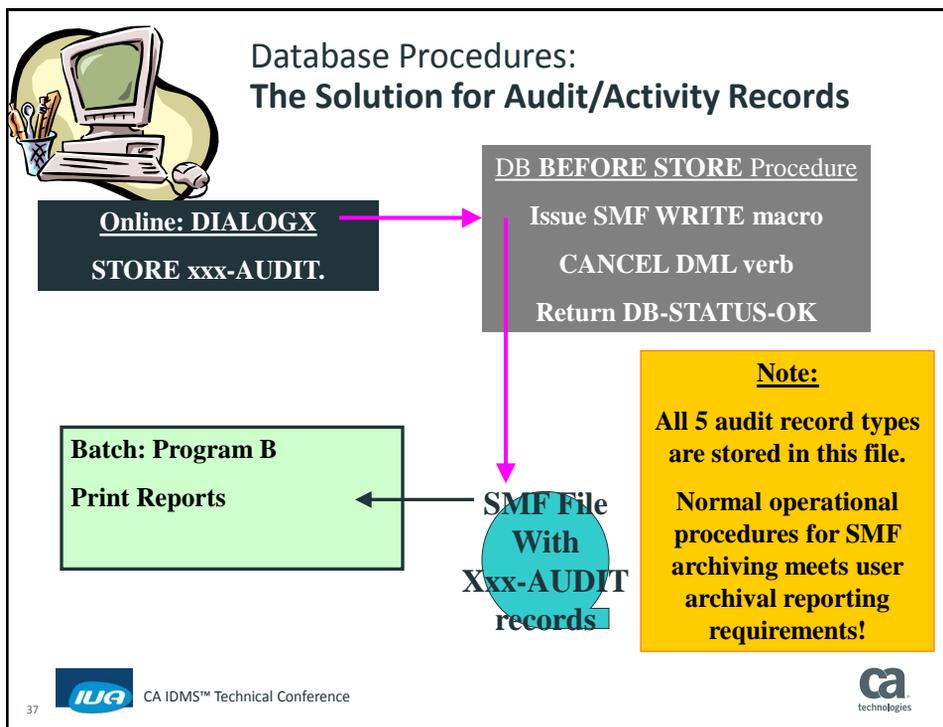
### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- ✓ CA ADS Arithmetic vs BIFs
- ✓ Look Aside Message Buffer
- Results
  - Conclusions
- ✓ Next Number Server Task
  - Database Procedures
  - Subschema Tailoring
  - Job Submission

### Database Procedures: The Problems with Audit/Activity Records

- Direct Dbkey of -1 to create a “sequential file” from IDMS
- Each STORE changes space available so creates page level locks
- Concurrent tasks competing for the same page leads to -
- Heavy deadlocking of online transactions
- Monthly processing to clear out the areas – and creation of 5 archive files
- Records are *never* looked at online





## Database Procedures: Results

- Reduced number of deadlocks (30%)
- Reduced I/O (20%)
- Reduced internal response time (20%)
- Reduced journal I/Os
- Replaced 5 monthly update jobs with one non-database job
- CAS-CPRAUD-AREA dropped out of sight on WRITE activity
- CAS-CAPAUD-AREA virtually disappeared from PMIM monitoring
- NO application code affected

## Agenda

- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- ✓ CA ADS Arithmetic vs BIFs
- ✓ Look Aside Message Buffer
  - Results
  - Conclusions
- ✓ Next Number Server Task
- ✓ Database Procedures
  - Subschema Tailoring 
  - Job Submission

## Subschema Tailoring: Problem – CPU

- Why large subschemas are a performance issue in CA ADS?
  - Currency save/restore processing
    - Very CPU intensive
    - It's a logical merge from the top level down – not just a straight copy of the currency tables
  - Cost goes up with an increase in the number of levels
  - Cost goes up with increasing size of subschema(s)

Subschema Tailoring:  
**Problem – CPU – A true Story**

- Problem
  - Heavy user of Logical Record Facility
  - Not a problem by itself – BUT
  - As LR’s needed additional components they were simply added to the subschema
  - End results
    - Very large subschemas
    - Application VERY expensive to run
- Solution
  - Had user tailor the subschemas to meet the needs of logical groups of programs
- Result – 25% cost reduction for the application

Subschema Tailoring:  
**Problem – CPU – “test harness” benchmark**

| Test Case          | Sys Mode | User Mode | Tot Time | Getstgs | Progs Called |
|--------------------|----------|-----------|----------|---------|--------------|
| Tailored subschema | 1.61     | 0.5407    | 2.1507   | 14023   | 8005         |
| Global subschema   | 16.41    | 0.6714    | 17.0814  | 14023   | 8005         |
| Difference         | 919.25   | 24.17     | 694.23   |         |              |

A 700% Reduction in CPU – and –  
 not a single line of application code was changed!

## Agenda

- ✓ Intro to System Monitoring and Tuning
  - ✓ Monitoring, Tuning, Objectives
  - ✓ Identifying Problem Areas

### Techniques Used

- ✓ Use of the RETURN Verb
- ✓ User-written CA ADS BIFs
- ✓ CA ADS Arithmetic vs BIFs
- ✓ Look Aside Message Buffer
  - Results
  - Conclusions
- ✓ Next Number Server Task
- ✓ Database Procedures
- ✓ Subschema Tailoring
- Job Submission 

## Job Submission: Problem – DB Key Deadlocking - Symptoms

- 4,000+ batch job submissions /day
- Heavy DB Key WAITS
- Frequent **nn29** abends

## Job Submission:

### Problem – DB Key Deadlocking - Why

- Job Submission with WRITE PRINTER
  - JCL images “printed” to JES Internal Reader
  - ‘/\*EOF’ and WRITE PRINTER .... ENDRPT to finish submission
- What causes the problems?
  - JCL goes into Queue area – with DB key locking and journalling of before+after images (STOREs)
  - JCL comes out of the Queue area when IDMS-DC “prints” the report to the INTRDR – with locking and journalling of before+after images (ERASEs)

45



CA IDMS™ Technical Conference



## Job Submission:

### Problem – DB Key Deadlocking - Solution

- Job Submission without Queue
- How?
  - CA Spool™ interface to send JCL images to JES via CA Spool
  - Some gotchas
- Results
  - Almost eliminated Queue area deadlocks
  - Stable and reliable
  - Sorry – no “benchmark” results on this one

46



CA IDMS™ Technical Conference



```

SYSK - EXTRA! Bundle for TCP/IP
File Edit View Tools Session Options Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      JIS.DEV.UTILONL.SRCLIB(GUTP0990) - 01.10      Columns 00001 00072
Command ==>                                     Scroll ==> CSR_
002016      650-WRITE-TO-CASPOOL SECTION_
002017      *****
002018      *** WRITE JCL TO THE CASPOOL INTERNAL READER
002019      *****
002020      650-BEGIN_
002021
002022      if  ESF-CLOSED
002023          PERFORM 660-OPEN-CASPOOL
002024      end-if
002025      -
002026      TRANSFER 'ESFWRITE' RETURN USING WRITE-PARM_
002027      IF WRITE-STATUS NOT = SPACES
002028          move  99          TO L-JCL-PARAM-ERROR-CODE
002029          MOVE WRITE-ERROR TO L-JCL-ERROR-MESSAGE
002030          perform 670-CLOSE-CASPOOL
002031          DC RETURN
002032      END-IF_
002033
002034      650-EXIT_
002035      EXIT_
-----
Page004 14/20
Connected to host 143.216.234.46 9:00 AM

```

Easy to use - this is what the API for CA Spool looks like!

```

SYSK - EXTRA! Bundle for TCP/IP
File Edit View Tools Session Options Help
-----
File Edit Confirm Menu Utilities Compilers T
-----
EDIT      JIS.DEV.UTILONL.SRCLIB(GUTP0990) - 01.10      Columns 00001 00072
Command ==>                                     Scroll ==> CSR_
000739
000740      01 WRITE-PARM_
000741          03 WRITE-STATUS          PIC XX VALUE SPACES_
000742          03 WRITE-RESV1          PIC X(2) VALUE LOW-VALUES_
000743          03 WRITE-CBA           PIC X(4) VALUE LOW-VALUES_
000744          03 WRITE-FILNO        PIC X(4) VALUE LOW-VALUES_
000745          03 WRITE-FLAG          PIC X(1) VALUE LOW-VALUES_
000746          03 WRITE-RESV2        PIC X(1) VALUE LOW-VALUES_
000747          03 WRITE-LRECL        PIC X(2) VALUE LOW-VALUES_
000748          03 WRITE-BUFPTR       PIC X(4) VALUE LOW-VALUES_
000749          03 WRITE-LINE         PIC X(80) VALUE SPACES_
000750          03 WRITE-PARM-END     PIC X      VALUE SPACES_
000751
000752      01 CLOSE-PARM_
000753          03 CLOSE-STATUS        PIC XX VALUE SPACES_
000754          03 CLOSE-RESV1        PIC S9(4) COMP VALUE +0_
000755          03 CLOSE-CBA         PIC X(4) VALUE LOW-VALUES_
000756          03 CLOSE-TYPE        PIC X VALUE '0'_
000757          03 CLOSE-RESV2        PIC X(3) VALUE LOW-VALUES_
000758          03 CLOSE-FILNO       PIC X(4) VALUE LOW-VALUES_
-----
Page004 06/19
Connected to host 143.216.234.46 9:03 AM

```

Clearly documented in the CA Spool documentation set

## Agenda

- ✓ **Intro to System Monitoring and Tuning**
    - ✓ **Monitoring, Tuning, Objectives**
    - ✓ **Identifying Problem Areas**
  - ✓ **Techniques Used**
  - ✓ **Use of the RETURN Verb**
  - ✓ **User-written CA ADS BIFs**
  - ✓ **CA ADS Arithmetic vs BIFs**
  - ✓ **Look Aside Message Buffer**
  - ✓ **Next Number Server Task**
  - ✓ **Database Procedures**
  - ✓ **Subschema Tailoring**
  - ✓ **Job Submission**
- **Results** 
  - **Conclusions**

49



CA IDMS™ Technical Conference



## Results: anticipated benefits in CAS

- Remove 1.1 I/O's for every dialog that issues a message, and
- Remove 2 to 2.n I/O's for every audit/activity record created, and
- Elimination of deadlocks and waits during number generation, and
- Elimination of deadlocks and waits due to creation of activity/audit records, and
- Improved response times due to all of the above

50



CA IDMS™ Technical Conference



## Results: actual achievements in CAS

- Reduced deadlocks by  $\geq 30\%$
- Eliminated 5 monthly update jobs
- Improved response times by 20% (in “hit list” dialogs)
- Reduced I/O activity by 20% (in “hit list” dialogs)
  - I/O profile excludes non-critical database areas (load area, message area, etc)
  - Only “business” areas show up
- DBAs able to cut Message run units from 4 to 2
- Only 4 dialogs had any code changes – a total of 8 lines of code were affected!

51



CA IDMS™ Technical Conference



## Agenda

- ✓ **Intro to System Monitoring and Tuning**
  - ✓ **Monitoring, Tuning, Objectives**
  - ✓ **Identifying Problem Areas**
- ✓ **Techniques Used**
- ✓ **Use of the RETURN Verb**
- ✓ **User-written CA ADS BIFs**
- ✓ **CA ADS Arithmetic vs BIFs**
- ✓ **Look Aside Message Buffer**
- ✓ **Next Number Server Task**
- ✓ **Database Procedures**
- ✓ **Subschema Tailoring**
- ✓ **Job Submission**
- ✓ **Results**
  - **Conclusions** 

52



CA IDMS™ Technical Conference



## Conclusions (1)

- Don't be afraid to **use the full functionality** of the products (e.g.):
  - Database procedures
  - User written BIFs for CA ADS
- Be sure to **follow traditional, good design and coding practices** for efficiency
  - What was good for COBOL is still good for CA ADS in many cases
- When all else fails
  - **Use assembler**

53



CA IDMS™ Technical Conference



## Conclusions (2)

- **Focus** your application tuning efforts on heavily used (application) utility programs
- **Avoid** changes that will affect many dialogs (requiring large migrations)
  - Ease of testing and validation
  - Makes for easier implementations
  - Backed out more easily if there's a problem
  - The value of focusing on utility routines

54



CA IDMS™ Technical Conference



### Conclusions (3)



55



CA IDMS™ Technical Conference



### Summary

- Introduced a variety of tools and techniques which were used to tune poorly performing applications
- Used a case study approach
- Discussed a number of common performance problems
- Examined design options
- Saw implementation techniques for solutions
- Reviewed specific code examples
- Looked at effects when applied to live problem situations
- Saw the results of a tuning exercise for CAS

56



CA IDMS™ Technical Conference



## Online Session Evaluation

Please provide your feedback about this session: A8

On the CA Communities web site  
<http://communities.ca.com>

[More details in your conference bag](#)

Questions  
and  
Answers  
and  
Discussion