

**IDMS System Level Tuning Issues: Current Status and Diary**  
**March 19<sup>th</sup>, 2007**

**Contents**

<b>1. MAX TASKS AND ERUS TASKS.....</b>	<b>4</b>
1.1 CONSIDERATIONS FOR MAX ERUS AND DEDICATED ERUS .....	4
1.2 FEEDBACK FROM CA – HOW MAX TASKS IS CALCULATED .....	4
1.3 CHANGED IN PRO – 2.JUL.1999 .....	4
1.4 CHANGED VARY MAX TASKS FROM 37 TO 38 – 20.AUG.1998 .....	4
1.5 CHANGED IN PRO – STILL VARIED TO 38 – WILL BE 40 – 2.SEP.1998 .....	5
1.6 MAX TASKS NOW VARIED TO 42 IN PRO – 16.NOV.1998 .....	5
1.7 CHANGED CHKUSER TASKS IN PRO - 11.OCT.1999 .....	5
<b>2. TIMER RELATED VALUES.....</b>	<b>6</b>
2.1 DISCUSSION .....	6
<b>3. SYSTEM RUN UNITS .....</b>	<b>7</b>
3.1 DISCUSSION .....	7
3.2 IDLE INTERVAL SYNTAX.....	7
3.3 CHANGE IMPLEMENTED 2.SEP.1998 .....	7
<b>4. XA RE-ENTRANT PROGRAM POOL .....</b>	<b>8</b>
4.1 DISCUSSION .....	8
4.2 CHANGED TO 32Mb - 8.JUN.1998 .....	8
4.3 CHANGED TO 35Mb - 19.OCT.1998 .....	8
4.4 CHANGED TO 45Mb – 6.AUG.1999.....	8
4.5 CHANGED TO 53Mb – 10.SEP.2000.....	8
4.6 CHANGED TO 63Mb – 4.FEB.2002.....	8
<b>5. STORAGE POOL SIZES .....</b>	<b>9</b>
5.1 DISCUSSION .....	9
5.2 CHANGED FROM 5Mb TO 9Mb - 14.SEP.1998 .....	9
5.3 ADDED POOL 129 - 28.SEP.1998 .....	9
5.4 POOL 128 FROM 14Mb TO 16Mb POOL 129 FROM 3Mb TO 4Mb - 19.OCT.1998.....	9
5.5 CHANGED STORAGE POOLS 128 AND 255 - 1.FEB.1999 .....	9
5.6 CHANGED STORAGE POOLS 255 (50 ADDITIONAL TERMINALS) – 15.FEB.1999.....	9
5.7 CHANGED STORAGE POOLS 255 (50 ADDITIONAL TERMINALS) – 23.JUN.1999.....	10
5.8 CHANGED STORAGE POOLS 255 (20 ADDITIONAL TERMINALS) – 23.JUL.1999 .....	10
5.9 CHANGED STORAGE POOLS 255 (50 ADDITIONAL TERMINALS) – 6.AUG.1999.....	10
5.10 CHANGED STORAGE POOL 128 – POOL GOING SOS – 8.JUN.2000 .....	10
5.11 CHANGED STORAGE POOL 0, 128, 255 – POOLS “TIGHT” WITH 430 USERS – 19.FEB.2002 .....	10
5.12 STORAGE POOL ADJUSTMENTS POST REL 16.0 – 13.MAR.2005-31.MAR.2005 .....	11
5.13 CURRENT STORAGE POOL ALLOCATIONS AND DISCUSSION FROM EDS.....	12
5.14 INCREASE MAX USERS ON VTAM LINE – 13.MAR.2007 .....	13
<b>6. SCRATCH POOL SIZES .....</b>	<b>13</b>
6.1 DISCUSSION .....	13
6.2 CHANGED SIZE - ?.JUN.1998.....	13
6.3 CHANGED SIZE FROM 4725 TO 5700 - ?.JUN.1998.....	13
6.4 CHANGED SIZE OF SCRATCH POOL - 17.FEB.1999 .....	13
6.5 CHANGED SIZE OF SCRATCH POOL - 2.FEB.2002 .....	13
<b>7. RCE/RLE/DPE COUONTS .....</b>	<b>14</b>
7.1 DISCUSSION .....	14
7.2 CHANGED RCE/RLE/DPE COUNTS – 29.JUL.2002.....	14
7.3 DISCUSSION AS OF REL 16.0 .....	15
<b>8. SYSTEM STATISTICS.....</b>	<b>15</b>

## JIS Mainframe Processing Management: Resource Management sub-Project

8.1	DISCUSSION .....	15
8.2	STATISTICS COLLECTION TURNED OFF – 27.AUG.1999 .....	15
8.3	STATISTICS COLLECTION- NOUSER WHEN TURNED ON – OCT.2001 .....	15
<b>9.</b>	<b>ADSO STATEMENT .....</b>	<b>16</b>
9.1	DISCUSSION .....	16
9.2	STATISTICS FROM PX52.....	17
9.3	PRIMARY RBB ALLOCATION CHANGED TO 24K EFFECTIVE DECEMBER 3 <sup>RD</sup> .....	17
<b>10.</b>	<b>SPLIT LOAD AREA ALONG SYSTEM/APPLICATION BOUNDARY.....</b>	<b>19</b>
10.1	DISCUSSION .....	19
10.2	IMPLEMENTED IN TRN 25.JUNE.1998.....	19
10.3	IMPLEMENTED IN PRO 27.JULY.1998.....	19
<b>11.</b>	<b>LOAD FIRST MIGRATION STRATEGY .....</b>	<b>20</b>
11.1	DISCUSSION .....	20
<b>12.</b>	<b>REMOVE PMIM FROM PRO ON SYSJ.....</b>	<b>20</b>
12.1	DISCUSSION .....	20
<b>13.</b>	<b>USE OF EXCEPTION RESPONSE COMMUNICATION PROTOCOL .....</b>	<b>21</b>
13.1	DISCUSSION .....	21
13.2	PROTOCOL SET TO EXCEPTION RESPONSE – 6.JUL.1999 .....	21
<b>14.</b>	<b>ADJUST RPL COUNT FOR LINE VTAM1 .....</b>	<b>21</b>
14.1	DISCUSSION .....	21
14.2	REDUCED FROM 100 TO 70 – 15.JUN.1999.....	21
<b>15.</b>	<b>ALLOCATE PERMANENT READ BUFFERS FOR ONLINE USERS.....</b>	<b>22</b>
15.1	TURN ON – 14.JUL.1998 .....	22
15.2	TURN OFF – 14.AUG.1998.....	22
15.3	RECONSIDER APPLYING AGAIN? .....	22
<b>16.</b>	<b>SIGNON RETENTION.....</b>	<b>23</b>
16.1	DISCUSSION .....	23
16.2	IMPLEMENTED PRO 23.JUN.1998.....	23
<b>17.</b>	<b>UNDEFINED PROGRAM COUNT .....</b>	<b>23</b>
17.1	DISCUSSION .....	23
17.2	IMPLEMENTED PRO 23.JUNE.1998.....	23
<b>18.</b>	<b>QUEUE JOURNAL BEFORE INSTEAD OF ALL.....</b>	<b>23</b>
18.1	DISCUSSION .....	23
<b>19.</b>	<b>USE MORE EFFICIENT COMPILATION OPTIONS IN DEV .....</b>	<b>24</b>
19.1	DISCUSSION .....	24
19.2	COBOL PRE-COMPILE IN RETRIEVAL IS NOW DEFAULT IN DEV .....	24
<b>20.</b>	<b>BUFFER TUNING .....</b>	<b>25</b>
20.1	DISCUSSION .....	25
20.2	INCREASED BUFFERS – 13.SEP.1999 .....	25
20.3	INCREASED BUFFERS – 20.SEP.2000.....	25
20.4	INCREASED BUFFERS – 25.SEP.2000.....	25
20.5	INCREASED BUFFERS – 4.OCT.2000 .....	25
20.6	INCREASED BUFFERS – 10.OCT.2000.....	26
20.7	DECREASED BUFFERS – 19.OCT.2000.....	26
20.8	DECREASED BUFFERS – 23.OCT.2000.....	26
20.9	DECREASED BUFFERS – NOV.2000 .....	27
20.10	SPECIAL BUFFER FOR “HARD HIT INDEXESS – APR.2002 .....	27
<b>21.</b>	<b>DC-COBOL TUNING.....</b>	<b>29</b>

## **JIS Mainframe Processing Management: Resource Management sub-Project**

21.1	DISCUSSION .....	29
21.2	DEFINE IGZ* MODULES TO IDMS-DC – 11.OCT.1999 .....	29
21.3	TAILORED COBPACKS TO IDMS-DC – 16.OCT.1999.....	29
<b>22.</b>	<b>SUPPRESS LOGGING OF SOME SECURITY MESSAGES TO THE IDMS-DC LOG AREA.....</b>	<b>30</b>
22.1	DISCUSSION .....	30
22.2	IMPLEMENTED – 30.JUL.1998 .....	30
<b>23.</b>	<b>SUPPRESS LOGGING OF DC091003 “REPORT PRINTED” MESSAGE .....</b>	<b>30</b>
23.1	DISCUSSION .....	30
23.2	IMPLEMENTED ALL ENVIRONMENTS – 5.JUN.1998.....	30
<b>24.</b>	<b>USE OF A “SINGLE SEGMENT” .....</b>	<b>31</b>
24.1	DISCUSSION .....	31
24.2	IMPLEMENTED IN PRO 30.JUNE.....	31
<b>25.</b>	<b>APPLY LS05738 - STORAGE VS CPU PTF .....</b>	<b>32</b>
25.1	DISCUSSION .....	32
25.2	IMPLEMENTED 14.SEP.1998.....	32
<b>26.</b>	<b>OFFICE/OFFICER IN DATASPACE 14.JAN.2003 .....</b>	<b>33</b>
26.1	DISCUSSION – DATASPACE ACCORDING TO CA .....	33
26.2	NOTE RE: STATS .....	33
26.3	IMPACT IN PRO .....	33
<b>27.</b>	<b>MP MODE – FEB.2004 – PLUS 12%.....</b>	<b>34</b>
27.1	NEW HARDWARE INSTALLED .....	34
27.2	CA-SPOOL AND MT.....	35
27.3	TUNING QUEUE DEPTH – MAR.2007 .....	35
<b>28.</b>	<b>RELEASE 16.0 INTO PRO – 13.MAR.2005.....</b>	<b>35</b>
<b>29.</b>	<b>USER BUILT-IN-FUNCTIONS AND MP MODE – JAN-AUG.2006.....</b>	<b>37</b>
29.1	USER BUILT IN FUNCTIONS BACKGROUND .....	37
29.2	MP MODE CHANGES .....	37
29.3	RESULTS .....	38
<b>30.</b>	<b>SYSTEM TRACE – JULY.2006 – ADDS APPROX 7% .....</b>	<b>38</b>
30.1	JULY.2006 .....	38
30.2	11.AUGUST.2006.....	38
<b>31.</b>	<b>DETERMINING HOW MANY TCP/IP BULK TERMINALS TO DEFINE.....</b>	<b>39</b>
31.1	RUNNING OUT OF BULK TERMINALS – FEB.2007 .....	39
31.2	DETERMINING NUMBER OF BULK TERMINALS – MAR.2007 .....	39
	<b>TUNING ISSUES STATUS AFTER RESOURCE MANAGEMENT .....</b>	<b>40</b>

# JIS Mainframe Processing Management: Resource Management sub-Project

## 1. Max Tasks and ERUS Tasks

Status:	<b>Changed 23.July.1998</b>
Scheduled for:	
Next Action:	
Result:	3.5% CPU reduction with other minor changes (jun.1998)

### 1.1 Considerations for Max ERUS and Dedicated ERUS

Needs some clarification. JIS to discuss with EDS and agree values. EDS to request information from CA re “missing” tasks in calculation below. Note that the DAMS External Region takes up 1 ERUS task. Is this the source of the DEDICATED ERUS = 1 spec? What impact will the many JANCO (PCIU960J) jobs have?

Vary to 35 at start up. Leaves 7 unused SD tasks + 6 non-dedicated ERUS tasks + 6 max tasks = 19 user tasks during day (in past 10-15 has been recommended value on other machines)

Don't see lots of concurrent batch jobs during the day - so main concern for MAX ERUS at 7 (because of DAMS only 6 available) - is degree of concurrency at night. **How do we find out how many batch CV jobs run at the same time?**

### 1.2 Feedback from CA – How Max Tasks is Calculated

The number of MAXTASKS calculated for your 12.0 CV is as follows:

\* MAXTASKS from SYSMO-170 record (SYSGEN record): 54

Number of MAXIMUM TASKS on system statement = 16

Number of MAXIMUM ERUS on system statement = 30

Number of LINE statements in sysgen = 6

Add +2 for MASTER task and DBRC task = 2

(Subtract one for each CONSOLE line defined in sysgen)

\* MAXTASKS from RHDCBLD1 at startup time: 19

Number of SDC control blocks allocated for RU Service Drivers = 12 - Note 1

Number of SDC control blocks for PERFMON DRIVERS = 2

Number of SDC control blocks for LOG SERVICE DRIVERS = 3

RHDCDEAD + RHDCPRNT = 2

Note 1: since we only allocate 5 RU Service Drivers it appears that there are 7 tasks left over for “user tasks”

### 1.3 Changed in PRO – 2.jul.1999

Varied MAX TASKS from 35 to 37 - system had already been at max 93 times when this was done. Trialling bigger numbers now that system is using less CPU/Txn - in preparation for Sysgen change on 22nd/23rd.

### 1.4 Changed Vary Max Tasks from 37 to 38 – 20.aug.1998

At request of Mark V. changed IDMSR.COMMON.SRCLIB(VARYSNA2) on SYSJ to change the max tasks figure up from 35 to 38 for next startup this can be varied dynam during the day anyway

## **JIS Mainframe Processing Management: Resource Management sub-Project**

### **1.5 Changed in PRO – still varied to 38 – will be 40 – 2.sep.1998**

Here's the finally agreed upon settings for PRO. These were trialled in RES where we ended up with 57 max tasks – because there are 7 more lines than in PRO.

<b>Count</b>	<b>Item</b>
8	Lines + Master + DBRC
19	Startup – RHDCBLD1
10	Max Erus – Change to Sysgen
3	Dedicated ERUS – Change to Sysgen – not included in calculation
13	Max Tasks – change to Sysgen
<b>50</b>	<b>Calculated Max Tasks</b>

When applied to PRO the startup job which varies MAX TASKS down to 38 should be changed to vary MAX TASKS to 40 (to allow for 2 additional DEDICATED ERUS).

### **1.6 Max Tasks now Varied to 42 in PRO – 16.nov.1998**

Altered max tasks as per JISS request (Gary C)

altered - D V AC TA MAX TASK 40 to 42 in PRO In line with 1st part of request completed on Friday.

### **1.7 Changed CHKUSER TASKS in PRO - 11.oct.1999**

MOD SYS 20 CHKUSER TASKS IS 17 to 10 .

## 2. Timer Related Values

Status:	
Scheduled for:	
Next Action:	
Result:	CPU neutral

### 2.1 Discussion

These changes are primarily a “limiting” mechanism rather than an absolute performance adjustment - they may help to catch looping programs sooner then they would be caught with current values - thus providing a cost saving to JIS.

This is fine for tasks running inside of IDMS - however the value for EXTERNAL WAIT actually needs to be increased to **1200**. This is because the mechanism that will be used to trap loops in batch programs will be an operating system tool which will put jobs that exceed their CPU limits for their job class into a “non-performance” status. While in that state the operator will be required to contact a designated Agency Help Desk or user - and then take an appropriate action - cancel the job or allow it to continue.

During this time the job will not be issuing any IDMS DML verbs - so with the current setting the job would be cancelled at the IDMS end after 10 minutes in the “non-performance” state, or after 5 minutes with the currently proposed value. Of course the batch job wouldn’t be aware of this until it resumed execution - at which time it would get an **nn69** status indicating it had lost its External Request Unit with the CV.

One of the reasons that the operating system “trap” is better than the IDMS trap is that the IDMS loop trap will only work if the loop does not include any DML verbs. It’s typical of batch IDMS programs that any loops will include DML verbs - which don’t re-establish area currencies properly (for example). Another reason for the preference for the operating system trap is that it will work for local IDMS and non-IDMS jobs as well as the CV jobs.

**Stop press:** EDS are having problems with putting jobs into non-performance group and restoring to original group. Not sure now whether this will be done manually (requiring 1200 setting) or whether job will be allowed to process while operator contacts Agency “on call” person (let’s go with the 300 originally suggested).

MOD SYS 20

```
      CHKUSER TASKS IS      10 was 17
INACTIVE INTERVAL IS      60 was 600
      INTERNAL WAIT IS     90 was 900
      RUNAWAY INTERVAL IS  30 was 90
      TICKER INTERVAL IS   1 was 2
MOD TASK ADS2
      INACTIVE INTERVAL IS 600 was SYSTEM
MOD TASK GUT0044T
      INACTIVE INTERVAL IS 180 was SYSTEM
```

## JIS Mainframe Processing Management: Resource Management sub-Project

### 3. System Run Units

Status:	<b>Implemented 3 Sept.1998</b>
Scheduled for:	
Next Action:	
<b>Result:</b>	3.5% CPU reduction – note done with 1.2 and 1.12

#### 3.1 Discussion

May have some performance benefits (shorten system control block chains) - will certainly tidy up Transaction details in PMRM.

#### 3.2 Idle Interval Syntax

More recent review by JIS has requested further information regarding syntax options for run units. TCC question to be raised for information on the availability of “idle interval” settings for System run units.

TCC answer to “idle interval” question was that such an interval could not be set for these run units. Current statistics are understood to indicate that, during busier times e.g. Mondays, the number of run units may not be excessive – apart from the one which isn’t used at all.

#### 3.3 Change Implemented 2.Sept.1998

Adjusted System Run Units as requested by JIS -

SIGNON from 5 to 2

SYSTEM/DEST from 5 to 0

SECURITY from 5 to 2.

To take effect 03:00 3 September (Thursday).

### 4. XA Re-entrant Program Pool

Status:	<b>Implemented 9.June.1998</b>
Scheduled for:	
Next Action:	
Result:	CPU Neutral

#### 4.1 Discussion

CPU and I/O saving by reducing amount of program loading. Even if no measurable change it should help to reduce WAIT time which will improve response time and help to reduce queue length (of currently executing tasks – length of DCE chain).

#### 4.2 Changed to 32Mb - 8.jun.1998

Was 21Mb – hope for CPU and I/O saving by reducing amount of program loading.

#### 4.3 Changed to 35Mb - 19.oct.1998

CCR# : SR6836

Increased XA REENT POOL up to 35 Mb from 32 Mb.

#### 4.4 Changed to 45Mb – 6.aug.1999

CCR# : 260372

MOD SYS 20 XA REENTRANT POOL IS 35000 to 45000.

#### 4.5 Changed to 53Mb – 10.sep.2000

MOD SYS 20 XA REENTRANT POOL IS 45000 to 53000.

#### 4.6 Changed to 63Mb – 4.feb.2002

MOD SYS 20 XA REENTRANT POOL IS 53000 to 63000.



### 5. Storage Pool Sizes

Status:	
Scheduled for:	
Next Action:	
Result:	CPU Neutral

#### 5.1 Discussion

#### 5.2 Changed from 5Mb to 9Mb - 14.sep.1998

Related to LS05738 PTF (the Storage for CPU trade-off – see Implemented 14.sep.1998) into Increased the XA Storage Pool (255) from 5Mb to 9Mb. This is expected to be far more than is required and will, if necessary, be tuned back at an appropriate time.

#### 5.3 Added Pool 129 - 28.sep.1998

CCR# : 17NNNN

At Gary Cherlet's request added storage pool 129 as follows: ADD XA STO POO 129 CUSHION 400 SIZE 4000 CONTAINS TYPES (DATABASE TERMINAL).

#### 5.4 Pool 128 from 14Mb to 16Mb Pool 129 from 3Mb to 4Mb - 19.oct.1998

CCR# : SR6836

Changed IDMSR.COMMON.SRCLIB(VARYSNA2) to include the line D V STO POO 129 CUSHION 0 since CA limits doing this in SYSGEN but 129 is meant to use pool 128 as buffer/cushion

SYSGEN

also changed XA STO POOL 128 to 14000 from 16000  
and 129 to 3000 from 4000  
increased XA REENT POOL up to 35 Mb from 32 Mb

#### 5.5 Changed Storage Pools 128 and 255 - 1.feb.1999

GEMS# : 70933

SOS in Production - ref gems for details

Gary agreed with us to make the following changes

MOD SYS 20 XA STORAGE POOL IS 10500. (from 9500)

MOD XA STO POO 128 SIZE IS 18000. (from 14000)

#### 5.6 Changed Storage Pools 255 (50 additional terminals) – 15.feb.1999

CCR# : TASK #36162

Increased XA Storage pool by 20% from 10500

## **JIS Mainframe Processing Management: Resource Management sub-Project**

MOD SYS 20 XA STORAGE POOL IS 12600.

Add additional 50 Lterm/Pterm pairs (301 - 350) - as requested by JISS

See also Changed size of Scratch Pool - 17.feb.1999.

### **5.7 Changed Storage Pools 255 (50 additional terminals) – 23.jun.1999**

In preparation for the PIMS Enhancement (Phase II) - SRB added an additional 50 LTERM/PTERM pairs - now totalling 400. Also increased the XA Storage Pool by 20% (from 12600): MOD SYS 20 XA STORAGE POOL IS 15100 .

### **5.8 Changed Storage Pools 255 (20 additional terminals) – 23.jul.1999**

Added an additional 20 LTERM/PTERM pairs - now totalling 420 (requested by JISS - Mark V as a result of all 400 terminals being utilised on Monday 19/07), also increased the XA Storage Pool from 15100 as follows: MOD SYS 20 XA STORAGE POOL IS 16000 .

### **5.9 Changed Storage Pools 255 (50 additional terminals) – 6.aug.1999**

CCR# : 260372

MOD SYS 20

XA STORAGE POOL IS 16000 to 20000.

MOD XA STO POO 128 SIZE IS 18000 to 22000 .

### **5.10 Changed Storage Pool 128 – Pool Going SOS – 8.jun.2000**

Claude had observed that since 13/14<sup>th</sup> April pool 128 had suddenly jumped from never going above an HWM of 80% to regularly hitting 88%+. Finally late in May it went to 98% and we got 000's of SOS's. Asked EDS to bump the size of the pool by about 10%ish.

From:

MOD XA STO POO 128 SIZE IS 22000 CUSHION IS 2200 .

To:

MOD XA STO POO 128 SIZE IS 25000 CUSHION IS 2500 .  
GENERATE.

### **5.11 Changed Storage Pool 0, 128, 255 – Pools “tight” with 430 users – 19.feb.2002**

See Sam's original message and our follow-up request for the increase.

-----Original Message-----

**From:** Cherlet, Gary (JTS) [mailto:Cherlet.Gary@saugov.sa.gov.au]  
**Sent:** Monday, 11 February 2002 12:07  
**To:** 'Mowbray, Sam'; Cherlet, Gary (JTS); Walker, Charles (JTS)  
**Cc:** Brennan, Simon R; Hansen, Ernst T; Best, Colin J; Leeflang, Alex; Vasileff, Mark (JTS); Fisk, Bob (JTS)  
**Subject:** RE: SYSJ PRO IDMS resource usage

Thanks Sam. We have been keeping an eye on the HWM's - and they are starting to look a bit dodgy.

Also - "times at max tasks" is 7300+ as of noon today (previous high this year was 2700) - leading me to suspect that task stats collect is "on" this week - which it does appear to be since we are getting a figure in the CPU bucket. We seem to hit max tasks more when task stats are being collected - and always use a bit more storage (ie. HWM is higher) when we sit on max tasks - it's to cope with those times when the system is a bit stressed that we need the extra resources - so that tasks can finish up without being abended due to shortages of storage pool space (for example).

## JIS Mainframe Processing Management: Resource Management sub-Project

When we hit 430 users storage pools 128, 255 and scratch all seem to be sailing a bit "too close to the wind". I note that pool 128 was within 6% of hitting the cushion today too. I would feel a lot more comfortable if storage pools 128, 255 and scratch were all increased by, just to pick a number, around 20%.

Cheers - Gary

This year to date:                      Maximum      Average  
<< OLE Object: Microsoft Excel Worksheet >>

-----Original Message-----

From: Mowbray, Sam [mailto:sam.mowbray@eds.com]  
Sent: Monday, 11 February 2002 11:42  
To: Gary Cherlet(JTD) (E-mail); Charles Walker (JTD) (E-mail)  
Cc: Brennan, Simon R; Hansen, Ernst T; Best, Colin J; Leeflang, Alex  
Subject: SYSJ PRO IDMS resource usage

Gary/Charles,

I note this morning in PRO we have hit our max of 430 users again. I also note that pool 255 is starting to get toward a high peak and Scratch is also starting to creep up again. Do you have any stats on how these are showing over time? Do we need to adjust these? I did note a few weeks ago that there were some abend stats messages indicating PRO had run out of scratch.

Sam

From:  
Mod sys 20 storage pool 5,000 XA storage pool 20,000..  
MOD XA STO POO 128 SIZE IS 36,000 CUSHION IS 2,500 .  
To:  
Mod sys 20 storage pool 6,000 XA storage pool 24,000..  
MOD XA STO POO 128 SIZE IS 40,000 CUSHION IS 6,000 .  
GENERATE.

### 5.12 Storage Pool Adjustments Post Rel 16.0 – 13.mar.2005-31.mar.2005

First couple of days that Rel 16.0 was in PRO we had storage pool 129 go to 100%. Eventually pinned it down to batch programs that called subroutines many times in succession – where the subroutines did BIND/FINISH sequence on every call. Turns out CV was holding onto the variable subschema block storage (VB50's) until the batch ERUS task ended. Saw thru PMRM at one time a retrieval only program with 7MB of storage. This was fixed by application of following APAR:

**TC03902 - Central Version may experience short on storage conditions in pools defined for DATABASE KEPT storage:**

Also around this time we bumped number of users to 540 max – and some of the Resource Control Area items were starting to look dodgy – so they were bumped to:

Values at 15.mar.2005			New values		
	Genned	HWM		Genned	HWM (after startup)
RLE	20625	18778	RLE	25781	1929
RCE	20000	18612	RCE	25000	1859
DPE	1700	1410	DPE	2125	881
Stack	1300	1179	Stack	1625	893

## JIS Mainframe Processing Management: Resource Management sub-Project

### 5.13 Current Storage Pool Allocations and Discussion from EDS

QO74381(SP2,SP3), QO74375(SP0,SP1) - Allow "CUSHION IS 0" in SYSGEN Syntax when defining storage pools. Without the apar, sysgen issues the message "DC301001 INVALID CUSHION SIZE" if 0 is specified, But the DCMT VARY STORAGE POOL nnn CUSHION 0 is allowed for XA and Non-XA storage pools.

Compare the following two storage pool definitions. The major difference is in the "Storage Cushion 0" defined for the XA storage pool 128, 129, 130 and 255 via the above apar. The question is whether this set up is better than the one (where the Storage Cushion for the XA pool 128, 129, 130, 266 is non-zero) in terms of storage utilization and system performance.

POOL	ADDRESS	SIZE	CUSHION	INUSE	HWM	TIMES SOS	PFIX	CONTAINS TYPES
0	0020D000	4000K	1000K	320K	340K	0	NO	SY
1	005F5000	1000K	100K	28K	108K	0	NO	US,UK
2	006EF000	1000K	100K	8K	8K	0	NO	SH,SK
3	007E9000	1000K	100K	0K	8K	0	NO	TR,DB
128	1B2FB000	40000K	0K	21012K	22480K	0	NO	US,UK
129	1DA0B000	4000K	0K	456K	460K	0	NO	SH,SK
130	1DDF3000	15000K	0K	5872K	6604K	0	NO	TR,DB
255	1EC99000	32000K	0K	21248K	21888K	0	NO	SY

POOL	ADDRESS	SIZE	CUSHION	INUSE	HWM	TIMES SOS	PFIX	CONTAINS TYPES
0	00111000	1200K	400K	244K	252K	0	NO	SY
1	0023D000	600K	92K	8K	80K	0	NO	US,UK
2	002D3000	40K	12K	12K	12K	0	NO	SH,SK
3	002DD000	500K	52K	0K	16K	0	NO	TR,DB
129	0B5D0000	500K	60K	0K	0K	0	NO	SH,SK
130	0B64D000	5500K	120K	564K	2288K	0	NO	US,UK
131	0BBAC000	2500K	252K	480K	988K	0	NO	TR,DB
255	0BE1D000	7000K	700K	1648K	2636K	0	NO	SY

Release 16.0 writes a new message, DC015007 if a short on storage condition occurs from one of two possibilities:

**Condition 0** A storage request has just been satisfied, but the total amount of free storage remaining in the pool is less than the storage cushion for that pool.

**Condition 1** A storage request has been issued. The DC system determined that this pool is eligible to satisfy the request, but that not enough contiguous storage is available in the pool to satisfy the request. Depending on the options specified on the GET STORAGE request and availability of storage in other eligible pools, one of four actions may have been taken:

- The task was abended
- The task has been put in a wait state until enough storage is available.
- A return code has been sent back to the requesting program.
- The request has been fulfilled from another pool.

When the amount of unused space in Storage Pool 0 is smaller than the storage cushion, the DC/UCF system is said to be short on storage. The system starts no new tasks and uses the cushion to satisfy storage requests from tasks that are already executing.

When the Storage Cushion in XA pool 129, 130, 131, 255 are used, an SOS condition is encountered, would it stop new task from starting as in the case of Storage Pool 0 going SOS. Since the non-XA pool 0, 1, 2 and 3 are defined for the same storage types as in XA pool 128, 129, 130 and 255, they are used as backup pools. In that case why would we want to allow SOS to occur for the XA pools by defining non-zero Storage Cushion. By defining "Storage Cushion 0" we would not get an SOS condition from XA pool 128, 129, 130 and 255, but might still get condition 1 of DC015007.

## **JIS Mainframe Processing Management: Resource Management sub-Project**

### **5.14 Increase Max Users on VTAM Line – 13.mar.2007**

Max Users increased to 660. There are 670 VTAM terminals generated – VTAM661 to VTAM670 are varied OFFline at startup – so the maximum number of users that can log on to IDMS is now 660, unless some additional terminals are brought online. Up until this change was made we were regularly hitting the 630 maximum – with no adverse impact on performance – and no stress occurring in Storage Pools. We brought on the additional allowable users without adjusting storage allocations – since the HWM's for all pools were well below 80% and this was only a 5% increase in Max Users. There have been no SOS's or other signs of stress in the Storage Pools.

## **6. Scratch Pool Sizes**

Status:	
Scheduled for:	
Next Action:	
Result:	CPU Neutral

### **6.1 Discussion**

Scratch is used by many of the system tools (IDD and sysgen compilers, pageable map sessions, etc) as well as by applications for user-written pageable maps, to pass variable length lists from one program to another – or between tasks and so forth. When there is insufficient scratch allocated applications start to abend.

There is an ongoing issue open with CA about a problem which JIS experiences where it appears that scratch is not cleaned up properly – and our utilisation will continue to grow even as the number of users decreases. See the “DBA Issues Diary” in the ApssSupport directory for more details on this. Basically this problem means that JIS can not run a 24\*7 shop until this is resolved.

### **6.2 Changed size - ?.jun.1998**

CCR# : 201919

PRO's Scratch down from 6300 pages to 4725 as requested by JISS.

### **6.3 Changed size from 4725 to 5700 - ?.jun.1998**

CCR# : 18818

Increased scratch from 4725 pages to 5700

### **6.4 Changed size of Scratch Pool - 17.feb.1999**

CCR# : TASK #36162

In PRO - increased scratch pool by approx 20% as requested by JISS.

Continuation of Changed Storage Pools 255 (50 additional terminals) – 15.feb.1999.

### **6.5 Changed size of Scratch Pool - 2.feb.2002**

CCR# : TASK #36162

In PRO - increased scratch pool by approx 20% as requested by JISS – from 7,000 to 9,000 pages. See also Storage Pool increase on same date for “back grounder e-mails”.

## 7. RCE/RLE/DPE Couonts

Status:	
Scheduled for:	
Next Action:	
<b>Result:</b>	CPU Neutral

### 7.1 Discussion

#### 7.2 Changed RCE/RLE/DPE Counts – 29.jul.2002

Got this message from Sam on July 22<sup>nd</sup>, 2002:

Gary,  
I notice we had our RCEs automatically adjusted on PRO on Monday at 14.57

DC010007 V20 T353501 THE NUMBER OF RCE'S HAS BEEN INCREASED TO 20025.  
D D MESS DC010007

```

,
MESSAGE ID      DC010007
LINE           1
SEVERITY        0
DESTINATION     LOG, OPER
O.S. DEST CODE  0
O.S. ROUTE CODE 0
MESSAGE TEXT    T&$0 THE NUMBER OF &01'S HAS BEEN INCREASED TO &02.
DEFINITION      DC010007 THE NUMBER OF <RESOURCE-TYPE> HAS BEEN INCREASED
                  TO <NEW-NUMBER>.
                  A shortage condition on a resource (RLE, RCE or DPE) has
                  occured. The corresponding resource pool has been dynamically
                  extended, up to the number specified in the message.
                  The DBA should use this message to update the corresponding
                  sysgen definition to avoid the overhead of using this facility
    
```

Then a week later I sent this message to EDS:

**From:** Cherlet, Gary (JTS) [mailto:Cherlet.Gary@saugov.sa.gov.au]  
**Sent:** Monday, 29 July 2002 12:22 PM  
**To:** Alex@EDS (E-mail)  
**Cc:** SAM@EDS (E-mail); Simon@EDS (E-mail); Vasileff, Mark (JTS)  
**Subject:** RCEs and RLEs

These are the Resource Management area figures at 12:15 today. Have had a user HWM of 450 and have put through 267k transactions. The users are really "going for broke" - and it's not even the first Monday of the month.

Cheers - Gary

	RCEs	RLEs	DPEs
Number			
Avail:	16020	16520	1873
In Use	84.83%	82.64%	38.12%
<b>HWM:</b>	<b>98.34%</b>	<b>96.18%</b>	65.72%

Later in the day we got "dynamically bumped" again – so Sam made the sysgen change.

RESOURCE MANAGEMENT			
	RCEs	RLEs	DPEs
Number			
Avail:	20020	20645	1873
In Use	62.61%	61.12%	42.76%
HWM:	72.09%	70.47%	61.45%
Threshold			
Times:	0	0	0
Now:	NO	NO	NO

### 7.3 Discussion as of Rel 16.0

Somewhere in the last couple of releases CA has changed IDMS-DC to create secondary allocations of RCE/RLE numbers if it runs out of the sysgen specified number. As with most situations it is always preferable to run with only a primary allocation, as there are overheads in using dynamically allocated secondary allocations. So while we still watch the HWM of these resources they are no longer as critical as they once were – that is when the CV used to fall over if you ran short.

## 8. System Statistics

Status:	Implemented in PRO 27.aug.1999
Scheduled for:	
Next Action:	
Result:	Reduced CPU by 18% from 27.aug.1999

### 8.1 Discussion

Saw a 20% CPU saving when turned off at CAA – decided to try at JIS also. First got approval from JIS Finance Committee – as this will impact the way in which chargeback will operate. There are papers written on this subject somewhere on the LAN.

### 8.2 Statistics Collection turned OFF – 27.aug.1999

At the request of Gary changed the sysgen system statement :

From :

STATISTICS INTERVAL OFF NOLINE TASK COLLECT USER NOTRANSACTION

To :

STATISTICS INTERVAL OFF NOLINE NOTASK

SMF stats won't be written unless reversed.

### 8.3 Statistics Collection- NOUSER when turned ON – oct.2001

Discovered that there is a 10% saving at JIS when we turn statistics ON - but when we specify NOUSER to suppress the braking out of USER mode and SYSTEM mode time as separate figures. Rather, the CV just gives the “total” cpu per task as “system mode”.

Unless doing tuning where we needed to know where the performance gains were taking place (ie system or application) – there is no need to break out USER mode time.

## 9. ADSO Statement

Status:	Implemented in PRO – unknown
Scheduled for:	
Next Action:	
Result:	

### 9.1 Discussion

In order to save or optimise storage there are a number of options in the ADSO statement – unfortunately each of these options can cost CPU.

ADD ADSO	
ACTIVITY LOG IS YES	Would like to have as NO in DEV/PED to reduce amount of work done during generates but require for AMI. Does not matter what setting is in PRO because we don't generate there.
DIAGNOSTIC SCREEN IS NO	We have our own diagnostic screen – talk to DBA's about how this is implemented.
FAST MODE THRESHOLD IS OFF	Prevents "relocatable" resources from going out to scratch during pseudo-converse – saves CPU and prevents single threading through SCRATCH.
PRIMARY POOL IS 8168	Use PX52 to help determine these values – output "snap shot" below shows the main items used to determine this – and next – setting. Note that no primaries are less than 50% used and only about half of 2ndary allocations are less than 50% – means this figure is pretty good.
SECONDARY POOL IS 8168	Same consideration as for PRIMARY – note in display below that about have of 2ndary buffers are less than 50% full. This is good as small storage requirements after pseudo-converse will be catered for in unused portion. Equally – the fact that about half are over 50% used means we're not really wasting storage. Could reduce number of GET STORAGE requests for ADS by making this a big number – say 12-20k – but would need to know average HWM for RBB's during task execution.
RESOURCES ARE FIXED	Related to FAST MODE THRESHOLD – don't want resources to be RELOCATABLE
RECORD COMPRESSION IS OFF	Chews up CPU if ON
DIALOG STATISTICS OFF	Only use in DEV
STORAGE MODE IS SGNSIZE	Chews up CPU if CALCULATED.



# JIS Mainframe Processing Management: Resource Management sub-Project

## 9.2 Statistics from PX52

The red, bold lines below are useful in determining ideal values for PRIMARY and SECONDARY buffer size in the ADSO statement. This output is the first two pages displayed by the PX52 task.

```
*****
*                               SYSTEM:  20                               *
*                               99/327   11:32:30                       *
*****
* Number of LTERMs examined                               509 *
* Number of USERS Signed On                               262 *
*****
* Number of RLEs Allocated to Users                       4,221 *
* 31 bit programs - not nucleus or driver                  807 *
* 24 bit programs - not nucleus or driver                 2,982 *
* Number of Storage Allocations                          4,221 *
*   24 bit Storage (non-XA)                                19 *
*   31 bit Storage (XA)                                  4,202 *
* Number of RBBs Allocated                                535 *
* Primary RBBs 261 *
* Primary RBBs less than 50% 0 *
* Secondary RBBs 274 *
* Secondary RBBs less than 50% 173 *
* Number of VDBs Allocated                                280 *
* Number of $CURCY Allocations                            150 *
* Number of RLTs Allocated                                0 *
* Number of BLLs Allocated                                0 *
*****
```

This output is the last page display by the PX52 task.

```
*****
*                               SYSTEM:  20                               *
*                               99/327   11:27:31                       *
*****
* Average RLEs per User - all types                        16 *
* Average RBBs per User 2 *
* Average RLTs per User                                    0 *
* Average VDBs per User                                    1 *
* Average $CURCYs per User                                  1 *
* Average Storage per user - all types                    31,472 *
* Average non-XA Storage per User                         35 *
* Average XA Storage per User                            31,436 *
* Average RBB Storage per User 16,888 *
* Average VDB Storage per User                            1,917 *
* Average $CURCY Storage per User                        2,155 *
* Average RLT Storage per User                            0 *
* Average BLL Storage per User                            0 *
* Average "other" Storage per User                       10,511 *
*****
```

## 9.3 Primary RBB allocation changed to 24k effective December 3<sup>rd</sup>

Have estimated a 3% saving - at current pricing and throughput that represents about \$60,000 per year. Here's how we got the number: (24\*1024=24576-12=24564)

PRIMARY POOL IS 24564	Used PX52 and also SHOWADSA to help determine these values – output “snap shot” below shows the main items used to determine this setting. Note that about 1/3 of the primaries are less than 50% used and most of the 2ndary allocations are less than 50%. This is good as it means that there is room for
-----------------------	--

## JIS Mainframe Processing Management: Resource Management sub-Project

	new records to be allocated in existing RBB's when the application becomes active again – and hopefully the system will not need to acquire a first or another 2ndary allocation.
--	---

Here is the PX52 output after the change:

```
*****
*                               SYSTEM:  20                               *
*                               02/346   16:03:54                       *
*****
* Number of LTERMs examined                               548 *
* Number of USERS Signed On - using ADS/O                364 *
*****
* Number of RLEs Allocated to ADS/O Users                5,277 *
* Number of Storage Allocations for ADS/O                5,277 *
*      24 bit Storage (non-XA)                            4 *
*      31 bit Storage (XA)                               5,273 *
* Number of RBBs Allocated                                393 *
* Primary RBBs                                           364 *
* Primary RBBs less than 50%                            114 *
* Secondary RBBs                                         29 *
* Secondary RBBs less than 50%                          27 *
* Number of VDBs Allocated                               394 *
* Number of $CURCY Allocations                           217 *
* Number of RLTs Allocated                               0 *
* Number of BLLs Allocated                               0 *
*****

*****
*                               SYSTEM:  20                               *
*                               02/346   16:03:54                       *
*****
* Average RLEs per ADS/O User - all types                 14 *
* Average RBBs per User                                1 *
* Average RLTs per User                                   0 *
* Average VDBs per User                                   1 *
* Average $CURCYs per User                                1 *
* Average Storage per ADS user - all types               45,336 *
* Average non-XA Storage per User                         50 *
* Average XA Storage per User                             45,285 *
* Average RBB Storage per User                          25,214 *
* Average VDB Storage per User                           2,144 *
* Average $CURCY Storage per User                        3,364 *
* Average RLT Storage per User                            0 *
* Average BLL Storage per User                            0 *
* Average "other" Storage per User                       14,666 *
*****
```

### 10. Split Load Area along System/Application Boundary

Status:	<b>Implemented PRO 27.July.1998</b>
Scheduled for:	
Next Action:	
Result:	CPU Neutral

#### 10.1 Discussion

Will eventually allow "load first" migrations to begin - there will be cost savings by eliminating 2 sets of Cobol compiles and 2 sets of ADS dialog generates.

#### 10.2 Implemented in TRN 25.June.1998

Implemented on schedule in TRN then PRO. Cleanup tasks remaining (removal of most UserIds from SYSDML) will wait until DAMS related modifications have been made to JAMSIDDC and DAMS Equaliser. These are progressing.

#### 10.3 Implemented in PRO 27.July.1998

CCR# : 203342 SR5998

As the final step in Task 5998 (split the system dictionary) I have deleted all "normal" users from SYSDICT. Those which remain include EDS users \*Z\*\*\*\*\*, but not the 5 FACS users WZ\*\*\*\*\*) and JDARCC, JDAVDN, a variety of JSC\*\*\*, JISDAM (!) and other "non-person" type users. Split completed and Users deleted.

### 11. Load First Migration Strategy

Status:	<b>Implemented in PRO October 1998 (??)</b>
Scheduled for:	
Next Action:	
Result:	????

#### 11.1 Discussion

“Load first” migrations will give cost savings by eliminating 2 sets of Cobol compiles and 2 sets of ADS dialog (map, subschema and so forth) generates, as well as 2 “explosions”.

### 12. Remove PMIM from PRO on SYSJ

Status:	Change in as scheduled 2.June - WR#6593
Scheduled for:	
Next Action:	
Result:	1.5% CPU reduction

#### 12.1 Discussion

Anticipate CPU savings.

### 13. Use of Exception Response Communication Protocol

Status:	<b>Implemented 7.July.1999</b>
Scheduled for:	
Next Action:	
Result:	CPU Neutral

#### 13.1 Discussion

Allows task to finish sooner than with DEFRESP. Hope for some reduction in CPU, although CA says it just helps response times – experience shows they are right that it doesn't affect CPU. Here's some notes:

The EXPRESP can be specified on LINE or TASK definitions. One CA paper mentions that EXPRESP is recommended for use on retrieval tasks and "serious consideration should be given to allowing update tasks to use exception response as well". JIS preference is to have EXPRESP specified at LINE level (simpler) and not at TASK, which is thought to be more for a COBOL-based environment where programs are called via task mechanism.

EDS is concerned about impact of this change on the complex network setup, particularly between SYSB and SYSJ. We would recommend a) clarifying impact on update tasks with CA and with EDS network technicians (EDS to request info) b) if change is to be made, trialing it in RES and then DEV. DEV will enable SAPOL to verify the SYSB communications link. on June 3<sup>rd</sup> EDS forwarded reply from CA. JIS reads this as indicating that EXPRESP should be used unless there is a site specific reason not to.

#### 13.2 Protocol set to Exception response – 6.jul.1999

MOD SYS 20.

MOD LINE VTAM1 PROTOCOL IS EXPRESP. GENERATE.

### 14. Adjust RPL Count for Line VTAM1

Status:	<b>Implemented 16.June.1999</b>
Scheduled for:	
Next Action:	EDS
Result:	0.5% CPU increase

#### 14.1 Discussion

Hoped for some CPU savings – not hopeful of much.

#### 14.2 Reduced from 100 to 70 – 15.jun.1999

CCR# : ----- WR#6836

Changed VTAM1 line's RPLs from 100 to 70 as part of the KickStart tuning process.

### 15. Allocate permanent Read Buffers for Online Users

Status:	<b>Implemented 14.July; Reversed out 14 August.</b>
Scheduled for:	
Next Action:	<b>Consider again – see if we get our something back</b>
Result:	3% CPU increase – had to back off – but when we did so we didn't get our 3% back – something “funny” happened in this period but don't know what

Implement in RES to determine the amount of permanent storage required per user. Create a new storage pool (below the line) which contains only terminal storage. Allocate to the size for the maximum number of users that can be supported by the CV.

On June 3<sup>rd</sup> EDS had no problems with the suggestion. Agree with new, separate storage pool to isolate TERM storage. Suggest trial in RES and DEV. Prefer to have same pool “population” in all environments - ie create pool 1 in each CV - although at an appropriate size for the CV's terminal population.

Implemented on RES 11.June. Issue (TCC #6270868) raised with CA as to why storage not being allocated in the new Storage Pool 1. CA response indicates that Terminal Storage is not yet recognised separately but that it is now able to be allocated above the line. Storage Pool 1 is simply wasting storage. Waiting for Gary to agree that it should go. - Agreed.

Implemented in DEV on 17.June and PED 24.June. – possibly causing minor side-effect relating to tasks defined as INPUT but which take no terminal input. The DBA's “O” task is one such. DBAs will correct definition of this task, and any others identified by them or the customer, in all environments.

#### **15.1 Turn ON – 14.jul.1998**

CCR# : SR#6836

AS part of KickStart tuning, have altered VTAM1 line in Pro to be PERMREADBUF as scheduled to be in effect as at startup 14/07/98.

#### **15.2 Turn OFF – 14.aug.1998**

CCR# : SR#6836

Changed VTAM1 from PERMREADBUF to NOPERMREADBUF in SYSGEN

#### **15.3 Reconsider applying again?**

### 16. Signon Retention

Status:	<b>Implemented PRO 23.June.1998</b>
Scheduled for:	
Next Action:	
<b>Result:</b>	CPU Neutral

#### 16.1 Discussion

Change “signon retention” value in SRTT to 10 minutes. Should have some CPU savings for batch jobs running under same “owner” user id. No problems. The documentation suggests that this impacts only ERUS signons, not online users, so the aborted online users signing back on won’t get any benefit.

#### 16.2 Implemented PRO 23.jun.1998

CPU neutral.

### 17. Undefined Program Count

Status:	<b>Implemented PRO 23.June.1998</b>
Scheduled for:	
Next Action:	
<b>Result:</b>	CPU Neutral

#### 17.1 Discussion

Uses storage spool space more efficiently - may save CPU when accessing dynamically defined programs (the majority at JIS). No problems. JIS to advise timing and grouping. Suggest isolating from other “memory-impacting” changes to see impact on CV’s region.

#### 17.2 Implemented PRO 23.June.1998

CPU neutral.

### 18. Queue Journal Before Instead of ALL

Status:	Dropped
Scheduled for:	n/a
Next Action:	n/a

#### 18.1 Discussion

JIS does retain persistent data in the DC-Queue Area which may need ROLL FORWARD.

### 19. Use More Efficient Compilation Options in DEV

Status:	
Scheduled for:	
Next Action:	
<b>Result:</b>	Not measurable

#### 19.1 Discussion

JIS to consider making these changes in DEV. Cobol compiles are controlled by Mig/Support team. Compile options for ADS requires a Sysgen change in ADSO statement.

In the end could not change ADS generate options permanently as change dates are required to drive the AMI explosion process. Cobol compile panels were eventually changed to default to RETRIEVAL mode for the pre-compile step – which now runs much more quickly and does less work in the CV side (updating the dictionary). There is an option on the compile panel to tell the job to update the dictionary – for AMI migration purposes.

#### 19.2 Cobol Pre-compile in RETRIEVAL is now default in DEV

Compiles now run faster (less CPU on both sides).



### 20. Buffer Tuning

Status:	
Scheduled for:	
Next Action:	
EDS	

#### 20.1 Discussion

DBD and EDS to consider separating the Address and ID areas into separate buffer pools - is it worth investigating use of *dataspaces* for these areas?

EDS to ask about dataspace. DBD and EDS to talk about buffer pools.

No problems. EDS attempting to clarify the impact of increasing memory/region on customer. Dataspace information also being sought.

#### 20.2 Increased Buffers – 13.sep.1999

CCR# : 7832

Added more buffer pages into IDMSDMCL

IDMSDMCL.P4820-BUFFER from 500 to 800

IDMSDMCL.P6356-BUFFER from 600 to 850

#### 20.3 Increased Buffers – 20.sep.2000

Added more buffer pages into IDMSDMCL

P4820-BUFFER from 800 to 1050

P6356-BUFFER from 850 to 1100

P11476-BUFFER from 600 to 850

P23476-BUFFER from 25 to 50 (dictionary load area only)

#### 20.4 Increased Buffers – 25.sep.2000

Added more buffer pages into IDMSDMCL

P4820-BUFFER from 1050 to 1300

P6356-BUFFER from 1100 to 1350

P11476-BUFFER from 850 to 1100

#### 20.5 Increased Buffers – 4.oct.2000

Added more buffer pages into IDMSDMCL

P4820-BUFFER from 1300 to 1550

P6356-BUFFER from 1350 to 1600

P11476-BUFFER from 1100 to 1350

By this stage we were showing about a 20% reduction IO's per task, for a 4% reduction in CPU per task.

## JIS Mainframe Processing Management: Resource Management sub-Project

### 20.6 Increased Buffers – 10.oct.2000

Added more buffer pages into IDMSDMCL

P4820-BUFFER from 1550 to 1800

P6356-BUFFER from 1600 to 1850

P11476-BUFFER from 1350 to 1600

Suspicious that this last increase had caused an increase in CPU with no further benefit in reducing IO's – so asked EDS to back off this last change so we can watch what CPU does – if CPU comes down again we know we've finally over allocated buffers – if it stays high it may indicate something else happening in the system.

### 20.7 Decreased Buffers – 19.oct.2000

Took away some buffer pages from IDMSDMCL – in an attempt to see at which point we can reduce IO's without causing CPU to increase back to previous level.

P4820-BUFFER from 1800 to 1600

P6356-BUFFER from 1850 to 1650

From: Mowbray, Sam A [mailto:sam.mowbray@eds.com]  
Sent: Wednesday, 18 October 2000 14:13  
To: 'Vasileff, Mark (JTD)'  
Cc: Leeflang, Alex A; Brennan, Simon; Hansen, Ernst; Best, Colin J; Crawford, David  
Subject: FW: Buffer Tuning

Mark,

I have dropped the vary buffer figures at next startup as follows :.

P23476-BUFFER left at 50 buffer pages (originally 25)

P11476-BUFFER left at +1000 for the moment since this seemed to show the greatest performance increase at this range

P6356-BUFFER dropped 200 rather than 250 to see if we can't eke a bit more out of the lowered figure without the CPU increase

P4820-BUFFER dropped 200 for the same reason above.

Can you take a look at this change with the stats and give us an indication as to how to proceed?

Sam

### Buffers as at 19.oct.2000

--- Data Buffer ---	Size	In-use	Max	Getstg
P2676-BUFFER	2676	125	125	OPSYS
P4276-BUFFER	4276	50	50	OPSYS
P4820-BUFFER	5724	1600	1600	OPSYS
P6356-BUFFER	7548	1650	1650	OPSYS
P11476-BUFFER	13680	1600	1600	OPSYS
P23476-BUFFER	23476	50	50	OPSYS

### 20.8 Decreased Buffers – 23.oct.2000

Due to performance probs with batch jobs have dropped buffers back (Sam Mowbray):

IDMSR.COMMON.CNTL(VARYSNA2)

D V BUFF P4820-BUFFER ADD PAGES 250

D V BUFF P4820-BUFFER INIT PAGES 1050

D V BUFF P4820-BUFFER MAX PAGES 1050

## JIS Mainframe Processing Management: Resource Management sub-Project

D V BUFF P4820-BUFFER CLOSE  
D V BUFF P6356-BUFFER ADD PAGES 250  
D V BUFF P6356-BUFFER INIT PAGES 1100  
D V BUFF P6356-BUFFER MAX PAGES 1100  
D V BUFF P6356-BUFFER CLOSE  
D V BUFF P11476-BUFFER ADD PAGES 250  
D V BUFF P11476-BUFFER INIT PAGES 850  
D V BUFF P11476-BUFFER MAX PAGES 850  
D V BUFF P11476-BUFFER CLOSE

### 20.9 Decreased Buffers – nov.2000

e-mail to Sam – 14.nov.2000

Sam - Mark and I have been looking at the "tuning diary" in conjunction with the spreadsheet - and to us it looks like our best overall performance was the period leading up to the 4th of October - we'd like to try dropping back to (around) the levels of that period - as noted below beside the current levels.

I don't know how you want to do this - perhaps commit the current levels into the DMCL but VARY the actual numbers down at start up - whatever. WRT to CPU overheads I wonder if there's some penalty for adding in buffer pages dynamically after start up - so that it might be better to over allocate initially and vary down (or is this what you've been doing)?

Anyway - current thinking is:

Buffer	Current	Suggested
P2676-BUFFER	2676	125
P4276-BUFFER	4276	50
P4820-BUFFER	5724	1550 <<<<- 1300
P6356-BUFFER	7548	1600 <<<<- 1300
P11476-BUFFER	13680	1350 <<<<- 1300 <<<<- left high because of indexes
P23476-BUFFER	23476	50

Cheers - Gary (and Mark)

### 20.10 Special Buffer for "Hard Hit Indexess – apr.2002

e-mail to DBA's and DBD 4.apr.2002

We had previously talked about splitting out some of the more active index areas into their own buffer pool - but never did anything about it because we had no hard evidence about which areas were being hit the hardest.

We now have a report that analyses the journal files to determine the (index) areas that are updated the most - assuming that this means that they are also inquired against a lot, and since it's update activity where extra buffers should give extra benefit, this report should help to make the call as to which buffers to map to the new pool.

Based on the attached report I'd suggest that we make the following changes to the DMCL buffer pools:

- 1) Create a new pool X13680-BUFFER Page Size = 13680 Number of pages = 250
- 2) If there is a concern about system paging - then take 100 pages from P11476 and P4820, and 50 pages from P6356
- 3) Map the following areas to the new buffer pool:

EI-EVTITMX3	PA-POLACTX1
HI-HLDITMX1	PD-PHYDESX2

## JIS Mainframe Processing Management: Resource Management sub-Project

```
HI-HLDITMX4      PE-POLEVTX3
IT-ITMTYPX4      PE-POLEVTX4
MO-MODOPIX2      SI-SUBEVTX1
```

Any comments or suggestions welcome - otherwise when can we go ahead and do this? In the "responsibility matrix" scheme of things - would EDS create the new buffer pool and DBD would do the area to buffer pool mapping?

Cheers - Gary

Suggestions implemented on night of 11.apr.2002. Made the following observations on the 12<sup>th</sup>:

We were getting a 20:1 found in buffer to pages read ratio for the new buffer (around noon this has gone down to 15:1), 10:1 for the old buffer, and around 7.0 +/- .5 for the system-wide ratio ! Next week we should start to see if this is significant enough to show up in the EXCP/Task ratio.

Buffer pools now look like this:

D BU							
--- Data Buffer ---	Size	In-use	Max	Getstg	Prf fetch-Min	Prefetch	
<b>X13680-BUFFER</b>	<b>13680</b>	<b>250</b>	<b>250</b>	<b>OPSYS</b>	<b>500</b>	<b>Not-Allowd</b>	
P2676-BUFFER	2676	125	125	OPSYS	500	Not-Allowd	
P4276-BUFFER	4276	50	50	OPSYS	500	Not-Allowd	
P4820-BUFFER	5724	1300	1300	OPSYS	500	Not-Allowd	
P6356-BUFFER	7548	1300	1300	OPSYS	500	Not-Allowd	
P11476-BUFFER	13680	1300	1300	OPSYS	500	Not-Allowd	
P23476-BUFFER	23476	50	50	OPSYS	500	Not-Allowd	
- Journal Buffer -	Size	# In-Use	Waits	DB	Ckpt		
JRNL-BUFFER	3860	6	0	0	58507		
		# of Recoveries		I/O's	in Buffer		
			12	10	21		
V20 Release 14.1 >>> JIS Production CV <<< Enter next task:							

### 21. DC-Cobol Tuning

Status:	
Scheduled for:	
Next Action:	
Result	

#### 21.1 Discussion

See separate paper(s) on Cobol tuning – basically there are a number of things that need to be done to ensure that CA's support for Cobol for MVS and LE is implemented for best performance.

#### 21.2 Define IGZ\* Modules to IDMS-DC – 11.oct.1999

CCR# : 8439 (Tuning / Gary) – no measurable impact on CPU.

Changes - re Cobol/LE performance as suggested by CA TCC#6841572

```
ADD PROG CEEBINIT CEEPIPI CEEPLPKA CEEEV005 IGZCPAC IGZCPCO IGZEPCL  
IGZEPLF IGZCFCC IGZCLNK IGZEINI IGZETRM - LANG ASS NOD NONO NOP  
REENTRANT .
```

#### 21.3 Tailored Cobpacks to IDMS-DC – 16.oct.1999

CAMS : 8450 ?

Saturday evening as part of System maint, and tuning change. Change due to come in next cycle  
Sun 17/10/99 at 6:00am:

Added - SYS3.CEE.IDMS.LINKLIB

Replaces - IDMSR.CEE.LINKLIB.R98

Introduced tailored Cobpacks – no measurable CPU reduction. Large reduction in program calls (to the Cobol run-time modules).

### 22. Suppress logging of some security messages to the IDMS-DC Log Area.

Status:	<b>July-August.1998</b>
Scheduled for:	
Next Action:	
Result:	CPU Neutral – log easier to find what you want

#### 22.1 Discussion

Currently many security related messages are output to both the log area and to the SMF “security log”. No use is made of the messages logged to the system log area, except during testing of the run-time security system in DEV (and PED). Messages involved include: “signon failed” before automatic retry is initiated, “password changed in dictionary” messages, and so forth. Only the “exceeded maximum attempts” message is worth logging to the DC Log area. Implementing this change may require modifications to: WSECDX71, GUT0001D, DAMS999D, and other run-time security programs.

See “Reducing Traffic on IDMS-DC Log Area” in the same directory.

#### 22.2 Implemented – 30.jul.1998

CCR# : SR8209

As part of reducing logging traffic to LOG I put in changes to DAMS999D/M and DAMS997D into IDMSR.COMMON.LOADLIB on SYSJ- changes affect signon to CV & password changes

### 23. Suppress logging of DC091003 “report printed” message

Status:	<b>Implemented PRO 5.June.1998</b>
Scheduled for:	
Next Action:	
Result:	CPU Neutral – easier to find what yiu want in the log

#### 23.1 Discussion

Clean up DC-Log to make it easier to find problem solving information.

#### 23.2 Implemented All Environments – 5.Jun.1998

CPU Neutral.

### 24. Use of A “Single Segment”

Status:	<b>Implemented 30.June.1998</b>
Scheduled for:	
Next Action:	
<b>Result:</b>	11.5% CPU reduction

#### 24.1 Discussion

Have automated procedure for collapsing 80+ segments into one - about ½ dozen steps – with manual checks in between each. Gary advises JIS happy to go with developing their own automatic code manipulator.

As of 25.june scheduled to be implemented in PRO on 30.June. JIS staff responsible for implementation, EDS DBA will be available if required. Result – 11.5% CPU reduction.

#### 24.2 Implemented in PRO 30.June

Single Segment implemented in PRO. Problem for FACS dealt with. Implementation into other environments eventually followed up on by DBD.

### 25. Apply LS05738 - Storage vs CPU PTF

Status:	<b>Implemented in PRO 13.Sep.1998</b>
Scheduled for:	
Next Action:	<b>Is this still in PRO?</b>
<b>Result:</b>	4.56% CPU reduction – this change by itself gave us a 2.26% reduction – but also included adding storage pool 129 to try to get full impact of the PTF (1.6%) and , tuning of storage pool sizes and transfer of excess storage pool allocations to XA Reentrant pool (0.7%)
<b>Performance Problem</b> <b>1.Feb.1999</b>	See item 6 below which describes a performance problem experienced on February 1 <sup>st</sup> , and changes to Storage Pool allocations made as a result! ST Pool 128 from 14Mb to 18Mb, pool 255 from 9.5Mb to 10.5Mb

#### 25.1 Discussion

We would like to have this PTF installed at CoB on Monday Sept 14th to take effect on Tuesday Sept 15th. As discussed, a copy of the storage manager load module without the PTF will be made so that the change can be backed out quickly if we start running into serious SOS problems.

In our testing we did not record any increase in the static amount of storage used by signed on users during a pseudo-converse, in ST POOL 128. We could not measure changes to the amount of dynamically acquired storage during task execution. Typically this runs to about 80-100Kb for a maximum of 20 concurrent user tasks - about 2Mb. Storage pool 128 which will be most affected is currently allocated at 16Mb and runs about 75% - so we have about 4Mb "spare" in the event that we (JIS Services) are wrong in this.

As an additional "safety net", if it is possible, we would like to add a further 2Mb to storage pool **255** at this time – as we did see an additional 10k per user allocated in that pool. If it is not required this 2Mb could be removed from the storage pool after monitoring for a week (we need to see the impact on a Monday which is our peak load day of the week).

#### 25.2 Implemented 14.sep.1998

Have installed LS05738 PTF (the Storage for CPU trade-off) into As requested by Gary Cherlet I have increased the XA Storage Pool (255) from 5Mb to 9Mb (see Storage Pool Sizes). This is expected to be far more than is required and will, if necessary, be tuned back at an appropriate time.



### 26. Office/Officer in Dataspace 14.jan.2003

#### 26.1 Discussion – dataspace according to CA

When IDMS needs a database page, it first looks in the IDMS buufer pool. If it's already there it will use it. If the page is not in the IDMS buffer pool then it will look in the dataspace. If it's in the dataspace then we move it to the IDMS buffer. If it's not there then we issue an EXCP to read the page into the dataspace and then move it to the IDMS buffer. Movement between dataspace and IDMS buffers is via machine instructions no EXCP is needed. If a database page needs to be written to the database, first the page is moved to the dataspace and then an EXCP issued to write the page to the dataset.

When DBIO gets a request (from DBMS) for a database page, it checks to see if it can find it in the buffer pool. If it can't, it verifies if the file is associated to a dataspace. If yes, it issues a cmd to retrieve it from the dataspace. If it is not found in the dataspace, then DBIO will request the I/O, and once it's read it, it will copy over to the dataspace.

#### 26.2 Note re: stats

-----Original Message-----

**From:** Cherlet, Gary (JTS) [<mailto:Cherlet.Gary@sau.gov.sa.gov.au>]  
**Sent:** Monday, 23 December 2002 11:47 AM  
**To:** 'Mowbray, Sam'; Cherlet, Gary (JTS); Leeflang, Alex; Fisk, Bob (JTS); Gadd, Robert (JTS); Camac, Ron (JTS); Walker, Charles (JTS)  
**Cc:** Best, Colin J; Warren, Debra E; Hansen, Ernst T; Brennan, Simon R; Vasileff, Mark (JTS); Camac, Ron (JTS)  
**Subject:** RE: DBA Issues - Dataspaces

After you set dataspace=yes I ran several OLQ queries: 4 area sweeps, 4 index sweeps. **Noticed that in DCMT D STAT BUFF the Phy-reads goes up by the same amount as Fnd-in-Cache (hmmm ?) - I think that's because IDMS counts as a physical read any page request that is not Fnd-in-Buf. Is this worth verifying with CA - I'll poke around the manuals and see if there is an answer there.**

[Response from CA on physical read counts](#)

You are correct, **the physical read statistic gives a count of the number of times a requested page is not found in the buffer pool. It is not the count of the number excp's actually issued.**

*Sam*

#### 26.3 Impact in PRO

As CA explains dataspace: when a page is requested the system looks in the buffer – if the required page is not there it adds 1 to “Phys reads” and then for files mapped to a dataspace looks in the dataspace – if not there it then reads the page into a buffer. “Found in Buffer” and “Found in cache” reflect the number of pages found in those areas of memory.

So – on that basis - I think that this is how we are meant to interpret these numbers. “Real Reads” has “Found in Cache” removed from “Phys Reads”. “Found in Buffer+Cache” is the sum of the two “Found in” columns. The ratio columns are the ratio of the “Found in” columns to “real Reads”. The end column on the right is the total effectiveness of cache plus buffers. A typical system-wide “buffer effectiveness ratio” is about 8-12 : 1 (our average is 7 : 1 ) – so **ratios of 444 : 1 for the data area and 2900 : 1 for the index area** is pretty impressive. **Tuesday a week ago the ratios were 6.2 : 1 for the index area and 3.5 : 1 for the data area.**

When looking at “Real Reads” you need to be aware that there are only 4,995 pages in Office/Officer data area and 270 pages in the index area. That there are more reads than pages is due to “rereads” owing to updates. The two found in cache numbers comes up to just over 506,000 – if we assume (and that's always dangerous) that is the number if IO's that were saved – we saved on average a bit over 1 IO per transaction

## JIS Mainframe Processing Management: Resource Management sub-Project

(on 440k+ transactions yesterday). We average 19 IO's per task – so for Office/Officer alone we're looking at an across the board reduction of a 5% reduction in IO's.

I'd dearly love to see what could be done for "ID". I wonder if there is any way of telling how much memory was actually used by the dataspace? The reason I ask is that depending on the number of pages actually read during the day is likely to only be a small percentage of the total number of pages for an area that is the size of "Identity" – that is 255,150 pages (at 5724 per page). So I'm wondering if there might be enough "real memory" on SYSJ to put ID into a dataspace.

Sam sent some numbers over yesterday part way through the day that would seem to indicate almost no impact on paging caused by Office/Officer (although that might have changed later in the day when more of the database had been accessed). I'm hoping we'll see some numbers for the whole day yesterday sometime today.

If I've misinterpreted the numbers – please let me know – I'll make the relevant corrections and re-circulate this note.

Cheers - Gary

Area	Found in Buffer	Phys Reads	Found in Cache	Phys Writes	Found in Buffer+Cache	Real Reads	F-I-B:Read Ratio	Cache:Read Ratio	Buffer+Cache to Read Ratio
JIS.OF-OFOFFCX1	280914	41168	41057	40	321971	111	2530.8	369.9	2900.6
JIS.OF-OFOFFC01	1768359	470128	465098	3009	2233457	5030	351.6	92.5	444.0

Note: "Phys Reads" includes "Found in Cache" - "Real reads" is (Phys Reads - Found in Cache)

### Prior to dataspace

JIS.OF-OFOFFCX1	243496	38987	0	55	243496	38987	6.2	0.0	6.2
JIS.OF-OFOFFC01	1685831	482363	0	2989	1685831	482363	3.5	0.0	3.5

## 27. MP Mode – Feb.2004 – plus 12%

MP Mode was activated to help EDS get by with the existing hardware. Because IDMS was unable to access two physical CPU's, its weighting had to be continually raised in order to give it sufficient MIPs to meet demand. This had dire consequences within the "real physical" machine. By being able to access MIPs from two physical CPUs – EDS was able to lower JIS's weighting but still meet peak demands. Turning MP Mode was benchmarked as costing 12% CPU.

The additional CPU costs ended up being shared by EDS/JTS – since JTS was only taking on the cost to help EDS to help JTS meet it's peak processing requirements (if memory serves it was an 8% EDS + 4% JTS split). When the new outsourcing contract is signed and the new (z/Series) hardware is installed it is hoped that multi-tasking can be turned off and return to uni-processing.

### 27.1 New Hardware Installed

Well – the contracts have been signed and the new hardware has been installed. While there have been tremendous improvements in throughput, and response times, due to faster DASD and CPU cycles – because of the total size of the physical machines and JIS's allocation of MIPS across 2 CPU's – multi-tasking will be a permanent feature of the landscape.

So – two tuning initiatives have been started: first is tuning of "queue depth", second is returning to CA-Spool version of "Online Job Submission" routine. A discussion of Queue Depth follows. The Job Submission change is to reduce deadlocking in the DC-Queue area when multi-tasking is in use.

## JIS Mainframe Processing Management: Resource Management sub-Project

### 27.2 CA-Spool and MT

To implement successfully (remember we had to back out in Feb.2004 when MT first went into PRO?) – we had to involve CA. While our “single threading” logic worked fine – the fact of the matter is that if an ESFOPNC is issued on one TCB and any of the subsequent ESFWrites for that “CA-Spool file number” are issued on another TCB CA-Spool closes the file prematurely – ending in a “partial job submission”. CA supplied a program that illustrated the use of CPU Affinity (internal CA) macros which forces our task to execute on a single TCB during the entire CA-Spool process. Everything has worked fine since this version of GUTP0990 went into PRO.

### 27.3 Tuning Queue Depth – Mar.2007

When MT Queue Depth is set at 3, there is a high wakeup count for subtask SUBT0002 averaging around 98% instead of more evenly spread across the different subtasks. The Maintask wakeup count averages 0.4% and SUBT0001 averages 1.4%. Looking at the figure, we make very little use of SUBT0001, and when SUBT0001 is woken up, it does 13-14 tasks and the Maintask does 16-19 tasks. It indicates that at times, we get quite long queues of tasks (and therefore delays in processing the work) in SUBT0001.

When MT Queue Depth is set at 2, there is a lower wakeup count for subtask SUBT0002 averaging around 95% and we get a more evenly spread across the different subtasks. The Maintask wakeup count averages 1.0% and SUBT0001 averages 3.6%. SUBT0001, when woken up, it does 8-10 tasks and the Maintask does 11-13 tasks. It indicates that we get shorter queues of tasks and therefore faster processing of the work in SUBT0001.

If we reduce the Queue Depth, we would wake up SUBT0001 more often, but there would be less work for it to do on each occasion. The result should be shorter queues on average, less waiting and better response times. We expect to see some slight increase in CPU usage with some slight benefit in improved task response time by reducing the Queue Depth from the current values of 3 to 2.

Another point of interest to look at is the total number of times the subtasks have been dispatched, compared to the actual number of tasks that have been processed. It tells us that for each processed task it causes how many subtasks to be dispatched, that is to execute a task, it involves dispatching the subtask how many times, dispatching DB mpmode subtask while processing the task's DB request and DC mpmode subtask when processing the task's DC requests. Each task will need to be dispatched many times to a subtask to complete its work.

With MT Queue Depth of 3 the number of subtasks dispatched to process each DC task averages around 50, while for MT Queue Depth of 2 it averages between 60-66 subtasks. This would partially explain why the CPU increases slightly for MT Queue Depth of 2, the additional CPU usage associated with the mode manager, RHDCMODE.

In summary, there is little to be gained with MT Queue Depth of 2. It is recommended that we stay with MT Queue Depth of 3

*Regards,  
Paul Mak*

## 28. Release 16.0 into PRO – 13.mar.2005

Had hoped for some CPU reduction owing to improvements in multi-tasking in Release 15.0. Remember we jumped from 14.1 to 16.0. At first did not appear to be anything in it – in fact may have cost a bit – but then we straightened up a problem with a batch program(s) that were generating lots of run units and eating up lots of storage (see Storage Pool 129 saga). Both fixed the program and applied a PTF to release the storage

## **JIS Mainframe Processing Management: Resource Management sub-Project**

and CPU figures dropped to be more in line with what they were under Rel 14.1. Certainly no saving has been measured at this stage.

### 29. User Built-in-functions and MP Mode – Jan-Aug.2006

#### 29.1 User Built in Functions Background

Removed all DML verbs for Run Unit Server - frees USERBIF5.

Added 2 new BIFs: SHODBNAME and SETDBNAME. Changed SORT to USERBIF5 instead of DBUT001P so that we can include it in ADSOMAIN for efficiency (some dialogs will use DBUT001P some USERBIF5 for some time to come). USERBIF2, USERBIF4 and USERBIF5 changes follow:

**USERBIF2** was updated to include two new built-in-functions (SETDBNAME and SHODBNAME).

**USERBIF4** had three functions related to "edited moves" for ADS/O. They are very old from the Motor Registration days - don't think they were ever completed and never worked correctly. They were currently "wasting" one of the precious "hooks" in ADSOMAIN for linking in user b-i-f routines for efficiency. As it turns out the DECODE function (previously implemented by DBUT008P and #LOADed dynamically at run-time by ADSOMAIN) is used nearly once for every task run on the system (if we have 850k DC tasks we seem to get around 850K references to DBUT008P - although many programs have no references to this routine and some have as many as 20+). DBUT008P has been copied to USERBIF4 and had appropriate entry points and such modified to fit into the USERBIFn naming requirement.

**USERBIF5** was completely deleted (removed all the "subschemaless access" b-i-f's") and was replaced by the single SORT b-i-f (previously implemented by DBUT001P and #LOADed dynamically at run-time by ADSOMAIN).

**USERBIF1** and **USERBIF3** are "unchanged" in this lot of maintenance.

All of these USERBIF changes have involved updating JISCEV51. It contains the "model XDE descriptors" for the built-in-functions (these are "expression descriptor elements" that name the b-i-f, how it is implemented, and what the syntax requirements are for it - number and type of data items, supported data transformations for different data types, result data type, etc). This module is referenced at ADS compile time only and is implemented by JTS in jis.&cv..utilonl.srclib/loadlib. The updated module is now in DEV and PED.

Whenever ADSOMAIN is re-linked it should always pick up the current load modules for all 5 USERBIFn routines from jis.PED.UTILONL.loadlib.

#### 29.2 MP Mode Changes

A series of changes have been scheduled in discussions with EDS in order to try to reduce the number of MPMODE Waits which appear to be impeding system throughput in Production (PRO on SYSJ). JTS has written a syntax generator for the sysgen changes and for DCMT commands to dynamically backout the change for the Asm/Cob change (defined below). CA has vetted the Exit 14 code change to run in a different MPMODE.

There are two changes:

- Asm/Cobol sysgen definition changed to MPMODE ANY
- IDMS System exit 14 has it's #START macro changed from MPMODE=DB to MPMODE=ANY ("Exit 14" in schedule)

Currently we have only subjective, anecdotal evidence that MPMODE waits are excessive - with no way of measuring the impact (if any) of the proposed changes).

To allow for the collection of some empirical data on MPMODE waits we will be requesting that PMIM be activated (in the #PMOPT macro for Perf Mon) along with the normal statistics collection. This will give us both a "before" baseline and an "after" basis for comparison - once the MPMODE changes have been made to PRO. As it turned out - doing a DCMT D MPMODE

## JIS Mainframe Processing Management: Resource Management sub-Project

gave us the figures we wanted to enter into a spreadsheet to determine frequency of usage of each MP Mode and number of MP Mode Waits.

### 29.3 Results

The last time we had a full week of IDMS statistics was for the week commencing 15 January 2006. A comparison of last week with the January stats shows some interesting comparisons:

- the total number of transactions for the week has risen by 17%
- total CPU usage has risen by nearly 12%
- the number of GIFs has risen by nearly 57%, but the CPU usage for GIFs has risen by less than 20%.

As for the CPU usage and the effectiveness of any tuning activities:

- CPU per transaction has dropped by 4.7%, but this can be distorted by an increased number of very small transactions
- Allowing for the different mix of transactions, total CPU consumption dropped by 2.5%, but this understates the improvement as we know that some of the transactions have been enhanced and now do more work. Also, the growth of databases has some impact on CPU usage
- Correcting for CPU per I/O, the apparent reduction in CPU consumption is 5.5%
- Correcting for CPU per Record Requested, the apparent reduction in CPU consumption is 5.6%

So, my(Charles Walker's) assessment is that the overall impact of tuning activities between January and August has been **approximately a 5.5% reduction in CPU usage**, which has been partly offset by an increase in the amount of work being done by transactions (as a consequence of enhancements and/or database growth).

The value of a 5.5% reduction in IDMS CPU consumption in recent months is approximately \$11,000 per month.

## 30. System Trace – July.2006 – adds approx 7%

### 30.1 July.2006

The system trace facility was enabled for IDMSP at the beginning of July in response to a recurrence of a problem which resulted in IDMSP abending on June 29th.

My best estimate of the cost impact of the additional overhead, based on a comparison of one day's statistics before tracing was enabled with one day's tracing after, is that the overhead is approximately 7%, which will add approximately \$17,800 to our CPU cost for a full month.

The intention was to leave the trace on until either the problem was fixed, or after a couple of months if there was no recurrence of the problem.

### 30.2 11.August.2006

I (Paul Mak @ EDS) took the liberty to turn SYSTRACE OFF in RES, PED, TRN and PRO leaving it in DEV until further advice from you. One point to note is that we can dcmt vary systrace on entries = 9999 to turn it on dynamically for the duration of the CV session or until it gets vary systrace off dynamically. That would be useful if we could recreate the problem within a short interval.

### 31. Determining How Many TCP/IP Bulk Terminals to Define

#### 31.1 Running out of Bulk Terminals – Feb.2007

We had an issue some weeks back when we ran out of "bulk terminals" on the TCP/IP line in DEV. This was caused by a couple of BEA Servers with IDMS Connection Pooling coming on line - they gobbled up all the available bulk terminals and nobody else could get in - either through the SQL listener for an individual desk top connection (eg. for JCF) or for one of the other application listeners. This begs the question of "how do we determine how many bulk terminals to have on our TCP/IP lines in each CV?".

The following 'formula' is what we came up with ;

**# of IPBULK terminals Required in DEV = Number Of Terminals In All IDMS Connection Pools + 5 for adhoc users + 5 \* Number of Listeners In CV**

which gives

**# of IPBULK terminals Required in DEV = 75 ( as per attached spreadsheet ) + 5 + 10 ( 5 \* 2 listeners in DEV ) = 90**

As per policy re adding Cobol program definitions ( added to all CVs at once ), all CVs need to be configured with 90 bulk terminals.

#### 31.2 Determining Number of Bulk Terminals – Mar.2007

##### IDMS Connection Pools

Mid-range Pool Name	Max Capacity		TRN	TRN	PRO
	INT	QA			
IDMSAddressFilePool	15				
IDMSIntake Pool	15				
idmsSapolConnectionPool	10	10	10		10
idmsSapolConnectionPool_COPY1	10	10	10		
IDMSSchoolPool	15				
IDMS_Pool	10	10		10	10
Port Number - SYSK	10999	15999	10999		
Port Number - SYSJ				30999	20999

# JIS Mainframe Processing Management: Resource Management sub-Project

## Tuning Issues Status after Resource Management

Tuning Issue – with post CRS settings	Current Setting	Date Last Changed	Next Change
<b>Non-sysgen Changes</b>			
Suppression of security messages			
DC091003 – take LOG out of destination			
Single Segment			
Turn PMIM off			
Signon retention			
Split System/Application Load Areas			
<b>Sysgen Parameter - MOD SYS 20.</b>			
<b>ERUS and Max Tasks</b>			
DEADLOCK DETECTION INTERVAL is 2			
CHKUSER TASKS IS 17	7 ?		
MAXIMUM ERUS IS 30	7 ?		
DEDICATED ERUS IS 1			
MAXIMUM TASKS IS 16	8 ?		
<b>Timer</b>			
EXTERNAL WAIT IS 600	300		
INACTIVE INTERVAL IS 600	60		
INTERNAL WAIT IS 900	90		
RUNAWAY INTERVAL IS 90	30		
TICKER INTERVAL IS 2	1		
QUEUE JOURNAL ALL	Before ?		
<b>System Run Units</b>			
RUNUNITS FOR LOADER = 5	3		
RUNUNITS FOR SECURITY = 5	2		
RUNUNITS FOR SIGNON = 5	2		
RUNUNITS FOR MSGDICT = 5	3		
RUNUNITS FOR QUEUE = 5	3		
RUNUNITS FOR SYSTEM/DEST = 5	0		
UNDEFINED PROGRAM COUNT IS (500 50) FOR (SUBSCHEMAS MAPS TABLES DIALOGS )	(3000 500)		
XA REENTRANT POOL IS 26000	32000		
<b>MOD LINE VTAM1</b>			
PROTOCOL IS DEFRESP	EXPRESP		
NOPERMREADBUF	PERMREADBUF		
RPL COUNT IS 100	70		