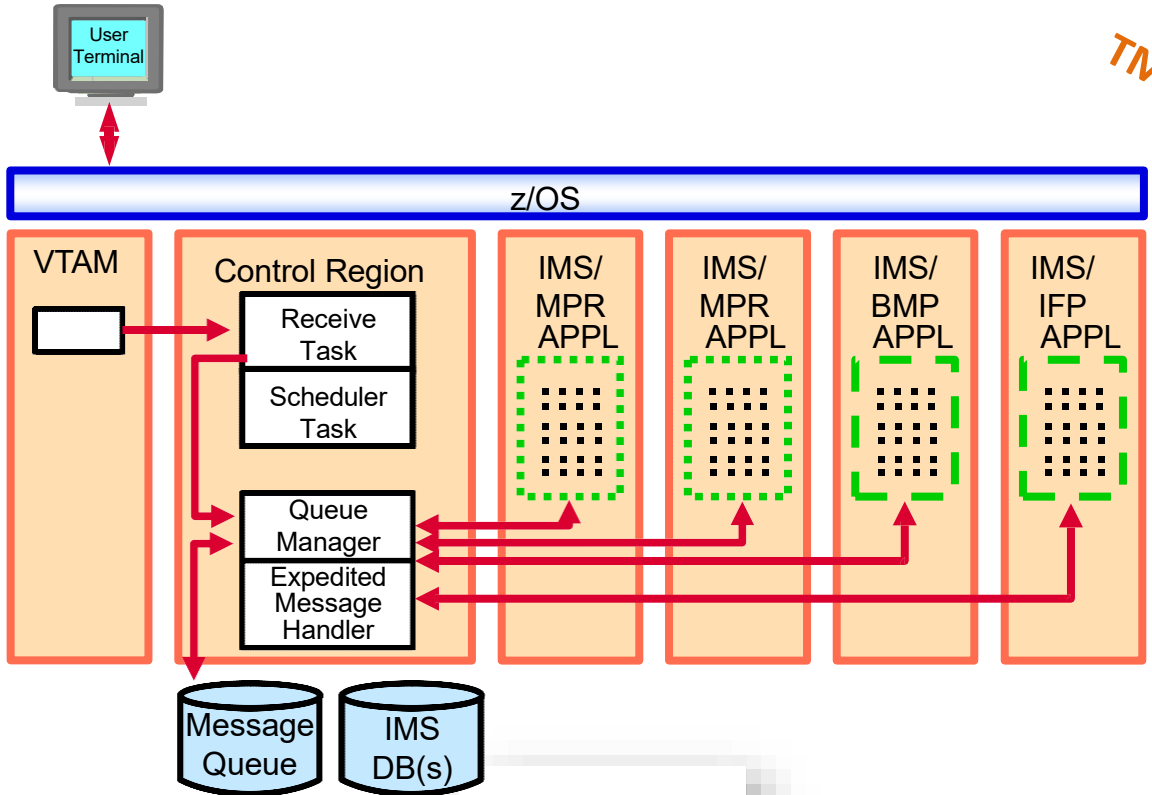


Unit 3 – Transaction Scheduling / Execution & IMS DB Processing



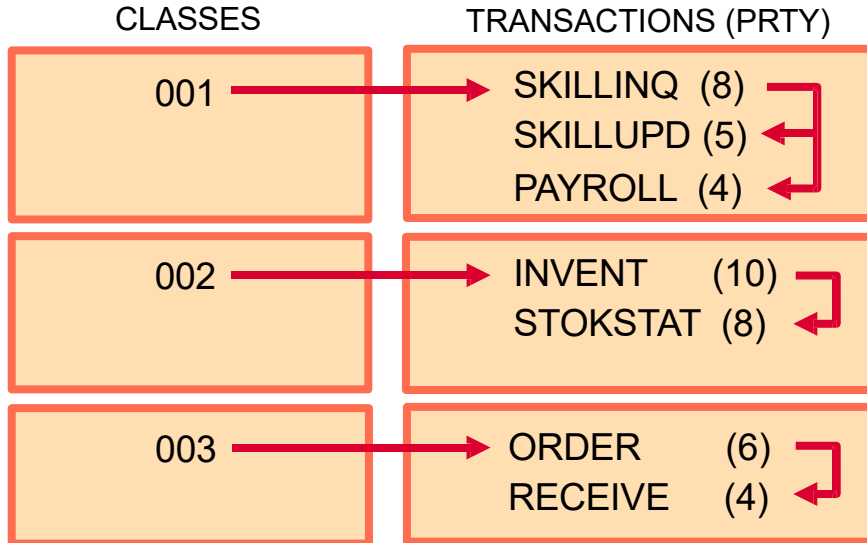
IMS/TM Tran/MSG Execution flow



Message Classes and Priorities

- TRANSACTIONS are assigned two (2) key attributes:
 1. Class, and
 2. Priority:

TM



→ Transactions are *ordered by* priority within class...

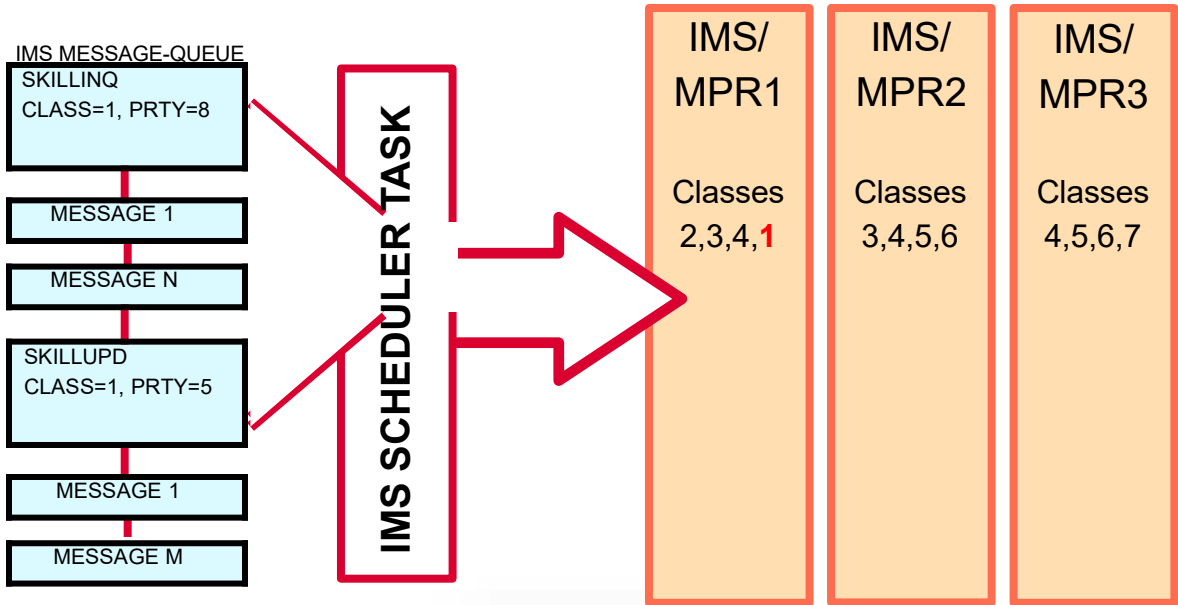
- Class and priority can be changed via /ASSIGN command



IMS Message Scheduling

TM

- Scheduling is the process in which IMS matches up messages to be processed with available Message Processing Regions (MPRs)
- Application programs are *automatically* scheduled into MPRs to process messages



→ Driven primarily by the enqueueing of *messages*...

Message Queue *Limit Priority*

TM

- Additional factors affecting scheduling

IMS MESSAGE-QUEUE

SKILLINQ
CLASS=1,
PRTY=(8,10,30)
PROCLIM=20

MESSAGE 1

MESSAGE N

SKILLUPD
CLASS=1,
PRTY=(9,12,4)
PROCLIM=5

MESSAGE 1

MESSAGE 2

MESSAGE M

	TIME 1	TIME 2	TIME 3
<div>Normal Prio</div> <p> NO LIMIT PRIORITY LIMIT COUNT CURRENT QUEUE # PROCESSING LIMIT </p>	<div>8</div> <p>10</p> <p>30</p> <p>8</p> <p>20</p>	<div>8</div> <div>10</div> <p>30</p> <p>31</p> <p>20</p>	<div>8</div> <p>10</p> <p>30</p> <p>11</p> <p>20</p>
<p> NORMAL PRIORITY LIMIT PRIORITY LIMIT COUNT CURRENT QUEUE # PROCESSING LIMIT </p>	<div>9</div> <p>12</p> <p>4</p> <p>2</p> <p>5</p>	<div>9</div> <p>12</p> <p>4</p> <p>1</p> <p>5</p>	<div>9</div> <p>12</p> <p>4</p> <p>3</p> <p>5</p>



SCHEDULING , PRTY and PROCLIM

TM

Please review by reading the IMS docs !!

- Here [\(MPP Scheduling\)](#)
- Here [\(PRTY\)](#)
- And here : [\(PROCLIM\)](#) or [HERE](#)
- Related reading: For information on scheduling options, see : [Choosing IMS options for performance](#)
- ... also here: Dave Viguers “[Scheduling](#)” summary



More scheduling options

TM

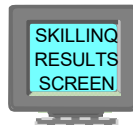
- *Wait-For-Input (WFI)* transactions:
 - Allows a program to remain scheduled to wait for the next transaction, subject to PROCLIM=
 - The idea here is to allow an already scheduled MPR to wait for message from an important transaction instead of permitting the MPR to be scheduled for another transaction
- Parallel scheduling:
 - Allows the same program to be scheduled in parallel address-spaces at the same time
 - Subject to queue-count of each transaction code Upper limit maximum
 - This is not the default and may not always be desirable
 - A consequence of parallel scheduling is that transactions can not always process in FIFO sequence
- After scheduling completes, the application program load modules must be loaded before execution begins:
 - This might take a relatively long time
 - *IMS Preload* options results in loading the application programs into certain regions as the regions initialize instead of after scheduling



Message Processing Programs

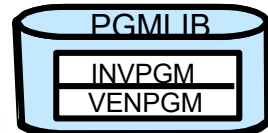
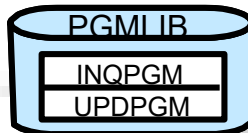
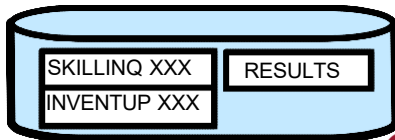
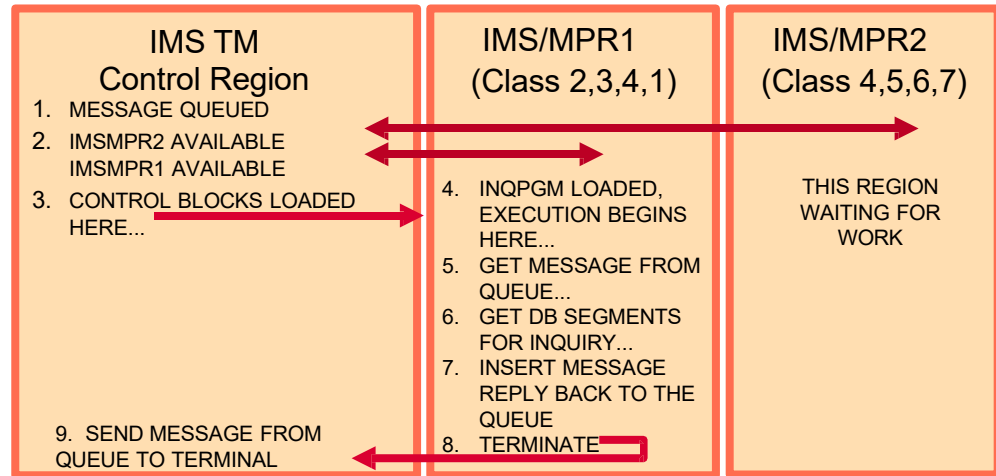


LTERM
CHI101



TM

z/OS

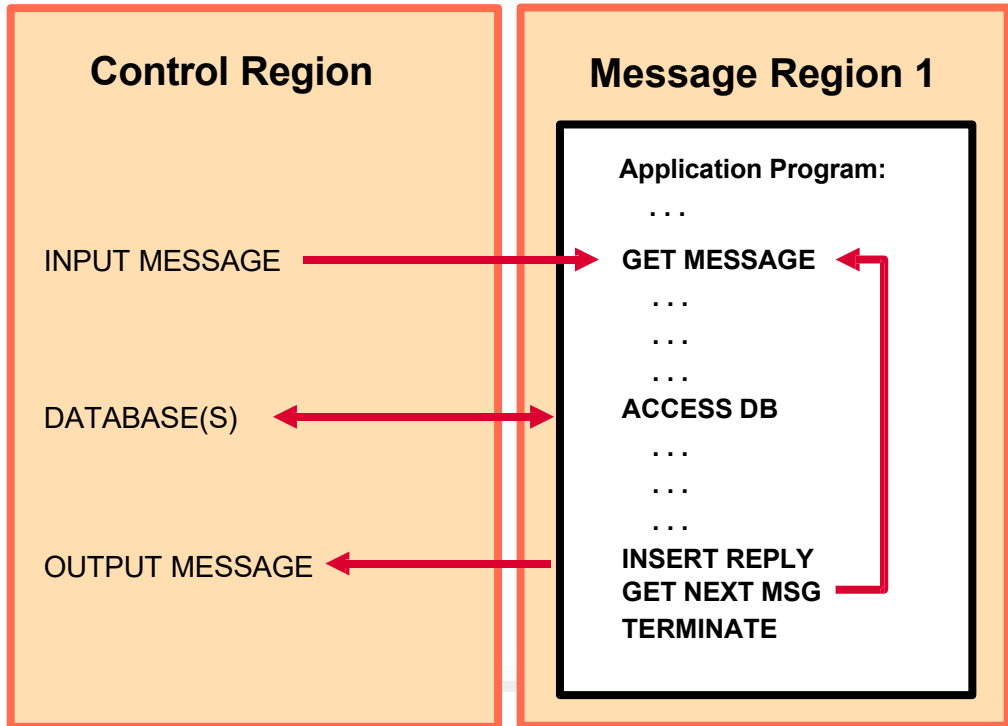


KISSLICH CONSULTING

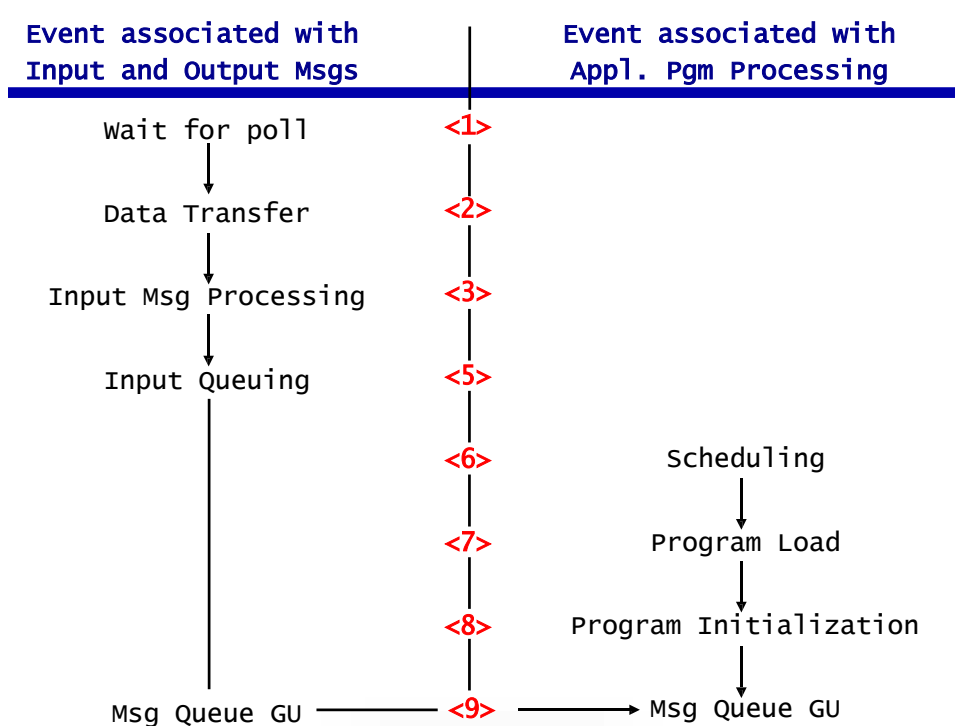
Scheduled Transaction flow

TM

```
// EXEC PGM=DFSRRRC00, PARM='MSG,002003004001,...'
```



Transaction Flow summary (1 of 2)

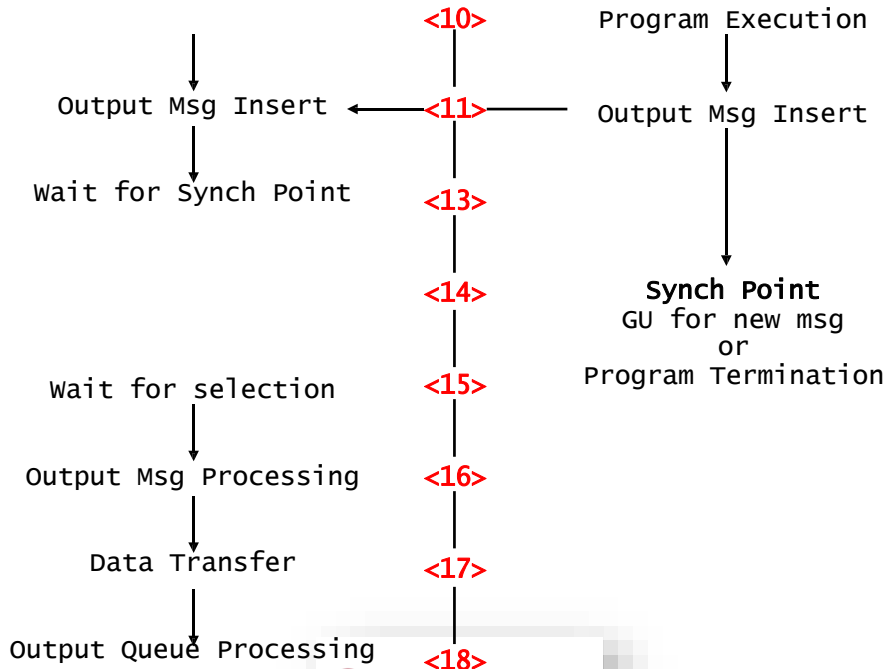


Transaction Flow summary (2 of 2)

Event associated with
Input and Output Msgs

Event associated with
Appl. Pgm Processing

TM



KISSLICH CONSULTING

Message Processing Calls

TM

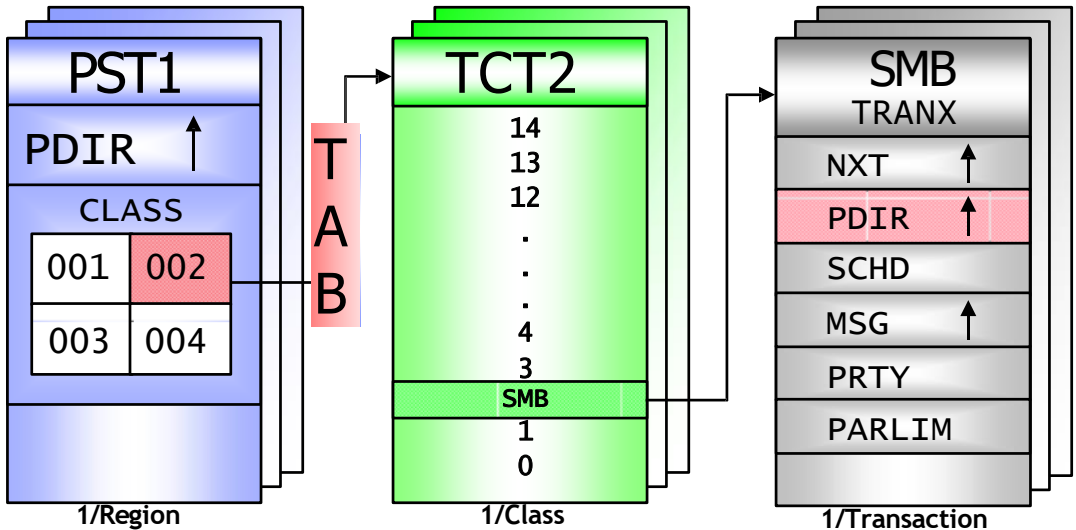
FUNCTION	PCB	CODE
RETRIEVING MESSAGES: GET UNIQUE GET NEXT	IO-PCB IO-PCB	GUbb GNbb
SENDING MESSAGES to ORIGINATING TERMINAL: INSERT	IO-PCB	ISRT

SENDING MESSAGES to ALTERNATE TERMINALS: - CHANGE - INSERT - PURGE - By EXPRESS PCB	ALT-PCB ALT-PCB ATL-PCB	CHNG ISRT PURG
--	-------------------------------	----------------------

CHECKPOINTING of the BATCH APPLICATION: CHECKPOINT SYNC-POINT	IO-PCB IO-PCB	CHKP SYNC
--	------------------	--------------

Scheduling a Transaction: The *Internal* Perspective

TM



No Messages



Idle for Intent

Procopt=EX
PSB busy
Pool Space Failure



WFI



IMS Transaction Schedule: Algorithm (1 of 2)

TM

- Scheduler gets control:
 - At MPR initialization
 - When a new message is enqueued on an SMB
 - At application program termination
 - If a CICS thread is released
- Schedule transaction to process in dependent regions on their request
- Select Transactions to schedule by priority within class
 - By class:
 - Assign a class to each TX code (1-999)
 - Assign 1-4 classes to each MPR
 - By priority within class:
 - Assign a priority to each TX code (0-14)
 - Optional automatic priority change (Limit Priority) when there is a large queue build-up
- Allocate and reserve resources for dependent regions/threads



PSB/DMB/PSBW sizes from ACBGEN output (1 of 2)

DFSUACB0 MESSAGES AND CONTROL STATEMENTS

BUILD PSB=ALL

DFS0940I DBD ADFASIGN HAS BEEN ADDED IN LIBRARY. DMB SIZE = 000640 BYTES
DFS0940I DBD ADFAAUDT HAS BEEN ADDED IN LIBRARY. DMB SIZE = 000904 BYTES
DFS0940I DBD ADFAMSGS HAS BEEN ADDED IN LIBRARY. DMB SIZE = 000696 BYTES
DFS0940I PSB ADFABCTL HAS BEEN ADDED IN LIBRARY. PSB SIZE = 008544 BYTES
DFS0941I PSB ADFABCTL IF USING DL/I SEPARATE ADDRESS SPACE, CSA SIZE = 000608,
SAS SIZE = 007888,
DFS0589I PROCESSING COMPLETED FOR PSB----ADTABCTL. PCB = 000960, PSB = 008544,
WORKAREA = 002848,
TOTAL SIZE = 011392
DFS0593I PSB--ADTABCTL WORKAREA BREAKOUT. NDX = 000256, XIO = 000008,
IOA = 001728, SEG = 000008, SSA = 00840
DFS0940I PSB ADFABCTP HAS BEEN ADDED IN LIBRARY. PSB SIZE = 008816 BYTES
DFS0941I PSB ADFABCTP IF USING DL/I SEPARATE ADDRESS SPACE, CSA SIZE = 000608,
SAS SIZE = 008160,
DFS0589I PROCESSING COMPLETED FOR PSB----ADTABCTP. PCB = 000960, PSB = 008816,
WORKAREA = 002848,
TOTAL SIZE = 011664
DFS0593I PSB--ADTABCTP WORKAREA BREAKOUT. NDX = 000256, XIO = 000008,
IOA = 001728, SEG = 000008,

DMB
Pool

PSB
Pool

Going into PSBP
going into DPSB

Needed from
PSBW



KISSLICH CONSULTING

DB Pools Backed by 64-bit Real Storage

- DB storage pools moved to 64-bit *real* storage (still in 31-bit virtual). When page fixed, these pools will now use 64-bit real storage:
 - DBWP: DB work pool
 - DLDP: DMB pool
 - DLMP: PSB CSA pool
 - DPSB: DLI PSB pool
 - PSBW: PSB work pool
- Target customers:
 - Customers with large pools (typically, PSB), who
 - Want to page fix them to avoid delays referencing old PSBs that have been paged out, but
 - Who cannot page fix them due to 31-bit real storage constraints.



ACBIN64 considerations (1 of 3)

- Scheduling considerations with non-resident ACB resources
 - At first scheduling of a program , a PSB and any related DMBs are loaded into the 31-bit non-resident pools and are also loaded into the 64-bit ACB storage pool.
 - At subsequent schedulings of this program , ACB members not found in the 31-bit non-resident pools are copied from the 64bit ACB storage pool (above the bar) back to the 31-bit non-resident pools (which avoids I/O to ACBLIB).
 - If the 64-bit ACB storage pool is full , the LRU algorithm will be used to remove old members to make room for new members



Querying the ACBIN64 storage pool

A new QUERY POOL TYPE(ACBIN64) can be used to monitor the usage of the 64-bit pool

```
QUERY POOL TYPE(ACBIN64) SHOW(STATISTICS)
```

PoolName	Type	CC	Size	Mbrs	Used	Free	Overflow
ACBIN64	Cache64	0	2048	10000	25	75	0

```
QUERY POOL TYPE(ACBIN64) SHOW(ALL)
```

PoolName	Type	CC	Size	Mbrs	Used	Free	Overflow	Gets	Hit	Miss
ACBIN64	Cache64	0	2048	3700	25	75	5	10000	90	10

Isrt	Del	Lmbr	Ltype	Lsize	Smbr	Stype	Ssize
300	20	PAYROLL	PSB	2000	DEBIT	INT	100



ACBIN64 storage pool – IMS Monitor REGION IWAIT report

IMS MONITOR
123, 08:11:48

*** REGION IWAIT ***
PAGE 0018

TRACE START 2008 123, 08:01:32 TRACE STOP 2008

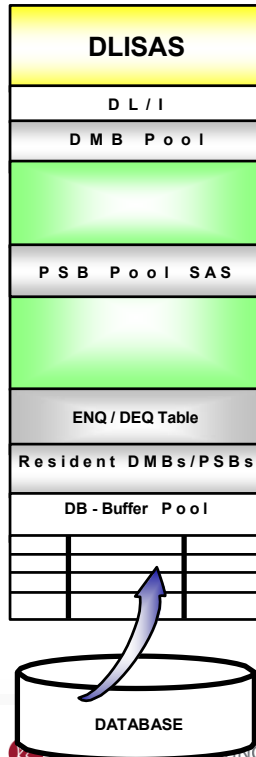
.....IWAIT TIME.....

**REGION	2	OCCURRENCES	TOTAL	MEAN	MAXIMUM	FUNCTION	MODULE
SCHEDULING + TERMINATION							
	2	32975611	16487805	23293621	NO MESSAGES	MSC	
...SUB-TOTAL...							
	2	32975611	16487805				
	1	5807	5807	5807	PSB=DDLTRN24	BLR-64BIT	
	1	1985	1985	1985	INT=DDLTRN24	BLR-64BIT	
	3	5115	1705	1965	PSB=BMPFPE07	BLR	
	1	1154	1154	1154	INT=BMPFPE07	BLR	
	3	3040	1013	1199	PSB=BMPFPE05	BLR	
	1	1028	1028	1028	INT=BMPFPE05	BLR	
	1	1739	1739	1739	PSB=BMPFPE02	BLR-64BIT	
	1	1628	1628	1628	INT=BMPFPE02	BLR-64BIT	
	1	3100	3100	3100	PSB=BMP255	BLR-64BIT	
..TOTAL...							
	15	33056434	972248				
DL/I CALLS							

IMS DB processing (Full Function)

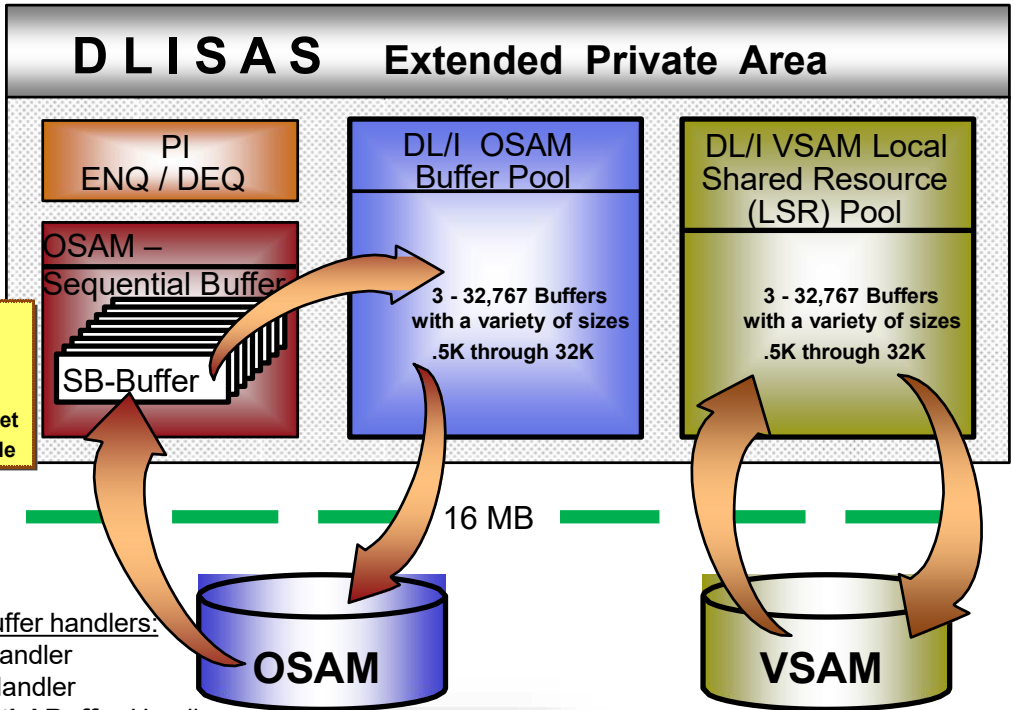
DB

IMS DB Basics now ?



IMS Database Buffers

DB



S/B Defaults:

- 4 Buffer Sets
- 10 Buffers per Set

Up to 25 Sets possible

IMS has three buffer handlers:

- VSAM Buffer Handler
 - OSAM Buffer Handler
 - OSAM **Sequential Buffer** Handler
- All have buffers in DLISAS EPA



KISSLICH CONSULTING

OSAM Buffer Pool Definition

CONTROL STATEMENTS (DFS VSM**) specifications:

IOBF=(bufsize,# buffers,fix1,**fix2**,id)
DBD=dbdname(data set number,id)



Sample:

IOBF=(512,12,Y,Y)
IOBF=(2048,5,Y)
IOBF=(4080,6,Y,Y,PROD)
IOBF=(4096,6,N,Y)
IOBF=(12288,4,Y,Y,CUST)
DBD=DBD3(1,PROD)
DBD=DBD4(2,CUST)

Three red curved arrows originate from the right side of the three IOBF statements that include a fifth parameter (PROD, CUST, and CUST). Each arrow points to the corresponding DBD statement (DBD3, DBD4, and DBD4) that follows it, indicating that the fifth parameter in the IOBF statement is the buffer pool name used in the DBD statement.

Structure of VSAM Shared Resource Pool

DB

VSAM Buffer Handler Pool

Buffer Pool Prefix
Sub - Pool 1 Prefix
Sub - Pool 2 Prefix
Sub - Pool 3 Prefix
Sub - Pool 4 Prefix

VSAM Shared Resource Pool

512	512	512	512	512	512	512	512
1024		1024		1024		1024	
1024		1024		1024		1024	
2048				2048			
2048				2048			
4096							
4096							
4096							
4096							

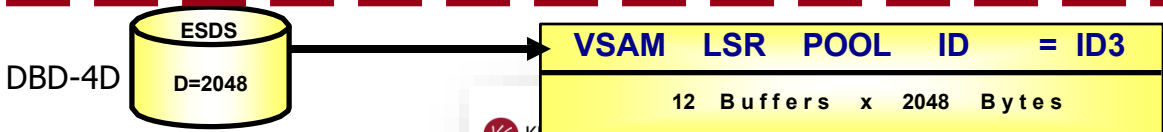
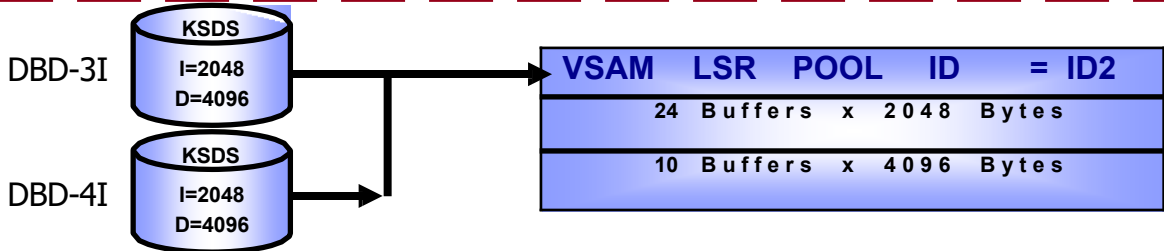
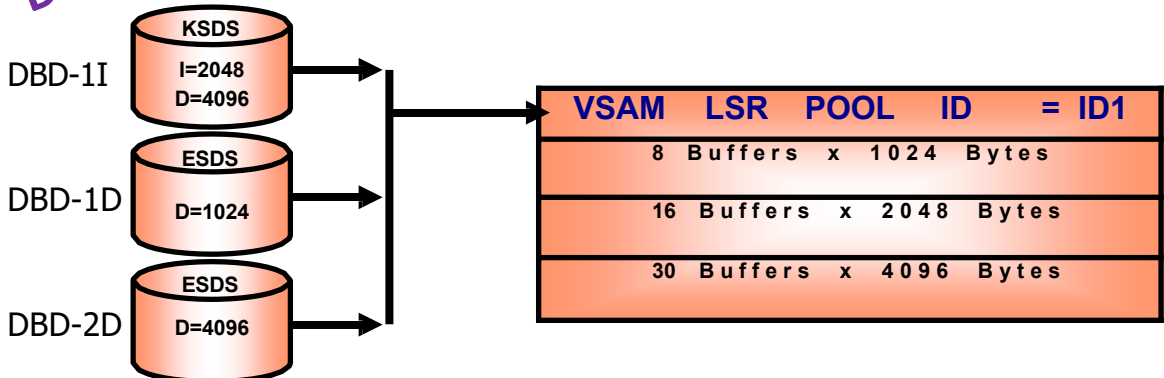
VSAM Buffer Pool Definition example

OPTIONS VSAMPL=LOCL (default) no HIPERSPACE is considered		
POOLID	= IDG	General Pool:
VSRBF	=.....	For all DBs not assigned to a specific POOLID and must be the first subpool
.....	=.....	
POOLID	= ID1	
VSRBF	= 1024,8	
VSRBF	= 2048,16	
VSRBF	= 4096,30	
POOLID	= ID2, FIXINDEX=YES	
VSRBF	= 2048,24,I	
VSRBF	= 4096,10	
POOLID	= ID3, FIXDATA=YES	
VSRBF	= 1024,12	
DBD	= DBD1I,(1,ID1)	
DBD	= DBD1D,(1,ID1)	
DBD	= DBD3I,(1,ID2)	
DBD	= DBD4I,(1,ID2)	
DBD	= DBD4D,(1,ID3)	

WHERE: POOLID : Pool Id used to assign data set to pool
 nn,nn,type : TYPE: I = Index Subpool, D = Data Subpool
 FIXDATA =, FIXINDEX= : Page fix buffers
 FIXBLOCK = : Page fix control blocks
 DBD = : Assign data set to matching Pool ID

Using Multiple VSAM Buffer Pools

DB



Klein's Consulting

BMP processing: Non-message-driven (1 of 2)

DB

- BMP started by JCL or z/OS console

```
// EXEC  IMSBATCH,MBR=pgmname, PSB=psbname,...
```

- Security verification
- PSB and DMBs loaded (if currently not in pools)
 - Databases authorized if registered with DBRC
- Region (PST) assigned or created (up to MAXPST=)
- UOR started (if UPDATE intent)
- Multiple (hopefully! *) UORs
- PSB remains scheduled until PGM termination



BMP processing: Non-message-driven (2 of 2)

DB

