

Exploring Onstat - an IBM Informix IDS utility

Lester Knutsen

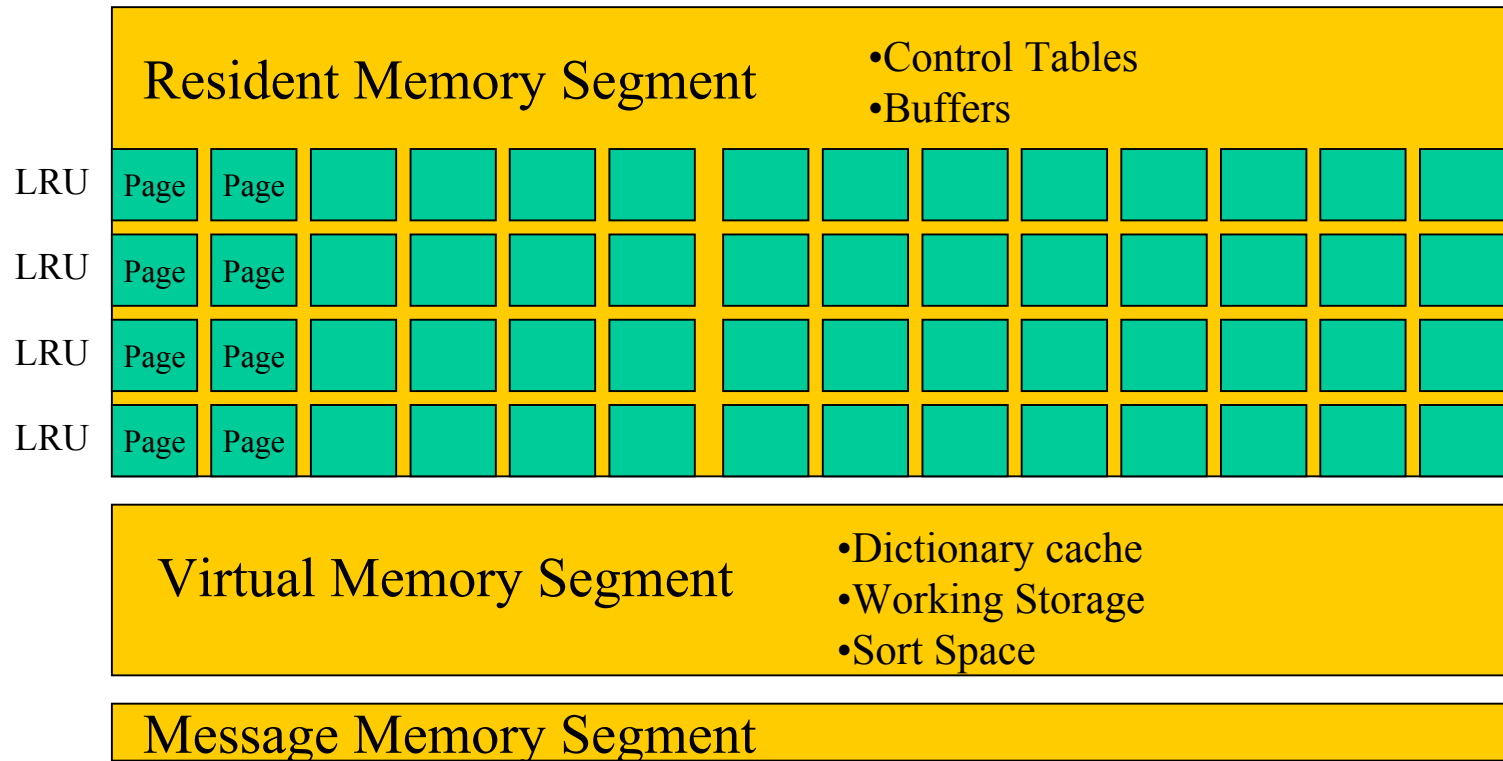
Advanced DataTools Corporation



Onstat – monitor IDS server operations

- **Onstat** utility reads shared-memory structures and provides statistics about the database server at the time that the command executes.
- The contents of shared memory might change as the **onstat** output displays.
- The **onstat** utility does not place any locks on shared memory, so running the utility does not affect performance.
- Onstat is a key utility to monitor the performance of your IDS server

Informix IDS Shared Memory



Onstat options:

onstat [-abcfgkhImpstuxzBCDFRX]

[-i]

[-r [<seconds>]]

[-o [<outfile>]] [<infile>]

-i Interactive mode

-r Repeat every <seconds> seconds (default: 5)

-o Put shared memory into specified dump file

<infile> Read shared memory information from specified
dump file

-z Zero profile counts

Key Onstat options:

- -d Print spaces and chunks
- -k Print locks
- -l Print logging
- -m Print message log
- -p Print profile
- -T Print TBLspaces
- -u Print user threads
- -D Print spaces and detailed chunk stats
- -F Print page flushers
- -R Print LRU queues

Key Onstat –g options (New monitoring commands)

- ath Print all threads
- rea Print ready threads
- sch Print VP scheduler statistics
- seg Print memory segment statistics
- iof Print disk IO statistics by chunk/file
- ses [<session id>]Print session information
- sql [<session id>]Print SQL information
- mgm Print Memory Grant Manager information

Current status of Server: onstat -

Current status: onstat -

```
lester@merlin >onstat -
```

```
INFORMIX-OnLine Version 9.2X.XXX-- On-Line -- Up 7 days  
11:54:44 -- 10656 Kbytes
```

Current status when Server is down

```
lester@merlin >onstat -
```

```
shared memory not initialized for INFORMIXSERVER 'merlindb'
```

```
lester@merlin >
```



Onstat Header Information

```
Informix Dynamic Server 2000 Version 9.21.UC6      -- On-Line -- Up 2 days 23:48:57 --  
3694592 Kbytes
```

- Product and Version
- Mode (and Type)
- (Optional Reason when Server is Blocked)
- Time Server has been up
- Size of Shared Memory in Kbytes



Mode of Server

- Off-Line Mode (does not show in header)
- Quiescent Mode
- On-Line Mode
- Read-Only Mode (DR Only)
- Recovery Mode
- Shutdown Mode

Reason when Server is blocked

- CKPT - Checkpoint
- LONGTX - Long transaction
- ARCHIVE - Ongoing storage-space backup
- MEDIA_FAILURE - Media failure
- HANG_SYSTEM - Database server failure
- DBS_DROP - Dropping a dbspace
- DDR - Discrete data replication (IDS)
- LBU - Logs full high-watermark

Server Profile:

onstat -p

Informix Dynamic Server 2000 Version 9.21.UC6 -- On-Line -- Up 2 days 23:48:57 --
3694592 Kbytes

Profile

dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached

412782302 493730321 2917750938 85.85 80916102 88592926 705858905 88.54

isamtot open start read write rewrite delete commit rollbk

1282651515 31183189 128955107 4056268803 334566515 47983470 4742621 125242 6

gp_read gp_write gp_rewrt gp_del gp_alloc gp_free gp_curs

3 0 0 0 0 0 2

ovlock ovuserthread ovbuff usercpu syscpu numckpts flushes

0 0 0 286397.69 18929.10 1382 3210

bufwaits lokwaits lockreqs deadlks dltouts ckpwaits compress seqscans

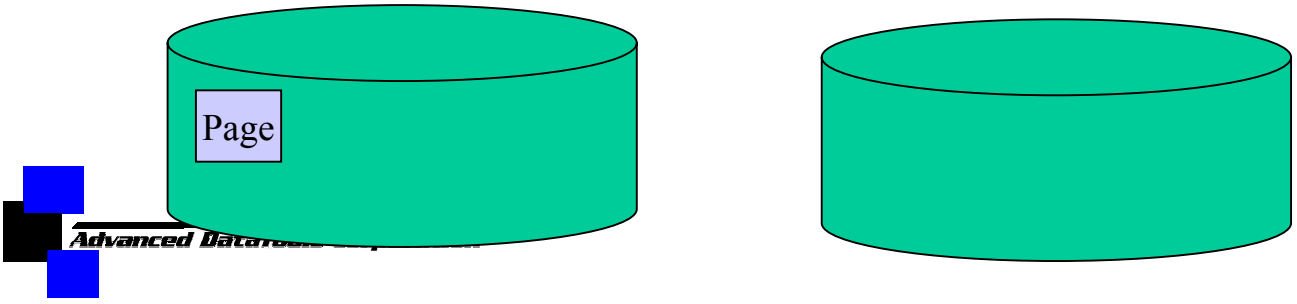
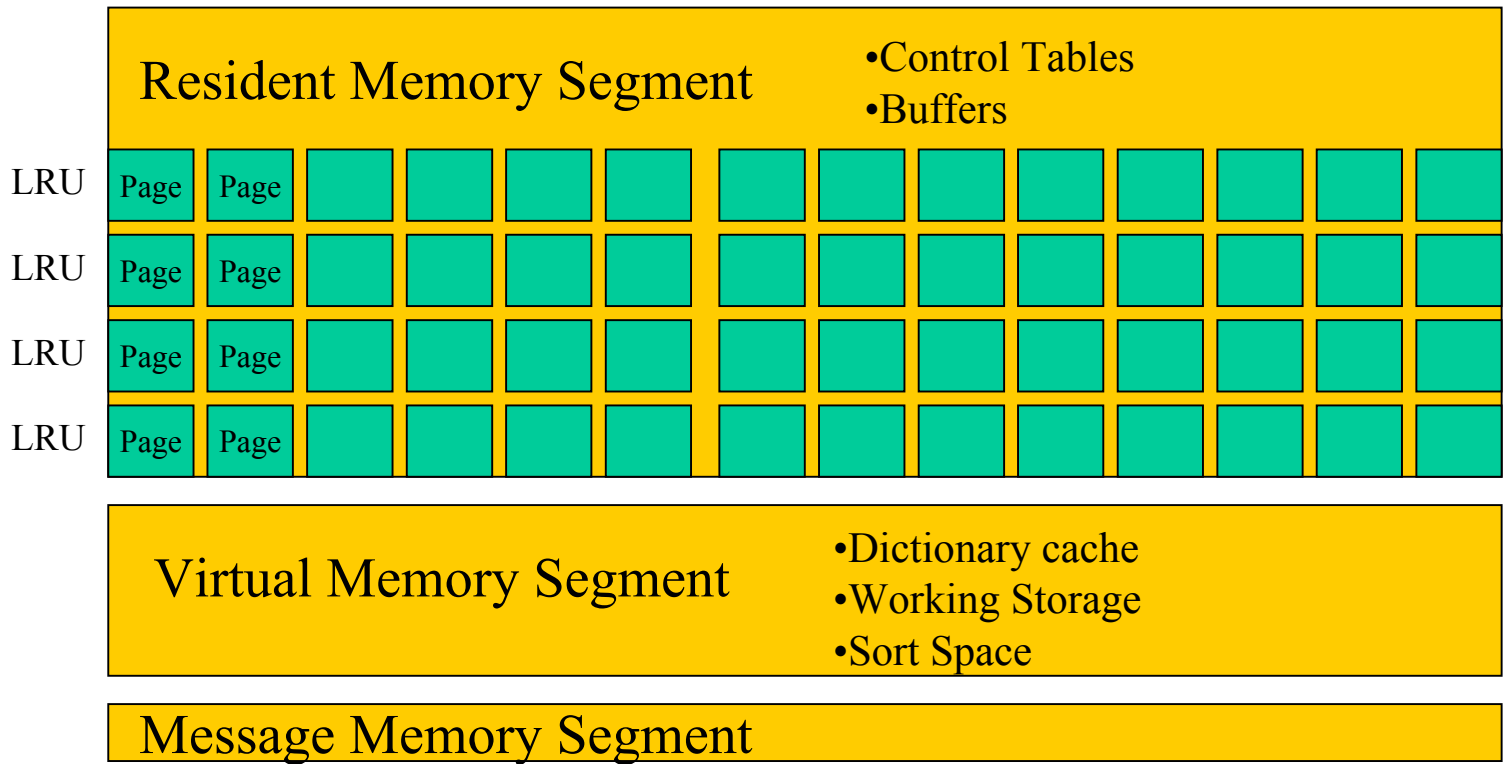
14692696 10 1065062633 0 0 2151 10377337 4240282

ixda-RA idx-RA da-RA RA-pgsused lchwaits

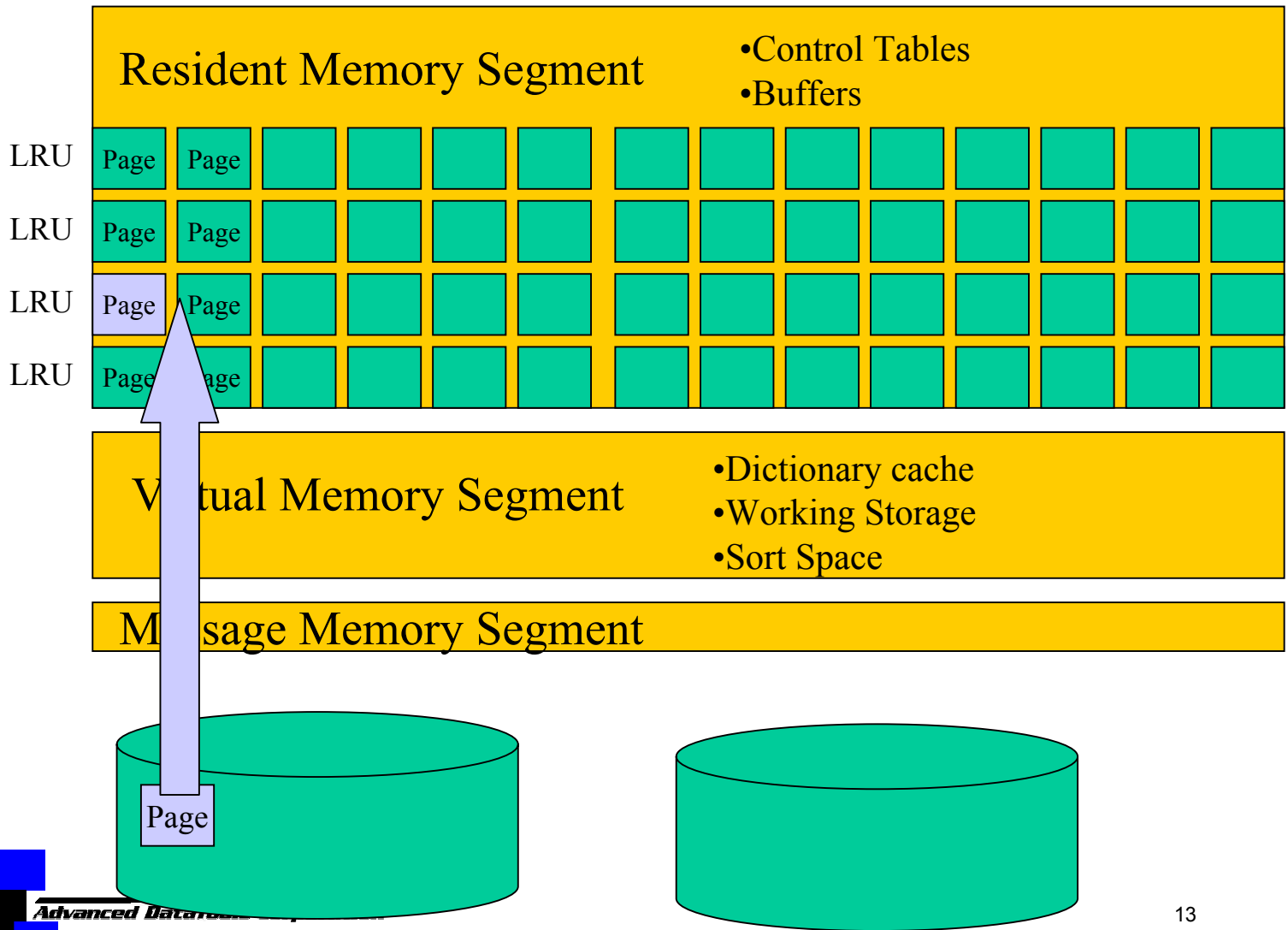
12294388 8635422 309229639 330153243 31011288



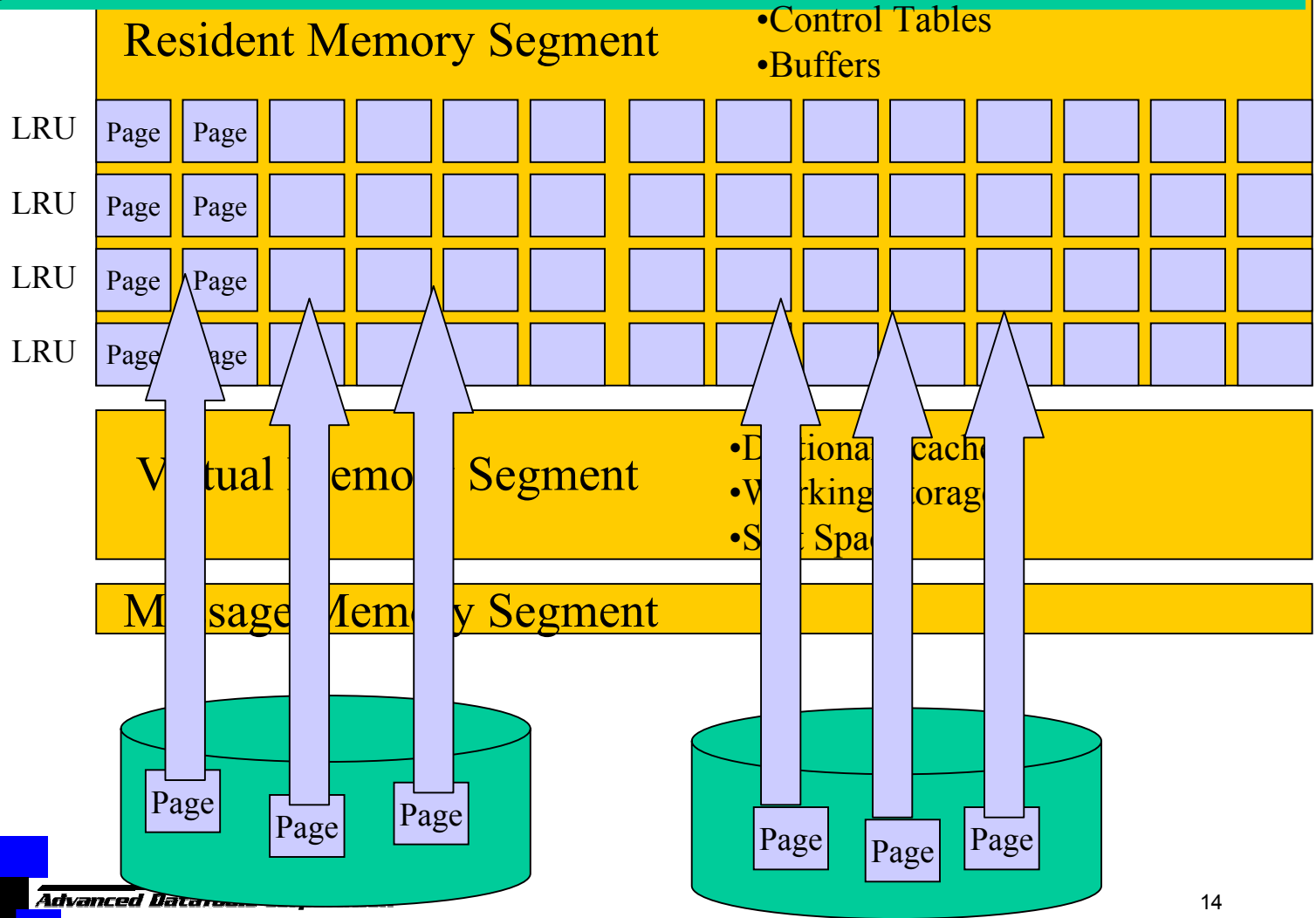
Informix IDS Shared Memory



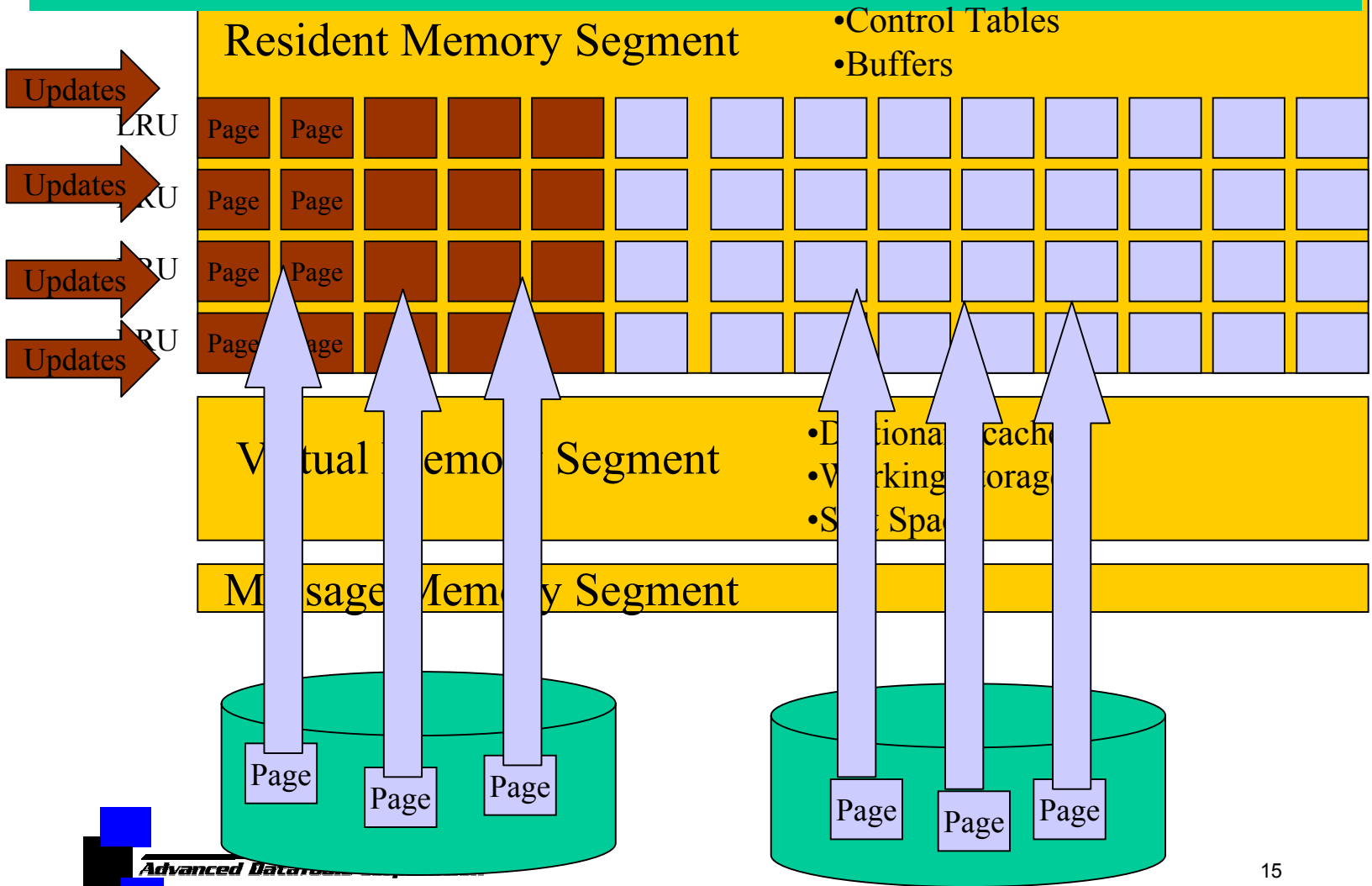
Page Gets Read into Memory by a Select



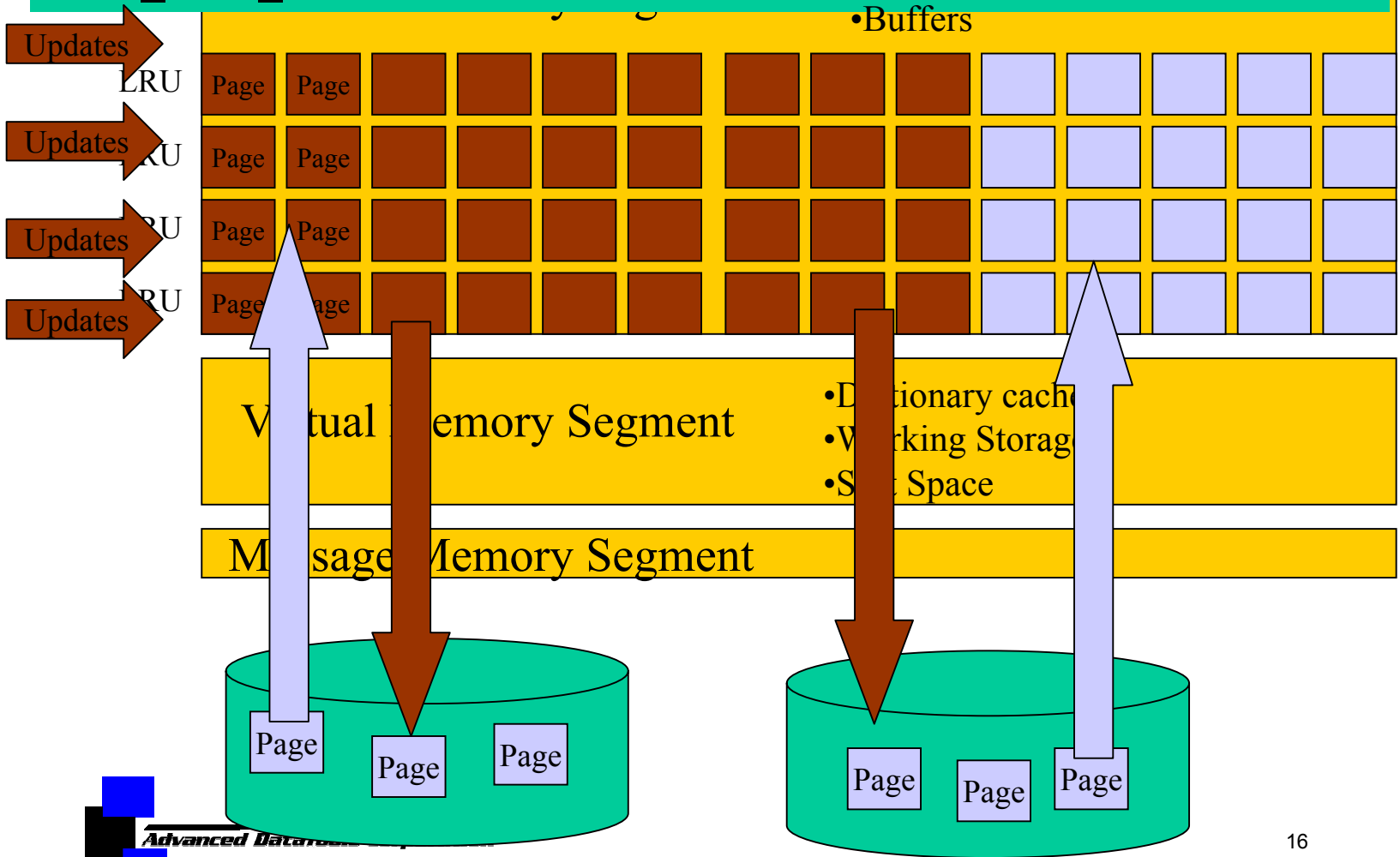
When all Buffers are full, Least Recently Used (LRU) Page is discarded to make room for more Data



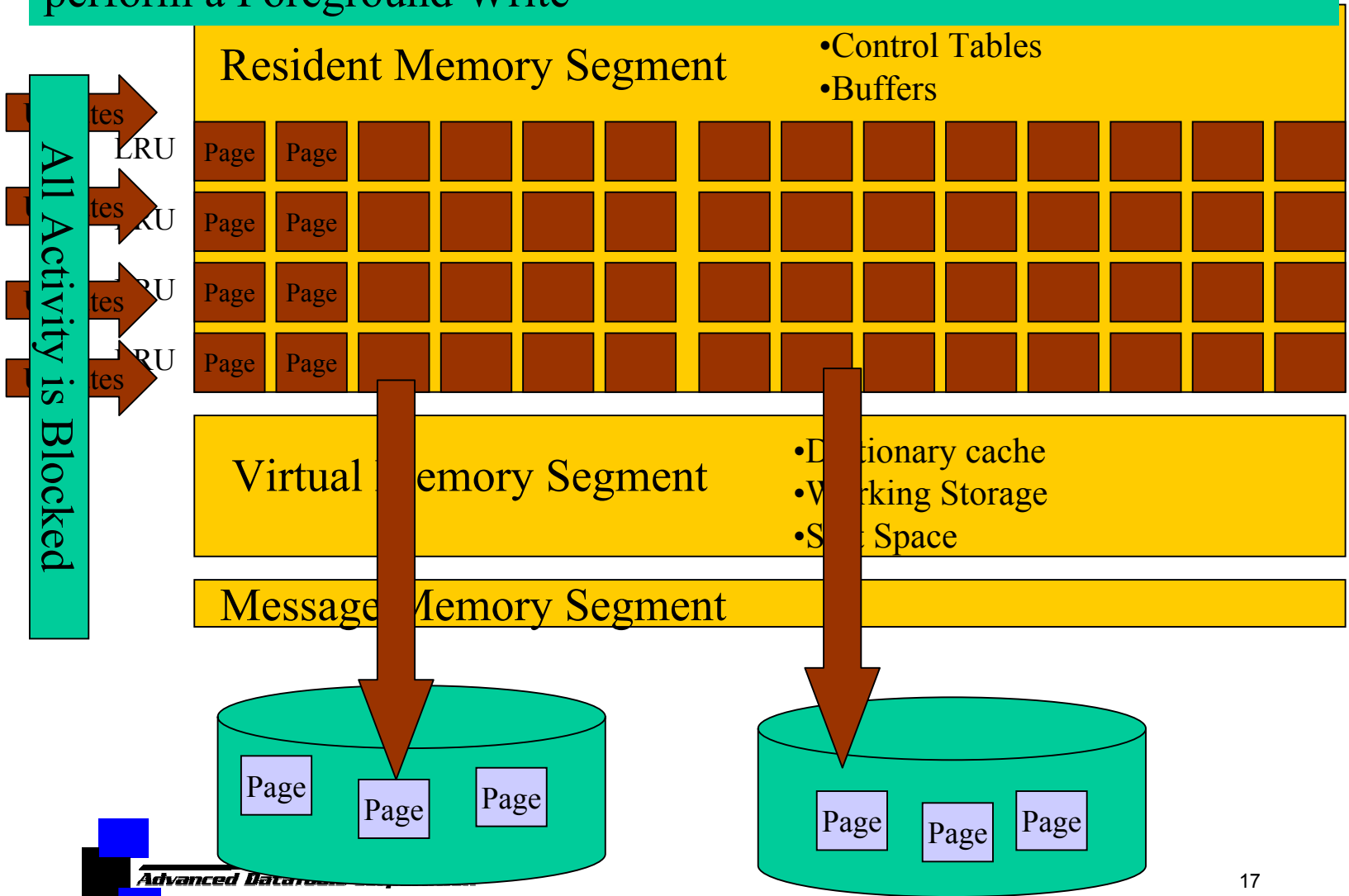
When a user updates a page, it is marked as dirty and must be written out to disk before it is discarded.



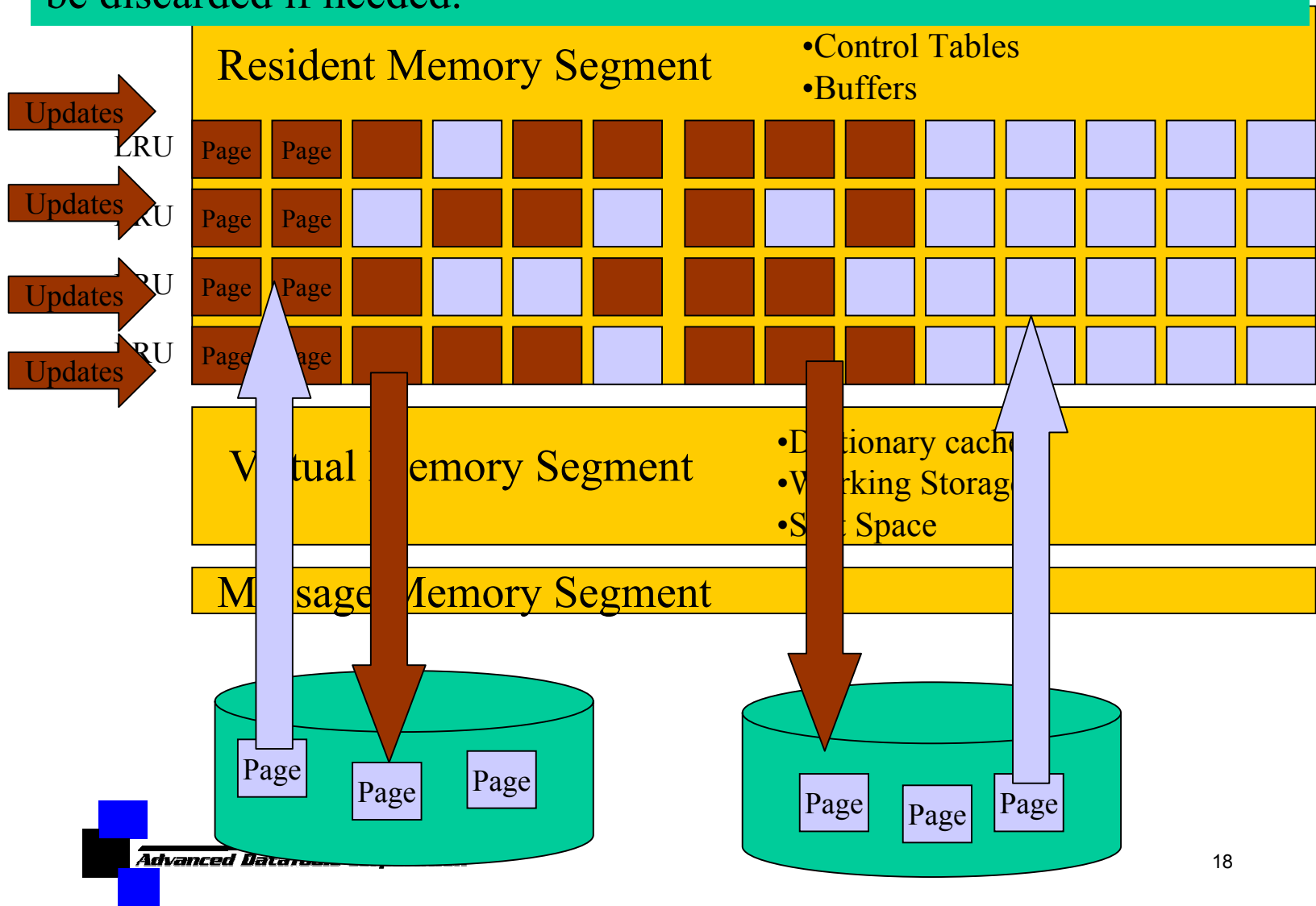
LRU Queues manage writing data to disk in the background when there is idle time based on LRU_MAX_DIRTY and LRU_MIN_DIRTY ONCONFIG values



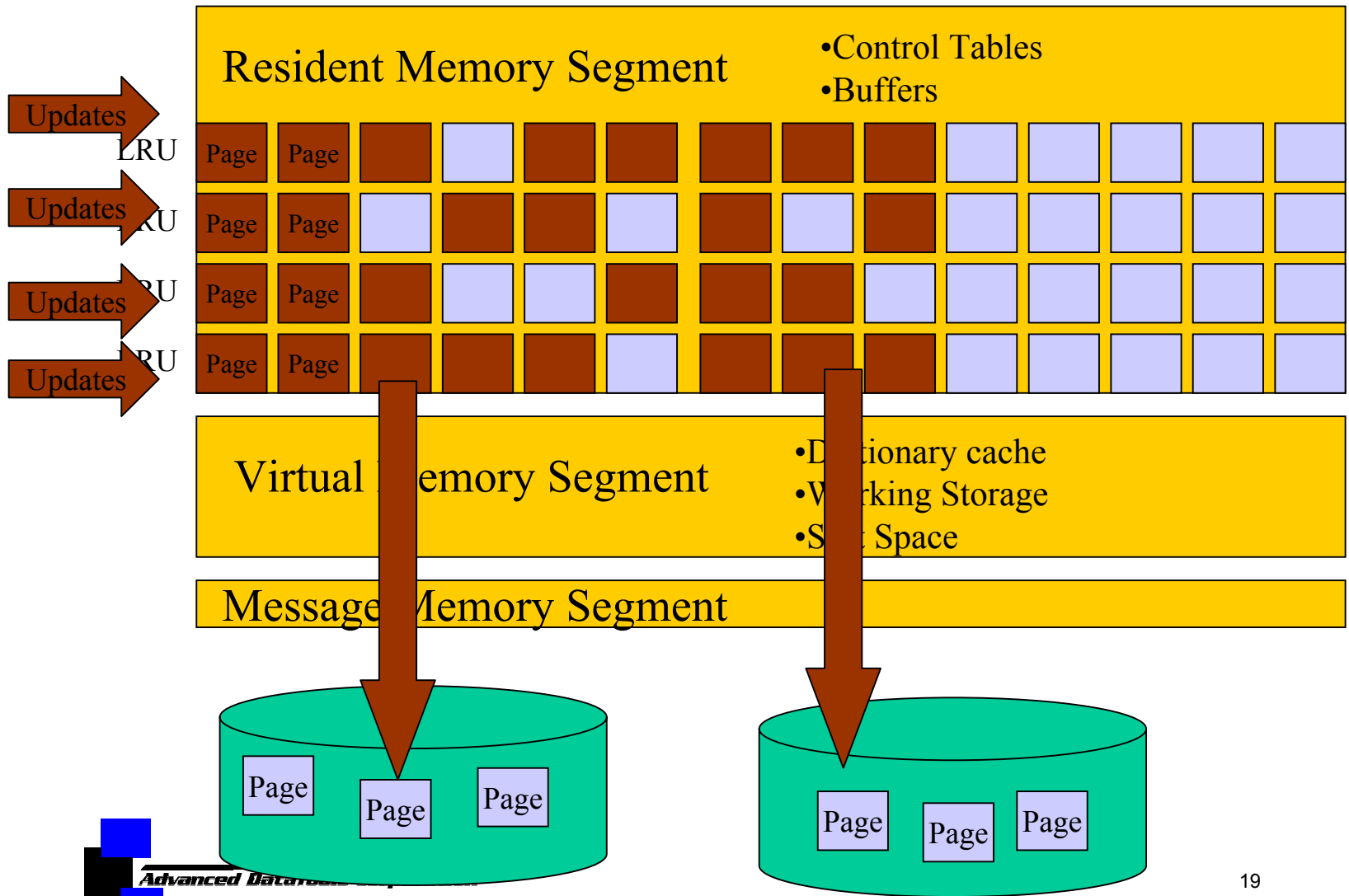
When all buffers are Dirty, the server must STOP all processing and perform a Foreground Write



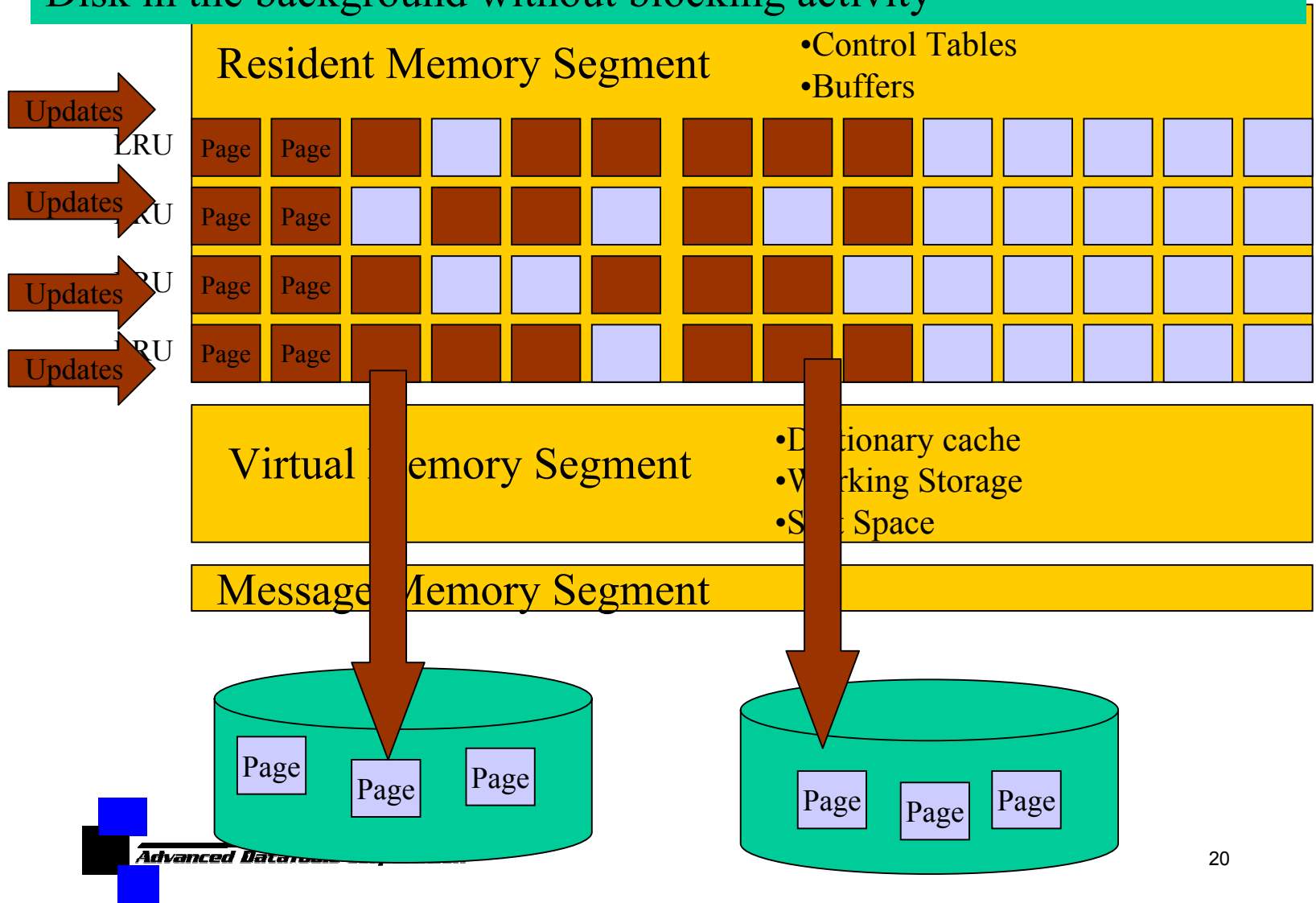
When a buffer is written to disk, it is marked as clean and may be discarded if needed.



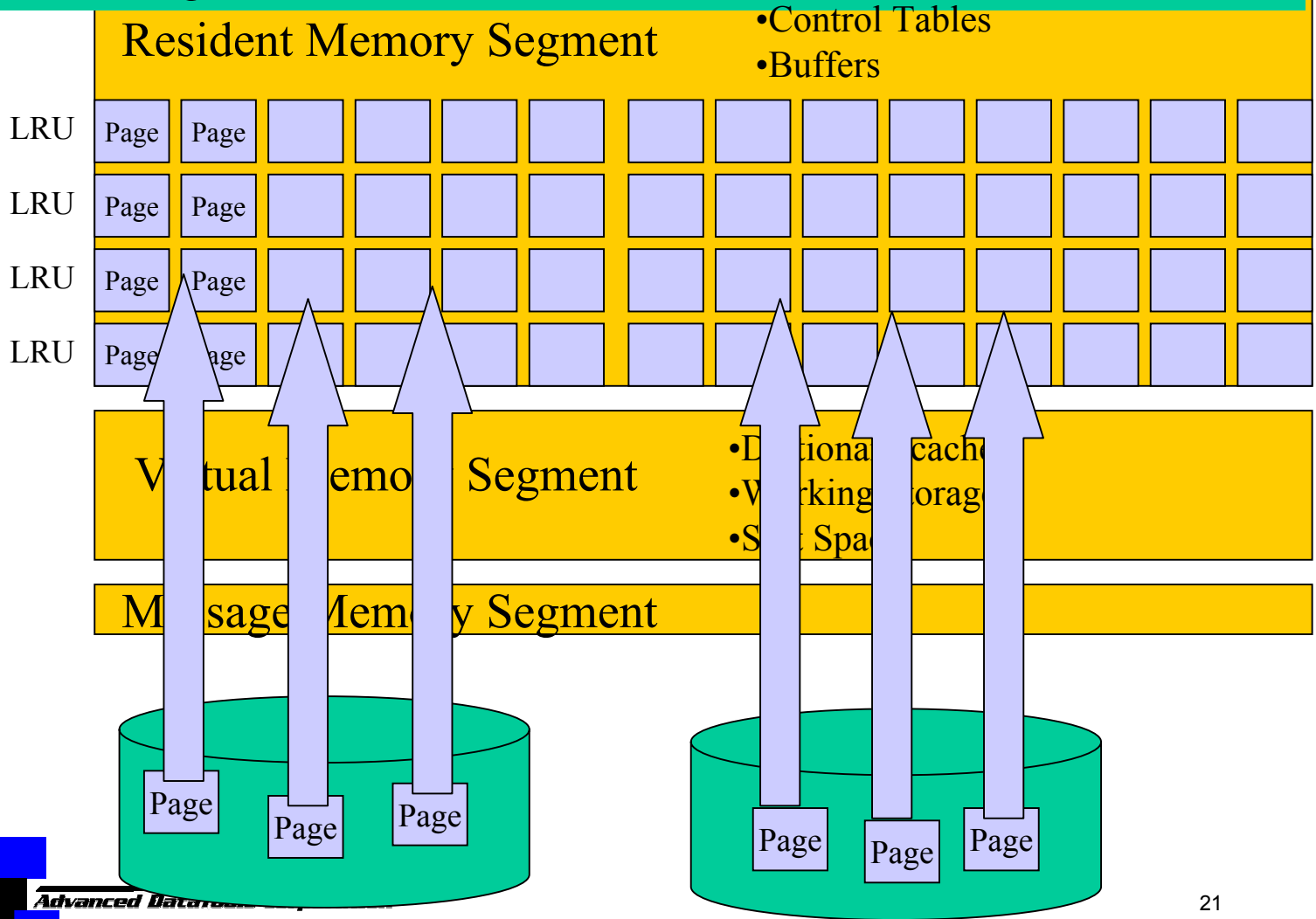
A Standard (7.X) Checkpoint writes all Dirty Buffers to Disk



A Fuzzy (9.21) Checkpoint flags all Dirty Buffers and writes them to Disk in the background without blocking activity



After a Checkpoint, all Buffers are Clean and the cycle Starts over again



Key Elements of onstat -p

- Reads %cached - The goal is $t > 95\%$
- Writes %cached - The goal is $> 85\%$
- The BUFFERS parameter in your ONCONFIG file will effect this value.
- **Be careful, if you make the BUFFERS too large this will take memory away from other processes and may slow down your whole system.**
- ovlock - This should be zero. Configure using LOCKS parameter in the ONCONFIG file.
- ovuserthread - This should be zero. Configure number of user threads in the third value of the NETTYPE parameter
- ovbuff - This should be zero. Configure using the BUFFER parameter in the ONCONFIG file.
- bufwaits - This should be zero. This indicates the number of times a user thread has waited for a BUFFER.
- lokwaits - This should be zero. This indicates the number of times a user thread has waited for a LOCK.
- deadlks - This should be zero. This indicates the number of times a deadlock was detected and prevented.
- dltouts - This should be zero. This indicates the number of times a distributed deadlock was detected.

Key Ratios Calculated from Onstat -p

- Kbytes read and written per minute/hour
- Buffer turnover ratio per minute/hour
- Buffer wait ratio
- Read Ahead Utilization

Display message log file: onstat -m

```
lester@merlin >onstat -m
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:41:12 -- 10656
Kbytes
Message Log File: /u3/informix/online1.log
21:46:51 Checkpoint Completed: duration was 8 seconds.
21:52:00 Checkpoint Completed: duration was 9 seconds.
21:57:09 Checkpoint Completed: duration was 8 seconds.
22:00:42 Logical Log 20 Complete.
22:00:43 Process exited with return code 1: /bin/sh /bin/sh -c
/u3/informix/log_full.sh 2 23 "Logical Log 20 Complete." "Logical Log 20 Complete."
22:02:17 Checkpoint Completed: duration was 8 seconds.
```

Using “tail -f” to continuously show the end of message log file

- Note: I like to have the OnLine log file always display in one of my windows on screen. The trick to doing this is to use the UNIX "tail" command with the "-f" option. This continually reads the last lines of a file as it is appended to. On my system I run the following command to continually monitor this log:

```
tail -f $INFORMIXDIR/online.log
```

User status: onstat -u

```
Informix Dynamic Server 2000 Version 9.21.UC6      -- On-Line -- Up 2 days 23:50:
12 -- 3694592 Kbytes
```

Userthreads

address	flags	sessid	user	tty	wait	tout	locks	nreads	nwrites
5911e018	---P--D	1	informix	-	0	0	0	6581	270848
5911e60c	---P--F	0	informix	-	0	0	0	0	3412274
5911ec00	---P--F	0	informix	-	0	0	0	0	2407535
59e4460c	---P---	12	informix	-	0	0	0	0	20434
59e44c00	---P--B	13	informix	-	0	0	0	11	258
59e457e8	--AP--M	3180	root	-	0	0	0	114	109
59e45ddc	---P--D	16	informix	-	0	0	0	0	0
59e463d0	Y-AP--M	3191	root	-	69542450	0	0	113	109
59e469c4	Y-----	2621	adtdba	-	6eb263d0	0	2	0	0
59e4ecbc	-----	3184	root	-	0	0	0	0	0
59e4f8a4	Y-----	2621	adtdba	-	5bf46b20	0	2	2742102	0
59e50a80	Y-----	2621	adtdba	-	5c1ee948	0	3	0	0
59e53a20	--AP--M	3186	root	-	0	0	0	116	109
59e551f0	-----	3186	root	-	0	0	0	0	0
59e57b9c	Y-----	2621	adtdba	-	5c973850	0	2	0	0
59e5cef4	Y-----	2621	adtdba	-	5c2c4af8	0	3	0	0
59e5d4e8	---P--M	3192	root	-	0	0	1	0	88



User status: onstat -u Flags

Flags in position 1

- B - Waiting on a buffer
- C - Waiting on a checkpoint
- G - Waiting on a logical log buffer write
- L - Waiting on a lock
- S - Waiting on a mutex
- T - Waiting on a transaction
- Y - Waiting on a condition
- X - Waiting on a transaction rollback

Flags in position 2

- * - Transaction active during I/O error

Flags in position 3

- A - Dbspace backup thread
- B - Begin work
- P - Prepared for commit work
- X - TP/XA prepared for commit work
- C - Committing work
- R - Rolling back work
- H - Heuristically rolling back work

User status: onstat -u Flags

Flags in position 4

P - Primary thread for a session

Flags in position 5

R - Reading call

X - Transaction is committing

Flags in position 6

None

Flags in position 7

B - Btree cleaner thread

C - Cleanup of terminated user

D - Daemon thread

F - Page flusher thread

M - ON-Monitor user thread

Logical Logs status: onstat -l

Informix Dynamic Server 2000 Version 9.21.UC6 -- On-Line -- Up 2 days 23:52:
51 -- 3694592 Kbytes

Physical Logging

Buffer	bufused	bufsize	numpages	numwrits	pages/io
P-1	8	128	34175115	268502	127.28
	phybegin	physize	phypos	phyused	%used
	10003f	500000	108581	8	0.00

Logical Logging

Buffer	bufused	bufsize	numrecs	numpages	numwrits	recs/pages	pages/io
L-3	0	64	1419836	62228	8468	22.8	7.3
	Subsystem	numrecs	Log Space used				
	OLDRSAM	1419836	113663544				

address	number	flags	uniqid	begin	size	used	%used
a8d8d70	25	U-B----	3769	10400035	5000	5000	100.00
a8d8d8c	26	U-B----	3770	104013bd	5000	5000	100.00
a8d8da8	27	U-B----	3771	10402745	5000	5000	100.00
a8d8dc4	28	U-B----	3772	10403acd	5000	5000	100.00
a8d8de0	29	U-B----	3773	10404e55	5000	2397	47.94



Logical Logs status: onstat -l Flags

The flags column provides status information about each log.

- A - Newly added, must run an archive before they can be used
- B - Backed up to tape or "/dev/null"
- C - Current logical log file
- F - Free and available for use. You will rarely see this flag as logs are not marked as free until right before they are needed.
- L - Last checkpoint is in this logical log
- U - Used logical log, it may be free if it is backed up and contains no active transactions.

One way to tell which logs can be reused is to use "onstat -l" with "onstat -x" to display all active sessions.

Display transactions: onstat -x

This option displays all current transactions. The most useful column is "log begin". This tells you in which logical log a transaction started. This may be used with the "onstat -l" command to determine which logs are free and may be reused. Find the earliest logical log number in the column "log begin". This tells you which logical log has the earliest active transaction. Any logical logs that are backed up before the log with the earliest transaction will be automatically reused by OnLine.

```
lester@merlin >onstat -x
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 01:21:38 -- 10656
```

```
Transactions
```

address	flags	userthread	locks	log begin	isolation	retrys	coordinator
a2f4384	A----	a2d1118	0	0	NOTRANS	0	
a2f44a8	A----	a2d1558	0	0	COMMIT	0	
a2f45cc	A-B--	a2d1118	2	21	NOTRANS	0	

```
6 active, 128 total, 7 maximum concurrent
```



Display locks: onstat -k

WARNING: If you have a large number of LOCKS defined in your ONCONFIG file and many users you could see thousands of rows from this command.

```
lester@merlin >onstat -k
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:53:31 -- 10656
```

```
Locks
```

address	wtlist	owner	lklist	type	tblsnum	rowid	key#/bsiz
a103e74	0	a2d2218	0	HDR+S	100002	20a	0

```
1 active, 20000 total, 16384 hash buckets
```



Who owns a lock

- The "owner" column lists the address in shared memory of the user who owns a lock. Use this with "onstat -u" to see all users, and compare this with the "address" column to identify username of the owner.

What table is locked?

- The "tblsnum" column identifies the table that is being locked. Compare this with the output of the following SQL statement to convert a table's partnum to hex. This will identify which table is locked.

1. Find a list of tblsnum

```
dbaccess database - <<EOF
    select tabname, hex(partnum) tblsnum
    from systables where tabid > 99;
EOF
```

database selected

tabname	tblsnum
genjournal	0x0010009E
gjsum	0x0010009F



What table is locked?

2. Find what is locked

```
onstat -k
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 01:47:38 -- 10656
```

```
Locks
```

address	wtlist	owner	lklist	type	tblsnum	rowid	key#/bsiz
a103e44	0	a2d1118	a103de4	HDR+X	10009f	0	0

```
3 active, 20000 total, 16384 hash buckets
```

3. Compare tblsnum from step 1 and step 2.

This identifies the table gjsun as the one that is locked.

- The tblsnum 100002 has a special meaning. This indicates a database lock. Every user who opens a database will place a shared lock on the database.

Types of locks

- Database - Lock is placed on tablespace 1000002
- Table - Lock is placed on actual tablespace with rowid of 0
- Page - Lock is placed on tablespace with rowid ending in 00
- Row - Lock is placed on tablespace with actual rowid (not 00)
- Byte - Lock is placed on tablespace/page with size of bytes
- Key - Lock is placed on tablespace hex rowid (starting with f)

Types of locks Flags

HDR	- Header
B	- Bytes lock
S	- Shared lock
X	- Exclusive
I	- Intent
U	- Update
IX	- Intent-exclusive
IS	- Intent-shared
SIX	- Shared, Intent-exclusive

Dbspaces and chunks status: onstat -d

```
lester@merlin >onstat -d
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:54:44 -- 10656  
Kbytes
```

Dbspaces

address	number	flags	fchunk	nchunks	flags	owner	name
a2ce100	1	1	1	1	N	informix	rootdbs
a2ce508	2	1	2	1	N	informix	dbspace1
a2ce578	3	1	3	1	N	informix	dbspace2
a2ce5e8	4	1	4	1	N	informix	dbspace3

4 active, 2047 maximum

Chunks

address	chk/dbs	offset	size	free	bpages	flags	pathname
a2ce170	1 1	0	250000	62047		PO-	/u3/dev/rootdbs1
a2ce280	2 2	0	10000	9587		PO-	/u3/dev/dbspace1
a2ce358	3 3	0	10000	9947		PO-	/u3/dev/dbspace2
a2ce430	4 4	0	10000	9947		PO-	/u3/dev/dbspace3

4 active, 2047 maximum



onstat -d Flags

The "flags" for Dbspaces are:

Position 1

M - Mirrored Dbpace

N - Not Mirrored Dbpace

Position 2

X - Newly mirrored

P - Physical recovery
underway

L - Logical recovery underway

R - Recovery underway

Position 3

B - Blobspace

The "flags" for Chunks are:

Position 1

P - Primary

M - Mirror

Position 2

O - On-line

D - Down

X - Newly mirrored

I - Inconsistent

Position 3

B - Blobspace Dbpace

T - Temporary Dbpace



Disk I/O: onstat -D

```
lester@merlin >onstat -D
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:55:09 -- 10656  
Kbytes
```

Dbspaces

address	number	flags	fchunk	nchunks	flags	owner	name
a2ce100	1	1	1	1	N	informix	rootdbs
a2ce508	2	1	2	1	N	informix	dbspace1
a2ce578	3	1	3	1	N	informix	dbspace2
a2ce5e8	4	1	4	1	N	informix	dbspace3

4 active, 2047 maximum

Chunks

address	chk/dbs	offset	page Rd	page Wr	pathname
a2ce170	1 1	0	36563	179558	/u3/dev/rootdbs1
a2ce280	2 2	0	3	0	/u3/dev/dbspace1
a2ce358	3 3	0	2	0	/u3/dev/dbspace2
a2ce430	4 4	0	2	0	/u3/dev/dbspace3

4 active, 2047 maximum



Writing to Disk - onstat -F

```
lester@merlin >onstat -F
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days
```

Fg Writes	LRU Writes	Chunk Writes
168	172280	5277

address	flusher	state	data
a2d0458	0	I	0 = 0X0

```
states: Exit Idle Chunk Lru
```



Writing to Disk - onstat -F

- Foreground writes occur when the Server needs a buffer and must interrupt processing to flush buffers to disk to free a buffer. These are the least desirable type of writes.
- Background writes (LRU Writes) occur when a set percent of the buffers are dirty. This is controlled by the LRU parameters in the ONCONFIG file. These do not interrupt user processing and are the best for interactive systems.
- Chunk writes occur at checkpoints, and all dirty buffer pages are written to disk. The more dirty pages, the longer a checkpoint will take. Checkpoint writes are sorted and optimized, but the longer a checkpoint is, the longer it will block user activity. Checkpoint writes are best for batch systems.

List all threads: onstat -g ath

INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:56:09 -- 10656

Threads:

tid	tcb	rstcb	prty	status	vp-class	name
2	a336b70	0	2	sleeping(Forever)	3lio	lio vp 0
3	a336dd0	0	2	sleeping(Forever)	4pio	pio vp 0
4	a337088	0	2	sleeping(Forever)	5aio	aio vp 0
5	a337340	0	2	sleeping(Forever)	6msc	msc vp 0
6	a337af8	0	2	sleeping(Forever)	7aio	aio vp 1
7	a337e00	a2d0018	4	sleeping(secs: 1)	1cpu	main_loop()
8	a34ab48	0	2	running	1cpu	sm_poll
9	a34b770	0	2	running	8tli	tlitcpoll
10	a34bce0	0	2	sleeping(Forever)	1cpu	sm_listen
11	a3c4a28	0	2	sleeping(secs: 2)	1cpu	sm_discon
12	a3c4e58	0	3	sleeping(Forever)	1cpu	tlitcplst
13	a3d0680	a2d0458	2	sleeping(Forever)	1cpu	flush_sub(0)
14	a3d0e40	a2d0898	2	sleeping(secs: 8)	1cpu	btclean
30	a35ea58	a2d1558	4	sleeping(secs: 1)	1cpu	onmode_mon
283	a39ef38	a2d2218	2	cond wait(sm_read)	1cpu	sqlxec



List Virtual Processor status: onstat -g sch

This option provides the means to identify which "oninit" UNIX process corresponds to which Informix Server server Virtual Processor.

```
lester@merlin >onstat -g sch
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:56:46 -- 10656  
Kbytes
```

VP Scheduler Statistics:

vp	pid	class	semops	busy	waits	spins/wait
1	230	cpu	21	0	0	0
2	231	adm	0	0	0	0
3	232	lio	277	0	0	0
4	233	pio	62	0	0	0
5	234	aio	144794	0	0	0
6	235	msc	756	0	0	0
7	236	aio	64028	0	0	0
8	237	tli	3	0	0	0



List SQL statement types: onstat -g sql

- This is the most interesting of the new options. This option allows you to drill down and see the actual SQL statement that a user is executing.

```
lester@merlin >onstat -g sql
```

```
INFORMIX-OnLine Version 9.2X.XXX  -- On-Line -- Up 7 days 12:52:02 --  
Sess  SQL                Current          Iso Lock        SQL  ISAM F.E.  
Id    Stmt type           Database        Lvl Mode        ERR  ERR  Vers  
264   INSERT              ffsdw          NL  Not Wait       -264 0    9.10
```



List SQL statement for a user: onstat -g sql sid

```
lester@merlin >onstat -g sql 264
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:51:10 -- 10656  
Kbytes
```

Sess	SQL	Current	Iso Lock	SQL	ISAM	F.E.
Id	Stmt type	Database	Lvl Mode	ERR	ERR	Vers
264	INSERT	ffsdw	NL Not Wait	-264	0	9.10

Current SQL statement :

```
insert into gjsum select exp_org, exp_prog, bud_obj_code, job_num,  
sum (exp_amount) from genjournal group by 1, 2, 3, 4
```

Last parsed SQL statement :

```
insert into gjsum select exp_org, exp_prog, bud_obj_code, job_num,  
sum (exp_amount) from genjournal group by 1, 2, 3, 4
```



List users sessions: onstat -g ses

- This option shows additional information about users' sessions, including how much memory each session is using.

```
lester@merlin >onstat -g ses
```

```
INFORMIX-OnLine Version 9.2X.XXX -- On-Line -- Up 7 days 12:57:48 -- 10656
```

```
Kbytes
```

session					#RSAM	total	used
id	user	tty	pid	hostname	threads	memory	memory
265	informix	-	0	-	0	8192	4680
264	lester	4	4249	merlin	1	106496	97840
10	informix	-	0	-	0	8192	4680
7	informix	-	0	-	0	16384	13144
6	informix	-	0	-	0	8192	4680
4	informix	-	0	-	0	16384	13144
3	informix	-	0	-	0	8192	4680
2	informix	-	0	-	0	8192	4680



Other Onstat Options

Repeat ONSTAT commands: -r

- To continually repeat an ONSTAT command use the "-r # of seconds" option. This is very useful when you need to monitor a situation. The following example displays the status of the logical logs every 10 seconds.

```
onstat -l -r 10
```

Clear ONSTAT shared memory statistics: onstat -z

- The Server statistics are reset every time OnLine is restarted. To reset all the statistics while OnLine is running, without shutting it down, use the following command:

```
onstat -z
```



For more information

- International Informix User Group Software Repository – <http://www.iiug.org>
- Mark Scranton's web site
<http://www.markscranton.com/informix>
- IBM Informix Documentation
- IDS Administrators Reference Manual

Advanced Training Options

- Advanced Informix IDS Performance Tuning and Optimization
 - March 31 - April 4, 2003
 - An advanced course for experienced Informix DBAs.
- Informix IDS for Database Administrators
 - February 10-14, 2003
 - Configuring, managing, and tuning the Informix IDS database server.

Lester Knutsen

Advanced DataTools Corporation

Web site: www.advanceddatatools.com

(download SQL scripts and papers)

Email: lester@advanceddatatools.com

Phone: 703-256-0267 x102



Contact information:

Lester Knutsen

Advanced DataTools Corporation

Web site: www.advanceddatatools.com

(download SQL scripts and paper)

Email: lester@advanceddatatools.com

Phone: 703-256-0267 x102

