

# Introduction to Informix Performance Tuning by Lester Knutsen

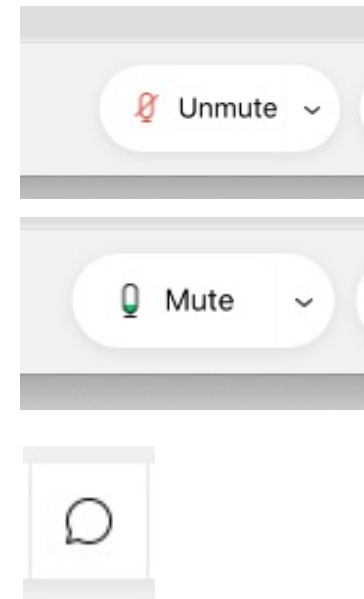
## 2022 Informix Tech Talks by the IIUG



We have launched a new channel on YouTube for Informix Users! Please subscribe to our channel on YouTube to stay informed. This will be a place for Informix how-to videos. Subscribe at: <https://www.youtube.com/c/InformixTechTalksbytheIIUG>

# Webcast Guidelines

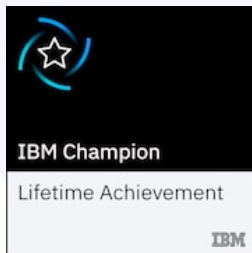
- **The Webcast is pre-recorded.**  
The replay and slides will be available on the IIUG Website
- **Please Mute your line.**  
Background sounds will distract everyone
- **Use the Chat Button** to ask questions



# Lester Knutsen



Lester Knutsen is President of Advanced DataTools Corporation and has been building and managing Informix databases systems since 1983. Lester is semi-retired but continues to teach Informix classes and provide Performance Tuning consulting. Lester is an Informix IBM Lifetime Champion. Lester was one of the founders of the Washington Area Informix User Group and the International Informix Users Group.



**[lester@advanceddatatools.com](mailto:lester@advanceddatatools.com)**  
**[www.advanceddatatools.com](http://www.advanceddatatools.com)**  
**703-256-0267**

# **Introduction to Informix Performance Tuning by Lester Knutsen**

**2022 Informix Tech Talks by the IIUG**

# Ten Informix Tuning Tips

- Correct Indexes
- Shrink Schema
- Reduce Buffer Turnover
- One SHMVIRT Segment
- One LOCK Segment
- Three or more Temp Dbspaces
- Balance Disk IO
- Use CPUVPs
- Row Level Locking
- Update Statistics

# Goal of Performance Tuning

- Balance resources to get the best performance
  - Memory
  - CPU
  - Disk
  - Network

# Scientific Method for Tuning

- **What is the Scientific Method?**
  - Ask a question - Define the Problem
  - Perform research - Observer and Measure
  - Construct a Hypothesis - Plan a Test
  - Test Your Hypothesis - Do an Experiment
  - Analyze Your Data - Draw a Conclusion
  - Communicate Results - Document Results
- **Repeat, Repeat, Repeat**

# Benchmark Worksheet

## Benchmark Worksheet

Lab: Benchmark 2

Date: 11/28/2017

Run #	Changes	Total Time	CPU % (usercpu + syscpu)	Disk I/O (pagreads + pagwrits)	Buffer I/O (bufreads + bufwrits)	Memory Used	Comments
1	<b>BASELINE</b>	17m49.455s	651.43	6494677	30566862	687428	buffers=250000
2	BUFFERPOOL=125000	8m37.551s	322.88	1402911	22158499	2933444	buffers=1250000
3	SHMVIRTSize 200000	8m36.411s	327.43	1402927	22168298	2982836	One SHMVIRT Segment
4	LOCKS 640000	5m20.191s	123.7	1402942	22180898	3067708	Resident Segment increased
5	RESIDENT -1	5m11.777s	107.3	680186	19748667	3070500	
6	VPCLASS cpu,num=4,noag	5m5.403s	108.79	1403024	22205882	3070500	
7	VP_MEMORY_CACHE_KB	5m6.878s	108.57	1403389	22223989	3070500	
8	PHYSBUFF 512 LOGBUFF	3m20.977s	78.13	673413	21319488	3072548	
9	DIRECT_IO 1	3m27.670s	77.58	665103	19944711	3072548	
10	PLOG and LOG	4m11.27s	86.68	1795234	22266939	3072548	
11	SAME	4m13.798s	88.15	1133695	20665709	3236388	
12	SAME no Server restart	3m38.265s	172.32	3439465	42449726	3236388	
13	SQL changes - run1	2m16.111s	27.96	2321957	7626794	3236388	
14	SQL changes - run2	3m17.596s	94.72	2172403	13550781	3236388	
15	SQL changes - run3	0m21.523s	9	486110	3738464	3236388	
16							
17							
18							
19							
20							



# Key Metrics for Tuning

```

real    590m16.824s
user    0m0.040s
sys     0m0.008s

IBM Informix Dynamic Server Version 14.10.FC3 -- On-Line -- 09:55:00 -- 4408904 Kbytes

Profile
dskreads  pagreads  bufreads  %cached dskwrits  pagwrits  bufwrits  %cached
135662    137436    12217396 0754 100.00 3158796   4171669   8263397   61.77

isamtot   open       start     read      write     rewrite   delete    commit    rollbk
26538791  59495     154488    7272863   1506243   1914195   641       7458      0

gp_read   gp_write   gp_rewrt   gp_del    gp_alloc  gp_free   gp_curs
0         0         0         0         0         0         0

ovlock    ovuserthread ovbuff     usercpu   syscpu    numckpts   flushes
0         0         0         35903.43  52.30     121        124

bufwaits  lokwaits   lockreqs  deadlks   dltouts   ckpwaits   compress   seqscans
163       0         61080286 217 0        0          6         34362     101419

ixda-RA   idx-RA     da-RA     logrec-RA RA-pgsused lchwaits
900       144       108919    2         109062    12056
    
```

# Benchmark Variance

- 8 Machines with same hardware repeatedly running the same script for 24 hours
  - 5% to 6% difference in times
- VM Machines:
  - 21% to 26% difference in times
- AWS Machines:
  - 14% to 58% depending on configuration
- **Best result is to run a test 3 times**

# One Change at a Time

- Make only one change at a time
- Measure the effect of the change
  - Clock Time
  - CPU Time
  - Memory Usage
  - Disk I/O Throughput
  - Network Traffic
- Evaluate the change

# Ten Informix Tuning Tips

- Correct Indexes
- Shrink Schema
- Reduce Buffer Turnover
- One SHMVIRT Segment
- One LOCK Segment
- Three or more Temp Dbspaces
- Balance Disk IO
- Use CPUVPs
- Row Level Locking
- Update Statistics

# 1 - Correct Indexes

Look for:

- Unnecessary sequential scans
- Excessive disk reads
- Avoid Duplicate Indexes

# Index Performance Example

- Same SQL Query (poorly written)
- Server – Default Configuration
- Server – Moderate Tuning (More Memory BUFFERS)
- Server – Default Configuration – Fixed one missing Index

# Correct Indexes Performance Example

<u>Server Configuration</u>	<u>Index</u>	<u>Minutes</u>	<u>Hours</u>	<u>Days</u>
Default Configuration	Missing Index	15,046.00	250.77	10.45
Moderate Tuning	Missing Index	590.00	9.83	0.41
Default Configuration	Added Index	7.00	0.12	0.00

# Correct Indexes Performance Example

Server Configuration	Index	Minutes	Hours	Days	Pages Read	Pages Write	BTR	CPU	Mem
Default Configuration	Missing Index	15,046	251	10	183,378,880,757	18,579,020	4,779	597,001	264,884
Moderate Tuning	Missing Index	590	10	0	122,173,960,754	8,263,397	1	35,956	4,408,904
Default Configuration	Added Index	7	0	0	71,578,783	13,517,654		664	256,692



# Find Tables with Scans

```
-- Module: @(#)table_with_seqscans.sql  2.3      Date: 2020/01/01
-- Author: Lester Knutsen  Email: lester@advancedatools.com
--       Advanced DataTools Corporation
-- Description: Find tables with sequential scans

-----

database sysmaster;

select  first 100
        dbname database,
        tablename table,
        partnum partnumber,
        ti_npdata  table_size_pages,
        sum(seqscans) total_scans,
        (ti_npdata * (sum(seqscans))) total_pages_scanned
from    sysptprof, systabinfo
where   sysptprof.partnum = systabinfo.ti_partnum
and seqscans > 0
and tablename not in ( select tablename from systables where tabid < 100 )
and dbname not in ( "sysmaster", "sysadmin" , "sysuser", "sysutils" )
group   by 1, 2, 3, 4
order   by 6 desc
```

# How to Monitor Index Usage

```
-- #####
-- ## Module: @(#)index_usage2.sql      2.0      Date: 08/25/2019
-- ## Author: Lester Knutsen  Email: lester@advancedatools.com
-- ##           Advanced DataTools Corporation
-- #####

select
    t.tabname,
    i.indexname,
    bufreads,
    bufwrites,
    case
        when bufwrites = 0 then bufreads
        when bufreads = 0 then 0
        else ( bufreads /bufwrites )
    end ratio
from   systables t, sysfragments i,  outer sysmaster:sysptprof p
where  t.tabid = i.tabid
and    i.fragtype = "I"
and    i.partn = p.partnum
and    t.tabid > 99;
```

# How to Monitor Index Usage

tablename	state
indexname	idx_state_1
bufreads	15
bufwrites	11
ratio	1.36363636363636

tablename	zip
indexname	idx_zip_1
bufreads	630617
bufwrites	931
ratio	677.354457572503

tablename	benchmark
indexname	idx_benchmark_1
bufreads	214154
bufwrites	4614
ratio	46.4139575205895

Poor Index Usage

Great Index Usage

Good Index Usage

## 2 - Shrink Table Schema

- Reduce Page Size so more rows fit on a page
- Divide Wide Table into two with 1 to 1 relationship
  - Master Table with the most used columns
  - Extra Table with the seldom used columns



# Shrink Table Schema Example

## 2 minutes to 5 seconds

```
real 2m20.711s
user 0m0.004s
sys 0m0.006s

IBM Informix Dynamic Server Version 14.10
2022-02-28 12:53:17

Profile
dskreads pagreads bufreads %cached
87604503 5286695366 20592636510 99.62

isamtot open start read
23077249940 2363060 1996032 8320542
```

```
real 0m5.506s
user 0m0.007s
sys 0m0.004s

IBM Informix Dynamic Server Version 14.10
2022-02-28 12:56:56

Profile
dskreads pagreads bufreads %cached
87604508 5286808892 20594375341 99.62

isamtot open start read
23080004364 2363194 1996152 8323561
```

Table Name	bills
Owner	informix
Row Size	1266
Number of Rows	605280
Number of Columns	19
Date Created	02/28/2022

Table Name	bills
Owner	informix
Row Size	116
Number of Rows	605280
Number of Columns	13
Date Created	02/28/2022

# Tools to Monitor Schema

- Art Kagel's Script – waste.bash
- Informix Compression Estimator

database sysadmin; -- Estimate Compression

Execute function task ("table estimate\_compression", "bills", "benchmark2" );

Execute function task ("table estimate\_compression", "customer", "benchmark2" );

- Waste Calculation

Rows per Page = (( Page\_Size - 28) / ( Row\_Size +4 ))

Data Size per Page = ( Rows per Page \* ( Row\_Size +4 ))

Waste per page = (( Page\_Size - 28) - Data Size per Page )

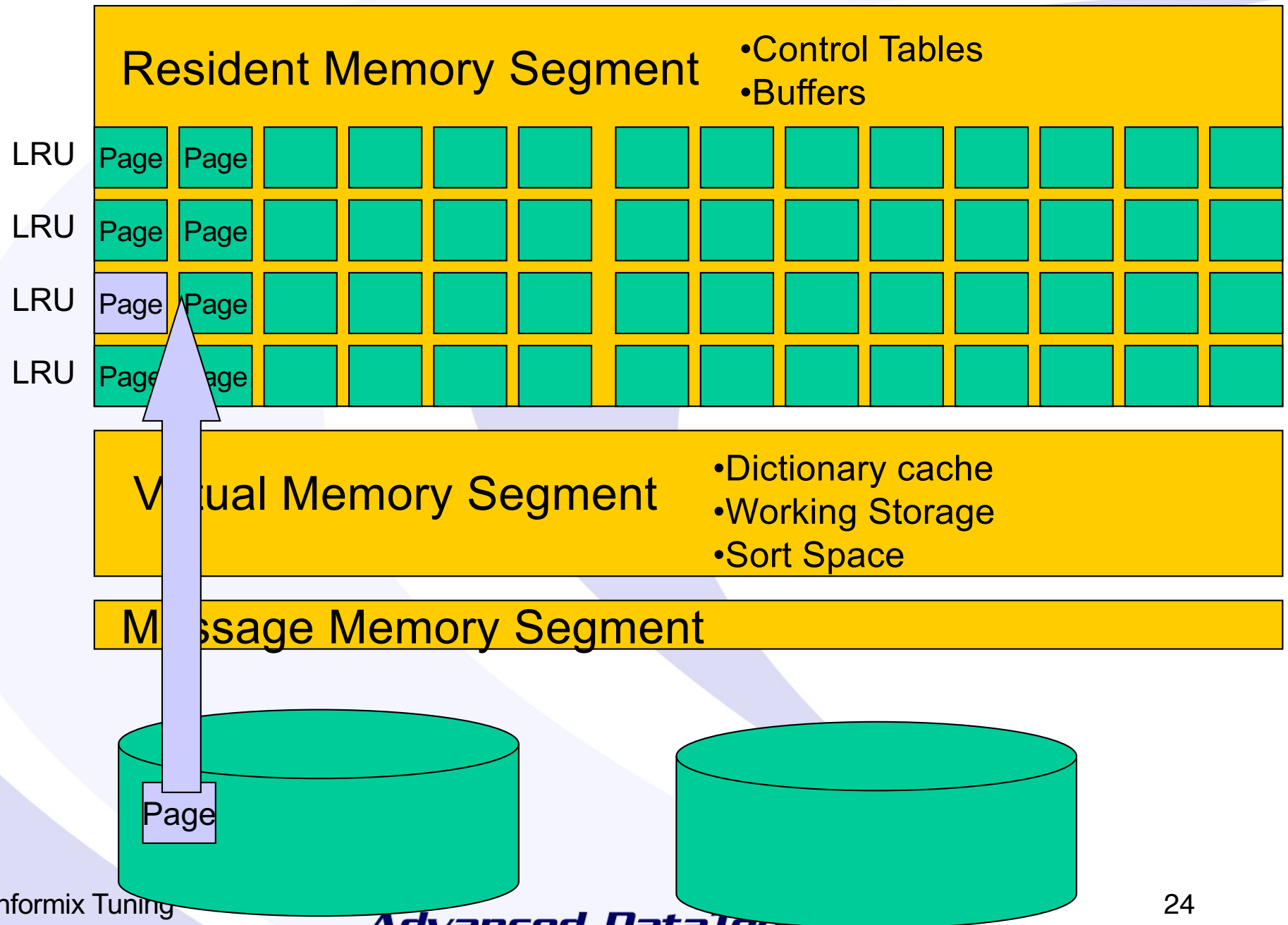
Estimated Table Waste = ( Waste per page \* npused )

# 3 - Reduce Buffer Turnover

## Informix Memory Best Practices

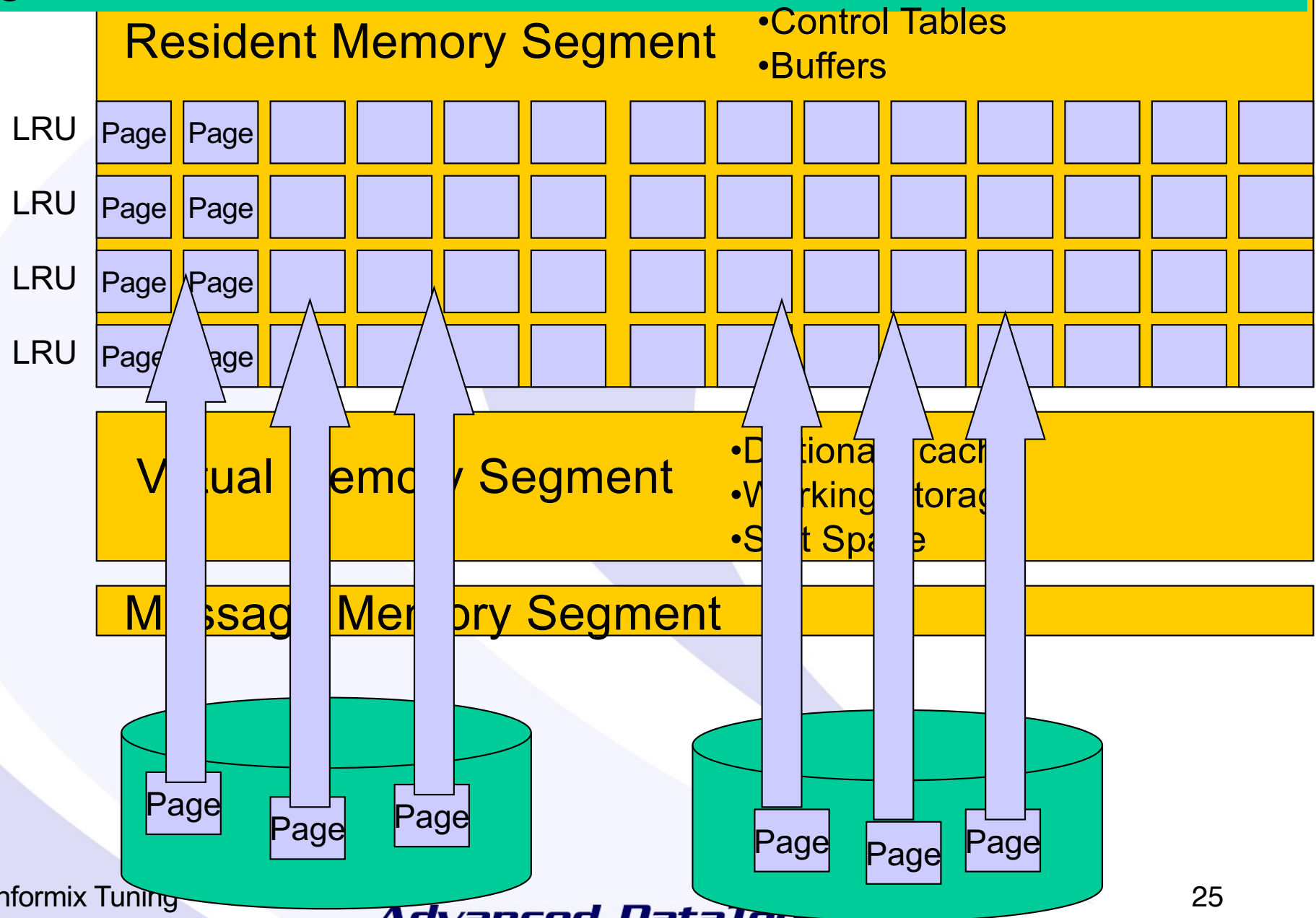
- How much memory is available on the machine?
- How much is used by the Operating System and other applications?
- How much will be assigned to Informix?
- ***DO NOT allow the machine to Swap memory to disk as this will SLOW everything down***

# Page Gets Read into Memory by a Select

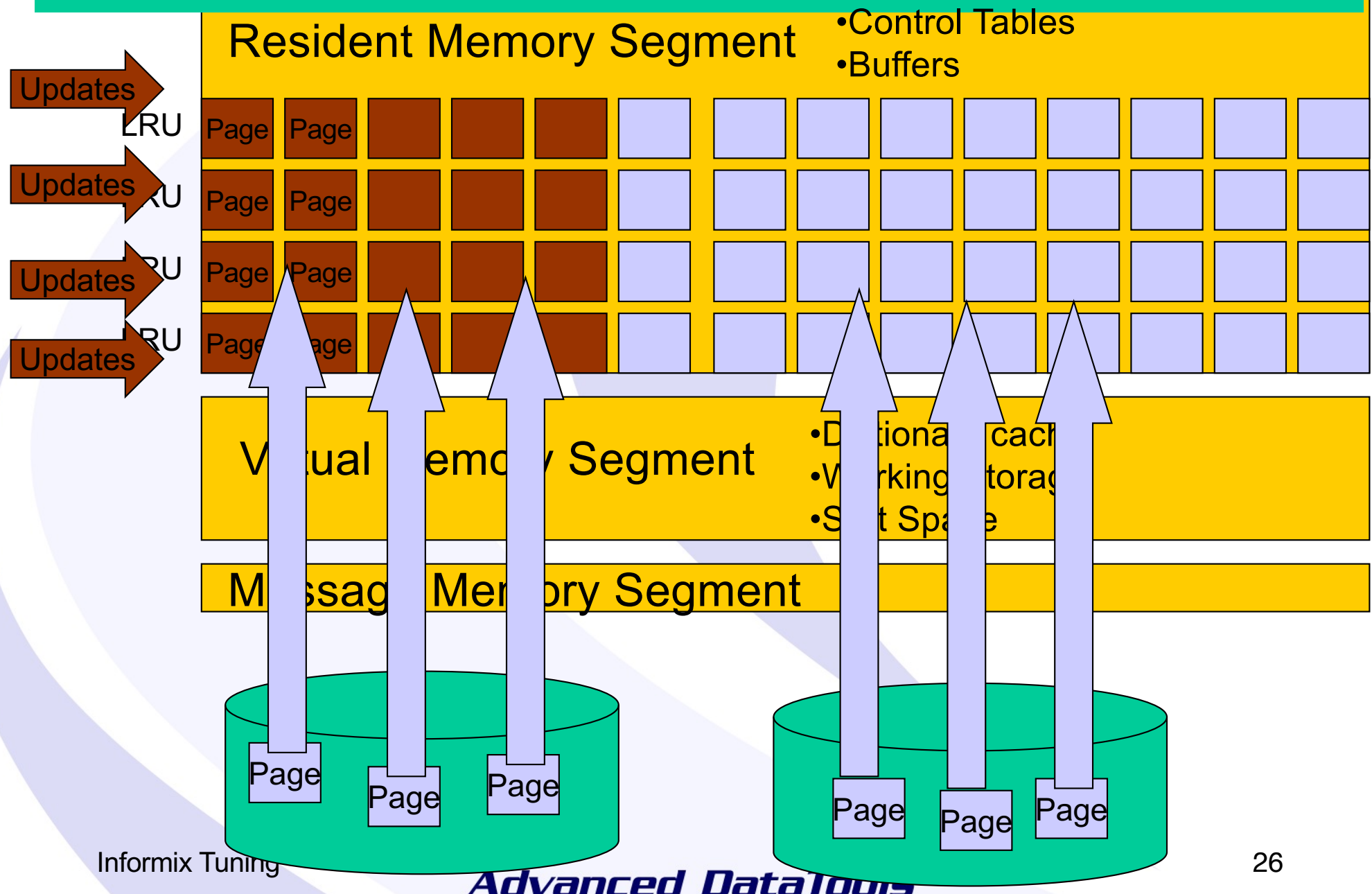




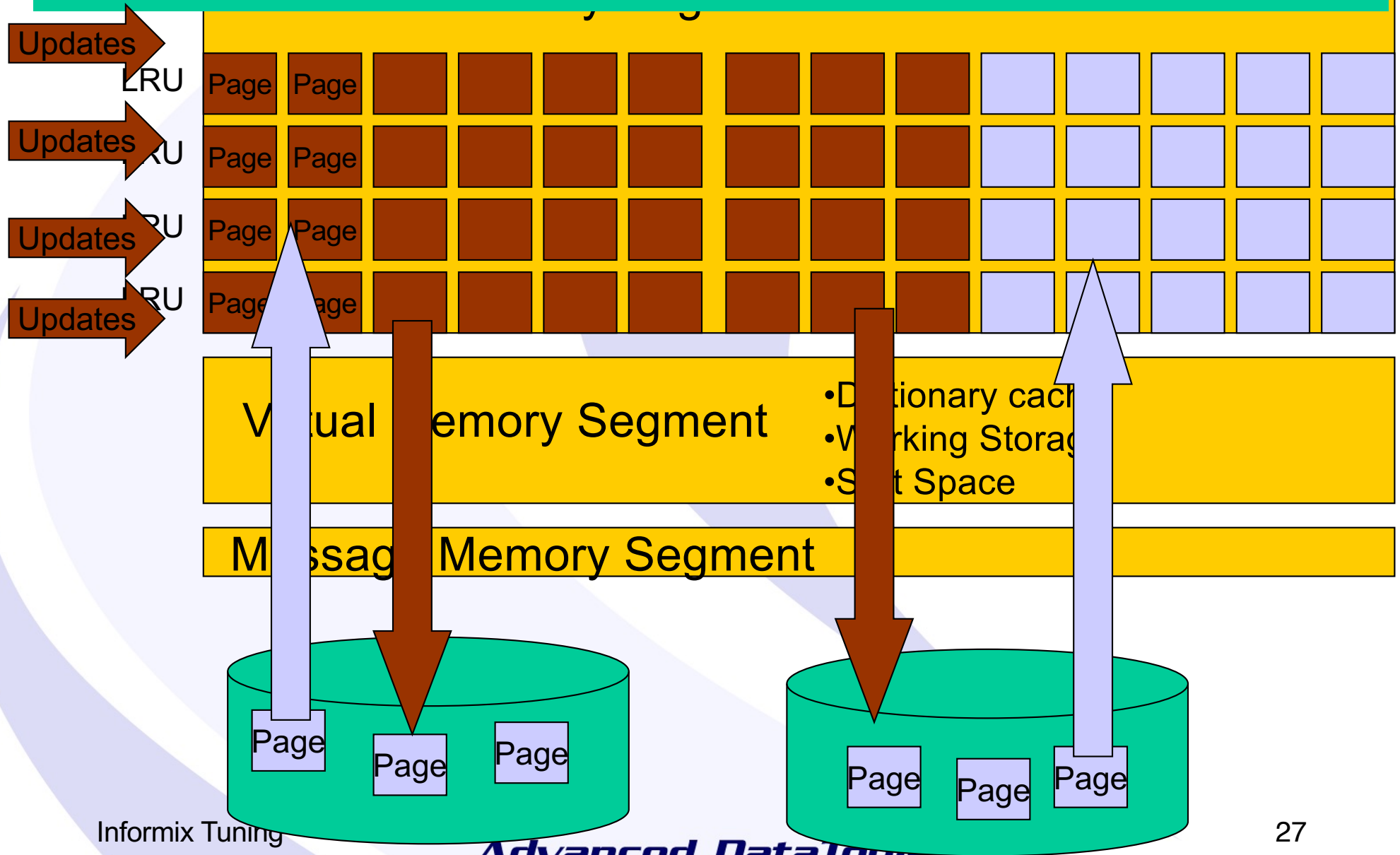
When all Buffers are full, Least Recently Used (LRU) Page is discarded to make room for more data.



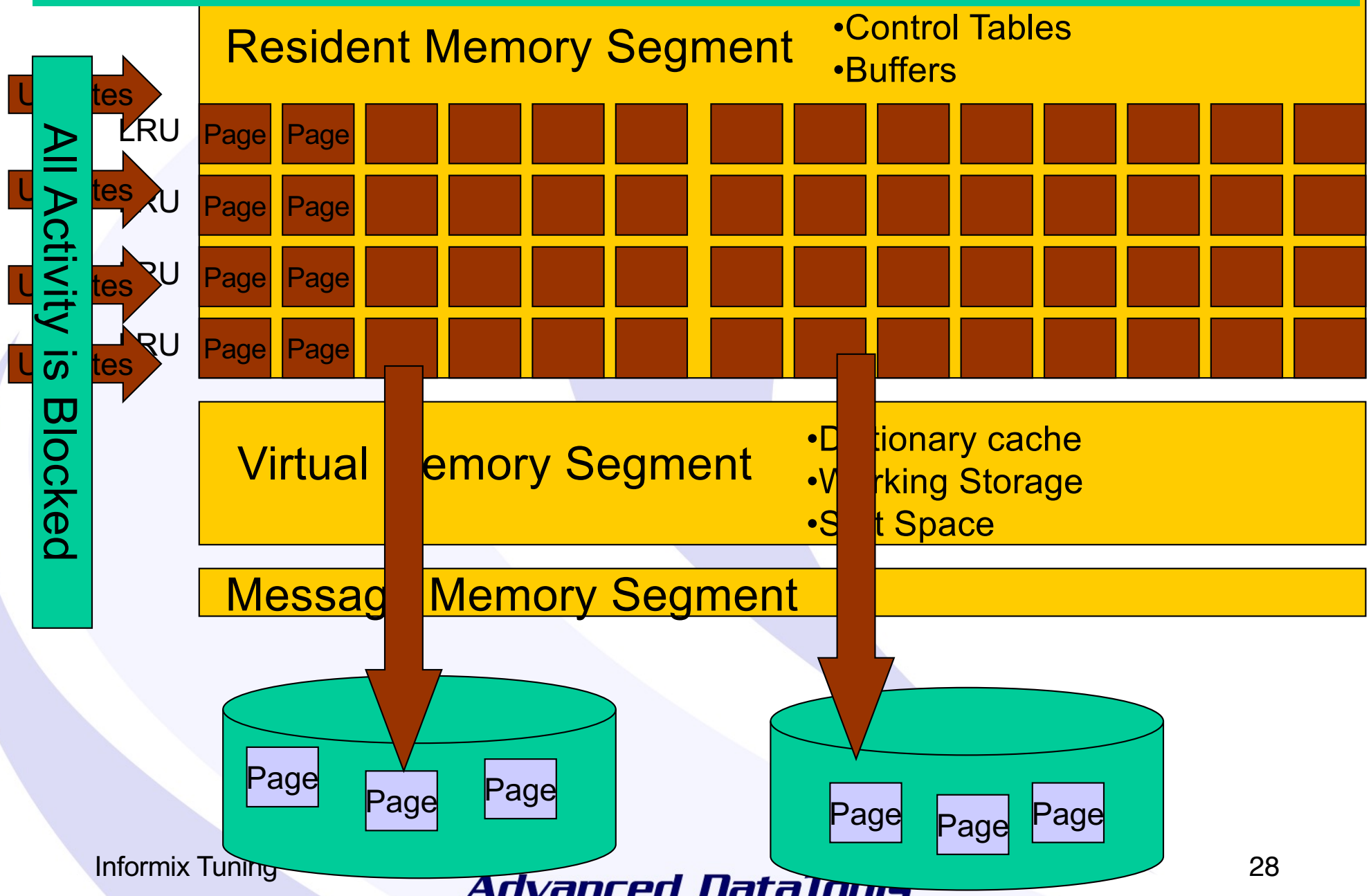
When a user updates a page, it is marked as dirty and must be written out to disk before it is discarded.



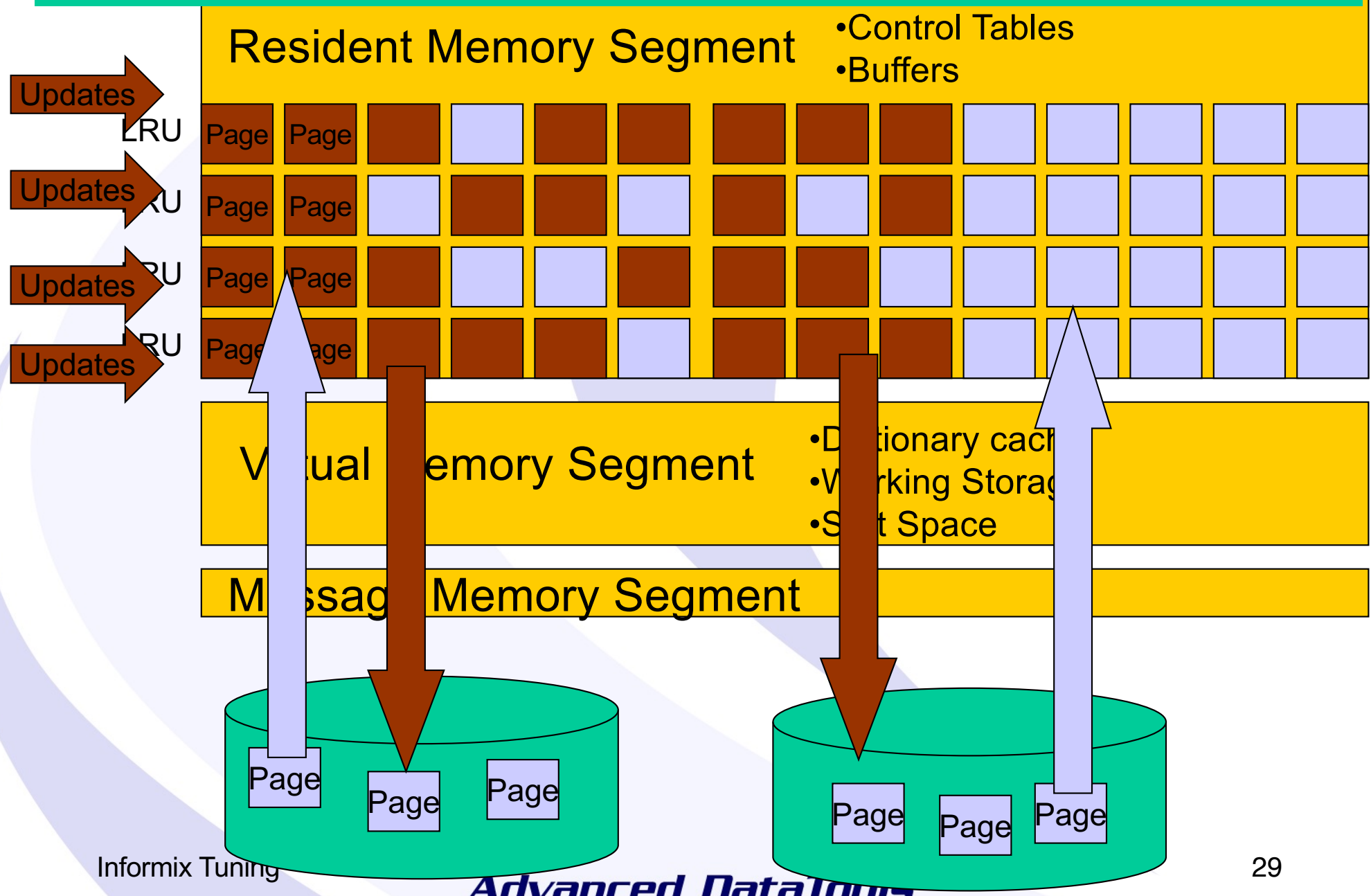
LRU Queues manage writing data to disk in the background when there is idle time, based on LRU\_MAX\_DIRTY and LRU\_MIN\_DIRTY ONCONFIG values.



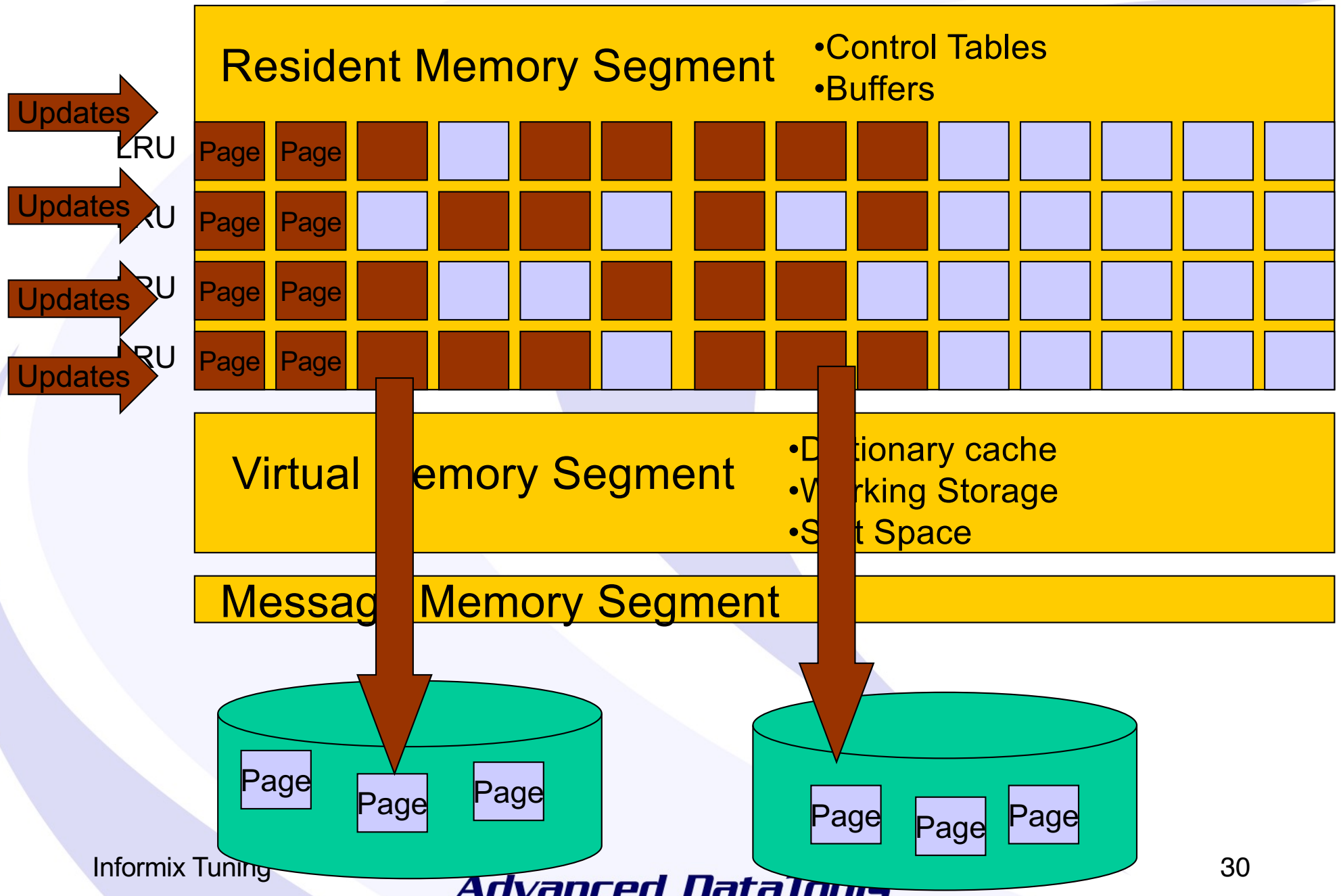
When all buffers are Dirty, the server must STOP all processing and perform a Foreground Write.



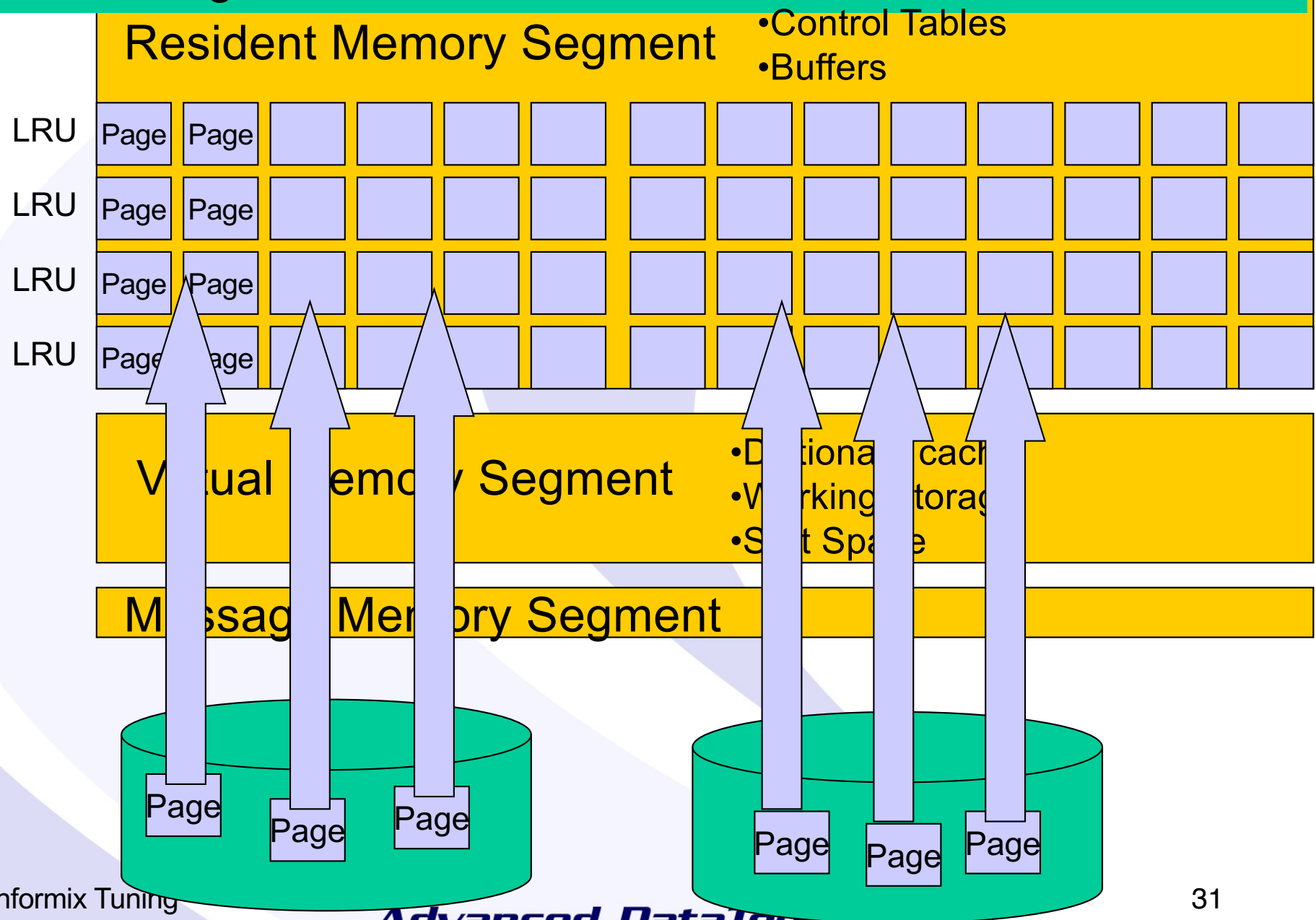
When a buffer is written to disk, it is marked as clean and may be discarded if needed.



# Checkpoint writes all Dirty Buffers to Disk.



After a Checkpoint, all Buffers are clean and the cycle starts over again.



# BUFFERPOOL Best Practices

- Biggest performance gain is to have enough Buffers to hold as many pages as possible in Memory
- ***More Buffers = the better and faster your database will perform***
- Goal is to put all the active data into Memory
- Goal is to prevent high Memory Buffers Turnover (Art Kagel's rule – less than 8 times and hour)



# Monitoring BUFFERPOOL Turnover

```
-- Module: @(#)buff_btr_ratio.sql      2.0      Date: 2013/04/10
-- Author: Lester Knutsen  Email: lester@advancedatools.com
--       Advanced DataTools Corporation
-- Discription: Display Buffer Turnovers per hour
--           Based on Art Kagels performance tuning tip on monitoring
--           how much buffer churn your server has.
--           Goal is BTR of less then 7 times per hour
--           Tested with Informix 11.70 and Informix 12.10
```

```

select
    bufsize,
    pagreads,
    bufwrites,
    nbuffs,
    ((( pagreads + bufwrites ) /nbuffs ) /
      ( select (ROUND ((( sh_curtime - sh_pfclrtype)/60)/60) )
        from sysshmvals ) ) BTR
from sysbufpool;
```

# 4 - One SHMVIRT Segment

## SHMVIRTSIZE Best Practices

- Controls the size of the Informix Virtual Memory Workspace, which can grow if needed.
- Best practice is to set it large enough so it does not need to grow.
  - Monitor with onstat -g seg

# Informix Shared Memory

## onstat -g seg

```
informix@train6:~ train6 > onstat -g seg
```

```
IBM Informix Dynamic Server Version 12.10.FC4 -- On-Line -- Up 21:28:30 -- 774588 Kbytes
```

Segment Summary:

id	key	addr	size	ovhd	class	blkused	blkfree
9109513	525c4801	44000000	4902912	490568	R	1197	0
9142282	525c4802	444ad000	33439744	393384	V	8155	9
9175051	525c4803	46491000	562749440	1	B	137390	0
9207820	525c4804	67d3f000	166359040	1	B	40615	0
9240589	525c4805	71be6000	561152	7848	M	136	1
9469966	525c4806	71c6f000	8388608	99720	V	2045	3
9764879	525c4807	7246f000	8388608	99720	V	1752	296
11829264	525c4808	72c6f000	8388608	99720	V	25	2023
Total:	-	-	793178112	-	-	191315	2332

# Additional Memory Setting

- **RESIDENT** - Controls whether shared memory is resident. Acceptable values are:
  - 0 off (default)
  - 1 lock the resident segment only
  - n lock the resident segment and the next n-1 virtual segments, where  $n < 100$
  - -1 lock all resident and virtual segments
- **SHMADD** - The size, in KB, of additional virtual shared memory segments

# 5 - One LOCKS Segment

- LOCKS in ONCONFIG = the number of LOCKS when Informix starts
- Determines the amount of Memory initially set for LOCKS
- Server will dynamically add as needed
- Dynamically adding LOCKS can cause a performance degradation

# LOCKS Memory Settings

- To monitor, look at the last line of:
  - onstat -k

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 02:34:23 -- 165016 Kbytes

Locks
address      wtlist      owner      lklist      type      tblsnum  rowid
44199028     0           44cd4668   0           S         100002   204
4423f068     0           44cd4f28   0           S         100002   204
442e50a8     0           44cd57e8   0           S         100002   204
442e5130     0           44cd57e8   442e50a8   HDR+S     100002   201
4438b0e8     0           44cd60a8   0           HDR+S     100002   204
5 active, 80000 total, 16384 hash buckets 2 lock table overflows
```

- This shows 2 lock table overflows
- This system requires 80,000 locks



# LOCK Best Practices

- LOCK Table Overflows will slow performance and should be avoided
- LOCK Table Overflows are a major contributor to SHMVIRT Memory additions
- Set your LOCK setting to a value that is the largest number required

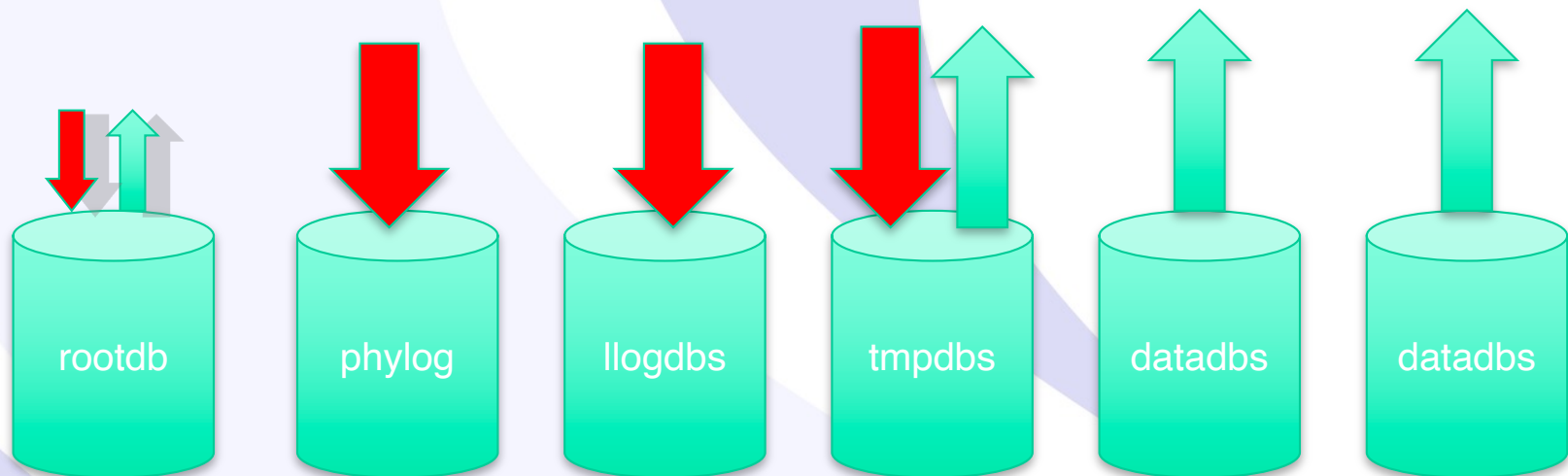
# 6 - Three or more Temp Dbspaces

- Most Disk Writes will be to Temp Space
- Create at least three Temp Dbspaces and Informix will automatically load balance between the temp dbspaces
- Example:
  - DBSPACETEMP tmp1dbs:tmp2dbs:tmp3dbs



# Database Disk I/O

- Most Reads are from Data and Tables
- Writes will be split between Physical Log, Logical Log, Temp, and Data



# Improving Sorts

- Avoiding Sorts
  - Improve indexes on tables
  - However, sorts consume less resources than reading the index when the number of index pages is large
- Narrow columns require less work to sort
- Reduce the data to sort - only select the rows you need

# Memory Required by Sorts

- The default memory used for a sort is 256KB
- If the sort takes more than 256KB of memory, the sort will go to disk
- ONCONFIG parameter:
  - DS\_NONPDQ\_QUERY\_MEM
    - The amount of memory given to sorts who have a PDQ of 0
    - Minimum Value 128KB
    - Maximum Value 25% of DS\_TOTAL MEMORY
    - onmode -wf DS\_NONPDQ\_QUERY\_MEM=1024

# Disk Used by Sorts

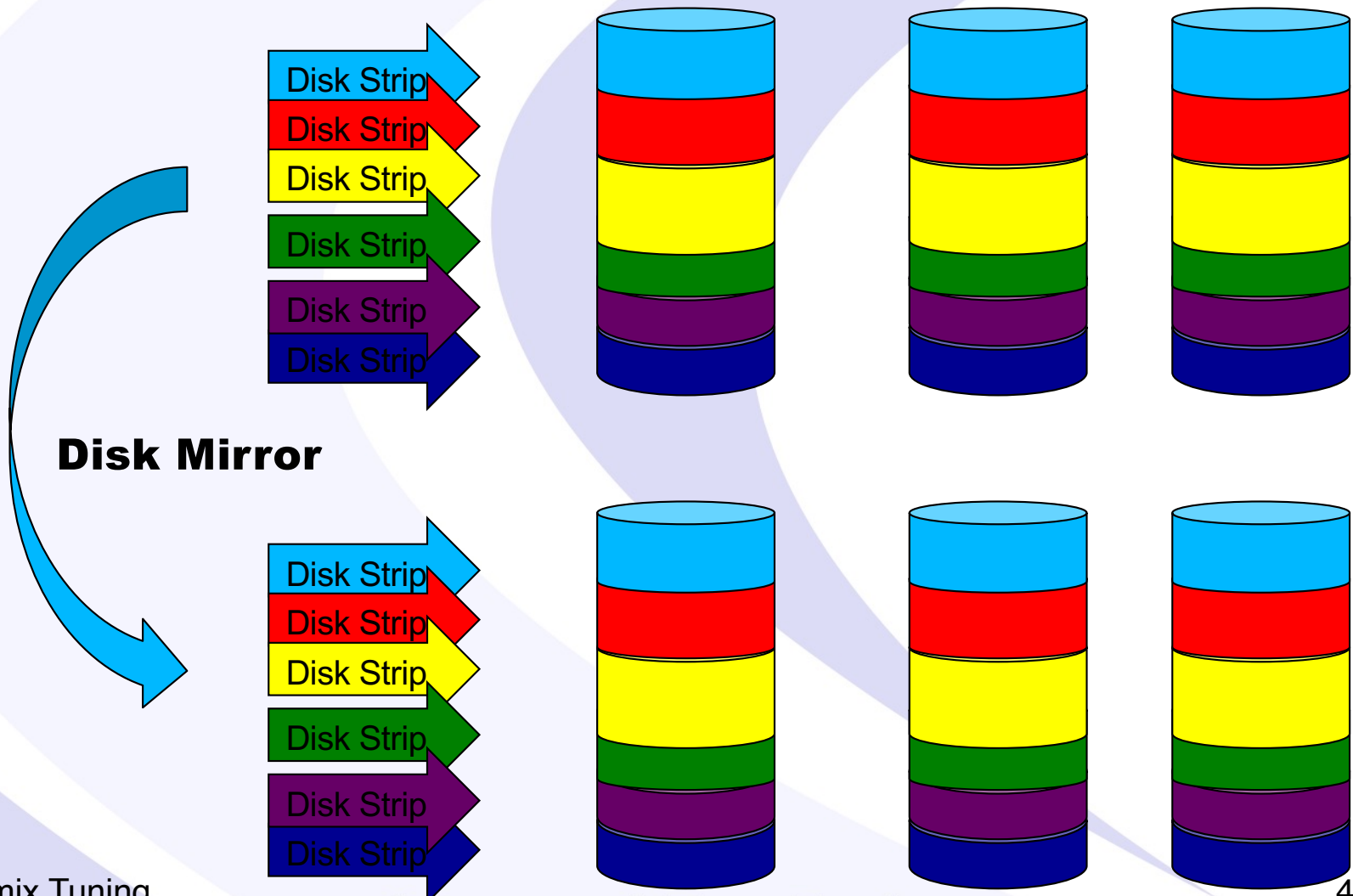
- DBSPACETEMP as defined in Onconfig
- Rootdbs
  - When the database is logged and DBSPACETEMP is a non-logged dbspace
  - When no DBSPACETEMP is defined
- /tmp
  - When PSORT\_DBTEMP or DBSPACETEMP is not set
- PSORT\_DBTEMP
  - Best performance for sorts – several file systems

# 7 - Balance Disk IO

## LRUS, KAIO or DIRECT\_IO

- More disks are better
- Spread the disk I/O across as many disks as possible
- RAID 10 – Stripes and mirrors the data across many disks
- Test DIRECT\_IO

# RAID 10 Configuration



# Move Physical and Logical Logs to Separate DBspaces

- Most disk writes will be to the Physical and Logical Logs
- Move the Physical and Logical Logs to separate dbspaces
- Separate disks are better!

# KAIO and RAW Disks

- Using RAW Disks can be up to 25% faster than using Cooked Disks
- If your operating system supports it, turn on KAIO
- See the Informix Release Notes for your machine on how to set up KAIO
- Test DIRECTIO options



# AIO Oninit Best Practices

- Default is
  - AUTO\_AIOVPS 1 – enable automatically adding AIO VPs as needed
  - This can lead to ***too many*** AIO VPs writing to the same disk system
- Recommended
  - AUTO\_AIOVPS 0
  - VPCLASS aio,num=<number of oninit's you need to write to disk>

# AIO Oninit Best Practices

- How many AIO Class Oninit's do you need? Test, Test, Test...
  - With KAIO or DIRECT\_IO on – only need 2 AIO oninit's
  - With KAIO off (default), it depends on how many processes can write to a disk at the same time
  - Never need more than twice the number of active chunks
  - Most disks can handle up to 8 processes writing

# LRU Writes – Key To Reducing Checkpoints

- LRU Writes are background writes with low overhead
- To reduce the time of a Checkpoint increase your LRU Writes
- Lower the settings for:
  - lrus=<Number of LRU QUEUES>,
  - lru\_max\_dirty=<Percent dirty to START cleaning>
  - lru\_min\_dirty=<Percent dirty to STOP cleaning>

# LRU Best Practices

- Enable AUTO\_LRU\_TUNING for turnkey or embedded systems
- Disable AUTO\_LRU\_TUNING for high performance systems where you do not want CHECKPOINTS to write a huge amount of data to disk and slow everything down

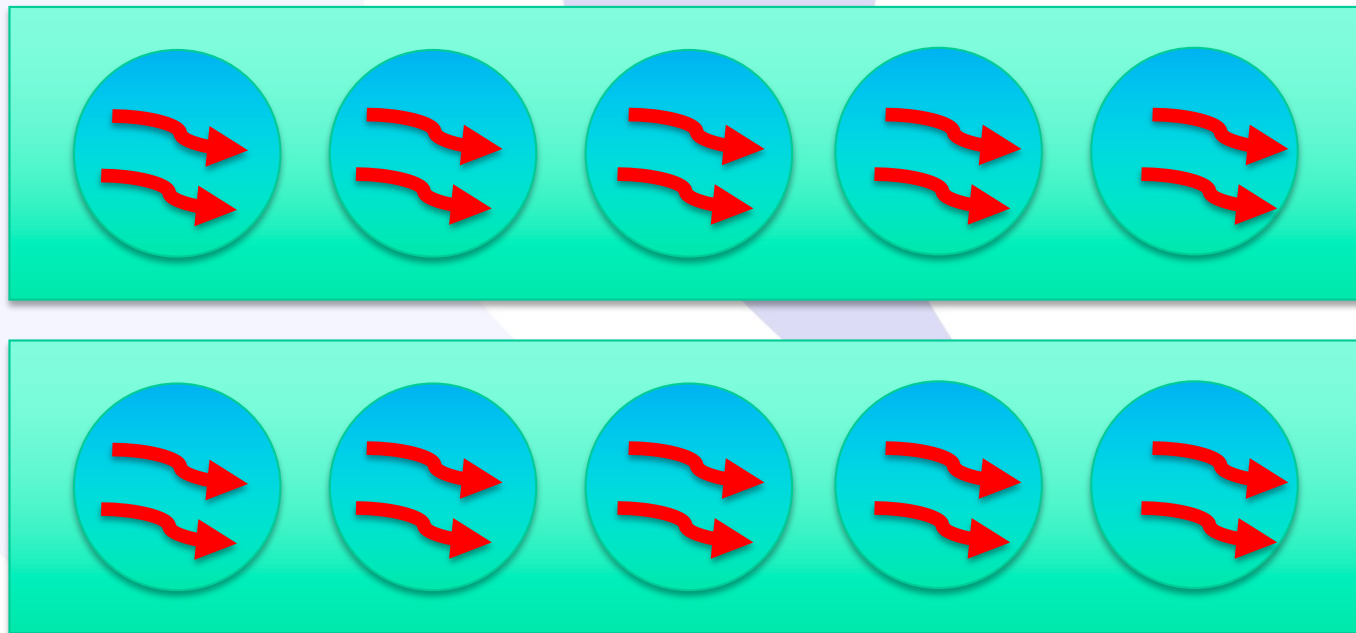
# 8 - Use Your CPUVPs

## CPU Terms

- Socket = One Chip or Processor
- Cores per Socket = How many cores run on a chip? A core only runs one process at a time.
- Hyper-Threads or SMT threads per core = Many cores have the ability to run multiple threads. No matter how many threads run on a core, only one thread can run at a time on a core. Hyper-Threads will appear as additional Virtual Cores.
- Chip speed is measured in gigahertz (GHz); this is the speed of a single core of your processor.
- PVU - IBM Processor Value Unit = A unit of measure used to differentiate licensing of software

# CPU Terms

- Example: 2 Sockets with 5 Cores and 2 Hyper-Threads per Core = 10 Cores and 20 Virtual Cores
- **20 Virtual Cores can ONLY run 10 processes at the same time**



# Informix CPU Best Practices

- How many Cores will be allocated for Informix? What else is running on the machine?
- Traditional best practice is number of physical CPU Cores minus 1
- Current CPU Cores are fast enough to handle 2-3 oninit per Core or 1 oninit per 1000 MHz



# CPU Usage Best Practices

- How busy are your CPUs?
- Tools to monitor:
  - `sar -u`, `vmstat`, `mpstat`, `top`, `prstat`
- Performance Guideline for Average CPU Usage:
  - < 30 % - Good
  - 30-60% - Fair
  - > 60% - Poor
- ***Save 60% of your CPU usage to handle Workload Spikes or you may not be able to handle busy loads***

# Hyper-Threads or SMT Threads Best Practices

- Test, Test and Test again; don't assume that more Hyper-Threads or SMT threads are better. Your workload will determine what is best.
- AIX – Try 2 SMT threads per Core on Power6 and Power7, 4 SMT on Power8
- Intel – Try 2 Oninitis per Core instead of 2 Hyper-Threads or 1 Oninit per Hyper-Thread

# Oninit Process

## onstat -g sch

```
informix@train6:~ train6 > onstat -g sch
```

```
IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line -- Up 00:02:12 -- 766404 Kbytes
```

```
VP Scheduler Statistics:
```

vp	pid	class	semops	busy waits	spins/wait	bsy lspins
1	22472	cpu	141	0	0	0
2	22473	adm	0	0	0	0
3	22474	lio	4115	0	0	0
4	22475	pio	30	0	0	0
5	22476	aio	7453	0	0	0
6	22477	msc	5	0	0	0
7	22478	fifo	2	0	0	0
8	22479	soc	2	0	0	0
9	22480	aio	2890	0	0	0
10	22481	aio	187	0	0	0
11	22482	aio	113	0	0	0
12	22483	aio	55	0	0	0
13	22484	aio	58	0	0	0
14	22485	aio	41	0	0	0
15	22486	aio	32	0	0	0
16	22487	aio	29	0	0	0
17	22489	aio	22	0	0	0

# Oninit Process Automatically Started

- Started Automatically
  - PIO - Handles physical log file when cooked disk space is used
  - LIO - Handles logical log file when cooked disk space is used
  - FIFO - Performs FIFO operations
  - ADM - Executes administrative threads
  - MSC - Handles request for system calls
- Started when Auditing is on
  - ADT - Executes auditing threads
- Started when UDRs are called
  - Java
  - User Defined Functions

# Oninit Process Controlled by VPCLASS

- ONCONFIG VPCLASS Setting
  - CPU - Executes all user and session threads and some system threads
  - AIO - Handles disk I/O

# CPU Oninit Configuration

- VPCCLASS CPU – Configure the number of Oninit CPU VPs to start for Informix
  - VPCCLASS cpu,num=<number> [,max=<max number cpu>] [,aff=<single CPU number> | <start cpu>-<end cpu> | ( <start cpu>-<end cpu>/<skip amount> ) ] ] [,noage]
- Examples for 8 Core machine:
  - VPCCLASS cpu,num=4,noage
  - VPCCLASS cpu,num=8,noage
  - VPCCLASS cpu,num=8,aff=0,noage
  - VPCCLASS cpu,num=8,aff=1-4,noage

# Additional CPU Best Practices

- Set MULTIPROCESSOR to 1 (Almost all machines today are multiprocessor)
- Set SINGLE\_CPU\_VP to 0 (Allows you to run more Oninits of CPU class as needed)
- Set NOAGE if your OS supports it



# Additional CPU Best Practices

- Set VP\_MEMORY\_CACHE\_KB <size in KB for private cache for each CPU VP>
- Format is: <size>[,DYNAMIC|STATIC]  
Acceptable values for <size> are: 0 (disable) or 800 through 40% of the value of SHMTOTAL
- Example:
  - VP\_MEMORY\_CACHE\_KB 4096

# 9 - Row Level Locking

- Informix defaults to creating tables with Page Level Locking
- For OLTP systems, make sure you are using ROW Level Locking
- Find Tables with Page Level Locking

```
select tabname, locklevel  
from systables  
where tabid > 99  
and locklevel = "P"  
and tabtype = "T"
```

# 10 - Update Statistics

**Why is Update Statistics important?**

- **Update Statistics collects metrics to enable the server to run your SQL faster**
- One of the most important tasks for a DBA
- Informix has Automated Update Statistics built into the server, but you still need to monitor it

# When Did AUS Last Run?

```
-- #####
-- ## Module: @(#)aus_last_run.sql      2.0      Date: 01/01/2018
-- ## Author: Lester Knutsen  Email: lester@advancedatatools.com
-- ##          Advanced DataTools Corporation
-- #####
-- ## Show when Auto Update Status (AUS) was last run for a server

-- unload to "aus_last_run.uld"
select
    substr(b.name,1,20) as db,
    substr(c.tabname,1,20) as table,
    aus_cmd_type as level,
    aus_cmd_time as when
from
    sysadmin:aus_command a,
    sysmaster:sytdatabases b,
    sysmaster:systabnames c
where  a.aus_cmd_dbs_partnum = b.partnum
and a.aus_cmd_partnum = c.partnum
and b.name not matches "sys*"
and c.tabname not matches "sys*"
order by aus_cmd_id;
```

# When Did AUS Last Run?

```
----- benchmark3@train1 ----- Press CTRL-W for Hel
```

db	table	level	when
benchmark1	state	l	2019-09-24 01:01:16
benchmark1	state	H	2019-09-24 01:01:16
benchmark1	benchmark	l	2019-09-24 01:01:16
benchmark1	benchmark	H	2019-09-24 01:01:16
benchmark2	state	l	2019-09-24 01:01:21
benchmark2	state	H	2019-09-24 01:01:21
benchmark2	customer	l	2019-09-24 01:01:21
benchmark2	customer	H	2019-09-24 01:01:21
benchmark2	product	l	2019-09-24 01:01:21
benchmark2	product	H	2019-09-24 01:01:21
benchmark2	bills	l	2019-09-24 01:01:21
benchmark2	bills	H	2019-09-24 01:01:21



# Update Stats Info by Database

```
-- #####
-- ## Module: @(#)updstats_info.sql      2.0      Date: 01/01/2018
-- ## Author: Lester Knutsen  Email: lester@advancedatools.com
-- ##          Advanced DataTools Corporation
-- #####
-- ## Show when update status for last run for a table or columns
-- ## in the current database

select  t.tabname as tabname,
        t.ustlowts as low_update,
        c.colname as column,
        d.constructed as upddate,
        d.mode as mode,
        d.constr_time as updtype,
        d.ustbuildduration as updduration,
        max(d.seqno) as maxseqno
from    systables t, outer ( sysdistrib d, syscolumns c )
where   t.tabid > 99
and     t.tabtype = "T"
and     t.tabid = d.tabid
and     d.tabid = c.tabid
and     d.colno = c.colno
group by 1,2,3,4,5,6,7 --,8,9, 10
order by 1, 3
```

# Update Stats Info by Database

```
tabname      bills
low_update   2022-02-28 12:56:52.00000
column       bill_number
update       02/28/2022
mode         H
uptime       2022-02-28 12:56:52.00000
updduration   0:00:00.00000
maxseqno     5

tabname      customer
low_update   2022-02-12 01:11:02.00000
column       customer_number
update       02/12/2022
mode         H
uptime       2022-02-12 01:11:02.00000
updduration   0:00:00.02167
maxseqno     9

tabname      product
low_update   2022-02-12 01:11:02.00000
column       product_number
update       02/12/2022
mode         H
uptime       2022-02-12 01:11:02.00000
updduration   0:00:00.00001
maxseqno     1
```



# Document ONCONFIG Changes - onconfig\_diff

- See changes in your ONCONFIG file:

usage: onconfig\_diff [-d] | [-c -f file\_name\_1 -s file\_name\_2]

-d	compare current onconfig to defaults
-c	compare file to another file
-f	file name
-s	second file name

- Compare to Another ONCONFIG:

onconfig\_diff -c -f \$INFORMIXDIR/etc/onconfig.std -s \$INFORMIXDIR/etc/\$ONCONFIG

- Compare to Defaults

onconfig\_diff -d

# Additional Information

- **Configuring a New Informix Server (ONCONFIG)**
  - <https://advancedatools.com/webcasts/informix-tutorial-configuring-a-new-informix-server/>
- **Managing Informix Disk Space**
  - <https://advancedatools.com/webcasts/informix-tutorial-managing-informix-disk-space/>
- **Informix Database Indexes**
  - <https://advancedatools.com/webcasts/informix-webcast-2019-database-indexes-best-practices-for-informix-dbas/>
- **Informix Update Statistics**
  - <https://advancedatools.com/webcasts/informix-webcast-2019-informix-update-statistics-best-practices-for-informix-dbas/>
- **Optimizing the Operating System**
  - <https://advancedatools.com/webcasts/informix-webcast-2018-optimizing-the-operating-system-for-the-best-informix-database-performance/>

# Informix DBA Training

## From the Informix IBM Champions

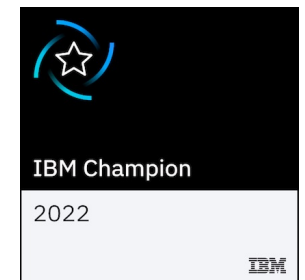


- **Informix for Database Administrators Training - May 9-12, 2022**
- **Advanced Informix Performance Tuning - May 23-26, 2022**

**More information and registration at:**

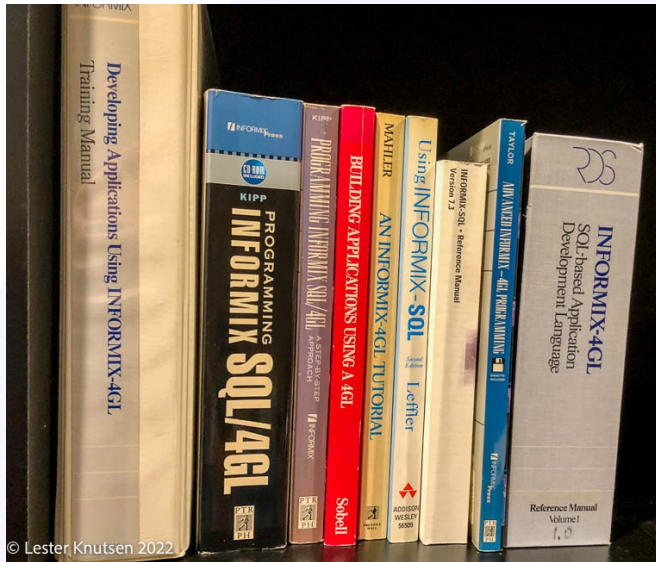
<https://advanceddatatools.com/training/>

- Instructor-led online classes by Zoom
- Four days of hands-on Informix experience
- Separate AWS Informix server for exercises and labs for each student
- Participate in Q&A and collaborative class discussions
- Taught by Lester Knutsen, IBM Informix Lifetime Champion, with over 40 years of Informix experience



# Informix 4GL for Developers

## Free Training coming in April



**Informix 4GL for Developers Course is for programmers developing and debugging Informix 4GL code. The FREE course previews the first eight lessons of the entire 18 lesson course.**

- Self-paced Online course
- Online Video instruction lectures
- Use your own server for the labs
- Participate in Q&A and collaborative class online discussions
- Developed by Lester Knutsen, IBM Informix Lifetime Champion, with over 40 years of Informix experience

**[Contact info@advancedatools.com if you are interested!](mailto:info@advancedatools.com)**

# Questions?



**Please ask your questions in the Chat!**

***Advanced DataTools***



# **Thank You**

## ***Advanced DataTools Corporation***



For more information:

**Lester@advancedatools.com**

**<https://www.advancedatools.com>**

***Advanced DataTools***

# Thank You

## Informix Tech Talks by the IIUG on YouTube

We have launched a new channel on YouTube for Informix Users! Please subscribe to our channel on YouTube to stay informed. This will be a place for Informix how-to videos.

### Subscribe at:

<https://www.youtube.com/c/InformixTechTalksbytheIIUG>







# Coming in April 2022

**All Presentations from IIUG World 2021  
will be available on  
Informix Tech Talks YouTube**

**Subscribe at:**

<https://www.youtube.com/c/InformixTechTalksbytheIIUG>



**International Informix User Group**

**We speak Informix**