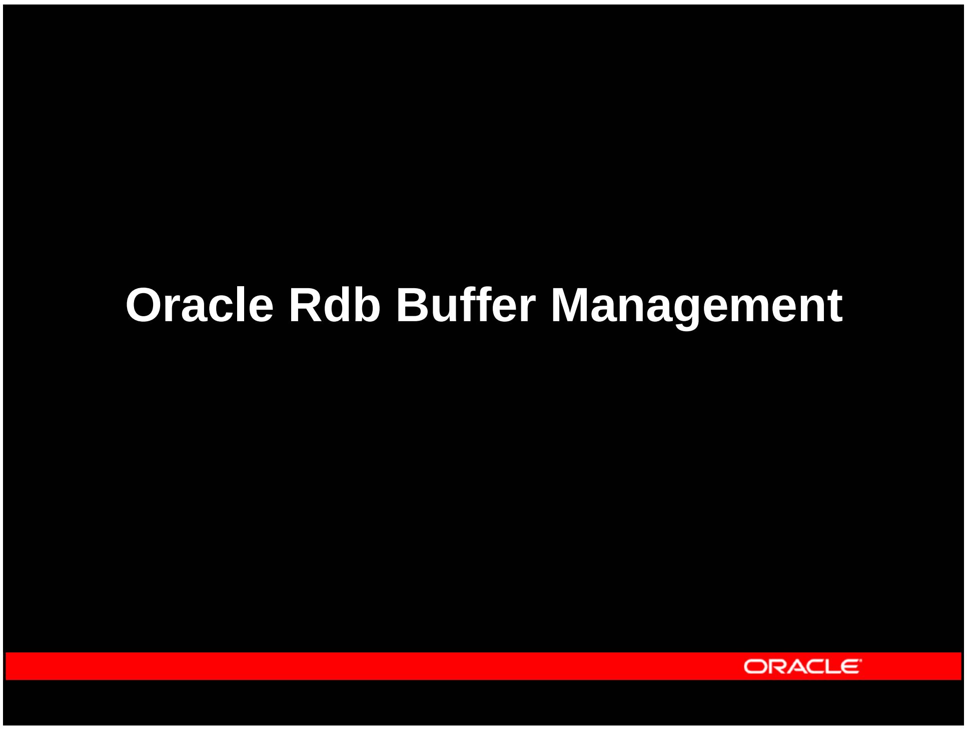
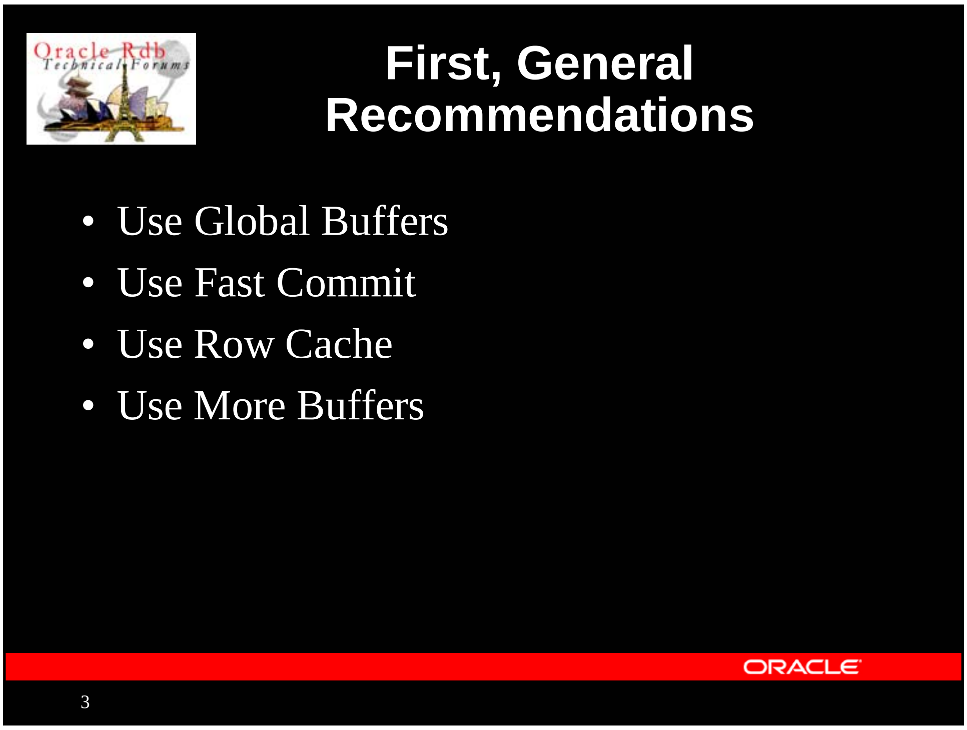
Rdb features for high performance application

Philippe Vigier

Oracle New England Development Center

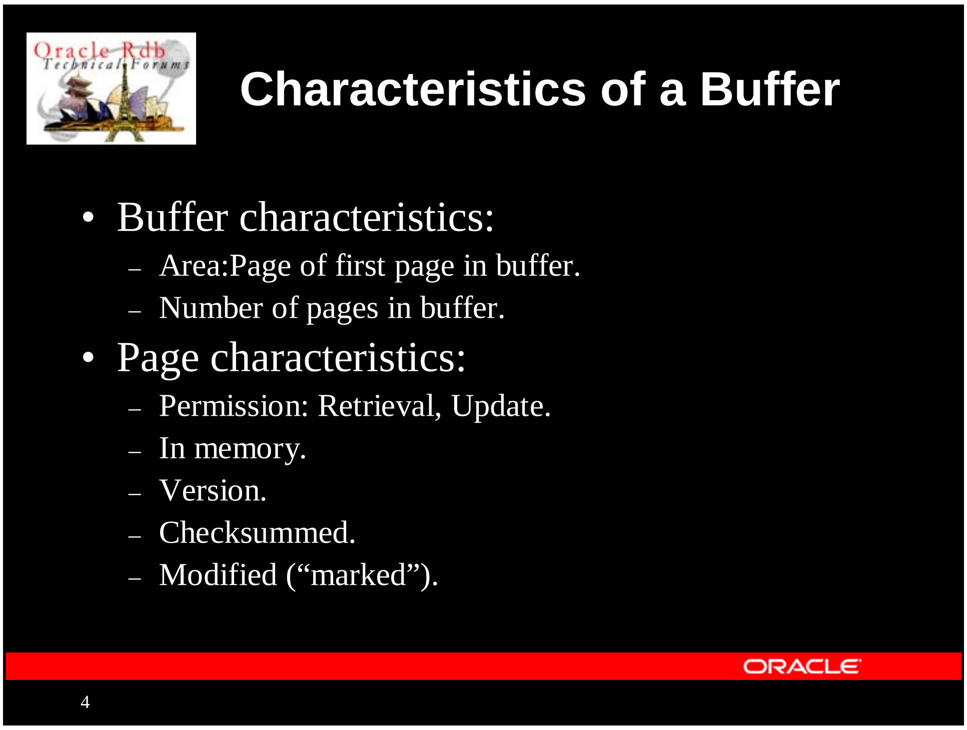


Oracle Rdb Buffer Management



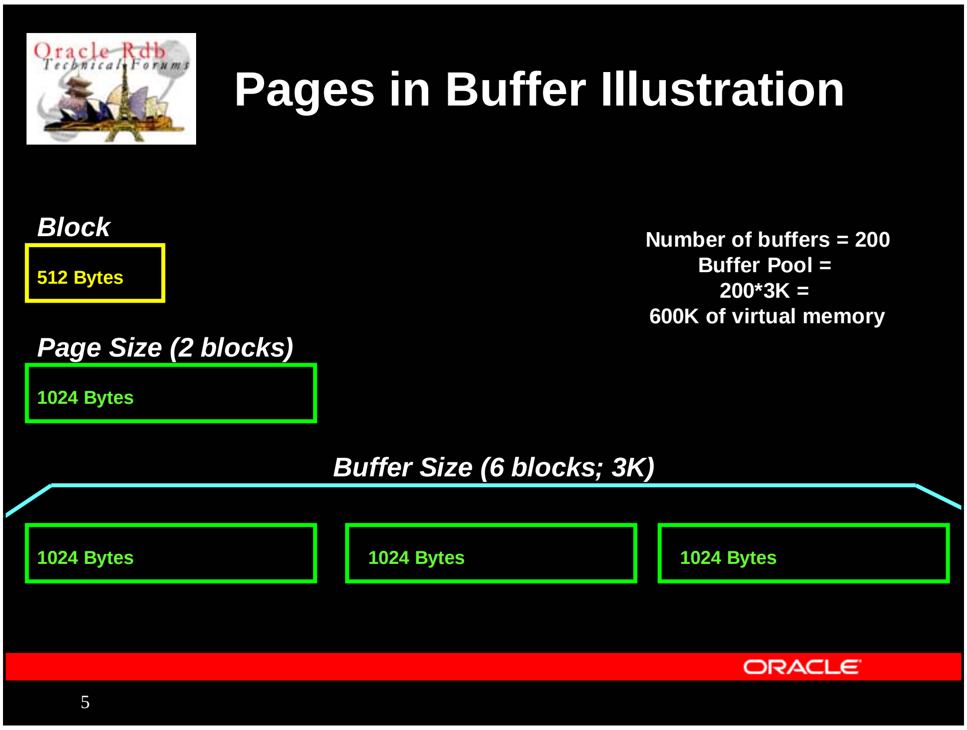
First, General Recommendations

- Use Global Buffers
- Use Fast Commit
- Use Row Cache
- Use More Buffers



Characteristics of a Buffer

- Buffer characteristics:
  - Area:Page of first page in buffer.
  - Number of pages in buffer.
- Page characteristics:
  - Permission: Retrieval, Update.
  - In memory.
  - Version.
  - Checksummed.
  - Modified (“marked”).



Pages in Buffer Illustration

Block

512 Bytes

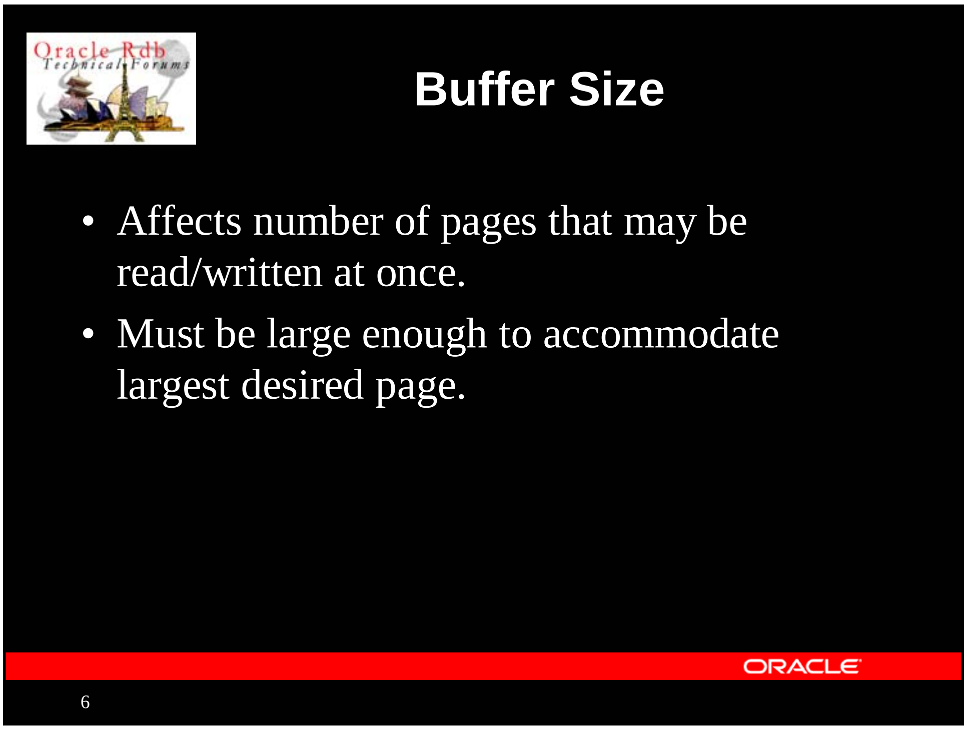
Page Size (2 blocks)

1024 Bytes

Number of buffers = 200
Buffer Pool =
200*3K =
600K of virtual memory

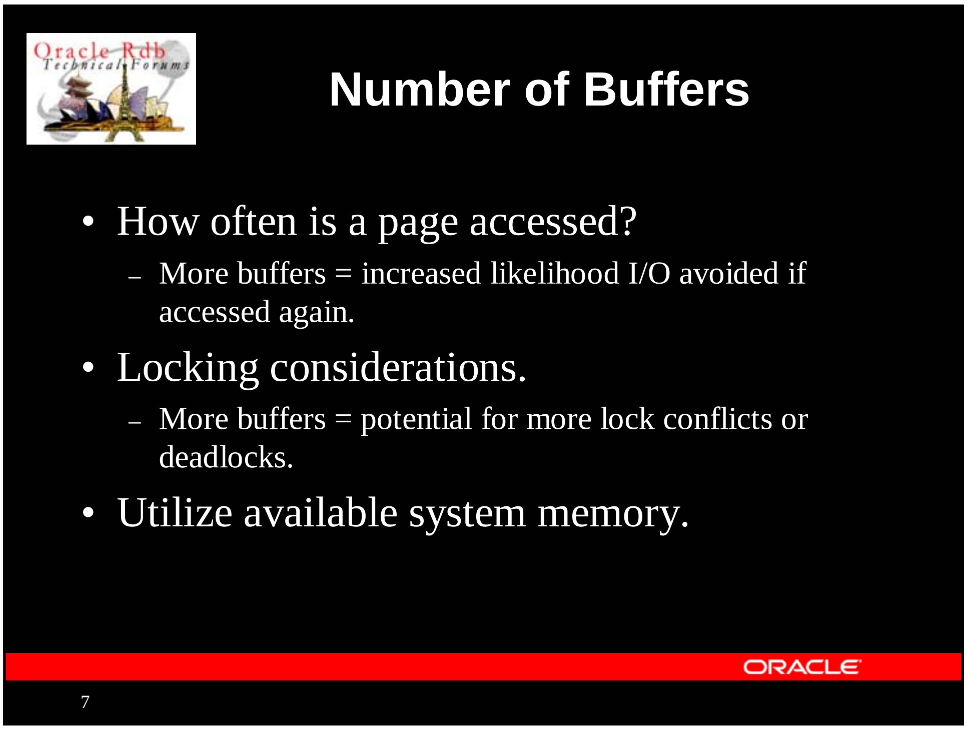
Buffer Size (6 blocks; 3K)

1024 Bytes | 1024 Bytes | 1024 Bytes



Buffer Size

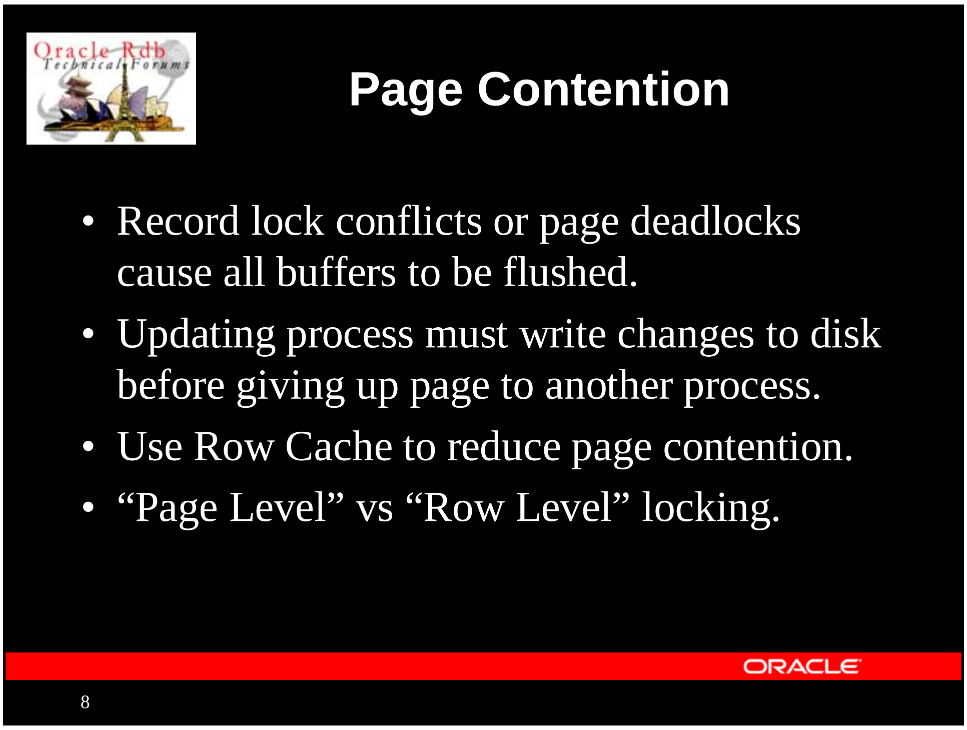
- Affects number of pages that may be read/written at once.
- Must be large enough to accommodate largest desired page.



Number of Buffers

- How often is a page accessed?
  - More buffers = increased likelihood I/O avoided if accessed again.
- Locking considerations.
  - More buffers = potential for more lock conflicts or deadlocks.
- Utilize available system memory.

7



Page Contention

- Record lock conflicts or page deadlocks cause all buffers to be flushed.
- Updating process must write changes to disk before giving up page to another process.
- Use Row Cache to reduce page contention.
- “Page Level” vs “Row Level” locking.

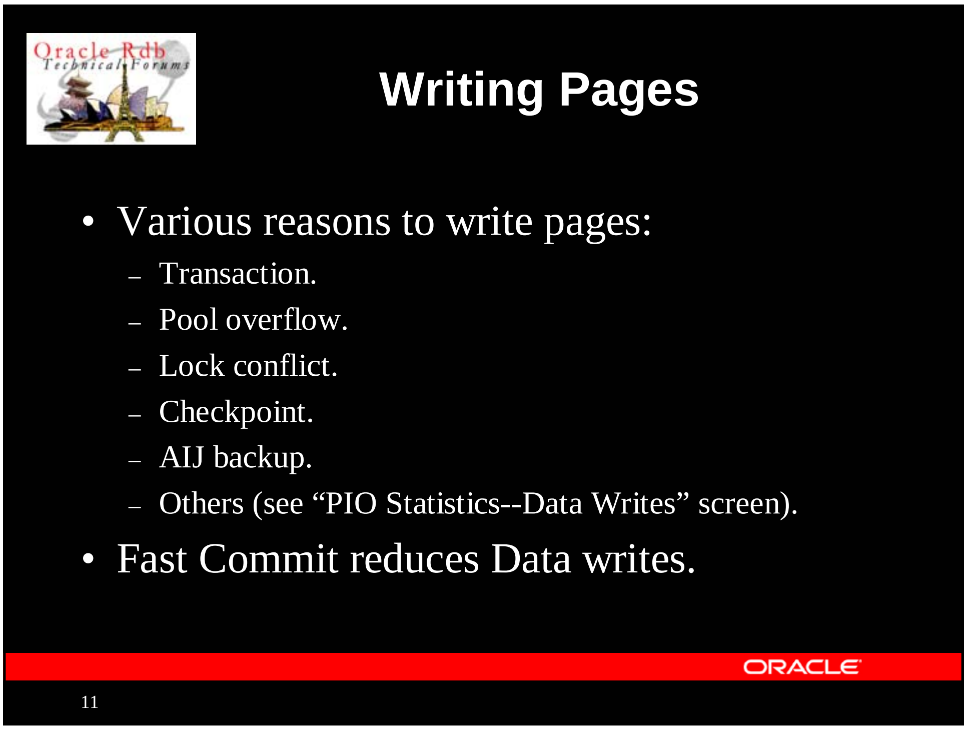
8

Page Buffer I/O

- Rdb attempts to read an entire buffer at one time; write only modified pages.
- I/O overview on first screen displayed by RMU/SHOW STATISTICS.
- Additional information under “IO Statistics Information” group.

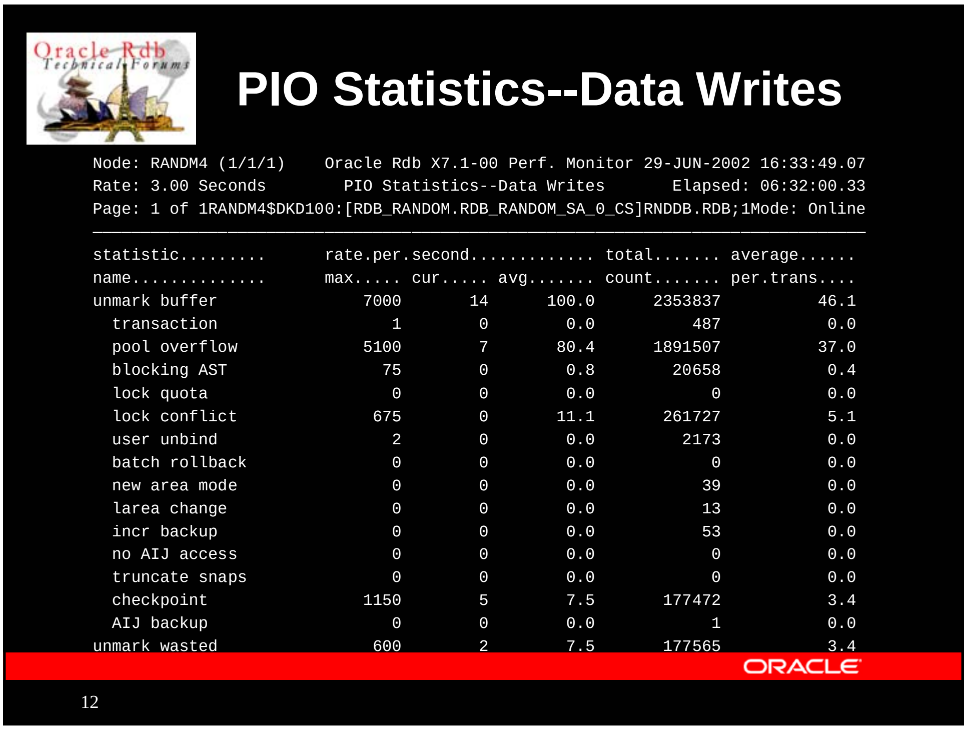
Summary IO Statistics

```
Node: RANDM4 (1/1/1)    Oracle Rdb X7.1-00 Perf. Monitor 29-JUN-2002 16:19:39.49
Rate: 3.00 Seconds           Summary IO Statistics          Elapsed: 06:17:50.75
Page: 1 of 1RANDM4$DKD100:[RDB_RANDOM.RDB_RANDOM_SA_0_CS]RNDDB.RDB;1Mode: Online
--------------------------------------------------------------------------------
statistic.........       rate.per.second............. total....... average......
name..............       max..... cur..... avg....... count....... per.trans....
transactions                    2        2        2.1       49245           1.0
verb successes                136      136       54.9     1245665          25.2
verb failures                   2        2        1.6       36570           0.7

synch data reads              801      801      128.1     2906832          59.0
synch data writes              31       31       17.3      393899           7.9
asynch data reads              45       45       23.5      534584          10.8
asynch data writes            442      442       82.1     1863479          37.8

.
.
.
```



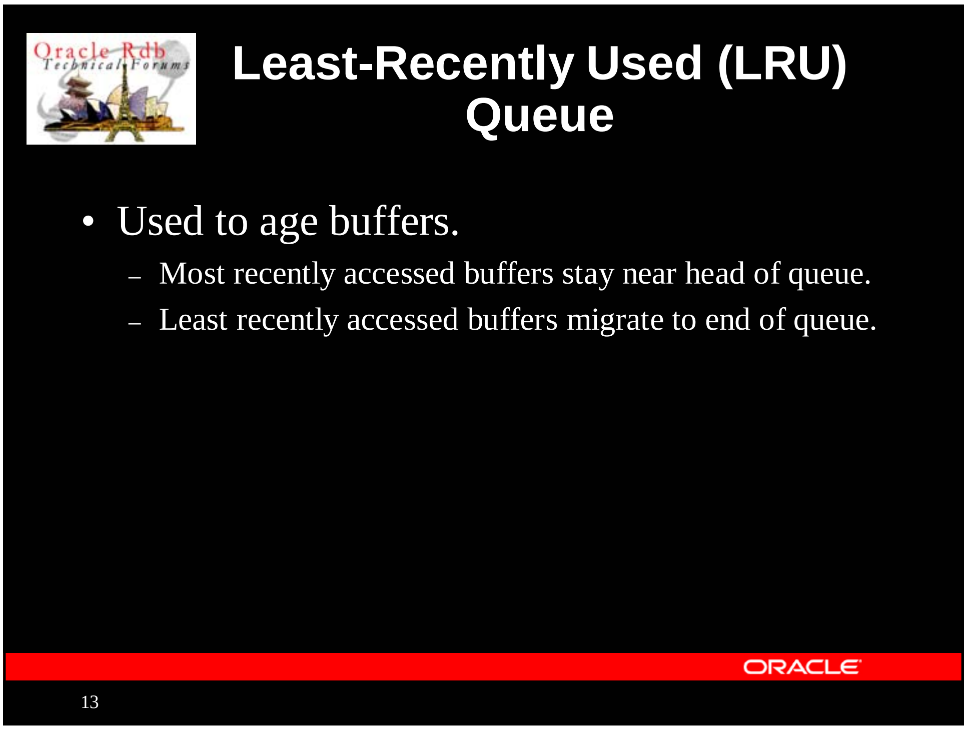
Writing Pages

- Various reasons to write pages:
  - Transaction.
  - Pool overflow.
  - Lock conflict.
  - Checkpoint.
  - AIJ backup.
  - Others (see “PIO Statistics--Data Writes” screen).
- Fast Commit reduces Data writes.



PIO Statistics--Data Writes

```
Node: RANDM4 (1/1/1)    Oracle Rdb X7.1-00 Perf. Monitor 29-JUN-2002 16:33:49.07
Rate: 3.00 Seconds        PIO Statistics--Data Writes       Elapsed: 06:32:00.33
Page: 1 of 1RANDM4$DKD100:[RDB_RANDOM.RDB_RANDOM_SA_0_CS]RNDDB.RDB;1Mode: Online
________________________________________________________________________________
statistic.........      rate.per.second............  total....... average......
name..............      max..... cur..... avg....... count....... per.trans....
unmark buffer                7000       14      100.0      2353837          46.1
  transaction                   1        0        0.0          487           0.0
  pool overflow              5100        7       80.4      1891507          37.0
  blocking AST                 75        0        0.8        20658           0.4
  lock quota                    0        0        0.0            0           0.0
  lock conflict               675        0       11.1       261727           5.1
  user unbind                   2        0        0.0         2173           0.0
  batch rollback                0        0        0.0            0           0.0
  new area mode                 0        0        0.0           39           0.0
  larea change                  0        0        0.0           13           0.0
  incr backup                   0        0        0.0           53           0.0
  no AIJ access                 0        0        0.0            0           0.0
  truncate snaps                0        0        0.0            0           0.0
  checkpoint                 1150        5        7.5       177472           3.4
  AIJ backup                    0        0        0.0            1           0.0
unmark wasted                 600        2        7.5       177565           3.4
```

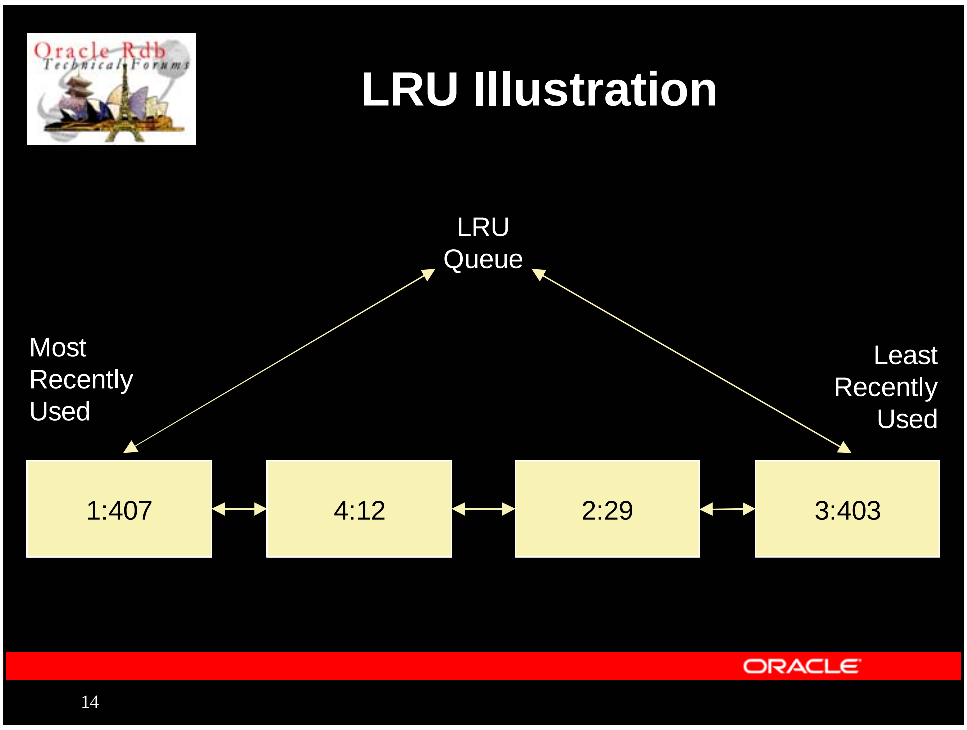


Least-Recently Used (LRU) Queue

- Used to age buffers.
  - Most recently accessed buffers stay near head of queue.
  - Least recently accessed buffers migrate to end of queue.

ORACLE

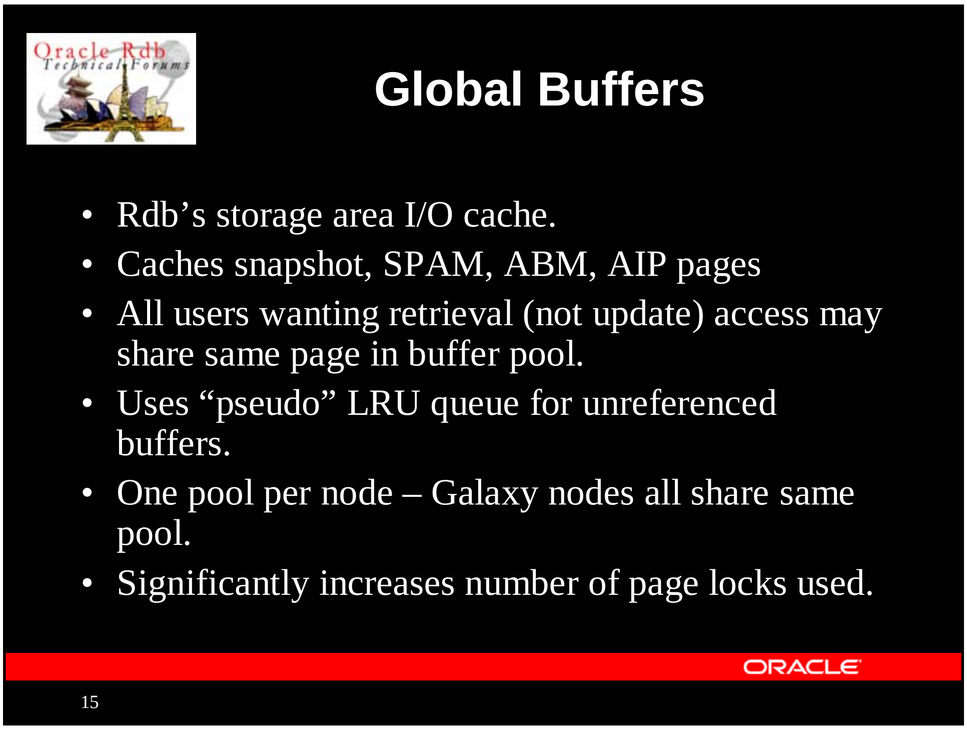
13



LRU Illustration

ORACLE

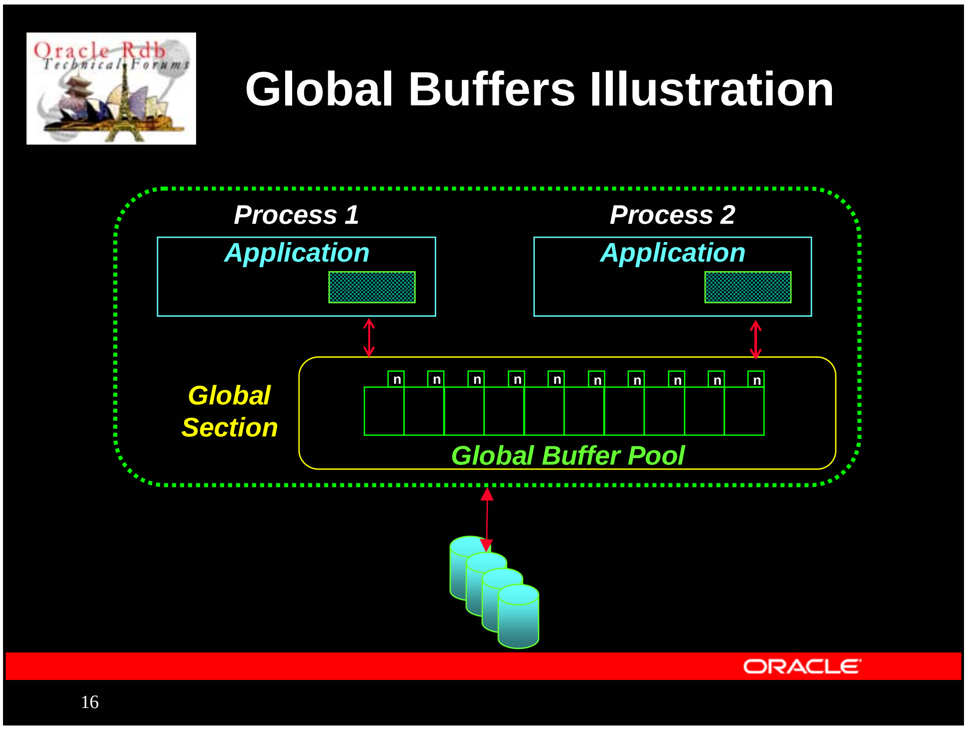
14



Global Buffers

- Rdb’s storage area I/O cache.
- Caches snapshot, SPAM, ABM, AIP pages
- All users wanting retrieval (not update) access may share same page in buffer pool.
- Uses “pseudo” LRU queue for unreferenced buffers.
- One pool per node – Galaxy nodes all share same pool.
- Significantly increases number of page locks used.

ORACLE



Global Buffers Illustration

Process 1

Application

Process 2

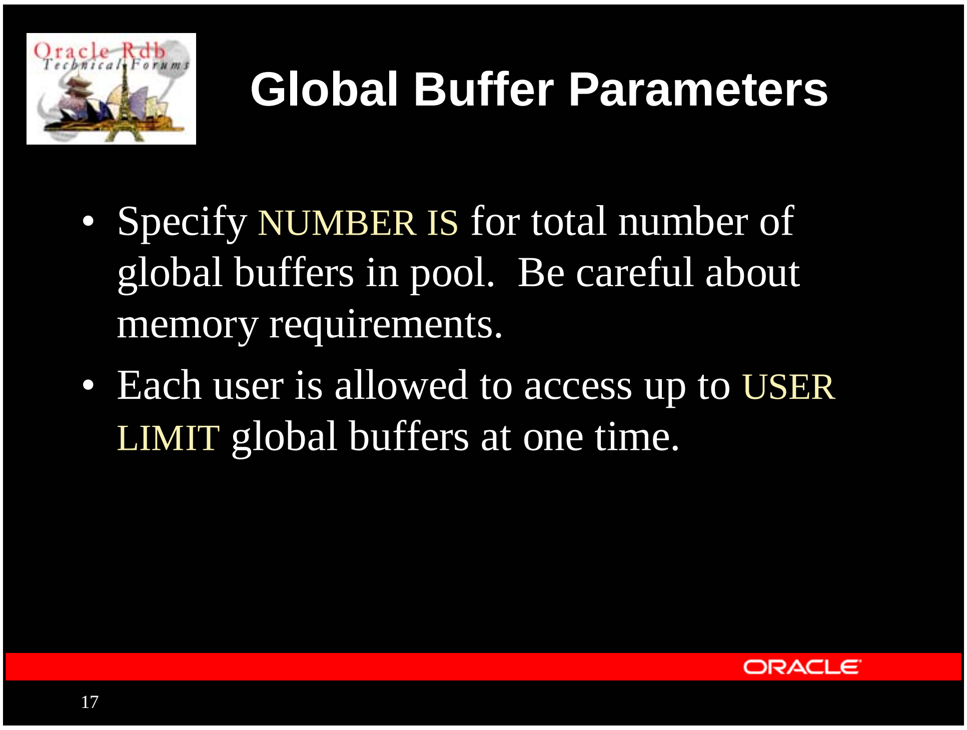
Application

Global Section

n n n n n n n n n n

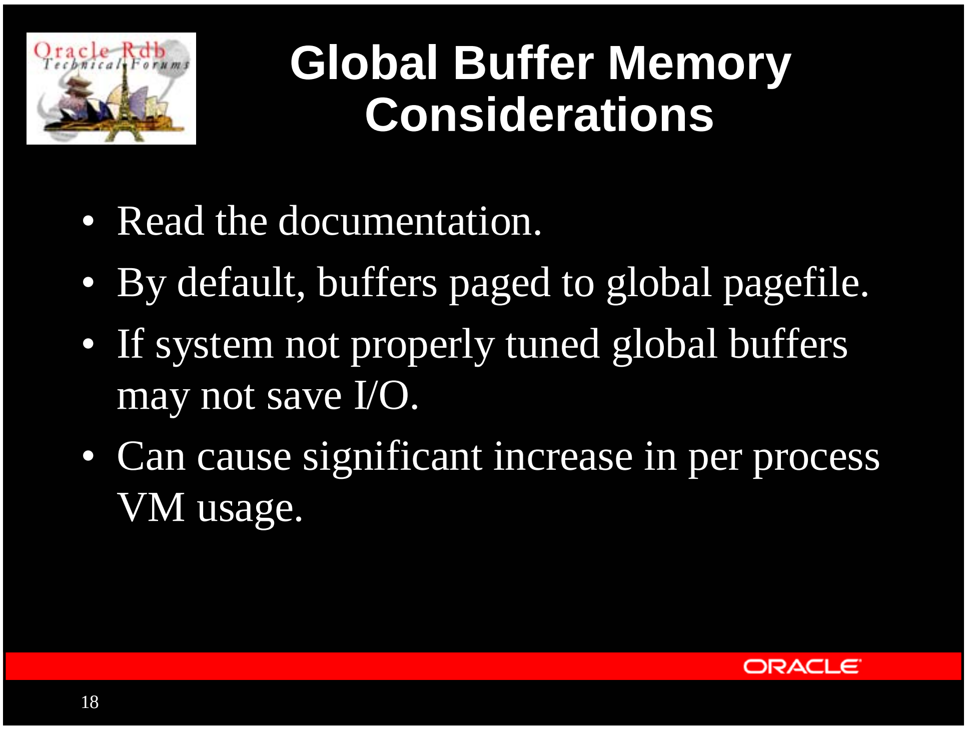
Global Buffer Pool

ORACLE



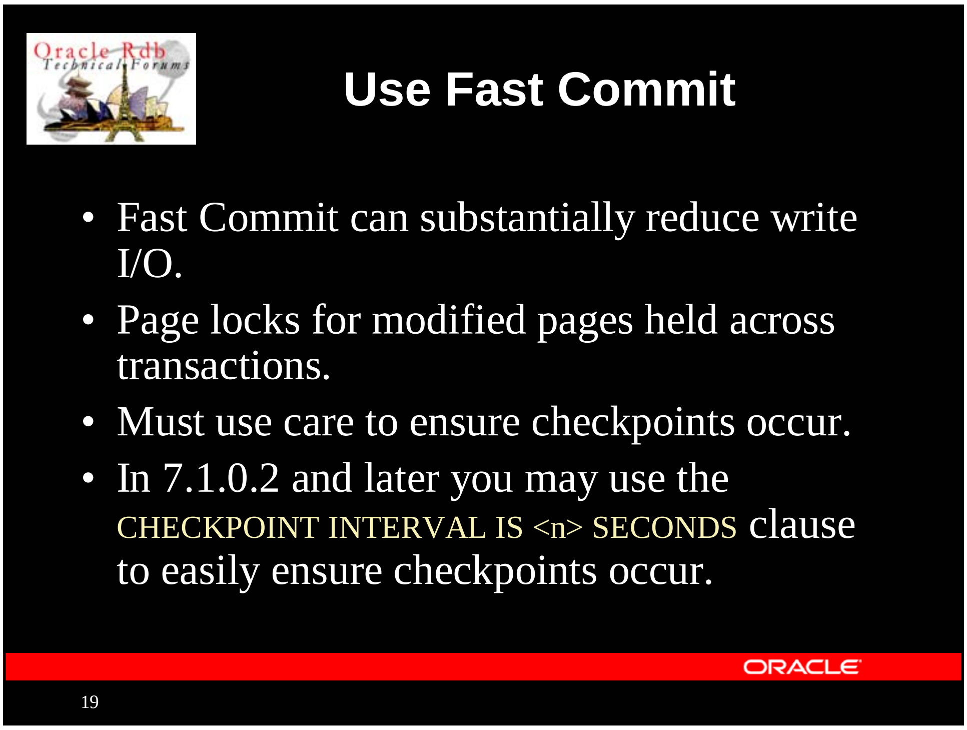
Global Buffer Parameters

- Specify NUMBER IS for total number of global buffers in pool. Be careful about memory requirements.
- Each user is allowed to access up to USER LIMIT global buffers at one time.



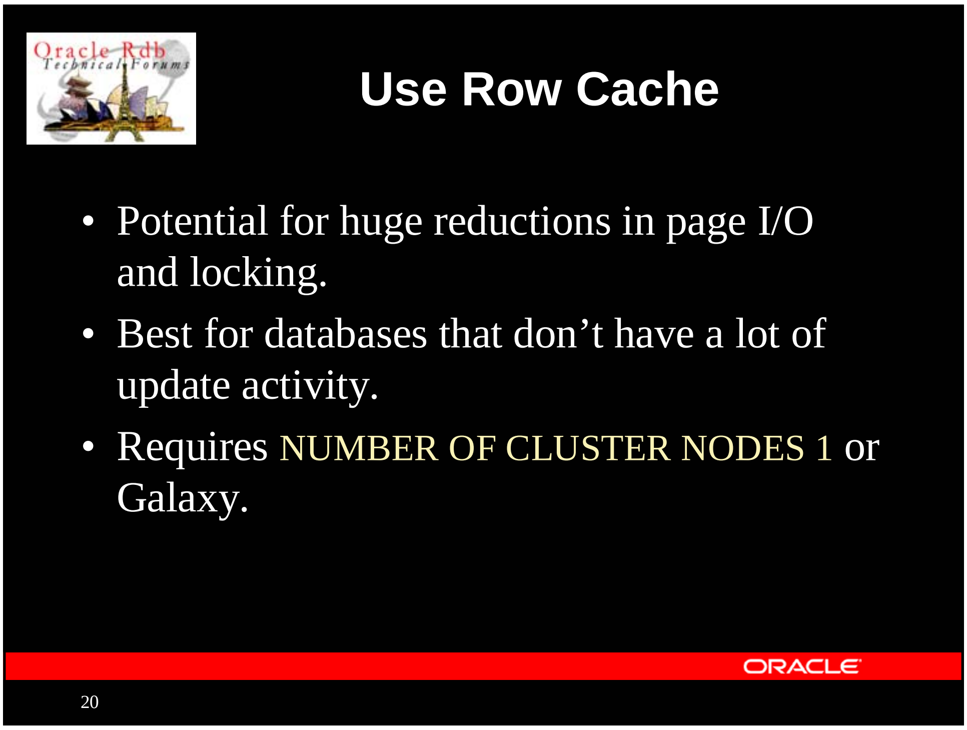
Global Buffer Memory Considerations

- Read the documentation.
- By default, buffers paged to global pagefile.
- If system not properly tuned global buffers may not save I/O.
- Can cause significant increase in per process VM usage.



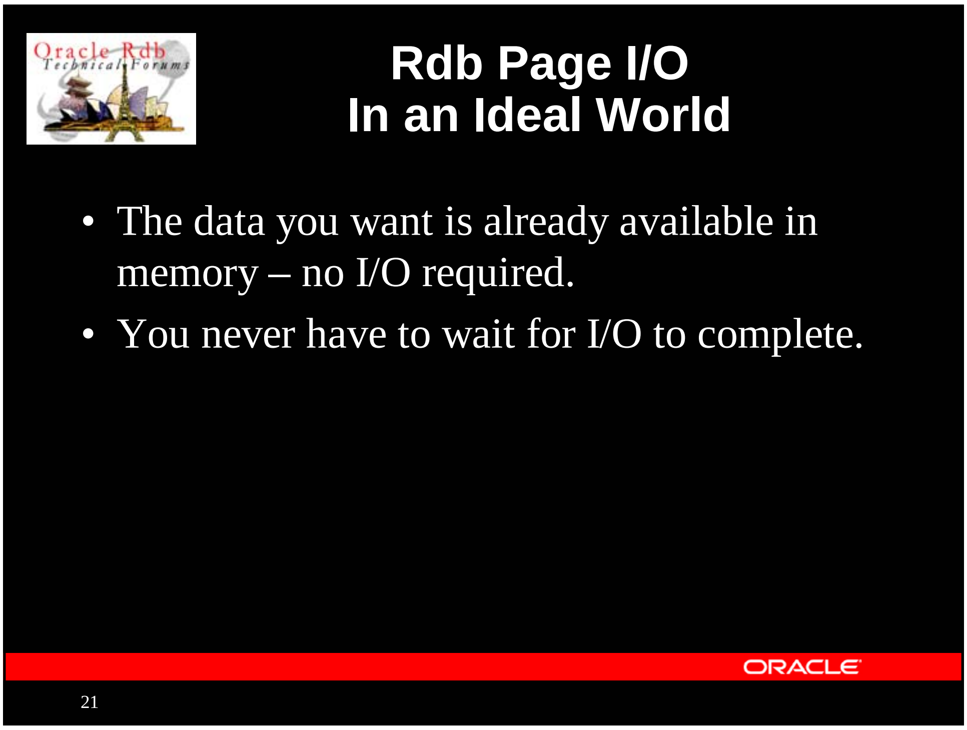
Use Fast Commit

- Fast Commit can substantially reduce write I/O.
- Page locks for modified pages held across transactions.
- Must use care to ensure checkpoints occur.
- In 7.1.0.2 and later you may use the CHECKPOINT INTERVAL IS <n> SECONDS clause to easily ensure checkpoints occur.



Use Row Cache

- Potential for huge reductions in page I/O and locking.
- Best for databases that don’t have a lot of update activity.
- Requires NUMBER OF CLUSTER NODES 1 or Galaxy.

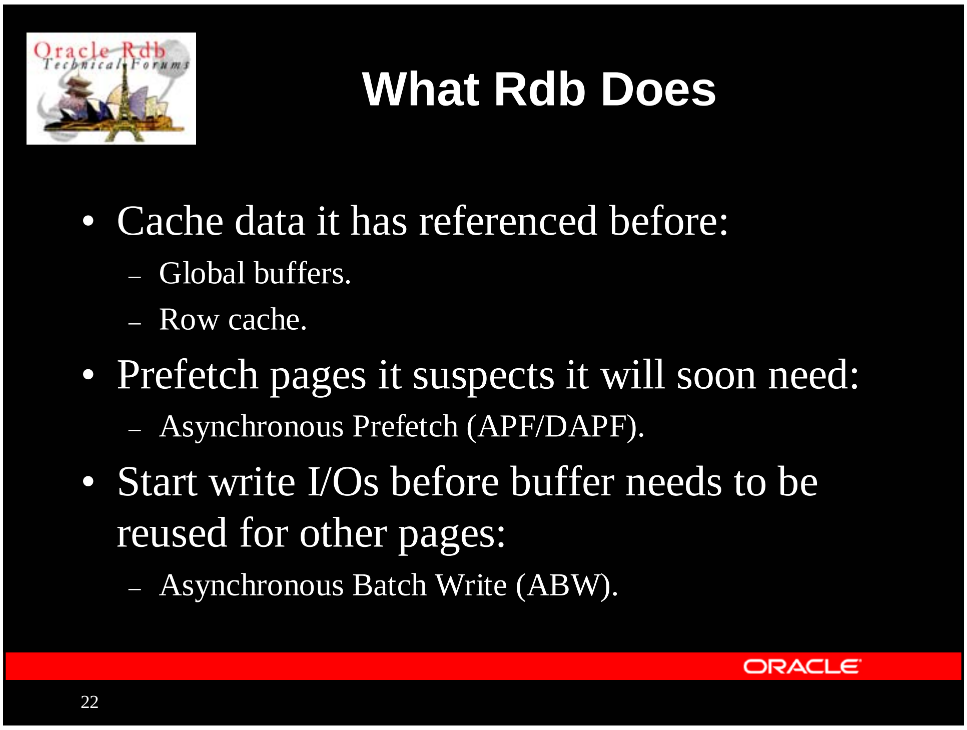


Rdb Page I/O In an Ideal World

- The data you want is already available in memory – no I/O required.
- You never have to wait for I/O to complete.

ORACLE

21

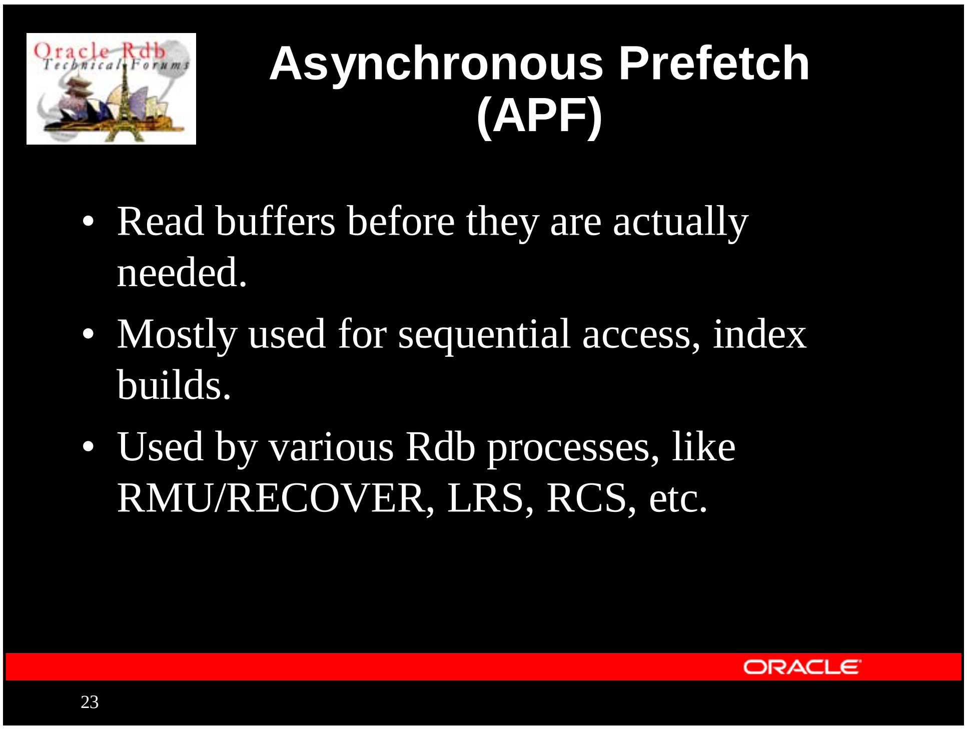


What Rdb Does

- Cache data it has referenced before:
  - Global buffers.
  - Row cache.
- Prefetch pages it suspects it will soon need:
  - Asynchronous Prefetch (APF/DAPF).
- Start write I/Os before buffer needs to be reused for other pages:
  - Asynchronous Batch Write (ABW).

ORACLE

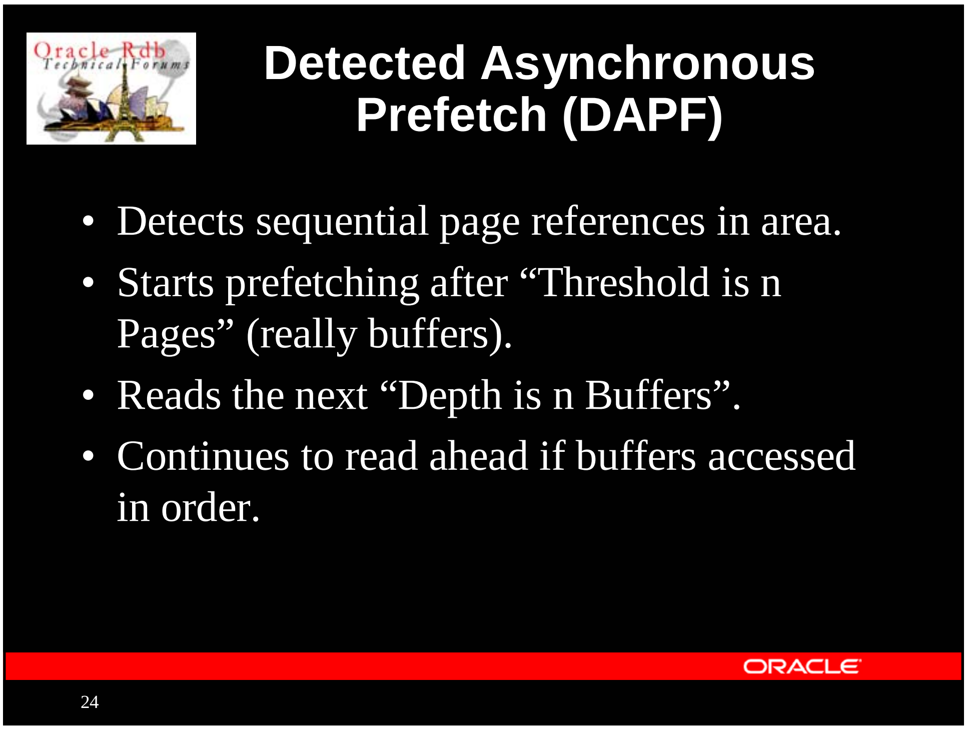
22



Asynchronous Prefetch (APF)

- Read buffers before they are actually needed.
- Mostly used for sequential access, index builds.
- Used by various Rdb processes, like RMU/RECOVER, LRS, RCS, etc.

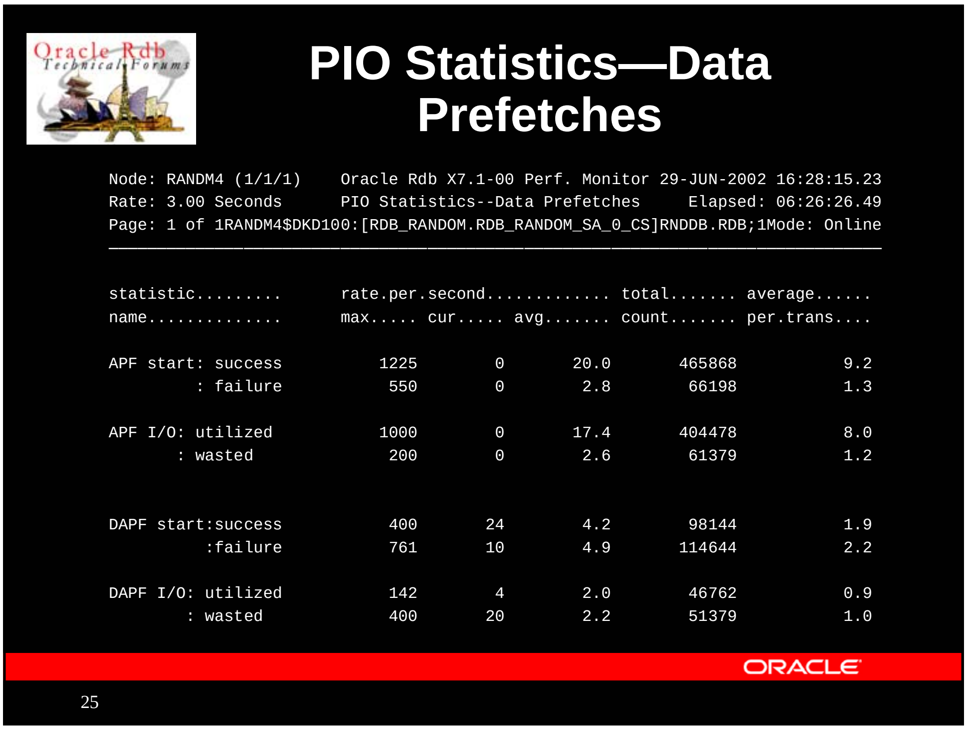
23



Detected Asynchronous Prefetch (DAPF)

- Detects sequential page references in area.
- Starts prefetching after “Threshold is n Pages” (really buffers).
- Reads the next “Depth is n Buffers”.
- Continues to read ahead if buffers accessed in order.

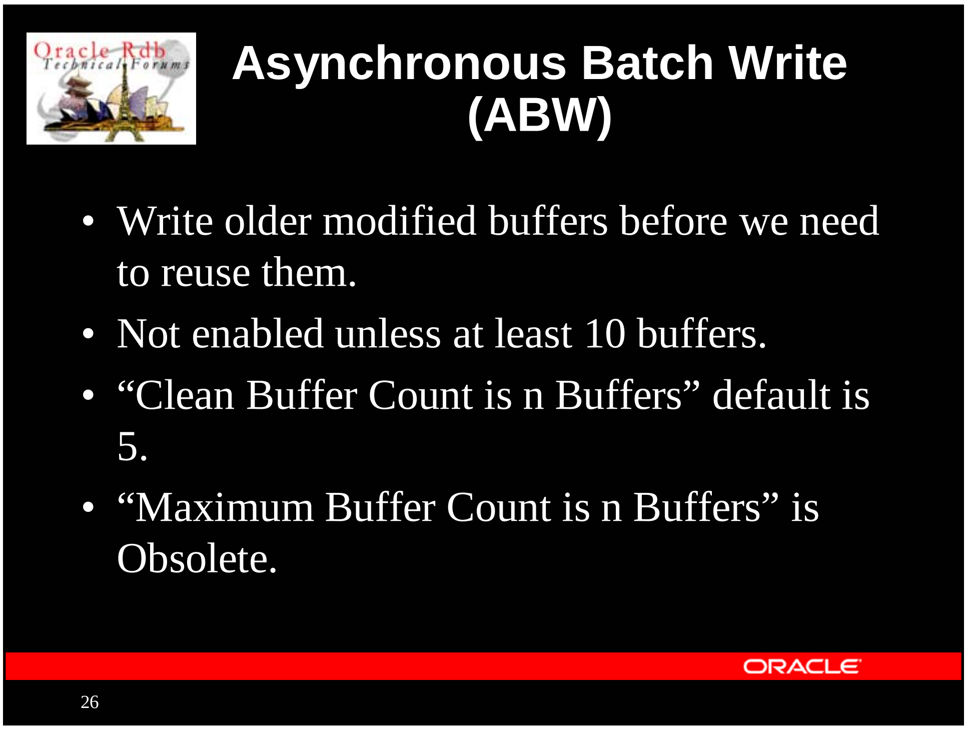
24



PIO Statistics—Data Prefetches

```
Node: RANDM4 (1/1/1)    Oracle Rdb X7.1-00 Perf. Monitor 29-JUN-2002 16:28:15.23
Rate: 3.00 Seconds      PIO Statistics--Data Prefetches     Elapsed: 06:26:26.49
Page: 1 of 1RANDM4$DKD100:[RDB_RANDOM.RDB_RANDOM_SA_0_CS]RNDDB.RDB;1Mode: Online
--------------------------------------------------------------------------------

statistic.........      rate.per.second............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....

APF start: success         1225        0       20.0      465868            9.2
         : failure          550        0        2.8       66198            1.3

APF I/O: utilized          1000        0       17.4      404478            8.0
       : wasted             200        0        2.6       61379            1.2


DAPF start:success          400       24        4.2       98144            1.9
          :failure          761       10        4.9      114644            2.2

DAPF I/O: utilized          142        4        2.0       46762            0.9
        : wasted            400       20        2.2       51379            1.0
```

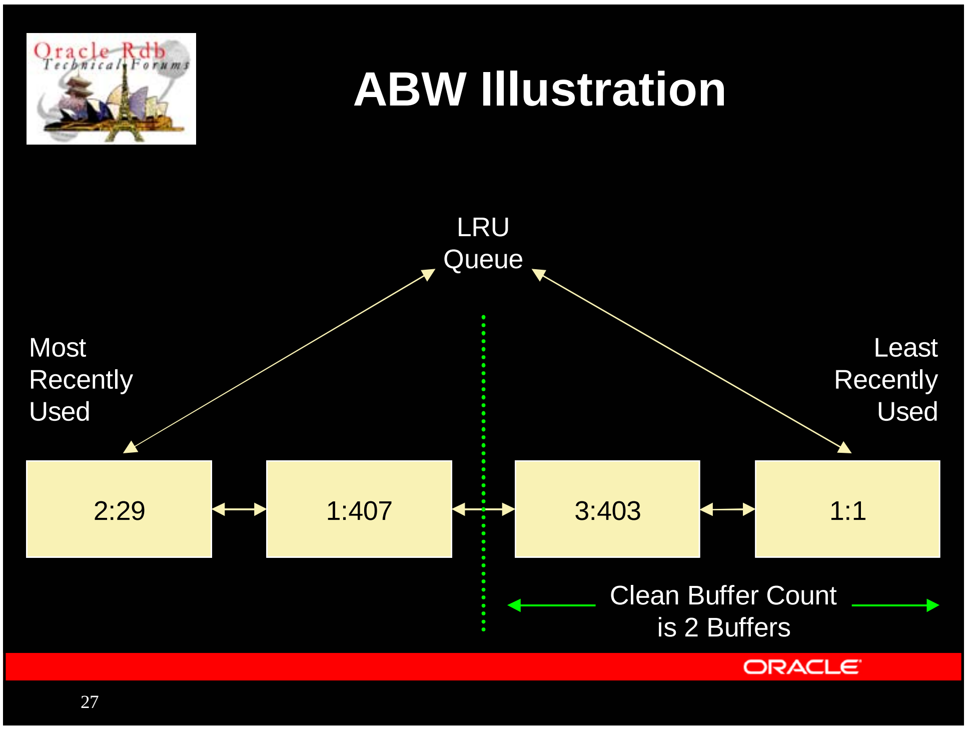
25



Asynchronous Batch Write (ABW)

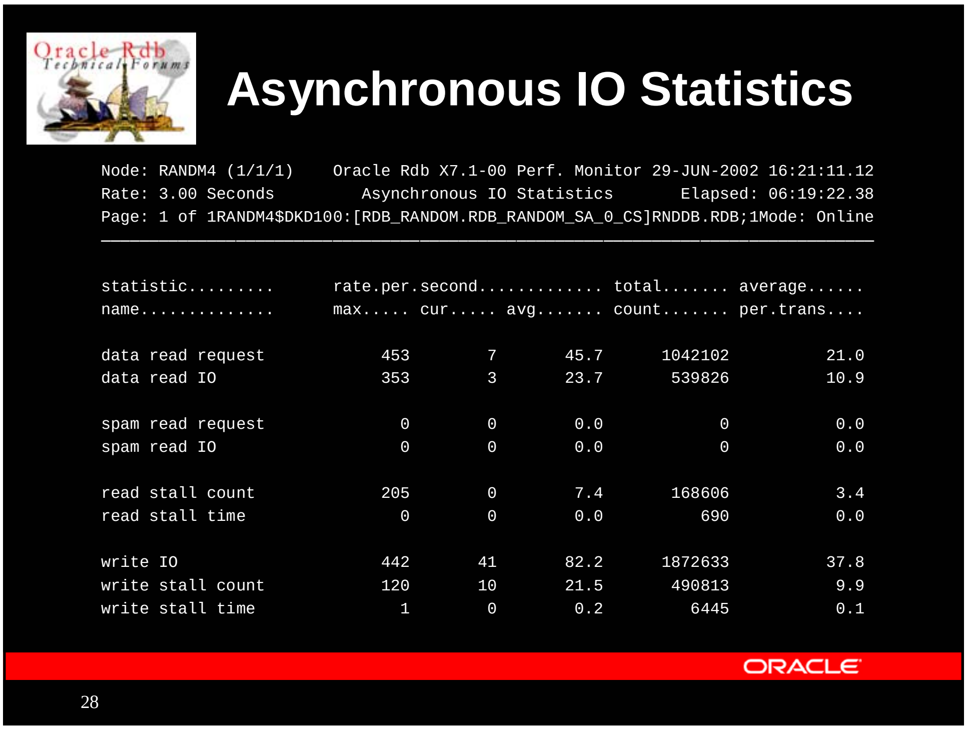
- Write older modified buffers before we need to reuse them.
- Not enabled unless at least 10 buffers.
- “Clean Buffer Count is n Buffers” default is 5.
- “Maximum Buffer Count is n Buffers” is Obsolete.

26



ABW Illustration

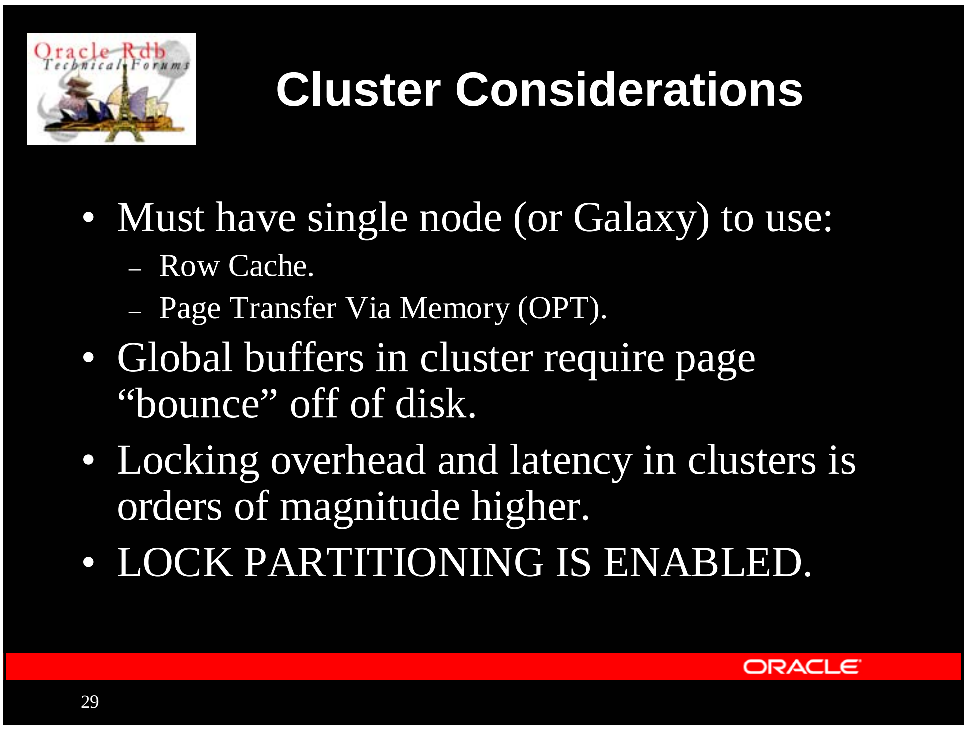
27



Asynchronous IO Statistics

```
Node: RANDM4 (1/1/1)    Oracle Rdb X7.1-00 Perf. Monitor 29-JUN-2002 16:21:11.12
Rate: 3.00 Seconds         Asynchronous IO Statistics       Elapsed: 06:19:22.38
Page: 1 of 1RANDM4$DKD100:[RDB_RANDOM.RDB_RANDOM_SA_0_CS]RNDDB.RDB;1Mode: Online


statistic.........      rate.per.second............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....

data read request            453        7       45.7      1042102          21.0
data read IO                 353        3       23.7       539826          10.9

spam read request              0        0        0.0            0           0.0
spam read IO                   0        0        0.0            0           0.0

read stall count             205        0        7.4       168606           3.4
read stall time                0        0        0.0          690           0.0

write IO                     442       41       82.2      1872633          37.8
write stall count            120       10       21.5       490813           9.9
write stall time               1        0        0.2         6445           0.1
```

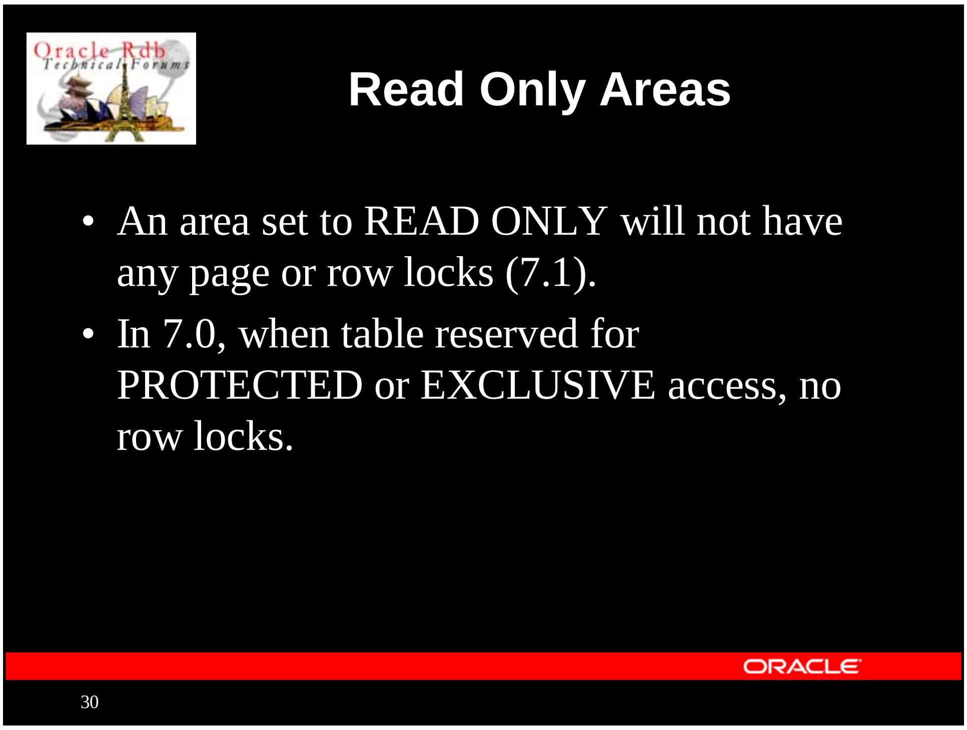
28



Cluster Considerations

- Must have single node (or Galaxy) to use:
  - Row Cache.
  - Page Transfer Via Memory (OPT).
- Global buffers in cluster require page “bounce” off of disk.
- Locking overhead and latency in clusters is orders of magnitude higher.
- LOCK PARTITIONING IS ENABLED.

29



Read Only Areas

- An area set to READ ONLY will not have any page or row locks (7.1).
- In 7.0, when table reserved for PROTECTED or EXCLUSIVE access, no row locks.

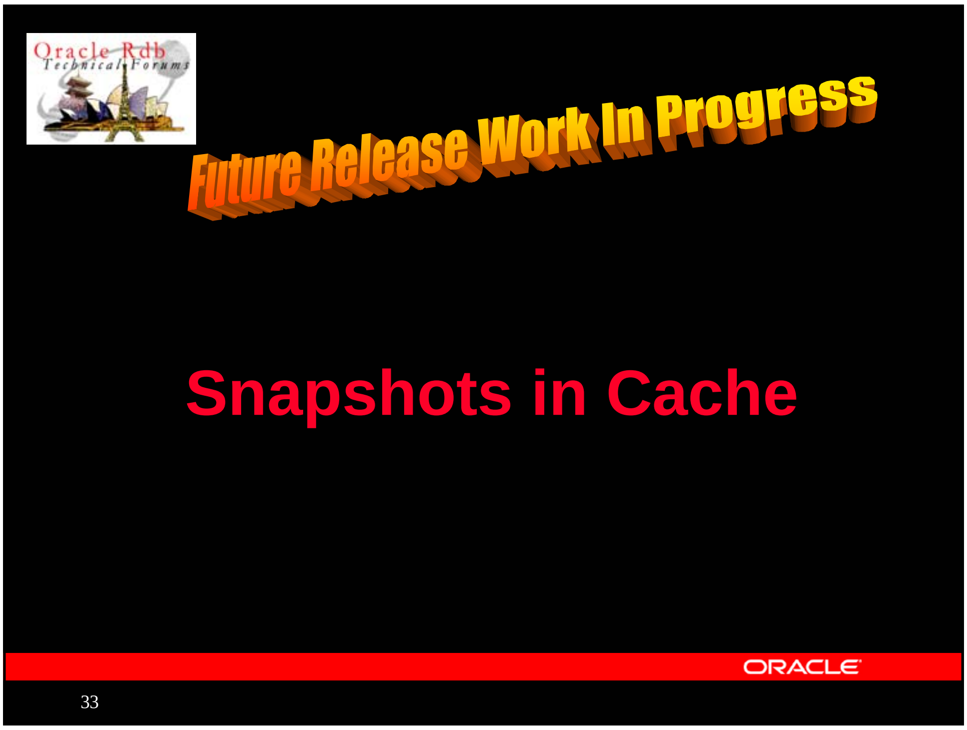
30

Use the Dashboard

- The dashboard allows you to test different db parameter settings.
- RMU/SHOW STATISTICS /OPTION=UPDATE

Fast I/O (Buffer Objects)

- Reduces system I/O overhead.
- Page buffers are memory resident.
- RMU/SET BUFFER_OBJECT /ENABLE=PAGE
- RDM$BIND_PAGE_BUFOBJ_ENABLED
  (prior to Rdb 7.1 the logical was RDM$BIND_BUFOBJ_ENABLED).
- See OpenVMS I/O User's Reference Manual.

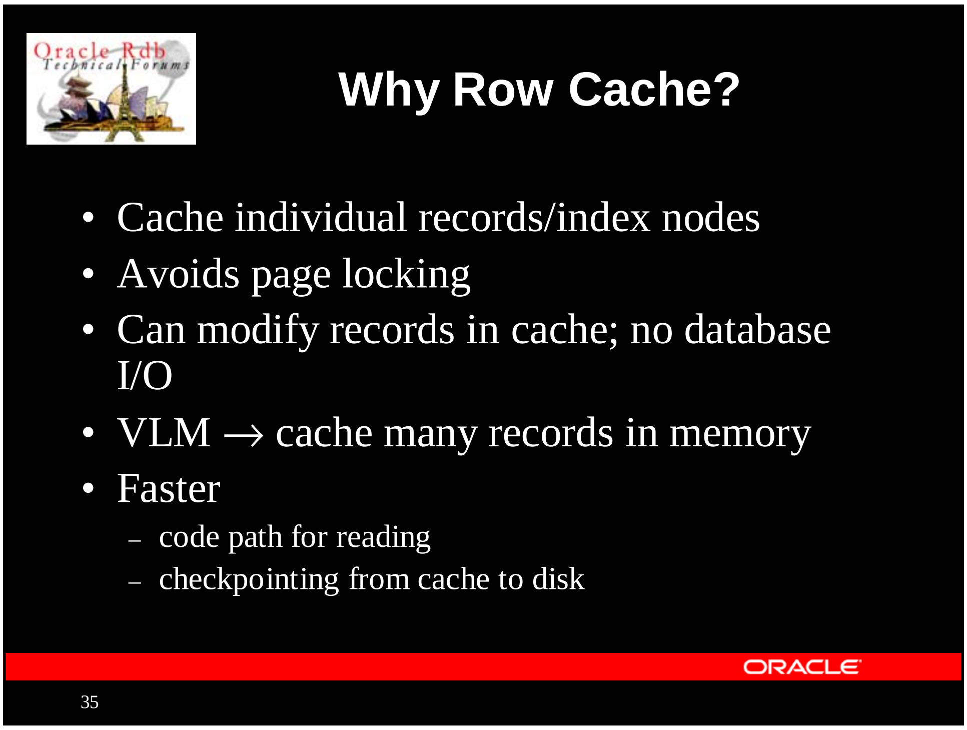


Future Release Work In Progress

Snapshots in Cache

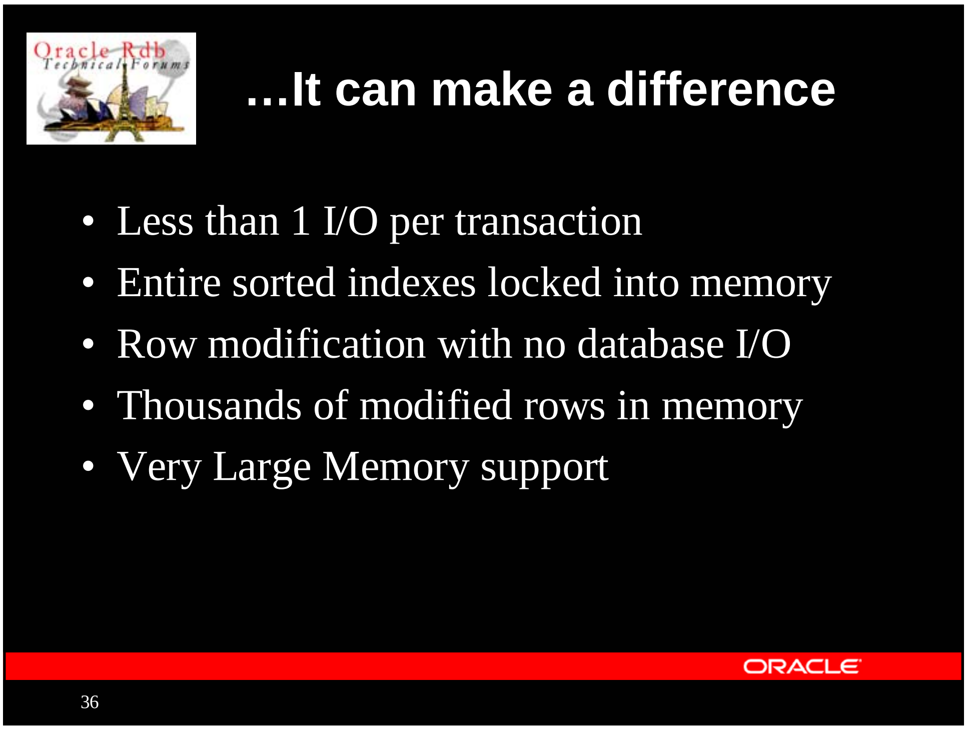
Agenda

- Row Cache Background
- Existing Limitations
- Improvements



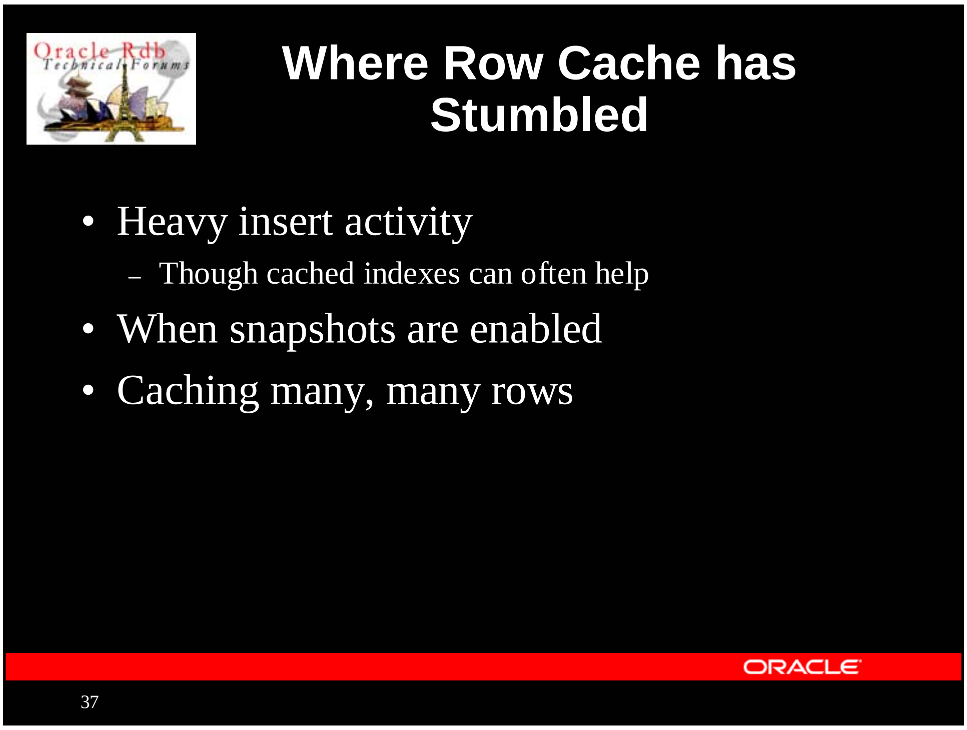
Why Row Cache?

- Cache individual records/index nodes
- Avoids page locking
- Can modify records in cache; no database I/O
- VLM → cache many records in memory
- Faster
  - code path for reading
  - checkpointing from cache to disk



…It can make a difference

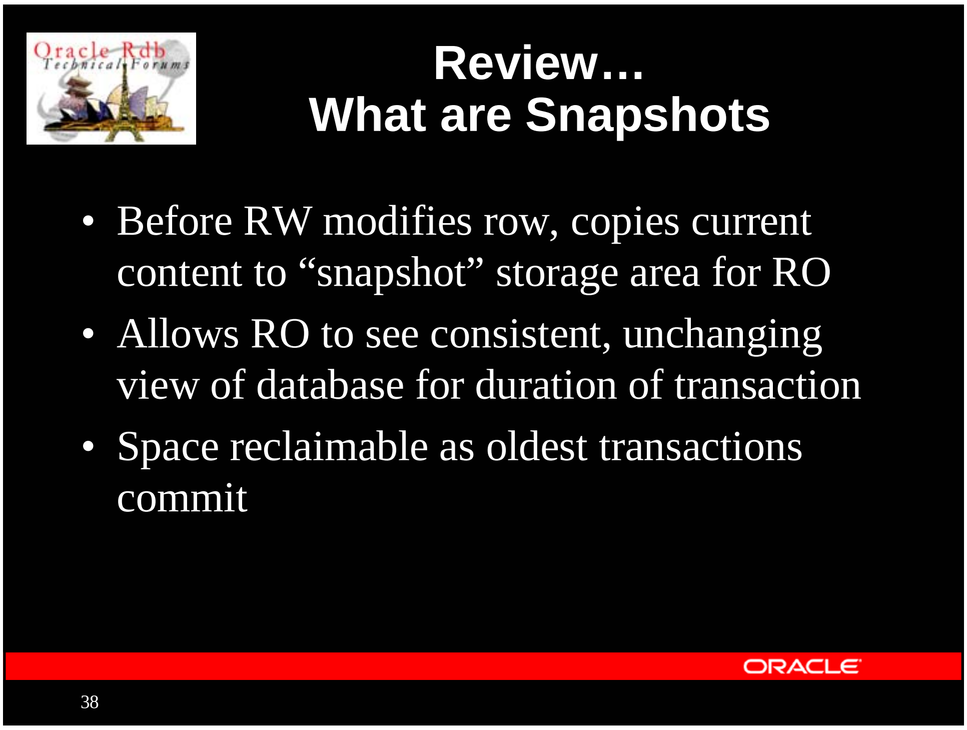
- Less than 1 I/O per transaction
- Entire sorted indexes locked into memory
- Row modification with no database I/O
- Thousands of modified rows in memory
- Very Large Memory support



Where Row Cache has Stumbled

- Heavy insert activity
  - Though cached indexes can often help
- When snapshots are enabled
- Caching many, many rows

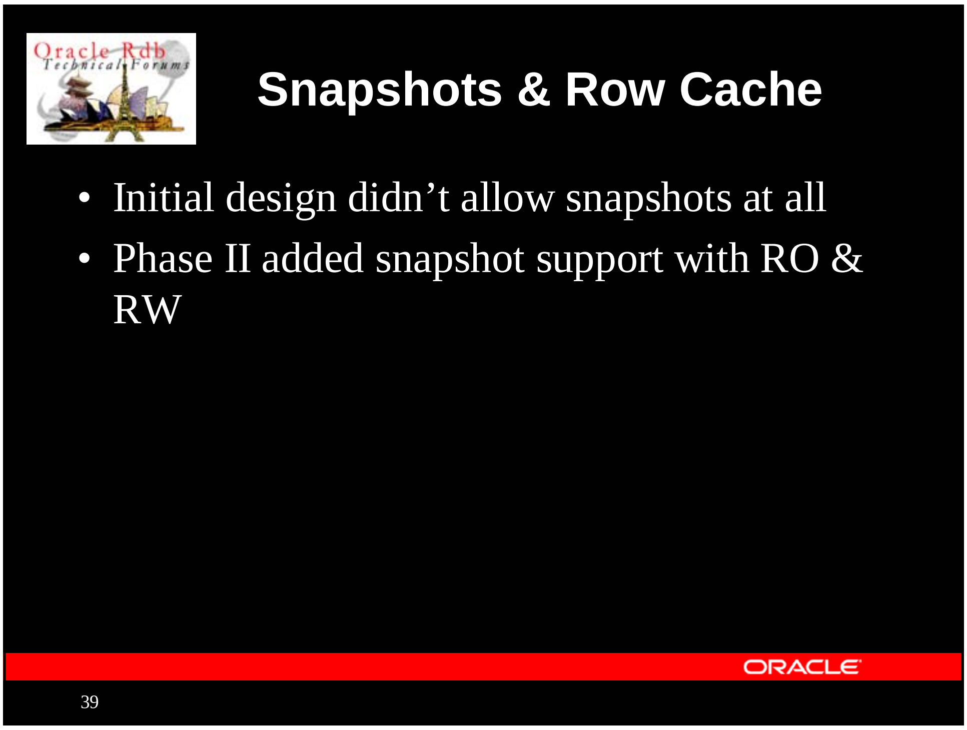
37



Review… What are Snapshots

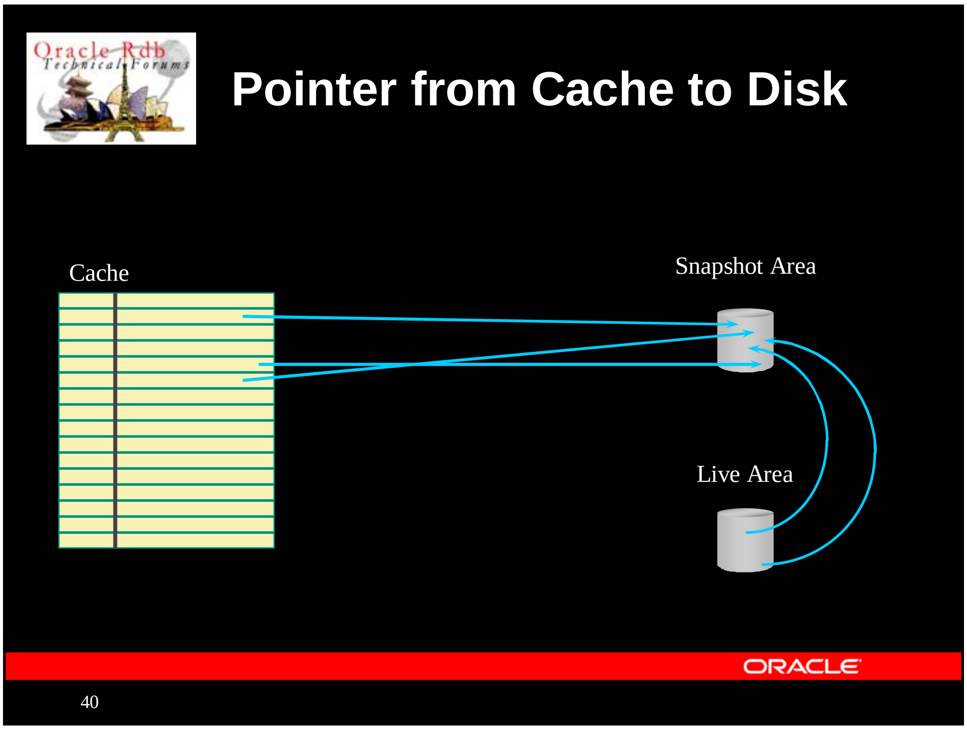
- Before RW modifies row, copies current content to “snapshot” storage area for RO
- Allows RO to see consistent, unchanging view of database for duration of transaction
- Space reclaimable as oldest transactions commit

38

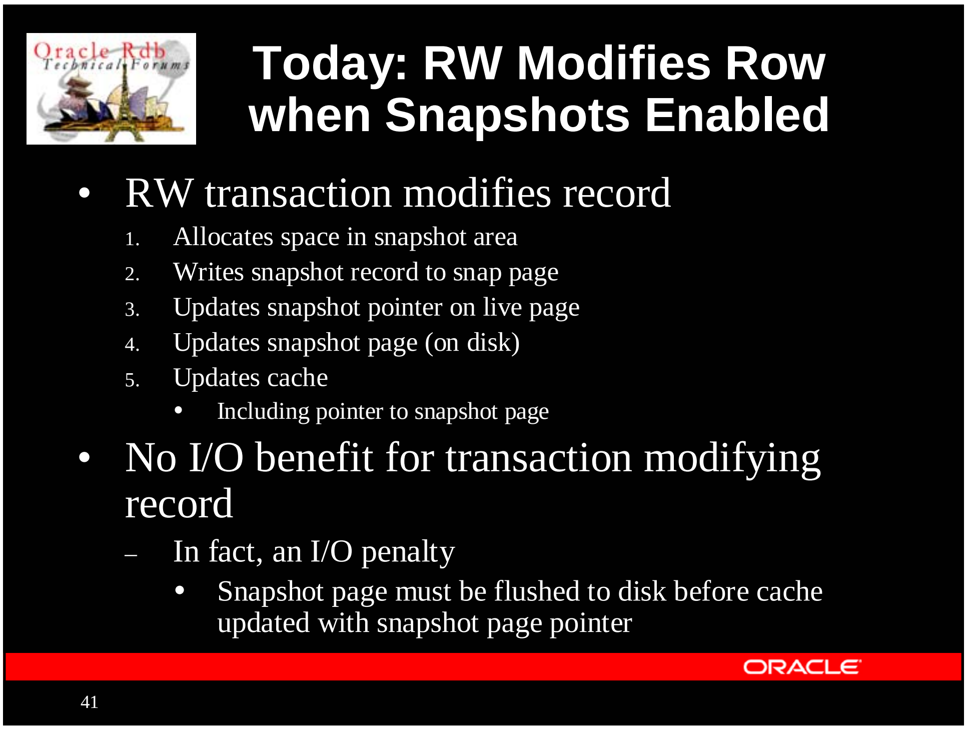


Snapshots & Row Cache

- Initial design didn’t allow snapshots at all
- Phase II added snapshot support with RO & RW



Pointer from Cache to Disk

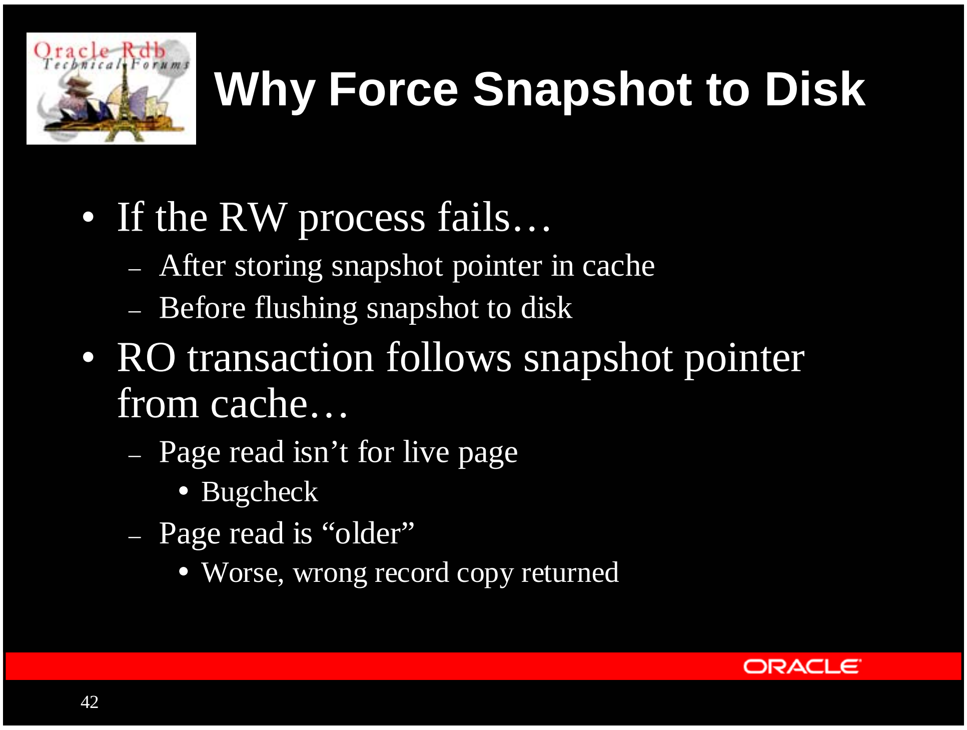


Today: RW Modifies Row when Snapshots Enabled

- RW transaction modifies record
  1. Allocates space in snapshot area
  2. Writes snapshot record to snap page
  3. Updates snapshot pointer on live page
  4. Updates snapshot page (on disk)
  5. Updates cache
     - Including pointer to snapshot page
- No I/O benefit for transaction modifying record
  - In fact, an I/O penalty
    - Snapshot page must be flushed to disk before cache updated with snapshot page pointer

ORACLE

41

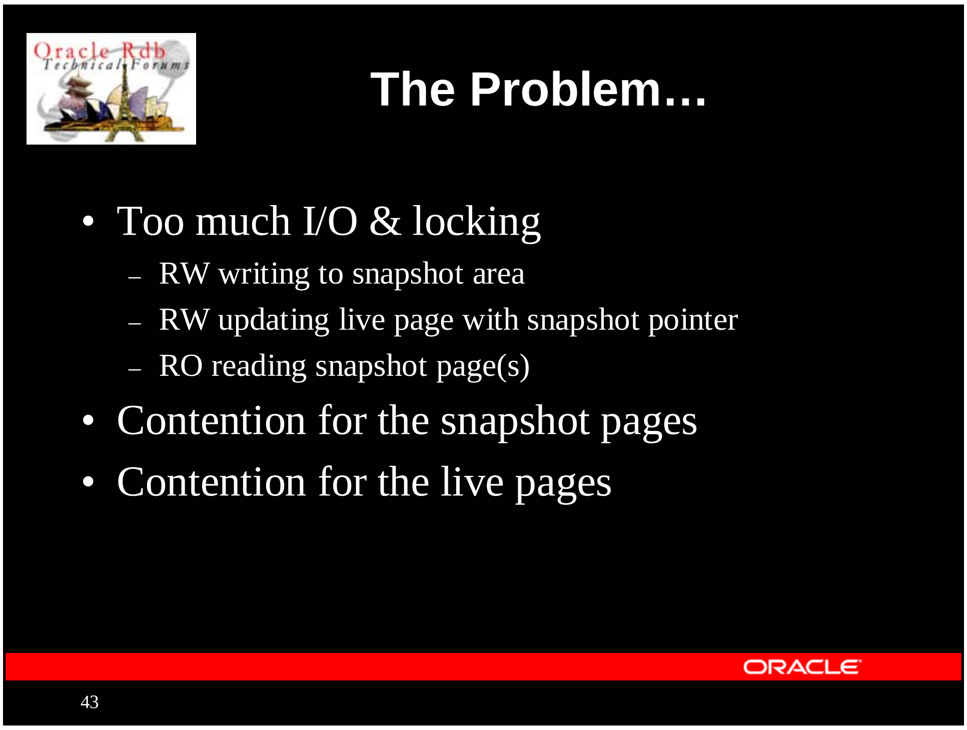


Why Force Snapshot to Disk

- If the RW process fails…
  - After storing snapshot pointer in cache
  - Before flushing snapshot to disk
- RO transaction follows snapshot pointer from cache…
  - Page read isn't for live page
    - Bugcheck
  - Page read is "older"
    - Worse, wrong record copy returned

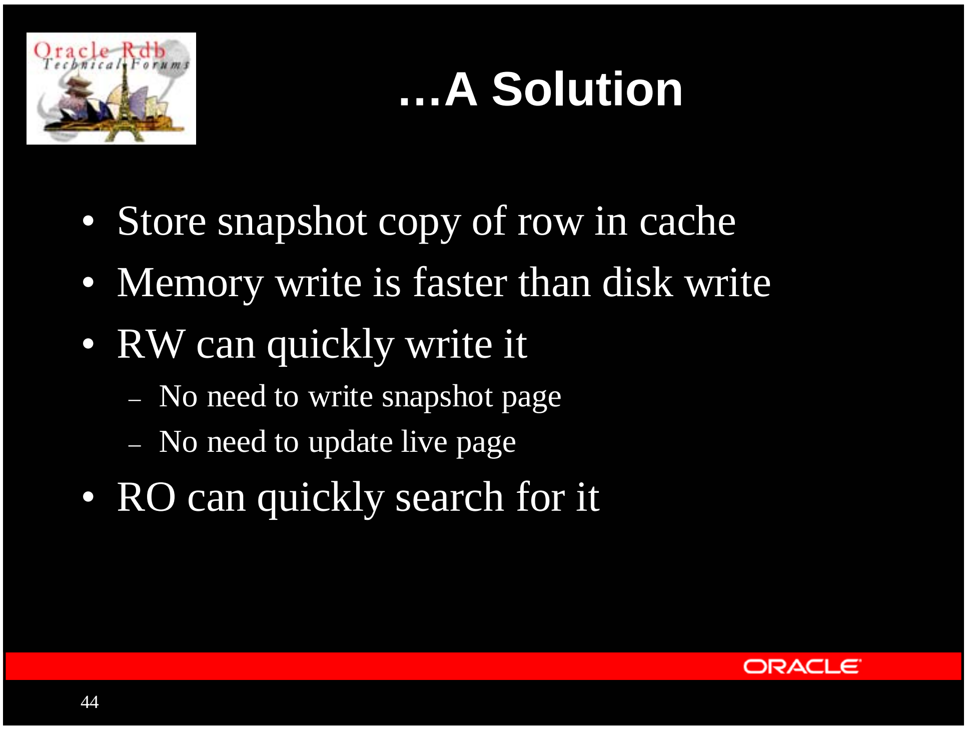
ORACLE

42



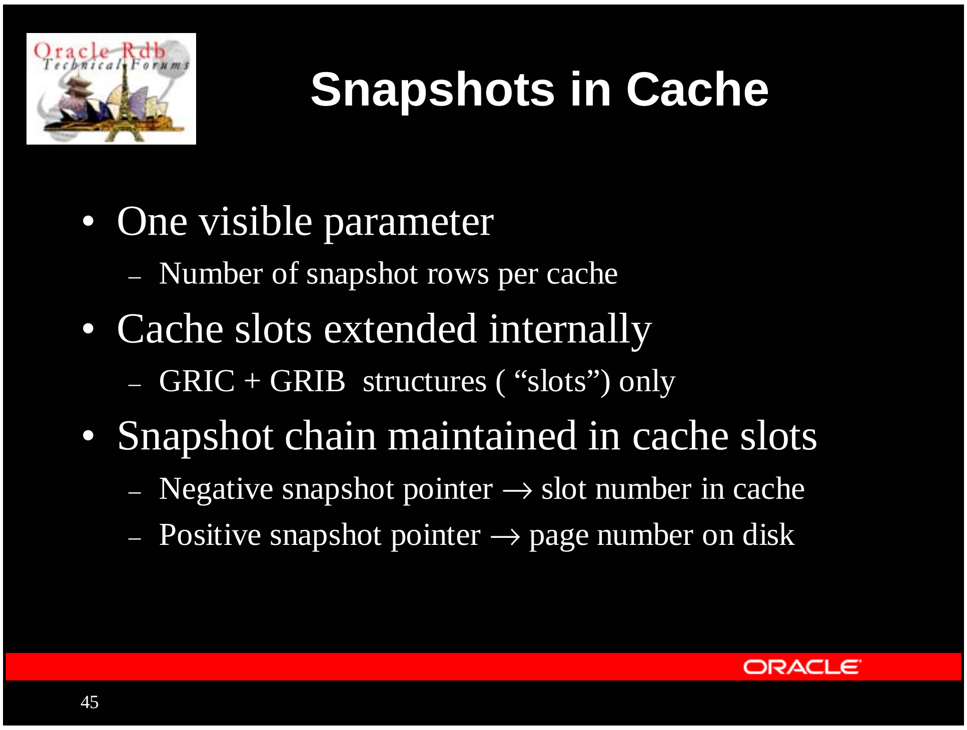
The Problem…

- Too much I/O & locking
  - RW writing to snapshot area
  - RW updating live page with snapshot pointer
  - RO reading snapshot page(s)
- Contention for the snapshot pages
- Contention for the live pages



…A Solution

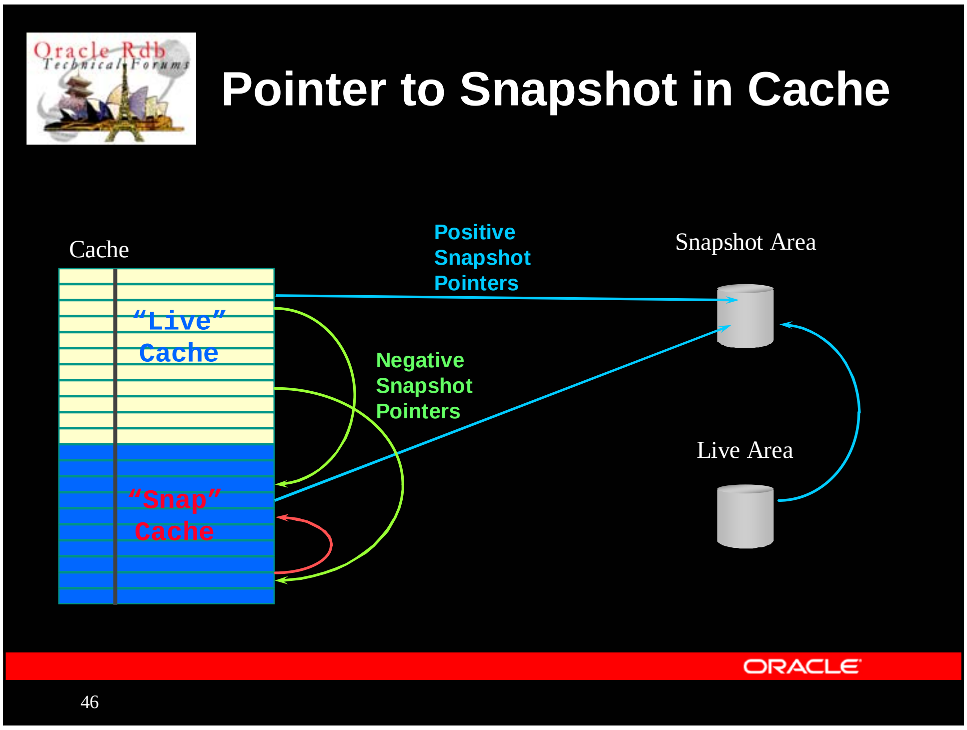
- Store snapshot copy of row in cache
- Memory write is faster than disk write
- RW can quickly write it
  - No need to write snapshot page
  - No need to update live page
- RO can quickly search for it



Snapshots in Cache

- One visible parameter
  - Number of snapshot rows per cache
- Cache slots extended internally
  - GRIC + GRIB structures (“slots”) only
- Snapshot chain maintained in cache slots
  - Negative snapshot pointer → slot number in cache
  - Positive snapshot pointer → page number on disk

45



Pointer to Snapshot in Cache

Cache

“Live” Cache

“Snap” Cache

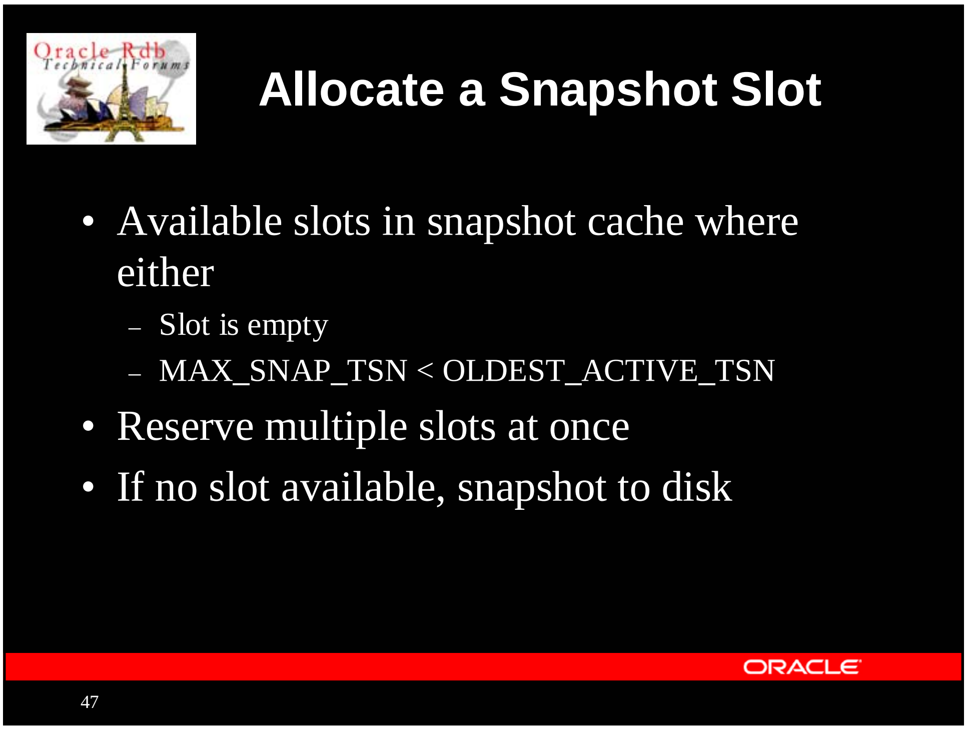
Positive Snapshot Pointers

Negative Snapshot Pointers

Snapshot Area

Live Area

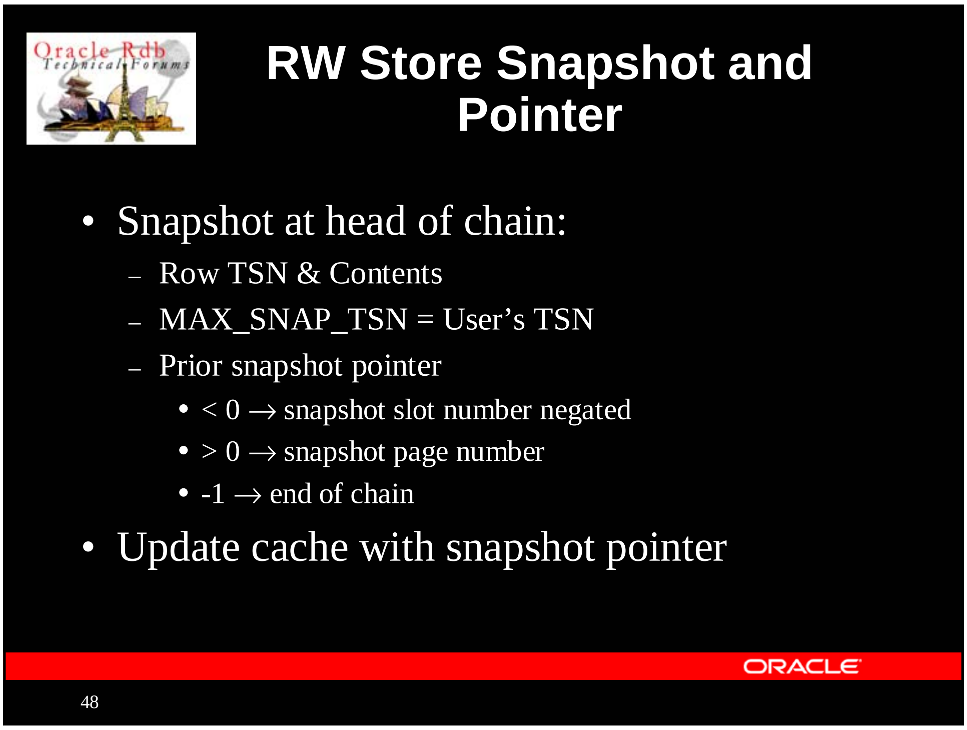
46



Allocate a Snapshot Slot

- Available slots in snapshot cache where either
  - Slot is empty
  - MAX_SNAP_TSN < OLDEST_ACTIVE_TSN
- Reserve multiple slots at once
- If no slot available, snapshot to disk

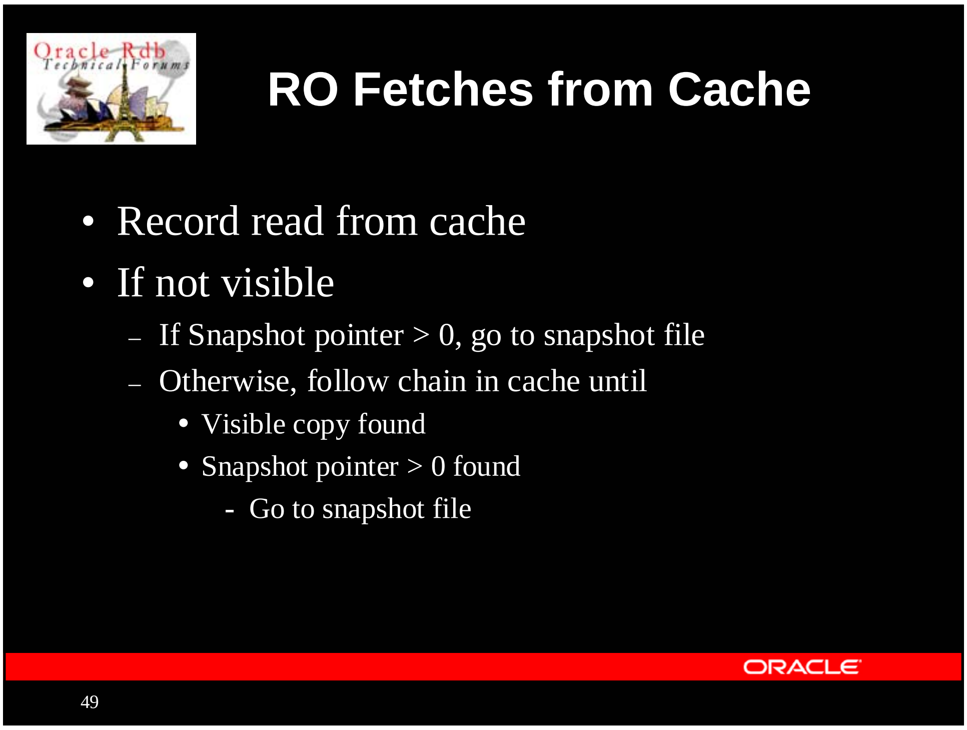
ORACLE



RW Store Snapshot and Pointer

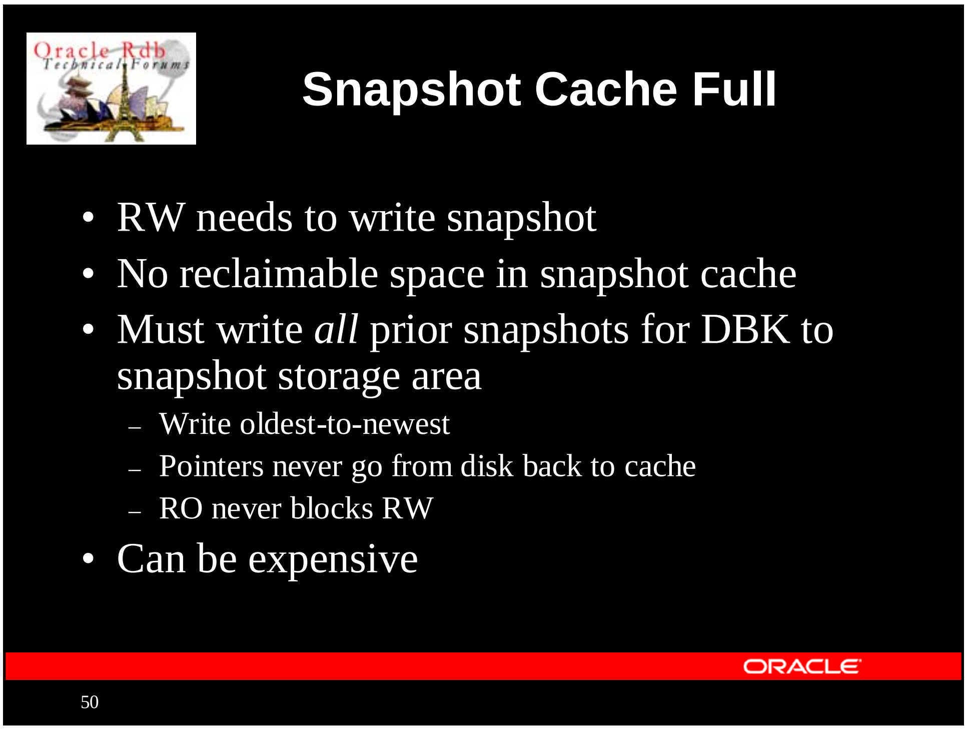
- Snapshot at head of chain:
  - Row TSN & Contents
  - MAX_SNAP_TSN = User’s TSN
  - Prior snapshot pointer
    - < 0 → snapshot slot number negated
    - > 0 → snapshot page number
    - -1 → end of chain
- Update cache with snapshot pointer

ORACLE



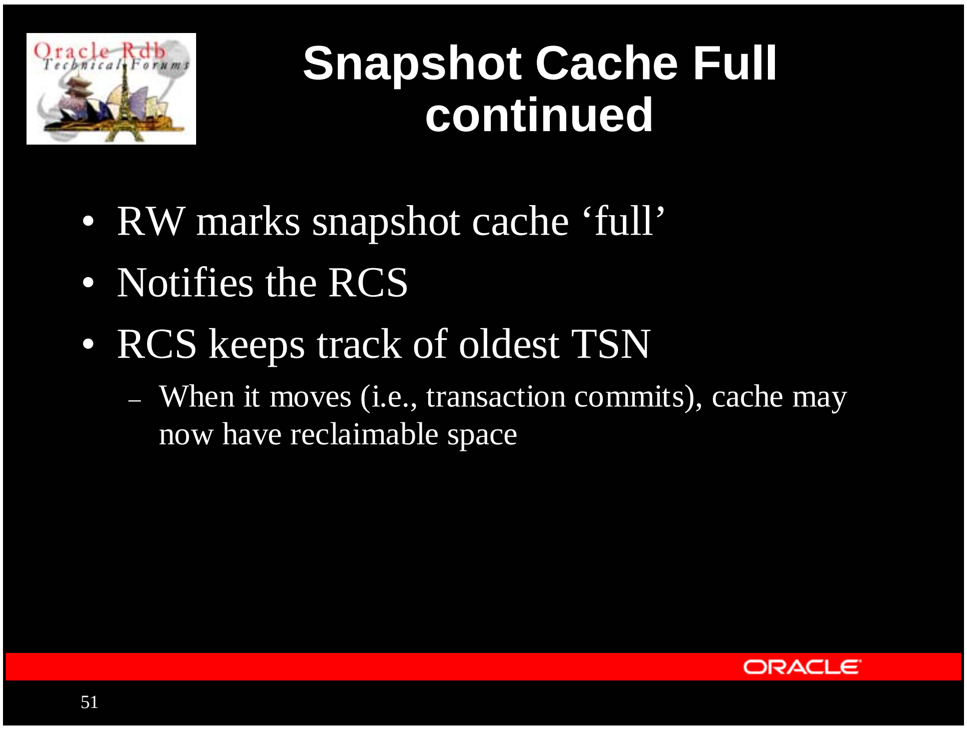
RO Fetches from Cache

- Record read from cache
- If not visible
  - If Snapshot pointer > 0, go to snapshot file
  - Otherwise, follow chain in cache until
    - Visible copy found
    - Snapshot pointer > 0 found
      - Go to snapshot file



Snapshot Cache Full

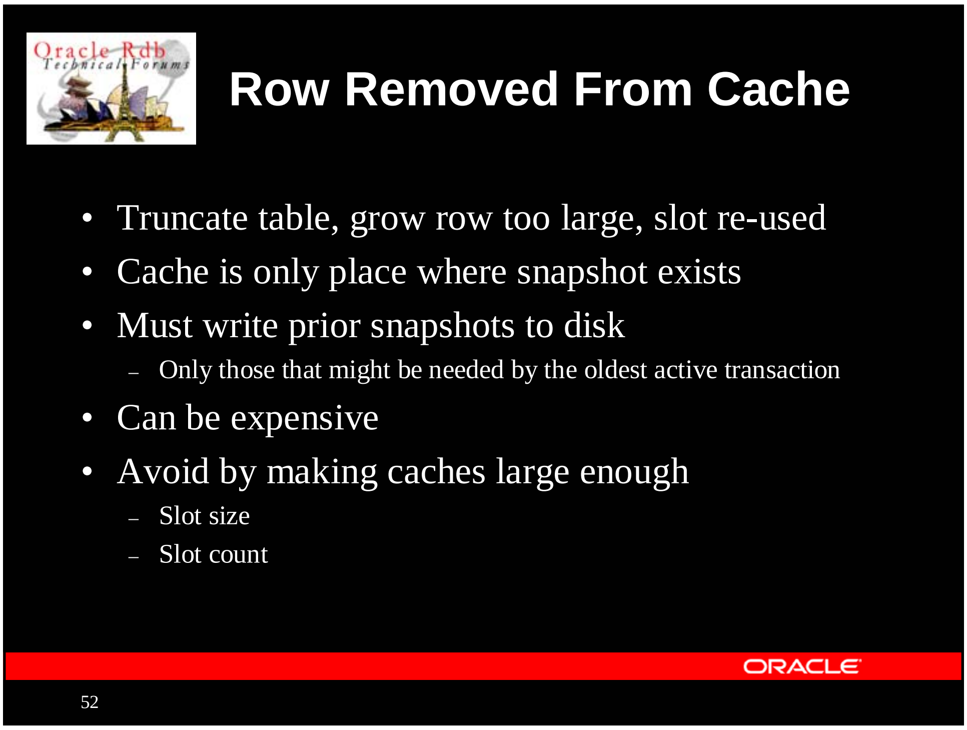
- RW needs to write snapshot
- No reclaimable space in snapshot cache
- Must write *all* prior snapshots for DBK to snapshot storage area
  - Write oldest-to-newest
  - Pointers never go from disk back to cache
  - RO never blocks RW
- Can be expensive



Snapshot Cache Full continued

- RW marks snapshot cache ‘full’
- Notifies the RCS
- RCS keeps track of oldest TSN
  - When it moves (i.e., transaction commits), cache may now have reclaimable space

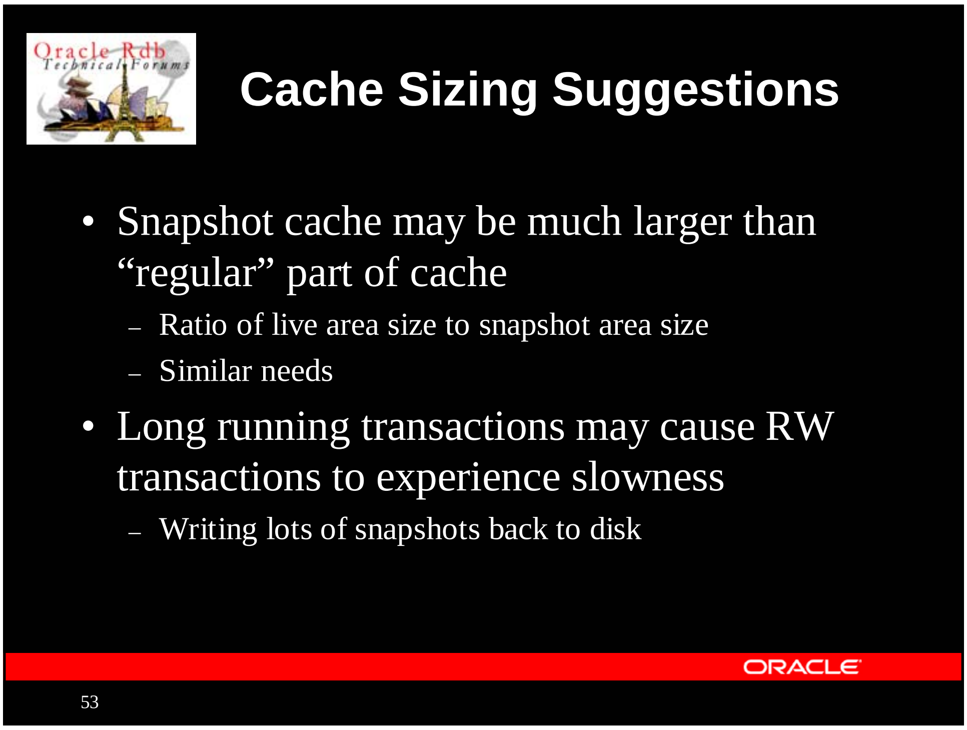
51



Row Removed From Cache

- Truncate table, grow row too large, slot re-used
- Cache is only place where snapshot exists
- Must write prior snapshots to disk
  - Only those that might be needed by the oldest active transaction
- Can be expensive
- Avoid by making caches large enough
  - Slot size
  - Slot count

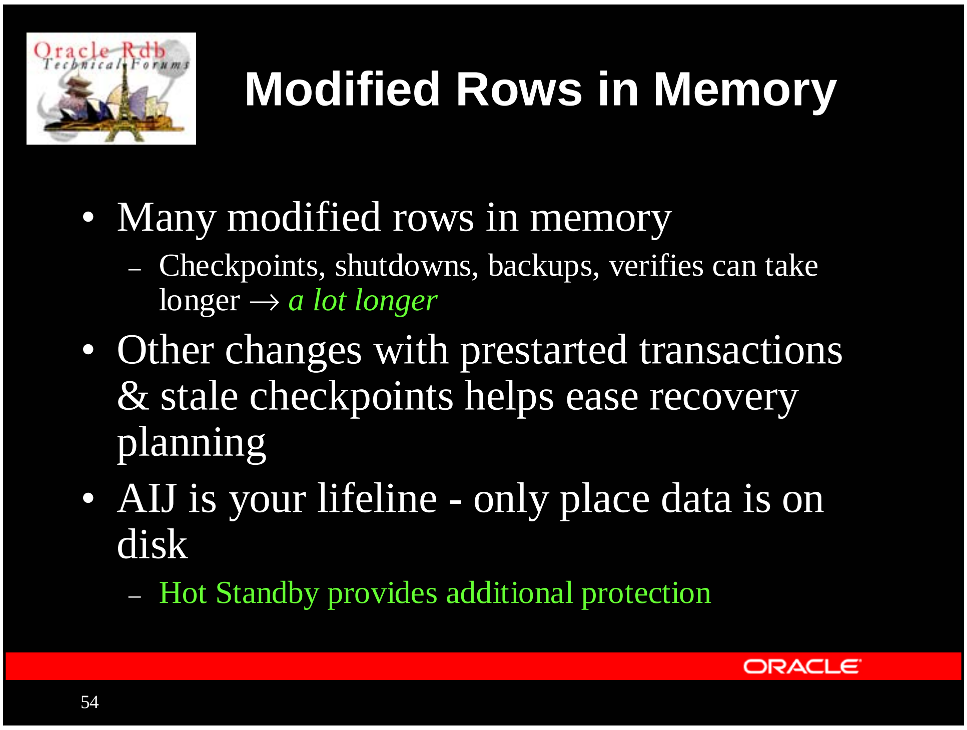
52



Cache Sizing Suggestions

- Snapshot cache may be much larger than “regular” part of cache
  - Ratio of live area size to snapshot area size
  - Similar needs
- Long running transactions may cause RW transactions to experience slowness
  - Writing lots of snapshots back to disk

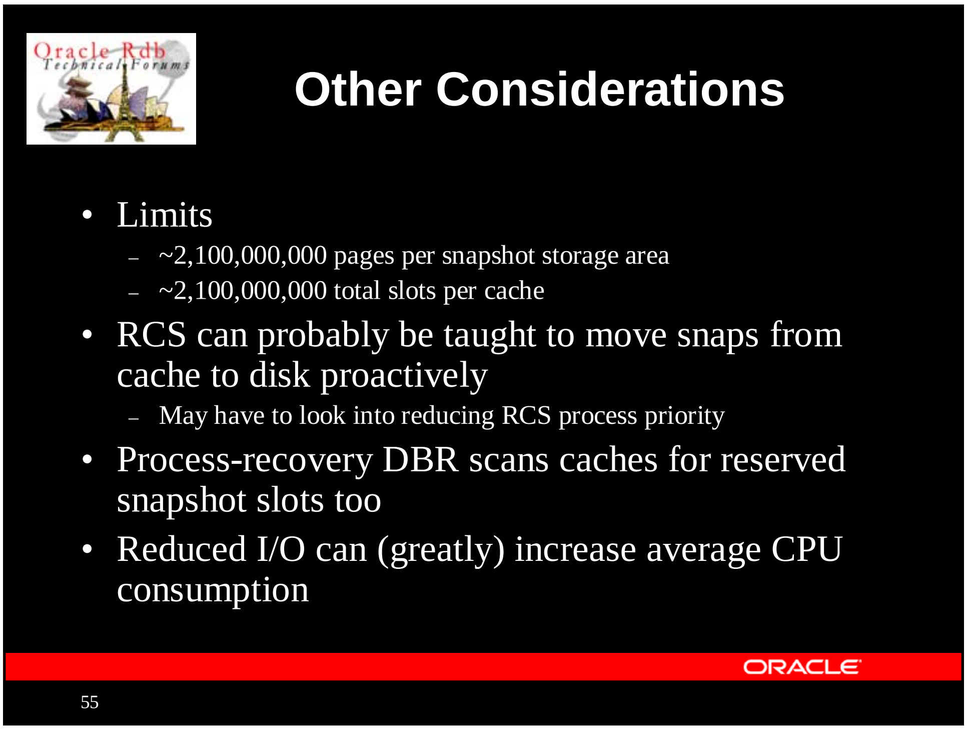
53



Modified Rows in Memory

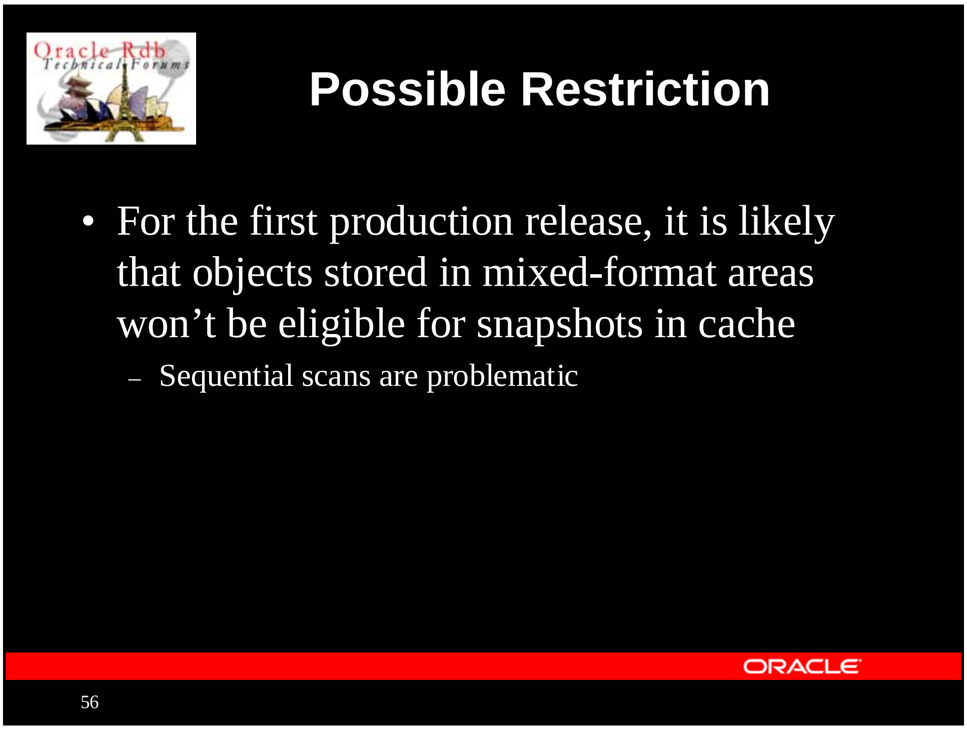
- Many modified rows in memory
  - Checkpoints, shutdowns, backups, verifies can take longer → *a lot longer*
- Other changes with prestarted transactions & stale checkpoints helps ease recovery planning
- AIJ is your lifeline - only place data is on disk
  - Hot Standby provides additional protection

54



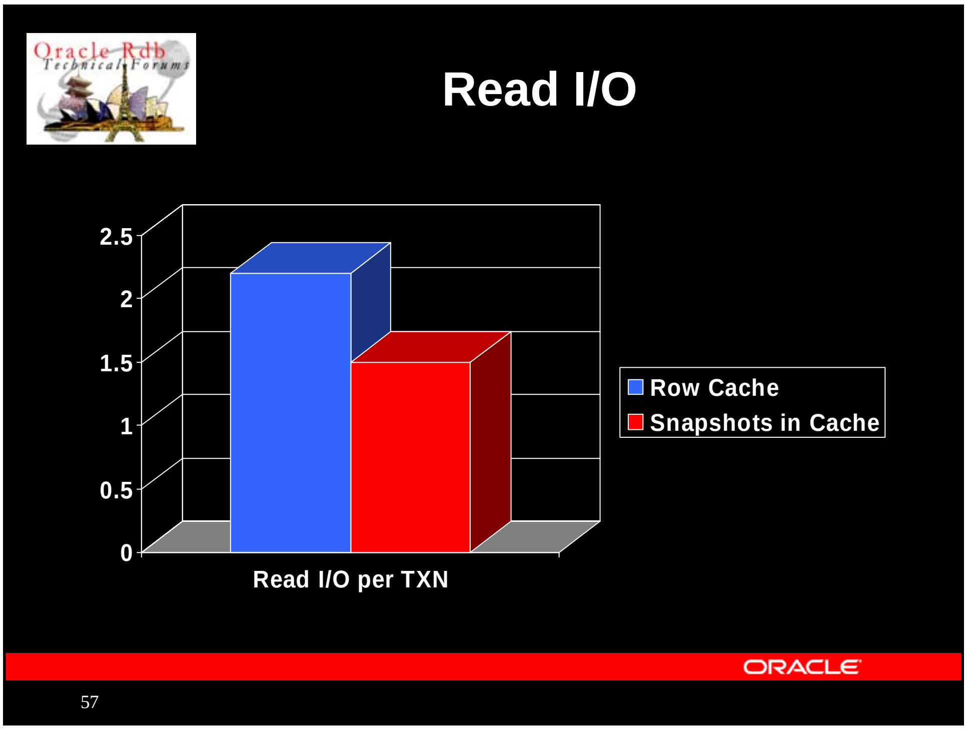
Other Considerations

- Limits
  - ~2,100,000,000 pages per snapshot storage area
  - ~2,100,000,000 total slots per cache
- RCS can probably be taught to move snaps from cache to disk proactively
  - May have to look into reducing RCS process priority
- Process-recovery DBR scans caches for reserved snapshot slots too
- Reduced I/O can (greatly) increase average CPU consumption



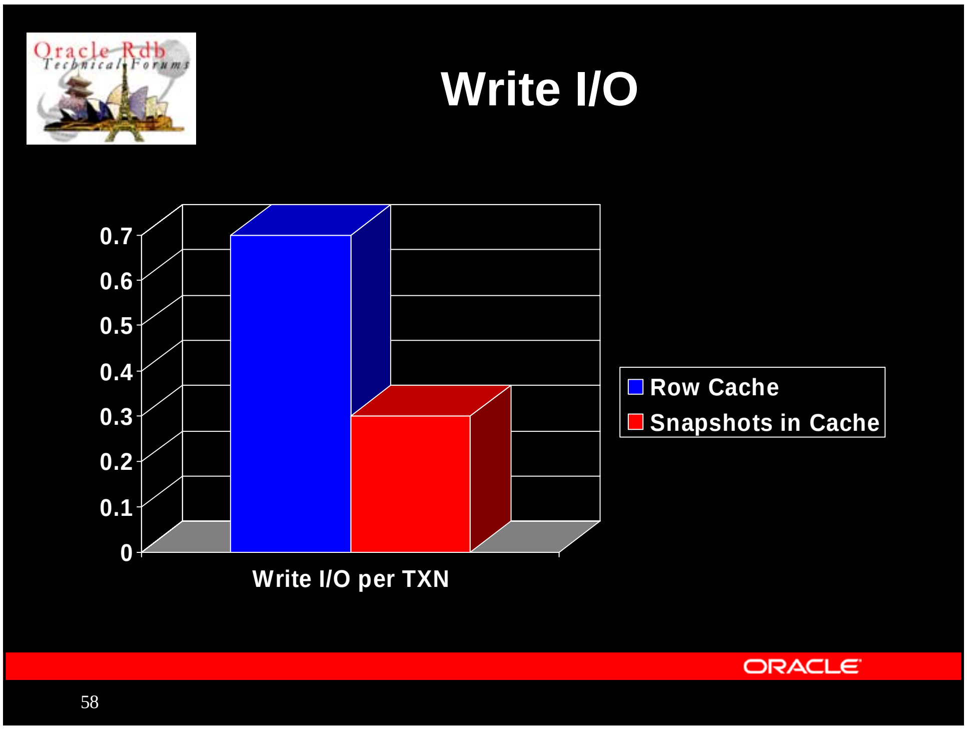
Possible Restriction

- For the first production release, it is likely that objects stored in mixed-format areas won’t be eligible for snapshots in cache
  - Sequential scans are problematic



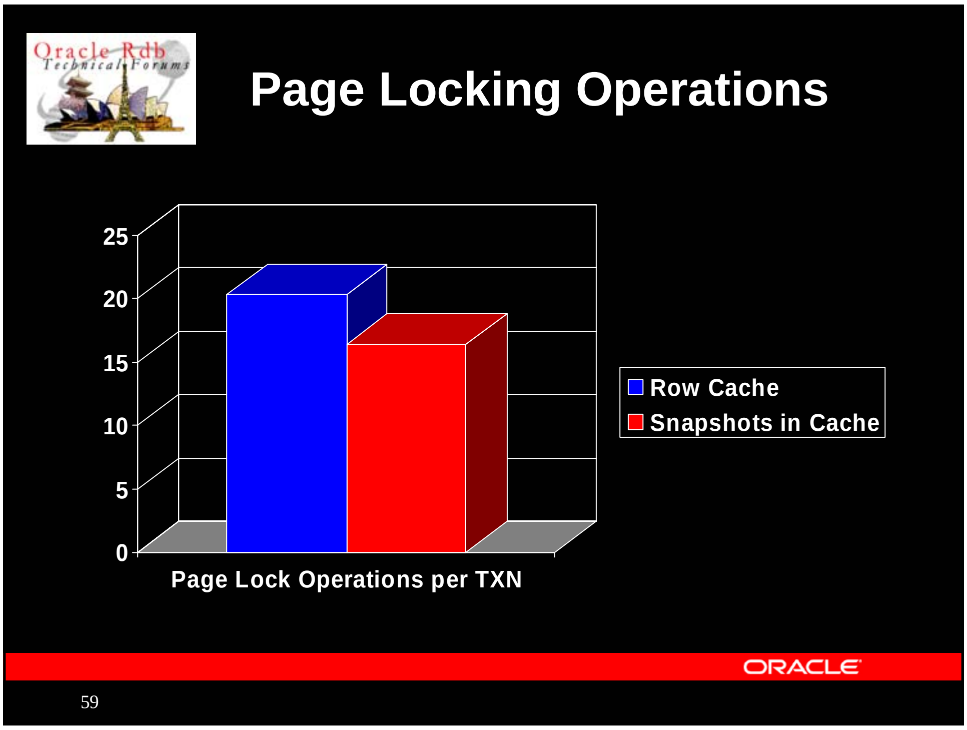
Read I/O

57



Write I/O

58



Page Locking Operations

59

Questions? Comments?

philippe.vigier@oracle.com

60

ORACLE®

http://www.oracle.com/rdb