

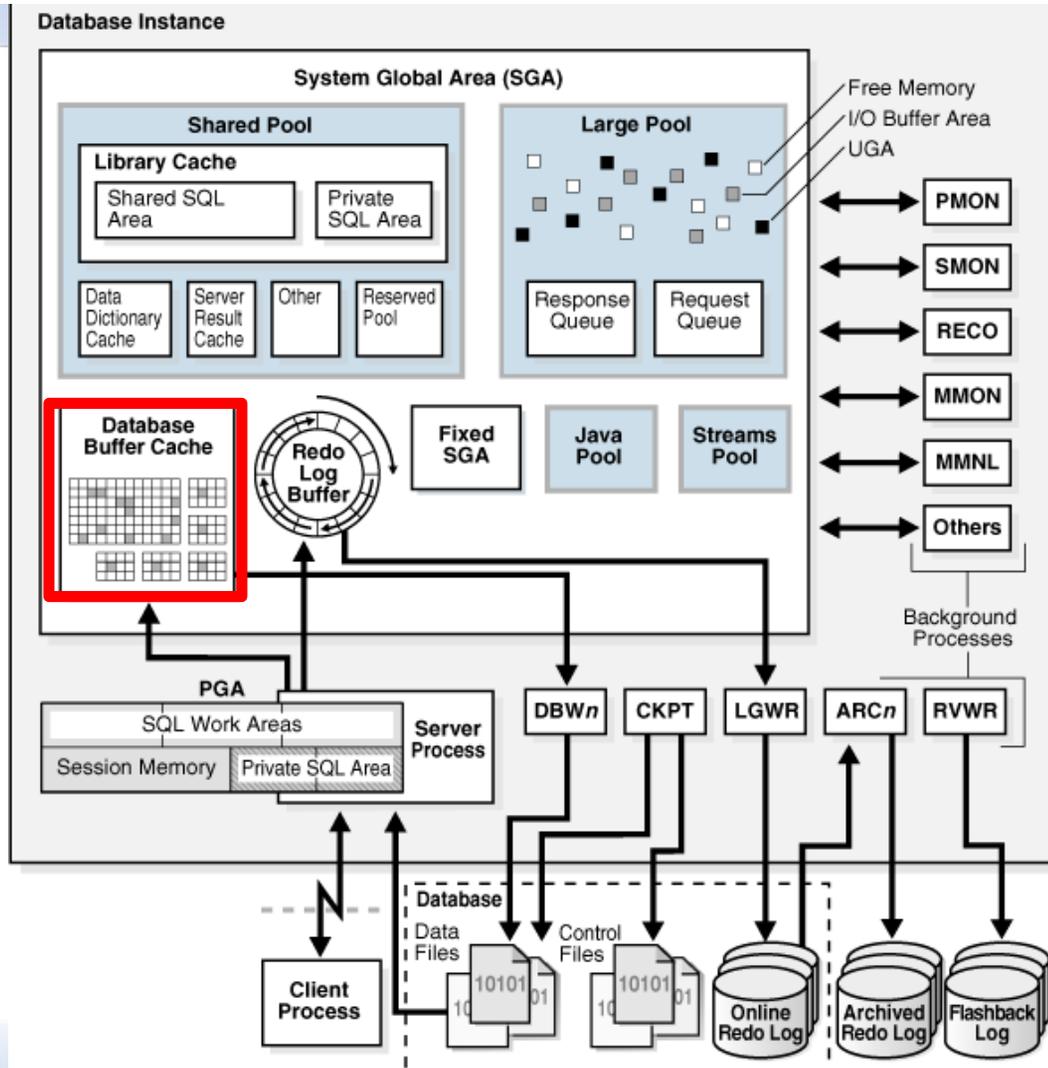
How Buffer Cache Works

Arup Nanda

Priceline

A Booking Holdings Company

Oracle Instance



Source: Oracle Database Documentation Concepts Guide

@ArupNanda

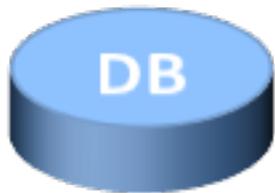
Arup Nanda

How Buffer Cache Works

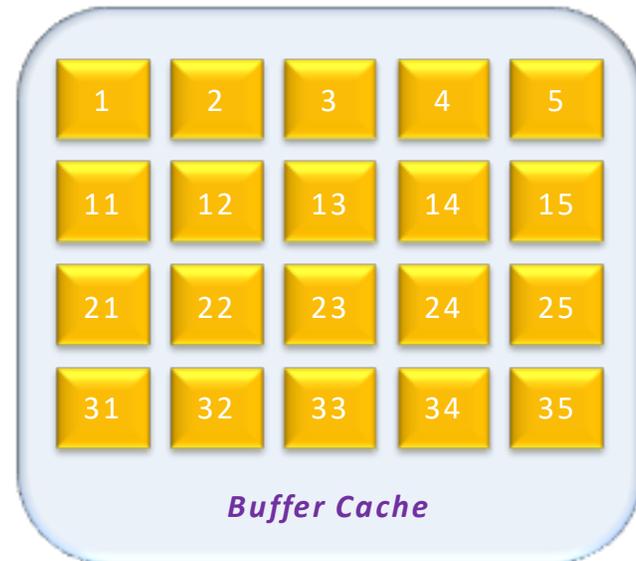
Agenda

- Difference between block and buffer
- How many buffers are created
- There is just one copy of the buffer in the cache, right?
- What is the “state” of a buffer
- What are hash buckets (chains)
- What is a cache buffer chain latch
- How many latches are there
- How do you increase the performance of the buffer cache

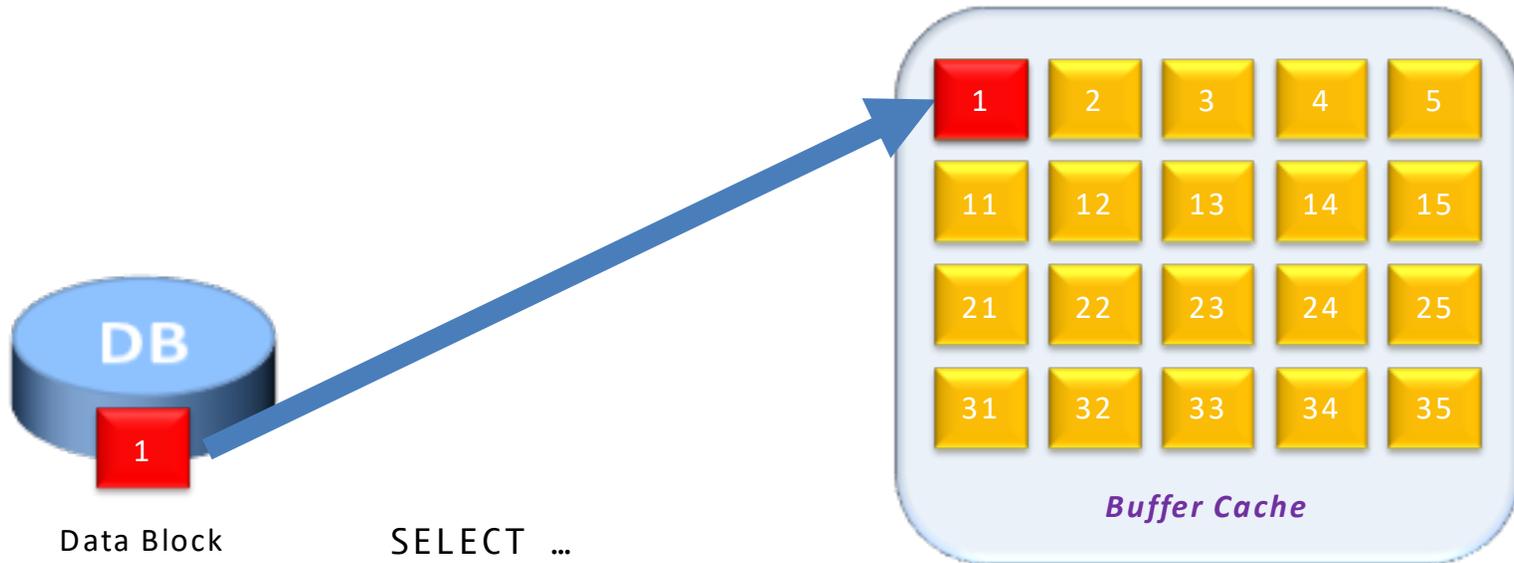
Buffer Operation



```
SELECT ...  
FROM EMP  
WHERE ...
```



Buffer Operation

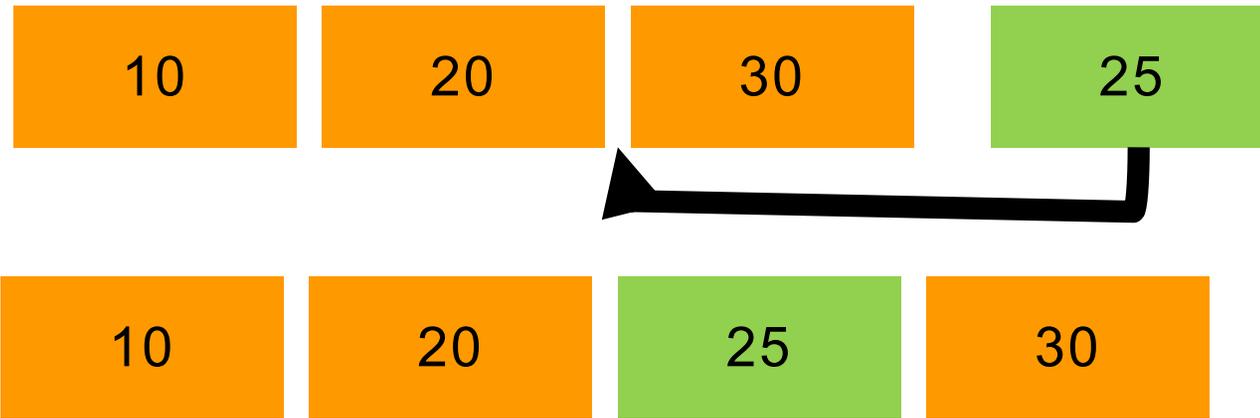


Data Block

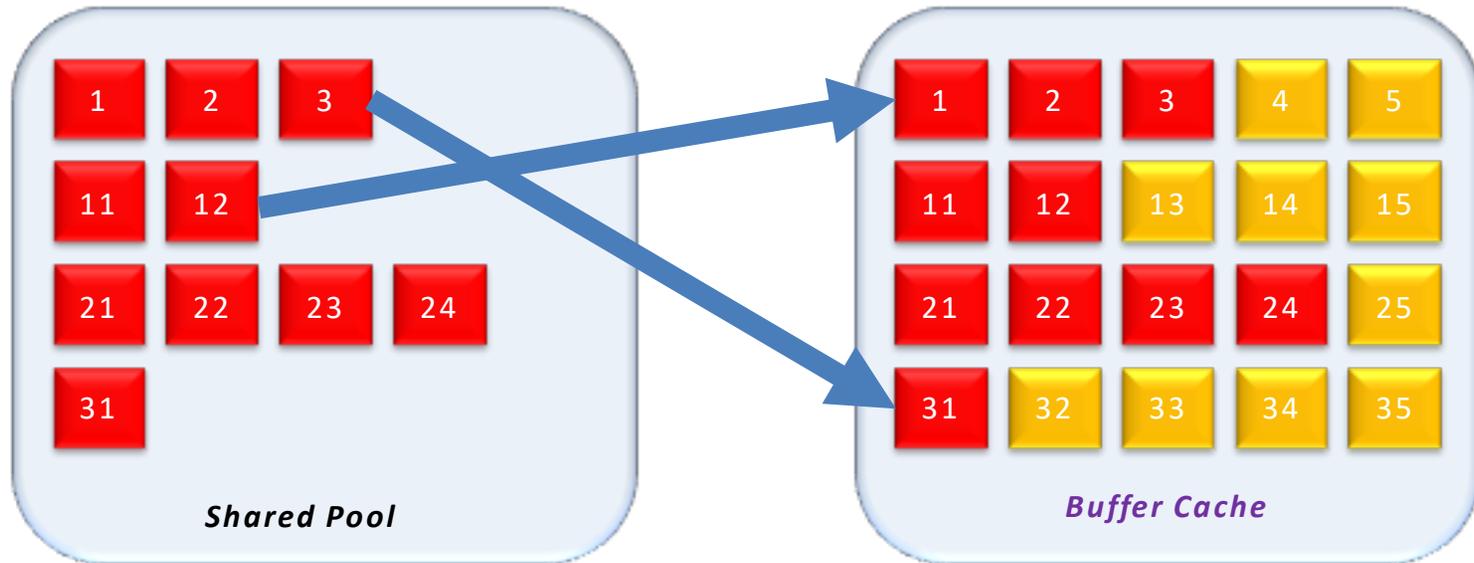
```
SELECT ...  
FROM EMP  
WHERE ...
```

Buffer Cache

Buffer Insertion



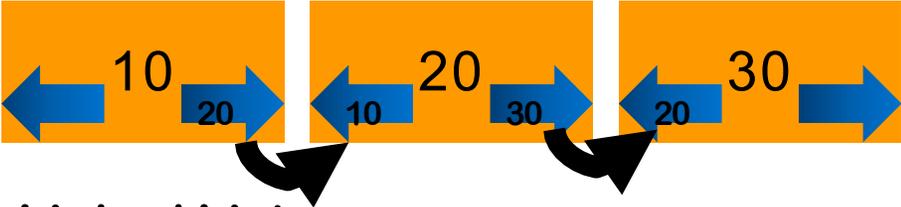
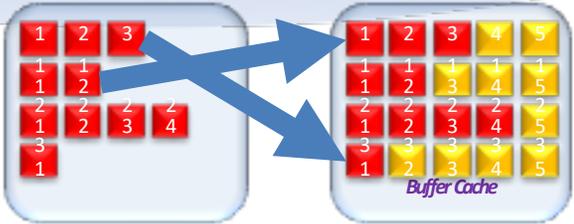
Buffer Header



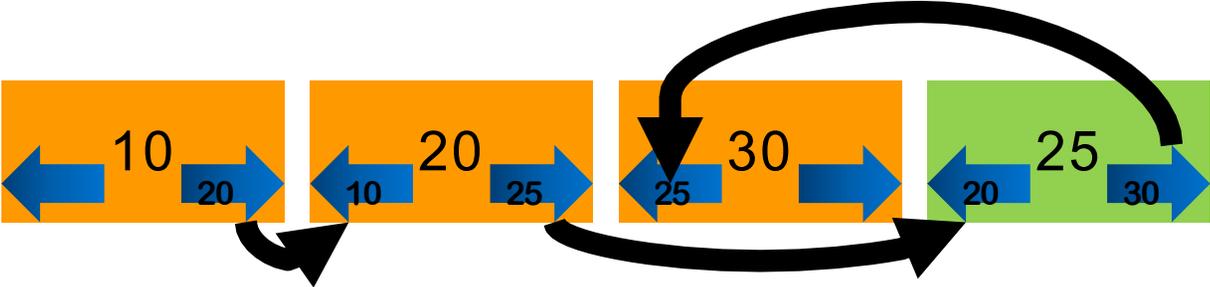
X\$BH
V\$BH

buffhan.sql

Buffer Header Management

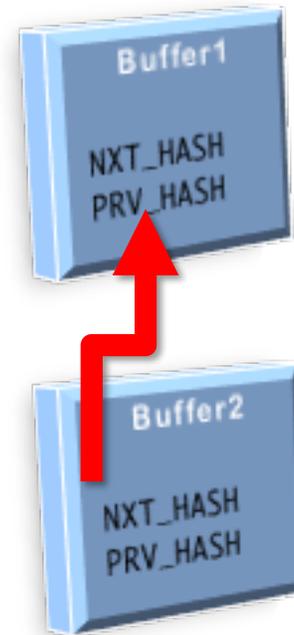


Linked List



When a new buffer comes in, only the pointers are updated

Linked List



Test for Buffer Header

```
select
  ltrim(addr, '0') buffer_address,
  ltrim(nxt_hash, '0') next_buffer,
  ltrim(prv_hash, '0') prev_buffer,
  case
    when nxt_hash = prv_hash then 'Unlinked'
  else
    'Linked'
  end
  as linked
from x$bh
where hladdr = '000007FF3C8B1568'
```

bh1.sql

Where did I Park My Car?

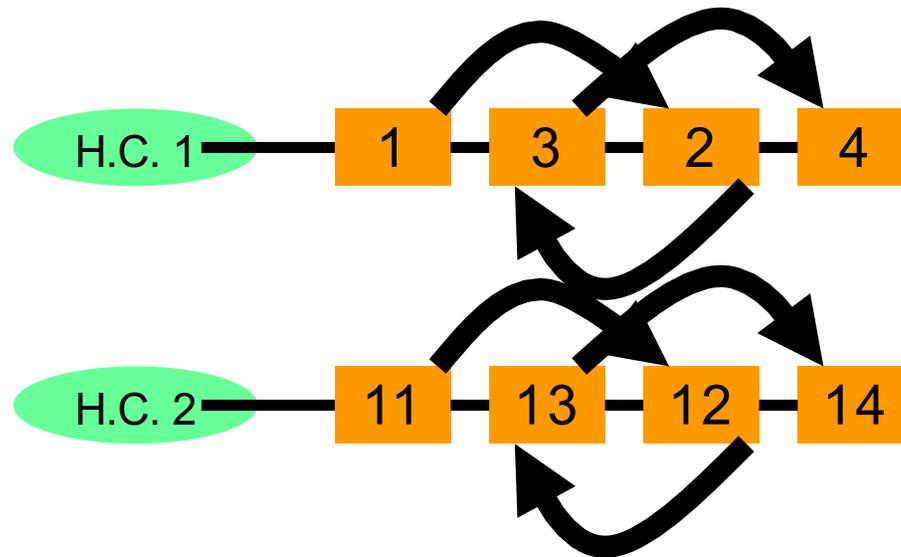


@ArupNanda

Arup Nanda

How Buffer Cache Works

Hash Chain



Data Block Address

- Get the relative file# and block#

```
select col1,  
       dbms_rowid.rowid_relative_fno(rowid) rfile#,  
       dbms_rowid.rowid_block_number(rowid) block#  
from latchtest;
```

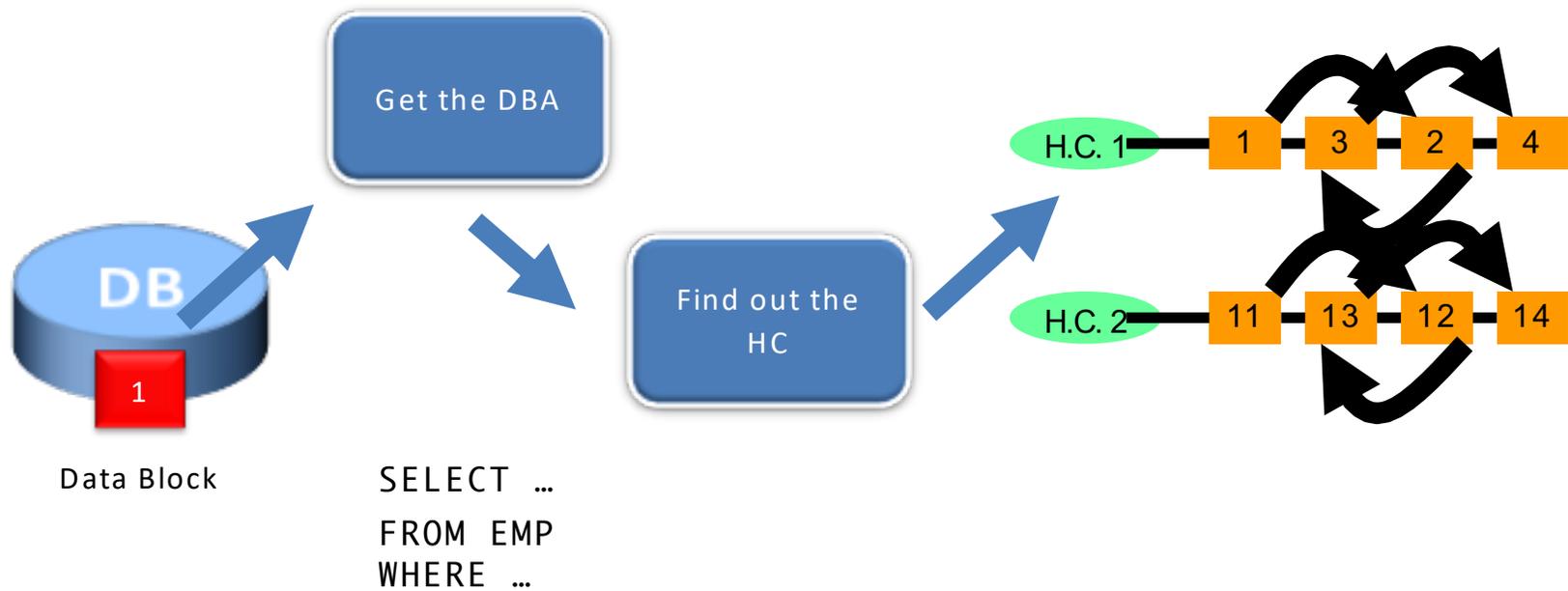
fbk.sql

- Get the DBA

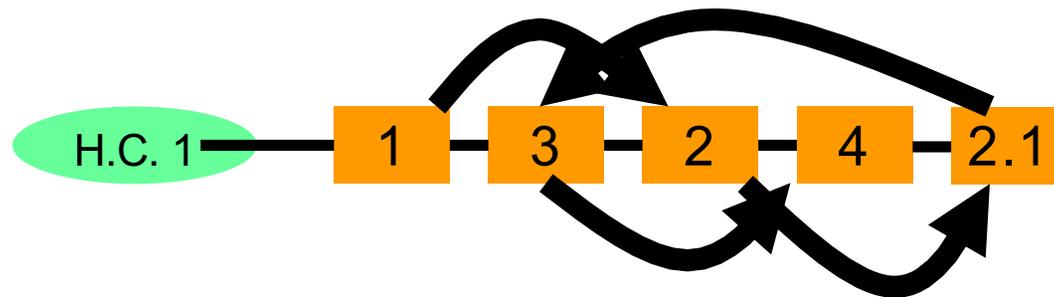
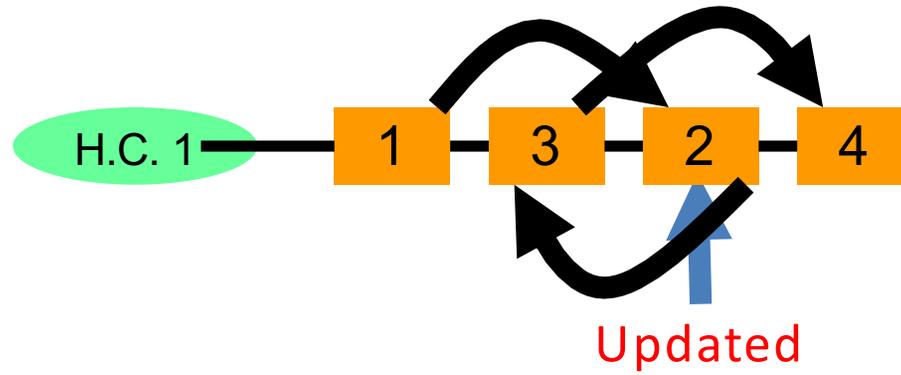
```
select dbms_utility.make_data_block_address  
(file#,block#) from dual;
```

get_dba.sql

Chain Placement



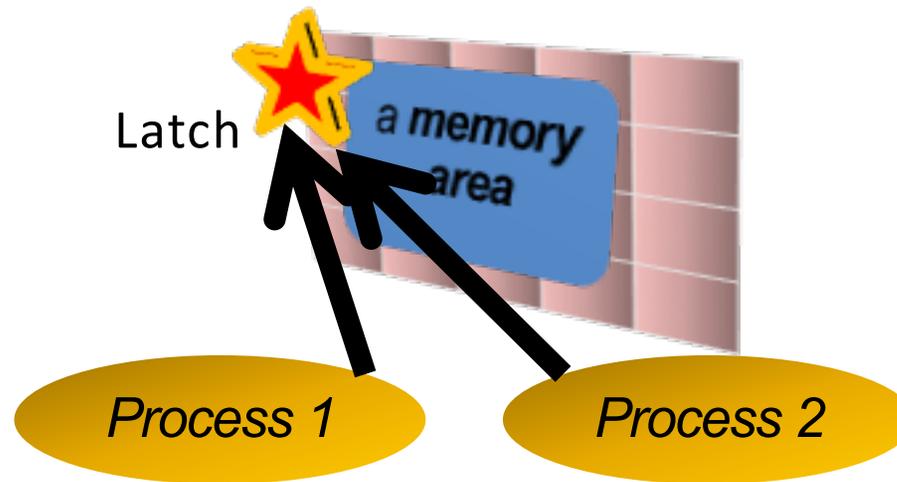
Buffer Clones



Buffer States

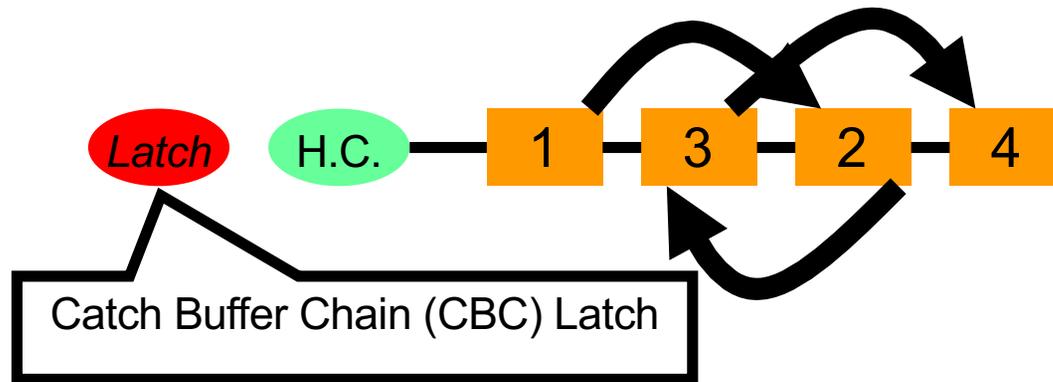
- What was the intention of the session when it brought the block to the buffer cache?
- If it was to merely read:
 - The state is CR – Consistent Read
- If it was to modify:
 - The state is CURRENT
- The column STATE
 - 1 – current
 - 3 - CR

Latches

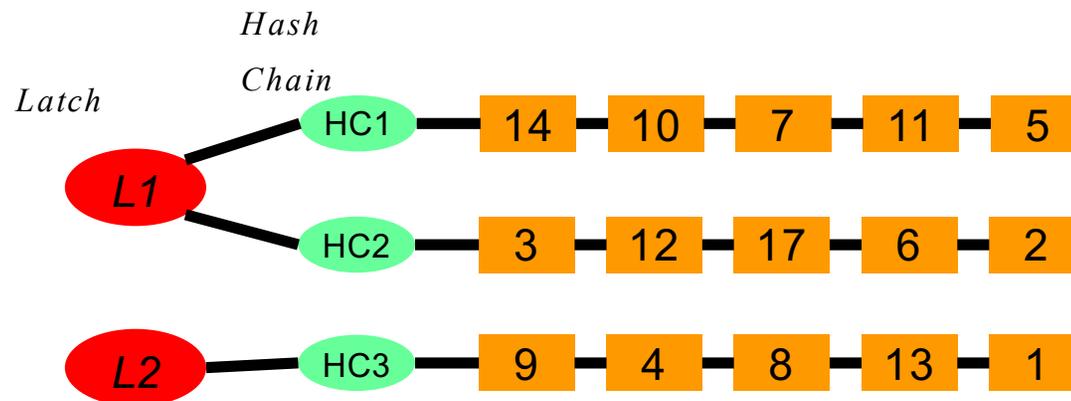


- Process 1 and 2 will try to get the “latch”, a area in memory that does not have any required data.
- Whoever gets the latch now gets to access the memory area exclusively
- When done, the process releases the latch

CBC Latch



Latches and Hash Chains



No. of hash buckets = `_db_block_hash_buckets`

No. of latches = `_db_block_hash_latches`

How to get the Hash Chain

- Dump the Buffer Cache
 - SQL> oradebug setmypid
 - SQL> oradebug dump buffers 4
 - SQL> oradebug tracefile_name
 - C:\APP\ARUP\VIRTUAL\diag\rdbms\al122\al122\trace\al122_ora_1132.trc
- Check the tracefile for the following:
CHAIN:

Here is the snapshot from the trace

CHAIN: 1030002 LOC: 0x00007FF95824EB48 HEAD:
[0x7ffa4cf94cc8,0x7ffa4ef9c5c8]

BH (0x7ffa4cf94c18) file#: 3 rdba: 0x00c0df25 (3/57125) class: 10

ba: 0x7ffa4c642000

set: 9 pool: 3 bsz: 8192 bsi: 0 sflg: 2 pwc: 0,0

dbwrid: 0 obj: 10634 objn: 10634 tsn: [0/1] afn: 3 hint: f

hash: [0x7ffa4cfc1928,0x7ff95824eb48] lru:
[0x7ffa4cf94e58,0x7ffa4cf94b98]

Buffer address from X\$BH

File# and Block# from X\$BH

And, latches?

- The column HLADDR
select hladdr
from x\$bh
where dbarfil = &file_no
and dbablk = &block_no
/
• This is a hexadecimal number

hladdr1.sql

How many buffers for a latch?

- This

select

```
    dbarfil      File#,  
    dbablk      Block#,  
    obj         Data_Obj#,  
    name        Object_Name,  
    subname     Sub_Obj,  
    tch         touch_count,  
    decode(state,1,'current',3,'CR',state)  
           state
```

```
from x$bh b, obj$ o  
where hladdr = '&hladdr'  
and o.dataobj# = b.obj
```

state1.sql

Identifying Buffer Latches

- Demo
 - Find out the rows and blocks – fblk.sql
 - Find out the data object id – dobjid.sql
 - Find out the data block address – get_dba.sql
 - Find out the child latch address – hladdr1.sql
 - Find out the objects protected by a latch – latchobjs.sql
 - Find out the total buffers per latch – clatchcount.sql

Summary

- Buffers are placeholders in memory
- Empty when the instance comes up
- Server process brings a data block from database to occupy a buffer
- There is no “database” of which block is in which buffer.
- A process has to scan the buffers to find what it needs
- The buffers are spread over several hash chains to help in searching
- A data block address (DBA) is fixed for the block
- DBA determines which chain the block may be found
- A latch prevents multiple processes from walking the chain



Thank You!

Blog: arup.blogspot.com

Tweeter: [@ArupNanda](https://twitter.com/ArupNanda)

Facebook: [facebook.com/ArupKNanda](https://www.facebook.com/ArupKNanda)

Download Scripts: proligence.com/pres/sangam18