Oracle SGA "Thrashing"

(aka buffer cache "thrashing")

by Mike Cuppett

SGA

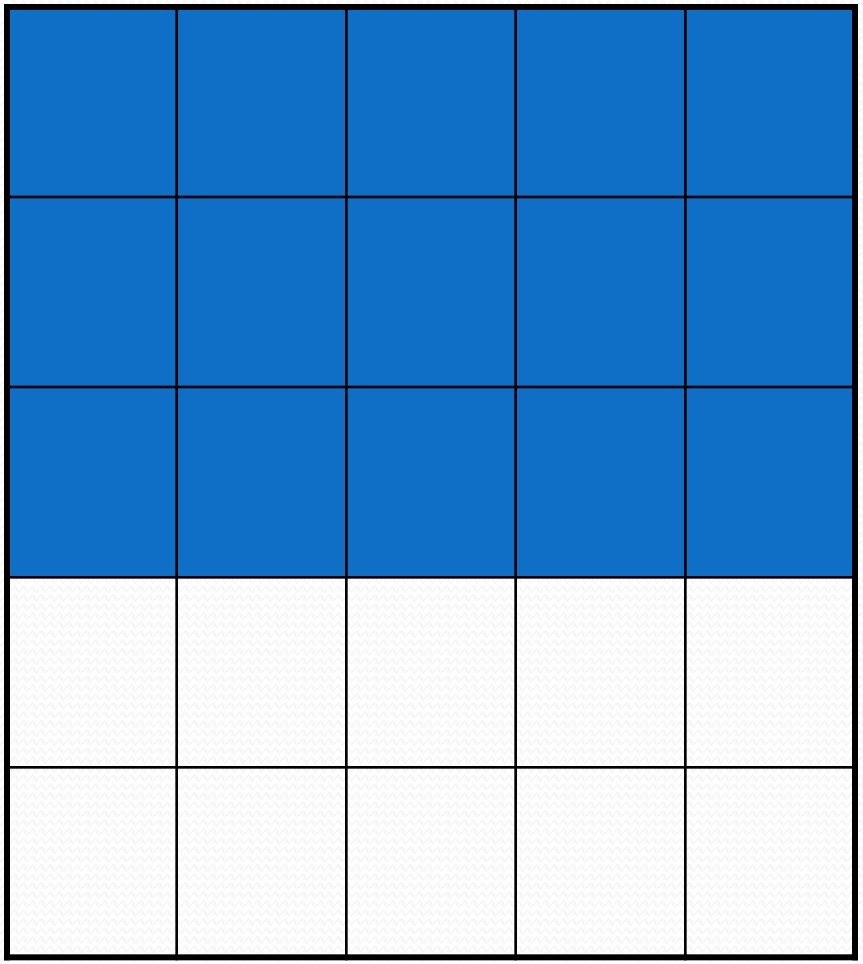
- Shared Global Area
  - Includes buffer cache (shared resource)
    - Can vary in size, but has finite limitation eventually
    - Key factor in database performance
  - Other SGA components
    - Out of scope for this presentation
  - Client Oracle version
    - 9.2.0.6 and 9.2.0.8 (Oracle mitigates this problem in later versions)

Buffer Cache

- Server (user session) processes read data into the cache
- Data changes occur in the cache
- DBWR process writes “dirty” cache blocks to files
  - Including uncommitted data, hence the reason for undo and redo stream
- “Thrashing” is a performance killer, but DBAs and developers can mitigate the negative impact

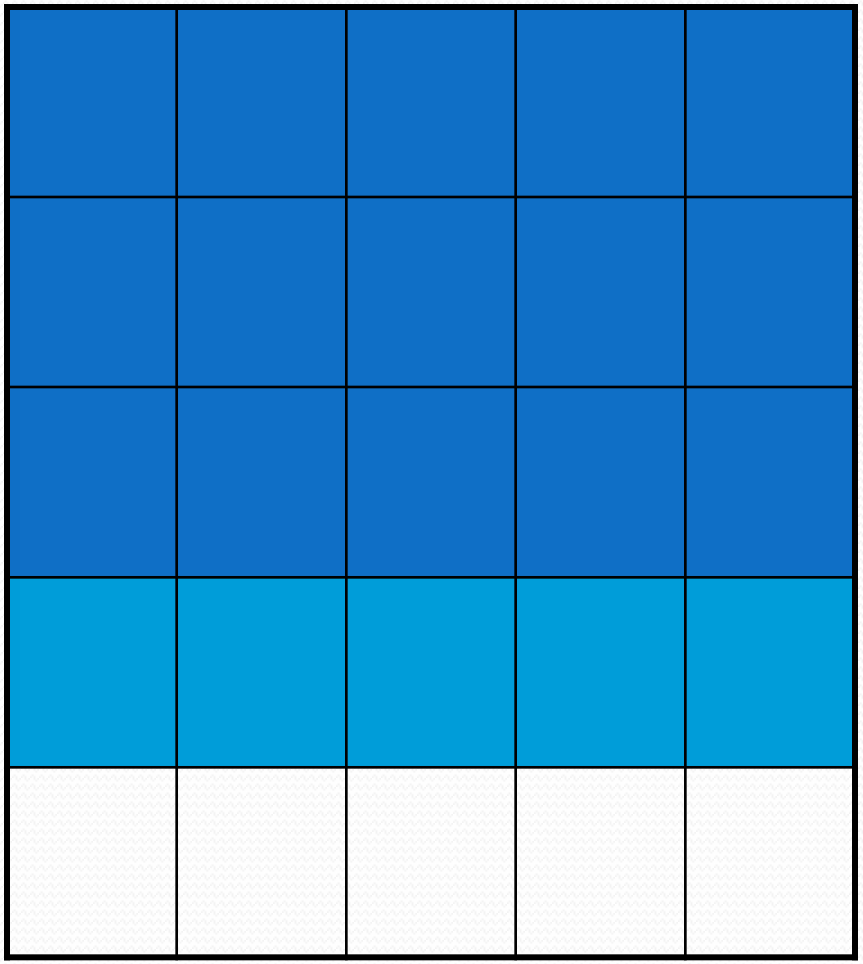
“Thrashing” Example

- Buffer cache holds 25 data blocks
- LRU manages the block releases
- User 1 SQL statement reads in 15 blocks



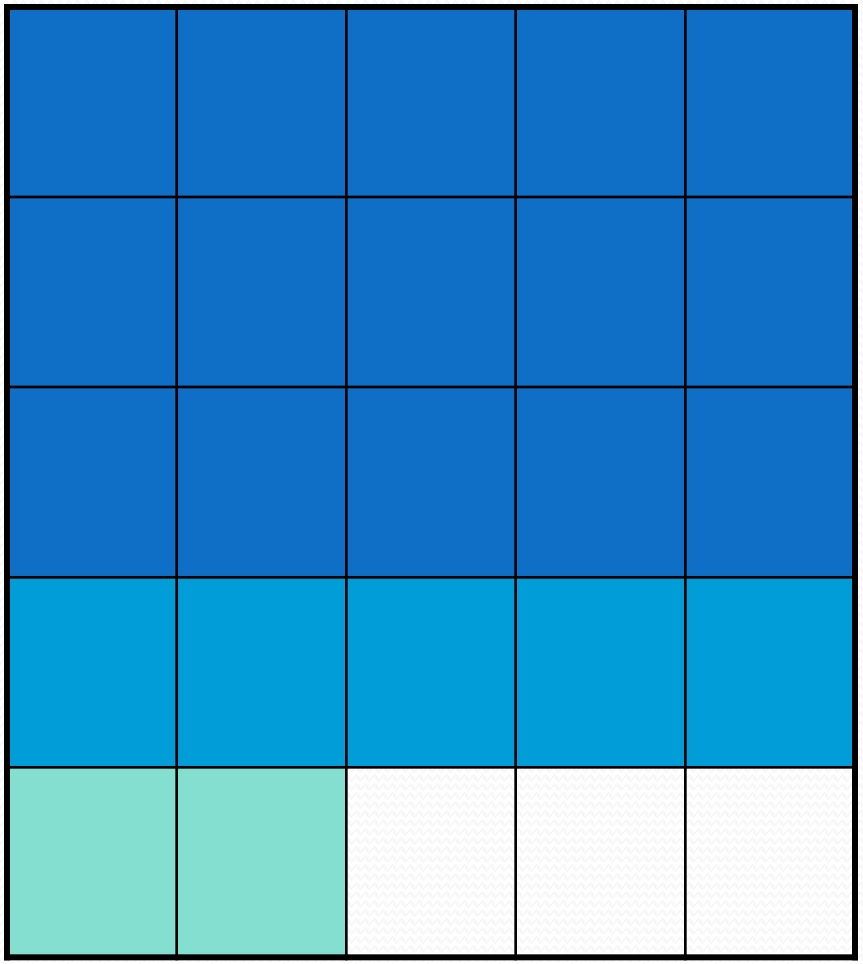
“Thrashing” Example

- User 2 SQL statement reads in 5 blocks – no problem



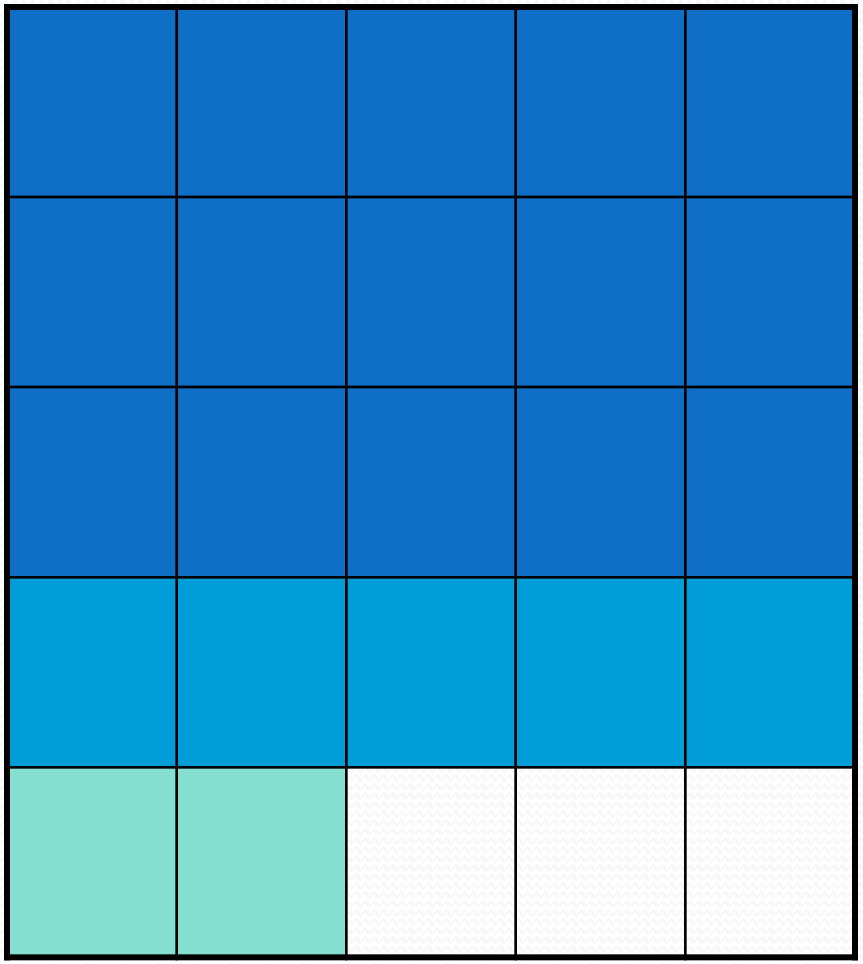
“Thrashing” Example

- User 3 SQL statement reads in 2 blocks – no problem



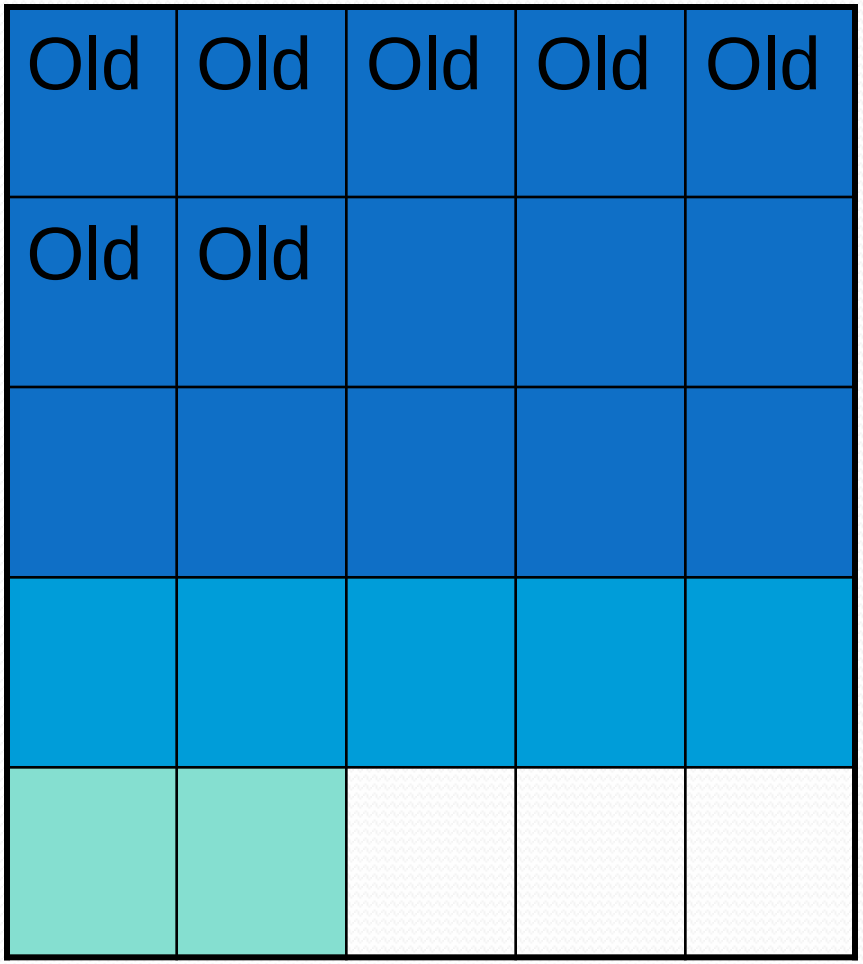
"Thrashing" Example

- User 4 SQL statement reads in 10 blocks – PROBLEM!
  - How does Oracle resolve this issue? The LRU mechanism goes to work identifying the oldest blocks for removal
    - Staying very high-level concerning the LRU algorithm



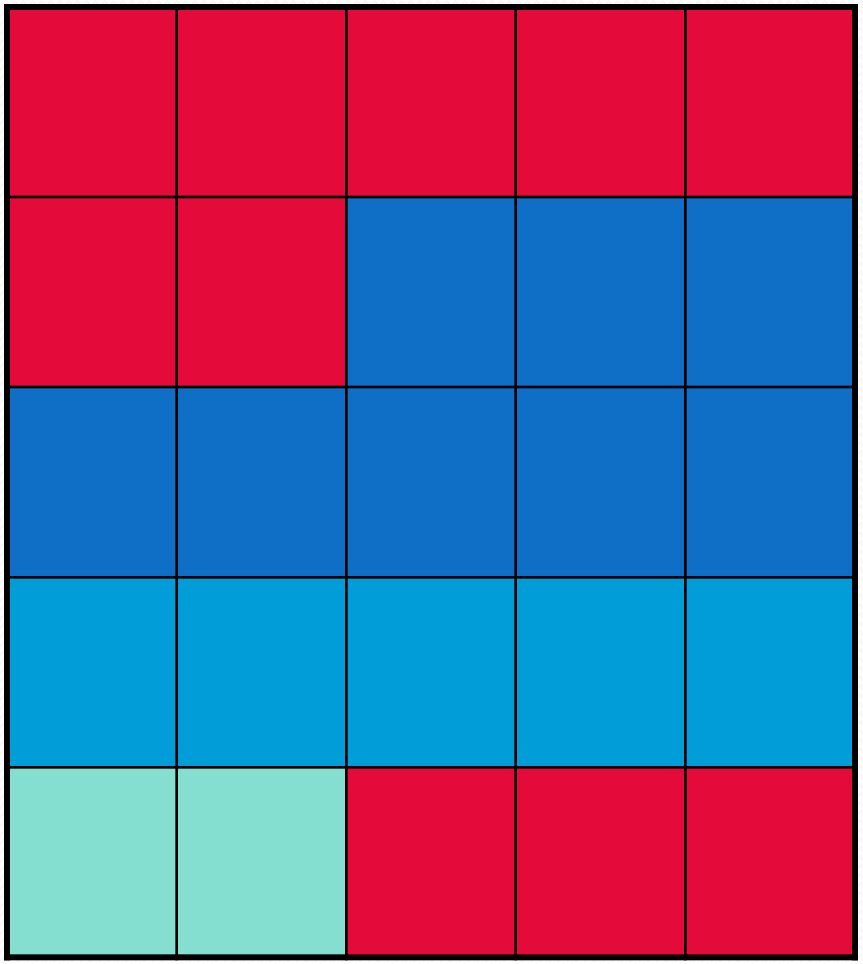
“Thrashing” Example

- LRU identifies 7 blocks for removal
- DBWR process writes out “dirty” blocks to files



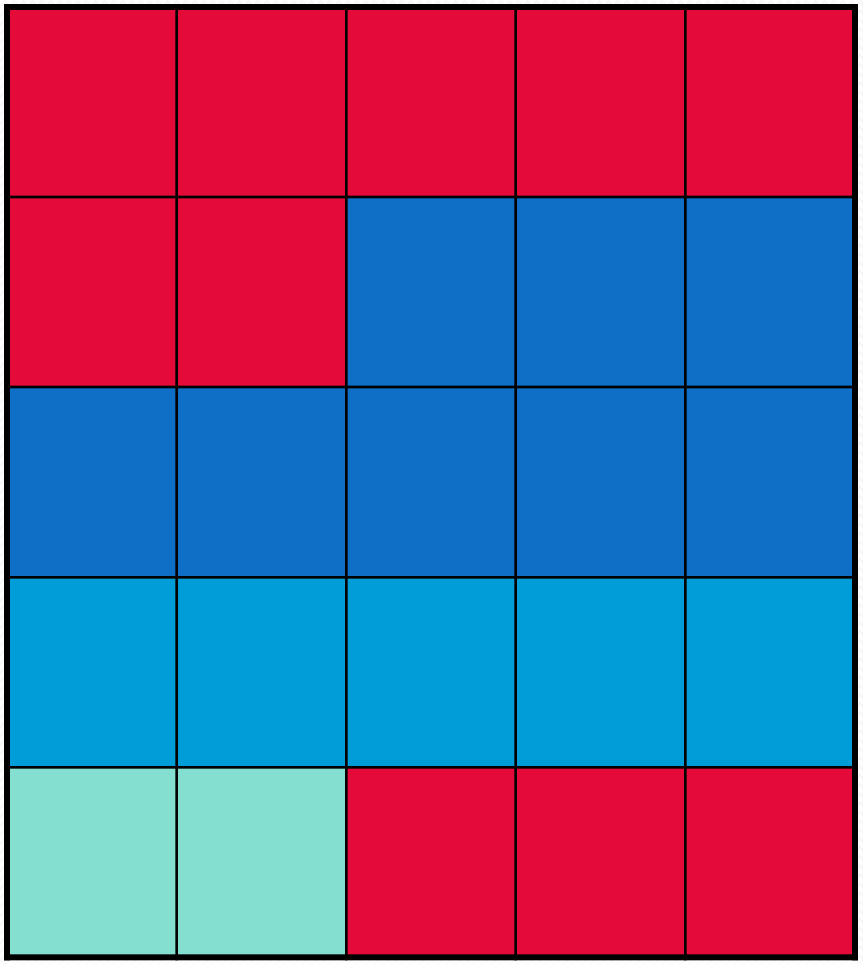
“Thrashing” Example

- User 4 SQL statement reads in 10 blocks – no problem, at least for user 4



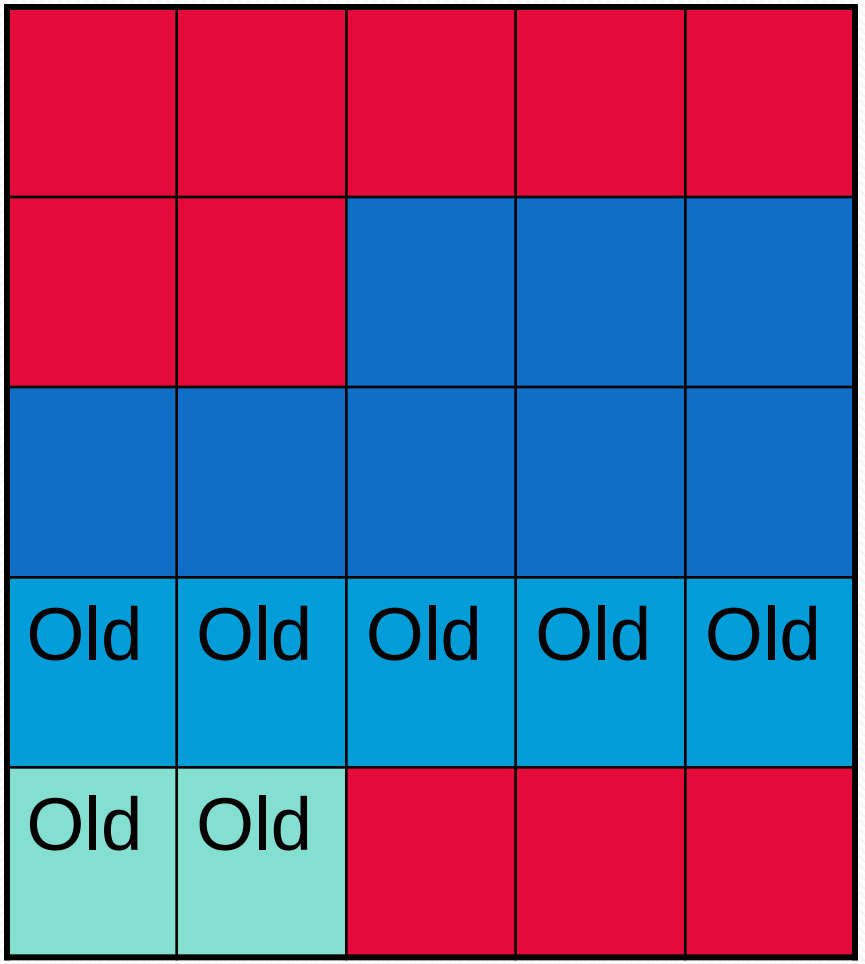
“Thrashing” Example

- User 1 still needs the data, but some of the data has been removed from memory
  - Data must be re-read into the cache
  - But to where?
  - Here we go again!



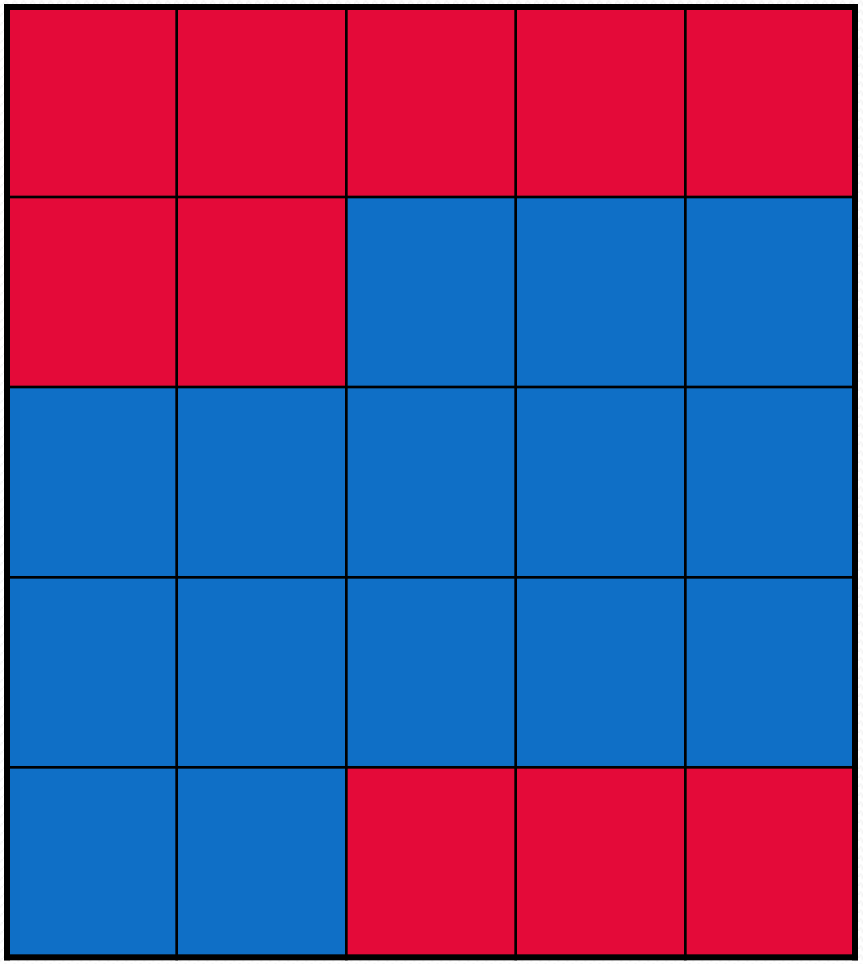
“Thrashing” Example

- LRU goes to work marking old blocks
- Fortunately, some of the user 1 data is still in memory, so it won’t be aged out
- Room is needed for 7 blocks



“Thrashing” Example

- User 1 data is back in memory
- But now, the data for users 2 and 3 are not in memory, so what if the data is still needed?
- That’s right, more reads
- DBAs call it “SGA or buffer cache thrashing” when user processes have to keep re-reading the data into the cache



“Thrashing” Scenario

- Think about an scenario using these values:
  - 1TB database size
  - 1000 user sessions a day
  - 1 million cache buffers allocated to the buffer cache
  - For a one day period, the database does **1 BILLION** physical reads (physical read = disk read)
  - For this example, let’s say each read is for one block
  - One query (really bad query) alone does 1,000,000 reads
  - How is that going to impact the other 999 sessions?
  - The above example is why DBAs talk about expanding the buffer cache and reducing data sets to reduce “thrashing”

“Thrashing” Prevention

- Manage the workload - # of jobs, reports, user sessions to reduce cache competition
  - Large batch jobs and reports that read a lot of data may cause user session data to be removed from memory – this is why a “mixed load” environment is NOT a best practice concerning performance
- Tune (expand) the buffer cache (DBA)
  - If RAM is available, consider expanding the buffer cache
- Reduce the data set (Developer)
  - Filter out as much data as possible to reduce the number of buffers needed
    - Also benefits performance concerning less physical reads, fewer disk sorts, etc

Conclusion

- DBAs can reduce “thrashing” by expanding the buffer cache
- Developers can reduce “thrashing” by minimizing query result sets