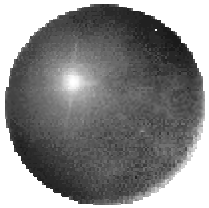


Applying a Blockcentric Approach to Oracle Tuning

Daniel W. Fink

www.optimaldba.com

Overview

- What is Blockcentric Approach?
  - Shifting Focus for Architectural and Tuning Decisions
- Myths and Fallacies
  - "Burn him at the stake!"
- Applying the Method
  - Will be addressed throughout the presentation

Trust, But Verify

- Information presented is general
- It will not cover all scenarios all databases
- Only covers 1 aspect of tuning and architecture

What is the Blockcentric Approach?

- Closely tied to Oracle I/O subsystem
  - Protecting memory structures
  - Single and Multiple Read operations
- Shifts focus from Rows to Blocks
  - Many traditional methods focus on rows
  - I/O system understands blocks
- Based upon many sources

Oracle's I/O Subsytem

- Blocks not Rows
  - Blocks are read, not rows
  - Larger block sizes means more rows per block
- Logical v. Physical
  - Logical is not just memory access
  - Physical is not always disk access
- Full Scans

Logical I/O

- Not just memory access
- Accessing Block in memory
- Requires locking of various memory structures

Physical I/O

- Not always disk access
- Block not found in memory
- Could be found in os buffer cache or prefetched from disk

Full Scans

- Able to read multiple blocks at one time
- Full Table Scan
  - Read all blocks up to the High Water Mark
- Fast Full Index Scan
  - Can access non-leading columns in concatenated indexes

High Water Mark

- The High Water Mark is the last block ever formatted to contain data.
- If the table has experienced high delete activity, the HWM may be set artificially high
- If indicated, the only solution is to reorganize the table

High Water Mark

H	D	D	D	D							
											HWM

Myths and Fallacies

- 99.999% Buffer Cache Hit Ratio is GOOD
- % of Rows Returned determines index usage
- Index all WHERE columns
- Full Table Scans are BAD
- DUAL is a low cost, useful table

Buffer Cache Hit Ratio

- Higher is not always better
- What happens when 99% is not good enough?
- Memory Speed Fallacy
- Determining Statement Cost

Higher Is Not Always Better

- It is an indication of hard work, not necessarily smart work
  - 99.999% BCHR requires 100,000 I/Os
- Not a good response to user performance problems

BCHR Method

- Statement #1 – 98% BCHR
  - 5000 Logical I/Os
  - 100 Physical I/Os
- Statement #2 – 50% BCHR
  - 400 Logical I/Os
  - 200 Physical I/Os

What Happens When 99% Is Not Good Enough?

- Everything is tuned, but the system is still slow…
- Add
  - More Memory
  - Faster CPU
  - Faster Disk
  - Faster Network
- Makes hardware sales reps happy…

Memory Speed Fallacy

- Disk access is 10,000 times more expensive than Memory Access
  - Measured in a 'pure' environment
- In practice, the ratio is closer to 40
  - Request does not go directly to block in cache
  - The requested block must be located
  - Memory structures must be accessed and manipulated
  - Disk access may not be actual disk access

Determining True Cost of Statements

- Compute total cost in terms of I/O
- Logical I/O costs 1
- Physical I/O costs 40
- Cost = LIOs + (PIOs * 40)

Blockcentric Method

- Statement #1 – 98% BCHR
  - 5100 Logical I/Os
  - 100 Physical I/Os
  - Cost = 5100 + (100 * 40) = 9100
- Statement #2 – 50% BCHR
  - 400 Logical I/Os
  - 200 Physical I/Os
  - Cost = 400 + (200 * 40) = 8400

% Of Rows Returned

- Index usage
- Number of Blocks Returned
- Data Considerations

Index Usage

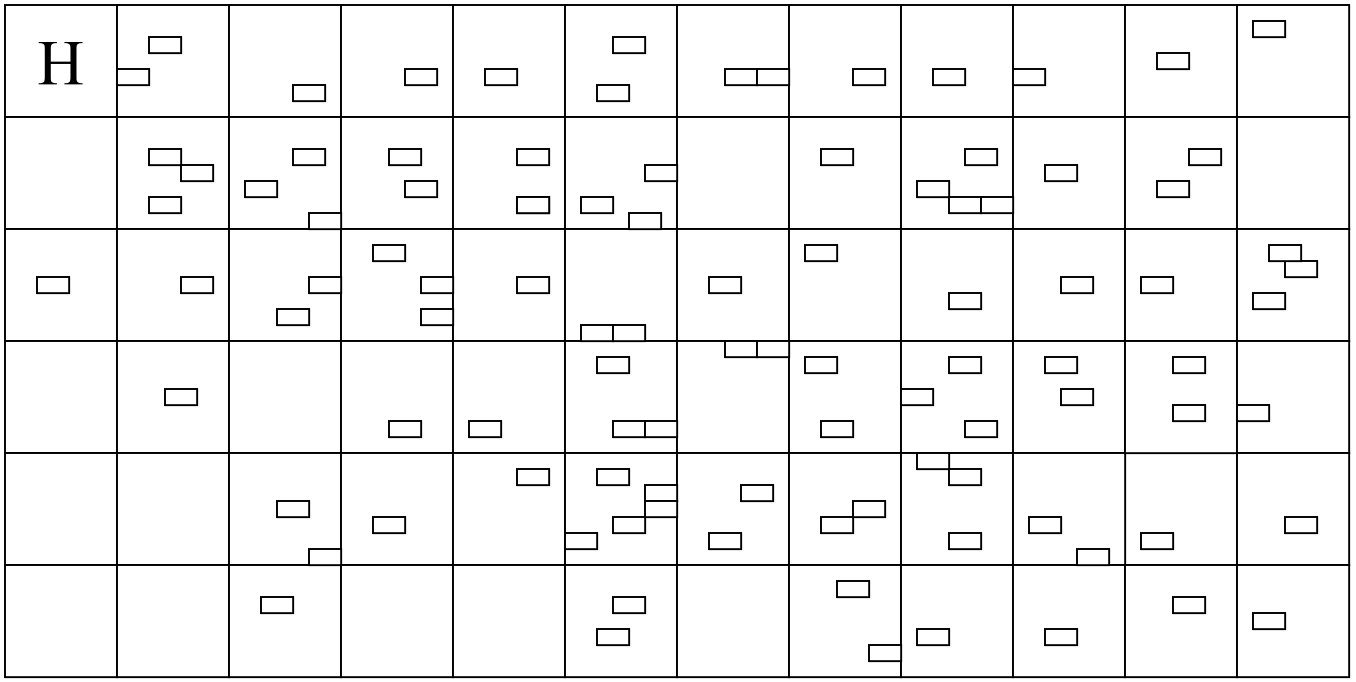
- Use index if % of rows returns is less than
  - 15% for nonparallel
  - 5% for parallel
- Assumes
  - rows are clustered
  - rows are read

Number Of Blocks Returned

- Focus on number of blocks to be read
  - Data scattered – many blocks to be read
  - Data clustered – fewer blocks to be read
  - Data fragmented – many/fewer blocks to be read
- Bottom Line – Reduce the Number of Blocks read

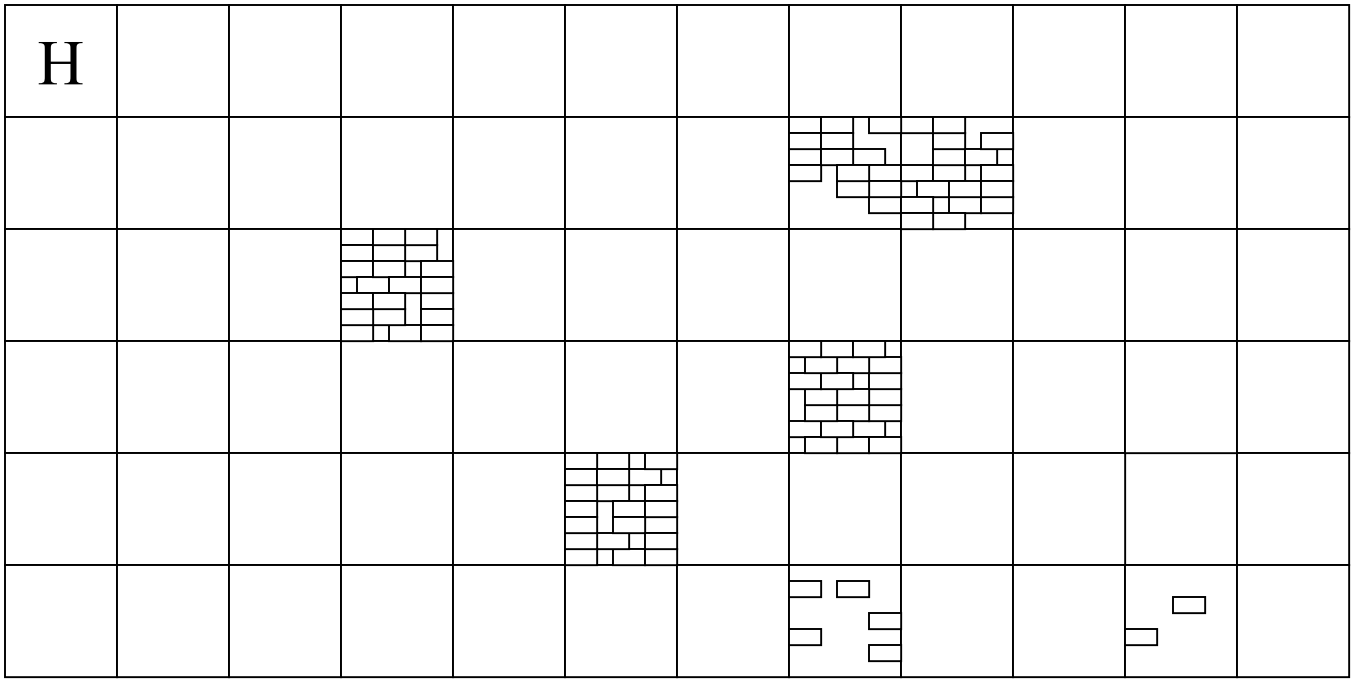
Scattered Data

- Data of interest is scattered among many blocks



Clustered Data

- Data of interest is present in a few blocks



Fragmented Data

- Blocks containing data are interspersed with empty blocks

H							D				
			D								
						D					
			D								D
										D	

Continued Data

- Rows are migrated or chained
- Multiple I/O operations

Index all WHERE Columns

- Indexes may speed queries
  - But will always slow other operations
- Unique constraints require indexes
- Index Foreign Keys
  - Locking problems
  - Join column

Full Table Scans are BAD

- It depends
- Focus on # of I/O operations
  - Multiblock reads
  - Data and Block clustering
- Focus on ends, not means

Determining Block Selectivity

- Identify the data to be returned by focusing on the WHERE clause
- Calculate the Actual Blocks to be accessed
  - # of blocks below HWM
  - # of blocks containing data
- Determine how many block selectivity
  - # of Rows with value / # of Blocks with Row
  - # of Rows with value / # of Blocks below HWM

DUAL

- Table often used for non-SELECT operations
- Very expensive,
  - Very few I/Os per access
  - May actually comprise the largest I/O consumer
- Solutions
  - Rewrite code
  - Create virtual table

Miscellaneous Tuning Tips

- Focus on overall length of operations
- Reduce the I/Os
- Identify resource intensive SQL, tune it first
- Table and Index Maintenance
- Make extent sizes multiples of
  `db_block_size *`
  `db_file_multiblock_read_count`

Focus on overall length of operations

- Record ‘Bad’ time
- Decide on ‘Good’ time
- Identify where statement is spending most time and tune accordingly
- When Good time is achieved, stop and move on to the next one

Reduce the I/Os

- Avoid the BCHR Myth
  - BCHR is not useless, it may indicate problems
    - Excessively High
    - Excessively Low
- Identify the most expensive statements
  - Find most I/O Intensive
  - Reduce Logical I/Os…Physical will follow
- Reorganize tables when needed
  - High Water Mark
  - Data or Blocks Not clustered

Identify resource intensive SQL – tune it first

- Look for statements consuming most I/O
  - Logical Reads + Physical Reads
- Identify tables with most activity
- Don’t tune 1 query to the detriment of 9 others

Table and Index Maintenance

- 'Sort tables according to primary key value' – Fallacy
- Sort tables to achieve common clustering
- Rebuild tables and indexes to
  - Reduce storage requirements
  - Reset high water mark

Extent Sizing

- Extent sizes should be multiples of multi block read configuration
  - Multiblock read configuration should match hardware limitations
- If extent is 1 block less, multiple I/O calls will be executed
  - Even if ‘extra’ blocks are empty, it may be more efficient

Sources

- "Oracle Performance Tuning 101" Vaidyanatha, Deshpande, Kostelac, Jr.
- "Oracle SQL Tuning Pocket Reference" Gurry
- Performance Sites
  - www.hotsos.com - Cary Millsap
  - www.orapub.com - Craig Shallahamer
  - www.evdbt.com - Tim Gorman and Jeff Maresh
  - www.oraperf.com - Anjo Kolk
- Rocky Mountain Oracle User Group Training Days 2003 – March 5 & 6

Questions?

or

Comments!