Meeting 11/8/17

Wednesday, November 8, 2017 11:00 AM

Agenda:
Spend one more day on data blocks

Next week:

- start scripts for measuring i/o
- Use same table space when comparing stuff
- Be prepared to talk about what we learned
- Create PowerPoint within next *two weeks*

Notes:
https://www.red-gate.com/simple-talk/sql/oracle/compression-oracle-basic-table-compression/

- "compression" is actually data de-duplication on the block
- Oracle doesn't decompress the table blocks, it simply re-constructs the row you need by hopping back and forth between the row directory and the row pieces
- It's possible to spend a significant amount of extra CPU time bouncing around a block to reconstruct a row - this could have a noticeable impact on CPU usage if you are doing lots of **tablescans**.
    - Because Oracle will have to hold (pin) the block for some time to reconstruct rows, you may find that your code will do fewer "consistent gets - examination," which means more activity on the "cache buffer chains" latch.
- Basic compression is only used with direct path inserts, not with ordinary DML (data manipulation language statements), and Oracle will, by default, set pctfree to zero on the table, which is a good indication that you are not supposed to modify the data once you've created it - basic compression is really only appropriate for data that is virtually read-only.
- The stored column order can change from block to block, allowing Oracle to maximize the options for creating tokens that represent multiple adjacent columns.
    - This deduplication mechanism means that Oracle doesn't have to decompress blocks, it keeps the blocks in the buffer cache just like any other block, but reconstructs rows (in the PGA) by following pointers to tokens - and chasing pointers is a CPU intensive process; the better your compression the more CPU intensive your queries (tablescans) are likely to be.

Questions:

- How many columns can you have before column order starts to matter?

- If you don't specify the order of columns, non-optimized, what's the CPU cost versus putting the columns in the optimized order?

- Can you minimize CPU overhead if you pre-arrange the columns?

Tests:
Trevor: Create a table with 5-6 columns, see how the Basic Compression optimizer organizes the columns
Dylan: Create a table with 5 columns and see what effect Advanced Compression optimizer has on CPU
Alex: Prove or disprove column order changes between two blocks (Basic Compression)

Try in a 5-8 column table