Meeting 10/25/17

Wednesday, October 25, 2017 11:00 AM

Attendees: Kirby, Andy
Trevor, Dylan, Alex

We will be measuring performance in Oracle Database.

They are currently employing Advanced Row Compression

HP currently has advanced row compression (OLTP) (row store compress advanced)
Structured data

Figure out the best way to compress and store data
- Reduce IO
- See how it affects CPU
- Store as much data in as little space as possible

What Kirby wants: no compress, basic compress, adv compress
Create tables that way, find block, open it up and read it
Figure out pieces of block header to figure out what thye mean / what they're doing

Data guard and data pump in future

Look into compression advisor (?)

Figure out how to read a block

Requirements:

Set up environment - due date: 11/1

Have a foundational understanding of blocks (good enough to reverse engineer blocks,) 11/15 Rick Van Dyke, Tim Goreman, oracle-base Tim Hall. Can trust jonathan lewis; tenel poder, tom kite carrie osbourn. Earlson - might be right

Reverse engineer data block by Winter break 11/15

Reverse engineer compression algorithms 11/29

Design experiments (based on findings) 1/17

Develop CPU, I/O, Mem usage tracking scripts 1/24

Analyze/Summarize results 2/15

Present conference Hotsos conference March

Last term is implementation (finish by June)

Stretch goals: network, index, storage compression

Compression options -->

Things we may want to explore:

*Basic compression (free)
-non-enabled
-enabled
-maybe change the order of the columns
-sort the columns that can be sorted

*Advanced compression (paid from here on out)
-compress (direct path inserts only, optimized for data warehouse systems)
-compress for all operations (suitable for OLTP, compression for all operations, including regular DML statements)

*HCC (Hybrid Columnar Compression) -- not an option. Don't have exadata
-normal
-query low
-query high
-archive low
-archive high

*In-memory compression (explored by earlier project)
-no memcompress
-memcompress for dml
-memcompress for query low
-memcompress for query high
-memcompress for capacity low
-memcompress for capacity high
-can also do it on a column by column basis

*Automatic Data Optimization (ADO with heat map)
-different compressions after certain periods of time
-uncompressed
-advanced
-HCC query low*
-HCC archive high*
-etc
-can move data to different storage tiers after certain periods of time

*Compression for backup data (RMAN)
-low
-medium
-high

*Advanced index compression (really need to fine tune it on a index by index basis)
-compress (basic)
-advanced low
-advanced high
-Can also do different compression levels for different partitions

*Advanced network compression (stretch)