PROGEEKING.COM

ABOUT ME

- Nikolay Kovachev - @KovachevPro
- Senior DBA @ TechnoLogica LTD
- Holder of more than 10 Oracle Certificates
  - Including Exadata Expert
- Blogger - PROGEEKING.COM

ORACLE Certified Expert
Oracle Exadata X3 Administrator

ORACLE Certified Expert
Oracle Database 11*g* Performance Tuning

ORACLE Certified Expert
Oracle Database 11*g* Release 2 SQL Tuning

ORACLE Certified Expert
Oracle Real Application Clusters 11*g* and Grid Infrastructure Administrator

ORACLE Certified Professional
Oracle Database 11*g* Administrator

ORACLE Certified Professional
Oracle Database 12*c* Administrator

ORACLE Certified Associate
Oracle Database 11*g* Administrator

ORACLE Certified Expert
Oracle Database SQL

ORACLE Certified Associate
Oracle Solaris 11 System Administrator

ORACLE Certified Specialist

ORACLE PartnerNetwork Certified Specialist

Techno Logica

ORACLE Platinum Partner
Specialized
Oracle Exadata Database Machine

Exadata

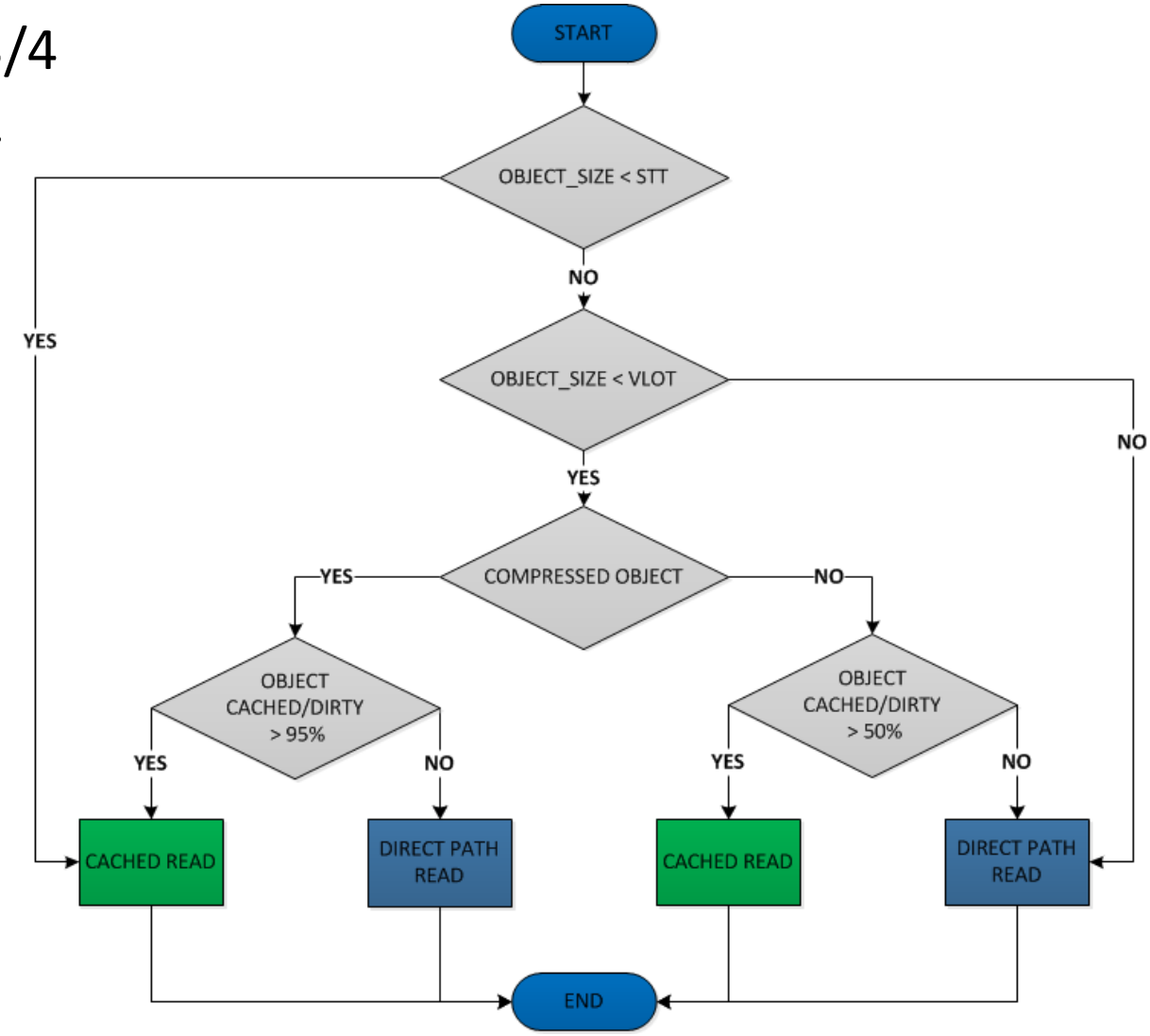
Must Know Internals

- **Cell Offloading**
  - The algorithm basics and some parameters
- **Hybrid Columnar Compression**
  - Compression levels, optimizations and how they work
- **Storage Indexes**
  - Structure, Maintenance and Control

Exadata
Cell Offloading

- **The Algorithm basics /v 11.2.0.3<>12.1.0.1/**
  - **Threshold Parameters:**
    - _small_table_threshold – STT
      - 2% of the buffer cache
      - Below - always perform cached read
    - _very_large_object_threshold – VLOT
      - 5 X buffer cache
      - Above - always perform DPR/direct path read/
  - **Thresholds between STT and VLOT:**
    - Non compressed objects: 50% dirty/cached blocks
    - Compressed objects: 95% dirty/cached blocks

- 11.2.0.3/4
- 12.1.0.1



- **12.1.0.2:**
  - **Threshold Parameters:**
    - _small_table_threshold – STT
      - 2% of the buffer cache
      - Below - always perform cached read
      - Above use ABTC
  - **New algorithm - Automatic Big Table Caching - ABTC**
    - db_big_table_cache_percent_target
      - Default: 0 – Disabled
    - Use object temperature for cache decisions

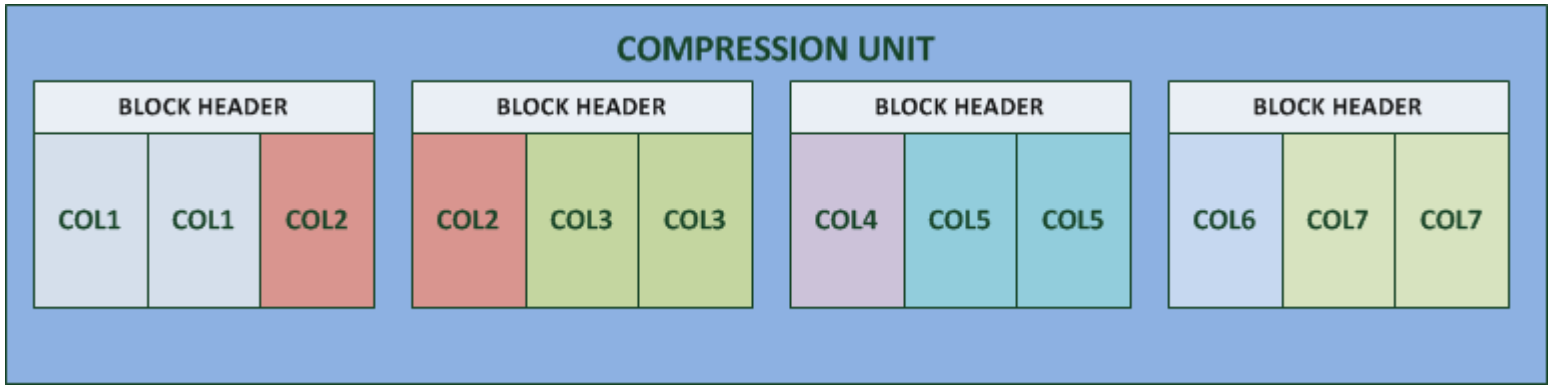
- **What about RAC & In-Memory Parallel?**
  - **Same threshold parameters**
    - _small_table_threshold – still valid
    - _very_large_object_threshold – till 12.1.0.1
  - **Thresholds between STT and VLOT:**
    - 10% smaller than single instance
  - **It checks only the local cache!**
    - The remote cache it's not checked for blocks!
  - **ABTC**
    - It’s not working for serial executions on RAC!

- **Controlling DPR/Offloading**
  - **Parameters:**
    - _serial_direct_read
      - It's underscore, but documented:
        - OU Exadata courses, MOS notes,Exadata Documentation
        - But still underscore
      - Values: auto, never, always
    - _direct_read_decision_statistics_driven
      - If true use statistics instead of the header
      - By default on true from version 11.2.0.2

Exadata

Hybrid Columnar Compression

- **HCC Basics**
  - **Compression units /CU/:**
    - Typically 32kb in size
    - Columnar format
  - **Single row access can read whole CU**
  - **Update = CU Lock**
    - 12.1.0.2 – row level locking!



- **Compression levels:**
  - Query Low:

```
oracle  oracle                [.] lzopro_lzo1x_1_12_compress_core
|
--- lzopro_lzo1x_1_12_compress_core
    lzopro_lzo1x_1_12_compress
    kgcclzodo
    kgcclzopseudodo
    kgccdo
```

  - Query High:

```
oracle  oracle                [.] deflate_slow
|
--- deflate_slow
    |
    |--------- deflate
    |          kgcczlibdo
    |          kgcczlibpseudodo
    |          kgccdo
```

- **Compression levels:**
  - Archive Low:

```
oracle  oracle                [.] deflate_slow
|
--- deflate_slow
    |
    |--------- deflate
    |          kgcczlibdo
    |          kgcczlibpseudodo
    |          kgccdo
```

  - Archive High:

```
oracle  oracle                [.] fallbackSort
|
--- fallbackSort
    BZ2_blockSort
    BZ2_compressBlock
    handle_compress
    BZ2_bzCompress
    kgccbzip2do
    kgccbzip2pseudodo
```

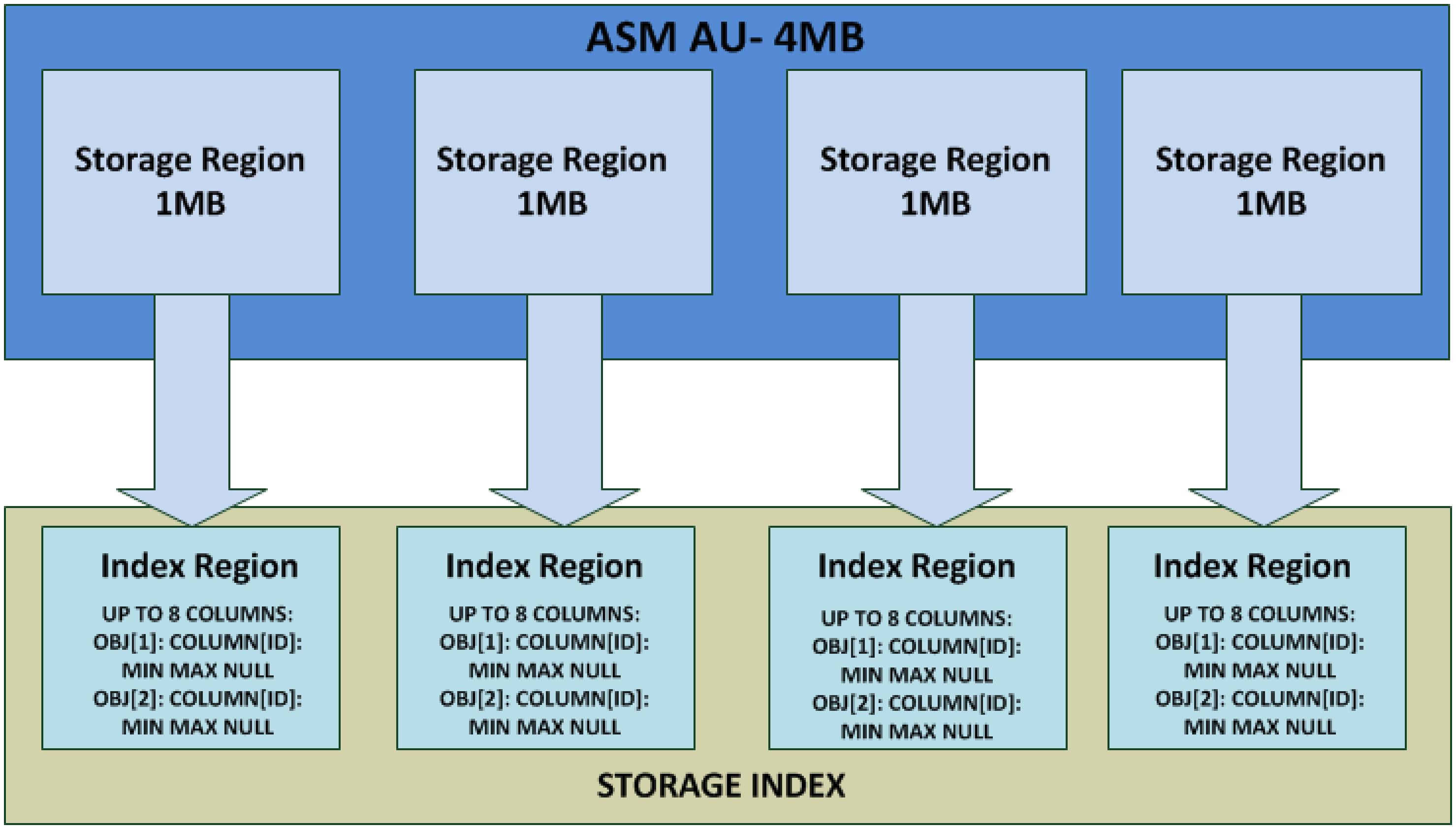
- **Single row access:**
  - Hint CLUSTER_BY_ROWID in version 11.2.0.4
  - Parameter _optimizer_cluster_by_rowid in 12.1.0.1

```
SQL> select /*+ CLUSTER_BY_ROWID(t) */ count(b)  from hcc t where a='asdf';

  COUNT(B)
----------
       336


Execution Plan
----------------------------------------------------------
Plan hash value: 2611202939

-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |         |     1 |     8 |   532   (0)| 00:00:07 |
|   1 |  SORT AGGREGATE               |         |     1 |     8 |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID| HCC     |   336 |  2688 |   532   (0)| 00:00:07 |
|   3 |    SORT CLUSTER BY ROWID      |         |   336 |       |     5   (0)| 00:00:01 |
|*  4 |     INDEX RANGE SCAN          | HCC_IDX |   336 |       |     5   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

Exadata

Storage Indexes

- # Storage Index Structure



- **Storage Index Creation**

```
00000656: BEFORE UPDATING RIDX SUMMARY
00000658: SQLID 0ury85nym5qkr Dumping RIDX for Read before update
RIDX(0x7f2a5eeb8530) : st 1 (RIDX_INVALID) validBitMap 0 tabn 0 id {0 0 0}
RIDX: strt 0 end 0 offset 0 size 0 rgnIdx 0 RgnOffset 0 scn: 0x0000.00000000 hist: 0
RIDX validation history:
0:Undef 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
OCL_RIDX: 0x7f2a5eeb8528 magic:caf1 rgnidx:2149 rgnhdlid:1state: 32 invalidatereason:7f2a5eeb8530 RIDX:
2014-02-09 20:43:18.760255 :0000065A: SQLID 0ury85nym5qkr ridxp 0x7f2a5eeb8530 [st: 1] (0, 2048) [io strt
 0 end 2048] isRead 1 newColCached 1 newColFound 0 someColNotFound 0 isRidxValid 0 overWriteSummary 1
colInRidxNotInPredCol 0 objd 87427 RIDX:diskOffset 2253389824 RIDXCtx:diskOffset 2253389824 RIDXCtx:
ioSize 1048576 regionSize 1048576
0000065B: AFTER UPDATING RIDX SUMMARY
RIDX(0x7f2a5eeb8530) : st 2 (RIDX_VALID) validBitMap 0 tabn 0 id {87427 7 2087871281}
RIDX: strt 0 end 2048 offset 2253389824 size 1048576 rgnIdx 2149 RgnOffset 0 scn: 0x0000.0011fdcb hist: 2
RIDX validation history:
0:FullRead 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [2] numFilt 4 flg 2 (HASNONNULLVALUES):
lo: 41 43 43 45 53 53 24 0
hi: 74 69 74 6c 65 39 33 5f
```

- **Storage Indexes and Data Density**

```
RIDX(0x7ffd3aecb474) : st 2 (RIDX_VALID) validBitMap 0 tabn 0 id {87410 4 2087871281}
RIDX: strt 0 end 2048 offset 1874853888 size 1048576 rgnIdx 1788 RgnOffset 0 scn: 0x0000.0011b157 hist: 2
RIDX validation history:
0:FullRead 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [4] numFilt 4 flg 2 (HASNONNULLVALUES):
lo: 43 4c 55 53 54 45 52 0
hi: 54 59 50 45 32 20 55 4e
RIDX(0x7ffd3aec98f4) : st 2 (RIDX_VALID) validBitMap 0 tabn 0 id {87410 4 2087871281}
RIDX: strt 0 end 2048 offset 1876951040 size 1048576 rgnIdx 1790 RgnOffset 0 scn: 0x0000.0011b157 hist: 92
RIDX validation history:
0:FullRead 1:FullRead 2:FullRead 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [1] numFilt 3 flg 2 (HASNONNULLVALUES):
lo: 41 50 45 58 5f 30 33 30
hi: 58 44 42 0 0 0 0 0
Col id [2] numFilt 2 flg 2 (HASNONNULLVALUES):
lo: 41 4c 45 52 54 5f 51 54
hi: 78 64 62 2d 6c 6f 67 31
Col id [4] numFilt 4 flg 2 (HASNONNULLVALUES):
lo: 43 4c 55 53 54 45 52 0
hi: 54 41 42 4c 45 20 53 55
```

- **Storage Indexes and Maintenance**
- Before Update:

```
RIDX(0x7fad85ecfe1c) : st 2(RIDX_VALID) validBitMap 0 tabn 0 id {87427 7 2087871281}
RIDX: strt 32 end 2048 offset 2156937216 size 1032192 rgnIdx 2057 RgnOffset 16384 scn: 0x0000.0012262f hist: 1
RIDX validation history:
0:PartialRead 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [2] numFilt 4 flg 2 (HASNONNULLVALUES):
lo: 41 43 43 45 53 53 24 0
hi: 78 64 62 2d 6c 6f 67 31
```

- After Update of column [1] – not indexed:

```
0006B4E5: BEFORE UPDATING RIDX SUMMARY
0006B4EB: SQLID dy3s2ghkxdasu Dumping RIDX for Read before update
RIDX(0x7fad85ecfe1c) : st 1(RIDX_INVALID) validBitMap 0 tabn 0 id {0 0 0}
RIDX: strt 0 end 0 offset 0 size 0 rgnIdx 0 RgnOffset 0 scn: 0x0000.00000000 hist: 0
RIDX validation history:
0:Undef 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
OCL_RIDX: 0x7fad85ecfe14 magic:caf1 rgnidx:2057 rgnhdlid:1state: 1 invalidatereason:7fad85ecfe1c
RIDX:2014-02-09 22:06:03.932207 :0006B4F0: SQLID dy3s2ghkxdasu ridxp 0x7fad85ecfe1c [st: 1] (32, 2048)
[io strt 32 end 2048] isRead 1 newColCached 1 newColFound 0 someColNotFound 0 isRidxValid 0 overWriteSummary 1
colInRidxNotInPredCol 0 objd 87427 RIDX:diskOffset 2156937216 RIDXCtx:diskOffset 2156937216 RIDXCtx:ioSize 1032192 regionSize 1032192
0006B4F3: AFTER UPDATING RIDX SUMMARY
RIDX(0x7fad85ecfe1c) : st 2(RIDX_VALID) validBitMap 0 tabn 0 id {87427 7 2087871281}
RIDX: strt 32 end 2048 offset 2156937216 size 1032192 rgnIdx 2057 RgnOffset 16384 scn: 0x0000.0012262f hist: 1
RIDX validation history:
0:PartialRead 1:Undef 2:Undef 3:Undef 4:Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [2] numFilt 4 flg 2 (HASNONNULLVALUES):
lo: 41 43 43 45 53 53 24 0
hi: 78 64 62 2d 6c 6f 67 31
```

- **Storage Indexes to Database Objects**
  - **cellsrv event:**
    - cellsrv_storidx('dumpridx','all',0,0,0)
      - all/griddiskname
      - objd - data_object_id from all_objects
      - tsn - tablespace number – ts# from ts$
      - dbid - ksqdngunid from x$ksqdn
      - Example: alter cell events="immediate cellsrv.cellsrv_storidx('dumpridx','disk',objd, tsn,dbid);
      - Trace shall be performed on all cell nodes
    - Tracing with disk filter is recommended!

- **Storage Indexes to Database Objects**
- Trace file example:

```
RIDX (0x7f360f4aec6c) : st 2 validBitMap 0 tabn 0 id {75577 4 3495522197}
RIDX: strt 32 end 2048 offset 3730849792 size 1032192 rgnIdx 3558 RgnOffset 16384
 scn: 0x0000.000fbc02 hist: 0x49
RIDX validation history: 0:PartialRead 1:PartialRead 2:PartialRead 3:Undef 4:
Undef 5:Undef 6:Undef 7:Undef 8:Undef 9:Undef
Col id [1] numFilt 2 flg 2:
lo: 61 61 61 66 0 0 0 0
hi: 7a 7a 7a 6d 0 0 0 0
Col id [2] numFilt 3 flg 2:
lo: 61 61 61 65 0 0 0 0
hi: 7a 7a 7a 61 0 0 0 0
Col id [5] numFilt 4 flg 2:
lo: 61 61 61 69 0 0 0 0
hi: 7a 7a 7a 6c 0 0 0 0
```

- **Controlling Storage Indexes**
  - **cellsrv event:**
    - cellsrv_storidx('purge','all ',0,0,0)
      - purge - purge storage indexes for specified disk/db/object
      - disable - disable storage indexes for specified disk/db/object
      - enable - disable storage indexes for specified disk/db/object
      - all/griddiskname, objd, tsn, dbid
    - We can purge instead of restarting cellsrv

- **For more information:**
  - **Cell Offloading:**
    - http://wp.me/p3QpUL-aM
    - http://wp.me/p3QpUL-fy
  - **Hybrid Columnar Compression:**
    - http://wp.me/p3QpUL-a5
    - http://wp.me/p3QpUL-eH
  - **Storage Indexes:**
    - http://wp.me/p3QpUL-8K
    - http://wp.me/p3QpUL-7B
    - http://wp.me/p3QpUL-2c

PROGEEKING.COM