enkitec

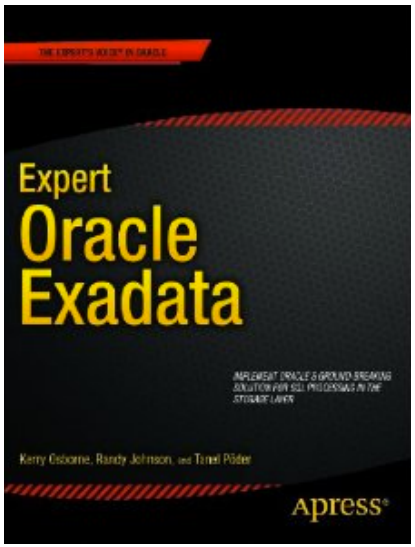
Understanding Exadata Offloading

Exadata Query Optimizations Enabled by Smart Scans

enkitec

whoami –

Work for Enkitec (www.enkitec.com)
Working with Exadata since early 2010
Many Exadata customers and POCs (40+)
Many Exadata Presentations (some to Oracle)
Coauthor Expert Oracle Exadata Book



OakTable.net

(kerryosborne.oracle-guy.com)

enkitec

whoami – (my prejudices)

I am an Exadata Fan Boy
– so please take everything I say with a grain of salt

Agenda

Exadata Basics
Offloading / Smart Scans
 - Optimizations
 - Requirements
 - How to tell if it's working ***this is very important***
Demo
Questions

enkitec

Exadata Basics

Exadata Database Servers

11gR2 / ASM

Infiniband

iDB / RDS

Exadata Storage Servers

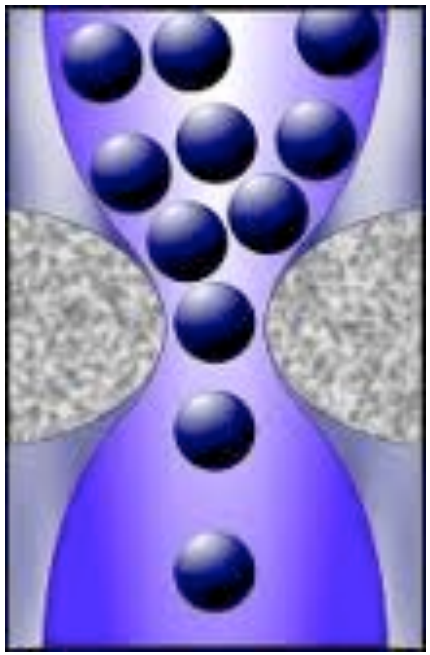
cellsrv

The Big Ah Ha!

The Bottleneck on Many (Most) Large Databases is between the Disk and the DB Server(s)!

How to Speed Up?

Make the Pipe Bigger/Faster
Reduce the Volume



* The fast way to do anything is not to do it ~ Cary Millsap

enkitec

Offloading – The "Secret Sauce"

Offloading vs. Smart Scan
(what's the difference)

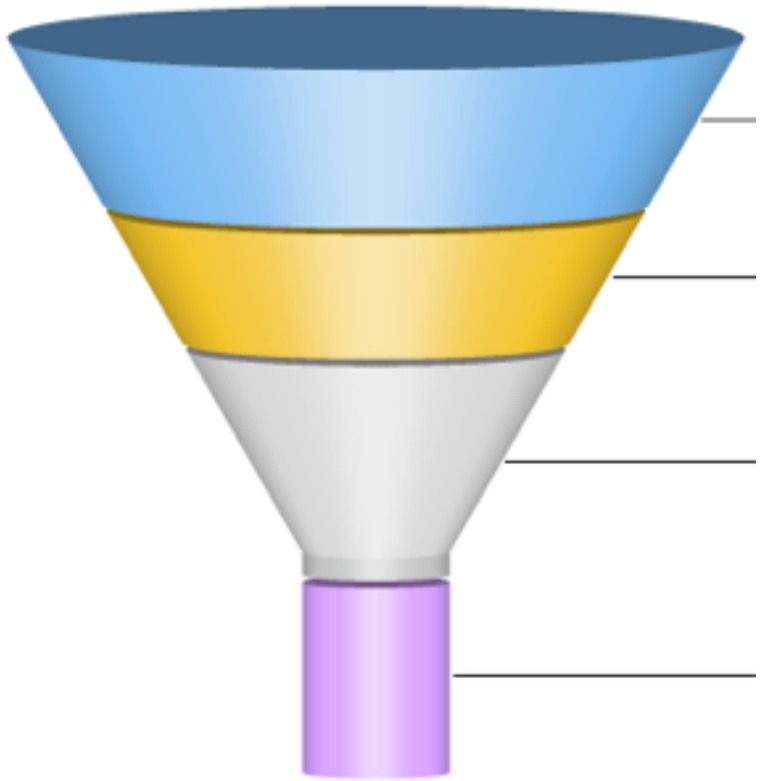
Offloading – generic term meaning doing work at the storage layer instead of at the database layer

Smart Scan – query optimizations covered by "cell smart table/index scan" wait events

enkitec

Smart Scan Optimizations

- Column Projection
- Predicate Filtering
- Storage Indexes
- Simple Joins
- Function Offloading
- Virtual Column Evaluation
- HCC Decompression
- Decryption



enkitec

You can Tune an Exadata (but not a fish)

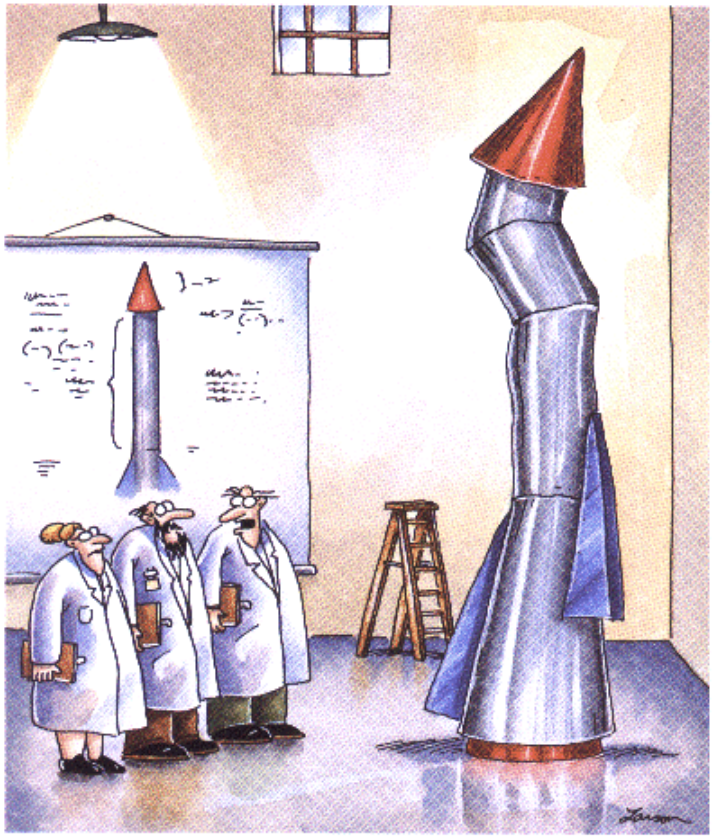
Check to see if you're getting Smart Scans!

If you're not, figure out why and correct the situation!

It's Pretty Simple.

3 things you'll need to know:

- the Optimizations
- the Requirements
- how to Measure



"It's time we face reality, my friends. … We're not exactly rocket scientists."

enkitec

Smart Scan Requirements

Full Scan
Direct Path Read
Object Stored On Exadata Storage

Why?

Very Simple Explanation:

Various full scan functions()
- kcbldrget() – direct path read function
  - kcfis_read() – kernel file intelligent storage read (Smart Scan)

*why it's there: checkpointing and non-block data return

Requirement 1: Full Scans

- Table
- Partition
- Materialized View
- Index (FAST FULL SCAN Only)

```
SYS@shareprd1> @op_event_awr.sql
Enter value for event: cell smart%

EVENT                          OPERATION                                 COUNT(*)
------------------------------ ---------------------------------------- ----------
cell smart index scan          INDEX STORAGE FAST FULL SCAN                     124
                               INDEX STORAGE SAMPLE FAST FULL SCAN              234

cell smart table scan          MAT_VIEW ACCESS STORAGE FULL                       1
                               TABLE ACCESS STORAGE FULL                      27747
```

*** Query from DBA_HIST_ACTIVE_SESS_HISTORY**

Digression - New Exadata Wait Events

cell list of blocks physical read – (db file parallel read)
cell multiblock physical read – (db file scattered read)
cell single block physical read – (db file sequential read)
cell smart file creation
cell smart incremental backup
cell smart index scan
cell smart restore from backup
cell smart table scan

* Note that there are others, these are the most interesting

Requirement 2: Direct Path Reads

Bypass buffer cache – direct to PGA
Decision not part of optimizer's job
Traditionally Used by Parallel Slaves
Non-Parallel Also Possible
- Serial Direct Path Reads (adaptive)
- algorithm not documented (but more aggressive in 11g) *
  - size of segment (table or index or partition)
  - size of buffer cache
  - number blocks already in buffer cache
  - _small_table_threshold
  - _very_large_table_threshold

* See MOS Note: 50415.1 - WAITEVENT: "direct path read" Reference Note

Requirement 3: Exadata Storage

Kind of Goes Without Saying

- Possible to have non-Exadata storage or mixed
- ASM Diskgroup has an attribute: **cell.smart_scan_capable**
- Must be set to TRUE for Smart Scans to work
- Can't add non-Exadata storage without changing to FALSE

How NOT to Tell if You got a Smart Scan

-- Explain Plan Output

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
SQL_ID  35tqjjq5vzg4b, child number 0
-------------------------------------
select count(*) from kso.temp_skew where col1_plus_pk=27998244

Plan hash value: 725706675

-----------------------------------------------------------------------------------------
| Id  | Operation                  | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |           |       |       | 44692 (100)|          |
|   1 |  SORT AGGREGATE            |           |     1 |     6 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| TEMP_SKEW |     2 |    12 | 44692   (1)| 00:08:57 |
-----------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage("COL1"+"PK_COL"=27998244)
       filter("COL1"+"PK_COL"=27998244)
```

* STORAGE keyword - means Smart Scans are possible, not guaranteed

How to Tell if You got a Smart Scan

Millsap It!
- (10046 trace)
- most fool proof?

TP It!
- Tanel's snapper
- v$sesstat, v$session_event
- great if it's happening now

~~Wolfgang It!~~
- unfortunately this doesn't work
- 10053 trace (and the optimizer) has no idea

Rahn It!
- DBMS_SQLTUNE.REPORT_SQL_MONITOR
- probably best

KO It!
- My fsx.sql script
- V$SQL family of views: IO_CELL_OFFLOAD_ELIGIBLE_BYTES
- saved in AWR so works on historical data as well

How to Tell if You got a Smart Scan

```
-- fsx.sql

select sql_id,
  decode(IO_CELL_OFFLOAD_ELIGIBLE_BYTES,0,'No','Yes') Offloaded,
  decode(IO_CELL_OFFLOAD_ELIGIBLE_BYTES,0,0,
  100*(IO_CELL_OFFLOAD_ELIGIBLE_BYTES-
         IO_INTERCONNECT_BYTES)/
         IO_CELL_OFFLOAD_ELIGIBLE_BYTES) "IO_SAVED_%"
  from v$sql
  where sql_text like '&sql_text';
```

* Warning: there are occasions where it's weird (negative IO_SAVED_%)

How to Tell if You got a Smart Scan

```
-- report_sql_monitor.sql

select DBMS_SQLTUNE.REPORT_SQL_MONITOR(
   session_id=>nvl('&&sid',sys_context('userenv','sid')),
   session_serial=>decode('&&sid',null,null,sys_context('userenv','sid'),
        (select serial# from v$session where audsid = sys_context
        ('userenv','sessionid')),null),
   sql_id=>'&sql_id',
   sql_exec_id=>'&sql_exec_id',
   report_level=>'ALL')
as report
from dual;
```

enkitec

The Wrong Tool for the Job?

Maybe:

Any of the tools can do the job. Just depends on the circumstance and you're preferences.

enkitec

Demo Time

Last Thoughts

Take Some Time to Test

- Just Because You Can Slam it in Doesn't Mean You Should

Take Some Time to Understand the Exadata Optimizations

- Know What to Expect

Take Some Time to Evaluate Indexes

- Migration is a Golden Opportunity to Get Rid of Some
- Make Sure the Ones You Keep Aren't Overused

Questions / Contact Information

Questions?

Contact Information : Kerry Osborne

kerry.osborne@enkitec.com
kerryosborne.oracle-guy.com
www.enkitec.com