Real World Exadata

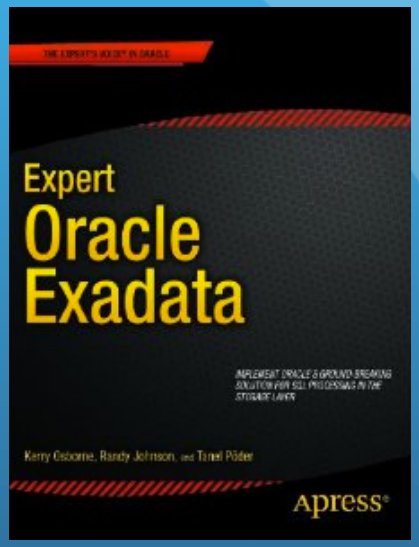
Presented by: Kerry Osborne

March 7, 2012

enkitec

whoami -

Worked with Oracle Since 1982 (V2)
Working with Exadata since early 2010
Work for Enkitec (www.enkitec.com)
(Enkitec owns a Half Rack – V2/X2)
Many Exadata customers and POCs
Many Exadata Presentations (many to Oracle)
Exadata Book



Blog: kerryosborne.oracle-guy.com

OakTable.net

ORACLE ACE Director

enkitec

What's the Point?

- Part 1: Real World Statistics
- Part 2: How To Be Sure Your Getting What You Paid For

enkitec

Poll – How Many In The Audience Are Already Using Exadata?

enkitec

Part 1: Statistics*

- 51 Database Machines
- 22 Companies
- Revenues ~$50M to ~$25B
- non-RAC to 16 node RAC cluster
- Basic (mini) – X2-8

* Not a statistically significant or random sample

enkitec

Sizes

Number of Racks:*

```
Full    - 11 or 22%
Half    - 12 or 24%
Quarter - 27 or 53%
Basic   -  1 or  2%
```

Average Size: 0.468

Number of Companies:

```
Full    -  3 - 11%
Half    -  9 - 33%
Quarter - 15 - 55%
```

Company Sizes:

```
Large (>$1B)    - 10 - 45%
Medium          -  5 - 23%
Small (<$200M)  -  7 - 32%
```

* A few have additional storage cells

enkitec

Models

Generations:

```
V1    -  3 or  6%
V2    - 10 or 20%
X2-2  - 28 or 73%
X2-8  -  1 or  2%
```

Drives Types:*

```
Hi Perf -  6 or 12%
Hi Cap  - 45 or 88%
```

* Flash Cache tends to cover up drive deficiencies

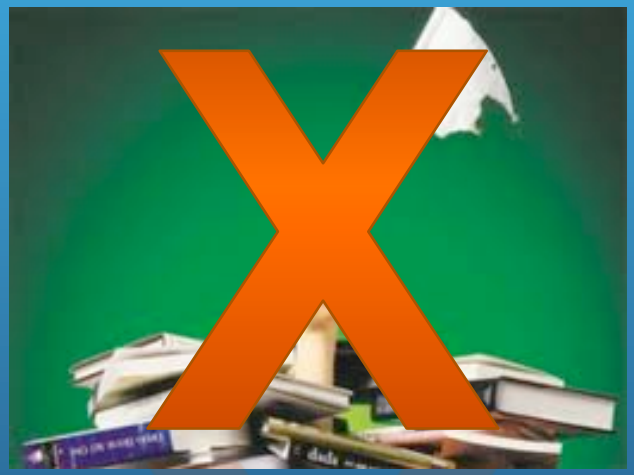
enkitec

Workload (DW, OLTP, Mixed)

This one is hard because there are few “pure” workloads!

Primary Usage:

```
DW       – 29 or 57%
OLTP     – 20 or 40%
Mixed    –  2 or  4%
```



Primary Application:

```
Custom      - 30 – 59%
PeopleSoft – 10 - 20%
eBiz        -  5 – 10%
Other       -  6 – 12%
```

enkitec

Consolidations

Most Companies in this Sample are Consolidating on Exadata

```
Yes     - 34 or 67%
No      - 16 or 31%
Unknown -  1 or  2%
```

Types:

- DW and OLTP
- Combining Many Disparate Systems
- “Cloud” Initiatives
- Unconsolidated Consolidations
  - Consider 1 Full Rack -> 4 X 2-Node RAC Clusters

enkitec

Multiple Racks

Most Companies in this Sample Bought More than One Rack

```
Single   -   3 or 14%
Multiple -  19 or 86%
```

Note: 4 of the 19 companies w/ multiple Exadatas started with a single DBM

Why?

- Patching
- Dev / Test
- DR

enkitec

Part 1a: Stories

- common migration strategies
- recommended parameter/configuration settings
- suitability for various workloads (OLTP vs. DW vs. mixed)
- indexing strategies
- compression strategies
- organizational challenges presented by Exadata

Common Migration Strategies

Logical

- Data Pump
- exp / imp
- Golden Gate
- CTAS Across DBLink

Physical

- RMAN
- TTS
- Dataguard Physical Standby
- ASM Rebalance

enkitec

Digression: Fork Lift Migrations

- Just Say No!
- 9i RBO to 11gR2 on Exadata
  - pour some more salt on the wounds
- Good News is Exadata Can Cover Up Many Sins
- Bad News is it Can't Cover Up Everything
- Typical Results (2-3X Faster Than Before)

enkitec

Digression: Fork Lift Migrations

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		21,685		49.1	
SQL*Net more data from dblink	52	7,296	1.E+05	16.5	Network
Streams miscellaneous event	12,525	6,274	501	14.2	Other
enq: TM - contention	671	5,478	8164	12.4	Applicatio
cell single block physical rea	3,935,035	2,231	1	5.1	User I/O

Digression: Fork Lift Migrations

```
ENKITEC@IOP> @offload_percent
Enter value for sql_text:
Enter value for min_etime:
Enter value for min_avg_lio:

     TOTAL  OFFLOADED OFFLOADED_%
---------- ---------- -----------
        41          3       7.32%

1 row selected.

Elapsed: 00:00:01.79
ENKITEC@IOP> @fsxo

SQL_ID         CHILD  PLAN_HASH  EXECS  AVG_ETIME AVG_PX OFFLOAD IO_SAVED_%        AVG_LIO
------------- ------ ---------- ------ ---------- ------ ------- ---------- --------------
0q3s48vg6ddhg      5 1556289656      2      88.74      0 No             .00     13,404,990
0sxmqjfv2xzf3      0 3139973520      1       6.54      0 No             .00        892,423
1n4rfxtg6zg4x      0 3889350093      1       8.85      0 No             .00        656,485
1r5u1bq8xq2xj      0 1092323636      1      19.73      0 No             .00      4,297,589
1vgh9gswphvdy      1 1599020674  2,787        .08      0 Yes          100.00        659,786
1vgh9gswphvdy      3 1599020674  5,944        .09      0 Yes          100.00        661,692
1vgh9gswphvdy      4 1599020674  3,877        .07      0 Yes          100.00        661,985
22nu7gq6awt2j      1 4032143122      2      27.85      0 No             .00      4,102,245
238fx4v7gsv95      1 3046949154      2      84.78      0 No             .00     13,323,956
2b8c16v3cjsua      1 1543662060      2      92.80      0 No             .00     14,404,649
2pbxscctg6cnj      1 3046949154      2      84.28      0 No             .00     13,323,720
```

Common Configuration

- Auto DOP - Off
- SPM - Off
- Buffer Cache – Smaller than on non-Exadata
- Flash – All Flash Cache
- Huge Pages – enabled (no AMM)
- parallel_max_processes < default
- Backups – generally RMAN to recovery area then to tape

Suitability for Various Workloads

- OLTP
  - Good
- Mixed
  - Excellent
- DW
  - Killer

Indexing Strategies

Some Suggestions We've Heard
Drop All Indexes
Don't Change Anything

enkitec

Indexing Strategies

- Single Row Access (OLTP) Needs Indexes
- Most Workloads are Mixed
- Optimizer Doesn't Know About Smart Scans
- Challenge is to Use Indexes When Appropriate
- You Probably Need Fewer Indexes
- You May Have to Get Creative
  - optimizer_use_invisible_indexes
  - optimizer_index_cost_adj

enkitec

Compression Strategies

- HCC Provides Exceptional Compression Ratios
  - 10X is pretty good guess
  - 6X – 60X in Practice
- Oddly Enough Many are Not Using HCC
- HCC Not Appropriate for Active Data
- HCC Needs Partitioning
  - Requires Direct Path Loads
  - Update Move
  - Single Row Update Locks Entire CU
  - Falls Back to OLTP

enkitec

Organizational Challenges

- Who Should Manage The Beast
- General Thinking is DBA's (DMA's ?)
- It is 11g DB with ASM After All
- Patching Requires More Knowledge Than Most DBAs Have
  - Linux
  - Network
  - Hardware
  - Storage
- Best Approach for Most is Combination of Sysadmin / DBA
- SAN Guys are Out of the Picture
- CAB Story – "What happens if I run out of space?"

enkitec

Part 2: How to Know You're Getting What You Paid For

enkitec

How to Tune an Exadata
(radio edit)

Check to see if you're getting Smart Scans!

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

It's Pretty Simple.

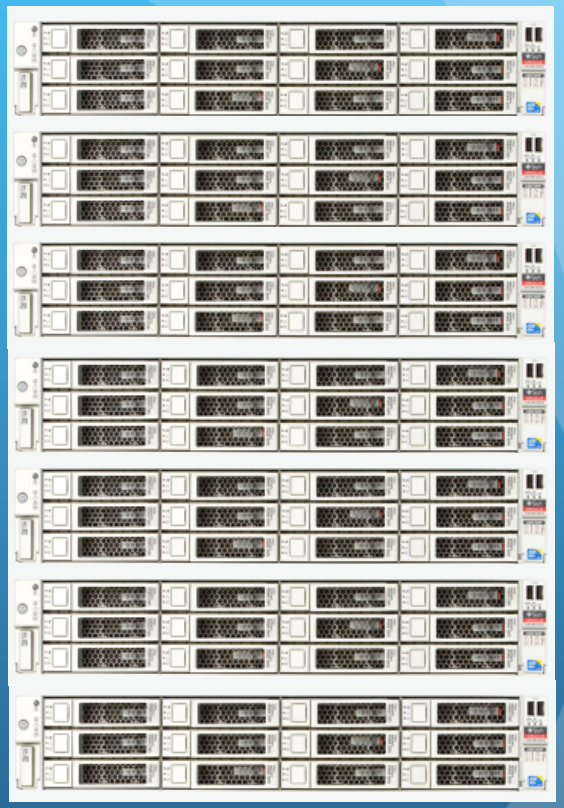
3 things you'll need to know:

- the Optimizations
- the Requirements
- how to Measure

enkitec

What is Exadata?

Exadata Storage Servers



Exadata Database Servers

Infiniband

11gR2 / ASM

iDB / RDS

cellsrv

*Half Rack

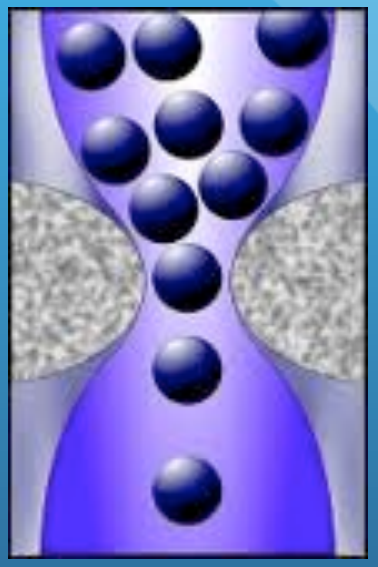
enkitec

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!

enkitec

Offloading – The "Secret Sauce"

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

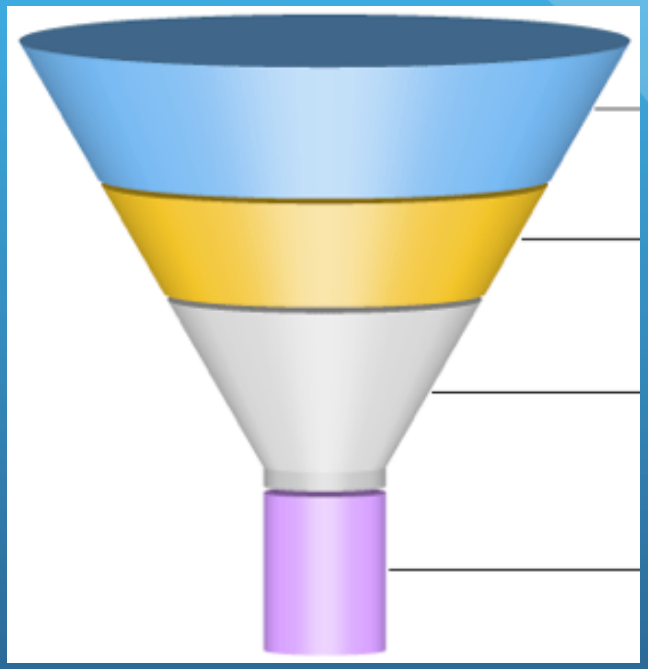
Offloading – generic term meaning doing work at the storage layer instead of at the database layer

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

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

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

enkitec

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

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 fully 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"

enkitec

Requirement 3: Exadata Storage

Kind of Goes Without Saying

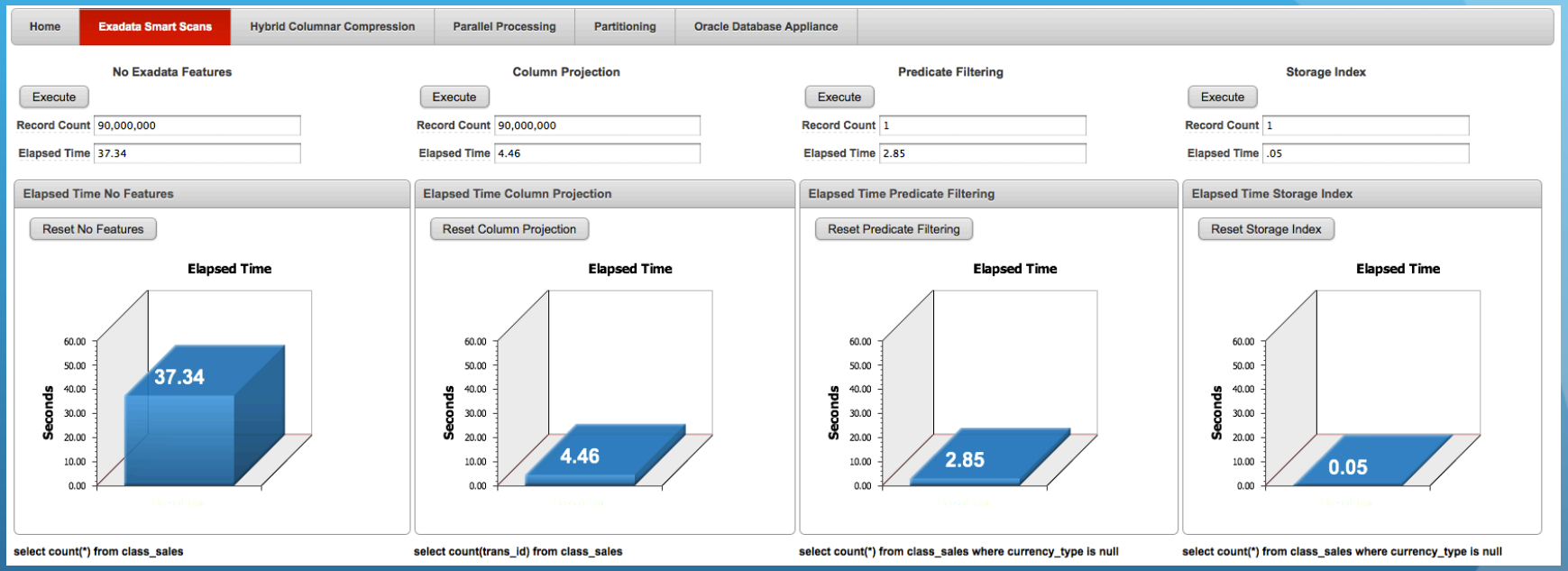
- Possible to have non-Exadata storage or mixed
- ASM Diskgroup has an attribute: cell.smart_scan_capable
- Must be set to TRUE for Smart Scans to work
- Can’t add non-Exadata storage without changing to FALSE

Demo Time

enkitec

```
------------------------------
select /*+ parallel 2 */ a.col2, sum(a.col1) from kso.skew a, kso.skew b where rownum < 30000000 group by a.col2

Global Information
------------------------------
 Status              :  DONE (ALL ROWS)
 Instance ID         :  1
 Session             :  SYS (1278:795)
 SQL ID              :  6f2dhcj7m3k2b
 SQL Execution ID    :  16777216
 Execution Started   :  03/07/2012 15:04:19
 First Refresh Time  :  03/07/2012 15:04:19
 Last Refresh Time   :  03/07/2012 15:04:33
 Duration            :  14s
 Module/Action       :  sqlplus@enkdb01.enkitec.com (TNS V1-V3)/-
 Service             :  SYS$USERS
 Program             :  sqlplus@enkdb01.enkitec.com (TNS V1-V3)
 Fetch Calls         :  2

Global Stats
=========================================================================================================
| Elapsed |   Cpu   |    IO    | Application |  Other   | Fetch | Buffer | Read | Read  | Write | Write |  Cell   |
| Time(s) | Time(s) | Waits(s) |  Waits(s)   | Waits(s) | Calls |  Gets  | Reqs | Bytes | Reqs  | Bytes | Offload |
=========================================================================================================
|      27 |      23 |     3.56 |        0.00 |     0.33 |     2 |  73767 | 3071 | 676MB |  1228 | 246MB |  -3.09% |
=========================================================================================================

Parallel Execution Details (DOP=2 , Servers Allocated=4)
=================================================================================================================================================================
|      Name      | Type  | Server# | Elapsed |   Cpu   |    IO    | Application |  Other   | Buffer | Read | Read  | Write | Write |  Cell   |          Wait Events          |
|                |       |         | Time(s) | Time(s) | Waits(s) |  Waits(s)   | Waits(s) |  Gets  | Reqs | Bytes | Reqs  | Bytes | Offload |          (sample #)           |
=================================================================================================================================================================
| PX Coordinator | QC    |         |    7.40 |    7.23 |          |        0.00 |     0.17 |   1058 |      |     . |       |     . |    NaN% |                               |
| p000           | Set 1 |       1 |    8.11 |    6.72 |     1.22 |             |     0.16 |    765 | 1123 |  61MB |   614 | 123MB | -66.67% |                               |
| p001           | Set 1 |       2 |    8.55 |    7.13 |     1.42 |             |          |    655 | 1103 |  59MB |   614 | 123MB | -66.67% | direct path read temp (1)     |
| p002           | Set 2 |       1 |    1.55 |    1.12 |     0.43 |             |          |  35646 |  428 | 278MB |       |     . |  38.65% |                               |
| p003           | Set 2 |       2 |    1.61 |    1.12 |     0.50 |             |          |  35643 |  417 | 278MB |       |     . |  38.27% |                               |
=================================================================================================================================================================

SQL Plan Monitoring Details (Plan Hash Value=1100917592)
=================================================================================================================================================================================================================
| Id |               Operation               |     Name     |  Rows   | Cost |   Time    | Start  | Execs |   Rows   | Read | Read  | Write | Write |  Cell   | Mem  | Temp | Activity |      Activity Detail      |
|    |                                       |              | (Estim) |      | Active(s) | Active |       | (Actual) | Reqs | Bytes | Reqs  | Bytes | Offload | (Max) | (Max) |   (%)    |       (# samples)         |
=================================================================================================================================================================================================================
|  0 | SELECT STATEMENT                      |              |         |      |         6 |     +9 |     1 |        1 |      |       |       |       |         |       |       |          |                           |
|  1 |   HASH GROUP BY                       |              |       2 | 331G |         7 |     +8 |     1 |        1 |      |       |       |       |     13M |       |       |    14.81 | Cpu (4)                   |
|  2 |    COUNT STOPKEY                      |              |         |      |         6 |     +9 |     1 |      30M |      |       |       |       |         |       |       |          |                           |
|  3 |     PX COORDINATOR                    |              |         |      |         6 |     +9 |     5 |      30M |      |       |       |       |         |       |       |    11.11 | Cpu (3)                   |
|  4 |      PX SEND QC (RANDOM)              | :TQ10001     |      1P | 175G |         7 |     +8 |     2 |      30M |      |       |       |       |         |       |       |     7.41 | Cpu (2)                   |
|  5 |       COUNT STOPKEY                   |              |         |      |         7 |     +8 |     2 |      30M |      |       |       |       |         |       |       |          |                           |
|  6 |        MERGE JOIN CARTESIAN           |              |      1P | 175G |        13 |     +2 |     2 |      30M |      |       |       |       |         |       |       |          |                           |
|  7 |         PX BLOCK ITERATOR             |              |     32M | 24649 |         1 |     +2 |     2 |       2 |      |       |       |       |         |       |       |          |                           |
|  8 |          TABLE ACCESS STORAGE FULL    | SKEW         |     32M | 24649 |        13 |     +2 |     2 |       2 |    3 |   3MB |       |       |  47.09% |       |       |          |                           |
|  9 |         BUFFER SORT                   |              |     32M | 331G |        13 |     +2 |     2 |      30M | 2223 | 117MB |  1228 | 246MB |     205M |  258M |       |    51.85 | Cpu (13)                  |
|    |                                       |              |         |      |           |        |       |          |      |       |       |       |         |       |       |          | direct path read temp (1) |
| 10 |          PX RECEIVE                   |              |     32M | 10683 |         7 |     +2 |     2 |      64M |      |       |       |       |         |       |       |     7.41 | Cpu (2)                   |
| 11 |           PX SEND BROADCAST           | :TQ10000     |     32M | 10683 |         2 |     +1 |     2 |       4 |      |       |       |       |         |       |       |     3.70 | Cpu (1)                   |
| 12 |            PX BLOCK ITERATOR          |              |     32M | 10683 |         1 |     +2 |     2 |      32M |      |       |       |       |         |       |       |          |                           |
| 13 |             INDEX STORAGE FAST FULL SCAN | SYS_C003992 |     32M | 10683 |         2 |     +1 |    26 |      32M |  845 | 557MB |       |       |  38.65% |       |       |     3.70 | Cpu (1)                   |
=================================================================================================================================================================================================================
```

Exadata Software Performance



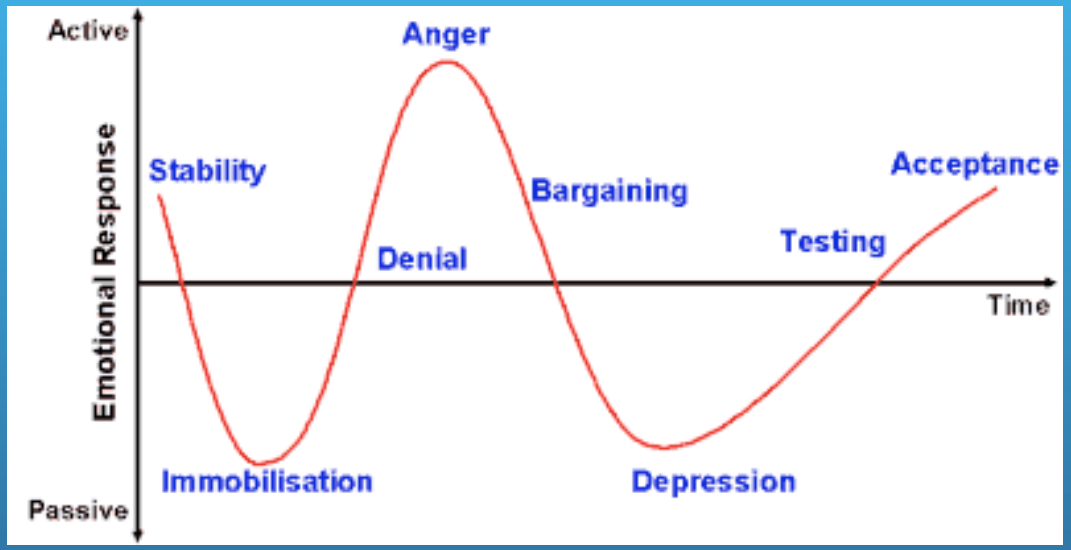
enkitec

High Transaction Volume: Telco Provider

- Customer Runs Dell, 16 Core Machines in Multiple RAC Instances
- Very High Volume of OLTP and Data Warehouse Type Queries on Same Database
- Performance Differences Were Too Excessive to Graph

SQL	Current	Exadata	Times Faster
Process 1: 6-Month Data Volume	52 min	19.5 sec	160 x
Process 2: 3-Month Data Volume	51 min	11.5 sec	269 x
Process 3: 1-Year Data Volume	50 min	37.5 sec	81 x
Process 4: 2-Month Data Volume	48 min	9.4 sec	308 x
Update SCN_CALL_PARTY_LOG	13 min	1.05 sec	744 x
Update SCN_CALL_PARTY_IDENT_LOG	7 min	.23 sec	1871 x
Select SCN_CALL_PARTY_EXTDATA_LOG	6.75 min	.47 sec	868 x

The Kübler-Ross grief cycle



Exposure to Exadata

Questions?

Contact Information : Kerry Osborne
kerry.osborne@enkitec.com
kerryosborne.oracle-guy.com
www.enkitec.com

ORACLE Platinum Partner
Specialized
Exadata

A Member Of The Inc. 5000

DALLAS 100
Fastest Growing Companies in Dallas