Tuning & Applying Best Practices On Very Large Databases (VLDB)

Tips, tricks & success stories from real life

Disclaimer

This views/content in this slides are those of the author and do not necessarily reflect that of Oracle Corporation and/or its affiliates/subsidiaries. The material in this document is for informational purposes only and is published with no guarantee or warranty, express or implied..

This material should not be reproduced or used without the authors' written permission.

Agenda

- Real-life scenario 1 – How a daily batch job performance consistency was achieved
- Real-life scenario 2 – ETL data loading and BI queries dramatic improvements
- Best Practices for configuration, sizing, performance, storage consideration
    - ➢ Golden Rules for VLDB configuration
    - ➢ Backup & Recovery best practices
    - ➢ Optimizer Statistics management
    - ➢ ASM best practices
    - ➢ Performance optimization
    - ➢ Exadata solutions
    - ➢ Active Data Guard solutions

Not a self marketing…

A tiny inspiration

Who am I

Over 20 years of IT experience
16+ years as an Oracle DBA
Oracle **ACE Director**
Oracle 10g Certified Master(**OCM**)
Oracle 10g RAC Certified Expert
OCP v8i,9i,10g & 11g
ITIL v3 Foundation Certified
Oracle Database 12c beta tester

SNC ID: @sjaffarhussain
http://jaffardba.blogspot.com

ORACLE
Certified Professional

ORACLE
10*g* Certified Master

ORACLE
Certified Expert
10*g* RAC Administrator

A ORACLE
ACE Director

ITIL

Who am I

TECHNOLOGIST OF THE YEAR

Syed Jaffer Hussain DBA

The learning never stops for Oracle RAC expert

[illegible]

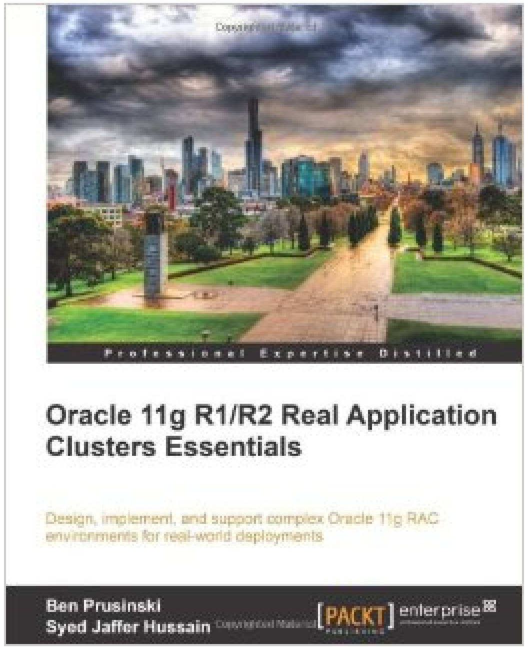
ORACLE MAGAZINE

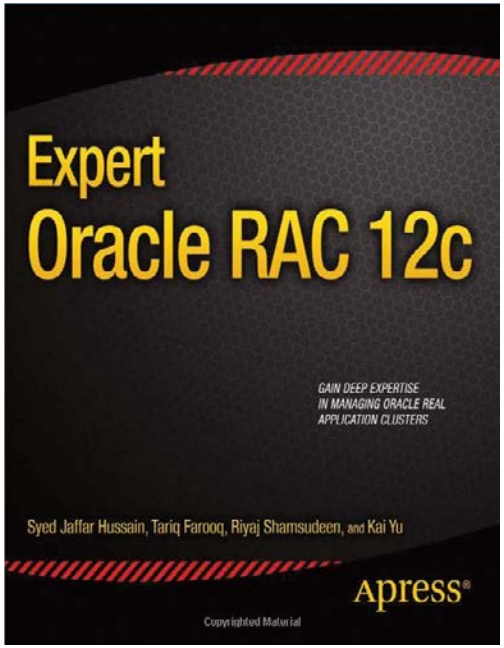
Technologist of the Year, DBA 2011

http://www.oracle.com/technetwork/issue-archive/2012/12-jan/o12awards-tech-1403083.html

Who am I

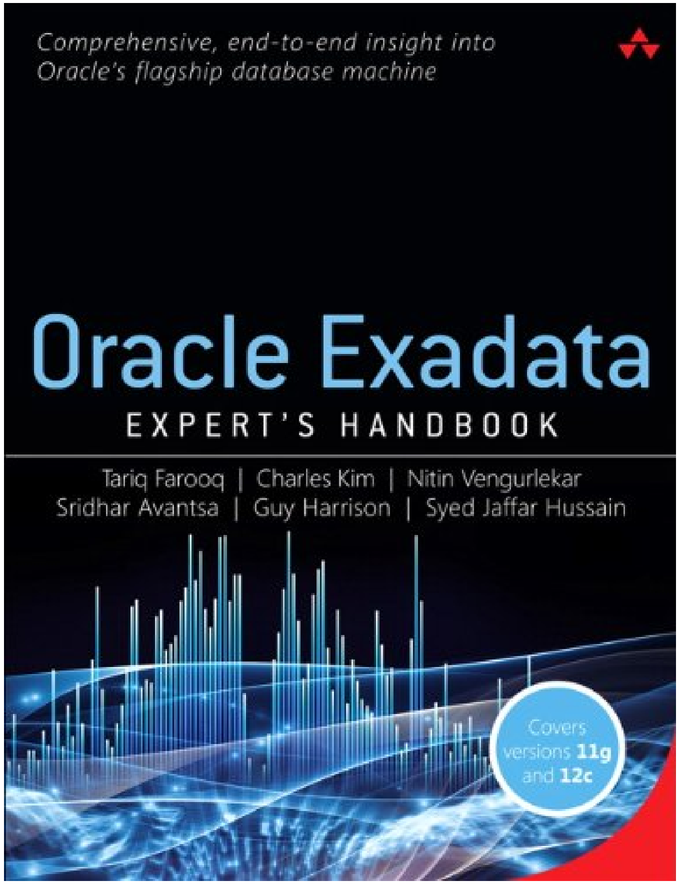
Co-authored …


Professional Expertise Distilled
Oracle 11g R1/R2 Real Application Clusters Essentials
Design, implement, and support complex Oracle 11g RAC environments for real-world deployments
Ben Prusinski
Syed Jaffer Hussain
[PACKT] enterprise


Expert
Oracle RAC 12c
GAIN DEEP EXPERTISE IN MANAGING ORACLE REAL APPLICATION CLUSTERS
Syed Jaffar Hussain, Tariq Farooq, Riyaj Shamsudeen, and Kai Yu
Apress®

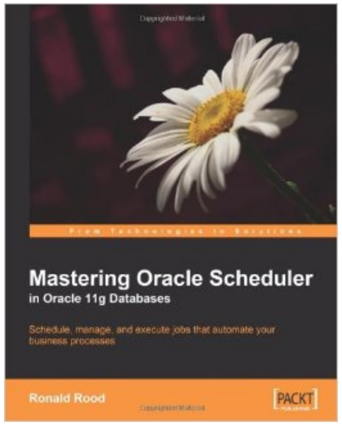
Who am I

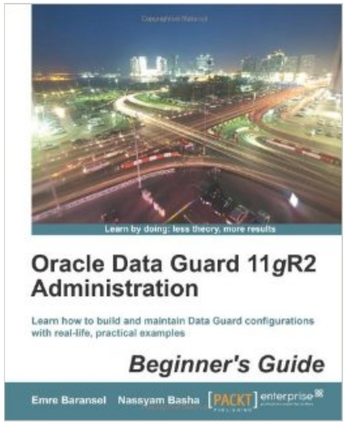
Up coming …

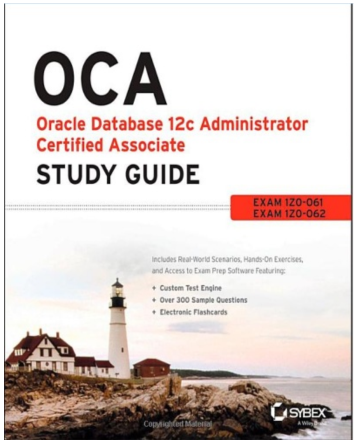

Comprehensive, end-to-end insight into
Oracle's flagship database machine
Oracle Exadata
EXPERT'S HANDBOOK
Tariq Farooq | Charles Kim | Nitin Vengurlekar
Sridhar Avantsa | Guy Harrison | Syed Jaffar Hussain
Covers versions 11g and 12c

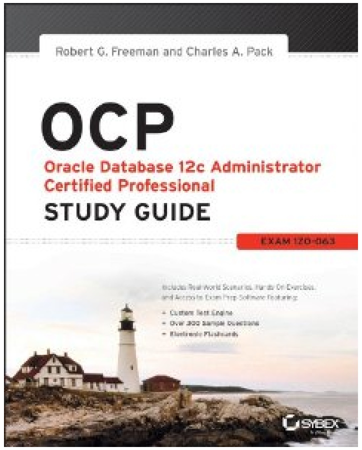
Who am I

Technical Reviewer …


Mastering Oracle Scheduler
in Oracle 11g Databases
Schedule, manage, and execute jobs that automate your business processes
Ronald Rood
PACKT


Learn by doing: less theory, more results
Oracle Data Guard 11gR2 Administration
Learn how to build and maintain Data Guard configurations with real-life, practical examples
Beginner's Guide
Emre Baransel
Nassyam Basha
PACKT enterprise


OCA
Oracle Database 12c Administrator Certified Associate
STUDY GUIDE
EXAM 1Z0-061
EXAM 1Z0-062
Includes Real-World Scenarios, Hands-On Exercises, and Access to Exam Prep Software Featuring:
Custom Test Engine
Over 300 Sample Questions
Electronic Flashcards
SYBEX


Robert G. Freeman and Charles A. Pack
OCP
Oracle Database 12c Administrator Certified Professional
STUDY GUIDE
EXAM 1Z0-063
SYBEX

VLDB - Definition

Data Warehouse –> VLDB –> Big Data

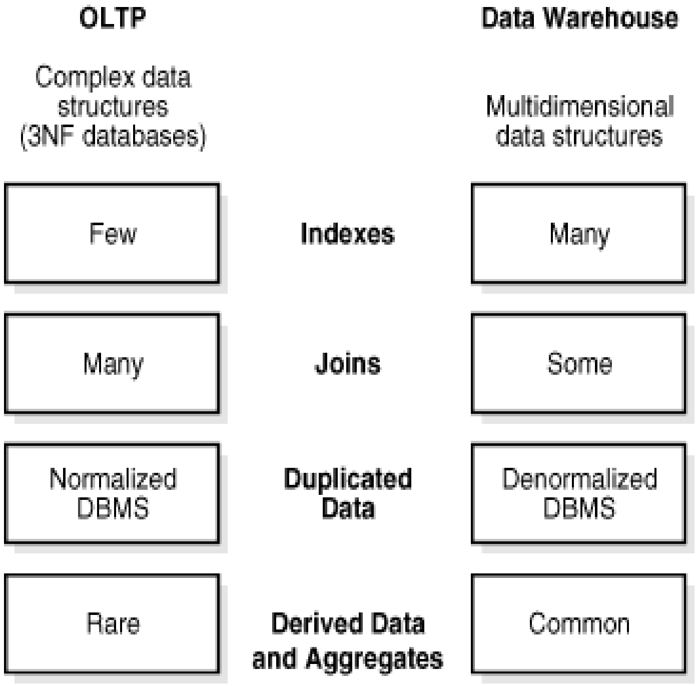
- Holds extremely high quantity of data
- Occupies large volume of storage
- Contains historical data for decision making
- Often updated through ETL tools
- A.k.a Big Data
- DW as a foundation for Big Data

VLDB – Key challenges

- Configuration
- Storage
- Performance
- Maintenance
- Administration
- Availability
- Server resources

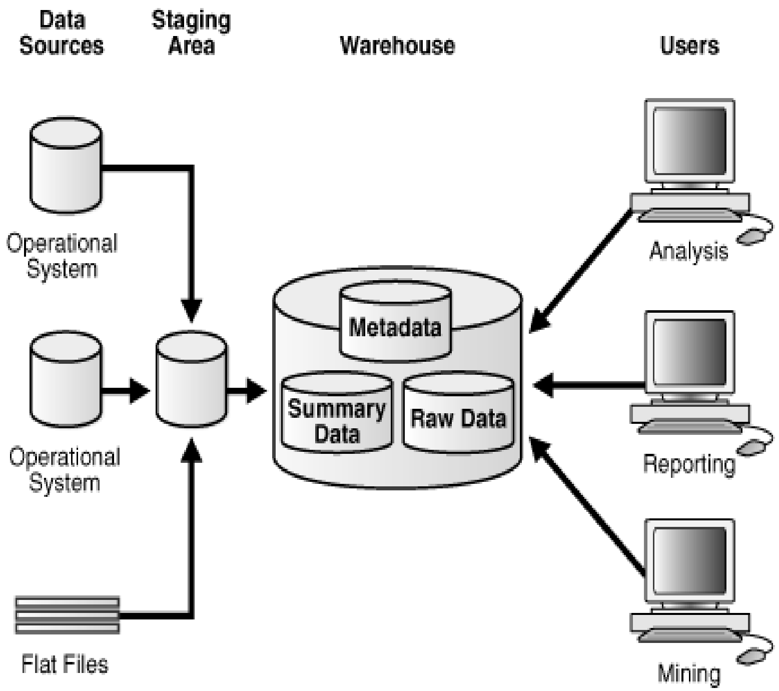
Data Warehousing Concepts - Overview

Contrasting OLTP and Data Warehousing Environments



OLTP		Data Warehouse
Complex data structures (3NF databases)		Multidimensional data structures
Few	**Indexes**	Many
Many	**Joins**	Some
Normalized DBMS	**Duplicated Data**	Denormalized DBMS
Rare	**Derived Data and Aggregates**	Common

Data Warehouse Architecture: with a Staging Area



Real Life Examples

Scenario 1

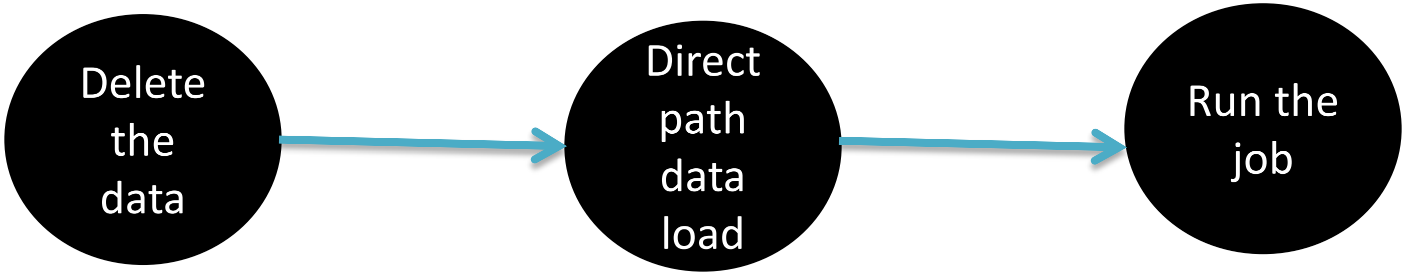
How a daily batch job performance consistency was achieved

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

Reconciliation batch job cycle



Oracle 10gR2
HPUX OS

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

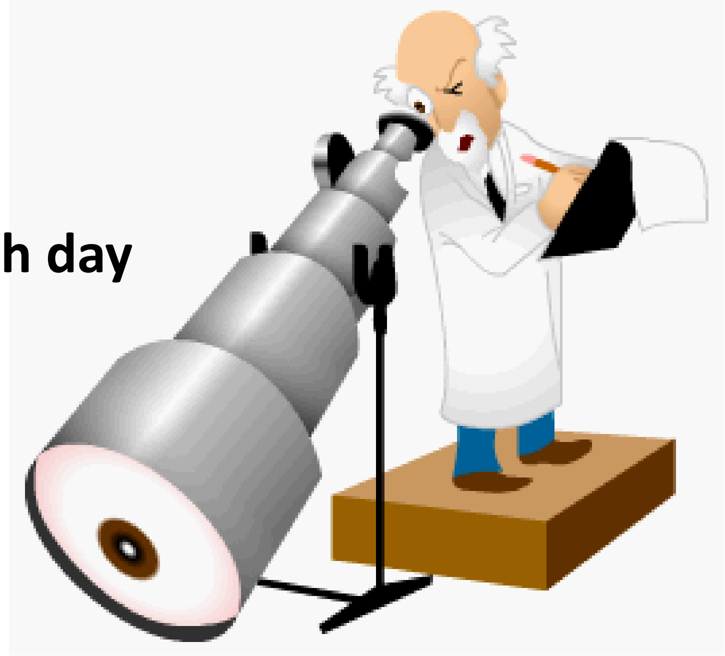
Major obstacles

- ➢ **Job completion inconsistency**
- ➢ **Taking 2 hours to 24 hrs to complete**
- ➢ **Impact on the job dependency**

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved



Observation

- ➢ **Consistent amount of data growth each day**
- ➢ **Queries favoring NESTED LOOP**

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

Temporary fix

- ➢ **Table and schema re-org**
- ➢ **Collecting stats**
- ➢ **Using Hints (HASH JOIN)**
- ➢ **Migration to other database**

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

Remedy

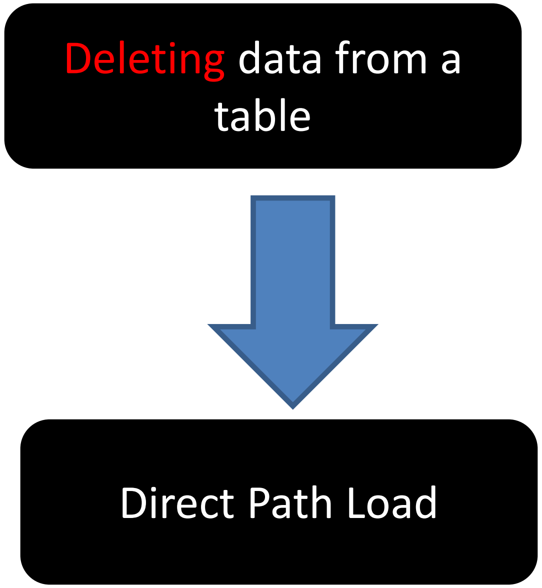
- ➢ **A <u>little ADJUSTMENT to the CODE</u> did the MIRACLE!**

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

Code review and Findings



Deleting data from a table

Direct Path Load

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

- **Delete was REPLACED with truncate**

Real Life Examples

Scenario 1

How a daily batch job performance consistency was achieved

- **Its been over 6 yrs or so, the JOB is not taking more than 45 min. to finish!**
- **No more fragmentation**
- **Queries opting HASH JOIN and finishing on TIME**

Real Life Examples

Scenario 2

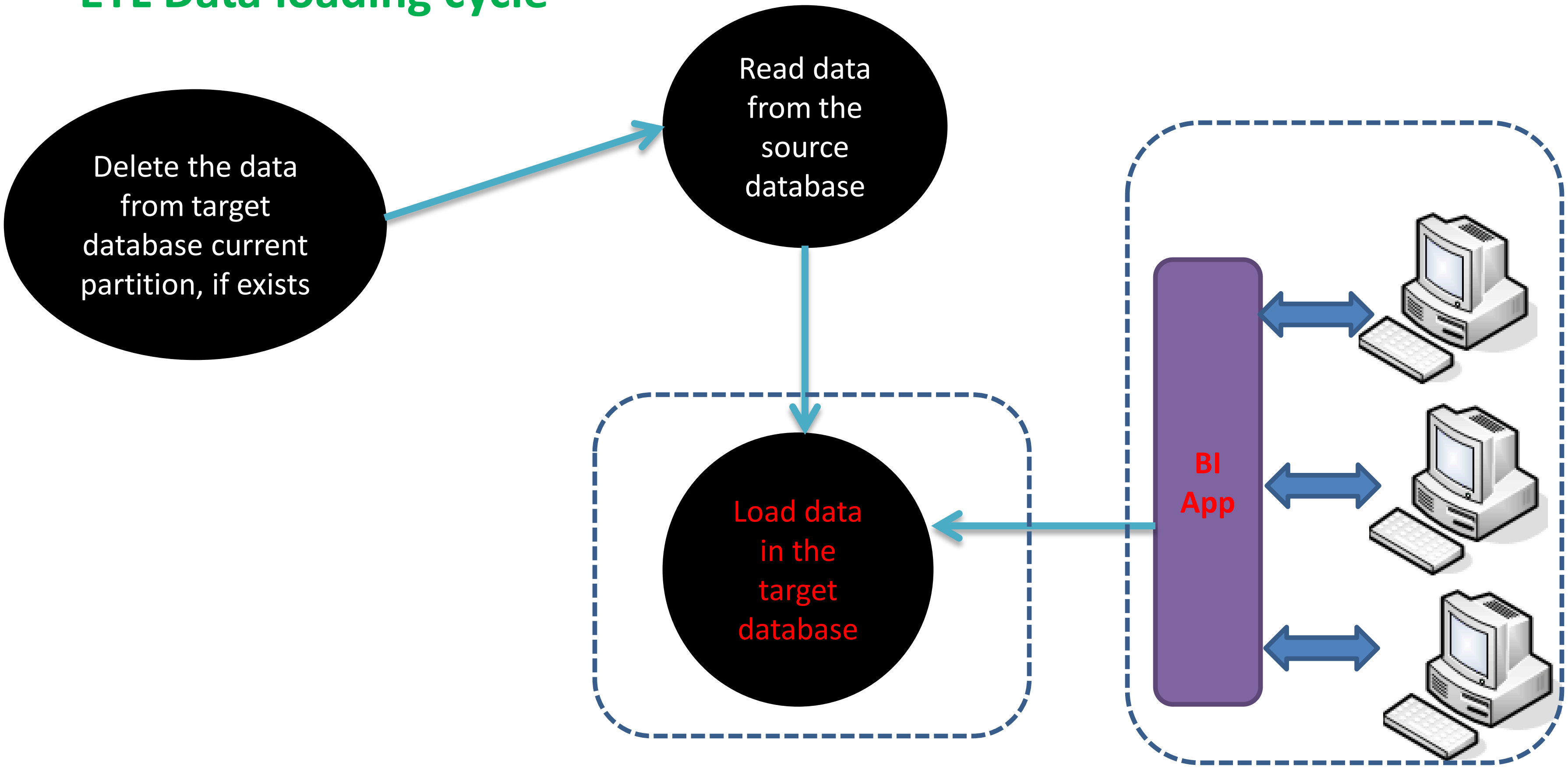
ELT data loading and BI queries dramatic improvements

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

ETL Data loading cycle


Delete the data from target database current partition, if exists
Read data from the source database
Load data in the target database
BI App

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

ETL Data loading cycle - statistics

- ✓ Over 1 millions are inserted a table using the plain inserts
- ✓ Monthly partitioned table
- ✓ Over 8 indexes
- ✓ 6 tables involved in the process

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

ETL Data loading cycle - Concerns

- ✓ Through put, <150 records per second
- ✓ Taking 4 hours to complete the load job
- ✓ Stats gather was taking 2-4 hours
- ✓ BI queries on the tables were taking over 2 hours
- ✓ Affecting subsequent job cycle, impacting other databases and busiess
- ✓ **Business users complains, escalations if the reports did not deliver by 7.30am**

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Action force committee

- ✓ Emergency meeting was called
- ✓ Discussed the seriousness of the issues
- ✓ Come-up with the list of action plans (around 27 actions)
- ✓ Targets/deadlines were set

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Database action plan

1. Monitor index usage to drop unwanted indexes
2. Verify chained rows
3. Splitting monthly partitions to weekly
4. Incremental gather stats
5. Increase redo log size
6. Enable Parallelism
7. Convert Global indexes to Local indexes
8. Adding resources, CPU & Memory to the database server
9. Increase SGA
10. Rewriting some of the queries(logic)
11. Enable Temporary Tablespace Group Management
12. Offloading queries to standby database
13. Restructured the partition key on the tables

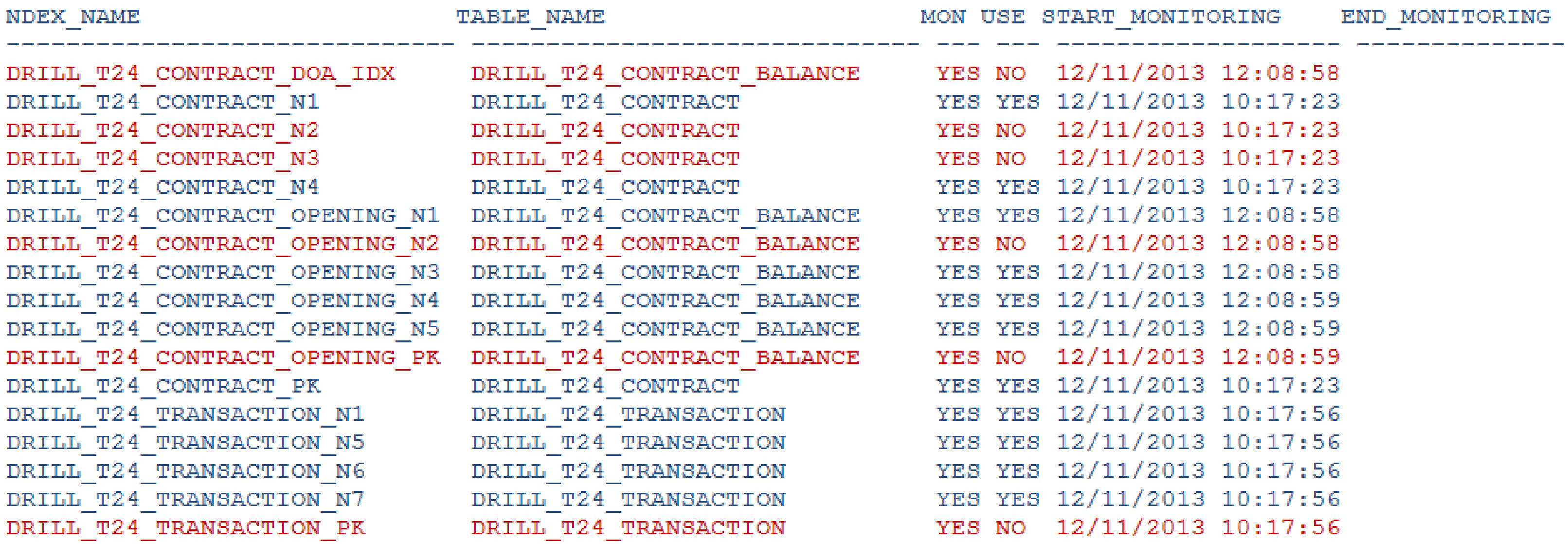
Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Get rid-off unwanted indexes

- ✓ Index usage monitoring / dropping unwanted indexes
  - ✓ Managed to reduce number of indexes, dropped and composite indexes
  - ✓ Made the indexes invisible before drop



```
NDEX_NAME                      TABLE_NAME                     MON USE START_MONITORING    END_MONITORING
------------------------------ ------------------------------ --- --- ------------------- ---------------
DRILL_T24_CONTRACT_DOA_IDX     DRILL_T24_CONTRACT_BALANCE     YES NO  12/11/2013 12:08:58
DRILL_T24_CONTRACT_N1          DRILL_T24_CONTRACT             YES YES 12/11/2013 10:17:23
DRILL_T24_CONTRACT_N2          DRILL_T24_CONTRACT             YES NO  12/11/2013 10:17:23
DRILL_T24_CONTRACT_N3          DRILL_T24_CONTRACT             YES NO  12/11/2013 10:17:23
DRILL_T24_CONTRACT_N4          DRILL_T24_CONTRACT             YES YES 12/11/2013 10:17:23
DRILL_T24_CONTRACT_OPENING_N1  DRILL_T24_CONTRACT_BALANCE     YES YES 12/11/2013 12:08:58
DRILL_T24_CONTRACT_OPENING_N2  DRILL_T24_CONTRACT_BALANCE     YES NO  12/11/2013 12:08:58
DRILL_T24_CONTRACT_OPENING_N3  DRILL_T24_CONTRACT_BALANCE     YES YES 12/11/2013 12:08:58
DRILL_T24_CONTRACT_OPENING_N4  DRILL_T24_CONTRACT_BALANCE     YES YES 12/11/2013 12:08:59
DRILL_T24_CONTRACT_OPENING_N5  DRILL_T24_CONTRACT_BALANCE     YES YES 12/11/2013 12:08:59
DRILL_T24_CONTRACT_OPENING_PK  DRILL_T24_CONTRACT_BALANCE     YES NO  12/11/2013 12:08:59
DRILL_T24_CONTRACT_PK          DRILL_T24_CONTRACT             YES YES 12/11/2013 10:17:23
DRILL_T24_TRANSACTION_N1       DRILL_T24_TRANSACTION          YES YES 12/11/2013 10:17:56
DRILL_T24_TRANSACTION_N5       DRILL_T24_TRANSACTION          YES YES 12/11/2013 10:17:56
DRILL_T24_TRANSACTION_N6       DRILL_T24_TRANSACTION          YES YES 12/11/2013 10:17:56
DRILL_T24_TRANSACTION_N7       DRILL_T24_TRANSACTION          YES YES 12/11/2013 10:17:56
DRILL_T24_TRANSACTION_PK       DRILL_T24_TRANSACTION          YES NO  12/11/2013 10:17:56
```

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Chained rows

- ✓ Verify chained rows
  - ✓ Over 60 thousand chained rows
  - ✓ The table had 1% of row chaining
  - ✓ Migrated to a higher block tablespace

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Database action plan

- ✓ Splitting monthly partitions to weekly with auto partitioning option
- ✓ Change in partition column
- ✓ Incremental gather stats
- ✓ Increase redo log size from 100MB to 500MB
- ✓ Convert Global indexes to Local indexes
- ✓ Adding resources, CPU & Memory to the database server
- ✓ Increase SGA

  - ✓ Incremental stats : from 2-4 hours to <30min
  - ✓ Inserts : increased from <150 RPS To >2000 RPS
  - ✓ The insert job completes in <6 min in contrast to 4 hrs earlier
  - ✓ BI reports improved from 18 min to 27 seconds

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Database action plan

- ✓ Rewriting some of the queries(logic)
  - ✓ 5hrs job was brought down to 3 min

Real Life Examples

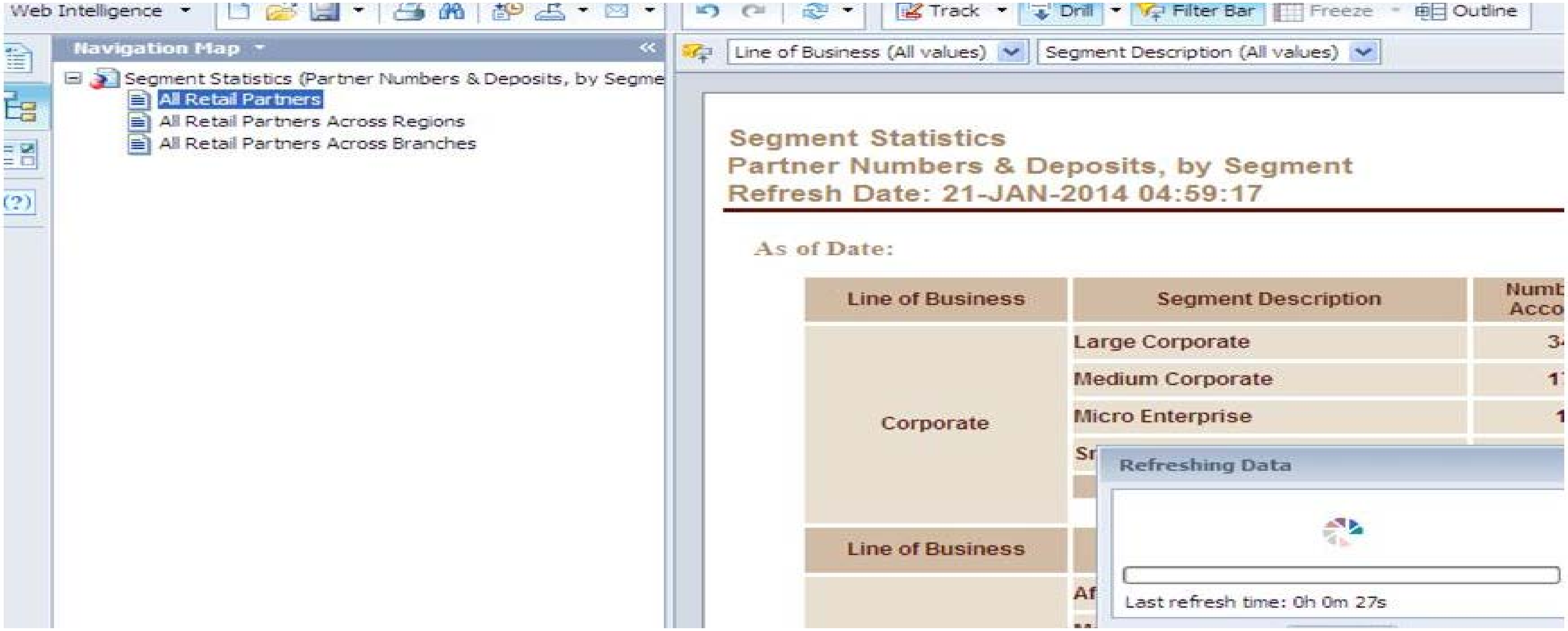
Scenario 2

ELT data loading and BI queries dramatic improvements

Testimonials

The Query which used to take 18 minutes is now taking only 27 seconds to get the data from the Database.

Below are the snapshots:



Segment Statistics (Partner Numbers & Deposits, by Segme

- All Retail Partners
- All Retail Partners Across Regions
- All Retail Partners Across Branches

Segment Statistics
Partner Numbers & Deposits, by Segment
Refresh Date: 21-JAN-2014 04:59:17

As of Date:

Line of Business	Segment Description	Numb Acco
Corporate	Large Corporate	3
	Medium Corporate	1
	Micro Enterprise	1
	Sr	
Line of Business		
	Af	

Refreshing Data

Last refresh time: 0h 0m 27s

Real Life Examples

Scenario 2

ELT data loading and BI queries dramatic improvements

Database action plan

- ✓ Restructured the partition key on the tables

```
SELECT FROM_DATE  , TO_DATE, ACCOUNT_NUMBER
FROM TABLE_BEFORE_MODIFIATION
WHERE '01-DEC-2012' BETWEEN FROM_DATE AND TO_DATE
```

SELECT STATEMENT, GOAL = ALL_ROWS			Cost=17373	Cardinality=2089220	Bytes=64765820	Time=244	IO cost=17264	CPU cost=1297607466
PARTITION RANGE ITERATOR			Cost=17373	Cardinality=2089220	Bytes=64765820	Time=244	IO cost=17264	CPU cost=1297607466
TABLE ACCESS FULL	Object owner=xxxx	Object name= TABLE_BEFORE_MODIFICATION	Cost=17373	Cardinality=2089220	Bytes=64765820	Time=244	IO cost=17264	CPU cost=1297607466

```
SELECT FROM_DATE , TO_DATE, ACCOUNT_NUMBER
FROM TABLE_AFTER_MODIFICATION
WHERE '01-DEC-2012' between from_date AND TO_DATE
```

SELECT STATEMENT, GOAL = ALL_ROWS			Cost=60	Cardinality=578568	Bytes=20249880	Time=1	IO cost=46	CPU cost=168083738
PARTITION RANGE ITERATOR			Cost=60	Cardinality=578568	Bytes=20249880	Time=1	IO cost=46	CPU cost=168083738
PARTITION HASH ALL			Cost=60	Cardinality=578568	Bytes=20249880	Time=1	IO cost=46	CPU cost=168083738
TABLE ACCESS FULL	Object owner= XXXX	Object name= TABLE_AFTER_MODIFICATION	Cost=60	Cardinality=578568	Bytes=20249880	Time=1	IO cost=46	CPU cost=168083738

Real Life Examples

		TABLE_BEFORE_MODIFICATION	TABLE_AFTER_MODIFICATION
PARTITION	RANGE	FROM_DATE	TO_DATE
SUB PARTITION	HASH(4)	N/A	FROM_DATE
INDEXES		BTREE GLOBAL INDEX	BTREE LOCAL INDEX

SELECT STATEMENT, GOAL = ALL_ROWS			Cost=3126	Cardinality=2072103	Bytes=64235193	Time=44	IO cost=3070	CPU cost=671176973
INDEX FAST FULL SCAN	Object owner=XXXX	Object name=TABLE_BEFORE_MODIFICATION	Cost=3126	Cardinality=2072103	Bytes=64235193	Time=44	IO cost=3070	CPU cost=671176973

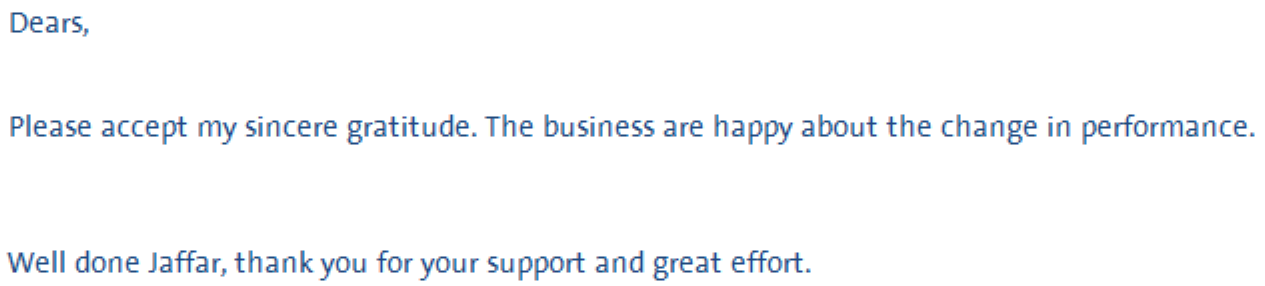
BTREE LOCAL INDEX ON FROM_DATE, TO_DATE ON AFTER_MODIFIATON_TABLE

SELECT STATEMENT, GOAL = ALL_ROWS			Cost=1037	Cardinality=821359	Bytes=28747565	Time=15	IO cost=1031	CPU cost=73807108
PARTITION RANGE ITERATOR			Cost=1037	Cardinality=821359	Bytes=28747565	Time=15	IO cost=1031	CPU cost=73807108
PARTITION HASH ALL			Cost=1037	Cardinality=821359	Bytes=28747565	Time=15	IO cost=1031	CPU cost=73807108
TABLE ACCESS BY LOCAL INDEX ROWID	Object owner=XXXX	Object name= TABLE_AFTER_MODIFICATION	Cost=1037	Cardinality=821359	Bytes=28747565	Time=15	IO cost=1031	CPU cost=73807108
INDEX RANGE SCAN	Object owner=XXXX	Object name= INDEX_ON_AFTER_TABLE_MODI	Cost=988	Cardinality=7392		Time=14	IO cost=982	CPU cost=73391858

BTREE LOCAL INDEX ON TO_DATE, FROM_DATE

SELECT STATEMENT, GOAL = ALL_ROWS			Cost=555	Cardinality=821359	Bytes=28747565	Time=8		IO cost=549	CPU cost=75584538
PARTITION RANGE ITERATOR			Cost=555	Cardinality=821359	Bytes=28747565	Time=8		IO cost=549	CPU cost=75584538
PARTITION HASH ALL			Cost=555	Cardinality=821359	Bytes=28747565	Time=8		IO cost=549	CPU cost=75584538
TABLE ACCESS BY LOCAL INDEX ROWID	Object owner=OFDM	Object name= TABLE_AFTER_MODIFIATION	Cost=555	Cardinality=821359	Bytes=28747565	Time=8		IO cost=549	CPU cost=75584538
INDEX RANGE SCAN	Object owner=OFDM	Object name= INDEX_ON_AFTER_TAB_MODIFICATION	Cost=502	Cardinality=8142		Time=8		IO cost=496	CPU cost=75131718

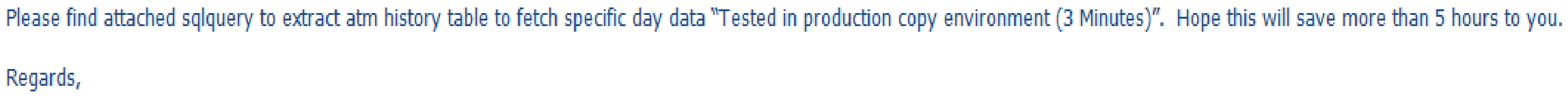
Real Life Examples



Dears,

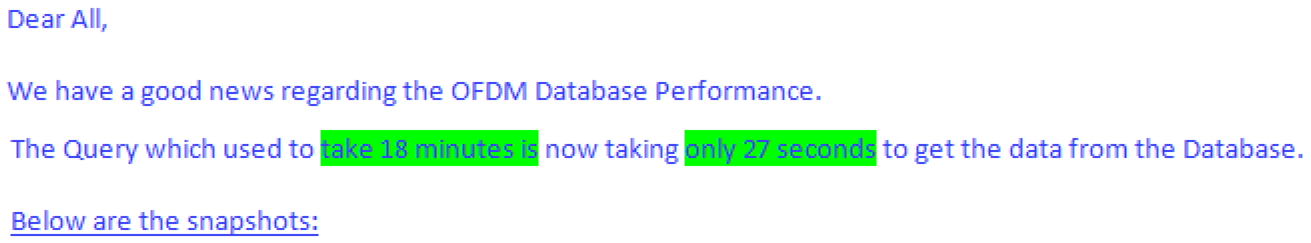
Please accept my sincere gratitude. The business are happy about the change in performance.

Well done Jaffar, thank you for your support and great effort.



Please find attached sqlquery to extract atm history table to fetch specific day data "Tested in production copy environment (3 Minutes)". Hope this will save more than 5 hours to you.

Regards,



Dear All,

We have a good news regarding the OFDM Database Performance.

The Query which used to take 18 minutes is now taking only 27 seconds to get the data from the Database.

Below are the snapshots:

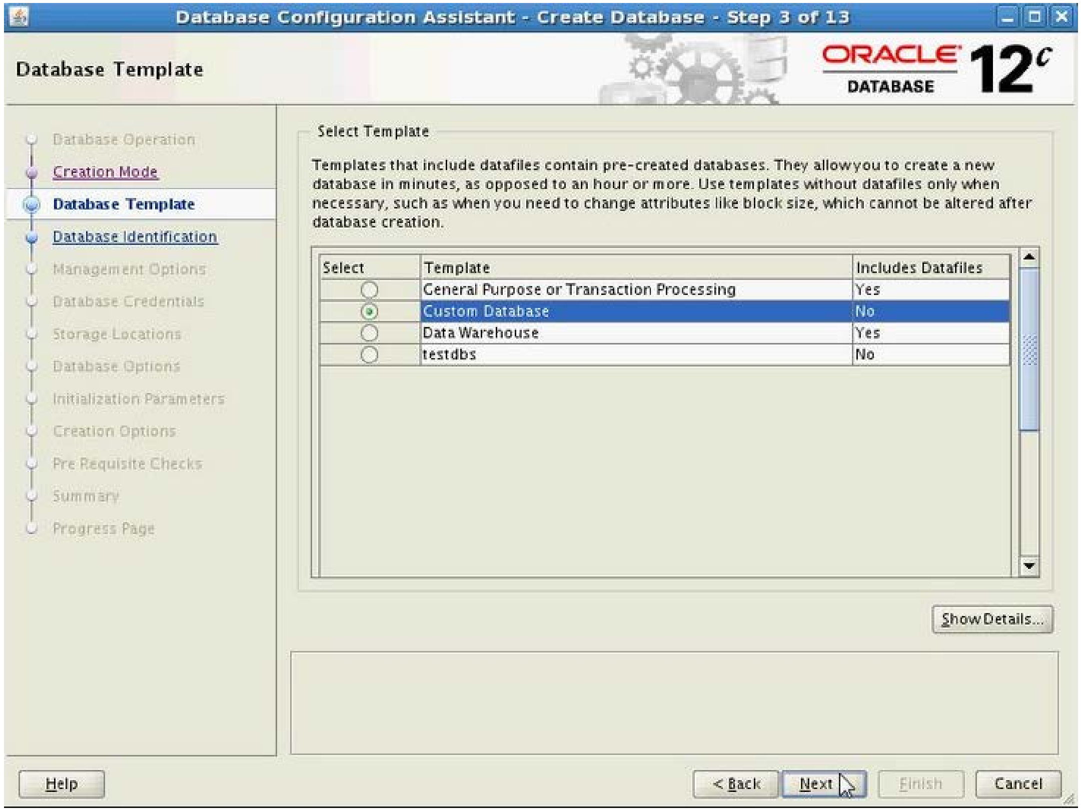
Standard/Golden Rules

Standard/Golden Rules for VLDB configuration

Standard/Golden Rules

Standard/Golden Rules for VLDB configuration

- ✓ Choose a right template in DBCA



Database Configuration Assistant - Create Database - Step 3 of 13

Database Template

- Database Operation
- Creation Mode
- Database Template
- Database Identification
- Management Options
- Database Credentials
- Storage Locations
- Database Options
- Initialization Parameters
- Creation Options
- Pre Requisite Checks
- Summary
- Progress Page

Select Template

Templates that include datafiles contain pre-created databases. They allowyou to create a new database in minutes, as opposed to an hour or more. Use templates without datafiles only when necessary, such as when you need to change attributes like block size, which cannot be altered after database creation.

Select	Template	Includes Datafiles
○	General Purpose or Transaction Processing	Yes
●	Custom Database	No
○	Data Warehouse	Yes
○	testdbs	No

Standard/Golden Rules

Standard/Golden Rules for VLDB configuration

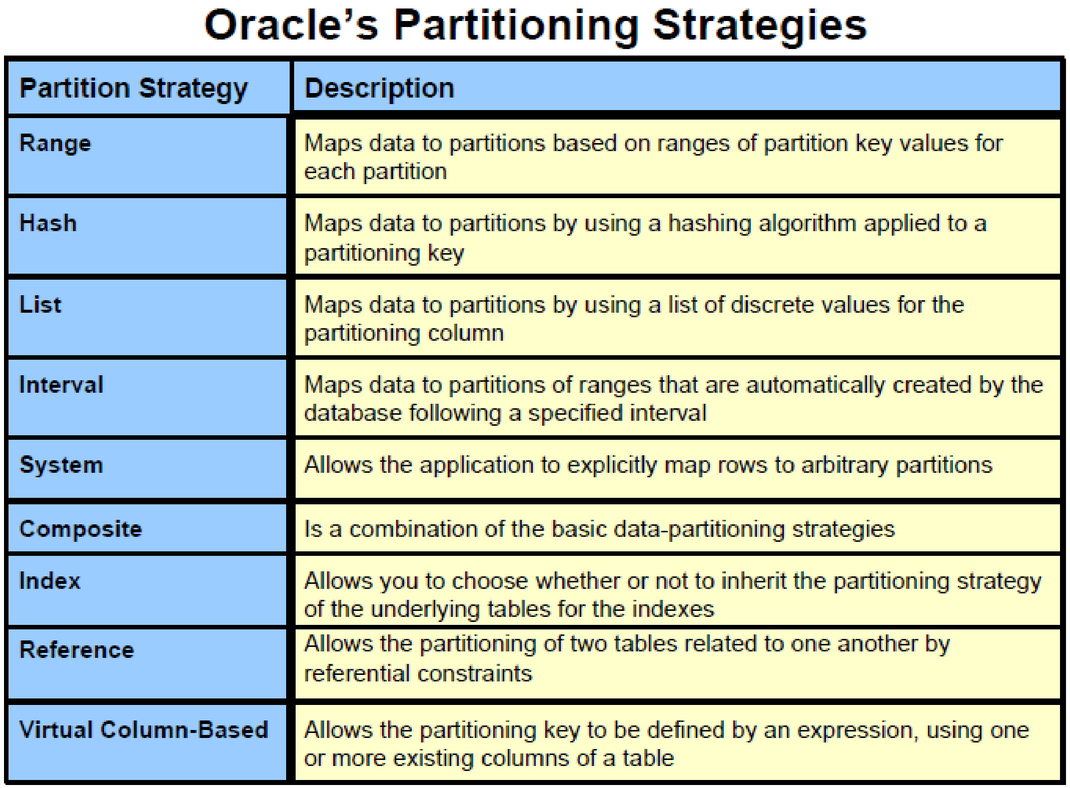
- ✓ Choose right template
- ✓ Configuring parameters
  - ❑ Block Size
  - ❑ db_file_multiblock_read_count
  - ❑ SGA (memory_target/sga_target/pga_aggregate_target)
  - ❑ Parallelism
    - o Parallel_min_servers
    - o Parallel_max_servers
    - o Parallel_execution_message_size
    - o Parallel_degree_policy

    - o DOP (degree of parallelism)
    - o Single instance ,DOP = PARALLEL_THREADS_PER_CPU X CPU_COUNT
    - o RAC , DOP = PARALLEL_THREADS_PER_CPU X CPU_COUNT X INSTANCE_COUNT

Standard/Golden Rules

Standard/Golden Rules for VLDB configuration

- ✓ Choose right template
- ✓ Configuring parameters
- ✓ Table and Index Partitioning



Oracle's Partitioning Strategies

Partition Strategy	Description
Range	Maps data to partitions based on ranges of partition key values for each partition
Hash	Maps data to partitions by using a hashing algorithm applied to a partitioning key
List	Maps data to partitions by using a list of discrete values for the partitioning column
Interval	Maps data to partitions of ranges that are automatically created by the database following a specified interval
System	Allows the application to explicitly map rows to arbitrary partitions
Composite	Is a combination of the basic data-partitioning strategies
Index	Allows you to choose whether or not to inherit the partitioning strategy of the underlying tables for the indexes
Reference	Allows the partitioning of two tables related to one another by referential constraints
Virtual Column-Based	Allows the partitioning key to be defined by an expression, using one or more existing columns of a table

Standard/Golden Rules

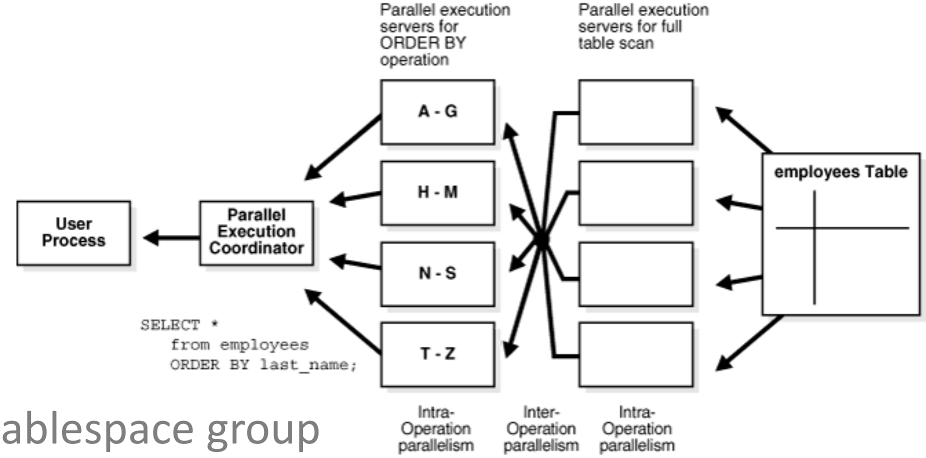
Standard/Golden Rules for VLDB configuration

- ✓ Choose right template
- ✓ Configuring parameters
- ✓ Table and Index Partitioning
- ✓ Avoid or minimize number of indexes
- ✓ Optimal use of index and MV
- ✓ Right index selection, b-tree vs bitmap
- ✓ Bigfile tablespace size and Temporary tablespace groups
- ✓ Data Compression
- ✓ Redo log size and multiplexing
- ✓ Database File System (DBFS)
- ✓ Parallelism

Standard/Golden Rules

Standard/Golden Rules for VLDB configuration

- ✓ Choose right template
- ✓ Configuring parameters
- ✓ Table and Index Partitioning
- ✓ Avoid or minimize number of indexes
- ✓ Optimal use of index and MV
- ✓ Right index selection, b-tree vs bitmap
- ✓ Bigfile tablespace size and Temporary tablespace group
- ✓ Data Compression
- ✓ Database File System (DBFS)
- ✓ Parallelism


Parallel execution servers for ORDER BY operation
Parallel execution servers for full table scan
A - G
H - M
N - S
T - Z
employees Table
User Process
Parallel Execution Coordinator
Intra-Operation parallelism
Inter-Operation parallelism
Intra-Operation parallelism

```
SELECT *
   from employees
   ORDER BY last_name;
```

- Large table scans, joins, or partitioned index will gain performance improvements
- Large index creation will be benefited
- Bulk inserts, updates, merges and deletion will gain performance

Standard/Golden Rules

Optimizer Statistics Management Approach

- AUTO SAMPLE SIZE
- INCREMENTAL STATISTICS
- CONCURRENT STATISTICS COLLECTIONS
- LOCK STATS COLLECTION

Standard/Golden Rules

Optimizer Statistics Management

- AUTO SAMPLE SIZE

  - ✓ Was much improved in Oracle 11gR1
  - ✓ Faster than sampling – speed of a 10% sample
  - ✓ More accuracy to compute statistics – same as 100% accuracy sample
  - ✓ Affects histogram gathering and index stats gathering
  - ✓ ESTIMATE_PERCENT is set to default AUTO_SAMPLE_SIZE

```
DBMS_STATS.GATHER_TABLE_STATS(‘SCHEMA',‘TABLE_NAME',estimate_percent=>DBMS_STATS.AUTO_S
AMPLE_SIZE,granularity=>'AUTO');
```

Standard/Golden Rules

Optimizer Statistics Management

- Incremental statistics

  - ✓ Was first introduced in Oracle 11gR1 to improve performance of gathering stats on large partitioned tables
  - ✓ Stats for touched partition(s)
  - ✓ Global stats are built from partition stats

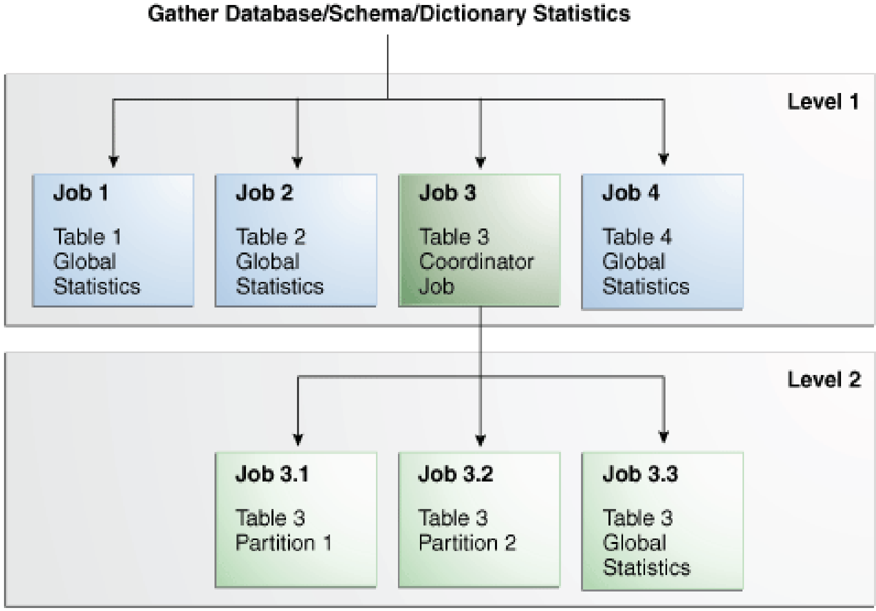
```
SQL> select dbms_stats.get_prefs('INCREMENTAL') from dual;

SQL>exec DBMS_STATS.SET_TABLE_PREFS(‘SCHEMA',‘TABLE_NAME','INCREMENTAL','TRUE');

SQL>exec DBMS_STATS.SET_TABLE_PREFS(‘SCHEMA',‘TABLE_NAME','PUBLISH','TRUE');
```

Standard/Golden Rules

Optimizer Statistics Management

- ➢ Concurrent statistics

- ✓ Was much improved in Oracle 11gR2
- ✓ Gather stats concurrently on multiple tables
- ✓ Goal is to reduce the overall time required for gather stats
- ✓ The global preference CONCURRENT must be set to TRUE
- ✓ Oracle uses Job Scheduler and Advanced Queuing (AQ) components to perform the action
- ✓ CREATE JOB, MANAGE SCHEDULER, MANAGE ANY QUEUE
- ✓ JOB_QUEUE_PROCESSES=>4



Standard/Golden Rules

Optimizer Statistics Management

- <u>Concurrent statistics</u>

```
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DEFAULT_PLAN';

SQL> ALTER SYSTEM SET JOB_QUEUE_PROCESSES=8;

SQL> exec BEGIN DBMS_STATS.SET_GLOBAL_PREFS('CONCURRENT','ALL|MANUAL|AUTOMATIC');

SQL> EXEC DBMS_STATS.SET_GLOBAL_PREFS('CONCURRENT','OFF');
```

Standard/Golden Rules

Backup & Recovery best practices on VLDB

- Developing optimal Backup/Recovery strategies
  - (RTO=recovery time object & RPO=recovery point objective)
- Periodic Full backup and frequent cumulative incremental backups
- Fast incremental backups - BCT
- Use parallelism, limit the backup piece size
- Leverage Read only tablespaces and backup compression
- Multi-section backups
- Backup compression
- Create a service for RAC database for backup workload management
- Offload backups to Data Guard
- Other backup solutions, snapshot, split mirror etc
- Compress data
- Backup to FRA (disk) and then to TAPE

Standard/Golden Rules

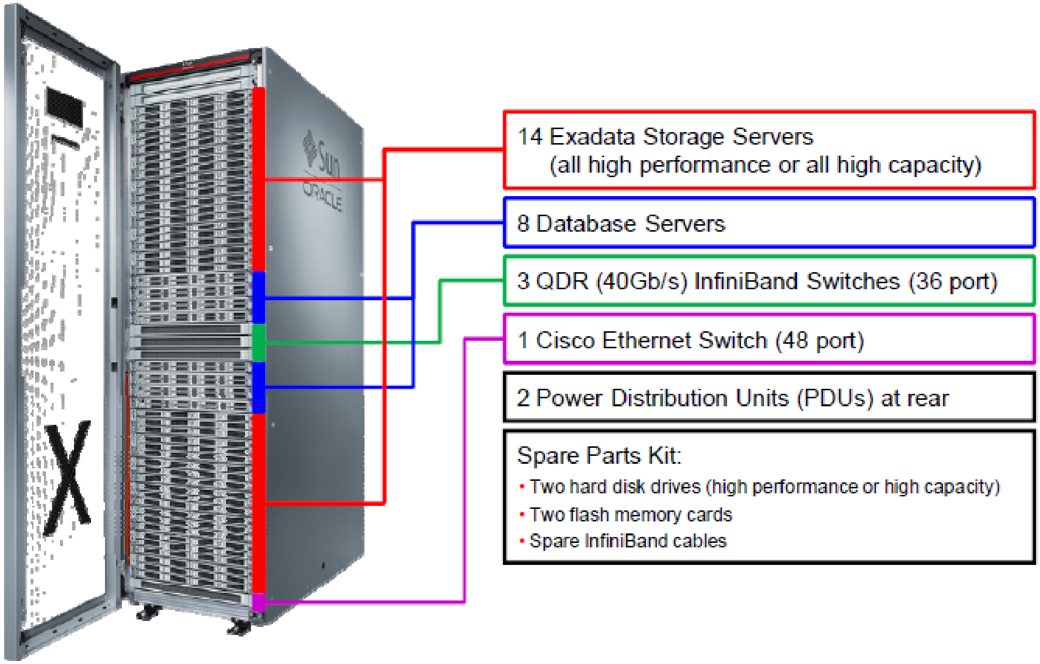
Backup approaches

- Level 0 (full) with BCT (Fast Incremental Backups)
- Image copy (full) and Fast Incremental + Incrementally updated backup
- Data Guard + BCT

Standard/Golden Rules

Exadata solutions

- Hybrid Columnar Compression
- Smart Scan
- Smart Flash Cache
- Smart Flash Cache Log
- IORM



14 Exadata Storage Servers
(all high performance or all high capacity)

8 Database Servers

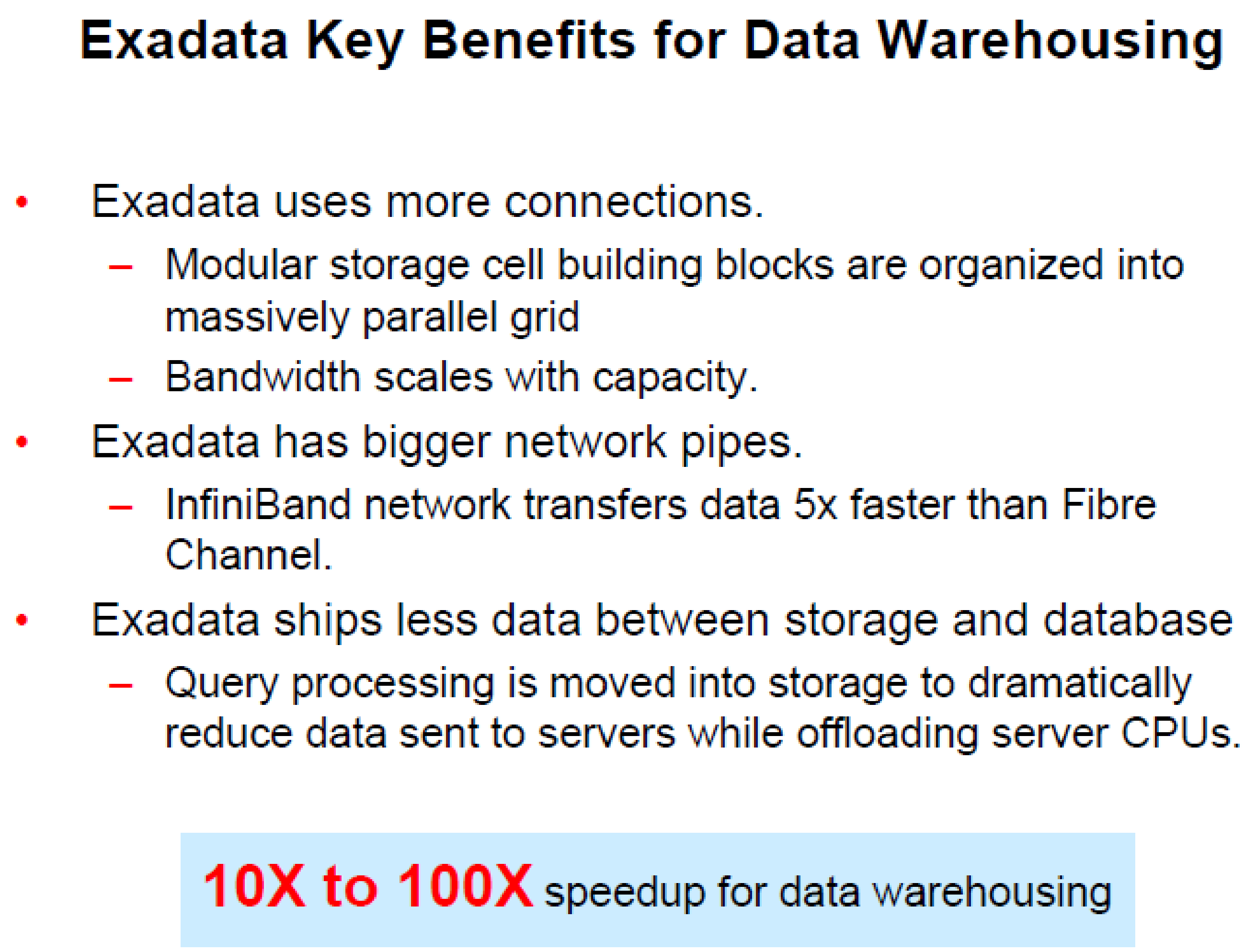
3 QDR (40Gb/s) InfiniBand Switches (36 port)

1 Cisco Ethernet Switch (48 port)

2 Power Distribution Units (PDUs) at rear

Spare Parts Kit:
- Two hard disk drives (high performance or high capacity)
- Two flash memory cards
- Spare InfiniBand cables

Standard/Golden Rules



Exadata Key Benefits for Data Warehousing

- Exadata uses more connections.
  - Modular storage cell building blocks are organized into massively parallel grid
  - Bandwidth scales with capacity.
- Exadata has bigger network pipes.
  - InfiniBand network transfers data 5x faster than Fibre Channel.
- Exadata ships less data between storage and database
  - Query processing is moved into storage to dramatically reduce data sent to servers while offloading server CPUs.

10X to 100X speedup for data warehousing

Standard/Golden Rules

ASM Best Practices for VLDBs

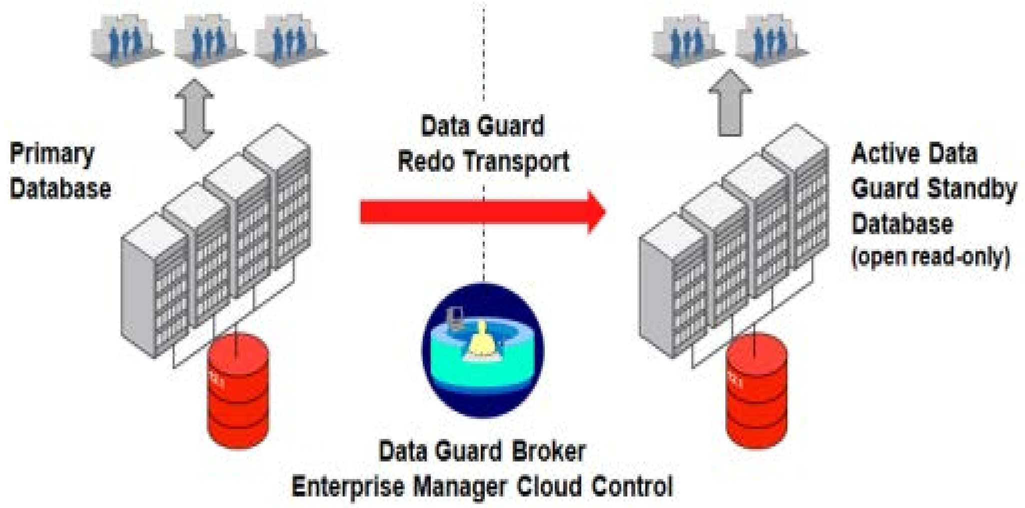
- Large allocation units
  - Between 1MB and 64MB (4,8,16,32 and 64)
  - Use 64MB for a very large data warehousing system, which will reduce ASM memory requirements, and also improve ASM instance startup time

- Disk groups with an External Redundancy

Standard/Golden Rules

Active Data Guard Solution

- Offload queries / reports to Standby Database
- Offload backups to Standby Database



Standard/Golden Rules

Performance optimization

- Indexing (bitmap index)
  - Bitmap indexes are widely used in DW application with huge data and low concurrent DML operations
  - Reduces storage requirements
  - Hardware performance gain, CPU and small amount of memory
- Initialization Parameters
  - OPTIMIZER_MODE = ALL_ROWS | FIRST_ROWS | FIRST_ROWS_n
  - QUERY_REWRITE_ENABLED

A big thank you all

for

listening ...

You can write me at **sjaffarhussain@gmail.com**