NoCOUG
Summer Conference 2009

Date: Thursday, August 20, 2009

Time: 01:00 PM – 02:00 PM

Venue: Chevron, Room 1240

San Ramon, CA

ORACLE®

Tuning Multi-Terabyte Database for High Performance - An Architecture Approach

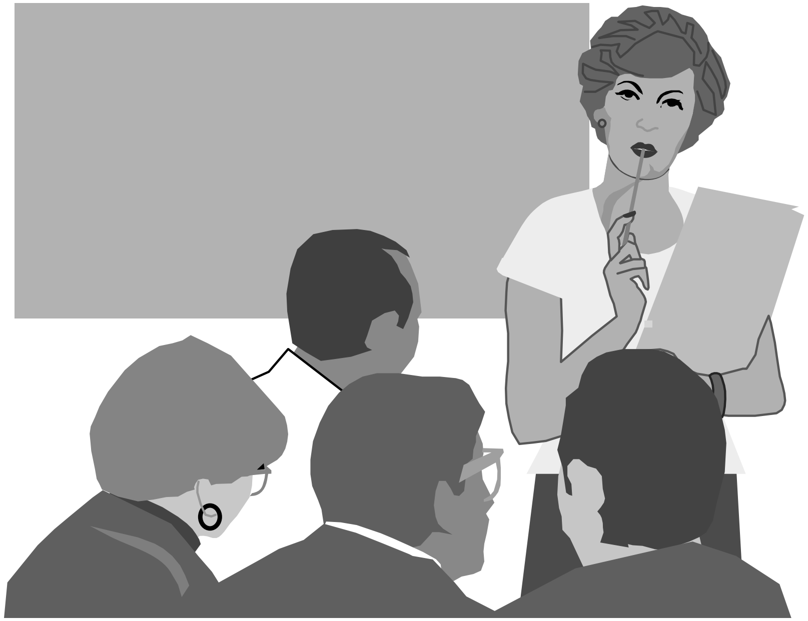
Daniel T. Liu
Principal Solution Architect

Agenda

1. **Introduction**
2. **Performance Tuning Approach**
3. **Indexing**
4. **Aggregation**
5. **ETL Loading**
6. **Database Configuration**
7. **Concurrency**

Agenda

8. **Server Configuration**
9. **Storage Configuration**
10. **Exadata**
11. **Monitoring and Testing**
12. **Others**



Introduction

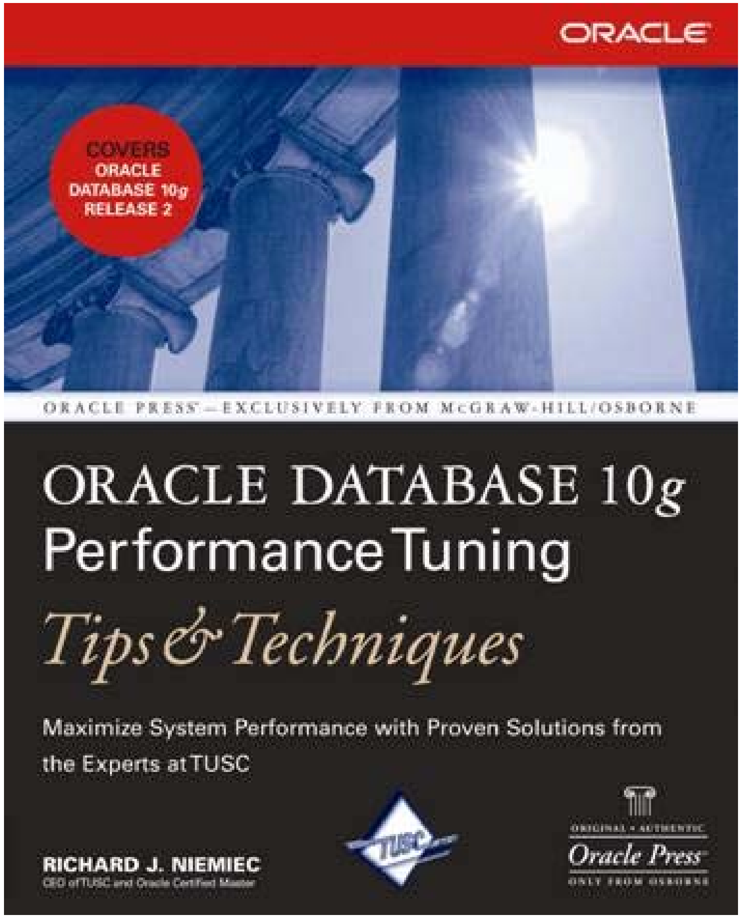
- People, Process, **Technology**
- Performance Tuning is everyone's business
  - DBAs
  - Developers
  - ETL team
  - System/Storage Admin
  - Architect Team
- Managing Expectations
- An Architecture Approach
- Data Volume Growth

Data Volume Growth

Byte	Value	Name	Value
1,000	1.E+03	kilobyte	(KB)
1,000,000	1.E+06	megabyte	(MB)
1,000,000,000	1.E+09	gigabyte	(GB)
1,000,000,000,000	1.E+12	terabyte	(TB)
1,000,000,000,000,000	1.E+15	petabyte	(PB)
1,000,000,000,000,000,000	1.E+18	exabyte	(EB)
1,000,000,000,000,000,000,000	1.E+21	zettabyte	(ZB)
1,000,000,000,000,000,000,000,000	1.E+24	yottabyte	(YB)

Data Volume Growth

- 2K – A typewritten page
- 5M – The complete works of Shakespeare
- 10 M – One minute of high fidelity sound
- 2 T – Information generated on YouTube in one day
- 10T – 530,000,000 miles of bookshelves at Library of congress
- 20P – All hard-disk drives in 1995 (or your database in 2010)



Data Volume Growth

- 700P – Data of 700,000 companies with Revenues less than $200M
- 1E – Combined Fortune 1000 company database (1P each)
- 1E – Next 9000 world company databases (average 100T each)
- 8E – Capacity of ONE Oracle10g Database (CURRENT)
- <u>12E to 16E – Info generated before 1999 (memory resident in 64-bit)</u>
- 16E – Addressable memory with 64-bit (CURRENT)
- <u>161E – New information in 2006 (most images not stored in DB)</u>
- 1Z – 1000E (Zettabyte – Grains of sand on beaches – 125 Oracle DBs)
- 100TY – Yottabytes – Addressable memory 128-bit (FUTURE)

ORACLE

Performance Tuning – An Architecture Approach

- End-to-End Approach
  - Web tier
  - Application tier
  - Database tier
  - Storage
  - Network
- Design and Configuration
  - Hardware
  - Logical model
  - Physical model
  - System Management

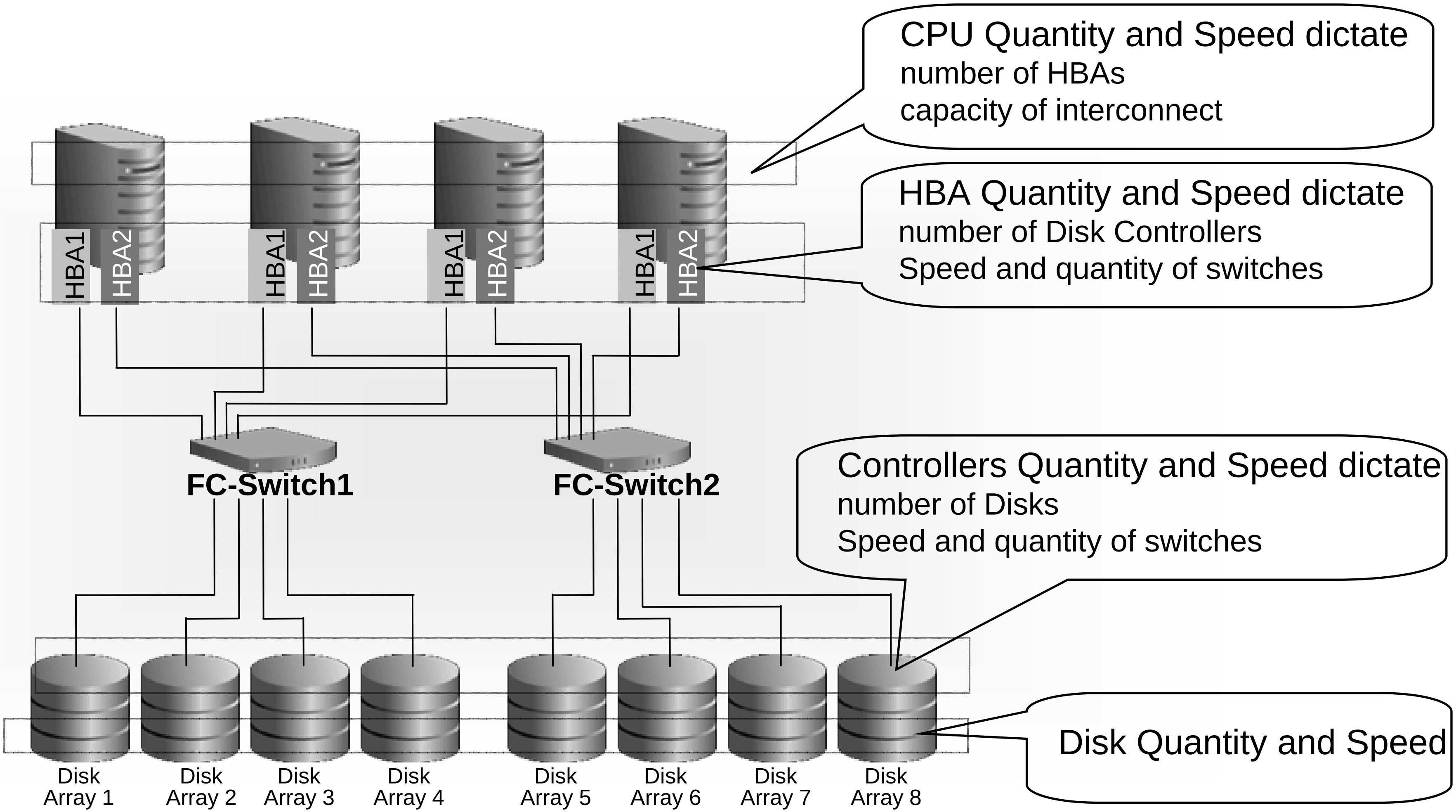
ORACLE

Performance Tuning – A Mathematical Approach

- A Balanced Configuration
- CPU Throughput
- HBA Throughput
- Network Throughput
- Disk Throughput
- Memory and CPU ratios

Balanced Configuration

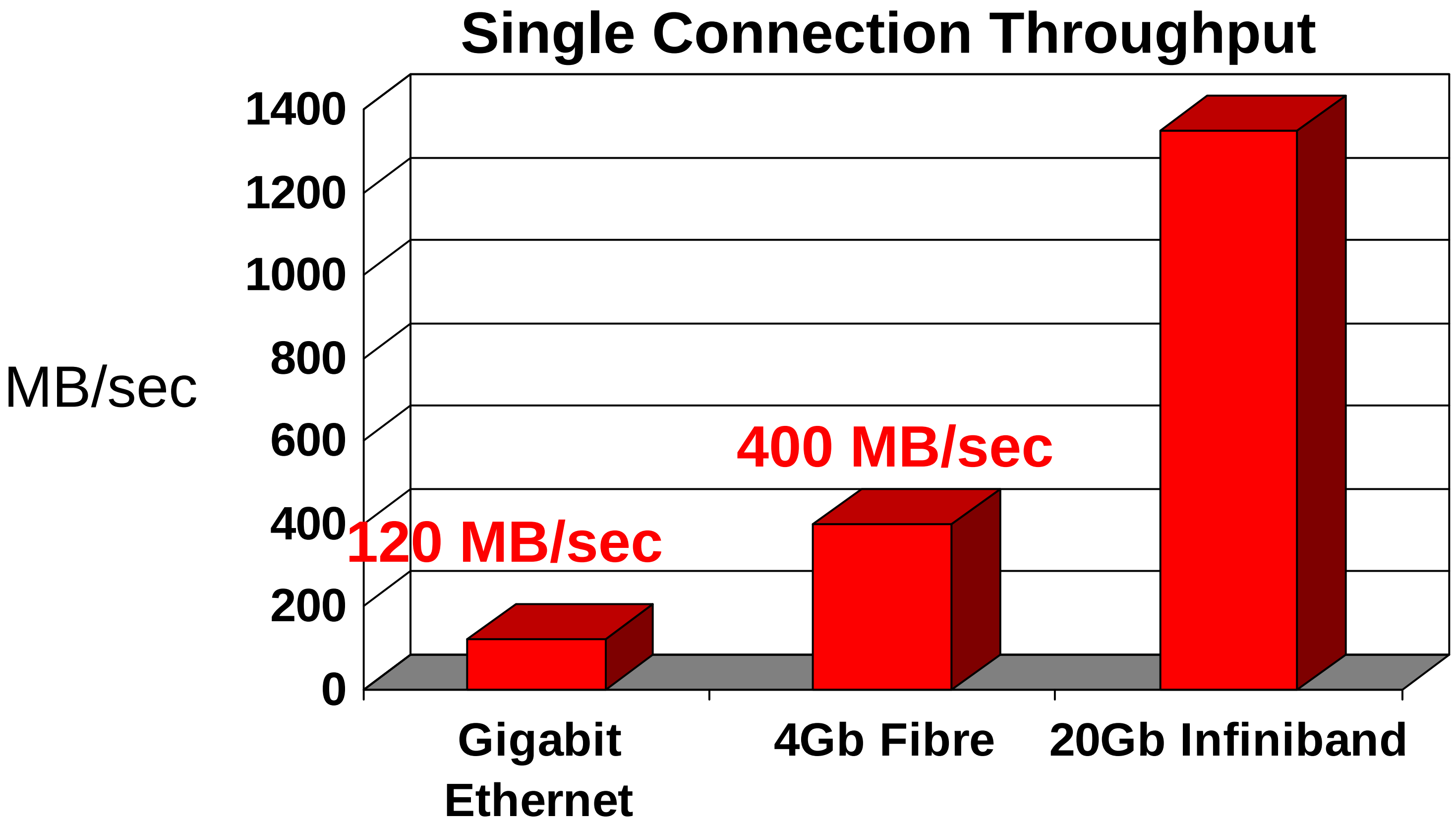
“The weakest link” defines the throughput



Data Warehouse hardware configuration best practices

- Build a balance hardware configuration
  - Total throughput = # cores X 100-200 MB (depends on chip set)
  - Total HBA throughput = Total core throughput
    - If total core throughput =1.6GB will need 4 4Gb HBAs
  - Use 1 disk controller per HBA Port (throughput capacity must be equal)
  - Switch must be same capacity as HBA and disk controllers
  - Max of 10 physical disks per controller(Use smaller drives 146 or 300 GB)
- Minimum of 4GB of Memory per core (8GB if using compression)

Throughput in Real Systems MB/sec



- Graph shows throughput achieved in real-world deployments
  - Infiniband is held back by PCIe 1.0 x8 bus on typical host systems

CPU Throughput

1 CPU	4 CPU	8 CPUs	16 CPUs	20 CPUs
100	400	800	1,600	2,000
200	800	1,600	3,200	4,000
MB/Sec	MB/Sec	MB/Sec	MB/Sec	MB/Sec

HBA Throughput

	1 HBA	2 HBAs	4 HBAs	8 HBAs	16 HBAs
2 Gb	200	400	800	1,600	3,200
4 Gb	400	800	1,600	3,200	6,400
	MB/Sec	MB/Sec	MB/Sec	MB/Sec	MB/Sec

15,000 RPM SAS Disk

1 Disk	2 Disks	4 Disks	8 Disks	12 Disks
90	180	360	720	1,080
MB/Sec	MB/Sec	MB/Sec	MB/Sec	MB/Sec

CPU and Memory

1 CPU	4 CPU	8 CPUs	16 CPUs	20 CPUs
4	16	32	64	80
8	32	64	128	160
GB	GB	GB	GB	GB

Indexing

- Issues
  - What’s the right amount of Indexes
  - Why indexes is not being used sometime
  - Why indexes is not helping on Query
  - Indexing and impact on ETL process
- Approaches
  - Monitoring index usage
  - Remove unused indexes
    - http://www.dbazine.com/oracle/or-articles/liu3
  - Invisible index (11g only)
  - Dropping/rebuilding Indexes for ETL process

ORACLE

Invisible Index

- An invisible index is an index that is ignored by the optimizer unless you explicitly set the OPTIMIZER_USE_INVISIBLE_INDEXES initialization parameter to TRUE at the session or system level. The default value for this parameter is FALSE.
- Making an index invisible is an alternative to making it unusable or dropping it. Using invisible indexes, you can do the following:
  - Test the removal of an index before dropping it.
  - Use temporary index structures for certain operations or modules of an application without affecting the overall application.

Invisible Index

- Here are a few examples:

```
SQL> alter index emp_id_idx invisible;
SQL> alter index emp_id_idx visible;
SQL> create index emp_id_idx on emp
  (emp_id) invisible;
```

Aggregation

- Issues
  - What's the right amount of aggregation?
  - Does the system need more aggregates/materialized views/pre-built DSS summaries?
- Approaches
  - Aggregates should be used strategically based on report requirements
  - SQL Query Result Cache (11g only)

Data Warehouse Workload

- Analyze data across large data sets
  - reporting
  - forecasting – trend analysis
  - data mining
- Use parallel execution for good performance
- Result
  - very IO intensive workload – direct reads from disk
  - memory is less important
    - mostly execution memory

Data Warehouse Query Example

```
select p.prod_category
,      sum(s.amount_sold) revenue
from products p
,    sales    s
where s.prod_id = p.prod_id
and   s.time_id
  between to_date('01-JAN-2006','dd-MON-yyyy')
  and     to_date('31-DEC-2006','dd-MON-yyyy')
group by rollup (p.prod_category)
```

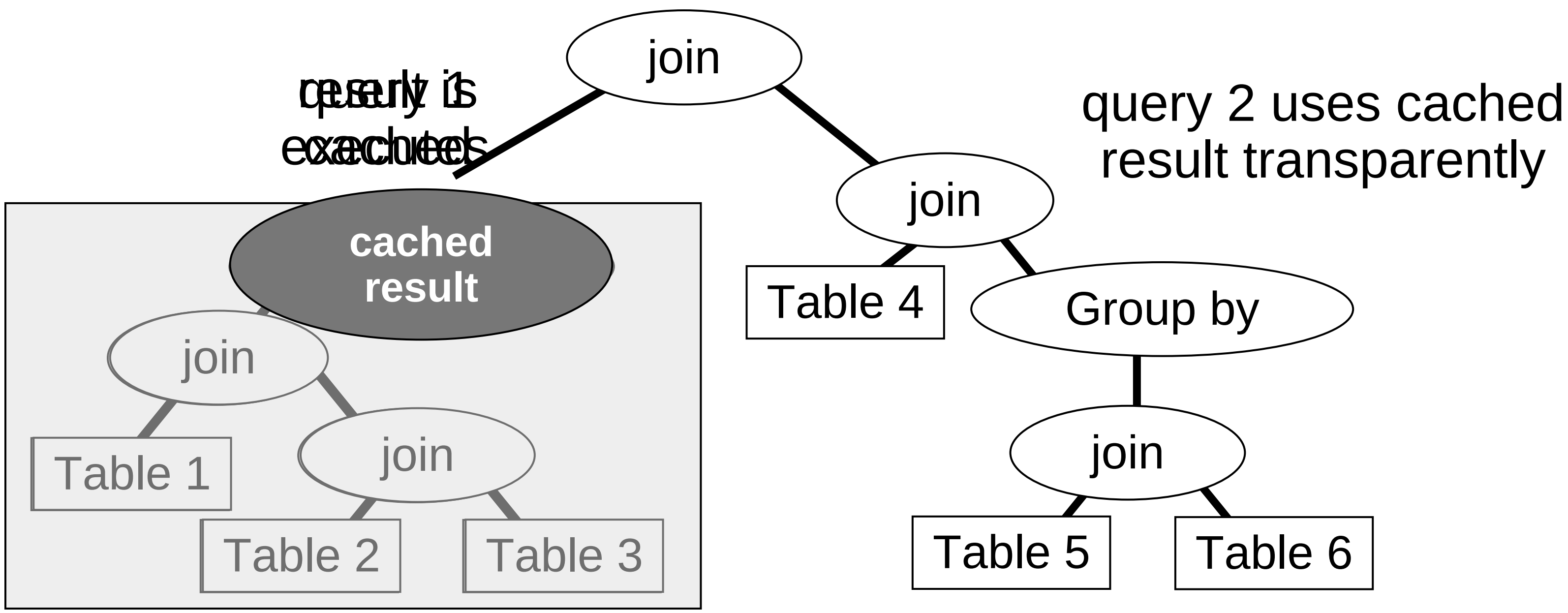
- accesses very many rows
- returns few rows

Data Warehouse Configuration Sizing

- Critical success factors
  - IO throughput
    - number of physical disks
    - number of channels to disks
  - CPU power
- Everything else follows
  - Storage capacity (500GB – 1TB common)
    - use surplus for high availability and ILM
  - Memory capacity (4GB/CPU is “standard”)
    - use surplus for... **RESULT CACHE**

SQL Query Result Cache Benefits

- Caches results of queries, query blocks, or pl/sql function calls
- Read consistency is enforced
  - DML/DDL against dependent database objects invalidates cache
- Bind variables parameterize cached result with variable values



SQL Query Result Cache Enabling

- result_cache_mode initialization parameter
  - MANUAL, use hints to populate and use
  - FORCE, queries will use cache without hint
  - AUTO, The optimizer determines the results that need to be stored in the cache based on repetitive executions
- result_cache_max_size initialization parameter
  - default is dependent on other memory settings

    (0.25% of memory_target or 0.5% of sga_target or 1% of shared_pool_size)
  - 0 disables result cache
  - never >75% of shared pool (built-in restriction)
- /*+ RESULT_CACHE */ hint in queries

SQL Query Result Cache Example

- Use RESULT_CACHE hint

```
select /*+ RESULT_CACHE */ p.prod_category
,       sum(s.amount_sold) revenue
from products p
,     sales    s
where s.prod_id = p.prod_id
and   s.time_id
  between to_date('01-JAN-2006','dd-MON-yyyy')
  and     to_date('31-DEC-2006','dd-MON-yyyy')
group by rollup (p.prod_category)
```

SQL Query Result Cache Example

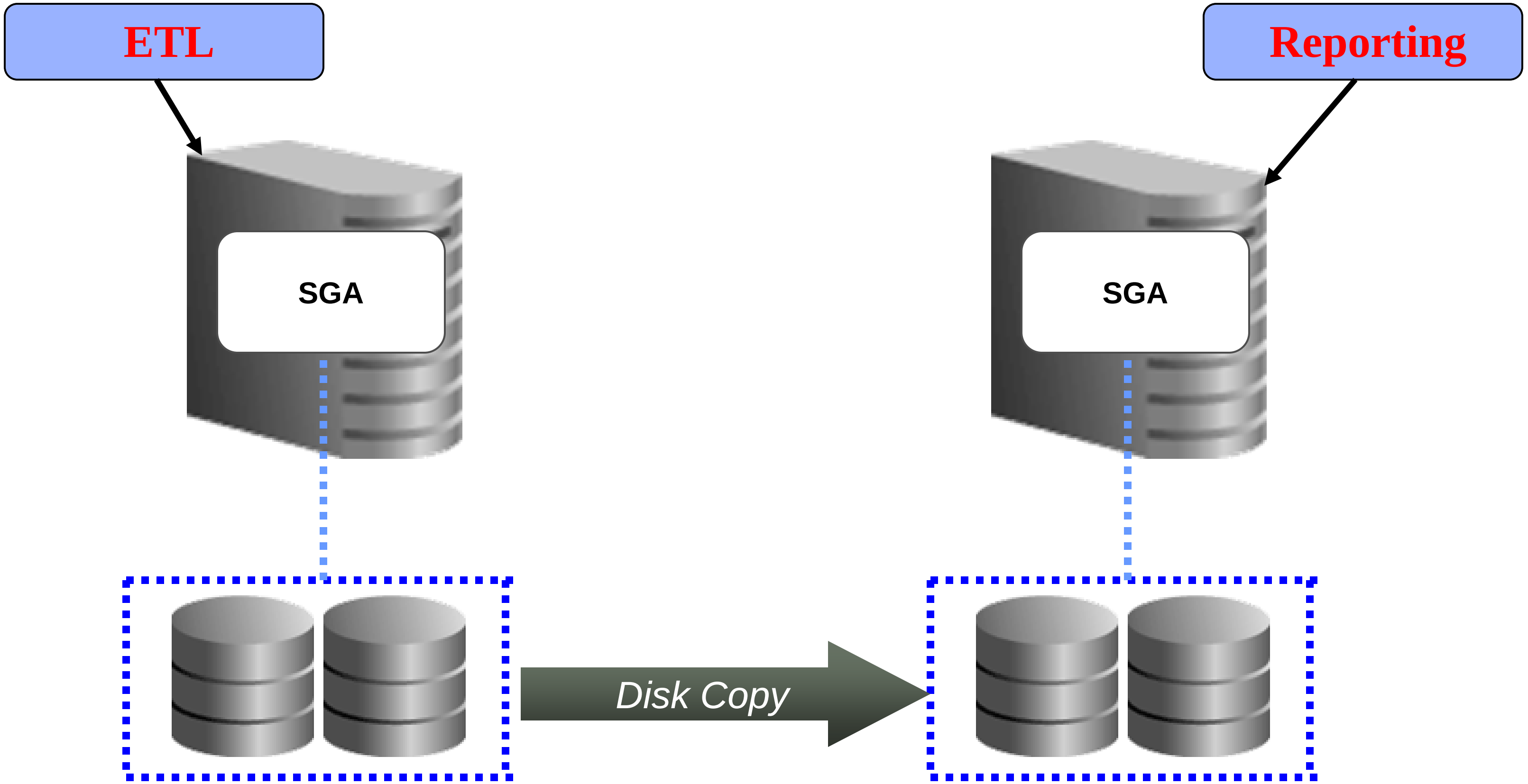
- Execution plan fragment

```
--------------------------------------------------------------------
| Id  | Operation                     | Name                       |
--------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                            |
|   1 |  RESULT CACHE                 | fz6cm4jbpcwh48wcyk60m7qypu |
|   2 |   SORT GROUP BY ROLLUP        |                            |
|*  3 |    HASH JOIN                  |                            |
|   4 |     PARTITION RANGE ITERATOR|                            |
|*  5 |      TABLE ACCESS FULL        | SALES                      |
|   6 |     VIEW                      | index$_join$_001           |
|*  7 |      HASH JOIN                |                            |
|   8 |       INDEX FAST FULL SCAN    | PRODUCTS_PK                |
|   9 |       INDEX FAST FULL SCAN    | PRODUCTS_PROD_CAT_IX       |
--------------------------------------------------------------------
```

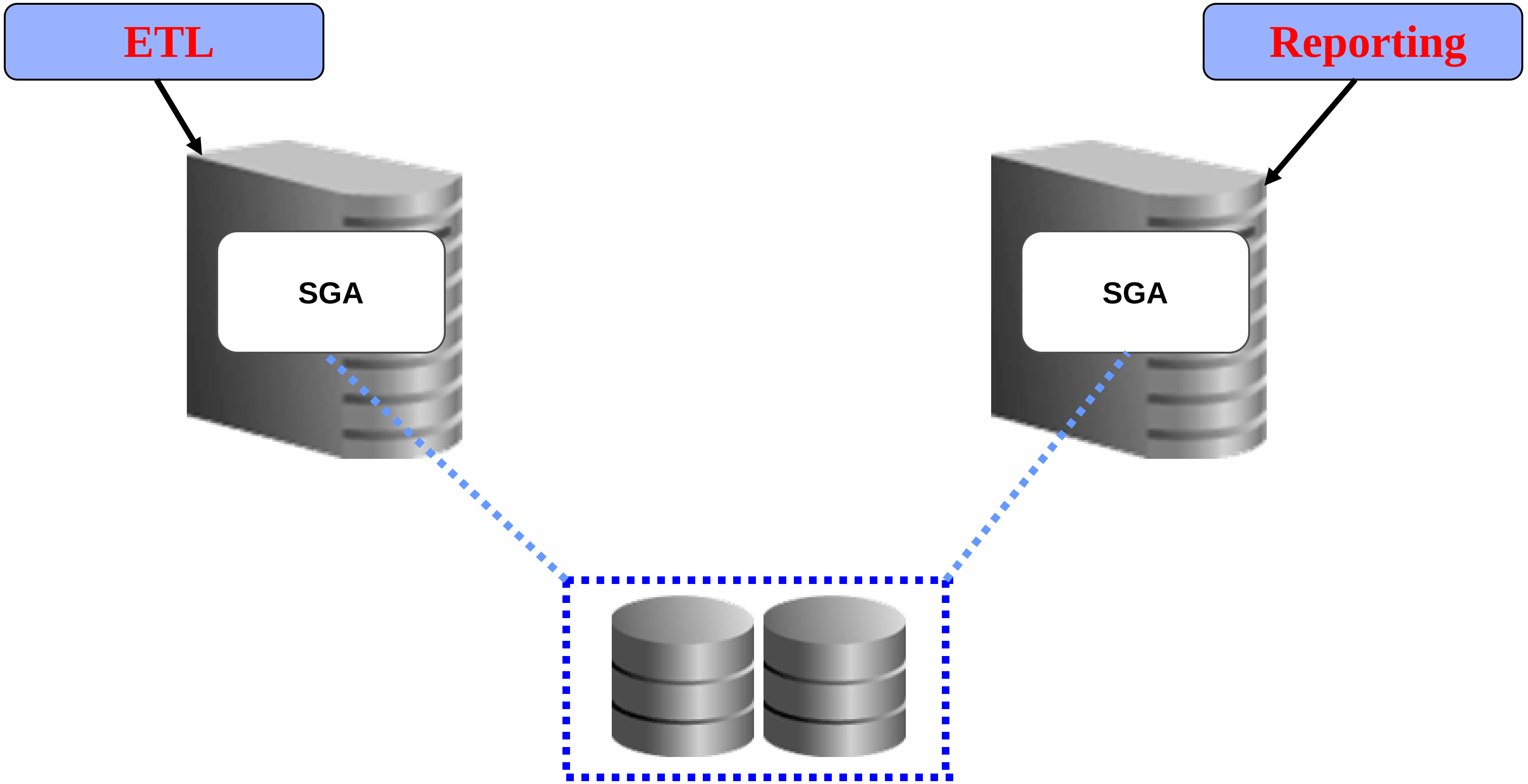
ETL Loading

- Issues
  - Reload vs. Update
  - Index rebuilding
  - Performance statistics collection
- Approaches
  - Real Application Clusters
  - Partitioning
  - Improve I/O throughput (including HBAs)

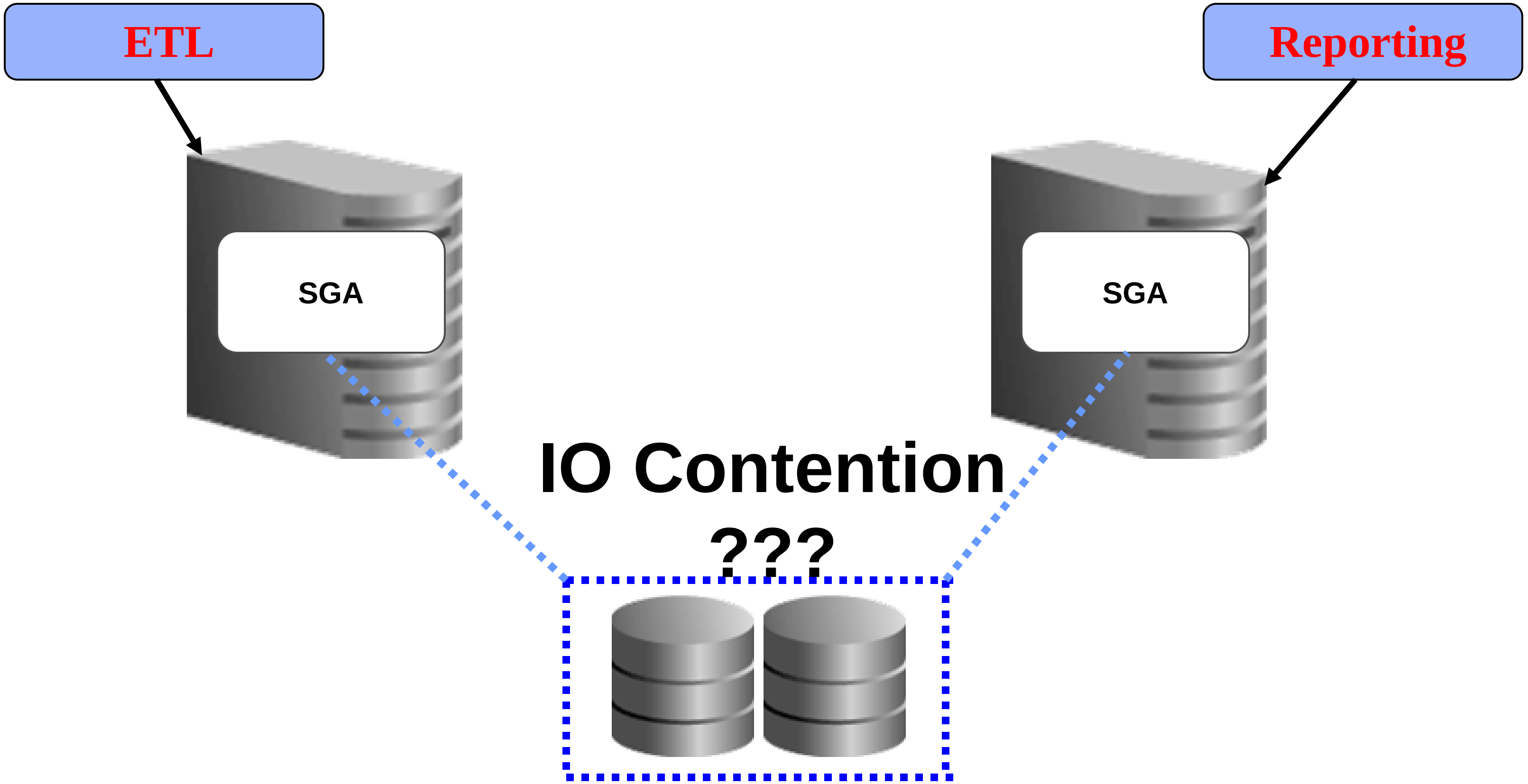
Common Architecture



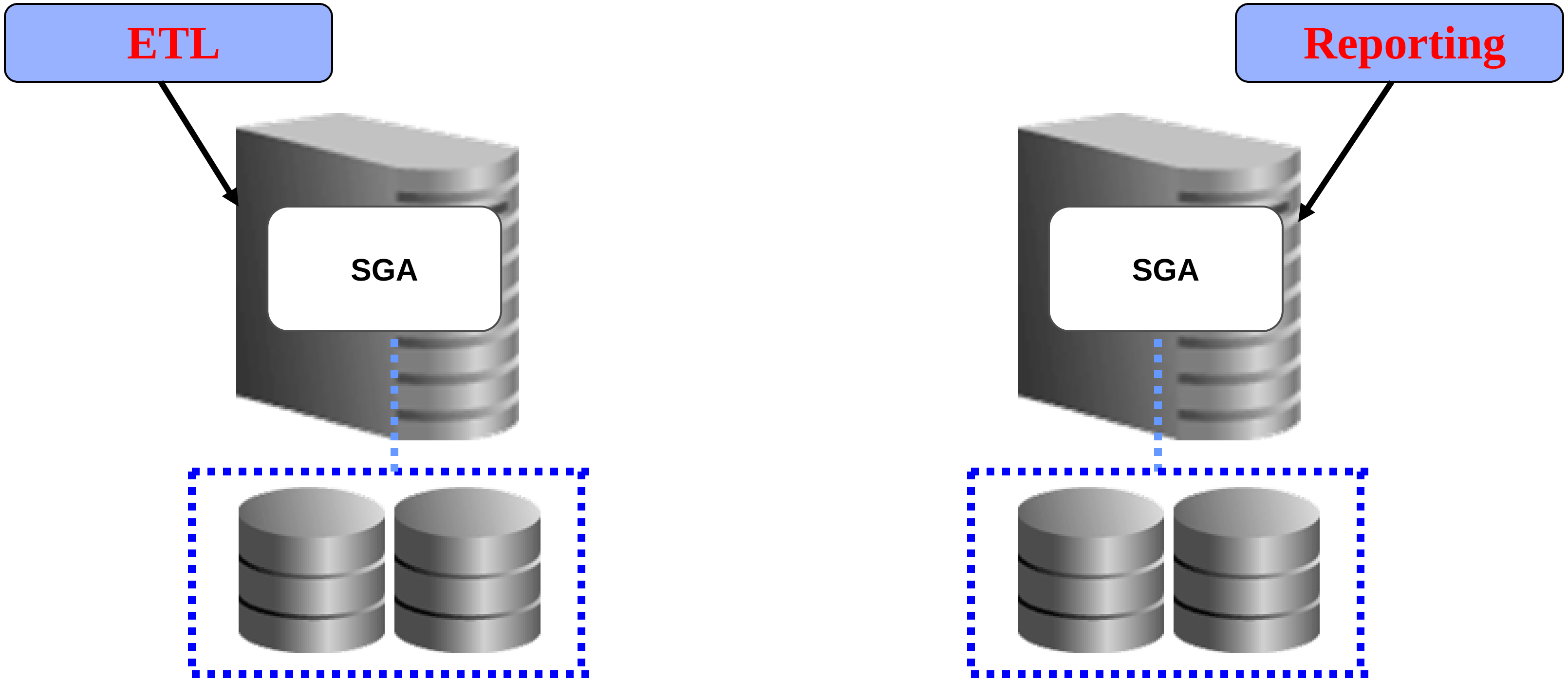
Real Application Clusters



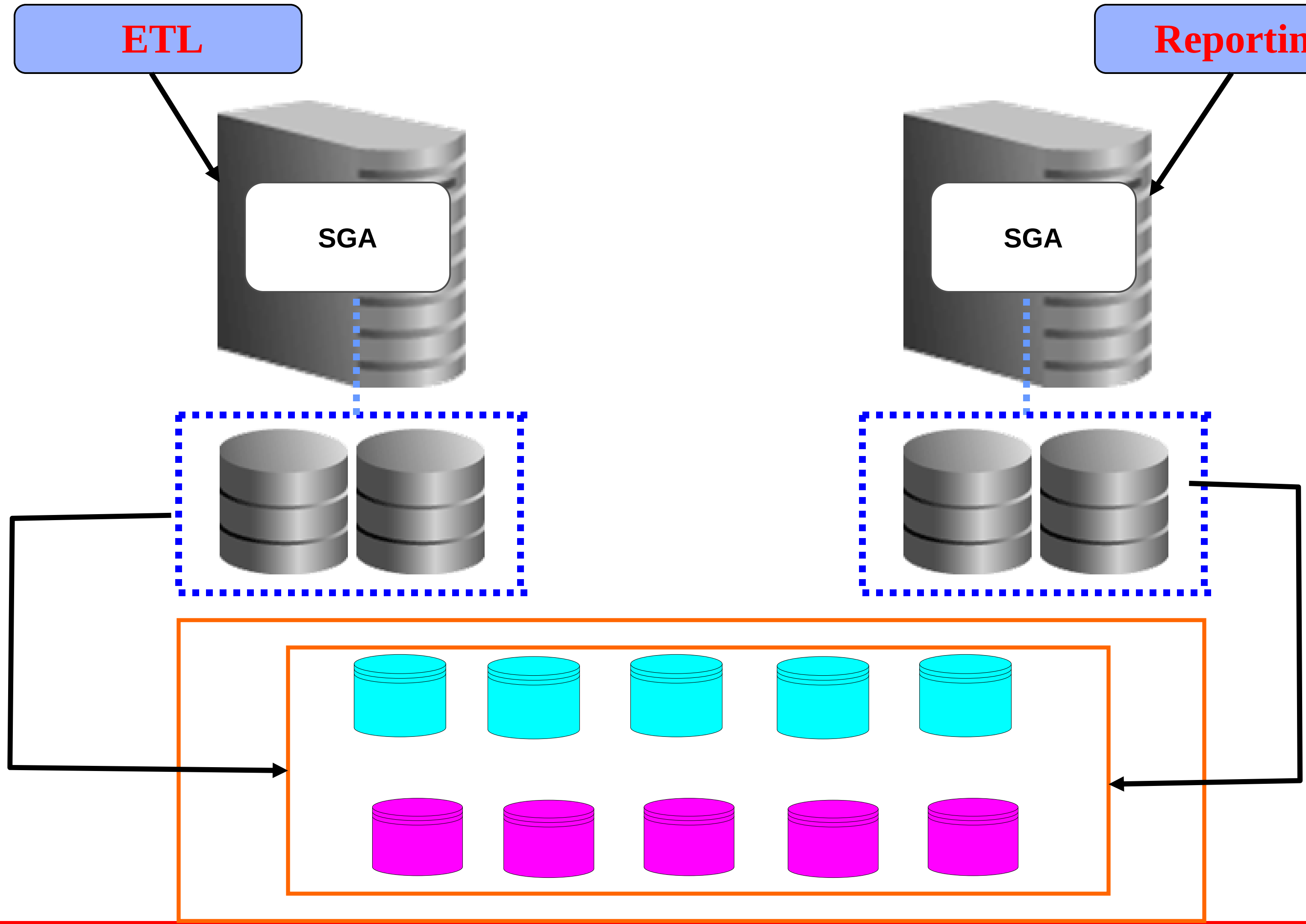
IO Issues with Real Application Clusters



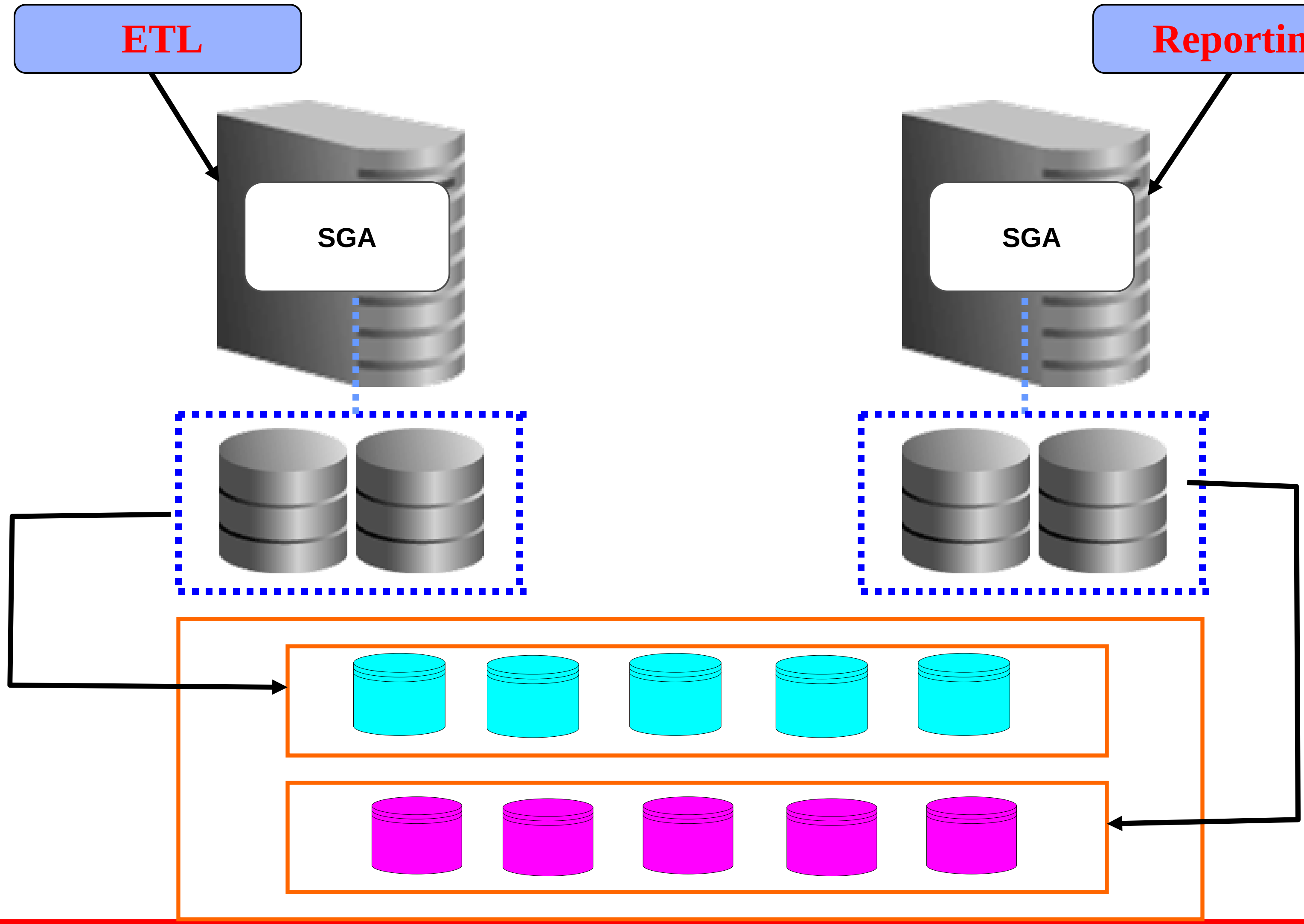
IO Issues



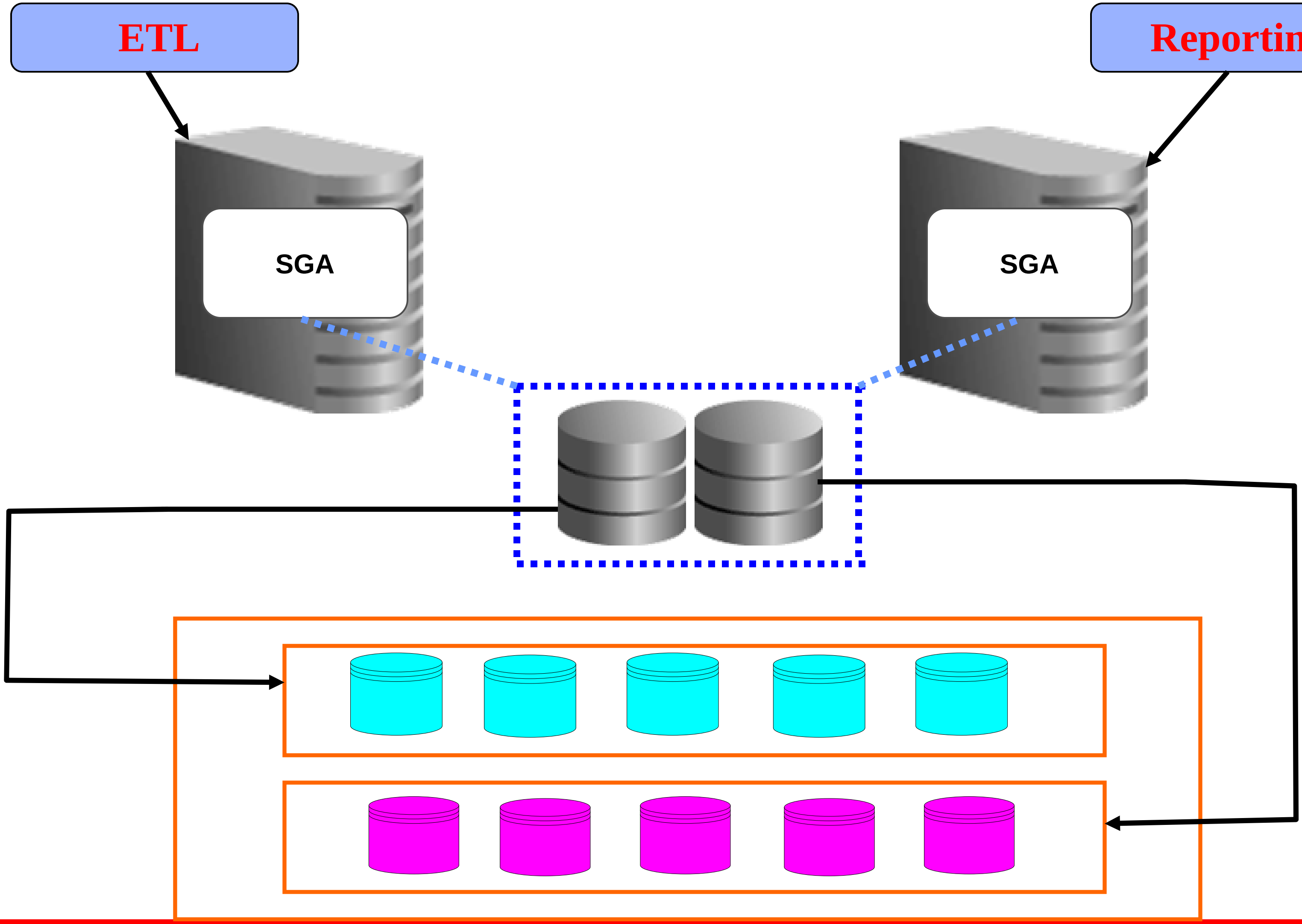
IO Issues



IO Issues



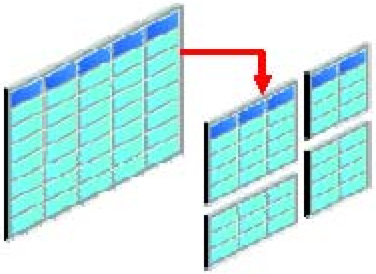
IO Issues



The Ideal Storage Configuration

- S.A.M.E.
  - Stripe And Mirror Everything
  - Optimize throughput across as many physical disks as possible – stripe across all devices
  - Exception: storage tiers
- Automatic Storage Manager (ASM)
  - Implements S.A.M.E. per disk group (mirroring optional)
  - Simplifies and automates database storage management
    - Automatic rebalancing
    - Separate disk groups for different storage tiers

Partitioning



- Range partition large fact tables typically on date column
  - Consider data loading frequency
    - Is an incremental load required?
    - How much data is involved, a day, a week, a month?
  - Partition pruning for queries
    - What range of data do the queries touch - a quarter, a year?
- Subpartition by hash to improve join performance between fact tables and / or dimension tables
  - Pick the common join column
  - If all dimension have different join columns use join column for the largest dimension or most common join in the queries

Partition Pruning

Q: What was the total sales for the weekend of May 20 - 22 2008?

```
Select sum(sales_amount)
From SALES
Where sales_date between
to_date('05/20/2008','MM/DD/YYYY')
And
to_date('05/23/2008','MM/DD/YYYY');
```

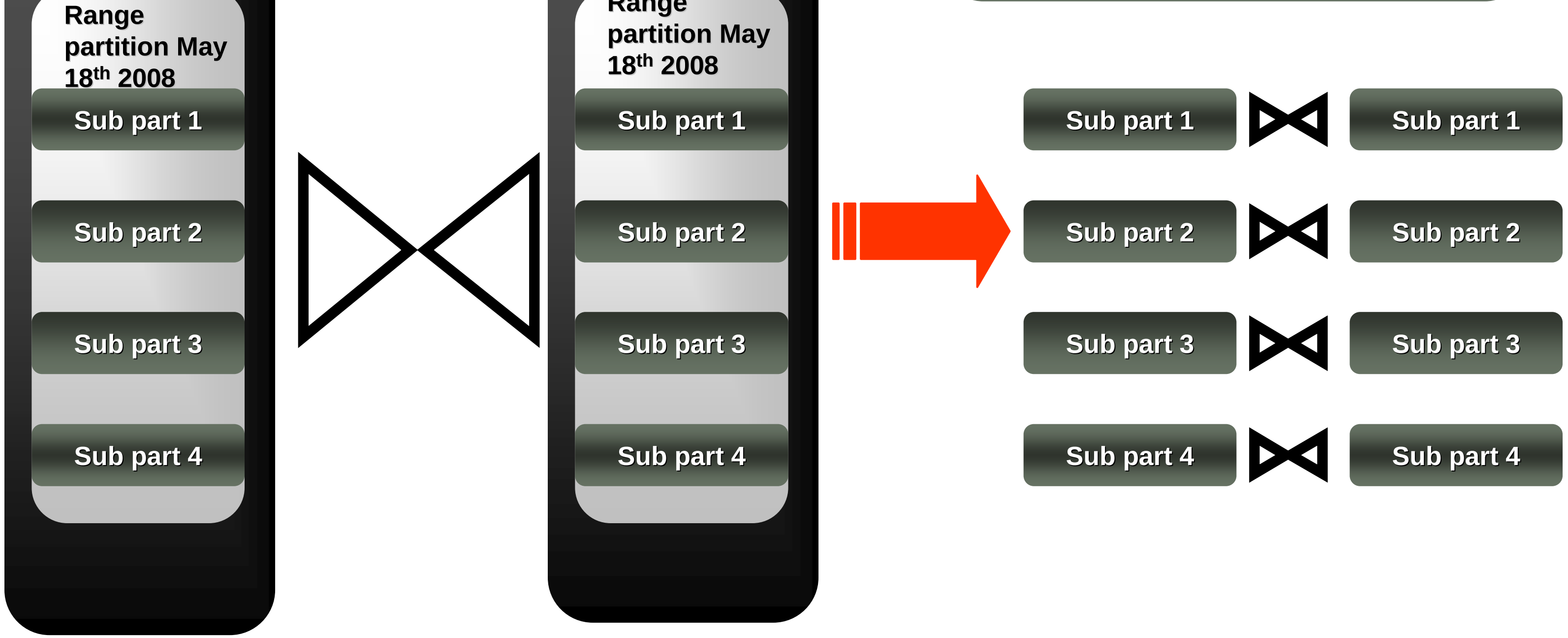
Sales Table

- May 18th 2008
- May 19th 2008
- May 20th 2008
- May 21st 2008
- May 22nd 2008
- May 23rd 2008
- May 24th 2008

Only the 3 relevant partitions are accessed

Partition Wise join

```
Select sum(sales_amount)
From
SALES s, CUSTOMER c
Where  s.cust_id = c.cust_id;
```



Both tables have the same degree of parallelism and are partitioned the same way on the join column (cust_id)

A large join is divided into multiple smaller joins, each joins a pair of partitions in parallel

Execution plan for partition-wise join

Partition Hash All above the join &

single PQ set indicate partition-wise join

ID	Operation	Name	Pstart	Pstop	TQ	PQ Distrib
0	SELECT STATEMENT					
1	PX COORDINATOR					
2	PX SEND QC (RANDOM)	:TQ10001			Q1,01	QC (RAND)
3	SORT GROUP BY				Q1,01	
4	PX RECEIVE				Q1,01	
5	PX SEND HASH	:TQ10000			Q1,00	HASH
6	SORT GROUP BY				Q1,00	
7	PX PARTITION HASH ALL		1	128	Q1,00	
8	HASH JOIN				Q1,00	
9	TABLE ACCESS FULL	Customers	1	128	Q1,00	
10	TABLE ACCESS FULL	Sales	1	128	Q1,00	

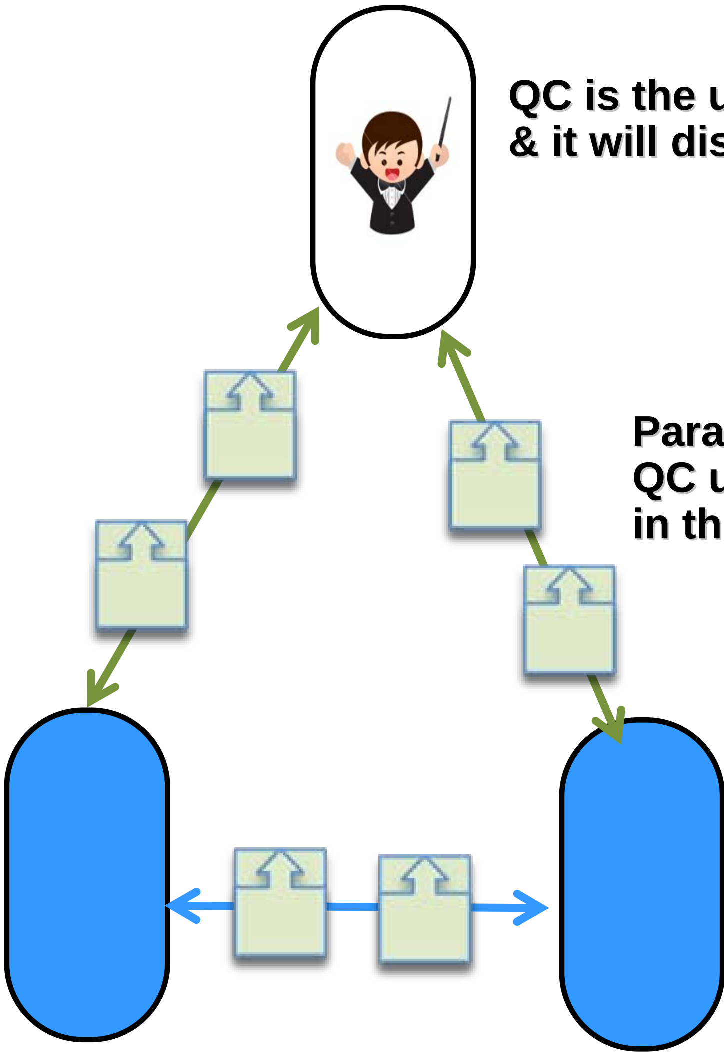
Database Configuration

- Parameters
  - db_block_size = 4k, 32 k
  - SGA, PGA
  - Increase parallelism with caution ! (concurrency vs. # of CPU)

Concurrency

- Issues
  - OLTP vs. OLAP
  - Latch Waits in the Shared Pool
  - Total Application Users
  - Concurrent users
  - Concurrent active query
- Approaches
  - Real Application Clusters
  - Parallelism

SQL Parallel Execution



QC is the user session that initiates the parallel SQL statement & it will distribute the work to parallel servers

Parallel servers communicate among themselves & the QC using messages that are passed via memory buffers in the shared pool

Parallel servers - individual sessions that perform work in parallel They are allocated from a pool of globally available parallel server processes and assigned to a given operation

Messages ⟷ Parallel server connection ⟷ QC connection

Server Configuration

- Approaches - A Balanced Architecture
    - # CPUs
    - # GB of RAM
    - # HBAs
    - Size of Swap Space
    - Kernel Parameters

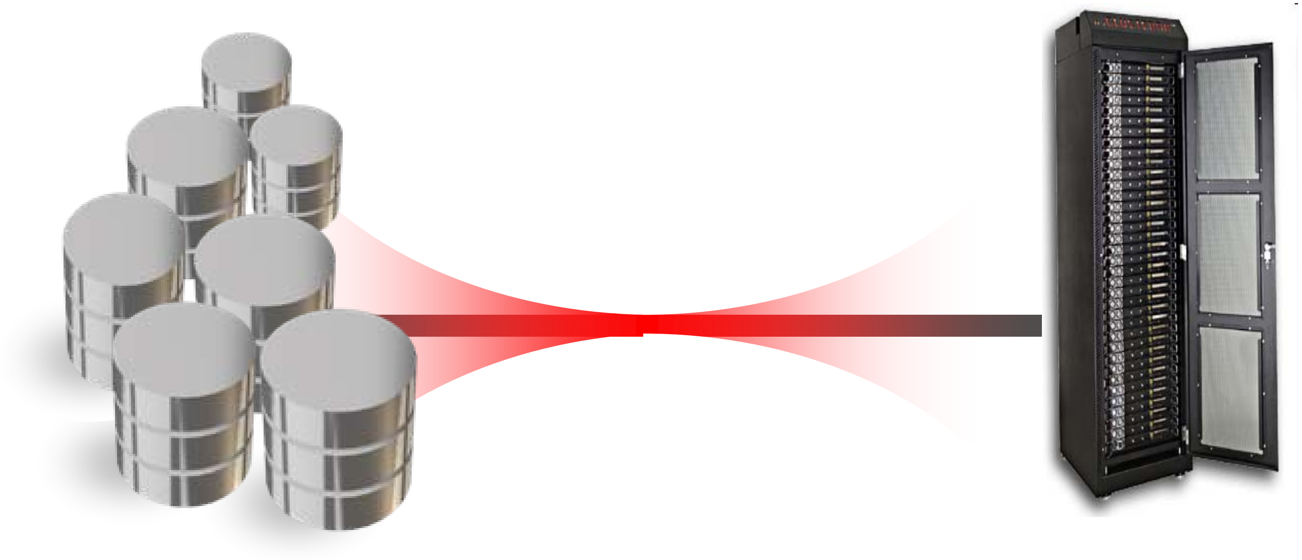
Storage Configuration

- Issues
  - Inconsistent I/O throughputs
  - RAID Configuration
  - Different type of disks (rpm)
- Recommendations
  - RAID Configuration
  - SAME & Zoning
  - ASM

Exadata

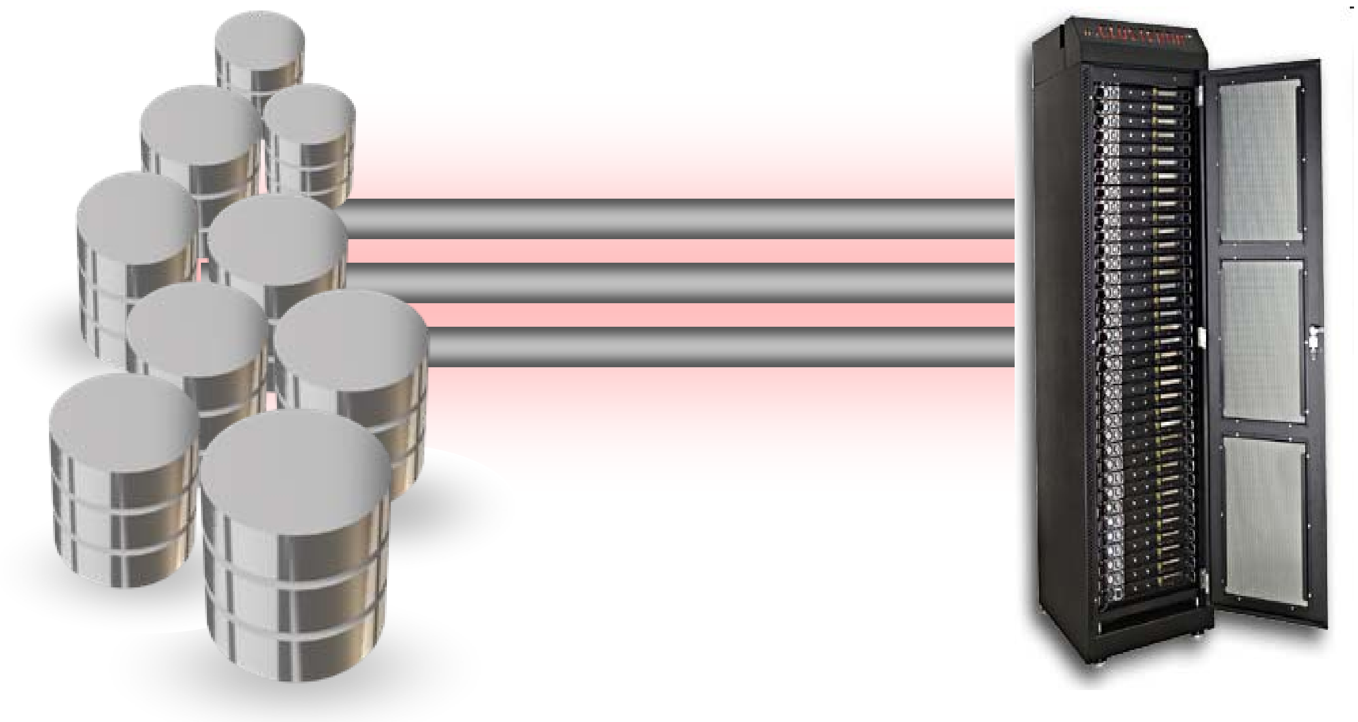
- Extreme Performance
  - **10X to 100X** speedup for data warehousing
- Database Aware Storage
  - Smart Scans
- Massively Parallel Architecture
  - Dynamically Scalable
  - Unlimited Linear Scaling of Data Bandwidth
  - Transaction/Job level Quality of Service
- Mission Critical Availability and Protection
  - Disaster recovery, backup, point-in-time recovery, data validation, encryption

The Performance Challenge
Storage Data Bandwidth Bottleneck



- Current warehouse deployments often have bottlenecks limiting the movement of data from disks to servers
  - Storage Array internal bottlenecks on processors and Fibre Channel Loops
  - Limited Fibre Channel host bus adapters in servers
  - Under configured and complex SANs
- Pipes between disks and servers are 10x to 100x too slow for data size

Solutions To Data Bandwidth Bottleneck



- Add more pipes – Massively parallel architecture
- Make the pipes wider – 5X faster than conventional storage
- Ship less data through the pipes – Process data in storage

HP Oracle Database Machine

Pre-Configured High Performance Data Warehouse

- 8 DL360 Oracle Database servers
  - 2 quad-core Intel Xeon, 32GB RAM
  - Oracle Enterprise Linux
  - Oracle RAC
- 14 Exadata Storage Cells (SAS or SATA)
  - Up to 21 TB uncompressed user data (SAS)
  - Up to 46 TB uncompressed user data (SATA)
- 4 InfiniBand switches
- 1 Gigabit Ethernet switch
- Keyboard, Video, Mouse (KVM) hardware
- Hardware Warranty
  - 3 YR Parts/3 YR Labor/3 YR On-site
  - 24X7, 4 Hour response time

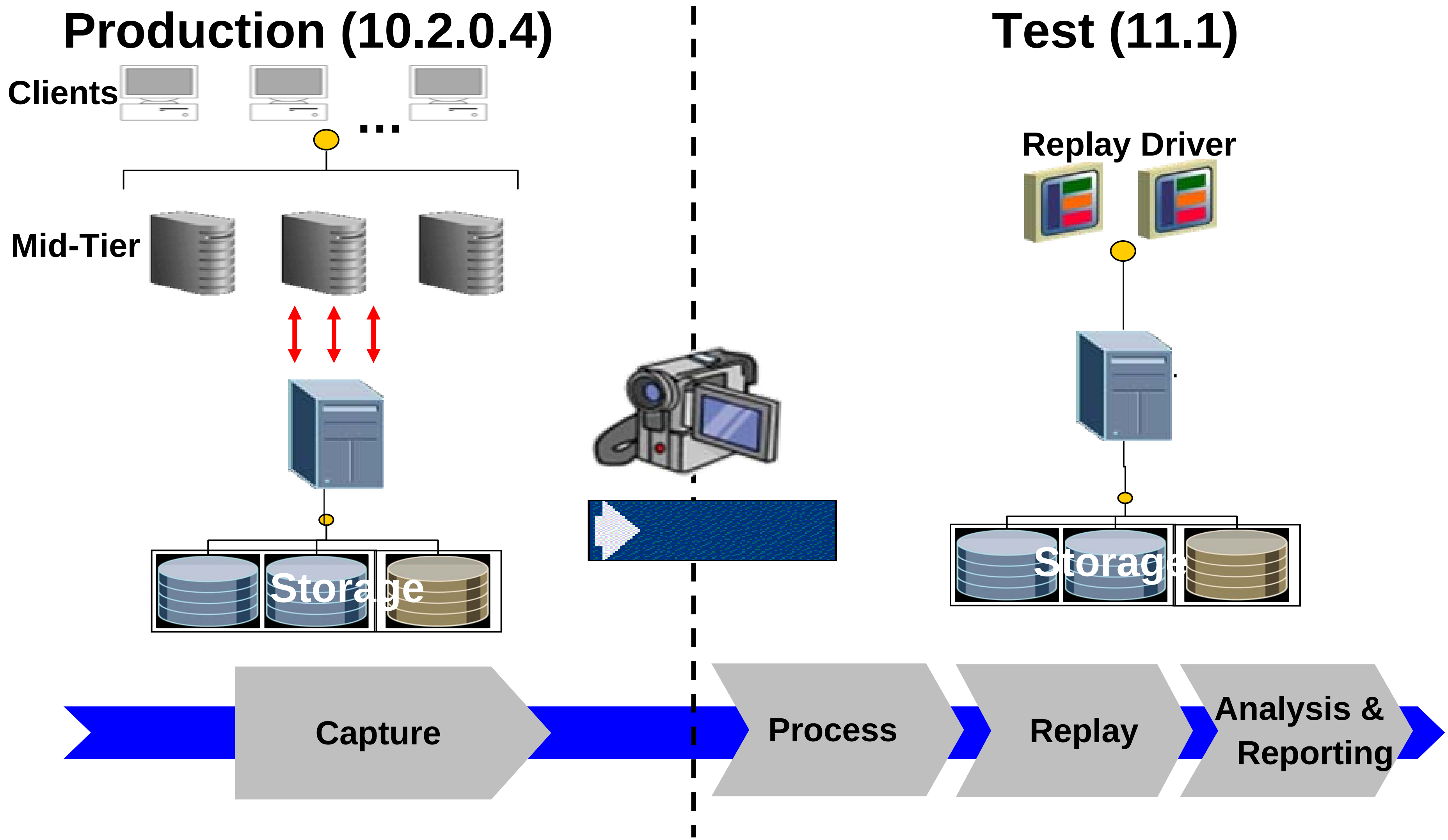
Add more racks for unlimited scalability

ORACLE

QA Test

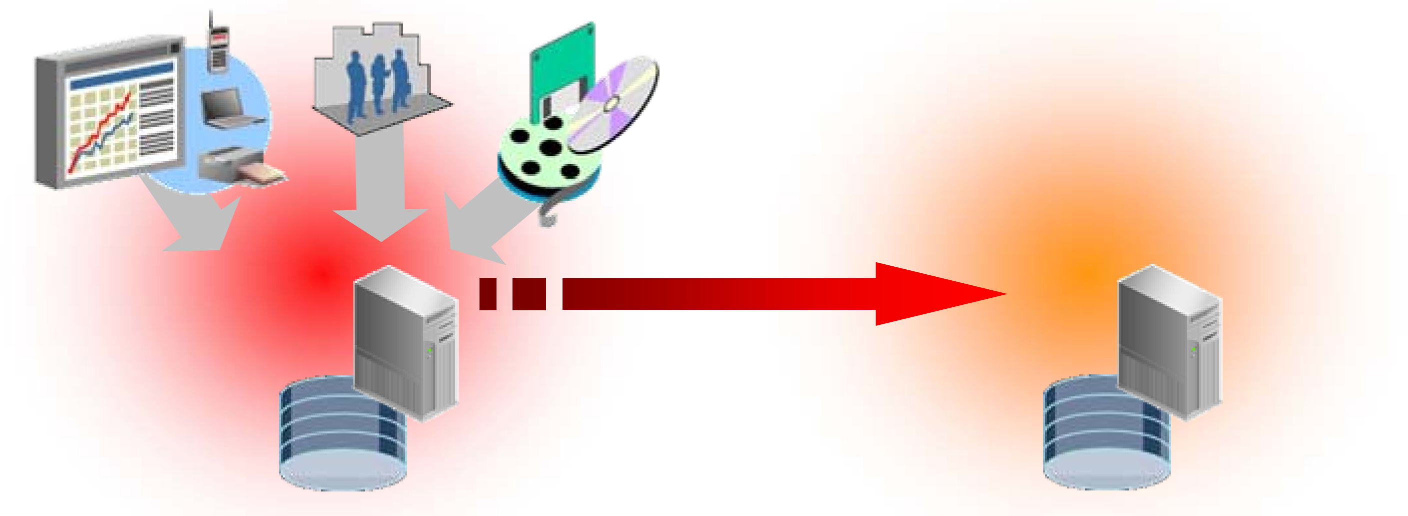
- Issues
  - No production-like QA environment
  - No standard QA tools
  - No synchronized end-to-end monitoring process
- Approaches
  - Need a QA environment closer to Production
  - Real Application Testing
  - Active Data Guard
  - Enterprise Manager

Database Replay Workflow



Traditional Physical Standby Databases

Investment in Disaster Recovery only



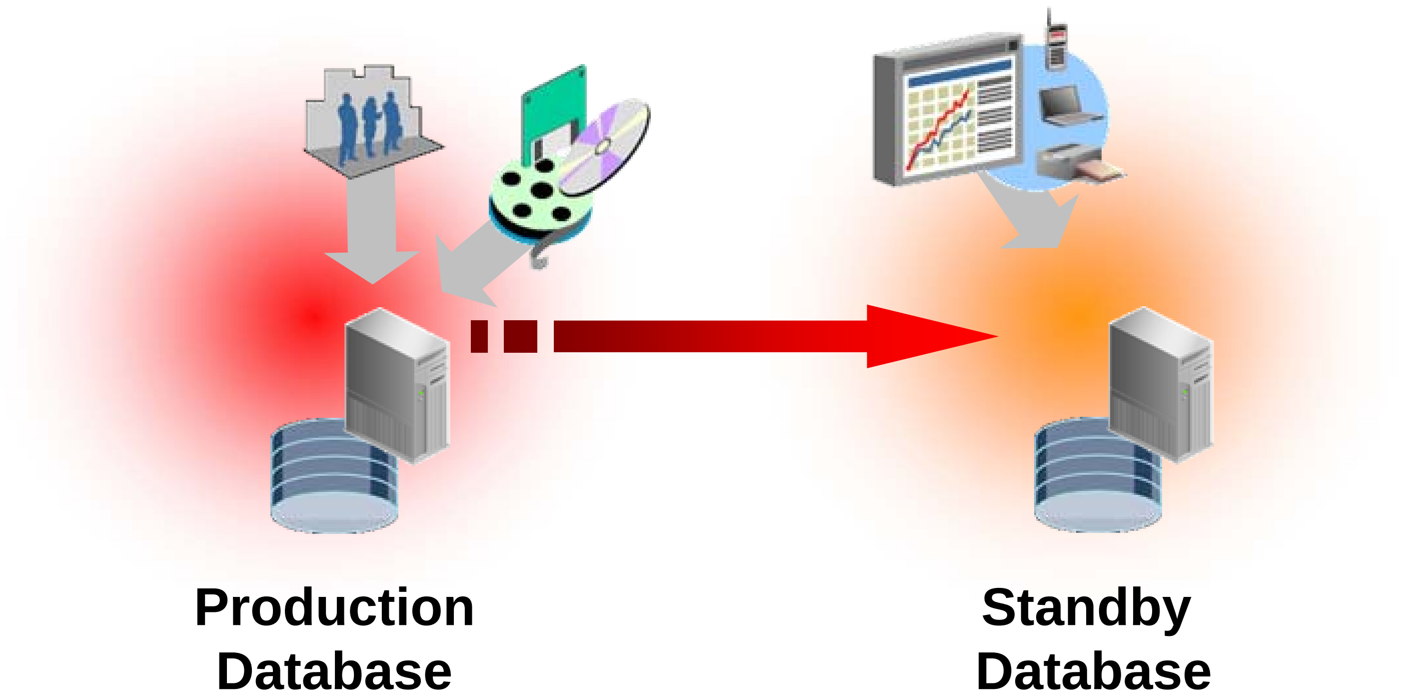
Production Database

Standby Database

- Applications, backups, reports run on production only

With Oracle Active Data Guard

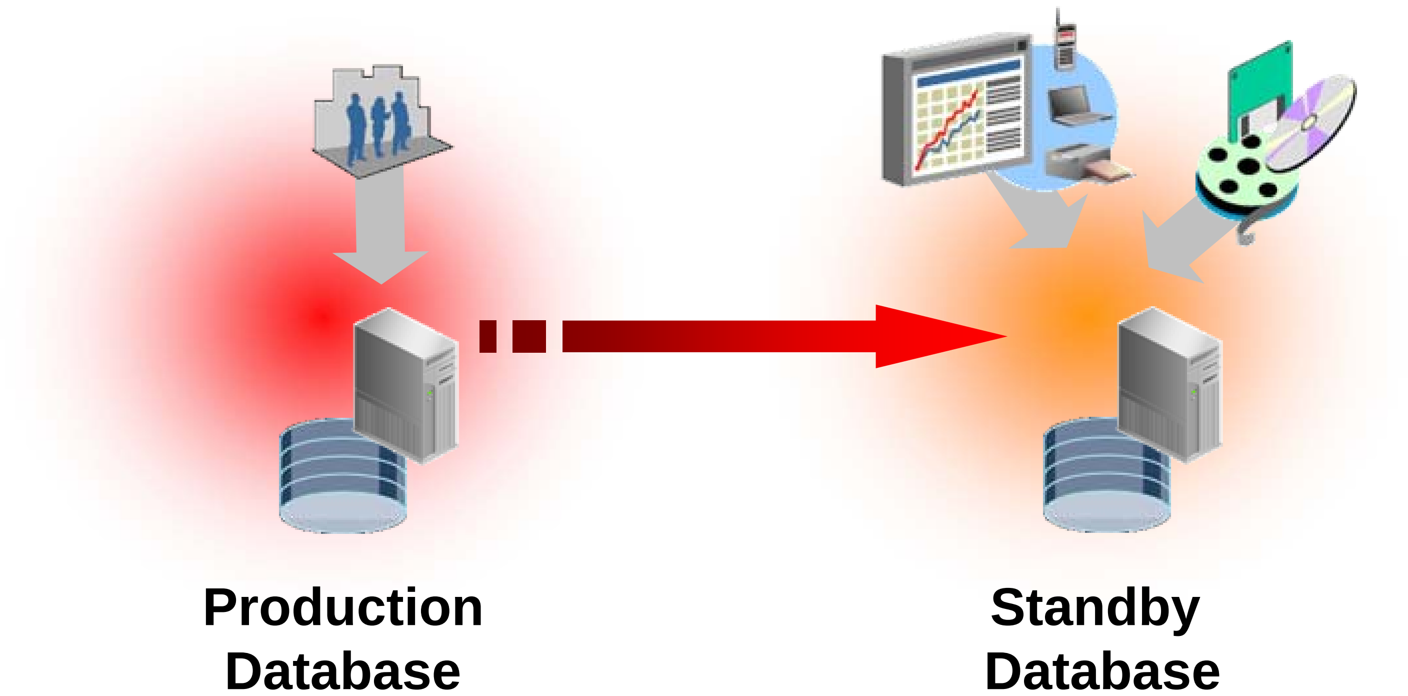
Offload production reporting to standby



- Simultaneously available in read <u>and</u> recovery mode

With Oracle Active Data Guard

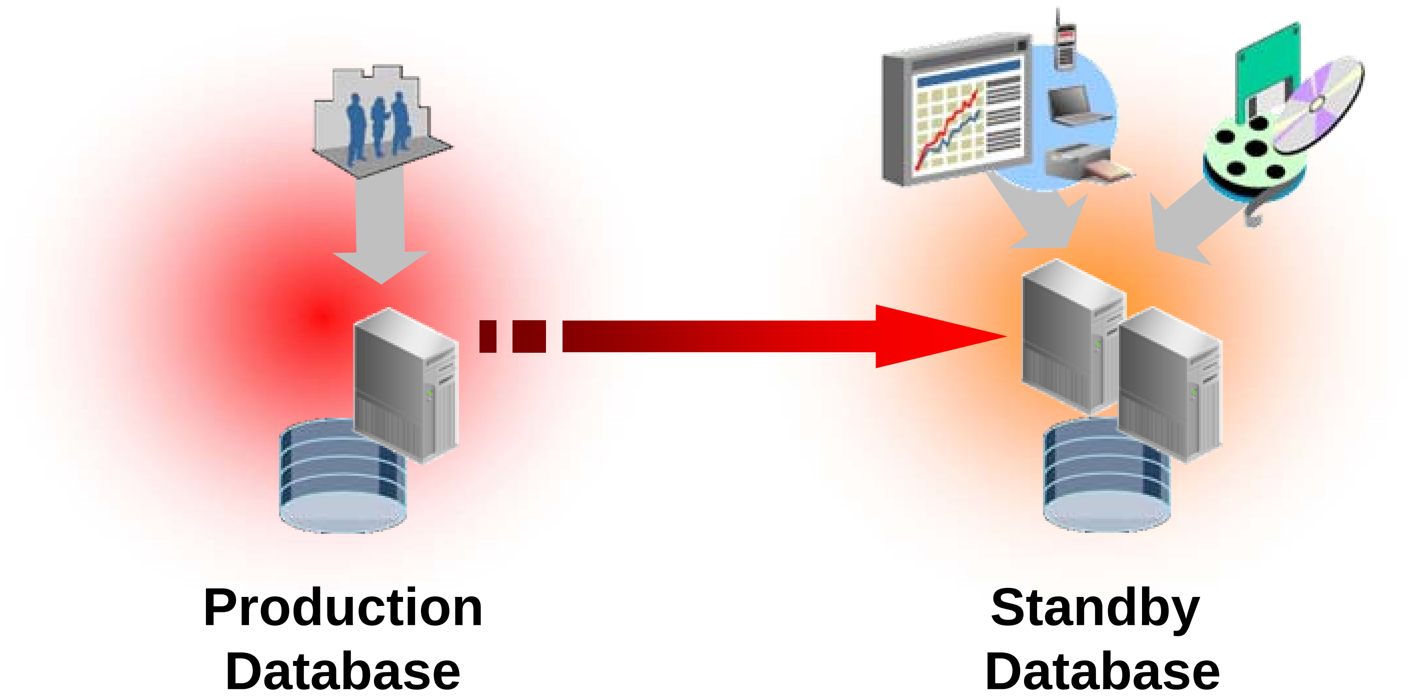
Offload fast incremental backups to standby



- Use RMAN block change tracking on standby database
- Fast incremental backups complete 20x faster

Snapshot Standby in Data Guard 11g

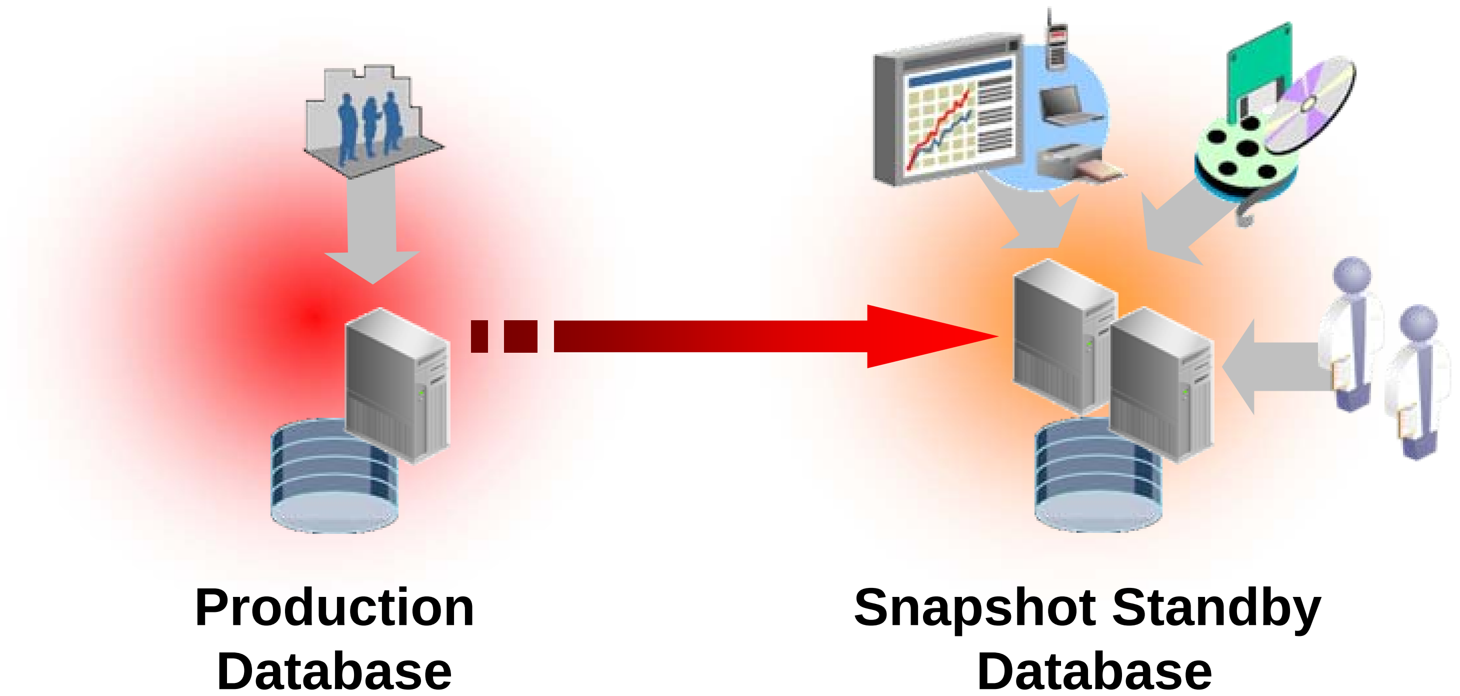
Use standby database for pre-production testing



- Convert to ‘snapshot’ standby for testing purposes
- Part of Data Guard 11*g*, no additional license required

With Oracle Snapshot Standby

Test changes – then resync standby with production



- Test on standby until guaranteed production-ready

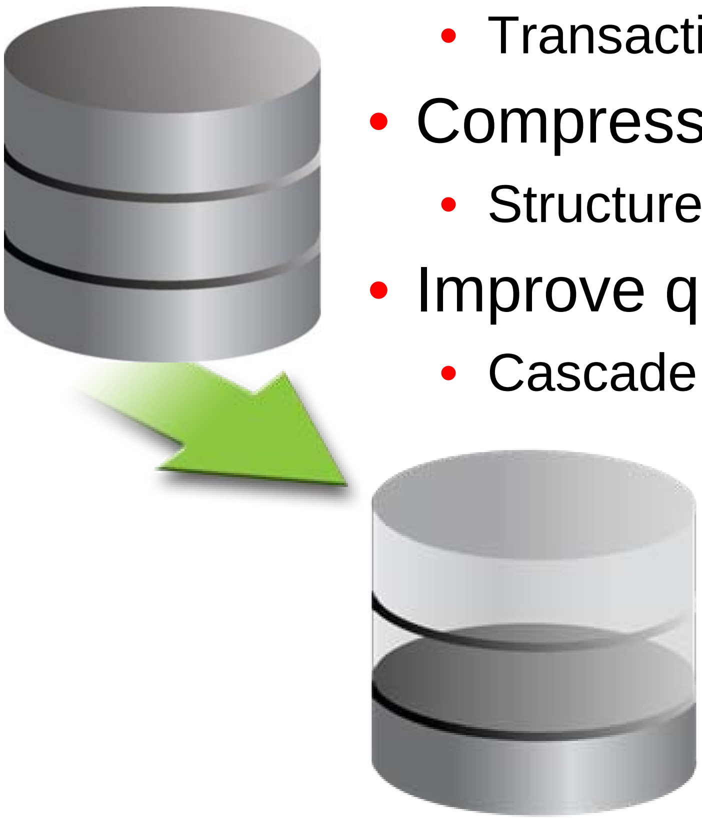
Others

- Approaches
  - Advanced Compression (11g only)
  - Backup strategy
  - Oracle Database 11g

Significantly Reduce Storage Costs

Advanced OLTP Compression

- Compress large application tables
  - Transaction processing, data warehousing
- Compress all data types
  - Structured and unstructured data types
- Improve query performance
  - Cascade storage savings throughout data center

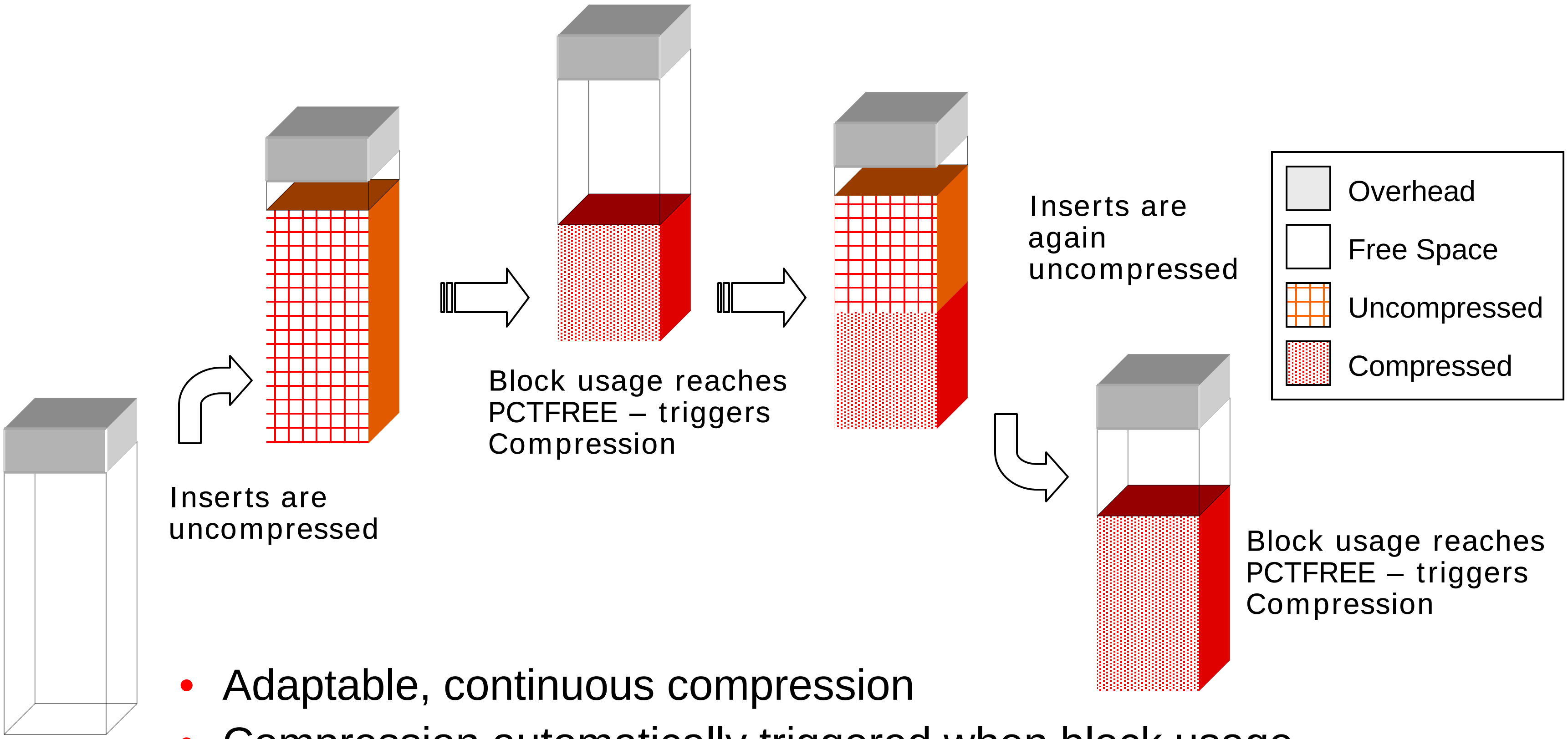


Up To

4X

Compression

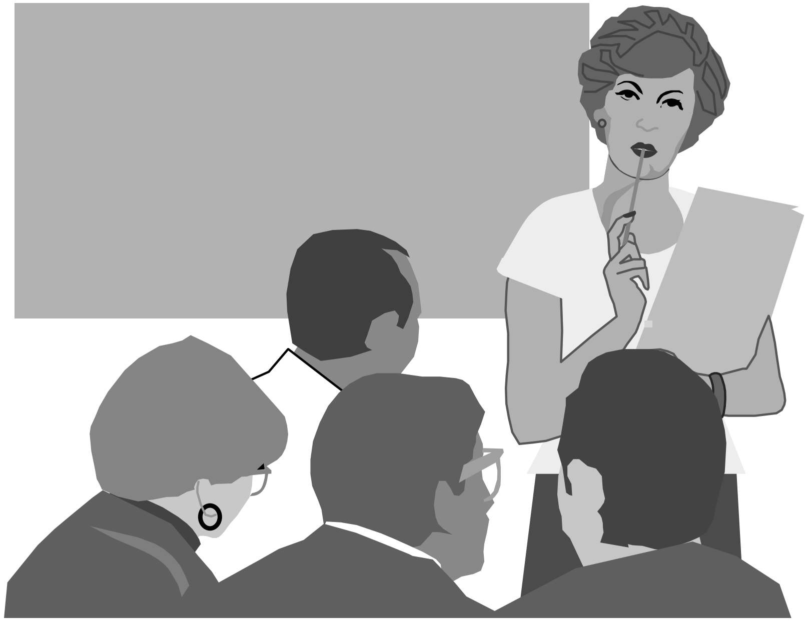
OLTP Table Compression



- Adaptable, continuous compression
- Compression automatically triggered when block usage reaches PCTFREE
- Compression eliminates holes created due to deletions and maximizes contiguous free space in block

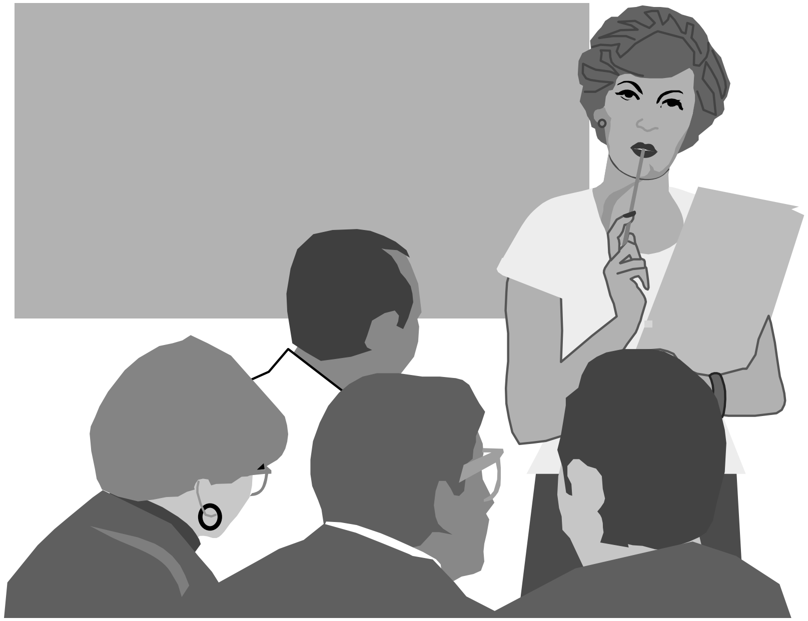
Summary

- Performance Tuning Approach
- Indexing
- Aggregation
- ETL Loading
- Database Configuration
- Concurrency



Summary

- Server Configuration
- Storage Configuration
- Exadata
- Monitoring and Testing
- Others



Q & A

ORACLE®

Daniel Liu Contact Information

Phone: (714) 376-8416

Email: daniel.liu@oracle.com

Email: daniel_t_liu@yahoo.com

Company Web Site:

http://www.oracle.com