# The Limits of Open Source in Extreme-Scale Storage Systems Design

## An MSST Panel on Open Source in Large System Design

1. Peter Braam, Campaign Storage
2. Sean Roberts, OpenStack Consortium
3. Matthew O'Keefe, Oracle
4. David Bonnie, Los Alamos National Laboratory

ORACLE

# Engineering at Cloud Scale

**Matthew O'Keefe, Ph.D.**
**Vice President, Cloud Infrastructure**

**Slides stolen from:**

**Kothanda (Kodi) Umamageswaran**
**Vice President, Exadata Development**

ExadataPM


ORACLE

## Changes Everything

Unprecedented:

- Simplicity
- Elasticity
- CapEx Savings
- Time to Market

**Enterprise Vendors MUST Have a Successful Public Cloud Solution**

ORACLE

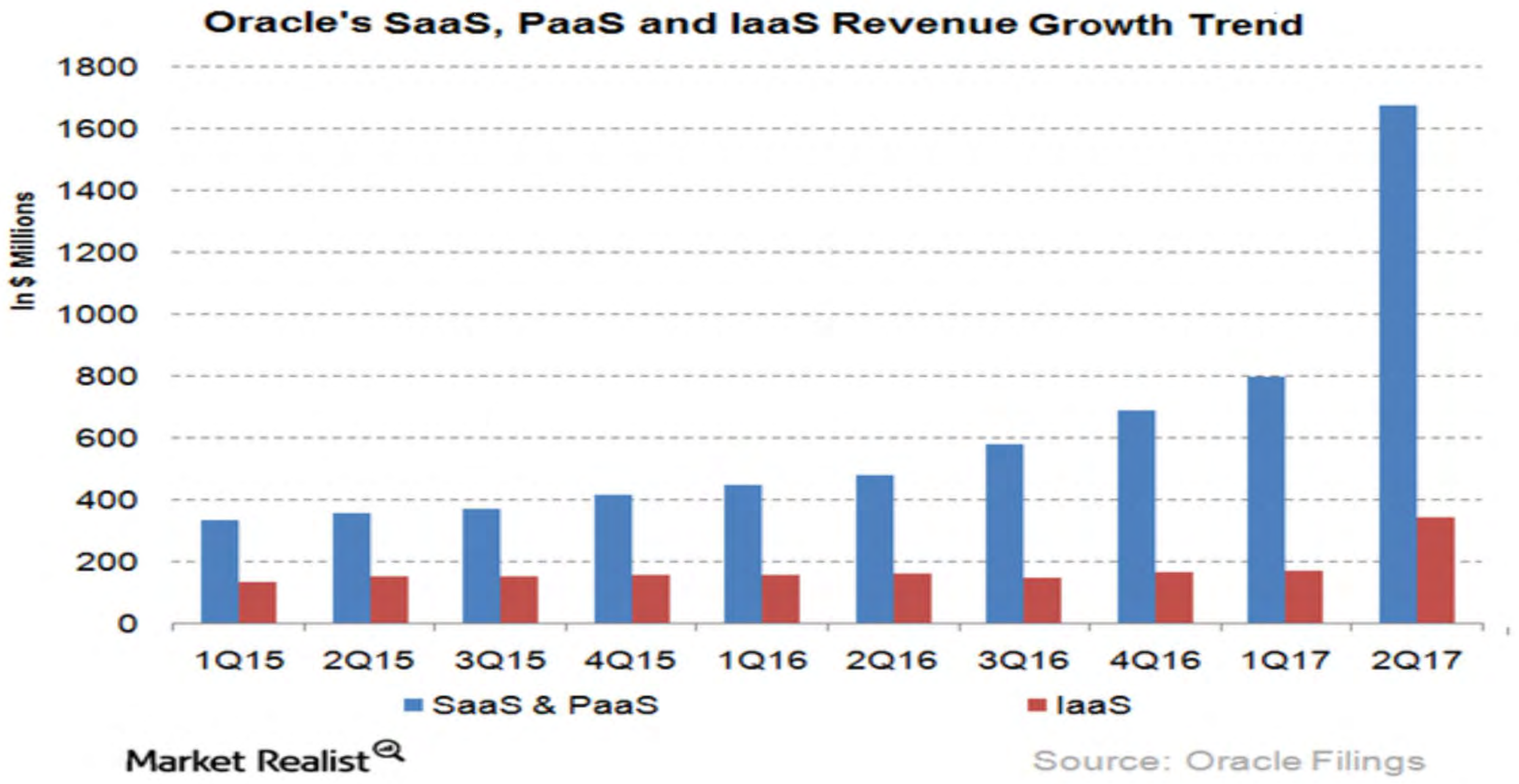
# Oracle Cloud: Summary

**30,000+** Devices
**400 PB+** Storage

**19** Data Centers
**2 new in Germany**

**62 Million+** Users/Day
**23 Billion+** Transactions/Day

ORACLE


Oracle's SaaS, PaaS and IaaS Revenue Growth Trend
1800
1600
1400
1200
1000
800
600
400
200
0
In $ Millions
1Q15
2Q15
3Q15
4Q15
1Q16
2Q16
3Q16
4Q16
1Q17
2Q17
SaaS & PaaS
IaaS
Market Realist
Source: Oracle Filings

# Rewriting Decades of Applications to Move to Cloud is not an Option

ORACLE

# Compromising Quality of Service to Move to Cloud is not an Option

ORACLE

# Compromising High Availability to Move to Cloud is not an Option

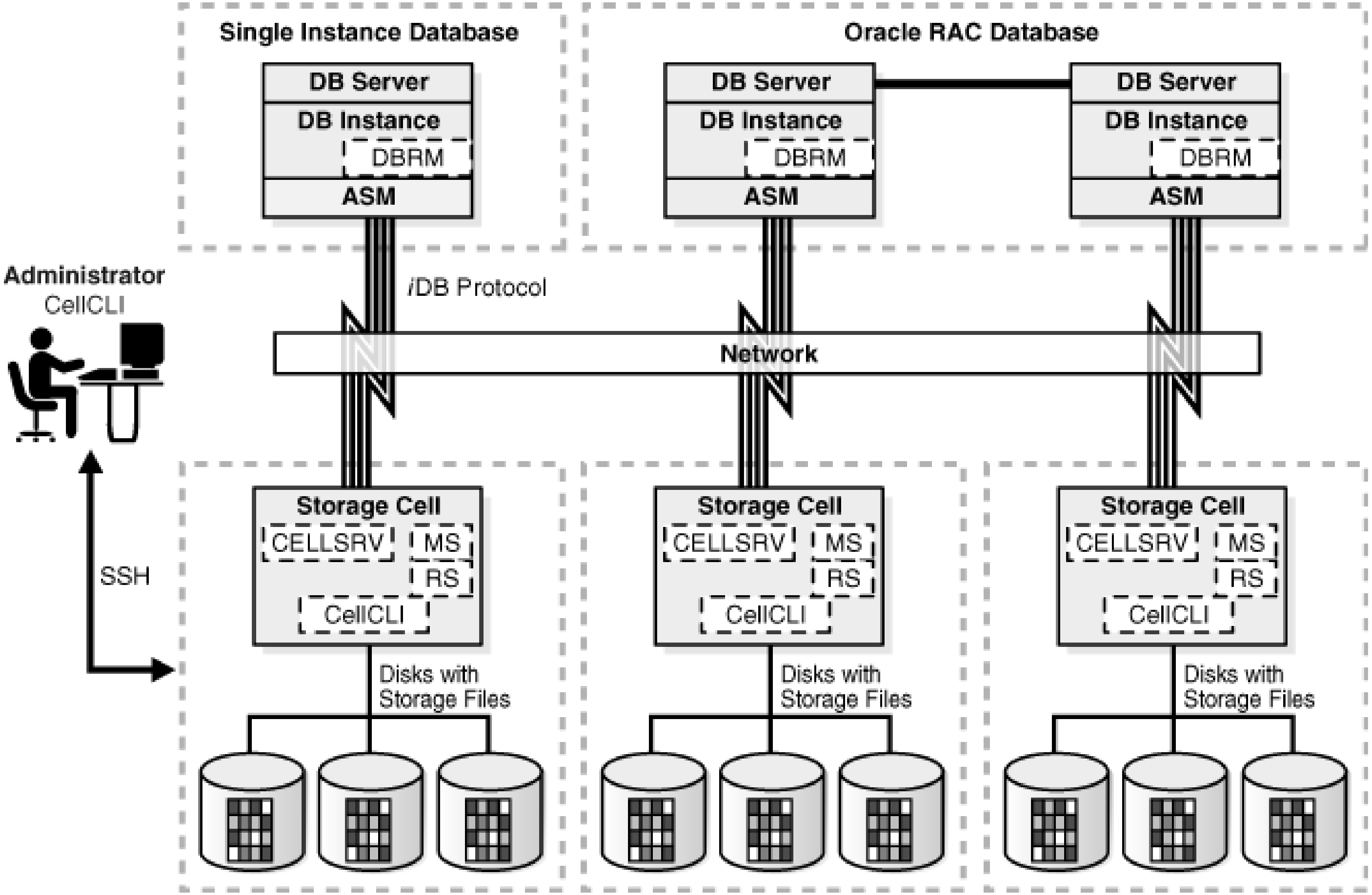
# Compromising Security to Move to Cloud is not an Option

ORACLE

# Cloud Infrastructure has to seamlessly scale as per business needs

ORACLE

# Oracle Exadata Architecture



# Ideal Database Cloud Platform

## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. Maintain Quality of Service for all Tenants
4. Fully Secure

# Previous Cloud Database Services are Severely Flawed

## Need Best of On-Premises with Best of Cloud

- **New cloud databases** are primitive – 30 years behind state-of-the-art
  - **Primitive or non-existent: SQL, transactions, analytics, functionality, security, standards, mixed workloads**
- **Mature databases** deployed in the cloud – 10 years behind state-of-the-art
  - **Non-existent: enterprise-class storage, high performance fabric, PCI flash, transparent OLTP scale-out**
- There is a **giant chasm** between on-premises and cloud
  - Applications must be changed, APIs are different, licensing is different, QoS is compromised

Amazon DynamoDB

Amazon Redshift

Amazon RDS

IBM Cloudant®

dashDB

SQL Database

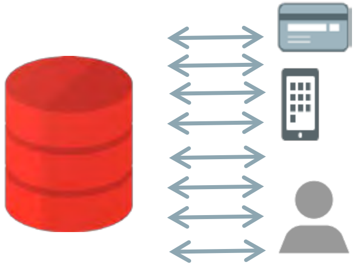
ORACLE

# Ideal Database Cloud Platform

## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. Maintain Quality of Service for all Tenants
4. Fully Secure

# Database Workloads in Cloud



- Online Transactional Processing Workloads
  – Lots of random small IOs
  – Low latency, predictable response times

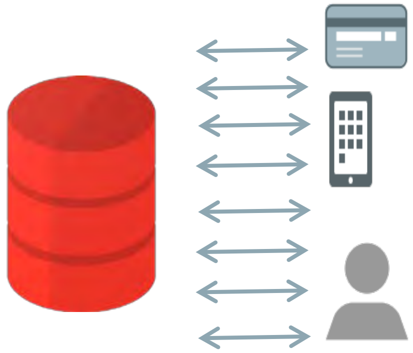
- Data Warehouse, Analytics Workloads
  – Lots of data to load and process
  – Succinct results in real time


- Test/Dev Workloads
  – Quick provisioning, simple management
  – Similar performance characteristics as production

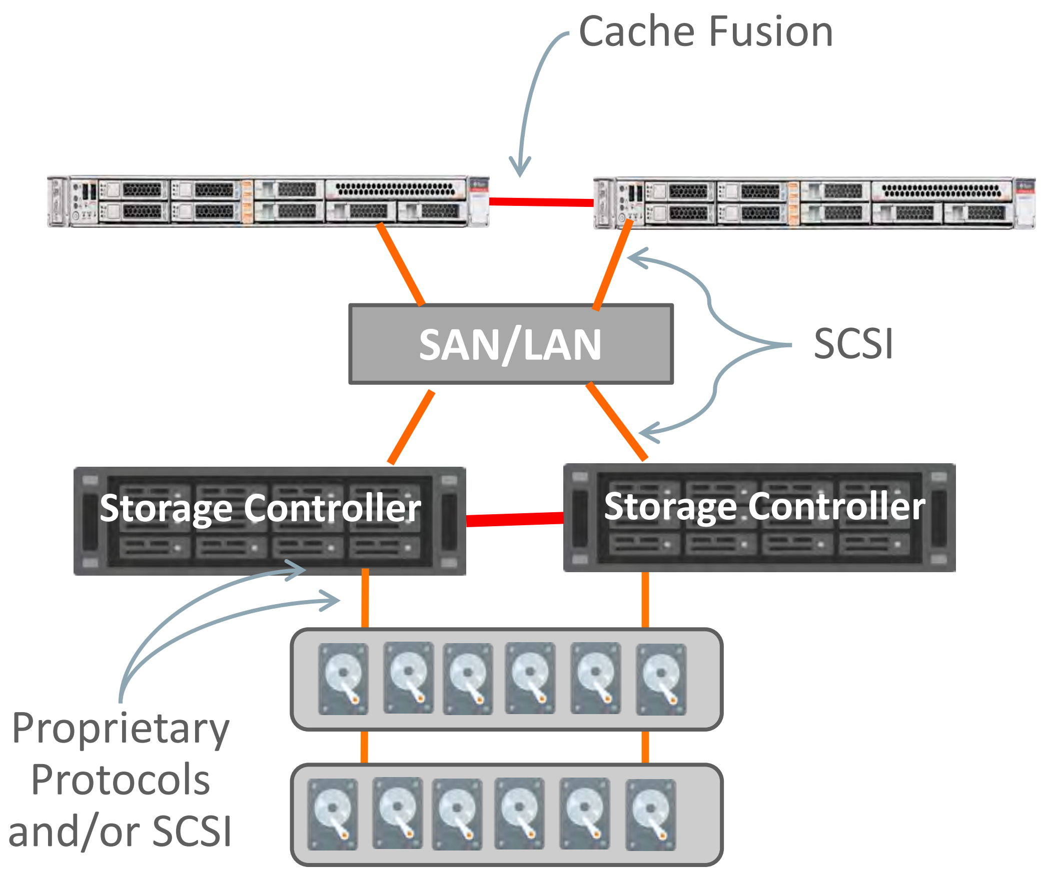
 

# Characteristics of an OLTP System

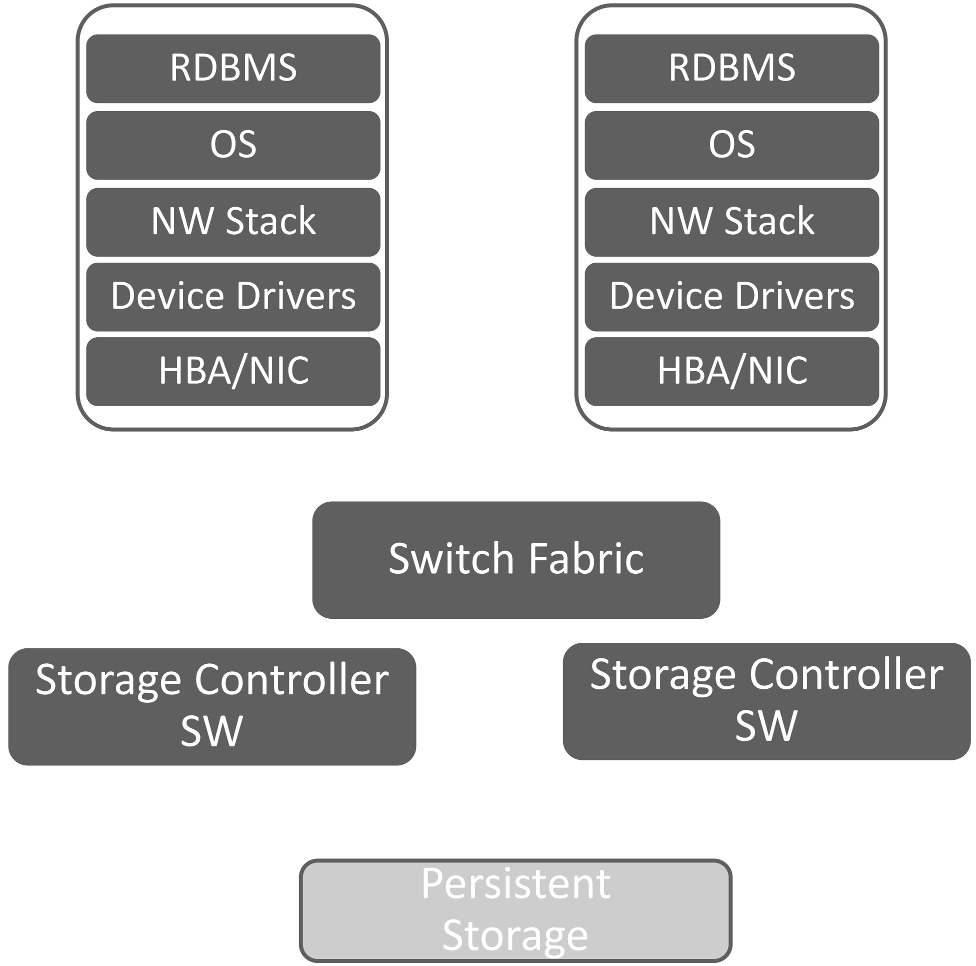
- Many concurrent users
- Lots of messages between database servers
- Lots small updates/deletes issued to the database
- Consistent response times are very critical
- Constant uptime is essential



# Traditional OLTP IO Path



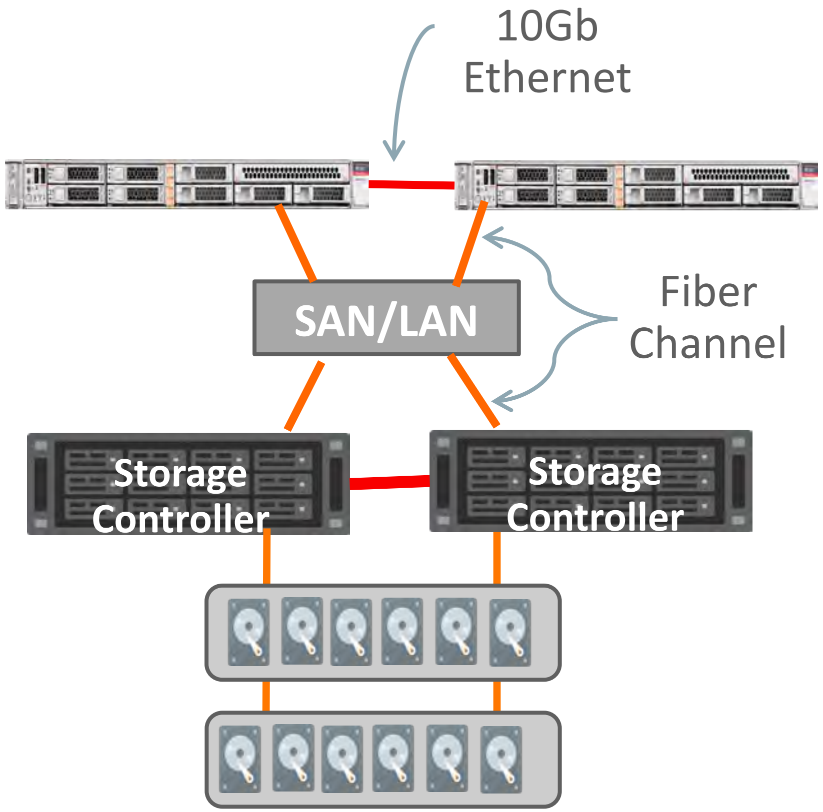

Hardware View




Software View

# Challenge: Traditional Networking Infrastructure Delays OLTP

- Database to Database server communication is on slow 10Gbps Ethernet
  - Ethernet though vastly popular has high latency and low throughput compared to other networking technologies
  - InfiniBand has 40Gb/s networks today and 100Gb/s networks (emerging) offering lower latency, zero copy, and higher throughput
- Database to storage communication, critical to RAC, is traditionally over Fiber Channel
  - Shared storage unable to take advantage of higher bandwidth, lower latency networks
  - Number of HBAs on the database server limits performance
  - InfiniBand enables lower latency and higher IOPS for OLTP



# Challenge: Traditional Networking Stacks Delay Cache Fusion Messages

- OLTP messages are small and relatively simple, so they require little time to transfer over the network and execute on the destination
- Most of the processing time for OLTP messages is due to the CPU and OS overhead of traversing the complex multi-layer network protocol stack
  - Both on the source and destination
  - Cache misses, Context switches, TLB flushes, Interrupt overhead, kernel buffer to user buffer copy

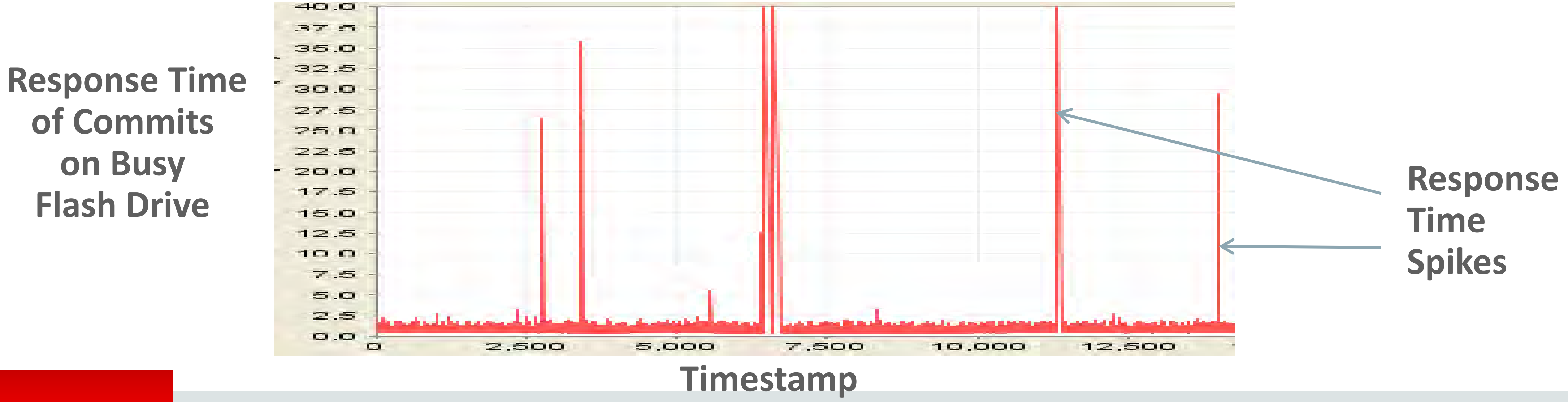

ORACLE

# Cloud: Enable low latency networking for OLTP

- InfiniBand hardware enables a low latency, high bandwidth infrastructure
- Cloud deployment hides InfiniBand network management from tenant
- Exafusion is a light-weight protocol custom designed for critical OLTP messages that bypasses the Operating System/networking stack
- InfiniBand hardware primitives called directly from oracle software
- Gets rid of huge amounts of networing stack overhead improving latency and CPU utilization

# Challenge: Achieving Fast OLTP Despite Flash Inconsistency

- Flash drives are great for OLTP because flash has very high I/O rates
- But Flash has occasional spikes in response time
  - Caused by complex internal algorithms: wear leveling (like garbage collection), erasing media, etc.
- Flash arrays try to hide by adding a DRAM cache with a UPS backup
  - Doesn't hide spikes when there is a surge in IO load
- I/O completion time outliers create a major problem for OLTP log writes
  - One slow log write stalls lots of foregrounds
  - Small changes in log write response time lead to large slowdowns in commit latency and OLTP throughput

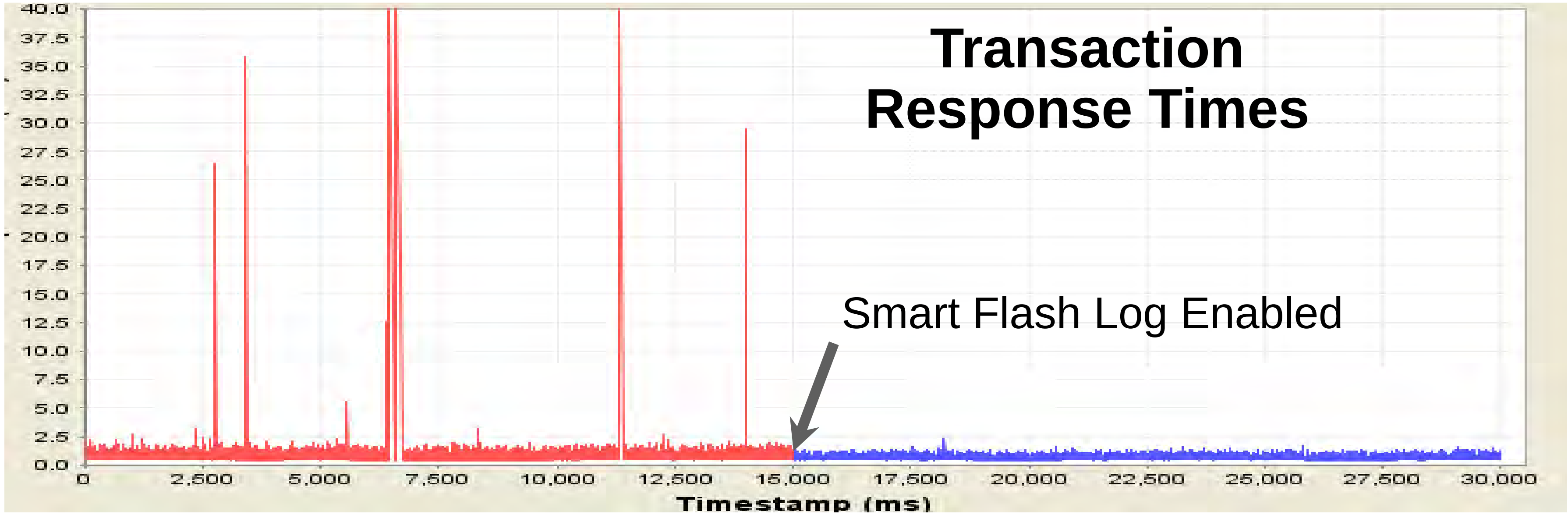


# Cloud: Writing Logs to 2 Devices in Parallel Eliminates Spikes

***Smart Flash Logging***

**Default (on left)**
- Choppy Response
- High Outliers

**Smart Flash Logging**
- 3x faster response
- Much lower outliers




- Smart Flash Logging transparently issues log writes to flash and a second device in parallel - log write completes when first of two writes finishes
  - Opposite of mirroring where both writes must complete
- If flash drive fails, commits recovered using DB log file specific algorithms
- Cloud deployment enables this transparently for all tenants

# Challenge: flaky devices slow down users

- Flash drives and hard drives don't fail instantly
- They can become sick
  - increase response times of I/Os
  - increase number of I/O errors
- I/Os potentially slow down 10x
  - One slow horse can slow down the horse cart
- User's queries and transactions start piling up and getting delayed
- Large backlog created takes long time to clear

ORACLE

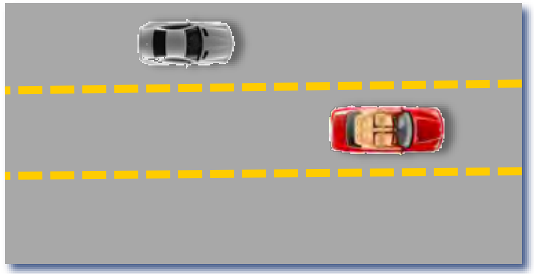
# Cloud: I/O latency capping

- Read I/O from the mirror copy of the data when read from primary copy of the data is slow
- Writes buffered in storage on a different media when the target device is slow
  - Resolve consistencies after testing the device or remirror by failing the device
- Isolate flaky devices and fail them using storage software even though the hardware is still functional
- Provide consistent response times for OLTP customers

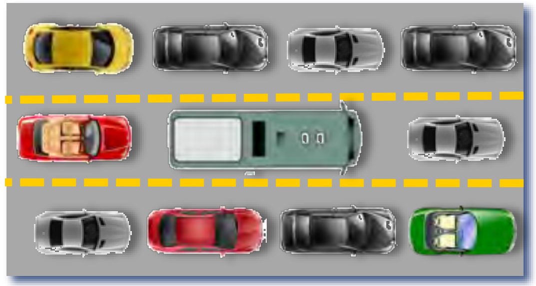
ORACLE

# Challenge: Mixing Workloads Degrades Response Times

**Only OLTP**



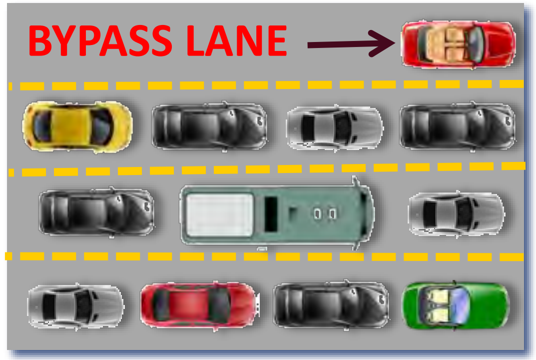
**Mixed Workload**



- OLTP often runs concurrently with high throughput workloads
  - Database consolidation, batch, real-time analytics, reporting, backups
- However, high throughput workloads can severely degrade OLTP
  - They create long network queues, delaying critical OLTP messages

# Cloud: Identify and Accelerate Critical OLTP Messages Across Full Stack

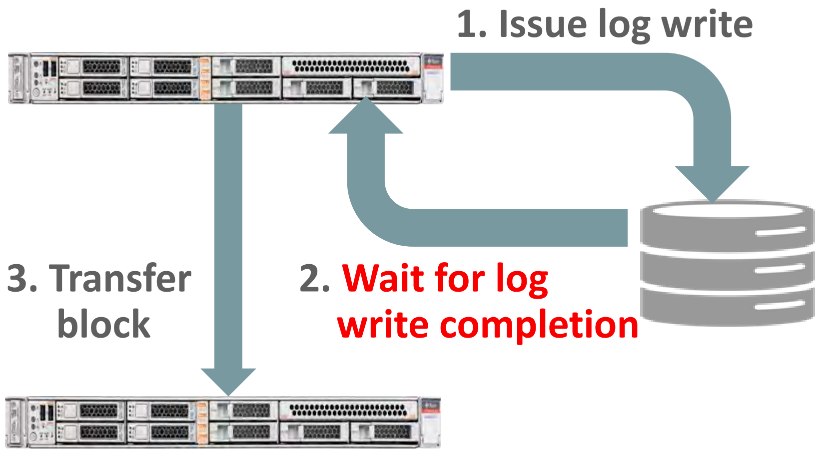
***Network Resource Management***




- Database tags messages that require **low-latency**
  - Log writes, cache-fusion messages, locks, etc.
- Low-latency messages bypass all other messages
  - Reporting, backups, batch, etc.
  - Even partially sent messages are bypassed
- Accelerate low-latency messages in all layers: database, network cards, switches, and storage
  - Otherwise bottleneck just moves
- Cloud deployment transparently enables network resource management

ORACLE

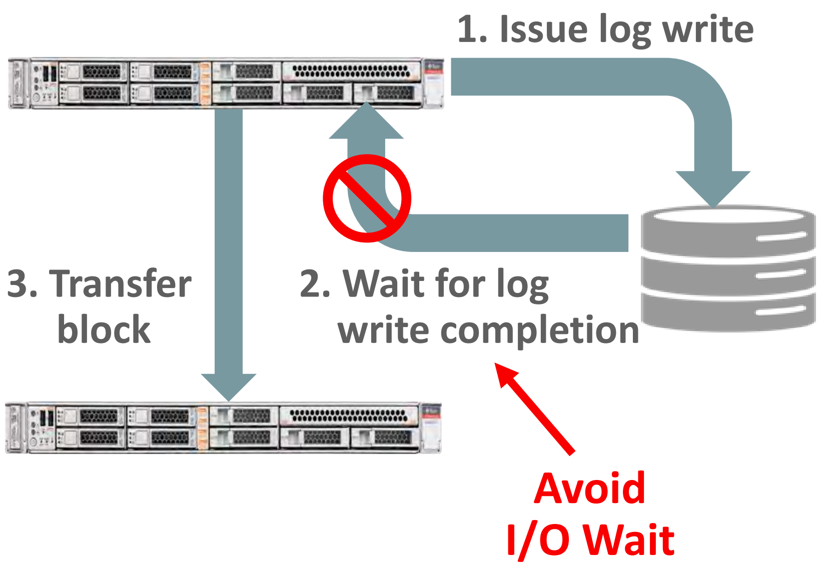
# Challenge: Transferring Hot Database Blocks Slows OLTP




- OLTP workloads can have hot blocks that are frequently updated
  – Right Growing Index for example
- Before transferring a block between nodes, all changes to the block must be written to the log
  – Ensures changes are not lost due to a node crash
- Waiting for a log write to complete delays critical OLTP communication

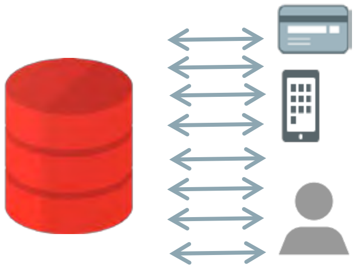
# Cloud: Postpone Log Write Synchronization to Avoid Delaying OLTP Block Transfers

***Smart Fusion Block Transfer***




- **Eliminate the wait** for log write completion before transferring a block
- Destination node can modify block but will wait at commit time if log write has not completed
  - Enabled by uniquely tracking of log writes across nodes
- Needs special fencing support in the storage
  - Cannot enable for all customers on-premises
- Transparently enabled in the Cloud

# Modern OLTP in the Cloud



- Use low-latency, high-bandwidth InfiniBand infrastructure
- Enable zero-copy, OS bypass networking for fast cache fusion transfers
- Enable writing to two parallel media to reduce “log file parallel write” outliers and “log file sync” waits
- Provide latency capping on I/Os transparently removing flaky devices for consistent response time
- Enable transparent network resource management prioritizing critical OLTP I/Os and cache fusion transfers
- Enable faster cache fusion transfers for contended blocks by removing log write wait

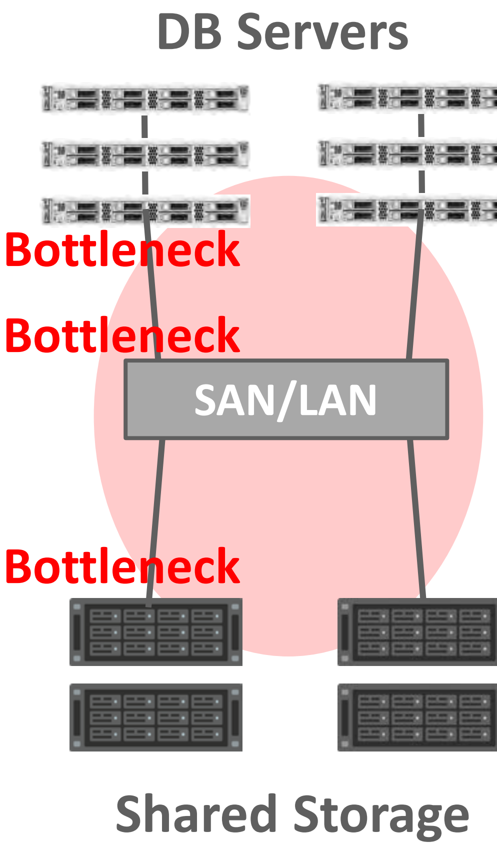
ORACLE

# Characteristics of an Data Warehousing System

- Ingest large amounts of data quickly
- Scan petabytes of data in a very short amount of time
- Minimize storage costs
- Provide Consistent response times
- Uptime is critical

# Challenge: Sharing the <u>Performance</u> of Flash Across Servers




<u>SAN Link = 40Gb</u>
5 GB/sec

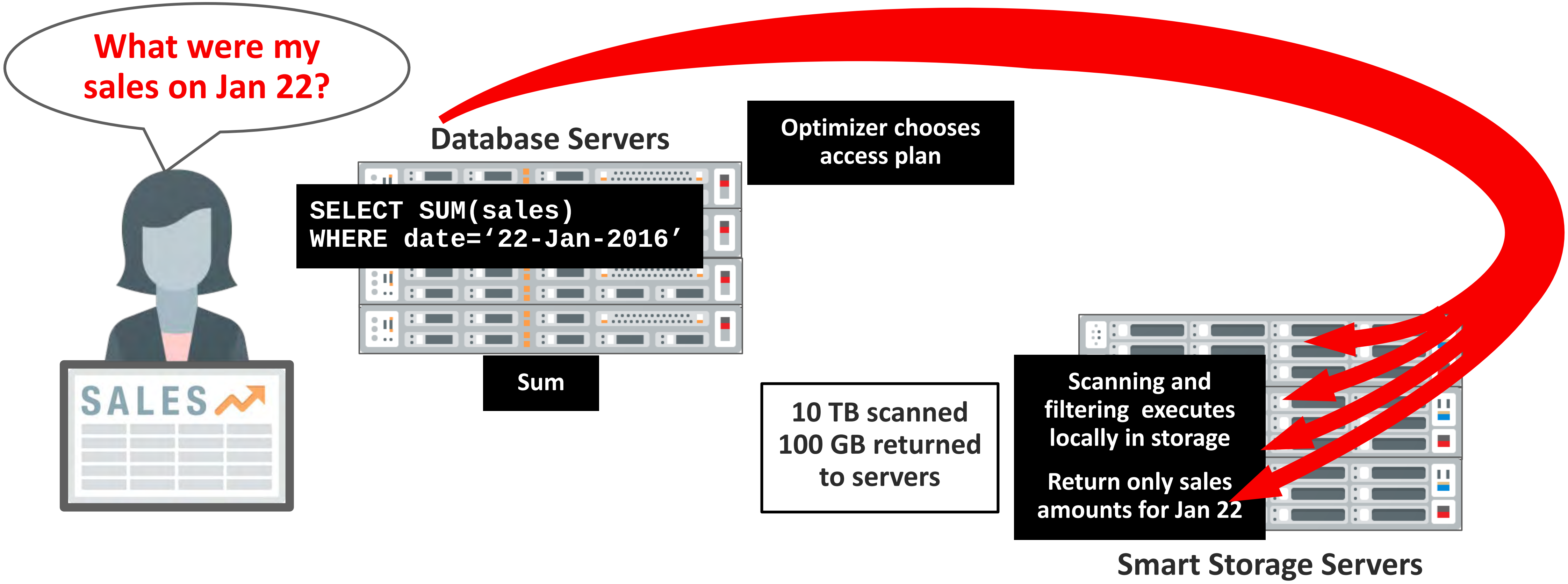
<u>Latest PCIe Flash</u>
5.5 GB/sec

- **Shared storage has many compelling advantages**
  - Much better space utilization, security, management, reliability
  - Enables DB consolidation, DB high availability, RAC scale-out
- **Sharing capacity is easy, sharing <u>performance</u> is hard**
  - Deliver performance of <u>all</u> shared flash drives to <u>any</u> server(s)
- **Flash performance has improved dramatically causing 100X bottlenecks across shared storage stack**
  - Speed of one flash card is now similar to fastest SAN or LAN link
  - A few flash cards deliver more throughput than
    - A storage array can output, a SAN/LAN can transfer, a server can input
  - Scale-out storage helps but does not solve the problem

ORACLE

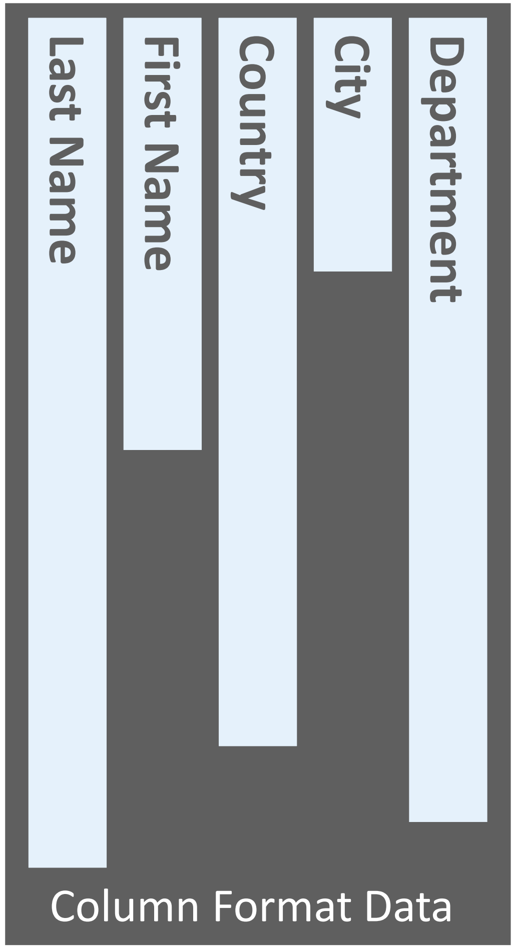
# Cloud: Move Queries to Data, Not Data to Queries

## *Smart Scan*




- Encrypted data can be decrypted in parallel by storage and need not use compute CPU
- Smart Storage keeps summaries of min and max values of columns to reduce I/Os

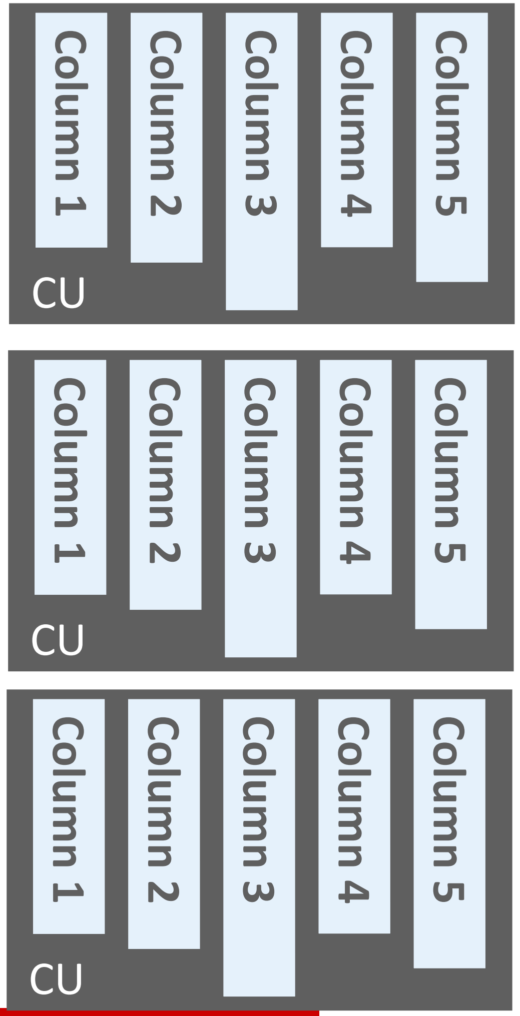
# Challenge: Columnar Format with Fast Row Access




- Columnar format stores the data in each column together rather than the data in each row
- Columnar format is great for Analytics
  - Enables fast scans of columns relevant to a query
- Columnar format is great for Compression
  - Values within a column are much more similar than across
- Pure Columnar format is horrible for Random Row Access
  - Requires an I/O for each column in a row rather than a single I/O for the entire row
  - 100x slower random row access – Columnar Cliff

# **Insight:** Columnar with Row Locality Enables Fast Row Reads

## *Hybrid Columnar Compression*




- Organize columns into sets of a few thousand rows
  - Compression Units (CUs)
- Within CU, data is organized by column, then compressed
  - Get all the compression benefits of full columnar format
  - For analytics, compression greatly reduces I/O, and the columnar format reduces CPU
- Each CU is small enough to be read from storage in a small number of I/O operations (usually 2)
  - Random row access does not require *one I/O for each column*

ORACLE

# Modern Data Warehousing in the Cloud

- Avoid moving data into the server and creating bottlenecks instead by moving queries to data and not data to queries
- Decrypt data if possible in parallel using storage CPUs and not database CPUs
- Use Smart Storage to find the blocks that are needed for the scan and only return results to the database
- Use Smart Storage Indexing to skip I/Os for blocks not needed in the scan

ORACLE

# Ideal Database Cloud Platform

## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. Maintain Quality of Service for all Tenants
4. Fully Secure

ORACLE

# Common HA Pain Points


- Application Brownout on Failure or Planned Maintenance
- Data Corruptions
- Disruptive Schema Changes
- Disaster Recovery System Doesn't Keep Up with Production

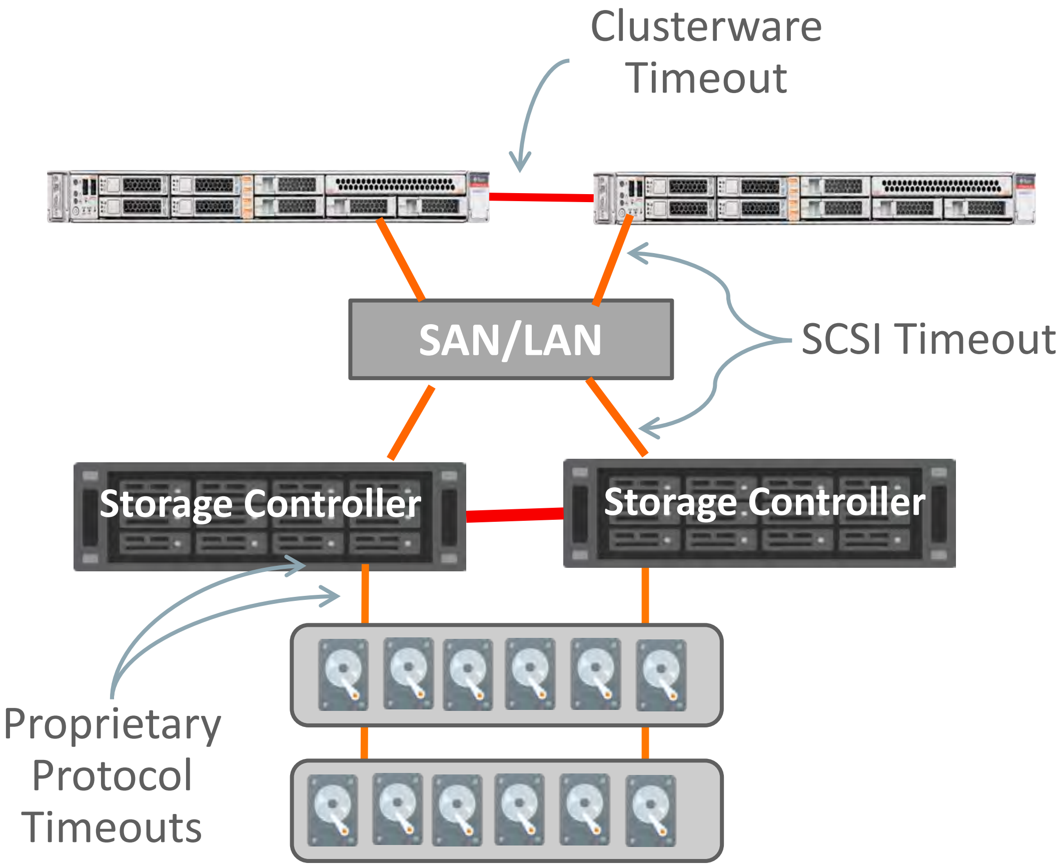
# Brownouts and Blackouts

## Its All about Service Levels

- A **brownout** is a significant service level degradation. A **blackout** is a complete service level interruption
- Brownouts and blackouts translate to lost productivity and revenue
- Systems are complicated with many components, and an issue at one layer can easily cascade to another layer and exacerbate the impact

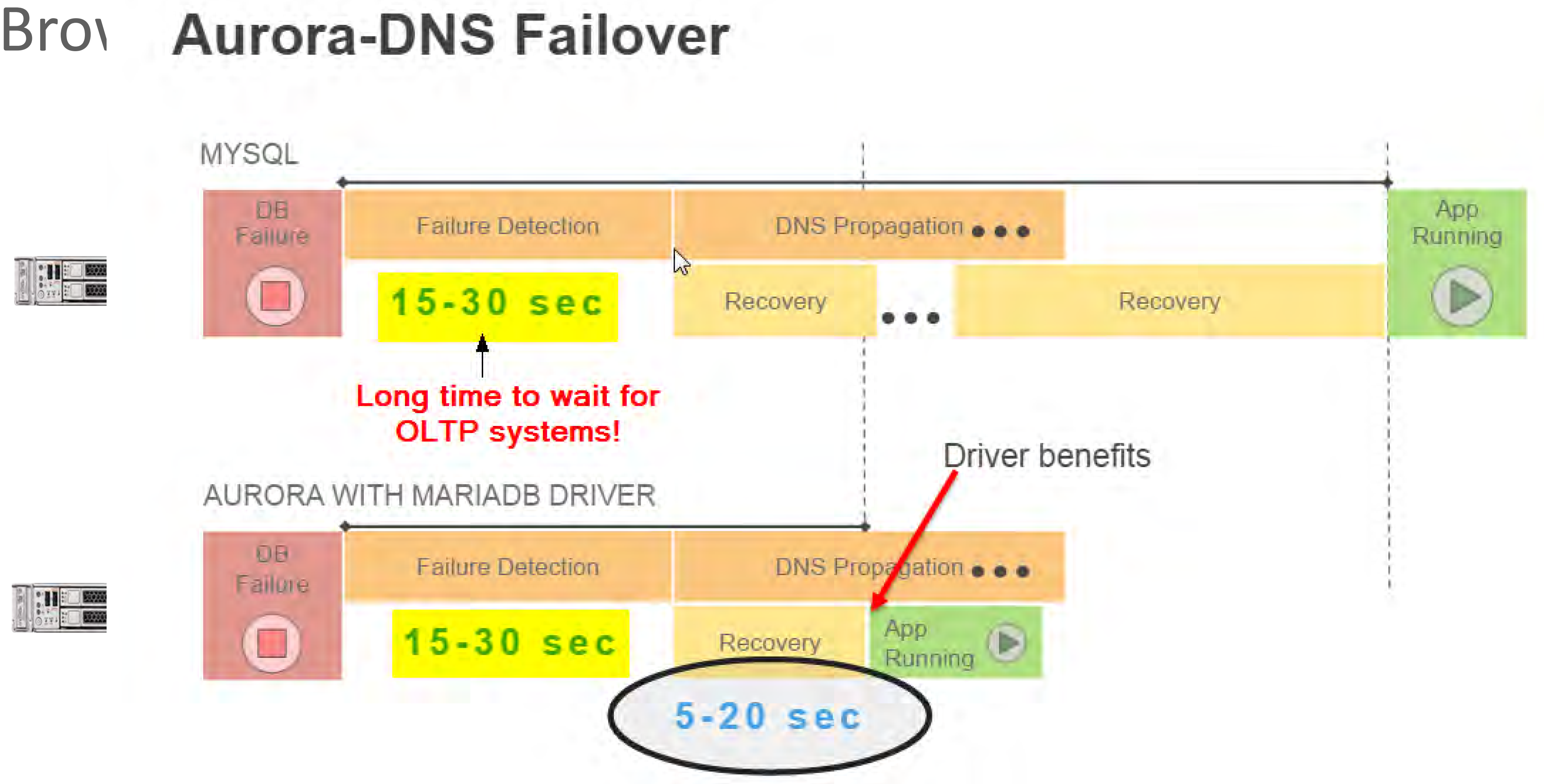
ORACLE

# Application Brownout in a Typical Configuration




- Each layer of the application stack has it’s own failure detection method
- Vendors try to obfuscate these details by quoting client side failure numbers
- In most cases the fault detection times are additive
- Multiplicative fault detection times created by layers of the stack
  - Large timeout = OS block layer tries * OS timeout * ( controller retries * controller timeout * ( disk retries * disk timeout ))

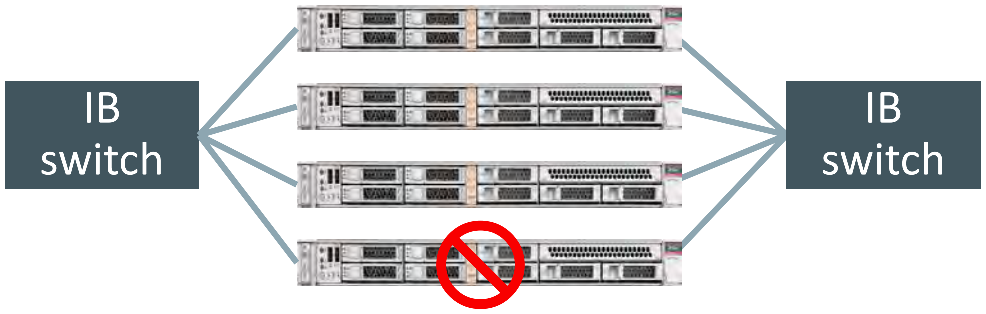
# Challenge: Slow Detection of Server Failures Causes Brov




ires

ction

# Cloud: Instant Server Failure Detection




- InfiniBand switches know immediately when a server has lost connectivity
  - Not available on commodity switches
- Upon missing a server heartbeat, software queries the two InfiniBand switches that are connected to the server
  - If both switches can't "see" the server, then Exadata can reliably declare the server failed
- Reduce brownout in the Cloud for OLTP

# IO Errors and Corruptions – Definitions and Impact

- IO errors occur when the disk sector cannot physically be read
- Corruptions exist when the disk sector can physically be read, but it contains incorrect information.
  - Possible causes are Lost Writes, Torn Writes, Mis-directed writes, Bit rot, Parity Pollution.
- Bad sectors and corruptions can live for a long time before being encountered.
- Encountering them at the wrong time can be extremely disruptive if they aren't handled properly and can result in application downtime and even data loss.

ORACLE

# What is the Exposure?

## Research Paper by Univ. of Wisconsin,  USENIX'07

- 1.5 million NetApp disk drives analyzed for latent sector errors and corruptions
  - Latent sector errors for nearline disk drives over 32 months
    - 8.5% of disk drives had errors detected
    - 13% of errors went undetected
  - Latent sector errors for enterprise class disk drives over 32 months
    - 1.9% of disk drives had errors detected
    - 38% of errors went undetected
  - Corruptions across all drive types over 41 months
    - 400,000 checksum mismatches
  - Newer, larger capacity drives have higher errors per drive

ORACLE

# What is the exposure?

- Tieto, a prominent Swedish IT service supplier, had a storage array fail causing five days of chaos …
  - ...SBAB bank was heavily affected, despite having a 99.8% uptime agreement with Tieto
  - The stoppage was caused by failures in a storage array and compounded by an inadequate disaster recovery plan involving tape backup files which could not be read

# Cloud: Database Aware Approach to Handle Errors

- Challenge: Database read encounters corruption
  - Solution: Database reads ASM mirror copy and repairs corruption
- Challenge: Disk sector goes bad
  - Solution: Scrubbing finds bad sector and ASM repairs it
- Challenge: Network packet containing database write is corrupted
  - Solution: Storage server prevents write of corrupt block using HARD checks
- Challenge: Lost storage device, need to restore redundancy
  - Solution: Restore the most critical files first
  - Priority restore: Control Files, Log Files, SP files, TDE key store, OCR, Wallet
- Challenge: Storage replication copies corruptions to replica
  - Solution: Use Data Guard for corruption detection and prevention

ORACLE

# Modern Cloud Database Platform Availability


- Use instant server failure detection to reduce OLTP brownouts
- Perform corruption checks in storage server before performing I/O
  - End to end validation of I/O
  - Validate file#, rDBA, checksum, redo log sequence number etc at storage
- While restoring redundancy for lost disks, restore redundancy for critical files first

# Ideal Database Cloud Platform

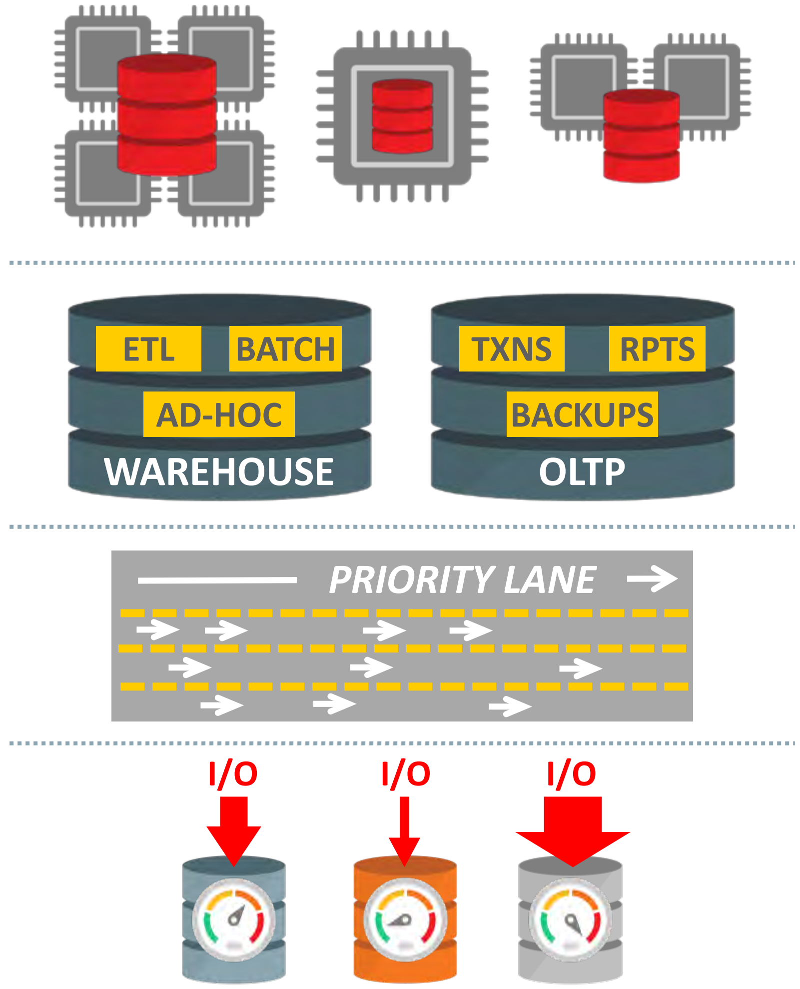
## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. **Maintain Quality of Service for all Tenants**
4. Fully Secure

ORACLE

# Cloud: Resource Management

## Prioritize System Resources by Database, Workload and Time of Day




- **Instance Caging**
  - Limits a database instance to a maximum number of CPUs
  - Prevents resource hogging when consolidating databases
- **CPU Resource Management**
  - Allocates CPU across different databases
  - Allocates CPU across workloads within a database
  - Implements parallel execution policies
  - Prevents runaway queries
- **Network Resource Management**
  - Automatically prioritizes critical messages on InfiniBand fabric
  - Log writes, RAC cluster messages, etc.
- **I/O Resource Management (IORM)**
  - Prioritizes I/O for critical workloads over non-critical workloads
  - Allows fair sharing for database consolidation, pluggable database aware

# Ideal Database Cloud Platform

## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. Maintain Quality of Service for all Tenants
4. Fully Secure

ORACLE

# Evolving Security Landscape

- Challenge: New OS vulnerabilities, almost every week
- Solution:
  - Limit the OS distribution to bare essentials
  - Online update procedures for all key components of the stack
  - Update the entire stack at once – OS, Storage, Database, Networking
  - Upcoming technologies: Security in Silicon
- Challenge: Widespread data breaches, rouge users
- Solution:
  - Encryption with little or no overhead (Use AES-NI with FIPS compliant decryption)
  - Limit access models and touch points for Test/Dev Snapshots

ORACLE

# Ideal Database Cloud Platform

## Key Tenets for Running a Database Workload in the Cloud

1. Same or Better Performance than current system
2. Extreme Availability
3. Maintain Quality of Service for all Tenants
4. Fully Secure

ORACLE

# Integrated Cloud

## Applications & Platform Services