Scaling Database Infrastructure @PayPal

Pramod Garre 05/21/2020

Agenda

1. Introduction
2. PayPal’s Scale
3. Scaling challenges
4. Scaling Methodology
5. Horizontal Scaling
6. Vertical Scaling
7. Q & A

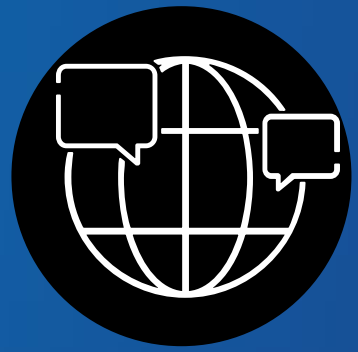
About me

- Database Engineer at PayPal for 8+ Years
- Working on ORACLE Technologies for more than a Decade and ORACLE Certified professional
- www.linkedin.com/in/pramodkgarre

About PayPal

Two decades ago, our founders invented payment technology to make buying and selling faster, secure, and easier; and put economic power where it belongs: **In the hands of people**

Our 300+ Million consumers can accept payments in > 100 currencies and interact with 20M+ Merchants across 19K+ corridors



Almost 8000 PayPal team members provide support to our customers in over 20 languages

We are a trusted part of people's financial lives and a partner to merchants in 200+ markets around the world

Database Infrastructure & Storage Footprint

5M+
Execs/Sec

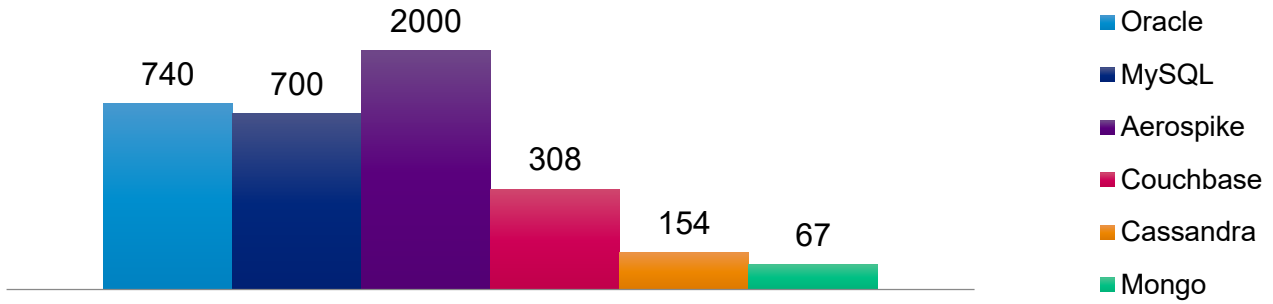
750+
ORACLE Instances

32% Y-o-Y
DB Storage Growth

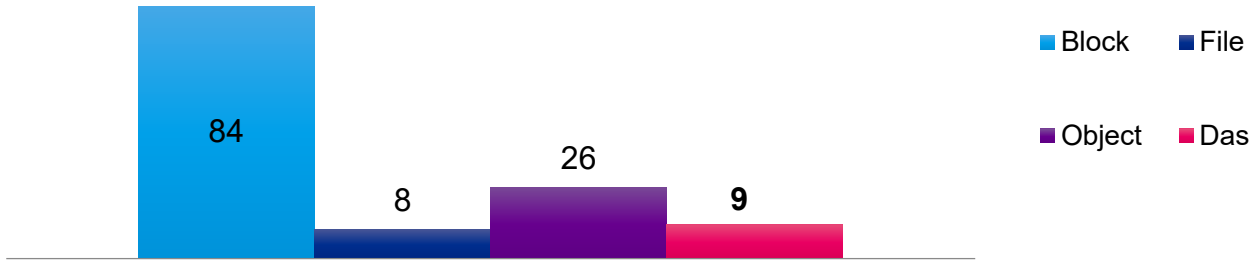
93 PB
Total DB Storage

PayPal

Host Count by Database Type



Storage Footprint (PB) by Type (Utilization)



Scaling challenges

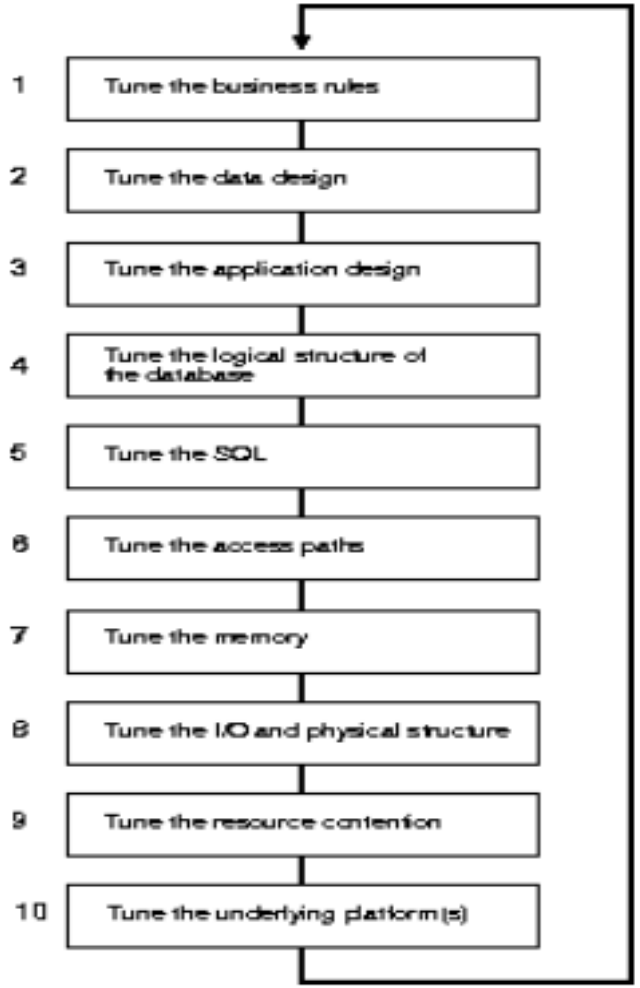
We are only as strong as the slowest component of the syster

- Hardware Limits
  - CPU
  - Memory
  - IOPs
  - Network
  - Interconnect
- Software Limitations at Scale
  - Concurrent waits- Enqueues (Table/Index/Sequences/LOBs)
  - REDO – LGWR contention
  - SGA contention – latches/Mutex waits

Tuning .. Scaling Methodology

Replace Tune with Scale, While the business demands change rest of the approach is still relevant.

- Data/Application Design
  - Right Data Normalization, choosing the right Datastore
  - Application layer caching , pagination and connection pooling etc.
- Logical Structure of the Database
  - Address object-level Bottlenecks - Divide and Conquer
- System Tuning
  - Scale up
    - Add more Power (CPU, Memory, Faster Disks, Storage Cache etc.)
  - Scale out
    - Add more instances – Replicas, Shards, split by Domains and A/A



Snippet from ORACLE 7 Document

Data/Application Design to Scale

Best way to utilize resources is not to utilize them

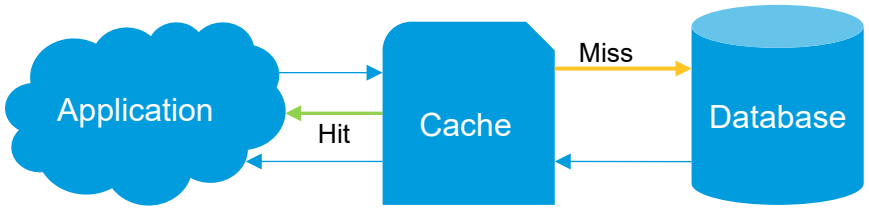
Application Layer Design considerations

- Application level caching
- Pagination of results
- Optimal SQLs
- Intelligent Mid-Tier
  - Persistent connections and Multiplexing
  - Slow SQL eviction
  - SQL caching/routing

Data Layer design considerations

- Right level of normalization and Design
- Avoid “hot spots”
- Design considerations based on the type of Data.

  Ex: Master Data Vs Transactional Data



Application Caching

Scaling Database logical structures

-Divide and Conquer with in the Database

"In computer science, **divide and conquer** is an algorithm design paradigm based on multi-branched recursion. A divide-and-conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem."

- **Table/LOB Contention**

  At **~20k inserts/sec**, table/lob can get into contention – "enq :HW contention "

  - Out-of-line Lob writes with similar size go behind the same latch resource and causes contention.
  - Partition the table and creating multiple entry point helps

    Ex: Range Hash sub partition
  - Number of sub-partitions and sub partitioning key was based on studying read patterns.
  - Range sub partitioning along with appropriate local/Global Hash indexes alleviate Table contention
  - Use secure file+ cache LOBs

To convert the heap table to new design chosen -
create a new table with chosen design -> Redirect reads to UNION ALL view of current and new table and Redirect writes to New table with instead of trigger on old table

Scaling Database logical structures
-Divide and Conquer

Scaling Indexes

Right-hand index contention– ”enq: TX - index contention”

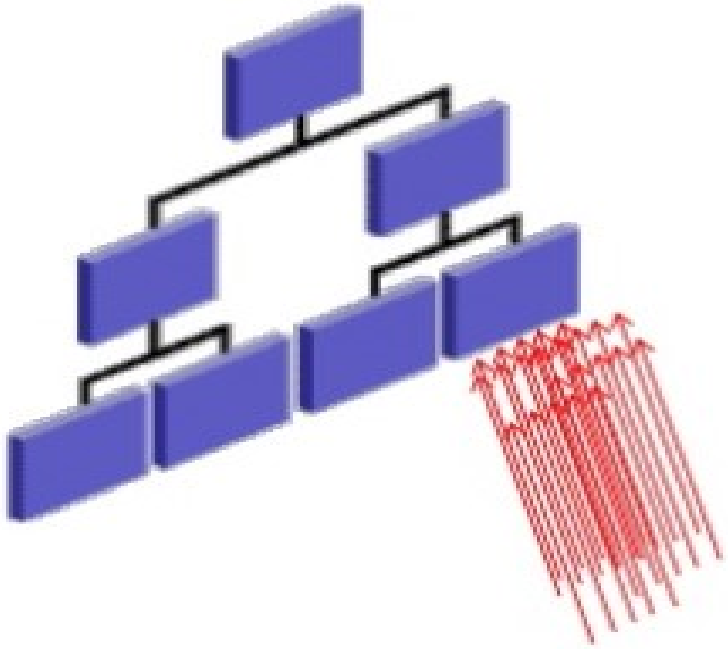
- Partition the table and index to create multiple entry points

  ex: Range-Hash partitions, Global Hash Indexes.

- Scatter the Index Key – Reverse-Key, Timed IDs etc.

Scaling Sequences

- CACHE, NO-ORDER Sequences in RAC
- Intelligent Mid Tier with Read-Write Split option



Scaling Database logical structures
-Divide and Conquer

Scaling IOTs @ ~20k inserts per second

IOTS accelerate Primary Key based access. Scaling IOT writes is critical

- Reduce Write contention for same Index Blocks
  - Choose appropriate partition/sub partition structure (Range, Range Hash etc)
  - Create the index on mod(Key) to scatter the records across multiple segments and blocks

Scaling/Tuning Instance components

Scaling Interconnect traffic

Scale the Interconnect for optimal RAC performance

- Database Service isolation to avoid interconnect overwhelming
- Better Data design to avoid Multi-table join queries overwhelming Interconnect Message Bandwidth
- RDS on InfiniBand (up to 40 Gbps speeds) to achieve ultra-low latency and throughput
- Critical Instance Background processes supporting Cache fusion to run in RT priority.

Scaling LGWR

LGWR is a single threaded process and often single most contention point in ORACLE

- Place REDO log files on faster Disks – flash, RAID 10 Disks
- Following application best practices like proper commit/sleep intervals

Scale up and Scale out

Scale UP

Add more resources -uplift to new powerful hardware that leverages the latest technology and add more such nodes. Cost is the main consideration

- CPU

  8 socket machines with 2.9GHz processors and up to 192 cores

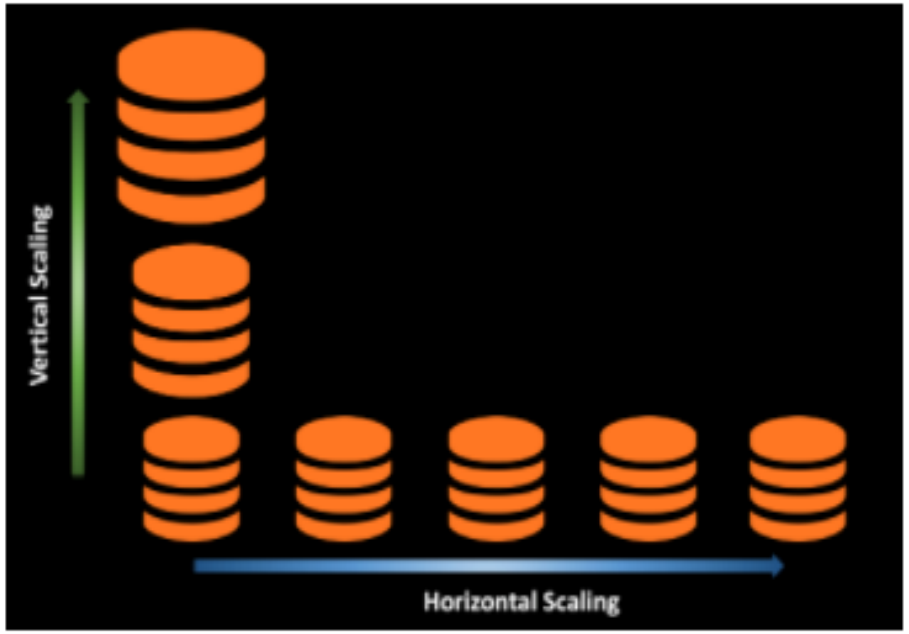
- Volatile Memory

  Up to 6 TB per node

- Non-Volatile Memory

  Nvme flash, Nvme-SSD etc-Bandwidths of 120GB/Sec &50K IOPs

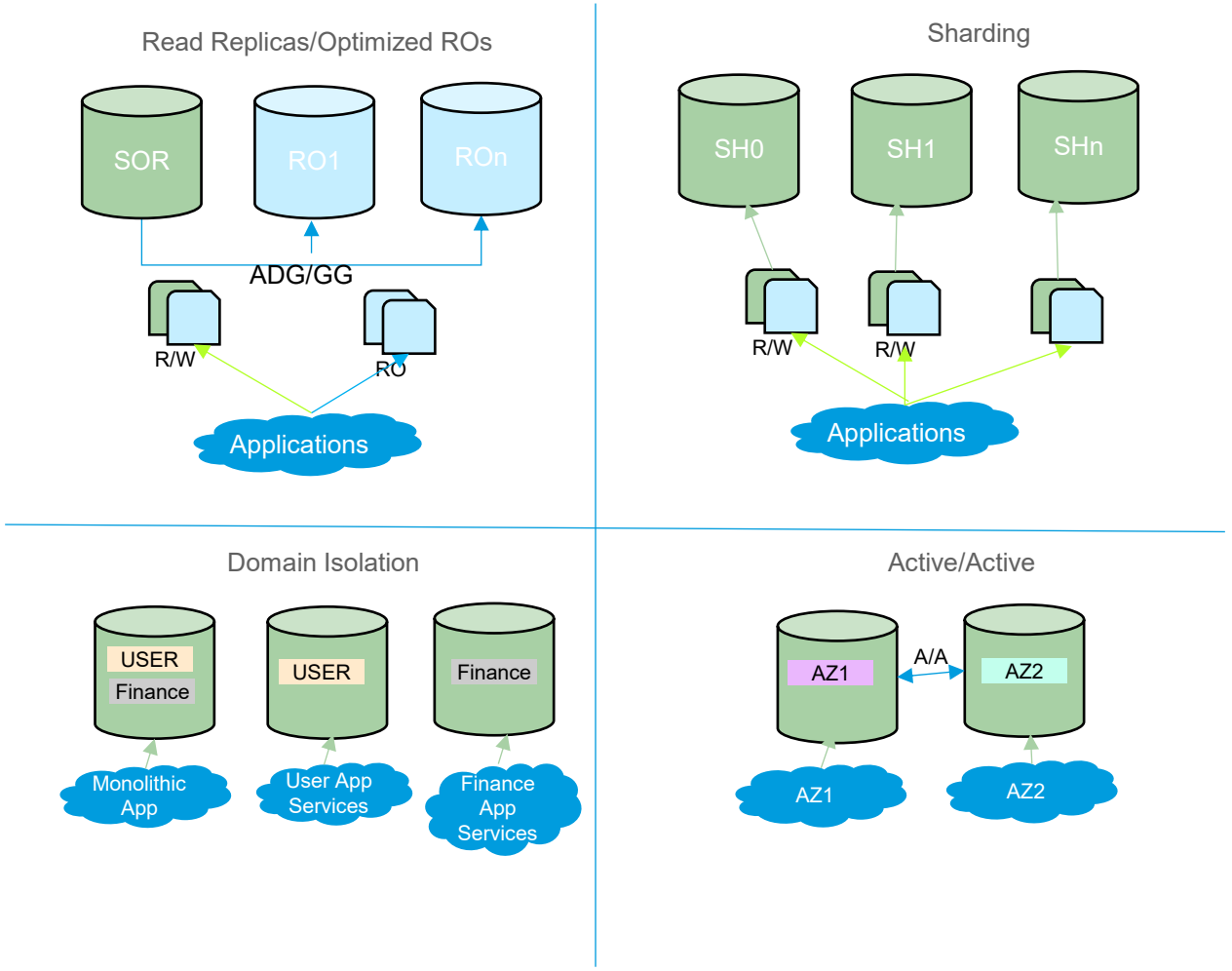
- Storage –High storage cache . 7200 RPM HDDs ,All flash storage
- InfiniBand up to 40GBPs, RoCE network Fabric up to 100GBPs



Scale out Patterns

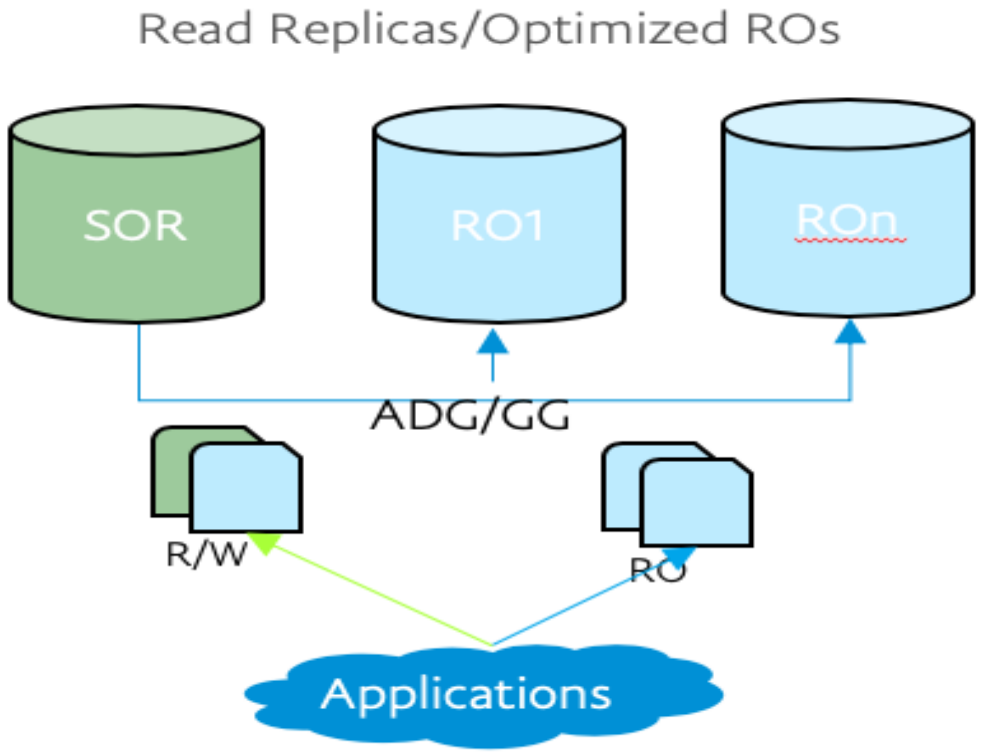
Scale up involves cost and Scale out enables elastic scaling

- Multi-AZ Read Replicas
- Sharding
- Domain Isolation
- Active-Active



Horizontal Scaling – Read Replicas

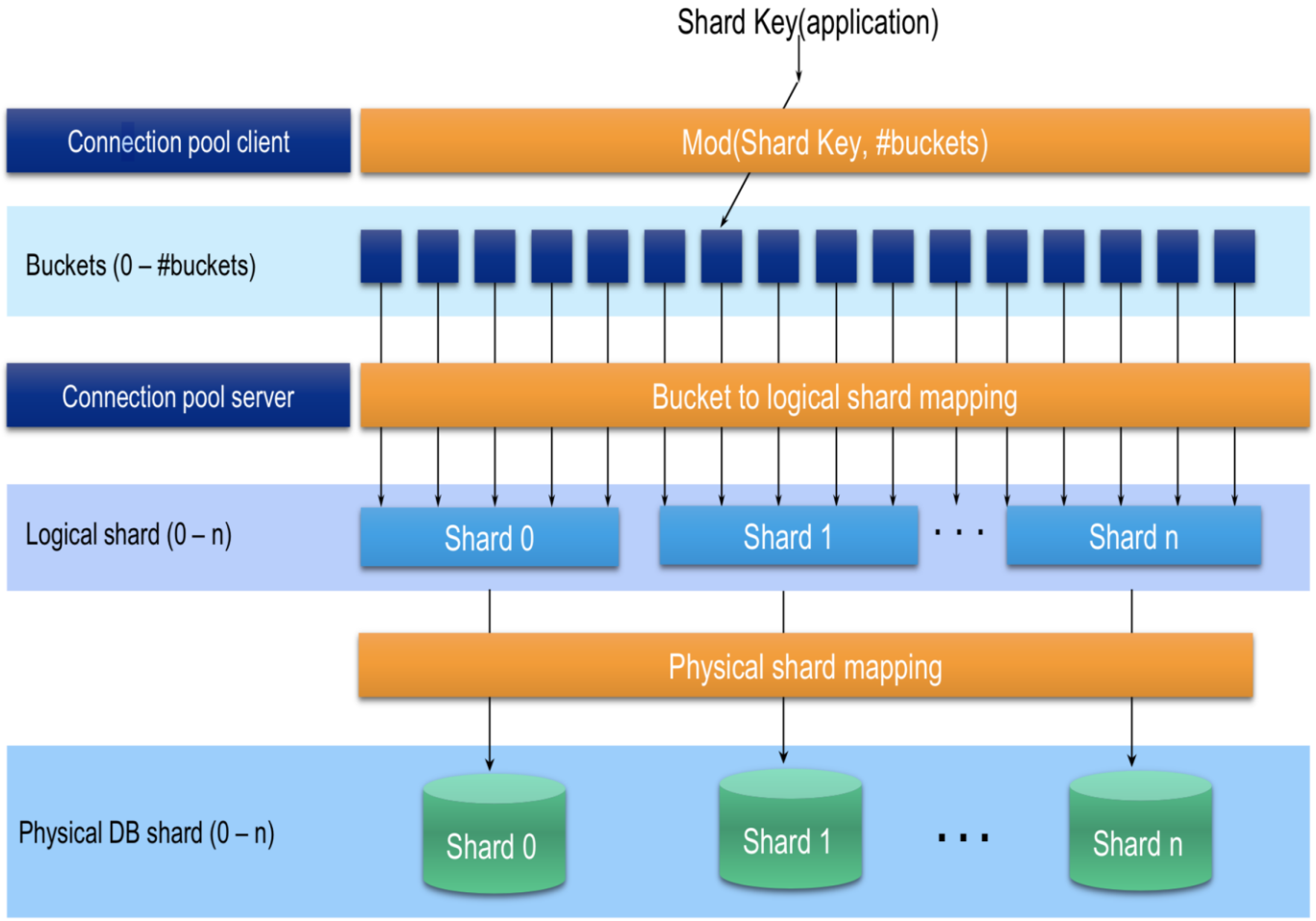
- Scale Read Workloads by adding more replicas
- Optimized Ros (GG Replicas) can also provide High Availability by assuming Primary Role
- Avoid Writes on replicas with DB services
- Full copy on each replica



Horizontal Scaling - Sharding

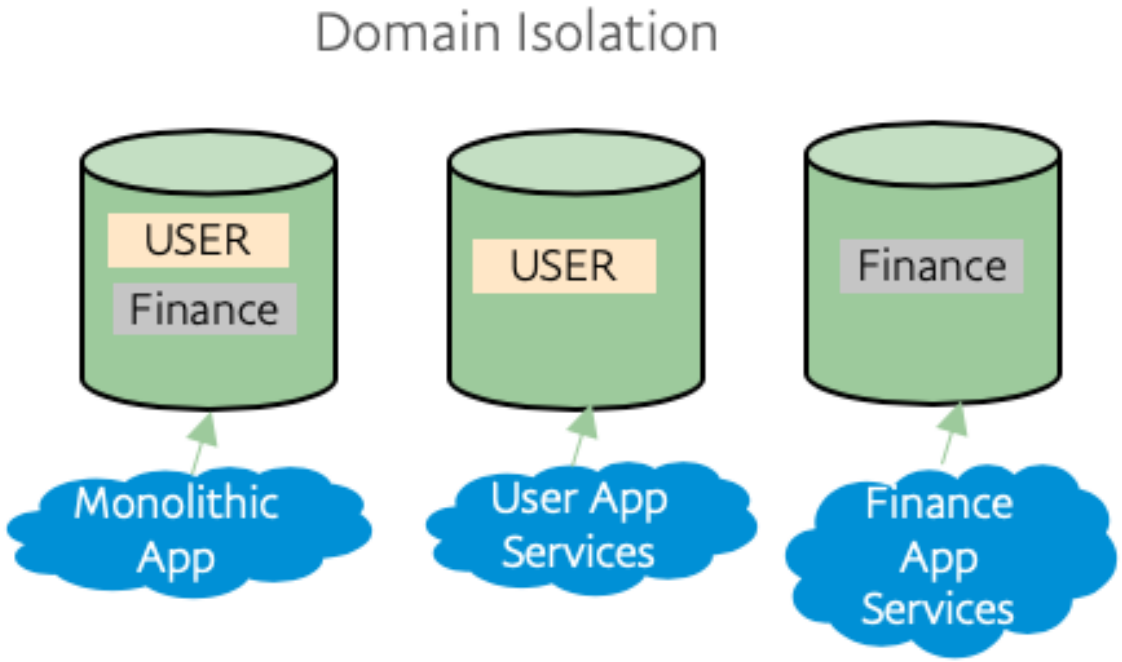
Sharding Rules

- Each table must have a shard key column
- Shard Key Must be Unique across the shards
- No cross Shard joins
- No cross shard writes
- Each SQL should have a shard key
- Tables may need to be denormalized to support above rules
- Each shard has a subset of Data



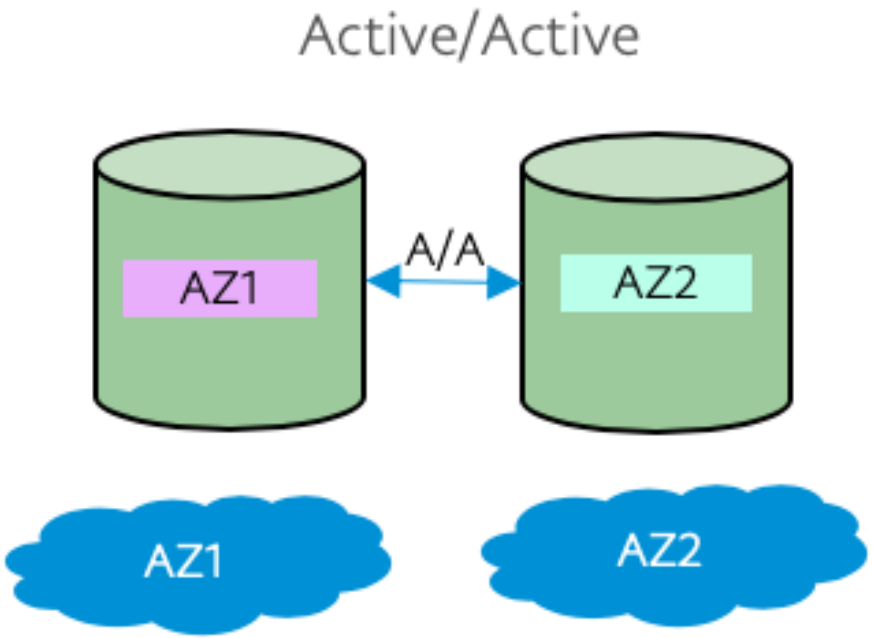
Horizontal scaling – Domain isolation

- Isolate self-contained domains to different physical database
- Logical isolation of tables and application user followed by Physical separation



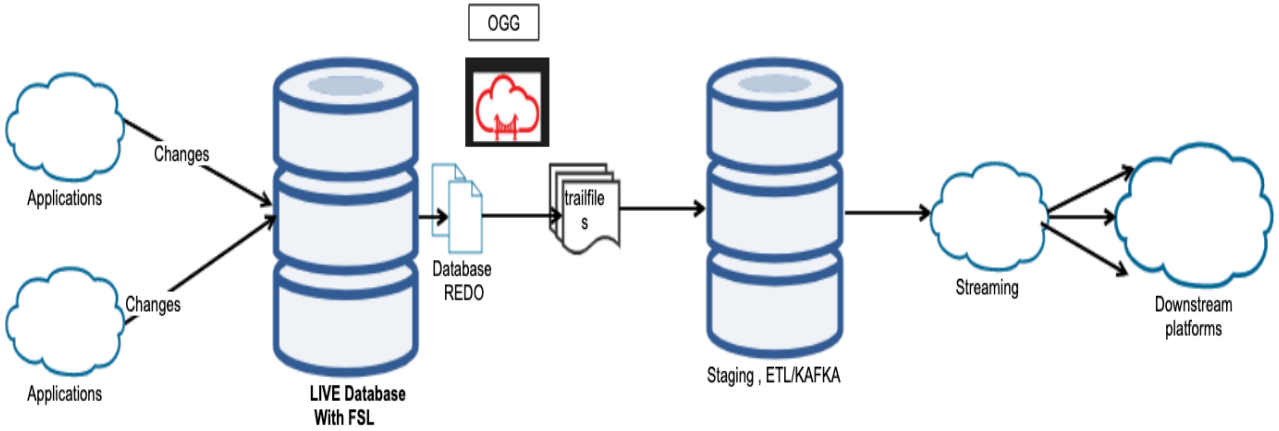
Horizontal scaling – Active/Active

- Active/Active is mainly for multi-region high availability But can also help in linear scaling
- Each Database has full copy of data
- Avoid mutations and collisions across the Databases – Use UUIDs for Keys , Even/Odd Sequences ,GG conflict resolution configuration and application stickiness.



Reporting/Analytic workload offloading

- Changes from FSL enabled SORs are replicated to Centralized data platform
- Near Real time replication with OGG and Kafka/Micro batch processing
- Replication Scaling using @RANGE replicats, Parallel extracts and parallel replicats

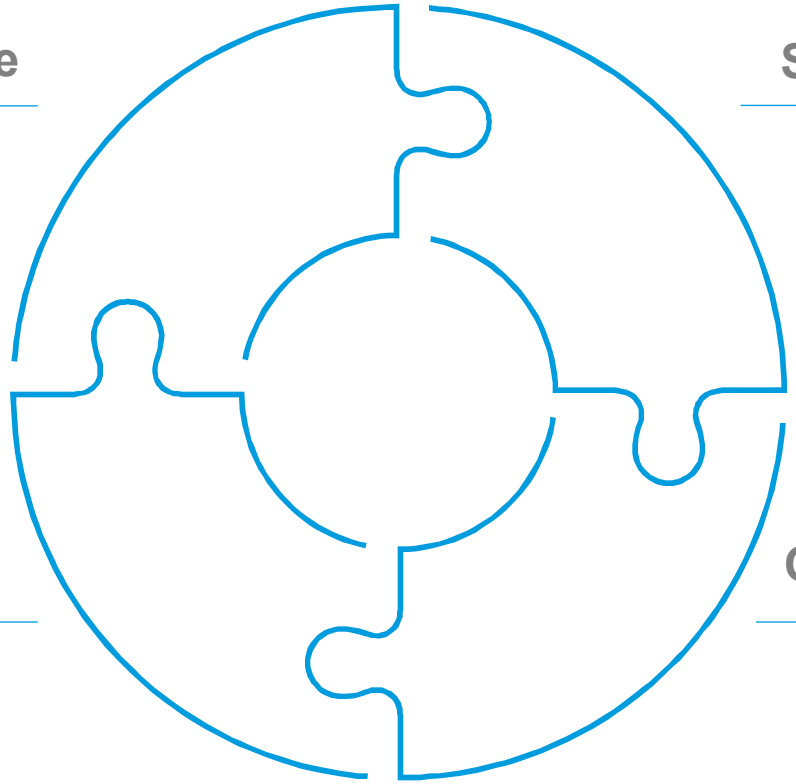


What Next .. Exadata & Oracle 19c

Focusing more on business impacting innovations

Reliability and Performance

- Integrated Hardware and Software designed for Scale
- Unique software optimizations
- Highly scalable & fault tolerant hardware



Security & Compliance

- Effective Data protection by eliminating Database sprawl
- Standard Encryption @ Rest (TDE) , Performance optimization for TDE
- Standard configuration
- Automated and fast patching of all components

Efficiency

- DB consolidation and Multi-tenancy
- Less day-to-day management, more business focusing activities

Optimized for Oracle Database

- Autonomous DB compatibility
- With an infrastructure that's engineered to work together with your Oracle Databases, Oracle Exadata delivers far more power with less hardware.


Q & A