

ORACLE

Oracle弹性分片 - 超大规模全球分布式数据库

Oracle Sharding – Hyperscale Globally Distributed Database

Shailesh Dwivedi

Vice President, Product Management

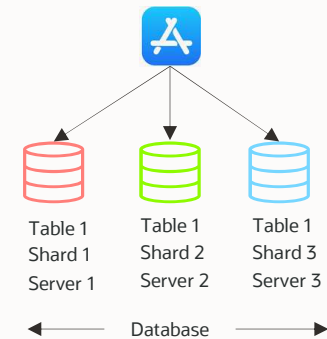




Oracle数据库分片 Oracle Sharding

扩展 Internet 应用程序的主要方法 Dominant approach for scaling Internet Applications

- 跨独立数据库的数据水平分区 (分片) Horizontal partitioning of data across independent databases (shards)
 - 每个shard持有数据的一个子集 Each shard holds a subset of the data
 - 支持高可用复制 Replicated for high availability
- Shared-nothing 架构 Shared-nothing architecture
 - Shard间不共享任何硬件(CPU、内存、磁盘)或软件(集群软件)
Shards don't share any hardware (CPU, memory, disk), or software (clusterware)
- 大规模并行处理 Massively Parallel Processing
- 应用程序直接连接到分片 Application connects directly to shard
- 通过协调器的多切分查询 Multi-Shard queries thru coordinator

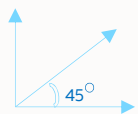


Sharding



分片的好处 Benefits of Sharding

线性可扩展性 Linear Scalability



联机添加分片，以扩大数据库规模和提升吞吐量。在线弹性 Add shards online to increase database size and throughput.
Online elasticity

超高的可用性 Extreme Availability



无共享硬件架构。一个分片发生故障不影响其他分片 Shared-nothing hardware architecture. Fault of one shard has no impact on others.

地理分布 Geographic Distribution



用户定义的数据放置，以满足性能、可用性、灾难恢复或监管要求 User defined data placement for performance, availability, DR or to meet regulatory requirements.



Oracle BlueKai 数据管理平台通过 Oracle 数据库分片扩展到每秒处理 100 万事务 Oracle BlueKai Data Management Platform scales to 1 Million transactions per second with Oracle Database Sharding

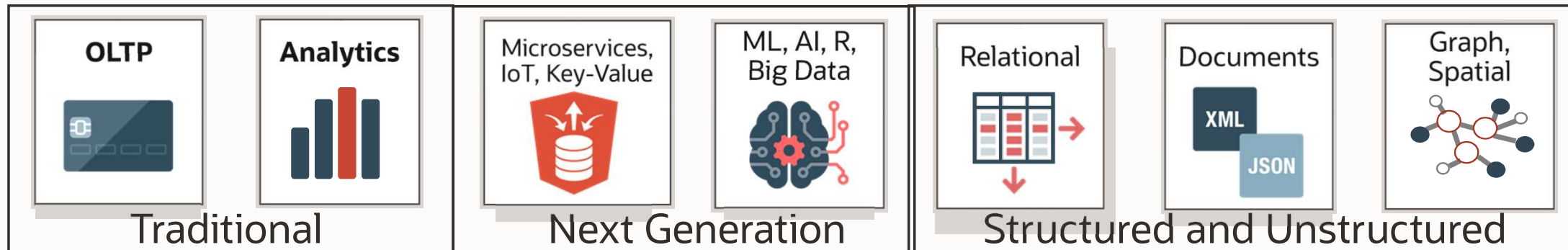
1 million/second transactions	2.5 petabytes total database size	30 billion/day API calls	125 Kilobytes/API call (Average) API payload size
1.6 milliseconds/API call average read time	2.5 milliseconds/API call average write time	22 billion rows in largest table	180 terabytes/hour redo generation rate
52 Oracle compute instances total machines	2,704 cores total CPU	38,740 gigabytes total memory	1 terabit/second network traffic



分片的 Oracle 融合数据库 - 实现简单性和数据协同 Sharded

Oracle Converged Database - Enables Simplicity and Data Synergy

Oracle 融合数据库支持所有工作负载和数据类型 Oracle Converged Database Supports all Workloads and **Data Types**

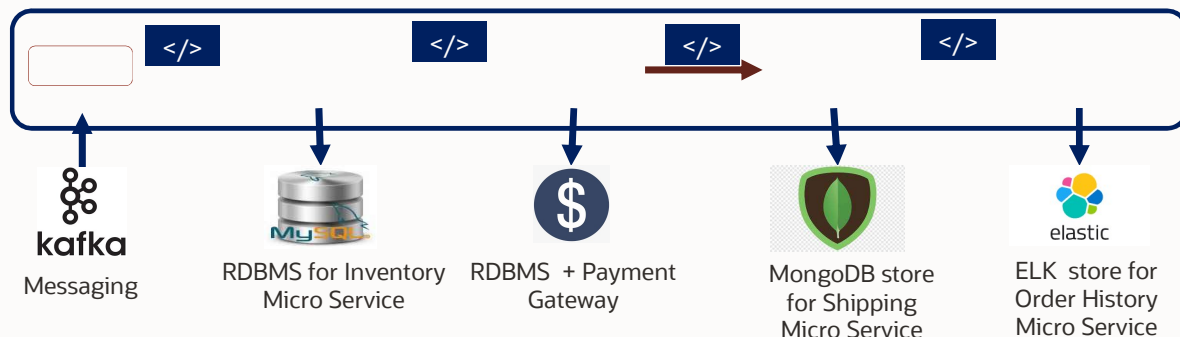


用于结构化和非结构化数据的单一片 Oracle 数据库

Single Sharded Oracle Database for structured and unstructured data

编程范式和持久性存储的激增

Proliferation in programming paradigms and persistence stores



具有专门持久存储的超大规模架构模式 Hyperscale Architecture patterns with specialized persistent store

这种方法需要采购不同的产品、管理它们需要不同的流程、使用它们需要开发人员具备不同技能 This approach requires different products to procure, different process to manage them, different skills to use them

这种架构通常会导致应用程序在问题诊断时遇到障碍 Such architecture often results in applications which are impossible to troubleshoot

Oracle 融合数据库 Oracle Converged Database



不同的数据类型对应独立的PDB
Separate PDBs for different data types



一个数据库支持多种数据类型
Different data types in same DB

关键客户用例

Key Customer Use Cases

客户用例 Customer Use Case	客户在选择 Oracle Sharding 之前评估的产品 Products Evaluated by Customers before choosing Oracle Sharding
Internet Scale Realtime OLTP	Cassandra, MongoDB, MemSQL, MariaDB, Couchbase, Aerospike, ScyllaDB
Global Databases/Data Sovereignty	Google Spanner, Azure Cosmos DB, AWS Aurora, CockroachDB
Log Store	Apache Lucene, Elastic Search, Solr
Metric/Time Series store, IoT, Infrastructure Monitoring, APM	AWS Redshift/EMR, Druid, Cassandra, Graphite, InfluxDB
Machine Learning	Apache Spark, HDFS, NoSQL and SQL Sharded DBs
Big Data Analytics	Apache Spark, MemSQL

客户选择 Oracle Database Sharding 是因为客户评估的许多产品都是分片系统，缺乏企业级特性，例如支持严格的数据一致性、事务、ACID 属性、复杂连接、完整的 SQL 支持、高级安全性、跨区域复制、性能优化器、备份和恢复、触发器、存储过程、定期安全补丁、大规模可管理性。 Customer chose Oracle Database Sharding as many of the products evaluated by customers are sharded systems and lack enterprise grade features like support for strict data consistency, transaction, ACID properties, complex joins, full SQL support, advanced security, cross region replication, performance optimizer, backup and recovery, triggers, stored procedures, regular security patches, manageability at scale



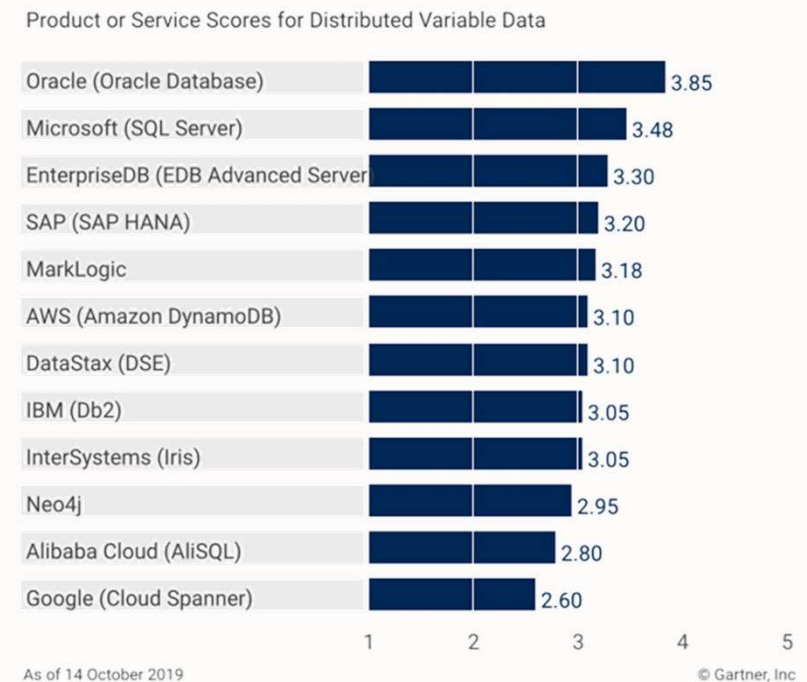
Gartner Sharding魔力象限 – Oracle Sharding排名第一

Gartner Magic Quadrant for Sharding – Oracle Sharding ranked #1

“自我管理”的分片和分发策略曾经是常态，现在被视为复杂而具有挑战性。分布式可变数据用例——特别是地理分布的用例——正在增加，尤其是在云端。在那些将自动化数据分发视为一项需求（并且有利于将其作为产品功能的供应商）而不是一项需要掌握的复杂技能的企业中，情况就是如此。Once the norm, “self administered” sharding and distribution strategies are now seen as complex and challenging. **Distributed variable data use cases** — particularly geographically distributed ones — are on the rise, especially in the cloud. This is very much the case among enterprises that view automated data distribution as a requirement (and that favor vendors that make it a product feature) rather than a complex skill to be acquired.

Oracle 数据库现在包括多种不同的分片功能，增强了自动化数据分发和混合云（分片可以驻留在本地和云中）。The **Oracle Database** now includes **several different sharding capabilities**, enhancing both automated data distribution and hybrid cloud (shards can reside on-premises and in the cloud).

Figure 2. Vendors' Product Scores for the Distributed Variable Data Use Case



[Gartner Critical Capabilities report – Oct 2019](#)



Oracle 分片 – 为客户提供更好的交易

Oracle Sharding – Better deal for Customers

- 超高可扩展性&可用性的NoSQL 数据库 Extreme scalability & availability of NoSQL databases

+

- 分片支持融合数据库 — 关系型、JSON、文本、空间、图形，我们可以对所有数据类型运行单个查询 Sharding support converged database – relational, JSON, Text, Spatial, Graph and we can run a single query across all the data types!
- SQL和您期望的所有编程接口（PL-SQL，OCI，JDBC等） SQL and all the programmatic interfaces (PL-SQL, OCI, JDBC, etc.) that you expect
- 严格的数据一致性、零数据丢失、完全支持ACID事务、2PC、复杂连接、触发器、存储过程 Strict data consistency, zero data loss, full support for ACID transactions, 2PC, Complex Joins, Triggers, Stored Procedures
- 更方便的应用程序维护–数据库中的模式而不是应用程序中的模式 Easier application maintenance – schema in database instead of application
- 企业功能：InMemory, Columnar, Partitioning and Sub Partitioning, Advanced Compression, Secondary Indices, Advanced Security, RMAN, ASM, Data Guard, Golden Gate, Parallel Query, high-performance storage engine, SMP scalability, RAC,
- 利用内部和全球范围的Oracle DBA和编程技能 Leverage in-house and world-wide Oracle DBA and programing skillset

&

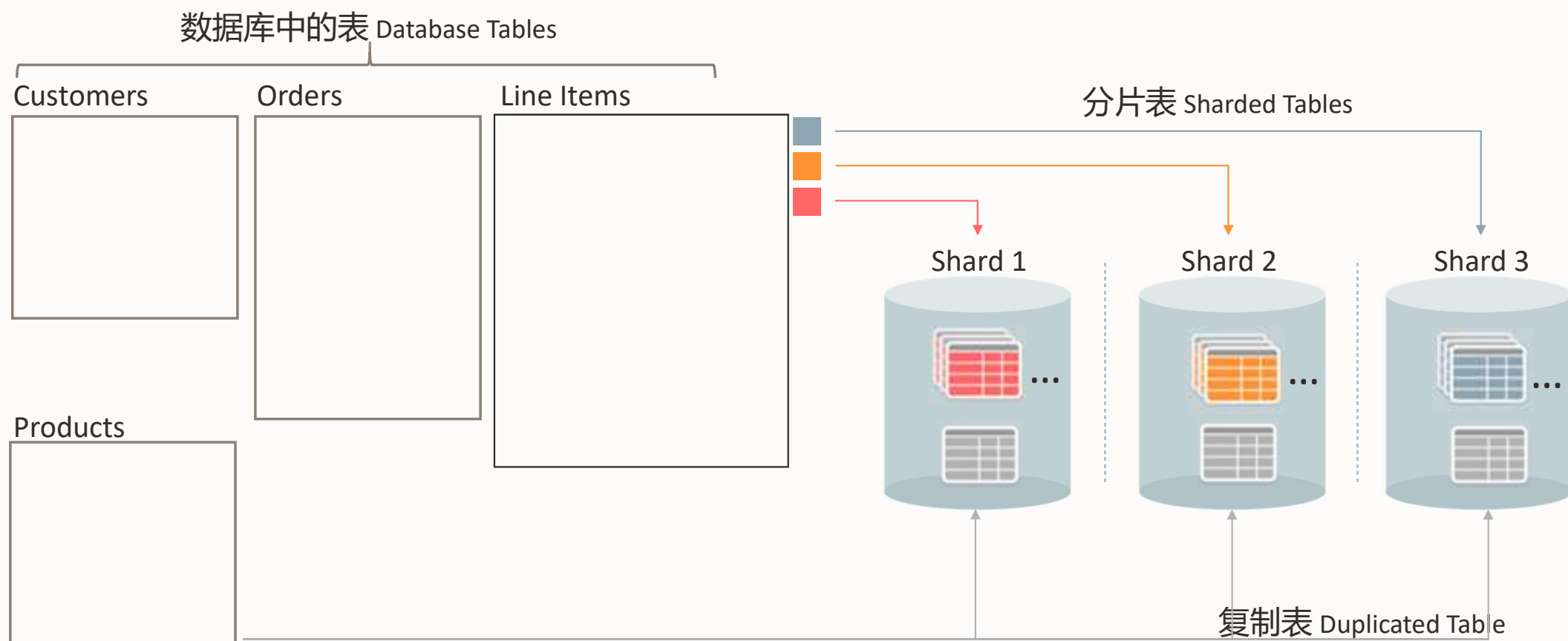
- Oracle数据库分片通常免费提供给客户—他们拥有许可证（DB企业版）、编程、部署和操作Oracle的技能和工具
Oracle Database Sharding is available to customers often at no extra cost – They have licenses(DB Enterprise Edition), skills and tooling to program, deploy and operate Oracle





Schema创建 - 分片表和重复表

Schema Creation – Sharded and Duplicated Tables



概念 Concept: Chunk

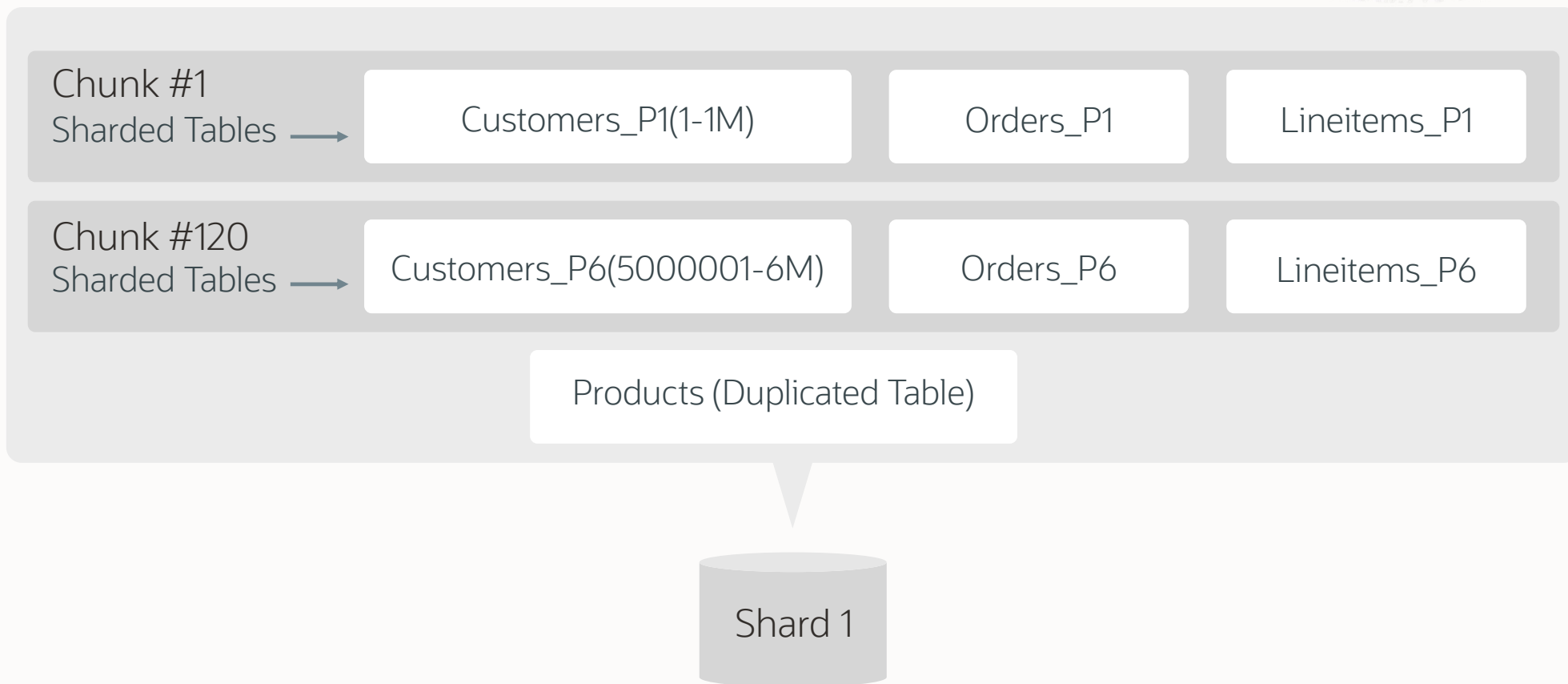
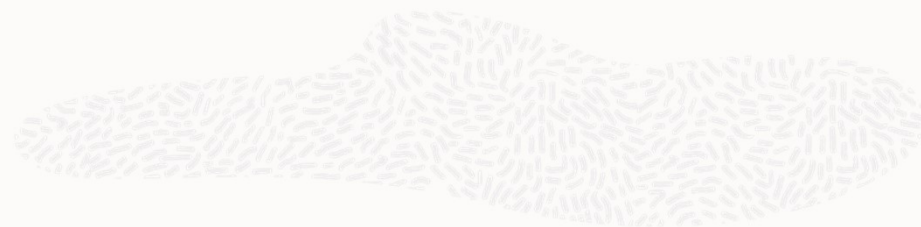
Chunk #1
Sharded Tables →

Customers_P1 (1-10000000)

- 分片表族中有相关分区的组 Group of related partitions of a sharded *table family*
 - 例Eg: Chunk#1 包含contains Customers_P1, Orders_P1, LineItems_P1
- 相同分片键的所有数据都位于给定的同一chunk中 All data pertinent to a sharding key resides in a given chunk
 - 无需访问多个shard数据库 No need to go to multiple shards
- 用于Re-sharding的数据移动单位 Unit of data movement for resharding

分片的内容

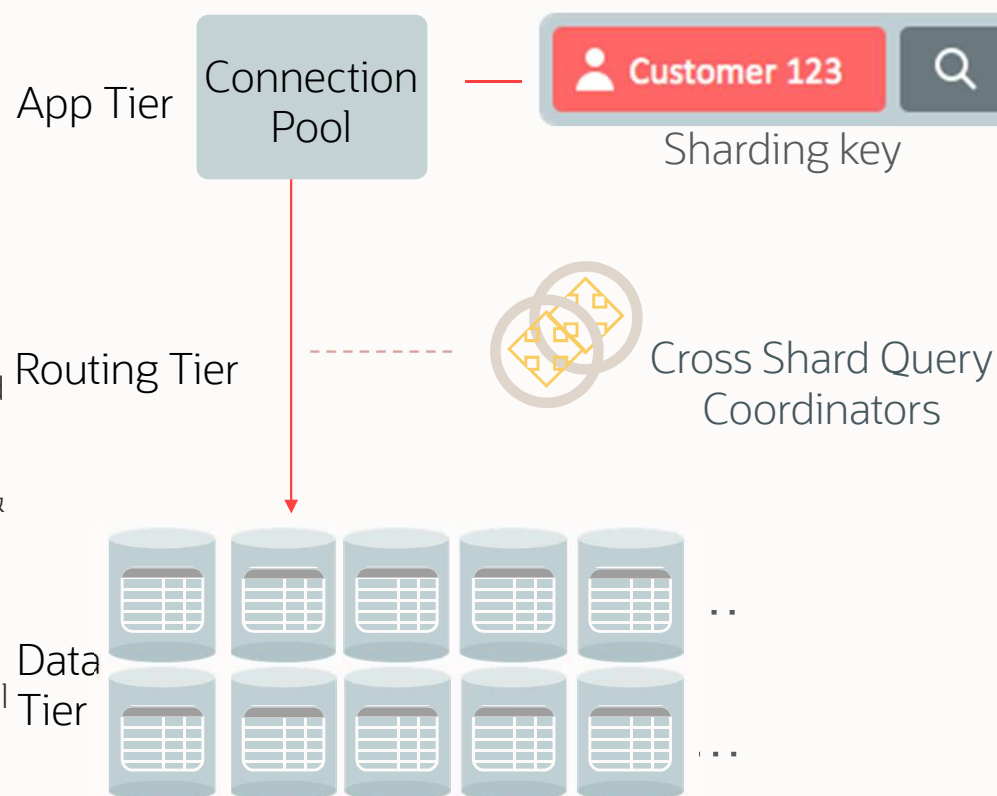
Contents of a Shard



客户端请求流程

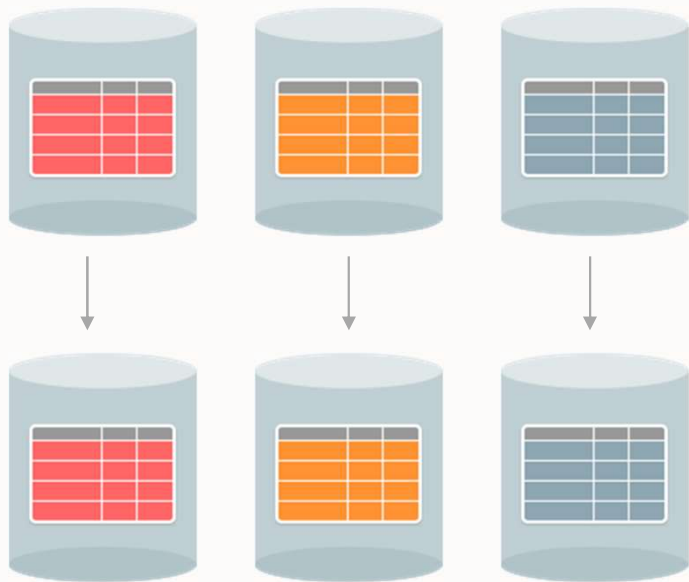
Application Request Flow

- 客户端路由 (JDBC, OCI, UCP, ODP.NET)
Client Routing (JDBC, OCI, UCP, ODP.NET)
 - 从连接池直接路由 Direct routing from Connection pools
 - 多分片查询的代理路由 Proxy routing for Multi-shard queries
- Shard Catalog
 - 存储分片数据库元数据 Stores SDB metadata
 - 充当多分片查询的协调器 Acts as a coordinator for multi-shard queries
 - 包含应用程序黄金架构和重复表 Contains app gold schema & duplicated tables
- Shard Director
 - 全局服务管理器，用于将连接请求直接路由到分片 A global service manager for direct routing of connection requests to shards
 - 通过ONS发布运行时分片数据库拓扑图，负载均衡咨询，FAN事件 Publishes run-time SDB topology map, load balancing advisory, FAN events via ONS

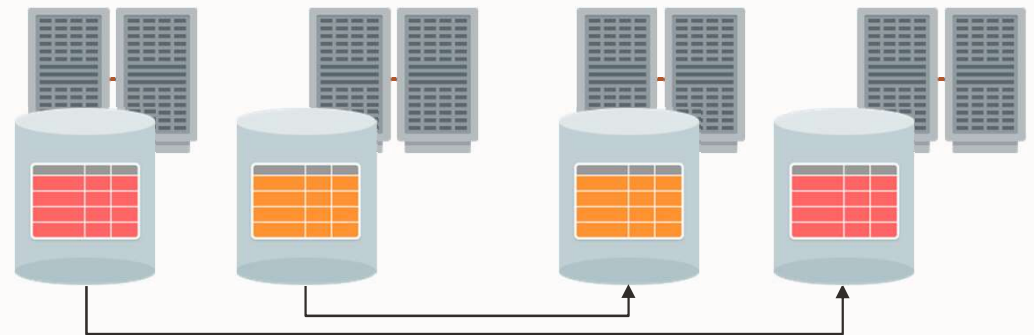


高可用/容灾 配置

HA / DR Configurations



Active Data Guard with Fast-Start Failover



Optionally – complement replication with **Oracle RAC** for server HA

分片方式

Sharding Methods

- 系统管理分片 System Managed Sharding
 - by *Consistent Hash*
 - 每个chunk给分片键分配哈希值范围 Range of hash values assigned to each chunk
- 用户定义分片 User-defined Sharding
 - by *Range*
 - 每个chunk给分片键值分配一个范围 Range of sharding key values assigned to each chunk
 - by *List*
 - 每个chunk给分片键值分配一个列表 Each chunk associated with a list of sharding key values
- 组合分片 Composite Sharding
 - by *Range - Consistent Hash or by List - Consistent Hash*
 - 两级分片, 使用两个键值 Two-level sharding, uses two keys

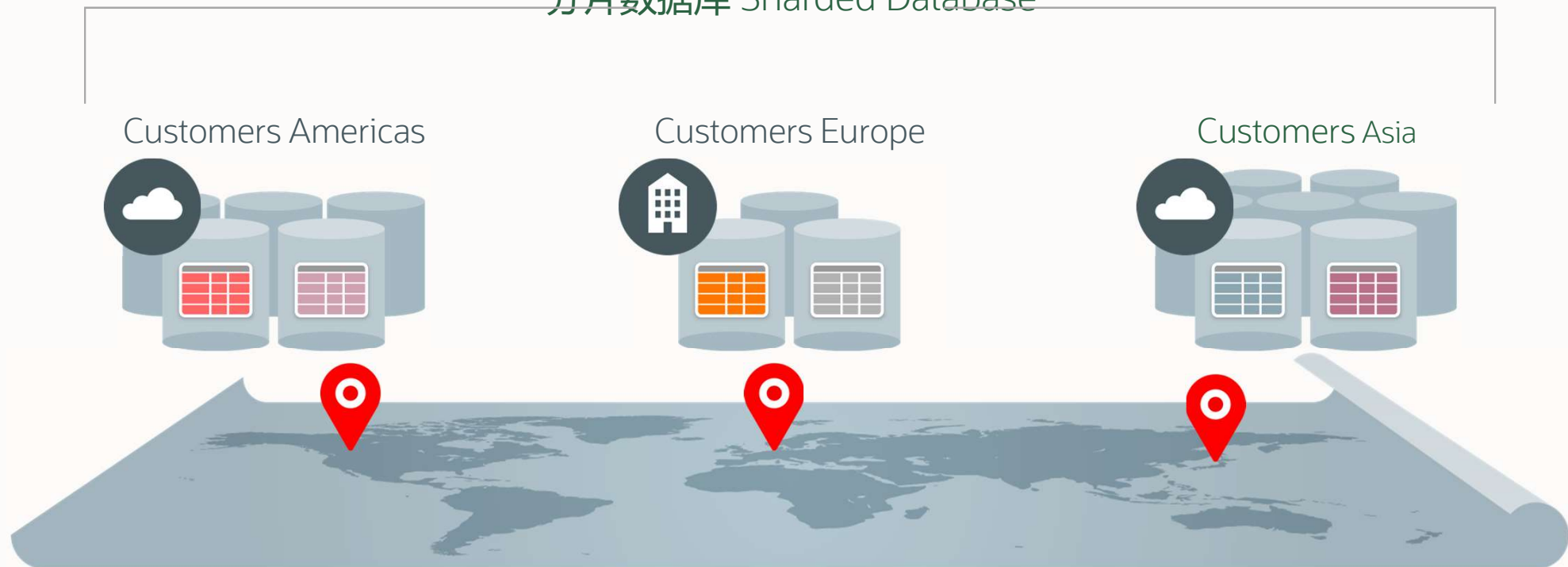


组合分片

Composite Sharding

地理分布和线性可扩展性 **Geographic Distribution and Linear Scalability**

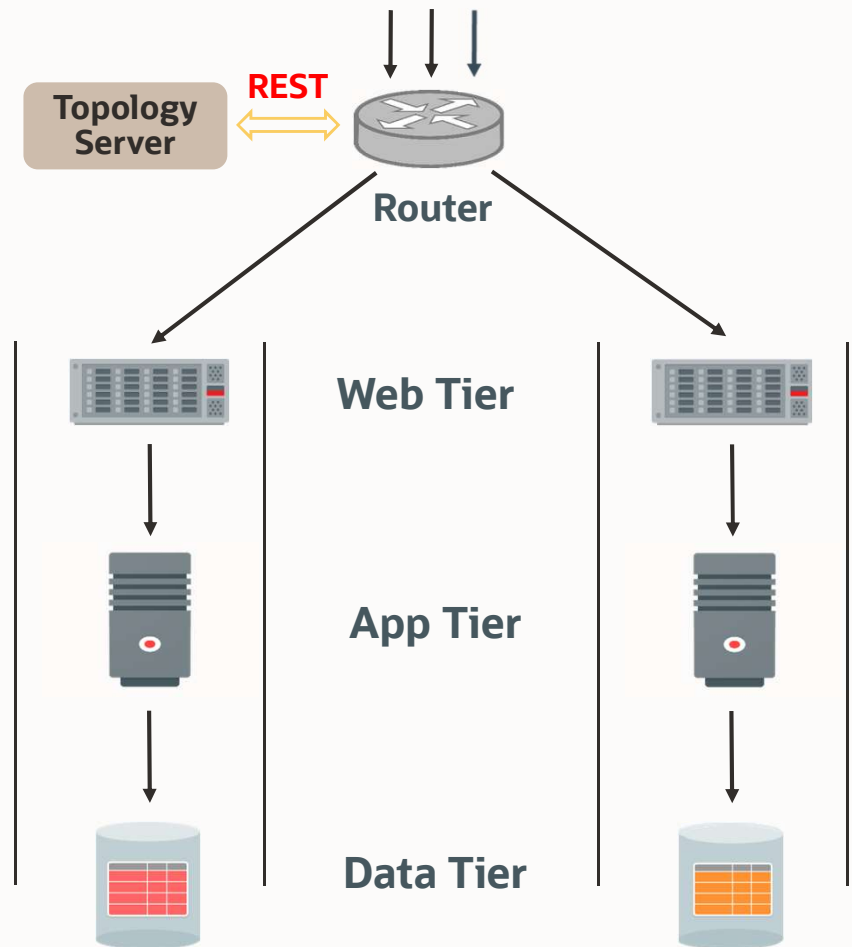
分片数据库 Sharded Database



中间层分片 – 跨层级的“泳道”

Midtier Sharding - “Swim Lanes” across tiers

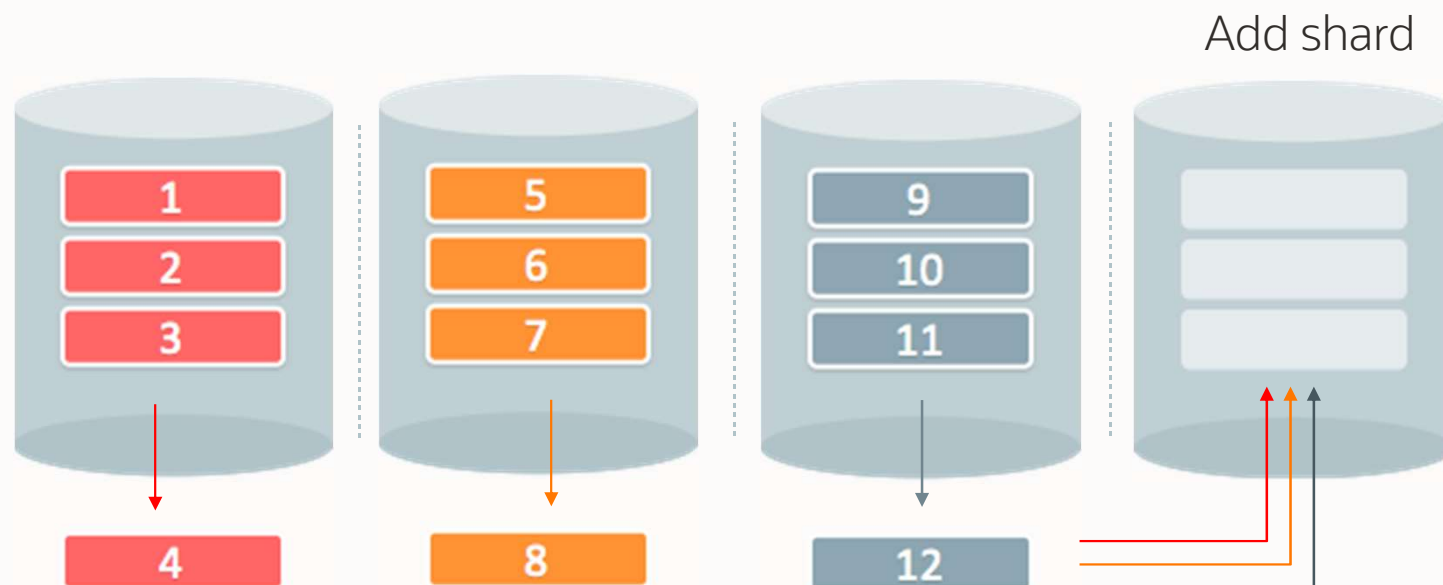
- 提供更好的故障隔离和可扩展性 Provides even better fault isolation and scalability
- 减少数据库连接数 Reduces the number of DB connections
- 改善中间层缓存的位置 Improves midtier cache locality
- 对于地理上分散的分片，消除了跨数据中心的中间层到数据库的连接 For geographically distributed shards eliminates chatty midtier-to-database connections across datacenters
- 允许A / B测试和部分用户访问新应用程序功能 Allows A/B testing with subset of users accessing new application functionality
- REST API将请求路由到与分片密钥关联的中间层 REST API to route requests to the midtier associated with a sharding key
 - 由客户的路由层调用 Called by customers' routing layer



在线添加和重新平衡分片

Online Addition and Rebalancing of Shards

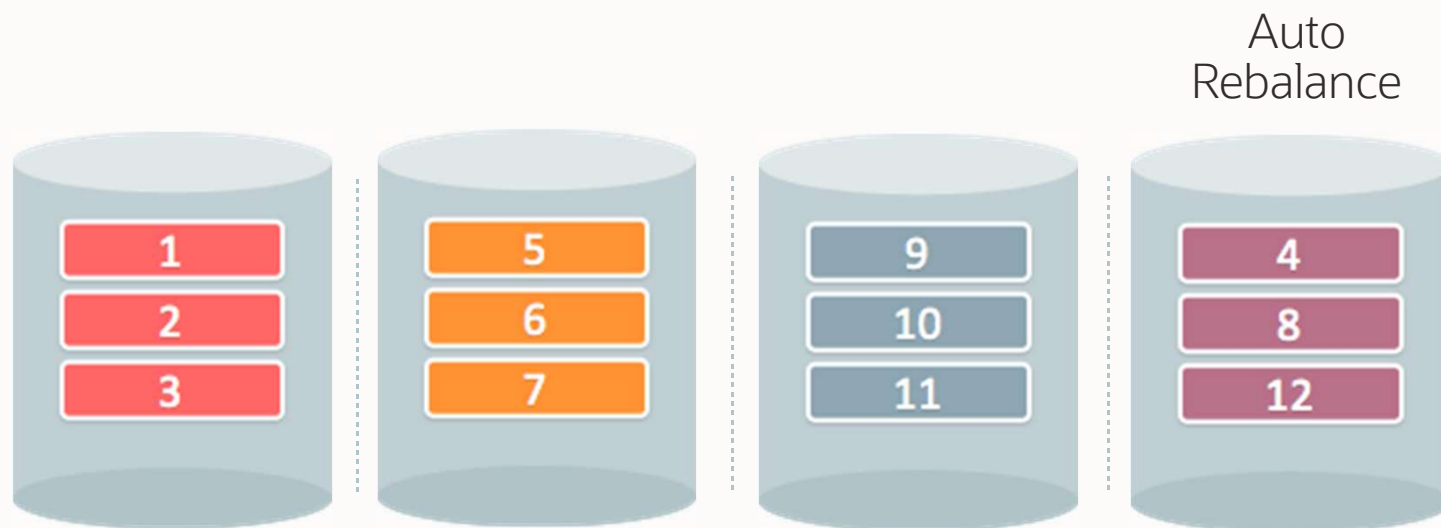
- chunk是Re-sharding的单位 A chunk is a unit of resharding
- 移动chunk可以自动或手动启动（由DBA） Chunk move is initiated automatically or manually (by DBA)
- 使用RMAN增量备份和传输表空间 Uses RMAN Incremental Backup & Transportable Tablespace



在线添加和重新平衡分片

Online Addition and Rebalancing of Shards

- chunk是Re-sharding的单位 A chunk is a unit of resharding
- 移动chunk可以自动或手动启动（由DBA） Chunk move is initiated automatically or manually (by DBA)
- 使用RMAN增量备份和传输表空间 Uses RMAN Incremental Backup & Transportable Tablespace



分片数据库自动化打补丁

Automated Patching of SDB

- OPatchauto 支持 OPatchauto supports
 - 所有分片schema和复制方法 All sharding schemes and replication methods
 - 单实例数据库和集群数据库(包括Grid Infrastructure) Single instance and clustered databases (also handles Grid Infrastructure)
- 为分片数据库打补丁 To patch a sharded database :
- `<CATALOG_DB_HOME>/OPatch/opatchauto apply <patch loc> -sdb -wallet <wallet file loc> -sid <sid of shardcat> -port <shardcat port>`
- 对于 Data Guard For Data Guard
 - 支持滚动模式 (默认: 并行模式) Supports rolling mode (default: parallel mode)
 - 对于给定的配置, 先修补备库, 然后再修补主库 For a given configuration, standbys are patched first followed by primary



shardcat

slc11g.us.oracle.c

Page Refreshed Sep 17, 2016 12:12:05 AM GMT

Summary

Sharded Database Name orasdb
 Configuration Name oradbccloud
 Catalog Database shardcat
 Catalog Version 12.2.0.1.0
 Sharding Type System-managed
 Replication Type Data Guard
 Shard Directors 1 (↑1)
 Master Shard Director shardedirector1

Shard Load Map

Total Active Sessions : 0.05

Instance: shardcat
 Total Active Load: 0.020 active sessions

Load Summary
 CPU: 0.013
 IO: 0.000
 WAIT: 0.007

View Level: Database Instance



Members

Shardspaces Shardgroups Shard Directors Shards

Name	Shardspace	Shardgroup	Data Guard Role	Region
sh1	shardspaceora	shgrp1	Primary	availability_domain1
sh1s1	shardspaceora	shgrp2	Active Standby	availability_domain2
sh2	shardspaceora	shgrp1	Primary	availability_domain1
sh2s1	shardspaceora	shgrp2	Active Standby	availability_domain2

Services

Name	Status	Data Guard Role
No services found.		

Incidents

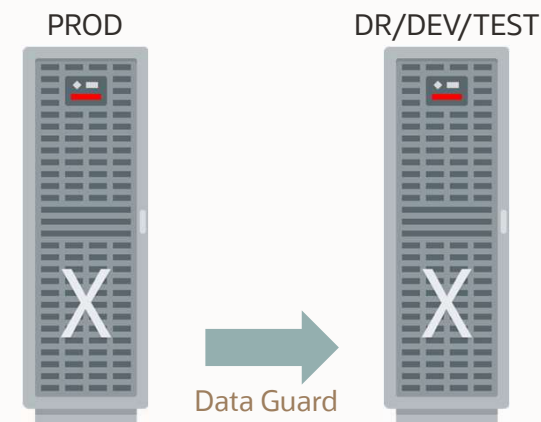
View Target Local target and Related targets Category All

Summary	Target	Severity	Status	Escalation Level	Type	Time Since Last Update
Problem: KUP 600			New	-	Problem	0 days 0 hours
Problem: KUP 600			New	-	Problem	0 days 2 hours
The Data Guard fast-start failover observer status is Error Fast-Start Failover observer is no longer observing this database.			New	-	Incident	0 days 2 hours
Checker run found 1 new persistent data failures.			New	-	Incident	0 days 4 hours

使用 Exadata 进行分片

Sharding with Exadata

- 部署Oracle数据库的最佳平台 Best platform for deploying Oracle database
- 分片提供 Exadata 的所有优势 All benefits of Exadata available with Sharding
- Oracle 数据库的多功能性加上 Exadata 基础架构的强大功能 Versatility of Oracle Database PLUS the power of the Exadata infrastructure
- 简化的生命周期操作 Simplified lifecycle operations
- 支持弹性扩展的处理器容量，无停机 Elastically scale enabled processor capacity with no downtime
- 单个平台上的所有分片工作负载 All Sharded workloads on a single platform
- 关键任务OLTP、分析、非结构化数据（文本/JSON）灾难恢复、报告 Mission-critical OLTP, analytics, unstructured data (Text/JSON) disaster recovery, reporting
- 具有查询卸载功能的存储单元提供数量级的性能提升 Storage cells with query offloading provide order of magnitude performance boost
- RAC 节点提供对节点故障的即时保护 RAC nodes provide instant protection from node failure





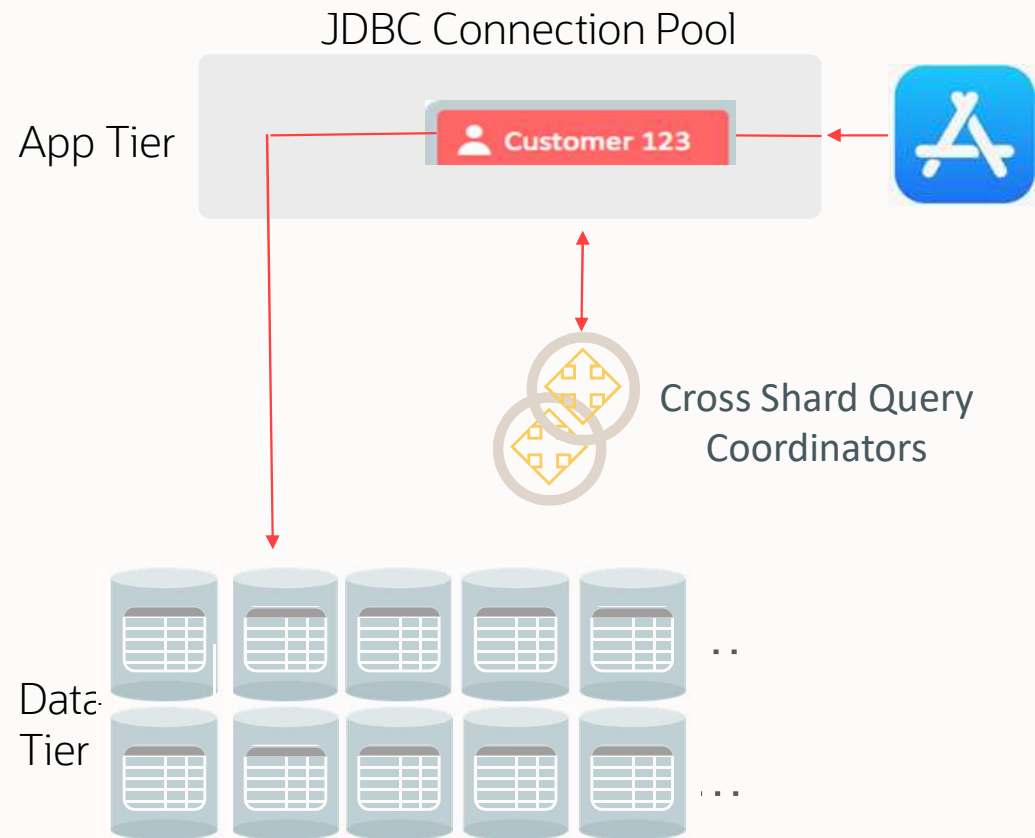
Oracle 分片的创新演变

自动识别 Sharding Key

Automatic Identification of Sharding Key

NEW IN
21^C

- 新的JDBC驱动程序，它使Java驱动连接到分片的数据库，而无需应用程序提供分片的密钥。A new JDBC driver that enables Java connectivity to sharded databases without the need for an application to furnish a sharding key
 - 自动识别给定SQL语句的绑定变量是分片键 Automatically identify which bind variable for a given SQL statement is the sharding key
- 自动查询路由 Automatic routing of queries
 - 单个分片查询到适当的分片 Single shard queries to the appropriate shards
 - 通过协调器跨分片查询 Cross shard queries to the coordinator



自动化部署

Deployment Automation

- 分片顾问 Sharding Advisor
 - 用于建议架构从非分片数据库迁移到分片的工具 Tool to advise on schema migration from non-sharded databases to Sharding
 - 主要目标是最大程度地提高并行度（在所有分片上扩展查询执行），最小化跨分片操作并最小化重复数据 Key goals are to maximize parallelism (spread query execution across all shards), minimize cross shard operations and minimize duplicate data
 - 分析现有的数据库架构，用户工作量并提出建议，例如要对哪些表进行分片，将哪个列用作分片键，要使用的分片方法，要复制的表 Analyze existing database schema, user workloads and makes recommendations like which tables to Shard, which column to use as Sharding Key, Sharding Method to use, which tables to duplicate
- 使用 Terraform, Kubernetes 和 Ansible 脚本实现部署自动化
Deployment Automation with Terraform, Kubernetes and Ansible
 - 简单的输入文件，描述了部署拓扑 Simple input file describing deployment topology
 - 从其中的一台主机运行以进行分布式设置 Run from one of the host for distributed setup
 - 出现错误时可重启/恢复/清理 Reentrant / Resume/Cleanup in case of errors
 - 独立扩展分片组件 Scale out sharding components independently
 - Terraform 部署下载链接在这里 · Terraform deployment download link [here](#)



Shard Advisor Sample Output

rank	tname	type	tlevel	parent	shardBy	cols	size	unenforced
1	CUSTOMER	S	1		HASH	C_CUSTKEY	44	CUSTOMERFK
1	ORDERS	S	2	CUSTOMER	REFERENCE	ORDERSFK	289	
1	LINEITEM	S	3	ORDERS	REFERENCE	LINEITEMFK1	1472	LINEITEMFK2
1	NATION	D			NONE		1	
1	PART	D			NONE		43945	
1	PARTSUPP	D			NONE		23340	
1	REGION	D			NONE		1	
1	SUPPLIER	D			NONE		260	

Terraform Script Input File

```
shards = {  
  "shard-1" = {  
    host = "den02ffv"  
    port = "1521"  
    sid = "sh1"  
    globalDBName = "sh1"  
    shard_group = "primary_shardgroup"  
  },  
  "shard-2" = {  
    host = "den02ffw"  
    port = "1521"  
    sid = "sh2"  
    globalDBName = "sh2"  
    shard_group = "primary_shardgroup"  
  }  
}
```

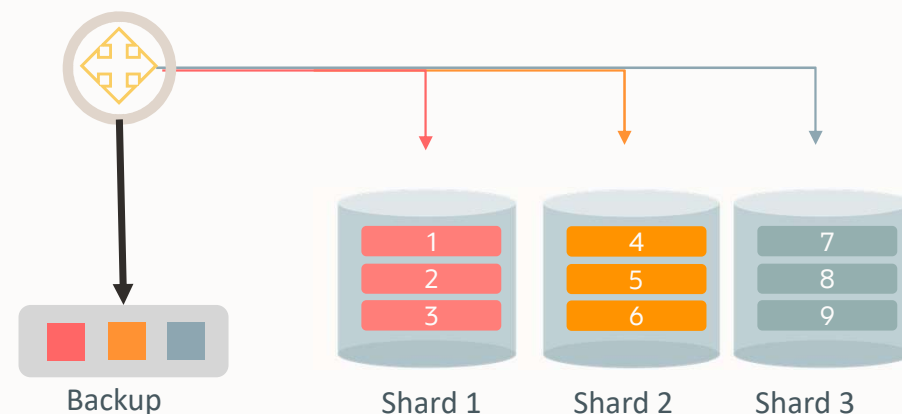


集中备份恢复

Centralized Backup Recovery

NEW IN
21C

- 分片数据库的备份/恢复自动化 Backup/recovery automation of sharded database
 - 整个分片数据库的一致备份 Consistent backup of entire sharded database
 - chunk、单个分片或整个分片数据库的恢复粒度 Recovery granularity at chunk, single shard or entire sharded database
- 备份目标包括磁盘（对象存储和ZDLRA在计划中） Backup destinations include disk(object store and ZDLRA on roadmap)
- 了解结构变化的操作，如chunk移动 Operations aware of structural changes like chunk move
- 使用GDSCTL进行管理 Management using GDSCTL



```
GDSCTL> RUN BACKUP -sync -shard ALL;
```

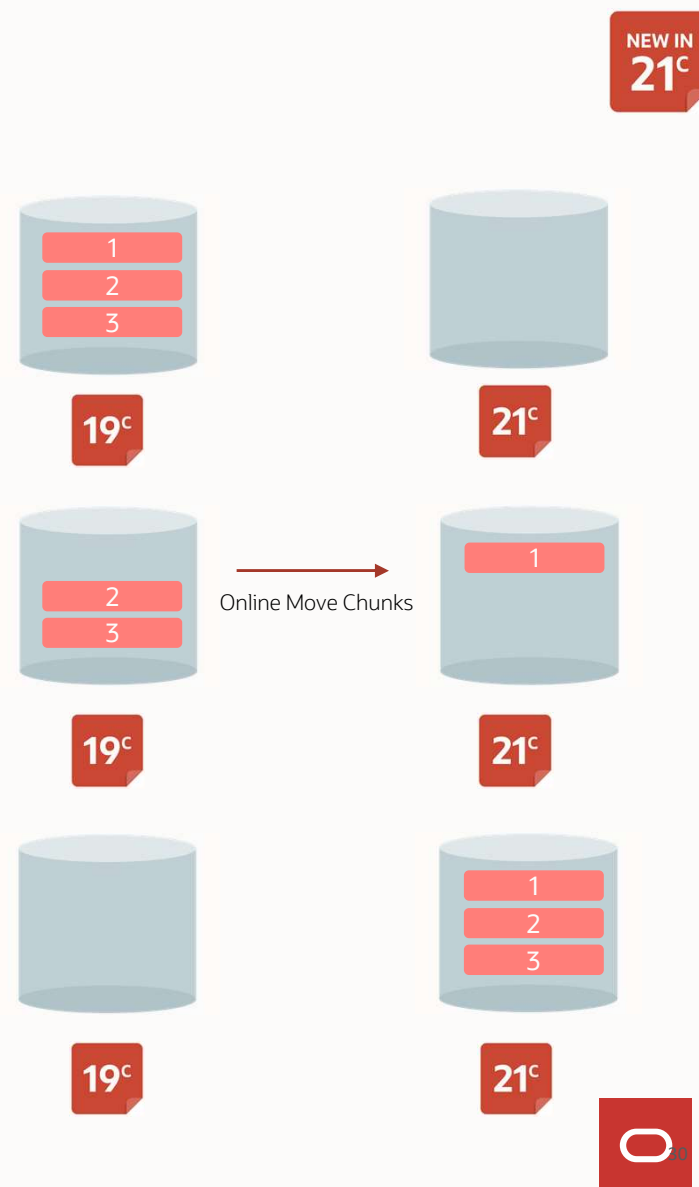
```
GDSCTL> RESTORE BACKUP -restorepoint  
grp010119000130 -controlfile -shard shard1 -  
sync;
```



使用切片替换进行升级

Upgrade using Shard Replacement

- 对于主要和次要升级 For major and minor upgrade
- 添加一个空分片 Add an empty shard
- 将chunks从旧版本分片移动到新分片 Move chunks from old version shard to new shard
- 停用旧版本分片 Decommission the old version shard

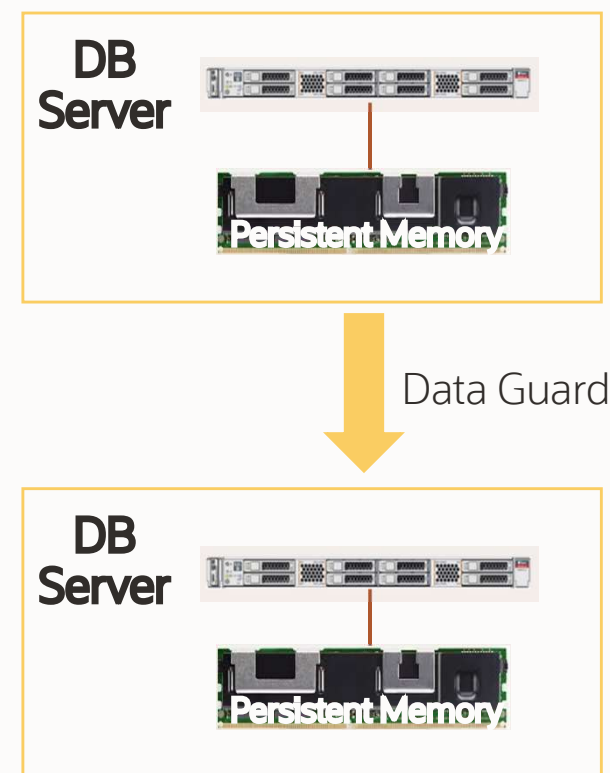


支持持久内存中的数据库

Support for Databases in Persistent Memory

NEW IN
21C

- 原生的持久化内存将数据库数据和redo保留在本地永久性存储器 (PMEM) 中 Native Persistent Memory stores database data and redo in local Persistent Memory (PMEM)
 - 对于需要比闪存更低延迟的工作负载 For workloads that need lower latency than flash memory
- SQL直接在访问的持久化内存存储文件系统中存储的数据 SQL runs directly on data stored in mapped Persistent Memory file system
 - 消除了IO代码路径，无需大型缓冲区高速缓存 Eliminates IO code path, no need for large buffer cache
 - 新的数据库算法可防止将部分或不一致的数据存储到持久性存储器 New database algorithms prevent partial or inconsistent stores to Persistent Memory
- 需要Data Guard来防止服务器故障或持久化内存故障 Requires Data Guard for protection from server failures or persistent memory failures
- 横向扩展使用原生的数据库 Sharding Scale-out using Native Database Sharding



反应流摄取库

Reactive Streams Ingestion Library

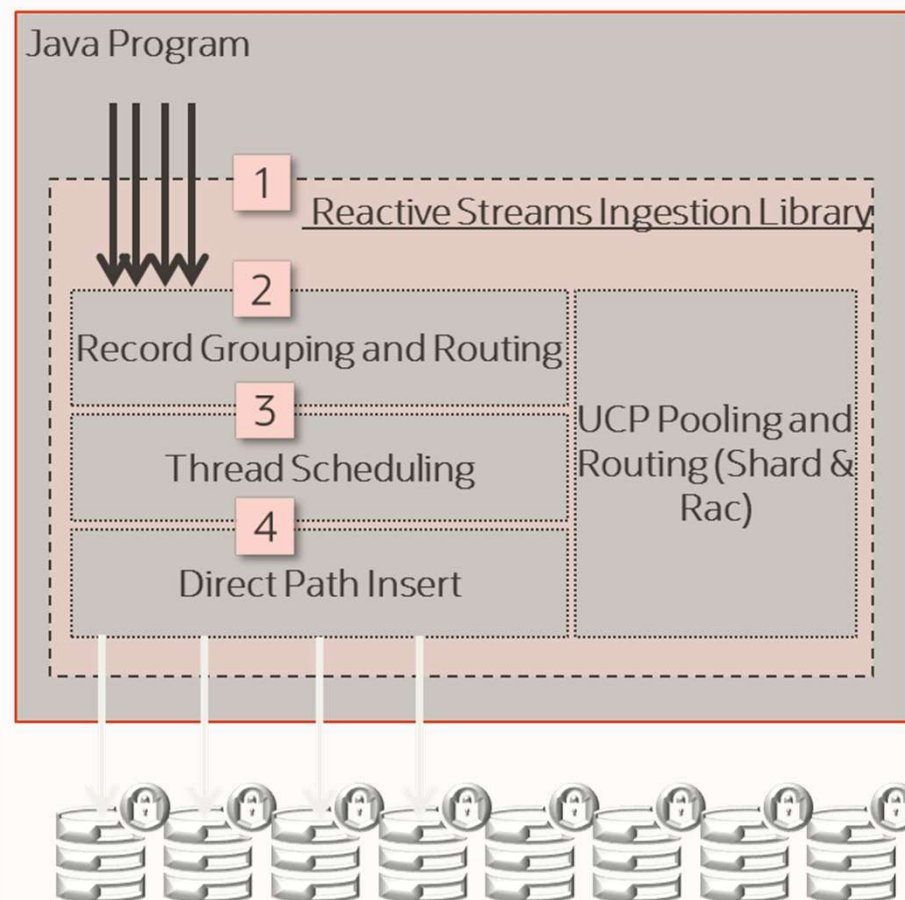
NEW IN
21^C

流功能：从大量客户端接收数据而不阻塞
Streaming capability: receive data from a large group of clients without blocking

使用原生 UCP 根据分片关联对记录进行分组
Group records according to Shard affinity using native UCP.

优化 CPU 分配，同时将记录处理与 IO 解耦
Optimize CPU allocation while decoupling record processing from IO.

通过直接路径插入、绕过SQL并直接写入数据库文件，实现Oracle数据库的最快插入方法。
Fastest insert method for the Oracle Database through Direct Path Insert, bypassing SQL and writing directly in the Database files.



总结 Summary

- Oracle Sharding 支持超大规模全球分布式数据库 Oracle Sharding enables Hyperscale Globally Distributed Database
- 它可以用于结构化和非结构化数据 It can be used for both structured and unstructured data
- 它可以用于OLTP和分析工作负载 It can be used for both OLTP and Analytical workloads
- 它是数据库企业版的一部分 It is included as a part of Database Enterprise Edition
- 你可能已经有了它，所以使用它吧 You probably already have it, so use it



资源 Resources

<https://www.oracle.com/goto/oraclesharding>

<https://docs.oracle.com/en/database/oracle/oracle-database/19/shard/index.html>

<https://github.com/oracle/db-sharding>

“甲骨文云技术”公众号文章：

<https://mp.weixin.qq.com/s/hll1DxchuUE4KYHMdHpN9w>

https://mp.weixin.qq.com/s/yQ_pwh235Mg_n73LO4EL5Q

Contact

Product Management – shailesh.dwivedi@oracle.com



Thank you
