DTCC 2010-2022

第十三届中国数据库技术大会

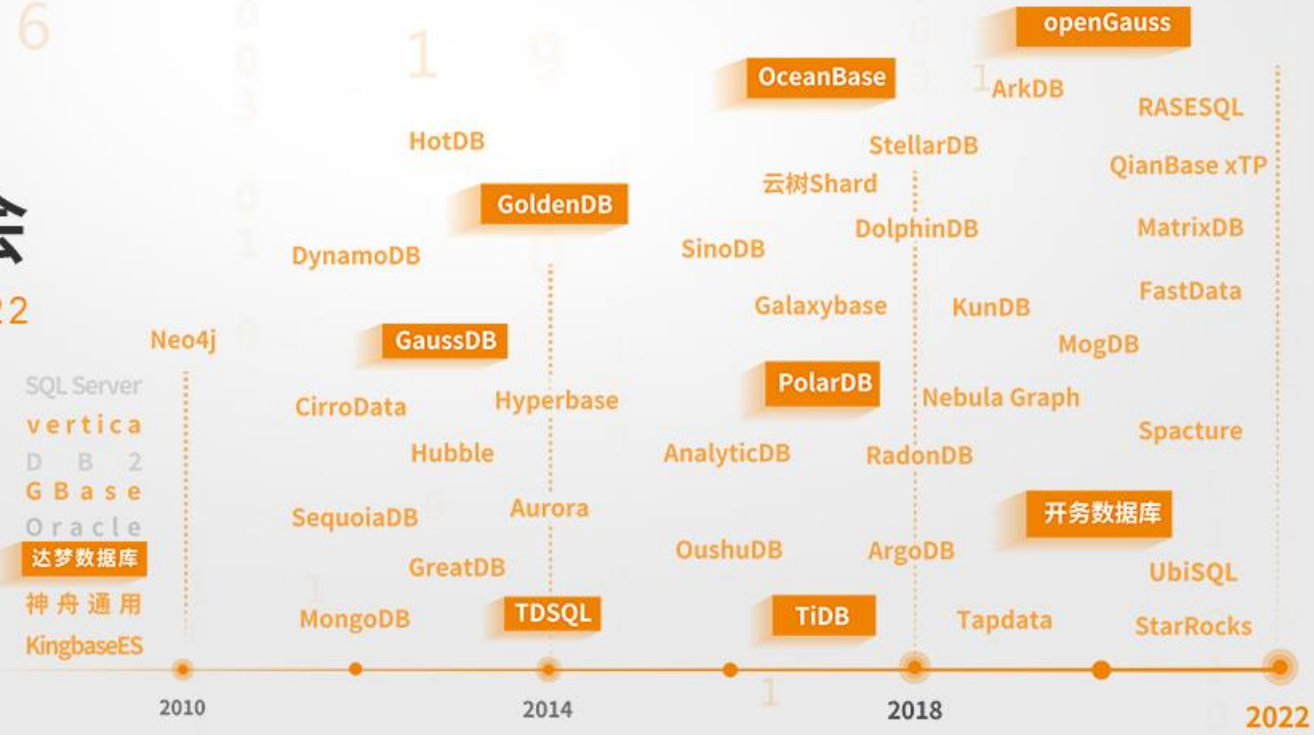
DATABASE TECHNOLOGY CONFERENCE CHINA 2022

数据智能 价值创新

IT168.com ChinaUnix.net ITPUB

线上直播 | 2022/12/14-16

数据来源：数据库产品上市商用时间



多region分布式数据库方案与实践

赵飞祥

Airwallex 空中云汇

赵飞祥

Airwallex空中云汇 数据架构师

曾就职于太极计算机、北京竞技世界网络技术有限公司、斗鱼等企业。Oracle 10g OCP，11g OCM，Oracle YEP年轻专家。喜爱技术总结和分享、多次行业会议和沙龙演讲嘉宾、IT Pub博客专家。2010年开始从事数据库相关运维、架构、开发工作，涉足postgresql、 mysql、Oracle、greenplum、 MongoDB、redis等数据库，目前主要研究PostgreSQL数据库和DevOps方向

个人博客： http://blog.itpub.net/24638123/

目录

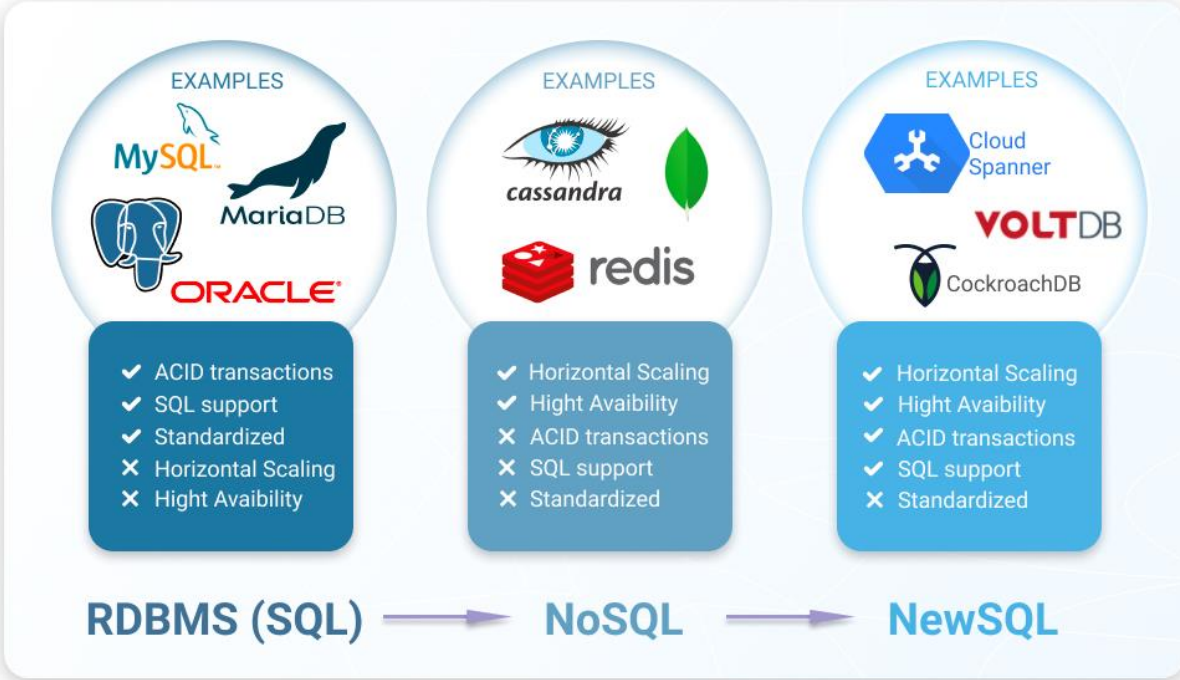
1.分布式数据库解决的问题
2.分布式数据库的常见方案
3.分布式数据库的实现原理
4.分布式数据库的实践经验

1.分布式数据库解决的问题

数据库发展脉络

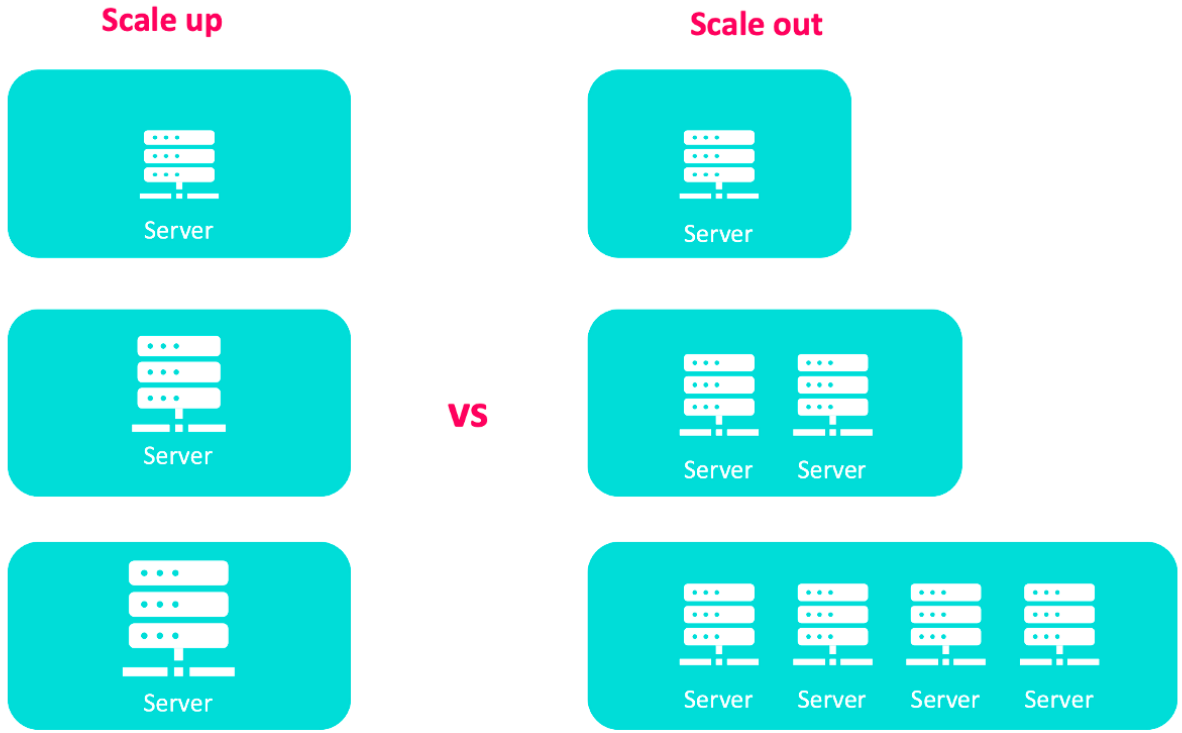
数据库技术发展三个阶段

- 关系型数据库 SQL
- 非关系型数据库 No-SQL
- 新型数据库 New-SQL



数据库水平扩展

- 垂直扩展与水平扩展
- 分布式算法应用到DB
- 可以实现的特性
  - 更高可用性和健壮性
  - 可扩展性
  - Rolling update
  - Rolling upgrade



多region数据写的业务需求

- 多region和单region的业务需求
- 数据库多点写入的意义
- 多 region/点 写是分部式数据库区别单体数据库的重要特征

2.分布式数据库常见方案

多节点数据写入的实现层面

多节点写入的不同实现：

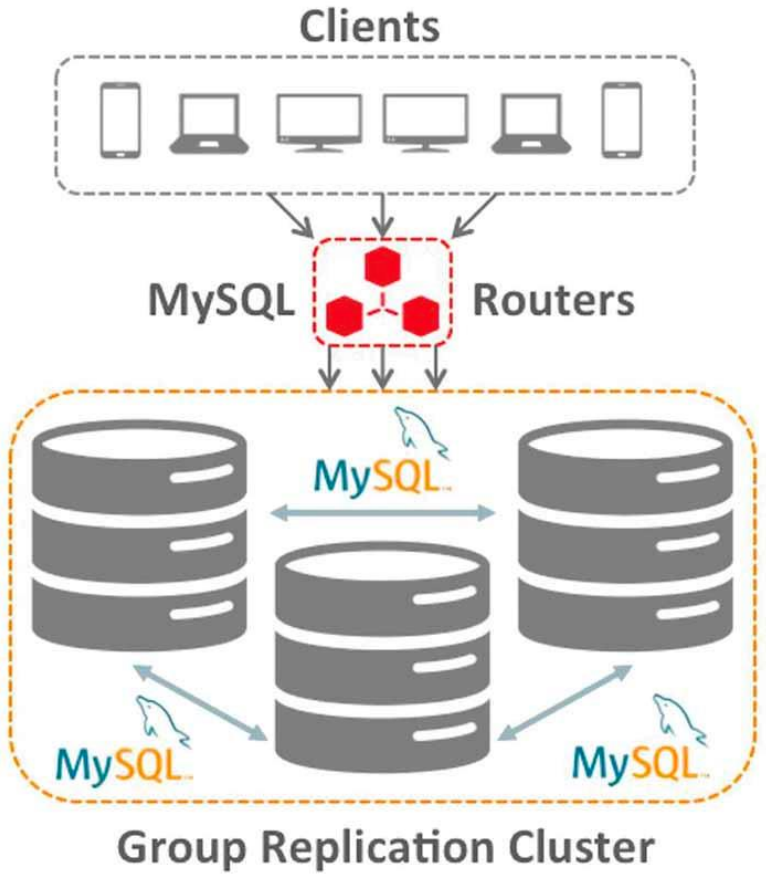
- 业务层：程序控制多节点写入
- 代理层：控制多节点写入
- 数据库层：实现多节点写入

单region的数据多点写入实现

- MySQL MGR
- MySQL innodb cluster

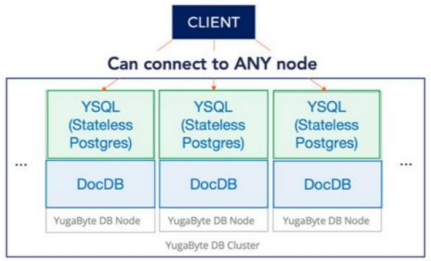
- Cloud provider distributed database

- Cochroach DB
- Yugabyte DB



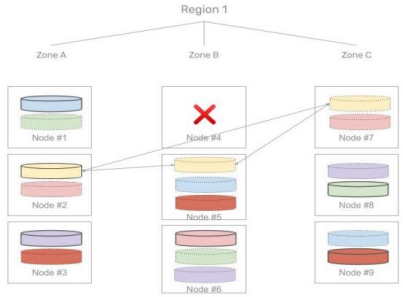
选择YugabyteDB作为分布式数据库的主要原因

PostgreSQL Compatible



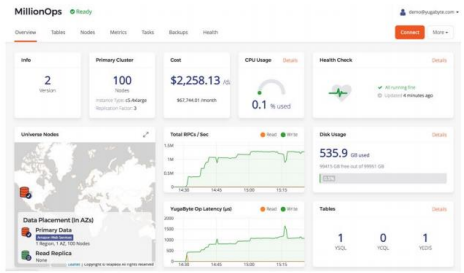
- Reuses the query layer of PostgreSQL
- Offers advanced features (triggers, stored procedures, partial indexes, security, etc)
- Build / modernize apps with high agility
- Developers get instantly productive
- Migrate legacy DBs to the cloud (Oracle / RAC, SQL Server, DB2)
- Example: A large bank migrated from DB2, Justuno from SQL Server

Resilience & HA



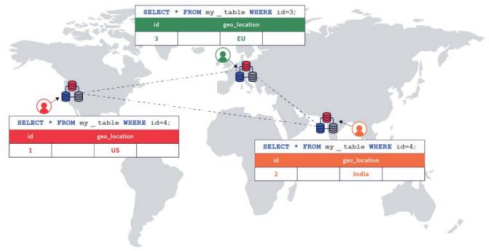
- Survives multiple failures - nodes, zones, regions/data centers
- Auto-heals with re-replication
- Zero-downtime upgrades & security patching
- Proven in real-world scenarios at scale
- Example: Sep'19 Plume continued handling billions ops/day during AWS outage of EU zone. Multiple rolling upgrades and cluster expansions

Horizontal Scalability



- Billions of ops/day, hundreds of TB
- Scale queries, storage and connections by adding nodes
- Reliably scale-out and scale-in on demand with large datasets without impacting the running application
- Example: Kroger and Narvar scale-out YugabyteDB to handle peak traffic (Black Friday and Cyber Monday), can scale-in after holiday season

Geo-Distribution

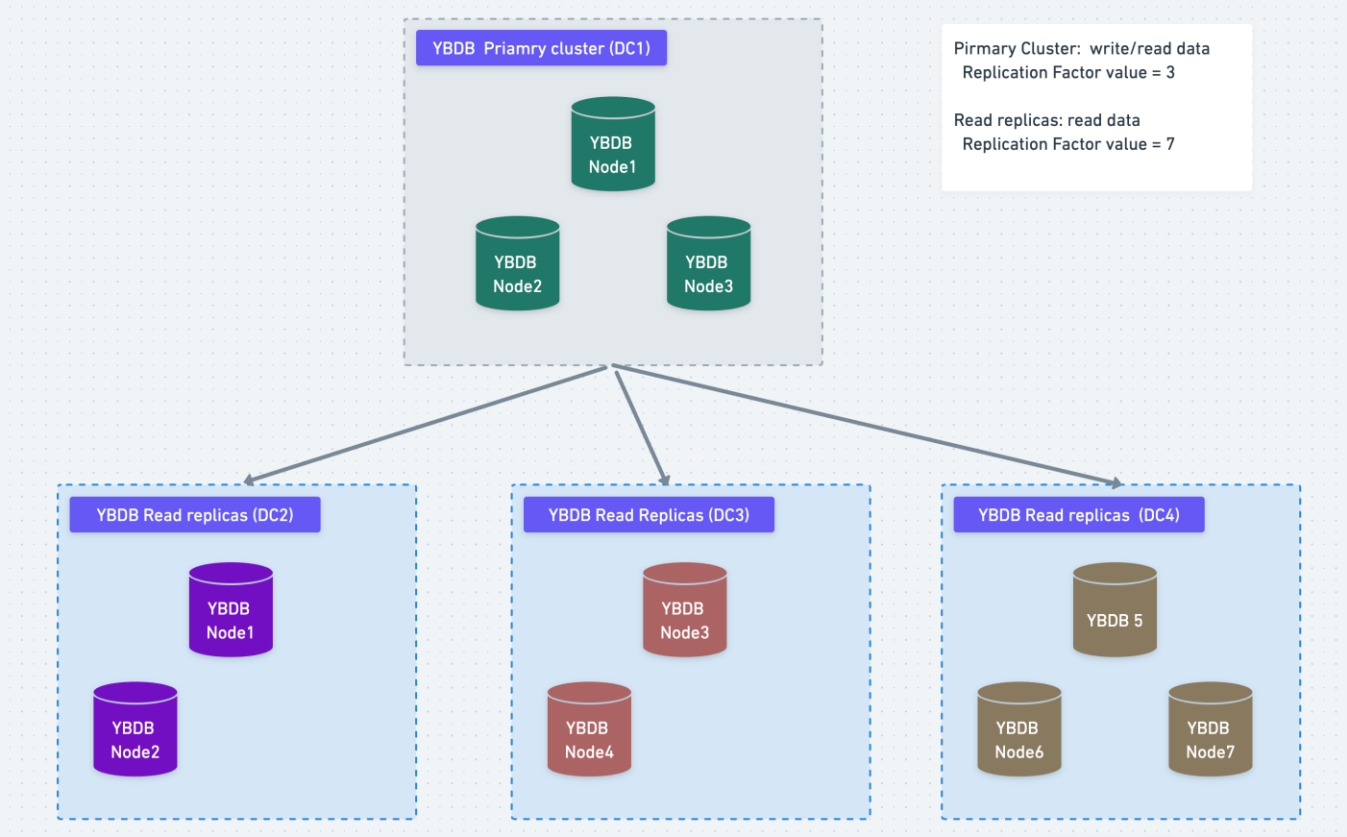


- Distribute data across zones, regions or clouds with ACID consistency
- Most complete global replication features of any database: sync, async, geo-partitioning, hybrid deployments
- Move data closer to customers and comply with regulations like GDPR
- Example: *Admiral geo-distributed data across 5 regions and 3 continents to improve user experience by achieving < 3ms read latencies*

Yugabyte DB单region多写架构

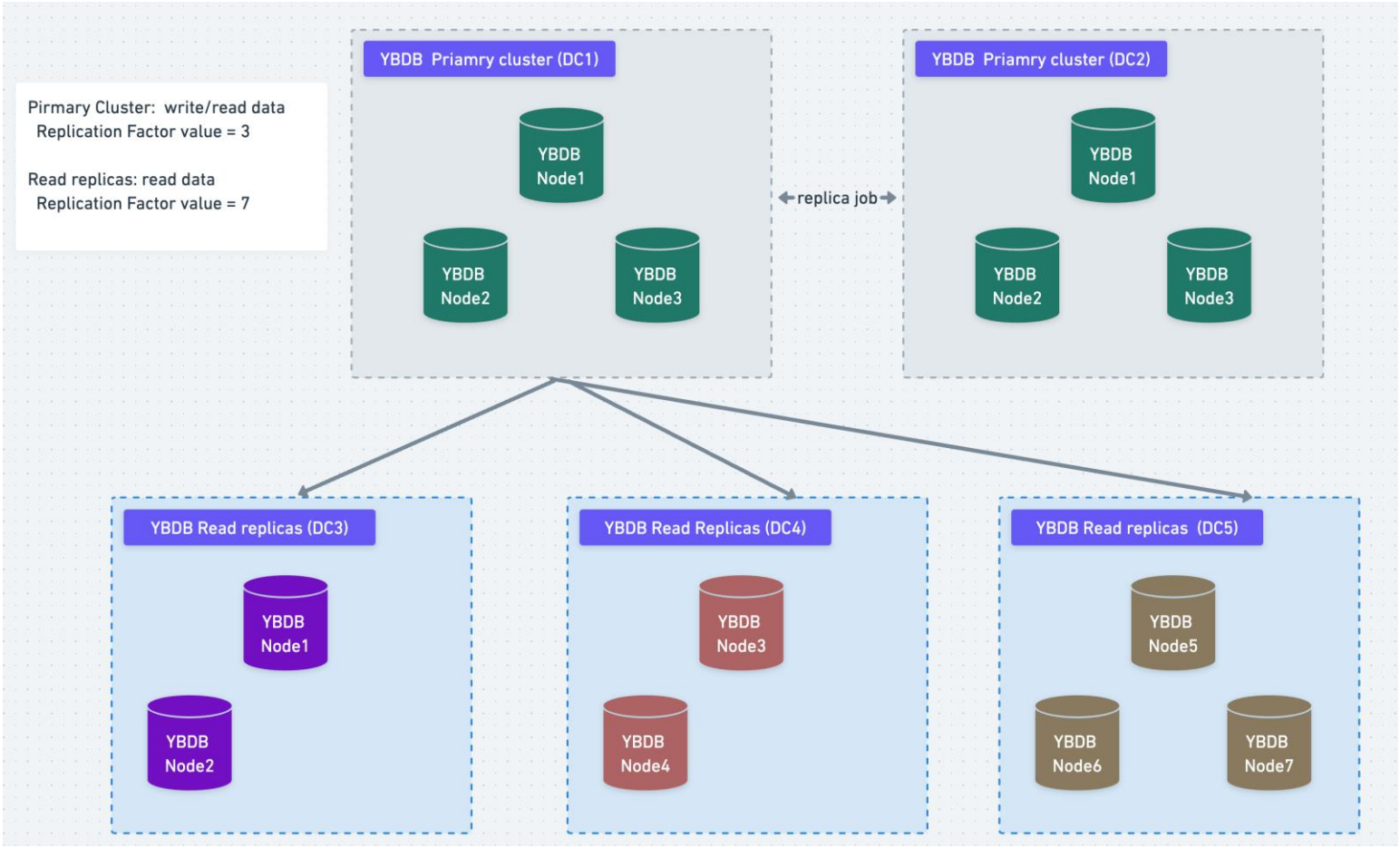
Yugabyte DB Universe

- Yugabyte Primary Cluster
- Yugabyte Read Replica



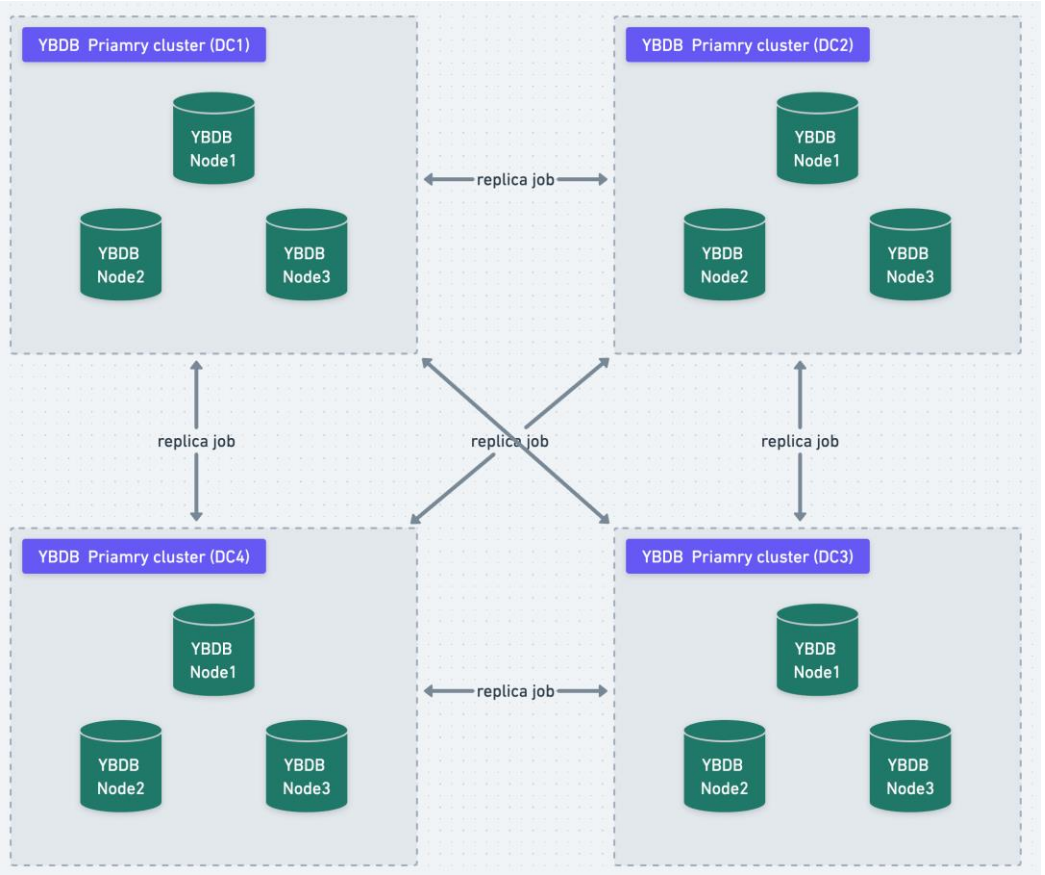
Yugabyte DB多region架构之 xCluster

- Primary Cluster
- Replica async job

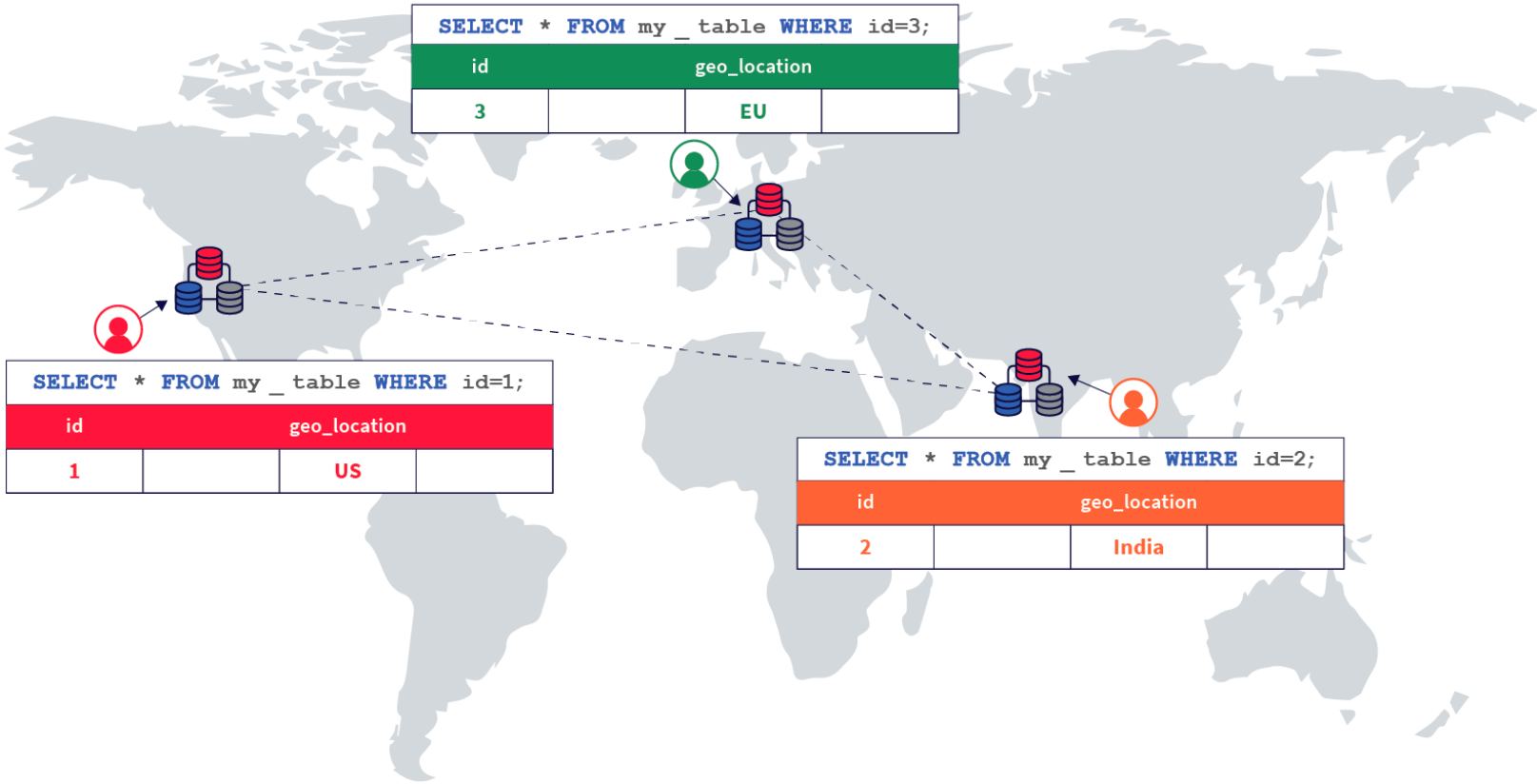


Yugabyte DB多region架构之 xCluster

- 1:1 xCluster
- 1:n xCluster
- n:n xCluster



Yugabyte DB多region架构之 GEO-parition cluster



Yugabyte DB多region架构之 GEO-parition cluster

- 可以在多个region部署，能够满足 ACID一致性的global集群
- 每个table都要geo-location列，用于定位数据存放和查找地
- 每个表都是按照geo-location列进行list分区的分布表
- 不同region的数据，可以存放在不同的region，满足法规要求
- region级别的事务表和元数据下推，避免跨region事务
- 如果有跨region查询，可以增加 read replica 进行 local read

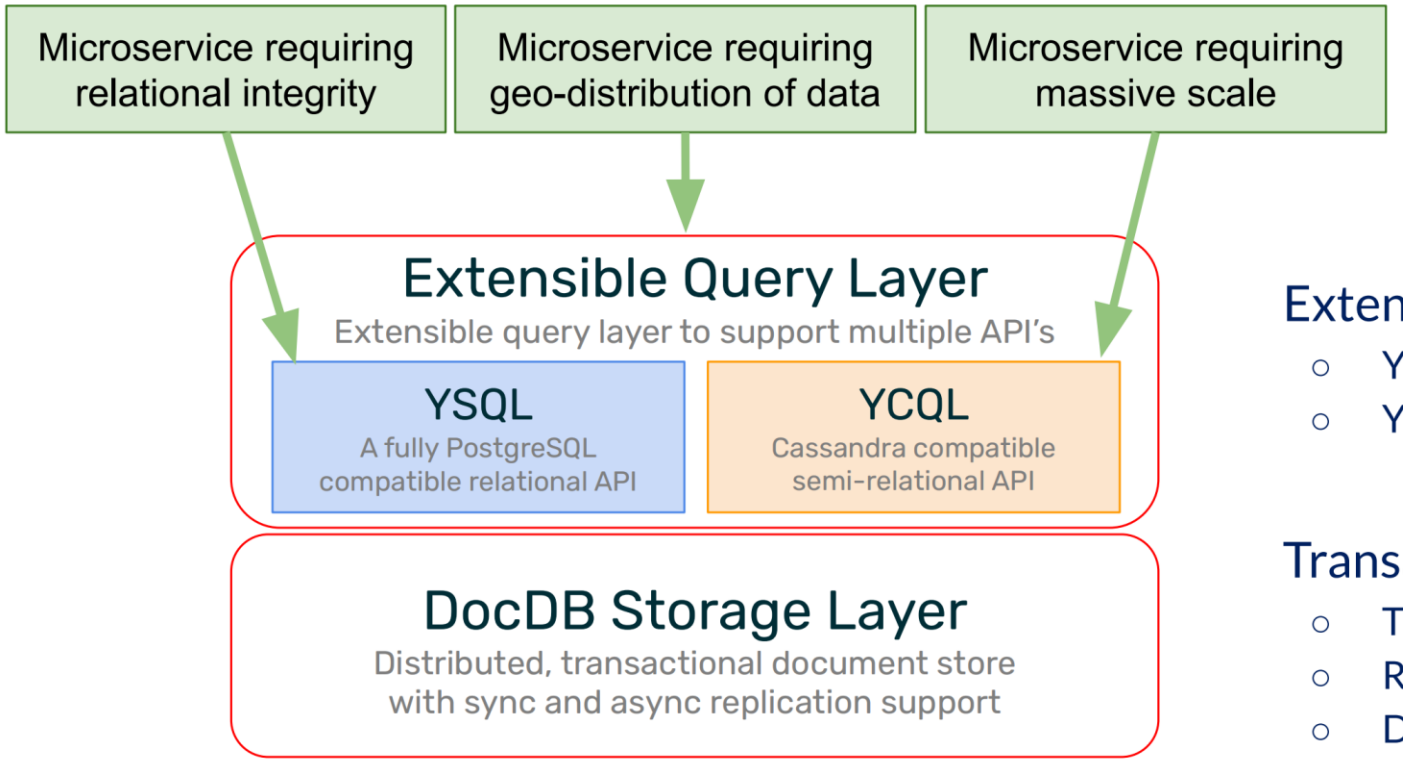
YugabyteDB 分布式数据库常用架构

三种常见架构：

- Primary cluster + Read Replica
  - primary部署单region，主要实现高可用
  - primary部署多region，主要实现 DR
- YugabyteDB xCluster
  - 多个region都是独立的 Primary cluster, 稳定性和性能好，注意数据冲突
- YugabyteDB GEO-parition Cluster
  - 多个region是一个大的primary cluster， 表是分区表，便于数据主权隔离

3.分布式数据库实现原理

分布式数据库与单体数据库架构区别



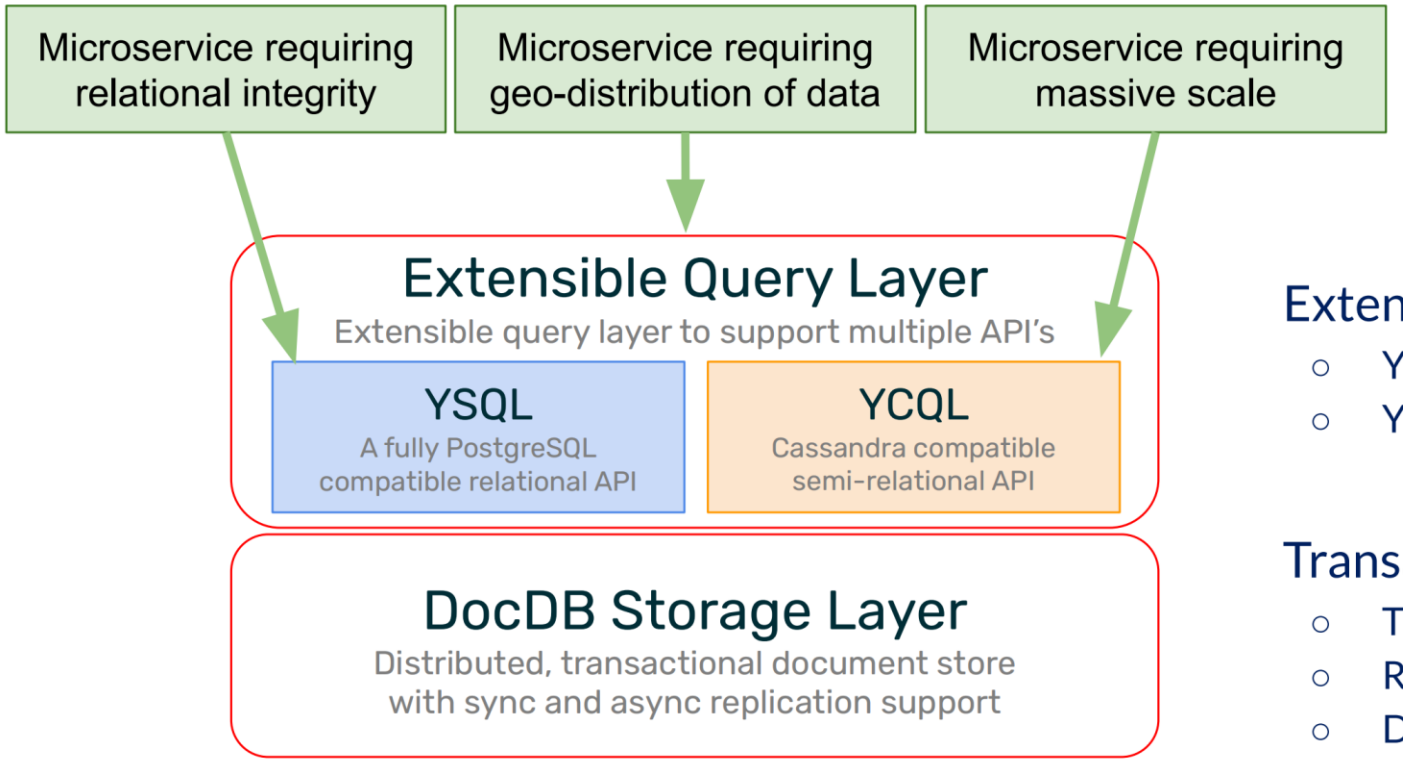
Extensible query layer

- YSQL: PostgreSQL-based
- YCQL: Cassandra-based

Transactional storage layer

- Transactional
- Resilient and scalable
- Document storage

YugabyteDB分层数据库架构



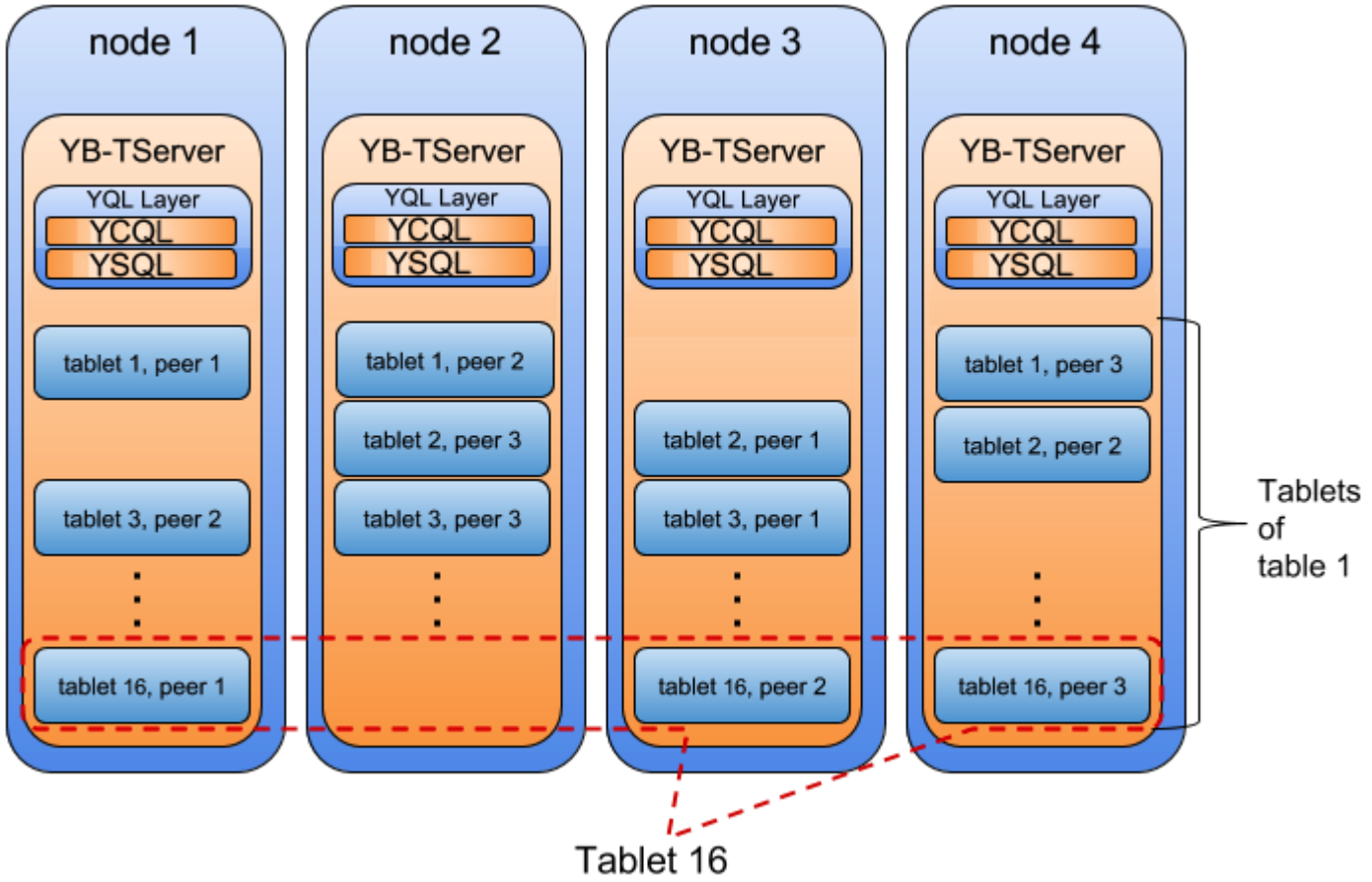
Extensible query layer

- YSQL: PostgreSQL-based
- YCQL: Cassandra-based

Transactional storage layer

- Transactional
- Resilient and scalable
- Document storage

YugabyteDB Universe之 YB-TServer Service



YugabyteDB Universe 之 YB-TSaster Service

YB Tablet Server的角色和功能：

- 将PG中的table再次分片成更小的 tablet，并按照 replciation factor 实现 raft 协议同步和调度
- tablet分片由YugabyteDB自动实现和完成，对应用透明，所有表必须由主键，且在第一次创建表时，就要创建好
- tablet也具有高可用，是由很多个 raft协议组实现
- tablet 的分片、存储、查询是分布式事务和分布式查询的基础

YugabyteDB Universe 之 YB-Master Service

YB master的角色和功能：

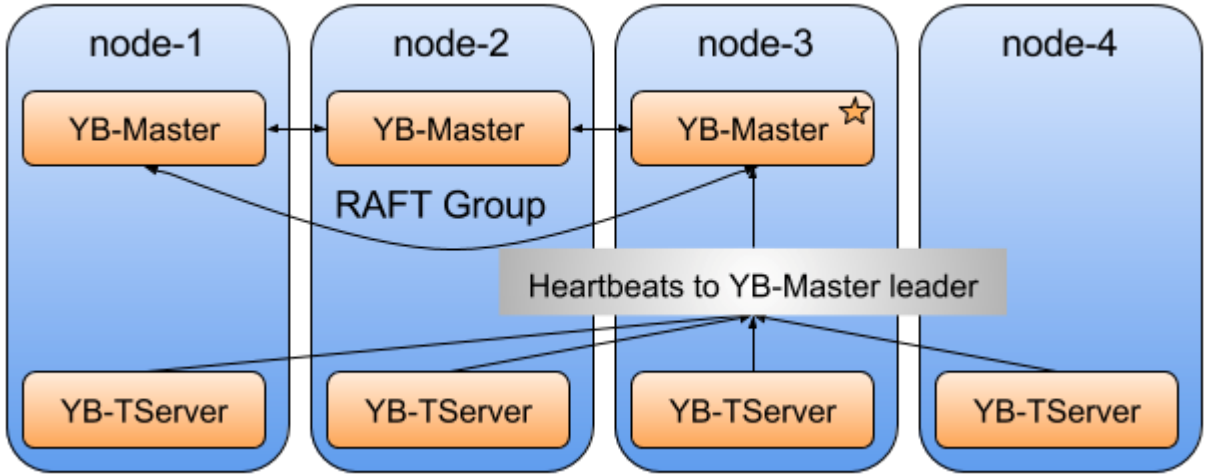
- 记录系统的元数据和记录

  表，tablet的位置，角色和权限

- 调度后台操作，执行管理操作

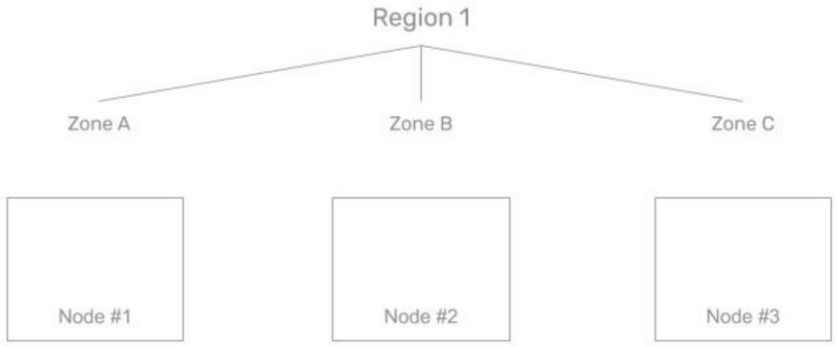
  负载均衡，启动复制，表DDL

- 具有高可用，是一个 raft协议组

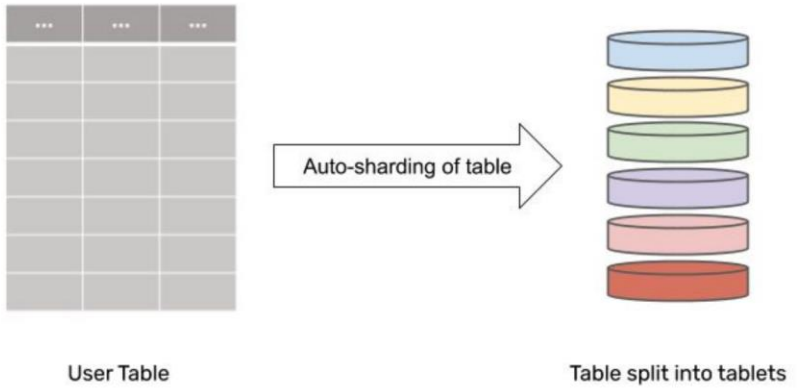


分布式数据库数据库存储

Distributing Data For Horizontal Scalability



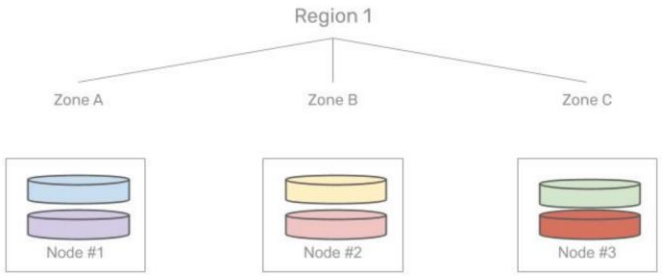
- Assume 3-nodes across zones
- How to distribute data across nodes?



- User tables sharded into tablets
- Tablet = group of rows
- Sharding is transparent to user

分布式数据库数据库存储

Distributing Data Across Nodes, Zones, Regions

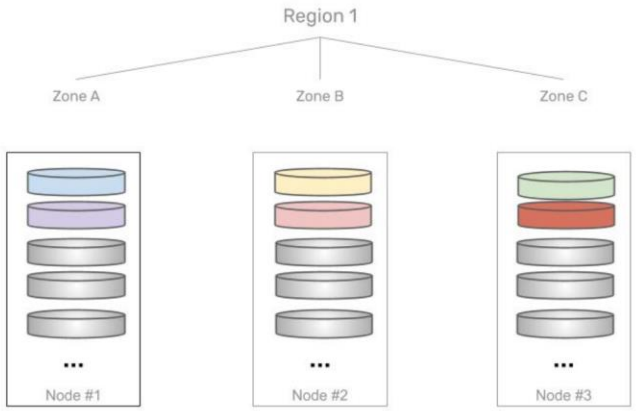


Tablets (per-table, across tables)

evenly distributed across nodes

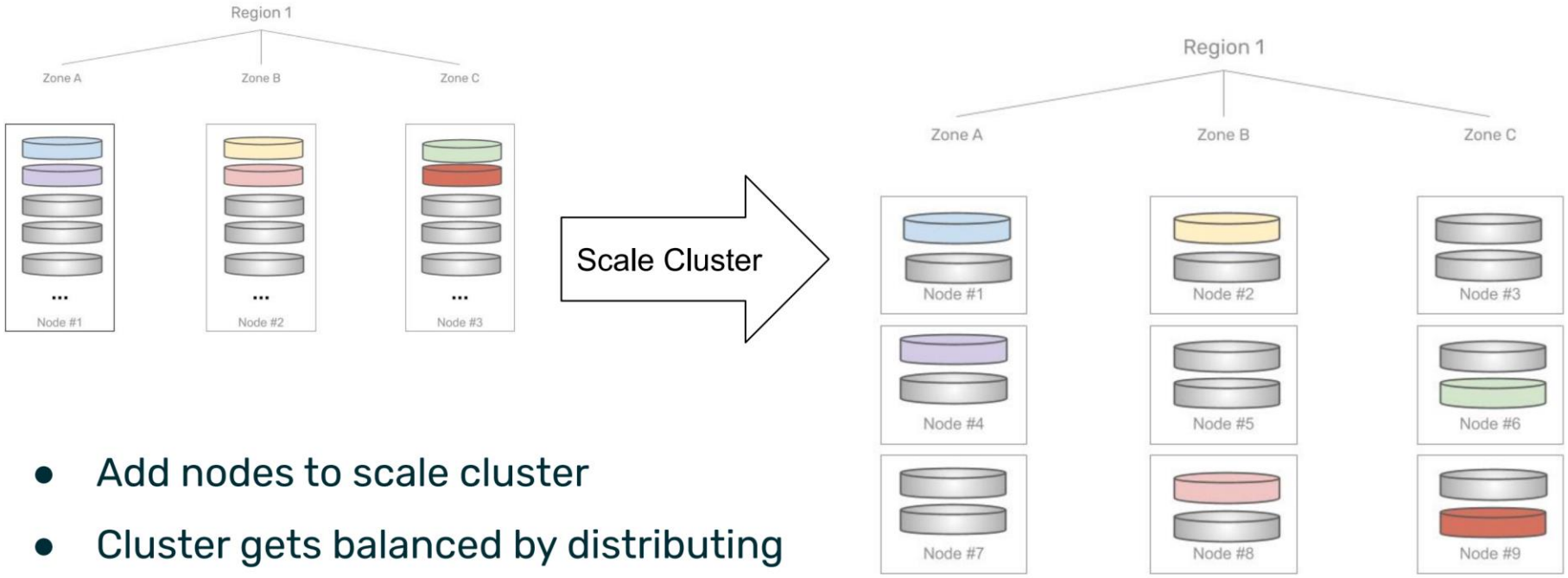
In real deployments,

many tablets per node



分布式数据库数据库存储

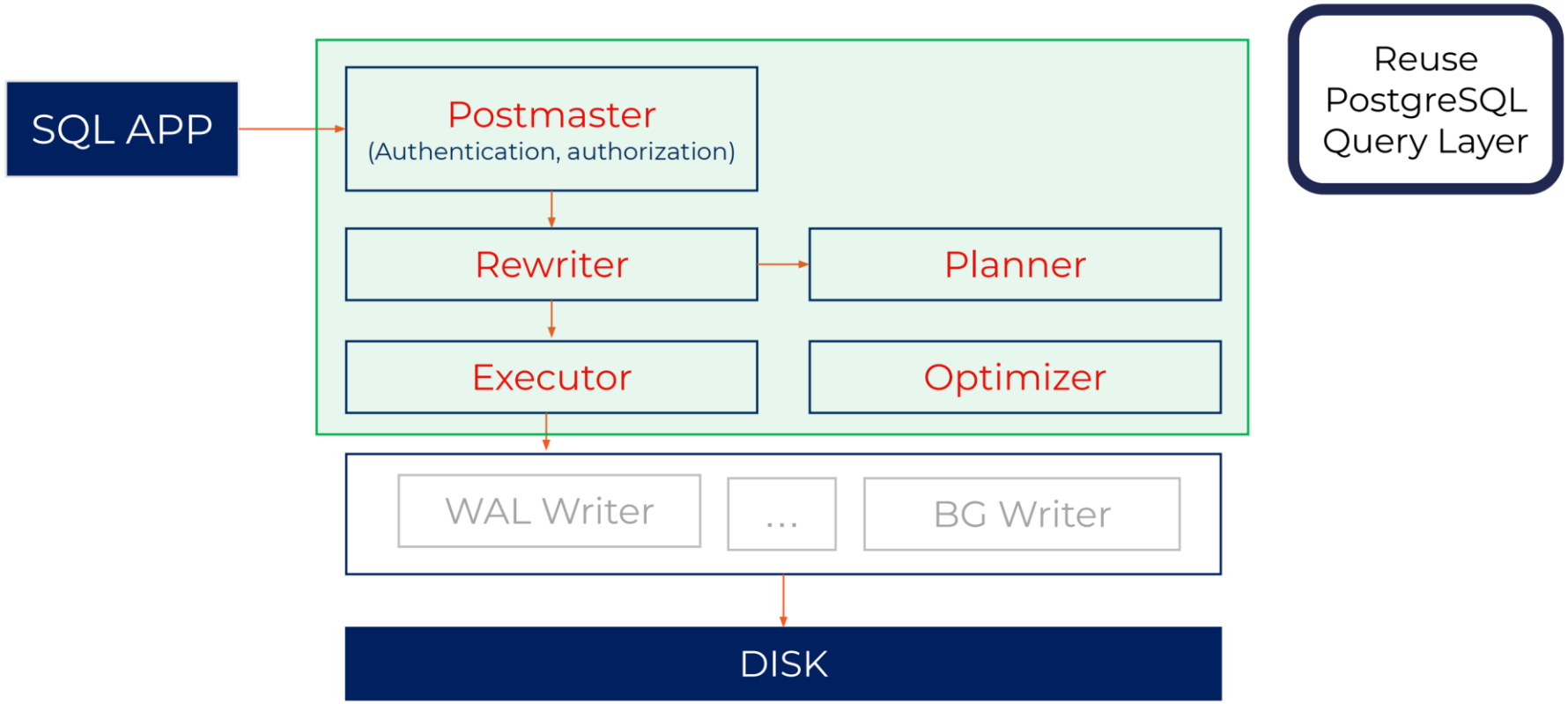
How Horizontal Scalability Works



- Add nodes to scale cluster
- Cluster gets balanced by distributing existing tablets to new nodes

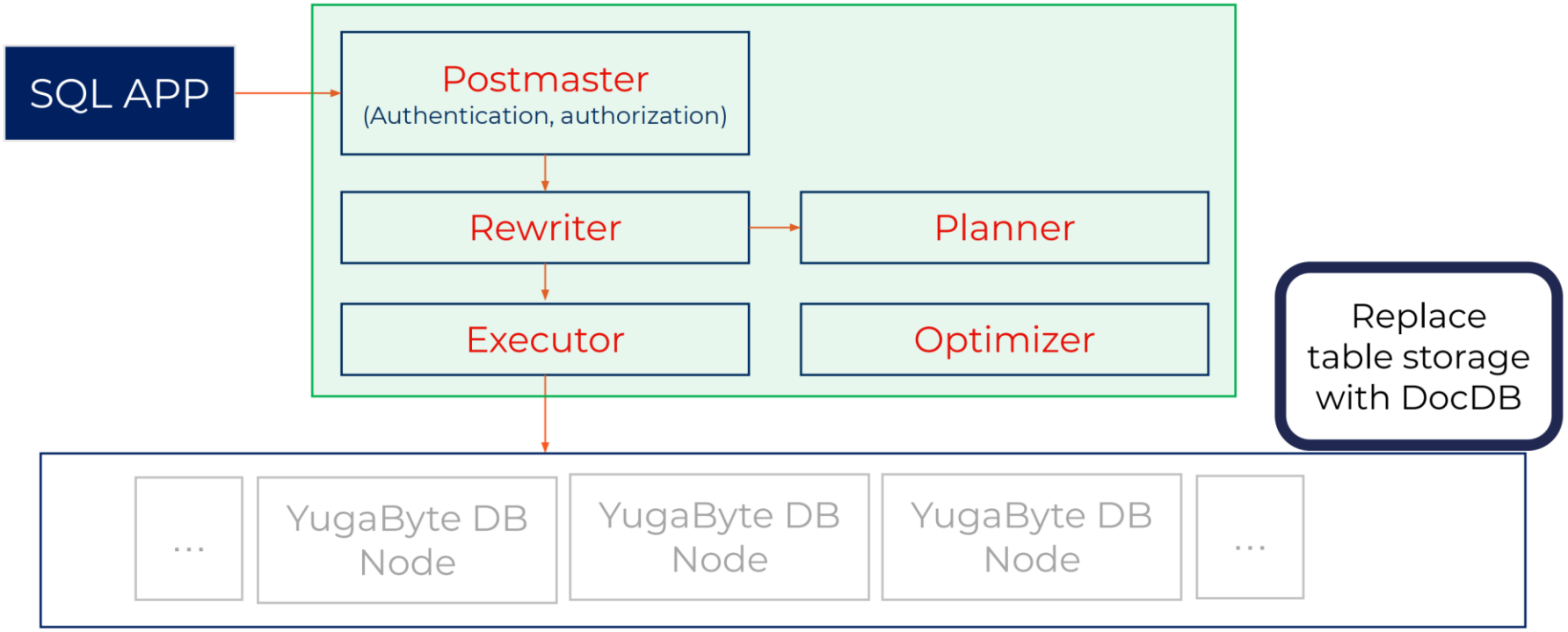
分布式数据库数据库查询

Existing PostgreSQL Architecture



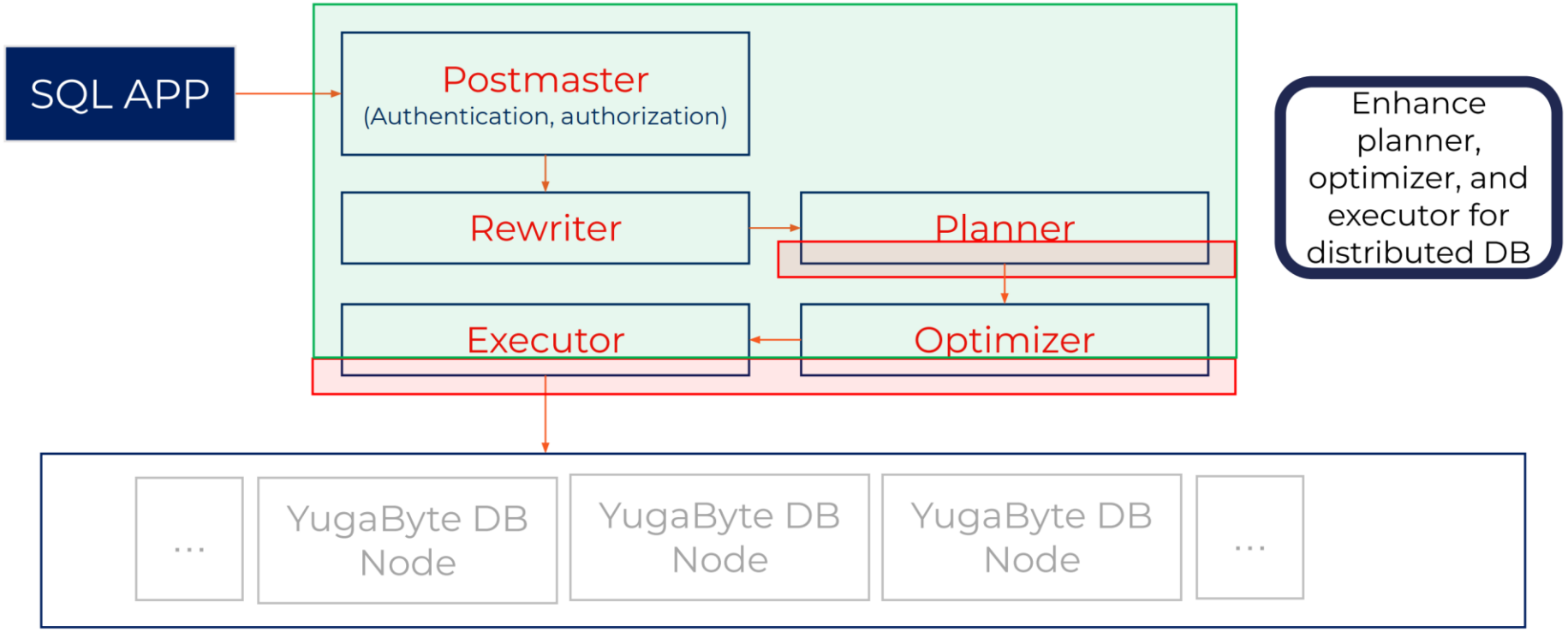
分布式数据库数据库查询

DocDB as Storage Engine



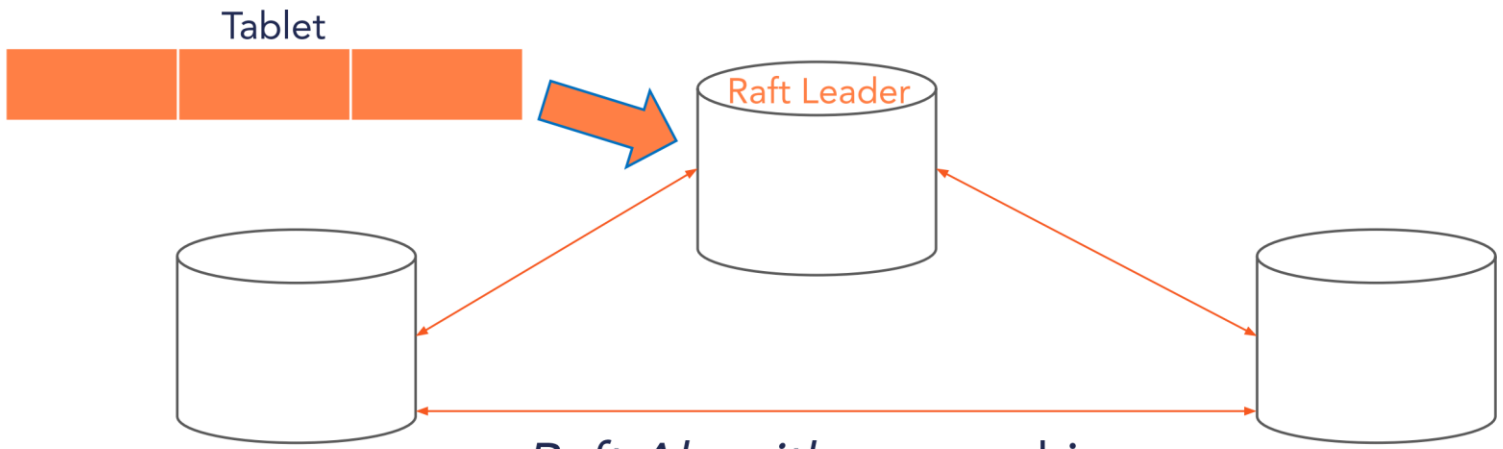
分布式数据库数据库查询

Make PostgreSQL Run on Distributed Store



分布式数据库分布式算法

Replication uses Raft Consensus algorithm



Raft Algorithm can achieve
per-row consistency across nodes

On failure, HA achieved because
new leader elected quickly

分布式数据库特殊功能

Isolation Levels

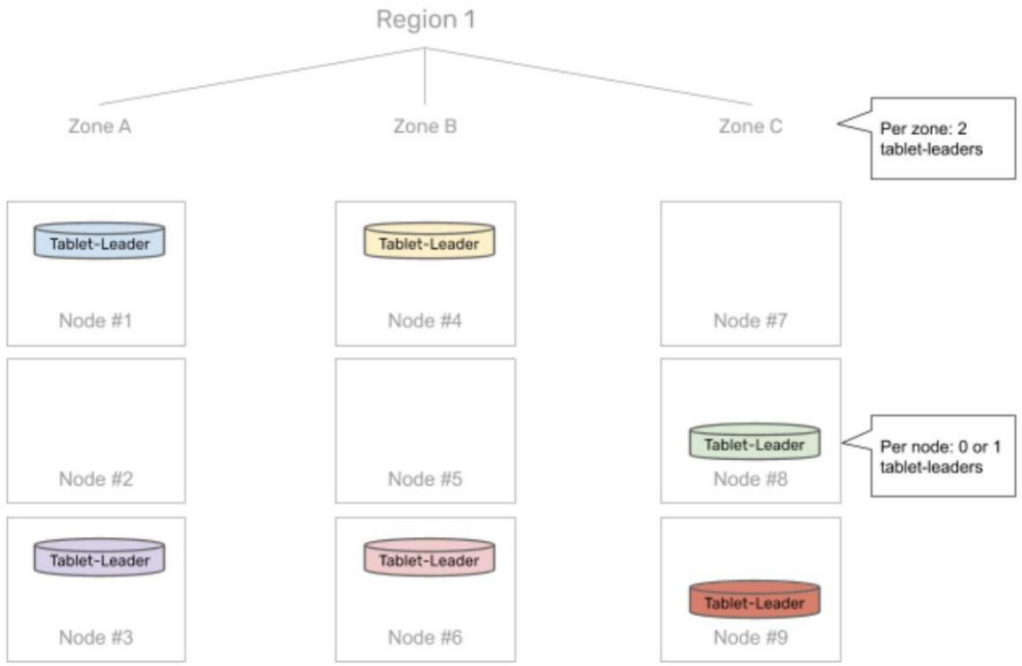
- **Serializable Isolation**
  - Read-write conflicts get auto-detected
  - Both reads and writes in read-write txns need provisional records
  - Maps to SERIALIZABLE in PostgreSQL

- **Snapshot Isolation**
  - Write-write conflicts get auto-detected
  - Only writes in read-write txns need provisional records
  - Maps to REPEATABLE READ, READ COMMITTED & READ UNCOMMITTED in PostgreSQL

- **Read-only Transactions**
  - Lock free

分布式数据库特殊功能

Balancing across zones and regions

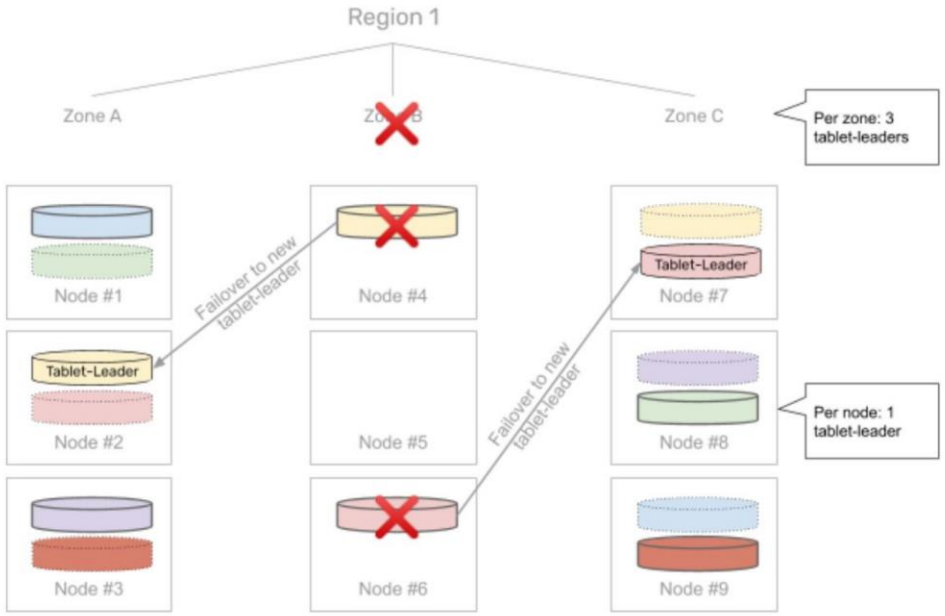


Tablet leaders balanced across:

- Zones
- Nodes in a zone
- Per-table balancing

分布式数据库特殊功能

Automatic rebalancing



- New leaders evenly rebalanced
- After 15 mins, data is re-replicated (if possible)
- On failed node recovery, automatically catch up

4.分布式数据库的实践经验

方向确定与选型测试

- 从最核心的需求确定分布式数据库的方向
  - 现有数据库全部是PG，必须对PG由充分的兼容性，服务可以在PG和分布式数据库之间切换
  - 必须能够实现数据强一致性，保证金融业务数据不丢失，不出错
  - 在性能上必须满足现有业务和后续业务发展的性能要求
  - 在功能上必须能实现分布式数据库的高可用，可扩展等功能
- 在明确目标的情况下，进行充分测试，最后确定选型
  - 找到国内国外所有市面上可用的分布式数据库
  - 筛选出基本符合要求的
  - 通过详细测试，对比，最后做出选择

架构方案和设计与选择

- 分布式数据库通常有多种部署方式和架构，架构选择是部署的方向，需要花足够的时间，考虑和论证足够的方面
- 可以列出所有可以实现的架构，找出每种架构的优缺点和适用常见
- 可以从具体业务需求出发，反向看一个或多个业务架构组合，可以满足具体的业务需求
- 将所有的架构都进行了充分的思考和论证后，结合业务需求，列出少量几种架构，纲举目张
- 每种架构都是适合场景和使用要求，可以追求，但不要执着完美的架构

细节验证与重点指标

- 细节验证是对架构方向确定后的部署和具体实现，是架构设计后的后续延伸
- 在实现时要特别注意细节，一个小的参数，或者一行测试数据，就可能对整个方向产生影响
- 分布式数据库的功能很多，业务场景也很多，需要区分重点指标，比如分布式数据库的性能指标，独特功能的验证需要多验证
- 一些相对常规的内容，比如YugabyteDB对PG数据库的兼容性，可以相对快速验证
- 有时验证也不是一蹴而就，一次运行就可以得到预期的结果，需要反复思考和推敲，完成验证过程，最终得到合理的结果

逐步上线与迁移细节

- 新技术，新架构的应用，最好先选择相对重要性较低的服务进行测试和上线，避免新技术对核心业务产生较大影响
- 业务上线应该在联调环境充分测试，验证所有方面都符合预期，满足prod上线要求，才能考虑prod环境上线
- 考虑prod环境线上，要遵循开发环境-联调环境-演示环境-生产环境的步骤以此上线，每个阶段上线，都要留有足够的时间观察状态
- 在迁移之前，先准备好详细的迁移步骤和注意事项，并在开发和联调环境迁移时进行确认和补充，保证线上迁移时全部都在预期内
- 一些迁移时的特殊操作，和特殊迁移事项，可以特殊记录和注意

持续迭代与应用实现

- 分布式数据库是一个还在发展和不断成熟的技术，它虽然带来SQL和NoSQL所不具备的特性，但也必然存在一定的逐步成熟的地方
- 不论分布式数据本身的特性，还是应用场景的需求变化，都是一个不论完善，不断找寻到更优解的过程
- 要以发展的眼光和开放的心态来看待和处理分布式数据库的现状和问题，能解决现有问题，也能满足未来发展需要
- YugabyteDB 中的 smart driver是一个 客户端服务发现和负载均衡工具，完善后可以替代现有 LB 的功能
- 分布式数据库架构，使用，SQL优化等与PG有一些不同的地方，在使用时需要特别注意

Q & A


openGauss
OceanBase
ArkDB
RASESQL
HotDB
StellarDB
QianBase xTP
云树Shard
GoldenDB
DolphinDB
MatrixDB
DynamoDB
SinoDB
FastData
Galaxybase
KunDB
SQL Server
vertica
D B 2
GBase
Oracle
达梦数据库
Spacture
开务数据库
UbiSQL
神舟通用
TDSQL
TiDB
Tapdata
StarRocks
KingbaseES
2010
2014
2018

THANKS