

分布式数据库时代

数据库应用和管理新方法

泽拓科技创始人&CEO 赵伟

目录



- 01** 数据库技术发展历史回顾
- 02** 数据库选型的新思考
- 03** KunlunBase的架构与核心能力
- 04** 应用系统架构师的新任务
- 05** DBA面临的新任务
- 06** KunlunBase的未来技术展望

数据库技术发展历史回顾

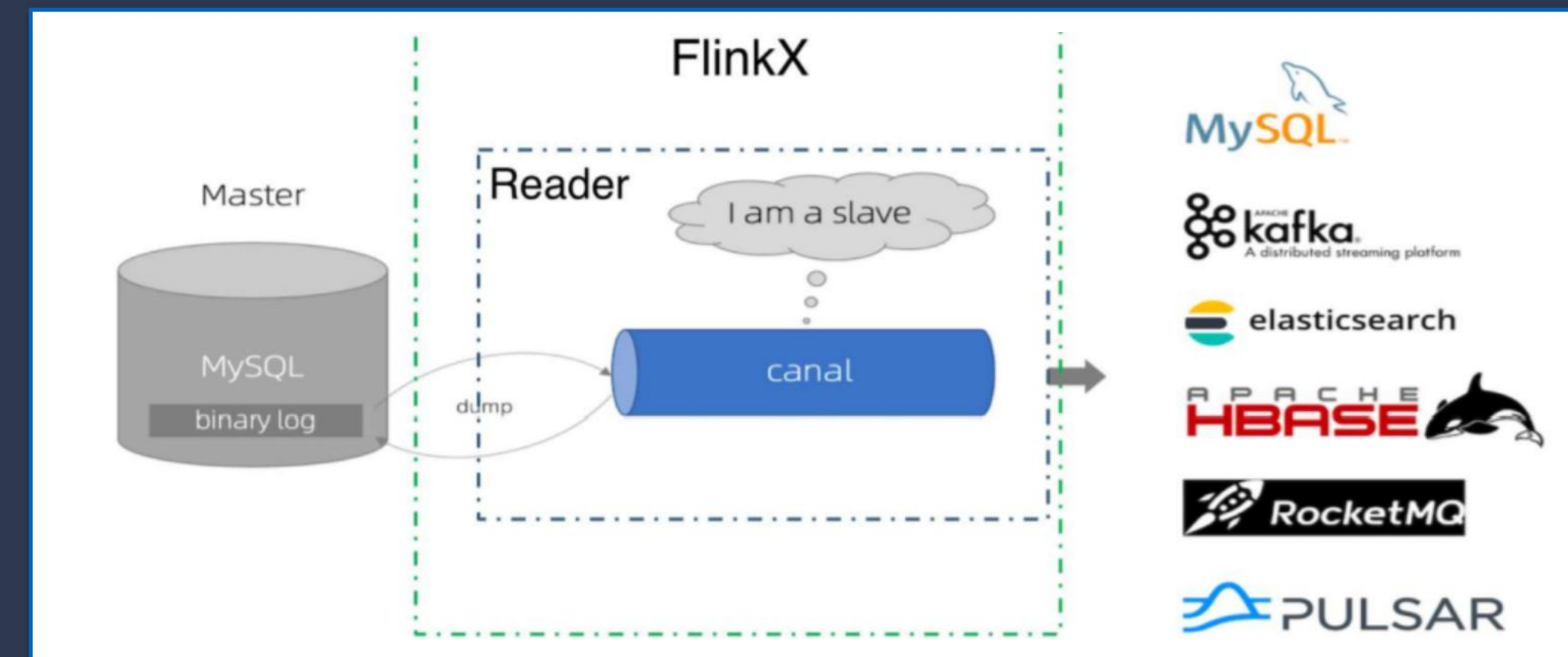
铭记经验和教训，不要重复踩坑

启蒙、探索与发现

- 拓荒时代的教训
 - 模块化和封装，软件复用；软件设计和研发流程；
 - 软硬件分离，操作系统负责硬件资源管理
 - 数据管理与应用逻辑隔离，由专用的数据库管理系统(DBMS)软件完成
- 走出迷昧：现代数据库技术的理论基础
 - 关系代数和SQL：“建立在数学基础上的理论才是科学” (1)
 - 事务模型和ACID：抽象的接口，分工合作 (2)

高可用与持续服务能力

- 基于复制实现高可用：服务持续性和数据价值变得重要 (3)
 - 物理复制的优缺点
 - 优：快，稳定，准确
 - 缺：数据传输量大；不容易消费
 - MySQL Binlog的优缺点
 - 优：事务一致性；开放生态
 - 缺：事务提交性能损耗；复制性能和稳定性
 - PostgreSQL 的逻辑复制
 - 基本上是CDC

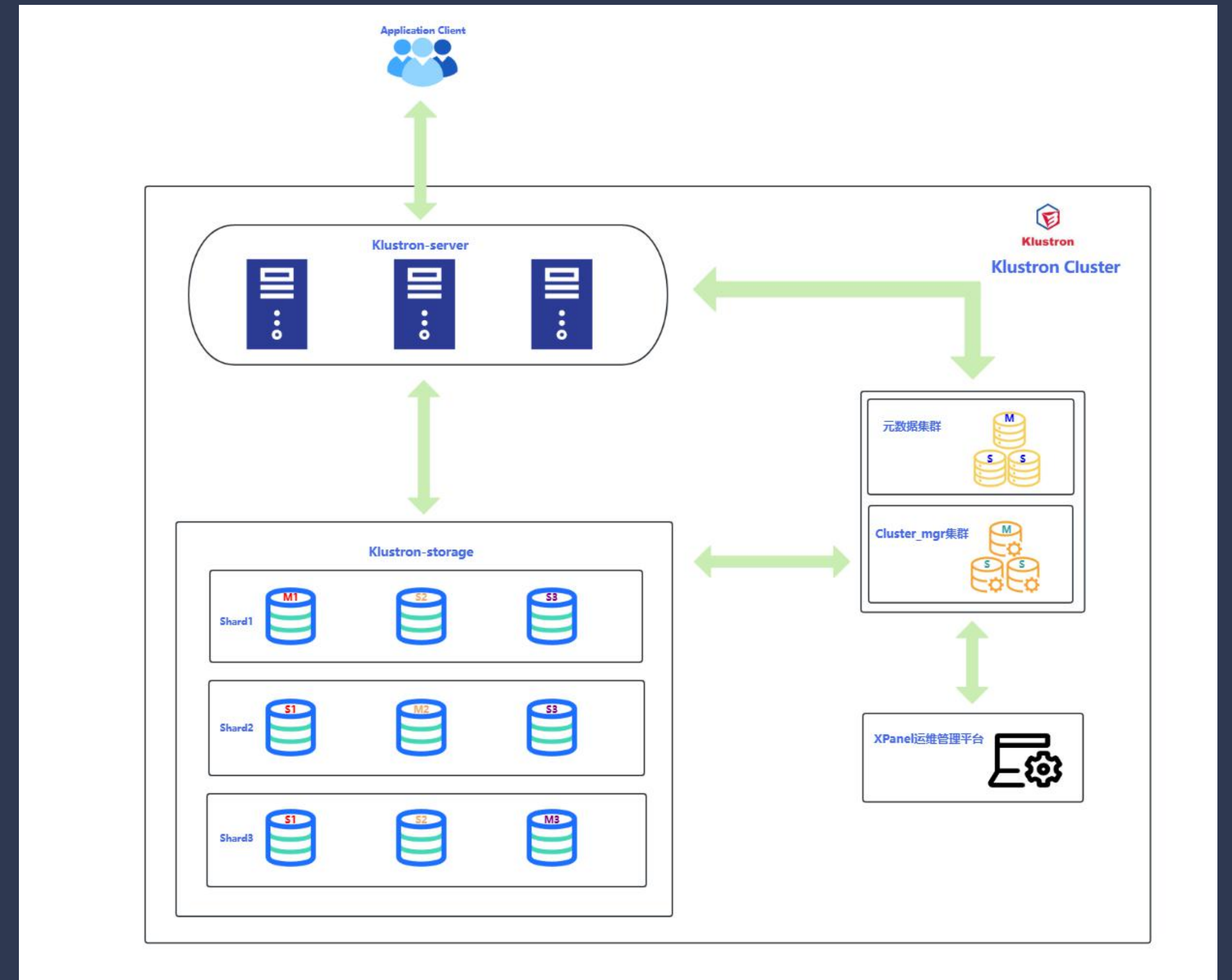


发展的曲折：NoSQL的演变

- NoSQL的变迁：No SQL! -> Not Only SQL -> No! SQL!
 - NoSQL是谁的救命稻草？
 - NoSQL的舍与得
 - NoSQL在哪里倒退了？

数据管理的螺旋上升与价值回归

- 分布式关系型数据库：与分库分表中间件有本质的不同 (4)
 - 使用多个服务器组成集群
 - 数据打散存储
 - 水平弹性伸缩
 - 真正的事务
 - 性能线性可扩展
 - 容错和自动故障切换



分布式数据库选型的新思考

应该选择什么样的分布式数据库系统？

必备的基础能力

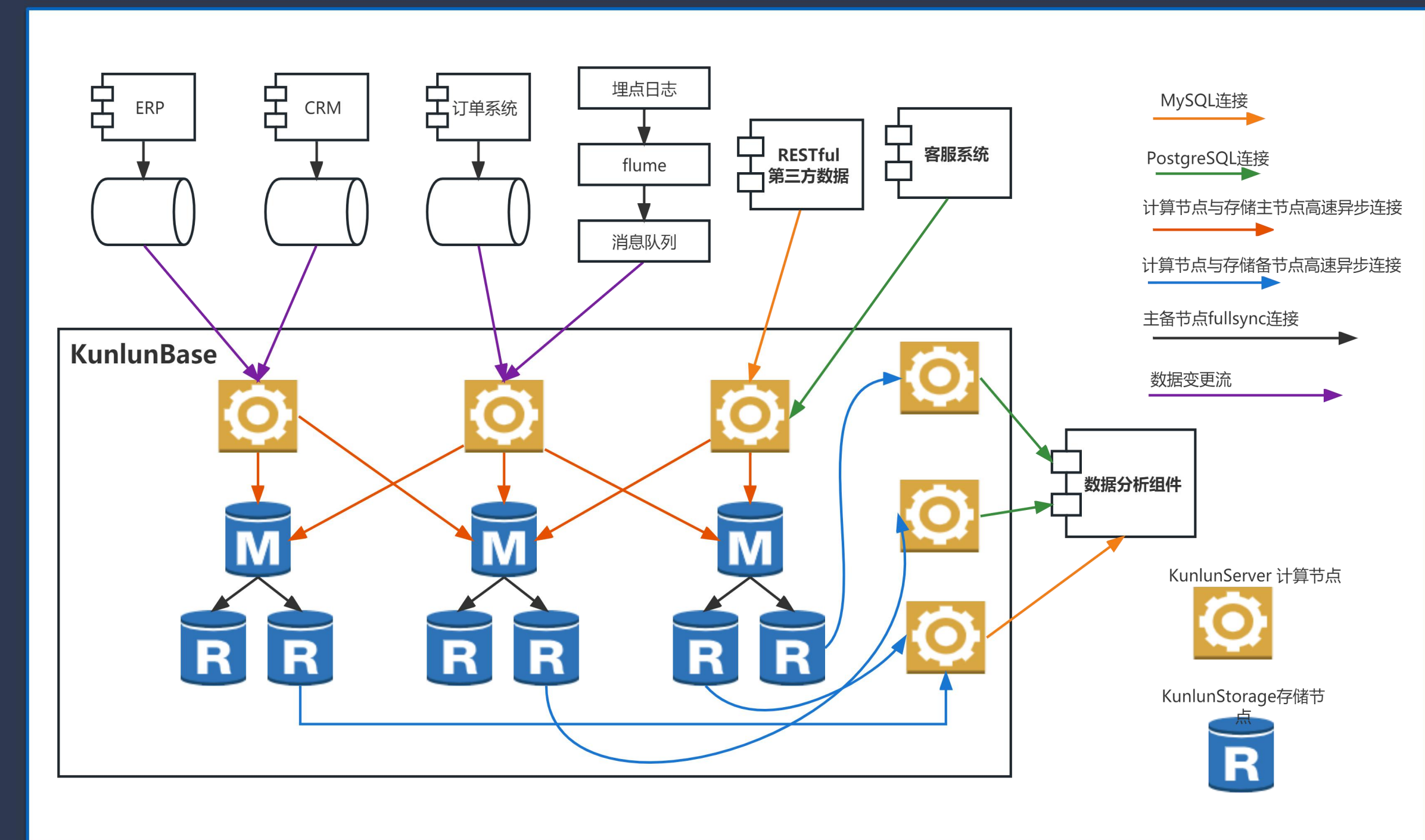
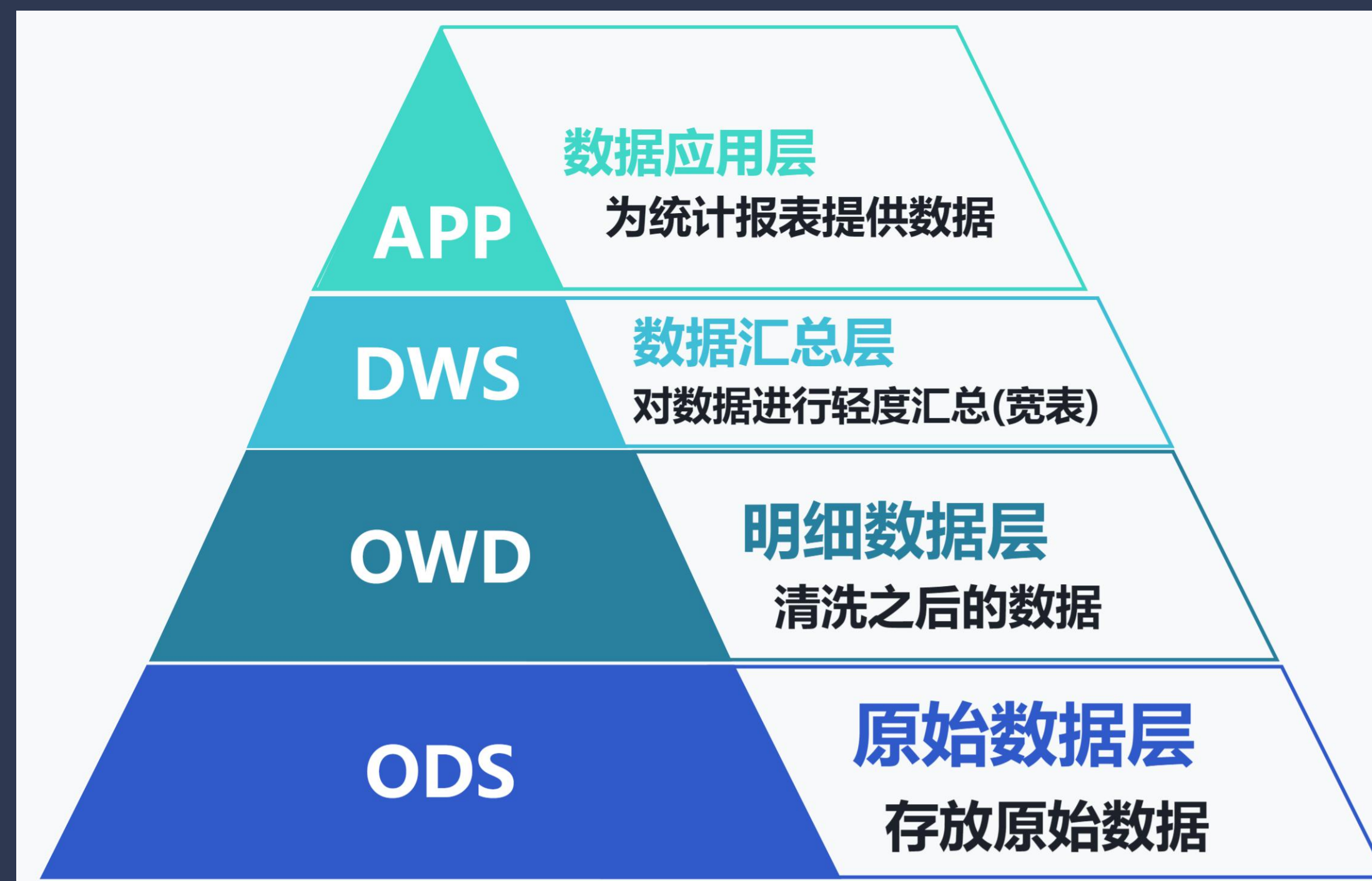
- To C业务的数据库容量能预测和规划吗？ 不能！
 - 必须具备水平弹性伸缩的能力
- 数据库停服30分钟的代价？ 可能过亿！
 - 数据库系统必须具备金融级高可靠性
 - 服务器、网络 and 机房故障发生
 - 自动处理、恢复和转移
 - 数据不丢不乱，服务持续在线

全面提升人效

- 人力成本是最高的成本，提升人的效率是最高的价值
 - 开发者：专注业务逻辑
 - 数据管理 -> DBMS
 - 项目进度和成本可控，质量可靠
 - DBA：运维监控管理全自动化
 - 自动故障恢复和高可用
 - 完全免除人力介入
 - 辅助性能调优和故障分析

数据分析

- 数据分析的新场景
 - 分析最新的业务数据：风控，推荐
 - 实时流式汇聚多个业务系统的数据更新：ODS



必备的高级SQL功能

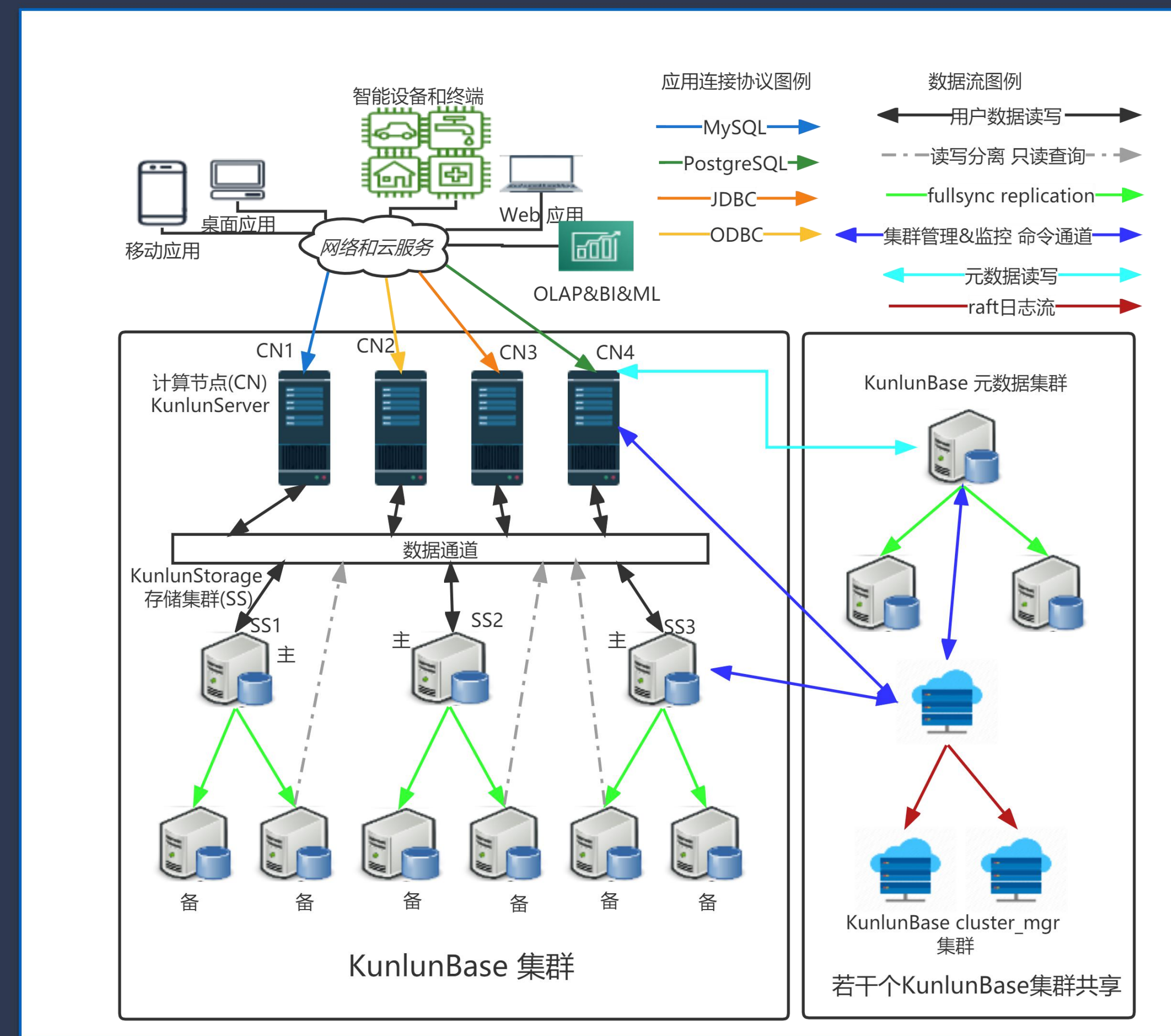
- SQL兼容性
- 视图
- 数据安全
 - 访问控制
 - 文件加密
 - 连接控制
- 数据有效性校验

KunlunBase架构与核心技术

KunlunBase 主要组件

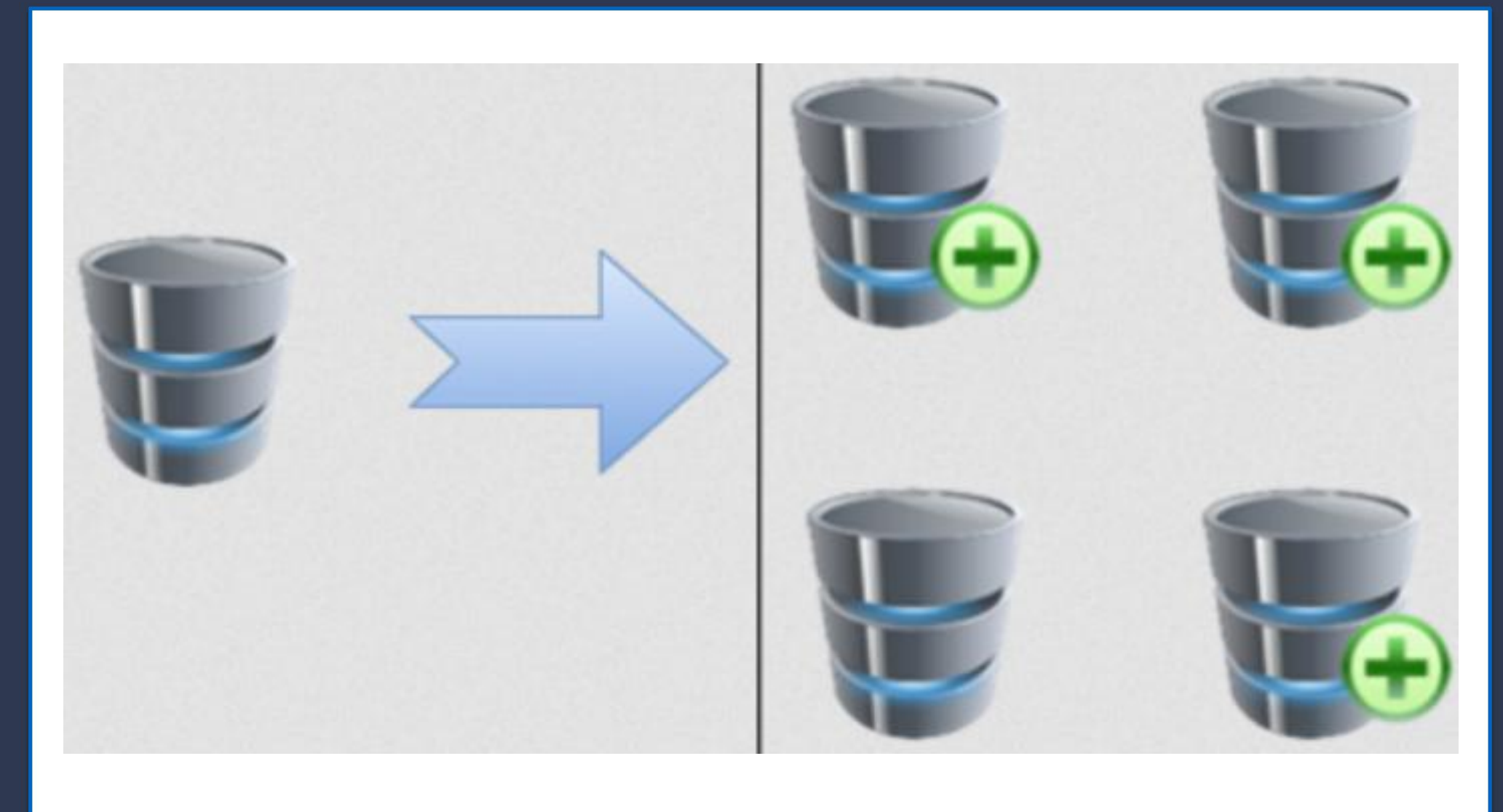
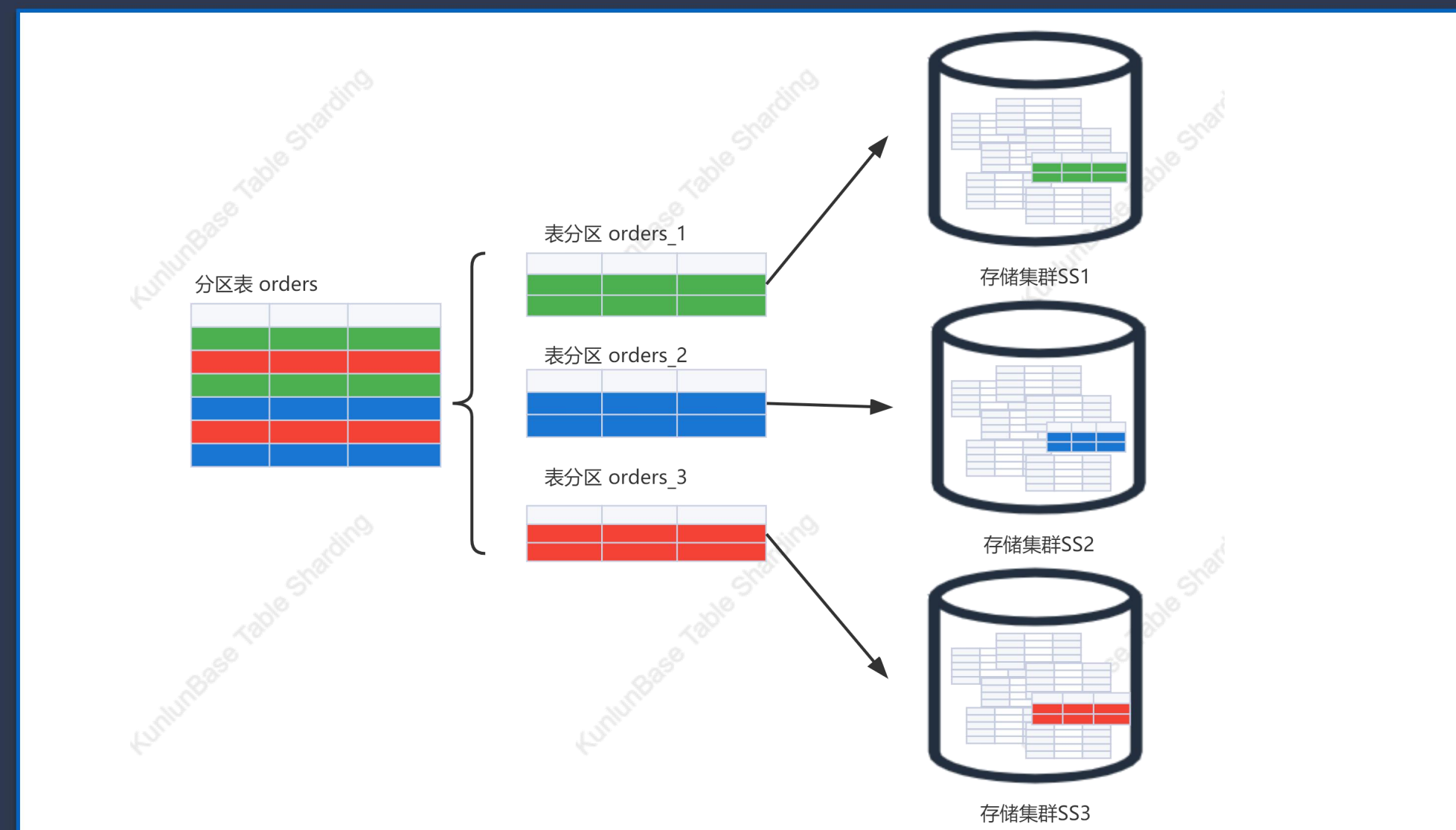


- 多个计算节点
 - MySQL&PostgreSQL 双协议双语法
 - 分布式事务&并行查询处理
 - 存储元数据
- 多个存储集群(shard) (HA)
 - 存储用户数据
- 元数据集群 (HA)
- cluster_mgr集群 (HA)
 - 故障恢复, shard HA
 - 集群管理 API
- XPanel
 - 集群管理、监控、告警
 - 故障分析



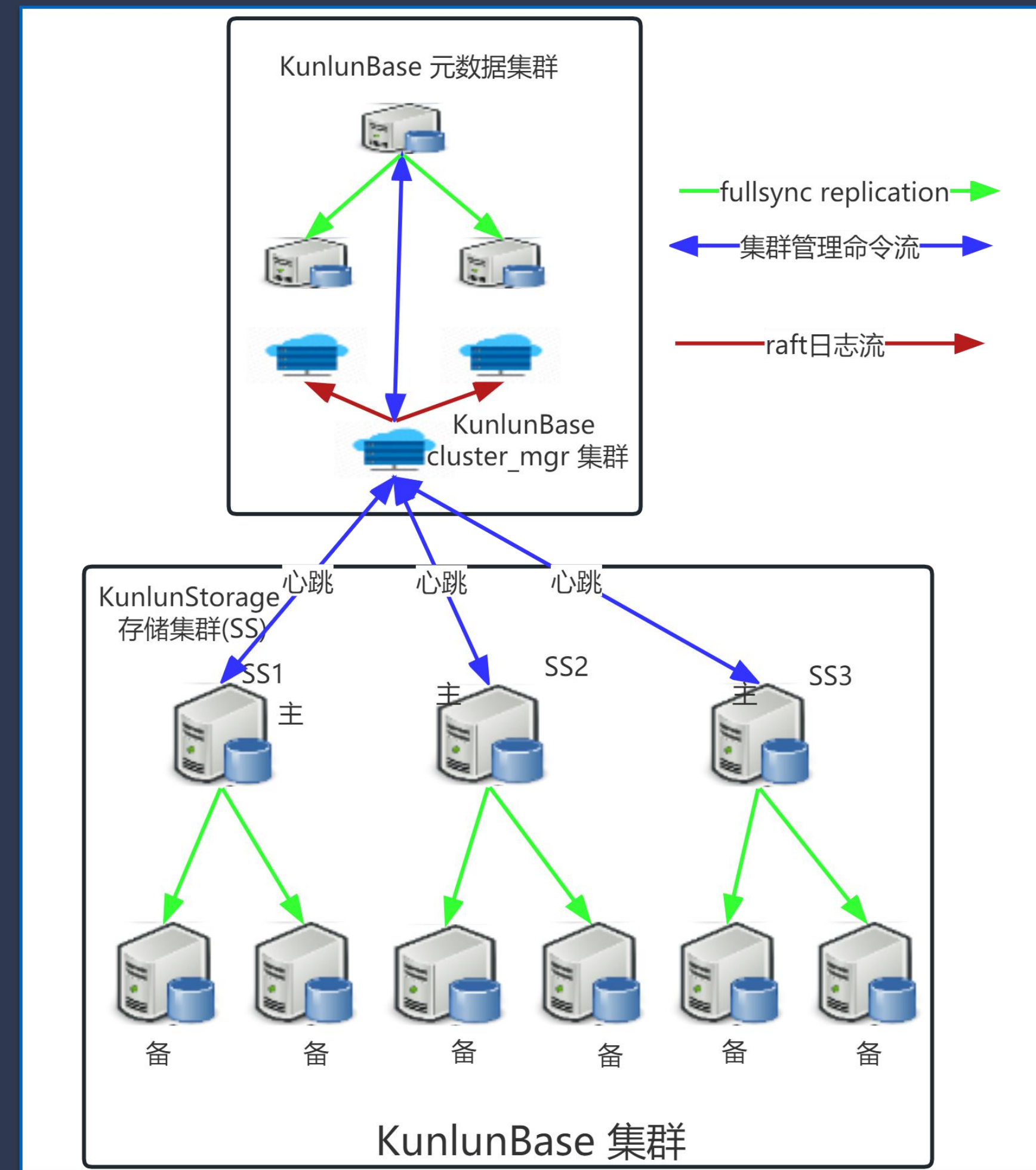
水平弹性伸缩

- 数据分区(partition): hash, range, list
 - 任意数量和类型的分区列
- 数据分布(distribution): auto, random, mirror, table grouping
- 扩缩容: 自动、柔性、不停服、无业务侵入、终端用户无感知
- 存储和计算分离, 多点读写, 按需增减存储和/或计算节点



金融级高可靠性

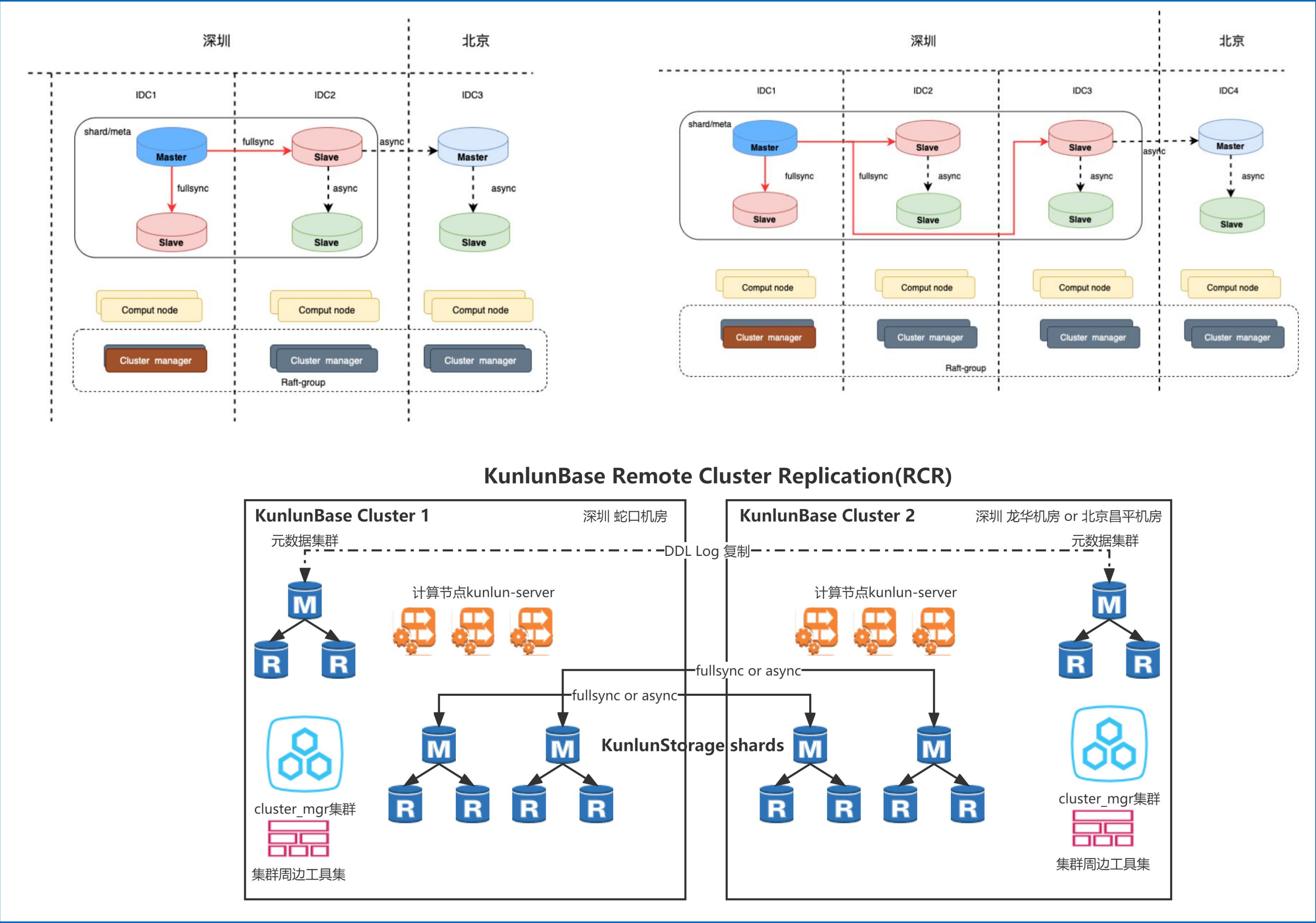
- 自动故障恢复、转移和切换
 - 数据不丢不乱，服务持续在线
 - 确保RTO < 30秒 & RPO=0
- shard故障处理和恢复
 - fullsync & fullsync HA
- 集群故障处理和切换
 - cluster_mgr



金融级高可靠性

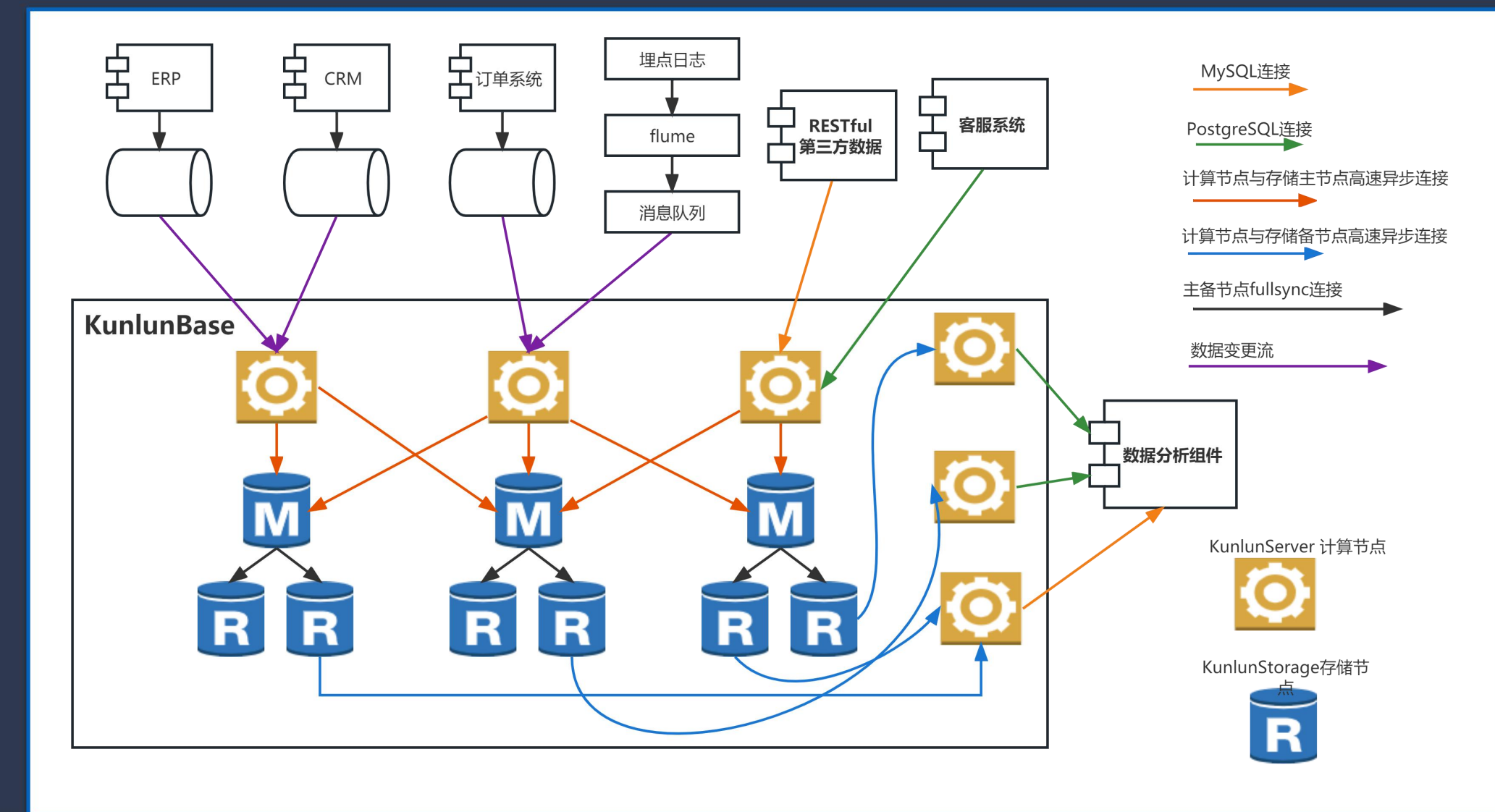


- 自动故障恢复、转移和切换
- 机房故障自动恢复
 - 多机房高可用
 - 同城/异地双活



HTAP 混合负载

- HTAP: OLTP & OLAP 一份数据，两类负载，互不干扰
 - OLTP为主：对应用软件等价于使用MySQL或PostgreSQL
 - OLAP为辅：多层级并行的分布式查询处理，充分利用大量硬件资源实现高性能
- 数据分析新场景
 - 分析最新数据：风控
 - 汇集多个数据库的数据变更做OLAP分析（ODS）
 - 大数据分析：推荐（ES）

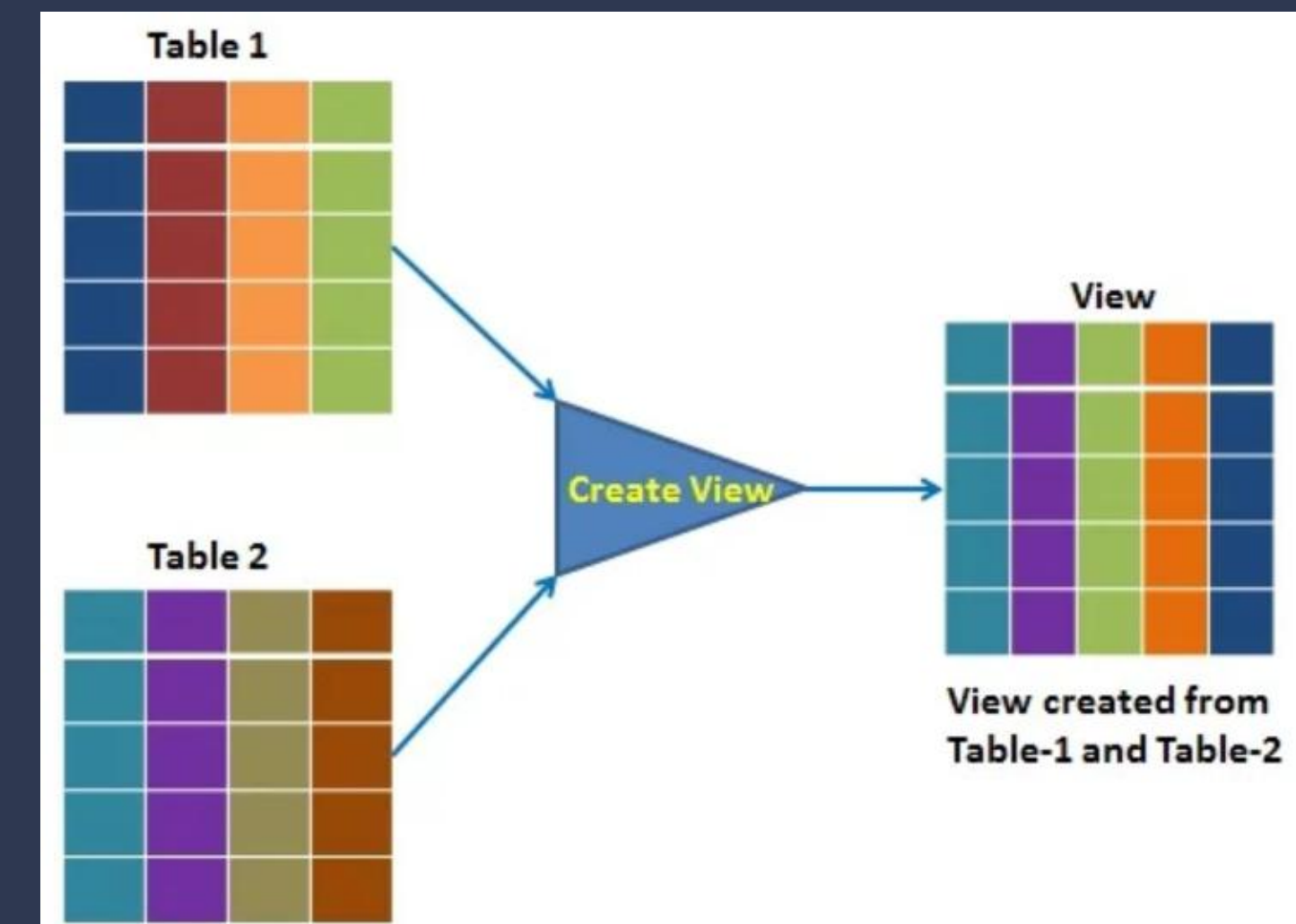


MySQL、PostgreSQL和SQL兼容性

- 融合标准SQL、PostgreSQL 和 MySQL 的应用和工具软件生态
 - 支持PostgreSQL的DDL 和DML语法和连接协议
 - 支持MySQL的DML和常用DDL 的语法和连接协议
 - 支持JDBC, ODBC, 所有常见编程语言的PostgreSQL和MySQL client connector
 - 支持Hibernate等常见的ORM映射中间件, 程序员无需编写SQL代码
- SQL兼容性: 高级查询功能
 - [物化]视图, domain (列级约束), CHECK约束, policy(RLS)
 - 多粒度的访问控制
 - 存储过程, 触发器

数据安全和校验机制

- 全方位的数据安全保障
 - 连接加密：全链路安全传输
 - 存储节点：数据文件和日志文件存储加密：不怕被拖库
 - 计算节点
 - 多层次细粒度访问控制
 - 库、模式、表、列、行、视图、存储过程
 - 有效性验证：CHECK约束(字段、行)；domain
 - 约束：数据类型、null-ability, unique-ness
- 数据访问控制和有效性检查 必须源头控制
 - 非法数据有毒，还会传染，久拖难治！



分布式数据库时代 应用架构师和DBA的新任务

应用架构和数据设计

简化应用架构设计：把数据管理任务完全交给分布式数据库系统

- 应用层只实现业务逻辑
- 应用层分库分表或者中间件分库分表已经过时了！

设计数据分布方案：理解业务需求和数据特征

- 目标：最优的性能
- 数据表的拆分方法
- 数据分片的分布策略

发挥高级SQL功能的价值

- 一般价值：存储过程、触发器
 - 存储过程：性能、可维护性；访问控制
 - 触发器：性能问题，可以用CDC代替，外部做流式变更处理
 - KunlunBase计算节点扩容，缓解资源开销和性能问题
- 高价值：CHECK约束、[物化]视图、domain（列类型）、RLS、访问控制
 - 视图：解耦，虚拟；访问控制
 - 物化视图：性能
 - CHECK约束：字段正确性规则
 - domain：面向业务逻辑
 - 访问控制：源头建立安全机制
 - RLS：细粒度

DBA：保障数据安全有效

- 在数据源头建立数据安全和校验机制
 - 数据访问控制规则
 - 用户名，密码，客户端IP范围：防止DDOS，不暴露到公网
 - 库、模式、表、列、行、视图、存储过程
 - 用户、角色、授权和回收
 - 数据有效性校验机制
 - PK，唯一索引
 - 数据类型和宽度，可否NULL
 - 字符集和collation：最好全database统一使用 UTF8
 - 混用易产生比较错误，导致排序，索引失效
 - CHECK 约束

DBA：系统性能调优

- 积极解决日常技术问题
 - 数据库性能监测和调优
 - 持续优化告警规则
 - 数据库系统故障分析和诊断



DBA：服务器资源规划与分配

- 服务器集群资源规划
 - 资源预留：何时开始和如何扩容？
 - 资源分配：服务器 -> 业务和数据
 - 何时清除备份的数据和日志文件？

KunlunBase未来技术展望

- Persistent Memory (PM)
 - WAL
 - WAL-less
- 计算节点多线程
- JIT
- 向量化
- 更高性能的replication
- KunlunBase-1.3的性能目标: TPS 翻倍, 延时缩短50%以上
 - 超越最新版本的Oracle和MySQL单机性能!

想一想，我该如何把这些 技术应用在工作实践中？

THANKS



个人微信
欢迎扫码添加



Klustron数据库公众号
欢迎扫码关注