分布式数据库如何实现高性能

SequoiaDB
巨杉数据库

公司简介：

SequoiaDB巨杉数据库（巨杉软件），成立于2011年专注于新一代企业大数据平台研发, 其核心产品SequoiaDB（巨杉数据库）是国内第一款新一代分布式数据库;

核心产品完全自主研发，数据库引擎没有基于任何开源数据库源代码, 已经成功部署并运行在多家世界500强企业的生产环境中;

获著名基金启明创投（A轮）与DCM（B轮）融资;

中国第一款商业开源数据库产品
www.github.com/sequoiadb/sequoiadb
www.sequoiadb.com
www.oschina.net/p/sequoiadb

DB GEEK

IT大咖说 知识分享平台

SequoiaDB 巨杉数据库

RED HERRING GLOBAL WINNER 100

"全球创新企业Top100"

——《红鲱鱼》

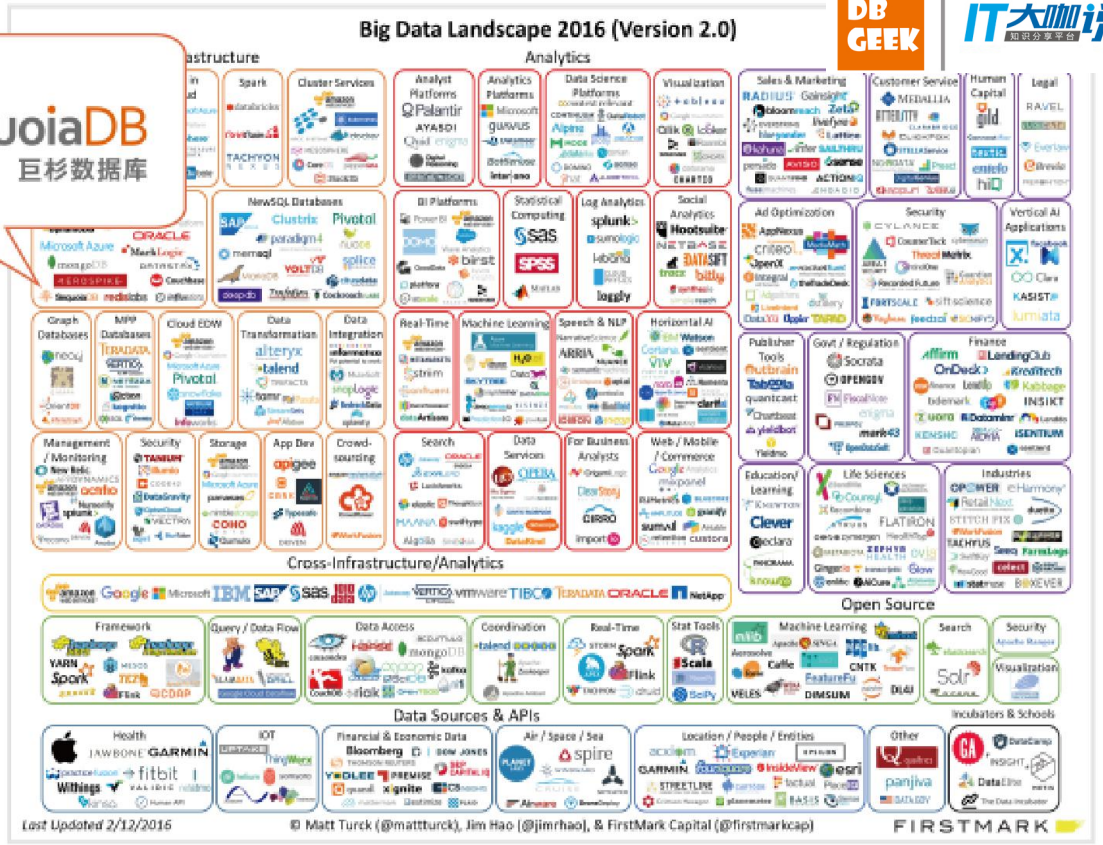
美国最具影响力商业媒体

FAST COMPANY 50 MOST INNOVATIVE COMPANIES

"中国创新企业50强"

——《快公司》

美国著名创新媒体



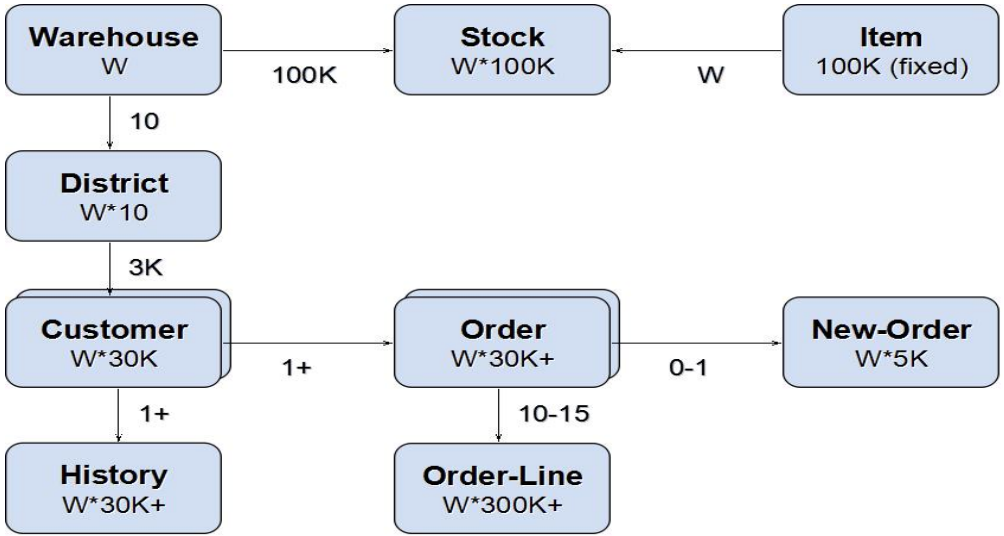
数据库高性能标准

数据库应用范畴

- 交易型数据库
  - Transactional DB
  - 传统OLTP业务
- 分析型数据库
  - Analytical DB
  - 分析报表型业务
- 联机型数据库
  - Operational DB
  - 在线高并发非交易型业务

交易型数据库性能核心指标

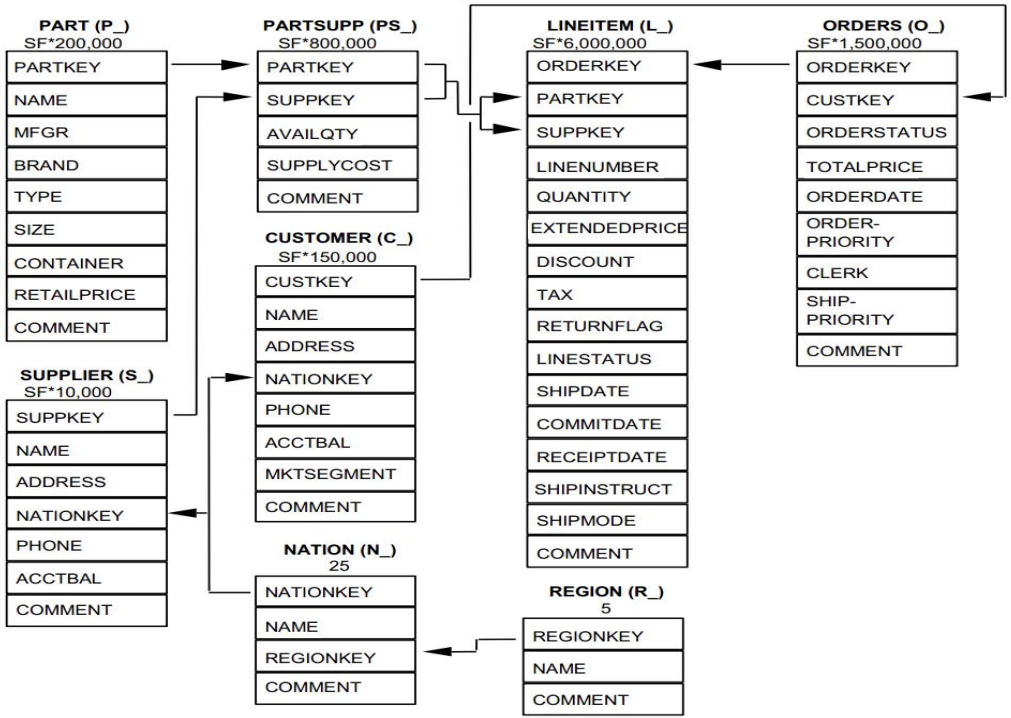
- 测评方式
  - TPC-C测试结果
- 读写性能
  - 随机读操作性能
  - 随机写操作性能
  - 随机更新性能
- 事务相关
  - 提交回滚开销
- 一致性相关
  - 悲观/乐观锁
  - 主外键能力
  - 触发器能力
- 可靠性
  - ACID
  - 高可用性
  - 准实时灾备功能
  - 备份恢复



分析型数据库性能核心指标

- 测评方式
  - TPC-H测试结果
  - TPC-DS测试结果
- 读写性能
  - 批量读操作性能
  - 批量装载操作性能
  - 大表关联性能
  - 大量数据聚集性能
- 批处理加工
  - 存储过程能力

Figure 2: The TPC-H Schema



联机型数据库性能核心指标

- 测评方式
  - YCSB测试结果
- 读写性能
  - 随机读性能
  - 随机写性能
  - 批量写性能
  - 随机更新性能
  - 并发能力
  - 可扩展性
- 可靠性
  - 高可用性
  - 准实时灾备功能
  - 备份恢复
  - 强一致性与最终一致性

Workload	Operations	Record selection	Application example
A—Update heavy	Read: 50% Update: 50%	Zipfian	Session store recording recent actions in a user session
B—Read heavy	Read: 95% Update: 5%	Zipfian	Photo tagging; add a tag is an update, but most operations are to read tags
C—Read only	Read: 100%	Zipfian	User profile cache, where profiles are constructed elsewhere (e.g., Hadoop)
D—Read latest	Read: 95% Insert: 5%	Latest	User status updates; people want to read the latest statuses
E—Short ranges	Scan: 95% Insert: 5%	Zipfian/Uniform*	Threaded conversations, where each scan is for the posts in a given thread (assumed to be clustered by thread id)

*Workload E uses the Zipfian distribution to choose the first key in the range, and the Uniform distribution to choose the number of records to scan.

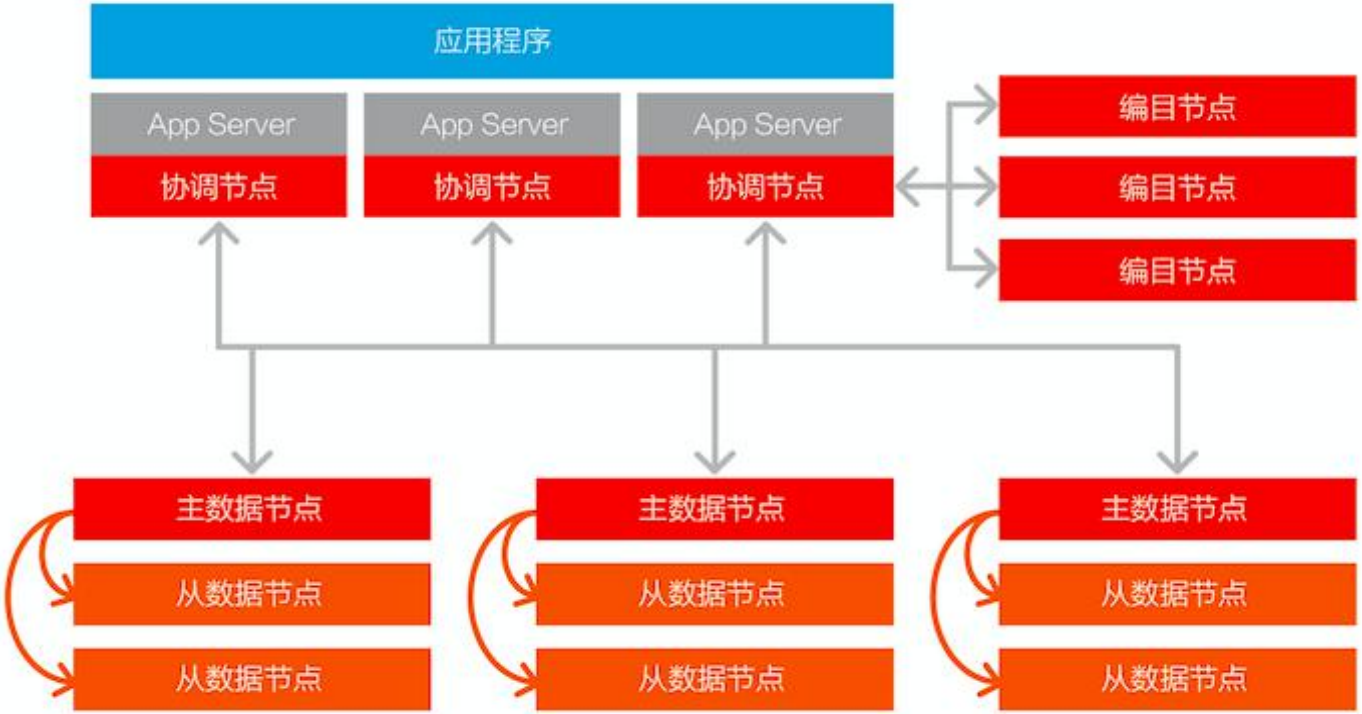
Table 2: Workloads in the core package

如何设计一个高性能交易型与联机型数据库？

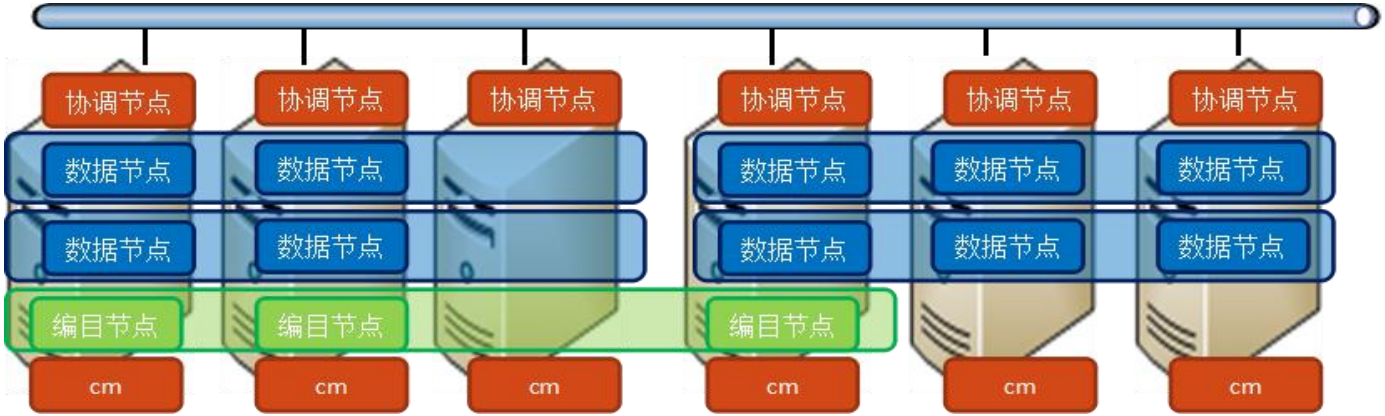
- 性能指标
  - 着重随机读写性能
  - 着重高并发性能
  - 满足高性能的提交回滚能力
- 功能指标
  - 支持高可用
  - 支持准实时灾备
  - 支持ACID
  - 强弱一致性切换
  - 锁功能切换

分布式数据库高性能实现

SequoiaDB分布式数据库架构



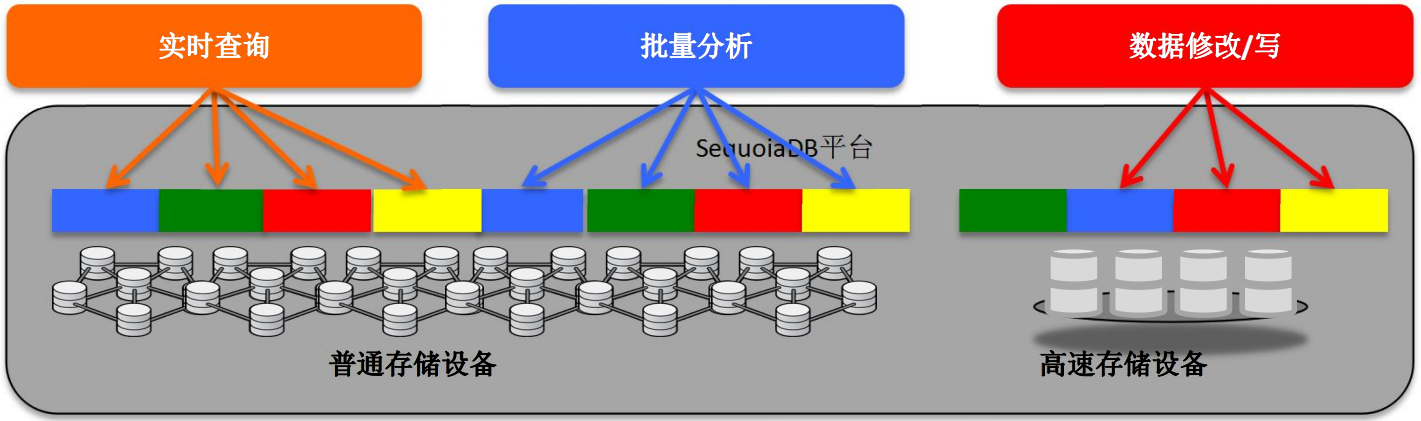
SequoiaDB物理架构



角色	功能
协调节点	胖客户层，从编目读取数据分布信息，从数据节点读取数据
编目节点	负责元数据信息存储，包括组信息、表切割信息
数据节点	负责数据表存储，提供查询、聚集、数据复制功能
CM节点	负责集群管理，包括watchdog, 节点增删启停

分布式架构优化：读写分离机制

- 数据在多个分布节点内自动复制，并实现写请求和读请求的自动分离，避免读请求对数据写入的影响。
- 此外，可进一步定制数据分布策略，保证不同类型业务可以运行在同一平台上，但同时又不会互相干扰，比如：
  - 冷/热数据区分离
  - 写交易的“强一致性”和“弱一致性”分离
  - 查询/批量分离



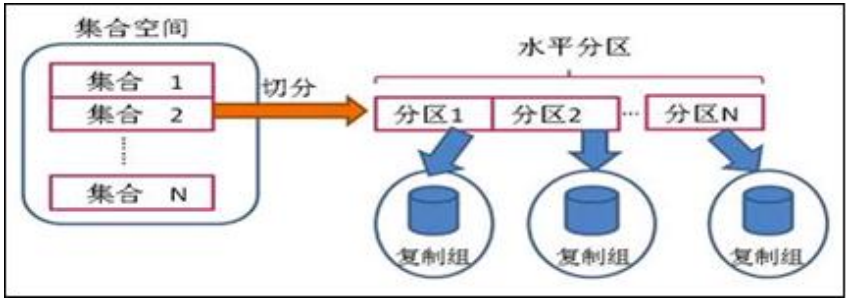
分布式架构优化：可配置强弱一致性机制

一致性因素	读主节点	读备节点
W=N	强一致，无数据丢失，事务操作符合传统关系型数据库模型	强一致，无数据丢失，读操作受节点分布影响会有脏读时间
W!=N	强一致型，极端情况下可能发生数据丢失，事务操作符合传统关系型数据库模型	最终一致，可能发生读取不到写入的数据，极端情况下可能发生数据丢失

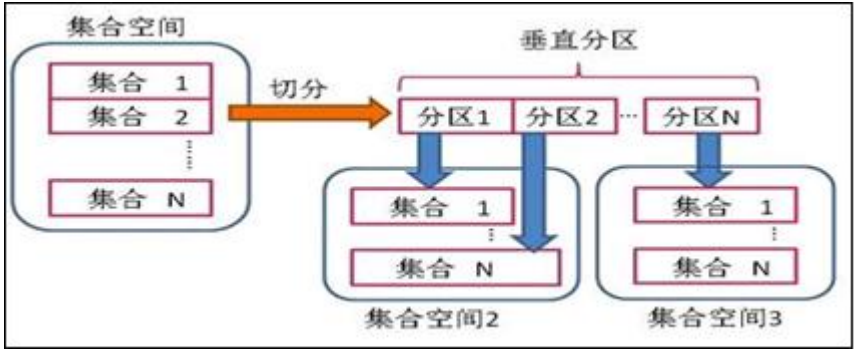
持久性因素	备节点响应策略	数据可靠性
物理同步	数据在备节点写入事务日志	全部节点宕机不会造成数据丢失，但会损失性能
逻辑同步	数据在备节点处理但未写入事务	主备节点同时宕机可能会造成数据丢失，数据查询强一致
半同步	数据在备节点成功接收但还未处理	备节点宕机可能会造成数据丢失，最终一致性
异步	数据不需要被备节点感知	主节点宕机可能会造成数据丢失

分布式架构优化：数据多维分区

SequoiaDB支持水平分区和垂直分区。水平分区尽可能选择唯一性较高的字段，垂直分区尽可能选择时间或区域这种相关性较高的字段。一个表可以同时为水平分区与垂直分区

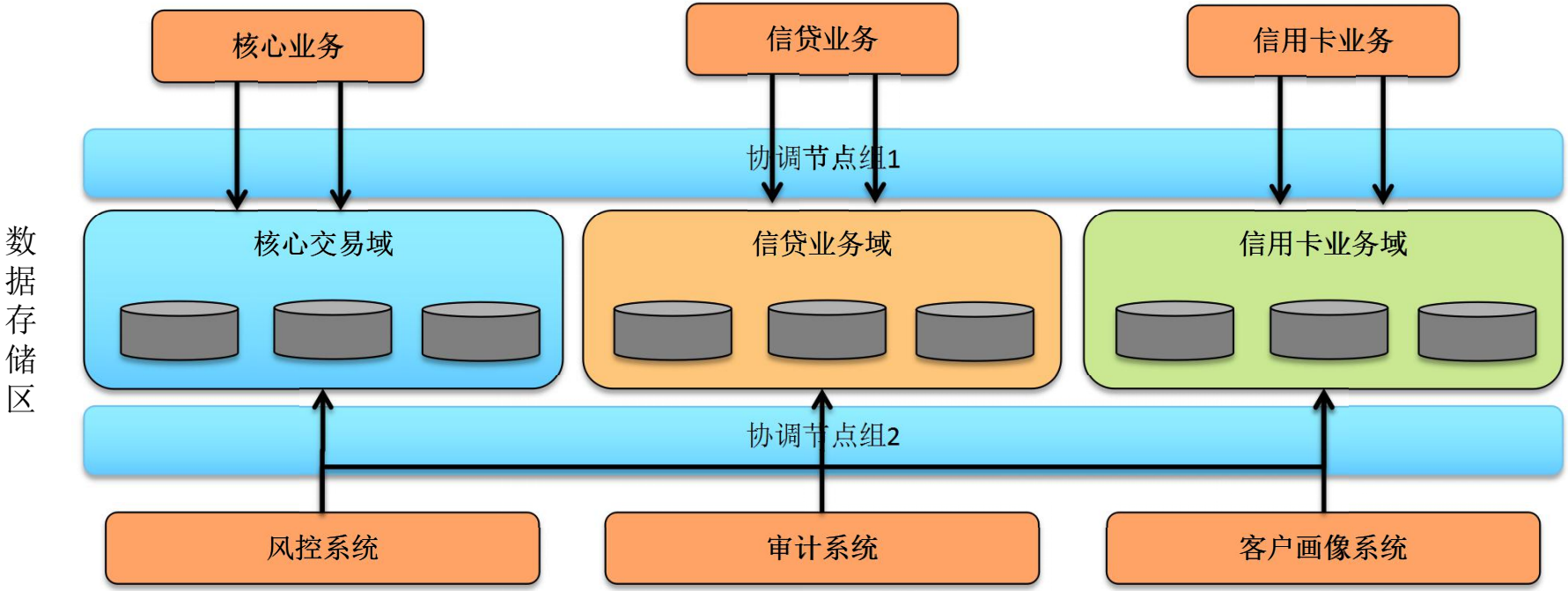


分别适合流水数据与快照数据

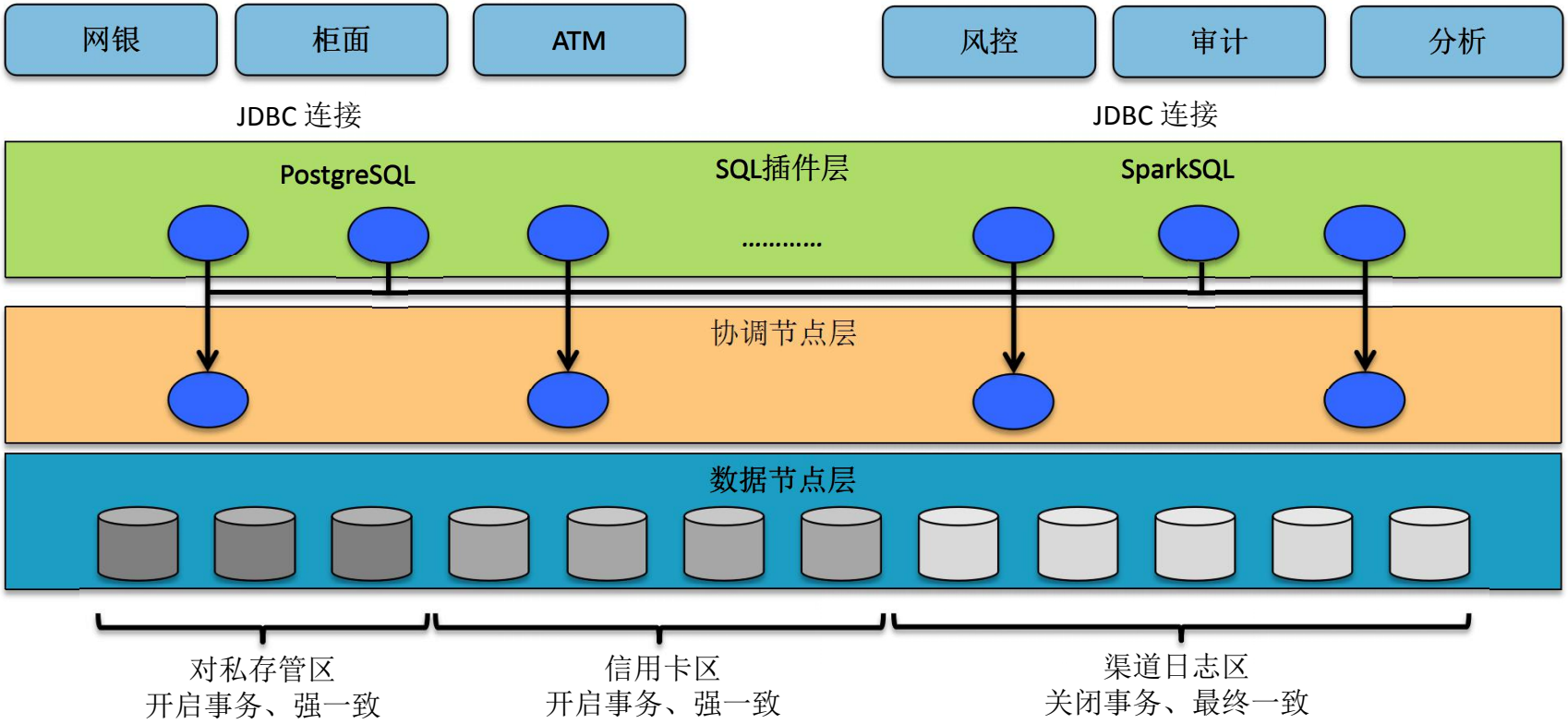


优势： 容量和性能可线性扩展

分布式架构优化：数据域逻辑与物理隔离



分布式架构优化：SQL与存储引擎隔离

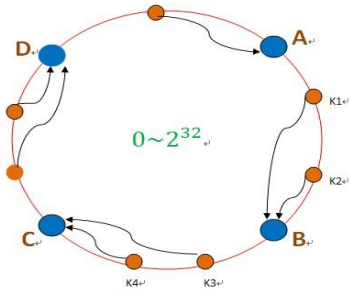


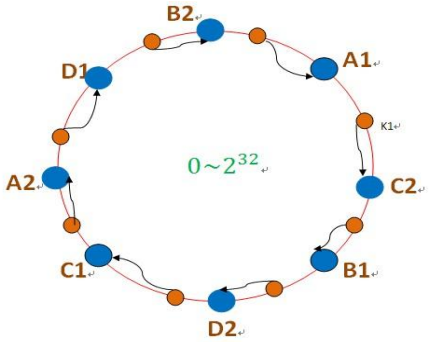
分布式架构优化：分布式事务机制

- 二段提交
- 协调节点首先发起预提交
- 当所有数据节点响应成功后，进行统一提交

分布式架构优化：一致性散列机制

- SequoiaDB可以指定自定义分区键，不指定分区键的情况下使用记录ID作为分区键，可以保证数据随机散列分布；
- 特殊情况下，当出现热点数据时，热点数据所在分区可以使用split命令进行切分，进一步细化粒度减少热点；
- 切分过程当中为全在线操作，对业务无感知；
- SequoiaDB支持多维数据分区的机制，在大容量磁盘的配置中性能表现最佳。

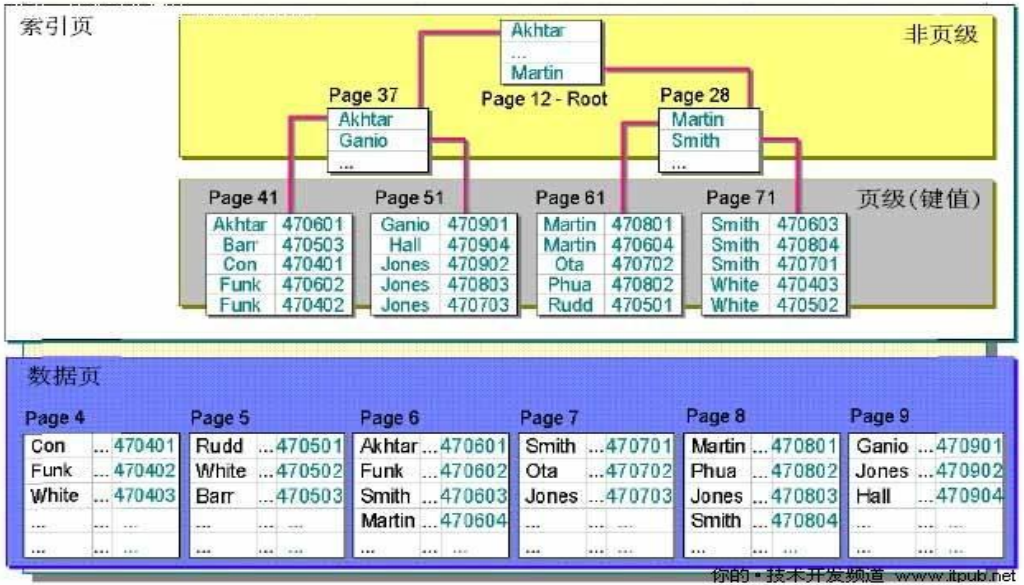




一致性散列：只需移动少量数据即可完成切分，不需全部数据导入导出。

引擎内部优化：B树多维度索引

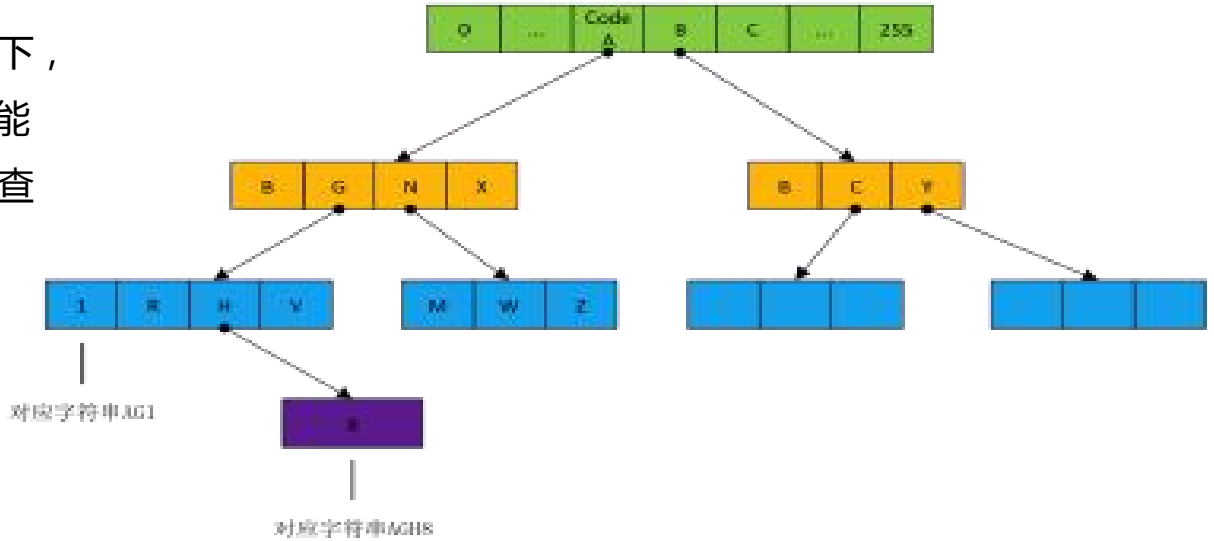
- 支持多字段索引
- 支持复合索引
- 支持唯一索引
- B树索引与数据保持强一致
- 支持全文检索索引
- 全文检索索引与数据保持最终一致



引擎内部优化：高效压缩机制

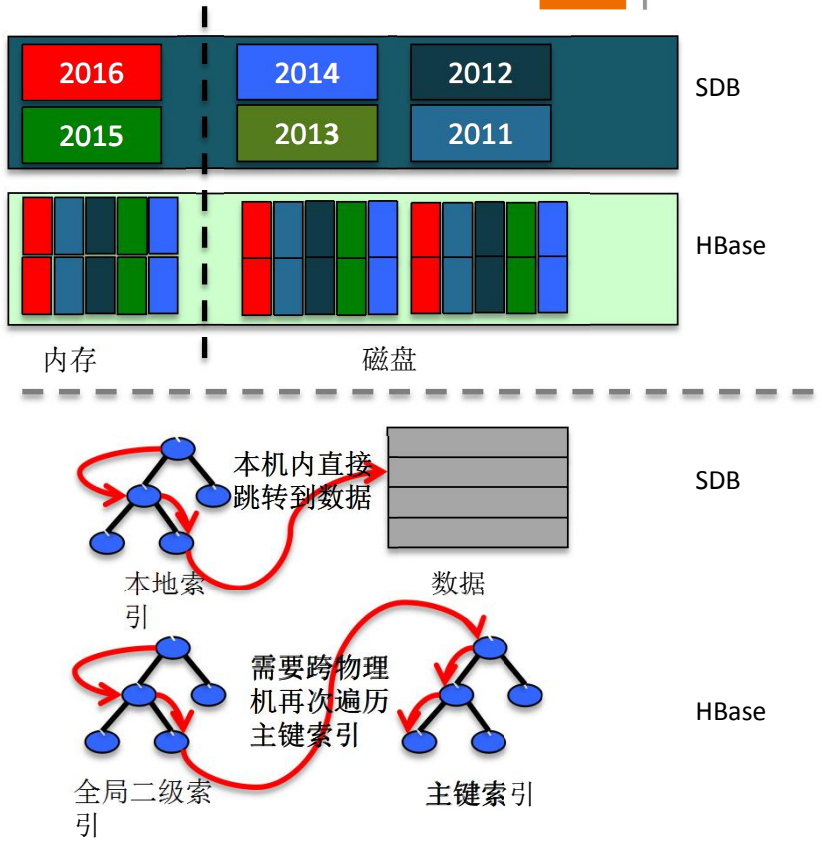
SequoiaDB支持Snappy和LZW两种压缩机制，既能实现快速压缩，也能满足深度压缩需求。

在IO吞吐量非常高的查询场景下，基于数据字典的深度压缩机制能够大幅降低IO开销，有效提高查询效率。

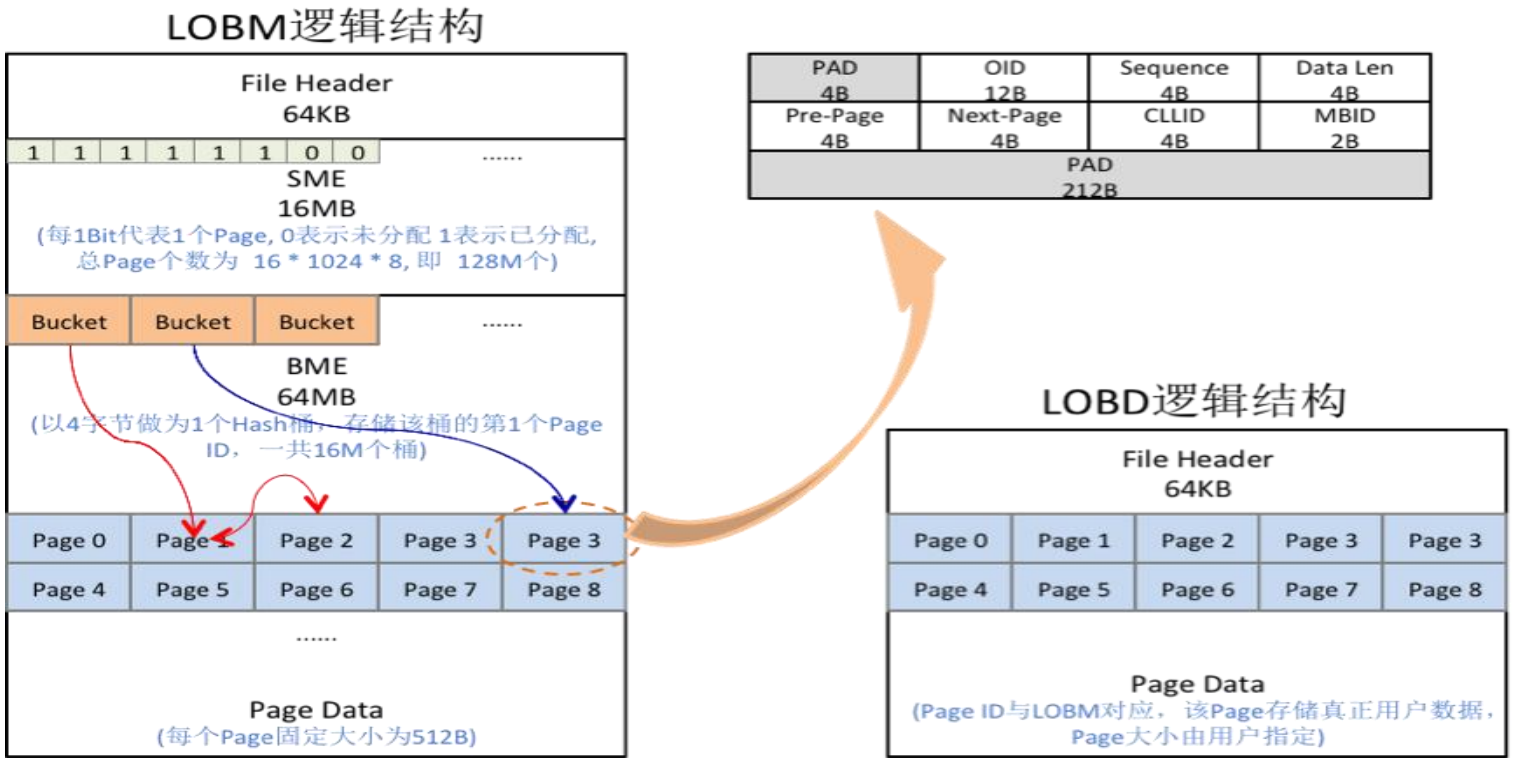


引擎内部优化：索引查询原理对比

- SequoiaDB多维分区机制能够在每个分区内部按照时间维度对数据进行汇聚
  - SDB吻合历史数据查询的主要业务流程，近期数据常驻缓存
  - HBase/Hyperbase/Hindex均不提供类似的机制，远期数据与近期数据无法进行隔离，在典型高并发实时响应场景中内存命中率不高
- SequoiaDB使用B树索引，而非HBase的二级索引
  - 二级索引的基本机制是先通过索引找到主键索引，再通过主键索引找到记录，造成两批随机I/O
  - 从性能的角度来看，SequoiaDB可以至少节省一半以上的随机I/O开销
  - HBase二级索引主要使用全局索引，在查询的过程当中会涉及到大量跨物理机的数据检索，对性能影响极大



引擎内部优化：非结构化块存储机制



引擎内部优化：块存储支持大小文件

- 其他实现方式

➢ 关系数据库+文件系统地址

问题：文件条目受关系型数据库的性能限制，比如超过3亿条后性能急剧下降

➢ HDFS

问题：受 Namenode限制无法处理大量的小文件，分配64MB存储浪费空间；小文件不定期后台自动聚合，影响系统的使用稳定性。

➢ HBase

问题：做Merge的过程造成I/O飙高，无法满足在线ECM服务场景。

- 分布式对象存储
- 文件按照数据块处理
  - 自动按照64/128KB的数据块进行切分，放在不同分区存储
  - 使用DIO避免二进制数据占用文件系统缓存
  - 并行处理
- 与GridFS相比不占用内存
- 与HDFS相比不存在Namenode限制

分布式数据库高性能实践案例

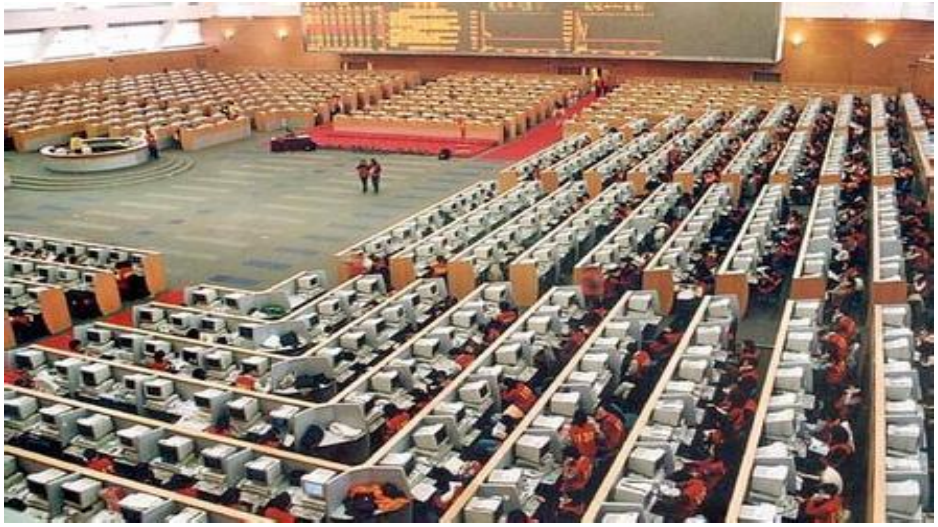
SequoiaDB

证券行业高并发查询

某证券监管机构的股票交易信息管理系统，存储全国交易所每天上传的所有的股票交易信息。如今通过APP，网页端等，开放给股民用户进行实时的查询。

通过搭建基于SequoiaDB的数据库存储，该机构将所有历史数据实现在线化，同时保证每天增量的及时写入。

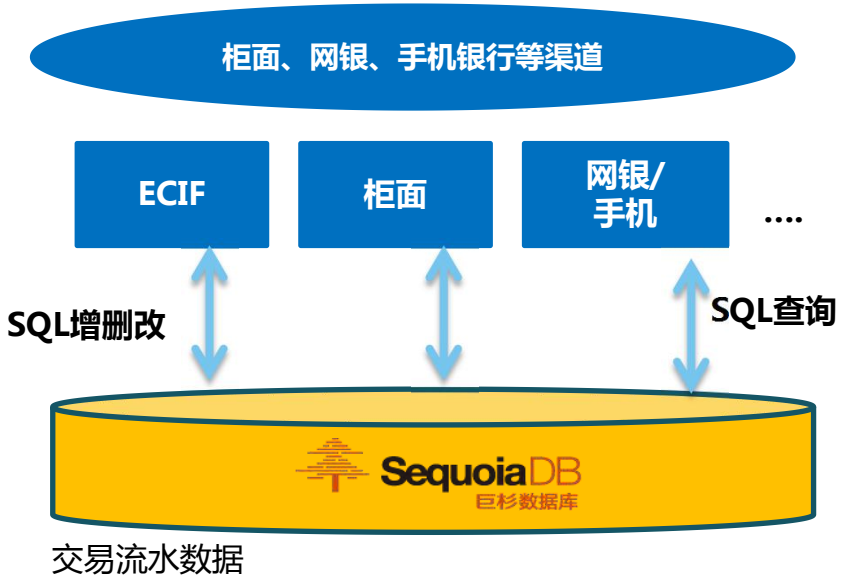
- 平均每日超过2亿条记录写入
- 高峰时段，同时有超过百亿级别的数据需要被检索、调用
- 系统保存3年内所有交易和持有数据
- 峰值并发量超过10000
- 高峰时段，查询返回时间小于100ms
- 实际测试性能10倍于原有MySQL
- 操作涉及3张数据表的关联，总量超过3000亿条数据



银行历史数据平台

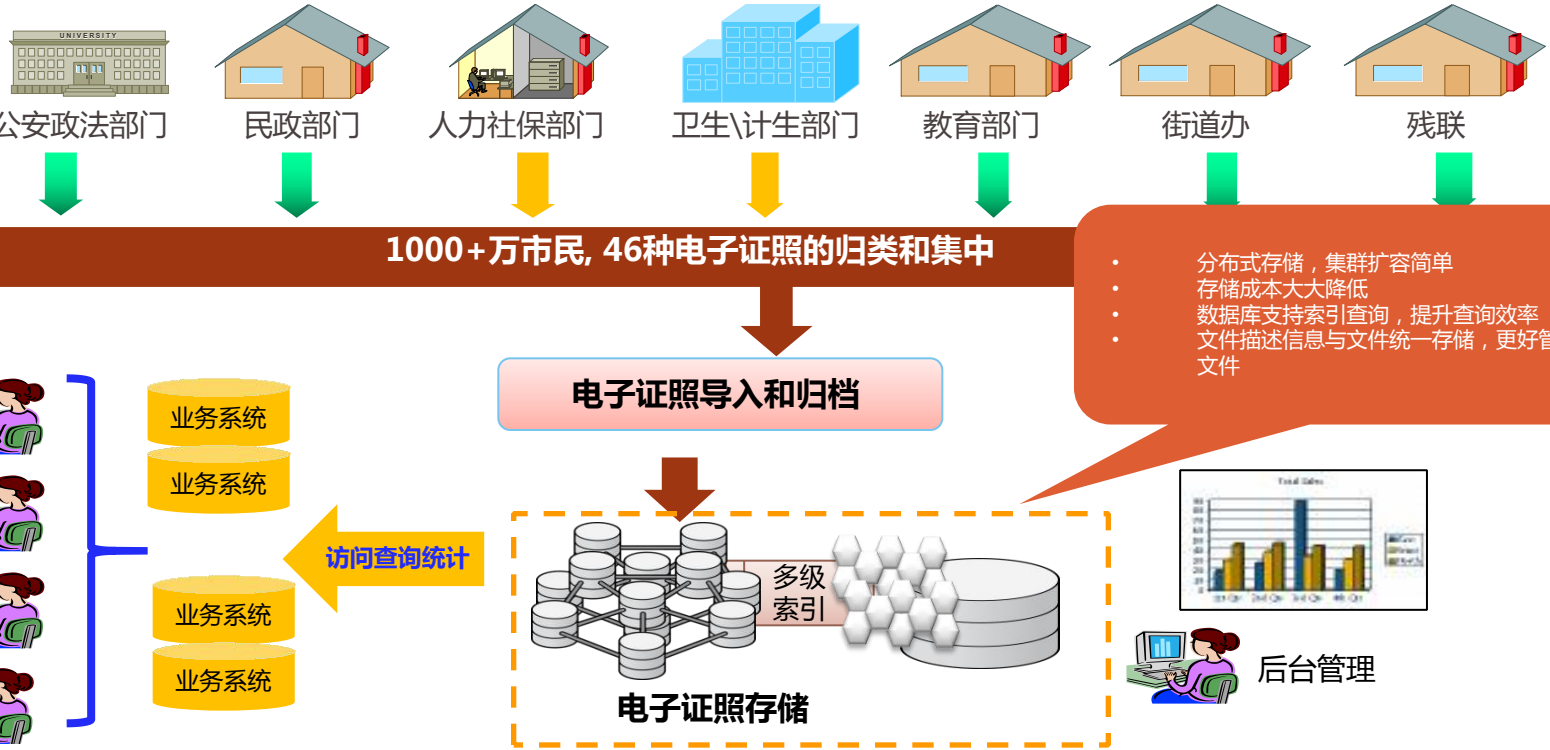
- 高扩展性和稳定性
- 数据分析的接口
- 平滑过渡，不影响原有数据处理流程， 对现有应用影响尽量

- 超过1PB数据存储，100+节点
- 50亿记录的实时查询性能<1s
- T+1 批量将生产数据全量及增量导入SDB
- SDB对外提供SQL接口，可继续使用现有查询应用



政务数据大数据湖

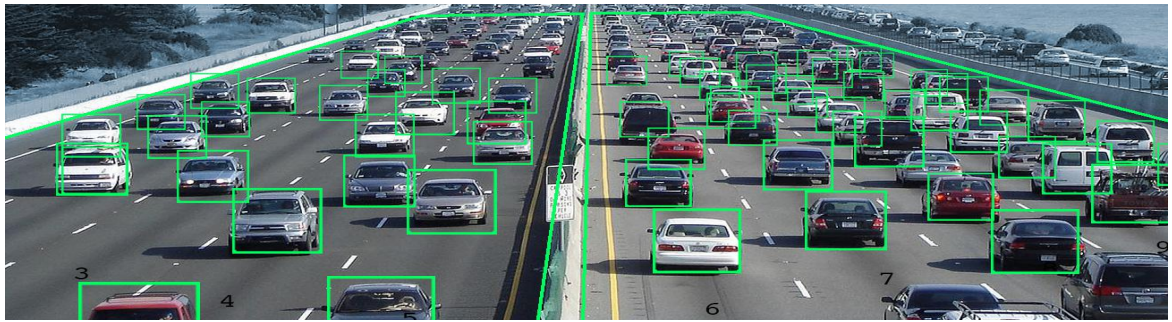
·过1000万市民的多维度信息存储，毫秒级查询反馈



交通/安防 监控视频影像管理

在公安与交通行业，针对视频卡口的大数据存储、分析与应用一直以来是最受关注的主题。借助SequoiaDB半结构化对象存储、分布式横向扩展能力以及非结构化影像存储引擎，交通部门可以从卡口视频文件中提取出的车牌信息、位置信息、以及时间信息按照三个维度汇总，进行道路拥堵预测、车辆轨迹跟踪、套牌车监控、尾随车辆监控等多种安防措施。

- 兼容各采集系统的数据，统一汇总统一管理，支持多索引多维度查询，查询数据ms级返回。
- 并发处理能力高，支持多警种多业务多用户同时使用，系统反应快速。
- 系统架构简单，易于部署和维护，易于横向扩展。
- 存储层和分析层松耦合，均可弹性扩充；数据加载快，分布式计算分析效率高。
- 可灵活配套多种分析工具；SQL通用性高，适合警务操作人员灵活查询；
- 系统实时计算实现车辆实时跟踪。



OTA旅游 /电商 - 多类型数据混合存储

互联网应用的特点，带来了几项重要的挑战。

- 数据量大
- 业务增长快
- 数据类型多样

途牛旅游网“资源系统”的另一个核心业务模块，负责存储和记录所有的旅游方案相关的资源信息，包括酒店，机票，门票，火车票，汽车票，地接，当地服务等。通过使用巨杉数据库，在满足海量存储的同时，也能实现高效的在线资源查询。

使得多个核心系统计算量从去年同期的每天1亿次，增加到今年每天100亿次以上。