DTCC

第十一届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2020

架构革新 高效可控

IT168.com ChinaUnix.net ITPUB

北京国际会议中心 | 2020/12/21-12/23

分布式列存数据库在同程艺龙的应用

王勇 2020年12月

目　　录

列存数据库选型

架构及核心特性

列存的技术实现

公司的业务应用

业界纷纷布局NewSQL

01 2013 Amazon Aurora

02 2014 Google Spanner

03 2016 PingCap TiDB CockroachDB

04 2017 阿里OceanBase
OceanBase 1.0版本在支付宝账号系统上线，支撑起12万笔/秒的支付峰值。

05 2018 华为GaussDB
2019年5月15日，华为在北京发布了人工智能原生的名为“高斯（GaussDB）”的数据库软件。

06 2019 百度BaikalDB
BaikalDB在百度凤巢广告平台上线

核心技术对比

NewSQL数据库	公司	年份	大事记	数据分片	数据复制	存储引擎	并发控制	隔离级别	开源情况	开发语言
Spanner	Google	2013	发布论文	分布式kv	Paxos	LSM-Tree PAX行列混存	MVCC+SS2PL	Serializable	云服务	C++
OceanBase	蚂蚁金服	2016	V1.0	分布式kv	Paxos	LSM-Tree	MVCC+SS2PL	Serializable	ToB	C++
TiDB	PingCAP	2017	V1.0	分布式kv	Raft	行存：LSM-Tree 列存：MergeTree	MVCC+SS2PL	SSI	Apache 2.0	Go/Rust
ShardingSphere	Apache	2016	首次开源	分库分表	Paxos	InnoDB	MVCC+2PL	SI	Apache 2.0	JAVA
TDSQL	Tencent	2017	分布式事务	分库分表	Raft	InnoDB	OCC+2PL+MVCC	RR	云服务	C++
BaikalDB	Baidu	2018	首次开源	分布式kv	Raft	行存：LSM-Tree 列存：LSM-Tree	MVCC+2PC	RC	Apache 2.0	C++

技术选型

纯自研：能力有限，投入有限

纯开源：无法及时满足定制化需求

云服务：安全与成本考虑，短期内核心业务自建IDC，k8s化

半自研：我们的选择，不重复造轮子，主体功能交由社区完成，集中有限力量满足公司需求，可供选择的NewSQL有： TiDB，BaikalDB，CockRoachDB等。

背景相似：BaikalDB来源于百度凤巢广告业务团队，由于广告业务的增长走过了从单机到分库分表到分布式的全过程，而我们面临类似的问题。

经受考验：已经有百度广告平台多个业务实际使用经验，千级别集群节点，PB级数据规模，我们跟进使用，风险可控。

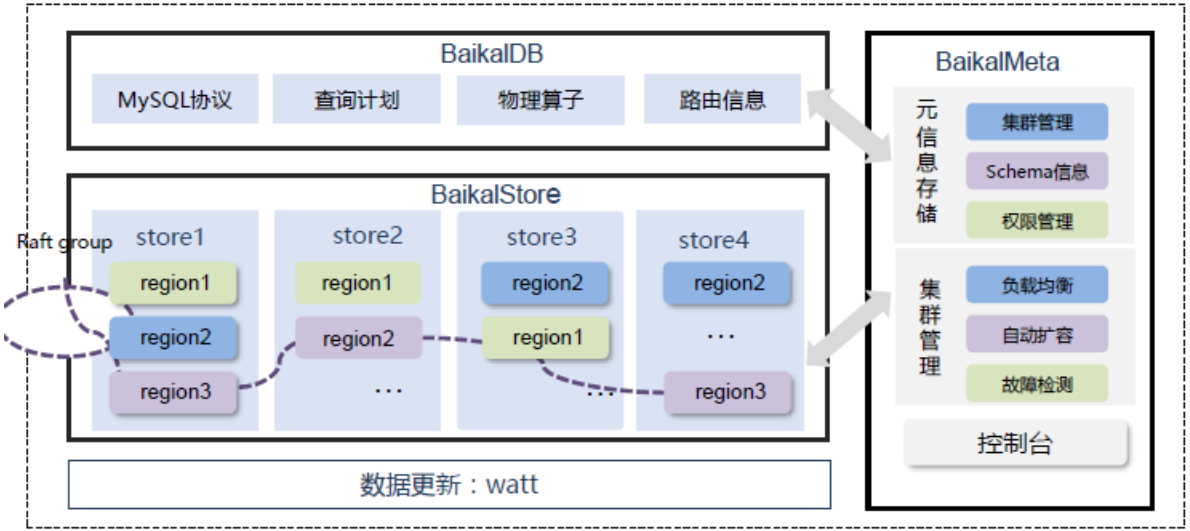
技术栈匹配：BaikalDB（c++实现，10万行代码精炼），依赖少而精（brpc，braft，rocksdb），社区友好，部署简单，技术栈匹配。

特性比较完善：基本满足我们需求，我们可以专注于满足公司需求。

基础架构

BaikalDB是面向商业业务系统的新一代存储系统，baikaldb.com

- ✓**全自主**管理，线性扩展，**应用无感知**
- ✓**高可用**，自动故障恢复和均衡
- ✓兼容MySQL协议
- ✓高性能加列加索引，应对业务需求变更
- ✓局部+全局二级索引
- ✓异地IDC部署，分布式事务，多表Join能力



BaikalDB：SQL查询引擎

- ✓ MySQL DDL、DML功能
- ✓ 分布式的聚合筛选

BaikalMeta：元信息管理

- ✓ 元信息存储，集群管理功能

BaikalStore：存储引擎

- ✓ 底层引擎：RocksDB
- ✓ 支持自定义索引，倒排索引
- ✓ 多region划分，multi-raft group

核心特性

BaikalDB

诞生于百度广告十年来的业务需求，
设计灵感来源于 Google F1/Spanner，
基础设施构架于brpc，braft，rocksdb，
结合了分布式与关系型数据库的最佳实践，
是一款开源云原生分布式关系型HTAP数据库。

https://github.com/baidu/BaikalDB

01 强一致

分布式事务
ACID强一致

02 高可用

- Raft协议，少数故障，数据不丢，服务不停
- RPO = 0；RTO < 30s

03 高扩展性

- 水平扩展，在线缩扩容，服务不停
- 动态Schema 30s完成
- 行> 1G； 列>1k

04 高性能

- 准内存数据库性能
- QPS > 3.6万/秒

05 高兼容性

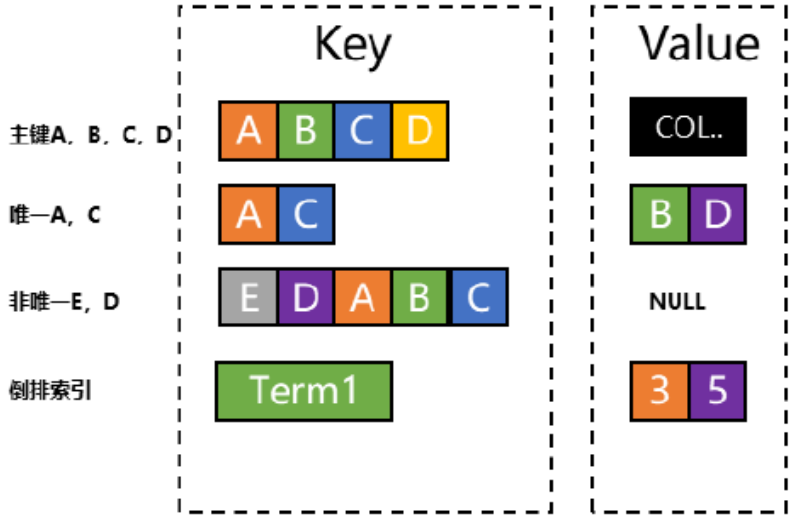
- 兼容MySQL5.6大部分功能

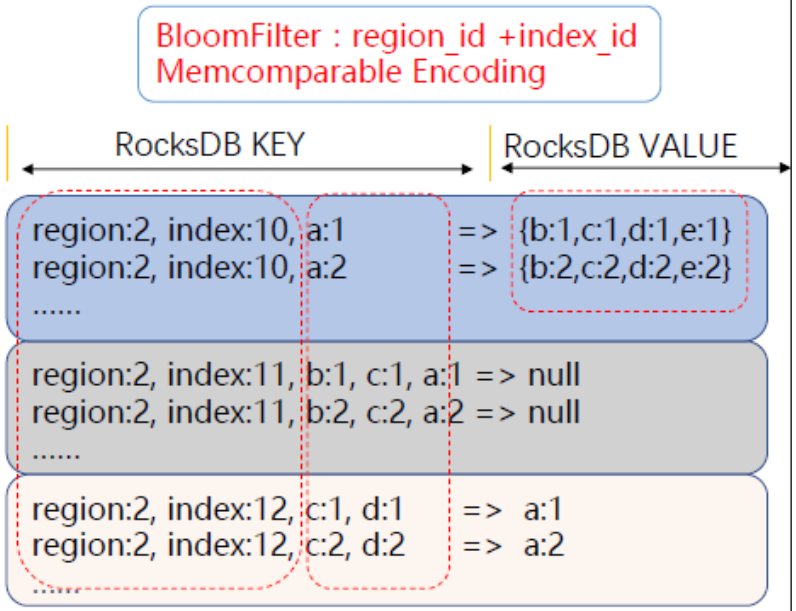
06 成本

- 基于普通PC服务器
- 充分发挥新硬件能

存储模型

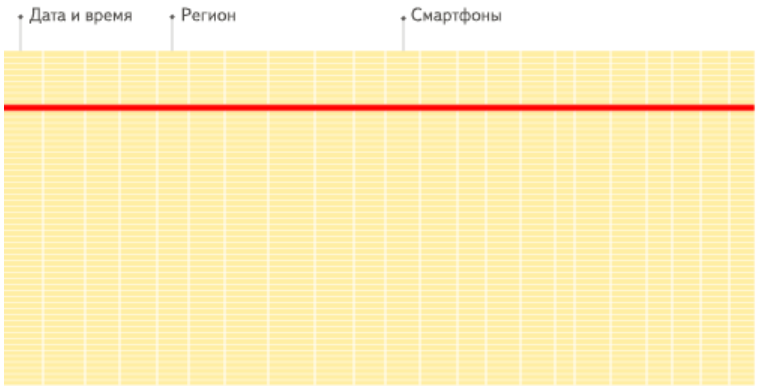
- ✓ RocksDB作为KV底层存储
- ✓ 支持索引：主键聚簇索引、联合索引、唯一索引、普通索引、倒排索引
- ✓ Value采用pb格式编码，业务可以灵活加列
- ✓ 动态解析pb格式，无需解析所有列



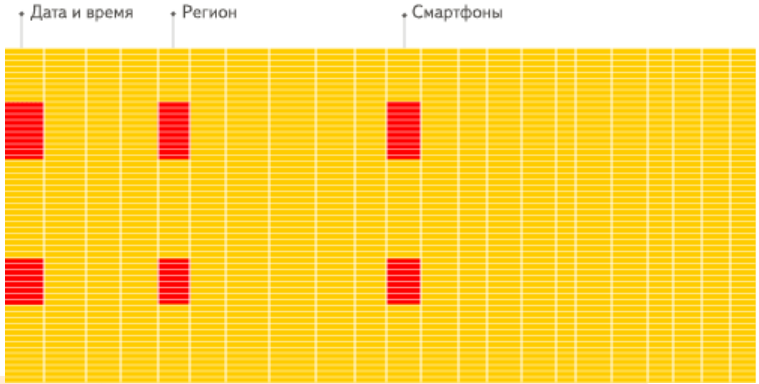


列式存储

行式存储



列式存储



减少IO：宽表少列情况下只读需要列，更新部分字段时只写涉及列

利于压缩：列针对性压缩同构的列值连续摆放在一起，字节上重复的部分很多，因而信息熵很低，可以充分利用压缩

利于缓存：I/O的降低，提高缓存效率

向量化查询：以块的形式处理数据

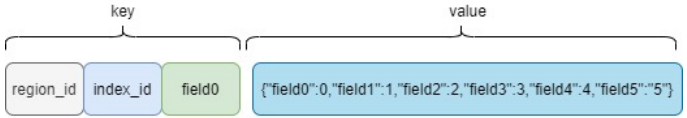
延迟物化：只解析与SQL有相关性的列并过滤，减少非必须列的扫描

设计实现

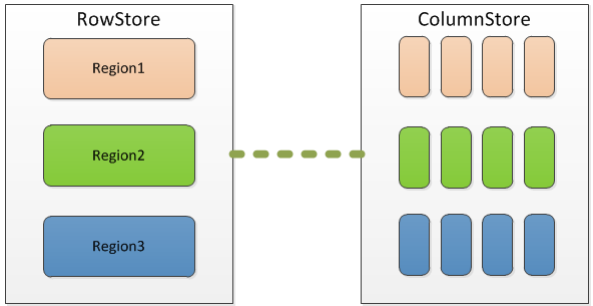
行式

primary

```
Key format: region_id(8 bytes) + table_id(8 bytes) + primary_key_fields;
Value format: protobuf of all non-primary key fields;
```



主键编码



列式

primary

```
Key format: region_id(8 bytes) + table_id(8 bytes) + primary_key_fields;
Value format: null;
```

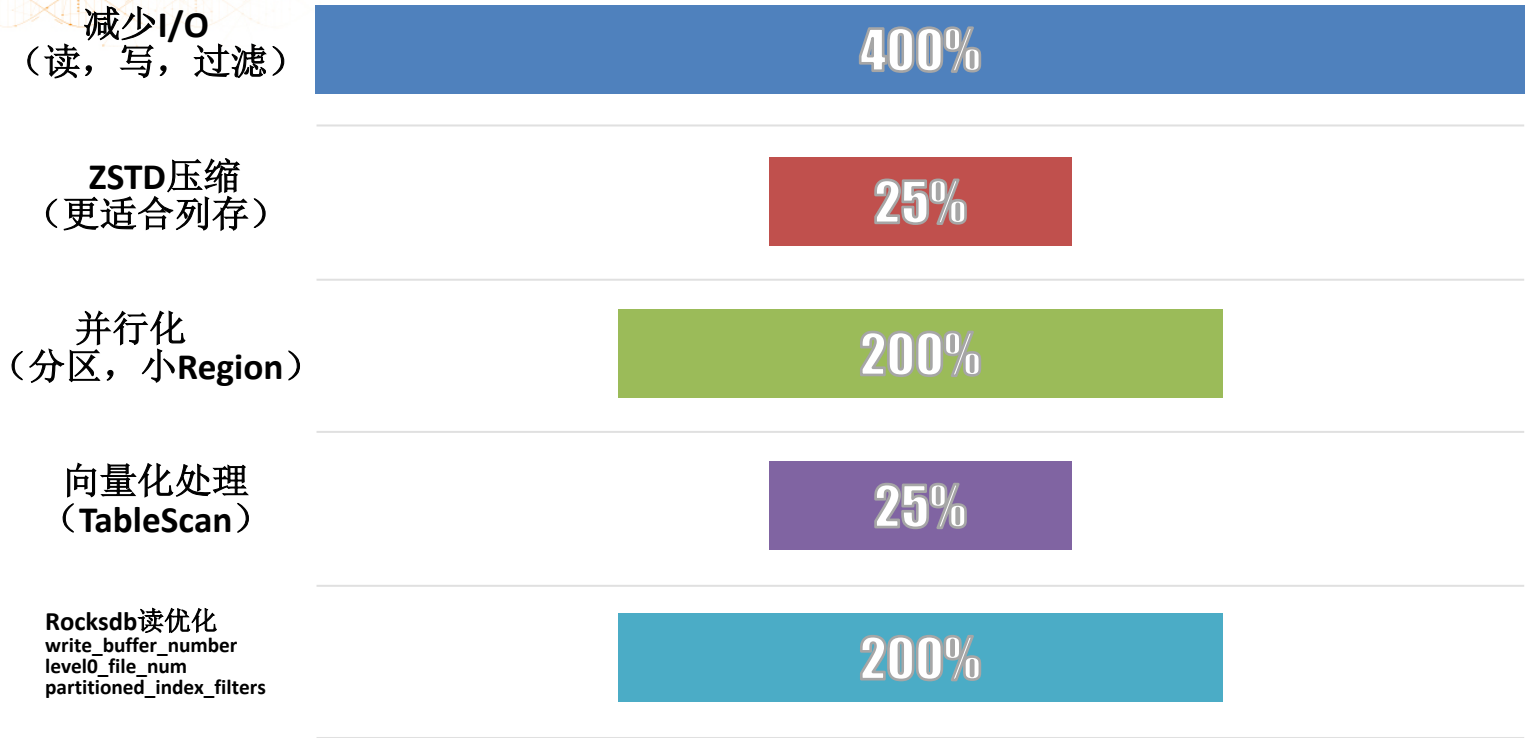
primary_columns

```
Key format: region_id(8 bytes) + table_id(4 bytes) + field_id(4 bytes) + primary_key_fields;
Value format: one of non-primary key fields encode value;
```

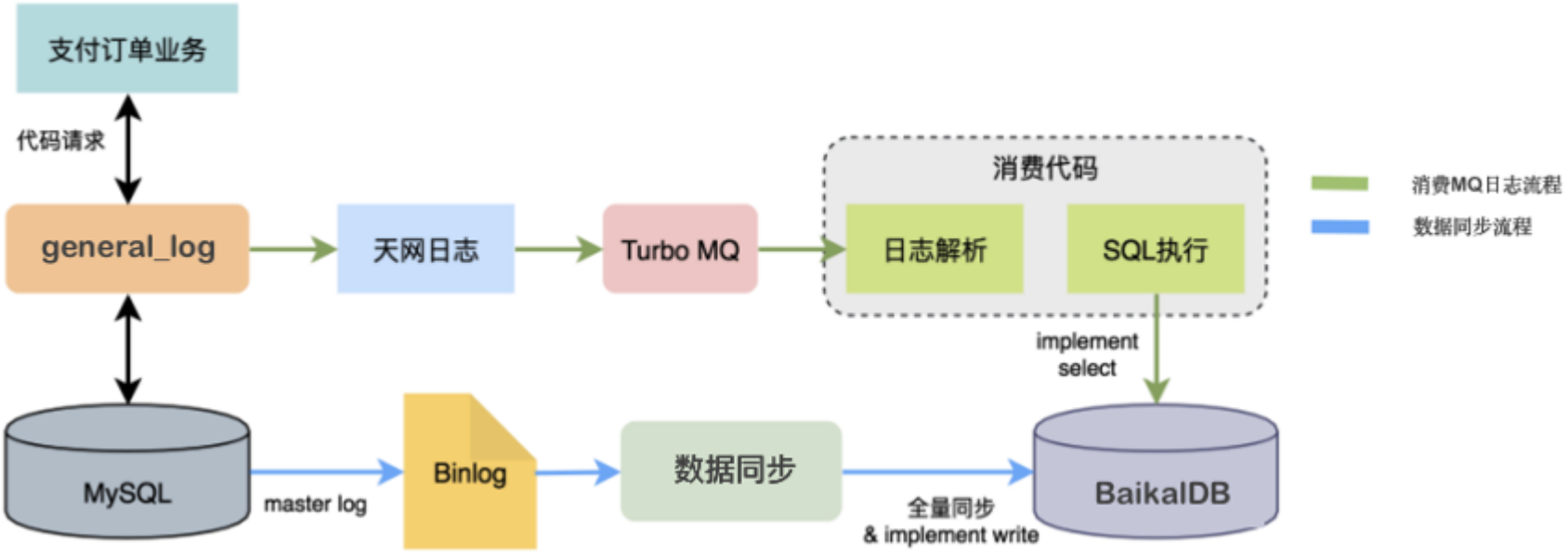
```
/8000000000000006/80000006/80000002/80000000 : AFRICA
 ^---------------- ^-------- ^-------- ^--------   ^-----
 |                 |         |         |          |
 Region ID (6)     |         |         |          Value encoding (AFRICA)
                   |         |         |
                 Table ID (6)        |
                             |       |
                           Column ID (2)
                                     |
                                   Primary key (0)
```

https://github.com/baidu/BaikalDB/pull/52

相关优化


减少I/O
（读，写，过滤）
400%
ZSTD压缩
（更适合列存）
25%
并行化
（分区，小Region）
200%
向量化处理
（TableScan）
25%
Rocksdb读优化
write_buffer_number
level0_file_num
partitioned_index_filters
200%

迁移与测试

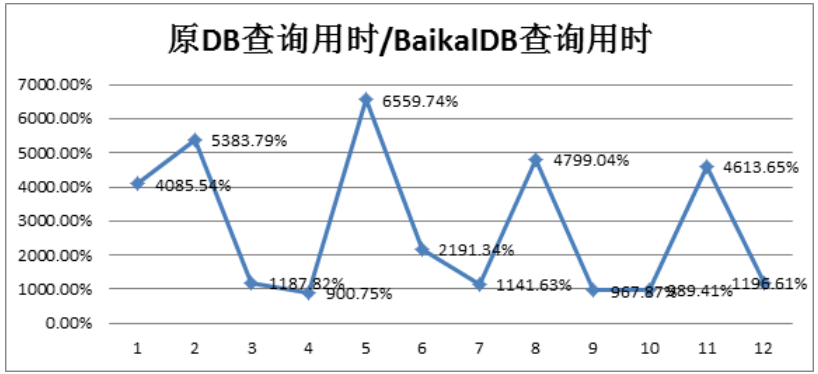

支付订单业务
代码请求
general_log
天网日志
Turbo MQ
消费代码
日志解析
SQL执行
消费MQ日志流程
数据同步流程
implement
select
MySQL
master log
Binlog
数据同步
全量同步
& implement write
BaikalDB

上线效果

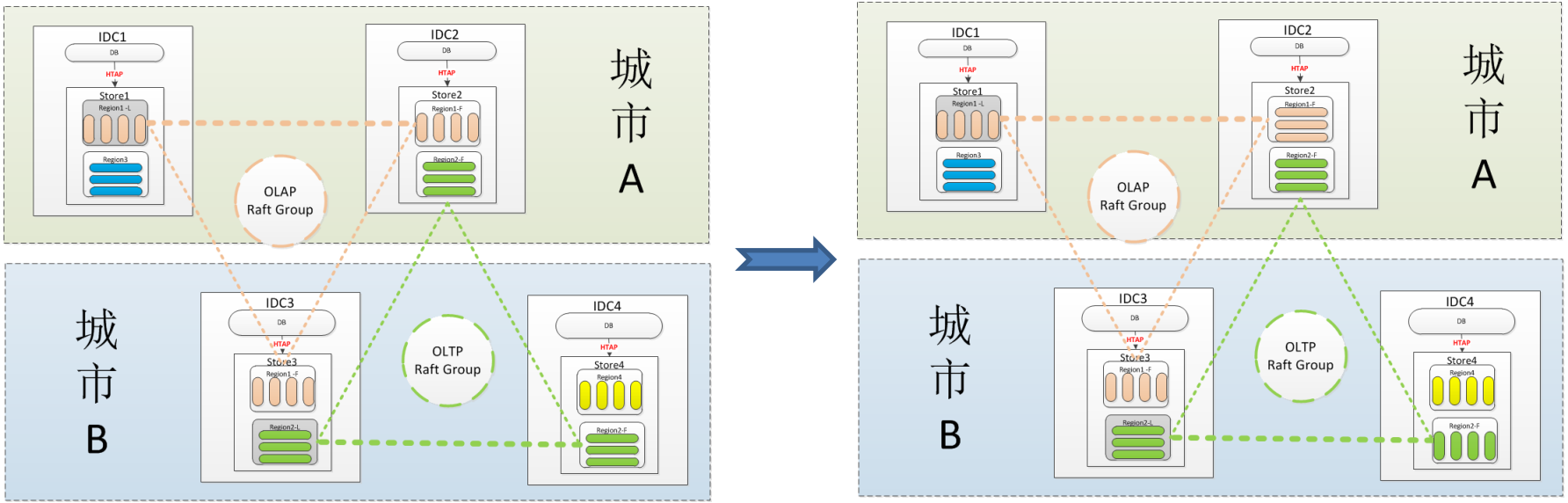
- 实时获取线上业务运行数据
- 生成关于健康状态的相关指标，如果指标超出阈值，则触发报警功能。
- 数据表约50列20亿行
- 查询sql均为聚合类查询，检索列数不超过4列，查询条件为一定时间区间的范围查询
- Sql示例：

```
SELECT project_code, count(*) as count
FROM trajectory
WHERE created_on > '2020-01-11'
AND    created_on <= '2020-01-12'
AND    flag = 2 group by project_code;
```

场景	频率	现状(ms)	BaikalDB(ms)	现状/BaikalDB
订单量	一分钟一次	726	17.77	4085.54%
成功率	一分钟一次	1003	18.63	5383.79%
支付宝扫码的请求量	一天一次	39282	3307.07	1187.82%
健康度监控-1分钟回调率	一分钟一次	962	106.8	900.75%
健康度监控-30秒成功率	30s一次	818	12.47	6559.74%
健康度监控-1分钟入库率	一分钟一次	339	15.47	2191.34%
微信web扫码的请求量	一天一次	36369	3185.7	1141.63%
成功率2	一分钟一次	1003	20.9	4799.04%
订单量2	一分钟一次	1003	103.63	967.87%
健康度监控-1分钟回调率2	一分钟一次	962	97.23	989.41%
健康度监控-30秒支付成功率2	30s一次	818	17.73	4613.65%
健康度监控-1分钟支付入库率2	一分钟一次	339	28.33	1196.61%



部署情况


城市A
城市B
IDC1
DB
HTAP
Store1
Region1 -L
Region3
IDC2
DB
HTAP
Store2
Region1-F
Region2-F
OLAP
Raft Group
IDC3
DB
HTAP
Store3
Region1 -F
Region2-L
IDC4
DB
HTAP
Store4
Region4
Region2-F
OLTP
Raft Group

THANKS