DTCC 2010-2022

第十三届中国数据库技术大会

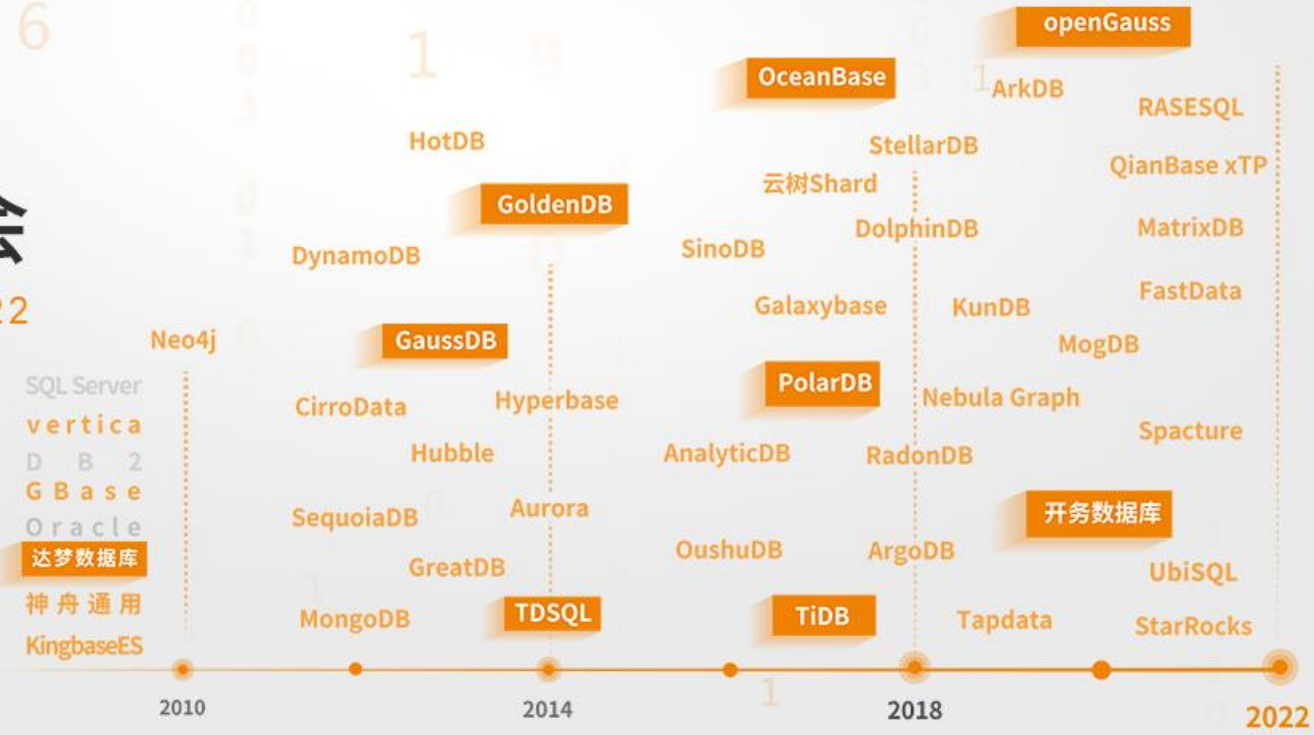
DATABASE TECHNOLOGY CONFERENCE CHINA 2022

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



Apache Pegasus的应用与实践

王伟 小米 高级软件工程师

讲师简介

王伟
高级软工程师

高级软件工程师，KV存储团队负责人

曾就职于百度、SHAREit，先后负责分布式Trace、表格存储、KV存储系统的研发。目前就职于小米，负责KV存储方向。专注于存储系统、性能优化等技术，对分布式、Trace、存储等技术领域有较深的技术积累，及丰富的存储系统实践经验。

目录

- 项目介绍
- 实现原理
- 功能特性
- 应用实践

项目介绍

项目介绍

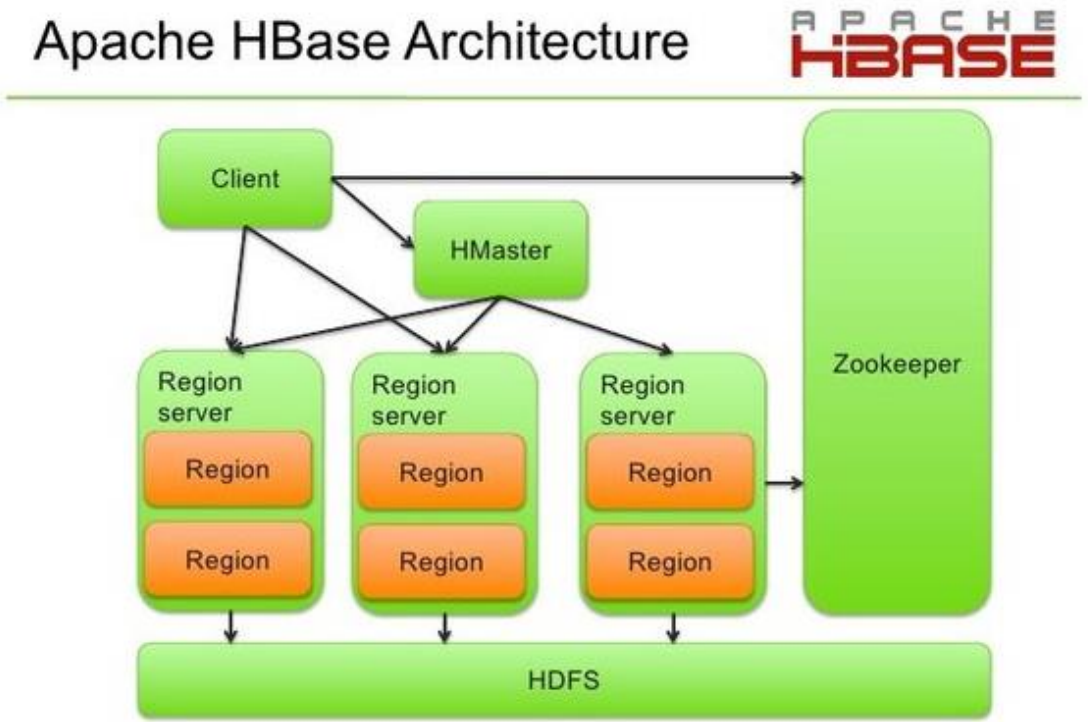
HBase的问题

数据非本地保存（性能较差）

Failover时间长（可用性低）

JVM垃圾回收（长尾问题）

Hadoop生态（运维困难）

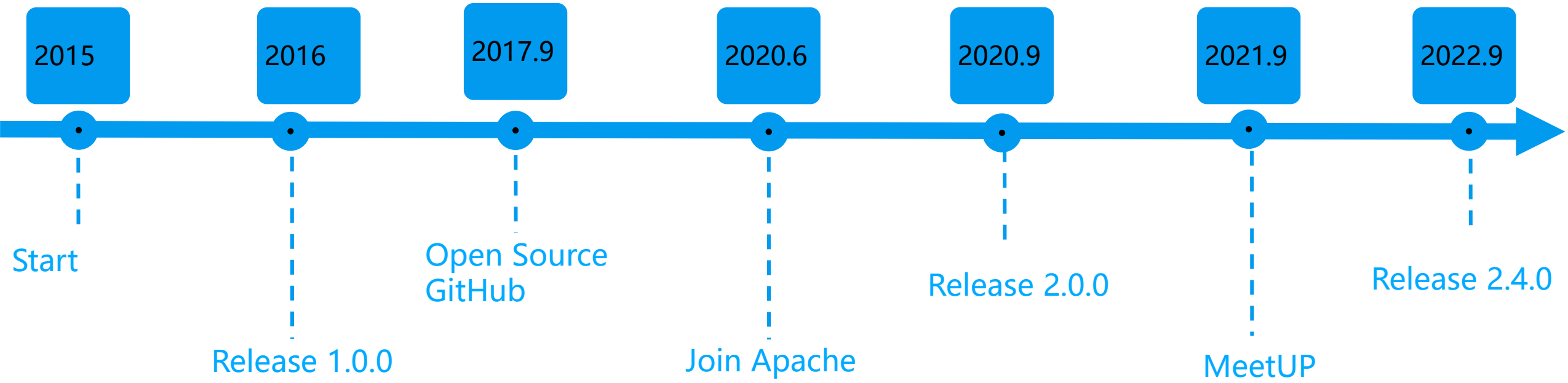


项目介绍

Apache Pegasus

一个高可用、高性能、强一致、轻量级分布式KV存储系统

项目介绍


2015
2016
2017.9
2020.6
2020.9
2021.9
2022.9
Start
Release 1.0.0
Open Source
GitHub
Join Apache
Release 2.0.0
MeetUP
Release 2.4.0

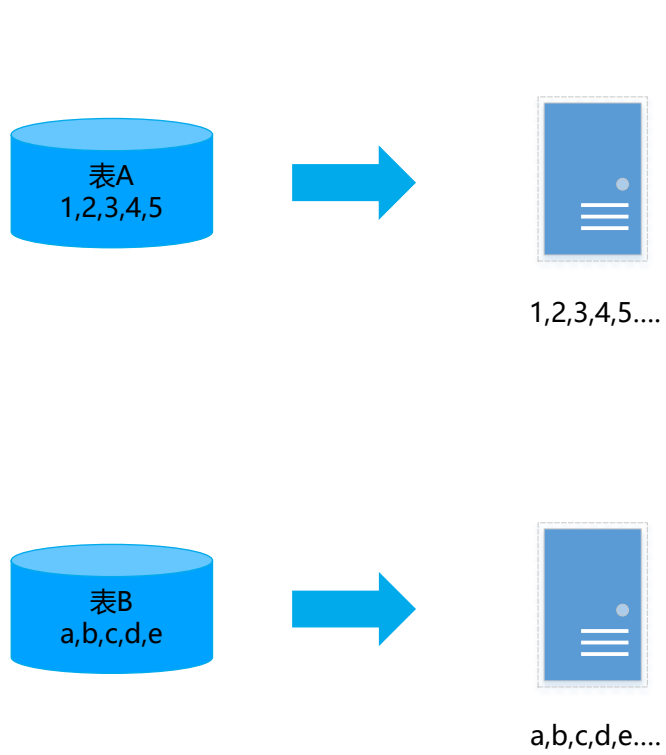
项目介绍

架构设计中的一些考量

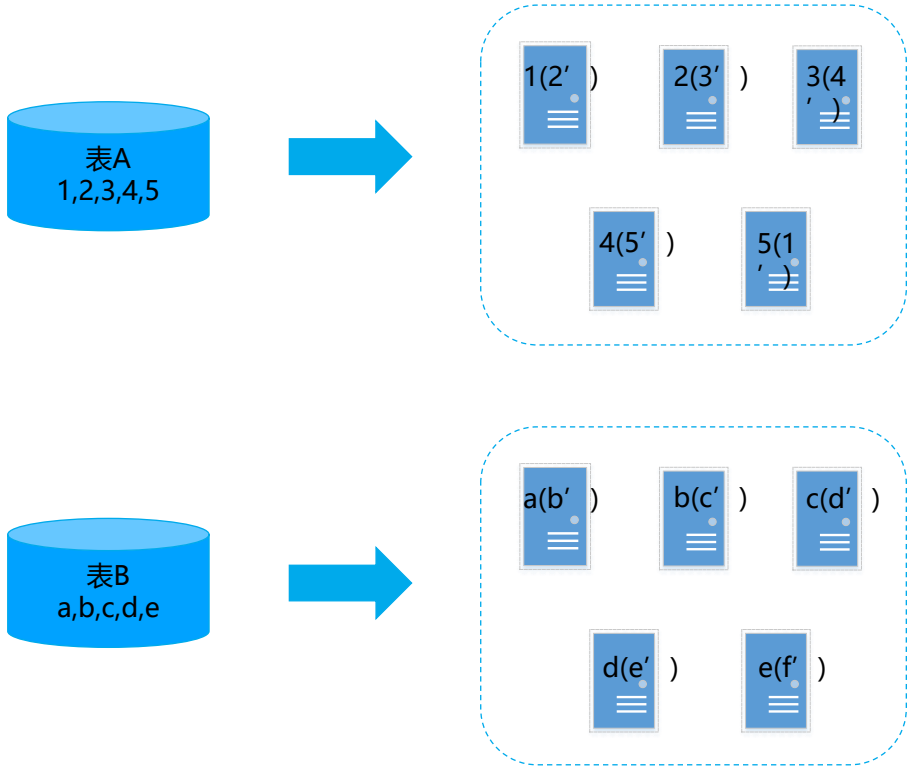
- 实现语言：C++、Java、Golang
- 存储介质：HDD、SSD、MEM
- 单机引擎：RocksDB、LevelDB
- 数据视图：KV、Tabular
- 数据分布：Hash、Range、一致性Hash
- 系统架构：中心化、去中心化
- 一致性协议：Raft、Paoxs、？

实现原理

实现原理—分布式系统概念



传统数据库

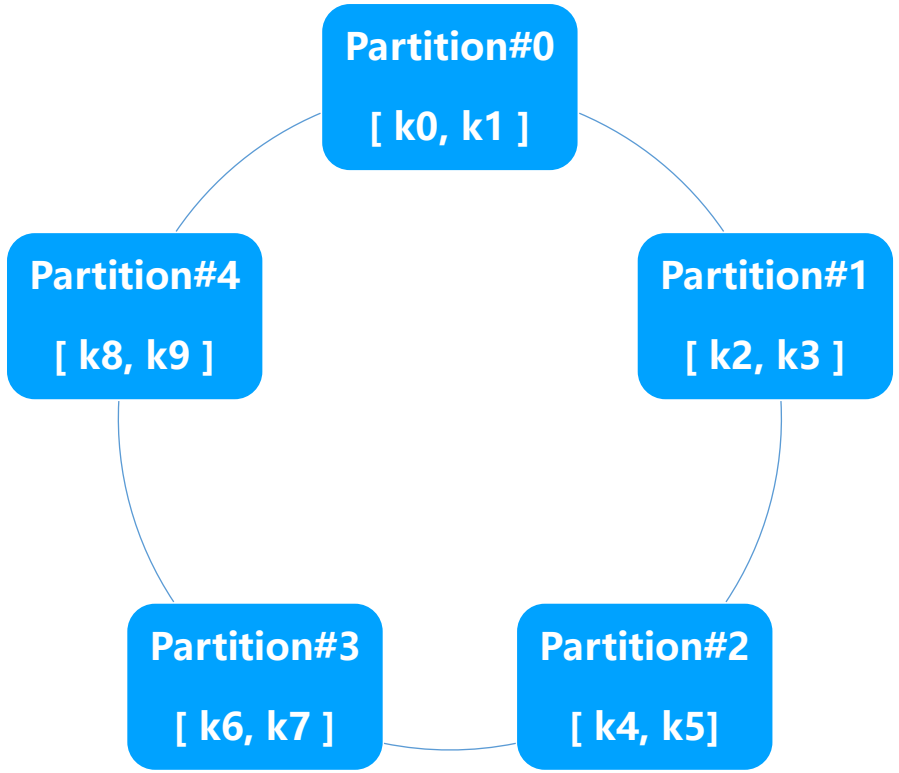


分布式数据库

实现原理—Partition

全局有序

Key Range



[k0<k1<k2<k3<k4<k5<k6<k7<k8<k9]

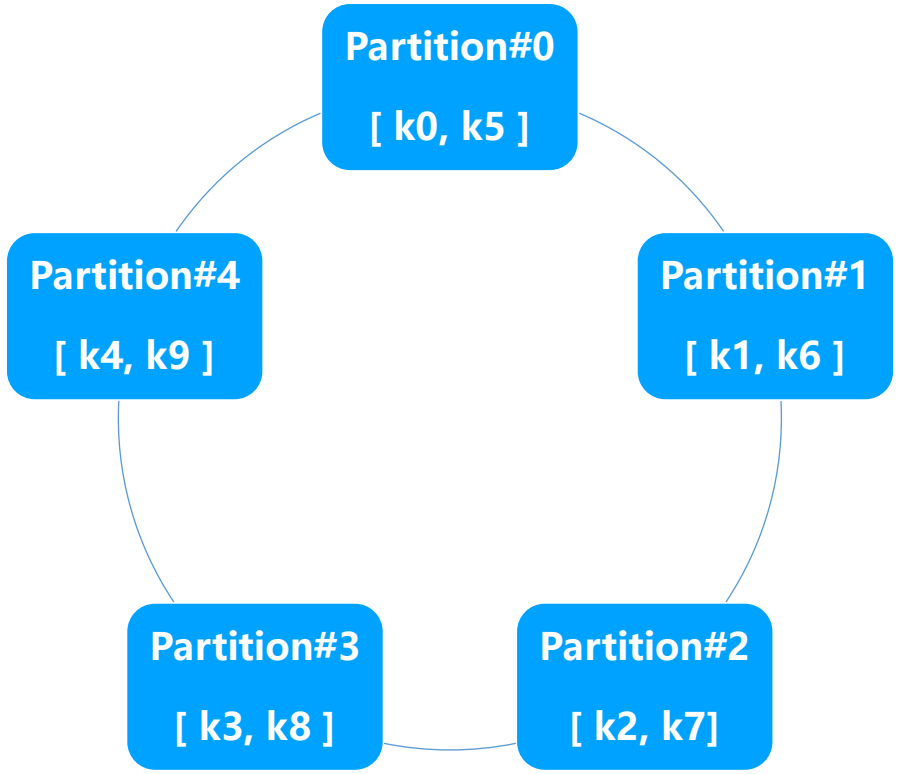
实现原理—Partition

Hash分环

全Key Hash

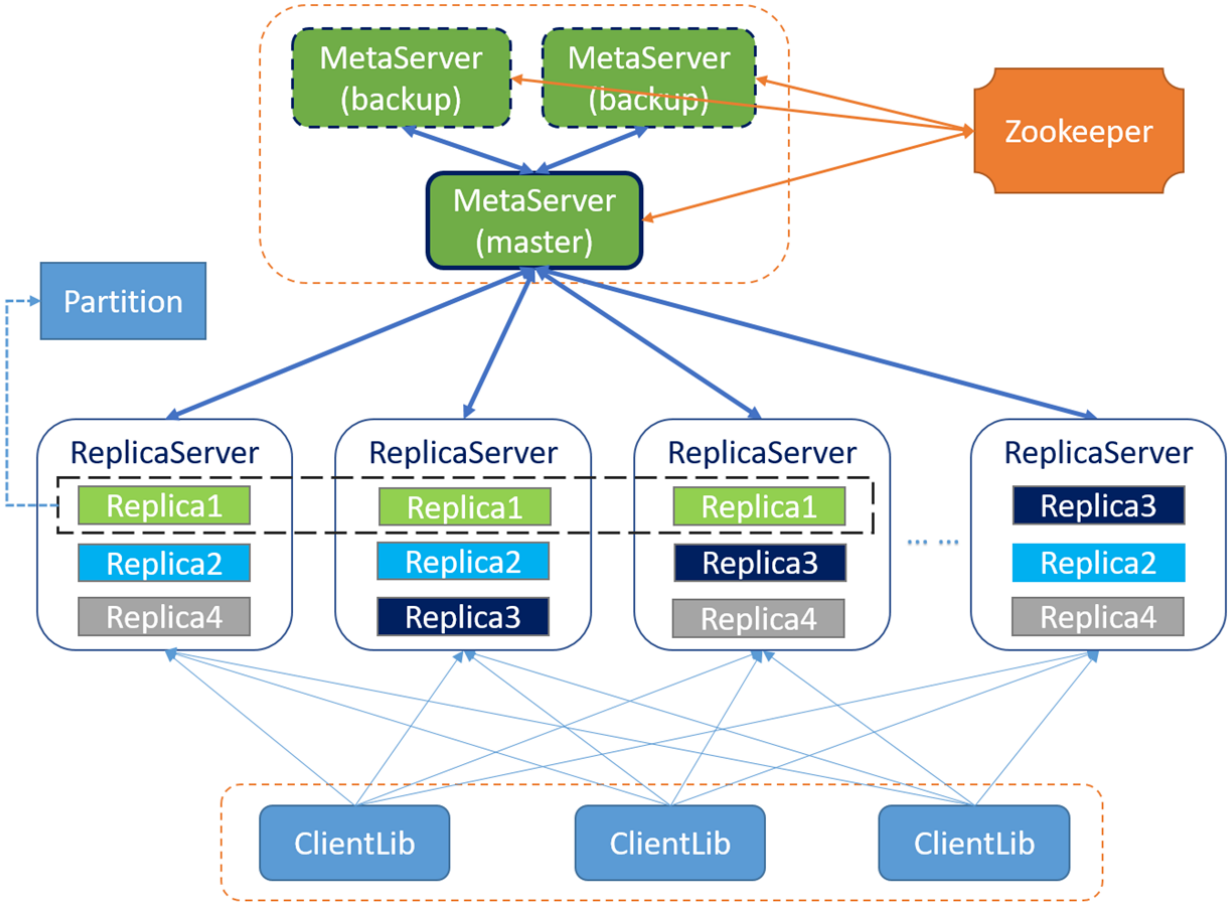
前缀Hash

Pegasus采用



[k0<k1<k2<k3<k4<k5<k6<k7<k8<k9]

实现原理—系统架构



Meta server

- 集群控制及配置管理
- 通过ZK选主实现高可用

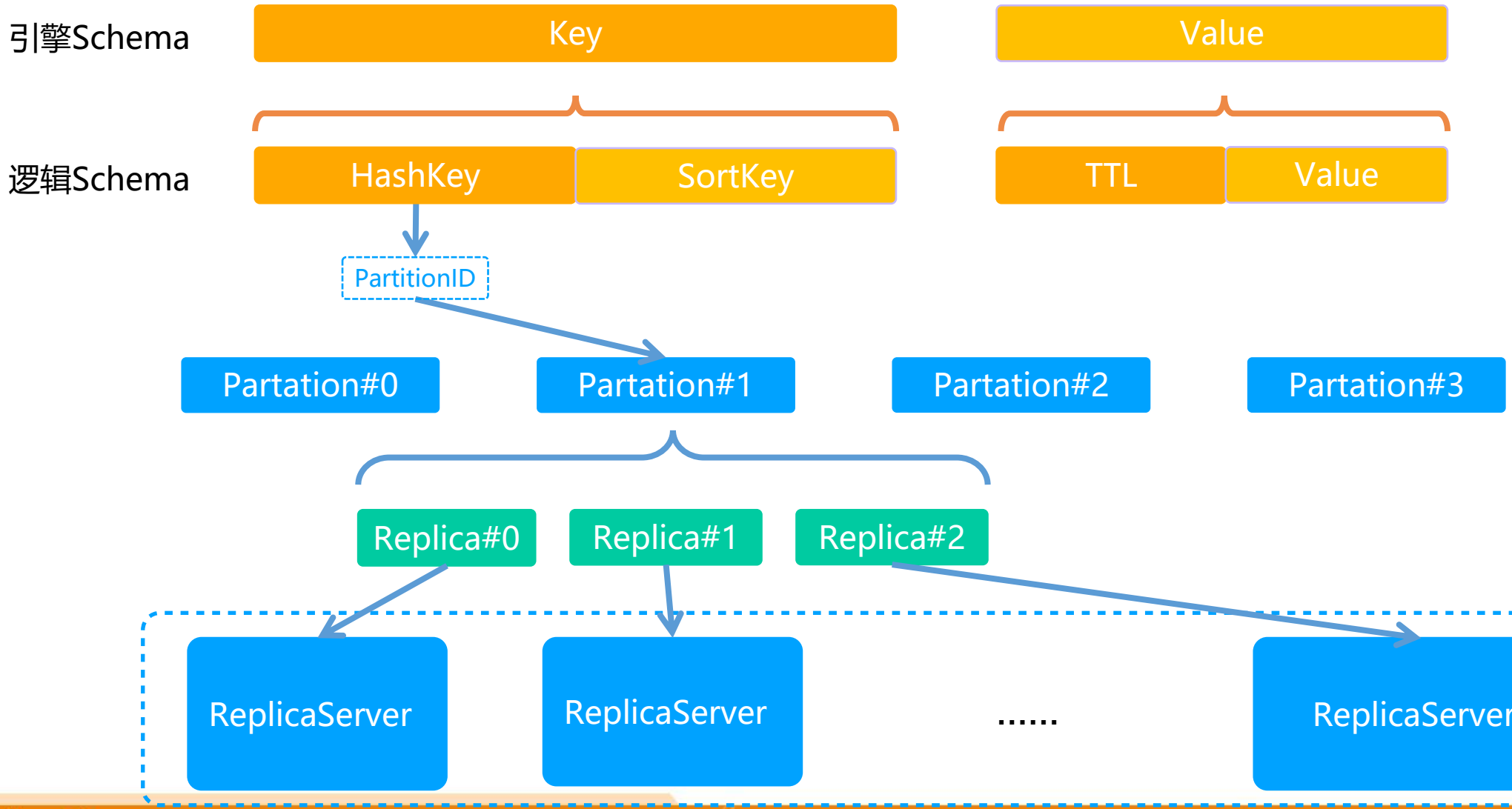
Replica server

- 数据节点，SSD存储介质
- Hash分片，个数固定
- PacificA实现数据强一致
- 单机引擎RocksDB
- 三副本高可用
- Replica⇔RocksDB

OverAll

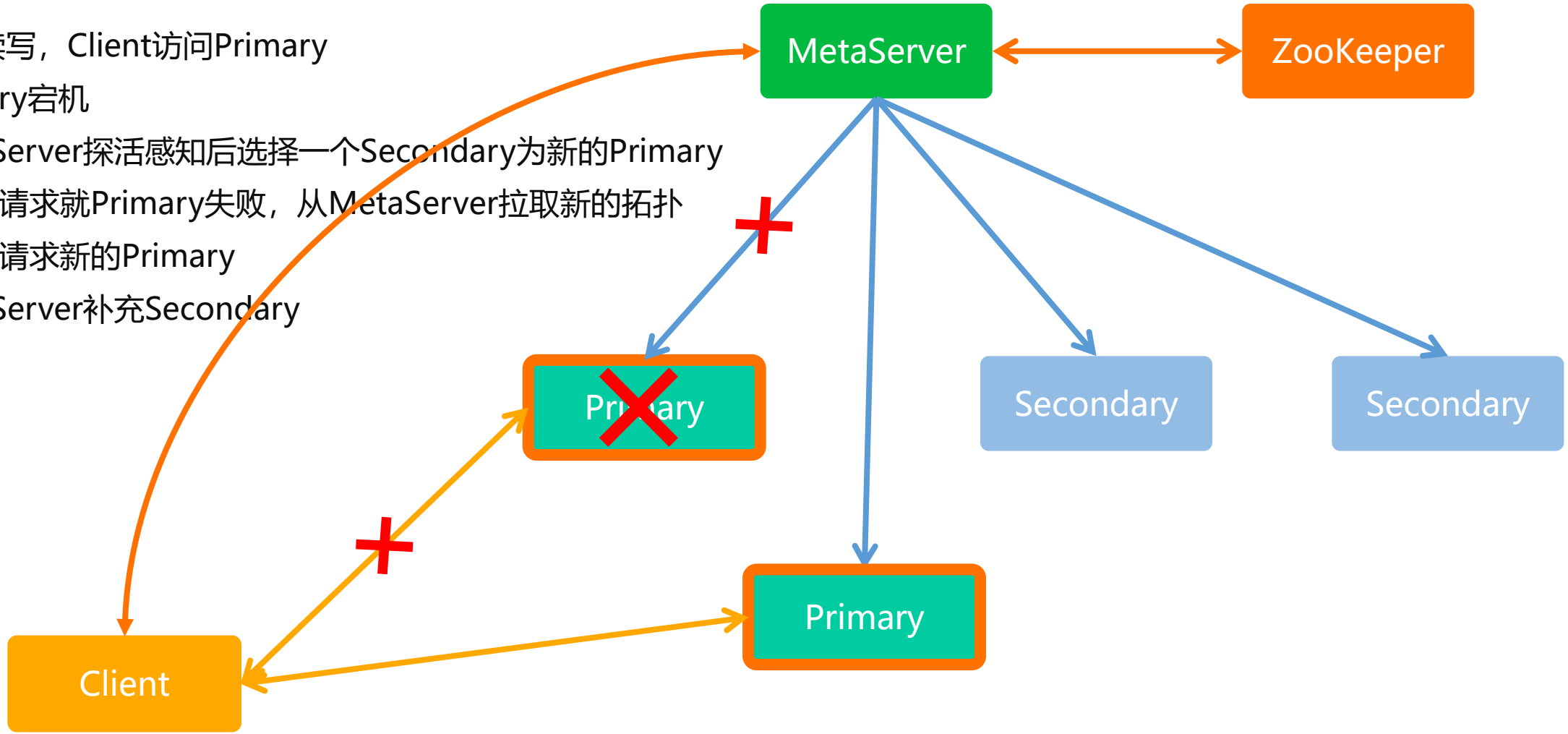
- C++，高性能，无GC
- Key-Value，易实现
- HashKey+SortKey，Tabular支持

实现原理—数据模型


引擎Schema
Key
Value
逻辑Schema
HashKey
SortKey
TTL
Value
PartitionID
Partation#0
Partation#1
Partation#2
Partation#3
Replica#0
Replica#1
Replica#2
ReplicaServer
ReplicaServer
.....
ReplicaServer

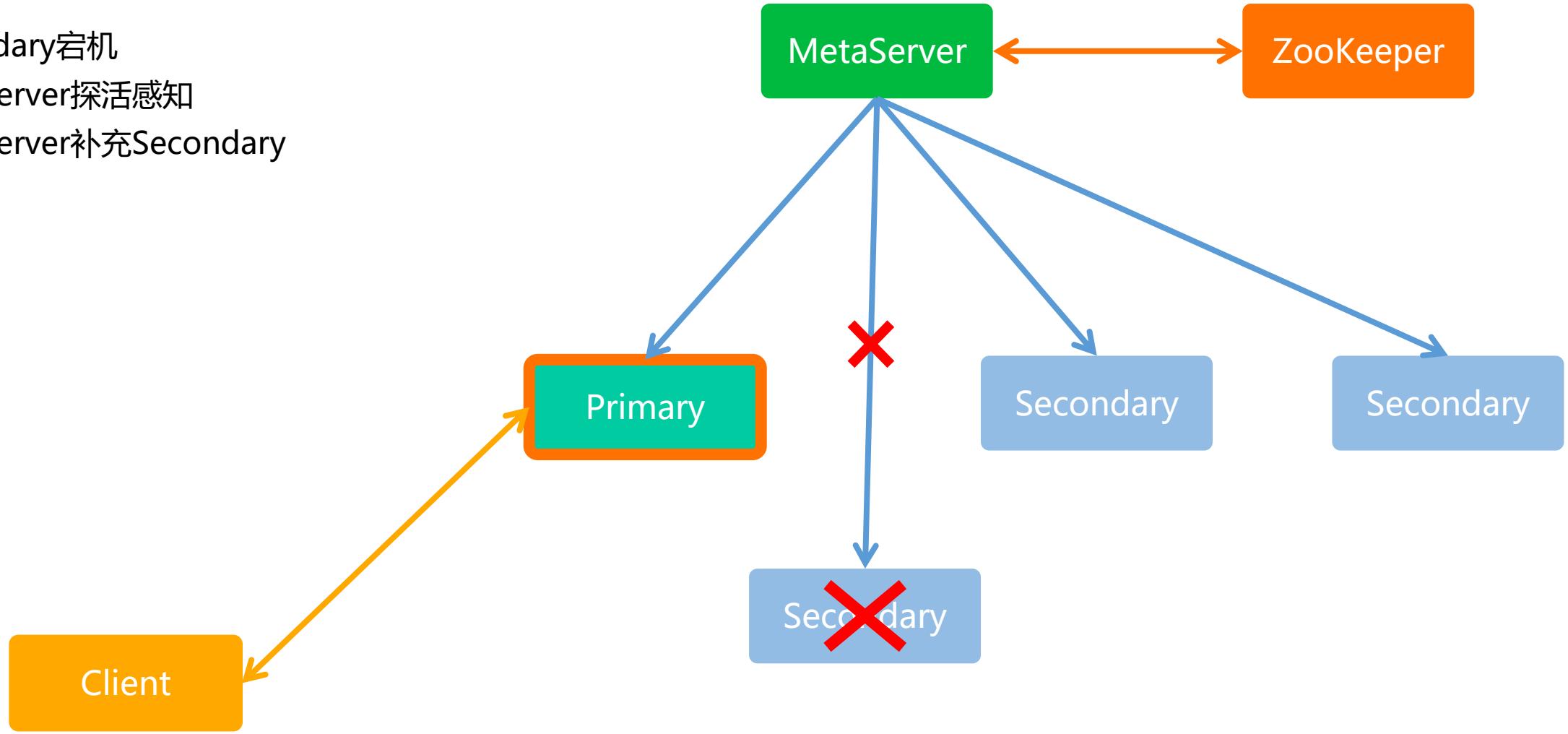
实现原理—高可用Primary

1. 正常读写，Client访问Primary
2. Primary宕机
3. MetaServer探活感知后选择一个Secondary为新的Primary
4. Client请求就Primary失败，从MetaServer拉取新的拓扑
5. Client请求新的Primary
6. MetaServer补充Secondary



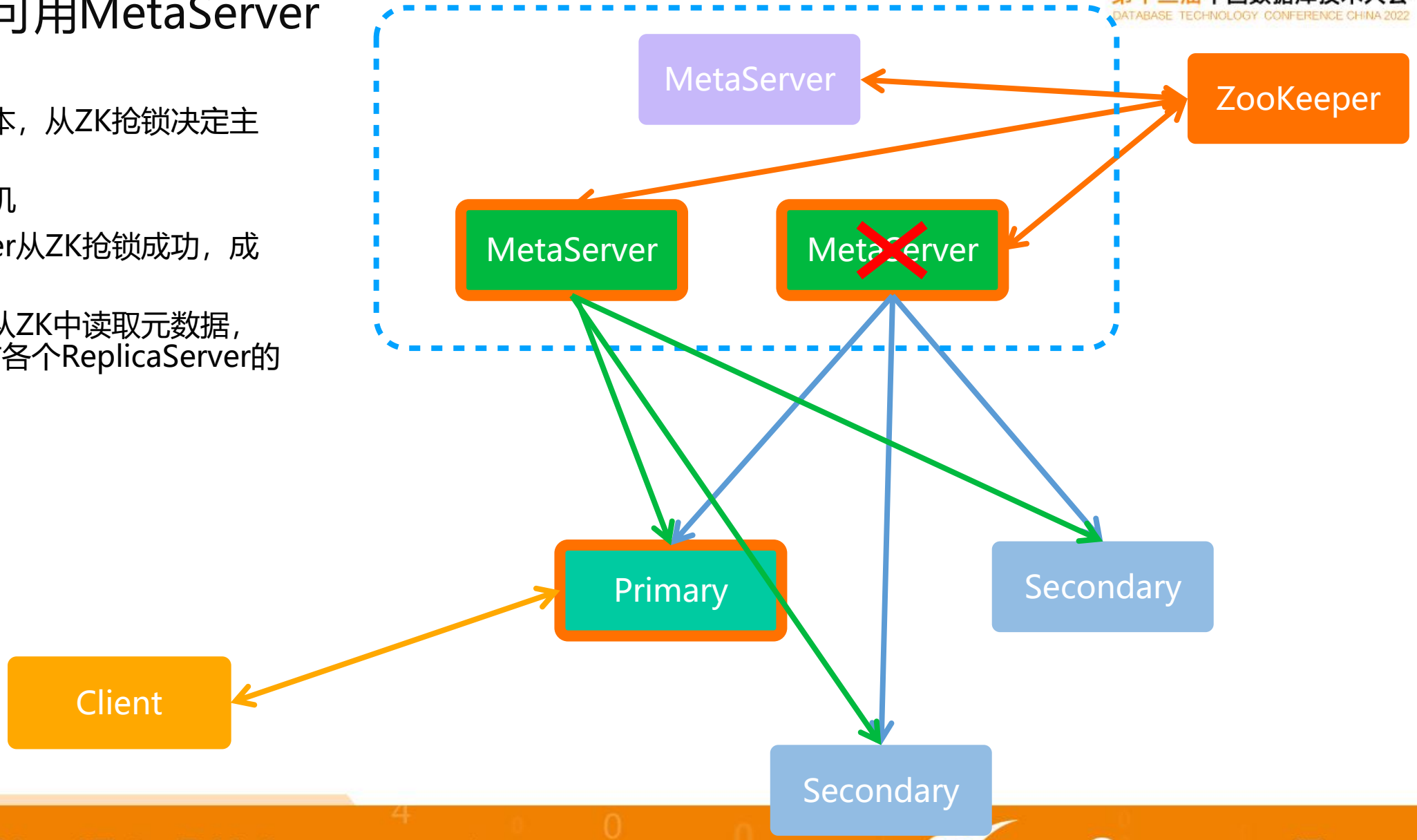
实现原理—高可用Secondary

1. Secondary宕机
2. MetaServer探活感知
3. MetaServer补充Secondary



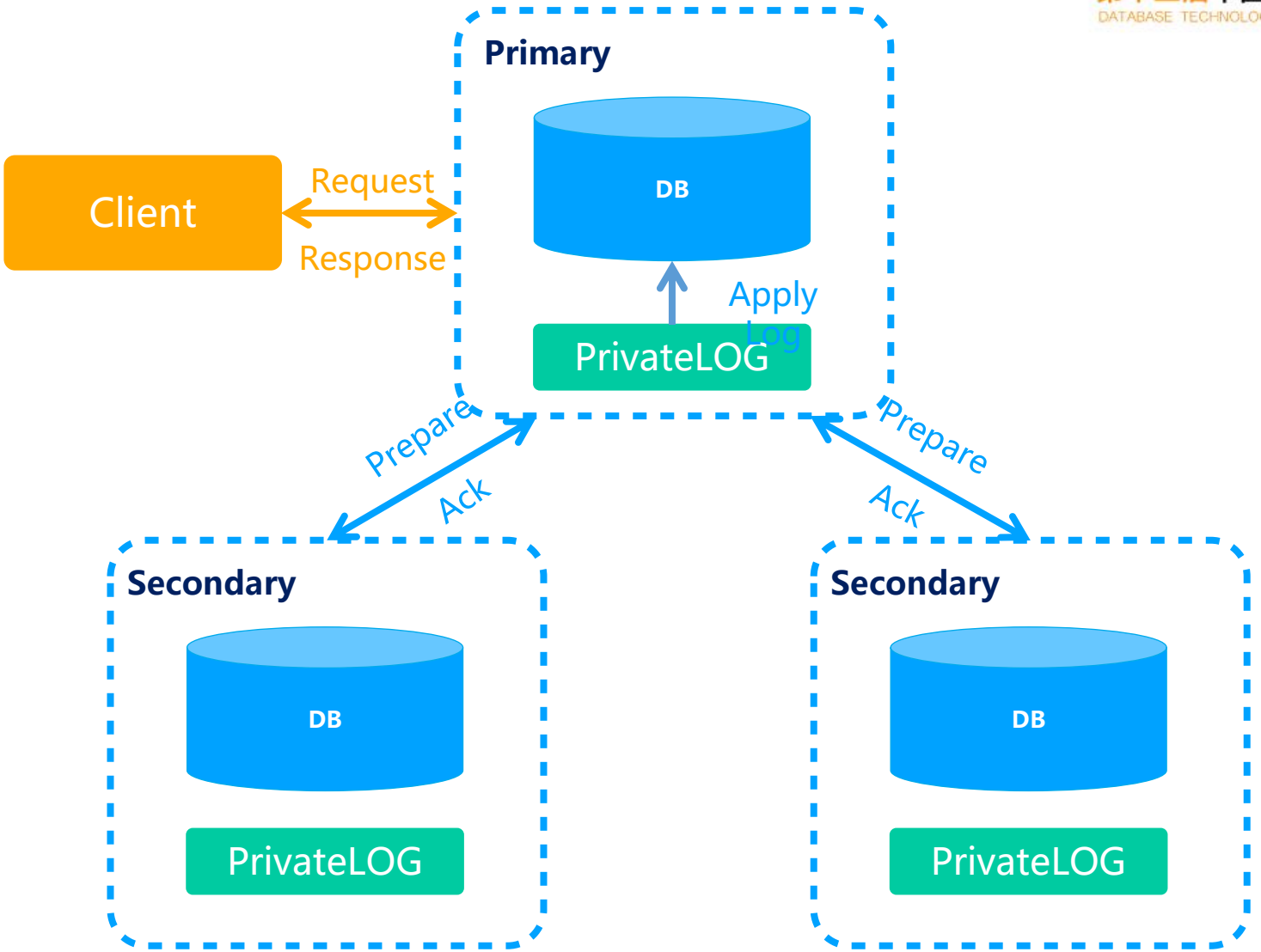
实现原理—高可用MetaServer

1. MetaServer三副本，从ZK抢锁决定主从
2. 主MetaServer宕机
3. 另一个MetaServer从ZK抢锁成功，成为新主
4. 新主MetaServer从ZK中读取元数据，接管集群，保持与各个ReplicaServer的心跳



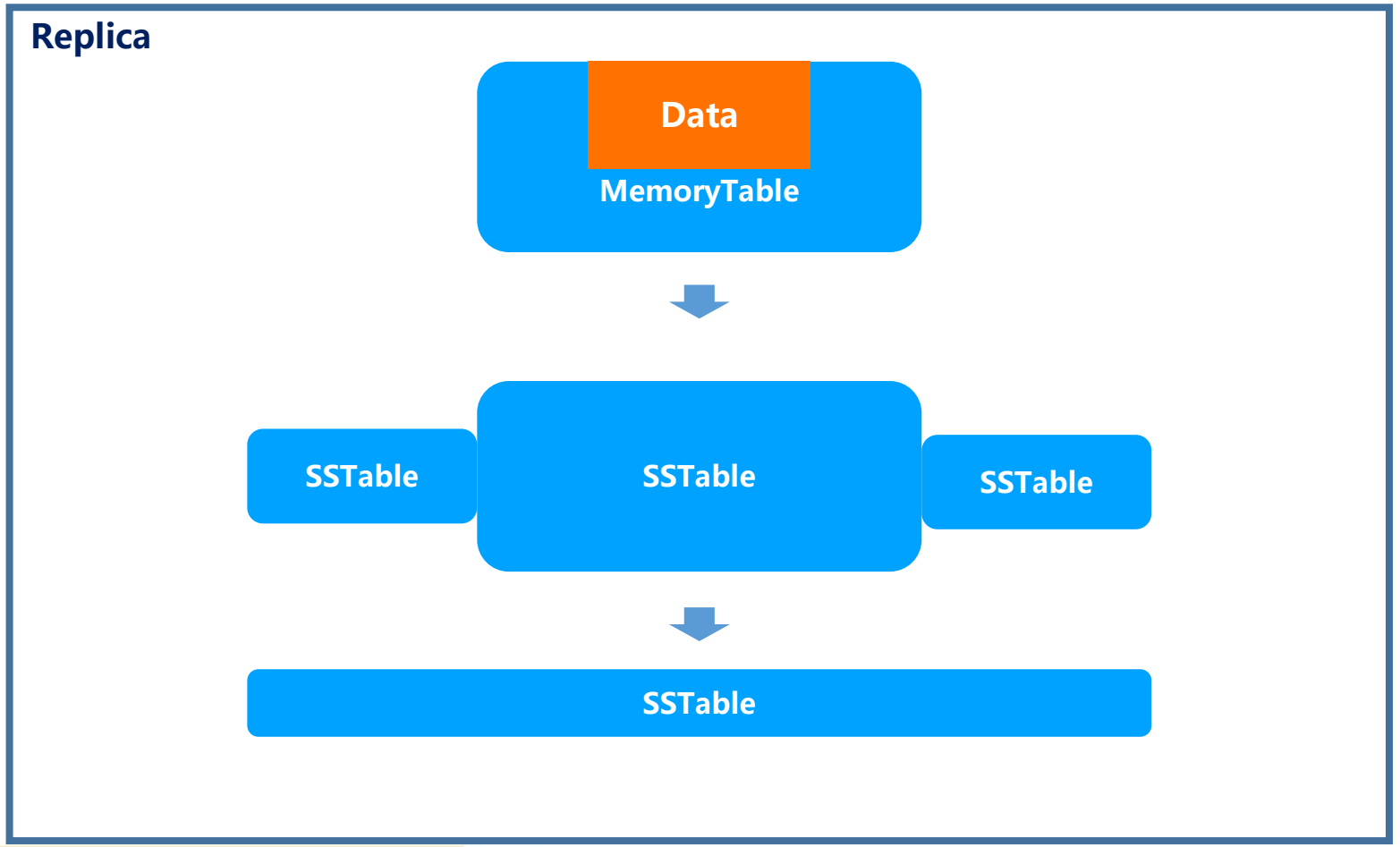
实现原理—强一致性

1. 使用PacificA协议
2. 全部副本同步复制
3. 两阶段提交

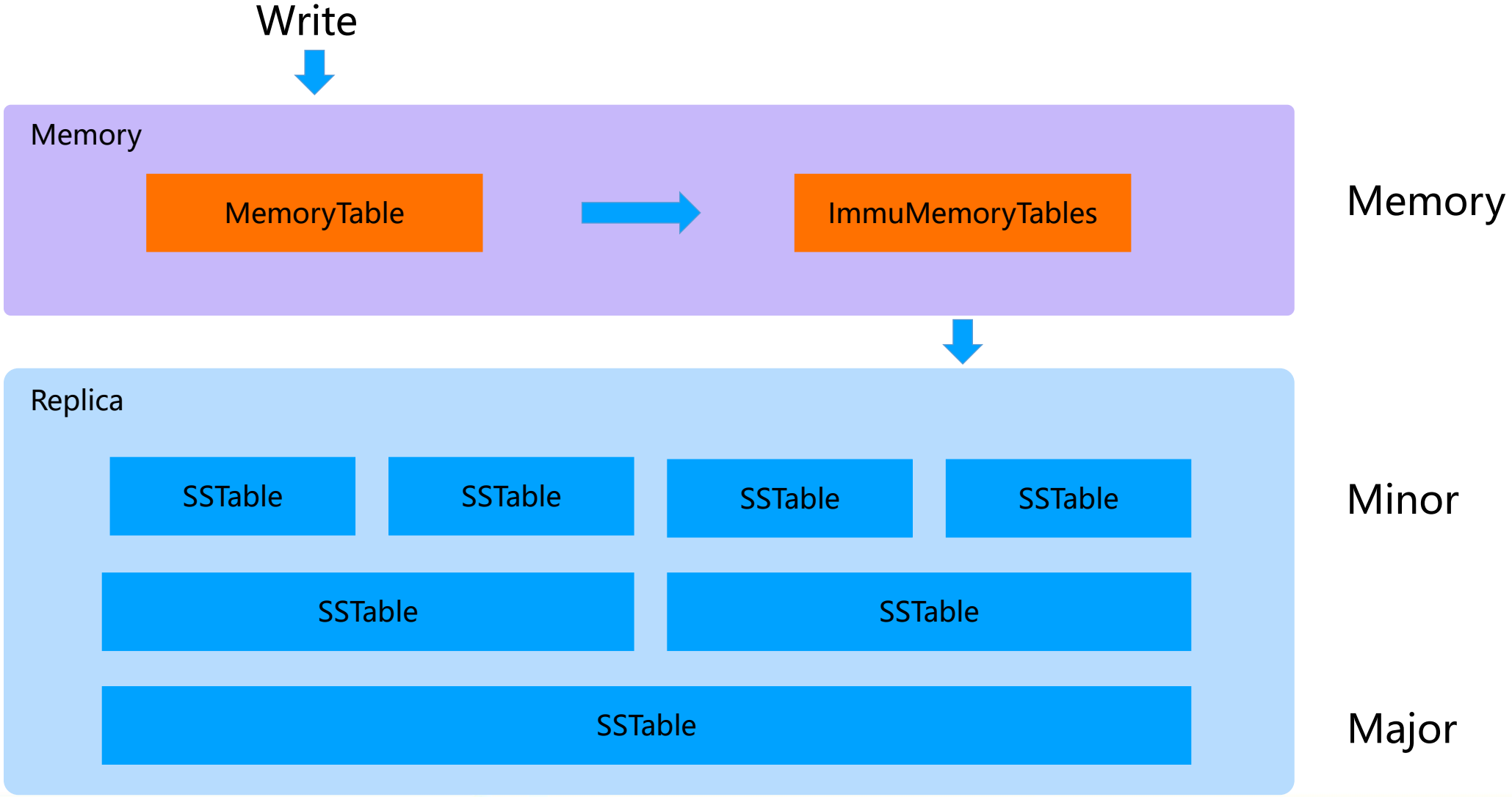


实现原理—底层数据结构

LSM-Tree



实现原理—Compaction流程



功能特性

功能特性—性能

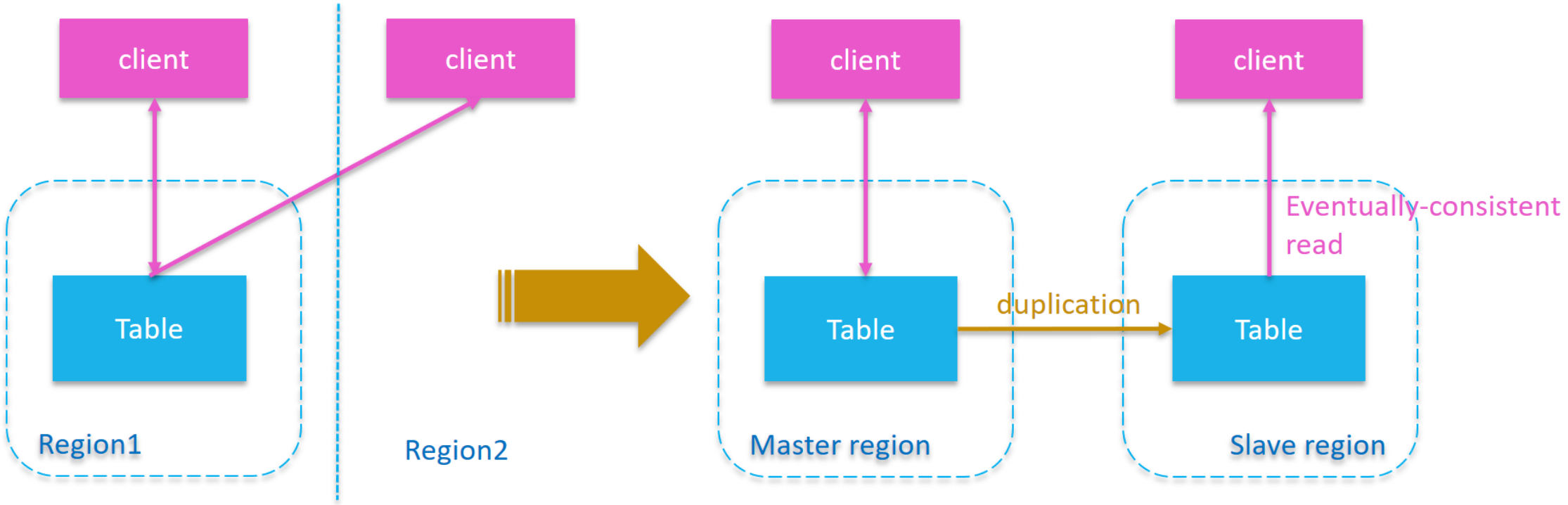
Version 2.4.0 Benchmark，5台节点，单条数据1KB

Read:Write	Client*Thread	---	QPS	AvgLatency	P99Latency(us)
0:1	3*15	read	---	---	---
		write	46128	972	5591
1:0	3*50	read	282648	542	1674
		write	---	---	---
1:1	3*30	read	36014	1068	15345
		write	36016	1421	8197
1:3	3*15	read	11622	779	10417
		write	34989	1021	5467

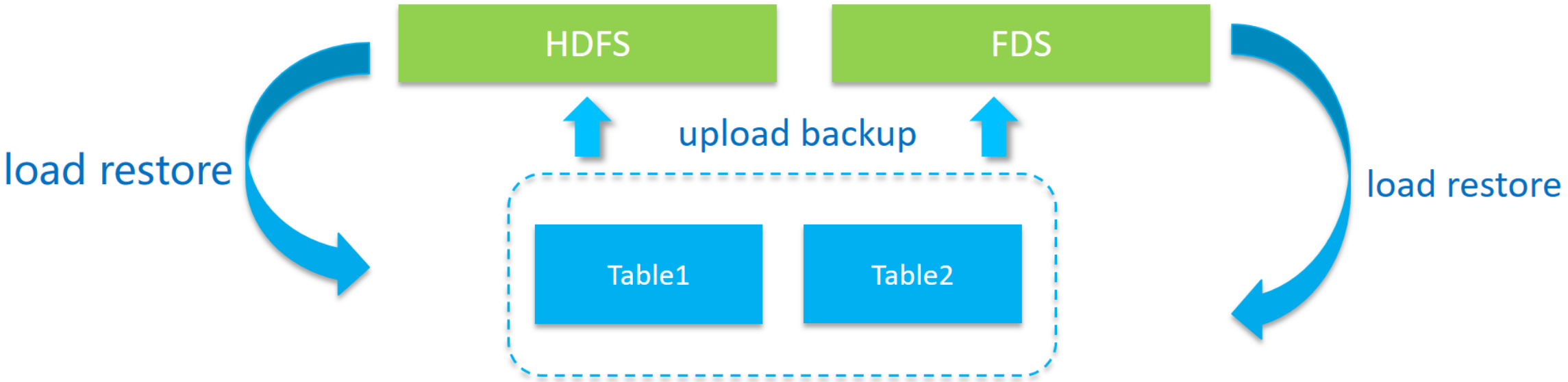
功能特性—读写接口

类型	接口	说明
基础操作	set/get/del	单条增删改查
	multiSet/multiGet/multiDel	同一HashKey下多条记录原子增删改查
	batchSet/batchGet/BatchDel	不同HashKey下多条记录非原子增删改查
扫描操作	hashScan	同一HashKey下范围扫描
	fullScan	全表扫描
CAS操作	checkAndSet	若记录符合请求条件，原子更新该记录
	checkAndMutate	若记录符合请求条件，原子执行批量操作
其他操作	incr	原子自增
	exist	是否存在某条记录

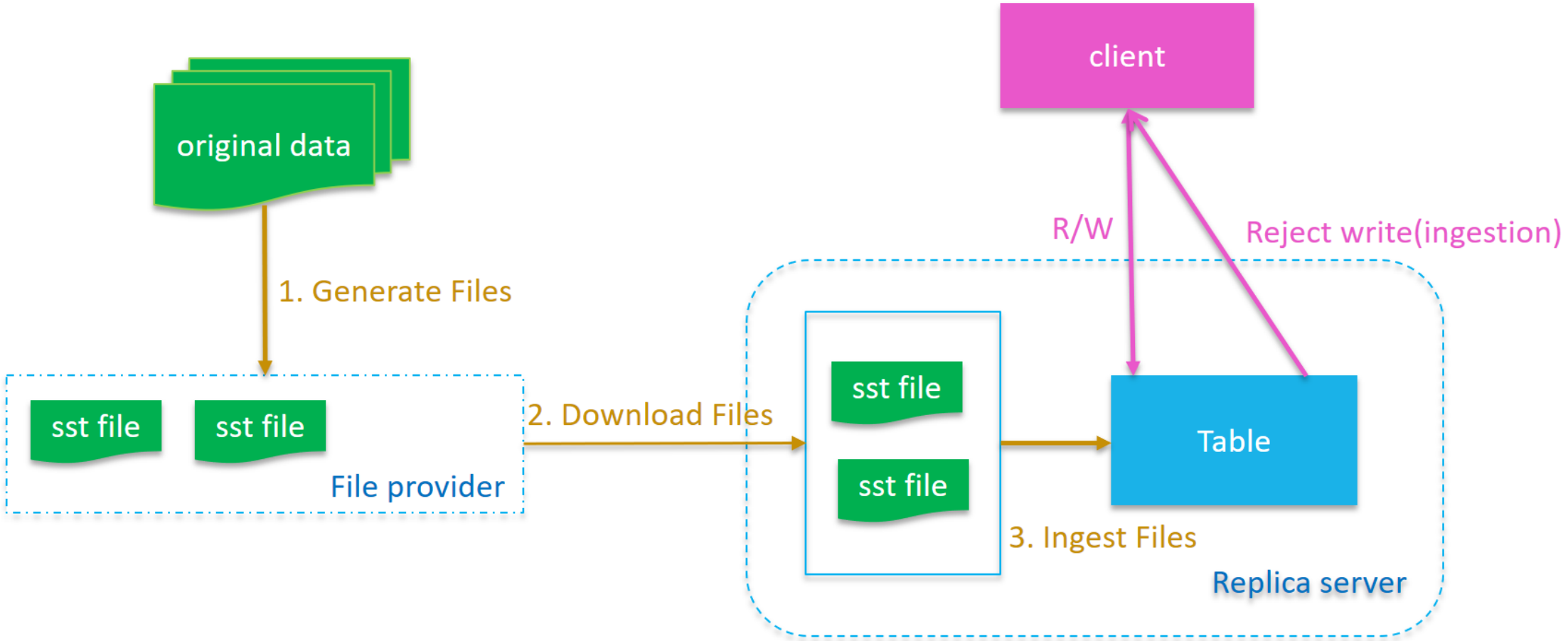
功能特性—数据热备


client
client
Table
Region1
Region2
client
client
Table
Table
duplication
Eventually-consistent
read
Master region
Slave region

功能特性—数据冷备


HDFS
FDS
upload backup
load restore
load restore
Table1
Table2

功能特性—BulkLoad



功能特性—离线生态

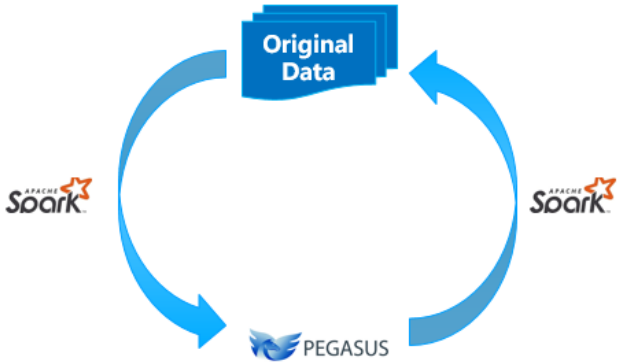
- 数据导出（在线 -> 离线）

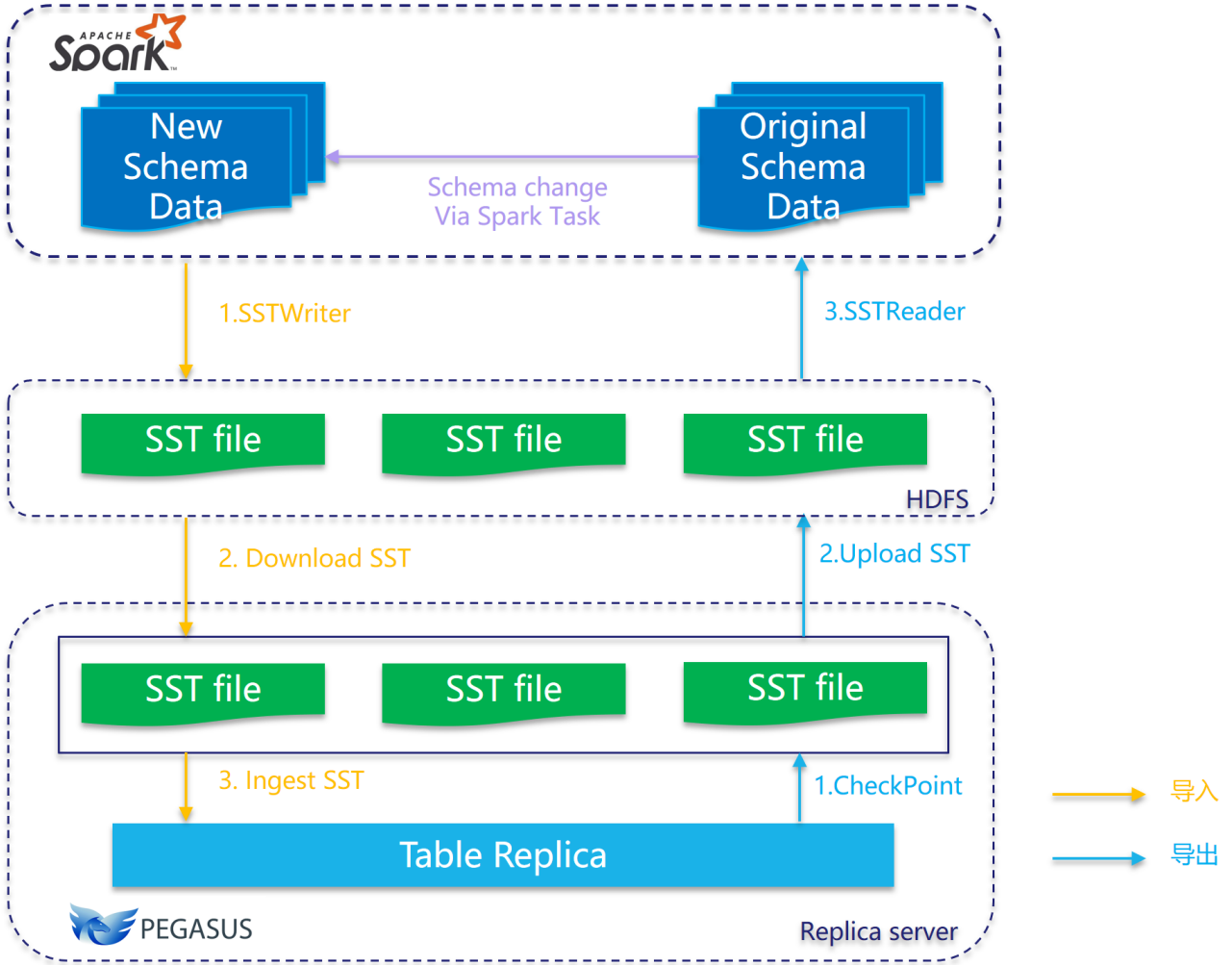
全量数据导出到HDFS，通过Spark进行离线计算，适用于在线数据离线分析场景

- 数据导入（离线 -> 在线）

离线生成数据文件，快速加载海量数据，

适用于周期性更新数据，在线提供读取服务，大幅提升灌库速度及稳定性



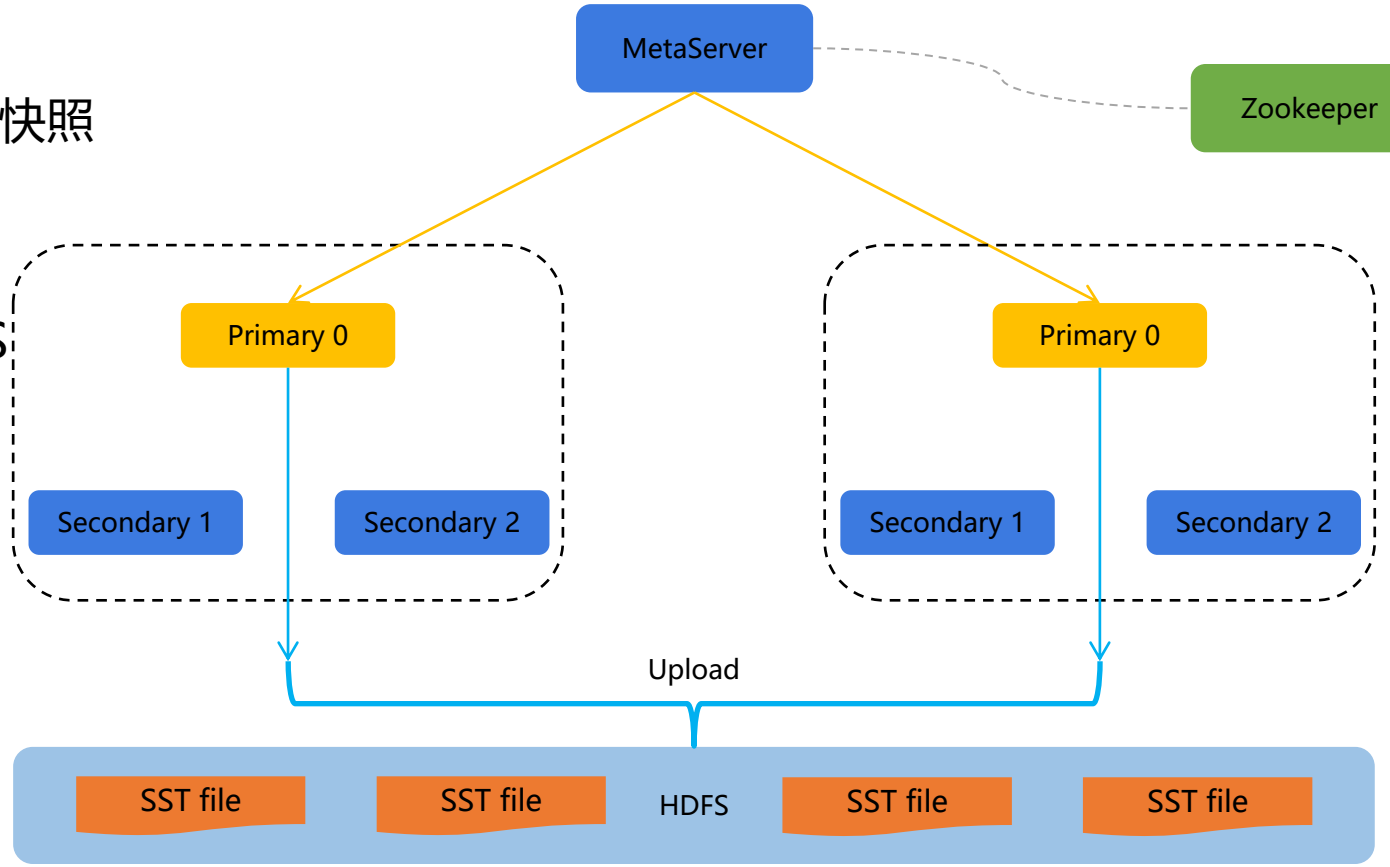


离线生态—数据导出实现

1 数据快照
- 通过RocksDB CheckPoint接口生成数据快照

2 数据上传
- 将数据快照从ReplicaServer上传到HDFS

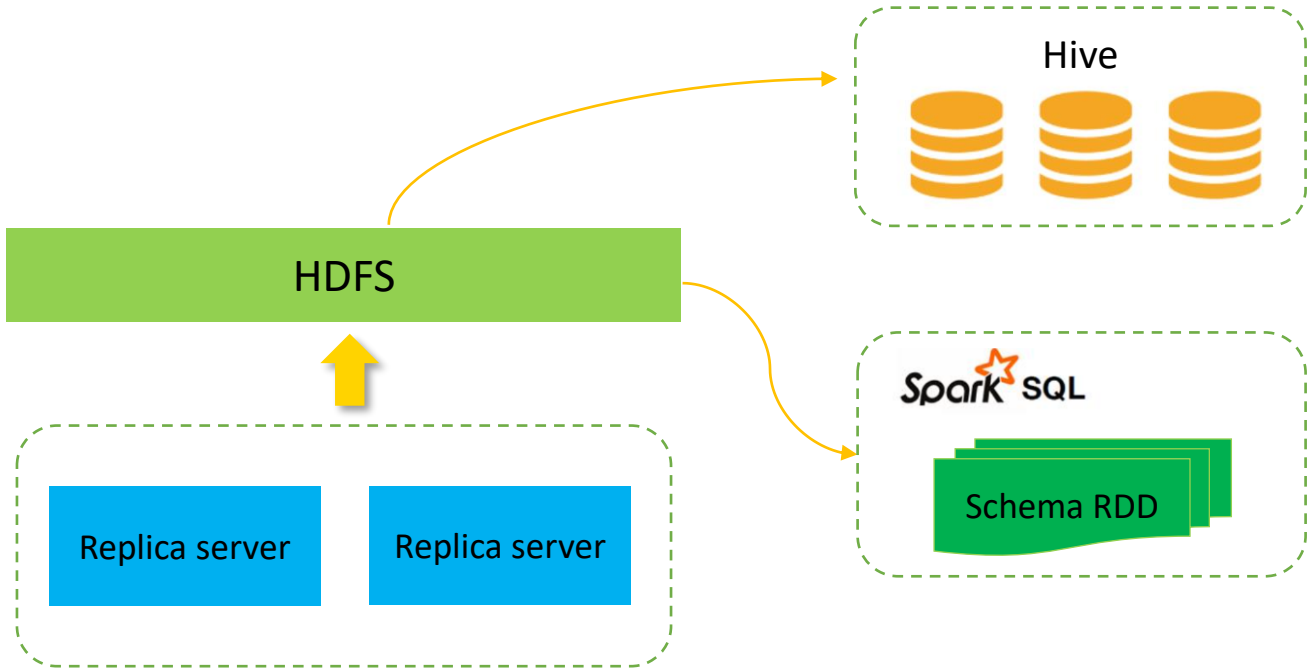
即Pegasus的冷备份流程



离线生态—数据导出实现

3 使用PegasusSpark解析数据快照，进行离线分析

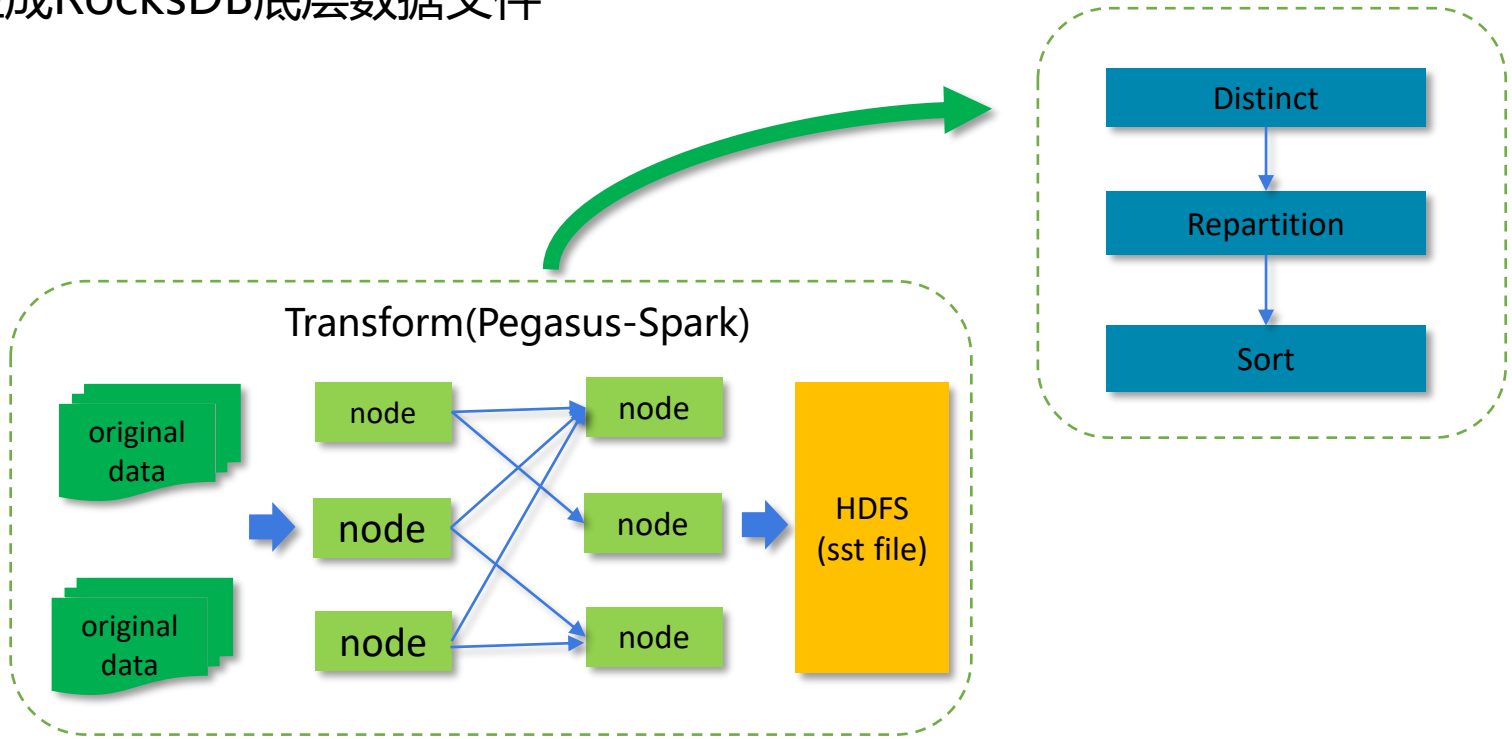
- 转换为Hive(parquet)
- 使用SparkSQL进行离线分析



离线生态—数据导入实现

1 数据生成

- 通过离线系统ETL生成RocksDB底层数据文件



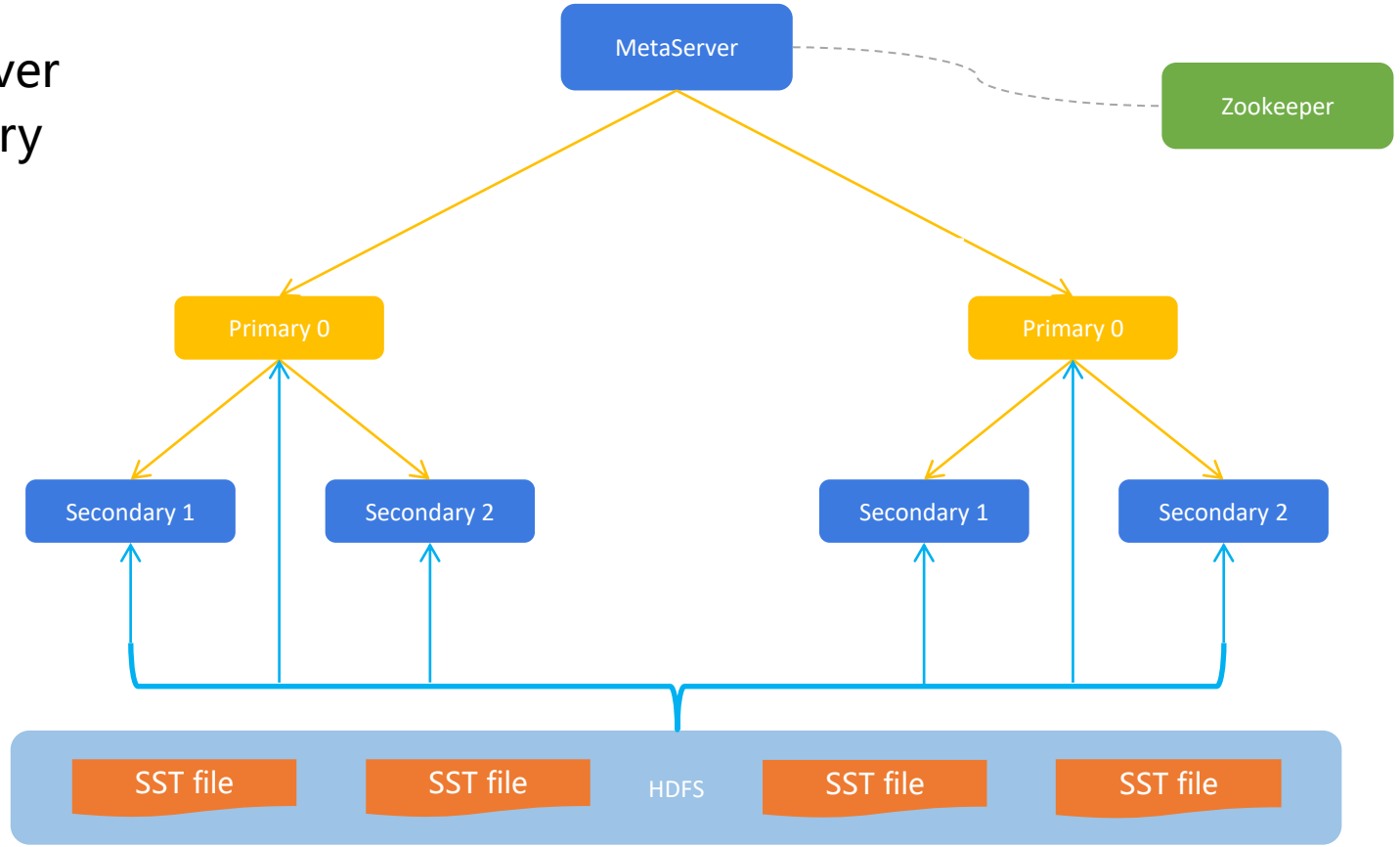
离线生态—数据导入实现

2 数据下载

- 将数据文件从HDFS下载到ReplicaServer
- 数据将同时下载到Primary和Secondary

3 导入引擎

- 待全部分片下载完成后进入导入环节
- 导入是特殊的写入，由Primary发起
- 通过IngestSST将数据文件加载到引擎
- 加载时引擎将瞬间阻写（分钟级别）



离线生态—数据导入实现

4 数据整理

- 数据导入后，必须进行全量数据Compaction，优化读取性能和空间占用
- ManualCompaction可以限制Compaction速度，减小线上读取抖动
- BulkLoad与Compaction解耦，支持多次BulkLoad之后，一次性Compaction

应用实践

应用实践—选型对比

	redis	PEGASUS	APACHE HBASE
性能	极低时延	低时延	高时延 长尾
数据规模	数据量小	数据量大	数据量极大
成本	成本高	成本低	成本低
一致性	非强一致	强一致	强一致
可用性	可用性高	可用性较高	可用性较低

Pegasus适用场景

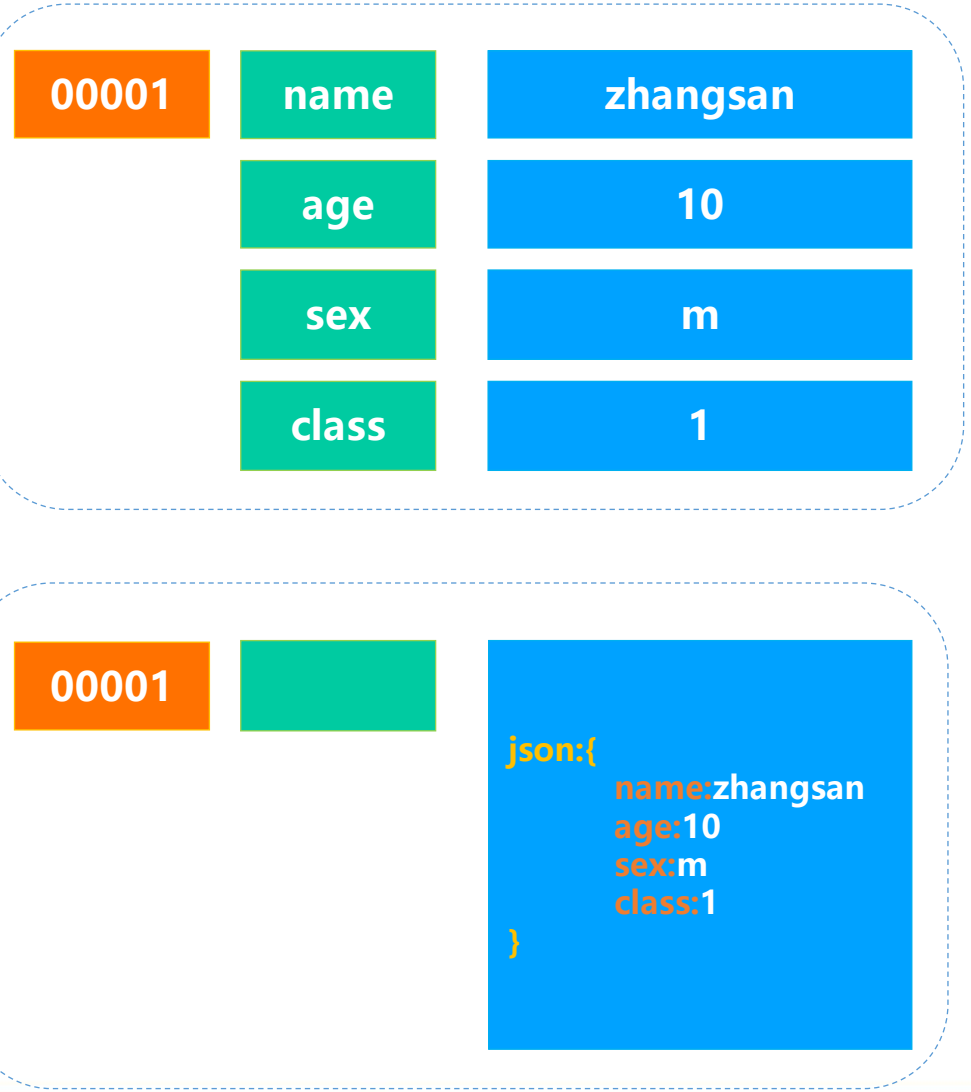
延迟敏感的在线业务、数据规模较大
可用性要求较高
数据强一致性及持久化需求

应用实践—使用场景

- 典型场景
  - 数据结构简单：单条记录长度短，Key-Value模型
  - 存储量大：100G到100T级别
  - 有持久化需求：数据高可靠，永久存储
  - 强一致、低延迟：能满足强一致性，延迟较低（P99<10ms）
- 不适用场景
  - 结构复杂：支持SQL查询的需求
  - 单条记录大：数据大小大于100KB
  - 数据量小：1GB 以下（可优先选择 Redis）

应用实践—表格存储

RowKey	TimeStamp	CF1		CF2	
		name	age	sex	class
00001	t1	zhangsan	10	m	1
	t2	lisi	20	m	
	t3	wangwu	30		2
00002	t1	zhaosi		m	3
	t2		10		1
	t3	yanger	20	f	4



HashKey尽量打撒，避免数据倾斜

应用实践—在广告业务落地

- 业务背景
  - 静态特征：年龄、性别、职业
  - 行为数据：APP的安装、卸载、使用、登录等
  - 在线喂给推荐引擎，预测打分，推荐App列表
  - 每天离线计算过去30天的用户行为，全量灌库
  - 3亿个HashKey，41个SortKey，100+特征，数据总量2.2TB

- 挑战
  - 线上读取流量高，需要控制写入速度，避免灌库对读取造成影响
  - 灌库时间尽可能短，最长不能超过1天
  - 数据写入后引擎Compaction挤占CPU、磁盘IO，极易造成抖动

应用实践—在广告业务落地

- 方案一：通过RocksDB调参实现灌库模式，提升灌数据场景的QPS和吞吐
  - 禁止自动Compactions
  - 允许Level0 SST数量无穷大
  - 取消Compactions过慢导致的写入限速及停写
  - 提高write_buffer大小及个数

- 实践过程
  - 切为灌库模式 -> 通过API写入数据 -> 写入完成->
    Manual Compact -> 切回Normal模式

应用实践—在广告业务落地

- 通过RocksDB调参，提升灌数据场景的QPS和吞吐

参数	值
disable_auto_compactions	true
level0_file_num_compaction_trigger level0_slowdown_writes_trigger level0_stop_writes_trigger	∞
soft_pending_compaction_bytes_limit hard_pending_compaction_bytes_limit	no limit
max_compaction_bytes	∞
write_buffer_size	raise to 256M form 64M
max_write_buffer_number	raise to 6 form 4

应用实践—在广告业务落地

- 方案二：通过离线计算+BulkLoad，实现海量数据灌库
  - 业务侧离线计算结束，以结算结果为数据源，对接PegasusSpark，生成数据文件
  - 每天流量低峰使用BulkLoad灌库，完成后进行Compaction

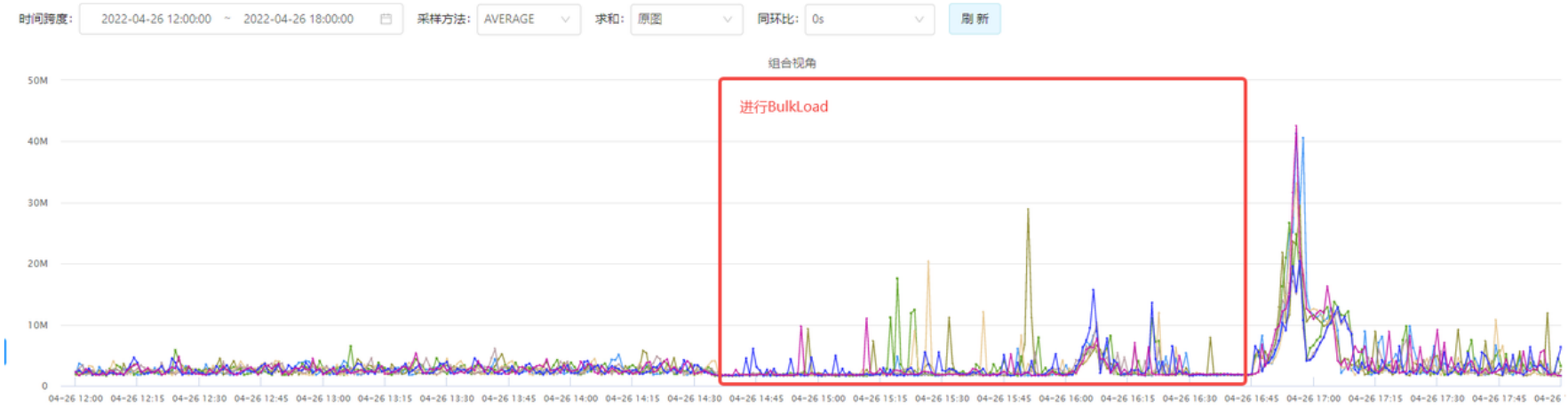
- 实践过程
  - 离线生成底层数据文件
  - 灌库
    - 增量：切为灌库模式 -> 下载数据文件到各节点 -> IngestSST -> Manual Compact -> 切回Normal模式
    - 全量：AB集群方式

      A提供在线读取 -> 全量灌入B -> 切流到B

      B提供在线读取 -> 全量灌入A -> 切流到A

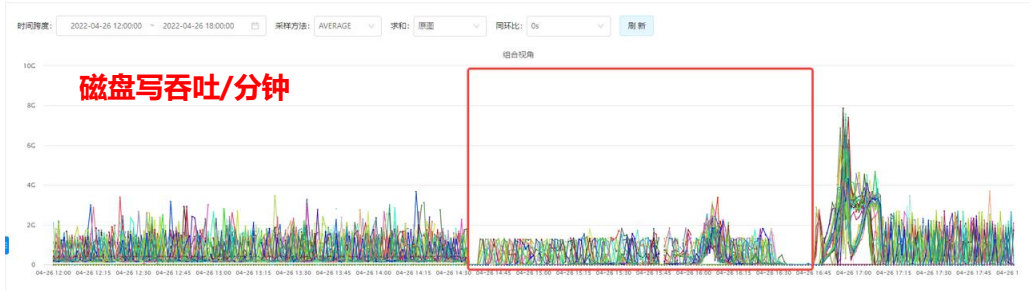
应用实践—在广告业务落地

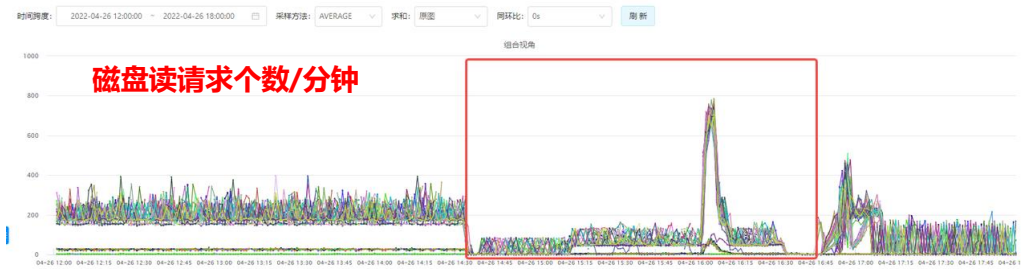
- 收益
  - 2.2T数据灌库时间从12小时缩短为1小时（20台节点，100M限速）
  - 灌库过程中multi_get请求P99时延在30ms之内

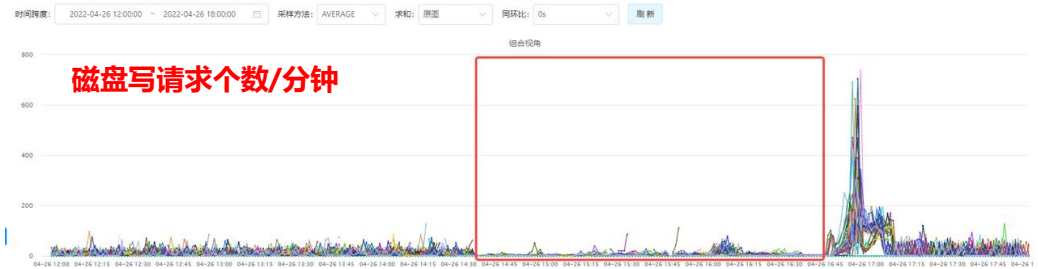


应用实践—在广告业务落地

- 收益
  - 相比于随机写入，磁盘IOPS、吞吐明显下降







未来规划

- 功能
  - 成本优化：更加丰富的负载均衡策略
  - SLA提升：多可用区部署能力
  - 性能提升：Client请求聚合，非强一致性支持
  - 架构优化：公有云对象存储（S3）支持，在线无损迁表，去除Zookeeper依赖
  - 新能力：探索PMEM给Pegasus带来的变化，引入RocksDB更多特性
- 开源
  - 推动Apache Pegasus合规建设，加速毕业进程
  - 社区活跃度提升


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