DTCC 2010-2022

第十三届中国数据库技术大会

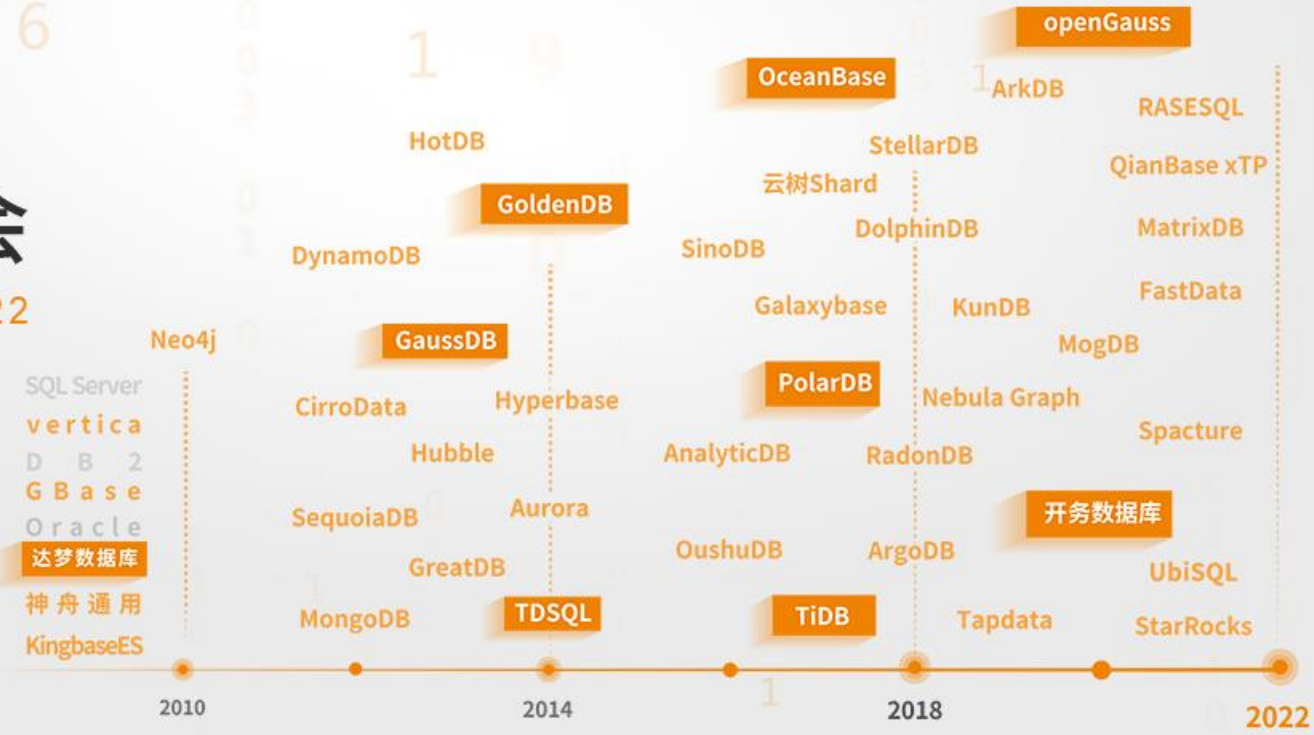
DATABASE TECHNOLOGY CONFERENCE CHINA 2022

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



百度智能云高性能KV数据库设计与实践

刘东辉

百度资深研发工程师

关于我

- 2013年毕业于南开大学，毕业后加入微博基础架构组，先后负责微博Redis、CounterService、CacheService、Redrocks等基础组件的设计与开发工作。
- 2020年加入百度基础架构部，担任Redis方向内核技术负责人，主要负责Redis和KV数据库PegaDB的设计与开发工作
- Apache Kvrocks(incubating) PMC Member

目录

- 百度智能云高性能KV数据库概述
- 百度智能云高性能KV数据库设计与实践
- 开源社区协作
- 未来规划

目录

百度智能云高性能KV数据库简介

PegaDB：完全兼容Redis协议大容量、低成本、高性能分布式KV数据库

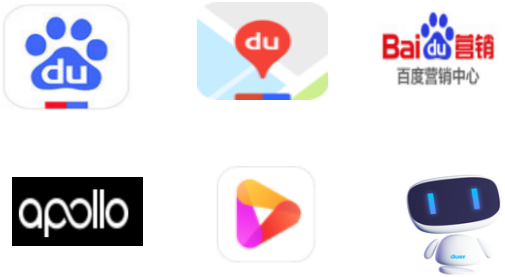
特点

- 全面兼容Redis, 支持业务平滑迁移
- 支持水平扩展，单集群PB级存储
- 基于SSD构建，单GB成本相比Redis降低80%
- 支持毫秒级在线数据处理
- 支持异地多活架构，提供多地域容灾能力
- 支持可调一致性、冷热分离、Json数据模型等增强特性

PegaDB应用场景

典型应用场景

- 大数据量场景，Redis存储成本高
- 开源KV数据库，无法完全满足需求
- 典型冷热分离场景，传统Cache+DB架构，业务开发复杂度高



目录

PegaDB设计与实践 | 背景

	业务痛点	需求
存储成本	Redis内存存储，开启持久化需要预留内存，存储成本高	低成本
容量	Redis容量有限（4TB），无法支撑大数据量存储	大容量
迁移成本	集团其它KV数据库存在兼容性、通用性、易用性问题	兼容Redis、通用KV存储
		高性能、可扩展、高可用

PegaDB设计与实践 | 业界方案

Ssdb Pika Ardb Kvrocks	Meitu Titan Tedis	Redis On Flash Redrocks
Disk Based 基于单机KV存储引擎	Disk Based 基于分布式KV引擎	Mem + Disk Redis + 单机KV存储引擎
扩展性问题 性能问题 不支持多活架构	兼容性问题 性能问题 不支持多活架构	通用性问题 性能问题 不支持多活架构

PegaDB设计与实践 | 设计选型

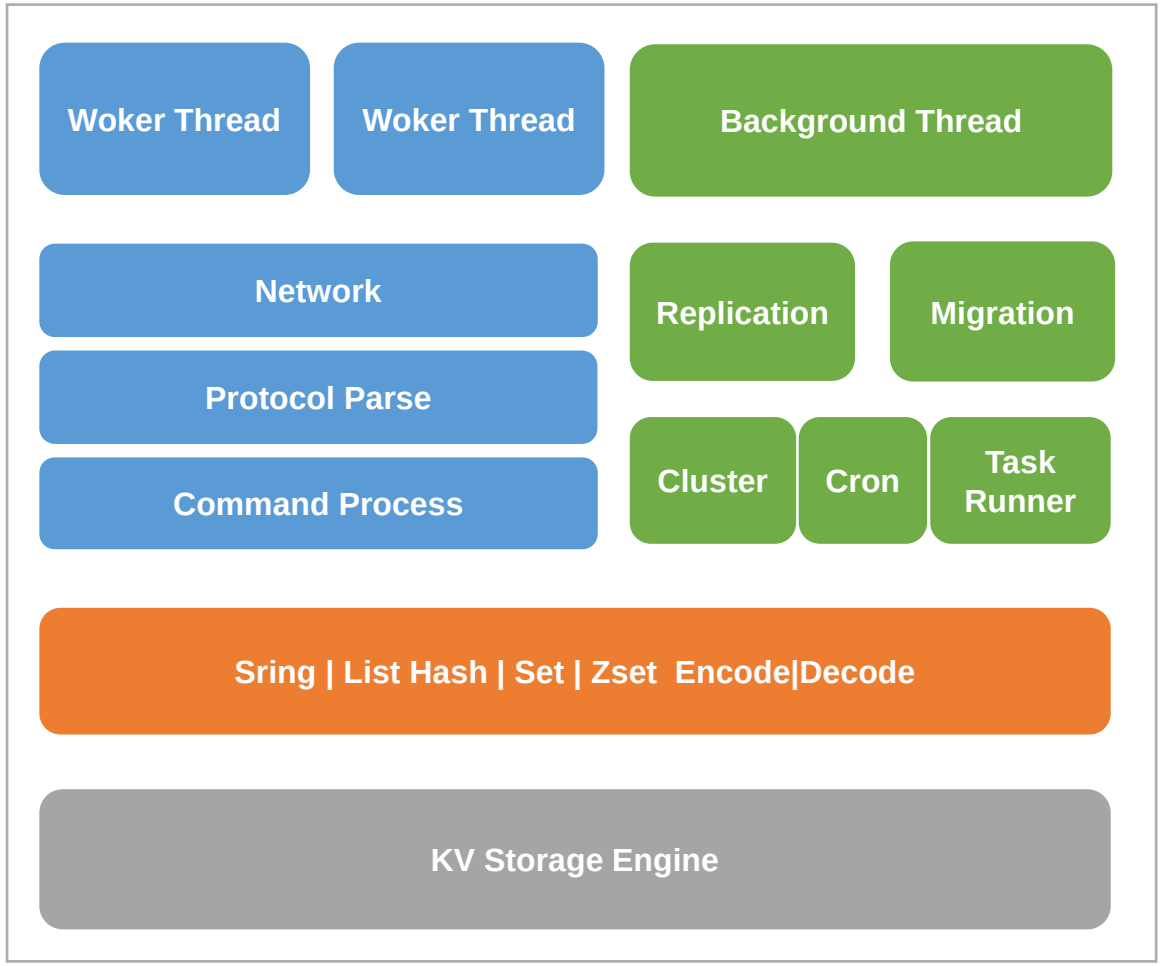
二次开发还是从0开始？
二次开发项目选型(**Ardb、Pika、Kvrocks ……**)？

选型考量

- 人力、交付时间
- 代码简洁，方便二次开发
- 设计思路及发展规划相似

Kvrocks二次开发
深度参与开源社区建设

PegaDB设计与实践 | Kvrocks介绍



Kvrocks 是美图公司开发的一款分布式 KV 数据库，并于2019年正式开源。使用 RocksDB 作为底层存储引擎并兼容 Redis 协议，旨在解决 Redis 内存成本高以及容量有限的问题

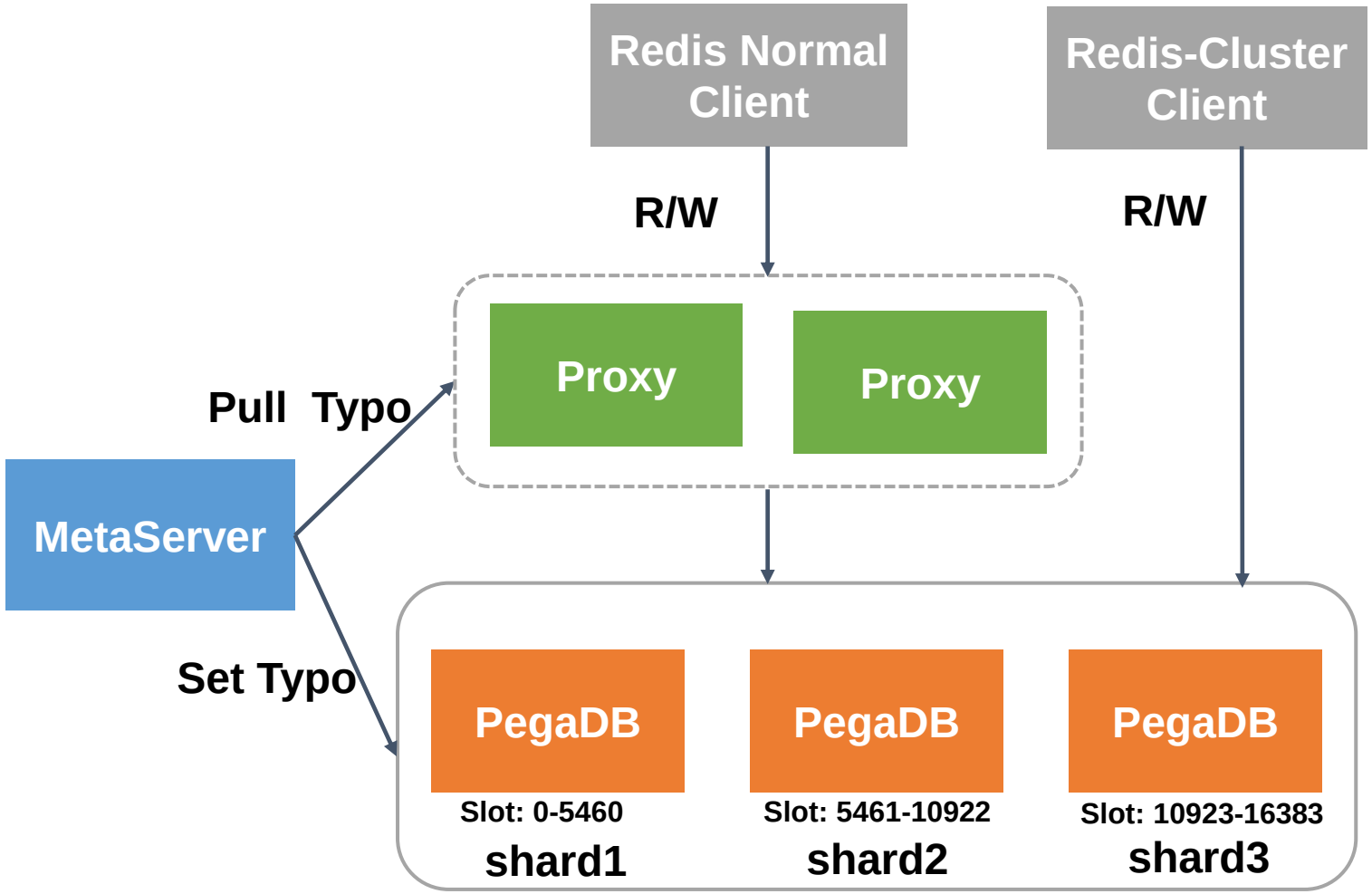
设计实现

- 基于RocksDB存储引擎封装Redis数据类型
- Hash等复杂数据类型拆分为多条KV数据
- 多Worker线程的处理模型
- 支持主从复制，增量复制基于WAL的“物理复制”
- Compaction Filter 实现过期数据删除
- 通过 Version 实现大 Key 秒删

PegaDB设计与实践 | Kvrocks不足

- 扩展性
  - 不支持集群
- 性能
  - 大Value场景、冷热明显场景存在性能问题
- 数据一致性
  - 异步复制模型，无法满足较高一致性需求
- 高可用
  - 不支持多活架构，无法满足地域级容灾需求
- 功能
  - 不支持Redis4.0以上版本命令、事务、Lua、多DB特性

PegaDB设计与实践｜集群方案



数据分布策略

- 同Redis-Cluster，预分固定数量Slot

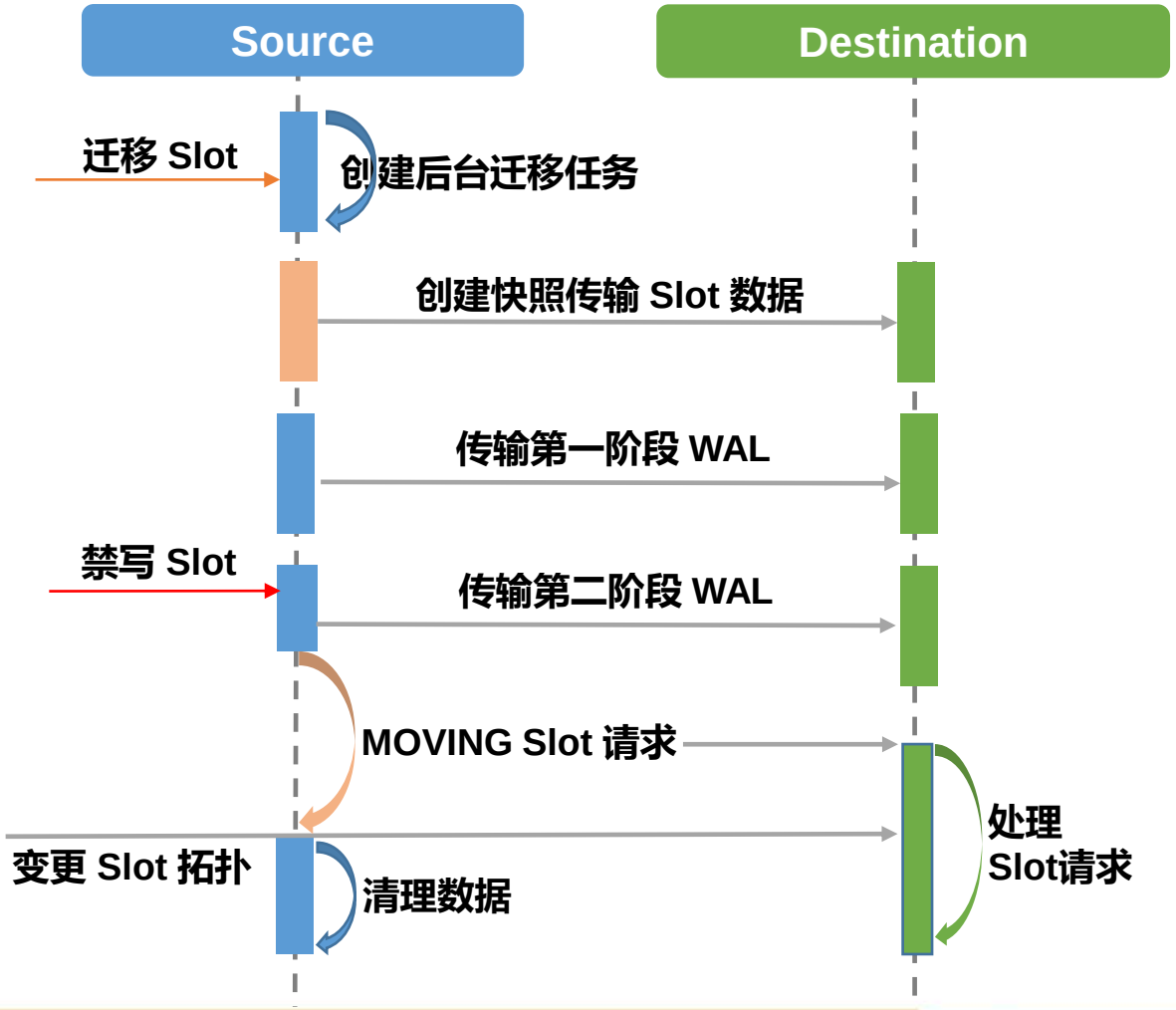
中心化架构

- MetaServer管理集群元信息

集群架构不强依赖代理层

- MetaServer向PegaDB下发拓扑
- 完全兼容redis-cluster SDK

PegaDB设计与实践｜扩缩容设计



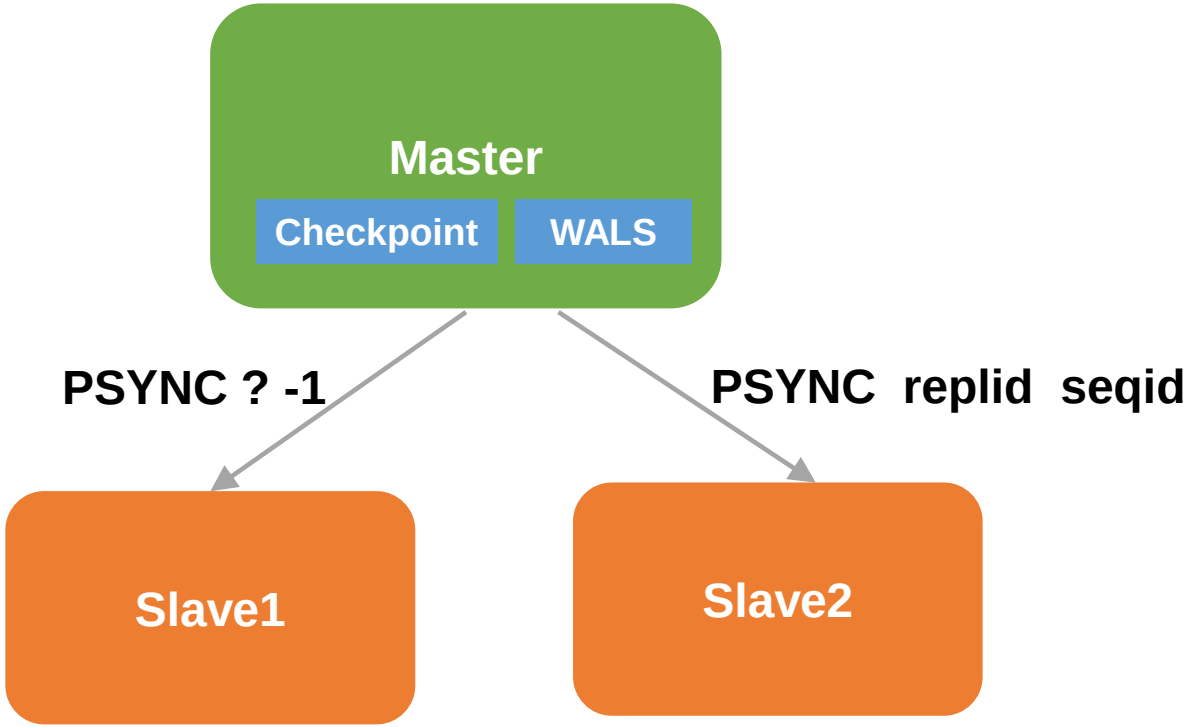
数据迁移

- 发送Slot数据快照、发送增量命令
  - 全量数据：RocksDB Snapshot
  - 增量数据：WAL日志
- Key编码中增加Slotid，提升迭代效率；
- 独立迁移线程，不影响正常请求；Slot并发迁移
- 清理源端数据：RocksDB Delete Range

拓扑变更

- 拓扑变更期间，短时间禁写，通常毫秒级

PegaDB设计与实践｜主从复制优化



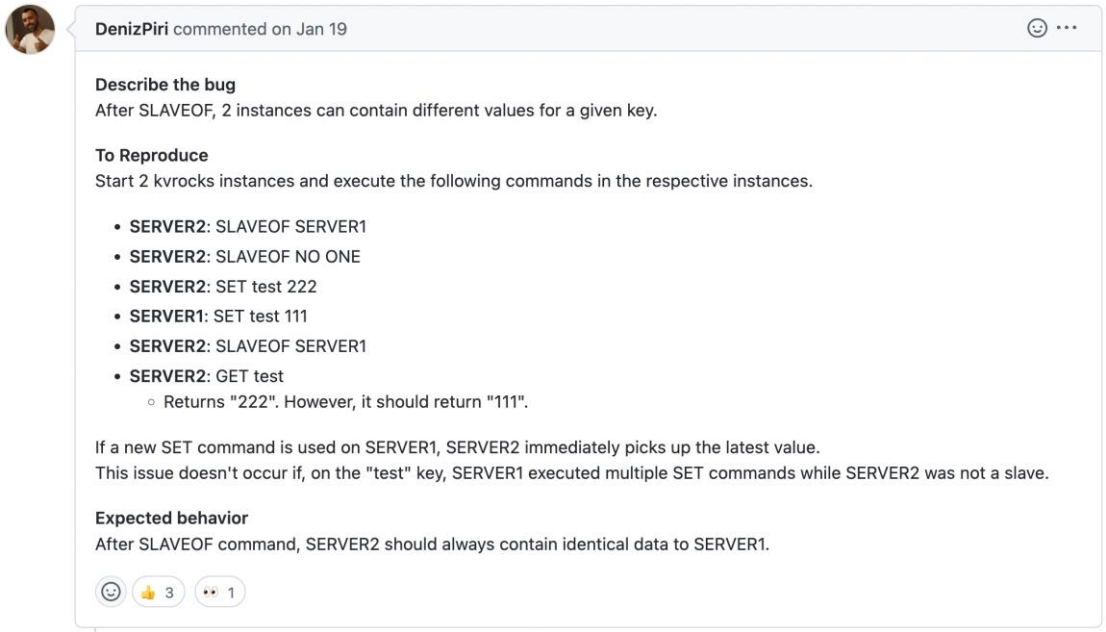
Kvrocks主从复制实现

全量复制：基于Checkpoint数据快照
增量复制：基于引擎层WAL的“物理复制”，基于WAL seq_id实现断点续传

问题

- 无同源增量复制保证主从切换会带来数据不一致
- 异步复制模型，主从切换可能会导致数据丢失

PegaDB设计与实践 | 主从复制优化



DenizPiri commented on Jan 19

Describe the bug
After SLAVEOF, 2 instances can contain different values for a given key.

To Reproduce
Start 2 kvrocks instances and execute the following commands in the respective instances.

- **SERVER2**: SLAVEOF SERVER1
- **SERVER2**: SLAVEOF NO ONE
- **SERVER2**: SET test 222
- **SERVER1**: SET test 111
- **SERVER2**: SLAVEOF SERVER1
- **SERVER2**: GET test
  - Returns "222". However, it should return "111".

If a new SET command is used on SERVER1, SERVER2 immediately picks up the latest value.
This issue doesn't occur if, on the "test" key, SERVER1 executed multiple SET commands while SERVER2 was not a slave.

Expected behavior
After SLAVEOF command, SERVER2 should always contain identical data to SERVER1.

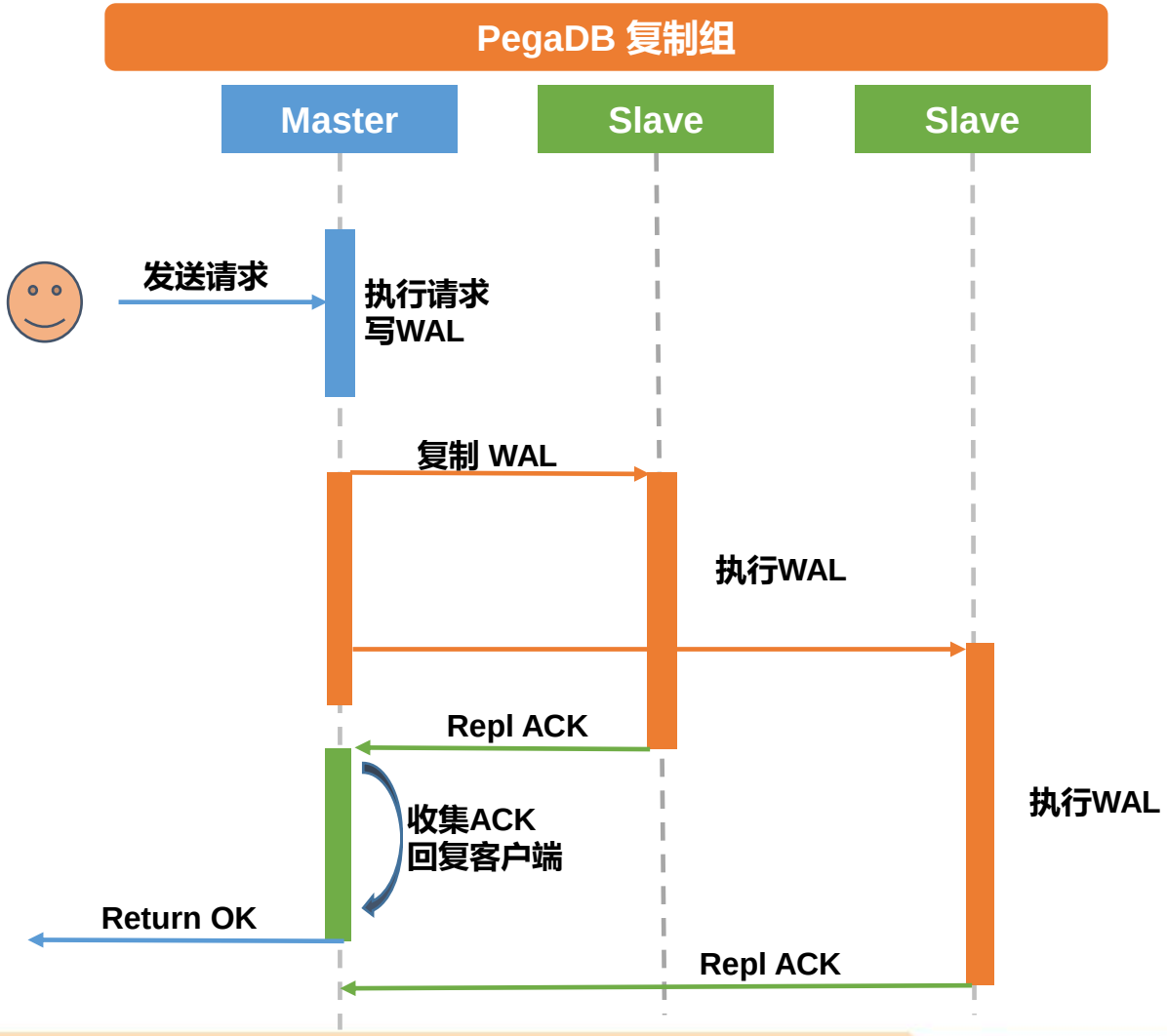
3 1

Replication Sequence ID PSYNC

Sequence: 4	K1 V1	Replication ID: ABC
Sequence: 5	K2 V2 | K3 V3	Replication ID: ABC
Sequence: 7	K4 V4	Replication ID: BCD

- 实例成为主库时生成新的 Replication ID（复制历史的标识）
- 每条写入 RocksDB 的操作都包含一个单调递增的 Sequence ID 和 Replication ID
- 支持 failover 后部分重同步、重启后部分重同步

PegaDB设计与实践｜主从复制优化



半同步复制

- 更强的一致性，支持配置同步的从库个数
- 支持超时机制

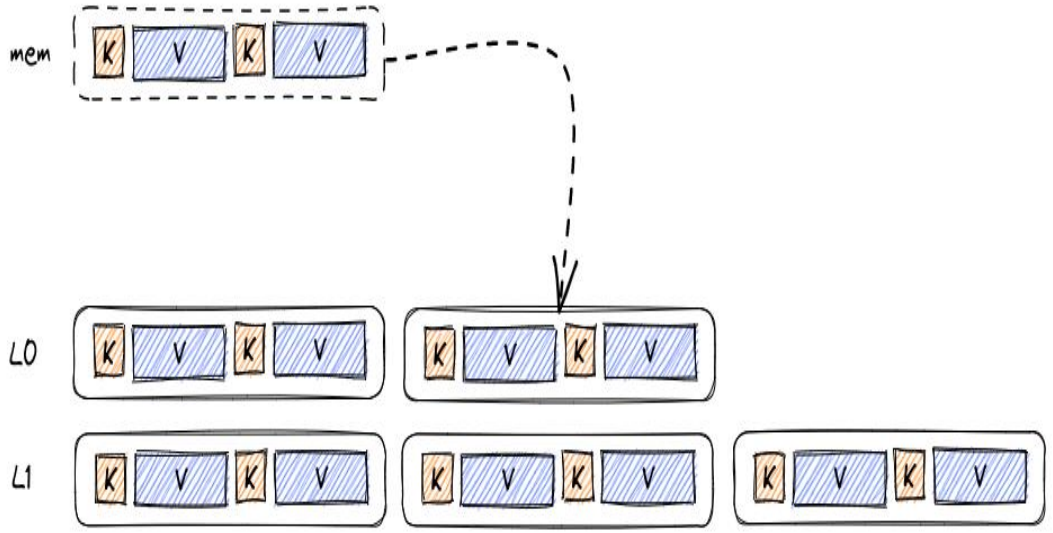
代理层支持配置请求粒度读取一致性

PegaDB设计与实践 | 性能优化

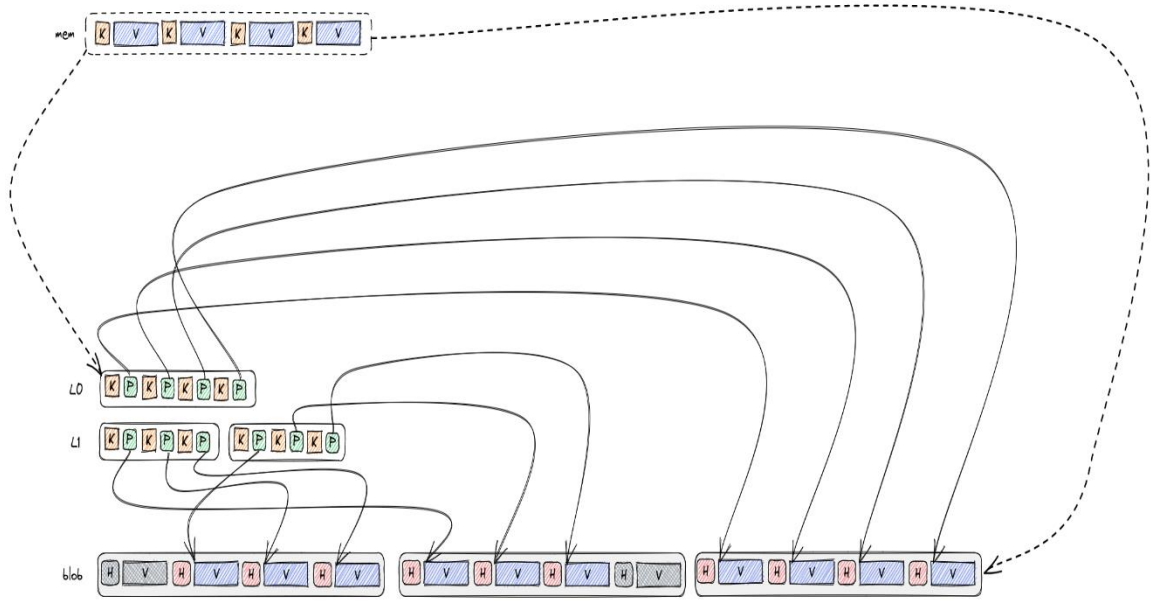
LSM引擎写放大问题

- WiscKey:　Key Value分离
- PebblesDB: 弱化全局有序的约束

- WiscKey、PebblesDB -> WiscKey
- Badger、TitanDB、BlobDB
  - TitanDB（二次开发支持Checkpoint: #207）
  - BlobDB



Traditional LSM tree. SST files contain keys (K) and values (V).



LSM tree with key-value separation. SST files (white background) contain keys (K) and blob pointers (P). Blob files (gray background) contain blob headers (H) and the actual values (V). Some blobs in the blob files may be unreferenced garbage.

PegaDB设计与实践 | 性能优化

存储引擎调优

耗时抖动优化

- 利用Rate Limiter对Compaction限速
- Partition index/filter
- 部分Compaction

读取优化

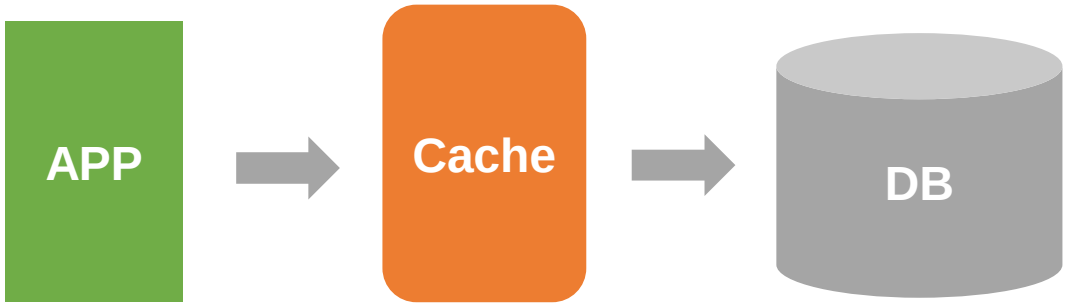
- Memtable开启全局Filter
- Data Block开启Hash索引
- L0和L1不压缩
- 自定义Prefix Extractor
- 支持配置多CF共享和独享Block Cache

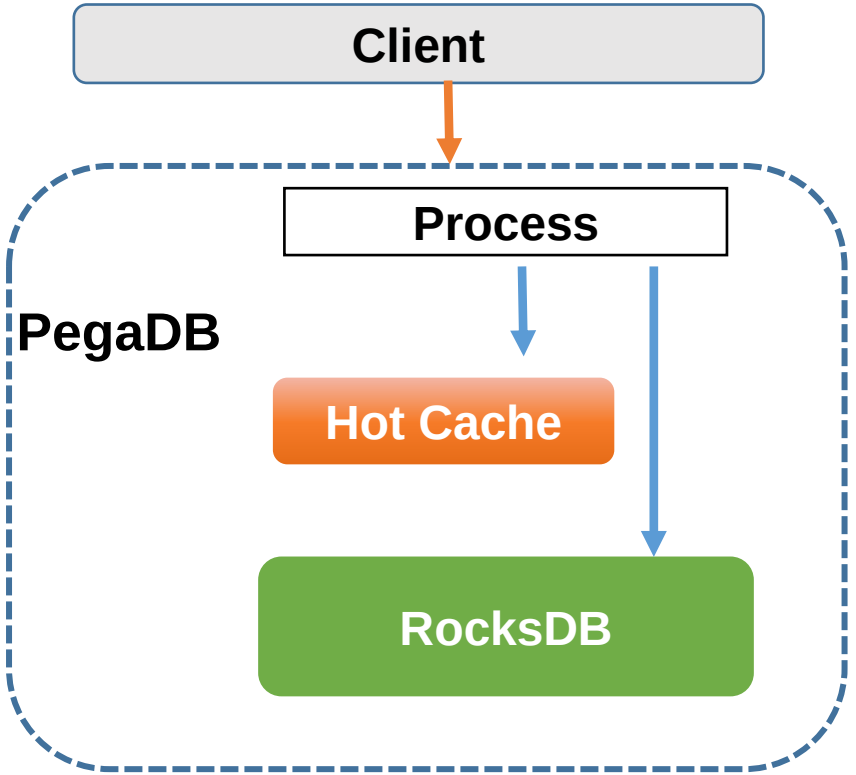
写入优化

- Key-Value分离，开启 GC 预读
- enable_pipelined_write
- sync_file_range，让刷盘更平稳

PegaDB设计与实践｜性能优化

冷热数据区分明显场景通常采用传统Cache（Redis） + DB(MySQL)架构



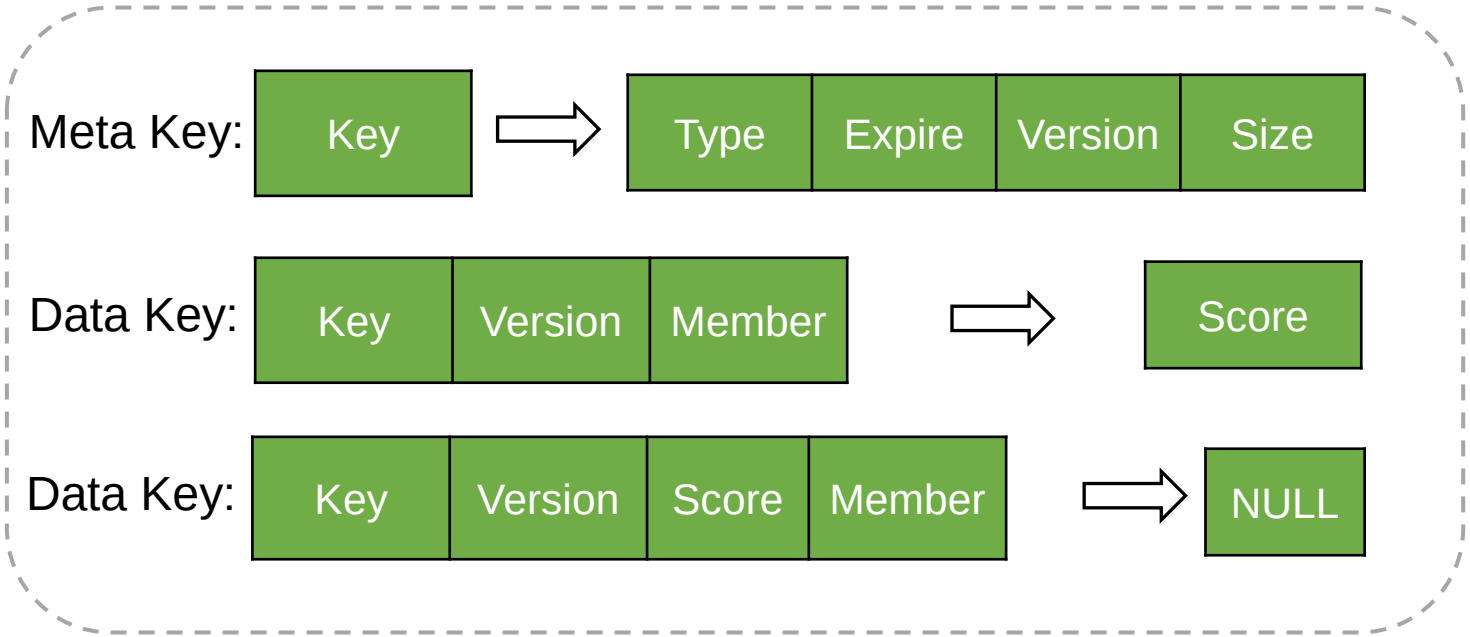


问题：传统Cache + DB，业务开发复杂度高

- **相比Block Cache, 更细粒度Cache，缓存利用率高**
- **相比Row Cache，没有Compaction导致的快速失效的问题**
- **热Key缓存，单节点支持百万级热Key访问**

PegaDB设计与实践｜性能优化

分散编码



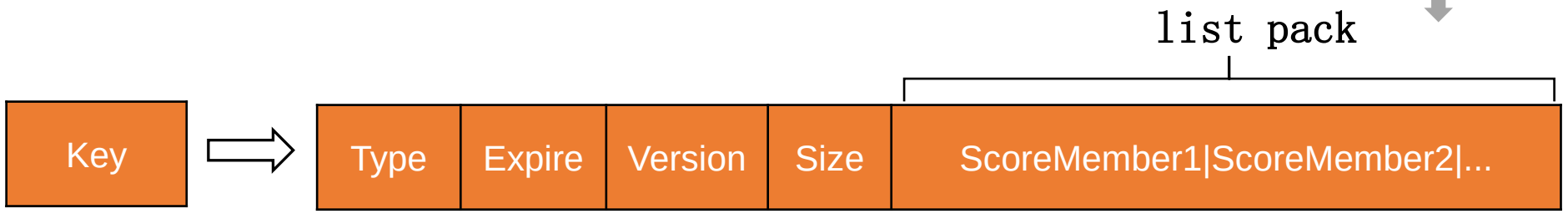
问题

- 批量、范围操作涉及多次磁盘IO性能差

优化

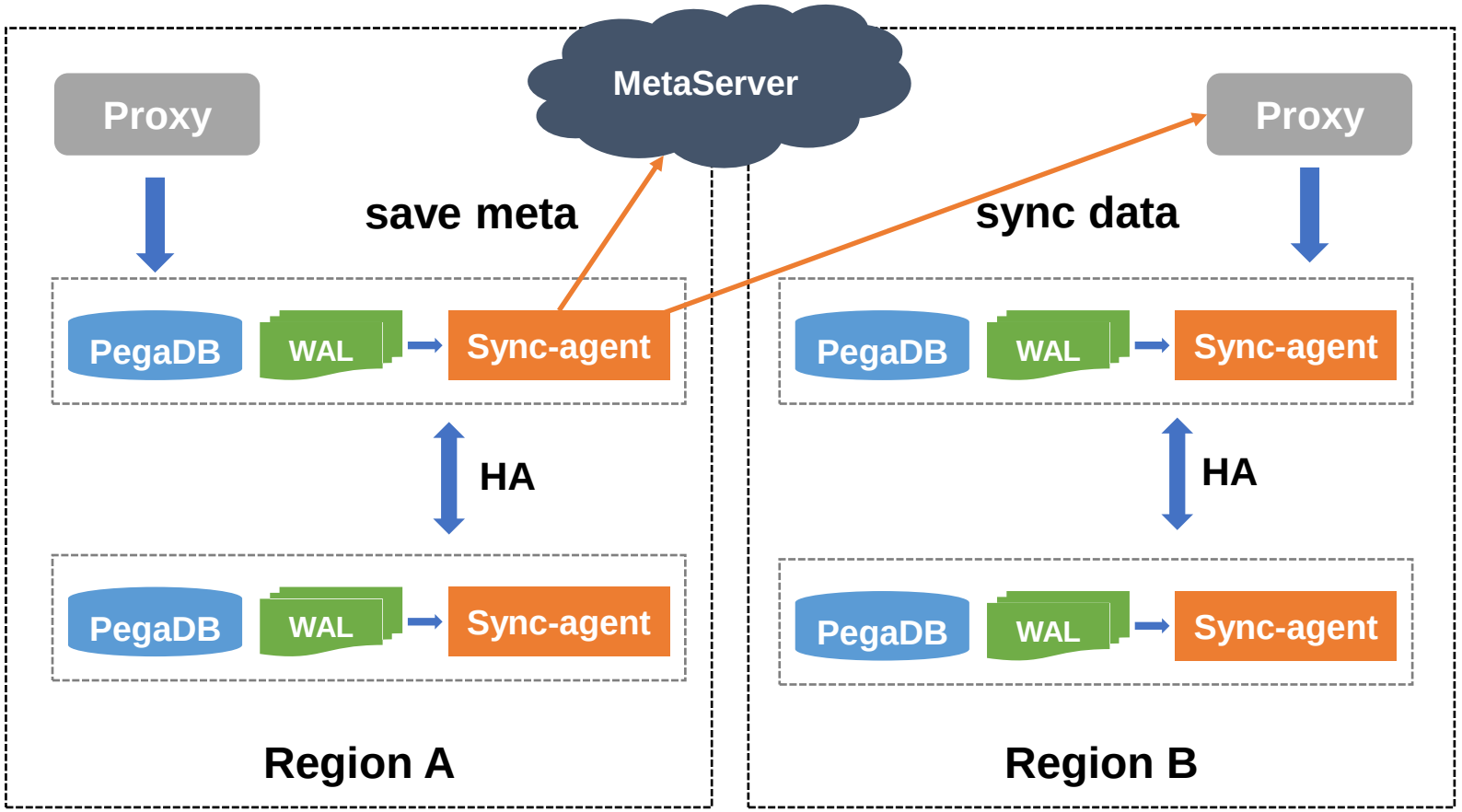
- 前缀迭代器
- 紧凑型编码

紧凑编码



PegaDB设计与实践｜异地多活架构

需求: 多地域快速容灾、降低跨地域写延迟



WRITE_CMD

OpHeader:
Version
ShardId
Opid： Wal SeqId | Cnt
Timestamp

设计实现

- SyncAgent同步组件，只在主库上工作
- 循环复制：Shardid
- 断点续传：Opid
- 多写冲突：LWW

PegaDB设计与实践 | Json数据模型


```
{
    "firstName": "John",
    "lastName": "Smith",
    "sex": "male",
    "age": 25,
    "address":
    {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    },
    "phoneNumber":
    [
        {
          "type": "home",
          "number": "212 555-1234"
        },
        {
          "type": "fax",
          "number": "646 555-4567"
        }
    ]
}
```

STRING/HASH数据类型存储问题

- 需要业务对数据进行序列化/反序列操作，增加开发复杂度
- 读取、更新部分字段，存在读写放大问题
- 并发更新字段时存在数据一致性问题

RedisJSON

RedisJSON - a JSON data type for Redis

GitHub ★ 3314

License: Other

Homepage

PegaDB设计与实践 | Json数据模型

- 完全兼容RedisJSON协议
- 支持JSONPath语法查询和更新文档中的元素
- 支持原子操作所有JSON Value类型
- 紧凑型编码存储
- 支持热Key Cache

操作	RedisJSON	SCS For Redis
增	JSON.SET <key> <path> <json> [NX | XX]	✓
删	JSON.DEL <key> [path] JSON.FORGET <key>	✓
改	JSON.SET <key> <path> <json> [NX | XX] JSON.NUMMULTBY <key> <path> <number> JSON.NUMINCRBY <key> <path> <number> JSON.STRAPPEND <key> [path] <json-string> JSON.ARRAPPEND <key> <path> <json> [json ...] JSON.ARRINSERT <key> <path> <index> <json> [json ...] JSON.ARRPOP <key> [path [index]] JSON.ARRTRIM <key> <path> <start> <stop>	✓
查	JSON.GET <key> [INDENT indentation-string] [NEWLINE line-break-string] [SPACE space-string] [NOESCAPE] [path ...] JSON.MGET <key> [key ...] <path> JSON.TYPE <key> [path] JSON.STRLEN <key> [path] JSON.ARRINDEX <key> <path> <json-scalar> [start [stop]] JSON.ARRLEN <key> [path] JSON.OBJKEYS <key> [path] JSON.OBJLEN <key> [path] JSON.DEBUG <subcommand & arguments> JSON.RESP <key> [path]	✓ JSON.DEBUG JSON.RESP 不支持

PegaDB设计与实践 | ZSET&HAHS命令增强

ZSET类型支持聚合、结果过滤操作

ZRANGEBYLEX key min max [LIMIT offset count] [withscores] [aggregation avg/sum/min/max/first/last/] [orderbyscore desc/asc] [between start end]

orderbyscore： 对结果重新排序，支持升序和降序
aggregation： 进行聚合运算，支持求和、平均值、最小值、最大值
between： 只对结果中特定访问内的数值查询

简单时序、前缀匹配场景

HASH类型支持Range操作

HRANGEByLex key min max [LIMIT offset count] [withvalues]

HRemRangeByLex key min max

HLEN key [min max]

范围查询场景

目录

开源社区协作

- 与社区共建，持续回馈社区
  - 主从复制优化: #200 #205 #538
  - 事务: #285
  - 存储引擎优化: #365 #395 #407 #438
  - 集群方案: #302
  - 基于Slot的扩缩容: #430
  - CAS/CAD： #415

- 百度拥有 2 名 Kvrocks PPMC（共4名），4 名 committer，主导多个重要功能的设计和开发

APACHE INCUBATOR

► **Apache Kvrocks Project Incubation Status**

This page tracks the project status, incubator-wise. For more general project status, look on the project website.

► **Description**

Kvrocks is a distributed key-value NoSQL database, supporting the rich data structure.

► **News**

- 2022-04-23 Project enters incubation.

https://github.com/apache/incubator-kvrocks

公众号
Kvrocks Community

目录

后续规划

- 借助云基础设施进一步提升弹性能力，发布Serverless产品
- 借鉴Redis Module生态，支持更丰富的数据模型
- 支持连接器，方便集成大数据生态
- 性能优化：io_uring、快慢请求线程分离


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