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

百亿级数据如何实现秒级复杂查询

光大银行 王磊

关于我

企业架构师，数据中台团队负责人，前IBM咨询顾问

研究方向：分布式数据库、Hadoop等基础架构与数据中台

Pharos架构设计和主要开发者

极客时间《分布式数据库30讲》专栏作者

个人公众号：金融数士

中国光大银行 CHINA EVERBRIGHT BANK

目 录

研究背景

现存产品与解决方案

1. 原生Filter，性能较差，多数场景下无法实际使用
2. HBase + Solr(ES)，架构复杂，性能不佳，维护成本高
3. Phoenix，整体方案较重，社区活跃度低
4. HBase On Cloud，不适合安全性要求较高的场景
5. ClickHouse，并发低，数据存储封闭

我们的需求

1. 非侵入性
2. 高性能
3. 通用性
4. 架构简单
5. 支持事务一致性

Pharos

产品名称来自英文单词 pharos ， 世界上第一座灯塔

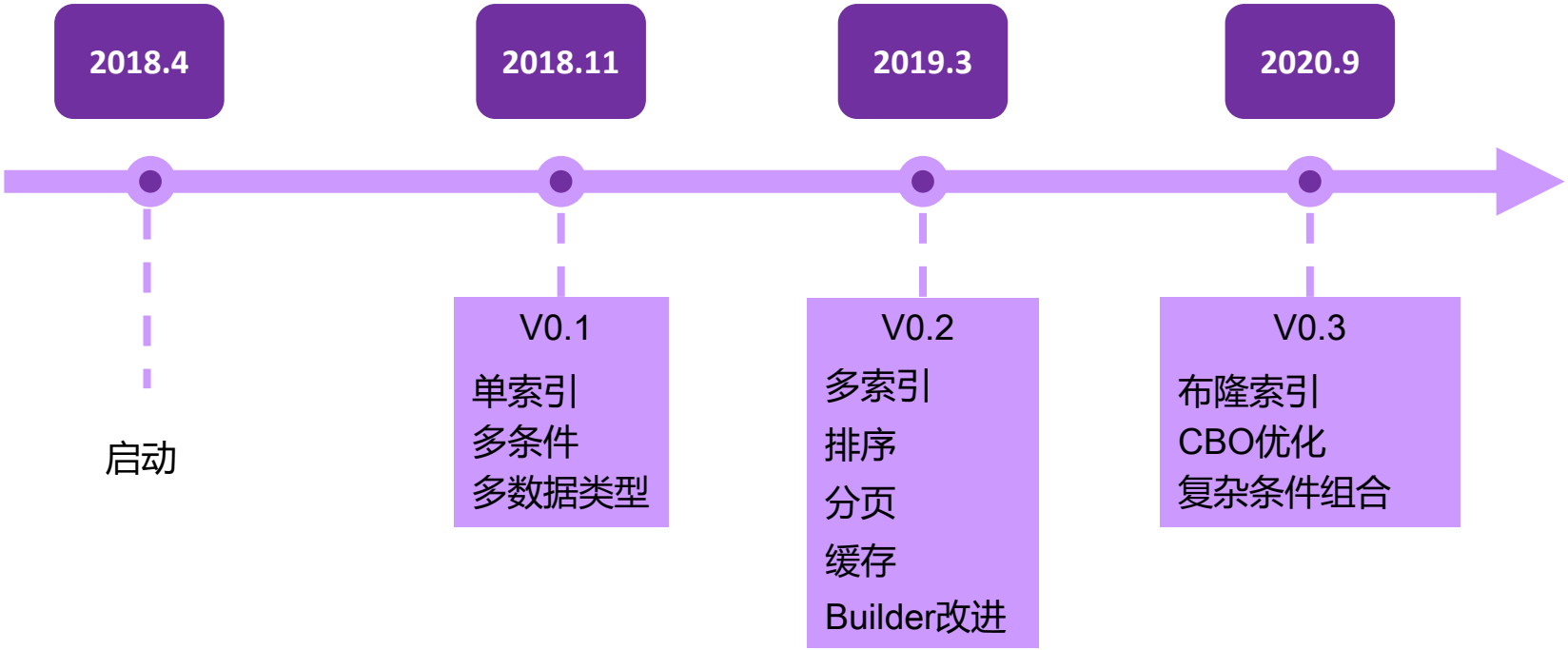
适用业务场景

- 只读场景，T+1数据加载
- 多读少写场景（实验版本）

设计原则

- 非侵入
- 架构简洁

研发过程


2018.4
2018.11
2019.3
2020.9
启动
V0.1
单索引
多条件
多数据类型
V0.2
多索引
排序
分页
缓存
Builder改进
V0.3
布隆索引
CBO优化
复杂条件组合

PharosV0.3 Features

Base on HBase1.2.6/CDH5.8.3-HBase1.2.0

1. 单索引（单列、多列复合），多索引
2. 分页，排序
3. 多条件查询，包括等于、大于、小于
4. 与或逻辑运算
5. 多种数据类型，包括Char、Data、Double等
6. 简单函数，例如count
7. 批量创建/更新索引
8. 布隆索引

目 录

组件构成

Client

API

协调器

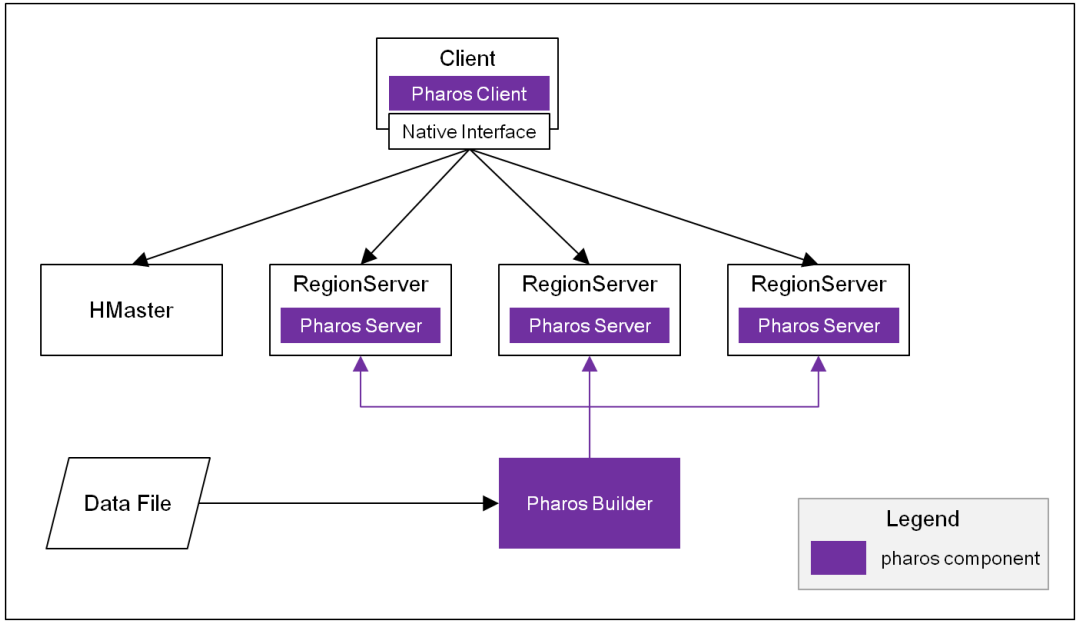
Server

协处理器Coprocessor

主要处理逻辑

Builder

索引创建/更新



索引创建过程

挑战

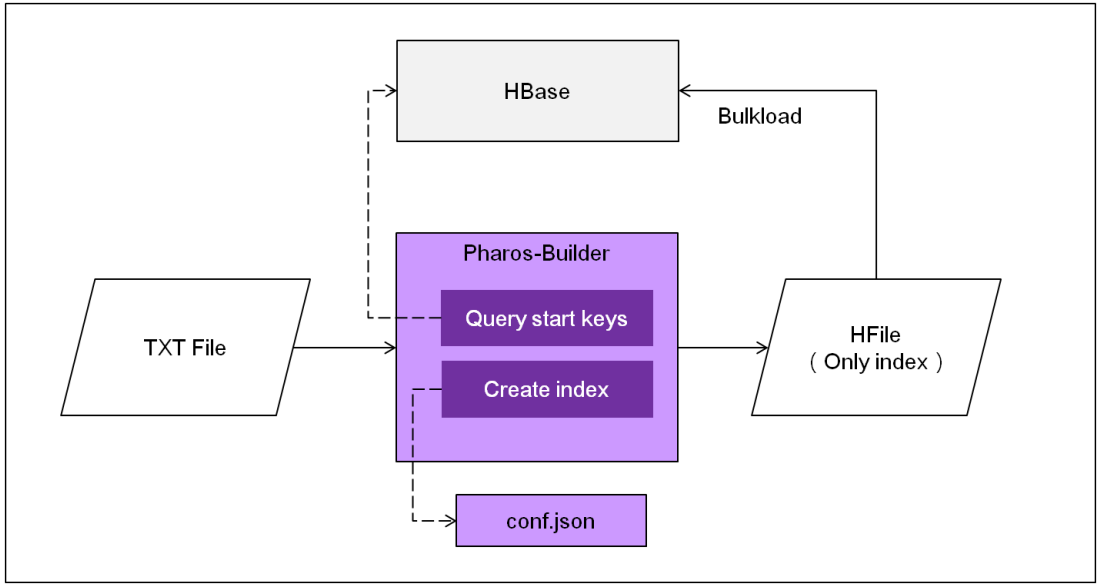
避免Region Split破坏**数据索引同分布**

根据应用中不同的rowkey设计策略，

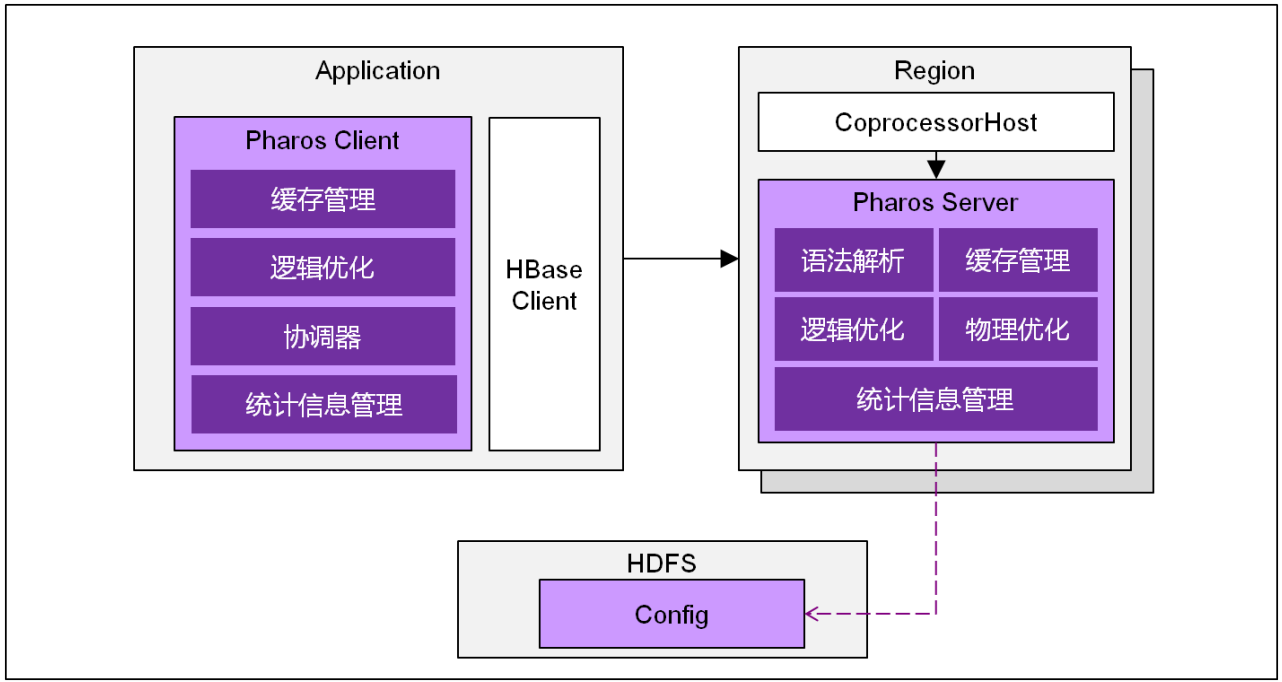
Bulkload可能会触发Split

解决方案（ V0.2 ）

在数据加载完成后，Region稳定后，再加载索引。



运行时架构


Application
Pharos Client
缓存管理
逻辑优化
协调器
统计信息管理
HBase Client
Region
CoprocessorHost
Pharos Server
语法解析
缓存管理
逻辑优化
物理优化
统计信息管理
HDFS
Config

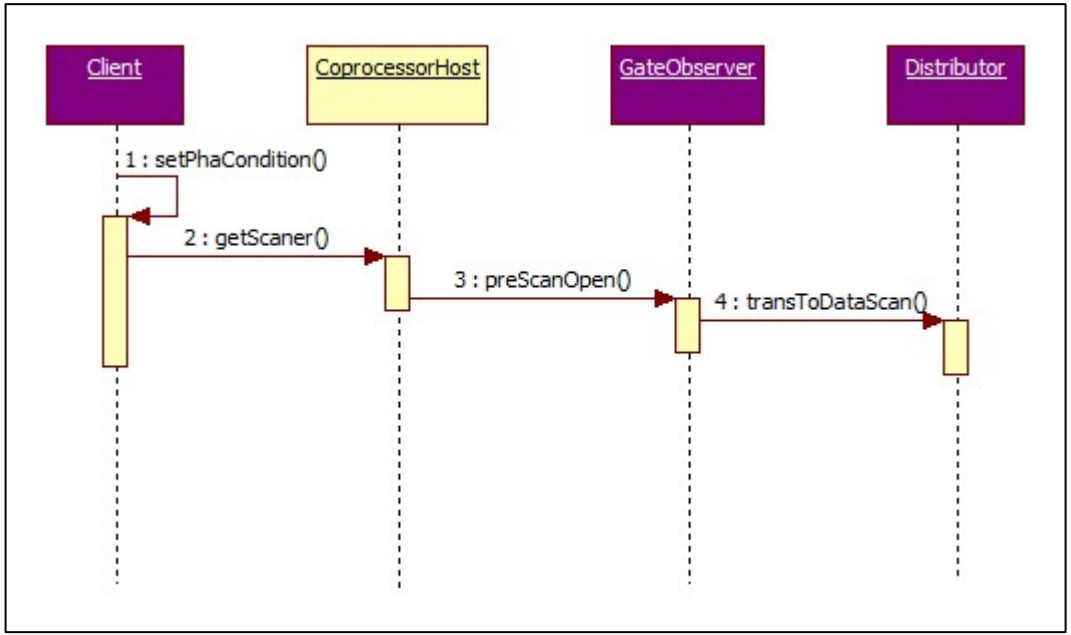
协处理器Coprocessor

Client

1. 定义查询条件
2. 置为Scan对象属性

Server

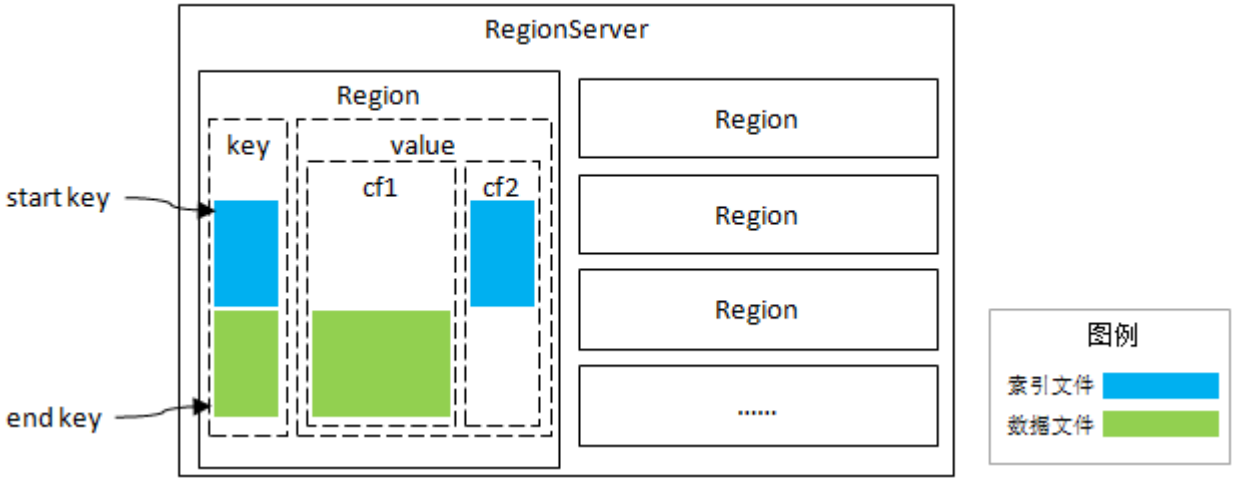
1. 拦截并解析scan
2. 扫描索引信息
3. 使用索引信息，重置Filter



索引存储策略

影子列族存储

通过控制索引rowkey的生成策略，保证数据与索引存储在同一Region不同列族(column family)中，侵入性较低



索引数据结构

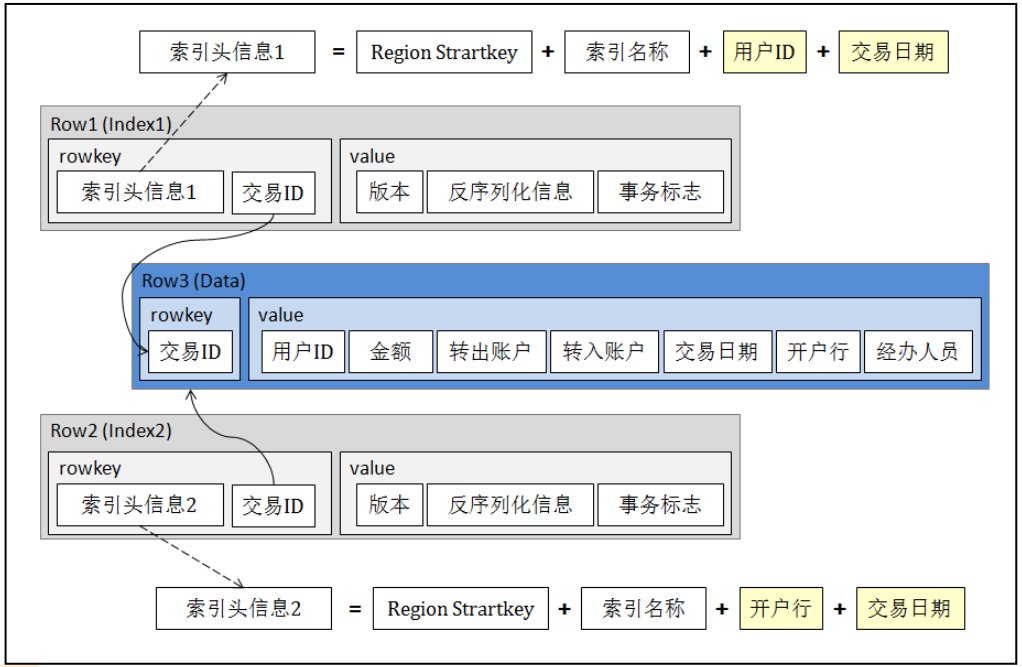
分区索引，索引与数据同分布，查询操作可以下推到每个节点，从而获得更好的性能，避免分布式事务。

Key 设计

1. 起始部分Region StartKey
2. 索引名称/序号
3. 被索引列值
4. 数据行rowkey

Value设计

1. 版本号
2. 反序列化信息
3. 事务标志



目 录

百亿数据量的挑战

PharoV0.2在亿级数据量下，性能较好，但在百亿数据量下存在问题。

挑战1. 查询性能骤降，延迟可能会长达数十秒。

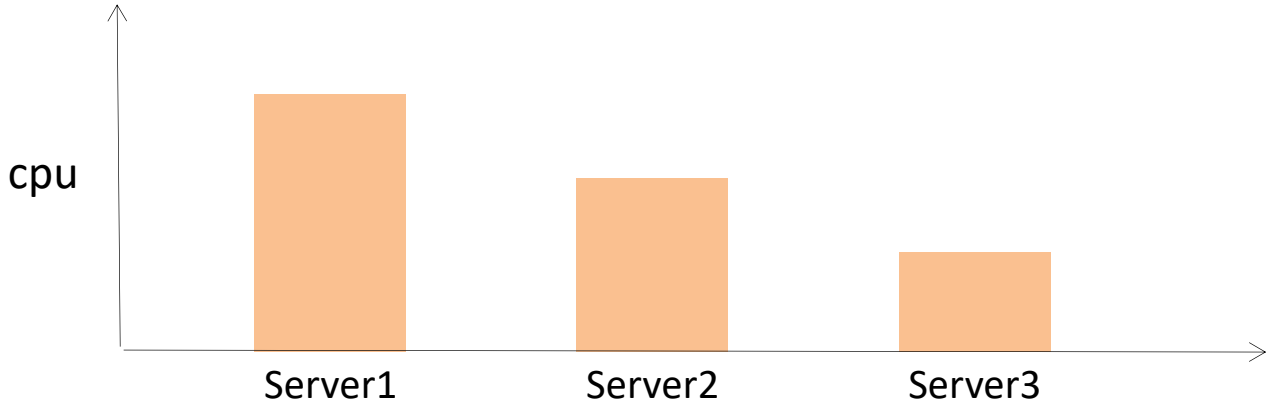
挑战2. 数据加载过程出现大量Region分裂，加载耗时过长，使得延迟加载索引策略的执行操作成本过高。

V0.3 做了三方面改进

1. 实现服务器端的并行处理
2. 借用逻辑桶优雅实现索引与数据的同分布，预处理HFile文件，减少Region被动分裂。
3. 增加布隆索引和数据块机制加速查询

SCAN的顺序访问导致高延迟

问题：Scan是同步串行访问RegionServer，服务器端的性能压力不均衡，海量数据、低选择性查询的延迟过长。



方案一：根据查询条件取哈希值，动态调整访问顺序，打散调整服务器特点。但是，在海量数据、低选择性查询下，仍存在问题。

方案二：用Gets方法替换Scan方法，实现对服务器的并行访问。

动态分区与逻辑桶

Split策略：KeyPrefixRegionSplitPolicy，将前N位相同的Rowkey保持在一个Region。

KeyPrefixRegionSplitPolicy.prefix_length设置为6，Rowkey的前6位构成一个”逻辑桶”

桶编号 = 4位日期（yymm） + 两位数字（十六进制，256）

考虑因素

1. 数据均衡，负载均衡。

桶数量与数据增量的关系，桶内的记录数量均衡

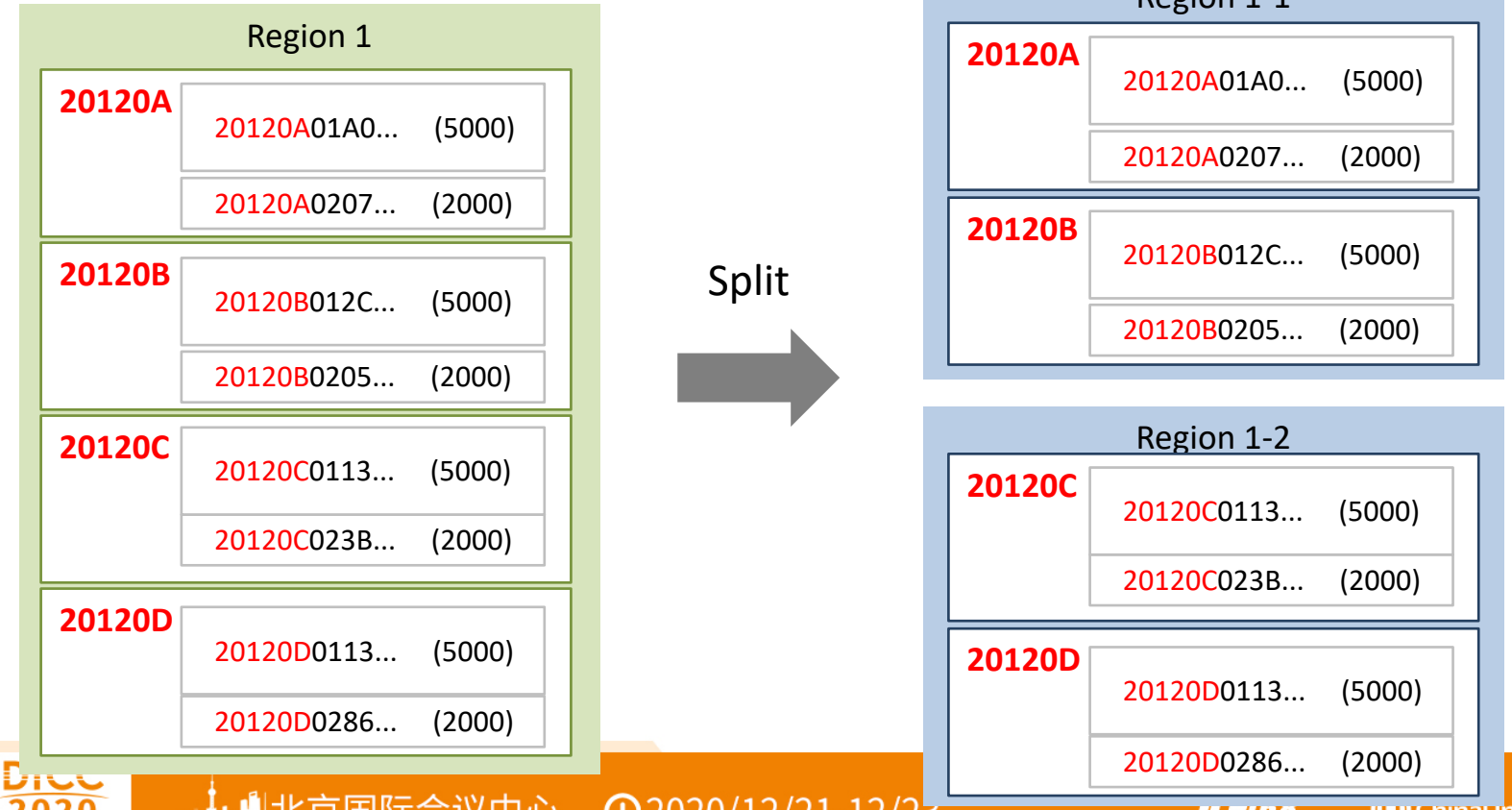
2. 集群规模，最大并发程度。

桶数量与节点数的关系

3. 单个Region的大小限制。

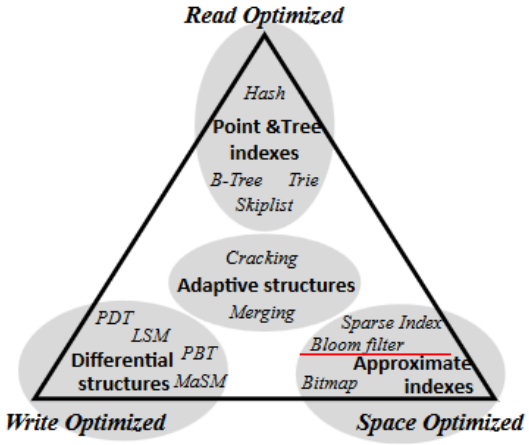
按照HBase默认Region大小为10G，每月增量数据约为2.5T。

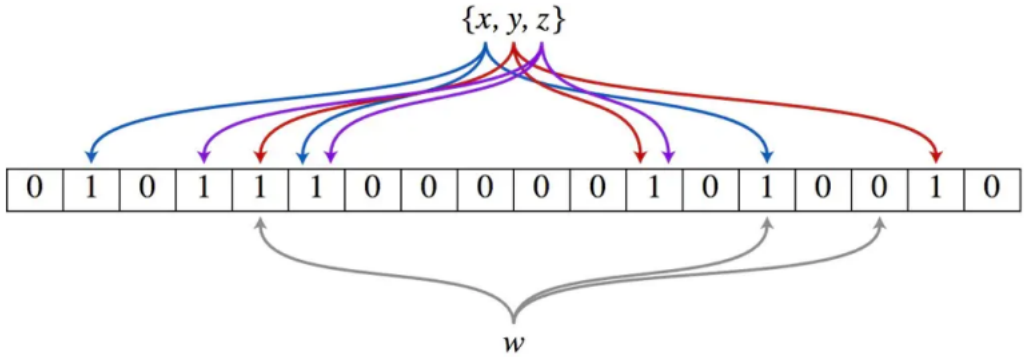
基于逻辑桶的Region拆分



布隆过滤器

BloomFilter 是一种空间效率非常高的数据结构，用于判断一个元素是否属于这个集合。BloomFilter 会出现假阳性，误判概率与存储空间大小呈负相关。

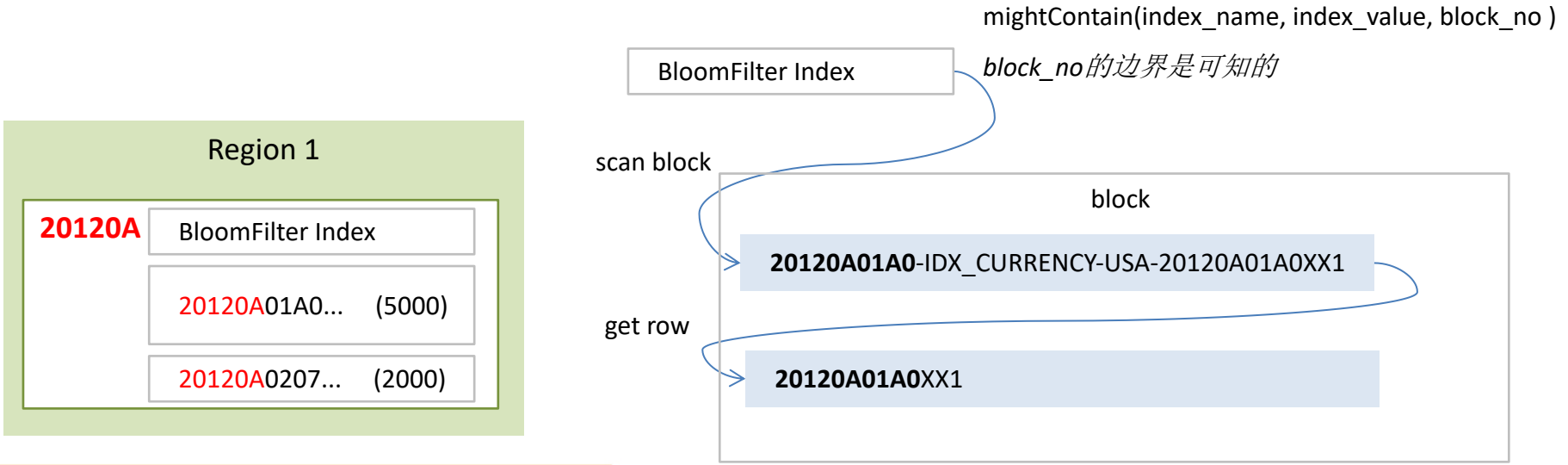




用数据块与布隆过滤器加速查询

问题： 在百亿数据下，使用普通二级索引的匹配低选择性条件，付出的代价过高，可以长达数十秒。

解决方案：以数据块为单位，使用布隆索引，可以提供粗粒度匹配，大幅压缩匹配时间。



加入我们

数据中台团队

没有996

这里有

数千万用户，

海量的数据，

丰富的金融场景，

严苛的稳定性挑战，

我们致力于打造业界领先的数据中台，如果你对微服务、分布式数据库、大数据等技术感兴趣，欢迎加入我们。

zh-wanglei@cebbank.com

公众号：金融数士

极客时间专栏
《分布式数据库30讲》

THANKS