从阿里HBase到Lindorm
大规模结构化存储的演进与思考

阿里巴巴-沈春辉(天梧)

2018年4月

自我介绍

Alibaba Group

- 沈春辉，花名：天梧
- @阿里巴巴-存储事业部，资深技术专家
- HBase Committer&PMC
- 专注在大数据领域，多次参与双11狂欢节的技术保障，在分布式、性能优化、系统架构等方面有丰富的积累与实战
- 目前负责新一代大规模NoSQL系统(Lindorm)的建设

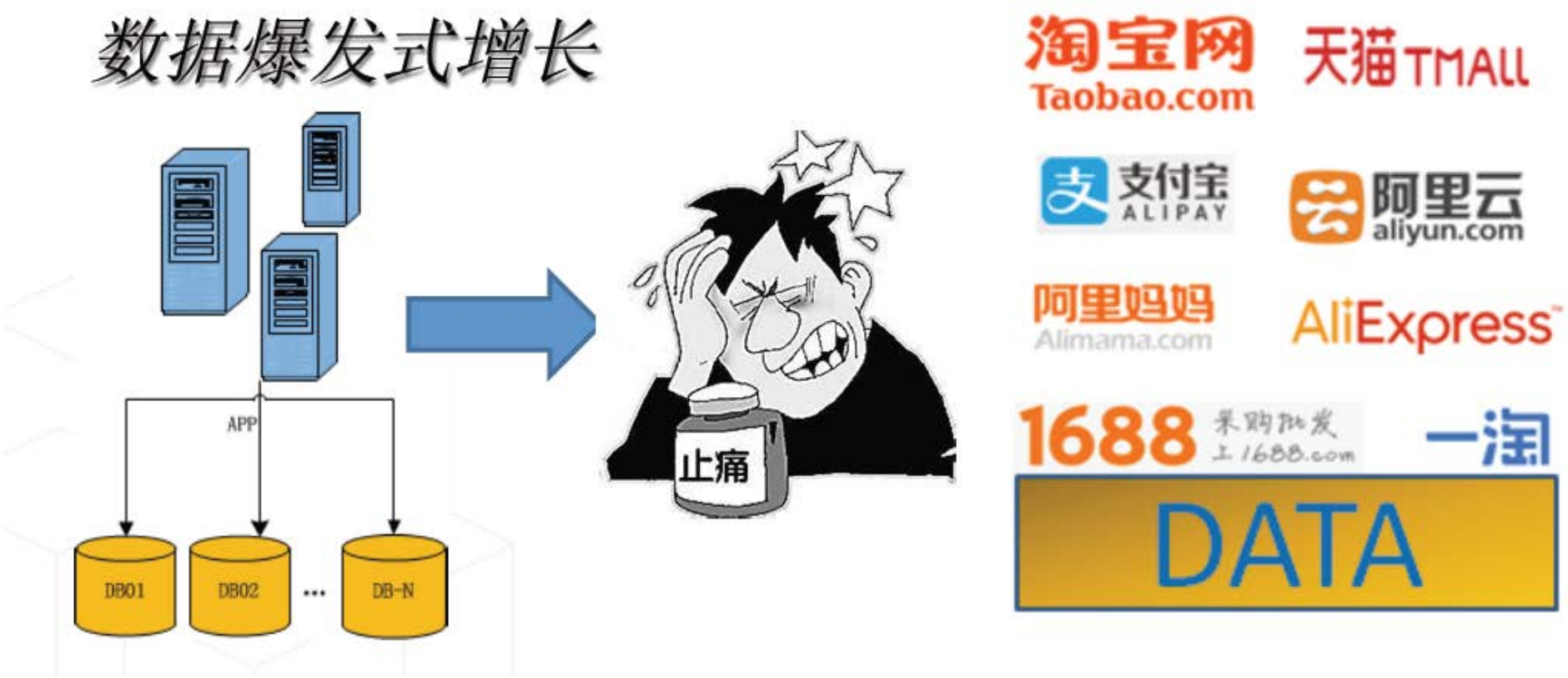
沈春辉

浙江 杭州

扫一扫上面的二维码图案，加我微信

Alibaba Group

互联网数据的爆发增长

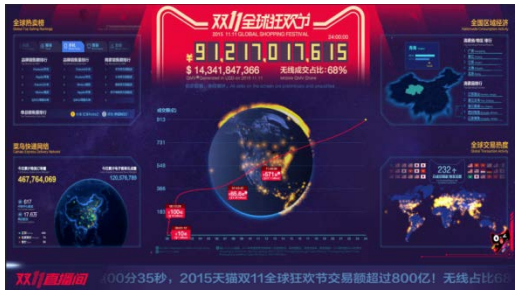


数据驱动的新兴场景

风控大脑

芝麻信用

借贷保险决策



实时大屏

生意参谋

智能客服

阿里HBase的发展

Alibaba Group

分布式NoSQL的选择

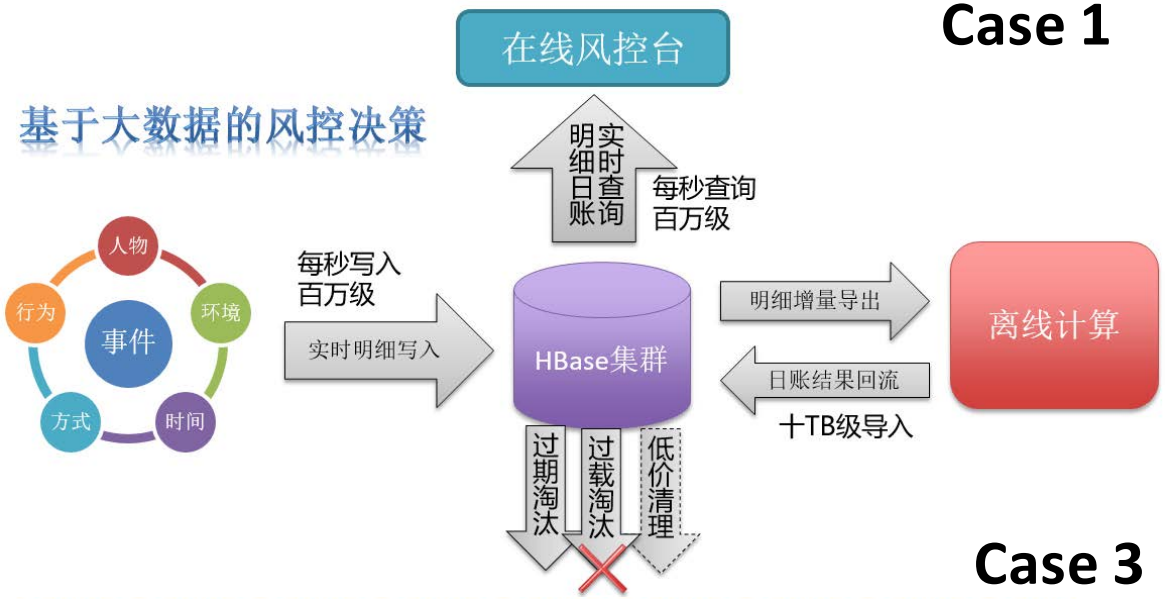
- Why HBase
  - 2011年启用
  - 活跃社区，Hadoop生态
  - Facebook成功案例
  - 理论信仰 (Google Paper: BigTable)
- Other Choice
  - Cassandra
  - 自研

- 表格模型
- 基本功能(~~Join~~、~~GroupBy~~)
- ~~分库分表~~

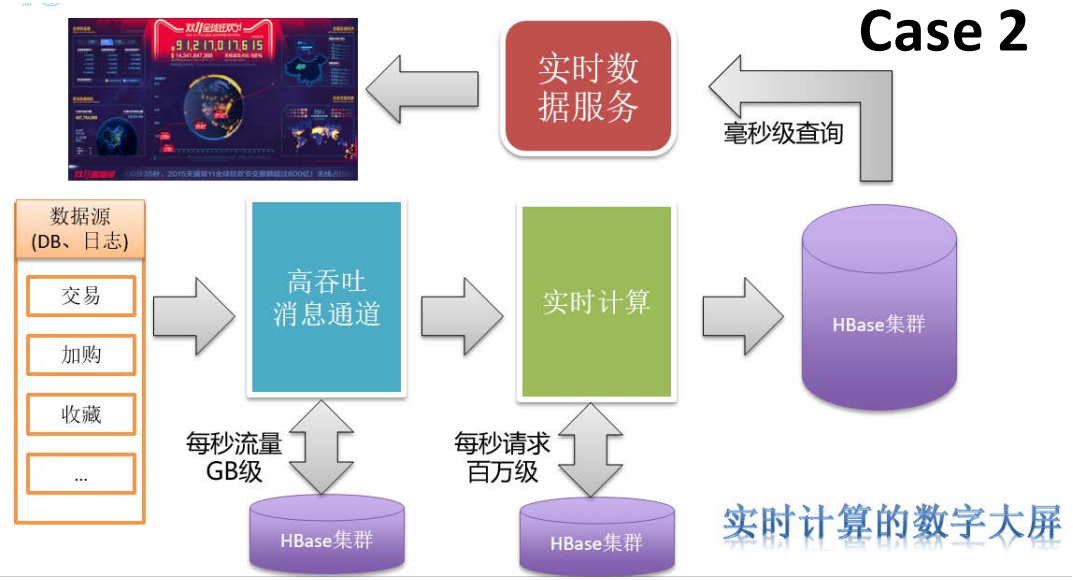
阿里HBase的发展

Alibaba Group

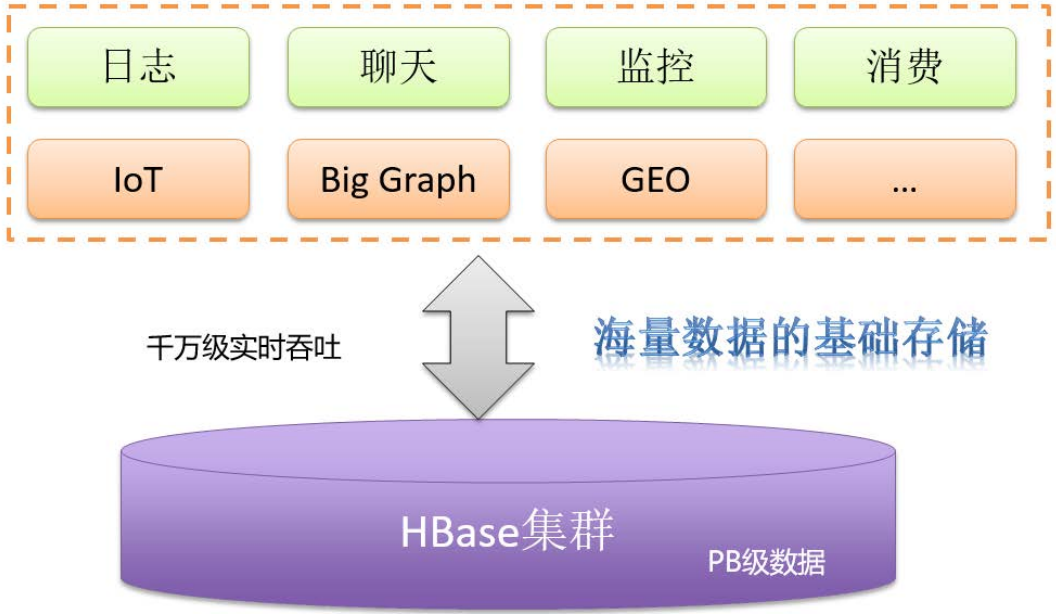
Case 1



Case 2

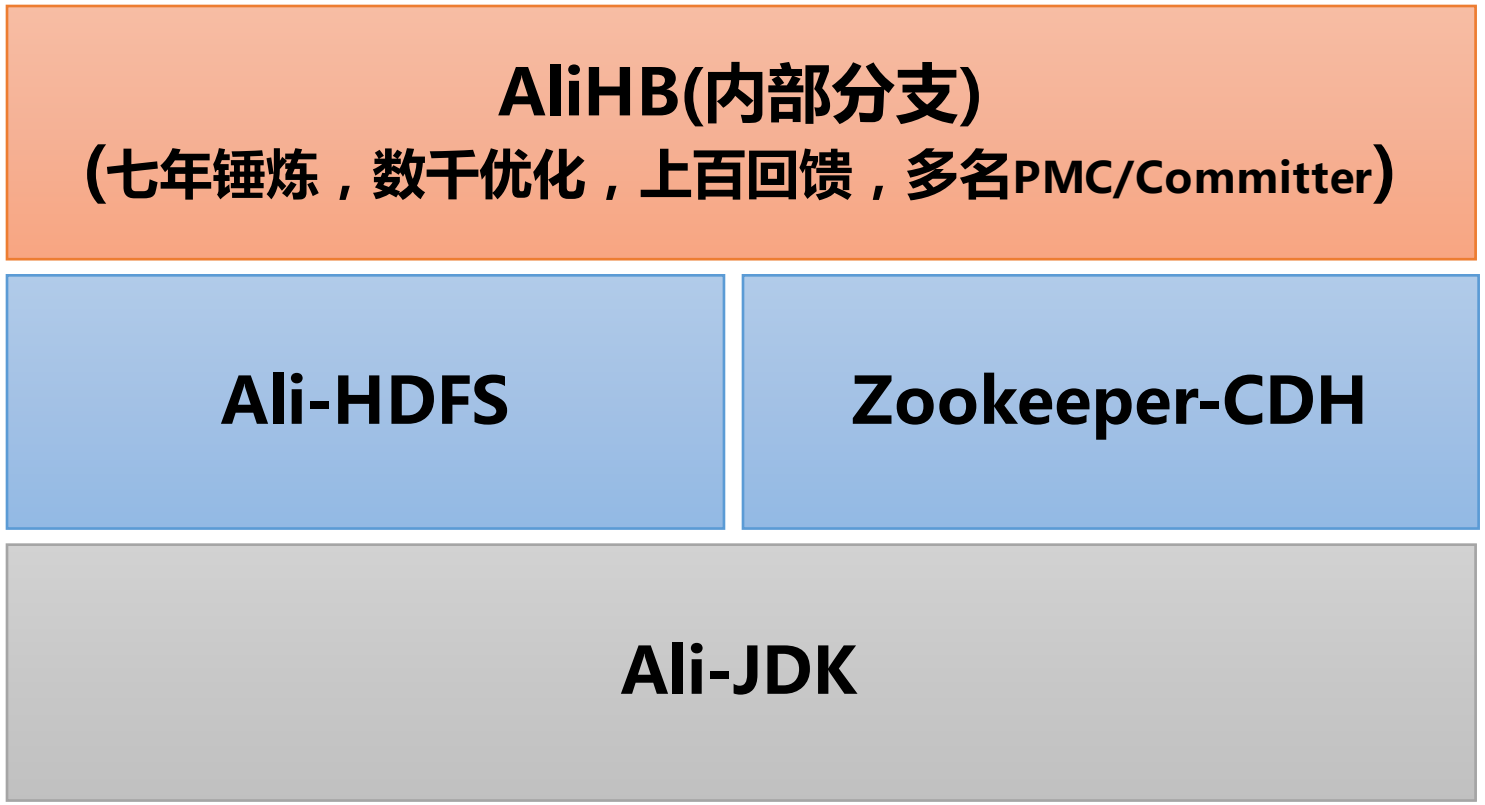


Case 3



面向大数据场景的结构化存储
高效支撑数据驱动、实时决策型业务

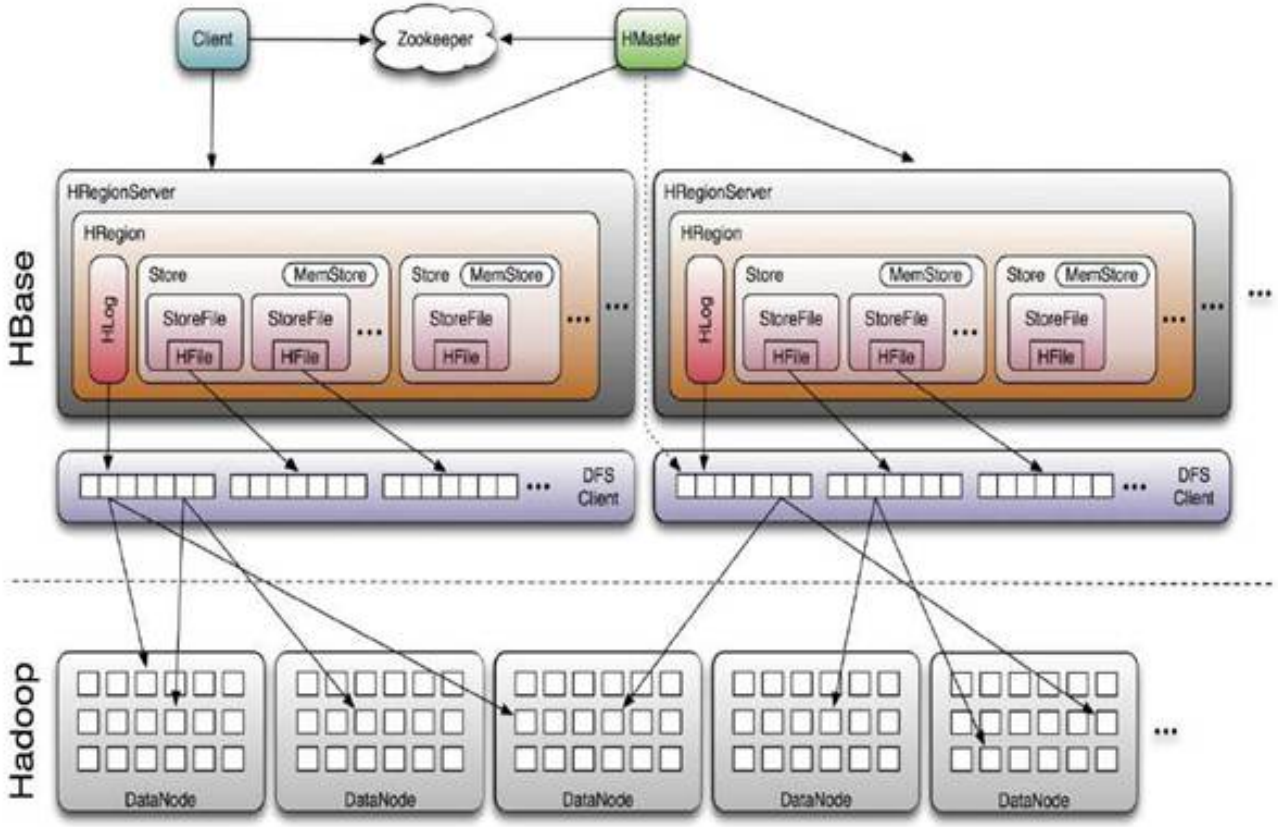
阿里HBase的发展



- **性能领先**
  - ✓ 高效数据结构、锁优化、请求合并等
- **功能丰富**
  - ✓ SQL、二级索引
  - ✓ 多租户、异构存储、异步接口等
- **稳定可靠**
  - ✓ 完善的HA架构
  - ✓ MTTR快2-10倍
  - ✓ 多年双11沉淀
- **运维灵活**
  - ✓ 全面的监控埋点
  - ✓ 全链路跟踪
  - ✓ 无缝迁移
- **版本迭代**
  - ✓ 0.90、0.92、0.94、1.1、1.2

- 面向阿里经济体中所有BU及6000+开发者，提供一站式大数据场景的结构化存储服务
- 维护管理1万+个物理节点，100+个集群，支撑亿级TPS，百PB数据

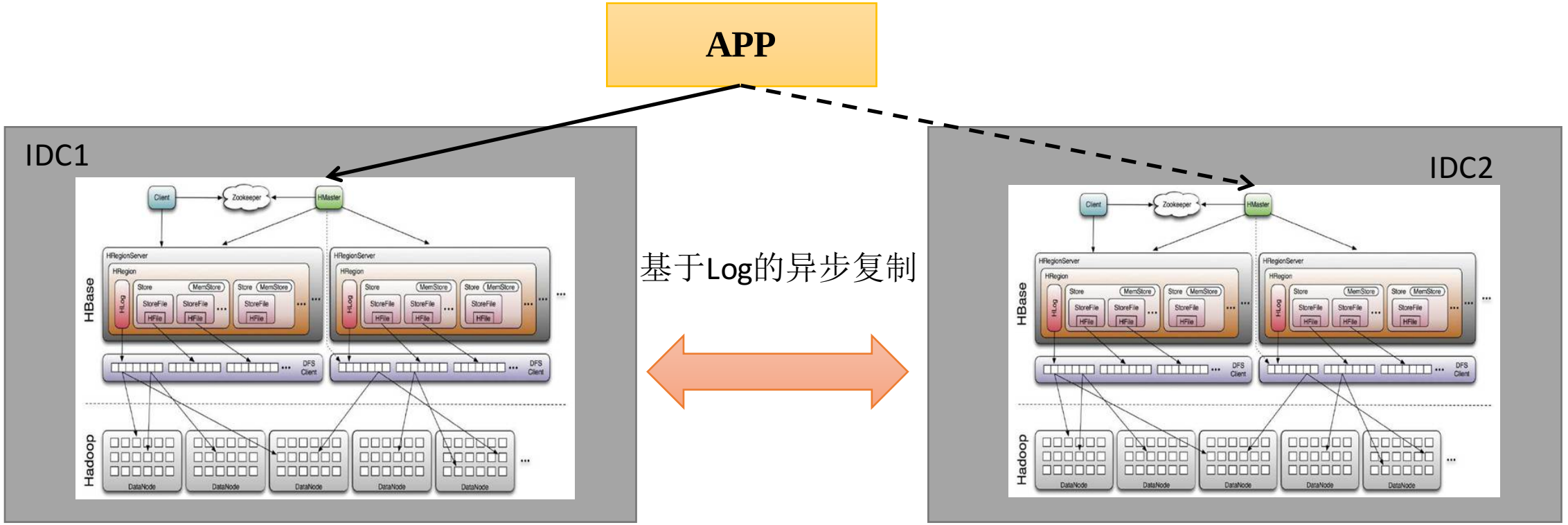
过去的需求及解决—跨机房容灾



- 不具备跨机房容灾
- 故障恢复慢
  - ✓ 单节点，3~10分钟
  - ✓ 大规模集群，小时级别

10

过去的需求及解决—跨机房容灾



- ~~不具备跨机房容灾~~
  - ✓ 数据非强一致
- 故障恢复慢
  - ✓ 单节点，3~10分钟
  - ~~✓ 大规模集群，~5分钟~~

- 成本双倍
- 维护复杂性大
  - 健康诊断、流量切换

过去的需求及解决—HBase As A Service

Alibaba Group

- 访问表达复杂
  - ✓ RowKey设计与开发绑定
  - ✓ 多条件查询
  - ✓ 与异构系统协接困难
  - ✓ 无数据类型，缺乏约束
- 客户端逻辑重
  - ✓ 性能Trace困难
  - ✓ 易发生缺陷，升级迭代不方便
- 安全性
  - ✓ 保障HDFS、ZK整个技术栈的安全，很复杂
  - ✓ 依赖外部的Kerberos体系
- 请求抖动

- 一些解决方法的尝试
  - ~~Proxy~~：运维成本、性能影响
  - Phoenix：有限帮助
  - 用户名+密码认证
  - Dual Services

小结

Alibaba Group

HBase’s good sides

- Auto Sharding
- LSM
- Layered Storage
- Schema Free
- M-S Framework

Not enough for some requirements

- Availability
- Query Ability
- Cross-Region Consistency
- Thin Client
- Latency Less-Spiking

Lindorm的目标

- 定位：大规模、高并发、快速灵活的NoSQL系统

- 技术方向：
  - ✓ 极高吞吐、极快伸缩、极低存储
  - ✓ 轻事务、弱分析
- 能力特征
  - ✓ PB级规模
  - ✓ MS级响应
  - ✓ 跨地域容灾
  - ✓ 99.999%可用性
  - ✓ 多模型

Lindorm的功能

Alibaba Group

- 基本概念
  - ✓ 用户(UserName + Password)
  - ✓ 访问权限(ACL)：Universe/Namespace/Table/
  - ✓ 命名空间(namespace)
  - ✓ 生命周期(TTL)
  - ✓ 多版本(Version)
- 数据模型
  - ✓ Relational Table
  - ✓ Wide Column
  - ✓ More in future…

Lindorm的功能

Alibaba Group

Relational Table

- ✓ 列(Column)
  - 数据类型：整型、浮点型、字符串、二进制等
  - 主键列（PK1、PK2）、普通列（Col3、Col4）
- ✓ 索引
  - 表的存储=1个主索引+N(≥0)个二级索引
  - 主索引=主键列1+…+主键列N
  - 二级索引
    - 主键重排
    - 单列倒排
    - 多列组合
- ✓ 数据访问
  - JDBC
  - LQL(Lindorm Query Language)

PK1	PK2	Col3	Col4
A	1	true	2017-01-01
B	2	true	2017-02-03
C	3	false	2017-04-05
…	…	…	…
J	5	true	2017-06-07
…	…	…	
Z	9	false	2017-08-09

Lindorm的功能

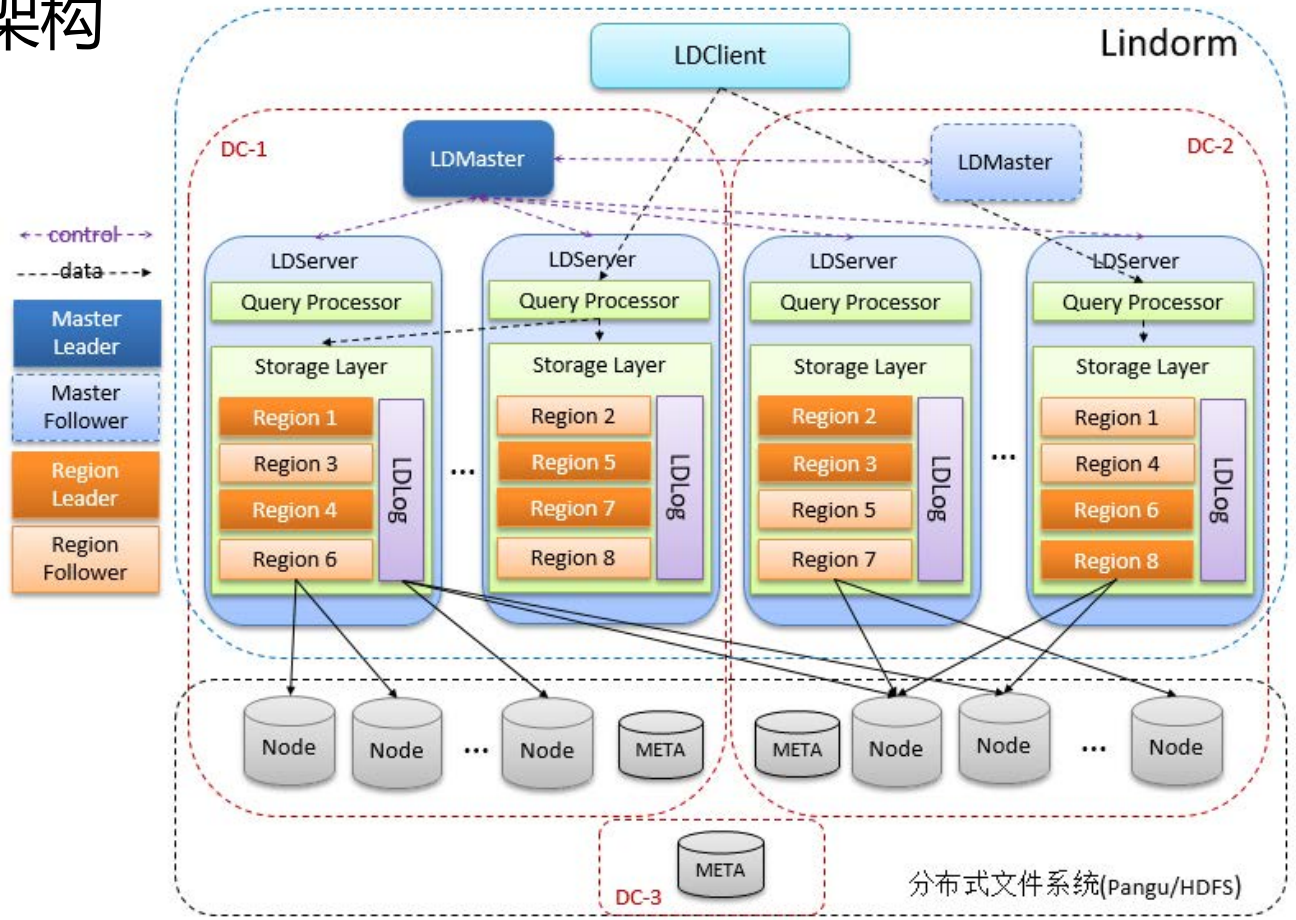
Alibaba Group

- Wide Column
  - ✓ RowKey
    - 行记录的唯一排序键
  - ✓ 列簇(Column Family)
    - 纵向切割，面向列的存储思想
  - ✓ 列(Column)
    - 无需定义，完全动态，无数量上限
    - 统一的数据类型byte[]
    - 独立的数据时间戳
  - ✓ 数据访问
    - 自定义API：Put/Get/Scan/Delete

Row Key	Time Stamp	ColumnFamily contents	ColumnFamily anchor
"com.cnn.www"	t9		anchor:cnnsi.com = "CNN"
"com.cnn.www"	t8		anchor:my.look.ca = "CNN.com"
"com.cnn.www"	t6	contents:html = "<html>..."	
"com.cnn.www"	t5	contents:html = "<html>..."	
"com.cnn.www"	t3	contents:html = "<html>..."	

Lindorm的设计

- 高可用->冗余多副本
  - ✓ Not only for persistent data (HBase)
  - ✓ But also for memory data
- 总体架构



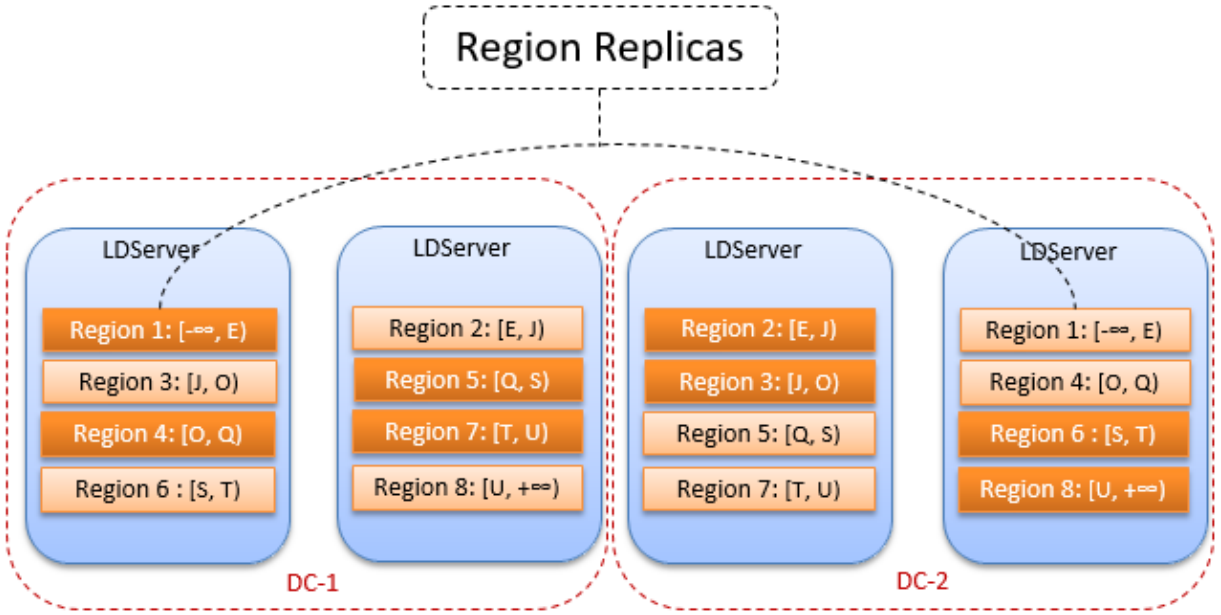
- ✓ 自动分区
- ✓ 多副本
- ✓ 存储计算分离
- ✓ 无状态访问
- ✓ 无单点

Lindorm的设计

Alibaba Group

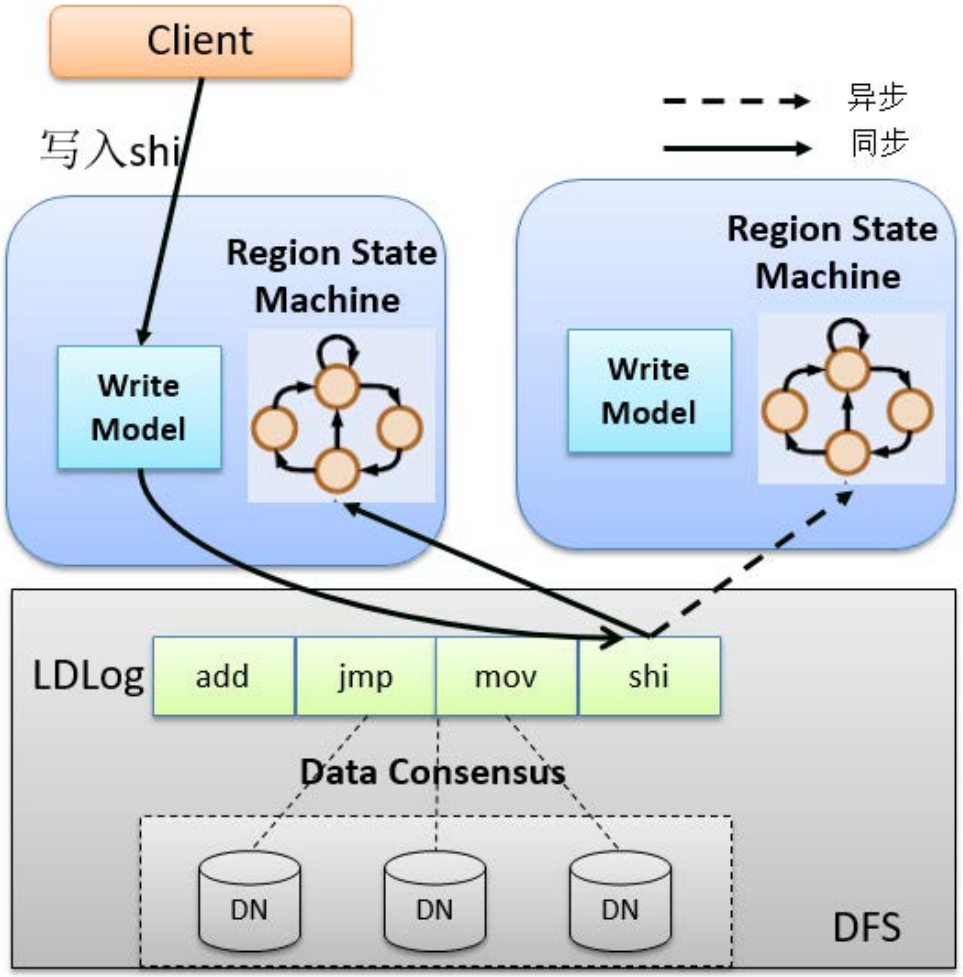
- 持久化存储在分布式文件系统
  - ✓ 成本更弹性
  - ✓ 抗热点更优秀
  - ✓ 负载更均衡
  - ✓ 数据管理更个性
- 使用盘古作为Lindorm的底层文件系统
  - ✓ 全分布式元数据管理
  - ✓ 充分发挥新一代网络和存储硬件的红利(e.g. SPDK, RDMA)
  - ✓ 高效远程访问，支持统一大存储

Lindorm的设计



- ✓ 一个Region拥有多个副本
- ✓ 副本间数据保持一致
- ✓ 1 Leader + N Follower
  - 基于心跳选举
- ✓ 副本分布在不同DC
  - LDMaster负责

Lindorm的设计



- Region副本间的数据一致性
  - ✓ LDLog是一种WAL（Write-ahead logging）
  - ✓ LDLog持久化存储于DFS（高可用）
  - ✓ 数据从LDLog中异步回放消费至其他Replicas
    - 回放点位
    - Log Buffer
  - ✓ 强一致
    - Only Leader Region提供读写
    - 数据从Leader的LDLog回放至Follower Region
    - Follower晋升为Leader之前保证数据完全一致
  - ✓ 最终一致
    - Leader和Follower Region都提供读写
    - 数据相互同步
  - ✓ 更多一致性等级
    - Timestamp Consistency
    - 会话一致性
    - 有界一致性

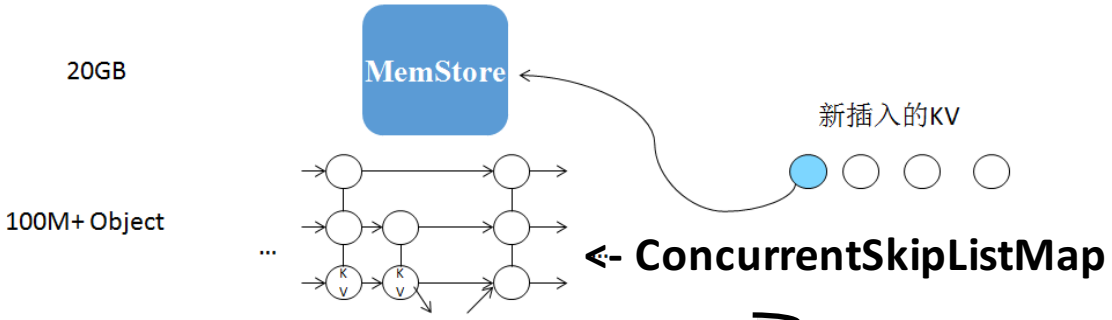
Lindorm的工程

Alibaba Group

- 在HBase上遇到的GC挑战
  - ✓ 内存规模越来越大，堆空间百GB以上
  - ✓ YGC频率高，停顿时间100ms以上
  - ✓ 内存碎片导致超长停顿的FullGC

- 在Lindorm消除JVM GC影响
  - ✓ 写buffer+读cache，使用新型数据结构，在应用态管理内存
  - ✓ 使用新GC算法(powered by Alibaba JVM)，优化频繁创建与回收的临时对象
  - ✓ GC停顿控制在5ms

Lindorm的工程



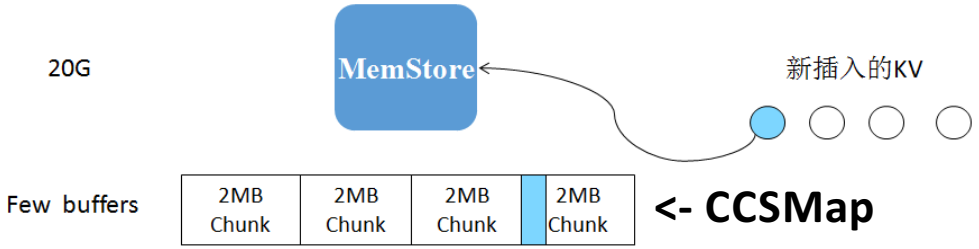
内存中保留对象多，小对象场景更加明显

KV对象频繁创建，几乎都要晋升到old区

对象间引用关系变化频繁

GC沉重

✓JDK自带Map的瓶颈



MemStore以2MB为Chunk申请，内部结构扁平无对象参与

插入MemStore时，不需要创建对象，不需要修改引用

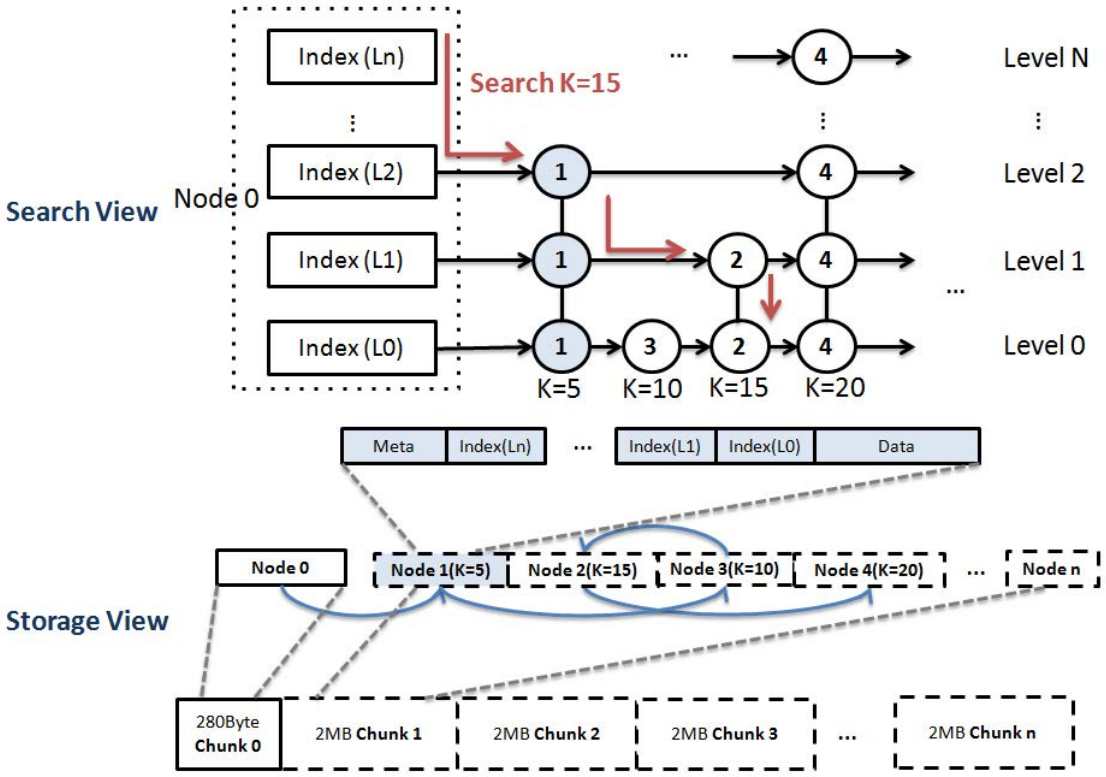
所有KV一直存储在Old区，不需要晋升

GC友好

✓面向场景的Map重构

Lindorm的工程

更快、更省的SkipListMap（CCSMap）

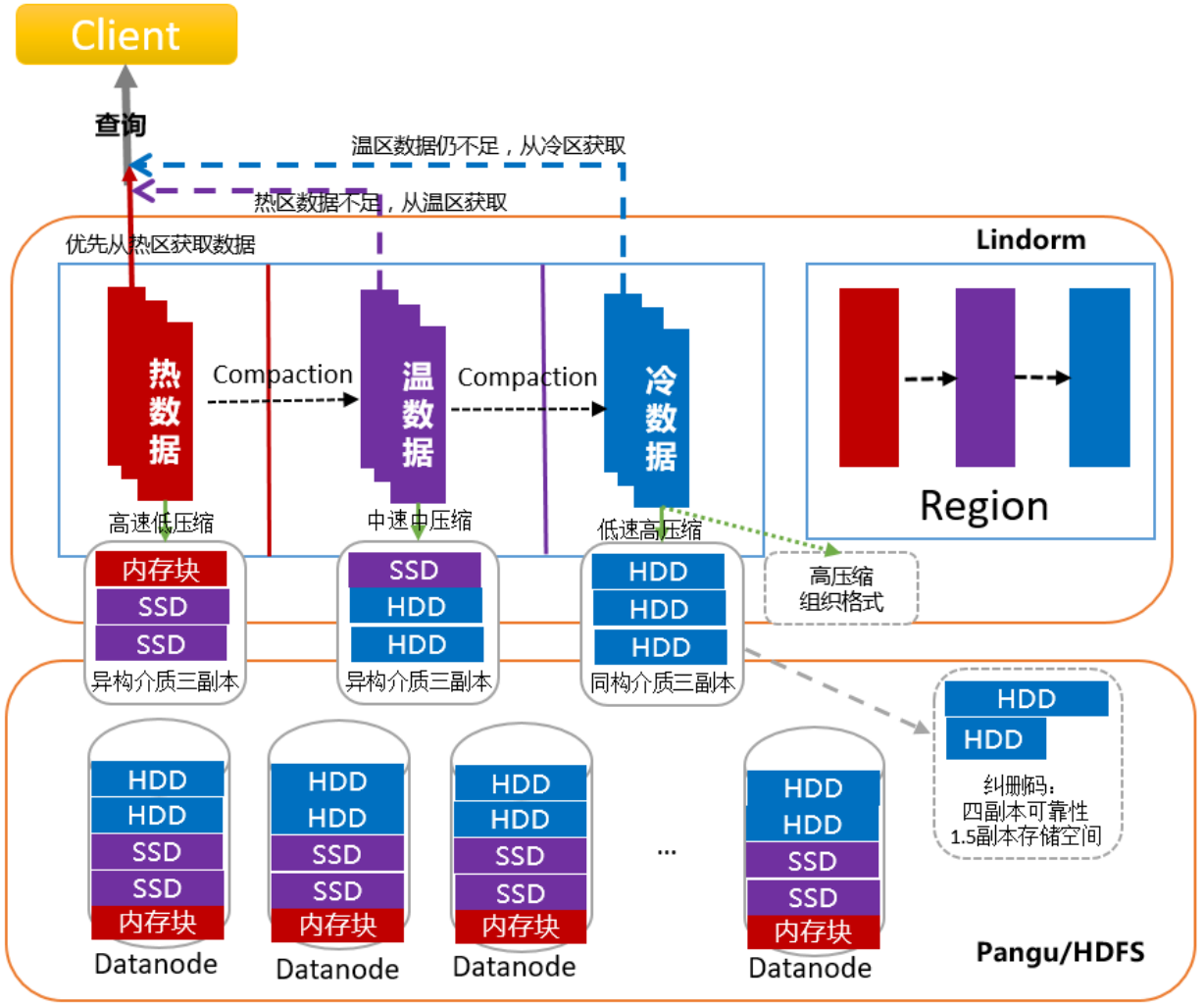


- ✓ 高压缩，内存开销减少一半
- ✓ 读写速度提升20%
- ✓ 支持批量接口，连续范围的插入性能提升500%
- ✓ GC优化24倍

注：CCSMap vs java.ConcurrentSkipListMap
更多：http://geek.csdn.net/news/detail/246136

Lindorm的工程

冷热分层异构存储



- ✓ 技术
  - 数据冷热识别
  - 分层compaction算法
  - 异构管理
  - 访问过滤
- ✓ 效果
  - 业务透明
  - 自动分层
  - 节省存储
  - 查询加速

Lindorm的总结（一）

Alibaba Group

	Lindorm	HBase
性能	基准测试，3-10倍于HBase	
	高性能无锁内存结构	
	合并提交、异步化、协程	
	高性能网络编程	
	GC-Less编程	
	支持盘古2	
	高效的多副本同步	
	可检索编码	
	CPU热点并行化	
	硬件加速(ing)	
功能	完全的二级索引	原生不支持
	Region级联Split	不支持
	冷热自动分层	不支持
	离散TTL	不支持
	实时订阅	不支持

Lindorm的总结（二）

Alibaba Group

	Lindorm	HBase
开发易用性	关系模型：支持JDBC驱动、标准SQL及类SQL的API	Get/Put/Scan等KV式API
	丰富的数据类型	无数据类型，统一为binary bytes
	丰富的查询支持	依赖主键(Rowkey)的拼接及Filter设计
	轻客户端，稳定，依赖少(10-)	重客户端，迭代频繁，依赖多(30+)
	使用简单的用户认证及ACL	复杂的kerberos认证
可用性	单机硬件故障，无感知	五分钟左右，服务不可用
	机房级故障，无感知	需额外备集群+配套系统， 五分钟以上，服务不可用
	分区移动/Split/Merge，无感知	分区移动/Split/Merge，短暂毛刺
	网络抖动，快速规避	网络抖动，服务抖动
	机器Hang或者慢盘，快速规避	机器Hang或者慢盘，服务抖动
运维性	一个集群的视角和操作	主备维护复杂
	一键迁移	

未来规划

Alibaba Group

- 统一存储、容器化
- Pluggable存储引擎
- 新硬件结合： AEP、FPGA
- 更多模型支持：对象、图、时空等
- 开源

沈春辉
浙江 杭州

谢谢指导

THANKS!

求才若渴，欢迎加入我们，杭州、硅谷、深圳、北京

扫一扫上面的二维码图案，加我微信