高可用HBase的技术实践

沈春辉

2016-3

内容大纲

- HBase特点简介
- HBase在阿里使用状况
- Ali-HBase的高可用实践

HBase的特点简介

- 描述
  - 以表的形式组织数据，提供实时更新、增量导入、随机查询、条件范围查询能力的分布式NOSQL数据库
- 基因
  - 自动分区
  - LSM-Tree
  - 基于K-V的行组织
  - 存储计算分离(base on HDFS)
  - 单点服务

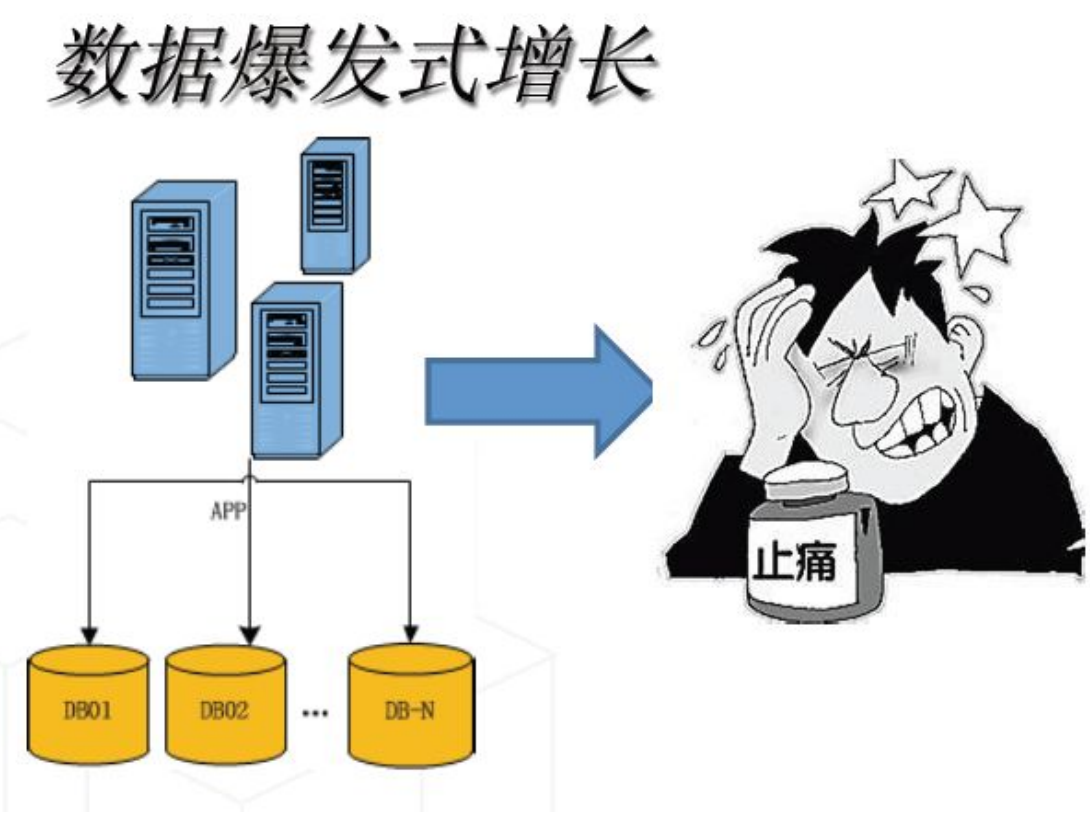
HBase的特点简介

数据模型	访问方式	性能与扩展	安全与稳定	数据导入导出
1. **松散的表Schema**，列的名字、数目、长度无需定义 2. 数据类型：唯一支持字节数组(byte[]) 3. 支持**多版本**	1. 实时写入/更新/删除，支持**批量、异步**等方式 2. 后台导入，直接**生成存储格式**的文件，十分高效 3. 设置**数据有效期**，过期自动删除 4. 指定主键(Row)的随机查询 5. 有条件(为主键或列设定条件)的范围查询 6. **协处理器** (类似于RDB中的触发器与存储过程) 7. 事务：支持单行、同分区跨行的事务 8. 索引：主键索引，不支持列的二级索引 9. 数据**强一致**性，强持久性 10.程序语言支持：原生客户端仅支持JAVA，C、PHP可以通过代理方式访问(自带Thrift框架) 11.外部扩展支持，SQL引擎(Apache-Phoenix，近SQL-92标准，JDBC驱动)	1. 水平扩展，支持千台物理机级规模 2. 扩容无需数据迁移，**即扩即用** 3. 大表自动分裂，支持分区在线合并 4. 扩展能力依赖表分区，Row设计需要**防热点** 5. 存储层采用LSM树，相比于B-Tree（读写对等），写能力>读能力	1. 存储层默认**三副本**，数据可靠性高 2. 支持**表快照**，方便冷备 3. 系统内部采用M-S架构 4. Master支持热备，Master故障影响DDL，不影响DML 5. Slave故障，影响可用性，部分数据区域的DML不可用，会自动恢复 6. 支持系统级的**M-M灾备** 7. 支持用户认证与授权	1. 跨系统，支持导入/导出CSV格式的数据 2. 同系统，支持distcp直接拷贝底层存储文件，快速导入 3. 使用sqoop，在HBase/mysql/oracle/hive等系统间相互迁移数据

HBase的特点简介

场景	适用描述
结构化数据 **在线存取**	用户的前台数据实时读写HBase，如电商交易行为产生的各种记录
高吞吐 数据写入	日志、消息、监控、聊天等需要高吞吐写入的数据存取
海量数据 **实时写入与查询**	安全风控场景，在线、离线写入大量用户行为数据，实时查询判断行为风险
实时流计算 的底层存储	作为流计算平台(Galaxy、JStorm)的中间计算和结果数据的存储

HBase在阿里的使用状况



淘宝网 Taobao.com
天猫 TMALL
支付宝 ALIPAY
阿里云 aliyun.com
阿里妈妈 Alimama.com
AliExpress
1688 采购批发 上1688.com
一淘
DATA

HBase在阿里的使用状况

- 2011.5 上线第一个HBase应用
- 到目前

高可用HBase---目标

– SLA

Level of Availability	Percent of Uptime	Downtime per Year	Downtime per Day
1 Nine	90%	36.5 days	2.4 hrs.
2 Nines	99%	3.65 days	14 min.
3 Nines	99.9%	8.76 hrs.	86 sec.
4 Nines	99.99%	52.6 min.	8.6 sec.
5 Nines	99.999%	5.25 min.	.86 sec.
6 Nines	99.9999%	31.5 sec.	8.6 msec

高可用HBase---目标

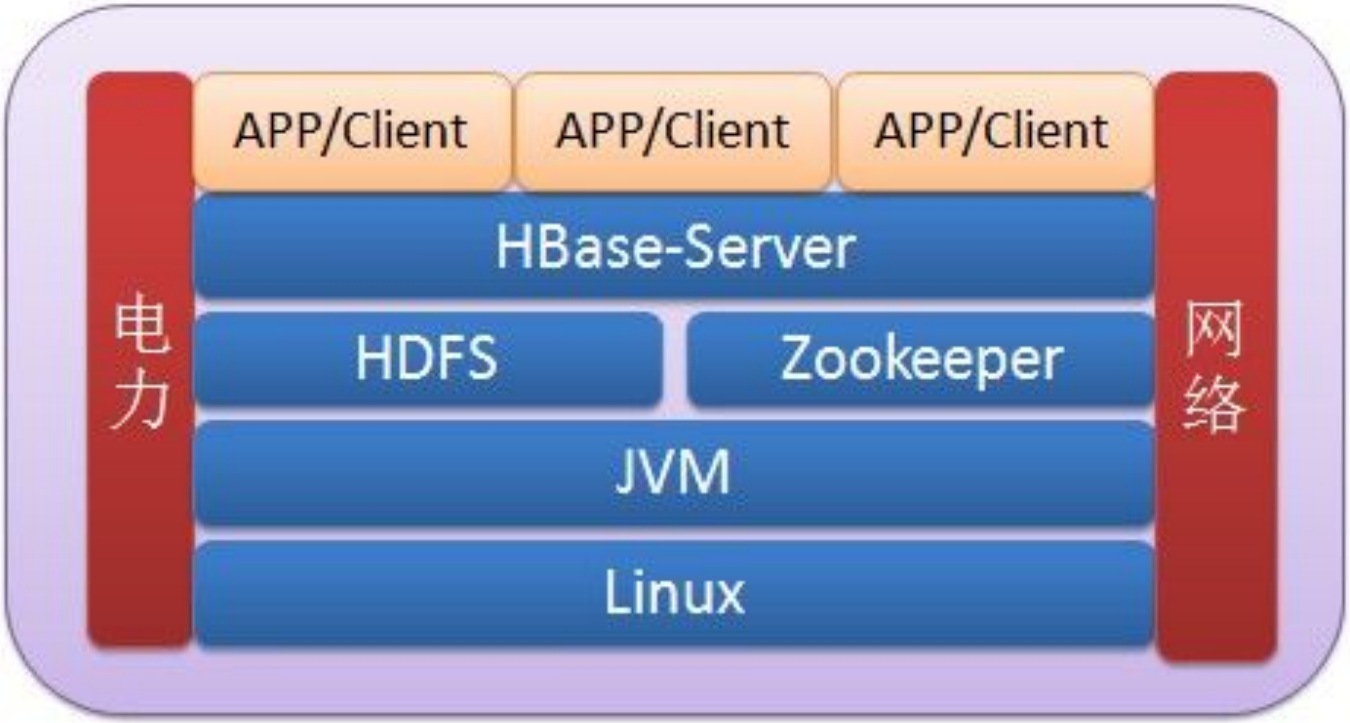
- MTTR（Mean Time To Recover）
- 影响因素
  - 集群规模
  - 系统压力
  - 故障范围
  - 故障缘由
- 目标
  - 多集群：1~5分钟， 大多数 < 2 分钟
  - 单集群：1~30分钟， 大多数 < 10分钟

高可用HBase---目标

- MTBF（Mean Time Between Failures）
- 影响因素
  - 升级
  - 变更
  - 迁移
  - 容量
  - 用户不规则活动
  - 软件不够健壮
- 目标
  - 变更行为的标准化、程序化
  - 增强软件健壮性

高可用HBase---运行环境

– 运行环境



高可用HBase---访问链路

- 电力
  - RPO 减少数据丢失量
  - Datanode
    - dfs.datanode.synconclose = true
  - Regionserver
    - 引入fsync (FSDataOutputStream #hsync())
    - 定期Flush

高可用HBase---访问链路

- 网络
  - 异常：快速感知
  - 中断：区域可用，快速恢复
    - 依赖部署避免跨机房

高可用HBase---访问链路

- Zookeeper
  - Zk层面
    - 提高 jute.maxbuffer
    - 降低 maxClientCnxns
  - HBase服务端与客户端争抢zk资源
    - 临时保护 iptables
    - TODO on ZK
      - » 请求Quota
      - » 服务隔离
  - Regionserver层面
    - 提高zookeeper.recovery.retry
    - 容忍zk不可用 (Replication)

高可用HBase---访问链路

- HDFS
  - Namenode
    - 提高 dfs.qjournal.start-segment.timeout.ms
    - 提高 dfs.qjournal.write-txns.timeout.ms
    - 屏蔽stale node
      - » dfs.namenode.avoid.read.stale.datanode = true
      - » dfs.namenode.avoid.write.stale.datanode = true
  - Datanode
    - 坏盘容忍 dfs.datanode.failed.volumes.tolerated
    - 提高 dfs.datanode.max.xcievers
  - Regionserver
    - 容忍HDFS不可用(Flush、Roll HLog)

高可用HBase---访问链路

HBase - MBTF

提高健壮，力求不跪

高可用HBase---访问链路

- HBase
  - 监视（WEBUI、监控、报警、脚本）
    - 系统、GC
    - 全局: Region大小、文件数目、完整性(hbck)
    - Memstore： blocking update
    - Hlog： .logs数目、.oldlogs数目、
    - Flush： delaying flush
    - Compaction： 文件选择的合理性(BulkLoad下建议使用ExploringCompaction)
    - RPC: Queue、Handler
    - Replication：积压Log

高可用HBase---访问链路

- HBase
  - 监视
    - 服务端Trace
      - » 跟踪满足匹配条件的请求在服务端的执行情况，包括来源、处理时间、资源开销等

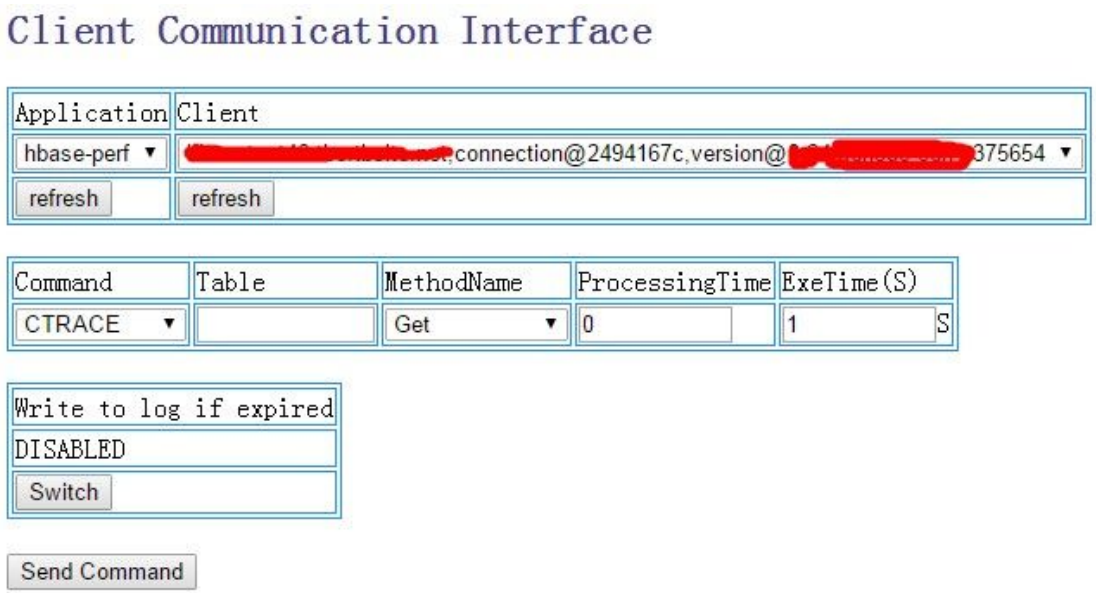
Call Tracker Current State 清屏

Enabled	Log Open	ProcessingTime(ms)	ResponseSize(byte)	Table	RegionEncodeName	MethodName	关键字过滤
false	false	1	0	null	null	null	null
Open	Open	1 重设	重设	重设	重设	重设	重设

id=739506684 Thu Feb 18 16:10:16 CST 2016 process_time=3 response_size=369 call_size=177
get(3249_2088902579836562,1441968951235.d98a90f9381b76e7c03203be8fbacbdb., {"timeRange":
[0,9223372036854775807],"totalColumns":1,"cacheBlocks":true,"families":{"c":["ALL"]},"maxVersions":1,"row":"3319_[redacted]"}), rpc version=1, client
version=29, methodsFingerPrint=1205343015 from [redacted]46404 ProfilingData:data_block_miss_cnt:3, bloom_block_hit_cnt:4,
data_block_hit_cnt:1, index_block_hit_cnt:4, server_process_time.ms:3, response_size.byte:369, server_queue_time.ms:0, total_block_read_time.ms:2, HFILE_NAME:
{883af67be98c4524b011e184892fd6eb[DATA Block offset=229300109/compressedSize=22030/decompressedSize=43075],52aee4da98714d51bb677d0934c0b5d9[DATA Block
offset=229317605/compressedSize=22020/decompressedSize=43075],dde5903fae5d4bdfb9b048c19858b725[DATA Block
offset=254083888/compressedSize=21238/decompressedSize=41360],}

高可用HBase---访问链路

- HBase
  - 监视
    - 全链路调试
      - » 服务端可以通过主动的方式获取客户端的配置，调用方式，运行开销，运行统计等信息，从而获得请求的全链路执行信息

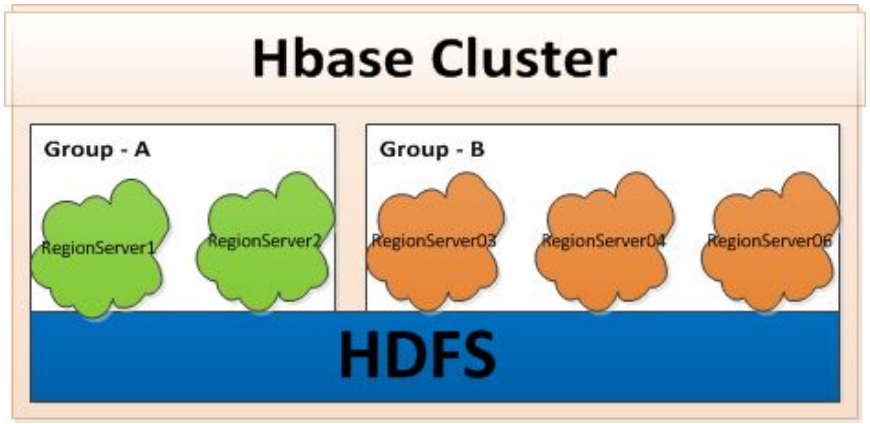


高可用HBase---访问链路

– HBase

– 控制

- 隔离-资源分组 (HBASE-6721)（多租户）



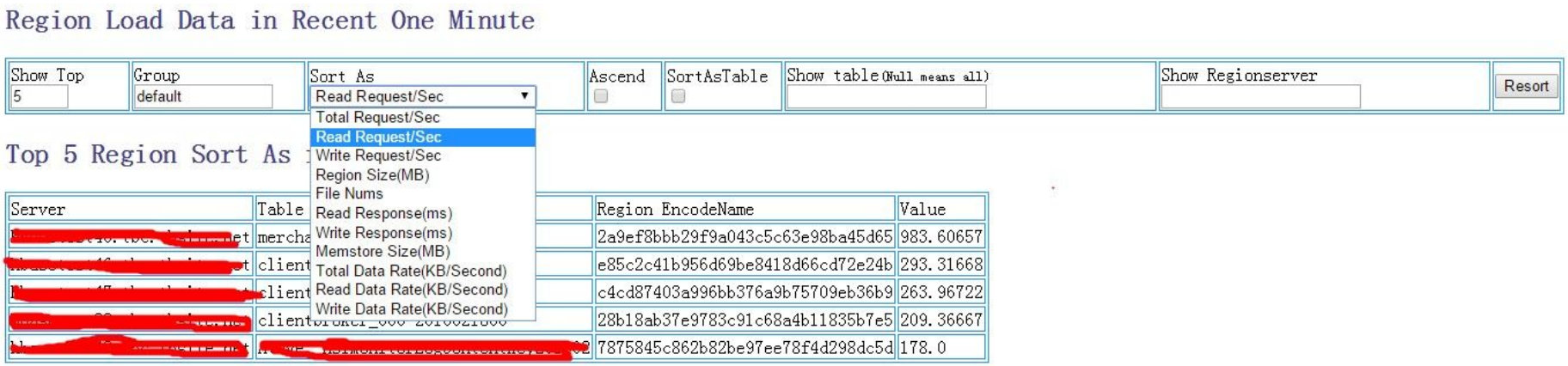
Group	ServerNum	TableNum	RegionNum	RequestsPerSecond	StoreFileSize	GroupAttributes
buffer	0	0	0	0	0 B	N/A
[redacted]	44	141	32,461	20,531	61.26 TB	BALANCE_SWITCH => 'true'
[redacted]	2	4	35	47	95.05 GB	N/A
[redacted]	24	40	4,634	278	8.19 TB	BALANCE_SWITCH => 'true'
[redacted]	30	10	3,202	211,770	13.07 TB	BALANCE_SWITCH => 'true'
Total:5	**100**	**195**	**40,332**	**232,626**	**82.62 TB**	

高可用HBase---访问链路

- HBase
  - 控制
    - 隔离
      - 关Balance + Move Region
      - Meta表打爆 (用户与系统隔离)
    - 拒绝访问
      - Iptables
      - ACL
      - 读写开关
      - 黑白名单机制，拒绝 HOST+ Method条件的RPC
    - 限制资源使用
      - 针对大Scan的资源保护
      - Compaction/Flush限速，控制线程数

高可用HBase---访问链路

- HBase
  - 热点问题
    - 预防
      - » Salted Table
      - » MD5
    - 发现
      - » Region Load 排序



高可用HBase---访问链路

- HBase
  - 热点问题
    - 发现
      - » 通过Trace发现行热点
    - 处理
      - » 分裂
      - » 隔离
      - » 业务改造

高可用HBase---访问链路

- HBase
  - 在线解决
    - 配置热调整
      - » 表： hbase.online.schema.update.enable true
      - » RS： update_all_config
      - » RS： update_config 'servername‘
    - 热补丁
      - » JSP + 反射
    - 滚动升级

高可用HBase---访问链路

- HBase
  - 业务迁移
    - 平滑切流
      - 地址虚拟化 (不直接使用zk1,zk2,zk3:2181:/hbase-znode)
        - VZNODE
        - 第三方地址服务
      - 查询强一致业务，需要停写开关
    - 不停服，移动数据
      - SNAPSHOT + Replication （推荐）
      - 建空表 -> 切流量 -> Bulk Load（适合迁移期间只写）
      - Distcp + HLog Replay (无snapshot依赖)

高可用HBase---访问链路

HBase - MTTR

争分夺秒，满血复活

高可用HBase---访问链路

- HBase
  - MTTR
    - 防止局部影响的连锁反应
    - 常规提速
      - » Distributed-Log-Replay（HBASE-7006）：先恢复写服务，再恢复数据，最后恢复读服务
      - » Meta-Log优先Split/Replay
      - » 设置合理的Recover Lease超时时间
      - » 持久化Region的最近Flush Sequence Id
      - » 支持脏读
      - » Assign Region提速 (Bulk Assign、并行ZK操作、提高RS中open-region线程数、减少对NN的重复访问)

高可用HBase---访问链路

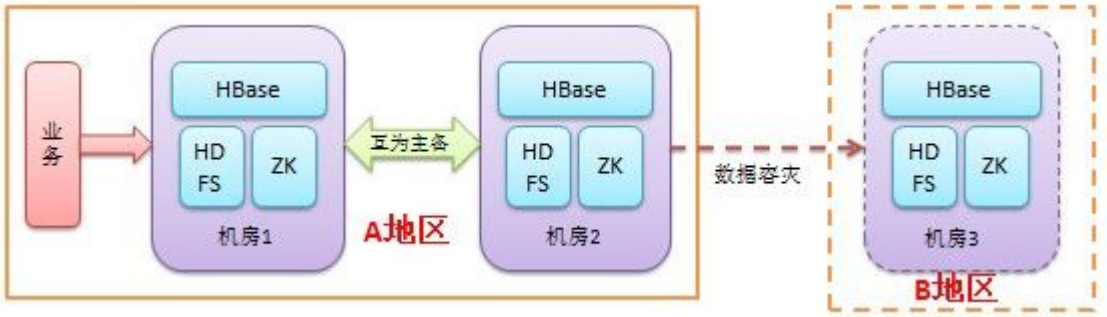
- HBase
  - MTTR
    - 非常规提速
      - » 设置恢复优先级
      - » 移除splitting-logs，灾后重建 （适合脏读业务）

高可用HBase---冗余切换

容灾，跑路

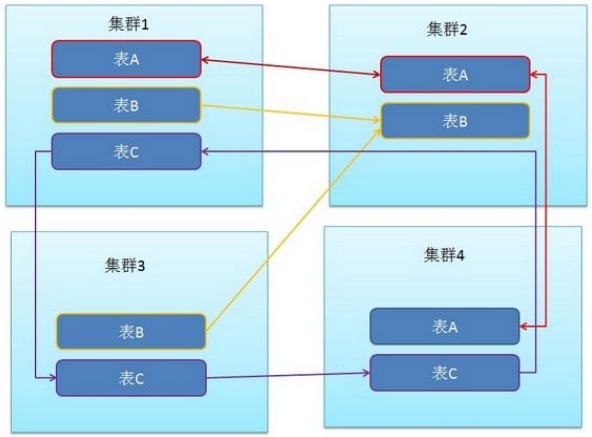
高可用HBase---冗余切换

- 多集群冗余
  - 使用HBase-Replication，构建数据复制链路
    - 同城服务容灾，异地数据容灾



高可用HBase---冗余切换

- 多集群冗余
  - 多集群链路下，支持数据的任意流动
  - 支持表复制只选择个别peer(s)（数据链路）
  - 循环链路下的数据去重
    - 无环 A<->B<->C √
    - 单向环 A->B->C->A √
    - 避免双向环 A<->B<->C<->A ×（或者进行去重优化）



高可用HBase---冗余切换

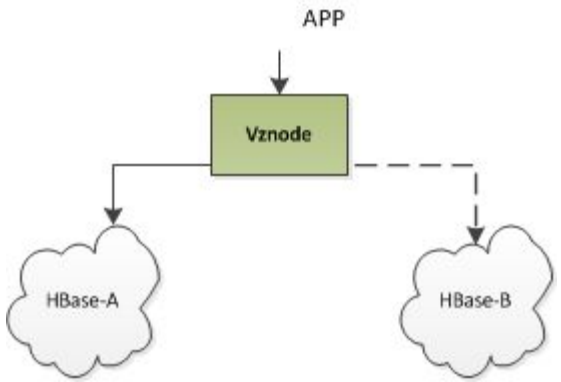
- 多集群冗余
  - 降低异步复制的弊端
    - 明确同步时间点

Replication Topology:

Master -> Slave	ReplSyncedTime	M-View Delay	LogQueue	RS-LEVEL-LOAD	M-REPL-STATUS
hbase-[redacted]-cell -> hbase-[redacted]-cell	2015-10-07 13:39:27 (1444196367751)	MAX: 18.00 ms AVG: 11.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-cell -> hbase-[redacted]-cloud	2015-10-07 13:39:25 (1444196365600)	MAX: 186.00 ms AVG: 70.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-cell -> [redacted]-P1	2015-10-07 13:39:27 (1444196367765)	MAX: 102.00 ms AVG: 8.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-cell -> [redacted]P2	2015-10-07 13:39:28 (1444196368543)	MAX: 0.00 ms AVG: 0.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-cell -> hbase-[redacted]-h5	2015-10-07 13:39:27 (1444196367716)	MAX: 199.00 ms AVG: 66.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-cell -> hbase-[redacted]-cell	2015-10-07 13:39:25 (1444196365067)	MAX: 0.00 ms AVG: 0.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-cell
hbase-[redacted]-h5 -> hbase-[redacted]cell	2015-10-07 13:39:33 (1444196373825)	MAX: 54.00 ms AVG: 20.00 ms	MAX: 0 AVG: 0	Detail	hbase-[redacted]-h5

高可用HBase---冗余切换

- 多集群切换
  - 不同集群的部署保持独立性
  - 平滑切流
    - 地址虚拟化 (客户端不直接使用服务端的ZK地址)
      - » VZNODE
      - » 第三方地址服务



    - 异步复制下，针对强一致业务，会先打开禁写开关，等到数据在主备完全同步后，再进行切换

高可用HBase---冗余切换

- 强一致与持续可用
  - 自动切换
    - 分布式数据库与单机数据库的故障区别
      - » 前者可以自我恢复
    - 可用性量化
    - 切换的容错性
  - 主备集群的强同步复制

Thank you! Join US!

邮箱: zjusch@163.com