

hbase上搭建广告实时数据处理平台

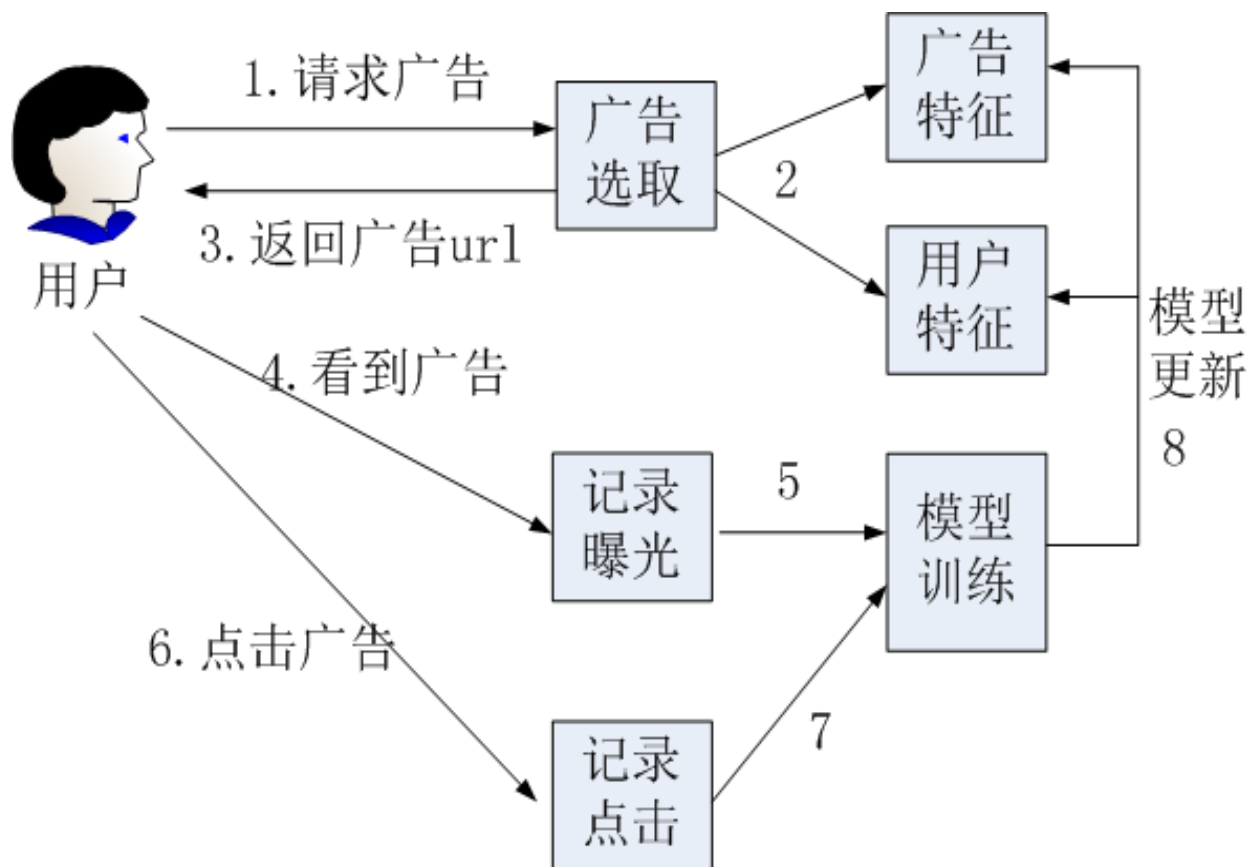
腾讯广点通WRAPENGINE项目

李锐

广点通

- 基于腾讯社交体系的效果广告平台
- 流量
 - QQ
 - QQ空间
 - 微信
 - 移动联盟
 - 等等
- 智能精准定向
 - 用户特征
- <http://e.qq.com>

背景：广告数据流



背景

- 实时storm+离线mapreduce计算，多维度分析
- 难点：如何把大量的信息从广告选取阶段传递至广告点击的模型训练？
 - Proto buffer
 - 实时日志关联

技术选型(1)

方案一：

- 广告请求时把信息编码到广告url里面。
- 曝光和点击的时候再上报。
- 缺点：
 - 广告url过长，“偷”用户流量，移动端优质APP难以接受。
 - 不能放入过多的信息。

技术选型(2)

方案二:

- 在storm里面实现流式关联
- 缺点:
 - storm数据无法落地, 需要有额外的存储方案解决storm重启的问题
 - 20%的曝光在广告请求20min以上, 内存无法缓存如此长时间的请求数据。

技术选型(3)

方案三：

- 在storm里面实现流式关联
- 中间数据写到磁盘排序文件，比如sstable
- 追加日志
- 排序文件的合并
- 读写数据缓存
- 过期数据淘汰
- 缺点：
 - 开发周期过长，跟hbase功能重合

技术选型(4)

方案四：

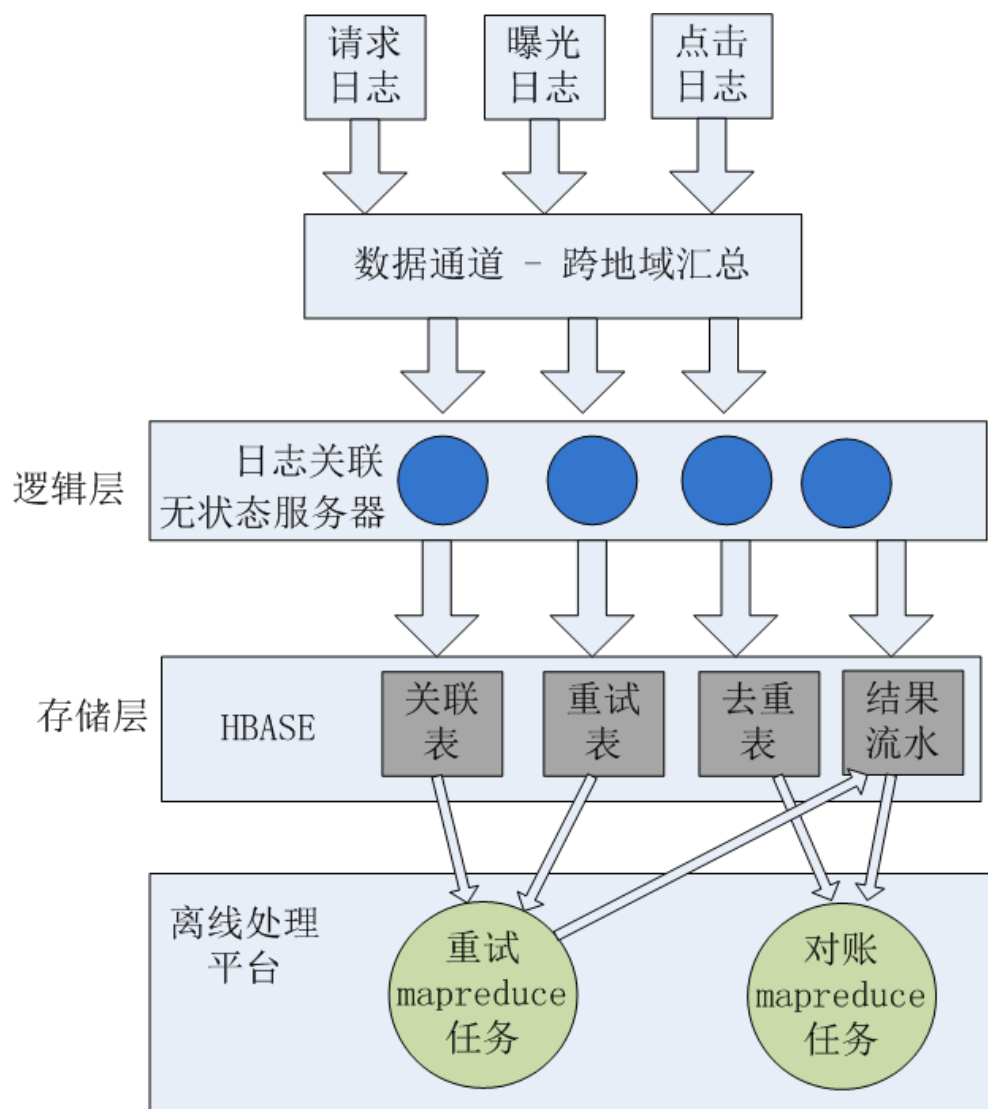
- 广告请求信息写入hbase
- 广告曝光和点击时查询hbase，做日志关联
- 优点：
 - 大部分的查询能命中cache

技术选型(5)

为何不用：

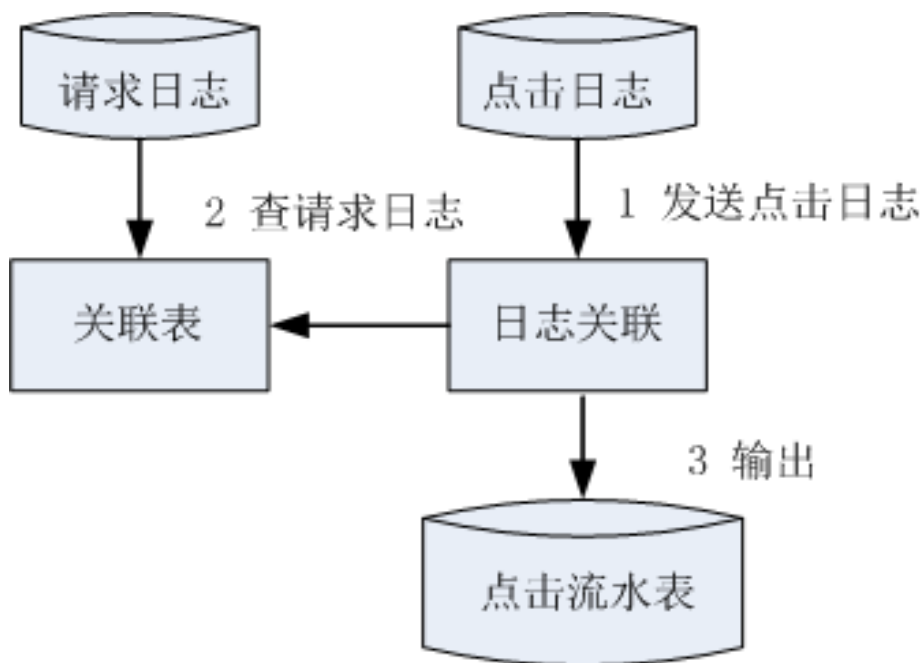
- Impala
- Spark
- Cassandra

技术方案 - 整体框架



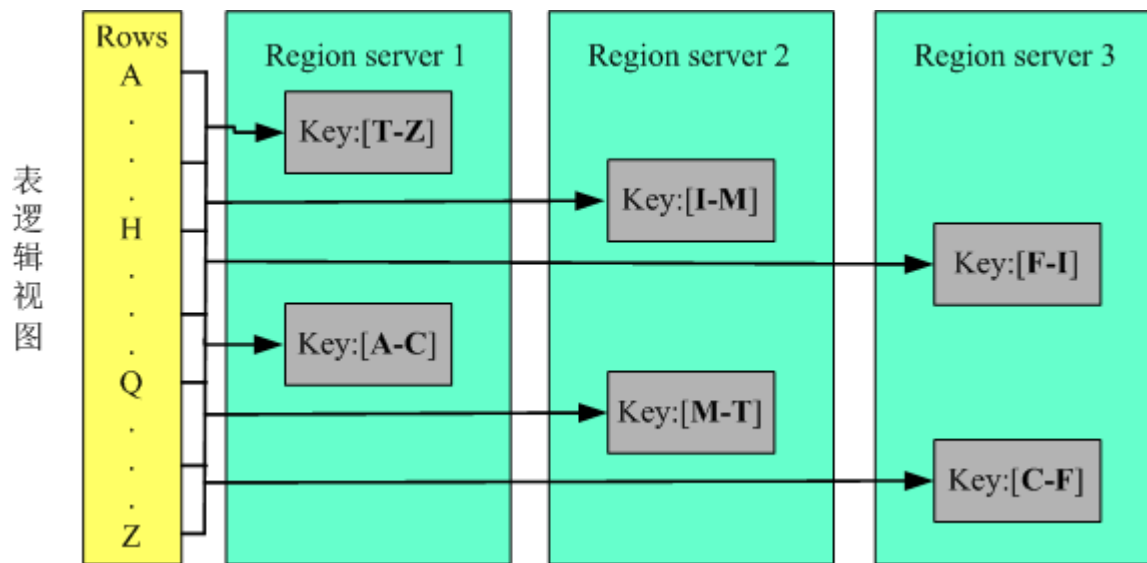
技术方案 - 整体框架

- 每次广告请求分配唯一id
- 关联表：
 - 收集请求日志，按id为key写到hbase关联表
 - 曝光和点击日志，按id查关联表



技术方案 - key的设计

- 避免热点
- Key前缀加分桶打散数据
- 支持按时间段小批量构建模型
- Key中间放时间戳



分桶ID

时间戳

日志ID

065	201408121450	agx13tdctggz
-----	--------------	--------------

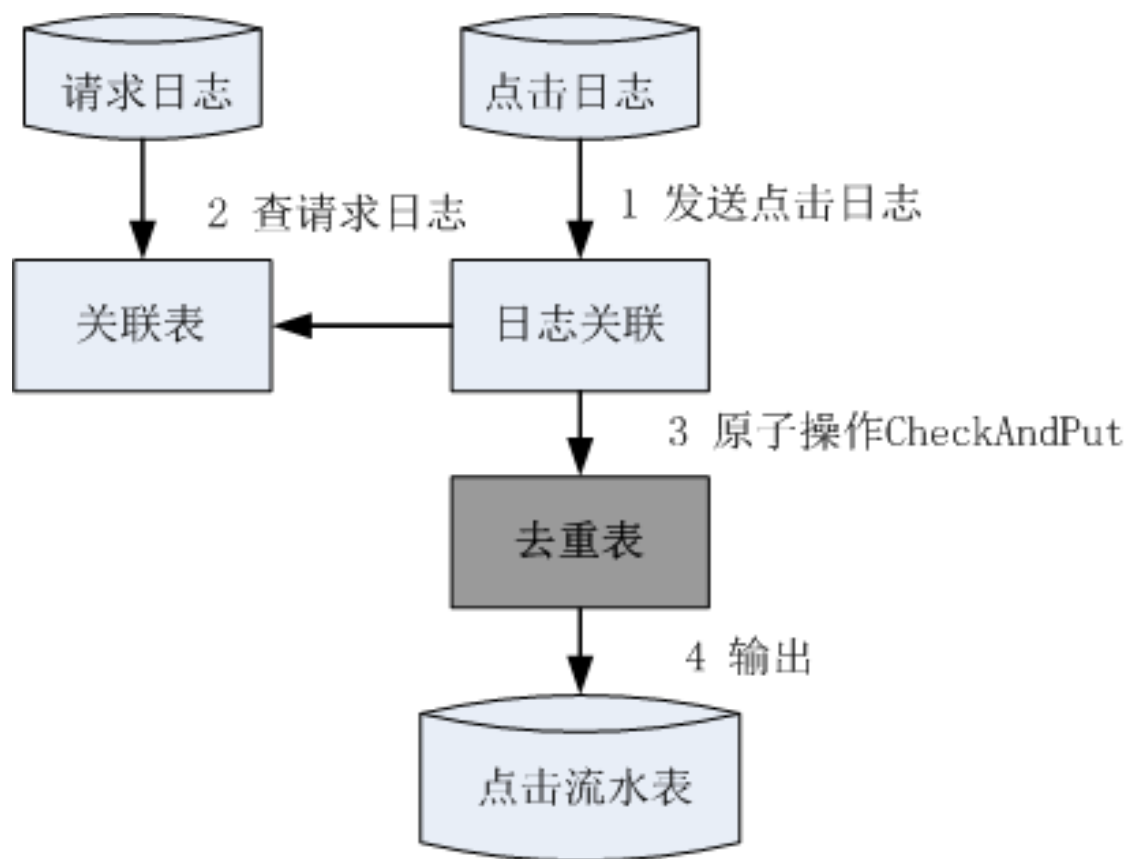
避免数据倾斜

扫描数据

查找日志

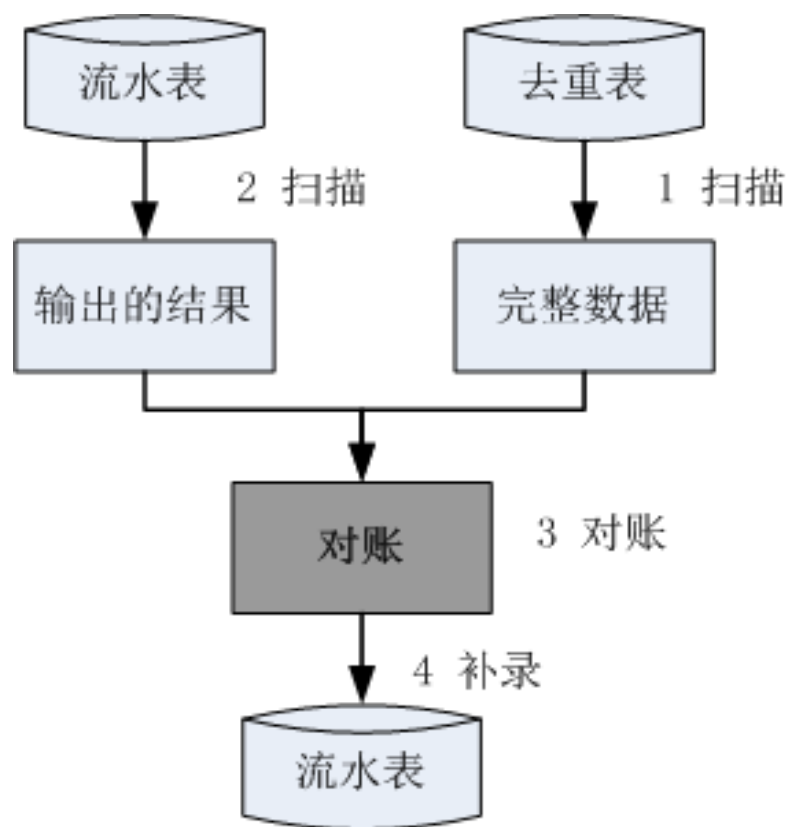
技术方案 - EXACT ONCE

- 如何保证点击等计费信息不重不丢？
 - 上游发送数据失败时，进行重试。
 - logjoin输出到下游的时候通过hbase进行去重



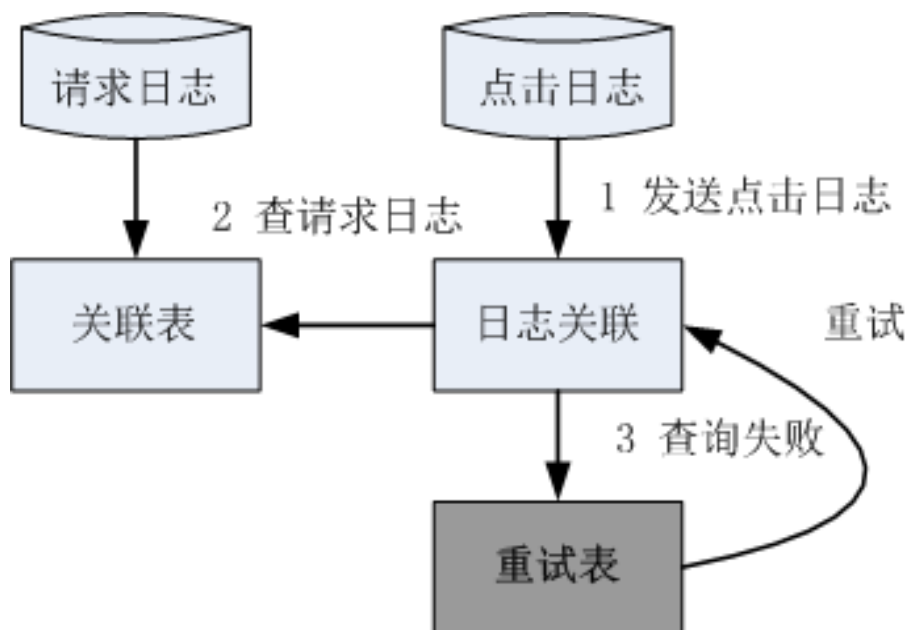
技术方案 - EXACT ONCE

- 后台离线对账。



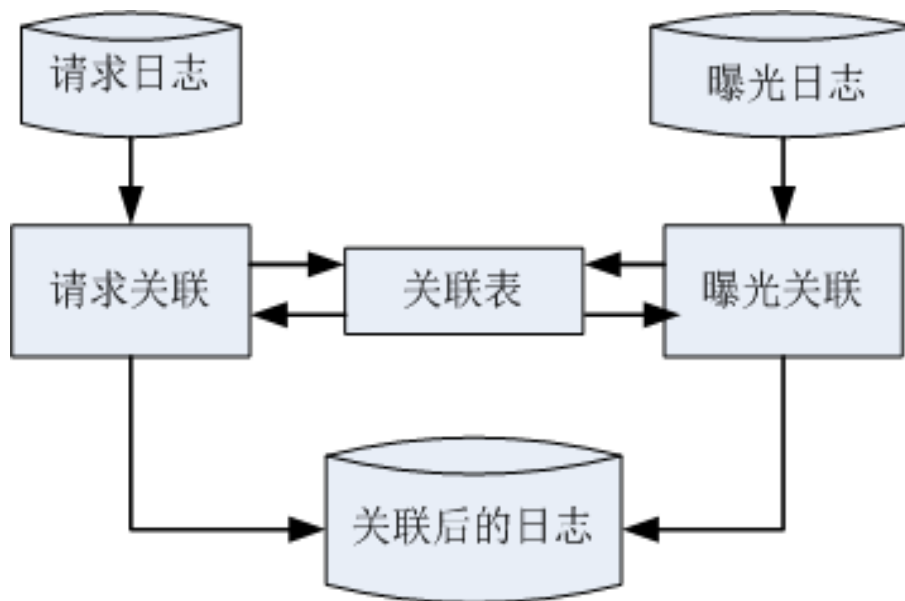
技术方案 - 处理数据乱序

- 由于跨地域传输，机器偶尔卡住等原因，有时曝光和点击比请求日志早到。
- 当请求日志延迟时，缓存点击日志到磁盘，然后离线重试。



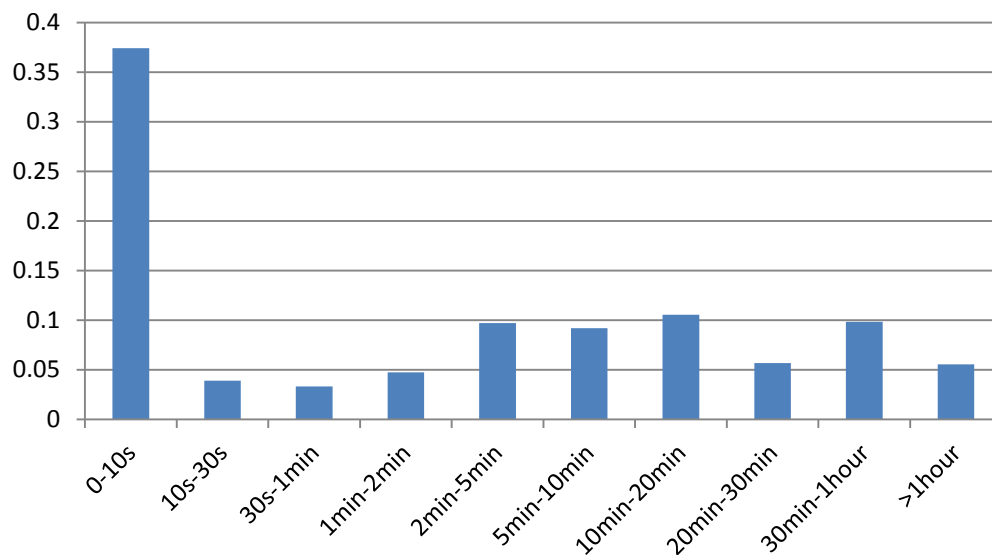
技术方案 - 处理数据乱序

- 约1/10的请求晚于曝光日志
- 让请求、曝光日志晚到的触发关联操作
- 通过checkandput原子操作保证数据不重复



性能优化 - 系统特点

- 访问特性:
- 写hbase: 每天300亿+
- 读hbase: 每天200亿+, 而且都是随机读!
- 但是:
 - 只有100多亿读操作是预期读到数据的
 - 大部分数据写入到读取的时间延迟很小



性能优化 - 内存cache

- 增大写cache, cache命中率95%
- 依靠Bloom filter过滤掉大部分读磁盘需求
- 部分访问频繁的小表配置为全内存表

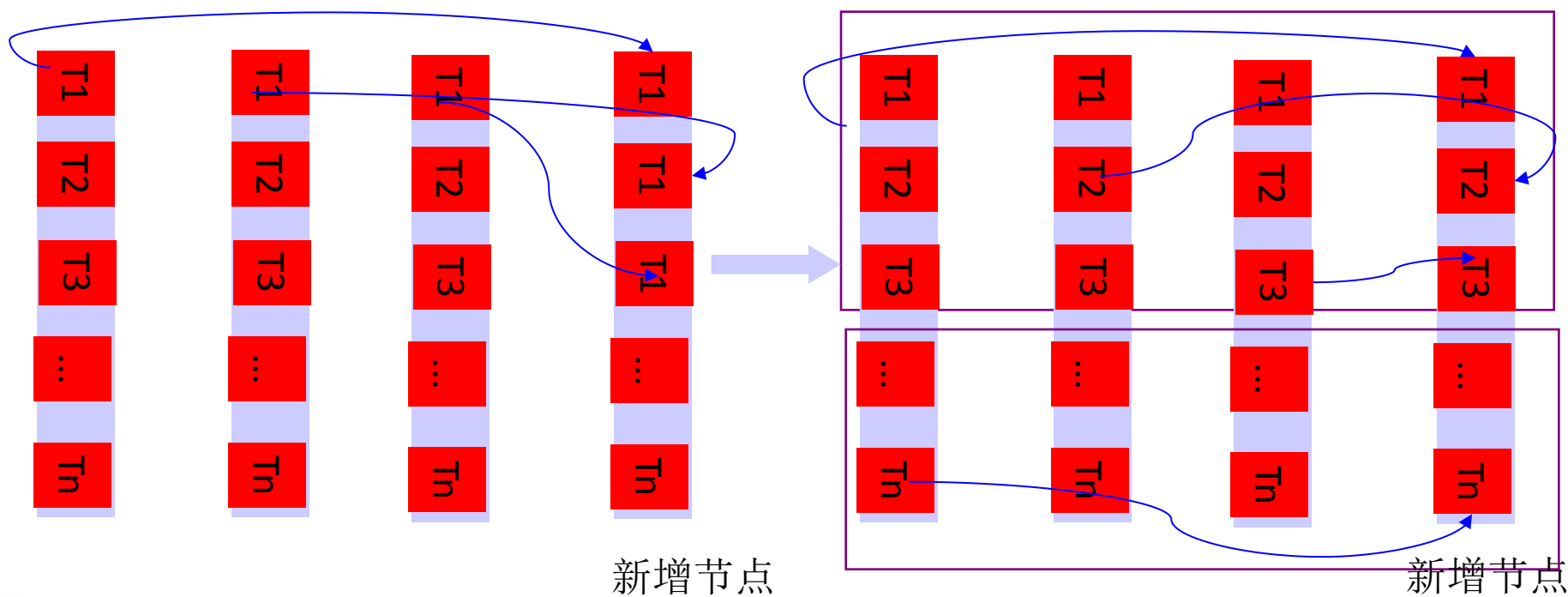
性能优化 - 优化负载均衡算法

■ 问题描述

对于时间序列的表，在新增节点或是节点故障重新加入时，热点Region会分布到一个Regionserver上。

■ 优化点

针对时间序列的balancer 算法优化。按时间分段，随机选取



性能优化 - 避免单点故障导致堵塞

■ 问题描述

为了更好的性能，我们使用批量操作读写hbase。

在同步的接口中，会启动与服务器个数相同的线程池，等待服务器的返回结果。

随着服务器数目增加，客户端线程数目增加非常多。

当有一个服务器处理缓慢时，会拖慢所有客户端的所有线程。

- 目前的解决办法：客户端把所有批处理操作按照Region归类分发。
- 更好的解决办法：采用异步的客户端。

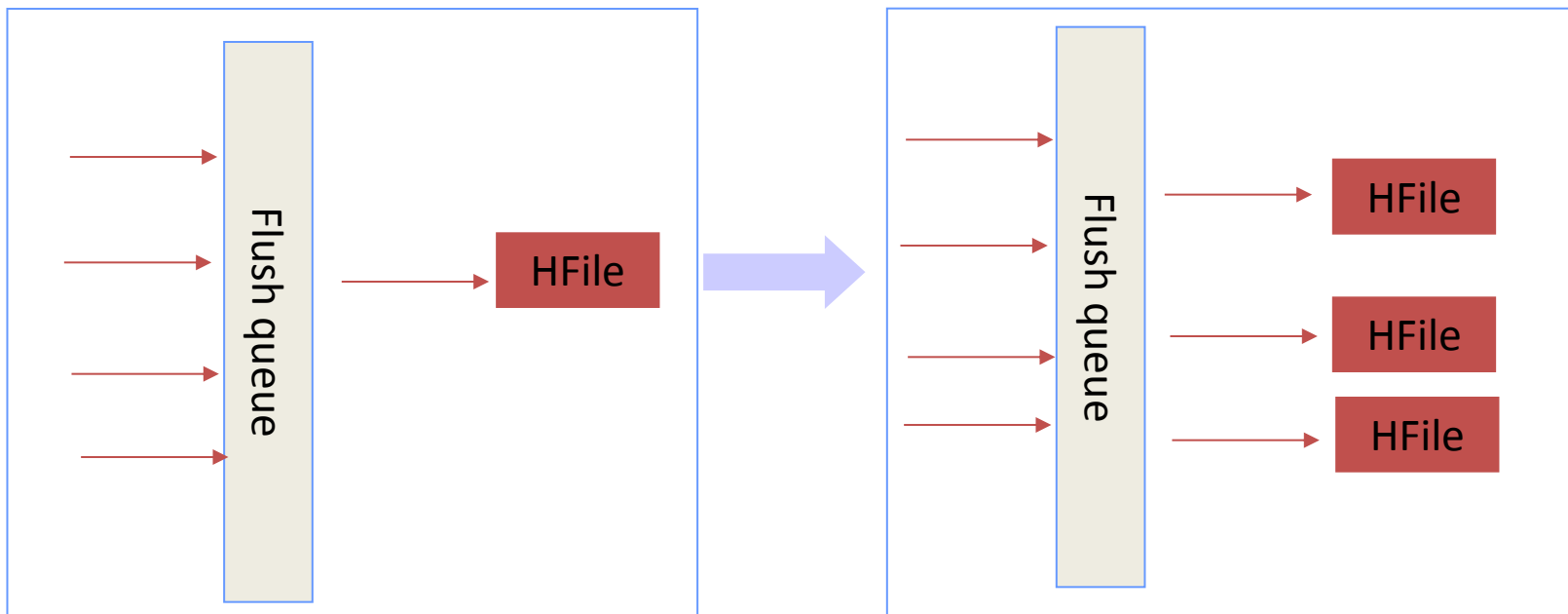
性能优化 - 避免rpc堵塞

■ 问题描述

在put请求过大的情况下，服务器的flushsize队列积压，导致RPC出现阻塞

■ 优化点

使用多线程进行flush



性能优化 - 减少网络开销

■ 问题描述

网络带宽消耗过大

■ 优化点

调整Memstore的大小以及Hlog等参数减少flush小文件的个数，减少compaction的需求

关闭大表的major compaction

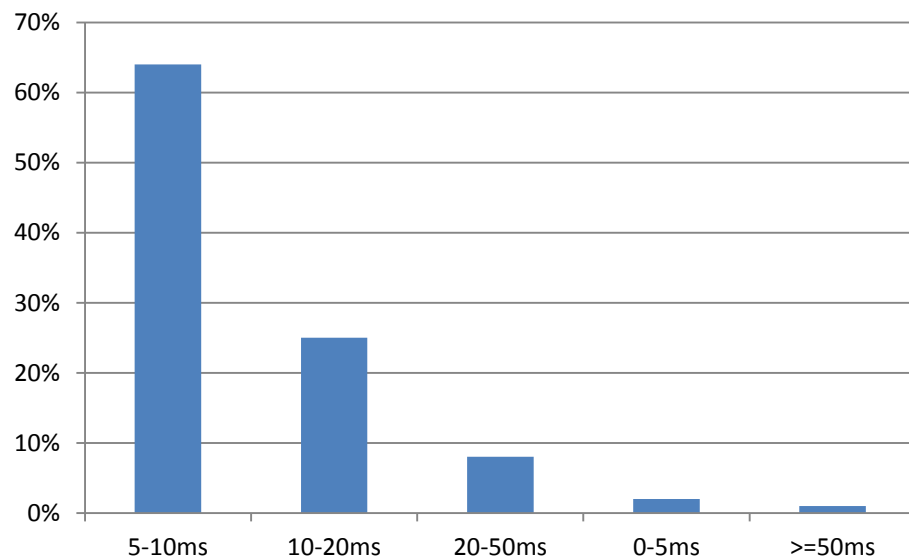
调整compaction的阈值参数和线程数目

采用针对单个region的compaction进行操作，通过外部配置，尽量在流量低峰进行compaction操作

应用情况

- 日志200+亿/天
- Hbase集群 200+台

Hbase访问时延分布



Thanks!

李锐
腾讯广点通
Phone: 13810095271
Email: rli@tencent.com

