

云数据库 HBase 版

使用案例和参考资料

使用案例和参考资料

案例集合

HBase X-Pack适合场景，下面收集和汇总了一些案例集合和参考资料供大家参考

HBase典型场景

再谈全局网HBase八大应用场景<https://yq.aliyun.com/articles/558255>

存储场景

亿方云(文档存储) <https://yq.aliyun.com/articles/346482>

车联网场景

车纷享(车厘子) <https://yq.aliyun.com/articles/563203>

HBase在新能源汽车监控系统中的应用<https://yq.aliyun.com/articles/673930>

金融场景

HBase在大搜车金融业务中的应用实践<https://yq.aliyun.com/articles/346456>

自动推荐场景

兑吧：从自建HBase迁移到阿里云HBase实战经验<https://yq.aliyun.com/articles/601483>

人工智能场景

HBase 在人工智能场景的使用<https://yq.aliyun.com/articles/672245>

离线分析场景

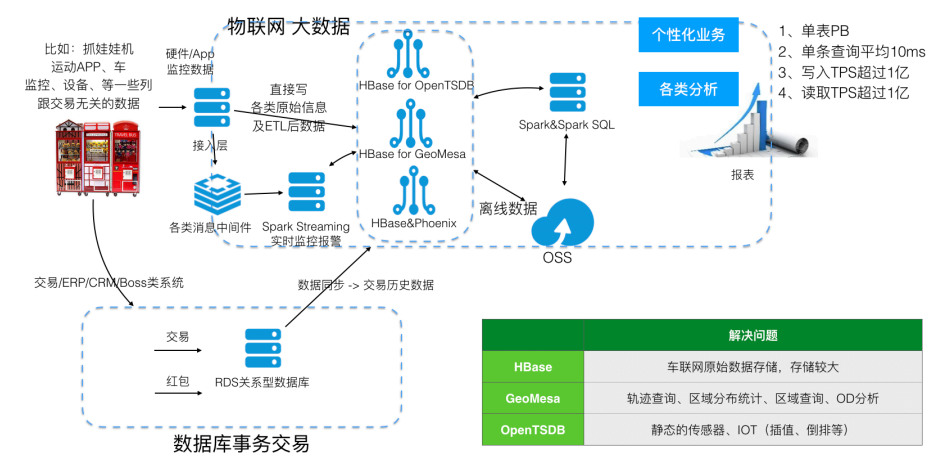
有赞 Hive 迁移到 Spark SQL实战<https://yq.aliyun.com/articles/686004>

社区技术资料汇总

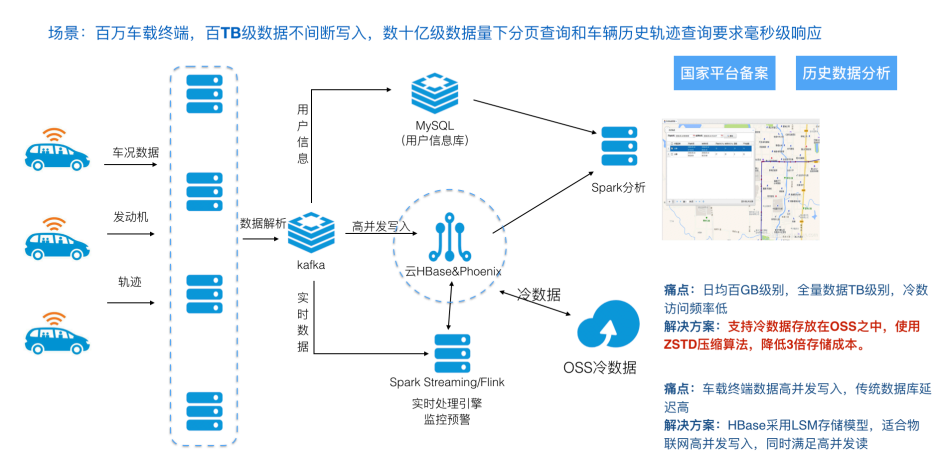
<https://yq.aliyun.com/teams/382>

物联网&车联网行业

物联网综合解决方案



车联网案例：使用HBase作为海量数据库存储



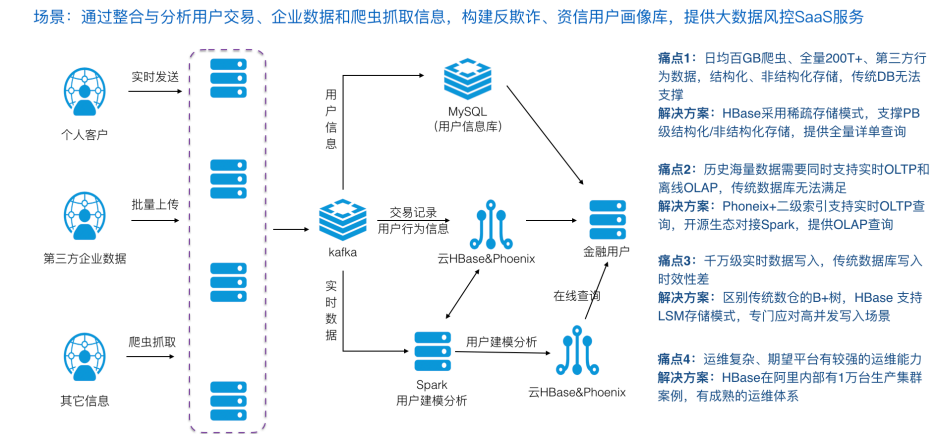
金融行业

金融行业：包括银行、保险、基金、P2P、大数据金融等公司，抽象目前HBase在金融行业的应用如下：

某保险公司 数据平台



某大数据风控平台



车联网 - 基于轨迹的风控

参考 HBase在大搜车金融业务中的应用实践



GeoHash, Left(GeoHash,6)聚合

痛点：运维复杂、期望平台有较强的运维能力
解决方案：HBase在阿里内部有1万台生产集群
案例：有成熟的运维体系

金融公司 - 历史订单在线查询

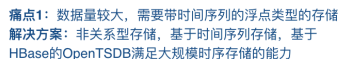


痛点2: 历史订单数据量较多, 存储成本较贵
解决方案: 云HBase基于共享存储, 让全局副本数从基于云盘的9副本降低为3副本, 成本降低60%+; HBase本身最高10倍压缩比例降低成本

痛点3: 在线查询需要预防灾难发生，如果机房故障可能业务中断

解决方案: 云HBase提供同城主备或者双活的方式，保障机房级别容灾

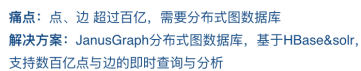
金融时序数据 - K线、分析



痛点2: 数据量较大，需要带时间序列的浮点类型的存储

解决方案: 如果业务类型有特殊的需求，可以参考 OpenTSDB 的思想，自由设计基于存储模型。比如 上证的代号

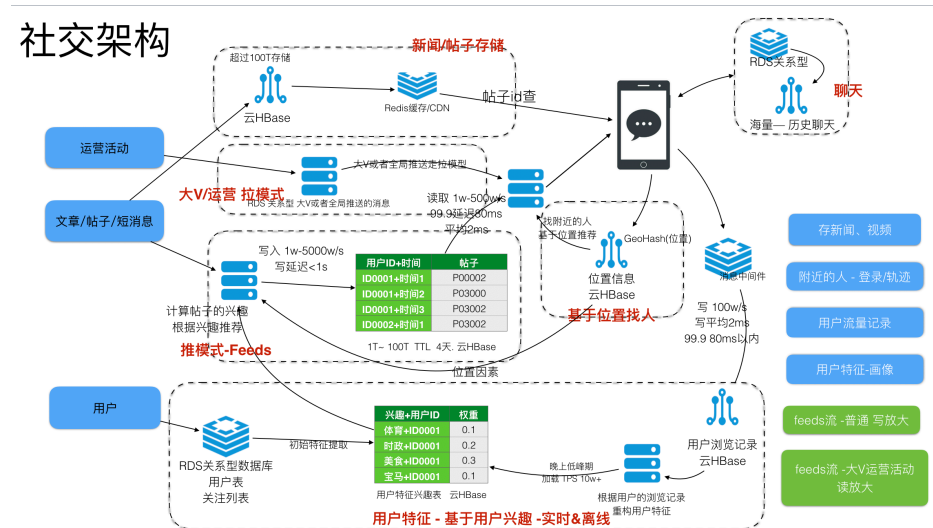
金融欺诈



简述

在社交领域之中，有很多数据的存储，使用数据的形态也多种多态。目前主流的社交都会带推荐模块的，也就是会把新闻/消息/短文 推动给合适的人，也会考虑地理位置的信息。本文简述社交类的基本架构，考虑的一些因素。细节点会在不同的社交类场景之中，有一定的优化，但是大同小异，基本架构如下：

社交领域的基本架构及流程流



上图是社交领域的基本架构，分为以下基本部分

客户登录及发帖及看帖

- 客户登录后，系统会记录位置信息，位置信息 GeoHash处理后，可以存在 云HBase 之中
- 客户可以 发帖子，此块帖子可以写到 云HBase 之中，发完之后，也可以立即写到推荐之中，自己可以看到自己发的帖子
- 客户浏览帖子，系统记录 客户看过的帖子的信息，把浏览记录 写到 云HBase 之中，以备后续 分析 反馈给 用户特性 - 用户画像 模块

会有 帖子/新闻 的表，量也比较大，一般在1T-1P左右。

用户特性 - 用户画像

- 用户注册后，会选择感兴趣的特征，此块数据会形成最初的用户特性
- 用户浏览信息后，会有浏览的历史记录，会写到 云HBase之中
- 晚上会启动 spark 或者 用户写的code 分析用户的历史记录，修正用户的标签画像

最终会形成一张 用户特性 - 用户画像表

推荐及Feeds流

一般有两个模型，分表为 推模型，拉模型，最主要是以帖子为维度还是以 查询的 人为维度。推模型 是 写放大的，拉模型是读放大的。HBase基于LSM，比较适合推模型

- 当帖子或者文章产生后，会写一份到 云HBase 存储上
- 再同时会计算，帖子的属性，分析帖子的归类，提取 特征值
- 根据 帖子的特征值与 用户特性 - 用户画像，把匹配的用户写入 到 推送的表之中。此块 根据不同的业务，可能涉及的逻辑比较复杂，比如加入 位置的因素、 权重、好友关注的列表、把不活跃的客户剔除。

会形成一张 帖子推荐表，数据量比较大，大约1T到100T左右。会有过期时间，一般在3天-4天左右。

对于大V或者写运营普发的信息，可以采取拉模式，可以显著减少写放大。一般实际的业务是 推模型 与 拉模型 混合使用

帖子新闻查询

- 用户打开APP，可以查询最近推荐的信息
 - 对于 大V 或者 运营推送的信息，可以单独链路查询
 - 查询 帖子推荐表，查询出所属的帖子ID后，再查询实际的帖子信息，这部分一般在 redis 或者 CDN之中有缓存，没有命中，可以查询到 业务库 云HBase 之中

基于位置的推荐人

- 可以基于存的客户位置的信息，推荐边上 兴趣相投的人
 - 基于地理信息，查询边上所有的客户，再查询画像表，找个兴趣相投的人，再推荐