从Storm到Flink，有赞五年实时计算效率提升实践

AI前线 2019-03-07 17:37:16

策划编辑 | Tina

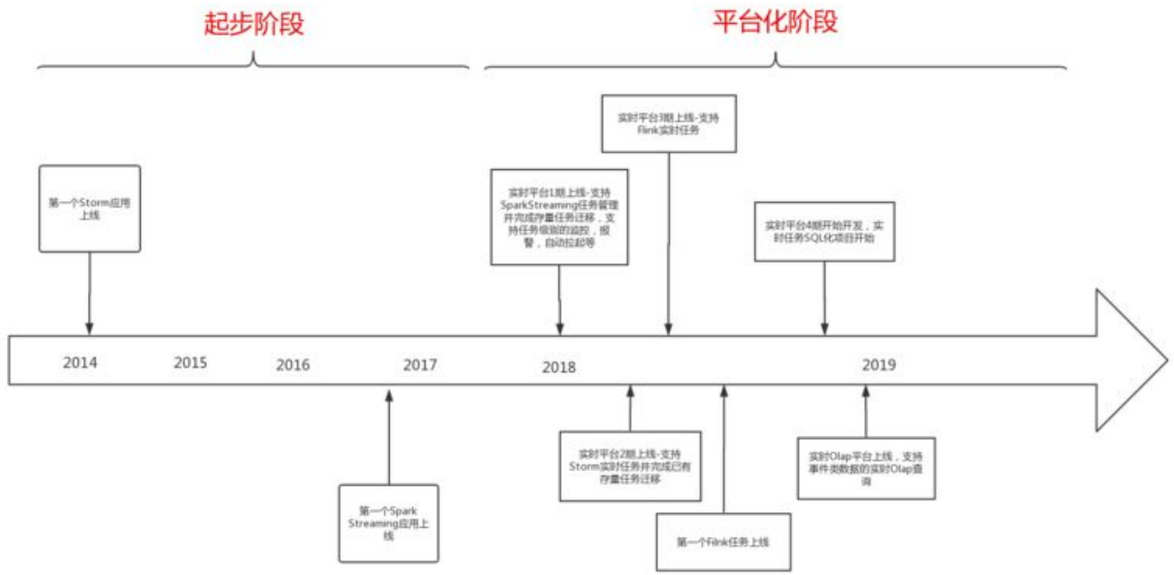
作者 | 贺飞

编辑 | Vincent

AI 前线导读： 有赞是一个商家服务公司，提供全行业全场景的电商解决方案。在有赞，大量的业务场景依赖对实时数据的处理，作为一类基础技术组件，服务着有赞内部几十个业务产品，几百个实时计算任务，其中包括交易数据大屏，商品实时统计分析，日志平台，调用链，风控等多个业务场景，本文将介绍有赞实时计算当前的发展历程和当前的实时计算技术架构。 **更多优质内容请关注微信公众号“AI 前线” （ID：ai-front)**

实时计算在有赞发展

从技术栈的角度，我们的选择和大多数互联网公司一致，从早期的 Storm，到 JStorm，Spark Streaming 和最近兴起的 Flink。从发展阶段来说，主要经历了两个阶段，起步阶段和平台化阶段；下面将按照下图中的时间线，介绍实时计算在有赞的发展历程。



2.1 起步阶段

这里的的起步阶段的基本特征是，缺少整体的实时计算规划，缺乏平台化任务管理，监控，报警工具，用户提交任务直接通过登录 AG 服务器使用命令行命令提交任务到线上集群，很难满足用户对可用性的要求。 但是，在起步阶段里积累了内部大量的实时计算场景。

2.1.1 Storm 登场

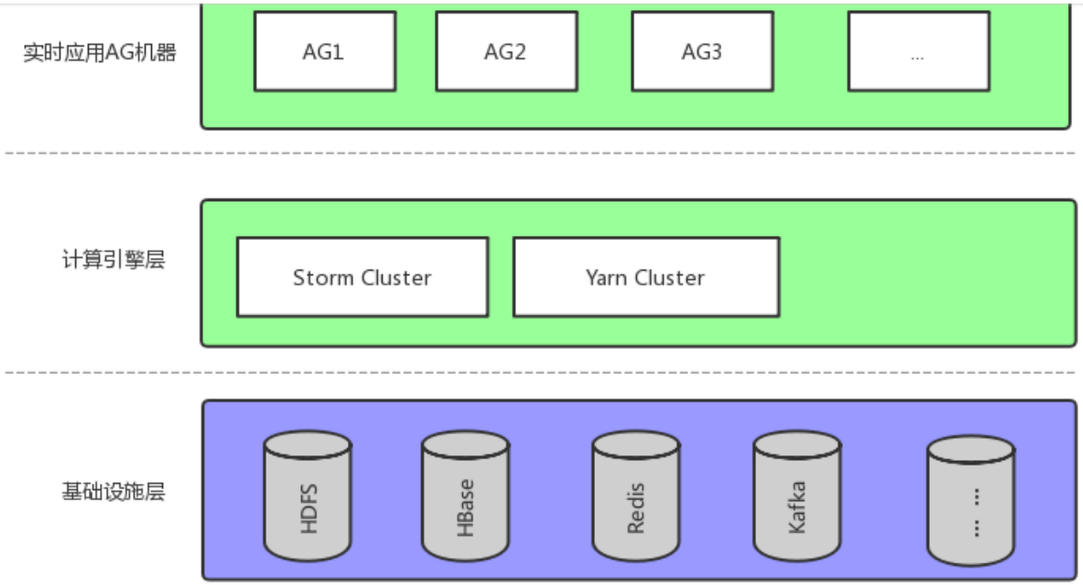
2014 年初，第一个 Storm 应用在有赞内部开始使用，最初的场景是把实时事件的统计从业务逻辑中解耦出来，Storm 应用通过监听 MySQL 的 binlog 更新事件做实时计算，然后将结果更新到 MySQL 或者 Redis 缓存上，供在线系统使用。类似的场景得到了业

早期，用户通过登录一组线上环境的 AG 服务器，通过 Storm 的客户端向 Storm 集群做提交任务等操作， 这样在 2 年多的时间里，Storm 组件积累了近百个实时应用。Storm 也同样暴露出很多问题，主要体现在系统吞吐上，对吞吐量巨大，但是对延迟不敏感的场景，显得力不从心。

2.1.2 引入 Spark Streaming

2016 年末，随着 Spark 技术栈的日益成熟，又因为 Storm 引擎本身在吞吐 / 性能上跟 Spark Streaming 技术栈相比有明显劣势，所以从那时候开始，部分业务团队开始尝试新的流式计算引擎。 因为有赞离线计算有大量 Spark 任务的使用经验，Spark Streaming 很自然的成为了第一选择，随着前期业务日志系统和埋点日志系统的实时应用的接入，大量业务方也开始逐渐接入。 同 Storm 一样，业务方完成实时计算应任务开发后，通过一组 AG 服务器，使用 Spark 客户端，向大数据 Yarn 集群提交任务。

初步阶段持续的时间比较长，差不多在 2017 年年末，有赞实时计算的部署情况如下图所示：



2.1.3 小结

这种架构在业务量少的情况下问题不大，但是随着应用方任务数目的增加，暴露出一些运维上的问题，主要在以下几个方面：

缺少业务管理机制。大数据团队平台组，作为集群管理者，很难了解当前集群上运行着的实时任务的业务归属关系，也就导致在集群出现可用性问题或者集群要做变更升级时，无法高效通知业务方做处理，沟通成本很高；

Storm 和 Spark Streaming 的监控报警，是各自实现的，处于工具化的阶段，很多业务方，为了可用性，会定制自己的监控报警工具，导致很多重复造轮，影响开发效率；

和 Spark Streaming 任务运行在同一组 Yarn 资源上，凌晨离线任务高峰时，虽然 Yarn 层有做 CapacityScheduler 的 Queue 隔离，但是 HDFS 层公用物理机，难免网卡和磁盘 IO 层面会相互影响，导致凌晨时间段实时任务会有大量延迟；

缺少灵活的资源调度。用户通过 AG 服务器启动实时任务，任务所使用的集群资源，也在启动脚本中指定。这种方式在系统可用性上存在很大弊端，当实时计算所在的 Yarn 资源池出现故障时，很难做实时任务的集群间切换。

总的来说就是缺少一个统一的实时计算平台，来管理实时计算的方方面面。

2.2 平台化阶段

2.2.1 构建实时计算平台

接上一节，面对上面提到的这四个问题，对实时计算平台的初步需求如下：

业务管理功能。主要是记录实时应用的相关信息，并且和业务的接口人做好关联；

提供任务级别的监控，任务故障自动拉起，用户自定义基于延迟 / 吞吐等指标的报警，流量趋势大盘等功能；

做好集群规划，为实时应用构建独立的计算 Yarn 集群，避免离线任务和实时任务互相影响；

提供任务零花的切换计算集群，保证在集群故障时可以方便迁移任务到其他集群暂避。

所以在 18 年初，我们立项开始做实时平台第一期，作为尝试起初我们仅仅完成对 Spark

的迁移。 试运行 2 个月后，明显感觉到对业务的掌控力变强。随后便开始了对 Storm 任务的支持，并迁移了所有的 Storm 实时计算任务. AG 服务器全部下线，业务方再也不需要登录服务器做任务提交。

2018 年中，有赞线上运行着 Storm，Spark Streaming 两种计算引擎的实时任务，可以满足大部分业务需求，但是，两种引擎本身也各自存在着问题。 Storm 本身存在着吞吐能力的限制。和 Spark Streaming 对比，选择似乎更难一些。我们主要从以下几个角度考虑：

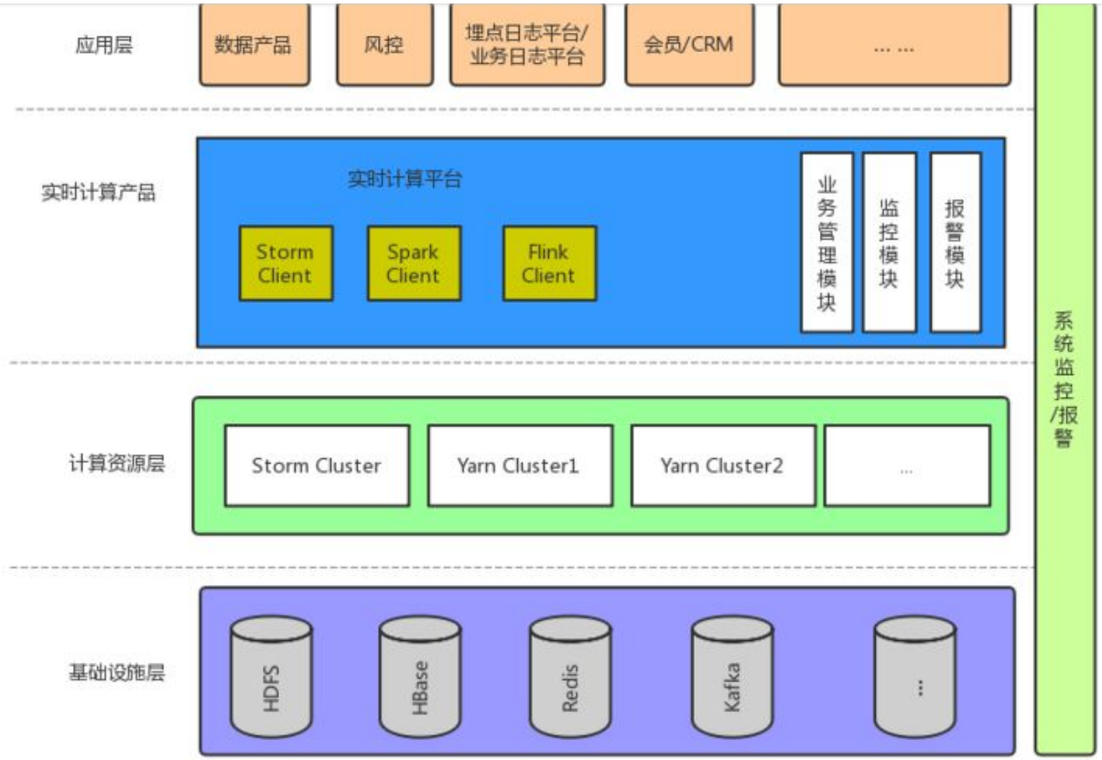
延迟， Flink 胜出，Spark Streaming 本质上还是以为微批次计算框架，处理延迟一般跟 Batch Interval 一致，一般在秒级别，在有赞的重吞吐场景下，一般 batch 的大小在 15 秒左右；

吞吐， 经过实际测试，相同条件下，Flink 的吞吐会略低于 Spark Streaming，但是相差无几对状态的存储支持， Flink 在这方面完胜，对于数据量较大的状态数据，Flink 可以选择直接存储计算节点本地内存或是 RocksDB，充分利用物理资源；

对 SQL 的支持，对当时两种框架的最新稳定版本的 SQL 功能做了调研，结果发现在对 SQL 的支持度上，Flink 也具有较大优势，主要体现在支持更多的语法；

API 灵活性， Flink 的实时计算 API 会更加友好。

出于以上几点原因，有赞开始在实时平台中增加了对 Flink 引擎的支持。在完成 Flink 引擎的集成后，有赞实时计算的部署情况如下图所示：



2.2.2 新的挑战

以上完成之后，基本上就可以提供稳定 / 可靠的实时计算服务；随之，业务方开发效率的问题开始显得突出。用户一般的接入流程包含以下几个步骤：

熟悉具体实时计算框架的 SDK 使用，第一次需要半天左右；

申请实时任务上下游资源，如消息队列，Redis/MySQL/HBase 等在线资源，一般几个小时；

对于复杂的实时开发任务，实时任务代码质量很难保证，平台组很难为每个业务方做代码 review, 所以经常会有使用不当的应用在测试环境小流量测试正常后，发布到线上，引起各种各样的问题。

整个算下来，整个流程至少需要 2~3 天，实时应用接入效率逐渐成了眼前最棘手的问题。 对于这个问题。在做了很多调研工作后，最终确定了两个实时计算的方向：

实时任务 SQL 化；

对于通用的实时数据分析场景，引入其他技术栈, 覆盖简单场景。

2.2.2.1 实时任务 SQL 化

实时任务 SQL 化可以大大简化业务的开发成本，缩短实时任务的上线周期。 在有赞，实时任务 SQL 化 基于 Flink 引擎，目前正在构建中，我们目前的规划是首先完成对以下功能的支持：

基于 Kafka 流的流到流的实时任务开发

基于 HBaseSink 的流到存储的 SQL 任务开发

对 UDF 的支持

目前 SQL 化实时任务的支持工作正在进行中。

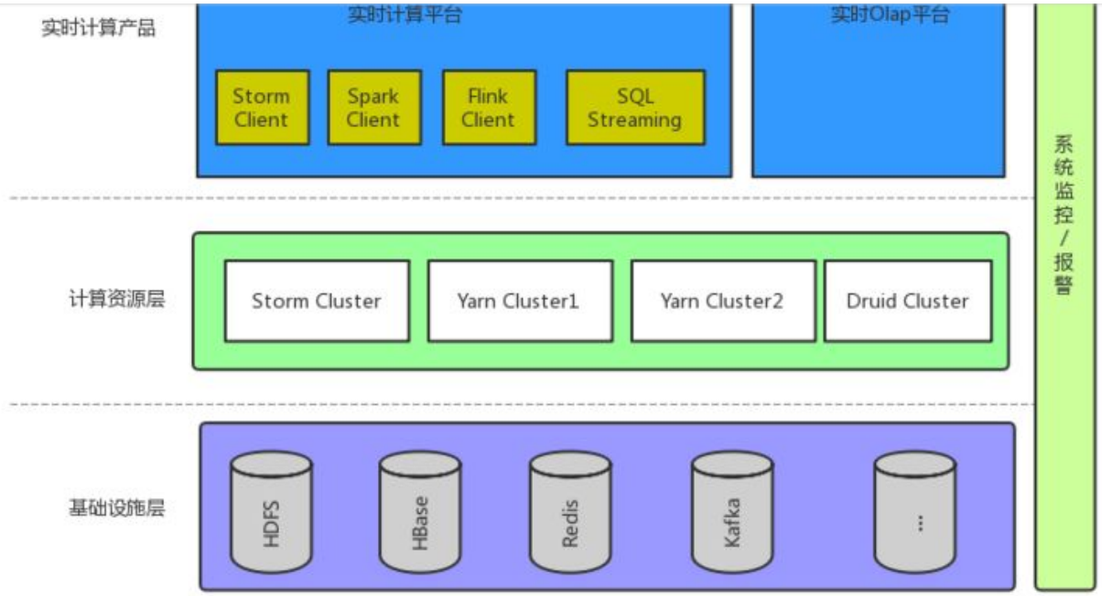
2.2.2.2 引入实时 OLAP 引擎

通过对业务的观察，我们发现在业务的实时应用中，有大量的需求是统计在不同维度下的 uv，pv 类统计，模式相对固定，对于此类需求，我们把目光放在了支持数据实时更新，并且支持实时的 Olap 类查询上的存储引擎上。

我们主要调研了 Kudu，Druid 两个技术栈，前者是 C++ 实现，分布式列式存储引擎，可以高效的做 Olap 类查询，支持明细数据查询；后者是 Java 实现的事件类数据的预聚合 Olap 类查询引擎~

综合考虑了运维成本，与当前技术栈的融合，查询性能，支持场景后，最终选择了 Druid。

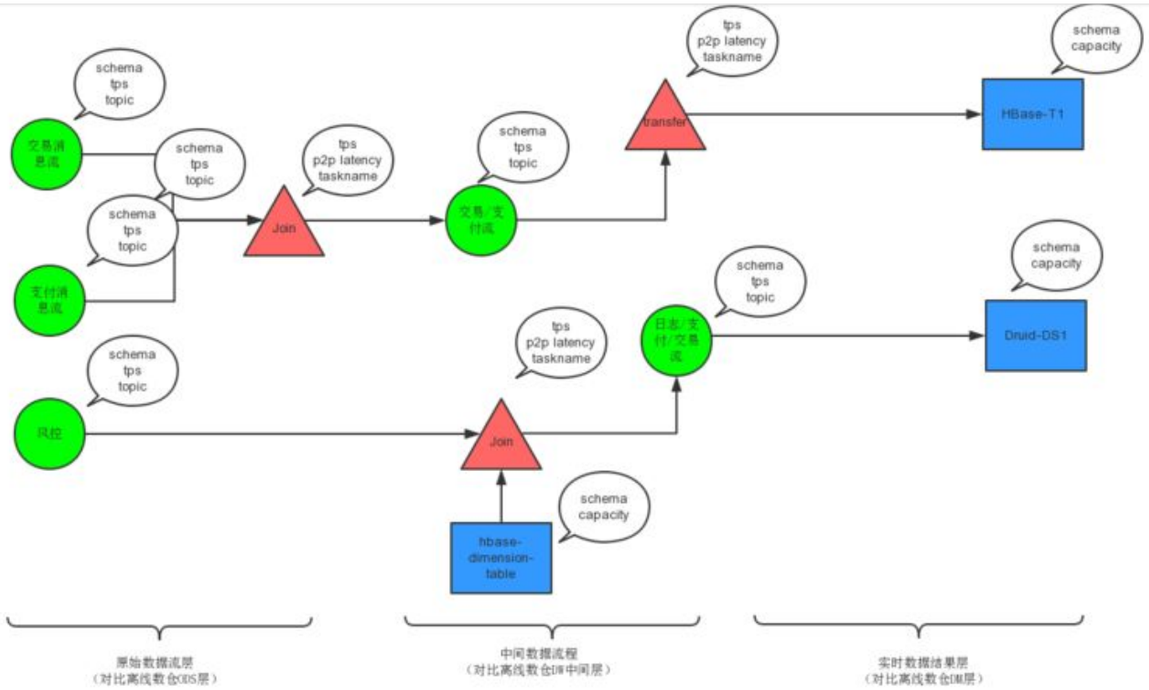
目前实时计算在有赞的整体技术架构如下图：



未来规划

首先要落地并的是实时任务 SQL 化，提高 SQL 化任务可以覆盖的业务场景（目标是 70%），从而通过提高业务开发效率的角度赋能业务。

在 SQL 化实时任务初步完成后，流数据的复用变成了提高效率上 ROI 最高的措施，初步计划会着手开始实时数仓的建设，对于实时数仓的初步设计如下图：



当然，完整的实时数仓绝没有这么简单，不只是实时计算相关的基础设施要达到一定的平台化水平，还依赖实时元数据管理，实时数据质量管理等配套的组件建设，路漫漫其修远~

总 结

有赞实时计算在业务方的需求下推动前进，在不同的阶段下，技术方向始终朝着当前投入产出比最高的方向在不断调整。本文并没有深入技术细节，而是循着时间线描述了实时计算在有赞的发展历程，有些地方因为作者认知有限，难免纰漏，欢迎各位同行指

作者介绍

贺飞，2017 年 7 月加入有赞大数据团队 - 基础平台组，先后负责有赞 HBase 存储的落地和数据基础各个组件的平台化工作。 有赞大数据团队是有赞共享技术核心技术团队之一，该团队主要由算法，数据产品，数据仓库和底层基础平台四个团队构成，目前共有 50 位优秀的工程师组成。

今日荐文

点击下方图片即可阅读

上学还是坐牢？百年老校“监控”学生惹争议

精品推荐

5 月 25-28 日，QCon 全球软件开大会广州站特别设置了“人工智能驱动业务实践”专