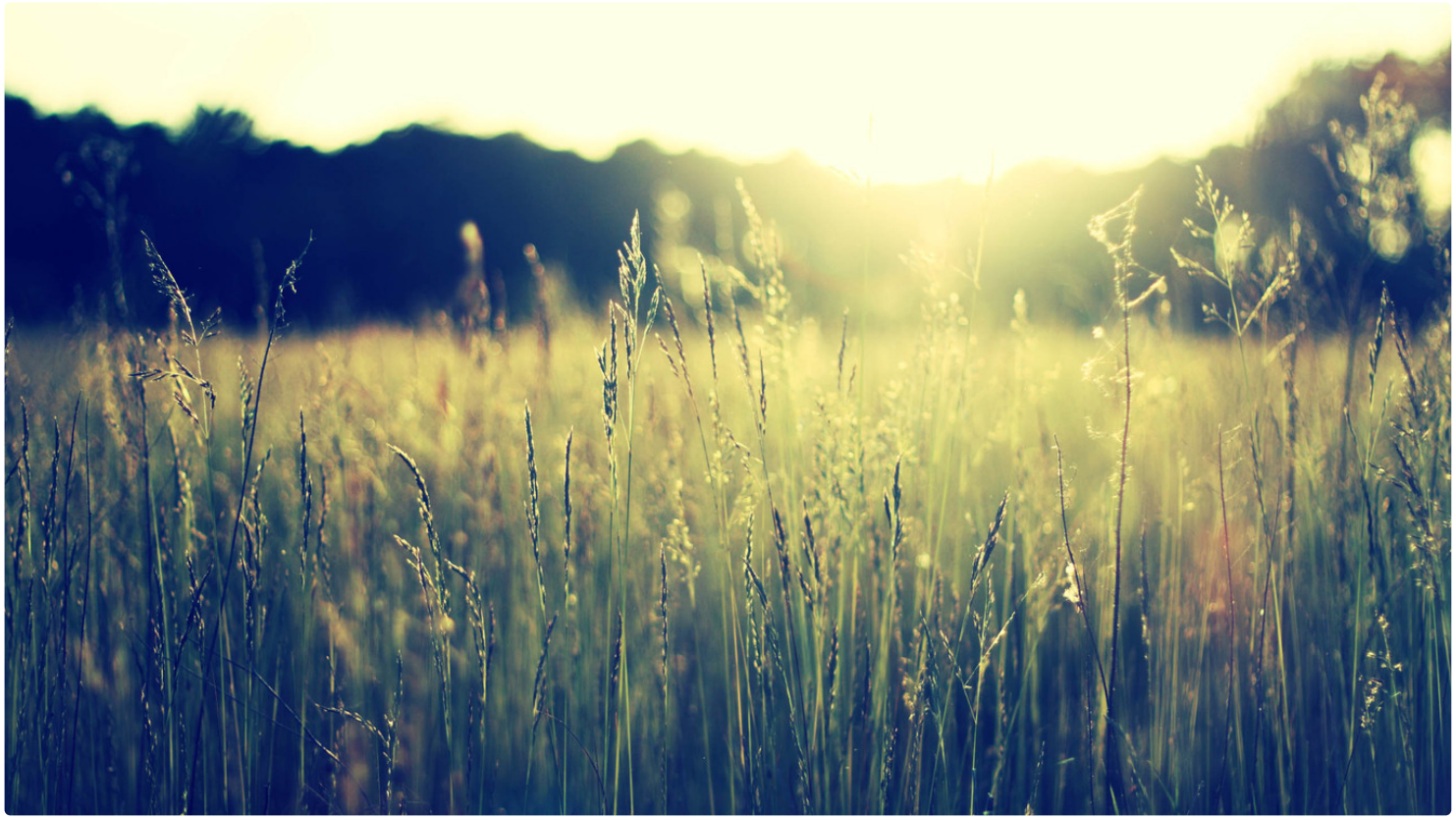
有赞百亿级日志系统架构设计

作者：饶榕

阅读数： 1571 2019 年 4 月 17 日



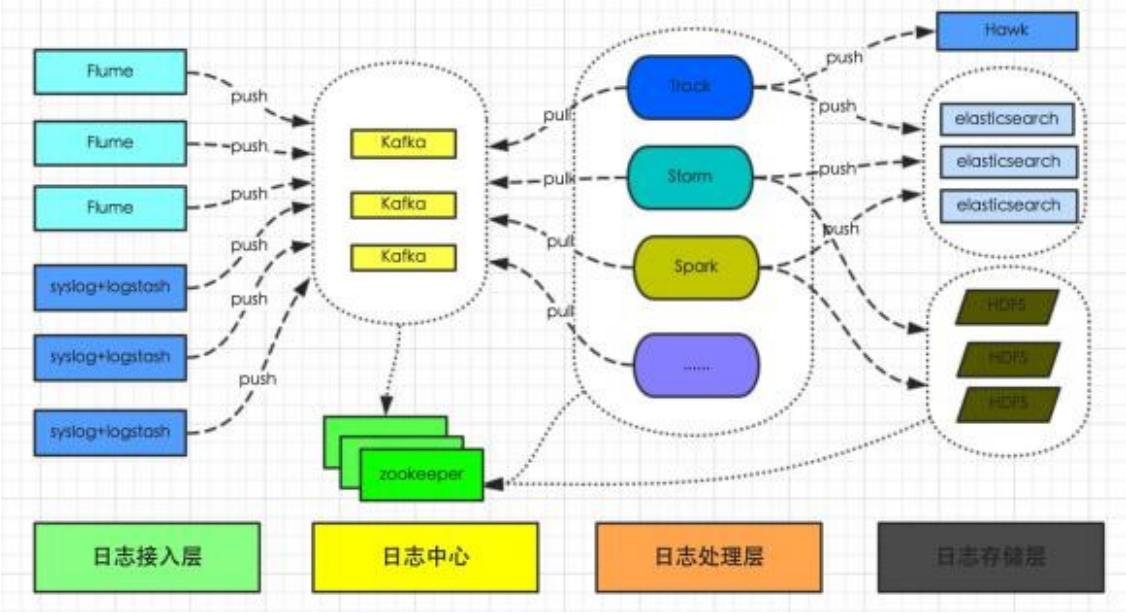
一、概述

日志是记录系统中各种问题信息的关键，也是一种常见的海量数据。日志平台为集团所有业务系统提供日志采集、消费、分析、存储、索引和查询的一站式日志服务。主要为了解决日志分散不方便查看、日志搜索操作复杂且效率低、业务异常无法及时发现等等问题。

随着有赞业务的发展与增长，每天都会产生百亿级别的日志量（据统计，平均每秒产生 50 万条日志，峰值每秒可达 80 万条）。日志平台也随着业务的不断发展经历了多次改变和升级。本文跟大家分享有赞在当前日志系统的建设、演进以及优化的经历，这里先抛砖引玉，欢迎大家一起交流讨论。

二、原有日志系统

有赞从 16 年就开始构建适用于业务系统的统一日志平台，负责收集所有系统日志和业务日志，转化为流式数据，通过 flume 或者 logstash 上传到日志中心 (kafka 集群)，然后共 Track、Storm、Spark 及其它系统实时分析处理日志，并将日志持久化存储到 HDFS 供离线数据分析处理，或写入 ElasticSearch 提供数据查询。整体架构如下图 所示。



随着接入的应用的越来越多，接入的日志量越来越大，逐渐出现一些问题和新的需求，主要在以下几个方面：

1. **业务日志没有统一的规范，业务日志格式各式各样，新应用接入无疑大大的增加了日志的分析、检索成本。**
2. **多种数据日志数据采集方式，运维成本较高**
3. **存储方面，**

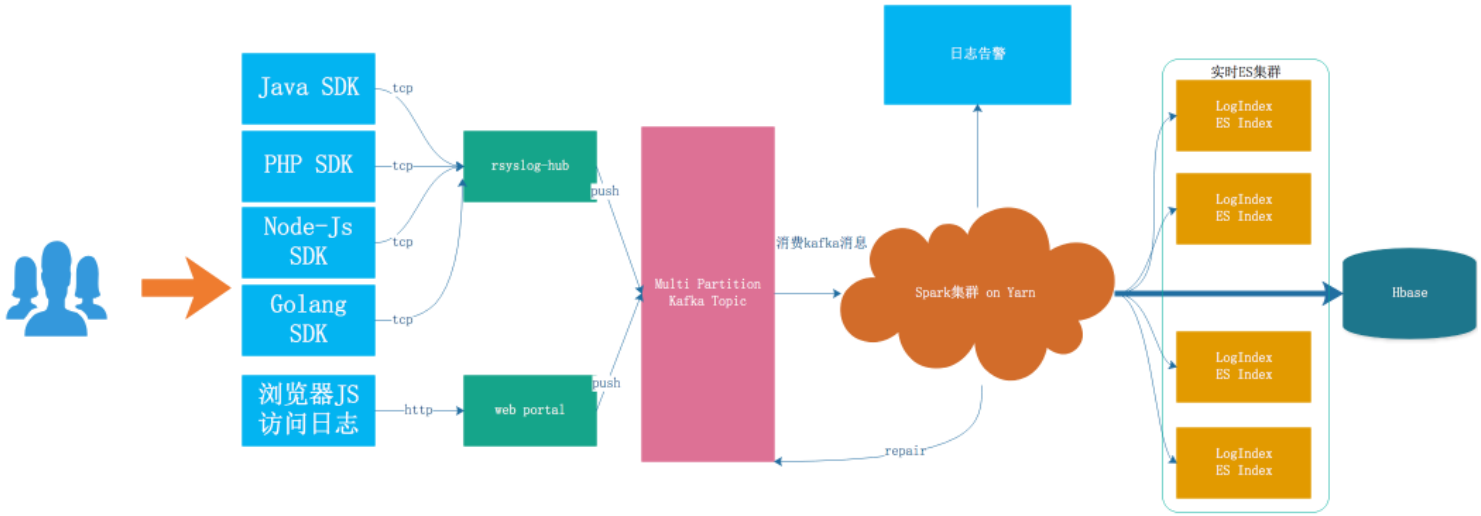
- **采用了** Es **默认的管理策略，所有的** index **对应** 3*2 **个** shard（3 **个** primary，3 **个** replica），**有部分** index **数量较大，对应单个** shard **对应的数据量就会很大，导致有** hot node，**出现很多** bulk request rejected，**同时磁盘** IO **集中在少数机器上。**

- **对于 bulk request rejected 的日志没有处理，导致业务日志丢失**
- **日志默认保留 7 天，对于 ssd 作为存储介质，随着业务增长，存储成本过于高昂**
- **另外 Elasticsearch 集群也没有做物理隔离，Es 集群 oom 的情况下，使得集群内全部索引都无法正常工作，不能为核心业务运行保驾护航**

4. **日志平台收集了大量用户日志信息，当时无法直接的看到某个时间段，哪些错误信息较多，增加定位问题的难度。**

三、现有系统演进

日志从产生到检索，主要经历以下几个阶段：采集 -> 传输 -> 缓冲 -> 处理 -> 存储 -> 检索，详细架构如下图所示：

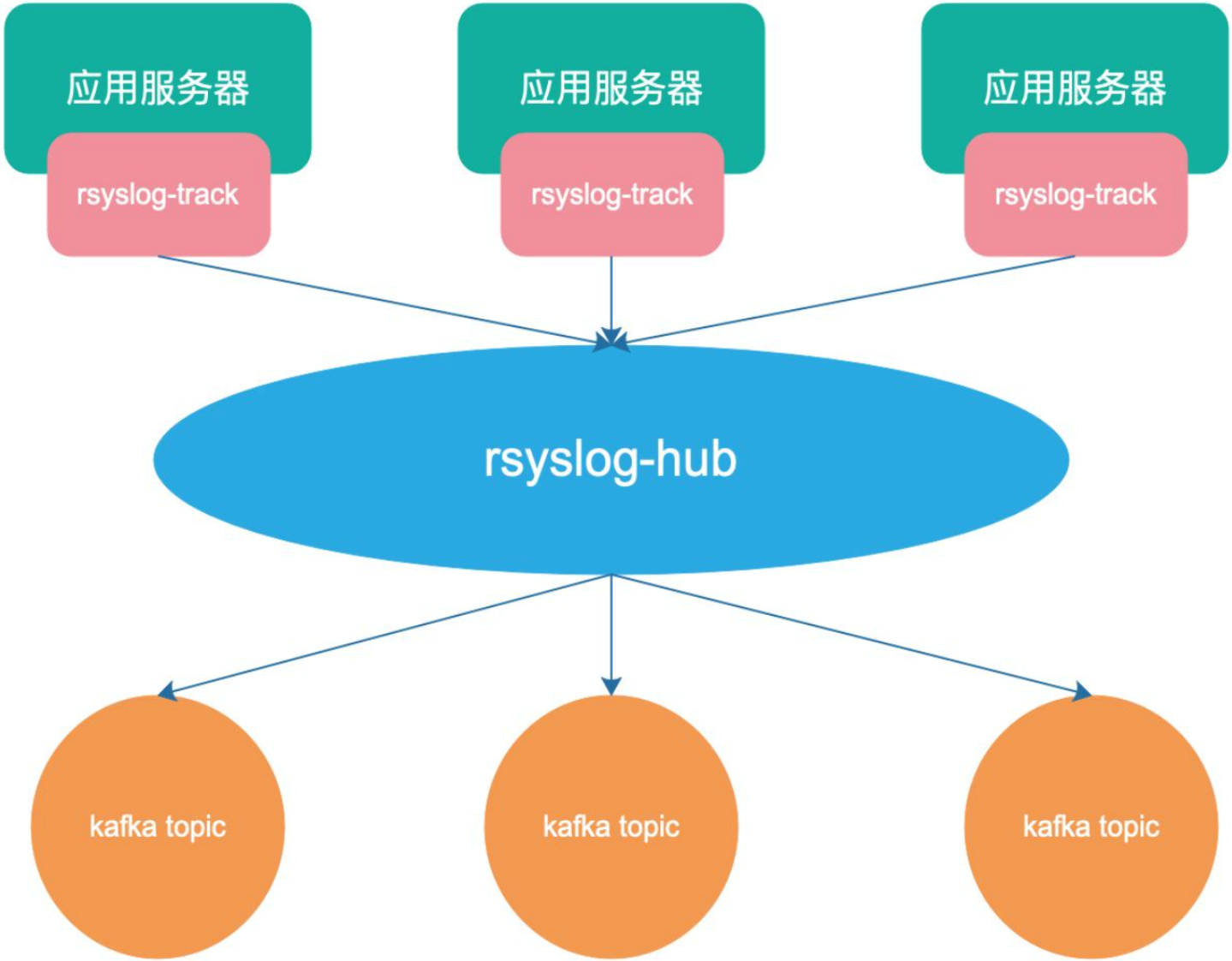


3.1 日志接入

日志接入目前分为两种方式，SDK **接入和调用** Http Web **服务接入**

- SDK **接入：日志系统提供了不同语言的** SDK，SDK **会自动将日志的内容按照统一的协议格式封装成最终的消息体，并最后最终通过** TCP **的方式发送到日志转发层（**rsyslog-hub**）**
- Http Web **服务接入：有些无法使用** SDk **接入日志的业务，可以通过** Http **请求直接发送到日志系统部署的** Web **服务，统一由** web protal **转发到日志缓冲层的** kafka **集群**

3.2 日志采集



现在有 rsyslog-hub **和** web portal **做为日志传输系统，**rsyslog **是一个快速处理收集系统日志的程序，提供了高性能、安全功能和模块化设计。之前系统演进过程中使用过直接在宿主机上部署** flume **的方式，由于** flume **本身是** java **开发的，会比较占用机器资源而统一升级为使用** rsyslog **服务。为了防止本地部署与**

kafka 客户端连接数过多，本机上的 rsyslog 接收到数据后，不做过多的处理就直接将数据转发到 rsyslog-hub 集群，通过 LVS 做负载均衡，后端的 rsyslog-hub 会通过解析日志的内容，提取出需要发往后端的 kafka topic。

3.3 日志缓冲

Kafka 是一个高性能、高可用、易扩展的分布式日志系统，可以将整个数据处理流程解耦，将 kafka 集群作为日志平台的缓冲层，可以为后面的分布式日志消费服务提供异步解耦、削峰填谷的能力，也同时具备了海量数据堆积、高吞吐读写的特性。

3.4 日志切分

日志分析是重中之重，为了能够更加快速、简单、精确地处理数据。日志平台使用 spark streaming 流计算框架消费写入 kafka 的业务日志，Yarn 作为计算资源分配管理的容器，会跟不同业务的日志量级，分配不同的资源处理不同日志模型。

整个 spark 任务正式运行起来后，单个批次的任务会将拉取的到所有的日志分别异步的写入到 ES 集群。业务接入之前可以在管理台对不同的日志模型设置任意的过滤匹配的告警规则，spark 任务每个 excutor 会在本地内存里保存一份这样的规则，在规则设定的时间内，计数达到告警规则所配置的阈值后，通过指定的渠道给指定用户发送告警，以便及时发现问题。当流量突然增加，es 会有 bulk request rejected 的日志会重新写入 kakfa，等待补偿。

3.5 日志存储

- 原先所有的日志都会写到 SSD 盘的 ES 集群，logIndex 直接对应 ES 里面的索引结构，随着业务增长，为了解决 Es 磁盘使用率单机最高达到 70%~80% 的问题，现有系统采用 Hbase 存储原始日志数据和

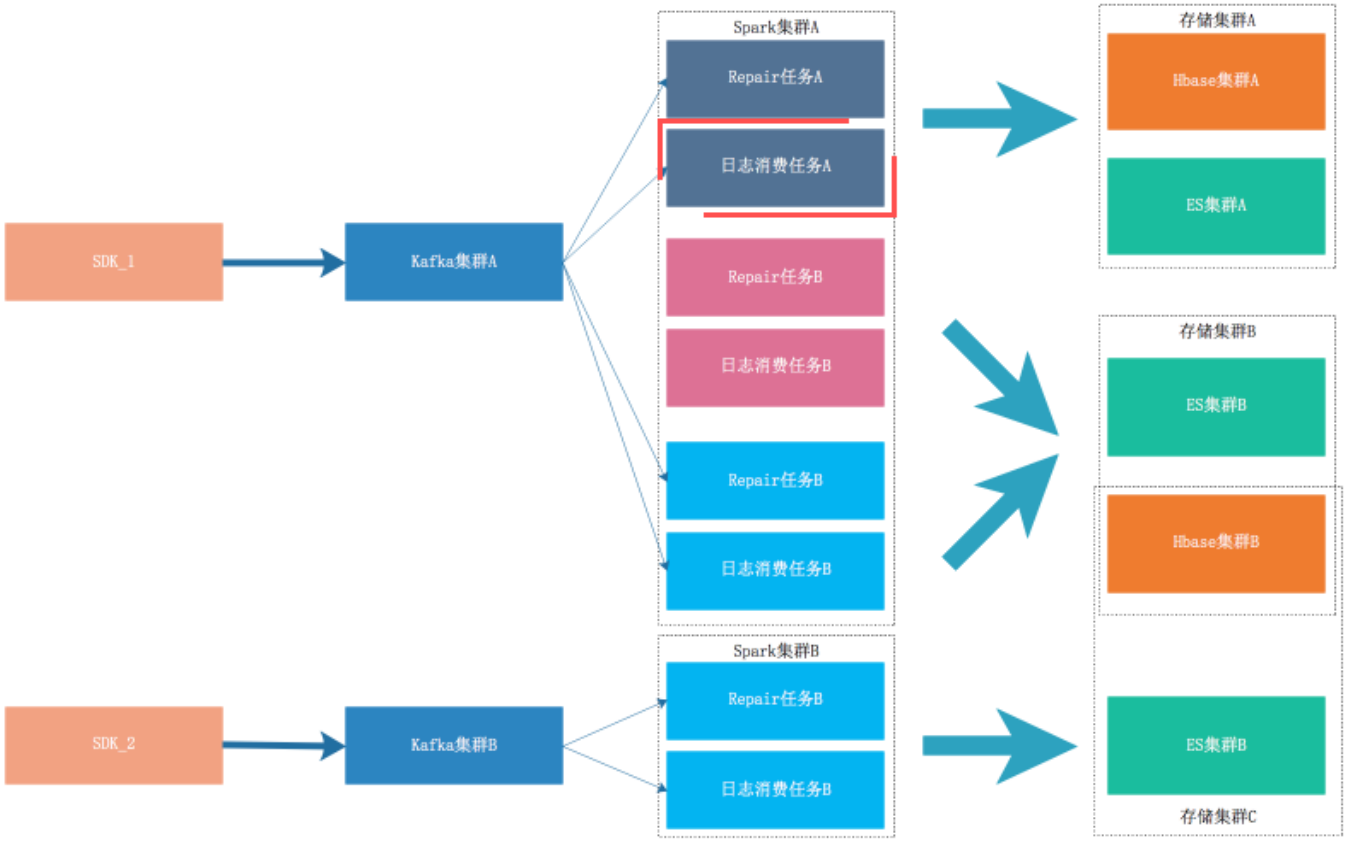
ElasticSearch 索引内容相结合的方式，完成存储和索引。

- Index 按天的维度创建，提前创建 index 会根据历史数据量，决定创建明日 index 对应的 shard 数量，也防止集中创建导致数据无法写入。现在日志系统只存近 7 天的业务日志，如果配置更久的保存时间的，会存到归档日志中。

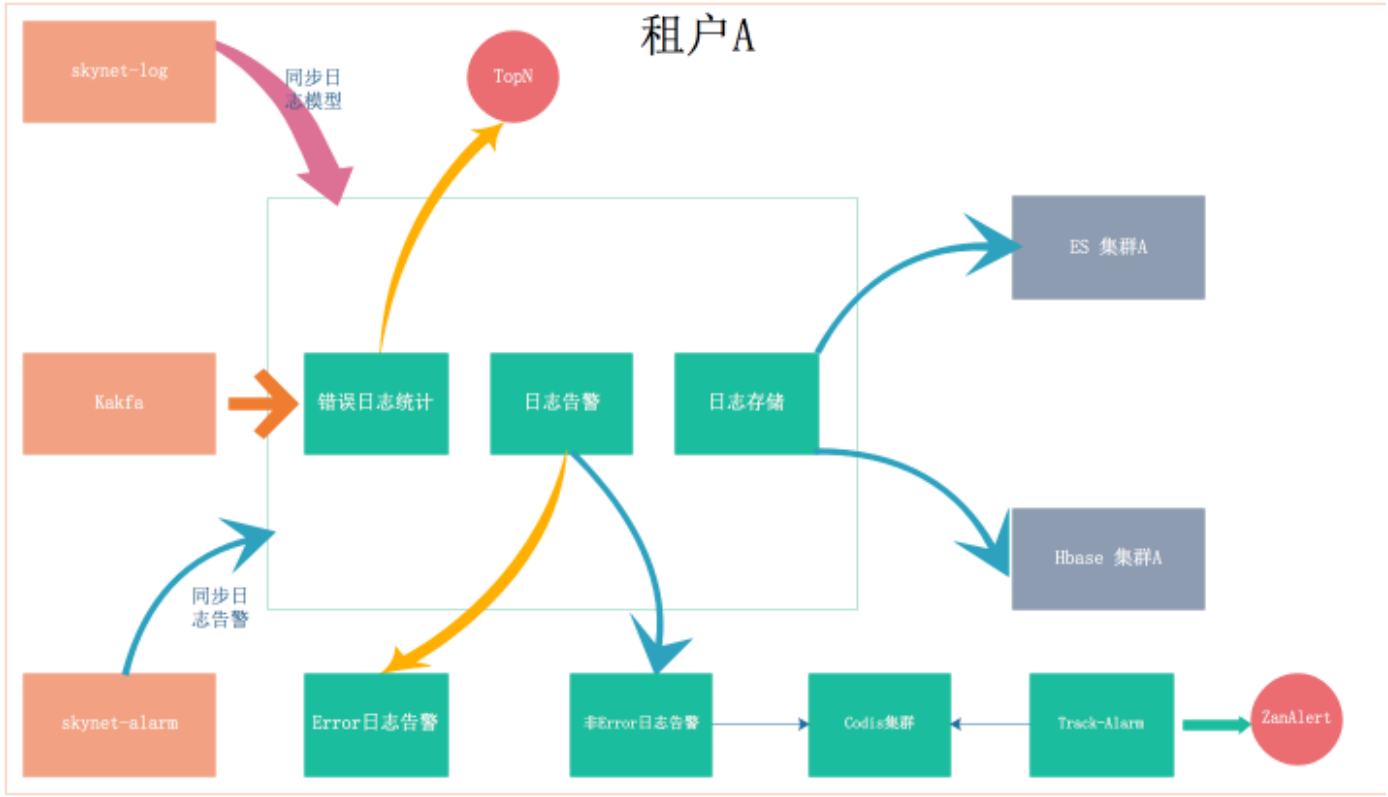
- 对于存储来说，Hbase、Es 都是分布式系统，可以做到线性扩展。

四、多租户

随着日志系统不断发展，全网日志的 QPS 越来越大，并且部分用户对日志的实时性、准确性、分词、查询等需求越来越多样。为了满足这部分用户的需求，日志系统支持多租户的的功能，根据用户的需求，分配到不同的租户中，以避免相互影响。



针对单个租户的架构如下：



- SDK：**可以根据需求定制，或者采用天网的** TrackAppender **或** SkynetClient
- Kafka **集群：可以共用，也可以使用指定** Kafka **集群**
- Spark **集群：目前的** Spark **集群是在** yarn **集群上，资源是隔离的，一般情况下不需要特地做隔离**
- **存储：包含** ES **和** Hbase**，可以根据需要共用或单独部署** ES **和** Hbase

五、现有问题和未来规划

目前，有赞日志系统作为集成在天网里的功能模块，提供简单易用的搜索方式，包括时间范围查询、字段过滤、NOT/AND/OR、模糊匹配等方式，并能对查询字段高亮显示，定位日志上下文，基本能满足大部分现有日志检索的场景，但是日志系统还存在很多不足的地方，主要有:

1. 缺乏部分链路监控：日志从产生到可以检索，经过多级模块，现在采集，日志缓冲层还未串联，无法对丢失情况进行精准监控，并及时推送告警。

2. 现在一个日志模型对应一个 kafka topic，topic 默认分配三个 partition，由于日志模型写入日志量上存在差异，导致有的 topic 负载很高，有的 topic 造成一定的资源浪费，且不便于资源动态伸缩。topic 数量过多，导致 partition 数量过多，对 kafka 也造成了一定资源浪费，也会增加延迟和 Broker 宕机恢复时间。

3. 目前 Elasticsearch 中文分词我们采用 ik_max_word，分词目标是中文，会将文本做最细粒度的拆分，但是日志大部分都是英文，分词效果并不是很好。

上述的不足之处也是我们以后努力改进的地方，除此之外，对于日志更深层次的价值挖掘也是我们探索的方向，从而为业务的正常运行保驾护航。

架构 运维 电商

4 人喜欢