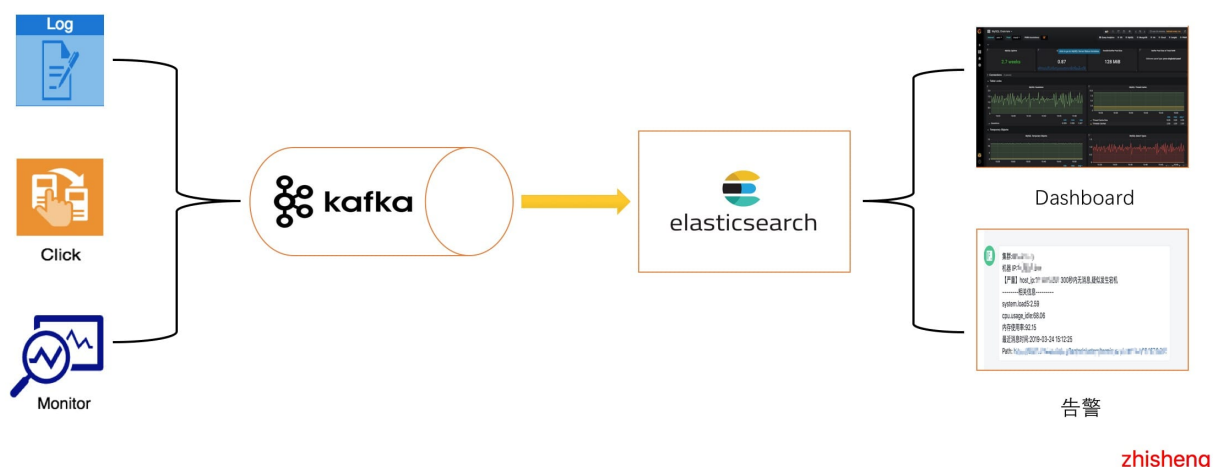


前言

大家好，我是 zhisheng，公众号（zhisheng）的作者，CSDN 博客专家，在公司主要负责监控平台的流式告警。欢迎加入我的专栏，和我一起开启 Flink 的学习之路。

我 2018 年 6 月加入公司，进入公司后一直负责监控平台的告警系统，从负责监控平台告警系统之后，我们的整个监控平台架构中途换过两次，其中有一次架构有巨大的变化。先来说下我们监控告警平台最早的架构是：



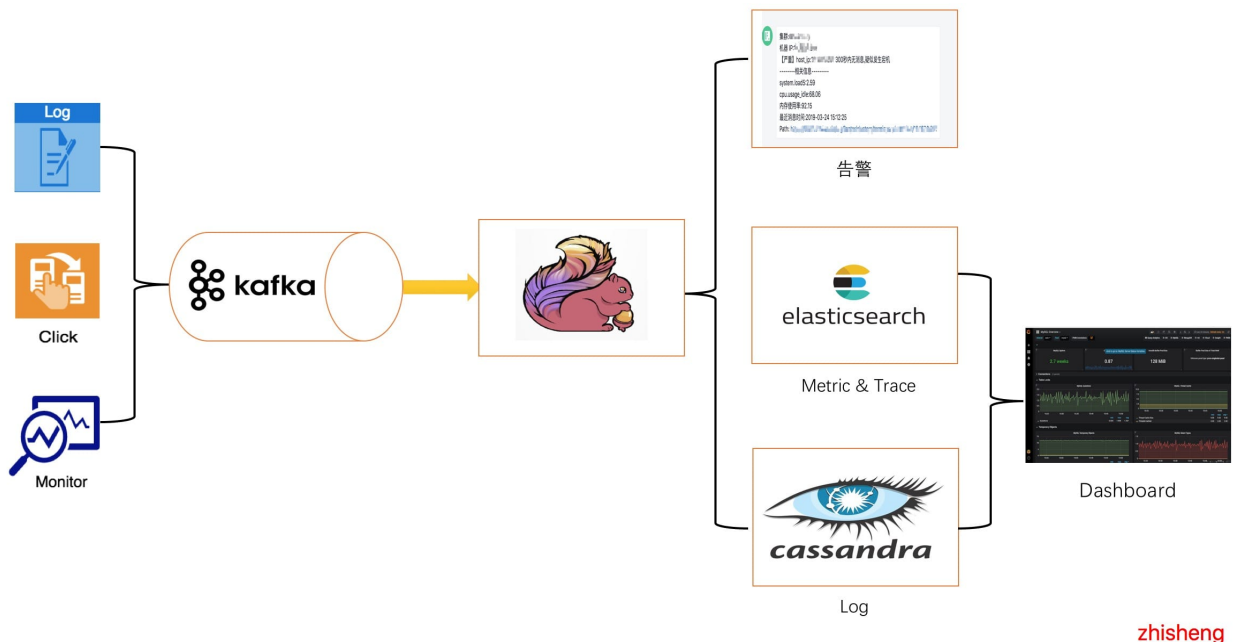
先来说下这个架构的挑战点：

- 海量的监控数据（Metric & Log & Trace 数据）实时写入 Elasticsearch；
- 多维度的监控指标页面展示（Dashboard）查 Elasticsearch 的数据比较频繁；
- 不断递增的告警规则需要通过查询 Elasticsearch 数据来进行判断是否要告警

从上面的几个问题我们就可以很明显的发现这种架构的瓶颈就在于 Elasticsearch 集群的写入和查询能力，在海量的监控数据（Metric & Log & Trace 数据）下实时的写入对 Elasticsearch 有极大的影响，我依稀清楚的记得，那会经常是因为写入的问题导致 Elasticsearch 集群经常挂掉，从而让我的告警和监控页面（Dashboard）歇菜（那会老被喷为啥配置的告警规则为啥没有触发告警 & 业务开发问为啥查看应用的 Dashboard 监控页面为啥没数据），这种情况下我也很无奈啊，只想祈祷我们的 Elasticsearch 集群稳一点。

初次接触 Flink

就这样的架构情况下，我们经历了几个月，后面由于一些特殊的原因，我们监控平台组整体做了一个很大的架构调整，如下图：



主要做了四点改变：

- 接入 Flink 集群去消费 Kafka 数据，告警的 Flink Job 消费 Kafka 数据去判断异常点，然后做告警
- Metric & Trace 数据存储到 ElasticSearch，之前还存储在 ElasticSearch 中的有 Log 数据
- Log 数据存储到 Cassandra
- Dashboard 查询数据增加 API 查询 Cassandra 的日志数据

原先因为 Metric & Trace & Log 的数据量组成一起全部实时写入到 ElasticSearch 中，对 ElasticSearch 的压力很大，所以我们将 Log 的数据拆分存储到 Cassandra 中，分担一些 ElasticSearch 的写入压力，但是分担压力后我们还发现有时候还会出现数据实时写入到 ElasticSearch 集群把 ElasticSearch 写挂的情况，所以那会不断的调优我们的写入数据到 ElasticSearch 的 Flink Job，然后也对 ElasticSearch 服务端做了不少的性能调优，另外那会我们的监控数据是以 10s 一次为单位将采集的数据发上来的，后面我们调整了下数据采集的策略（变成 30s 一次为单位采集数据），多种调优策略后，终于将我们的 ElasticSearch 弄稳定了。

遇到 Flink 相关的挑战

替换成这种新架构后，组里因为也没人熟悉 Flink，再加上那会 Flink 的资料真的很少很少，所以当时在组里对 Flink 这块大家都是从 0 开始学习，对大家的挑战还不少。

所以在那会我们跑在 Flink 上面的 Job 也遇到各种各样的问题：

- 消费 Kafka 数据延迟
- checkpoint 失败
- 窗口概念模糊、使用操作有误
- Event Time 和 Processing Time 选择有误
- 不知道怎么利用 Watermark 机制来处理乱序和延迟的数据

- Flink 自带的 Connector 的优化
- Flink 中的 JobManager 和 TaskManager 经常挂导致 Flink Job 重启
- Flink 集群模式的选型

...

因为碰到的各种各样的问题，所以才会促使我们不断的学习 Flink 的原理和内部机制，然后慢慢的去解决上面遇到的各种问题，并逐步的稳定我们监控平台所运行的 Flink Job。

Flink 系列博客

在这个过程中自己也在自己的 [博客](#) 里面不断的记录自己的 Flink 学习之路，到写这篇文章的时候已经对外公布了 20+ 篇 Flink 的个人学习博客，在这期间自己收到一大堆出版社的编辑来邀请我写本 Flink 的书籍（或者整理现有的博客然后写本书），但无奈自己那会也一直处于学习阶段，怕自己写不好反倒是误导读者。因为写书的流程比较长：书本大纲编写 -> 大纲审核调研 -> 写稿 -> 交稿 -> 排版 -> 初审 -> 复审 -> 终审 -> 申请书号 -> 印刷 -> 入库 -> 物流 -> 销售 等，耗时比较长，所以自己都是直接先拒绝了。在这个阶段，因为自己写了一系列的博客，所以好多对 Flink 感兴趣的童鞋也加我一起讨论问题，后面自己干脆创建了微信交流群，截止目前已经两个微信群了，里面快 700 人了，每天里面的童鞋会提很多遇到的 Flink 问题，但是我发现一个问题，回答的很少，其实并不是因为不活跃，而是因为对 Flink 了解的还不是很多，比如有的是大数据工程师但之前是搞 Spark 这块的、有的是转大数据开发的后端开发工程师、有的是对 Flink 这块比较感兴趣的研究生 等。

因为自己就是从 Flink 小白过来的，所以知道初学者可能会遇到的哪些问题，当你回首的时候，你会发现这么简单的问题当时自己怎么还这么傻逼的折腾了半天都出不来。

所以自己在心里萌生了一个想法：多写点文章来帮助大家尽快的从小白阶段来过渡到入门阶段，然后再从入门到能够将 Flink 用起来，在生产环境真正的把你的 Flink Job 运行起来，然后能够根据你生产环境出现的错误进行排查并解决，然后还能不断的根据你的 Job 的运行状况进行进一步的优化，另外就是如果你对 Flink 源码感兴趣的话，欢迎加入我的知识星球：<https://t.zsxq.com/zV7MnuJ>，里面专门分享 Flink 源码解析，然后还进行各种答疑。

为什么要写 Flink 专栏？

上面也说了自己把出书的编辑都拒绝了，然后可能会有读者会问，那你为啥还开着门 Flink 专栏的课程啊，我这里针对这个问题回答一下：

- 个人博客宣传的力度太小，对 Flink 国内的推广比较有限，希望多通过平台级别的推广，帮助 Flink 社区在国内更快的落地，希望更多的公司能够接入新一代的实时计算引擎 —— Flink
- 市面上讲 Flink 的太少太少，国内的中文资料太欠缺，已有的几本书籍也是不堪入目
- 自己将上面讲的那个想法实现掉，抽空闲时间写个系列文章来帮助更多的人更快的入门和掌握 Flink
- 自己也是一个提高的机会，一个人会很简单，但是要将一个东西讲给 100 个人会、1000 个人会、甚至 10000 个人会就真有点挑战了，不过我接受这挑战！勇敢的尝试一次，教学过程中也得强迫自己去梳理知识，找到相通之处，深入挖掘，总结成一个知识体系，从而达到另一个层

次！

自己之前在 GitChat 写过了一篇 Chat：[大数据“重磅炸弹”：实时计算框架 Flink](#)，然后自己也在微信群里面给了很多童鞋做了答疑，从答疑的过程中知道大家对 Flink 还是比较关注的，好多公司在陆续的调研。

虽然市面上 Flink 的资料很少，但是国内在阿里的推动下，我相信 Flink 会越来越火的，并且阿里内部也将 Flink 做了一定的优化和修改，叫 Blink，今年年初也将源码贡献到 Flink 上面，后面在 Flink 1.9 版本会将 Blink 的功能进行合并到 Flink 上去，所以自己也开始编写这门专栏，并在这门专栏定了四个篇幅：预备篇 -> 基础篇 -> 进阶篇 -> 高级篇 -> 实战篇 -> 系统案例篇。

专栏目录大纲

预备篇：

在预备篇中我将先介绍实时计算、并讲解常见的使用场景，从而引申到我们的主角 —— Flink，接着对比 Flink、Spark Streaming 和 Storm，然后开始准备我们的本地运行环境和安装 Flink，然后通过两个入门的程序（Wordcount 和处理 Socket 数据）先来讲解 Flink 程序如何执行，并讲解了如何部署我们开发的 Flink 应用程序。

基础篇：

在基础篇内我将先讲解 Flink 的基础概念，然后深入讲解 Flink 里面的 Time、Window、Watermark、Connector，并有大量的文章篇幅讲解 Flink 中的 Connector，因为这块估计真实场景用的也非常多，我除了会讲解如何使用这些 Connector 外，还是深入其源码做一定的讲解，除此之外我还会讲解如何自己去实现一个 Connector（Source & Sink）。

进阶篇：

在进阶篇中我将讲解 Flink 里面的 State、Checkpoint、Savepoint、内存管理机制、CEP、Table API、SQL API、Machine Learning、Gelly。在这篇中我不仅只讲概念，还会讲解如何去使用 State、如何配置 Checkpoint 和如何利用 CEP 处理复杂事件。

高级篇：

在高级篇中我将讲解 Flink 中的一些监控运维相关的，以及一些最佳实践。比如如何查看 Flink Job 的执行计划，根据 Job 运行的 Metric 数据进行调优，处理 Job 出现背压的情况，如何合理的设置并行度，如何处理数据倾斜问题，如何管理 Flink Job 里面的配置，如何处理数据源中数据的多种格式，还有就是详细的讲解 Flink 的配置文件并讲解如何配置高可用的 JobManager？

实战篇：

在实战篇中我将讲解如何用 Flink 去处理我们常见的实时场景需求，比如统一一天内的 PV 和 UV、统计商品在大促时一分钟的销量、应用 Error 异常日志实时告警、机器宕机告警等。

系统案例篇：

在系统案例篇中我将讲解如何基于 Flink 实时处理海量的的日志、如何基于 Flink 做实时监控告警系统、如何基于 Flink 实时处理并存储数据、如何基于 Flink 做实时监控大屏、如何基于 Flink CEP 处理复杂事件！多个系统案例帮助你更好运用 Flink。

在这里，有一份目录，你也可以先了解下整个专栏的知识结构：

这里放到时候专栏的大纲目录结构