



ES在华为电信软件 运维中的应用

肖曙旭

华为电信软件云运维开发部

xiaoshuxu@huawei.com

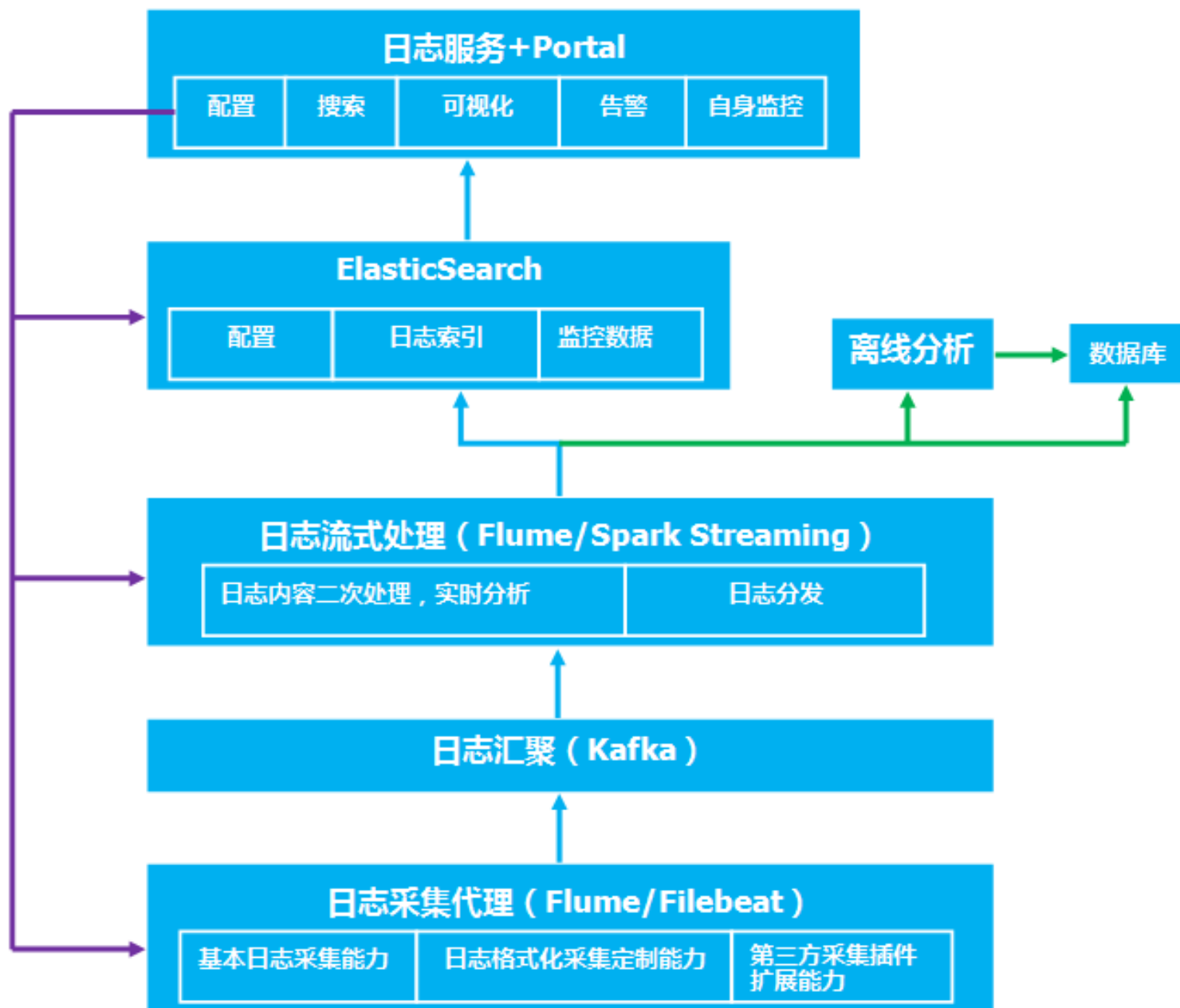
2017-06-05



目录

- 系统架构概述
- 日志采集格式化定制
- 日志采集监控
- 基于日志的告警
- 电信领域应用实例
- 实践过程中的优化经验

系统架构概述




日志采集格式化定制

日志采集代理使用**Flume**实现，通过**Flume**的**Source**和**Sink**的定制能力实现日志采集格式化定制。

- 支持采集目录，文件名的黑白名单；
- 支持正则表达式或者分隔符匹配逻辑行，且支持逻辑行的黑白名单；
- 支持按分隔符或者正则表达式将逻辑行分割为格式化字段列表；
- 支持直接使用正则表达式从日志中提取字符串作为独立字段；
- 支持一些简易的算子对字段做初步处理，比如数值运算，字段串截取等。

<<<A|B|C|D|E|F|G>>>



```
{  
  Field_A:A,  
  Field_B:B,  
  Field_C:C,  
  Field_D:D,  
  Field_E:E,  
  Field_F:F,  
  Field_G:G  
}
```

日志采集监控

依赖**Flume**自身的**metric**数据，定制计算逻辑，获取我们需要的信息，比如每条通道采集的日志总数，采集速率，数据发送的成功率，异常问题列表等。将这些信息以单独的日志类型通过数据通道入**ElasticSearch**保存，最终监控微服务定时查询相关索引获取采集状态并在界面呈现。

基于日志的告警

日志中体现了一个系统的运行状态，通过对日志的分析，我们能够了解系统是否正常。当日志中出现一些确定的关键词或者某项指标超出阈值的情况下，我们希望运维能够以告警的方式上报风险。出于上述考虑，并结合ES的搜索能力，我们构建了相关能力：

- 统计最近一个周期内某个关键词出现的次数，满足一定条件则告警；
- 统计最近一个周期内某个数值指标的和/平均值/最大值/最小值，满足一定条件则告警；
- 统计最近两个周期内某个关键词出现的次数，两者的差异超过阈值设置则告警；
- 统计最近两个周期内某个数值指标的和/平均值/最大值/最小值，比较两者差异超过阈值则告警

电信领域应用实例——某省移动BES

业务规模：1000+节点，日志平均5W+条/秒

ES规模：10个节点，每个节点6C56G

应用场景：

- 故障定界定位：通过界面异常信息，业务节点监控等可以了解问题出现的业务集群甚至节点，以及导致问题的关键词，从业务监控直接下钻到日志搜索，根据相关信息过滤得到错误日志，再结合日志钻取功能获取错误日志的上下文，从而定位出问题根因；
- 状态监控：通过ES的搜索和聚合能力配置分析图表或者告警，比如配置Nginx平均时延趋势图来监控Nginx响应状态，配置最近5分钟内Nginx的error日志中出现time out的次数超过一定阈值就告警；
- 统计分析：基于ES的聚合能力针对采集提取的字段进行有效的分析，比如从业务日志中提取错误码字段，配置TopN表格统计出现频率最高的10种错误码；分析调用链日志，统计服务调用时延，找到系统耗时瓶颈等，作为后续系统优化的有效输入。

实践过程中的优化经验（一）

◆ 文档索引速度优化

这里简单列举几个有明显作用的优化项：

- 指定字段格式，包括字段类型，是否索引，是否分词；不需要搜索的字段设置成不索引，简单且一般全词匹配的字段设置成不分词 `not_analyzed`。
- 禁用 `_all` 字段，节省磁盘空间，降低磁盘IO； `"_all" : {"enabled" : true}`
- 非必须的日志不设置副本； `index.number_of_replicas: 1`
- 设置合理的刷新闻隔；增加刷新闻隔有效降低磁盘IO的占用，但是由于ES的索引数据只有刷新后才能够被搜索到，所以增加刷新闻隔会在一定程度上影响搜索的实时性。例如： `index.refresh_interval: 30s`
- 使用SSD替代传统机械磁盘。在实际应用中，可以发现往往是由于磁盘的IO瓶颈限制了索引的速率，而非CPU和内存。
- 升级ES到最新版本。实测相同条件下，所有字段均指定格式的情况下5.X版本比1.X版本的索引性能提升30%左右。

实践过程中的优化经验（二）

◆ 通过设置字段的doc_values属性解决聚合查询容易导致OOM的问题

ES在排序和聚合查询的时候会将fielddata全部写入内存，大量占用heap空间，从而可能引起OOM。将所有不分词的字段（包括数值型字段）的doc_values属性设置成true，该字段的fielddata就会被写入磁盘，从而降低对内存的压力，避免OOM。同时，实测其对检索的影响也不大，相同条件下，检索速度大概慢了10%左右。

◆ 检索性能优化

- 按照时间段创建索引，搜索的时候可以基于时间条件指定范围内的索引名搜索，减少了搜索范围，对搜索性能有一定的提升，同时减小了索引分片的大小，能够防止分片过大导致索引效率降低；
- 默认按时间排序，禁止相关度分值计算；
- 使用最新的ES版本。实测相同条件下，所有字段均指定格式的情况下5.X版本比1.X版本的索引性能提升50%左右

实践过程中的优化经验（三）

◆ 使用**Beats**替代**Flume**作为采集代理，降低资源占用

Flume的Channel会占用大量的内存，特别是日志类型较多的情况下，需要配置更多的采集通道，会导致内存占用大幅增加。使用Elastic Stack中的Beats替代Flume能够大幅降低内存占用。Beats中能够定制Processor，之前在Flume上定制的逻辑基本都可以迁移到processor中。当然，使用过程中也遇到一些问题：比如Beats的metric信息没有flume丰富，也无法针对不同的日志类型区分，集成的Kafka Output默认不支持kerberos鉴权等，需要自己定制开发。



谢谢大家

Q&A