Log Search

1. 问题描述

该问题是需要在分布式环境下收集和有效存储各服务的日志，并根据查询corr\_id来获得整个调用链。限制条件如下：

1. 响应时间要小于1分钟
2. 单个文件的大小平均为200M
3. 平均调用链长度可以假设为20
4. 文件的数量以万记
5. 问题分析

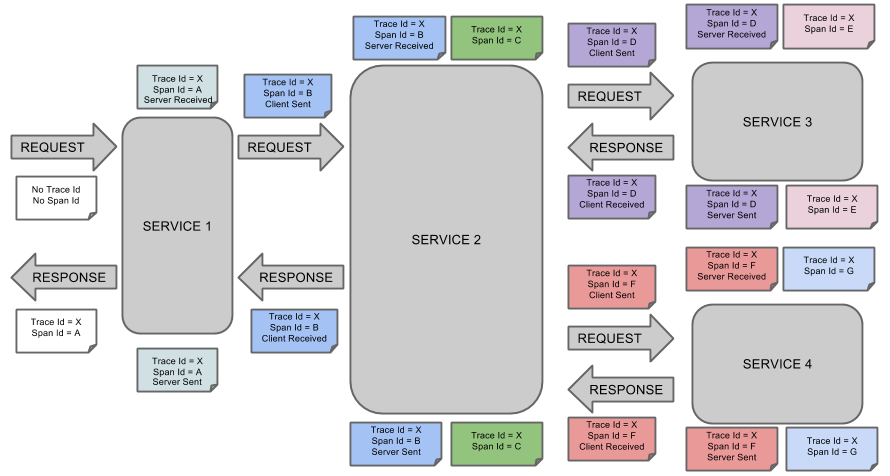
要解决的问题：**1. 找几万个文件里找到corr\_id所在的文件；**

**2. 在所在的文件里找到corr\_id相关的内容，构建成链**

类似的工具有很多，例如ELK分布式日志分析搜索引擎，zipkin分布式系统监控等等工具。

1. 解决方案

根据题目中日志的意思，每个请求会有corr\_id、当前生成的node\_id和请求的来源ri。如图所示：



用已有的工具实现的话就可以参考<https://github.com/liaokailin/zipkin>，但是感觉照着教程做一遍还是不懂意思，感觉意义不大。通过阅读Elk相关资料后，我认为这个问题如果我个人来完成，步骤应该是：

1. 对每个corr\_id做倒排索引，存储该id出现在哪些文件中，如表所示。这种方法查询时间复杂度为Log(N)，N指的是term数目，占用的空间大小是O(N\*str(term))。排序数组需要完整存储每一个term，当term数目多达上千万时，占用的内存将不可接受。

corr\_zxc->12\23\28…

corr\_vbn ->13\18\22…

corr\_bnm ->9\10\29…

Lucene解决该问题的方法为：FST，一种有限状态机，优点为：1. 空间占用小，通过对词典中单词前缀和后缀的重复利用，压缩了存储空间；2. 查询速度快，O(len(str))的查询时间复杂度。

1. 在文件里搜索corr\_id，构建正向索引表。即将文件中的关键词，关键词的偏移量和文档标识进行索引，构建正向信息，如索引－段－文档－域－词。就本题来看，索引结构应为：文档号－corr\_id-偏移量。这样能很快找出文档里包含的corr\_id的位置。
2. 整个分布式日志文件的管理过程

Analysis

**Logstash**

**Log File**

Filter

Input

Kafka

Filter

Output

**Kibana可视化展示**

**Elasticsearch存储**

位于各个节点上的服务器先将数据/日志传递给Kafka，并将队列中消息或数据间接传递给Logstash，Logstash过滤、分析后将数据传递给Elasticsearch存储。最后由Kibana将日志和数据呈现给用户。这样即使远端Logstash server因故障停止运行，数据将会先被存储下来，从而避免数据丢失。

1. 编码实现：

参照附件，因为时间有限，没有使用FST算法，就用了简单的HashMap和正则表达式，日志的流式增加也简化成了静态文件。思路一致，每新增一个日志文件，提取文件里的corr\_id,并增加到corr\_id的索引表中，同时将corr\_id,ri和node\_id三者构建成一个对象，存到该文档对应的hashmap中。

Main函数中输入一个corr\_id, 找到该corr\_id对应的所有Span对象，然后输出即可。