## 搜索引擎技术核心基础-Lucene

1. **ELK问题回顾**

目前主要有以下几个问题：

1. 启动错误：logstash启动错误。90%都是配置文件错误。# - {} log-
2. 启动正常但es没有数据:

a.确定logstash以及es启动成功：log不报错其成功，Debug。Es启动成功：访问x.x.x.x:9200/\_pugin/head：看到json.安装head

b.配置了end。不追加log是不会收集日志到es。

c.我们配置了换行合并。那么日志格式一定要满足你配置的换行正则，我们配置的是日期。所以你写的log一定要以日期开头。2019-01-02 11:22:00。

d.出现在es和log通信问题。是网咯太慢.就是收集不到实时的.云服务一定要用内网的地址。正常情况下es的log延时不会超过10s.

1. es高版本安装不成功

怎么解决问题：

(1)一定要看log。80%

(2)查资料，一般我建议查官网，英文版。百度 csdn 都是转载过来 看原创的。200M

(3)逐步排查问题。

(4)ELK：es logstsh kibana。

1. **文本检索介绍**

百度：文本检索。like。千亿级 100TB,绝对不能用like。Like ‘sdf%’搜索引擎应运而生：高效率，存储大，支持tb级数据。

B+tree：nlogn（计算时间复杂度的代表，算法课程有讲的）

Hash索引：O(1) O表示时间要多久。O(1)表示查一次。

1. **IK分词**

**分词概念**:就是将一句话把他分词成词组

**英文情况下分词**：I like tuling：

I / like /tuling 比较简单就用空格分开。

**中文情况下分词**：我是一个中国人：

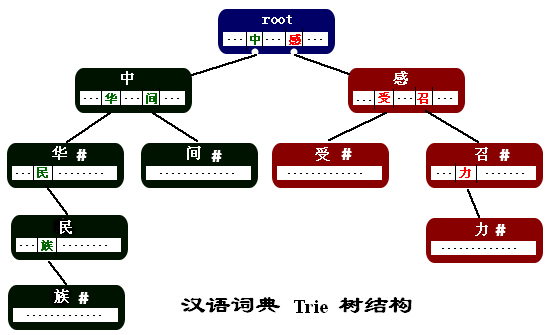
分词1:我/是/一个/中国/人

分词2：我/是/一个/中国人

武汉南京市长江大桥：  
 武汉/市长/江大桥

武汉市/长江大桥

**分词算法**:Tire：字典树。英文字典树：比较简单。双数组链表的概念。其算法设计的核心就是以空间换速度。可以去看我的字典树视频，里面讲了原理实现；字典树会把我们的词打散成字。形成的结构如下所示：



**实践步骤：**

1. IK配置分词器：主词典，停用词：建议网上找一下有一个非常好的停用词库。
2. 注意如果没有IKAnalyzer.cfg.xml文件 在自己的工程下加就可以。
3. 分词粒度：IKSegmenter ik = new IKSegmenter(new StringReader(str), true); True表示粗，false细
4. 建索引的时用细粒度
5. 搜索的时候用粗粒度

参考我的源码文件

**分词应用场景**

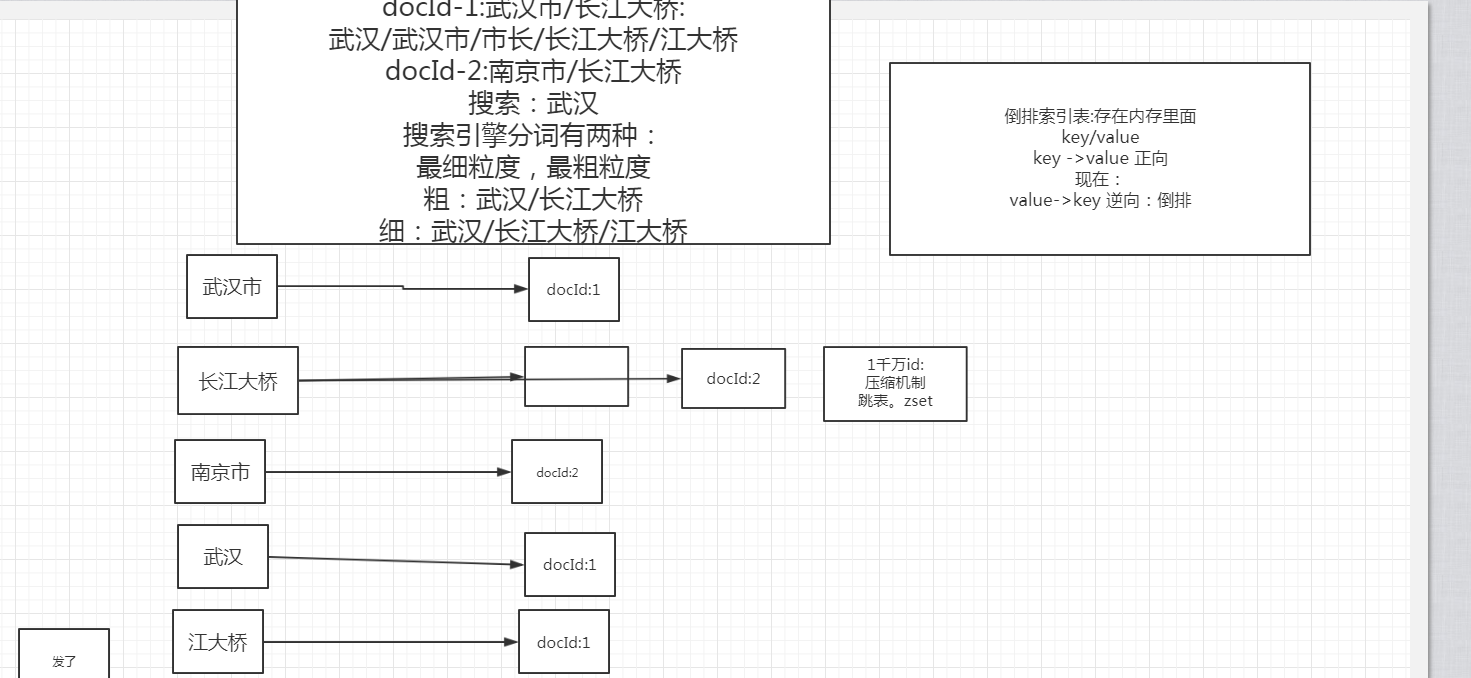
1. 敏感词过滤：只有我改写的源码才有效.网上自行下载的没有此功能
2. 做NLP。文本分析：
3. 配合like查询：10w like。但是like又不能匹配句子,因为句子中有很多无用词

比如：图灵的哈哈 图灵/哈哈

(Like ‘%图灵%’ or like ‘%哈哈%’)

1. **Lucene原理**

**倒排索引**:因为我们形成的是一个value->key的结构所以称之为倒排索引。



**分词对Lucene的意义**：

1. Lucene其核心就是一个倒排索引库，在建立索引的时候需要先将文本进行分词。再根据分好的词建立与文档的对应关系。
2. 分词的好坏直接决定lucene查询的 精准度
3. 分词还可以直接决定索引的大小。

**Es默认的分词**：一个个字的分开。内存爆炸。搜到很多没用的。

**Es和Lucene是什么关系？**

Es基于Lucene。只是在这基础上做出来一个分布式索引库，提供了集群的概念，还提供了api查询接口。Es和solr是一样的原理。

演示：

1000w 商品数据，要你做一个搜索功能。

分词加like不行（适用10w级）这个不必要用es。维护成本高 ，适用千万级以上，稳定性搞，查询量大。分布式查询。

最佳方案：分词+Lucene（单机版），不适用高并发。

1. **总结**

此节课重点为IK分词器和倒排索引原理.