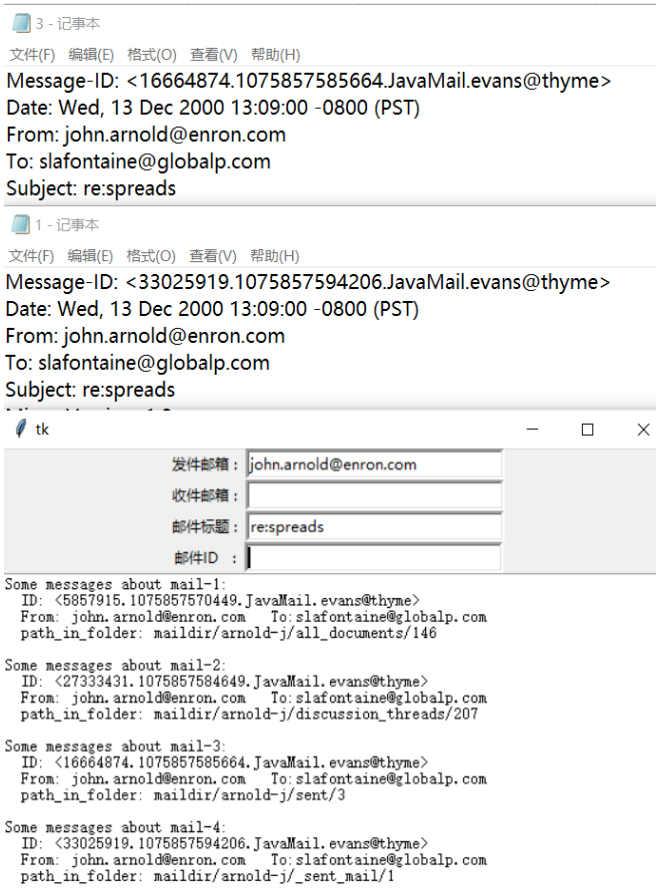
邮件检索系统实现

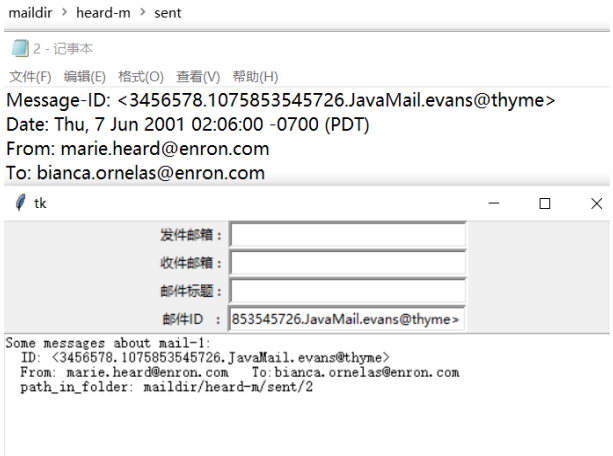
计算机科学与技术　1910378　范毓哲

一、实验代码与代码分析

这部分内容在 es-email.ipynb 文件中进行了展示与详细分析。

二、实验效果





最终程序实现了根据发件人、收件人、邮件标题和 ID 这四项信息，进行检索的功能。

邮件 ID（Message-ID）可以唯一地确定一封邮件。

如图所示，左图查询了 john.arnol 所发送的所有标题为 re:spreads 的邮件，上图则指定查询了特定 ID 下的邮件。

三、ES 中的优化

1. 首先，ES 基于 Lucene 构建，利用倒排索引将文件组织进字典，倒排索引是本课程最主要的数据结构，对提高查询速度有着十分显著的能效。
2. 引擎包 Lucene 在倒排索引的基础上，利用一种类似于前缀树的数据结构进行词项到字典的查询——有穷状态转换机 FST，它对于词项的公共前缀进行了提取，按典序组织成有序状态图，使得查询的时间性能和空间性能都有很大提升。
3. 缓存频率较高的过滤器。
4. Frame of Reference

对每个文档进行 δ-编码（该文档的编号值减去前一个文档的编号值），每 256 个文档

分一块，再算出这个块里占位最多的文档编号，以它为基准，决定头信息。这样一来，缩短了文档编号所需的位数，节省了空间，也更加方便查询。
（下图为 es 索引空间信息）

index	uuid	pri	rep	docs.count	docs.deleted	store.size
.geoip_databases	HkVfjQiCT16VPOheIHHNTw	1	0	42	4	40.6mb
test-index	rWFdaXkTT6-wZEKQhxkLsg	1	1	1	0	4.9kb
megacorp	h12tG1PCSTqeDHdrscSoIg	1	1	1	0	6.3kb
em-index1	i11NzXdKTGOGj1O1S5x1hg	1	1	517401	0	240.7mb
index1	gm4NLUx6S6m4-rbEI7pS4g	1	1	4	0	54kb
megacorp2	bH_V4hnCRry9fXojC3_IIQ	1	1	1	0	6.3kb

四、遇到的问题与解决

1. es 默认只显示 10 条数据，解决：在 query 中对“size”字段进行设置。
2. 问题：query = {"query": {"term":{"first_name" : "John"}}}，返回结果为空。
   解决：term 是基于词项的查询，不能区分大小写、不能查询特殊字符、不能查询含空白字符的内容。
   问题：query = {"query": {"match ":{ "From" : "phillip.allen@enron.com" }}}
   这样一条查询，会把所有邮件都返回，显然不是期待的答案，这是因为 es 将点号视为分隔，所有含 com 的邮件都会返回。如果用 term，则会无法匹配任何邮件。
   解决：用 match_parse，会匹配单词组合。
3. 问题：每次只能插入 15 条数据，解决：用 helpers 批量插入。
4. 有的文件是乱码，用常用方式都无法解码，解决：把它们跳过。
5. 如果在建立索引时导入的第一条 json 数据里包含 int 类型的字段，那么之后的数据，这部分都必须是 int（或可以转化为 int），这时直接删除索引比较好，es.indices.delete('em-index1')。

五、参考文献

[1] Elastic Stack Company. Elastic Stack and Product Documentation [EB/OL] . [2021-11-17] . https://www.elastic.co/guide/index.html.
[2] https://zhuanlan.zhihu.com/p/266116262