搜索引擎 之 倒排索引

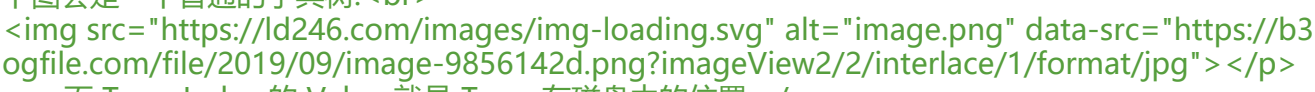
作者：eindex
原文链接：https://ld246.com/article/1567568659271
来源网站：链滴

世界上最伟大的互联网产品,说是搜索引擎,绝对没有别的产品可以替代,尤其是伟大的先在市场占率最高的搜索引擎,Google Search.
还有很多差一大截的,比如 Bing, Yahoo 和 YANDEX.
<h2 id="什么是搜索引擎">什么是搜索引擎</h2>
所谓搜索引擎，就是根据用户需求与一定算法，运用特定策略从互联网检索出制定信息反馈给用的一门检索技术。
搜索引擎技术的核心模块一般包括爬虫、索引、检索和排序等，同时可添加其他一系列辅助模块以为用户创造更好的网络使用环境。
<h2 id="搜索引擎干了些什么">搜索引擎干了些什么</h2>
简单的说搜索引擎从网络上爬取网页,然后对网页信息进行提取,构建正排索引,然后分析网页内容,立倒排文件.
接下来我将依次介绍 正排索引、 倒排索引 等知识点.
<h2 id="正排索引">正排索引</h2>
正排索引通常是
id-document 的键值对
<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>context</th>
<th>eng_context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>小明</td>
<td>今天吃了 3 个包子</td>
<td>today eating 3 baozi</td>
</tr>
<tr>
<td>2</td>
<td>小仓</td>
<td>最后一天看了一本书</td>
<td>read a book last day</td>
</tr>
<tr>
<td>3</td>
<td>仓颉</td>
<td>去年造了个新华字典</td>
<td>make a xinhua dictionary last year</td>
</tr>
</tbody>
</table>
<h2 id="倒排索引">倒排索引</h2>
倒排索引是关键词到 id 列表的映射关系.
倒排索引里面左边的列叫做 term , 里面存储的通常是一个关键词;
右边的列叫做 posting list,里面存储的是文档的 id.
左侧的一列一起叫做 term dictionary.
首先上面有每个 document 有 3 个 field , 所以会创建 3 个倒排索引.
如果使用默认的英文分词器,那么表格会创建的索引大概是下面这个样子:
name index:
<table>

<thead>
<tr>
<th>term</th>
<th>posting list</th>
</tr>
</thead>
<tbody>
<tr>
<td>小</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>仓</td>
<td>[2, 3]</td>
</tr>
<tr>
<td>明</td>
<td>[1]</td>
</tr>
<tr>
<td>颉</td>
<td>[3]</td>
</tr>
</tbody>
</table>
context index:
<table>
<thead>
<tr>
<th>term</th>
<th>posting list</th>
</tr>
</thead>
<tbody>
<tr>
<td>个</td>
<td>[1, 3]</td>
</tr>
<tr>
<td>了</td>
<td>[1, 2, 3]</td>
</tr>
<tr>
<td>今</td>
<td>[1]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
eng_context index:
<table>

<thead>
<tr>
<th>term</th>
<th>posting list</th>
</tr>
</thead>
<tbody>
<tr>
<td>last</td>
<td>[2, 3]</td>
</tr>
<tr>
<td>a</td>
<td>[2, 3]</td>
</tr>
<tr>
<td>dictionary</td>
<td>[3]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
<hr>
读到这里想来你也会有些许疑问?

比 MySQL Index 的优势在哪?
当有 1 亿 Term 时如何高效查询?
联合查询怎么办?
相关度排序?

<h2 id="Term-index">Term index</h2>
假象一下我们现在有 1 亿个 term, 那么这个 trem dictionary 会占多少内存呢?
假设每个 trem 占用内存 6 byte, 那么亿个大约占用 0.6 GB, 那么如果是一个非常复杂的文档, 岂不是存不够用?
而且为什么搜索引擎的速度远远快于 MySQL 的索引速度?
首先 Trem Index 实际上是一种字典树(前缀树), 可以将 term dictionary 里的内容组成树状结构.
br>
下图会是一个普通的字典树.


而 Term Index 的 Value 就是 Term 在磁盘中的位置.
而且 Trem Index 通常会被缓存到内存中, 再加上优秀的树结构, 所以查询速度极快, 字典树的数结构优势, 使体积远小于 term dictonary.
快于 MySQL 的主要原因是, 减少了访问磁盘的次数.MySQL 的 B+ 树会在磁盘中多次读写不同置, 而使用 term index , 则只会在匹配到 term 时进行一次磁盘访问.
<h2 id="联合查询">联合查询</h2>
如何进行索引的联合查询, 当程序获得多个 Posting list 那么程序需要从 list 中进行多次交集并操作.如何减少这些操作呢?
通常有两个解决方案.

bitset

跳表

可以自己去看一下相关的工作原理。
<h2 id="END">END</h2>
想必看完之后，你也对倒排索引工作原理有一定的了解了吧。感谢阅读！