Index Construction

Database Model,
Index Construction Process

Johannes Gosda, Gerald Töpper
## Index

<table>
<thead>
<tr>
<th>word</th>
<th>articleid</th>
<th>frequency_title</th>
<th>frequency_header</th>
<th>frequency_body</th>
<th>positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>307</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>1, 262, 322, 346, ...</td>
</tr>
<tr>
<td>lincoln</td>
<td>307</td>
<td>1</td>
<td>1</td>
<td>151</td>
<td>2, 31, 43, 66, ..., 263, ...</td>
</tr>
<tr>
<td>lincoln</td>
<td>620</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1001</td>
</tr>
</tbody>
</table>
Index Constructor

- Concurrency
  - Program
  - Threads
- Parsing
  - Apache Lucene Wikipedia Tokenizer
  - No markup, no footnotes
  - 'a-zA-Z' & '0-9' only
  - Stop word removal
  - Porter stemming algorithm
- Multiple index tables
### Configurable parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start article</td>
<td>Range of articles to index</td>
</tr>
<tr>
<td>Article count</td>
<td>Number of articles that should be fetched from database at once (per thread)</td>
</tr>
<tr>
<td>Articles per cycle</td>
<td>Number of threads</td>
</tr>
<tr>
<td>Table count</td>
<td>Number of tables the index is distributed on</td>
</tr>
</tbody>
</table>

- Low main memory usage
- 1 million articles
  - index with > 140 million records
  - 4.5 GiB Data
  - 17h creating time
Future prospects

- Better distribution of articles (to threads)
- Make articles per cycle depended from a given main memory value
- Dependable parser
- Individual stop word lists
- Support updates without complete new indexing