Concept Identification Using Lookup Approach
Mohammed Asif  Praveen Kumar  Senthil Madhappan

**Problem Statement**
- Design an engine to tag the concepts for the given text using lookup approach. The design should be scalable and update-friendly.
- List of all valid concepts reside on a HBase / NoSQL database and the initial set would be 10 million concepts.
- Benchmark the performance for tagging the document of size 250 words with concepts.

**Motivation**
Concept identification is becoming an area of great interest and has been extensively used in various applications, notably
- Information Retrieval & Extraction
  - Concept Based Indexing
  - Summarization
  - Question Answering
- Data Mining
  - Document Classification & Categorization
  - Ontology Engineering
  - Literature Based Knowledge Discovery

**High-level Solution Approach**
**Issues:**
- Loading the complete known concepts into memory is expensive in terms of memory footprint that gets added to the JVM heap.
- Lookup cost to check for a concept from known concepts.
- Number of lookups required to find an existing concept.

**Solution Approach:**
- Store the concepts on a central storage off the JVM heap.
  - The possible options would be HBase (or) NoSQL database like MongoDB
  - Time required for the lookup operation can be reduced by
    - Using local/distributed caches between the Hadoop task and the storage.
    - Identifying the words that will never form a concept and avoid lookup for those words e.g. stop words.

**Architecture**

**Performance Tuning**

<table>
<thead>
<tr>
<th>Concept Size</th>
<th>Execution Time (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
</tr>
<tr>
<td>7</td>
<td>700</td>
</tr>
</tbody>
</table>

**Dataset Used:**
- Seeded Concepts – 10 Million Wikipedia Titles
- Lookup Documents – 100 Gyananidhi Documents (English)

**Future Work**
- Experiment with MongoDB replica set configuration
  - Use local database server instead of single centralized server over network.
- Experiment with various database tuning strategies
  - Create additional index and/or change database schema.
  - Tuning database cache parameters.
- Experiment with fully-distributed Hadoop cluster environment.
- Experiment with distributed JVM cache.

**Hardware/Software Configuration**

<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>Host OS</th>
<th>Guest OS</th>
<th>Processor</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows 7 (64bit)</td>
<td>Ubuntu 12.04 (64bit)</td>
<td>2</td>
<td>1.5GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>Hadoop</th>
<th>Mongo DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridged Adapter</td>
<td>Pseudo-Distributed</td>
<td>Single Centralized Server</td>
</tr>
</tbody>
</table>