A fast approach for parallel deduplication on multicore processors

Guilherme Dal Bianco, Renata Galante, Carlos A. Heuser
Overview

- General Blocking
- MD-Approach Overview
- MapReduce Implementation
- Evaluation
- Discussion
# General Blocking

<table>
<thead>
<tr>
<th>DiscID</th>
<th>DiscName</th>
<th>Genre</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From The Cradle - Eric Clapton</td>
<td>Blues</td>
<td>1994</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
<td>1975</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The Beatles - A Hard Day’s Night</td>
<td>Blues</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Eric Clapton - From the Cradle</td>
<td>Blues</td>
<td>1995</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Beatles - A Hard Day’s Night</td>
<td>Rock</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Curtis Mayfield - Curtis</td>
<td>Soul</td>
<td>1970</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## General Blocking - Blocking Key

| DiscID | DiscName                                      | Genre | Year | ...
|--------|-----------------------------------------------|-------|------|------
| 1      | From The Cradle - Eric Clapton               | Blues | 1994 | ...  
| 2      | Marvin Gaye - Here, My Dear                  | Soul  | 1975 | ...  
| 3      | The Beatles - A Hard Day’s Night             | Blues | 1964 | ...  
| 4      | Eric Clapton - From the Cradle               | Blues | 1995 | ...  
| 5      | Beatles - A Hard Day’s Night                 | Rock  | 1964 | ...  
| 6      | Curtis Mayfield - Curtis                     | Soul  | 1970 | ...  
| ...    | ...                                           | ...   | ...  | ...  

## General Blocking - Balance Problem

<table>
<thead>
<tr>
<th>DiscID</th>
<th>DiscName</th>
<th>Genre</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From The Cradle - Eric Clapton</td>
<td>Blues</td>
<td>1994</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
<td>1975</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The Beatles - A Hard Day’s Night</td>
<td>Blues</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Eric Clapton - From the Cradle</td>
<td>Blues</td>
<td>1995</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Beatles - A Hard Day’s Night</td>
<td>Rock</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Curtis Mayfield - Curtis</td>
<td>Soul</td>
<td>1970</td>
<td></td>
</tr>
</tbody>
</table>
## General Blocking - Keys Problem

<table>
<thead>
<tr>
<th>DiscID</th>
<th>DiscName</th>
<th>Genre</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From The Cradle - Eric Clapton</td>
<td>Blues</td>
<td>1994</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
<td>1975</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The Beatles - A Hard Day’s Night</td>
<td>Blues</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Eric Clapton - From the Cradle</td>
<td>Blues</td>
<td>1995</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Beatles - A Hard Day’s Night</td>
<td>Rock</td>
<td>1964</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Curtis Mayfield - Curtis</td>
<td>Soul</td>
<td>1970</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Blocking Functions & Multipass

- blocking functions are defined as followed:
  - \(bf_1(\text{record}) = \{\text{genre}\}\)
  - \(bf_2(\text{record}) = \{\text{year, genre}\}\)
  - \(bf_3(\text{record}) = \{1^{\text{st}} 3 \text{ letters of genre, } 1^{\text{st}} 3 \text{ digits of year}\}\)

- in a n-multipass several blocking functions are applied to each record
  - \(BFS = \{bf_1, bf_2, ..., bf_n\}\)
MD-Approach - Idea

Blocking Step
MD-Approach - Idea

Blocking Step

Match

D → B₁ → M
D → B₂ → M
D → B₃ → M
D → B₄ → M
MD-Approach - Idea

D → B₁ → M
D → B₂ → M
D → B₃ → B₃,₁ → M
D → B₃ → B₃,₂ → M
D → B₄ → M

Blocking Step

MD-Approach

Match
MD-Approach - Idea

Blocking Step → MD-Approach → Match

D → B_1 → M → M → M → M

D → B_2 → M → M → M → M

D → B_3 → M → M → M → M

D → B_4 → M → M → M → M
MD-Approach - MapReduce Overview
Map-Reduce Implementation
Phase I - First Blocking Step

- create dataset segments
- only map phase
- emits key-value pair
  - generated blocking key as key, e.g. \(bf(\text{record}) = \{1^{\text{st}} 3 \text{ letters of genre, } 1^{\text{st}} 3 \text{ digits of year}\}\)
  - record as value

<table>
<thead>
<tr>
<th></th>
<th>Marvin Gaye - Here, My Dear</th>
<th>Soul</th>
<th>1975</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Sou197</td>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
</tr>
</tbody>
</table>
Map-Reduce Implementation
Phase I - First Blocking Step

- multi-passing
  - set of n several blocking functions
    - BFS = \{bf_1, bf_2, ..., bf_n\}
  - for each record emit at once:
    - \(<k_{bf1} : record_1> \ ... \ <k_{bf1} : record_n>\)
    - \(<k_{bf2} : record_1> \ ... \ <k_{bf2} : record_n>\)
    - \(<k_{bfn} : record_1> \ ... \ <k_{bfn} : record_n>\)

<table>
<thead>
<tr>
<th>2</th>
<th>Marvin Gaye - Here, My Dear</th>
<th>Soul</th>
<th>1975</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sou197</td>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
<td>1975</td>
</tr>
<tr>
<td>MarvSou</td>
<td>2</td>
<td>Marvin Gaye - Here, My Dear</td>
<td>Soul</td>
<td>1975</td>
</tr>
</tbody>
</table>
Map-Reduce Implementation
Phase II - Sort Blocks & Match

- identify unbalanced blocks
  - compare the record count of each block with a threshold
  - use reduce function until a certain threshold is reached

- reduce step (match step)
  - receives all records with the same key (here same block)
  - nested-loop pairwise comparing
  - outputs pairs of similar records
Map-Reduce Implementation
Phase III - Second Blocking Step

- only unbalanced blocks
- map: expand blocking key from first blocking step
  - e.g. $bf_1(record) = \{1^{st} 3 \text{ letters of genre}, 1^{st} 3 \text{ digits of year}\} \rightarrow bf_1'(record) = \{\text{all letters of genre, all digits of year}\}$
  - creates very fine granular blocks

<table>
<thead>
<tr>
<th>Blu199</th>
<th>1</th>
<th>From The Cradle - Eric Clapton</th>
<th>Blues</th>
<th>1994</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blu199</td>
<td>4</td>
<td>Eric Clapton - From the Cradle</td>
<td>Blues</td>
<td>1995</td>
<td>...</td>
</tr>
<tr>
<td>Blues1994</td>
<td>1</td>
<td>From The Cradle - Eric Clapton</td>
<td>Blues</td>
<td>1994</td>
<td>...</td>
</tr>
<tr>
<td>Blues1995</td>
<td>4</td>
<td>Eric Clapton - From the Cradle</td>
<td>Blues</td>
<td>1995</td>
<td>...</td>
</tr>
</tbody>
</table>
Map-Reduce Implementation
Phase III - Second Blocking Step

- to avoid loss of true positives use 'sliding window approach'
  - create an index structure for fine-grained keys after map phase
  - compare with k-nearest neighbors
  - if the similarity is high enough merge records with very similar keys to bigger blocks again
- reduce step (match) is same as in Phase II
Map-Reduce Implementation
Phase IV - Merge Pairs

- short map-reduce operations to clean output file
  - identify and remove replicated pairs
  - multipass generates duplicates of detected records
Evaluation

- Phoenix MR framework was used for implementation - shared memory-architecture
- synthetic dataset generated by Febrl (1M, 2M, 4M, each with 10% duplicates)
- compared with BTO-BK
- used different similarity metrics for different approaches
Relevance for the seminar

- interesting and intuitive main idea
- due to weaknesses in English language, sometimes hard to understand
- the MR-specific implementation details are very rare
- the mapping from a shared-memory (Phoenix) onto a shared-nothing (Hadoop, Stratosphere) architecture will be challenging
- to sum best things up:
  - single-run multi-pass
  - load balancing through re-blocking
Sources

Map-Reduce Implementation
First MR-Step

- map-step
  - emits (blocking-key, value)

- identify unbalanced blocks

- reduce-step (balanced blocks only)
  - similarity function
  - arithmetic average
  - find duplicate by threshold
Map-Reduce Implementation

Second MR-Step

- **map-step**
  - emits expanded blocking-key

- "sliding window sort" (binary search)

- **reduce-step**
  - same as in First MR-Step