Enabling Efficient Random Access to Hierarchically-Compressed Data

Feng Zhang †, Jidong Zhai ◊, Xipeng Shen #, Onur Mutlu ⋆, Xiaoyong Du †

†Renmin University of China
◊Tsinghua University
#North Carolina State University
⋆ETH Zürich
Outline

1. Background
2. Motivation
3. Operations to Support
4. Challenges
5. Our Solution
6. Evaluation
7. Conclusion
1. Background

• TADOC: Text Analytics Directly on Compression

Input:

file0: w1 w2 w3 w1 w2 w4
   w1 w2 w3 w1 w2 w4

file1: w1 w2 w1

(a) Original data

Rules:

R0 \rightarrow R1 R1 SPT1 R2 w1
R1 \rightarrow R2 w3 R2 w4
R2 \rightarrow w1 w2

(b) TADOC compressed data

(d) Numerical representation

w1: 0  w2: 1  w3: 2  w4: 3  R0: 4  R1: 5  R2: 6  SPT1: 7

4 \rightarrow 5 5 7 6 0
5 \rightarrow 6 2 6 3
6 \rightarrow 0 1

(e) Compressed data in numerical form

(c) DAG Representation

1. Background

• Example: word count

Step 1

<R0: R1 R1 SPT1 R2 w1>

<w1,1>, <w2,1>

Step 2

<w1,2>, <w2,2>
<w3,1>, <w4,1>

R1: R2 w3 R2 w4

Step 3

<w1,6>, <w2,5>, <w3,2>, <w4,2>

R2: w1 w2

R1: 2 × 2 + 1 + 1 = 6
R2: 2 × 2 + 1 = 5
R3: 1 × 2 = 2
R4: 1 × 2 = 2

<w1,1>, <w2,1>

CFG Relation

Information Propagation

Word table
2. Motivation

- search
- extract
- count
- insert
- append
3. Operations to Support

- Random Access
  - `extract(file, offset, length)`
  - `search(file, word)`
  - `count(file, word)`

- Compatibility
  - `insert(file, offset, string)`
  - `append(file, string)`

Locality

Compatibility

User Transparency
4. Challenges

• Hierarchical Structure of the DAG
  • Example: R2 belongs to both file0 and file1

• Uni-Directionality
  • Edges
4. Challenges

• Special Complexities on Insert
  • Example: insertion to R2

• Tradeoff between Space Savings and Time Cost
  • Index space cost
5. Our Solution

Solution Techniques: Five Data Structures

Challenges: Four Sources of Complexity

Five Operations to Enable Random Access on Compressed Data

- extract
- search
- count
- insert
- append

Direct processing on compressed data

- hierarchical structure of the DAG
- uni-directionality
- special complexities on insert
- tradeoff between space savings & time cost

Solution Techniques:
- rule2location
- word2rule
- rootOffset
- bitmap
- records
5. Our Solution

• Relations for data structures
5. Our Solution

- `extract(file, offset, length)`

```
Offset: 0
Rule: R2
Start: 0   End: 1

Offset: 4
Rule: R1
Start: 1   End: 1

Offset: 6
Rule: R2
Start: 0   End: 1

Offset: 10
Rule: R1
Start: 3   End: 3
```

```
R0: R1  R1  SPT1  R2  w1

R1: R2  w3  R2  w4

R2:  w1  w2
```
5. Our Solution

- search(file, word)
  - word2rule
  - rule2location

- count(file, word)
  - word2rule
  - rule2location

Illustration of rule2location optimization.
5. Our Solution

- `insert(file,offset,string)`
- `append(file,string)`

**The Record Data Structure**

```c
struct Record{
    int fileId; // file, such as file1
    int fileOffset; // file offset to insert, such as 100
    int ruleID; // the rule ID to insert, such as 0
    int ruleLocation; // the inserted location, such as 2
    int replaceWord; // the replaced word, such as w2
    string content; // content string
    int ptr; // the recordID inserted at the same place. Default is -1
    int ruleStartOffset; // the starting offset of the rule to insert, such as 0
};
```
6. Evaluation

- Five operations
  - search, extract, count, insert, append
- Five datasets
  - 580 MB ~ 300 GB
- Two platforms
  - Single node
  - Spark cluster (10 nodes on Amazon EC2)
6. Evaluation

- On average, the overall throughput of our proposed techniques (CD) is $3.1 \times$ of Succinct’s throughput in a distributed environment.
6. Evaluation

• The average compression ratio we observe is 3.9, which is still much more compact than the 1.8 compression ratio of Succinct.

• compression ratio = original size / compressed data size
7. Conclusion

• We provide a set of new techniques that enable efficient random access operations on hierarchically-compressed data

• Compatible with TADOC: data traversal operations

• We remove a major barrier against practical adoption of direct text analytics on compressed data.
Thanks!

• Any questions?

Feng Zhang †, Jidong Zhai ◊, Xipeng Shen #, Onur Mutlu ※, Xiaoyong Du †

†Renmin University of China
◊Tsinghua University
#North Carolina State University
※ETH Zürich