Efficient Document Analytics on Compressed Data: Method, Challenges, Algorithms, Insights

Feng Zhang †◊, Jidong Zhai ◊, Xipeng Shen #, Onur Mutlu ★, Wenguang Chen ◊

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

• Every day, 2.5 quintillion bytes of data created – 90% data in the world today has been created in the last two years alone[1].

[1] What is Big Data?
https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html
Outline

• **Introduction**
  • Motivation & Example
  • Compression-Based Direct Processing
  • Challenges

• **Guidelines and Techniques**
  • Solution Overview
  • Guidelines

• **Evaluation**
  • Benchmarks
  • Results

• **Conclusion**
Motivation

How to perform efficient document analytics when data are extremely large?

• Challenge 1:
  • SPACE: Large Space Requirement

• Challenge 2:
  • TIME: Long Processing Time
Motivation

• Observation
  • Using Hash Table to check redundant content for Wikipedia dataset

Whether we can perform analytics directly on compressed data?
Our Idea

- Compression-based direct processing
- *Sequitur* algorithm meets our requirement

Input:

```
abcabdabcabd
aba
```

Rules:

- \( R_0 \rightarrow R_1 \)  \( R_1 \)  \( R_2 \)  a
- \( R_1 \rightarrow R_2 \)  c  \( R_2 \)  d
- \( R_2 \rightarrow a \)  b

(a) Original data  (b) Sequitur compressed data  (c) DAG Representation

Numerical representation:

- a: 0  b: 1  c: 2  d: 3
- R0: 4  R1: 5  R2: 6

(d) Numerical representation

Compressed data in numerical ID:

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

(e) Compressed data in numerical ID
Double benefits

- Challenge 1: Space
  - Appear more than once, but only store once!

- Challenge 2: Time
  - Appear more than once, but only compute once!
Optimization

• We can make it more compact.

Some applications do not need to keep the sequence.

In each rule, we may remove sequence info too.

Further saves storage space and computation time.
Example

• Word Count

\[ \begin{align*}
\langle a, 2 \times 2 + 1 + 1 \rangle &= \langle a, 6 \rangle \\
\langle b, 2 \times 2 + 1 \rangle &= \langle b, 5 \rangle \\
\langle c, 1 \times 2 \rangle &= \langle c, 2 \rangle \\
\langle d, 1 \times 2 \rangle &= \langle d, 2 \rangle
\end{align*} \]
Challenges

- How to organize data.
- How to accommodate the order for applications that are sensitive to the order.
- How to perform parallelism on large datasets.
- How to utilize the attributes of datasets.

CHALLENGES

- Unit sensitivity
- Parallelism barriers
- Data attributes
- Overhead in saving and propagating
- Reuse of results across nodes

Order sensitivity
Outline

• Introduction
  • Motivation & Example
  • Compression-Based Direct Processing
  • Challenges

• Guidelines and Techniques
  • Solution Overview
  • Guidelines

• Evaluation
  • Benchmarks
  • Results

• Conclusion
Solution Overview

SOLUTION TECHNIQUES

Adaptive traversal order and information to propagation

Compression-time indexing

Two-level table with depth-first traversal

Double compression

Double-layered bit vector for footprint minimization

Coarse-grained parallel algorithm and automatic data partition

CHALLENGES

Unit sensitivity

Parallelism barriers

Order sensitivity

Data attributes

Reuse of results across nodes

Overhead in saving and propagating

Reuse of results across nodes

Order sensitivity

Data attributes

Overhead in saving and propagating

Unit sensitivity

Parallelism barriers
Data Attributes

Problem: How to utilize the attributes of datasets

• Guideline I: minimize the footprint size
• Guideline II: Traversal order is essential for the efficiency

The best traversal order may depend on the data attributes of input.
Parallelism Barriers
Problem: How to perform parallelism on large datasets

- Guideline III: Coarse-grained distributed implementation is preferred
Order Sensitivity (detailed in paper)
Problem: Some applications are sensitive to the order

- Guideline IV: depth-first traversal and a two-level table design
Unit Sensitivity (detailed in paper)
Problem: How to organize data

- Guideline V: use of double-layered bitmap if unit information needs to be passed across the CFG
Short Summary for Six Guidelines

• Data Attributes Challenge
  • Guideline II

• Parallelism Barriers
  • Guideline III

• Order Sensitivity
  • Guideline IV

• Unit Sensitivity
  • Guideline V

• General insights and common techniques
  • Guideline I and Guideline VI
Outline

• Introduction
  • Motivation & Example
  • Compression-Based Direct Processing
  • Challenges

• Guidelines and Techniques
  • Solution Overview
  • Guidelines

• Evaluation
  • Benchmarks
  • Results

• Conclusion
Benchmarks

• Six benchmarks
  • Word Count, Inverted Index, Sequence Count, Ranked Inverted Index, Sort, Term Vector

• Five datasets
  • 580 MB ~ 300 GB

• Two platforms
  • Single node
  • Spark cluster (10 nodes on Amazon EC2)
Time Savings

• *CompressDirect* yields 2X speedup, on average, over direct processing.
Space Savings

- *CompressDirect* achieves 11.8X compression ratio, even more than *gzip* does.

Compression ratio = original size / compressed data size
Conclusion

• Our method, *compression-based direct processing*.

• How the concept can be materialized on Sequitur.
  • Major challenges.
  • Guidelines.

• Our library, *CompressDirect*, to help further ease the required development efforts.
Thanks!

• Any questions?

Feng Zhang †○, Jidong Zhai ○, Xipeng Shen #, Onur Mutlu ★, Wenguang Chen ○

†Renmin University of China
○Tsinghua University
#North Carolina State University
★ETH Zürich