G-TADOC: Enabling Efficient GPU-Based Text Analytics without Decompression

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. Challenges
4. Our Solution
5. Evaluation
6. 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

(d) Numerical representation
- w1: 0 w2: 1 w3: 2 w4: 3 R0: 4 R1: 5 R2: 6 SPT1: 7

(b) TADOC compressed data

Rules:
- R0 → R1
- R1 SPT1 R2
- w1
- R1 → R2
- w3 w2
- R2 → w1 w2

(c) DAG Representation

(e) Compressed data in numerical form
- 4 → 5 5 7 6 0
- 5 → 6 2 6 3
- 6 → 0 1

1. Background

• Example: word count

- Step 1: \(<w1,1>, <w2,1>, <w1,6>, <w2,5>, <w3,2>, <w4,2>\)
- Step 2: \(w1: 1 \times 2 = 2\), \(w2: 1 \times 2 = 2\), \(w3: 1\), \(w4: 1\)
- Step 3: \(w1: 2 \times 2 + 1 + 1 = 6\), \(w2: 2 \times 2 + 1 = 5\), \(w3: 1 \times 2 = 2\), \(w4: 1 \times 2 = 2\)
2. Motivation

• GPU – popular in data science
• limited GPU memory
• different from CPUs
• Previous GPU-based methods does not apply
3. Challenges

- Challenge 1: GPU parallelism for TADOC
  - Example: R2 depends on R0 and R1, and R1 depends on R0

- dependencies
  - Edges
3. Challenges

- Challenge 2: TADOC final result update conflict of massive GPU threads
  - Example: writing to a global buffer

```
<table>
<thead>
<tr>
<th>thread</th>
<th>global buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w1</td>
</tr>
<tr>
<td></td>
<td>w2</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>w_i</td>
</tr>
</tbody>
</table>
```

write conflicts
3. Challenges

• Challenge 3: sequence maintenance of TADOC compressed data on GPUs
  • Example: cross-rule sequence

• sequence:
  • w1-w2-w3
    • R1 and R2
4. Our Solution

G-TADOC

- TADOC compressed data
- TADOC program
- results

modules

- data structures
- parallel execution engine
- sequence support

phases

- initialization
  - data structure preparation
  - light-weight scanning
- graph traversal
  - top-down traversal / bottom-up traversal
  - result merging

TADOC program

- light-weight scanning
- top-down traversal / bottom-up traversal
- result merging
4. Our Solution

- Fine-Grained Thread-Level Execution Engine

- Top-down / bottom-up traversal design
4. Our Solution

• **G-TADOC Data Structures**
  • G-TADOC maintained memory pool
  • Thread-safe data structures
  • Head and tail structures for sequence support

<table>
<thead>
<tr>
<th>Locks</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entries</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Keys</td>
<td>126</td>
<td>163</td>
<td>78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Next</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(d) Add key = 78 (suppose hash to 1), and value = 1.
4. Our Solution

- **Sequence Support in G-TADOC**
  - Phase 1: initialization for head and tail buffers
  - Phase 2: graph traversal with sequence support
5. Evaluation

- Six benchmarks
  - Word Count, Inverted Index, Sequence Count, Ranked Inverted Index, Sort, Term Vector from [1]

- Five datasets
  - 62 MB ~ 50 GB

- Four platforms
  - three generations of Nvidia GPUs
    - Pascal, Volta, and Turing micro-architectures
  - Spark cluster (10 nodes on Amazon EC2)

5. Evaluation

- On average, G-TADOC achieves $31.1 \times$ speedup over TADOC.
5. Evaluation

- Speedups in different phases

(a) Phase 1: initialization.
(b) Phase 2: traversal.
5. Evaluation

• Analysis of performance metrics

(a) DRAM throughput.

(b) Achieved occupancy.
6. Conclusion

• G-TADOC, the first framework enabling efficient GPU-based text analytics directly on compressed data

• Our work can help put much larger content directly in GPU memory.
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