Reducing Replication Bandwidth for Distributed Document Databases

Lianghong Xu¹, Andy Pavlo¹, Sudipta Sengupta²
Jin Li², Greg Ganger¹
Carnegie Mellon University¹, Microsoft Research²
The Peer and the Peri is a comic [[Gilbert and Sullivan]] [[operetta ]] in two acts... just as predicted,...The fairy Queen, however, appears to ... all live happily ever after."
Replication Bandwidth

Primary Database

Operation logs

WAN

Secondary

Operation logs

Secondary

Secondary

{  
"_id": "55ca4cf7bad4f75b8eb5c25d",  
"pageId": "46780",  
"revId": "4128520",  
"timestamp": "2002-03-30T20:06:22",  
"sha1": "6i81h1zt22u1w4sfxoofyzmxd",  
"text": "The Peer and the Peri is a comic [[Gilbert and Sullivan]] in two acts... just as predicting...The fairy Queen, however, appears to... all live happily ever after."
}

{  
"_id": "55ca4cf7bad4f75b8eb5c25c",  
"pageId": "46780",  
"revId": "41173",  
"timestamp": "2002-03-30T20:06:22",  
"sha1": "6i81h1zt22u1w4sfxoofyzmxd",  
"text": "The Peer and the Peri is a comic [[Gilbert and Sullivan]] in two acts... just as predicting...The fairy Queen, however, appears to... all live happily ever after."
}
Replication Bandwidth

Goal: Reduce WAN bandwidth for geo-replication

The Peer and the Peri is a comic [[Gilbert and Sullivan]] in two acts… just as predicted,…The fairy Queen, however, appears to … all live happily ever after. 

The Peer and the Peri is a comic [[Gilbert and Sullivan]] in two acts… just as...
Why Deduplication?

• Why not just **compress**?
  – Oplog batches are small and not enough overlap

• Why not just use **diff**?
  – Need application guidance to identify source

• **Dedup** finds and removes redundancies
  – In the entire data corpus
Traditional Dedup: Ideal

Incoming Data

{BYTE STREAM}

Deduped Data

Send dedup’ed data to replicas
Traditional Dedup: Reality

Chunk Boundary  Modified Region  Duplicate Region

Incoming Data

Deduped Data
Traditional Dedup: Reality

Chunk Boundary  Modified Region  Duplicate Region

Incoming Data

Deduped Data

Send almost the entire document.
Similarity Dedup (sDedup)

- Chunk Boundary
- Modified Region
- Duplicate Region

Incoming Data

Dedup’ed Data

Only send delta encoding.
Compress vs. Dedup

20GB sampled Wikipedia dataset
MongoDB v2.7 // 4MB Oplog batches
sdedup Integration

Client

Insertion & Updates

Oplog

Source documents

Database

Source Document Cache

sdedup Encoder

Dedup’ed oplog entries

Oplog syncer

Reconstructed oplog entries

Replay

Database

Source documents

sdedup Decoder

Oplog

unsynchronized oplog entries

Oplog batch
sDedup Encoding Steps

- Identify Similar Documents
- Select the Best Match
- Delta Compression
Identify Similar Documents

Target Document

Rabin Chunking

Consistent Sampling

Similarity Sketch

Feature Index Table

Candidate Documents

Similarity Score

1

Doc #1

2

Doc #2

Doc #3

13
Select the Best Match

Initial Ranking

<table>
<thead>
<tr>
<th>Rank</th>
<th>Candidates</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Doc #2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Doc #3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Doc #1</td>
<td>1</td>
</tr>
</tbody>
</table>

Final Ranking

<table>
<thead>
<tr>
<th>Rank</th>
<th>Candidates</th>
<th>Cached?</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Doc #3</td>
<td>Yes</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>Doc #1</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Doc #2</td>
<td>No</td>
<td>2</td>
</tr>
</tbody>
</table>

Is doc cached?

Source Document Cache

If yes, reward +2
Evaluation

• MongoDB setup (v2.7)
  – 1 primary, 1 secondary node, 1 client
  – Node Config: 4 cores, 8GB RAM, 100GB HDD storage

• Datasets:
  – Wikipedia dump (20GB out of ~12TB)
  – Additional datasets evaluated in the paper
Compression

Compression Ratio

Chunk Size

20GB sampled Wikipedia dataset
Memory

- **sDedup**
- **trad-dedup**

20GB sampled Wikipedia dataset

<table>
<thead>
<tr>
<th>Chunk Size</th>
<th>Memory (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4KB</td>
<td>34.1</td>
</tr>
<tr>
<td>1KB</td>
<td>80.2</td>
</tr>
<tr>
<td>256B</td>
<td>133.0</td>
</tr>
<tr>
<td>64B</td>
<td>57.3</td>
</tr>
<tr>
<td></td>
<td>272.5</td>
</tr>
<tr>
<td></td>
<td>780.5</td>
</tr>
</tbody>
</table>

- 61.0
Other Results (See Paper)

- Negligible client performance overhead
- Failure recovery is quick and easy
- Sharding does not hurt compression rate
- More datasets
  - Microsoft Exchange, Stack Exchange
Conclusion & Future Work

• **sDedup**: Similarity-based deduplication for replicated document databases
  – Much greater data reduction than traditional dedup
  – Up to 38x compression ratio for Wikipedia
  – Resource-efficient design with negligible overhead

• Future work
  – More diverse datasets
  – Dedup for local database storage
  – Different similarity search schemes (e.g., super-fingerprints)