#### Reducing Replication Bandwidth for Distributed Document Databases

#### Lianghong Xu<sup>1</sup>, Andy Pavlo<sup>1</sup>, Sudipta Sengupta<sup>2</sup> Jin Li<sup>2</sup>, Greg Ganger<sup>1</sup> Carnegie Mellon University<sup>1</sup>, Microsoft Research<sup>2</sup>



#### **Document-oriented Databases**

Update

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

"\_id" : "55ca4cf7bad4f75b8eb5c25d". "pageld" : "46780", "revld" : "128520"] "timestamp" : [2002-03-30T20:11:12", "sha1" : [q08x58kbjmyljj4bow3e903uz" "text" : "The Peer and the Perriss a comic [[Gilbert and Sullivan]] [[operetta ]] in two acts... just as predicted, ..The fairy Queen, on the othe hand, is "not" happy, and appears to ... all live happily ever atter. "

#### Update: Reading a recent doc and writing back a similar one

## **Replication Bandwidth**



# 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: Reality**

Chunk Boundary

Modified Region

Duplicate Region



# **Similarity Dedup**

Chunk Boundary

Modified Region

Duplicate Region





## **sDedup: Similarity Dedup**



# **sDedup Encoding Steps**

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



#### Select the Best Match **Initial Ranking Final Ranking** Rank Candidates Rank Candidates Cached? Score Score Doc #2 **Doc #3** 2 Yes Doc #3 Doc #1 2 Yes 3 Doc #2 Doc #1 2 2 No 2 \*\* If yes, reward +2 Is doc cached? Source Document 12 Cache

## **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





20GB sampled Wikipedia dataset



20GB sampled Wikipedia dataset

# **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)

## **Backup Slides**

#### **Compression: StackExchange**



10GB sampled StackExchange dataset

# Memory: StackExchange

sDedup

trad-dedup



10GB sampled StackExchange dataset

## **Throughput Overhead**



## **Failure Recovery**



# **Dedup + Sharding**



# **Delta Compression**

- Byte-level diff between source and target docs:
  - Based on the xDelta algorithm
  - Improved speed with minimal loss of compression

#### • Encoding:

Descriptors about duplicate/unique regions + unique bytes

#### • Decoding:

- Use source doc + encoded output
- Concatenate byte regions in order