Reducing Replication Bandwidth for Distributed Document Databases

Lianghong Xu[1], Andy Pavlo[1], Sudipta Sengupta[2]
Jin Li[2], Greg Ganger[1]
Carnegie Mellon University[1], Microsoft Research[2]

Document-oriented Databases

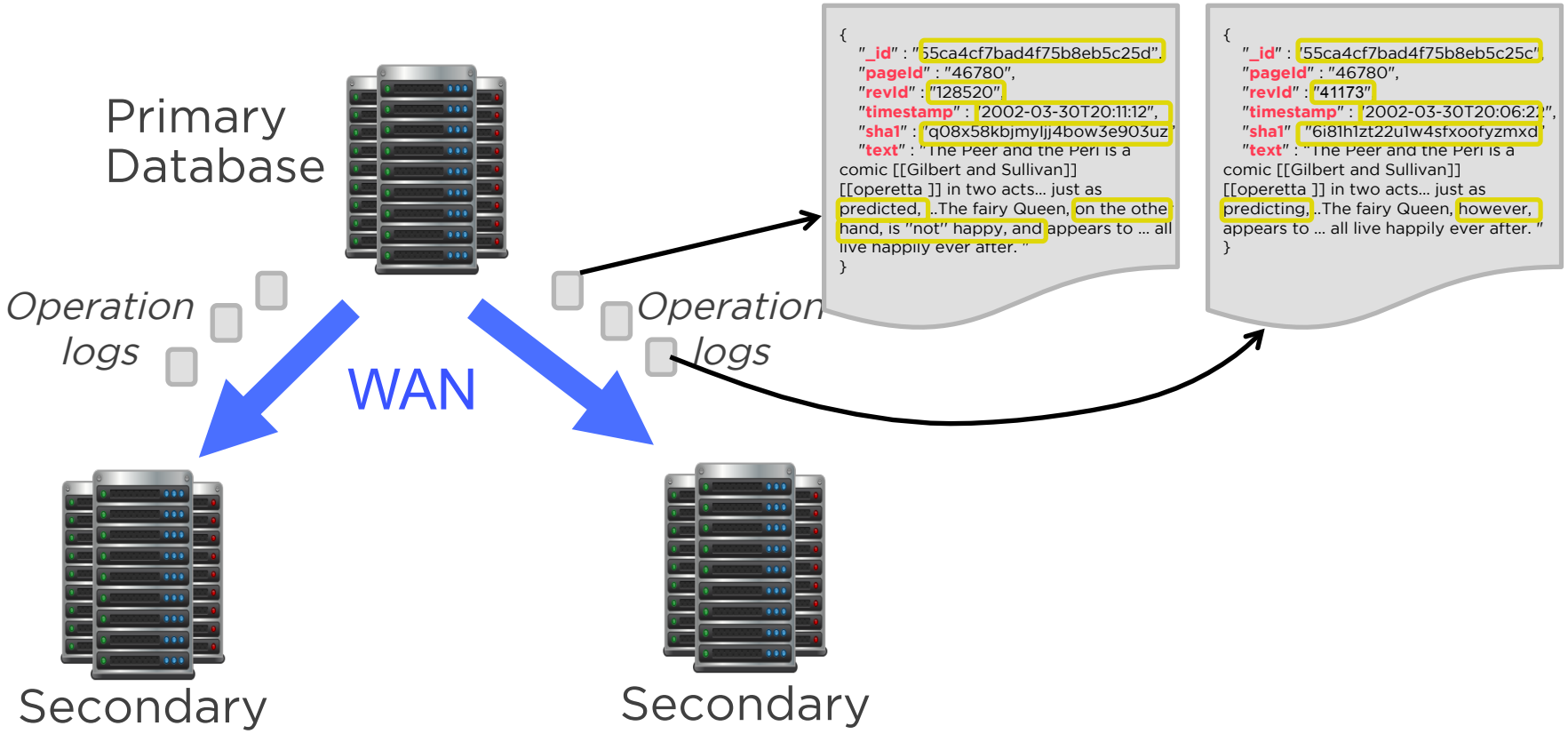

```
{
    "_id" : "55ca4cf7bad4f75b8eb5c25c",
    "pageId" : "46780",
    "revId" : "41173",
    "timestamp" : "2002-03-30T20:06:22",
    "sha1" : "6i81h1zt22u1w4sfxoofyzmxd"
    "text" : " The Peer and the Peri is a
comic [[Gilbert and Sullivan]]
[[operetta ]] in two acts... just as
predicting, ..The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


Update


```
{
    "_id" : "55ca4cf7bad4f75b8eb5c25d",
    "pageId" : "46780",
    "revId" : "128520",
    "timestamp" : "2002-03-30T20:11:12",
    "sha1" : "q08x58kbjmyljj4bow3e903uz"
    "text" : " The Peer and the Peri is a
comic [[Gilbert and Sullivan]]
[[operetta ]] in two acts... just as
predicted, ..The fairy Queen, on the other
hand, is "not" happy, and appears to ... all
live happily ever after. "
}
```

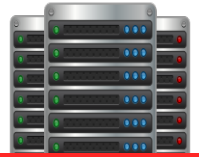

Update: Reading a recent doc and writing back a similar one

Replication Bandwidth



Replication Bandwidth

Primary Database




```
{
   "_id" : "55ca4cf7bad4f75b8eb5c25d",
   "pageId" : "46780",
   "revId" : "128520",
   "timestamp" : "2002-03-30T20:11:12",
   "sha1" : "q08x58kbjmyljj4bow3e903uz",
   "text" : "The Peer and the Peri is a
comic [[Gilbert and Sullivan]]
[[operetta ]] in two acts... just as
```

```
{
   "_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]]
[[operetta ]] in two acts... just as
```


Goal: Reduce WAN bandwidth for geo-replication

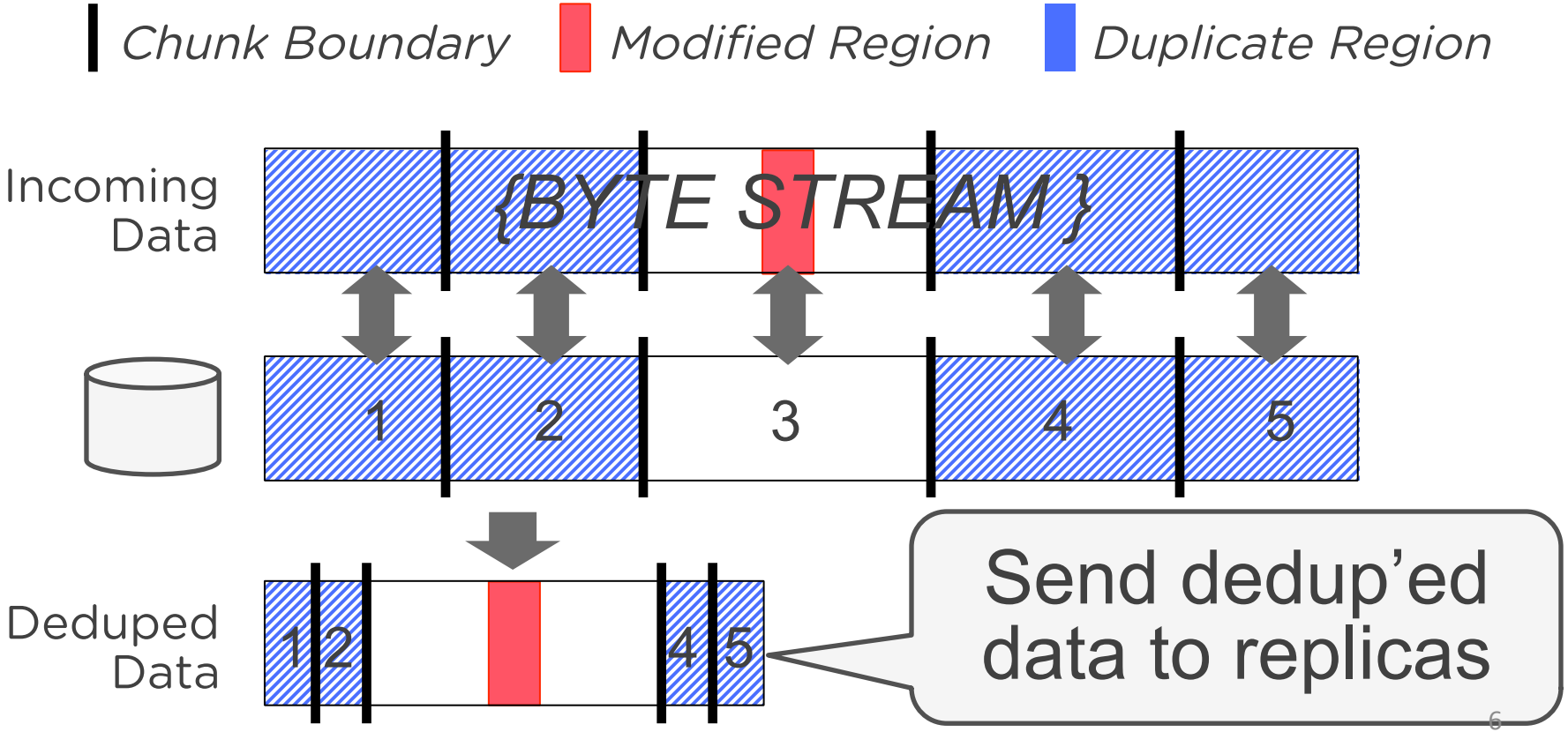
Secondary

Secondary

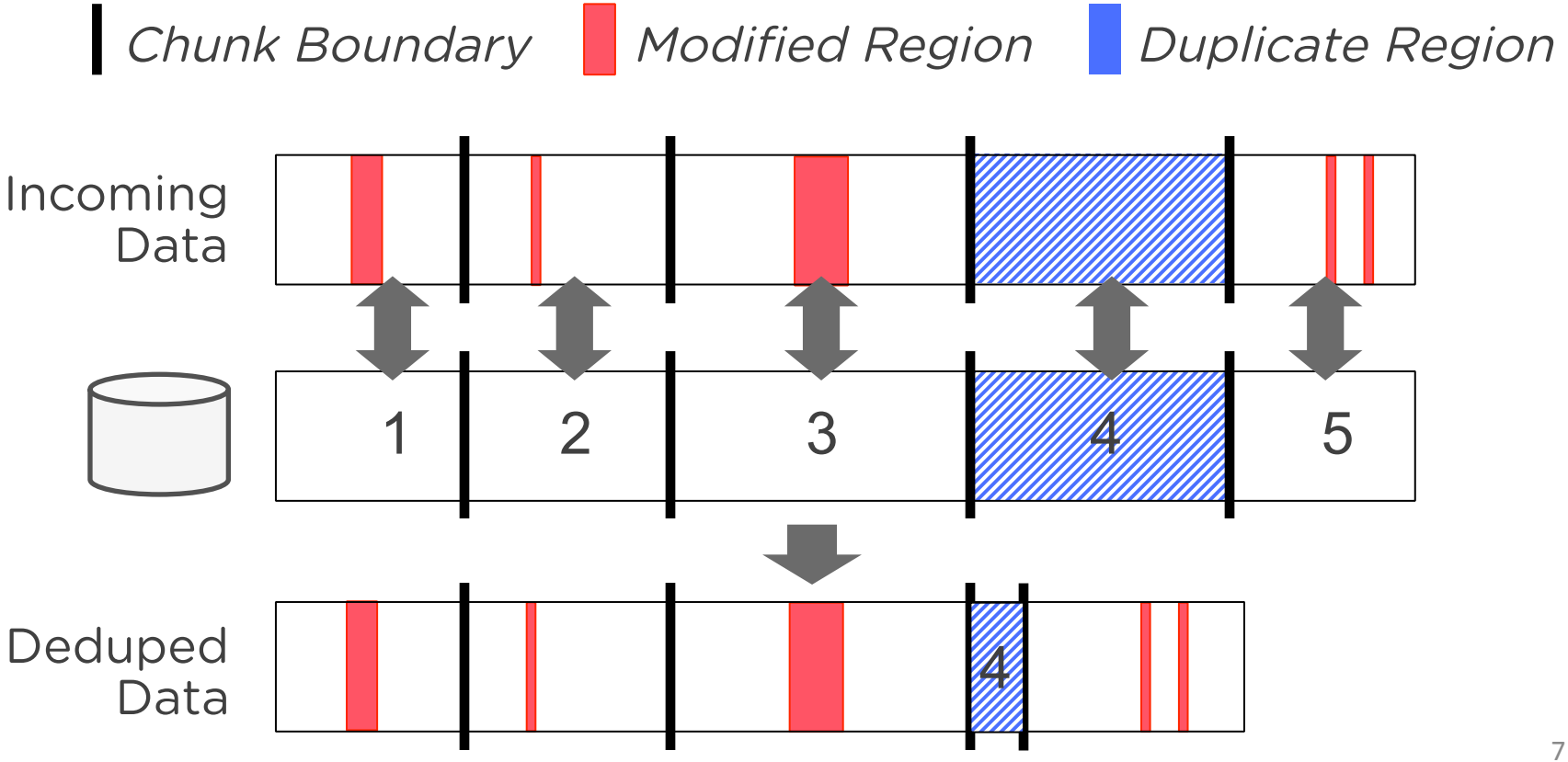
Why Deduplication?

- 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*

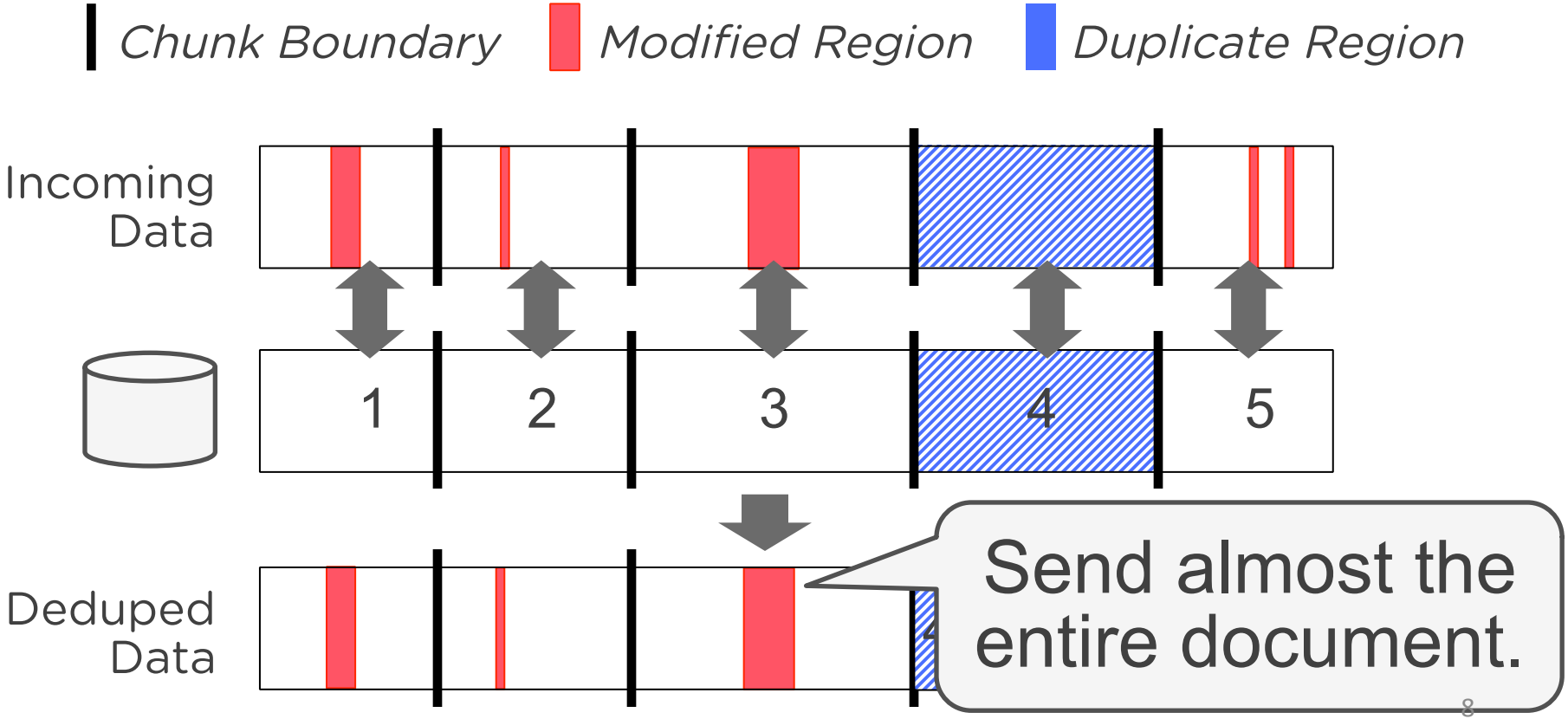
Traditional Dedup: Ideal



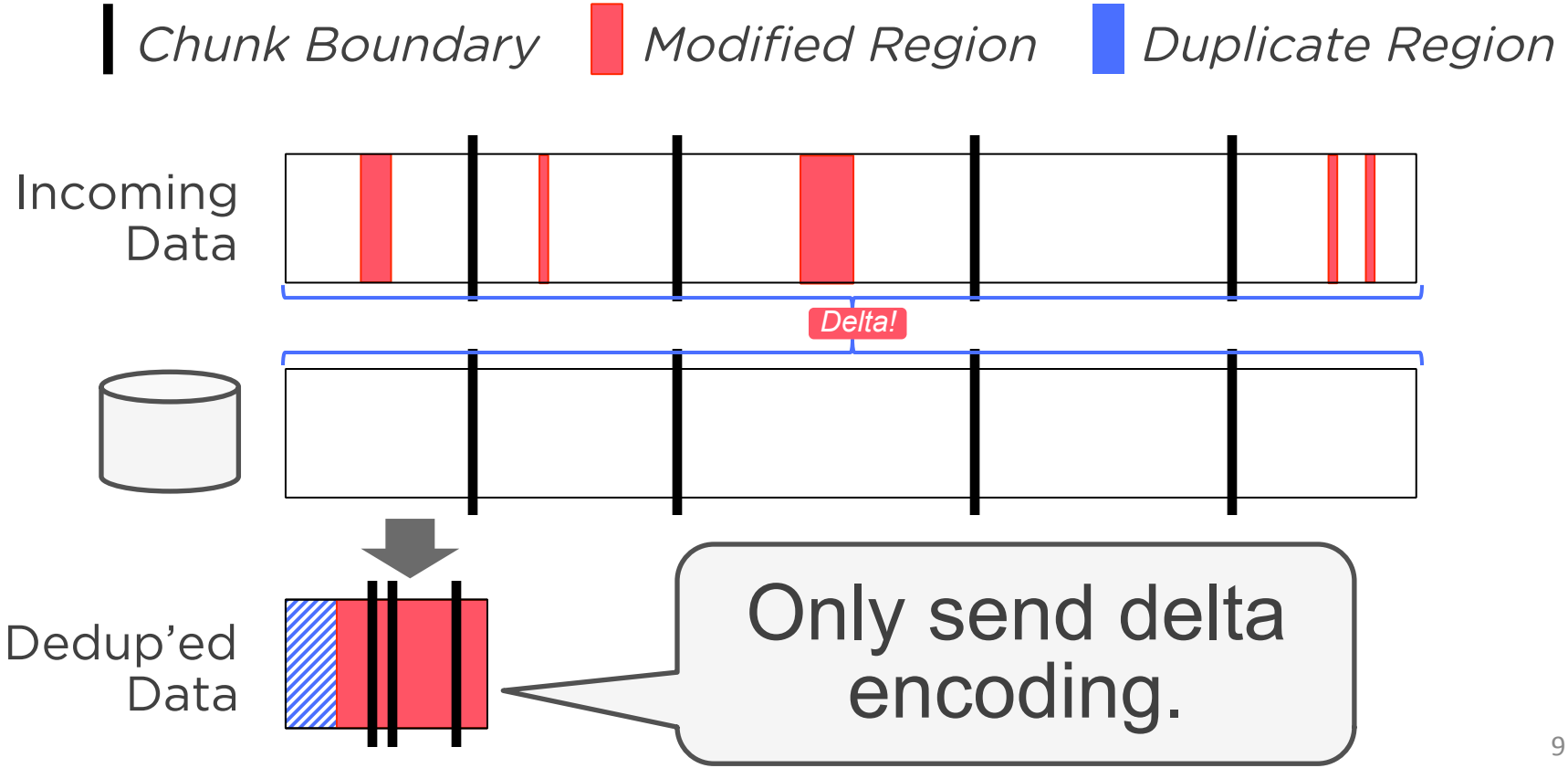
Traditional Dedup: Reality



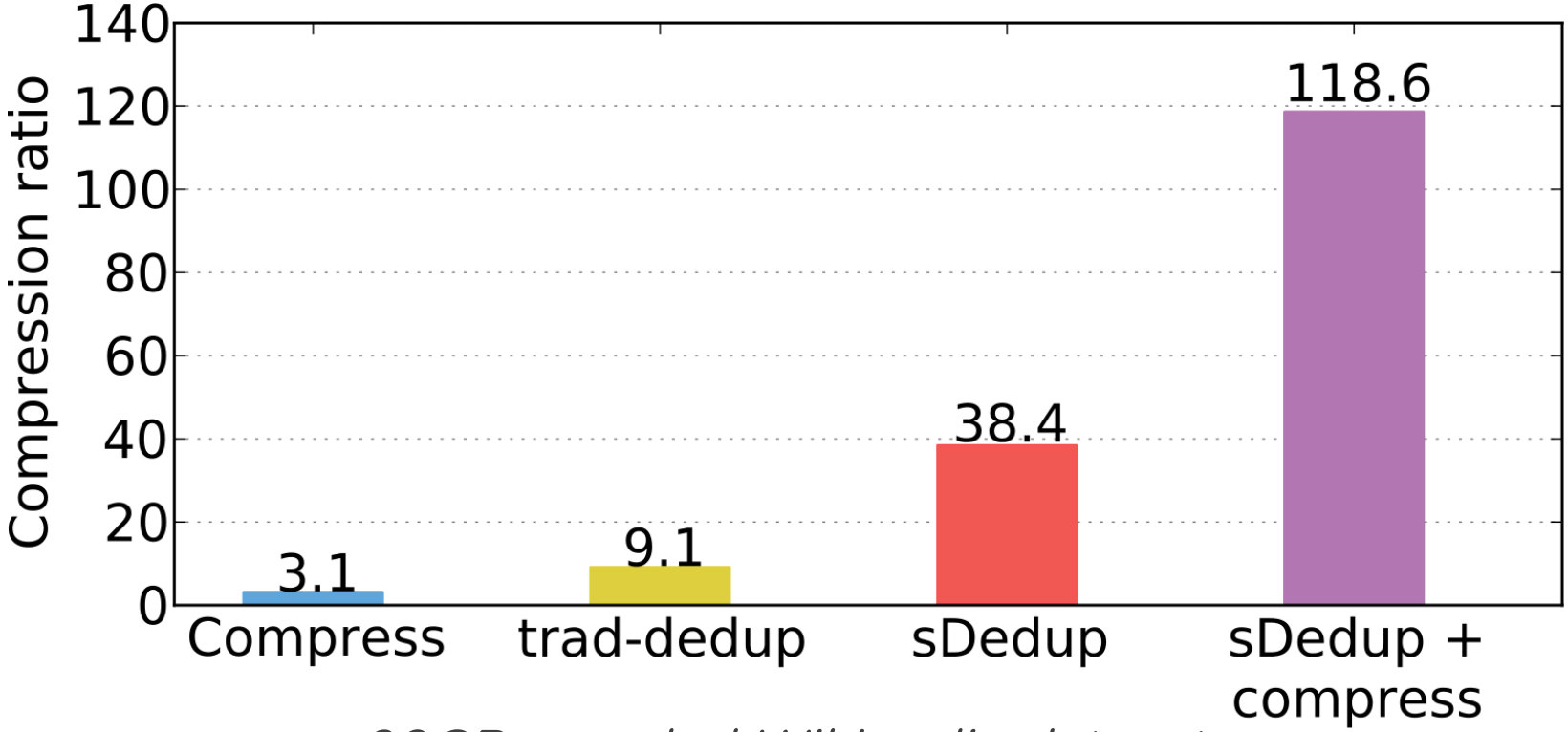
Traditional Dedup: Reality



Similarity Dedup (sDedup)

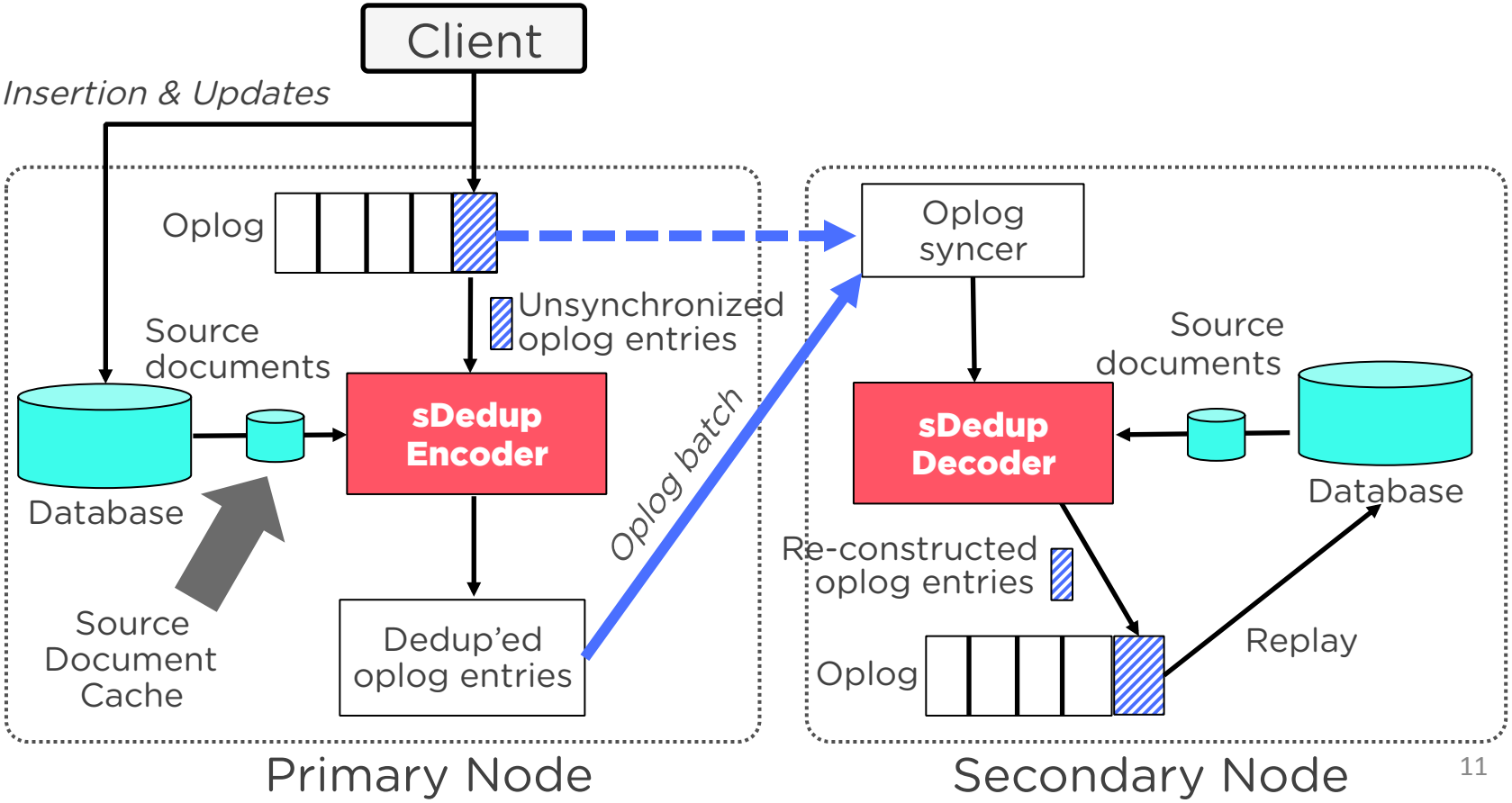


Compress vs. Dedup



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

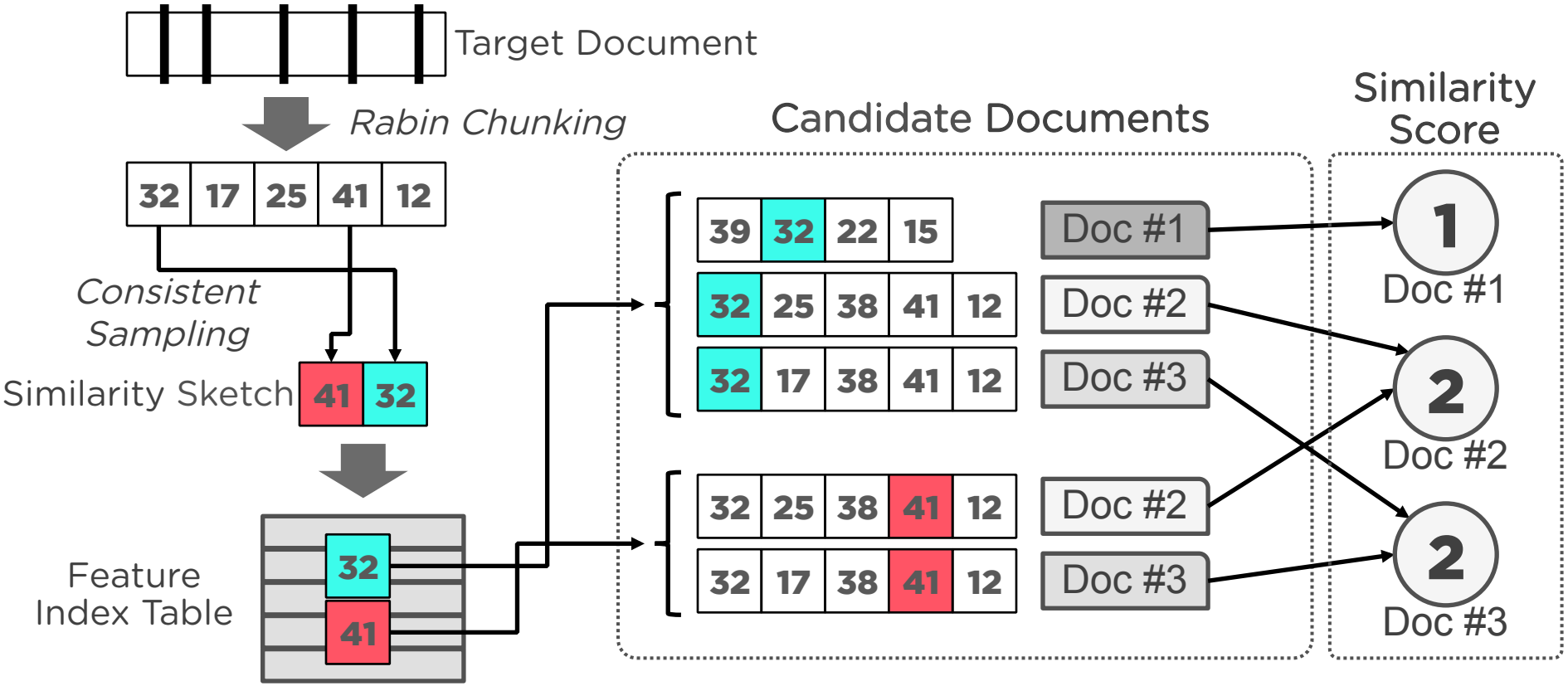
sDedup Integration



sDedup Encoding Steps

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

Identify Similar Documents



Select the Best Match

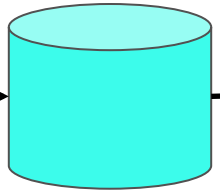
Initial Ranking

Rank	Candidates	Score
1	Doc #2	2
1	Doc #3	2
2	Doc #1	1

Final Ranking

Rank	Candidates	Cached?	Score
1	Doc #3	Yes	4
1	Doc #1	Yes	3
2	Doc #2	No	2

Is doc cached?



Source Document Cache

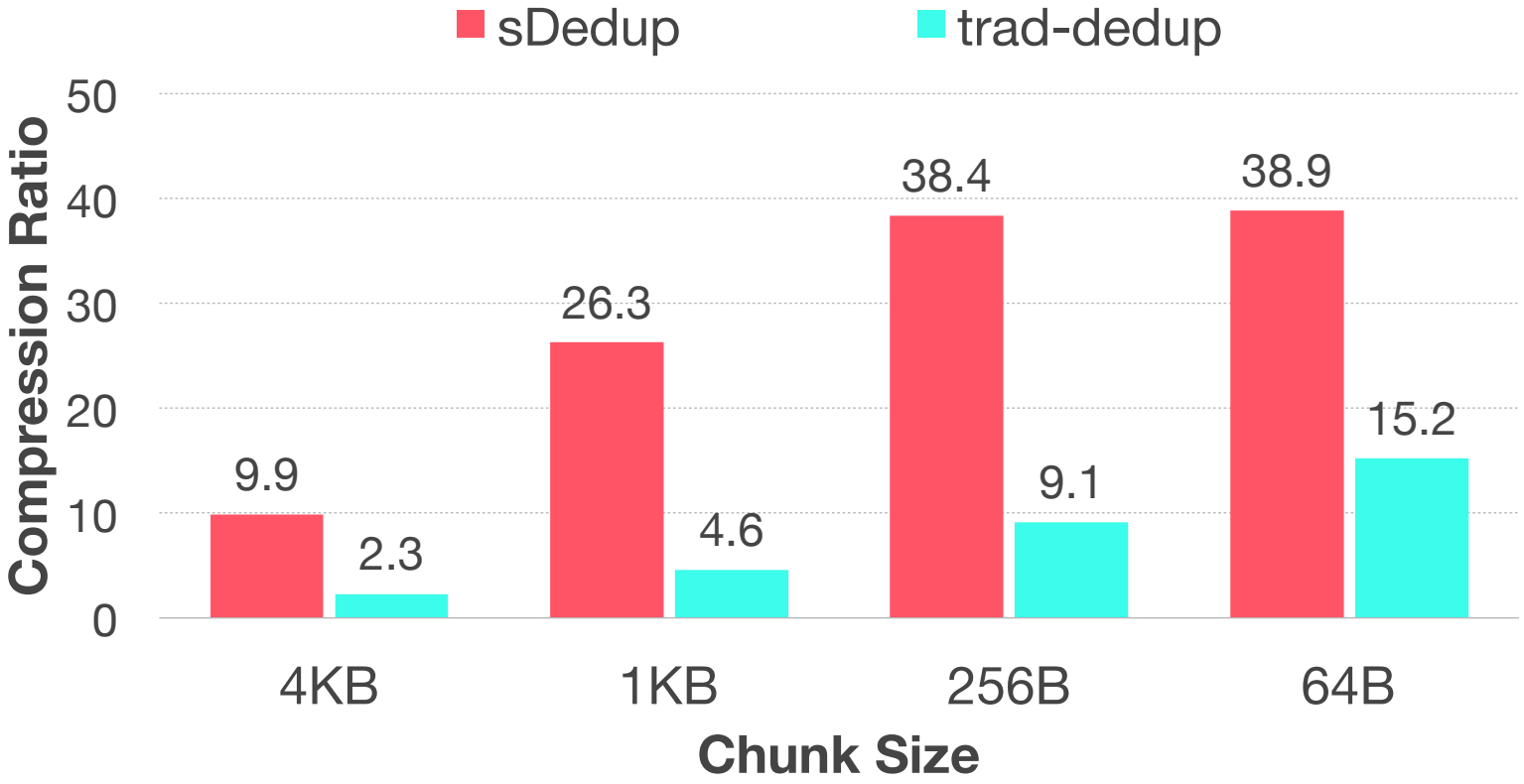
If yes, reward +2

Evaluation

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

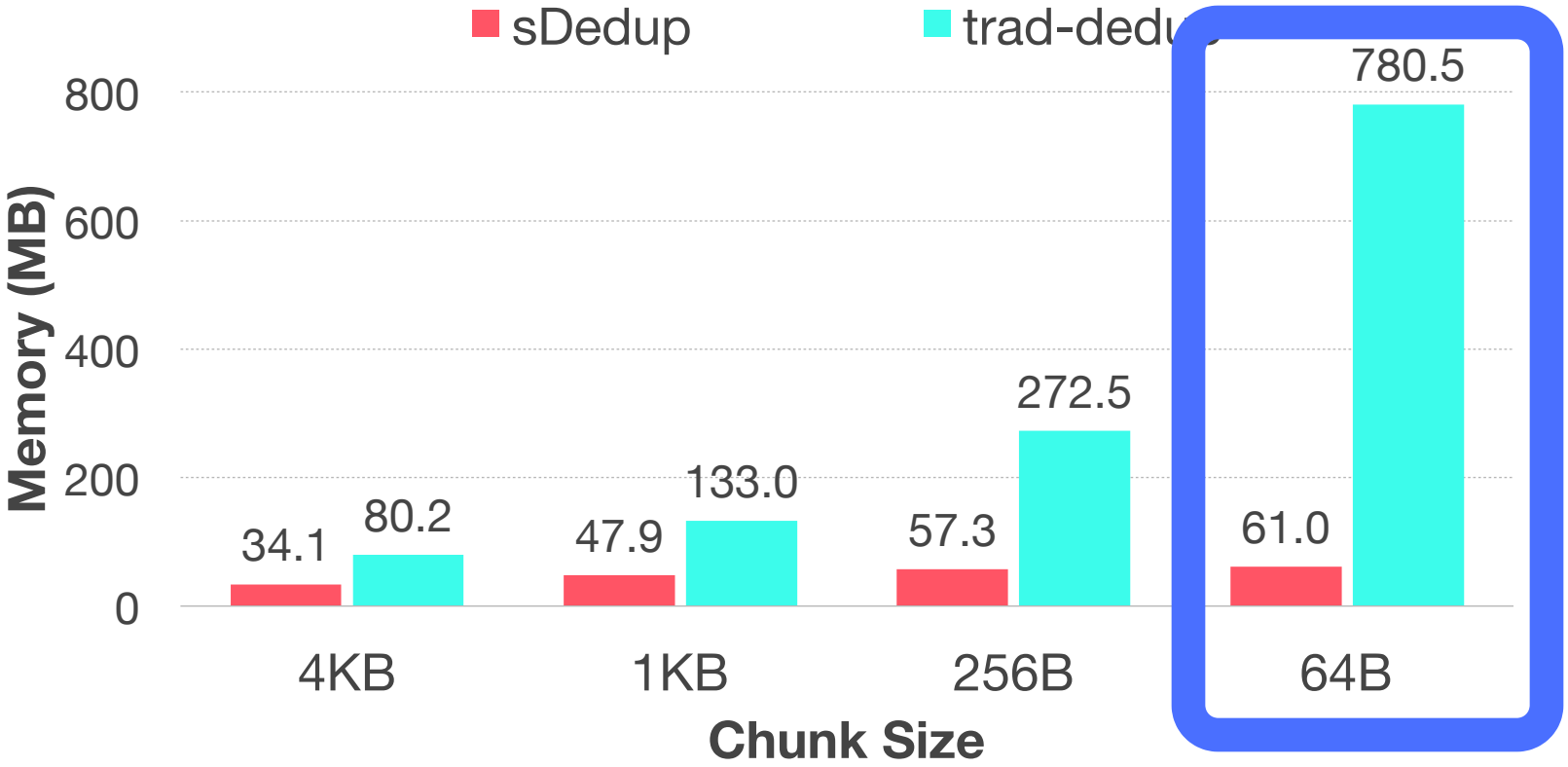
- Datasets:
  - *Wikipedia dump (20GB out of ~12TB)*
  - *Additional datasets evaluated in the paper*

Compression



20GB sampled Wikipedia dataset

Memory



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)*