iDedup: Latency-aware, inline data deduplication for primary storage

Kiran Srinivasan, Tim Bisson, Garth Goodson, Kaladhar Voruganti.

NetApp, Inc

FAST 12

Deduplication Techniques

- Primary Storage
  - Focus on performance and latency
  - Network file systems (RPC-based protocols) is latency sensitive
  - Only developed offline deduplication techniques
- Secondary Storage
  - Focus on data reliability and storage efficiency
  - No motivation to build inline deduplication techniques

Deduplication Techniques

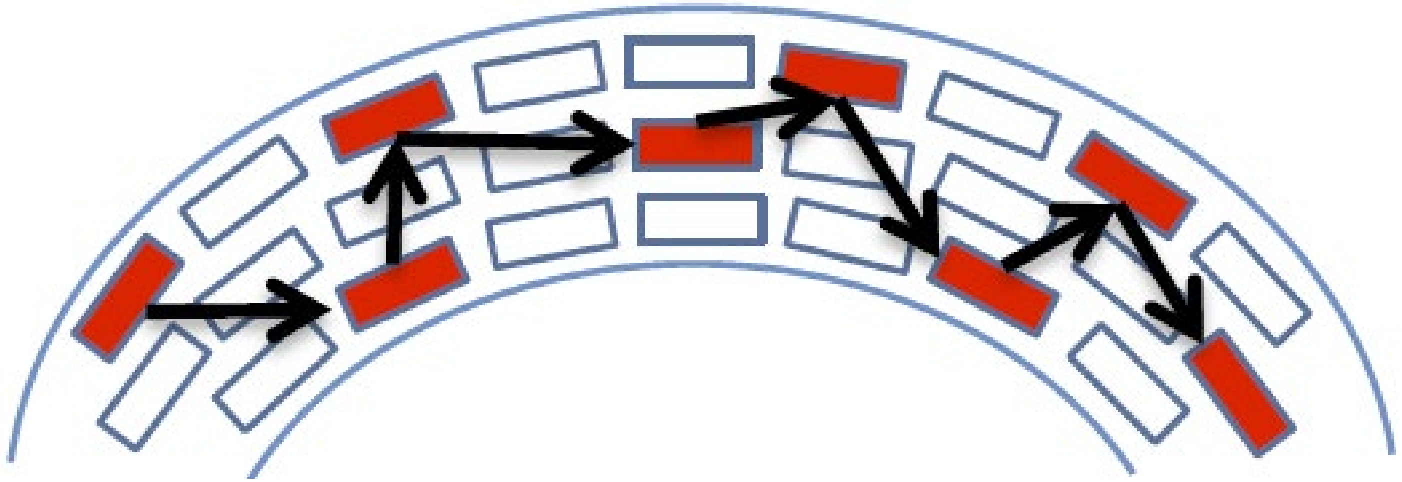
- Inline Deduplication
  - Dedup before storing first copy
  - Primary: affect write latency (no previous work)
  - Secondary: affect throughput (dedupe at idle time)
- Offline Deduplication
  - Dedup after storing first copy
  - Deduplication is a background activity

Necessity of Inline Deduplication for Primary

- No post-processing activities
  - No background processes
  - Not affect front-end workloads
  - Not require limited maintenance windows
- Efficient use of resources
  - No offline I/O usage
- Performance challenges is the key obstacle
  - Overheads (CPU & I/Os) for reads/writes hurt latency

Problems in Inline Deduplication – Read Path

- Deduplication causes disk-level fragmentation
  - Sequential reads turn random, leads to more seeks (more latency)
  - Workload/Dataset property
- Primary workloads are read-intensive
  - The read/write ratio is ~ 7:3
  - Inline deduplication must not affect read performance



Problems in Inline Deduplication – Write Path

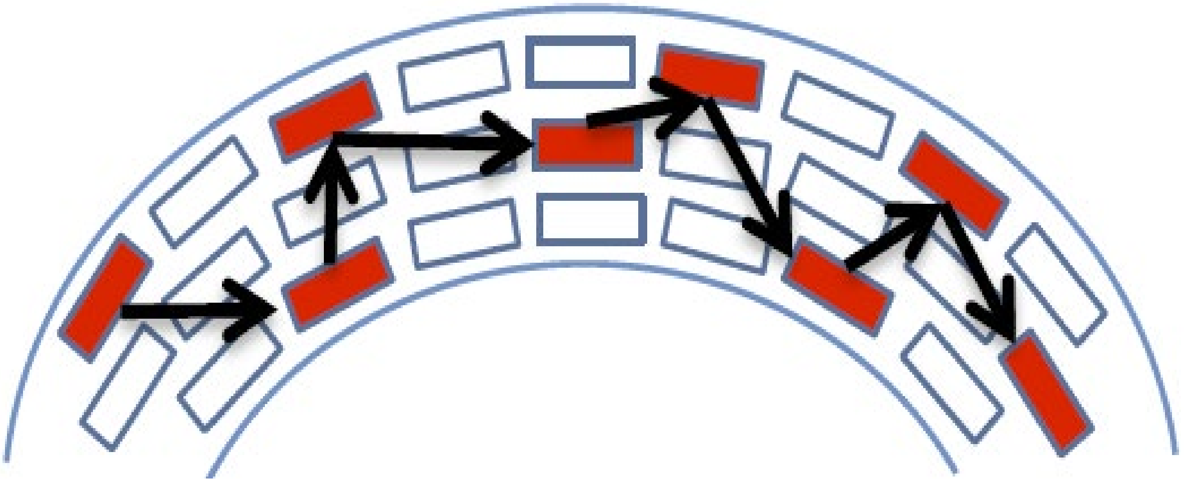
- CPU overheads in write path
  - Computing fingerprint for each block
  - Deduplication algorithm requires extra cycles
- Extra random I/Os in write path due to deduplication algorithm
  - Fingerprint queries and updates
  - Update block reference counts (for delete operation)
- Target: Find tradeoff between capacity saving and latency performance

Key Findings from Workload

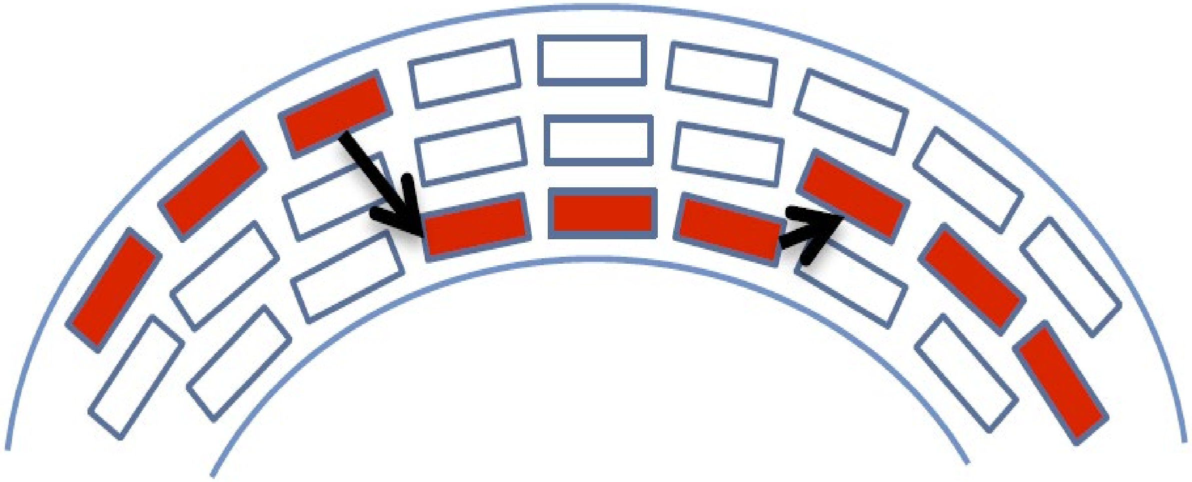
- Spatial locality
  - Dropping in deduplication ratio is less than linear with increasing block size.
  - Duplicated data is clustered
- Temporal locality
  - Dropping in deduplication ratio is less than linear with decreasing fingerprint table size (deduplication index)
  - Duplicate data is written repeatedly close in time

iDedup – Solve Read Path Issues

- Only dedup sequences of disk blocks
  - Solves fragmentation (amortized seeks during reads)
  - Configurable minimum sequence length
  - Perform selective dedup to leverage spatial locality



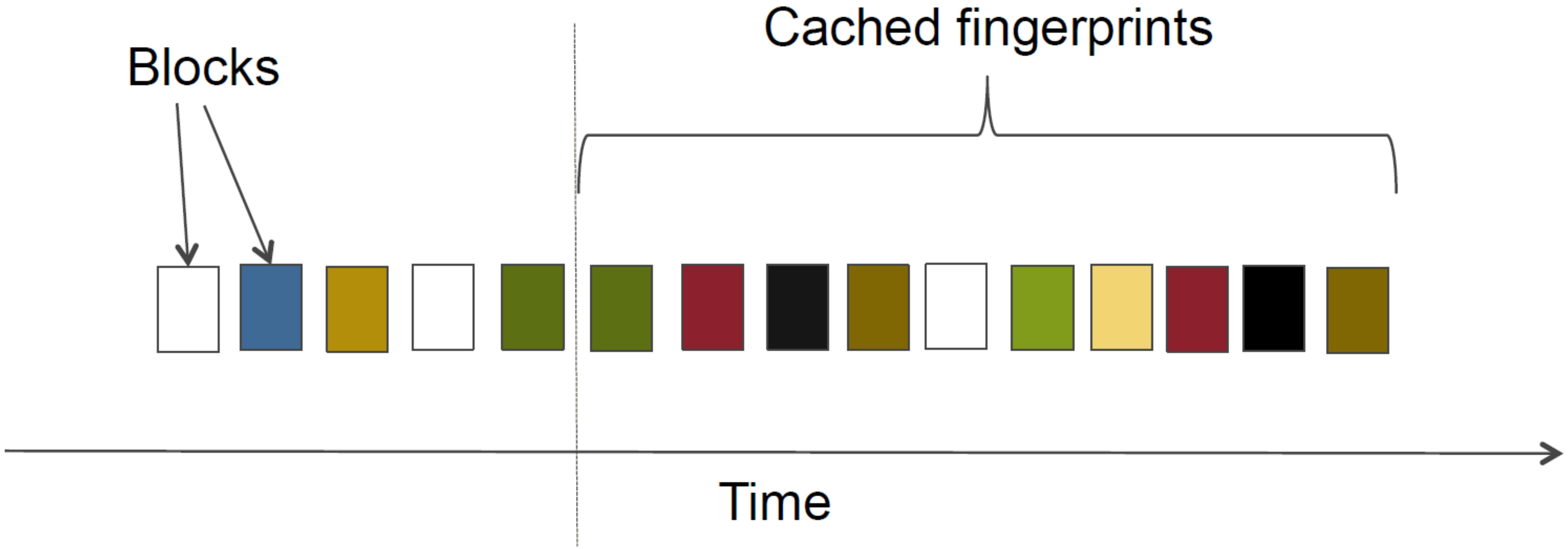
Fragmentation with random seeks



Sequences with amortized seeks

iDedup – Solve Write Path Issues

- Keep smaller dedup metadata as in-memory cache
  - No extra I/Os for fingerprint query and update
  - Leverages temporal locality in primary deduplication
  - Near-exact dedup (only subset of blocks are used for deduplication query)

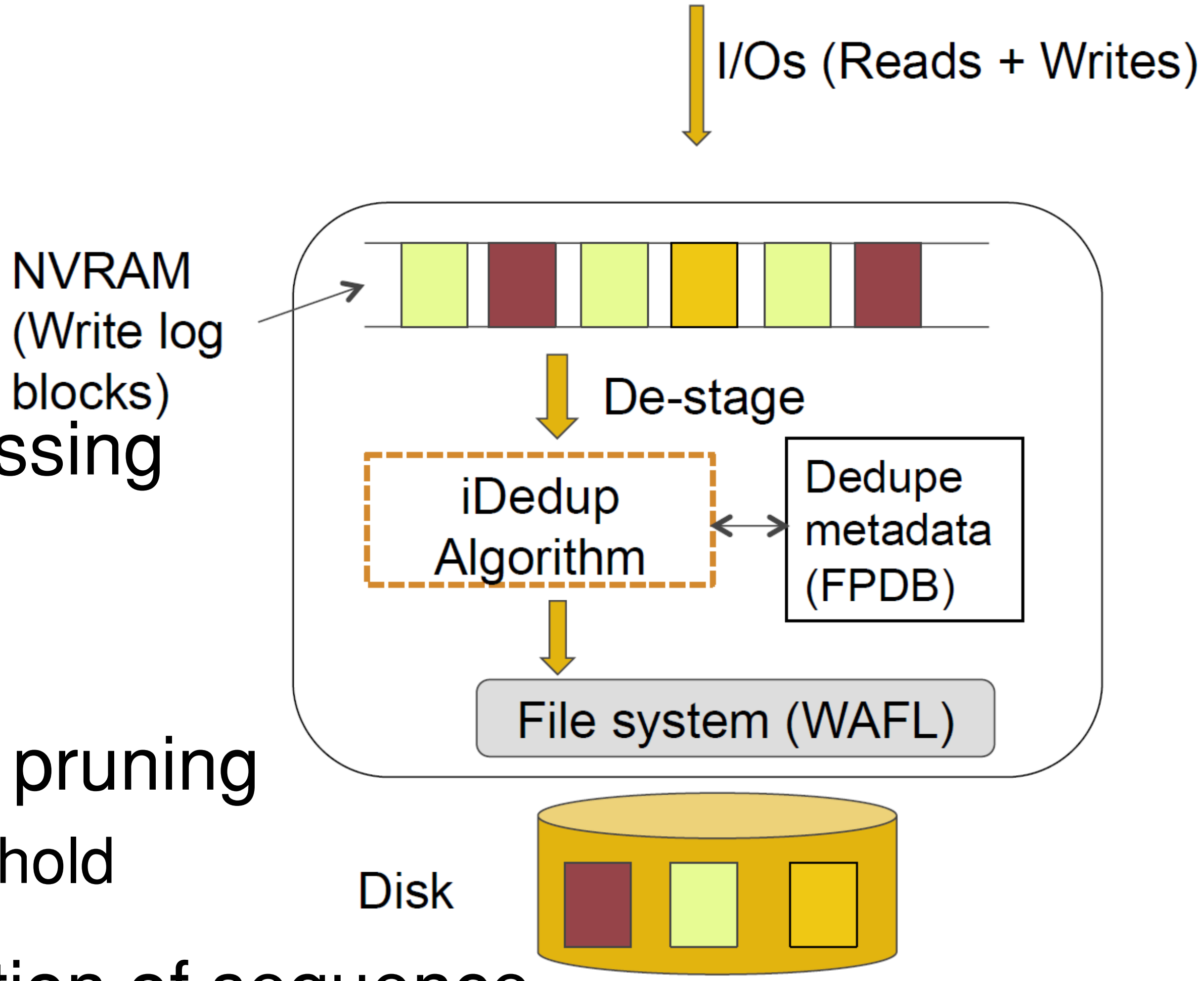


iDedup – Two key Parameters

- Minimum sequence length (threshold)
  - Minimum number of sequential duplicate blocks on disk
  - Dataset property => ideally set to expected fragmentation
  - Knob between performance (fragmentation) and dedupe
- Dedupe metadata (Fingerprint DB) cache size
  - Workloads working set property
  - Increase in cache size => decrease in buffer cache
  - Knob between performance (cache hit ratio) and dedupe

iDedup – Architecture

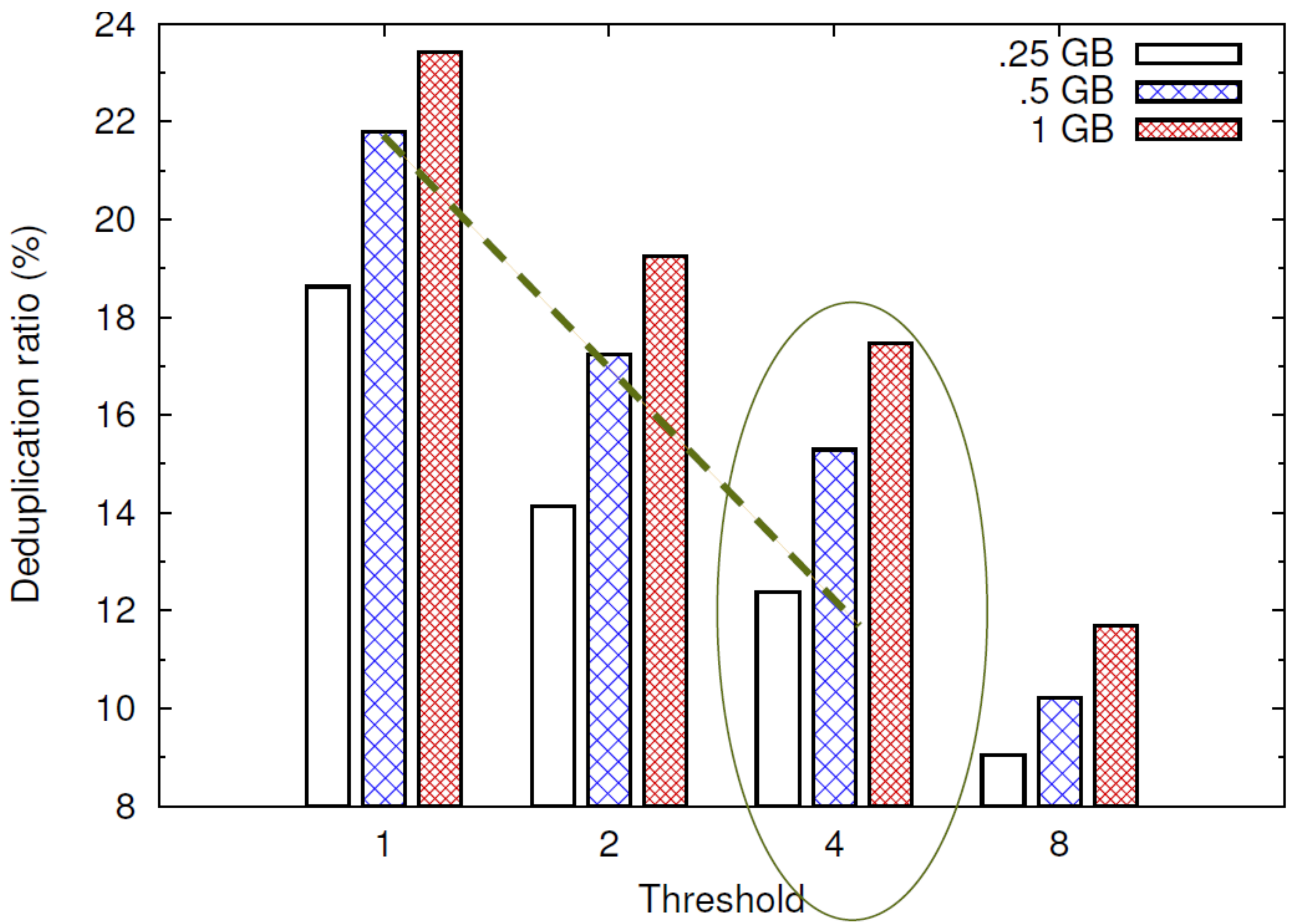
- Phase 1 (per file):Identify blocks
  - Process only data blocks
  - Ignore metadata blocks, special files
- Phase 2 (per file) : Sequence processing
  - Uses the dedupe metadata cache
  - Keeps track of multiple sequences
- Phase 3 (per sequence): Sequence pruning
  - Eliminate short sequences below threshold
- Phase 4 (per sequence): Deduplication of sequence



Evaluation

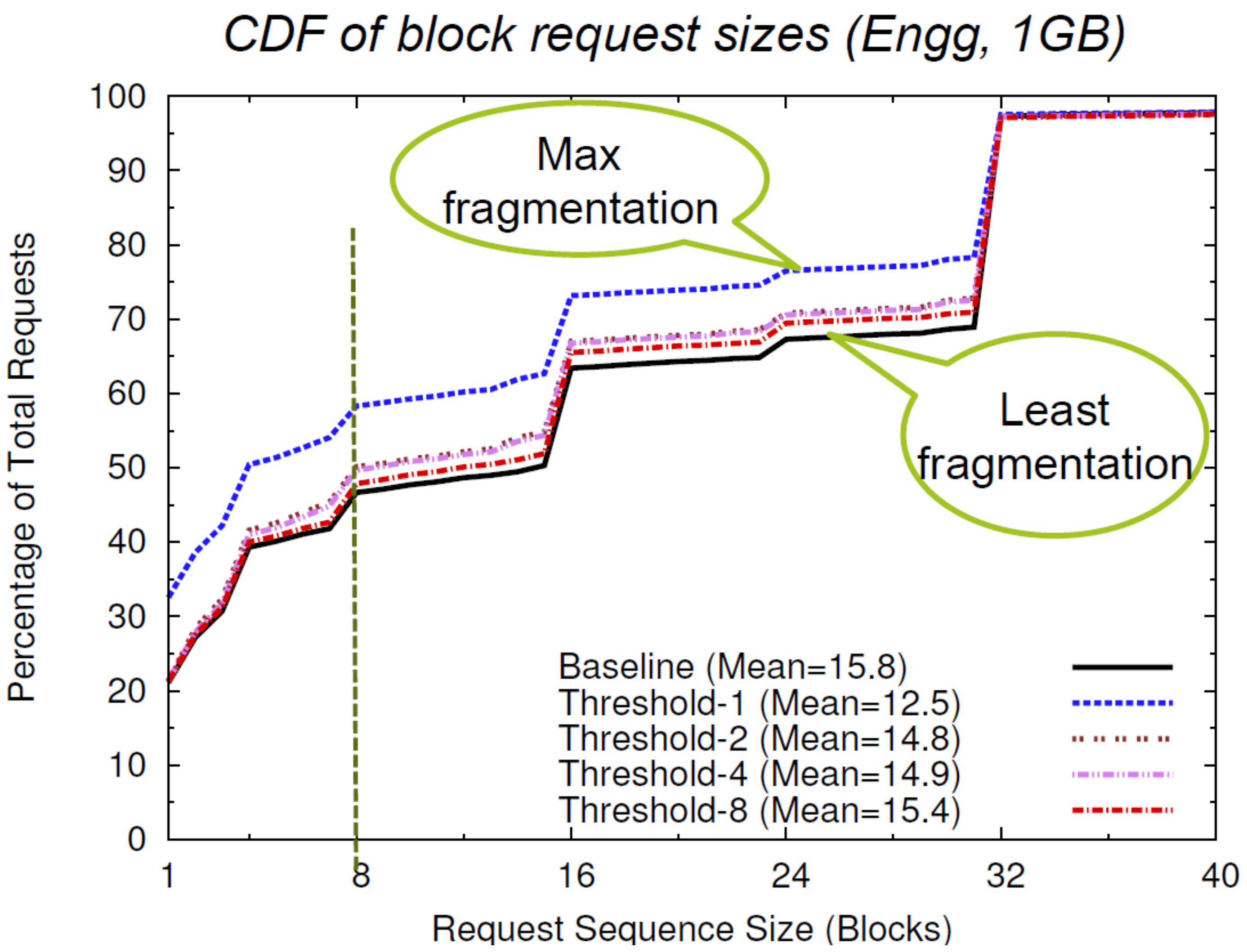
- Dataset: CIFS network trace (Claimed to be public, but not found)
- Comparison
  - Baseline: system with no iDedup
  - Threshold-1: system with full deduplication (1 block)
- Deduplication metadata cache size: 0.25, 0.5, 1 GB

Deduplication Ratio vs. Threshold



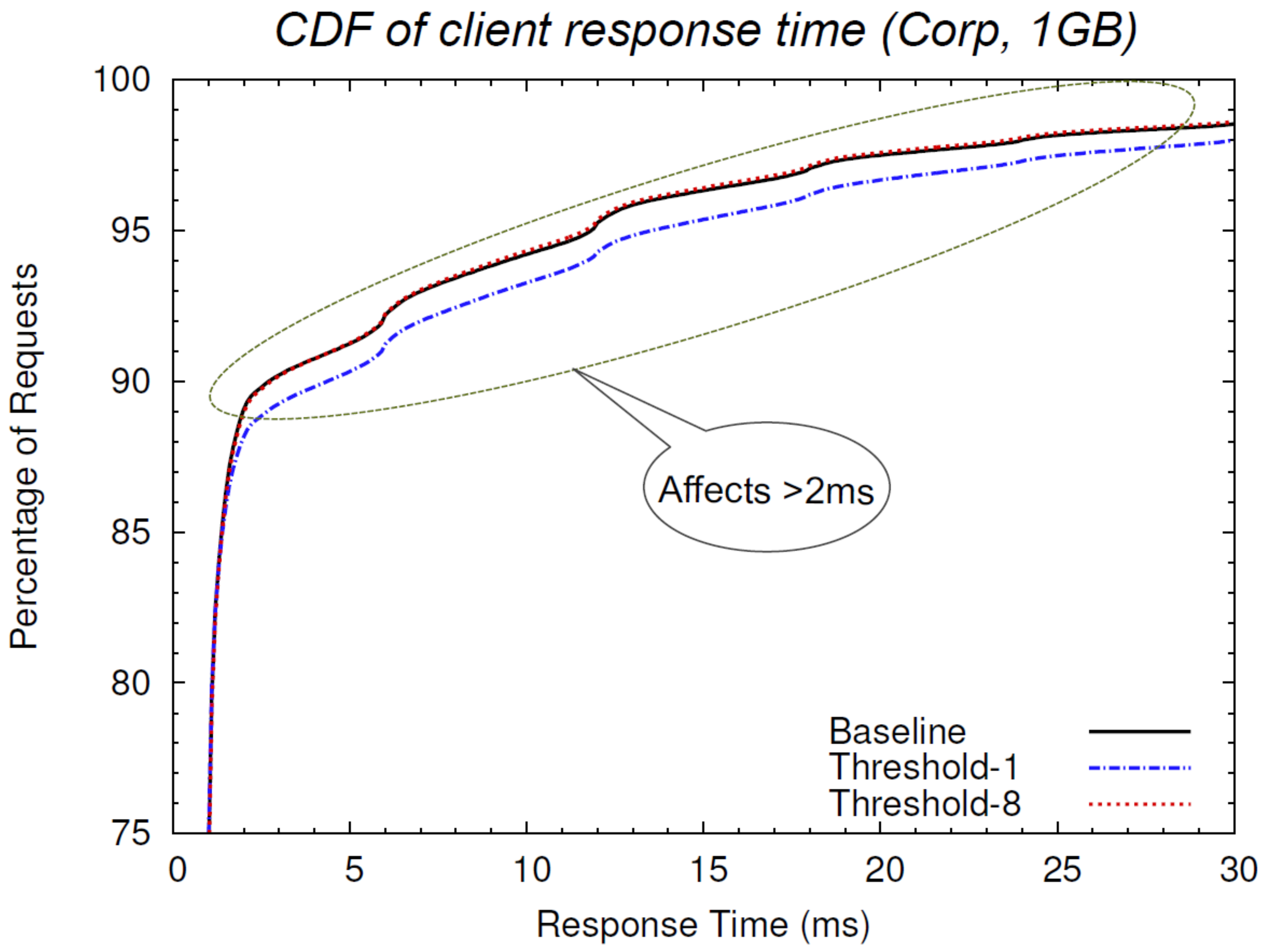
- Ideal Threshold = biggest threshold with least decrease in dedup savings.
- Threshold-4 achieve ~60% of max deduplication ratio

Disk Fragmentation vs. Threshold



- Fragmentation for other thresholds are between Baseline and Threshold-1
- The fragmentation is tunable with the threshold.

Latency Impact



- Threshold-1 mean latency affected ~13% vs. Baseline
- Different between Threshold-8 and Baseline < 4%

Conclusions

- Inline dedupe has significant performance challenges
  - Reads : Fragmentation, Writes: CPU + Extra I/Os
- iDedup creates tradeoffs between storage savings and performance
  - Leverage dedupe locality properties
  - Avoid fragmentation - dedupe only for sequences
  - Avoid extra I/Os - keep dedupe metadata in memory
- Experiments for latency-sensitive primary workloads
  - ~60% of max dedup, ~4% impact on latency