用於超大規模資料去重複化之資料指紋庫的截斷型日誌結構合併樹

Enabling Ultra-Scale Data Deduplication with Truncated Log-Structured Merge-Trees in Fingerprint Stores

指導教授：謝昀珊　組員姓名：黃紋綺、溫珮伶、李良駿

Introduction

Inline deduplication can immediately eliminate redundant writes, but it requires fingerprint lookup on the write critical path, making it highly sensitive to latency. As data volume grows, keeping the entire fingerprint store in memory becomes impractical. Moving fingerprint lookup to disk reduces memory pressure but introduces extra I/O, leading to the *chunk-lookup disk bottleneck*.

Our observation shows that *old fingerprints contribute little to deduplication effectiveness*. Therefore, this work focuses on retaining recent and valuable fingerprints in bounded memory to achieve low-latency lookup while preserving most deduplication benefits.

System Architecture

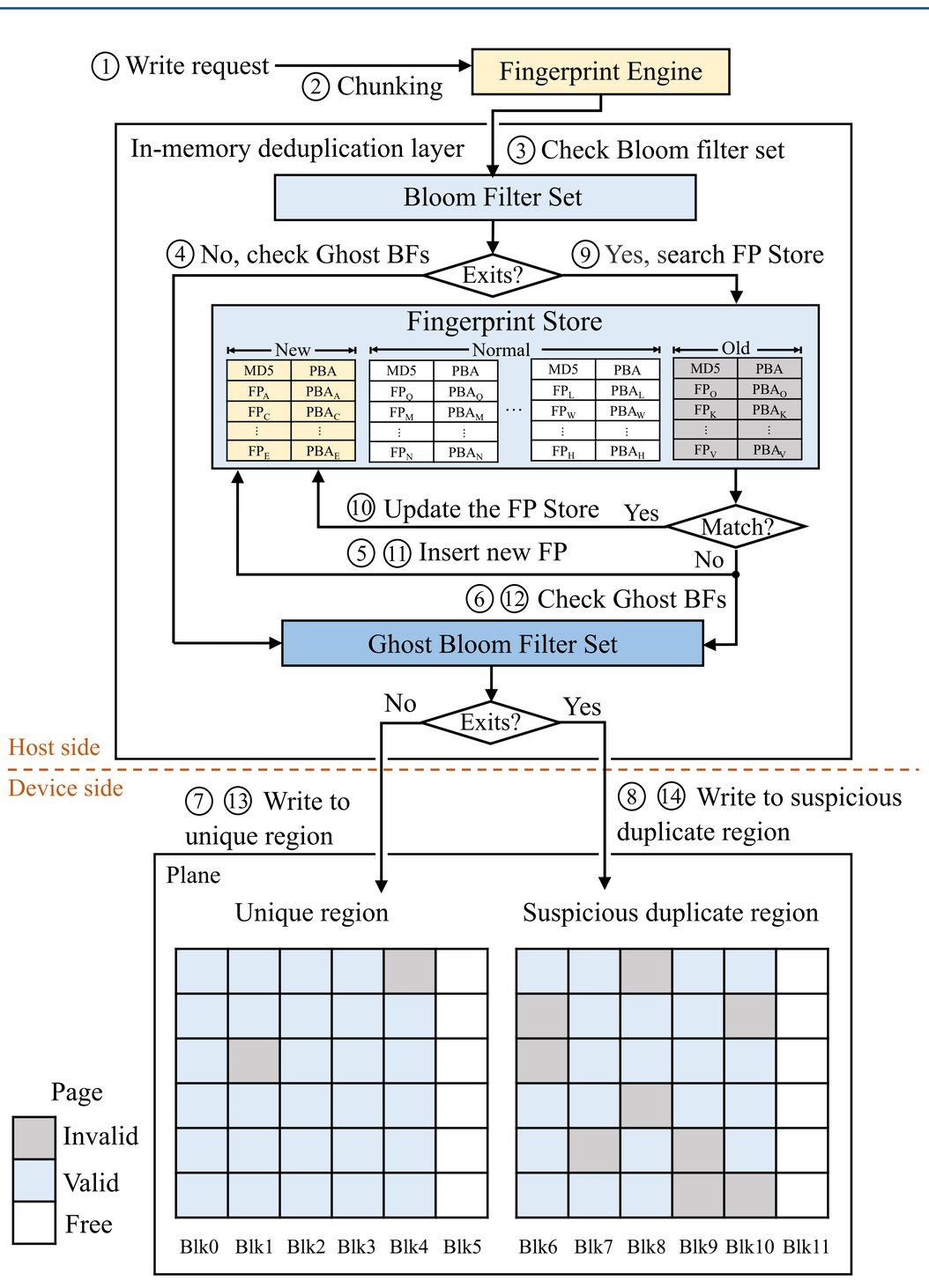

Figure 1: System architecture

1. Chunk and fingerprint each incoming write.
2. Detect recent duplicates using active Bloom filters and the fingerprint store.
3. Use Ghost Bloom Filters to identify possible aged duplicates.
4. Classify writes as normal or suspicious duplicates.
5. Place them into separate regions to improve data locality.

Results

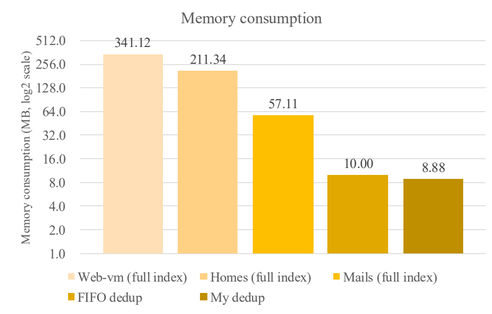

Figure 2: Memory consumption

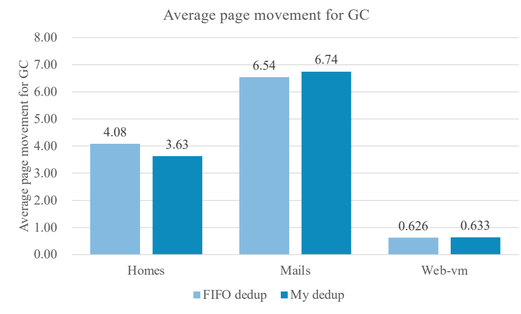

Figure 3: Avg page movement for GC

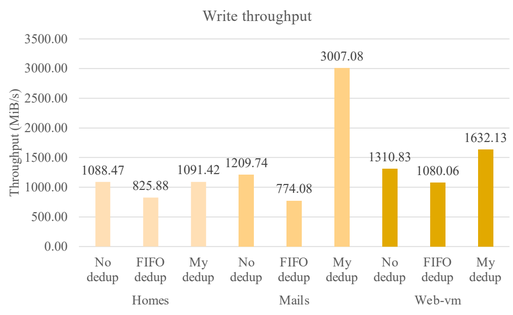

Figure 4: Write throughput

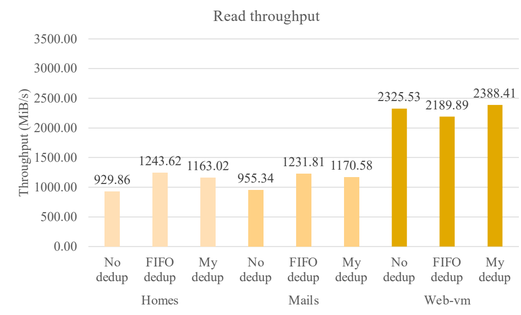

Figure 5: Read throughput

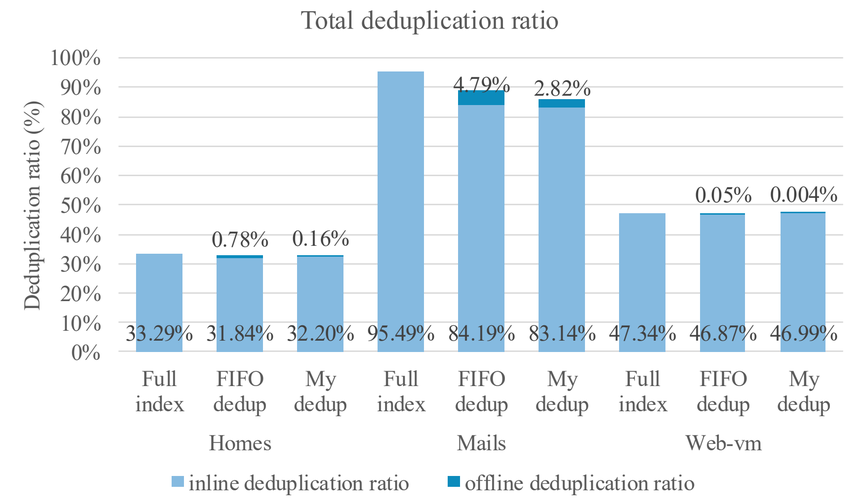

Figure 6: Total deduplication ratio

Conclusion

The proposed design reduces memory consumption compared with perfect inline deduplication while still maintaining a good deduplication ratio. By keeping recent and valuable fingerprints in bounded memory, the system lowers fingerprint lookup overhead and improves both write and read throughput. In addition, the separate-region write design improves invalid-page locality and reduces GC overhead, especially in the *Homes* dataset. These results show that the proposed method can improve memory efficiency, storage efficiency, and GC behavior with limited loss in duplicate elimination effectiveness.

Future Work

Future work will make the system more adaptive by adjusting the number of Ghost Bloom Filters according to system loading. The classification mechanism can also be extended from binary classification to multiple classes, such as *Unique*, *Normal*, and *Suspicious*. When system overhead is low, the system can further inspect suspicious and normal writes to recover more duplicate data, improve space efficiency, and reduce GC overhead.