FUJITSU

shaping tomorrow with you

DBLK: Deduplication for primary storage

Yoshi Tsuchiya and Takashi Watanabe
Fujitsu

5/26/2011 IEEE MSST Research Track

This work is supported by the Green IT project of New Energy and Industrial Technology Development Organization (NEDO)

Outline

FUJITSU

- Background
- DBLK outline
- DBLK Hash index
- Multilayer Bloom Filter (MBF)
- MBF Bitwise Transposition (MBF BT)
- Summary

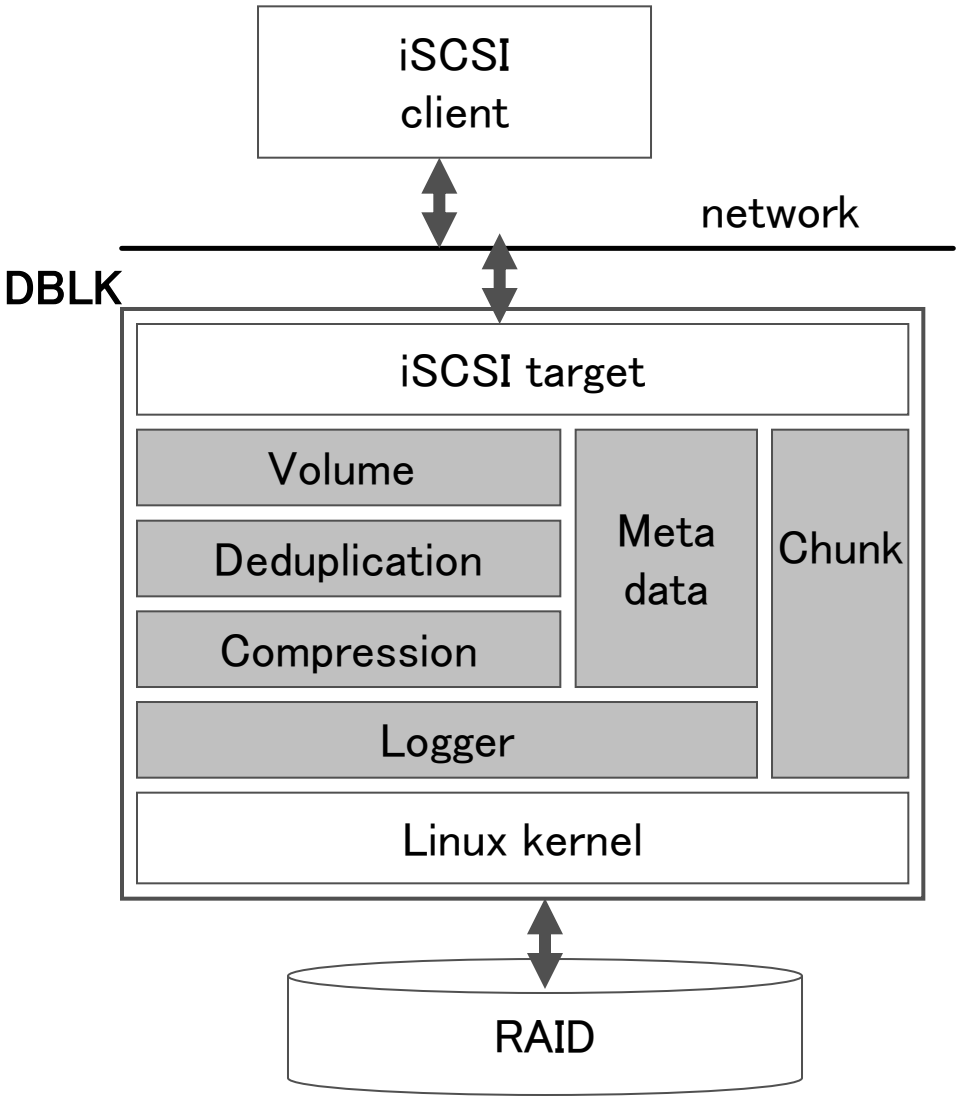
Deduplication Background

FUJITSU

- Deduplication: Reduce size of storage by removing redundancy in data
- Dedup for Primary storage/Backup storage
  - Primary storage: file servers, small latency is required
- Inline/Post-process dedup
  - Inline: immediately deduplicate and write data
  - Post process: write data first, deduplicate later, needs extra storage
- Deduplication unit
  - Fixed size block
  - File
  - Variable-sized segment

DBLK iSCSI implementation

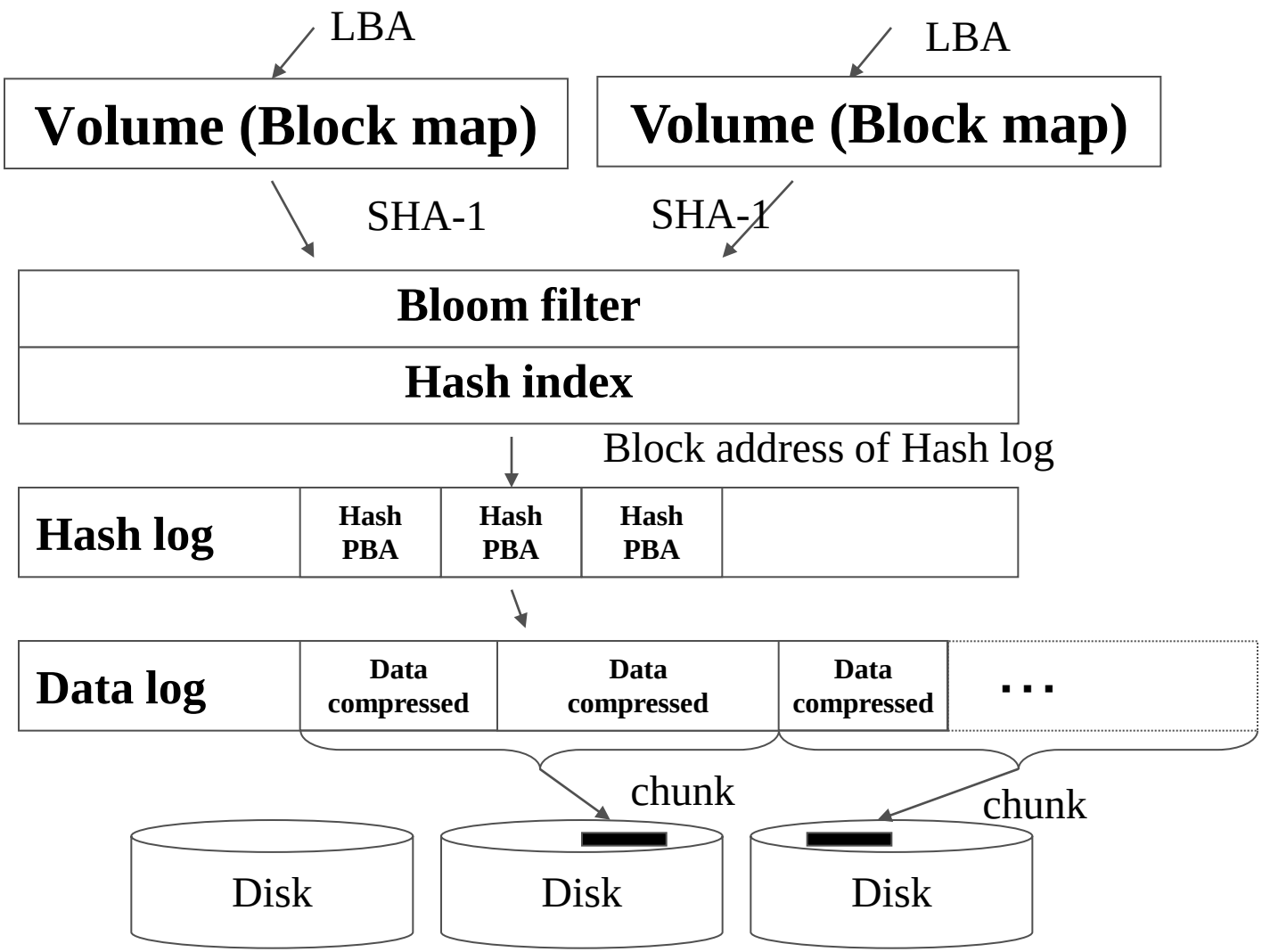
FUJITSU



Primay block storage
Inline, block-wise deduplication, thin provisioning and compression

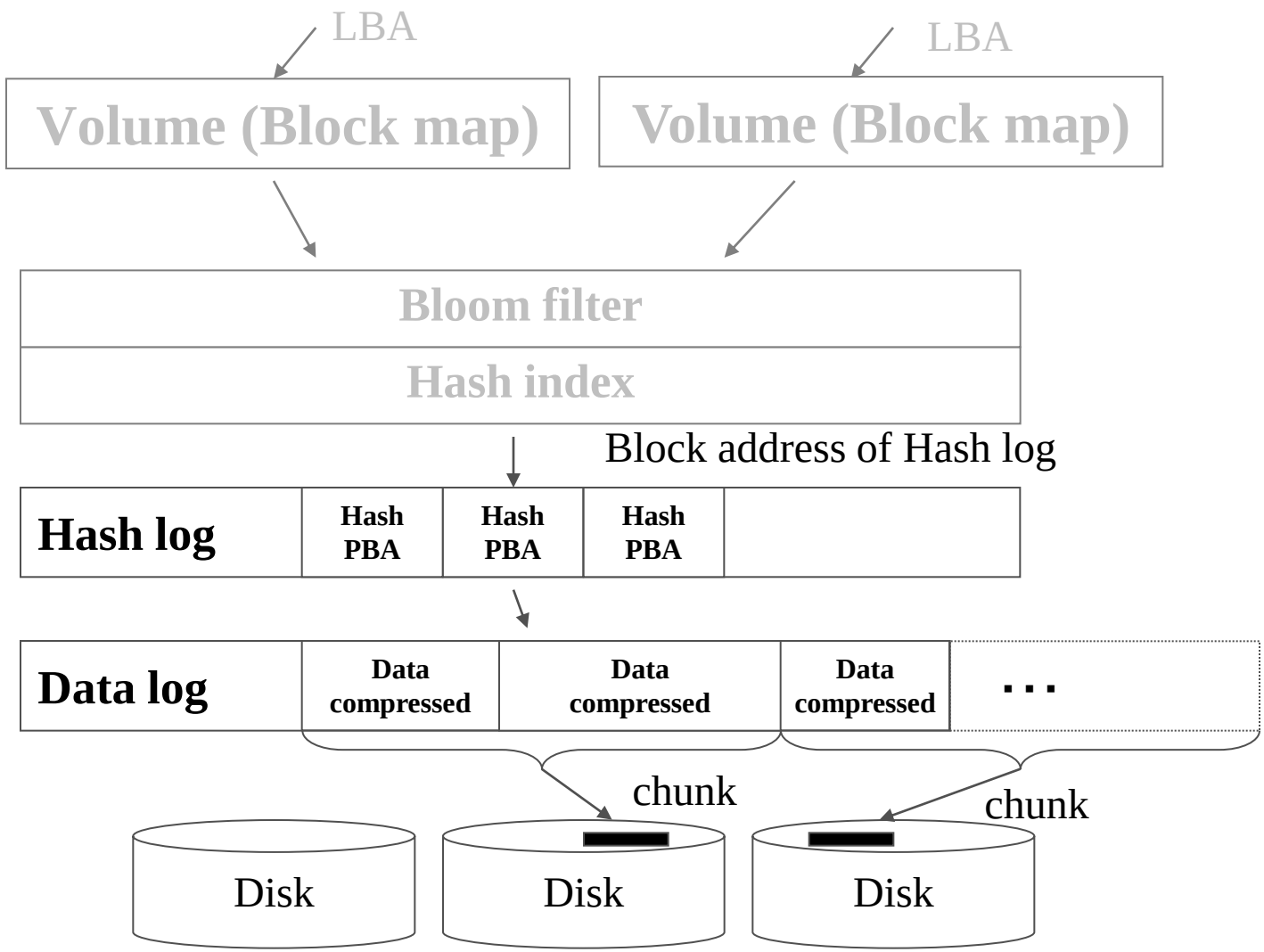
DBLK Data Structure

FUJITSU

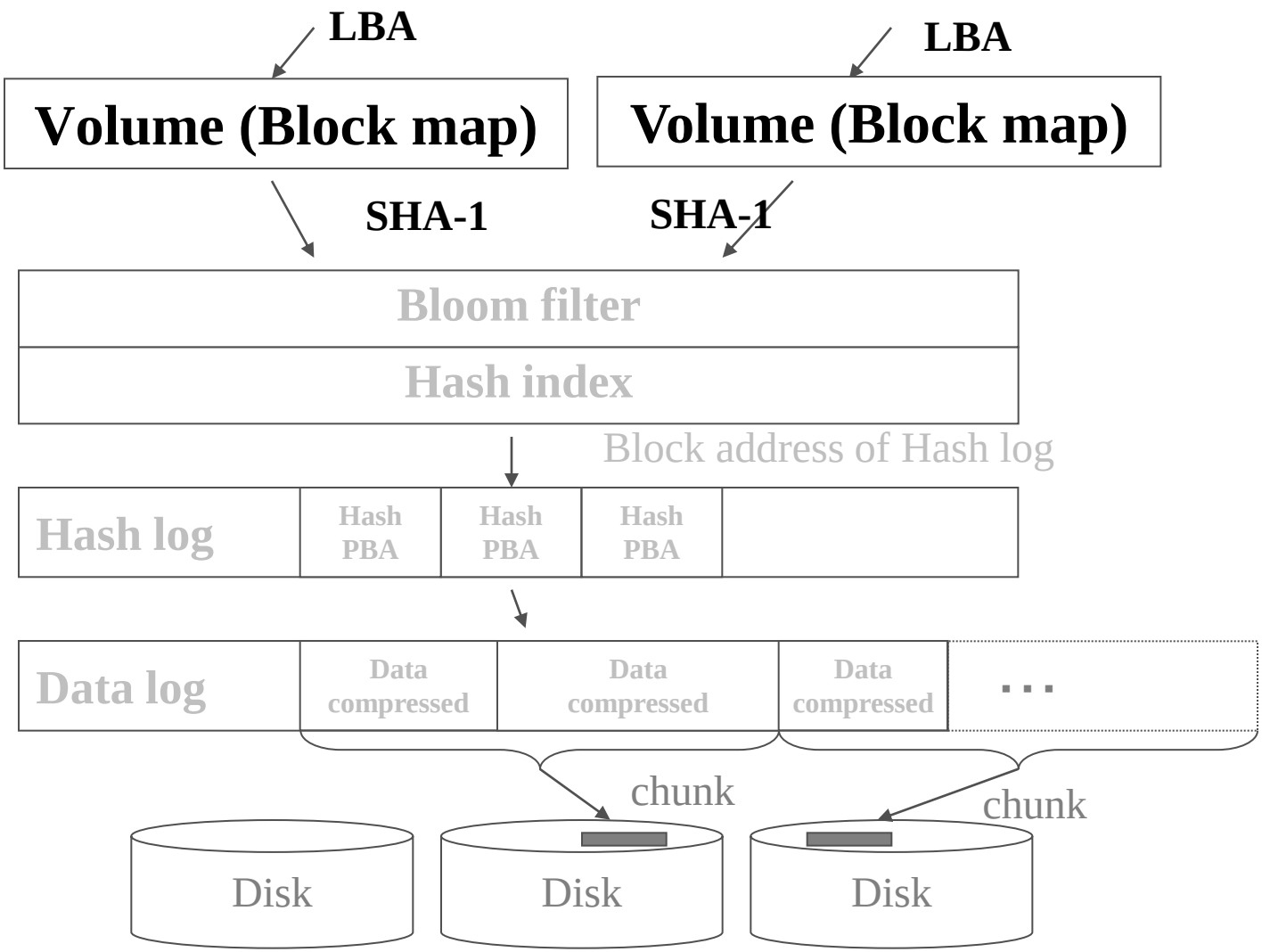


DBLK Data Structure: Logs

FUJITSU

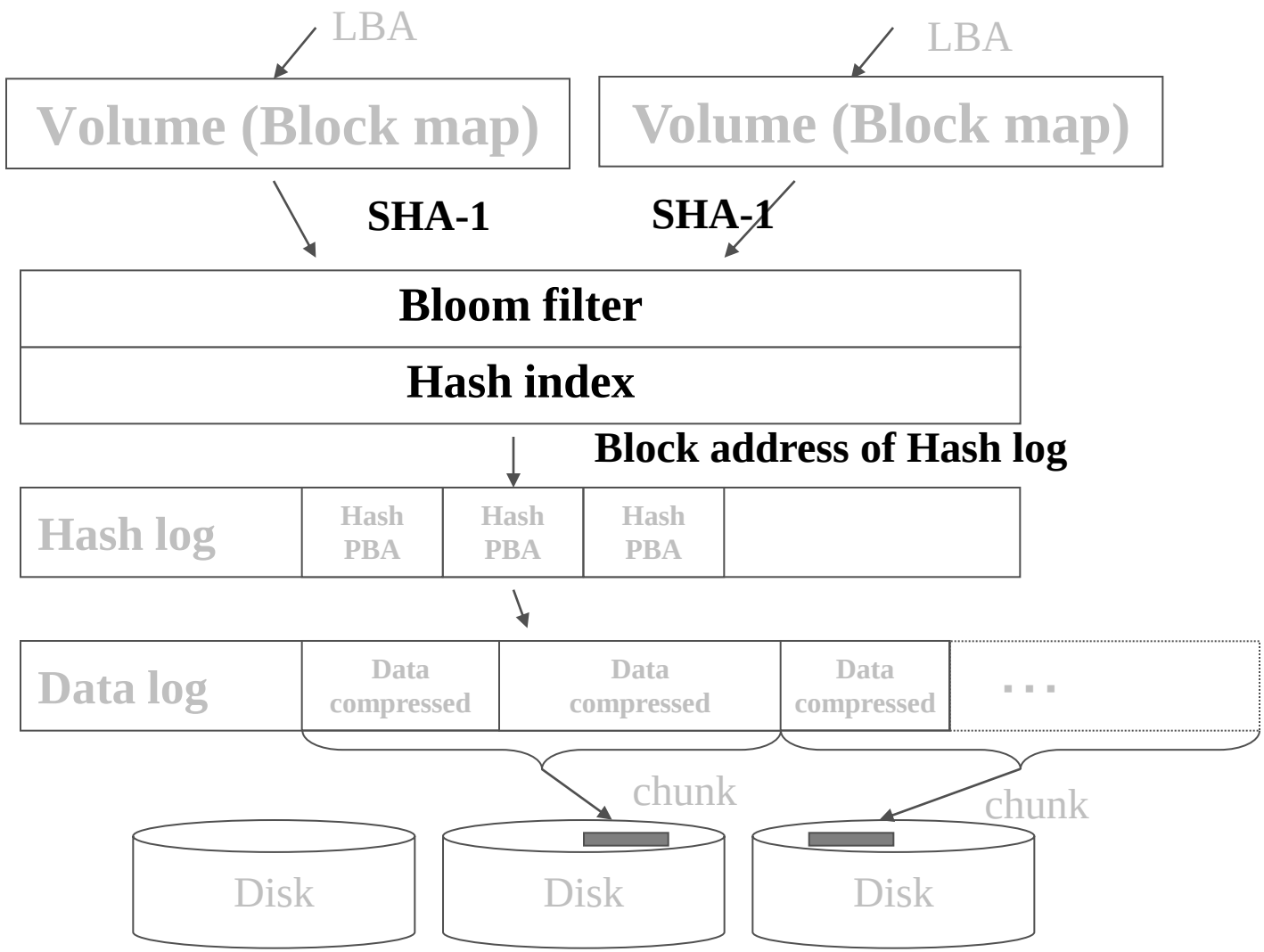


DBLK Data Structure: Block maps

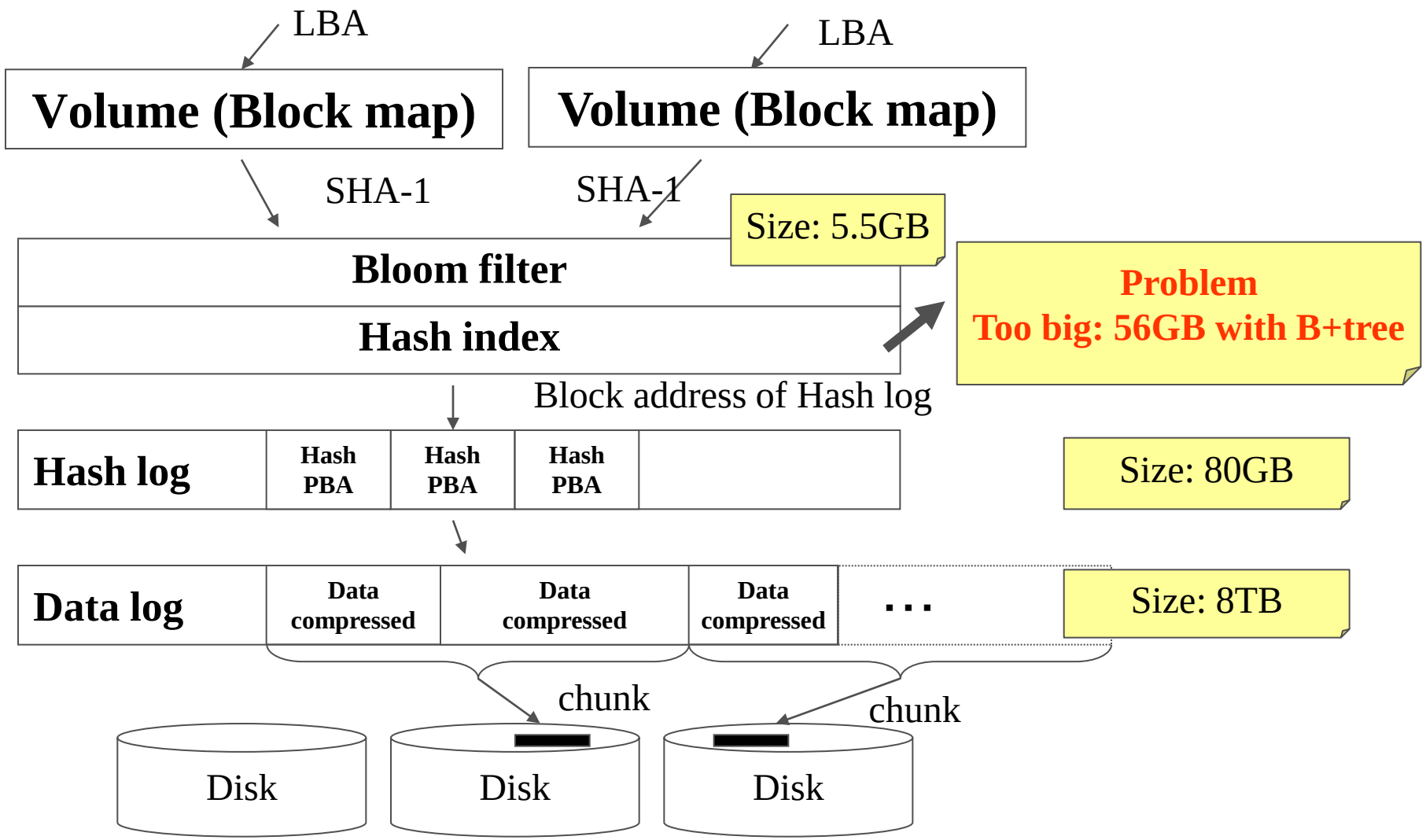


DBLK Data Structure: Hash index

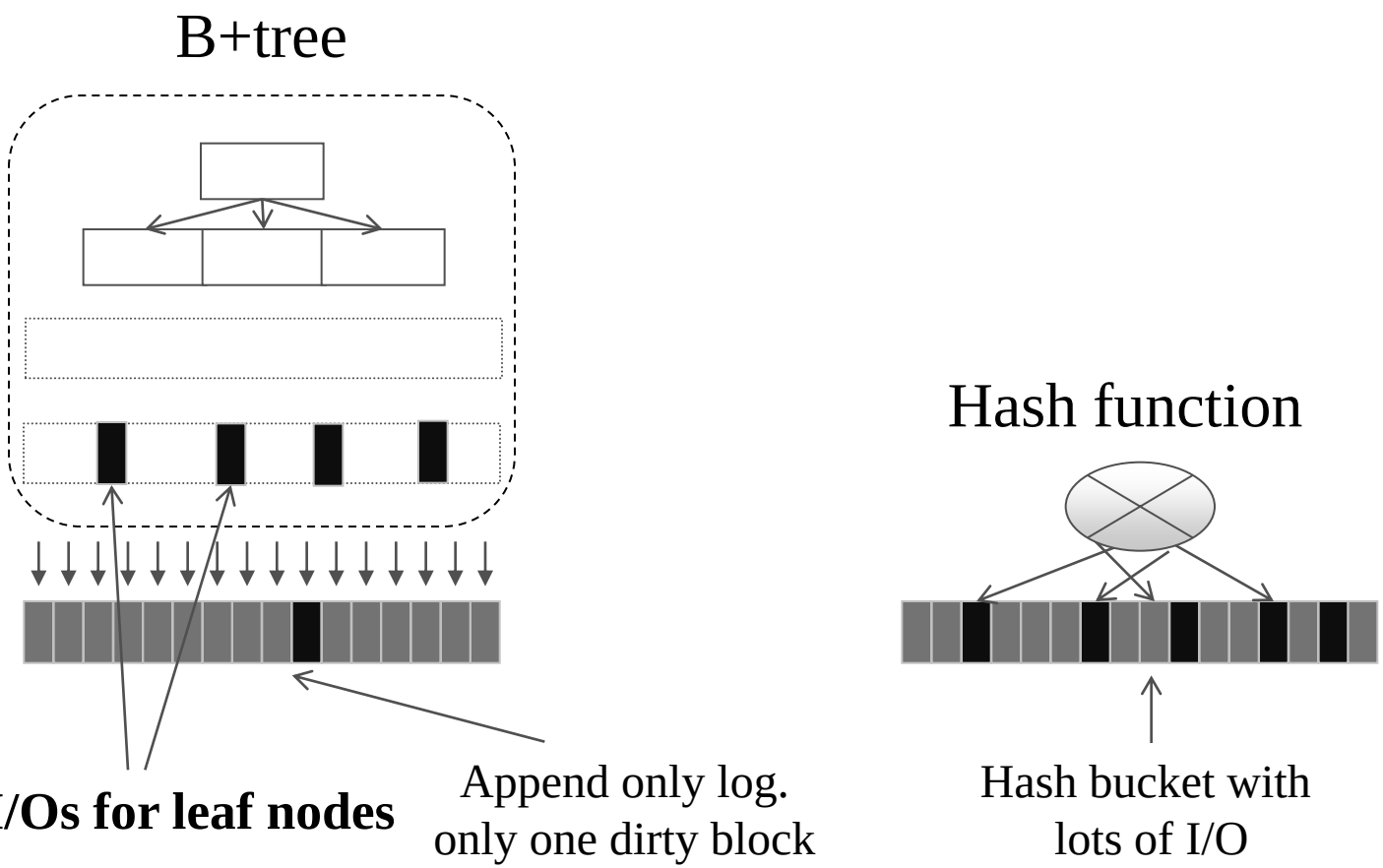
FUJITSU



DBLK Data Structure



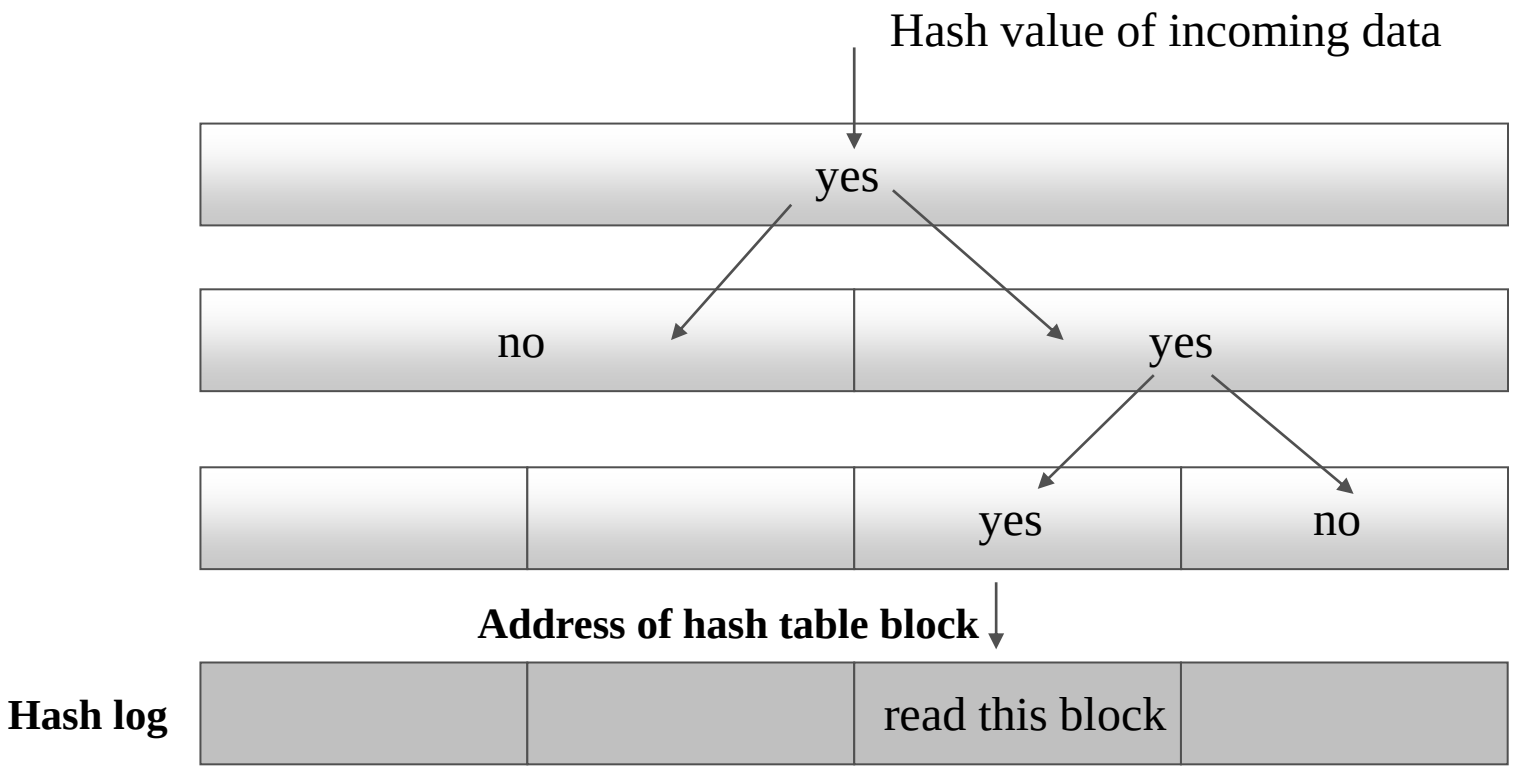
Hash index implementations



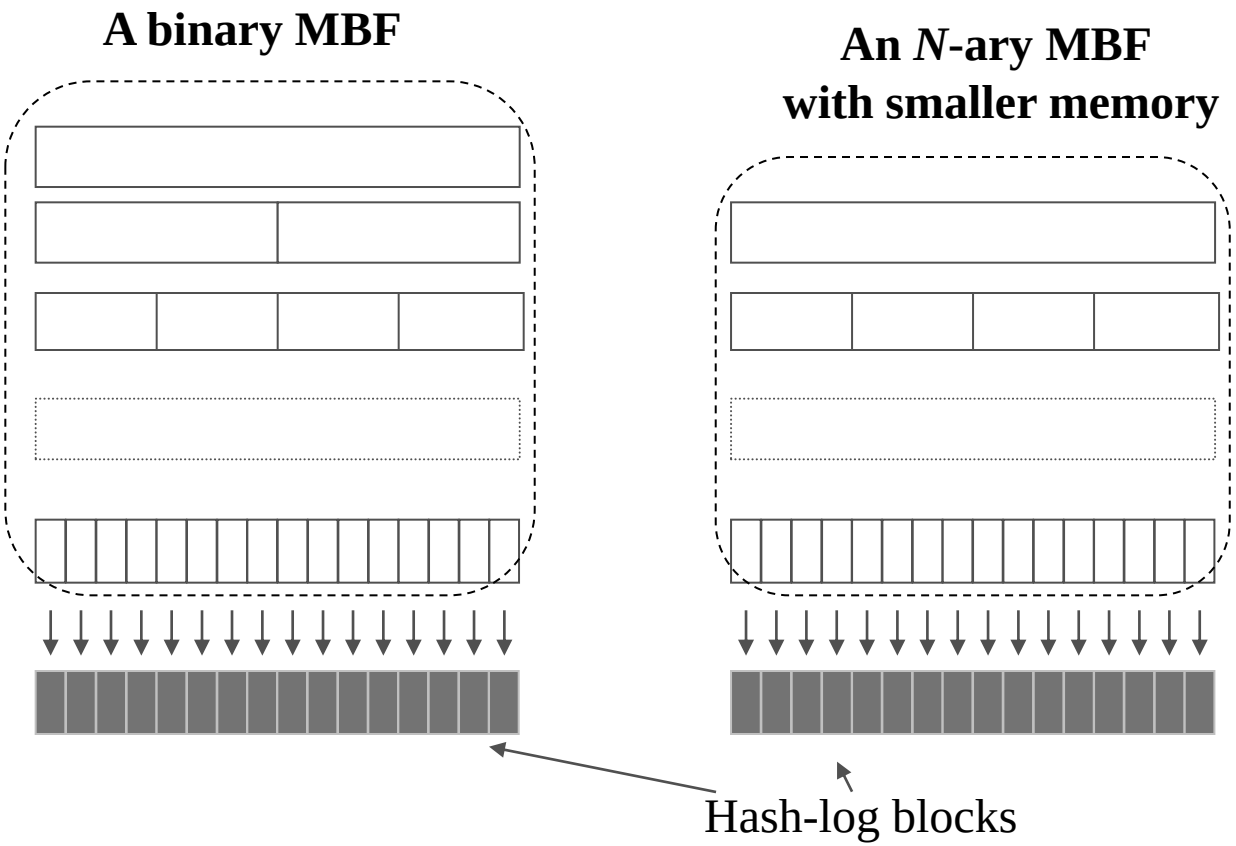
Solution: Multilayer Bloom filter

FUJITSU

- Layers of Bloom filters
  - If it reaches to a lowest filter, read the hash-log block
- False positive means ….
  - Extra I/Os but a correct answer

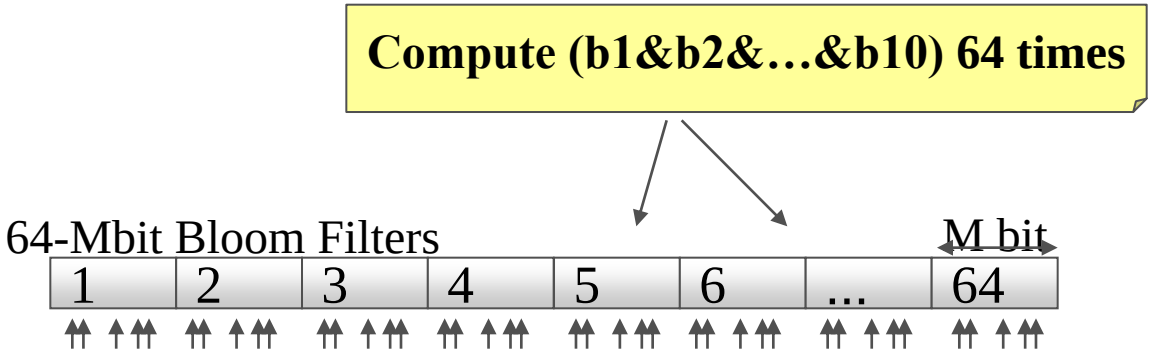


MBF implementation

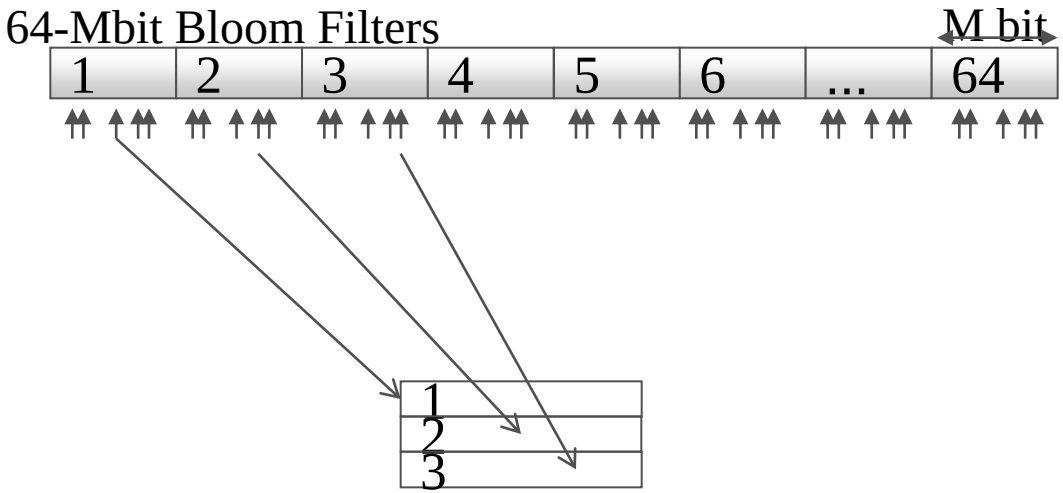


The bigger *N*, the shorter MBF = less memory.

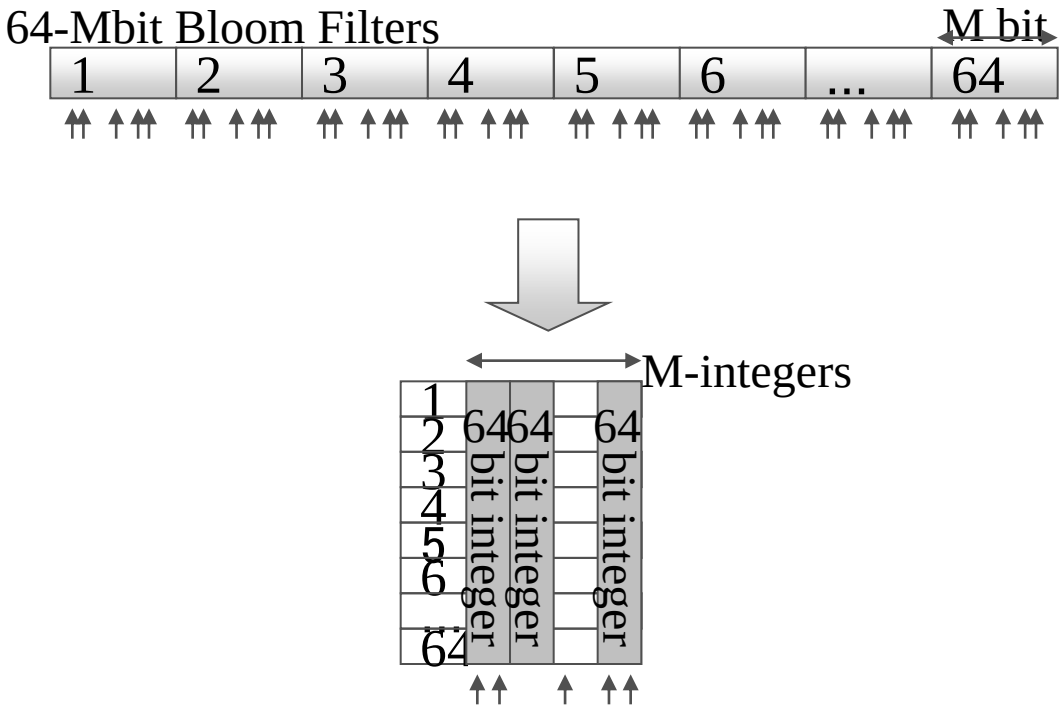
MBF Optimization: MBF-BT



MBF Optimization: MBF-BT

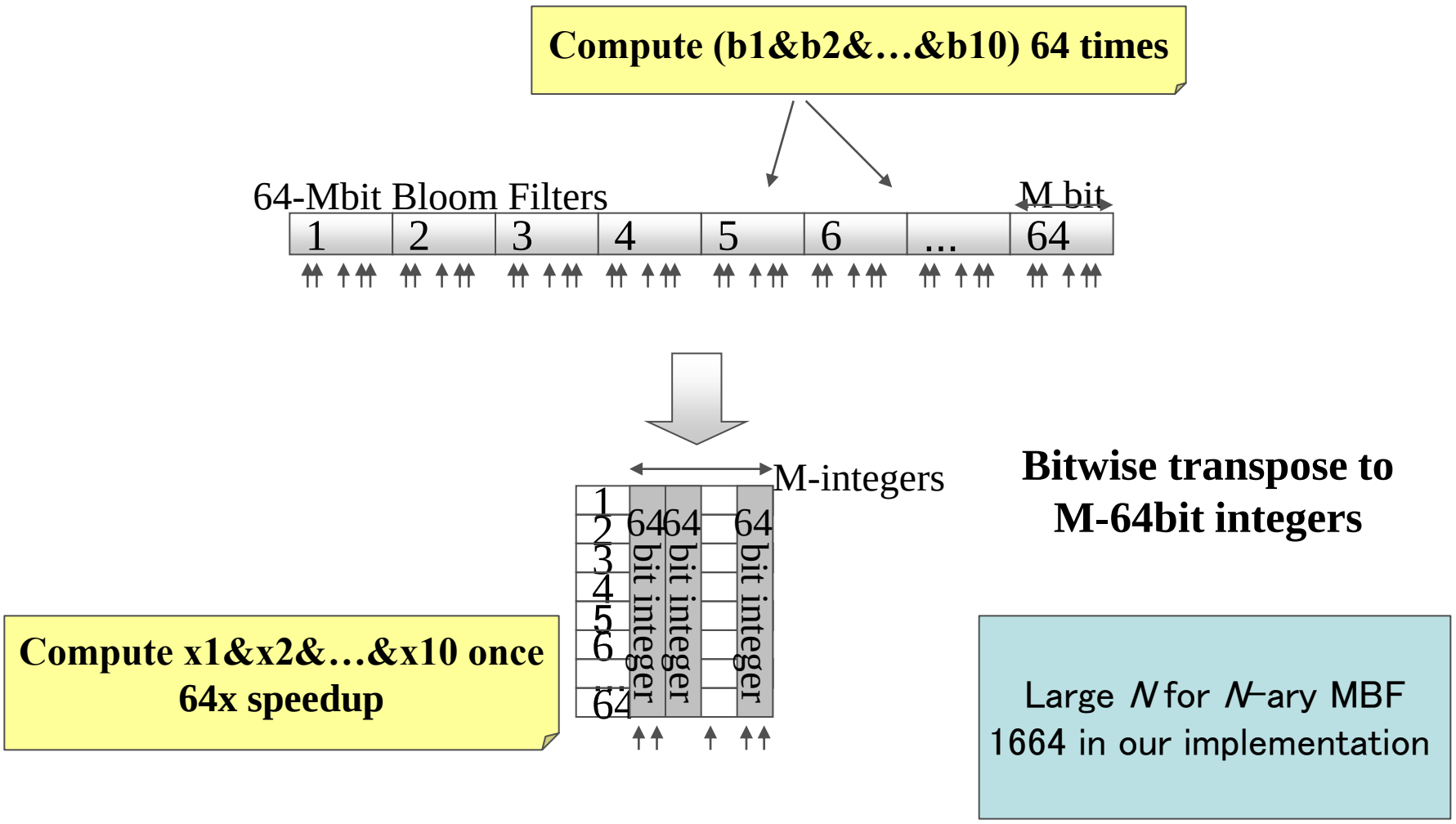


MBF Optimization: MBF-BT



MBF Optimization: MBF-BT

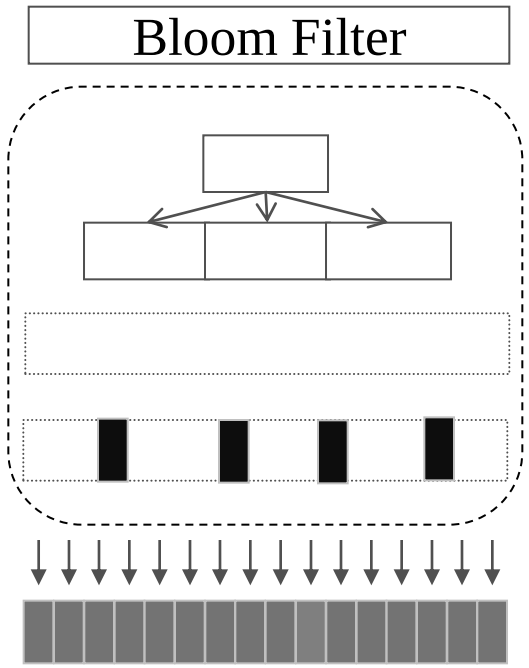
MBF Bitwise Transposition



MBF implementation

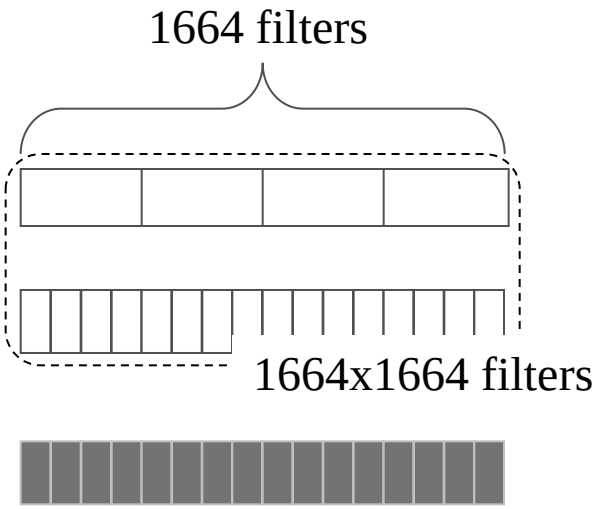
FUJITSU

Bloom filter and B+tree



5.5GB+56GB

Result: 2-layer MBF



5.5GB for each layer, i.e. 11GB

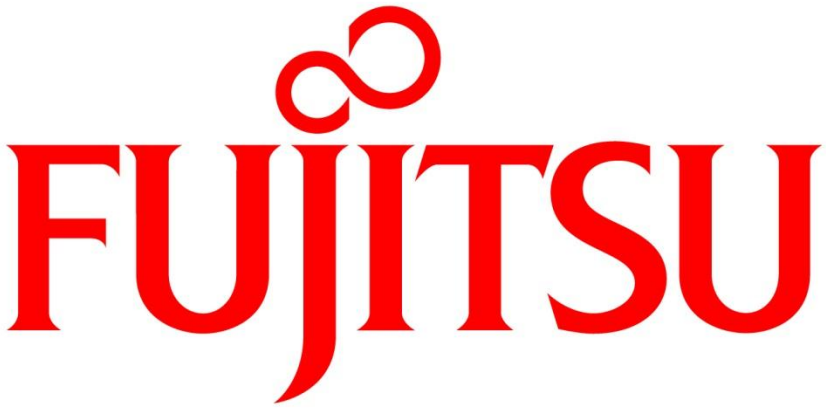
Summary

FUJITSU

- MBF
  - On-memory data structure
  - Helps reduce the latency of primary dedup
- MBF bitwise transposition
  - Reduces CPU cost to lookup bunch of Bloom filters
  - Optimizes access to MBF
- DBLK
  - Smaller latency than other implementation
  - No I/O for hash index

FUJITSU

Thank you!



shaping tomorrow with you