

ShieldReduce: Fine-Grained Shielded Data Reduction

Jingyuan Yang*, Jun Wu*, Ruilin Wu*, Jingwei Li*, **Patrick P. C. Lee**#,
Xiong Li*, and Xiaosong Zhang*

*University of Electronic Science and Technology of China (UESTC)

#The Chinese University of Hong Kong

USENIX ATC 2025

Outsourcing Storage

- Data outsourcing is common in organizations
 - 60% of business data is now stored in cloud
- Two goals:
 - **Storage efficiency:** small storage footprints
 - **Security:** no unauthorized access to outsourced data



The main challenge facing cloud decision-makers is **managing cloud spend** (82%).

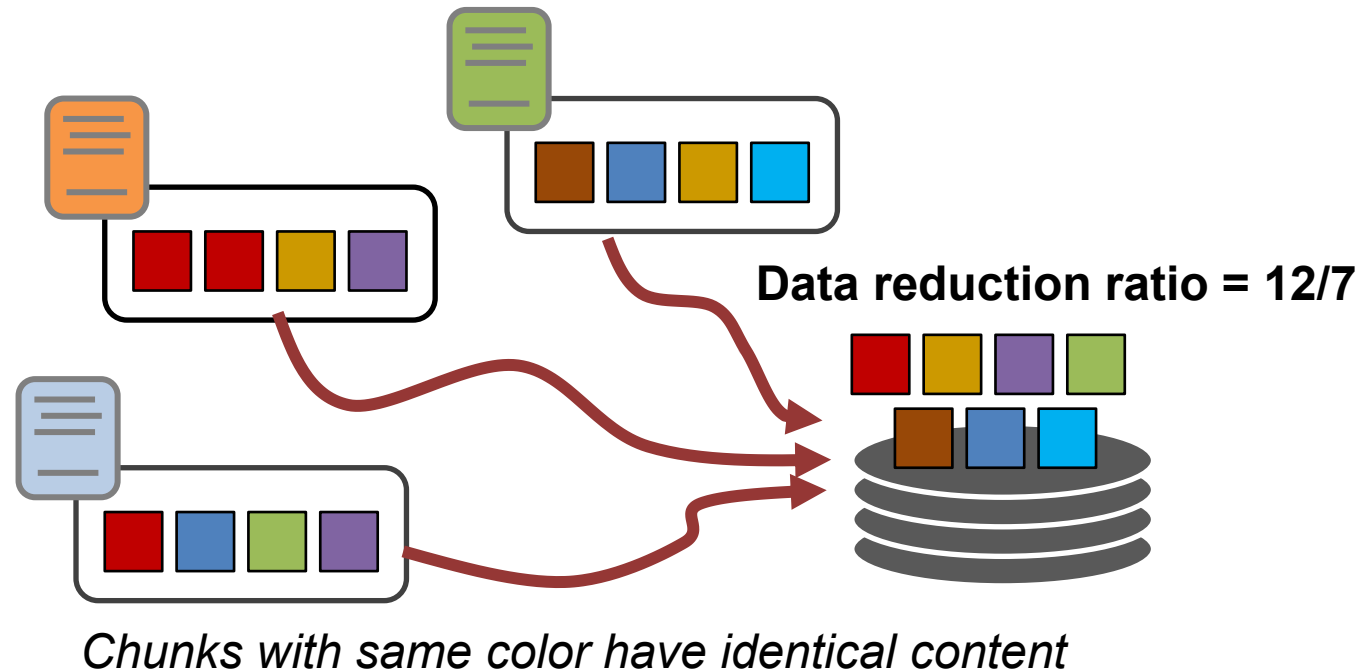


94% of businesses noted **improvements in their security** after moving to the cloud.

Deduplication

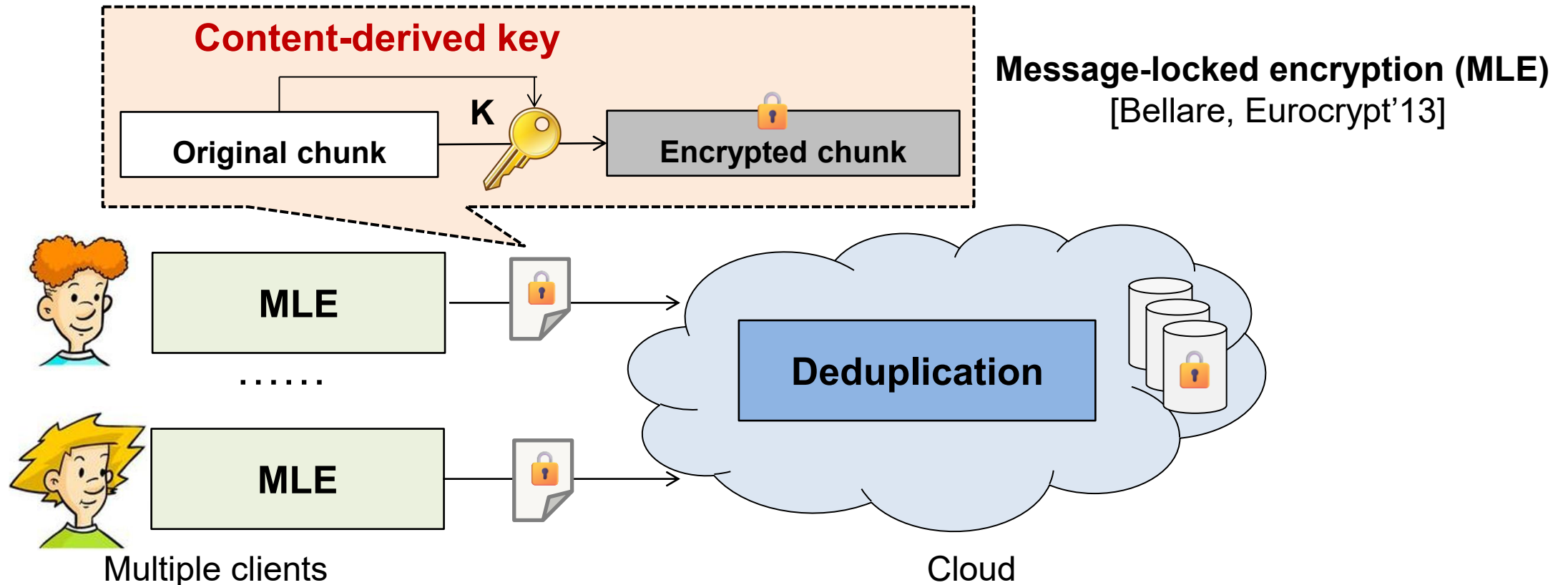
➤ Coarse-grained compression

- Stores only one physical copy for **chunks** with same content
- ~10x data reduction in backup workloads [Wallace, FAST'12]



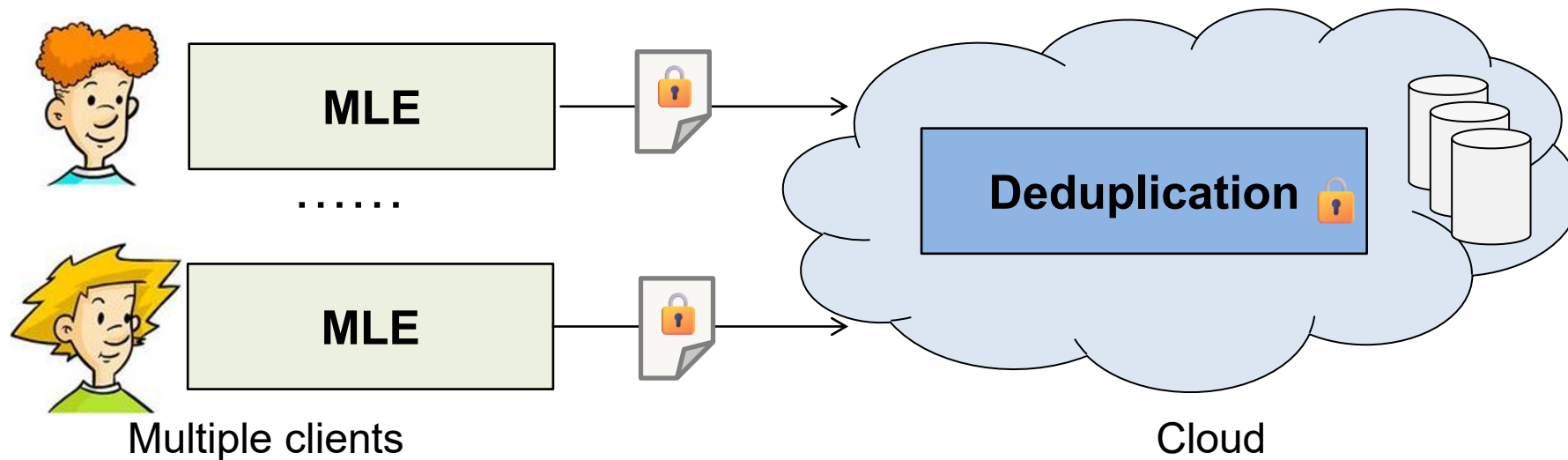
Encrypted Deduplication

- Combine deduplication with encryption for data confidentiality



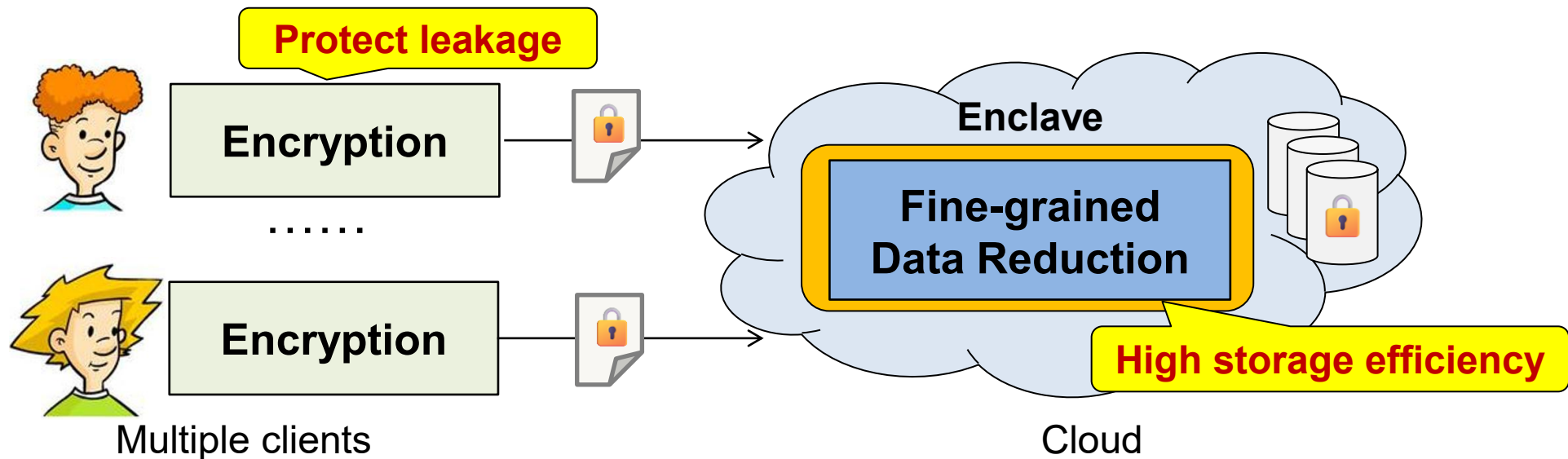
Limitations of Encrypted Deduplication

- Deterministic encryption leaks number of occurrences of chunks
- Encrypted chunks are incompressible



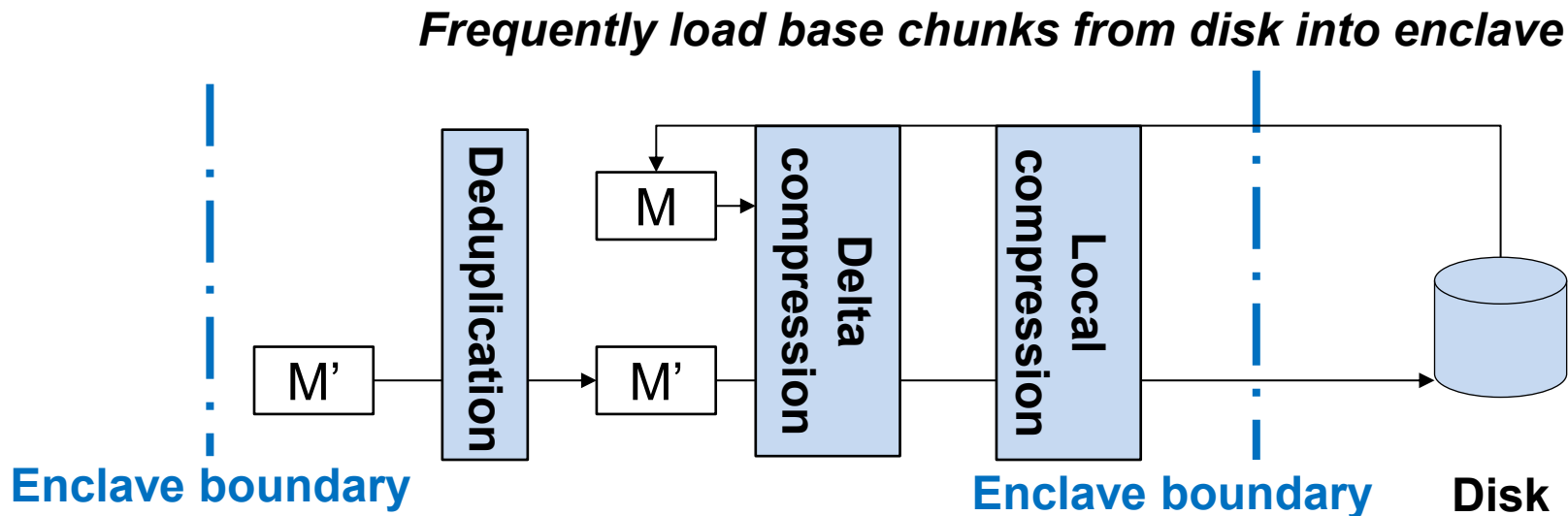
Fine-grained Shielded Data Reduction

- Apply deduplication, delta compression, and local compression
 - **Delta compression** encodes content difference between new chunk M' and previously stored **base chunk** M similar to M'
- Protect whole data reduction via SGX enclave



Challenges

- SGX incurs high context-switch overhead when moving data across enclave boundaries [Weisse, ISCA'17]
- Delta compression incurs **high I/O overhead** in disk I/O and SGX context switching

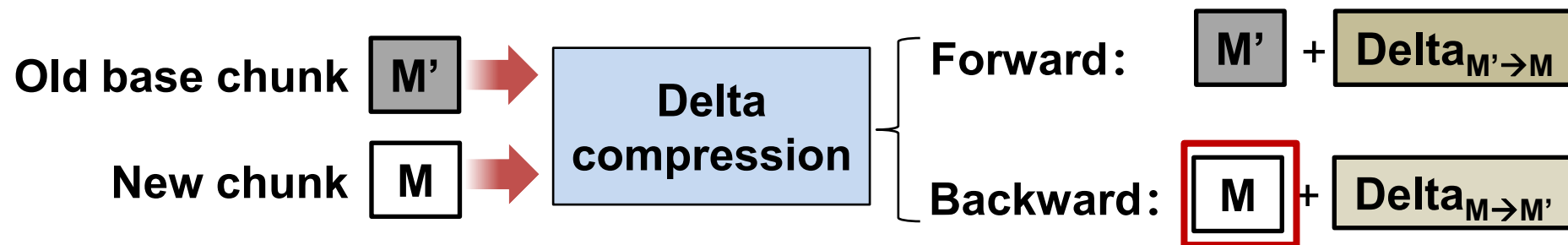


Our Contributions

- **ShieldReduce**: an outsourced backup storage system based on fine-grained shielded data reduction
 - **Bi-directional delta compression** to mitigate I/O overhead
 - **Hybrid inline and offline compression** to limit I/O interference with normal operations
- Testbed experiments
 - Up to **3.5x** upload speedup with bi-directional delta compression
 - Up to **3.6x** storage savings
- Open-sourced prototype

Main Idea

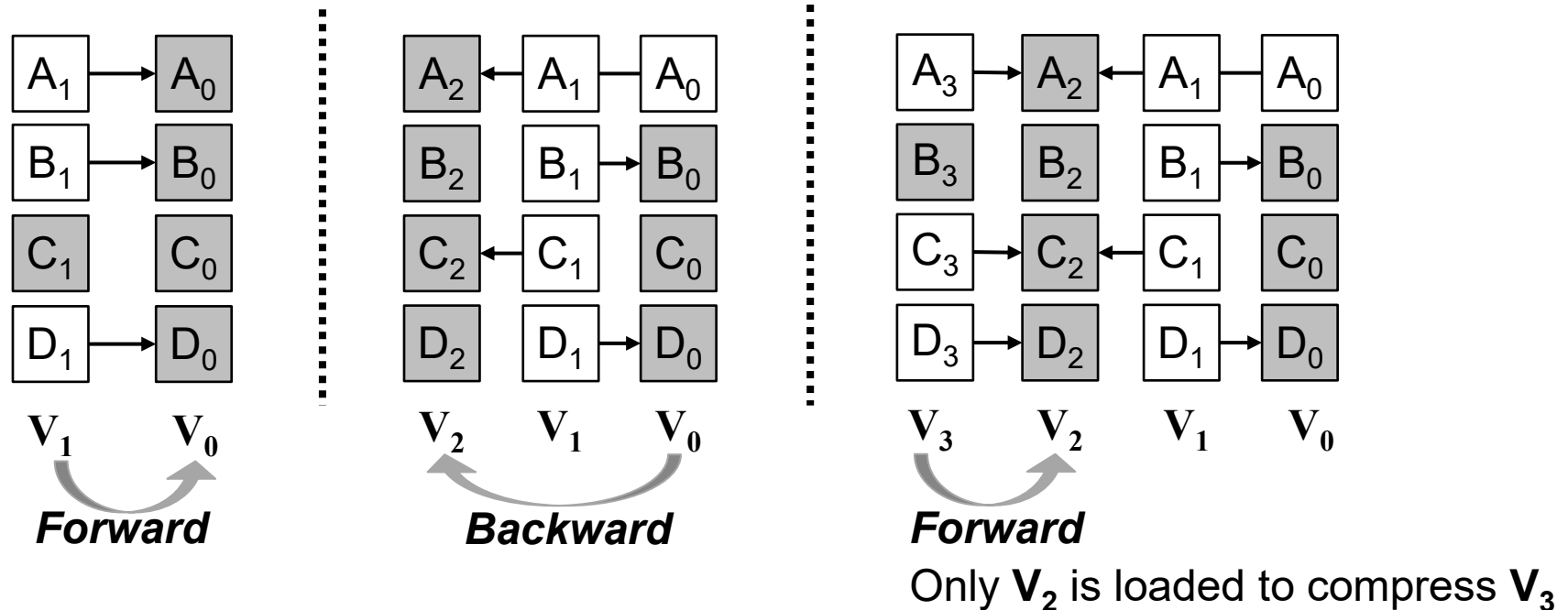
- **Insight:** If M' is modified from M (i.e., M is base chunk of M'), neighbors of M are likely base chunks of neighbors of M'
 - Extend **locality** in deduplication to chunk similarity
 - Batch-loading base chunks from disk to mitigate I/O overhead
- **Bi-directional delta compression**
 - If locality degrades, reconstruct locality via backward delta compression



New base chunks can be stored together

Example

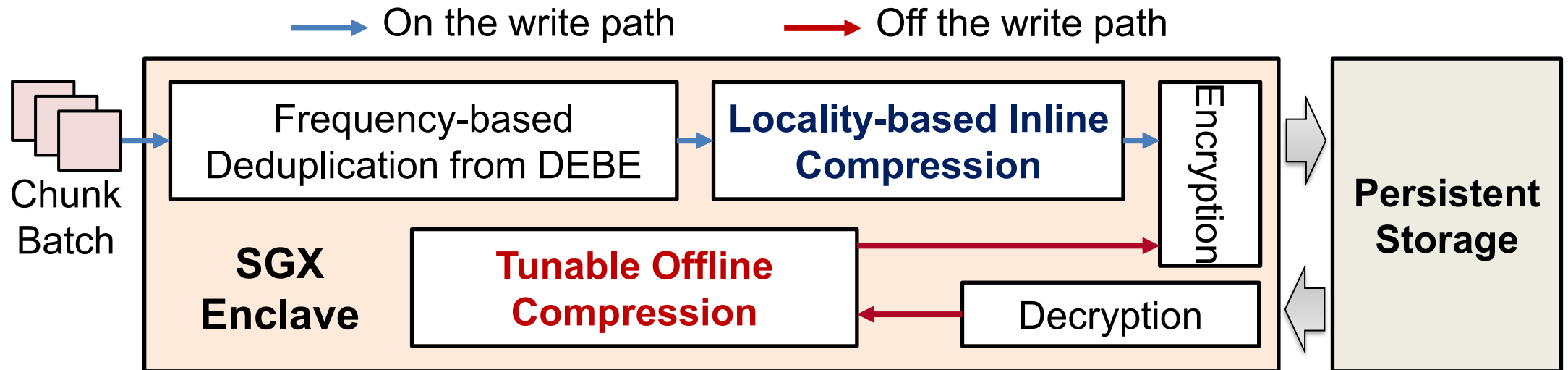
- Suppose backup version V_0 is stored



- **Challenge:** Mitigate backward delta compression's overhead

Design Overview

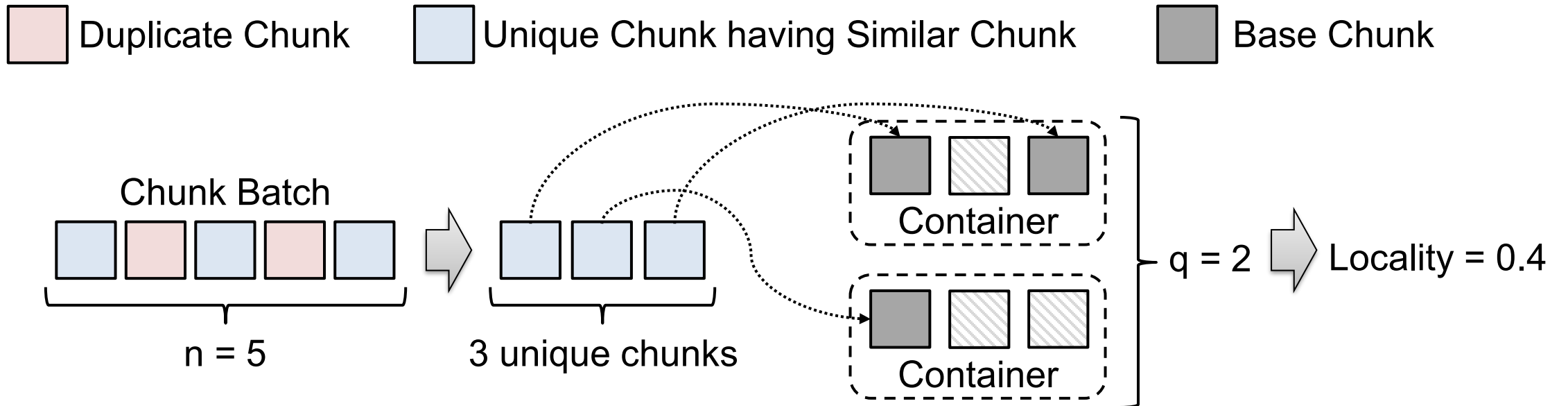
- Extend DEBE^[*] with fine-grained data reduction inside enclave
 - Perform forward delta compression for high performance on the write path
 - Offload backward delta compression off the write path for storage efficiency



[*] "Secure and Lightweight Deduplicated Storage via Shielded Deduplication-Before-Encryption", USENIX ATC 2022.

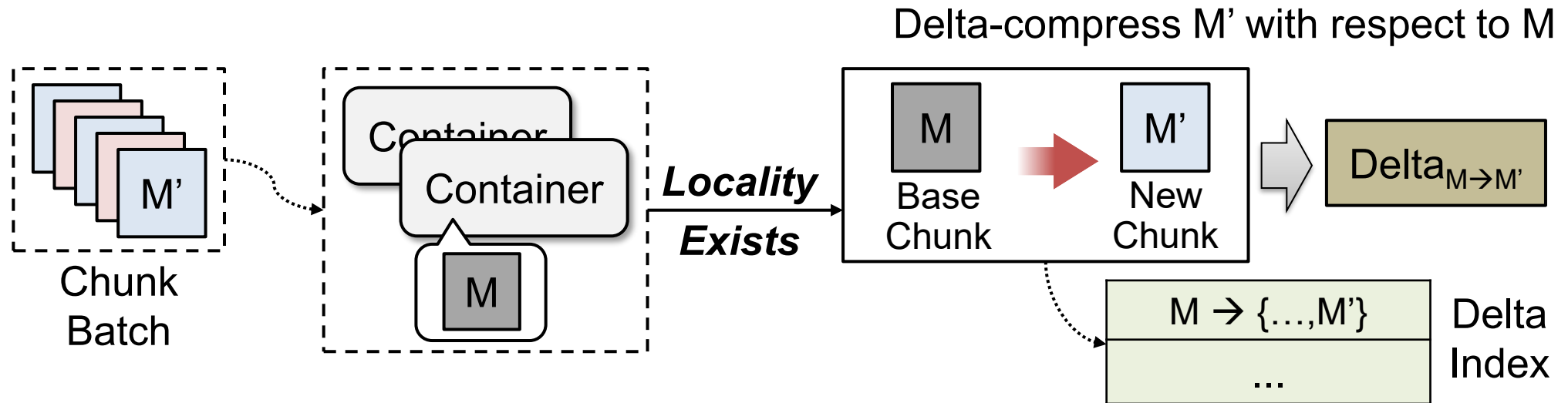
Locality-based Inline Compression

- Measure **physical locality of a chunk batch** based on how base chunks are spread across containers
 - For a batch with n chunks, unique chunks have base chunks across q containers \rightarrow **locality = q/n**



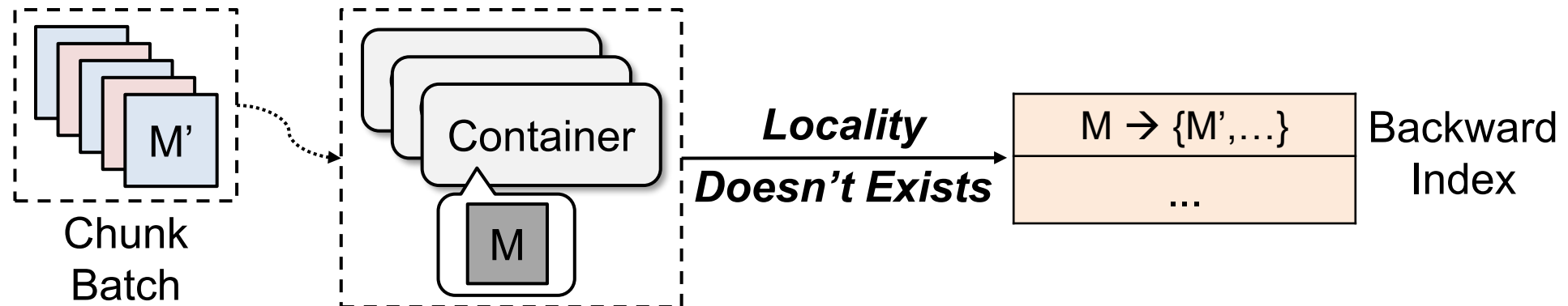
Locality-based Inline Compression

- If locality exists (i.e., q/n is less than some threshold), perform forward delta compression inline
 - Maintain a **delta index** that tracks all delta compression relationships



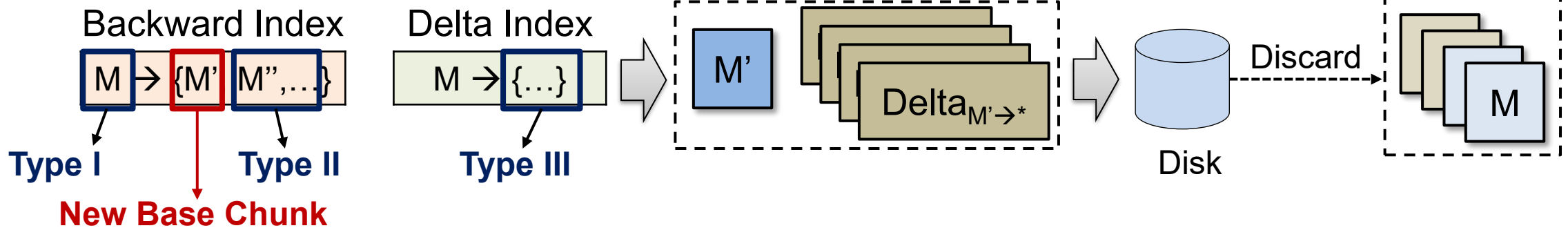
Locality-based Inline Compression

- If locality doesn't exist, prepare for offline compression
 - Maintain a **backward index** that maps a base chunk M to data chunks $\{M'\}$ that will be delta-compressed offline



Tunable Offline Compression

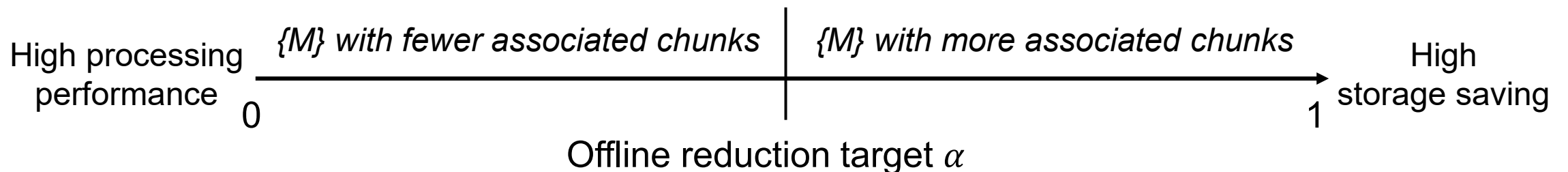
- Perform backward delta compression of M
 - Choose **most recent** chunk M' in M's backward index as new base chunk
 - Delta-compress chunks with respect to M'
 - Type I: Old base chunk M
 - Type II: Deferred chunks (e.g., M'') similar to M
 - Type III: Already delta-compressed with respect to M
 - Replace old persistent copies by new ones corresponding to M'



Tunable Offline Compression

- Tunable performance-storage trade-off via a user-specified **offline reduction target** α (smaller α means to reduce more data)
 - Sort old base chunks by number of associated data chunks
 - Apply backward delta compression, starting with old base chunks with fewest associated chunks
 - Stop backward delta compression for old base chunks when α is reached

Backward delta compression on old base chunks **in order**
until α is satisfied



Experimental Setup

- Testbed based on Alibaba Cloud
 - Three real-world datasets (Linux, Web and Docker) and one synthetic dataset (SimOS)
- Three shielded execution baselines:
 - **DEBE** [Yang, ATC'22]: deduplication and local compression
 - **SecureMeGA**: performance-oriented design [Zou, ATC'22] adapted to shielded execution
 - **ForwardDelta**: forward delta compression only

Overall Data Reduction

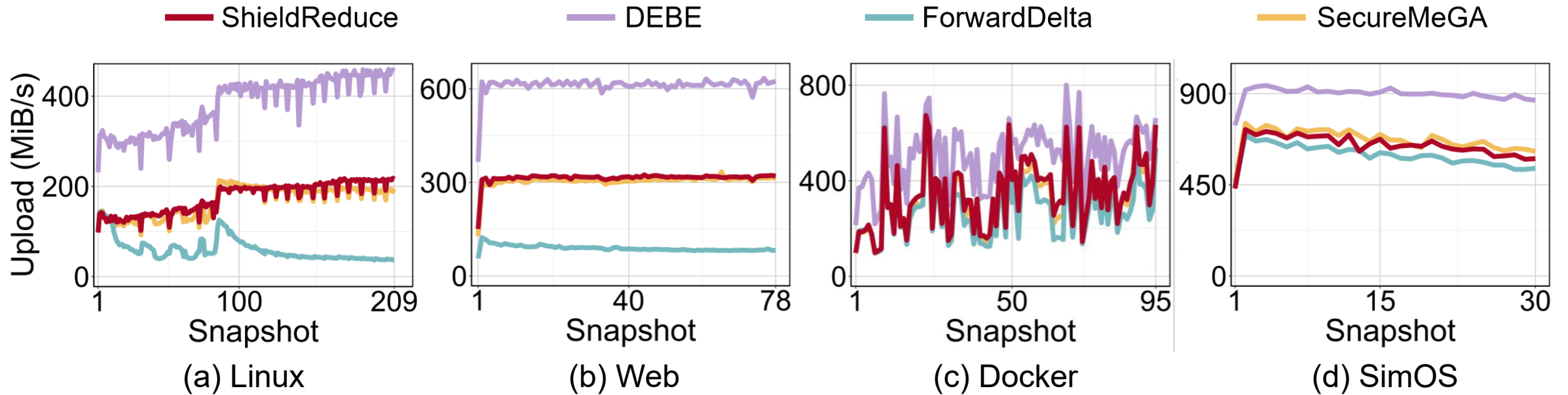
A parameter that configures performance-storage trade-off in offline compression

Datasets	DEBE	ShieldReduce				Secure MeGA	Forward Delta
		$\alpha=0$	$\alpha=0.5$	$\alpha=0.7$	$\alpha=1$		
Linux	5.8	25.8	12.2	8.9	6.3	12.1	25.1
Web	13.1	58.6	27.9	19.9	14.0	16.1	60.6
Docker	8.6	14.9	14.4	12.9	10.3	14.0	15.0
SimOS	59.3	63.6	63.6	63.6	61.0	60.2	63.3

➤ ShieldReduce achieves high storage savings

- Up to 4.5x and 3.6x data reduction ratios compared with DEBE and SecureMeGA, respectively

Inline Performance



- ShieldReduce preserves high performance on the write path
 - 1.1-3.5x throughput gains compared to ForwardDelta
 - Slower than DEBE due to delta compression

Enclave Overhead

Dataset		DEBE	ForwardDelta	ShieldReduce		
				Inline	Offline	
Linux	#ECall		0.2M		209	
	#OCall	Index queries	0.2M	0.4M		963.6M
		Index updates	-	-	0.04M	
		Data transfers	0.03M	2.6M	0.06M	
		Total	0.23M	3.0M	0.5M	

- ShieldReduce reduces #OCalls for ForwardDelta by up to 83.3% on the write path
- ShieldReduce (with offloading to reconstruct locality) has up to 4.6x data reduction per OCall

Conclusion

- ShieldReduce: a secure outsourced backup storage system based on fine-grained shielded data reduction
 - **Bi-directional delta compression + Hybrid inline and offline compression**



- Source code: <https://github.com/YangJingyuan99/shieldreduce>

