Yaofeng TU, Rong XIAO, Yinjun HAN, Zhenghua CHEN, Hao JIN, Xuecheng QI, Xinyuan SUN, 2023. DDUC: an erasure-coded system with decoupled data updating and coding. *Frontiers of Information Technology & Electronic Engineering*, 24(5):716-730. https://doi.org/10.1631/FITEE.2200466

DDUC: an erasure-coded system with decoupled data updating and coding

Key words: Concurrent update; High reliability; Erasure code; Consistency; Distributed storage system

Corresponding author: Yinjun HAN
E-mail: han.yinjun@zte.com.cn
ORCID: https://orcid.org/0000-0001-5578-2351

Motivation

1. The existing distributed erasure code (EC) systems do not support high-concurrency scenarios well. To support strong consistency among EC stripes, the performance of concurrent reads and writes needs to be sacrificed. This study is dedicated to solving the problem of how to support high-concurrency reads and writes while ensuring data consistency and reliability.

2. The persistent memory (PMem) technology has been developed rapidly. PMem has byte-addressable, low-latency, and non-volatile characteristics, which provides new opportunities for designing high-performance distributed storage systems.

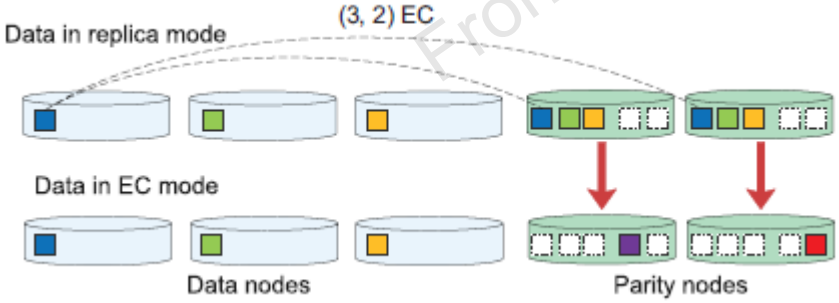
Main idea

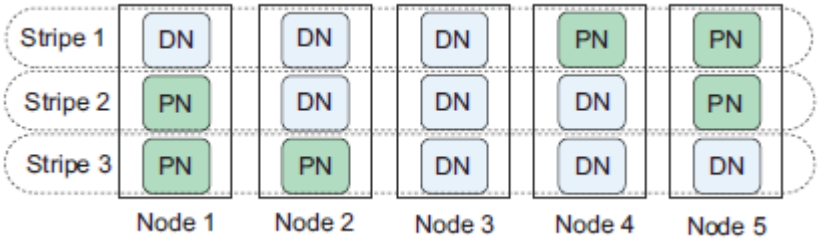
We use a hybrid redundancy mode of replication and EC to achieve high concurrency and reliability in decoupling data updating and EC encoding using an innovative placement policy and an update method. The main contributions include the following:

1. A placement policy that combines replicas and parity blocks to realize the decoupling of data updating and EC encoding.
2. A two-phase data update method.
3. A lightweight log mechanism based on persistent memory.

Method

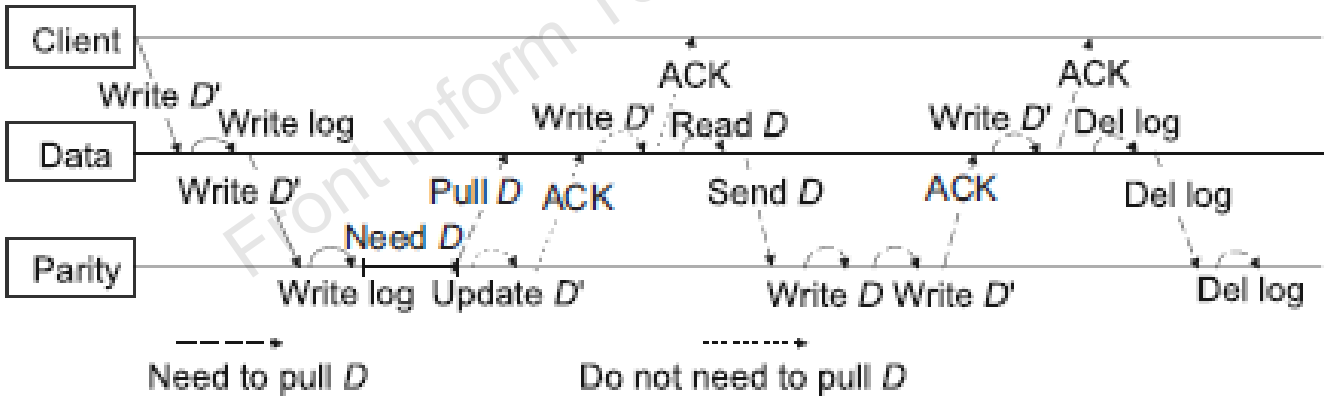
1. For the hybrid placement policy, hot data use replication and cold data use EC. The data blocks in the (*N*, *M*) EC system are managed as *N* groups with *M*+1 replication blocks, and the redundant blocks in the same stripe are all placed in the parity nodes, so that the parity nodes can implement EC encoding locally. To balance the system load, parity nodes are represented by different physical nodes in different stripes.





Method (Cont'd)

2. For the (*N*, *M*) EC system, the update phase is updated according to the *M*+1 replica mode, and the EC phase is completed by the parity node independently for EC coding, and the update process is as follows:



Method (Cont'd)

3. The DDUC system ensures data reliability by saving both old and new versions of the data block in the parity node during concurrent updates.

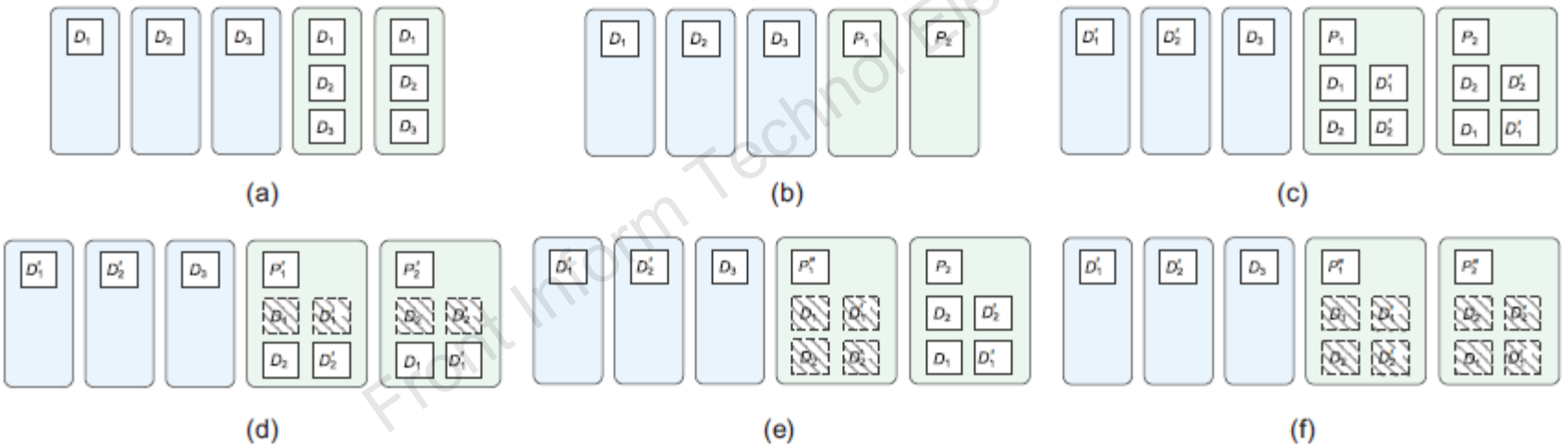


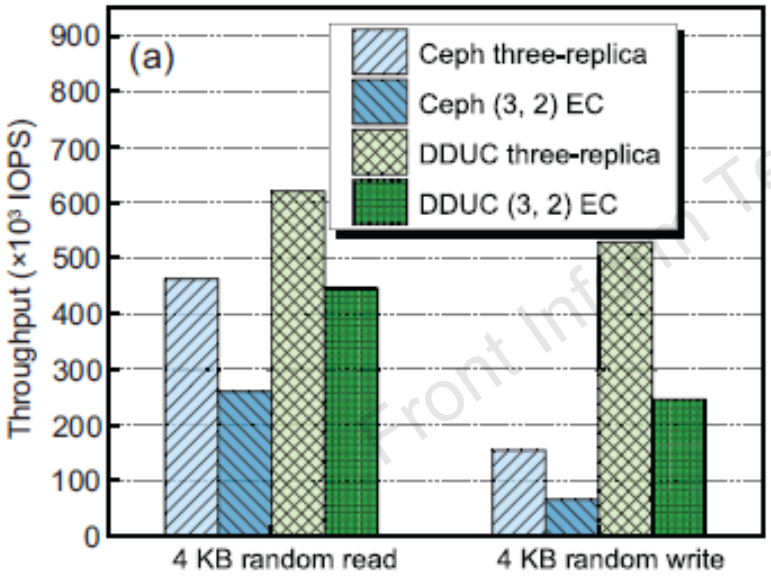
Fig. 5 Data reliability: (a) data blocks in the replica mode; (b) data blocks and parity blocks in the EC mode; (c) clients update D_1 and D_2 into D_1' and D_2', respectively; (d) parity node 1 and parity node 2 perform EC encoding inconsistently; (e) parity node 1 performs EC encoding but parity node 2 does not; (f) parity nodes go back in the EC mode again (D: data block; P: parity block; EC: erasure code)

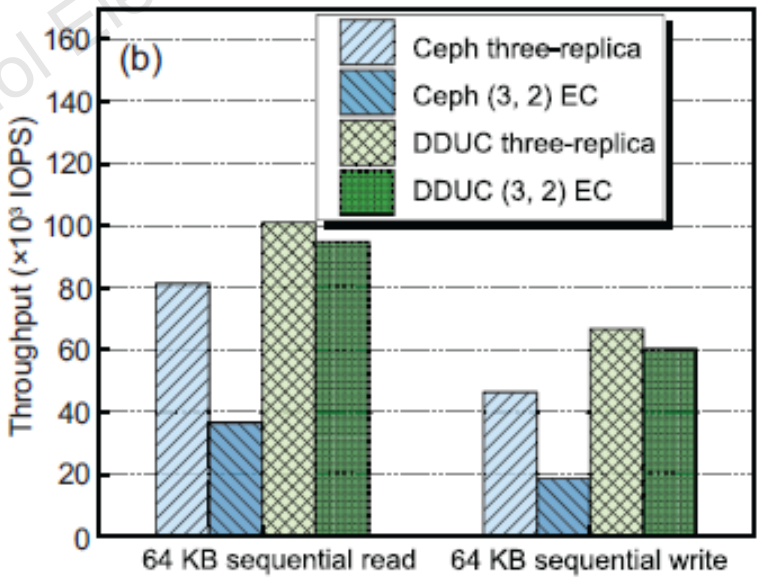
Method (Cont’d)

4. Based on the byte-addressable, low-latency, and non-volatile characteristics of PMem, we design a lightweight log mechanism. We design the log in the form of a 64-bit block ID and store it to PMem using append write. The log is written before data block updating, and deleted after updating. So, we can guarantee the consistency between data blocks.

Major results

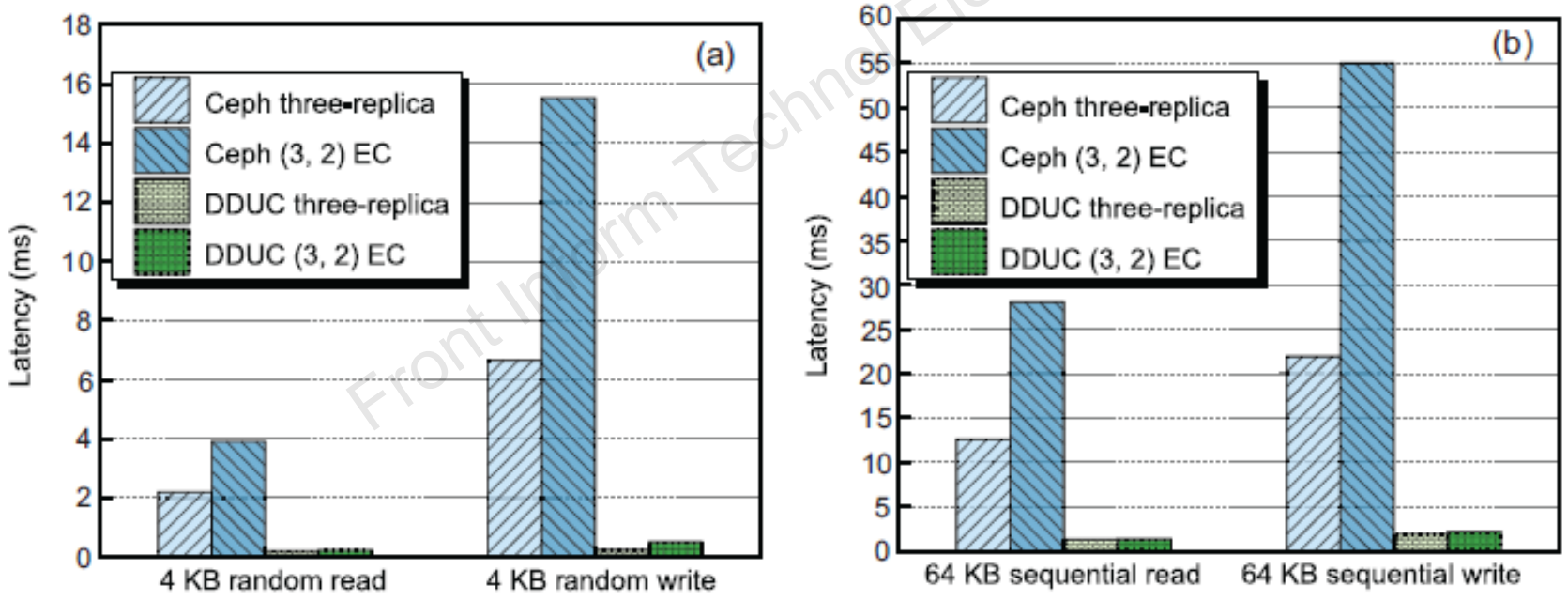
IOPS test





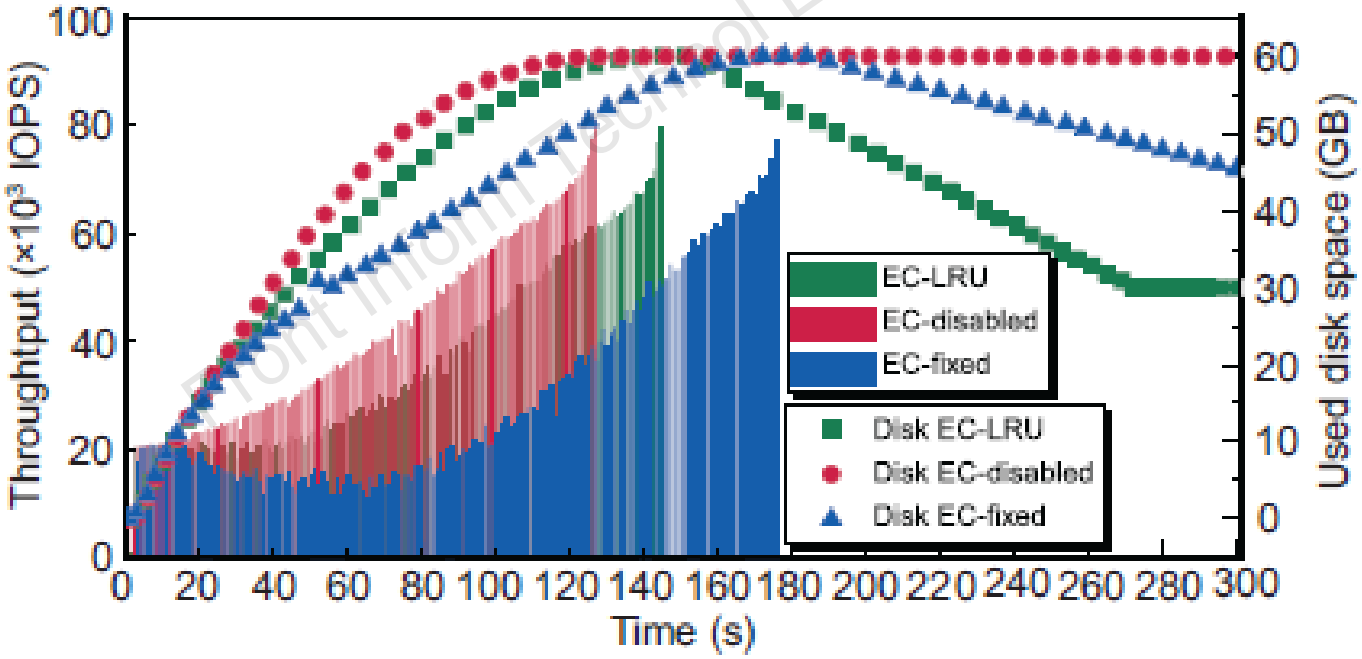
Major results (Cont'd)

Latency test



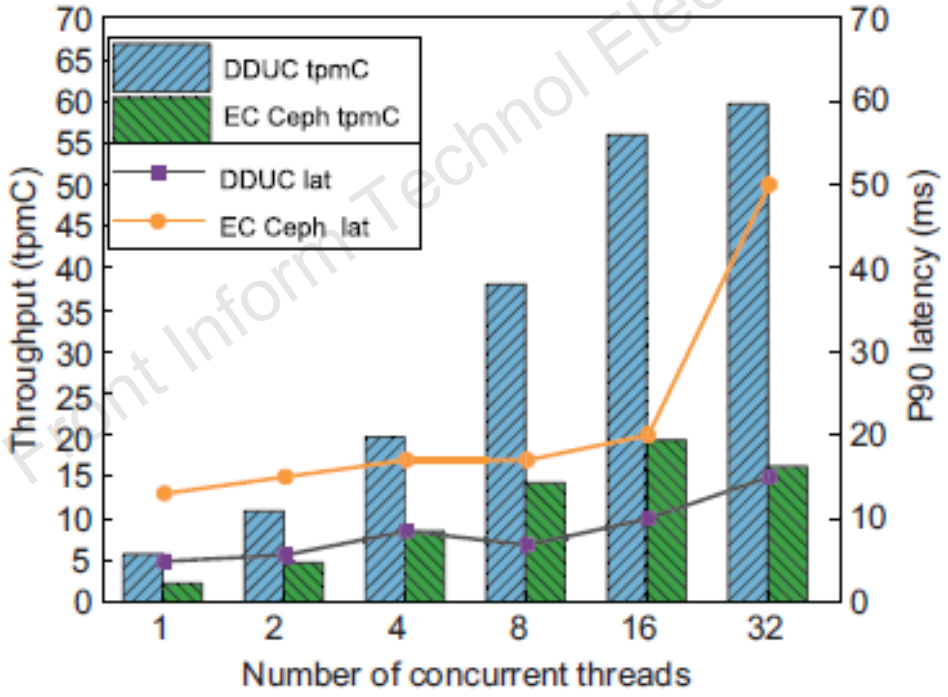
Major results (Cont'd)

Disk space efficiency test



Major results (Cont'd)

Work load test



Conclusions

1. We observed: the existing distributed EC systems do not support high-concurrency scenarios well.
2. We designed:
   a placement policy that combines replicas and parity blocks;
   a two-phase data update method;
   a lightweight log mechanism based on persistent memory.
3. We implemented: a storage system which decouples data updating and EC encoding, and takes features of EC's high storage efficiency and replication's good performance into account.
4. We evaluated: the concurrent access performance of the proposed storage system is 1.70–3.73 times that of Ceph, and the latency is only 3.4%–5.9% that of Ceph.

Yaofeng TU received the PhD degree from Nanjing University of Aeronautics and Astronautics, China, in 2019. He is currently a research professor in ZTE Corporation. His main research interests include distributed computing, big data system, and machine learning.

Rong XIAO, master, senior engineer. She is currently a research engineer in ZTE Corporation. Her main research interests include storage systems and distributed systems.

Yinjun HAN, master, senior engineer. He is currently a research engineer in ZTE Corporation. His main research interests include storage systems and database systems.

Zhenghua CHEN, senior engineer. He is currently a research manager in ZTE Corporation. His main research interests include storage systems and distributed systems.

Hao JIN, master, senior engineer. He is currently a research manager in ZTE Corporation. His main research interests include storage systems and distributed systems.

Xuecheng QI received the PhD degree in software engineering from the East China Normal University, Shanghai, China, in 2022. He is currently a research engineer in ZTE Corporation. His main research interests include in-memory database and distributed storage.

Xinyuan SUN received the MS degree of Computer Science from Boston University, USA, in 2022. He is currently a junior engineer in ZTE Corporation. His main research interests include data analysis, distributed database system, and artificial intelligence.