Технічні науки

УДК 004.41:004.65

Шабатин Павло Євгенійович

магістр комп'ютерних наук,

асистент кафедри інженерії програмного забезпечення

Київський національний університет імені Тараса Шевченка

Shabatin Pavlo

Master of Computer Science,

Assistant at the Department of Software Engineering

Taras Shevchenko National University of Kyiv

Лукашов Олександр Валентинович

магістр комп'ютерної інженерії,

асистент кафедри інженерії програмного забезпечення

Київський національний університет імені Тараса Шевченка

Lukashov Oleksandr

Master of Computer Engineering,

Assistant at the Department of Software Engineering

Taras Shevchenko National University of Kyiv

DOI: https://doi.org/10.25313/2520-2057-2026-7-12106

АНАЛІТИЧНЕ ПОРІВНЯННЯ СТРАТЕГІЙ ЛІНИВОГО ТА НЕТЕРПЛЯЧОГО ВІДНОВЛЕННЯ У РОЗПОДІЛЕНИХ СХОВИЩАХ

ANALYTICAL COMPARISON OF LAZY AND EAGER REPAIR IN DISTRIBUTED STORAGE

Анотація. *Розподілені сховища з надлишковим кодуванням потребують стратегії відновлення блоків після відмов вузлів. Нетерпляче відновлення (ER) запускається негайно, ліниве (LR) відкладає ремонт до критичного порогу, задаючи компроміс між надійністю та витратами пропускної здатності. Запропоновано марківську модель обох стратегій; виведено вирази для ймовірності втрати даних і витрат на відновлення та замкнену умову переходу λ*, що визначає межу доцільності лінивого відновлення.*

Ключові слова: *розподілені сховища, надлишкове кодування, ліниве відновлення, нетерпляче відновлення, ланцюги Маркова, ймовірність втрати даних, витрати на відновлення.*

Summary. *Distributed storage systems with erasure coding require a strategy for repairing blocks after node failures. Eager repair (ER) starts immediately; lazy repair (LR) defers repair until a critical threshold, trading a wider vulnerability window for bandwidth savings. We propose a Markov model that formalizes both strategies and derive expressions for the data loss probability, repair costs, and a closed-form transition condition λ* that determines when lazy repair is appropriate.*

Key words: *distributed storage, erasure coding, lazy repair, eager repair, Markov chains, data loss probability, repair bandwidth.*

Вступ. Сучасні децентралізовані сховища – Filecoin, Codex, Walrus – використовують надлишкове кодування (erasure coding) для захисту від відмов вузлів [1]. У схемі (n, k) об'єкт розбивається на n блоків, і для його відновлення достатньо будь-яких $k < n$ з них. При відмові вузла система втрачає блок, зменшуючи запас надлишковості. Якщо до наступного відновлення відбудеться ще $n - k$ відмов, дані втрачаються безповоротно.

Ключове проектне рішення – *коли* запускати відновлення. Стратегія ER усуває дефіцит негайно після кожної відмови, підтримуючи повну надлишковість, але генерує постійний ремонтний трафік. Стратегія LR відкладає відновлення до моменту, коли кількість наявних блоків досягає критичного порогу δ або минає часовий ліміт D, знижуючи витрати пропускної здатності ціною ширшого вікна вразливості.

Системи Walrus і Codex декларують використання LR [2], проте не надають аналітичного обґрунтування умов його доцільності. Існуючі теоретичні роботи аналізують надійність розподілених сховищ через ланцюги Маркова [3], але не порівнюють LR та ER у єдиній формальній рамці.

Внески цієї роботи: - Єдина марківська модель для ER та LR із загальними параметрами λ, D, n, k. - Аналітичний вираз для P_loss та порівняльний аналіз B_repair для обох стратегій. - Замкнена умова переходу λ*, що визначає межу доцільності LR.

Аналітична модель

Модель системи

Розглядаємо один об'єкт, закодований у схемі (n, k): n блоків, мінімум k для декодування. Стан системи $s \in \{k, k+1, \ldots, n\}$ – кількість наявних блоків. Відмови блоків незалежні; кожен блок відмовляє зі швидкістю λ. Відновлення одного блоку займає час, розподілений експоненційно з середнім t_r (мережева передача + запис), $\mu = 1/t_r$.

Стан втрати: $s = k - 1$ – поглинаючий стан, з якого відновлення неможливе.

Нетерпляче відновлення (ER)

При ER відновлення починається негайно після кожної відмови. Система поводиться як ланцюг народження-загибелі зі станами $\{k, \ldots, n\}$:

- **Перехід $s \to s - 1$** (відмова): швидкість $\alpha_s = s \cdot \lambda$
- **Перехід $s \to s + 1$** (відновлення, $s < n$): швидкість $\mu_{ER} = 1 / t_r$

Ймовірність втрати визначається як першопрохідна ймовірність досягнення стану k−1 до повернення до стану n [3]. При μ_ER >> λ втрата вимагає n−k послідовних відмов до першого завершення ремонту. Ймовірність відмови до ремонту у стані s становить s·λ / (s·λ + μ_ER) ≈ s·λ / μ_ER. Перемноживши по всіх станах від n до k+1, отримуємо формулу (1):

$$P_{\text{loss}}^{\text{ER}} \approx \prod_{s=k+1}^{n} \frac{s\cdot\lambda}{\mu_{\text{ER}}} = \frac{n!}{k!}\cdot\left(\frac{\lambda}{\mu_{\text{ER}}}\right)^{n-k} \quad (1)$$

Очікувані витрати на відновлення: B_repair^ER ≈ n·λ·B_block за одиницю часу, де B_block – розмір одного блоку. При μ_ER >> λ система переважно перебуває у стані n, тому частота ремонтів приблизно дорівнює n·λ.

Ліниве відновлення (LR)

При LR відновлення не запускається, поки s > δ = k + 1. Коли s досягає δ, система чекає D одиниць часу перед початком відновлення. Відповідно, вікно вразливості між входом у стан δ і завершенням відновлення становить D + t_r.

У стані s = k + 1 система перебуває перед єдиним кроком до втрати. Ймовірність втрати обчислюється як ймовірність того, що наступна відмова відбудеться раніше, ніж завершиться відновлення, відповідно до формули (2):

$$P_{\text{loss}}^{\text{LR}} = \frac{(k+1)\cdot\lambda}{(k+1)\cdot\lambda+\mu_{\text{LR}}} \quad (2)$$

де μ_LR = 1 / (D + t_r). Підставляючи, отримуємо формулу (3):

$$P_{\text{loss}}^{\text{LR}} = \frac{(k+1)\cdot\lambda\cdot(D+t_r)}{1+(k+1)\cdot\lambda\cdot(D+t_r)} \quad (3)$$

При LR ремонт відбувається лише при досягненні порогу δ. Нехай ν_LR – частота досягнення δ з початкового стану n. Оскільки у LR ремонту немає вище δ, система дрейфує вниз виключно через відмови; час першого проходження від стану n до k+1 складається з n−k−1 незалежних фаз, і ν_LR ≈ λ / Σ_{s=k+2}^{n} (1/s). Звідси отримуємо формулу (4):

$$B_{\text{repair}}^{\text{LR}} \approx \nu_{\text{LR}} \cdot (n-k) \cdot B_{\text{block}}, \quad \nu_{\text{LR}} \ll n\lambda \text{ при } \lambda \ll \mu_{\text{ER}} \tag{4}$$

Умова переходу λ*

Визначимо λ* як значення λ, за якого обидві стратегії забезпечують однакову P_loss. При малих λ, де (k+1)·λ·(D + t_r) << 1, права частина лінійна, як показано у формулі (5):

$$P_{\text{loss}}^{\text{LR}} \approx (k+1) \cdot \lambda \cdot (D + t_r) \tag{5}$$

Прирівнюючи до P_loss^ER і ділячи на λ*, отримуємо формулу (6):

$$\frac{n!}{k!} \cdot \frac{(\lambda^*)^{n-k-1}}{\mu_{\text{ER}}^{n-k}} = (k+1) \cdot (D + t_r) \tag{6}$$

Розв'язок для n − k ≥ 2 подано у формулі (7):

$$\boxed{\lambda^* \approx \left[\frac{(k+1)! \cdot (D+t_r) \cdot \mu_{\text{ER}}^{n-k}}{n!}\right]^{\frac{1}{n-k-1}}} \tag{7}$$

Згідно з формулою (7), для n − k ≥ 2 існує єдина точка перетину λ* (ліва частина зростає як λ^(n−k−1), права є константою відносно λ). Для n − k = 1 ліниве та нетерпляче відновлення мають різні P_loss при будь-якому λ > 0, і LR завжди гірше за ER на величину, пропорційну D.

Практичний наслідок: λ* спадає зі збільшенням D – чим довша затримка LR, тим вужчий діапазон λ, за якого LR є безпечним. При D → 0 λ* → ∞ (LR завжди безпечне). При D >> 1/((k+1)λ) ймовірність втрати при LR прямує до 1.

Висновки. Ми запропонували єдину марківську модель для порівняння стратегій лінивого та нетерплячого відновлення у розподілених сховищах. Ключовий результат – замкнений аналітичний вираз для порогу λ, *що дозволяє проектувальникам системи обирати стратегію без симуляції: при* $\lambda < \lambda$ ліниве відновлення забезпечує еквівалентну надійність при значно менших витратах пропускної здатності; при λ > λ* нетерпляче відновлення є необхідним.

Обмеження. Модель передбачає незалежні відмови вузлів, пуасонівський процес відмов та експоненційно розподілений час відновлення. Корельовані відмови (збій стійки, відмова дата-центру) виходять за межі цієї роботи. **Майбутня робота:** валідація моделі на реальних трасах відмов (Filecoin, Storj) [4] та розробка адаптивної стратегії, що динамічно перемикається між LR та ER на основі поточної оцінки λ.

Література

1. Li C. et al. SoK : Decentralized Storage Network. *High-Confidence Computing*. 2024. Vol. 4, No. 3. Art. 100239. DOI: 10.1016/j.hcc.2024.100239.

2. Danezis G. et al. Walrus : An Efficient Decentralized Storage Network. *arXiv*. 2025. arXiv:2505.05370. URL: https://arxiv.org/abs/2505.05370 (дата звернення: 07.07.2026).

3. Greenan K. M., Plank J. S., Wylie J. J. Mean Time to Meaningless : MTTDL, Markov Models, and Storage System Reliability. *2nd USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage '10)* : proceedings, Boston, 2010. P. 1–5. URL: https://www.usenix.org/legacy/event/hotstorage10/tech/full_papers/Greenan.pdf (дата звернення: 13.07.2026).

4. Wu Z. et al. Degrees of Decentralized Freedom: Comparing Modern Decentralized Storage Platforms. *Network Traffic Measurement and Analysis Conference (TMA 2025)* : proceedings, IFIP, Copenhagen, 2025. P. 1–11.