

МЕТОД НАДІЙНОГО ЗБЕРІГАННЯ ДАНИХ НА ОСНОВІ НАДЛИШКОВОЇ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

В роботі запропоновано метод надійного зберігання даних на основі коригуючих кодів системи залишкових класів. Важливою характеристикою систем зберігання даних є висока швидкість запису та зчитування інформації з накопичувачів. Враховуючи, що система зберігання базується на коригуючих кодах, необхідно отримати підвищену швидкість процесів кодування та декодування даних. Перевагою запропонованого рішення є менша надлишковість для відновлення даних, порівняно з технологією RAID та інших систем, що базуються на коригуючих кодах.

Ключові слова: система залишкових класів, коригуючі коди.

V.V. YATSKIV, S.V. KULYNA

Тернопільський національний економічний університет

THE METHOD OF RELIABLE DATA STORAGE BASED ON REDUNDANT RESIDUE NUMBER SYSTEM

The main advantages and disadvantages of existing real-time data storage methods used to improve the reliability of storage systems are discussed. The main purpose of the study is the storage system reliability improvement by using coding methods based on redundant residue number system. The proposed method of secure data storage is based on the corrective codes of the residue number system. The essence of this method is that the array of obtained data is divided into fragments which are used to calculate the residuals by the selected system of relatively prime modules. The resulting residuals are stored on separate disks, which can be placed on one device or connected remotely. Considering that during encoding an extended module system is used, in case of disk distortion or failure, the redundant residue number system allows to recover the data fragment using only one validation module, if it is known which data block has been lost. The important characteristic of storage systems is the high speed of data writing / reading. Due to the fact that the storage system will use corrective codes, it is necessary to ensure a high speed of data encoding / decoding processes. The advantage of the proposed solution is that there is less redundancy in data recovery compared to RAID technology and systems that use other correction codes. The authors conducted experimental studies on the dependence of the required time for data packet processing on the selected module systems. According to the obtained results, the transition to a system with more modules allows to increase the speed by 13-15% compared to the previous one. Also the bit rate increasing allows to increase the encoding and decoding speed by 1.81-2.11 times. The developed system provides high reliability of data storage through the possibility of data recovery in case of some disks failure, and also high data security due to the distributed storage.

Keywords: residue number system, correction codes, storage, reliability.

Вступ

Останнім часом все більшого попиту на світовому ринку досягають системи надійного зберігання даних. В даний час вони використовуються як під час побудови баз даних, так і для сховищ даних, для корпоративних рішень окремих компаній і для організації мережних та хмарних сховищ даних. З цього зрозуміло, що розробка та впровадження методів підвищення ефективності вказаного класу систем дозволить значно розширити їх область застосування і відповідно збільшить ринок таких систем.

Згідно з світовою практикою більшість організацій використовують методи розподіленого (віддаленого) зберігання та резервного копіювання, а також методи зберігання даних в режимі реального часу для ліквідації наслідків втрати інформації.

Для зберігання даних в режимі реального часу використовують наступні типи систем [1]:

- 1) пряме сховище (direct-attached storage, DAS);
- 2) мережний пристрій зберігання даних (network attached storage, NAS);
- 3) мережа зберігання даних (storage area network, SAN);
- 4) хмарне сховище (Cloud storage);
- 5) резервний масив незалежних дисків (Redundant array of independent disks, RAID).

DAS (Direct-attached storage – система зберігання даних із прямим підключенням) – це сховище, яке складається тільки з накопичувача даних (жорсткий диск або USB-накопичувач), який безпосередньо під'єднаний до комп'ютера чи робочої станції через контролер шини даних. Іншими словами, DAS не є частиною мережі зберігання даних.

Основними перевагами DAS є простота та низька вартість встановлення. Використання мережних систем зберігання вимагає більшого планування, а також придбання та розгортання мережевого обладнання, такого як маршрутизатори та комутатори, на додаток до відповідних кабелів і з'єднань.

NAS (Network attached storage – мережева система зберігання даних) – це пристрій, підключений до мережі, який дозволяє авторизованим користувачам зберігати та отримувати дані з централізованого сховища.

Перевага NAS полягає в тому, що він спрощує обмін файлами між декількома користувачами, потенційно забезпечуючи більш високу продуктивність, ніж традиційний файловий сервер.

SAN (Storage area network – мережа зберігання даних) – це високошвидкісна мережа пристроїв зберігання даних, які також з'єднують ці пристрої зберігання даних з серверами. Вона надає систему

блочного копіювання даних у сховище, до якої можуть звертатися програми, запущені на будь-яких мережесерверах. Завдяки високошвидкісним з'єднанням, SAN часто забезпечує кращу продуктивність, ніж DAS. Крім того, оскільки SAN, як правило, пропонує кілька підключень до серверів центру обробки даних, що також підвищує доступність даних. Крім того, відокремлення сховища від серверів звільняє обчислювальні ресурси на серверах для інших завдань, не пов'язаних зі зберіганням інформації.

Хмарне сховище (Cloud storage) – це вид віддаленого сховища, під час застосування якого використовується простір на накопичувачах в дата-центрі (Google Drive, iCloud і Dropbox та ін.).

Існує безліч переваг використання хмарних сховищ, найбільш значимою є доступність даних. Доступ до інформації, що зберігаються в хмарі, можна отримати в будь-який час з будь-якого місця, якщо у вас є доступ до Інтернету. Іншою перевагою є те, що хмарне зберігання надає організаціям можливість здійснювати віддалене резервне копіювання даних, що знижує витрати, пов'язані з аварійним відновленням.

На жаль, найбільшим недоліком для хмарних сховищ є те, що користувачі обмежені пропускнуою здатністю. Якщо підключення до Інтернету повільне чи нестабільне, можливо, виникнуть проблеми з доступом до файлів або їх спільного використання.

RAID (англ. Redundant Array of Independent Disks) – технологія віртуального сховища даних, яка об'єднує декілька дисків у логічний том з надлишковістю даних, з метою забезпечення відмовостійкості або поліпшити загальну продуктивність накопичувачів.

RAID дозволяє зберігати однакові дані надлишково (в декількох кроках) збалансованим способом, щоб поліпшити загальну продуктивність.

Диски RAID часто використовуються на серверах, але, як правило, не потрібні для персональних комп'ютерів. Набором пристроїв, з яких складається система зберігання, керує особливий контролер масиву (RAID-контролер). Він забезпечує зв'язок між дисками, розміщення даних та дозволяє відобразити увесь масив, як логічний пристрій зберігання. За рахунок виконання операцій читання та запису на декількох дисках одночасно, масив забезпечує вищу швидкість обміну в порівнянні з одним великим диском.

Масиви дисків також забезпечують надійніше зберігання інформації завдяки чому дані не будуть втрачені у випадку виходу з ладу одного із них. Залежно від обраного рівня RAID, інформація віддзеркалюється або рівномірно розподіляється на дисках системи.

Протягом останніх років у сфері безпечного зберігання інформації значну нішу займають саме RAID технології. Загальна структура RAID системи зображена на рис. 1.

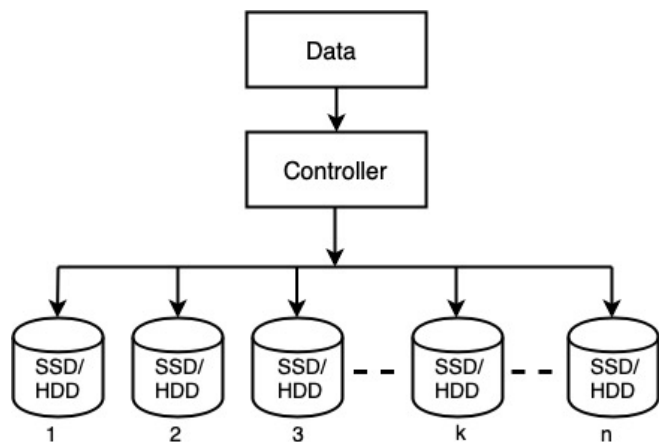


Рис. 1. Структура системи зберігання даних

Аналіз існуючих робіт

У [2–5] пропонуються нові механізми побудови RAID, що працюють як звичайна система введення-виведення і оптимізація навантаження на більший дисковий простір. Запропоновані нові архітектури зберігання даних в RAID масивах. Розроблені методики оцінки початкових параметрів надійності, які використовуються в моделі надійності та приклади розрахунку середнього часу втрати даних.

Стаття [6] присвячена питанням побудови систем зберігання даних на твердосплавних накопичувачах (SSD), які широко використовувались на персональних комп'ютерах та в центрах обробки даних. Зокрема, розглядалася проблема впливу різних розподілів паритетів на кілька пристроїв, що впливає на надійність масиву SSD RAID. Слід відмітити, що автори не приділяють належну увагу виявленню та виправленню помилок, які виникають внаслідок обмеженого ресурсу використання твердосплавних накопичувачів.

Послуги зберігання даних дають нові можливості для технології розподіленого збереження інформації, а саме хмарні технології. В роботі [7] представлено систему, яка дозволяє використовувати одночасно декілька різних провайдерів дискового простору та забезпечити доступність і шифрування. Проте в роботі не досліджено можливості втрати зв'язку з одним з провайдерів, що унеможливить доступ до інформації. Використання коригуючих кодів на основі модулярної арифметики для цієї системи надало б можливість мати постійний доступ до даних, незалежно від втрати одного або декількох з них та забезпечило б шифрування інформації.

В [8] розроблені ефективні коригуючі коди СЗК, здатні виявляти та виправляти пакети помилок. Оскільки виявлення та виправлення помилок відбувається з використанням алгоритмів зворотного перетворення, тобто потребує відновлення позиційного подання повідомлення, то це значно ускладнює процес декодування при зростанні кількості символів у повідомленні та розрядності символів.

Частково виправити вказані недоліки дозволяють коригуючі коди в системі залишкових класів, однак вони також характеризуються високою обчислювальною складністю алгоритмів виправлення помилок, а також необхідно перетворювати дані сенсорів у систему залишкових класів [9]. Враховуючи викладене, можна окреслити наступні проблеми при розробці методів для підвищення ефективності систем

зберігання даних: вибір коригуючих кодів на основі досконалої та модифікованої форм системи залишкових класів з мінімальною обчислювальною складністю алгоритмів декодування та можливістю адаптивної зміни коригуючої здатності коду без зміни алгоритму кодування; розробка алгоритмів зберігання розподілених даних поданих в системі залишкових класів.

Надлишкова система залишкових класів

Ідея авторів, полягає у тому, що підвищення надійності роботи системи зберігання даних можна досягти шляхом використання надлишкової системи залишкових класів (НСЗК), яка забезпечує відновлення даних при втраті одного із залишків по певному модулю (модулях).

Система залишкових чисел (СЗК) визначається набором k цілих чисел (модулів) $p_i (i = 1, 2, \dots, k)$.

Якщо всі модулі є попарно взаємно простими числами, то будь-яке ціле число X в діапазоні $[0, P_k)$, де $P_k = \prod_{i=1}^k p_i$ може бути однозначно представлено послідовністю залишків $x_i: X = (x_1, x_2, \dots, x_k)$, де число X у діапазоні $[0, P_k)$ може бути відновлено з k залишків (x_1, x_2, \dots, x_k) , використовуючи китайську теорему про залишки [10].

Відновлення даних здійснюється за формулою $X_j = \left(\sum_{i=1}^k x_i \cdot M_i \right) \text{mod } P_k$, де k – кількість інформаційних модулів; $M_i = \delta_i \cdot m_i$, $\delta_i = \frac{P}{p_i}$, $m_i = \delta_i^{-1} (\text{mod } p_i)$, $P_k = \prod_{i=1}^k p_i$.

Надлишкову СЗК отримуємо додаванням $r = n - k$ додаткових модулів $(p_{k+1}, p_{k+2}, \dots, p_n)$, до раніше вибраної системи модулів. В результаті формується НСЗК-код з n додатних попарно взаємно простих модулів.

Тепер ціле число X в діапазоні $[0, P_k)$ подається послідовністю n залишків за модулями p_1, p_2, \dots, p_n :

$$X = (x_1, x_2, \dots, x_k, \dots, x_n)$$

Відповідно, інтервал $[0, P_k)$ називається інформаційним (робочим), а інтервал $[P_k, P_n)$, сформований із додаткових модулів r , називається перевірочним діапазоном.

Пропонований метод

Суть методу полягає в наступному: масив отриманих даних, розділяється на фрагменти (блоки) X_j , з яких обчислюються залишки x_i за системою взаємно простих модулів $p_i: x_i = X_j (\text{mod } p_i)$.

Отримані залишки розподілені на окремі носії (диски). Враховуючи, що при кодуванні використовується розширена система модулів, для відновлення даних нам необхідна тільки частина залишків (рис. 2).

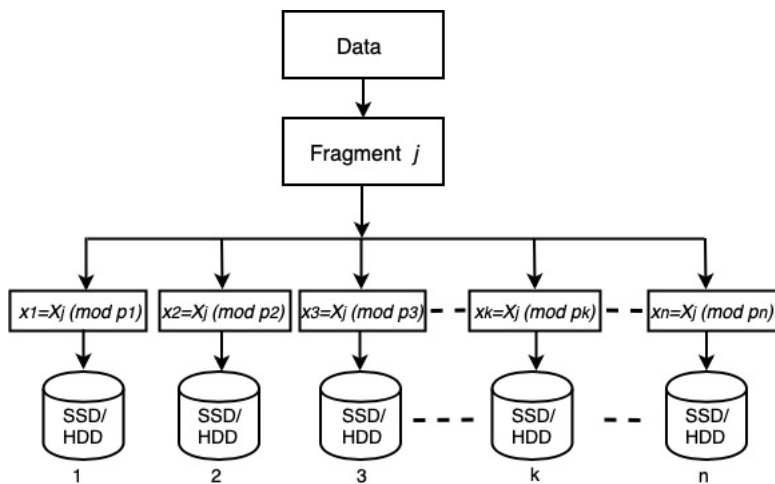


Рис. 2. Система надійного зберігання даних на основі НСЗК

Відновлення спотворених або втрачених даних здійснюється за формулою:

$$X_j = \left(\sum_{i=1}^n x_i \cdot M_i \right) \text{mod } P_n, \tag{1}$$

де n – кількість модулів; $P_n = \prod_{i=1}^n p_i$.

Коригуюча здатність надлишкової СЗК визначається кількістю введених додаткових модулів. При додаванні r додаткових модулів, надлишкова СЗК здатна виявляти r виправляти помилки в $r/2$ залишках [11].

Основною перевагою НСЗК є рівноцінність (взаємозамінність) інформаційних та перевірочних символів, що дозволяє відновлювати фрагменти даних при спотворенні чи виході з ладу цілого блоку залишків за одним з модулів.

Надлишкова системи залишкових класів дозволяє відновити втрачений фрагмент даних з використанням тільки одного перевірочного модулю, якщо відомо, який саме з блоків даних був втрачений. У системах зберігання даних, на відміну від передачі даних, визначити неробочий носій можна, застосовуючи інші методи та засоби діагностики.

Приклад. Розглянемо приклад використання системи залишків. Виберемо систему модулів: $p_i = [227, 229, 233, 239, 241, 251]$. Загальний діапазон системи модулів становитиме: $P_n = 175\ 107\ 974\ 924\ 611$.

Робочий діапазон: $P_k = 697\ 641\ 334\ 361$. Базисні числа $M_i = [132680932541996, 16822600211098, 174356438551544, 114296418779244, 51587826637541, 35579708052411]$.

Нехай блок даних у десятковій системі числення становить $X_j = 12350$, то подання цього числа в системі залишків матиме вигляд: $X_j = [92, 213, 1, 161, 59, 51]$. Згідно з формулою (1)

$$\begin{aligned} X_j &= (X_0 * M_0 + X_1 * M_1 + X_2 * M_2 + X_3 * M_3 + X_4 * M_4 + X_5 * M_5) \% P_n = \\ &= (92 * 132680932541996 + 213 * 16822600211098 + 1 * \\ &\quad * 174356438551544 + 161 * 114296418779244 + 59 * 51587826637541 + \\ &\quad + 51 * 35579708052411) \% 175107974924611 = 12350 \end{aligned}$$

При спотворенні одного і з залишків, за умови, що ми знаємо який із залишків зазнав спотворення, ми можемо відновити втрачені дані. Припустимо, що втрачено $X_4 = 59$. Тоді формула відновлення даних буде мати вигляд:

$$\begin{aligned} X_j &= (X_0 * M_0 + X_1 * M_1 + X_2 * M_2 + X_3 * M_3 + X_5 * M_5) \% \left(\frac{P_n}{P_4}\right) = \\ &= (92 * 132680932541996 + 213 * 16822600211098 + \\ &\quad + 1 * 174356438551544 + 161 * 114296418779244 + 51 * 35579708052411) \% \left(\frac{175107974924611}{241}\right) = 12350. \end{aligned}$$

Проведений аналіз використання систем модулів для кодування та декодування інформаційних фрагментів кратних 8 біт показав, що використання таких модулів дозволяє мінімізувати надлишкову інформацію. На основі проведених досліджень можна сказати, що збільшення кількості модулів зменшує надлишковість інформації, яка зберігається на носіях даних. Для виявлення однієї помилки у кодах НСЗК достатньо одного модуля, проте для її виправлення необхідно два перевірочних модулі. Вибрані системи модулів та їх характеристики відображено в таблиці 1.

Таблиця 1

НСЗК з двома перевірочними модулями

Набір модулів НСЗК	Кількість модулів, $(n, k)^*$	Розмір фрагмента, біт	Загальний діапазон, біт	Надлишковість, %
[227, 229, 233, 239, 241, 251]	(6, 4)	24	48	100,
[223, 227, 229, 233, 239, 241, 251]	(7, 5)	32	56	75,0
[211, 223, 227, 229, 233, 239, 241, 251]	(8, 6)	40	64	60,0
[199, 211, 223, 227, 229, 233, 239, 241, 251]	(9, 7)	48	72	50,0
[197, 199, 211, 223, 227, 229, 233, 239, 241, 251]	(10, 8)	56	80	42,9
[193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251]	(11, 9)	64	88	37,5

де n – загальна кількість модулів, k – кількість інформаційних модулів.

Як видно з таблиці 1, використання кількості модулів меншої або рівної шість є неефективним, оскільки надлишок інформації становить 100%, а отже, відповідає технології RAID рівня 1. При збільшенні кількості модулів надлишковість інформації зменшується завдяки зростанню фрагмента інформації, який опрацьовується за один цикл роботи.

Таблиця 2

НСЗК з одним перевірочним модулем

Набір модулів НСЗК	Кількість модулів, $(n, k)^*$	Розмір фрагмента, біт	Загальний діапазон, біт	Надлишковість, %
[227, 229, 233, 239, 241, 251]	(6, 5)	32	48	50,0
[223, 227, 229, 233, 239, 241, 251]	(7, 6)	40	56	40,0
[211, 223, 227, 229, 233, 239, 241, 251]	(8, 7)	48	64	33,3
[199, 211, 223, 227, 229, 233, 239, 241, 251]	(9, 8)	56	72	28,6
[197, 199, 211, 223, 227, 229, 233, 239, 241, 251]	(10, 9)	64	80	25,0
[193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251]	(11, 10)	72	88	22,2

де n – загальна кількість модулів, k – кількість інформаційних.

Проте таке використання перевірочних модулів є не доцільним при роботі з масивами дисків. При втраті зв'язку чи виході з ладу одного із дисків нам відомо, який саме з них не працює. Це дає змогу зменшити необхідну кількість перевірочних модулів для виправлення фрагмента даних. В таблиці 2 відображено систему наборів модулів використану в попередньому прикладі, проте в ній тільки один модуль виконує роль перевірочного.

Завдяки тому, що кількість перевірочних модулів зменшилась, ми можемо опрацювати за один цикл роботи на 8 біт більше. Завдяки цьому надлишок інформації зменшився у 2 рази при використанні системи з 6 модулів.

Одним з значних недоліків використання системи модулів з більшою кількістю k , є збільшення необхідної кількості носіїв для зберігання даних, проте завдяки цьому зменшується надлишковість та час опрацювання даних. Час, необхідний для опрацювання пакету даних розміром 50 Мб залежно від вибраної системи рівнянь, наданий на рис. 3. Дослідження проводились на комп'ютері з двоядерним процесором Intel Core i5-2430M (2.4 ГГц) та обсягом пам'яті 6 ГБ.

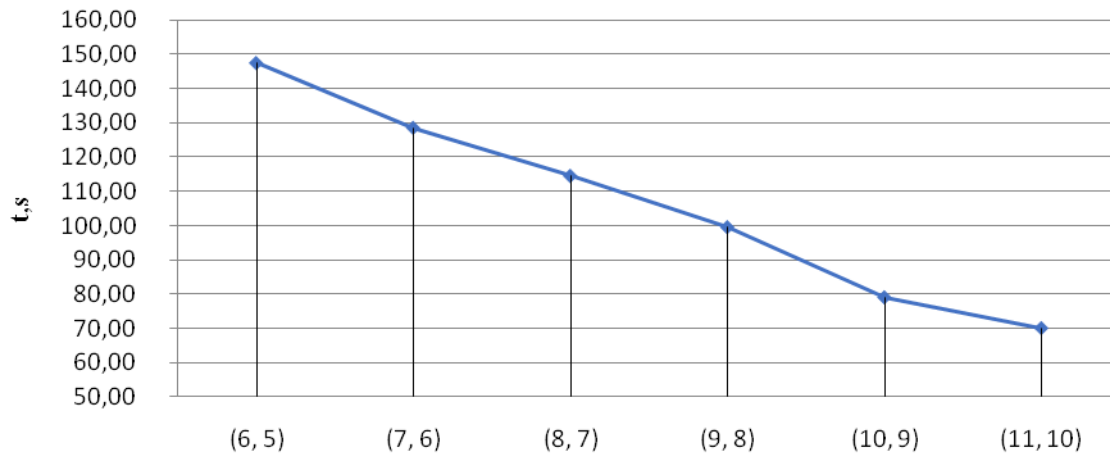


Рис. 3. Залежність часу від обраної системи модулів при опрацювання фрагменту розміром 50 Мб

Як видно з рисунку 3 перехід до використання системи з більшою кількістю модулів дозволяє збільшити швидкодію на 13–15% в порівнянні з попередньою. Проте варто враховувати, що збільшення кількості модулів потребує потужнішої апаратної бази для кодування та декодування даних.

Іншим шляхом підвищення швидкодії роботи з фрагментами даних є використання системи із значно більшими значеннями модулів, наприклад, $p_i = [10313, 10321, 10331, 10333]$. Загальний діапазон такої системи залишків становить 54 біти, а робочий 40. Завдяки більшим значенням модулів, цю систему по розміру робочого фрагмента можна порівняти з системою (7,6) із таблиці 2. Графік зростання часу, затраченого на кодування залежно від вибраного методу, наведено на рис. 4, а час, затрачений на декодування – на рис. 5.

Як видно з рисунків 4 та 5, процес кодування та декодування даних на основі НСЗК при обраних системах модулів є лінійним, а отже не залежить від розміру робочого фрагмента. Проте, завдяки збільшенню розрядності використаних модулів ми отримуємо зменшення часу обробки фрагмента при кодуванні та декодуванні у 1,8-2,1 рази.

Порівняння ефективності відомих RAID масивів та запропонованих з використанням системи залишкових класів наведено в таблиці 3.

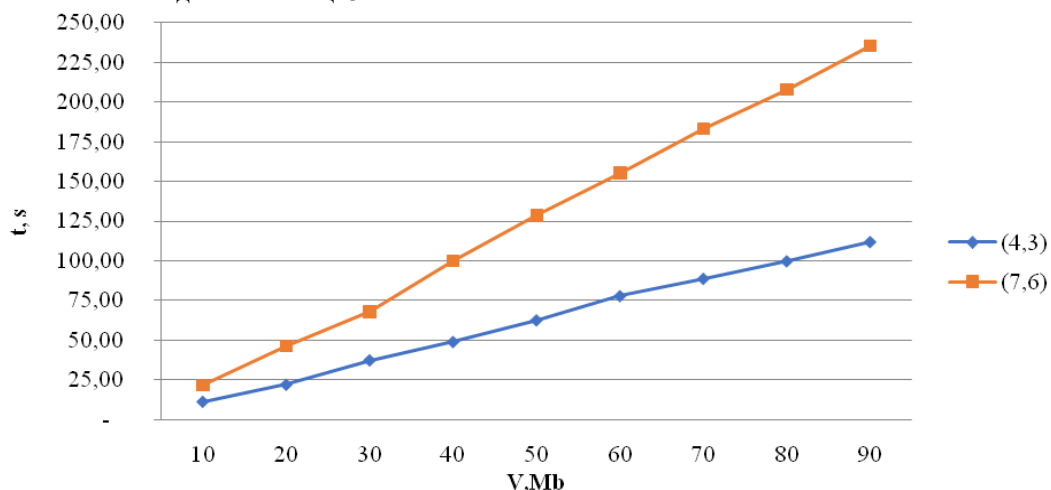


Рис. 4. Кодування даних заданими наборами модулів

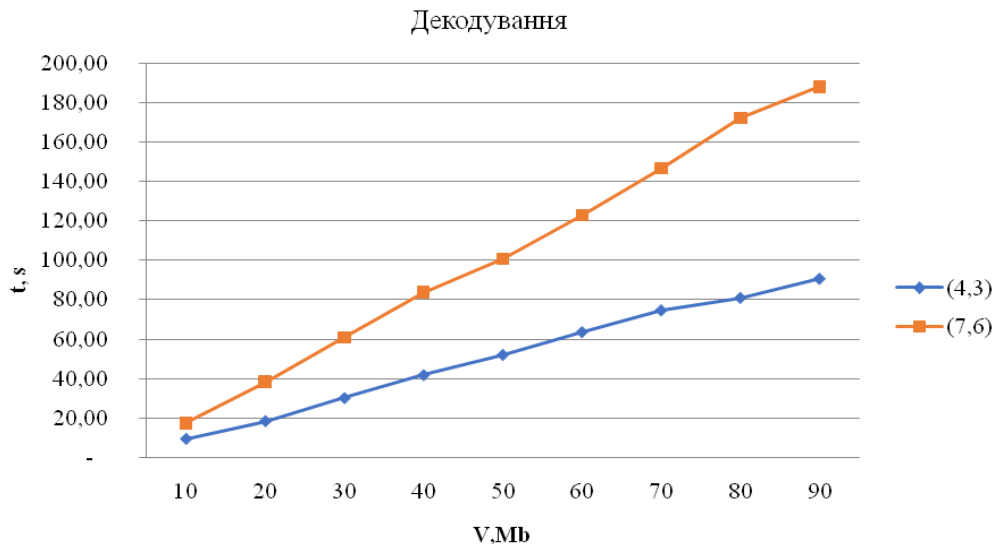


Рис. 5. Декодування даних заданими наборами модулів

Таблиця 3

Порівняння ефективності різних типів RAID масивів

Тип RAID масиву	Ефективний обсяг	Мінімальна кількість дисків	Допустима кількість дисків, що вийшли з ладу (Відмовостійкість)
1	2	3	4
RAID 0	$I = S \cdot N$	2	Немає
RAID 1	$I = S$	2	$(n - 1)$ несправний диск
RAID 2	$I = S \cdot (N - 2)$		Один несправний диск
RAID 3	$I = S \cdot (N - 1)$	3	Один несправний диск
RAID 4	$I = S \cdot (N - 1)$	3	Один несправний диск
RAID 5	$I = S \cdot (N - 1)$	3	Один несправний диск
RAID 6	$I = S \cdot (N - 2)$	4	Два несправних диски
RAID DP	$I = S \cdot (N - 2)$	6	Два несправних диски
RAID 10	$I = S \cdot N / 2$	4	від 1 до $N/2$ дисків (інформація не втратиться, якщо вийдуть з ладу диски в межах різних дзеркал)
RAID 50	$I = S \cdot (N - 2)$	6	від 1 до 2 дисків (інформація не втратиться, якщо вийде з ладу однакова кількість дисків в різних масивах)
RAID 60	$I = S \cdot (N - 4)$	8	від 2 до 4 дисків (інформація не втратиться, якщо вийде з ладу однакова кількість дисків в різних масивах)
RAID RRNS1	$I = S \cdot (N - 1)$	3	Один несправний диск
RAID RRNS2	$I = S \cdot (N - 2)$	4	Два несправних диска

Тут N – кількість дисків в масиві, S – об'єм найменшого диска.

Розроблена система забезпечує високу надійність під час зберігання даних завдяки можливості відновити фрагмент даних при виході з ладу одного з дисків, а також захищеність інформації при розподіленому зберіганні на віддалених носіях. Це відбувається завдяки тому, що для відновлення будь-якого фрагменту даних необхідно отримати фізичний або віртуальний доступ до більшості дисків.

Перевагою пропонованого методу є значно менша надлишковість для відновлення даних, у порівнянні з технологіями RAID та системами на основі інших коригуючих кодів.

Висновки

На основі проведених досліджень було запропоновано метод використання надлишкової системи залишкових класів для підвищення надійності роботи систем зберігання даних. Завдяки рівноцінності інформаційних та перевірочних символів ми можемо відновити втрачений блок даних з використанням тільки одного перевірочного модулю. У системах зберігання даних, на відміну від передачі даних, визначити неробочий носій можна, застосовуючи інші методи та засоби діагностики.

Проведено експериментальні дослідження залежності необхідного часу опрацювання пакету даних від обраної системи модулів. Отже, перехід до використання системи з більшою кількістю модулів дозволяє

збільшити швидкодію на 13–15% в порівнянні з попередньою, відповідно збільшення розрядності дозволяє підвищити швидкість кодування та декодування у 1,81–2,11 рази.

Розроблена система забезпечує високу надійність зберігання даних за рахунок можливості відновлення даних при виході з ладу частини дисків, а також високу захищеність даних при їх розподіленому зберіганні.

Література

1. Garth A. Network attached storage architecture / Garth A. Gibson and Rodney Van Meter // *Communications of the ACM*. – New York, 2000. – Vol. 43 (№ 11). – P. 37–45.
2. Guangyan Zhang. RAID+: Deterministic and Balanced Data Distribution for Large Disk Enclosures / Guangyan Zhang, Zican Huang, Xiaosong Ma, Songlin Yang, Zhufan Wang, Weimin Zheng // This paper is included in the Proceedings of the 16th USENIX Conference on File and Storage Technologies. – Oakland, February 12–15, 2018. – P. 280–283.
3. Wan J. Thin RAID: Thinning Down RAID Array for Energy Conservation / J. Wan, X. Qu, N. Zhao, J. Wang and C. Xie // *IEEE Transactions on Parallel & Distributed Systems*. – 2015. – Vol. 26 (№ 10). – P. 2903–2915.
4. Rahman P. A. Reliability model of disk arrays RAID-5 with data striping / P. A. Rahman and G. D'K Novikova, Freyre Shavier // *IOP Conf. Series: Materials Science and Engineering*. – 2018. – Vol. 327. – P. 788–794.
5. Keval Kachhala. RAID (Redundant Array of independent Disks) / Keval Kachhala, Rupali Gangarde // *International Journal of Engineering Trends and Technology (IJETT)*. – May 2016. – Vol. 35 (№ 12). – P. 574–577.
6. Yongkun Li. Analysis of Reliability Dynamics of SSD RAID / Yongkun Li, P. C. Lee, John C. S. Lui // *IEEE TRANSACTIONS ON COMPUTERS*. – 2014. – P. 1–14.
7. Antonio Celesti. Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems / Antonio Celesti, Maria Fazio, Massimo Villari, Antonio Puliafito // *Journal of Network and Computer Applications*. – January 2016. – Vol. 59. – P. 208–218.
8. Tay Thian Fatt. A new algorithm for single residue digit error correction in Redundant Residue Number System / Tay Thian Fatt // *Circuits and Systems (ISCAS)*, IEEE International Symposium IEEE. – 2014. – P. 1748–1751.
9. Alsheikh M. A. Rate-distortion Balanced Data Compression for Wireless Sensor Networks / Alsheikh M. A., Lin S., Niyato D., & Tan H. P. // *IEEE Sensors Journal*. – 2016. – Vol. 16 (№12). – P. 5072–5083.
10. Xiao H. New Error Control Algorithms for Residue Number System Codes / Xiao H., Garg H. K., Hu J., & Xiao G // *ETRI Journal*. – 2016. – Vol. 38 (№ 2). – P. 326–336.
11. Акушский И. Я. Машинная арифметика в остаточных классах / Акушский И. Я., Юдицкий Д.И. – М. : Сов. радио. 1968. – С. 460.
12. Кулина С.В. Виявлення помилок на основі коригуючих кодів системи залишкових класів / С.В. Кулина // *Фізико-технологічні проблеми передавання, оброблення та зберігання інформації в інфокомунікаційних системах : матеріали VII Міжнародної науково-практичної конференції. – Чернівці : «Місто», 2018. – С. 126–127.*

References

1. Garth A. Network attached storage architecture / Garth A. Gibson and Rodney Van Meter // *Communications of the ACM*. – New York, 2000. – Vol. 43 (№ 11). – P. 37–45.
2. Guangyan Zhang. RAID+: Deterministic and Balanced Data Distribution for Large Disk Enclosures / Guangyan Zhang, Zican Huang, Xiaosong Ma, Songlin Yang, Zhufan Wang, Weimin Zheng // This paper is included in the Proceedings of the 16th USENIX Conference on File and Storage Technologies. – Oakland, February 12–15, 2018. – P. 280–283.
3. Wan J. Thin RAID: Thinning Down RAID Array for Energy Conservation / J. Wan, X. Qu, N. Zhao, J. Wang and C. Xie // *IEEE Transactions on Parallel & Distributed Systems*. – 2015. – Vol. 26 (№ 10). – P. 2903–2915.
4. Rahman P. A. Reliability model of disk arrays RAID-5 with data striping / P. A. Rahman and G. D'K Novikova, Freyre Shavier // *IOP Conf. Series: Materials Science and Engineering*. – 2018. – Vol. 327. – P. 788–794.
5. Keval Kachhala. RAID (Redundant Array of independent Disks) / Keval Kachhala, Rupali Gangarde // *International Journal of Engineering Trends and Technology (IJETT)*. – May 2016. – Vol. 35 (№ 12). – P. 574–577.
6. Yongkun Li. Analysis of Reliability Dynamics of SSD RAID / Yongkun Li, P. C. Lee, John C. S. Lui // *IEEE TRANSACTIONS ON COMPUTERS*. – 2014. – P. 1–14.
7. Antonio Celesti. Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems / Antonio Celesti, Maria Fazio, Massimo Villari, Antonio Puliafito // *Journal of Network and Computer Applications*. – January 2016. – Vol. 59. – P. 208–218.
8. Tay Thian Fatt. A new algorithm for single residue digit error correction in Redundant Residue Number System / Tay Thian Fatt // *Circuits and Systems (ISCAS)*, IEEE International Symposium IEEE. – 2014. – P. 1748–1751.
9. Alsheikh M. A. Rate-distortion Balanced Data Compression for Wireless Sensor Networks / Alsheikh M. A., Lin S., Niyato D., & Tan H. P. // *IEEE Sensors Journal*. – 2016. – Vol. 16 (№12). – P. 5072–5083.
10. Xiao H. New Error Control Algorithms for Residue Number System Codes / Xiao H., Garg H. K., Hu J., & Xiao G // *ETRI Journal*. – 2016. – Vol. 38 (№ 2). – P. 326–336.
11. Akushskij I. Ya. Mashinnaya arifmetika v ostatochnyh klassah / Akushskij I. Ya., Yudickij D.I. – M. : Sov. radio. 1968. – S. 460.
12. Kulyna S.V. Vyiavlennia pomylok na osnovi koryhuiuchykh kodiv systemy zalyshkovykh klasiv / S.V. Kulyna // *Fizyko-tekhnolohichni problemy prepravannia, obrobлення ta zberihannia informatsii v infokomunikatsiinykh systemakh : materialy VII Mizhnarodnoi naukovo-praktychnoi konferentsii. – Chernivtsi : «Misto», 2018. – S. 126–127.*

Рецензія/Peer review : 10.11.2019 р.

Надрукована/Printed : 02.01.2020

Рецензент: к.т.н., доц. Кочан В.В.