

Системы телекоммуникации, связи и защиты информации

УДК 004.6

М.Є. Боданюк, О.К. Карнаухов, О.І. Ролік, канд. техн. наук, **С.Ф. Теленик**, д-р техн. наук
Національний технічний університет України "Київський політехнічний інститут",
НТУУ "КПІ": Україна, 03056, м.Київ-56, проспект Перемоги, 37.

Управління системами збереження даних

Розглянуті аспекти управління багаторівневими системами збереження даних (СЗД). Розроблені математичні моделі для задач кластеризації та управління ресурсами СЗД. Розроблені генетичний та евристичний алгоритми для розв'язання задачі розподілення ресурсів СЗД за критерієм мінімізації витрат на збереження даних. Наведені результати експериментальних досліджень ефективності алгоритмів для СЗД різних об'єктів при широкому діапазоні потреб користувачів у ресурсах збереження. Запропоновані рекомендації щодо використання запропонованих алгоритмів для розв'язання задачі управління СЗД в інформаційно-телекомунікаційних системах. Бібл. 9, рис. 5, табл. 1.

Ключові слова: система збереження даних, управління, ресурси, математична модель, генетичний алгоритм, евристичний алгоритм.

Вступ

Сьогодні інформаційно-телекомунікаційні системи охопили усі сфери діяльності суспільного та ділового життя людини [5, 7]. Дома, в офісі, у відрядженні і навіть на відпочинку кожна людина є користувачем різноманітних інформаційно-комунікаційних послуг. Це призводить до стрімкого збільшення об'ємів даних користувачів, вимагає виокремлення у складі інформаційно-телекомунікаційні систем та центрів оброблення даних (ЦОД) систем збереження даних (СЗД), породжує проблеми їх створення, впровадження та експлуатації. У СЗД особлива увага приділяється високому рівню доступності до даних і надійності їх збереження. В умовах високої конкуренції провайдери інформаційно-комунікаційних послуг намагаються забезпечити високий рівень якості сервісів при порівняно невеликих витратах. Потреба зменшувати витрати на створення і підтримку своєї ІТ-інфраструктури спонукає провайдерів інформаційно-комунікаційних послуг дотримуватися чіткого балансу між виділеними ресурсами

СЗД та цінністю даних, що зберігаються. Попит на СЗД зростає, оскільки вони є важливою складовою корпоративних і хмарних ІТ-інфраструктур. У свою чергу, зростання попиту на СЗД в сьогоденних умовах формує стійку потребу в ефективних системах управління СЗД.

Загальна модель поведінки провайдерів інформаційно-комунікаційних послуг щодо СЗД будується на застосуванні середніх СЗД і забезпеченні максимального використання їхніх потужностей. Це призводить до реалізації багаторівневої концепції збереження даних, згідно якої кожному типу даних виділяються ресурси СЗД із відповідними характеристиками швидкодії, доступності, надійності та безвідмовності. Багаторівнева концепція дозволяє побудувати СЗД без втрати продуктивності, але за більш низькою ціною. Але ефективне втілення цієї концепції вимагає розроблення відповідних математичних моделей і алгоритмів управління СЗД, на їх основі технологій управління СЗД та їх експериментального дослідження.

Суть і проблематика управління СЗД

У загальному випадку СЗД може містити велику кількість рівнів збереження, які розрізняються за:

- ступенем захищеності даних — рівні будуються на основі складної системи захисту на базі показників цільової точки відновлення даних (RPO) та часу, необхідного на відновлення резервної копії даних (RTO), дзеркальованих томів, захисту на рівні RAID, захисту засобами HDD;
- рівнем продуктивності — рівні розрізняються високою, середньою чи низькою продуктивністю для випадкового, потокового, змішаного доступу та іншими параметрами;
- рівнем захисту даних від змін — рівні можуть допускати зміну даних, не допускати зміну даних, представляти змішаний тип двох перших;

- способом доступу до даних — рівні можуть відрізнятися за способом доступу до даних (файловий, блоковий) і типом підключення: пряме підключення (DAS), мережеве підключення (NAS), мережеві блокові пристрої (SAN);
- терміном зберігання — 1 день, 1 місяць, 1 рік, 10 років, більше 10 років;
- типом системи зберігання та фізичним типом доступу — флеш, HDD, накопичувальна магнітна стрічка, MO, CD, DVD і прямий, послідовний, змішаний.

Сучасний ринок інформаційних технологій вимагає максимально недорогих СЗД із забезпеченням високої надійності та масштабованості. На зміну нарощуванню об'ємів збереження приходять такі нові підходи до збереження даних, як віртуалізація СЗД, багаторівневе збереження даних. Основними вимогами до СЗД є: надійність — СЗД повинні забезпечувати цілодобовий режим роботи і готовність надати по запиту збережені дані з вірогідністю 0,999(999); горизонтальна масштабованість — екстенсивне нарощування наявних ресурсів СЗД без зміни архітектури [4]; продуктивність — до ресурсів СЗД можуть мати доступ декілька серверів, які підтримують різні додатки (необхідно забезпечити можливість ефективного використання ресурсів СЗД декількома користувачами одночасно); повнофункціональність — широкий набір можливостей, які надає СЗД.

Згідно [1], при створенні сучасних СЗД виникають такі основні проблеми: збільшення кількості критичних бізнес-додатків; швидке зростання об'ємів інформації; низька гнучкість архітектури СЗД; проблеми безпеки; втрата або невірне розміщення інформації; простой серверів; невідповідність вимогам щодо зберігання даних та ін.

На ефективне розв'язання перелічених проблем націлені базові підходи до збереження даних: багаторівневе збереження — баланс між ресурсами СЗД і цінністю даних; віртуалізація сховищ — процес повного абстрагування логічного сховища даних від фізичного сховища даних (рис. 1).

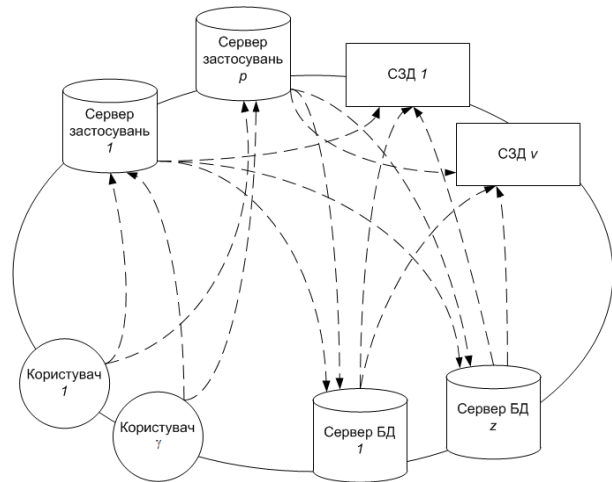


Рис. 1. Мережева взаємодія компонентів надання даних користувачам

Для забезпечення реальної ефективності цих підходів система управління функціонуванням ІТ-інфраструктури має включати підсистему управління СЗД [2], здатну вирішити наведені вище проблеми. За мету функціонування підсистеми управління СЗД доцільно прийняти забезпечення вимог користувачів економічно обґрунтованим способом. Критерієм вибору підсистеми управління СЗД доцільно прийняти мінімальне співвідношення «вартість збереження/інтегральний показник якості збереження». Безумовно при цьому виборі повинні виконуватися обмеження — часові, вартісні, технологічні, якісні та ін.

Функціонал підсистеми управління СЗД повинен охоплювати:

- зберігання даних у необхідних об'ємах з визначеним рівнем якості;
- розподілення користувачів між СЗД;
- забезпечення традиційних операцій введення, пошуку і модифікації даних з дотриманням вимог бізнес-процесів користувачів;
- розподілення даних за пристроями зберігання.

Особливе місце в умовах поширення розподілених і насамперед хмарних обчислень належить функціям розподілення даних за пристроями зберігання (рис. 2).

Дані повинні бути розподілені таким чином, щоб вимоги користувачів були задоволені і залишалася можливість провайдеру сервісів збереження отримати економічну вигоду. Останнє можливе за умови досягнення максимальної продуктивності СЗД при мінімальних витратах.

Для цього підсистема управління СЗД повинна базуватися на комплексі математичних моделей, що пов'язують критерії та обмеження функціонування ІТ-інфраструктури (через критерії функціонування СЗД) з параметрами СЗД.

Виділяють такі незалежні управляючі змінні СЗД:

- об'єм СЗД (Мб);
- пропускну здатність СЗД (Мб/с);
- продуктивність у кількості операцій вводу/виводу в секунду (IOPs);
- затримка (мс).

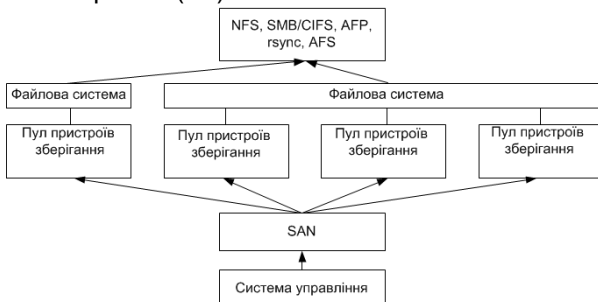


Рис. 2. Схема взаємодії систем управління і збереження даних

Критерії функціонування СЗД:

- мінімізація витрат на збереження файлів на рівнях пам'яті та пристроях зберігання СЗД;
- рівномірно заповнення файлами рівнів пам'яті та пристроїв в СЗД.
- Обмеження:
 - обмеження об'єму пристроїв зберігання;
 - обмеження часу доступу до файлів;
 - інтенсивність обміну між користувачами і СЗД.

Крім поділу сховища на рівні залишається проблема управління циклом життя даних. Критично важливі сьогодні дані вже через рік можуть значною мірою втратити свою актуальність. Відповідно знижуються вимоги до ресурсів, які виділяються для збереження цих даних.

Огляд існуючих рішень

Найчастіше на ринку ІТ пропонуються готові рішення СЗД — як універсальні, так і концептуальні, орієнтовані на задачі певного типу (провідні виробники: EMC, NetApp, IBM, DELL, HP, Fujitsu, Hitachi Data Systems). На їх

основі можна будувати комплексні СЗД різного рівня із використанням DAS-, NAS- та SAN-технологій. Математичні моделі та методи, покладені в основу функціонування перелічених систем, невідомі. До того ж, завжди залишається потреба у підвищенні ефективності за рахунок використання методів оптимізації параметрів СЗД і режимів функціонування.

Ефективність СЗД суттєво залежить від моделей і методів, розв'язання наукових проблем, пов'язаних із розподілом даних у СЗД, управління даними, балансування, тощо. У [3] запропонована модель управління збереженням даних при використанні багаторівневого середовища, але не проведено порівняння ефективності різних алгоритмів розміщення даних. Крім того, важливі задачі розподілення даних користувачів за перспективними критеріями, наприклад, мінімізації витрат на збереження даних в СЗД, як і задачі розподілення користувачів між СЗД залишилися поза увагою дослідників.

Усі ці чинники сприяли формуванню передумов для розроблення нових математичних моделей і алгоритмів управління СЗД, створення на їх основі ІТ управління СЗД, їх експериментального дослідження з метою напрацювання рекомендацій для їх ефективного застосування.

Постановка проблеми

СЗД характеризуються набором технічних параметрів, а користувачі — набором вимог до СЗД. Рівні пам'яті характеризуються набором технічних параметрів, а файли — набором вимог до рівнів. Головною вимогою файлів до об'єму пам'яті є повне його збереження на одному рівні та логічному томі. Вимоги файлів до часу доступу залежать від типу даних, що зберігаються.

Необхідно розробити моделі та методи визначення такого розподілення користувачів між СЗД та розміщення даних на рівнях СЗД, щоб витрати на обслуговування користувачів та збереження їх даних були мінімальними, а також виконувалися обмеження на об'єм розміщуваних даних і на час доступу до даних.

Математичні моделі

Введемо необхідні для побудови математичної моделі позначення. Нехай $V = \{V_1, \dots, V_V\}$ — множина СЗД, кількість яких дорівнює v ; $Y = \{Y_1, \dots, Y_\gamma\}$ — множина користувачів СЗД, кількість яких дорівнює γ ; $N = \{N_1, \dots, N_n\}$ — множина файлів, кількість яких

дорівнює n ; $K = \{K_1, \dots, K_m\}$ — множина рівнів (видів) пам'яті, які визначаються типом пам'яті (наприклад, флеш, HDD, магнітна стрічка, CD, DVD), кількість яких дорівнює m .

Звичайно, у сучасних обчислювальних центрах використовується три рівні, розділені за швидкістю доступу до даних.

Кожний файл користувача N_j , $i = 1, \dots, n$, має такий набір параметрів:

ρ_i — об'єм файлу N_j ;

d_i — обмеження по часу доступу до файлу N_j .

Кожний рівень пам'яті K_j , $j = 1, \dots, m$, має такий набір параметрів:

h_j — час доступу до одиниці збереження на K_j -му рівні пам'яті;

$R_j = \{r_1^j, \dots, r_s^j\}$ — множина засобів зберігання даних на K_j -му рівні пам'яті.

Кожний засіб збереження даних r має такий набір параметрів:

a_{sj} — наявний об'єм пам'яті r_s^j -го пристрою зберігання;

γ_s^j — використаний об'єм пам'яті r_s^j -го пристрою зберігання;

b_s^j — вартість використання r_s^j -го пристрою зберігання.

Математичні моделі розбиті на дві групи: кластеризації ресурсів (розподілення користувачів між СЗД) та управління ресурсами СЗД.

Задача кластеризації ресурсів.

Модель 1.1. Задача мінімізації сумарних витрат на передачу даних між користувачами і СЗД.

$$\min \sum_{i=1}^{\gamma} \sum_{j=1}^{\nu} \Gamma_{ij} x_{ij}. \quad (1)$$

де

$x_{ij} = \begin{cases} 1, & \text{якщо користувач } i \text{ розподілений на СЗД } j, \\ 0, & \text{в супротивному випадку.} \end{cases}$

Γ_{ij} — вартість передачі даних користувача i на СЗД j .

Обмеження інтенсивності обміну

$$\sum_{i=1}^{\gamma} A_{ij} x_{ij} \leq \Pi_j, \forall j = 1, \dots, \nu \quad (2)$$

де Π_j — пропускна здатність j -ї СЗД, A_{ij} — потреби i -го користувача у пропускній здатності j -ї СЗД.

Задача (1)—(2) становить лінійну задачу булевого програмування.

Модель 1.2 Задача максимізації вигоди від обслуговування користувачів СЗД з урахуванням надійності.

$$\min \sum_{i=1}^{\gamma} \sum_{j=1}^{\nu} \sum_{k=1}^{\Omega} \Gamma_{ijk} x_{ijk}, \quad (3)$$

де

$x_{ijk} = \begin{cases} 1, & \text{якщо користувач } i \text{ розподілений на СЗД } j \\ & \text{при рівні надійності } k, \\ 0, & \text{в супротивному випадку,} \end{cases}$

Ω — максимальне значення рівня надійності.

При цьому необхідно дотримуватись обмеження (2).

Задача (3), (2) також становить лінійну задачу булевого програмування.

Задача управління ресурсами СЗД.

Модель 2.1. Задача розподілення даних за критерієм мінімізації витрат на зберігання файлів.

Потрібно розмістити дані на рівнях пам'яті СЗД (рис. 3) так, щоб досягти мінімальних витрат на збереження даних при таких обмеженнях:

- об'єм розміщуваних на рівні даних не перевищує наявного об'єму пам'яті рівня;
- час доступу до даних не перевищує час доступу до даних рівня;

Пристрій, на якому не розміщено жодного файлу можна вважати вимкненим, тобто вартість його використання дорівнює нулю.

Мінімізацію витрат можна забезпечити, якщо розподілити дані користувачів таким чином, щоб кількість пристроїв, виділених під ці потреби, була найменшою. Тоді не задіяні пристрої можна вимкнути і тимчасово вивести із активної експлуатації, що й дозволить зменшити експлуатаційні витрати.

Зазначимо, що на етапі проектування СЗД така модель може бути використана для обґрунтування проекту з найменшою вартістю володіння.

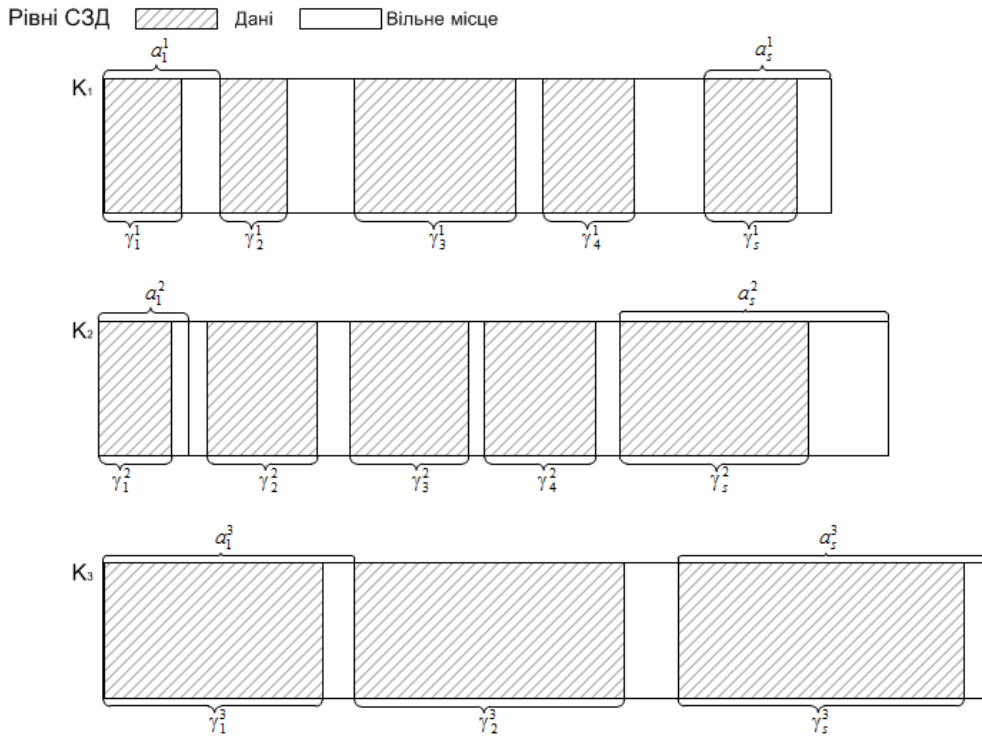


Рис. 3. Модель СЗД з трьома рівнями

Введемо матрицю X вигляду:

$$X = \begin{pmatrix} x_{11}^1 & x_{12}^1 & \dots & x_{1s}^1 & x_{11}^2 & x_{12}^2 & \dots & x_{1s}^2 & x_{11}^m & x_{12}^m & \dots & x_{1s}^m \\ x_{21}^1 & x_{22}^1 & \dots & x_{2s}^1 & x_{21}^2 & x_{22}^2 & \dots & x_{2s}^2 & x_{21}^m & x_{22}^m & \dots & x_{2s}^m \\ x_{31}^1 & x_{32}^1 & \dots & x_{3s}^1 & x_{31}^2 & x_{32}^2 & \dots & x_{3s}^2 & x_{31}^m & x_{32}^m & \dots & x_{3s}^m \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n1}^1 & x_{n2}^1 & \dots & x_{ns}^1 & x_{n1}^2 & x_{n2}^2 & \dots & x_{ns}^2 & x_{n1}^m & x_{n2}^m & \dots & x_{ns}^m \end{pmatrix}, \quad (4)$$

де $x_{is}^j = \begin{cases} 1, \text{ файл } N_i \text{ зберігається на пристрої } r_s \text{ рівня } j, \\ 0, \text{ файл } N_i \text{ не зберігається на пристрої } r_s \text{ рівня } j. \end{cases}$

Задачу розміщення файлів на різних видах пам'яті СЗД доцільно розглядати як задачу визначення матриці X , яка відображає фізичне розміщення файлів на рівнях пам'яті.

Введемо вектор

$$\bar{w} = (w_1^1, w_2^1, \dots, w_s^1), \quad (5)$$

де

$$w_s^j = \begin{cases} 1, \text{ якщо хоча б один файл розміщений на пристрої } r_s^j, \\ 0, \text{ якщо жодного файлу не розміщено на пристрої } r_s^j. \end{cases}$$

Тоді для задачі мінімізації витрат на збереження даних критерій запишеться у вигляді

$$\sum_{j=1}^m \sum_{k=1}^s w_k^j b_k^j \rightarrow \min, \quad (6)$$

обмеження по об'єму пам'яті для кожного пристрою набуде вигляду

$$\forall r_s, \gamma_s < a_s, \quad (7)$$

а обмеження часу доступу до даних —

$$\forall d_j, x_{is}^j \cdot d_j < h_j. \quad (8)$$

Задача (6)—(8) становить лінійну задачу булевого програмування.

Модель 2.2. Задача рівномірного розподілення файлів між пристроями зберігання.

У цьому випадку основна мета полягає в тому, щоб рівномірно розподілити дані користувачів між пристроями зберігання. Це дозволить, з одного боку, уникнути перевантаження, з другого боку, створити умови для

швидкого нарощування ресурсів збереження за потреби в умовах високої динаміки вимог користувачів.

Критерій для задачі рівномірного розподілу даних між пристроями можна записати у вигляді:

$$\sum_{j=1}^m \sum_{k=1}^s w_k^j \rightarrow \max \quad (9)$$

При цьому необхідно дотримуватись визначених вище обмежень (7)—(8).

Задача (9), (7)—(8) також становить лінійну задачу булевого програмування.

Методи розв'язання задач

Оскільки усі задачі є лінійними задачами булевого програмування, до них можна застосувати такі методи:

- обмеженого перебору;
- генетичні алгоритми;
- евристики.

Розглянемо задачу (6)—(8), модель 2.1. Оскільки в СЗД зазвичай можуть зберігатись десятки мільйонів файлів, використання алгоритмів повного перебору не доцільне внаслідок великої трудомісткості задачі. Тому генетичний та евристичний алгоритми видаються більш перспективними для розв'язання задачі.

Генетичний алгоритм. Розглянемо розроблений на основі праць [6,8] універсальний варіант такого алгоритму для наведеної задачі.

Крок 0. Установлюємо $i = 0$. Утворення випадкової початкової популяції $P_k(0) = \{p_1(0), \dots, p_k(0)\}$ з урахуванням можливо-го відсіювання під час перевірки обмежень.

Крок 1. Оцінюємо кожний ланцюжок популяції щодо виконання обмежень (4)—(5). Для тих ланцюжків $p_q(0)$, для яких обмеження виконуються, виконується оцінка $f(p_q(0))$ цільової функції, здійснюється вибір найкращого рішення.

Крок 2. Відбираємо представників для нової популяції (ланцюжків з найкращим значенням функції $f(p_q(0))$).

Крок 3. $l = l + 1$. Утворюємо нову популяцію застосуванням оператора рекомбінації.

Крок 4. Поліпшуємо популяцію застосуванням оператора мутації.

Крок 5. Оцінюємо кожний ланцюжок популяції щодо виконання обмежень і функцію $f(p_q(l))$, якщо обмеження виконуються.

Крок 6. Завершення після виконання заданої кількості ітерацій.

Вибирається найкраще рішення в утвореній популяції $P_k(l)$.

У наведеному варіанті генетичного алгоритму кожний ланцюжок (особина) будь-якої популяції становить двійковий вектор довжини $t = o \cdot n$, який визначається кількістю рівнів пам'яті та кількістю файлів, що розглядається на ітерації роботи алгоритму. Кількість файлів o вибирається експертним шляхом в залежності від наявних обчислювальних можливостей. Елемент q вектора набуває значення 0 або 1 відповідних змінних x_{is}^j . Довжина ланцюжків постійна.

Відбір особини для нової популяції виконується на основі ранжування за значенням цільової функції особин попередньої популяції, причому ранжуються лише особини, які задовольняють ресурсним та іншим обмеженням.

Для рекомбінування застосовується одноточковий кросинговер з випадковим вибором точки кросинговеру. Реалізація мутації здійснюється на основі схеми інверсії ділянок ланцюжків.

Евристичний алгоритм. Оскільки задача полягає в тому, щоб знайти розподіл даних по СЗД, при якому мінімізуються витрати на їх збереження, то буде логічним спершу розмістити файли, які мають вищі потреби в об'ємі пам'яті. Взнявши це за правило, необхідно лише врахувати особливості критерію. Кожному файлу N_i припишемо коефіцієнт P_i , значенням якого приймемо об'єм файлу N_i , кожному пристрою зберігання r_s^j — коефіцієнт ρ_j , значенням якого приймемо вартість використання b_s^j .

Попередній етап для будь-якого з наведених нижче варіантів евристичного алгоритму полягає у сортуванні файлів за значенням їх коефіцієнта P у порядку зростання і сортуванні пристроїв зберігання за значенням їх коефіцієнта ρ у порядку спадання.

Попередній етап виконано.

Крок 1. Пошук пристрою зберігання, який має найменше значення коефіцієнта ρ .

Крок 2. Пошук файлу з найбільшим значенням коефіцієнта P .

Крок 3. Якщо для вибраних файлу і рівня пам'яті не виконуються обмеження стосовно вимог щодо часу доступу рівня пам'яті K_j , то повертаємося на крок 2 для пошуку наступного файлу.

Крок 4. Якщо для вибраних файлу і рівня пам'яті не виконуються інші обмеження, то повертаємося на крок 2 для пошуку наступного файлу.

Крок 5. Вибраний файл розміщується на вибраному пристрої та вилучається із упорядкованого списку файлів.

Крок 6. Якщо упорядкований список файлів порожній, то кінець роботи алгоритму, інакше перехід до кроку 1.

Обчислювальна складність евристичного алгоритму становить $m \times n$.

Для інших критеріїв використовується інша логіка вибору пристроїв зберігання та файлів у їх впорядкованих списках.

Експериментальні дослідження

Розмір і кількість файлів, що зберігаються в СЗД, можуть суттєво змінюватись. З метою вивчення ефективності роботи евристичного та генетичного алгоритмів для різних випадків, експериментальні дослідження проводилися для СЗД, в яких зберігаються:

- 1) файли малого об'єму;
- 2) середні за об'ємом файли;
- 3) файли будь якого об'єму.

Моделювання проводилось для СЗД які мають три рівні, тобто $m = 3$. За основу взята модель 2.1 — задача мінімізації використаних пристроїв зберігання в СЗД. Параметри рівнів та файлів визначались випадковим чином, враховуючи дані, отримані в результаті моніторингу і аналізу діючих СЗД. Вартість використання b_s^j кожного з пристроїв зберігання варіюється в цілочисельному діапазоні значень від 1 до 5. Середні витрати на збереження даних визначались як середнє арифметичне вартості використання увімкнених пристроїв з усіх рівнів

$$B = \frac{w_1^1 b_1^1 + w_2^1 b_2^1 + \dots + w_s^1 b_s^1 + \dots + w_s^j b_s^j}{s_r}$$

Програмне забезпечення для здійснення досліджень спроектоване і реалізоване на мові програмування C#, платформа .NET.

На рис. 4 зображена відносна різниця між середніми витратами, за варіантами розподілів,

отриманих шляхом застосування евристичних та генетичних алгоритмів $\frac{B_{EA}}{B_{GA}} - 1$.

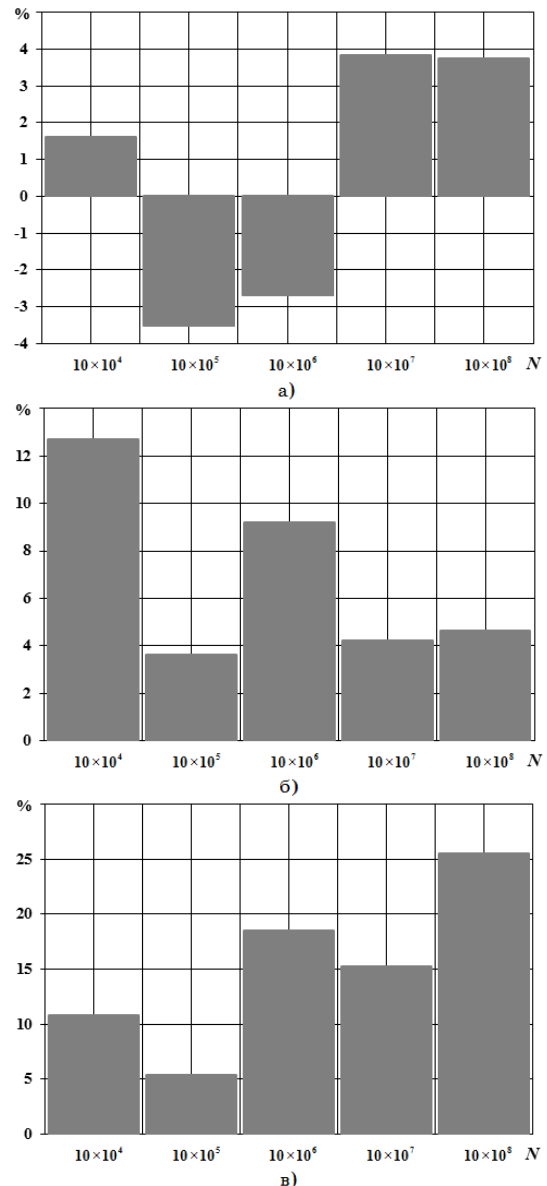


Рис. 4. Економія на збереженні файлів при використанні генетичного алгоритму відносно евристичного алгоритму: а) маленьких файлів розміром 100 Кб. — 5 Мб. б) середніх файлів 1 — 100 Мб. в) великих файлів розміром 100 — 1000 Мб

Для задач з малим розміром файлів (рис.4а) генетичний та евристичний алгоритми дають приблизно однакові результати. Різницю у $\pm 5\%$ можна розглядати, як похибку генерації параметрів пристроїв зберігання та файлів.

На файлах середнього розміру (рис. 4б) генетичний алгоритм розміщує файли помітно економніше, ніж евристичний. Це можна пояснити

ти тим, що евристичний алгоритм не враховує збільшення діапазону розмірностей файлів, тоді як генетичний обов'язково знайде якнайщільніше заповнення просторів файлами.

При використанні генетичного алгоритму для розподілення файлів з максимальним діапазоном розмірів (табл. 1) економія від за-

стосування генетичного алгоритму може сягати 20—25% у порівнянні з евристичним алгоритмом (рис. 4в). Це найкращий результат серед усіх розглянутих випадків і пояснюється він зростанням можливих варіантів розподілу і можливостями генетичного алгоритму швидко наблизитися до області найкращих рішень в цих умовах.

Таблиця 1. Розмір файлів становить 1 Мб — 1000 Мб

Кількість файлів, N_n	Кількість задіяних засобів збереження даних, генетичний алгоритм	Середні витрати на збереження даних, генетичний алгоритм, $B_{ГА}$	Кількість задіяних засобів збереження даних, евристика	Середні витрати на збереження даних, евристика, B_{EA}
$10 \cdot 10^4$	124	3,190387759	123	3,534859705
$10 \cdot 10^5$	1216	3,33726031	1224	3,515829847
$10 \cdot 10^6$	12840	3,414263153	12789	4,201988564
$10 \cdot 10^7$	118893	3,769870591	119769	4,344447618
$10 \cdot 10^8$	1312432	3,427397925	1229125	4,300784743

Загалом, генетичний алгоритм у більшості випадків дає результат, помітно кращий за евристичний алгоритм. Евристичний алгоритм показує гірший результат, оскільки не перебирає варіанти з метою вибору кращого за визначеним критерієм, а вибудовує один визначений варіант, який не буде найгіршим. Помітно, що із зростанням діапазону між мінімальним та максимальним розміром файлів ефективність алгоритмів збільшується, але за генетичним залишаються переваги, пов'язані із зручністю налаштування на різні критерії, обмеження, особливості СЗД.

Результати експериментального дослідження генетичного та евристичного алгоритмів для розв'язання задачі рівномірного розподілу файлів між пристроями з метою уникнення перевантаження дозволяють зробити висновок, що зазначені алгоритми працюють на реальних даних в умовах жорстких часових обмежень.

Проведено порівняння витраченого на розподіл часу для задачі (6). На рис. 5 наведені усереднені за результатами трьох експериментів залежності часу роботи t у секундах (вісь ординат) від розмірів N задачі (кількість файлів по осі абсцис) для генетичного (ГА) і евристичного (ЕА) алгоритмів.

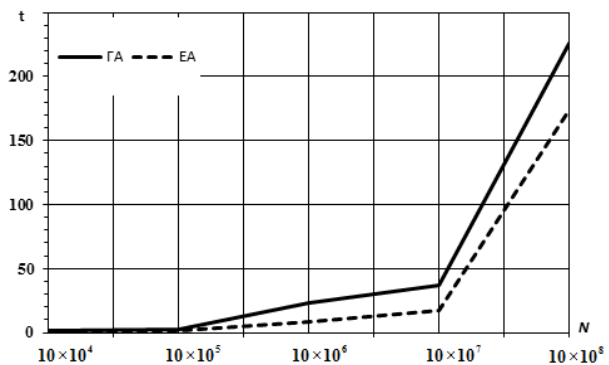


Рис. 5. Середні витрати часу на розподіл файлів між пристроями зберігання даних

Евристичний алгоритм завжди швидше знаходив розв'язок. З іншого боку, генетичний алгоритм давав розвиток, більш близький до оптимального. Різницю у часі в одну хвилину (для $10 \cdot 10^5$ файлів) можна вважати несуттєвою в задачах такого роду. Навпаки, зменшення задіяних пристроїв збереження даних від 1% до 9% для різної кількості файлів говорить про вищу ефективність генетичного алгоритму у порівнянні з евристичним.

Висновки

На основі результатів розгляду проблем, пов'язаних із створенням і використанням су -

часних СЗД обґрунтовано необхідність включення до складу систем управління ІТ-інфраструктурами підсистеми управління СЗД, побудованої на основі математичних моделей і методів розв'язання задач кластеризації та управління ресурсами СЗД.

Встановлено, що математичні моделі кластеризації та управління ресурсами СЗД доцільно будувати у вигляді булевих задач математичного програмування за декількома критеріями, насамперед мінімізації витрат і рівномірного заповнення рівнів та пристроїв, з урахуванням ресурсних, технологічних і часових обмежень.

Показано, що для розв'язання задач кластеризації та управління ресурсами СЗД можна використати евристичні алгоритми та варіанти генетичного алгоритму, які враховують специфічні особливості цих задач. Вибір методу залишається за провайдером оскільки перші вимагають менший час на розв'язання задачі, а другі дають розв'язок, ближчий до оптимального.

Перспективними напрямками досліджень видаються підбір характеристик СЗД і моделей та методів управління СЗД в залежності від типу даних (документи, шаблони, архіви, бази даних та ін.), технологій, насамперед Hadoop, Map Reduce та ін.

Список використаних джерел

1. State of the Data Center Survey: Global results // Symantec Corp. — Sept. 2012. — 12 p.
2. Дубова Н. Платформи управління бизнес-процессами / Н. Дубова // Открытые системы. — 2005. — №10.
3. Жукова С.А. Оптимизация хранения корпоративных данных /С.А. Жукова, И.Н.Ефимов//Международный журнал. Программные продукты и системы.—2007.— №4.
4. Орлов С. СХД: унифицированное и многоуровневое хранение / С. Орлов // LAN. Журнал сетевых решений — 2013 — № 1 — С. 38—40.
5. Павлов О.А. Інформаційні технології та алгоритмізація в управлінні / О.А.Павлов, С.Ф.Теленик. — К.: Техніка, 2002. — 344 с.
6. Теленик С.Ф. Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, С.А. Андросов // Автоматика. Автоматизація. Електротехнічні комплекси та системі. — 2010. — №1 (25). — С. 106—120.
7. Теленик С.Ф. Система управління інформаційно-телекомунікаційною системою корпоративної АСУ / С.Ф.Теленик, О.І. Ролік, М.М.Букасов, Р.Л.Соколовський // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: «ВЕК+». — 2006. — №45. — С. 112—126.
8. Теленик С.Ф. Управляемый генетический алгоритм в задачах распределения виртуальных машин в ЦОД / С.Ф. Теленик, А.И. Ролик, П.С. Савченко, М.Е. Боданюк // Вісник ЧДТУ. — 2011. — № 2. — С. 104—113.
9. Холланд Дж. Генетические алгоритмы / Дж. Холланд // В мире науки. — 1992. — №9 — 10. — С. 32—40.

Поступила в редакцию 21 октября 2012 г.

УДК 004.6

М.Е. Боданюк, А.К. Карнаухова, А.И. Ролик, канд. техн. наук, **С.Ф. Теленик**, д-р техн. наук
Национальный технический университет Украины "Киевский политехнический институт",
Украина, 03056, г.Киев-56, проспект Победы, 37.

Управление системами хранения данных

Рассмотрены проблемы управления многоуровневыми системами хранения данных (СХД). Разработаны математические модели для задач кластеризации и управления ресурсами СХД. Разработаны генетический и эвристический алгоритмы для решения задачи распределения ресурсов СХД по критерию минимизации затрат на хранение данных. Приведены результаты экспериментальных исследований эффективности алгоритмов для СХД различных объектов при широком диапазоне потребностей пользователей в ресурсах хранения. Предложенные рекомендации по использованию предложенных алгоритмов для решения задачи управления СХД в информационно-телекоммуникационных системах. Библи. 9, рис. 5, табл. 1.

Ключевые слова: система хранения данных, управление, ресурсы, математическая модель, генетический алгоритм, эвристический алгоритм.

UDC 004.6

M.E. Bodaniuk, O.K. Karnaukhov, O.I. Rolik, Ph.D., S.F. Telenyk, Dr.Sc.

National Technical University of Ukraine "Kyiv Polytechnic Institute",

Address: 37 Prospect Peremogy, Kiev 03056, Ukraine.

Storage area network management

The problems of effective management of tiered storage area network (SAN) are considered. The main independent variables and criteria for the operation SAN are defined. The criterion for selecting SAN management subsystem is proposed. The functional should covered with the SAN management subsystem is defined. Mathematical models for distribution users between SAN resources and management of tiered SAN resources are proposed. Genetic and heuristic algorithms for solving the problem of minimizing storage costs are proposed. The experimental results for different groups of tasks are given. It is shown that the genetic algorithm allows find a more effective solution than a heuristic algorithm. The average time spending on distribution of data depending on the dimension of the problem is estimated. The heuristic algorithm finds a solution faster than the genetic algorithm, but gives a less accurate result. Reference 9, figures. 5, tables 1.

Keywords: storage area network, tiered storage, management, resources, mathematical model, genetic algorithm, heuristic algorithm.

Reference

1. Dubova N. (2005), "Business process management platforms." Otkrytye sistemy. No.10. (Rus.)
2. Holland D. (1992), "Genetic algorithms". V mire nauki. No.9 -10, Pp.32—40 (Rus.)
3. Orlov S. (2013), "SAN: unified and tiered storage" .LAN. Zhurnal setevykh reshenij. No.1 pp.38—40 (Rus.)
4. Pavlov O.A., Telenyk S.F. (2002), "Information technology and algorithmic in managing". P. 344 (Ukr.)
5. State of the Data Center Survey: Global results. Symantec Corp. Sept. 2012. P. 12.
6. Telenyk S.F., Rolik A.I., Savchenko P.S., Bodaniuk M.E. (2011), "Controlled genetic algorithm for solving task of allocation virtual machines in data centers". Visnyk ChDTU. No. 2., Pp.104—113 (Rus.)
7. Telenyk S.F., Rolik O.I., Bukasov M.M., Androsov S.A. (2010), "Genetic algorithms for solving resource management data tasks in data centers". Avtomatyka. Avtomatyzatsiia. Elektrotekhnichni kompleksi ta systemi. No.1 (25), Pp. 106—120 (Ukr.)
8. Telenyk S.F., Rolik O.I., Bukasov M.M., Sokolovskyi R.L. (2006), "Management system of corporate ICS". Visnyk NTUU «KPI». Informatyka, upravlinnia ta obchysluvalna tekhnika. K. «VEK+». No.45, Pp. 112—126 (Ukr.)
9. Zhukova S.A., Efimov I.N. (2007), "Corporate data storage optimization". Mezhdunarodnyj zhurnal. Programmnye produkty i sistemy. No. 4. (Rus.)