8. Hernandez, E. Adaptive Sampling for Network Management [Text] / E. Hernandez, M. Chidester, A. George // Journal of Network and Systems Management. – 2001. – Vol. 9, Issue 4. – P. 409–434.
9. Andrey, L. Survey of SNMP performance analysis studies [Text] / L. Andrey, O. Festor, A. Lahmadi, A. Pras, J. Schönwälder // International journal of network management. – 2009. – Vol. 19, Issue 6. – P. 527–548. doi: 10.1002/nem.729
10. Pras, A. Comparing the Performance of SNMP and Web Services-Based Management [Text] / A. Pras, T. Drevers, R. van de Meent, D. Quartel // IEEE: Transactions on network and service management. – 2004. – Vol. 1, Issue 2. – P. 72–82. doi: 10.1109/tnsm.2004.4798292
11. Айвазян, С. А. Прикладная статистика и основы эконометрики [Текст] / С. А. Айвазян, В. С. Мхитарян. – Юнити. – 1998. – 1005 с.
12. Стрижов, В. В. Методы выбора регрессионных моделей [Текст] / В. В. Стрижов, Е. А. Крымова. – Вычислительный центр РАН, 2010. – 60 с.
13. Карабутов, Н. Н. Выбор структуры модели при обработке результатов измерений в системах управления [Текст] / Н. Н. Карабутов // Измерительная техника. – 2008. – № 9. – С. 29–33.
14. Саенко, В. И. Информационные модели наблюдаемых процессов для мониторинга компьютерных сетей [Текст] / В. И. Саенко, А. И. Гриценко // Вестник национального технического университета «ХПИ». Серия: «Новые решения в современных технологиях». – 2014. – № 48. – С. 55–66.
15. Lavy, M. Windows Management Instrumentation (WMI) [Text] / M. Lavy, A. Meggitt. – New Riders Publishing, 2001. – 432 p.
16. DMTF Std. DSP0004, Specification Version 2.6.0. Common Information Model (CIM) Infrastructure [Text] / March 2010. – 186 p.

У даній роботі розглянуто процес побудови мультибазових сховищ даних на основі властивостей даних, для яких воно будується, і запитів, які виконуються до цих даних. Проаналізовано показники швидкодії побудованого сховища, зокрема на різних етапах побудови. Досліджено вплив параметрів двофазного алгоритму, зокрема порогу адаптації процесу пошуку на основі генів

Ключові слова: мультибазові сховища даних, структурованість даних, модифікований генетичний алгоритм з адаптацією

В данной работе рассмотрен процесс построения мультибазовых хранилищ данных на основе свойств данных, для которых оно строится, и запросов, которые выполняются с этими данными. Проанализированы показатели быстродействия построенного хранилища, в том числе на различных этапах построения. Исследовано влияние параметров двухфазного алгоритма, в частности порога адаптации процесса поиска на основе генов

Ключевые слова: мультибазовые хранилища данных, структурированность данных, модифицированный генетический алгоритмы с адаптацией

УДК 004.652

DOI: 10.15587/1729-4061.2015.36646

ПОБУДОВА МУЛЬТИБАЗОВИХ СХОВИЩ ДАНИХ НА ОСНОВІ СТРУКТУРОВАНОСТІ ДАНИХ ТА ЗАПИТІВ

А. Ю. Яцишин
Провідний прикладний програміст
ДП «Головфінтех»
вул. Дегтярівська, 38-44, м. Київ,
Україна, 04119
E-mail: andrew.yatsyshyn@hotmail.com

1. Вступ

На сьогодні існують різні роботи та літературні джерела, що описують сховища даних, їх проектування та інтеграцію даних. Зокрема, запропоновані підходи є у авторів У. Інмона [1], Р. Кімболла [2] та Д. Хекні [3]. Для сховищ даних формалізовано низку моделей що визначають сховище даних і дозволяє описувати процес їх побудови. Ці моделі будуються на основі схеми «сутність-зв'язок» або похідної від неї схеми «зірка». Також представлена низка робіт, що описують різні методики оптимізації баз даних – побудову індексів, вибір матеріалізованих представлень, горизонтальну та вертикальна фрагментацію.

Автором у роботі [4] було введено мультибазові сховища даних, які включать не тільки реляційну і багатовимірну базу даних, а також бази даних з відсутністю жорсткої схеми – XML та NoSQL, і файлове сховище.

Такі сховища даних проектуються для оптимального зберігання даних різного рівня структурованості, спрямованого на підвищення швидкодії виконання запитів.

2. Аналіз існуючих досліджень та постановка проблеми

Існуючі архітектури сховищ даних будуються на підходах, які передбачають наявність деякого носія сховища – корпоративна інформаційна фабрика Інмона з використанням реляційних БД як носія, шина даних Кімболла з використанням багатовимірних БД як носія, і гібридне сховище Хекні з використанням як реляційної, так і багатовимірної БД в якості носія даних. Також широко використовуються сховища даних з застосуванням паралельних обчислень, такі як Hadoop [5], які базуються на відсутності жорсткої схеми. Однак рішення, що пропонуються в цих роботах, пропонують збереження даних на основі деякої моделі (реляційна – для реляційних БД, вимірна – для багатовимірних БД, ієрархічна – для БД XML). Підходи, представлені в цих роботах не передбачають вибору моделі, у якій краще представити дані для оптимізації виконання запитів до неї.

На сьогодні проектування (визначення схем) сховищ даних здійснюється здебільшого вручну, виходячи з предметної області, а інтеграція даних здійснюється за допомогою автоматизованих засобів. Крім того, окремо можна виділити консолідацію даних, описану у вигляді просторів даних у [6], яка також створює уніфіковане джерело даних, але оптимізація збереження не розглядається.

Для оптимізації сховищ даних використовуються матеріалізовані подання [7, 8] та фрагментація [9, 10]. Оптимізація, як правило, здійснюється евристичними алгоритмами. Однак у роботах, що пропонують такі рішення, розглядається лише задача мінімізації вартості обслуговування подань і не розглядається швидкодія запитів.

Крім того, були описані рішення з проектування сховища даних за допомогою механізмів інтеграції даних [11, 12]. Однак у цих роботах не розглядається питання визначення структур даних і проектування сховища на основі інформації про структурованість даних.

Грунтуючись на вищесказаному, можна стверджувати, що на сьогодні немає робіт, які одночасно описують і проектування сховища, і оптимізацію виконання запитів з врахуванням як структури даних, так і визначень запитів. Тому це питання є невирішеним і актуальним.

3. Мета та завдання дослідження

Метою роботи є підвищення швидкодії виконання запитів мультибазових сховищ даних та зменшення витрат часу на їх побудову за рахунок розміщення даних за їх структурованістю і вибору відповідної моделі для представлення даних на етапі проектування сховища, а також використання модифікованого генетичного алгоритму з використанням адаптації за генами.

Для досягнення поставленої мети вирішуються такі завдання:

– Розробка механізму вибору моделі за структурованістю.

– Розробка механізму адаптації процесу пошуку за генами.

– Дослідження впливу фаз проектування і оптимізації на швидкодію сховища.

– Дослідження впливу параметрів генетичного алгоритму на його ефективність та збіжність.

4. Матеріали та методи дослідження процесу побудови мультибазових сховищ даних

4. 1. Схеми даних та запити, на яких проводилося дослідження

Для експериментів використовувалися дані та запити, що використовуються у впроваджених рішеннях, зокрема «Системі управління державними фінансами» (СУДФ) та ІАС «Прозорий бюджет», тобто ці результати експериментів відображають вплив розроблених наукових результатів на роботу реальних систем.

Також використовується схема Adventure Works, яка є прикладом схеми корпоративних бізнес-застосувань і містить дані, представлені як реляційною, так і багатовимірною моделями та має різну структурованістю.

Сховище, на якому проводилися експерименти містить наступні носії: реляційна БД – Microsoft SQL Server 2014 Database Services, багатовимірна БД – Microsoft SQL Server 2014 Analysis Services, база даних XML – XBase, база даних NoSQL – MongoDB.

Для проведення досліджень використовуються набір генерованих простих запитів.

Простим запитом q з визначенням

$$\mathrm{def}_q =< W_q, T_q, S_q, O_q, G_q >, \quad (1)$$

називається запит наступного вигляду:

$$q = \Theta_{O_q} \pi_{W_q} \Sigma_{G_q} \sigma_{S_f} t_1 \triangleright\!\triangleleft_{s_{j1}} t_2 \triangleright\!\triangleleft_{s_{j2}} t_3 \triangleright\!\triangleleft_{s_{j3}} t_4 .. \triangleright\!\triangleleft_{s_{jn-1}} t_n, \quad (2)$$

де W_q – множина значень, що повертаються запитом q, які описуються атрибутами відповідних стовпчиків або функцій на цих атрибутах f(b); T_q – множина таблиць, до яких виконується запит Q. Заголовки цих таблиць, повинні включати всі атрибути, що повертаються запитом; S_q – множина умов вибору даних для запиту q, визначених на стовпчиках таблиць T_q, O_q – множина ознак сортування для запиту *q*, G_q – множина ознак групування для запиту *q*.

Для оцінки ефективності виконання запитів до вихідних баз даних виконано серію запитів до них. Всього було виконано 3200 запитів, які належать до 166 класів запитів.

4. 2. Методика визначення показників процесу побудови сховищ даних

До побудови сховища швидкодія сховища оцінюється за наступним співвідношенням:

$$\frac{1}{Q}\sum_{i=1}^{n}\frac{n_i q_i^s}{t_i^s(S)}, \quad (3)$$

де f_i^s – доля всіх запитів вибору даних класу K_i по відношенню до всіх запитів вибору у сховищі за звітний період; f_i^u – доля всіх запитів оновлення даних класу K_i по відношенню до всіх запитів оновлення за звітний період; n_i – розмір даних для еталонних запитів, визначається як кількість даних, отриманих у резуль-

таті виконання запитів, або, якщо запит не повертає даних, сумарну загальну площу таблиць, що беруть участь у запиті; t_i^s – час виконання еталонного запиту вибірки даних; t_i^s – час виконання еталонного запиту оновлення даних.

Для визначення показників процесу побудови сховищ даних потрібно виконати його дві фази :

1. Проектування сховища даних, що складається із завдання джерел даних, концептуального, логічного та фізичного проектування сховища даних.

1. 1. Концептуальне проектування (формування концептуальної моделі), при якому згідно метаданих джерел відбувається визначення структури цих джерел, представлення їх у вигляді таблиць (елементів структурованої частини), аналіз структурованості цих таблиць і виділення файлів (елементів неструктурованої частини), а також при необхідності можливе визначення додаткових таблиць та файлів.

1. 2. Логічне проектування (формування логічної моделі), яке полягає у визначенні обмежень цілісності сховища, областей та подань.

На цьому етапі знаходяться елементи сховища :

$$E_{Dw} = C(E_{Src}, M_{Src}, M_{Dw}),\ L_{Dw} = C(L_{Src}, M_{Src}, M_{Dw}). \quad (4)$$

Виділяються області сховища – множини таблиці, кожна з яких складає зв'язний граф: ($r_1 \Lambda r_2$ позначає зв'язність відношень r_1 та r_2)

$$A = \{a_i \mid a_i = \{r_z\}, \forall t_j \in a_i \exists t_k \in a_i (r_j \Lambda r_k),\}, i = \overline{1,n};$$
$$\forall t_j, t_k \in a_i \exists \{t_l\} \subset a_i (r_j \Lambda r_{l_1}, r_{l_1} \Lambda r_{l_2}, ..., r_{l_2} \Lambda r_{l_n}, r_{l_n} \Lambda r_k). \quad (5)$$

Для кожної області обчислюється структурованість її даних на основі структурованостей відношень. Структурованість для відношення r визначається за формулою:

$$ST_r = \frac{m \times n - \sum_{i=1}^{n} (n - u(a_i, r))}{m \times n}, \quad (6)$$

де $m \times n$ – розмір таблиці, $u(a_i, r)$ – невизначеність атрибута ai у відношенні r (визначається як кількість кортежів, на яких цей атрибут приймає невизначене значення).

Структурованість зв'язаної області ST_a визначається за формулою:

Після визначення структурованості ми можемо визначити розташування даних по носіях сховища.

При розподілі даних використовується принцип достатньої підтримки структурованості. Він означає, що для даних D , структурованість яких St(D) , вибирається джерело Ds , яке має підтримку даних з такою структурованістю.

$$SL(D) = Ds \mid StSup(St) \le St(D). \quad (7)$$

Для випадку мультибазових сховищ даних з реляційною, багатовимірною БД, БД XML та NoSQL значення підтримуваності даних StSup наступні:

$$StSup(M_{Rel}) \le 1,$$
$$StSup(M_{MD}) \le 2,$$
$$StSup(M_{XML}) \le 3,$$
$$StSup(M_{NoSQL}) \le 3. \quad (8)$$

1. 3. Фізичне проектування (формування фізичної моделі), яке полягає у розміщенні даних таблиць та областей у носіях сховища відповідно до їх структурованості

На цьому етапі знаходяться елементи носіїв сховища :

$$E_{Ds} = C(E_{Dw}, M_{Dw}, M_{Ds}), \quad (9)$$
$$L_{Ds} = C(L_{Dw}, M_{Dw}, M_{Ds}).$$

2. Оптимізація сховища, яка полягає в побудові класів запитів, визначення стану сховища, пошук оптимального стану сховища та зміна стану сховища.

Знаходиться такий стан сховища S, що наступна функція приймає максимальне значення

$$\sum_{i=1}^{n} n_i \left(\frac{f_i^s}{t_i^s(S)} - m(S,i) \cdot \frac{f_i^u}{t_i^s(S)} \right) \to \max,$$
$$r(S_0, S) \le L \quad (10)$$

при умові $r(S, S_0) \le L$, де S_0 – стан сховища перед оптимізацією.

де $m(S,i)$ – умова застосування компоненту штрафу по оновленню; f_i^s – доля всіх запитів вибору даних класу K_i по відношенню до всіх запитів вибору у сховищі за звітний період; f_i^u – доля всіх запитів оновлення даних класу K_i по відношенню до всіх запитів оновлення за звітний період; n_i – розмір даних для еталонних запитів (визначається як кількість даних, отриманих у результаті виконання запитів, або, якщо запит не повертає даних, сумарну загальну площу таблиць, що беруть участь у запиті); t_i^s – час виконання еталонного запиту вибірки даних; t_i^s – час виконання еталонного запиту оновлення даних; L – сумарний ліміт часу на оптимізацію.

Для розв'язання даної задачі використовується генетичний алгоритм з наступними параметрами:

– мутація є одноточковою. Імовірність (частота) мутації є змінною і залежить від довжини хромосоми наступним чином:

якщо N>32, то m=0.8, якщо N<=32, то m=0.15.

– схрещування є одноточковим. Імовірність (частота) схрещування є змінною і залежить від довжини хромосоми наступним чином:

якщо N>32, то c=0.8, якщо N<32, то c=0.4.

– селекція відбувається за методом колеса рулетки.

– поріг звірки (адаптація процесу пошуку за генами) є змінним і залежить від довжини наступним чином:

якщо N<32, то $\alpha = \lceil N/4 \rceil$, *якщо* $N \ge 32$, *то* $\alpha = \lceil 3N/4 \rceil$.

При застосуванні генетичного алгоритму використовується також адаптація за генами, яка полягає в тому, що для кожного наступного покоління знаходи-

мо пари хромосом з найбільшою кількістю спільних і найменшою кількістю змінених генів та співставляємо ті гени, які змінилися від їх батьків і якщо маємо значення цільової функції гірше, то оцінка цих особин здійснюється на основі особин з протилежним геном з певним штрафом. Це відбувається наступним чином.

Нехай на деякому кроці генетичного алгоритму маємо множину оцінених особин, що представляють собою різні стани сховища

$$I = \{i_1, i_2, ..., i_k\}, \quad i = <S, e>. \tag{11}$$

У результаті кросинговеру або мутації отримуємо інші особини та обчислюємо їх придатність.

$$I' = \{i'_1, i'_2, ..., i'_l\}, \quad i' = <S', e'>. \tag{12}$$

Потім попарно порівнюємо особини i та i', визначаємо генетичну відстань d між цими особинами як кількість генів, що змінилися між ними:

$$S = <x_1, x_2, ..., x_n>, \quad S' = <x'_1, x'_2, ..., x'_n>,$$
$$d(i,i') = d(S,S') = \sum_{i=1}^{n} |x'_i - x_i|. \tag{13}$$

Якщо знайдена генетична відстань =1, то оцінюємо придатність особини, що складається з генів початкового стану сховища, а ген, по якому є генетична відстань (13), встановлюємо у те значення, на якому придатність більша. Таку особину назвемо «початковою з модифікацією і».

Порівнюємо різницю між початковою особиною i^0 і «початковою з модифікацією і» i^0_{+j}, а також особину без гена і i^1 та оцінювану особину i^2, якщо різниці мають однаковий знак, то штраф для оцінки «гіршої» особину встановлюємо як максимум по модулю цих різниць.

Якщо знайдене $1 < d < \alpha$, де α – параметр, який назвемо *порогом звірки*, то для кожного гена, що відрізняється, змінюємо значення на протилежне, отримуємо нові значення цільової функції і фіксуємо параметри у значенні «кращої» цільової функції.

4. 3. Результати досліджень впливу фаз проектування та оптимізації на швидкодію сховища

Для оцінки ефективності виконання запитів до вихідних баз даних виконано серію запитів до них. Всього було виконано 3200 запитів, які належать до 166 класів запитів.

Виконаємо ці ж запити у сховищі. Тоді отримаємо наступні показники:

Розрахована за цими формулами мінімальна швидкодія виконання запитів складає 421.874 1/мс, середня – 517.0587 1/мс, максимальна – 608.1161 1/мс.

Запустимо алгоритм оптимізації МСД на побудованому сховищі та наборі запитів, що використовувалися раніше.

У результаті виконання алгоритму отримаємо мінімальну швидкодію виконання запитів до сховища 480.5466 1/мс. Без використання адаптивності отримуємо значення швидкодії 482.574 1/мс.

Після повторного проведення всіх запитів обчислюємо функцію (10) Розраховані значення за формулами дають мінімальну швидкодію виконання запитів до сховища – 464.9081 1/мс, середню – 560.4368 1/мс, максимальну –668.3728 1/мс.

Загальний час без використання адаптивності складає 109720.3 с, з адаптивністю – 53058.47 с.

Узагальнимо отримані результати у табл. 1.

Таблиця 1

Показники роботи сховища на різних етапах його побудови

Виконання запитів	Швидкодія виконання запитів, 1/мс		
	Мінімальна	Середня	Максимальна
Джерела даних	413.0272	502.6794	587.2584
Сховище даних	421.874	517.0587	608.1161
Оптимізоване СД	464.9081	560.4368	668.3728

У результаті проведення експериментів також було проведено порівняння розробленого алгоритму (2) зі стандартним ГА (1), з відомими методами – гілок і границь (3) і методом повного перебору (4). Результати порівняння приведені в табл. 2.

Таблиця 2

Порівняння запропонованого алгоритму з відомими

№ алг.	Результат, 1/мс				Час оптимізації, мс			
	N=10	N=20	N=30	N=60	N=10	N=20	N=30	N=60
1	1510	1201	798	468	$5 \cdot 10^6$	$34 \cdot 10^6$	$64 \cdot 10^6$	$123 \cdot 10^6$
2	1514	1201	806	478	$2 \cdot 10^6$	$2 \cdot 10^6$	$1 \cdot 10^6$	$83 \cdot 10^6$
3	1518	1201	810	–	$27 \cdot 10^6$	$30 \cdot 10^6$	$268 \cdot 10^6$	–
4	1528	–	–	–	$380 \cdot 10^6$	–	–	–

Дані результати обговорюються в п. 4. 4.

Також було проведено дослідження впливу параметрів генетичного алгоритму. Для цього були проведені експерименти з різними довжинами хромосом, а саме $N = 12$, $N = 23$, $N = 28$, $N = 37$, $N = 65$). Такий вибір областей дозволяє аналізувати вплив досліджуваних показників у залежності від розміру хромосоми.

Було отримано наступні результати:

Поріг звірки адаптації процесу пошуку за генами α. При проведенні експерименту для $N = 12$ бачимо, що при збільшенні порогу адаптації ГА зменшується час, необхідний на оптимізацію. Однак, швидкодія оптимізованого сховища суттєво зменшується при збільшенні α до 7. Тому для $N = 12$ доцільно вибирати $\alpha = 4$, при цьому значенні час виконання алгоритму зменшується в 2.5 рази порівняно зі звичайним ГА, а швидкодія сховища залишається аналогічною тією самою.

При $N = 23$ був виявлений невеликий потенціал для оптимізації, в цьому випадку досягається однакове значення швидкодії за набагато менший час (до 30 % від початкового результату), а саме при $\alpha = 20$.

Для $N = 28$ максимальне значення швидкодії досягається при α =6, схожого значення було досягнуто

при α =13 при 40 % часу порівняно з α =6 . Окремо треба відзначити, що значення швидкодії між α =6 і α =13 та α =13 і α =2 аналогічні, що може казати про кратність мінімальній генетичній відстані між особинами 7.

Для $N = 37$ був виявлений великий діапазон значень як часу проектування (від 1 до 7 мільйонів мс), так і значень функції швидкодії (від 700 до 820 1/мс). Найбільше значення досягається при α =1, а найбільшого значення за менший час можна досягти при α =25.

Для $N = 65$ найбільша швидкодія досягається при α =25, в той час як значення функції придатності аналогічно повторюються з кроком 6, виключною ситуацією є значення для α =25.

Мутація. Аналогічні випробування проводилися й для дослідження впливу значення імовірності мутації m на швидкодію та збіжність алгоритму.

Для $N = 12$ дослідження показали, що зі збільшенням імовірності мутації збільшується й час виконання алгоритму. Це пов'язано з появою особин з більшою генетичною відстанню. Однак при цьому підвищується й генетичне різноманіття, тому алгоритм дає ближчий до оптимуму результат.

Для $N = 23$ з огляду на низький оптимізаційний потенціал значення швидкодії не змінюється після $m > 0.1$, тому доцільно вибрати таке значення.

Для $N = 28$ значення швидкодії слабо змінюється після $m > 0.1$, тому також доцільно вибрати таке значення.

Для $N = 37$ вище значення швидкодії досягається на $m = 0.9$, що свідчить про необхідність частої мутації.

Для $N = 63$ вище значення швидкодії досягається на $m = 0.15$. Хоча хромосома є довгою, однак алгоритм є добре збіжним в цьому випадку в силу низького оптимізаційного потенціалу.

Кросинговер. Аналогічні випробування проводилися й для імовірності схрещування c .

Для $N = 12$ дослідження показали, що найкраще значення досягається при імовірності схрещування значенні $c = 0.9$, однак за найдовший час, хоча аналогічне значення досягається й при значенні $c = 0.35$ за набагато менший час.

Для $N = 23$ з огляду на низький оптимізаційний потенціал значення швидкодії не змінюється і достатньою імовірністю схрещування є значення $c = 0.05$.

Для $N = 28$ найкраще значення досягається при $c = 0.95$, хоча можна досягти схожого значення при $c = 0.45$ за менший час.

Для $N = 37$ найкраще значення досягається при $c = 0.8$, схожий результат з трохи меншим часом отримуємо при значенні $c = 0.95$.

Для $N = 63$ найкраще значення досягається при $c = 0.9$, схожий результат з трохи меншим часом отримуємо при значенні $c = 0.65$.

4. 4. Обговорення результатів досліджень показників процесу побудови сховищ даних

За результатами проектування сховища бачимо, що завдяки проектуванню сховища приріст швидкодії складає до 5 %, оптимізації, до 10 %. Даний результат може бути досягнутий як з використанням адаптивності, так і без неї, однак з її використанням алгоритму потрібно на 50 % менше часу на досягнення цих результатів.

У порівнянні з іншими алгоритмами, результати яких також отримано в роботі, запропонований алгоритм є найбільш ефективним серед вищезазначених і дозволяє отримати кращий результат за меншу кількість часу. Зокрема, він затрачає у 10 раз менше часу, ніж метод гілок і границь, і у 100 раз менше, ніж метод повного перебору для довжини хромосоми N=10, а на довших хромосомах ще менше, при цьому знаходячи рішення, близьке до оптимуму.

Окремо визначимо, що на етапі проектування сховища приріст швидкодії по областям, що мають слабкоструктуровані дані, оскільки саме за рахунок виконання запитів на відповідних носіях (БД XML) і отримується цей приріст.

Результати дослідження процесу оптимізації сховища для виявлення впливу параметрів дозволяють зробити наступні висновки щодо окремих параметрів.

Поріг адаптації. Можливий виграш від використання адаптивності за генами сильно залежить від оптимізаційного потенціалу області, а точніше від віддаленості оптимальних значень від початкових. При малих значеннях N доцільно вибирати малі значеннях $\alpha < N/2$, на великих значеннях – великі значеннях $\alpha > N/2$

Мутація. Як показали проведені випробування, імовірність мутації доцільно вибирати невелику, зі значенням близьким до 0.15. Це пояснюється тим, що при високих ймовірностях мутації збільшується генетична відстань, що призводить до більшого часу виконання алгоритму, однак не забезпечує краще значення функції швидкодії. В деяких випадках, зокрема при великих значеннях N, доцільно вибирати $m > 0.8$ для швидшого дослідження області допустимих рішень.

Кросинговер. У результаті аналізу даних можемо стверджувати, що для малих значеннях N доцільно вибирати малу імовірність схрещування $[0.35, 0.45]$ при великих значеннях N доцільно вибирати велику імовірність схрещування $[0.8, 0.9]$, хоча в окремих випадках можливе отримання схожих результатів при значеннях $c \in [0.45, 0.65]$.

Результати проведених досліджень дозволяють зробити висновки про підтвердження наступних гіпотез.

Гіпотеза про вплив розподілу даних за структурованістю. Розподіл даних по структурованості дозволяє пришвидшити виконання запитів до цих даних за рахунок використання відповідних носіїв. За рахунок виконання операцій над слабкоструктурованими даними за допомогою БД XML та MongoDB отримуємо ефективніше виконання операцій над частково-структурованими даними (до 5 %).

Гіпотеза про вибір еталонного запиту. Вибір найменш швидкодійного запиту в якості еталонного дозволяє оптимізувати все сховище, тобто оптимізуються всі запити.

Гіпотеза справджується, хоча існує деяка множина запитів з низькою швидкодією, що не мають потенціалу для оптимізації. Водночас, як показали результати експериментів, використовувані методи оптимізації підвищують швидкодію всіх запитів до сховища, що мають такий потенціал, а саме – розмір даних який перевищує деяку величину s (у наведених дослідженнях використовувалося s=100).

Гіпотеза про вплив адаптації за генами на процес пошуку. Використання адаптивності за генами дозволяє визначити «хворобливі» гени і відсіктиці лі «підобласті» області допустимих рішень, що спрощує задачу і дозволяє скоротити час, необхідний на виконання алгоритму. Гіпотеза підтверджується, оскільки завдяки низькій імовірності мутації і близькості особин у поколіннях ми отримуємо малу відстань між мутованими і немутованими особинами, що дає нам можливість виявити впливи окремих генів і щвидко зменшити розмірність задачі за рахунок фіксації значень. Крім того, використання схрещування для отримання особин з великою відстанню дозволяє нам «відійти» від ділянок з низькими значеннями.

Оскільки інші дослідження не розглядали питання оптимізації функції інтегральної швидкодії виконання запитів до сховища даних, побудовану за класами запитів, порівняння результатів досліджень з іншими дослідженнями не є можливим.

У подальших дослідженнях варто проаналізувати вплив окремих блоків стану сховища на зміну цільової функції. Також варто провести описані в даній роботі експерименти на великих обсягах даних і з використанням інших оптимізацій (розпаралелювання зберігання даних і обробки запитів, розміщення частин сховища даних у оперативній пам'яті тощо).

Дослідження виконувалося на невеликих обсягах даних. Також дослідження виконувалося без використання паралельних БД.

Це дослідження є продовженням робіт [4, 13–17] автора з проектування та оптимізації сховищ даних. Зокрема, воно представляє задачу проектування сховищ даних [13] як задачу перетворення даних.

У роботах [15] було описано використання даних про структурованість для побудови сховищ даних. У цій роботі формалізовано сам механізм вибору моделі для представлення даних.

Двофазний алгоритм проектування та оптимізації [14] був деталізований процедурою адаптації процесу пошуку за генами

Результати досліджень впроваджені у вигляді інформаційних технології для інтегрованої системи управління державними фінансами, Інформаційно-аналітичні системи «Прозорий бюджет» [18], а також у Інформаційних системах підвищення компетентності фахівців фінансової сфери та Порталах «Віртуальний університет» [17], Інституту післядипломної освіти, журналу «Фінанси України».

7. Висновки

1. Розроблено механізм вибору моделі за структурованістю, це дає можливість оптимізувати сховище даних, коли не відомі запити, які до нього виконуються. Вибір здійснюється за допомогою порівняння обчисленої структурованості даних, для яких вибирається модель, і значення підтримки структурованості моделей носіїв даних сховища. Вибирається та модель, якої достатньо для збереження даних з такою структурованістю.

2. Розроблено генетичний алгоритм з адаптацією процесу пошуку за генами, яка дозволяє визначити вплив генів на функцію оптимізації під час виконання алгоритму. Даний алгоритм відрізняється від класичного генетичного алгоритму тим, що після селекції, після мутації і після схрещування відбувається знаходження особин з малою генетичною відстанню. Якщо відстань не перевищує так званий поріг звірки, визначається вплив генів, що відрізняються в початкових та цих особинах. Якщо виявляється негативний вплив гену на цільову функцію, надалі використовується обернене значення.

3. У результаті дослідження впливу фаз проектування і оптимізації на швидкодію сховища було виявлено, що проектування сховища з вибором моделі дозволяє підвищити швидкодію виконання запитів від 5 %. Оптимізація сховища з використанням механізмів розміщення, індексації, матеріалізації, вертикальної і горизонтальної фрагментації та злиття дозволяє підвищити швидкодію виконання запитів від 10 %.

4. Дослідження впливу параметрів генетичного алгоритму на його ефективність та збіжність показали, що використання адаптації процесу пошуку за генами дозволяє скоротити час виконання алгоритму за рахунок визначення впливу параметрів до 50 %.

Література

1. Inmon, W. H. Corporate Information Factory Components [Electronic resource] / W. H. Inmon. – Inmon Data Systems. – Available at: http://www.inmoncif.com/view/26
2. Kimball, R. The data warehouse toolkit: the complete guide to dimensional modeling [Text] / R. Kimball. – New York, Wiley, 2002. – 436 p.
3. Hackney, D. Architectures and Approaches for Successful Data Warehouses [Electronic resource] / D. Hackney. – Available at: http://www.egltd.com/presents/ArchitecturesApproaches.pdf
4. Томашевський, В. М. Особливості проектування гібридних сховищ даних з врахуванням джерел даних [Текст] / В. М. Томашевський, А. Ю. Яцишин // Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі: збірник наукових праць. – 2011. – № 715. – С. 246–254.
5. Thusoo, A. Hive – a petabyte scale data warehouse using Hadoop [Текст] / A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang et. al. // Data Engineering (ICDE), 2010 IEEE 26th International Conference, 2010. – P. 996–1005. doi: 10.1109/icde.2010.5447738
6. Шаховська, Н. Б. Організація просторів даних у складних інформаційних системах [Текст] : автор. ... дис. ... д-р техн. наук : 05.13.06 / Н. Б. Шаховська. – Національний університет «Львівська політехніка», 2012. – 39 с.
7. Zhou, L. An Improved Approach for Materialized View Selection Based on Genetic Algorithm [Text] / L. Zhou, X. He, K. Li // Journal of Computers. – 2012. – Vol 7, Issue 7. – P. 1591–1598. doi: 10.4304/jcp.7.7.1591-1598
8. Mami, I. A survey of view selection methods [Text] / I. Mami, Z. Bellahsene // ACM SIGMOD Record. – 2012. – Vol. 41, Issue 1. – P. 20–29. doi: 10.1145/2206869.2206874

9. Dimovski, A. Horizontal partitioning by predicate abstraction and its application to data warehouse design [Text] / A. Dimovski, G. Velinov, D. Sahpaski // Advances in Databases and Information Systems. Lecture Notes in Computer Science. – 2010. – Vol. 6295. – P 164–175. doi: 10.1007/978-3-642-15576-5_14
10. Elmansouri, R. The fragmentation of data warehouses: An approach based on principal components analysis [Text] / R. Elmansouri, E. Ziyati, O. Elbeqqali, D. Aboutajdine // International Conference on Multimedia Computing and Systems (ICMCS), 2012. – P. 18–23. doi: 10.1109/icmcs.2012.6320319
11. Jarke, M. Architecture and Quality in Data Warehouses [Text] / M. Jarke, M. A. Jeusfeld, C. Quix, P. Vassiliadis // Seminal Contributions to Information Systems Engineering, 2013. – P. 161–181. doi: 10.1007/978-3-642-36926-1_13
12. Siebert, J. C. The Stanford Data Miner: a novel approach for integrating and exploring heterogeneous immunological data [Text] / J. C. Siebert, W. Munsil, Y. Rosenberg-Hasson, M. M. Davis, H. T. Maecker // Journal of Translational Medicine. – 2012. – Vol. 10, Issue 1. – P. 62. doi: 10.1186/1479-5876-10-62
13. Яцишин, А. Ю. Проектування мультибазових сховищ даних на основі двохфазного алгоритму [Текст] / А. Ю. Яцишин // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка : збірник наукових праць. – 2012. – № 55. – С. 125–134.
14. Яцишин, А. Ю. Проектування гібридних сховищ даних з врахуванням структурованості даних [Текст] / А. Ю. Яцишин // Управління розвитком складних систем. – 2012. – Вип. 9. – С. 59–65.
15. Роль віртуального університету у забезпеченні прозорості бюджетного процесу в монографії Державний бюджет і бюджетна стратегія в умовах економічних реформ: у 4 т. Т. 2 [Текст] / за заг. ред. М.Я. Азарова. – ДННУ «Акад. фін. управління». К, 2011. – С. 878–902.
16. Соціальна технологія «Прозорий бюджет» як інновація в монографії Державний бюджет і бюджетна стратегія в умовах економічних реформ: у 4 т. Т. 4 [Текст] / за заг. ред. М.Я. Азарова. – ДННУ «Акад. фін. управління». К, 2011. – С. 327–381.

Визначено поняття соціально-інформаційного інтерфейсу. Виділено стратегії інформаційної діяльності вищого навчального закладу в соціальних середовищах Інтернету. Визначено показник важливості для генератора інформаційного образу. Сформовано задачу про призначення відповідальності підрозділів за генератори та визначено прогнозовану ефективність інформаційної діяльності підрозділу у генераторі

Ключові слова: генератор, інтегрований показник, підрозділи у генератора, вищий навчальний заклад (ВНЗ)

Определено понятие социально-информационного интерфейса. Выделены стратегии информационной деятельности высшего учебного заведения в социальных средах Интернета. Определены показатель важности для генератора информационного образа. Сформирована задача о назначении ответственности подразделений за генераторы и определены прогнозируемую эффективность информационной деятельности подразделения в генераторе

Ключевые слова: генератор, интегрированный показатель, подразделения в генераторах, высшее учебное заведение (ВУЗ)

УДК 004.738.5

DOI: 10.15587/1729-4061.2015.37031

ФОРМУВАННЯ СТРУКТУРИ СОЦІАЛЬНО-ІНФОРМАЦІЙНИХ ІНТЕРФЕЙСІВ ЯК ВИРІШЕННЯ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ ВІДПОВІДАЛЬНОСТІ

А. М. Пелещишин
Доктор технічних наук, професор
Кафедра соціальних комунікацій
та інформаційної діяльності*
E-mail: apele@ridne.net
Р. О. Корж
Кандидат технічних наук, доцент
Кафедра електронних засобів
інформаційно-комп'ютерних технологій*
E-mail: korzh@lp.edu.ua
*Національний університет «Львівська політехніка»
вул. С. Бандери, 12, м. Львів, Україна, 79013

1. Вступ

Соціально-інформаційним інтерфейсом (СІІ) називатимемо генератор інформаційного образу ВНЗ, у якому офіційно зареєстровано представників ВНЗ. Статус інтерфейсу для спільноти означає, що ВНЗ використовує її як один з каналів взаємодії зі суспільством, несе відповідальність за його інформаційну підтримку. Для ВНЗ спільноти, що є визначені як інтерфейси, повинні розглядатися як певного роду