УДК 621.391
С.В. Гринюк, О.І. Міскевич
Луцький національний технічний університет

ДОСЛІДЖЕННЯ АЛГОРИТМУ ПОСИМВОЛЬНОГО СТИСНЕННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ В АДРЕСНИХ БАЗАХ ДАНИХ

Гринюк С.В., Міскевич О.І. Дослідження алгоритму посимвольного стиснення текстової інформації в адресних базах даних. В статті роглядається посимвольне стиснення інформації в адресних базах даних на основі модифікованого алгоритму Хаффмана. Даний алгоритм покращує показники ефективності стиснення текстової інформації в базах даних, що сприятиме зменшенню вартості їх зберігання та передачі.
Ключові слова: стиснення даних, алгоритм хаффмана, блочно-статистичний алгоритм, бази даних.

Гринюк С.В., Мискевич О.И. Исследование алгоритма посимвольного сжатия текстовой информации в адресных базах данных. В статье роглядается символьная сжатия информации в адресных базах данных на основе модифицированного алгоритма Хаффмана. Данный алгоритм улучшает показатели эффективности сжатия текстовой информации в базах данных, что приведет к уменьшению стоимости их хранения и передачи.
Ключевые слова: сжатие данных, алгоритм Хаффмана, блочно-статистический алгоритм, базы данных.

Grunjuk S., Miskevuch O. Research compression algorithm spelling of text information in the address database. The article rohlyadayetsya character-compression in the address database based on modified Huffman algorithm. This algorithm improves the compression performance of text information in databases to help reduce the cost of storage and transmission.
Keywords: data compression, Huffman algorithm, block-statistical algorithm database.

Вступ. Інформація – це не просто наукова категорія, а комерційна, яка є таким же принциповим фактором розвитку, як сировина, енергія. Тепер для відновлення запасів сировини і енергії людство гостро потребує інформації. Інформація відкриває нові шляхи більш раціонального та економного отримання коштів для подальшого науково-технічного прогресу, розвитку всіх сфер людської діяльності.

Інформаційні ресурси – продукт інтелектуальної діяльності найбільш кваліфікованої й творчо активної частини працездатного населення. Таким чином необхідне знаходження та застосування принципово нових методів і засобів сприйняття, передачі, обробки, зберігання і розповсюдження інформації, здатних оперувати з великими масивами інформації в реальному часі.

В даний час це реалізовано – на базі комп'ютерів створена інформаційна індустрія, що визначила перехід до без паперових технологій обміну інформацією на основі відеотелефонів, факсимільної передачі документів, електронної пошти, телеконференцій, локальних і глобальних мереж передачі даних, супутникового зв'язку, баз і банків даних , інформаційно-пошукових систем, автоматизованих робочих місць.

У зв'язку з великими обсягами інформації актуальним стало питання їх економного зберігання та передачі. Інтерес до задачі стиснення даних в СУБД спочатку був обумовлений прагненням зменшити фізичний обсяг баз даних. Ціна підсистеми вводу-виводу становила основну частину вартості апаратури. Тому при належному інтегруванні в СУБД методів стиснення без втрат даних досягалася значна економія.

Постановка проблеми. Характерною особливістю більшості типів даних є їх надлишковість. Коли мова йде про зберігання та передачу інформації засобами комп'ютерної техніки, то надлишковість відіграє негативну роль, оскільки вона приводить до зростання вартості зберігання та передачі інформації.

Особливо актуальною є ця проблема у випадку необхідності обробки величезних обсягів інформації при незначних об'ємах носіїв даних. У зв'язку з цим постійно виникає проблема позбавлення надлишковості або стиснення даних.

Основний принцип, на якому базується стиснення даних, полягає в економічному описі повідомлення, згідно якому можливе відновлення початкового його значення з похибкою, яка контролюється [1, 3].

Основоположником науки про стиснення інформації прийнято рахувати Клода Шеннона. Його теорема про оптимальне кодування показує, до чого потрібно прагнути при кодуванні інформації і на скільки та або інша інформація при цьому стиснеться.

Один з перших алгоритмів ефективного кодування інформації був запропонований Д. А. Хаффманом в 1952 році. Ідея алгоритму полягає в наступному: знаючи ймовірності

символів у повідомленні, ми можемо описати процедуру побудови кодів змінної довжини, що складаються з цілої кількості бітів. Символам з більшою ймовірністю ставляться у відповідність коротші
коди. Коди Хаффмана мають властивість префіксності (тобто жодне кодове слово не є префіксом іншого), що дозволяє однозначно їх розкодувати.

Класичний алгоритм Хаффмана на вході отримує таблицю частот зустрічальності символів у повідомленні. Далі на підставі цієї таблиці будується дерево кодування.

Питання економного кодування інформації в системах управління базами даних було поставлено в першій половині 1970-х років [8], але воно не втратило актуальності і до сих пір. Багато сучасних дослідників відзначають недостатню теоретичну опрацьованість проблеми і неефективність підтримки стиснення даних в промислових СУБД [6].

Застосування в СУБД економного кодування без втрат інформації призводить до ряду позитивних результатів. Найбільш очевидним ефектом є зменшення фізичного розміру бази даних, журнальних та архівних файлів. Але також часто досягається збільшення швидкості виконання запитів і зниження вимог до обсягу оперативної пам'яті, що відзначається практично у всіх роботах в даній галузі знань. Тому ефективна реалізація підтримки стиснення даних істотно покращує якість СУБД.

Економне кодування сприяє криптографічному захисту інформації. Усунення статистичної надмірності підвищує крипостійкість алгоритмів шифрування інформації і часто є попереднім етапом в схемах шифрування даних [59].

Для застосування в СУБД потрібні специфічні методи і прийоми економного кодування, оскільки звичайні методи не задовольняють ряду вимог. Практична цінність реалізації досягається тільки при забезпеченні швидкого доступу до довільного запису або елемента даних. Затребувані так звані методи стиснення зі збереженням упорядкованості, що дозволяють виконувати операції порівняння без декодування даних [5, 7].

Проблема реалізації стиснення в СУБД має також такі аспекти, як оптимізація виконання запитів до стиснених даних, ефективне стиснення результатів виконання запитів, економне кодування метаданих, оцінка доцільності стиснення окремих елементів, вибір алгоритму стиснення і його параметрів з урахуванням типових запитів і характеру даних.

Великі обсяги інформації, що збирається, і вимоги до скорочення строків її надання споживачам обумовлюють необхідність використання для її обробки та узагальнення сучасної електронної техніки. Для зберігання, автоматизації та узагальнення інформації створюються спеціальні банки даних.

Ефективний та відкритий доступ до інформаційних ресурсів базується на використанні інформаційного сервісу глобальної мережі Інтернет, тобто на основі Web-технологій. З цією метою розробляються системи для роботи зі структурами метаданих, що забезпечують збір і розподіл експериментальних даних і результатів тематичної обробки, при цьому архів об'єднується з регіональними центрами глобальною мережею Інтернет. Важливим елементом є розробка структури інтерфейсу, архівування та мережевого обміну даними. Це вимагає розвитку пошукових систем та реалізації вилученого інтерактивного доступу зовнішніх користувачів по мережі Інтернет до даних і електронних каталогів, надання користувачам можливості для інтерактивного доступу до них в режимі on-line [2, 4].

Метою роботи є дослідження модифікованого алгоритму Хаффмана.

Методи стиснення даних можна розділити на два типи:

1. Методи без спотворення (loseless) - методи стиснення (звані також методами стиснення без втрат) гарантують, що декодовані дані будуть в точності збігатися з вихідними;
2. Методи з втратами (lossy) - методи стиснення (звані також методами стиснення з втратами) можуть спотворювати вихідні дані, наприклад за рахунок видалення несуттєвою частини даних, після чого повне відновлення неможливо [40].

Методи стиснення даних без втрат інформації засновані на усуненні надмірності подання інформації. Економне кодування досягається за рахунок подання малоймовірних подій більш довгими словами, чим подій з високою ймовірністю настання. Якщо ймовірність настання події дорівнює p, то, відповідно до теореми Шеннона про кодування

джерела інформації, така подія найвигідніше кодувати словом завдовжки $-log_2 p$ бітів. Методи стиснення даних явно або неявно спираються на цей факт.

В результаті процесу економного кодування одиниці вихідних даних (символу, слову, рядку, числу і т.п.) ставиться у відповідність так зване кодове слово. Кодове слово складається з послідовності цифр, звичайно двійкових. Сукупність усіх кодових слів утворює код. Якщо довжини всіх кодових слів однакові, то використовується код має фіксовану (постійну) довжину, інакше -змінну. Якщо вихідні дані можуть бути однозначно відновлені по масиву відповідних кодових слів, то кодування не призводить до втрати інформації.

Ефективність стиснення як характеристика скорочення розміру подання інформації щодо початкового визначається ступенем стиснення. Ступінь стиснення приймається рівною відношенню обсягу вихідних даних до обсягу відповідних їм стиснутих даних і вимірюється в бітах.

Алгоритм Хаффмана. Алгоритм Хаффмана – є адаптивним алгоритмом оптимального префіксного кодування алфавіту з мінімальною надмірністю.

В даний час використовується в багатьох програмах стиснення даних.

Цей метод кодування складається з двох основних етапів:

1. Побудова оптимального кодового дерева.
2. Побудова відображення код-символ на основі побудованого дерева.

Один з перших алгоритмів ефективного кодування інформації був запропонований Д. А. Хаффманом в 1952 році. Ідея алгоритму полягає в наступному: знаючи ймовірності символів у повідомленні, можна описати процедуру побудови кодів змінної довжини, що складаються з цілої кількості бітів. Символам з більшою ймовірністю ставляться у відповідність коротші коди. Коди Хаффмана мають властивість префіксності (тобто жодне кодове слово не є префіксом іншого), що дозволяє однозначно їх декодувати.

Класичний алгоритм Хаффмана на вході отримує таблицю частот зустрічальності символів у повідомленні. Далі на підставі цієї таблиці будується дерево кодування Хаффмана [7].

1. Символи вхідного алфавіту утворюють список вільних вузлів. Кожен лист має вагу, який може дорівнювати або ймовірності, або кількістю входжень символу в повідомлення що стискається.
2. Вибираються два вільних вузла дерева з найменшими вагами.
3. Створюється їх батько з вагою, рівною їх сумарній вазі.
4. Батько додається в список вільних вузлів, а два його нащадка видаляються з цього списку.
5. Одній дузі, котра виходить з батьківського вузла, ставиться у відповідність біт 1, інший - біт 0.
6. Кроки, починаючи з другого, повторюються до тих пір, поки в списку вільних вузлів не залишиться тільки один вільний вузол. Він і буде вважатися коренем дерева.

На відміну від алгоритму Шеннона - Фано, алгоритм Хаффмана залишається завжди оптимальним і для вторинних алфавітів з більш ніж двома символами.

Класичний алгоритм Хаффмана має ряд істотних недоліків. По-перше, для відновлення вмісту стиснутого повідомлення декодер повинен знати таблицю частот, якою користувався кодер. Отже, довжина стиснутого повідомлення збільшується на довжину таблиці частот, яка повинна надсилатися попереду даних, що може перекреслити всі зусилля зі стиснення повідомлення. Крім того, необхідність наявності повної частотної статистики перед початком власне кодування потребує двох проходів за повідомленням: одного для побудови моделі повідомлення (таблиці частот і Н-дерева), іншого для кодування.

Стиснення інформації є однією з тих проблем, яка нерозривно пов'язана з обробкою даних з використанням засобів обчислювальної техніки. Текст і звук, графіка і відео - для кожного з цих видів інформації існують свої найбільш відповідні методи стиснення. Метою процесу стиснення, як правило, є отримання компактнішого способу представлення початкових даних, який мінімізує об'єм займаної пам'яті за допомогою деякого їх перетворення.

Існує декілька різних підходів до проблеми стиснення інформації, які базуються або на складних математичних алгоритмах або засновані на властивостях інформаційного потоку і алгоритмічно досить прості. Проте усі способи стиснення можна розділити на дві категорії:

оборотне і безповоротне стиснення. У першому випадку можливо повне і безпомилкове відновлення початкових даних, які були піддані стисненню. У другому - це неможливо.

Проте з точки зору практичного застосування результат, отриманий в процесі стиснення, може бути цілком задовільний, наприклад, для графіки або звукових повідомлень.

Особливий інтерес викликає застосування методів стиснення у базах даних для зменшення об'єму текстових полів. По-перше, для таких даних неприпустимо використання алгоритмів безповоротного стиснення, а по-друге, застосування алгоритмів, закладених в широко відомих архіваторах, наприклад, RAR або ZIP, є неефективним з причини того, що стисненню піддаються текстові поля невеликого об'єму і в цьому випадку отриманий при стисненні результат, з урахуванням даних для декодування, може займати об'єм пам'яті більший, ніж початкові дані.

Одним з ефективних методів, який застосовується у базах даних для стиснення, є класичний алгоритм Хаффмана, що базується на знанні частоти розподілу символів в тексті і дає в результаті, набір кодів змінної довжини для усього вхідного алфавіту. В результаті цього, хоча окремі символи і рідко зустрічаються у тексті, з причини їх довгої кодової послідовності, середня довжина коду символу може бути досить великою.

Блок-схема модифікованого алгоритму зображена на рис. 1.

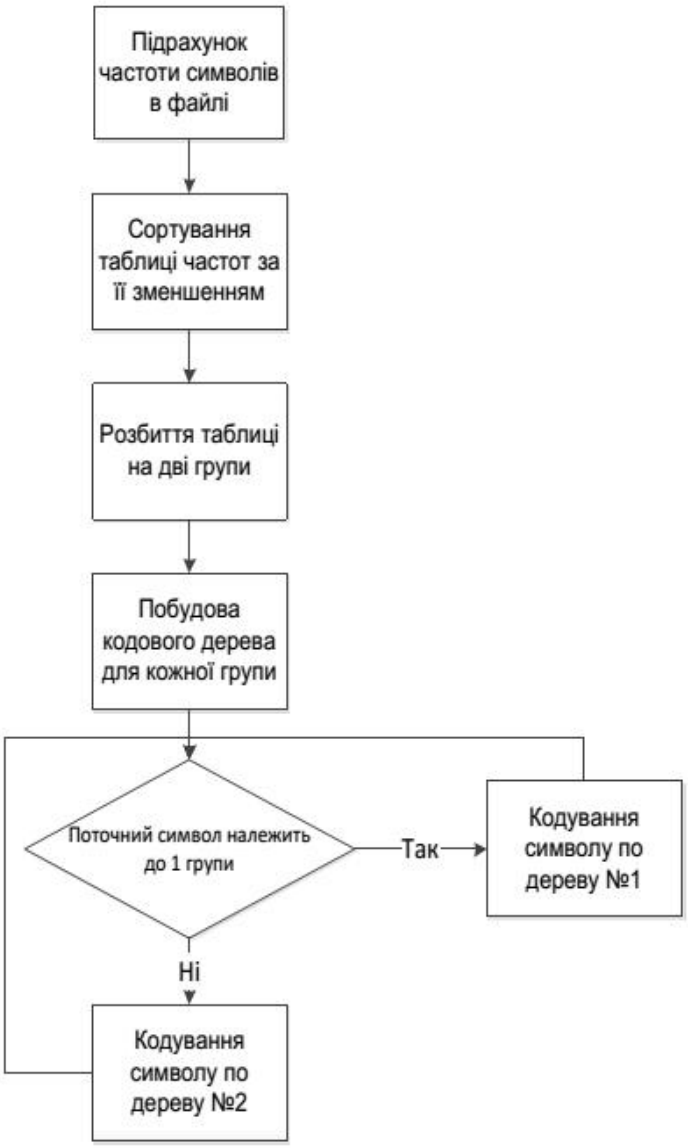

Рис. 1 – Блок-схема модифікованого алгоритму Хаффманна

Висновки. Основною ідеєю алгоритму Хаффмана є те, що можна кодувати всі символи різним числом біт. Символи, які зустрічаються частіше, будуть закодовані меншим числом біт, ніж ті, які зустрічаються рідше. Отриманий код буде оптимальний або, іншими словами, мінімально-надлишковий. Алгоритм Хаффмана - є адаптивним алгоритмом оптимального префіксного кодування алфавіту з мінімальною надмірністю. Є одним з перших алгоритмів ефективного кодування інформації.

Список використаних джерел

1. Артюшенко В. Цифровое сжатие информации / Артюшенко В. М., Шелухин О.И. – М.: Дашков и Ко, 2004. – 426 с.
2. Мартынов Н. Информатика. С для начинающих / Мартынов Н. - М.: Кудиц-Образ, 2006. – 304с.
3. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Морелос-Сарагоса Р. – М.: Техносфера, 2006. – 320с.
4. Сем□нов Ю. Телекоммуникационные технологи / Семенов Ю. – М.:Бином. Лаборатория знаний, 2007. – 640с.
5. Cannane A. A General-Purpose Compression Scheme for Databases / Cannane A., Williams H. E., Zobel J. - Proc. IEEE Data Compression Conference, p. 519, 1999.
6. Chen Z., Query Optimization in Compressed Database Systems / Chen Z., Gehrke J., Korn F. - Proc. 2001 ACM-SIGMOD Int. Conf. Management of Data, pp. 271-282, Santa Barbara, CA, May 2001.
7. Cormack G. Data Compression in Database Systems / Cormack G. - Comm. of ACM, 28(12):1336-1342, December 1985.
8. Moffat A. Coding for compression in full-text retrieval systems / Moffat A., Zobel J. - In Proc. IEEE Data Compression Conference, pp. 72-81, Snowbird, Utah, March 1992. IEEE Computer Society Press, Los Alamitos, California.
9. Roth M., Database compression / Roth M., Van Horn S. - ACM SIGMOD Record, 22(3):31-39, Sept. 1993.