Аналогічно доводиться, що $l_q \neq 2q_{m-1}$ і $l_q \neq 1$.

Отримані суперечності доводять, що не існує простого дільника q числа q_m, такого що $q < q_m$. Це означає, що q_m - просте число.

II. Нехай $2q_{m-1} < l_q$. Тоді $2q_{m-1} \equiv l_q (\mathrm{mod}\, 2q_{m-1})$. Тобто $2q_{m-1} | l_q$ і $l_q = L_q 2q_{m-1}$, де натуральне $L_q < l_q$.

Нерівність $2q_{m-1} < L_q$ відповідно до АНЗ не може виконуватися.

Отже, $L_q < 2q_{m-1}$, $L_q | (q - 1)$ і $2^{L_q} \equiv 2^{q_{m-1}} (\mathrm{mod}\, q)$. Тоді, як в п.І., аналогічно доводиться, що $L_q \neq 1, 2, q_{m-1}, 2q_{m-1}$, що є суперечністю, з якої випливає, що q_m - просте число. Теорема 2 доведена.

Наслідок 3. Досконалих парних чисел E_n - нескінченна множина, оскільки, як відомо [3, с.42], таке число має вигляд $E_n = 2^{n-1}M_n$, де M_n - просте число Мерсенна. Зауважимо, що кожне парне досконале число є значенням полінома $h(x) = 2x(4x - 1)$, оскільки при $x = 2^{n-1}$, $n \geq 2$, отримуємо що $h(2^{n-1}) = 2^{n-1}M_n$.

Наслідок 4. У зв'язку з простими числами Мерсенна і простими числами Ферма має місце наступна теорема: існує нескінченне число натуральних n, для яких кожне з чисел n і $n + 1$ має тільки один простий дільник. Доведення випливає з твердження [12, с.23]: теорема, за якою існує нескінченне число натуральних n, для яких кожне з чисел n і $n + 1$ має тільки один простий дільник рівносильна теоремі про те, що існує нескінченне число простих чисел Мерсенна, або нескінченне число простих чисел Ферма.

1. Эдвардс Г. Генетическое введение в алгебраическую теорию чисел. – М.: Мир, 1980. – 484с.
2.Трост Э. Простые числа. – М.: Госиздат. физ.-мат. л-ры, 1959. – 135 с.

Ю.Рашкевич, Д.Пелешко, М.Пасєка
Національний університет "Львівська політехніка"

УДК 621.372

ОПТИМІЗАЦІЯ ПРОЦЕСУ ПОШУКУ ІНФОРМАЦІЇ В БАЗАХ ДАНИХ СИСТЕМ УПРАВЛІННЯ НАВЧАННЯМ

Пропонується метод прискорення пошуку символьних рядків в системах зберігання даних, побудований на основі представлення слова чи фрази у вигляді дискретного сигналу.

There are proposed method of acceleration of large string search in database of learning management system.

Вступ

Сучасний стан економіки спричинив появу нових та розвиток існуючих різноманітних форм навчання, таких, наприклад, як корпоративне, дистанційне, електронне та ін. З іншого боку, стрімкий розвиток новітніх інформаційних технологій (ІТ) призвів до широкого їх використання практично в усіх (традиційних та нетрадиційних) формах навчання. Більше того, почали з'являтись і розвиватись наукові концепції використання ІТ в навчальному процесі. В світлі цього з'явилась і активно впроваджується в практичне використання концепція систем управління навчанням. У загальному випадку її основними завданнями є спрощення навчального процесу як з боку викладача, так і з боку студента, а також спрощення управління самим навчальним процесом [3, 4]. Залежно від форми навчання основна направленість системи може зміщуватись у бік вирішення одного чи другого завдання.

Опис проблеми

Розвиток концепції системи управління навчанням в сукупності з розвитком комп'ютерної техніки призвів до появи теорії архітектур таких систем. Так, зокрема, системи, які володіють достатньо широкими можливостями та використовують технології комп'ютерних мереж, будуються за клієнт-серверною технологією. Типова схема наведена на рис.1 [4].

Як видно з рис.1, основою таких систем є система управління базою даних. У більшості випадків база даних (БД) системи містить навчальну інформацію, інформацію про учасників навчального процесу, нормативні документи тощо [3, 4].

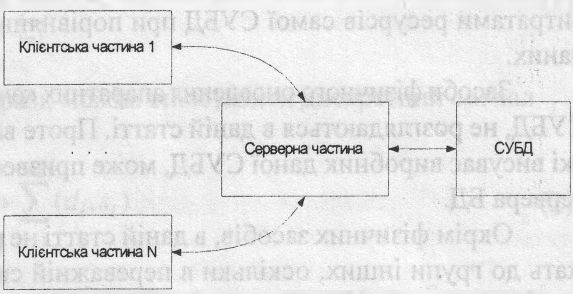

Рис. 1. Типова схема системи уравління навчанням

У міру використання системи розміри бази даних значно зростають, і виникає проблема пришвидшення вибірки інформації із сховища даних. Ця проблема є особливо актуальною у випадку використання розподілених та віддалених СУБД. Оскільки якість транспортних засобів передавання достаньо великих обє'мів інформації є низькою, а її покращення вимагає значних коштів, то вирішення проблеми полягає в оптимізації роботи з БД.

Сучасний стан питання та характеристика об'єкта дослідження

Сьогодні існує достатньо багато засобів оптимізації процесу вибірки чи пошуку даних із СУБД. Усіх їх умовно можна поділити на:

- засоби самої СУБД, тобто алгоритми вибірки чи пошуку даних, які вбудовані у саму СУБД;
- оптимізаційні засоби [5], тобто попередня логічна, оптимізаційна обробка складних запитів до СУБД;

- апаратні засоби, які передбачають фізичне оновлення (Upgrade) пристроїв, на яких працює СУБД;
- логічні засоби, які умовно також можна розбити на такі дві категорії: засоби першої категорії, передбачають внесення в СУБД додаткової інформації, яка забезпечує пришвидшення пошуку даних; засоби другої категорії грунтуються на методах передбачення даних;
- інші засоби, зокрема такі, які не входять до жодної з описаних груп. Як приклад можна навести логічну оптимізацію диску, на якому зберігається БД.

Оскільки існуючі системи управління навчанням використовують стандарні комерційні реляційні СУБД, то практично не існує жодної можливості розглядати оптимізаційні алгоритми самої СУБД.

Стосовно групи оптимізаційних засобів варто зауважити, що вони є достатньо повно досліджені і описані в науковій літературі [5]. Показано, що не існує єдиного ефективного механізму покращення *SQL*-запиту, який б забезпечував найкращу (з точки зору швидкості) вибірку даних в середовищі будь-якої СУБД. Для вирішення цього завдання треба проводити цілий набір оптимізаційних дій (логічна оптимізації, семантична тощо).

Проведені в сукупності ці дії дають прийнятний результат лише у випадку баз даних великого обсягу. Але при цьому поля в таблицях БД повинні бути або числовими, або символьними короткої довжини.

У випадку, коли поля таблиці бази є довгими рядками, або бінарними наборами даних, ефективність логічних засобів суттєво понижується. Це зумовлено значними витратами ресурсів самої СУБД при порівнянні великих масивів одного екземпляру даних.

Засоби фізичного оновлення апаратних компонентів комп'ютера, на якому працює СУБД, не розглядаються в даній статті. Проте варто додати, що не дотримання вимог, які висуває виробник даної СУБД, може призвести до суттєвого сповільнення роботи сервера БД.

Окрім фізичних засобів, в даній статті не розглядаються також засоби, які належать до групи інших, оскільки в переважній своїй більшості вони залежать від персоналу, який адмініструє дану СУБД.

Логічні засоби на відміну від оптимізаційних виявляють свою ефектиність у випадку, коли поля таблиць є великі за розміром. Так, зокрема, прогностичні засоби логічної групи з достатньою імовірністю дозволяють прогнозувати ті дані, які будуть потрібні в деякий момент часу. Проте варто зауважити, що ці засоби меньшою мірою використовуються для пришвидшення пошуку. Значно більшою мірою вони використовуються при вирішення завдань розпізнавання логічного змісту, який закладається у дані.

Оскільки таблиці бази даних систем управління навчанням можуть містити змішані набори даних, то найбільш ефективним є використання цілого комплексу оптимізаційних засобів: від оптимізаційних до інших.

Предметом розгляду даної статті є розробка ефективного логічного засобу першої категорії, який у великих за об'ємом БД давав б можливість пришвидшити пошук та вибірку інформації у випадку існування полів великої довжини.

Характеристика методу

Пошук інформації є найпростішою задачею теорії алгоритмів і може бути представлений як пошук інформації, яка збережена з конкретним індентифікатором. В загальному випадку вважається, що існує набір записів і завдання полягає в знаходженні кожного з них. Припускається, що кожен запис має спеціальне поле, яке називається ключем, атрибутом чи характеристикою запису. Це поле повинно однозначно (або з деякою похибкою у випадку розпізнавання) ідентифікувати запис, але при цьому мати значно меньший розмір. Полів, які задіюються в однозначному визначенні запису, може бути декілька (випадок поділу ключів на первинні та вторинні і т.д.). Але тоді однозначна індентифікація досягається комбінацією цих полів. Тобто, у загальному випадку вважається, що в кожному записі міститься декілька атрибутів і необхідно віднайти усі записи з деякими значеннями цих атрибутів.

На основі цього завдання пошуку запису зводиться до пошуку за ключем чи характеристикою, які за довжиною є значно меншими ніж сам запис.

Метод прискореного пошуку

Будь-яке слово чи фраза, яку можна розглядати як набір слів, надалі S, складається із символів (s_i), кожен з який має свій код (d_i) в таблиці *UNICODE*. Це означає, що кожному S, яке подається у вигляді суми символів і може бути представлене як

$$S \to \sum_{i=1}^{m} s_i \,, \tag{1}$$

тут m - кількість символів у слові чи фразі, можна співставити дискретний сигнал

$$S \to \sum_{i=1}^{m} (d_i, s_i) \,. \tag{2}$$

Якщо додатково розглядати ще й імовірность появи символа, то S подаємо у такому вигляді

$$S \to \sum_{i=1}^{m} (p_i, n_i) \,, \tag{3}$$

де n_i - порядковий номер симола у слові; p_i - імовірність появи символа, яка є наперед визначеною, статистичною характеристикою кожного символа.

Тобто слово стає функцією від коду символа та імовірності появи даного символа

$$S \to f(p_i, n_i) \,, \tag{4}$$

Фунціональну залежність (4) можна посилити, якщо n_i-му поставити у відповідність код символу, тобто n_i розглядати як лінійну функцію від коду

$$n_i = n_i(d_i), \tag{5}$$

На основі (4) розглянемо посимвольну інформативність у вигляді

$$P' = \int_l pdl, \tag{6}$$

де $l = l(n_i)$ - траєкторія слова чи фрази. Беручи до уваги (6), введемо першу ознаку - *питому посимвольну інформативність* слова чи фрази, яка буде визначатись за формулою

$$P = \frac{P'}{L} = \frac{1}{L}\int_l pdl, \tag{7}$$

де $L = L(l)$ - довжина S.

Оскільки, окрім p_i, існує ще одна статистична характеристика, а саме g_i - імовірність появи окремих складів (в даному випадку двосимвольних), то подібно до (6) та (7) можна отримати другу характеристику для S

$$G' = \int_{l_g} gdl_g; \tag{8}$$

$$G = \frac{G'}{L} = \frac{1}{L}\int_{l_g} gdl_g, \tag{9}$$

де $l_g = l_g(n_i)$ - траєкторія слова відносно g_i. Характеристику G назвемо *питомою сполучною характеристикою S.*

Таким чином кожному слову чи фразі відповідає характеристична пара

$$S \to (P, G), \tag{10}$$

де P, G, які є цілимим додатніми числами, повинні зберігатись в базі поряд з кожною фразою. Власне на основі них і буде прийматись рішення про збіг шуканої фрази з поточною в базі. Рішення про збіг пропонується приймати на основі введеної міри подібності

$$\mu(S_1; S_2) = \mu\big((P_1, G_1); (P_2, G_2)\big) < \varepsilon, \tag{11}$$

де ε - точність збігу. Міру (11) можна розглядати як набір мір, тобто

$$\mu\big((P_1, G_1); (P_2, G_2)\big) = \begin{cases} \mu(P_1; P_2) < \varepsilon_P; \\ \mu(G_1; G_2) < \varepsilon_G, \end{cases} \tag{12}$$

де ε_P, ε_G - є мірами збігу за окремими характеристиками P і G. У загальному випадку вони можуть бути рівними. Кожна з мір в (12) є звичайним модулем, тобто

$$\mu(P_1;P_2)=|P_1-P_2|<\varepsilon_P; \qquad \mu(G_1;G_2)=|G_1-G_2|<\varepsilon_G. \tag{13}$$

Приклад практичної реалізації

З метою апробації запропонованого алгоритму було попередньо проведено статистичний аналіз частоти зустрічання українських символів та пар українських символів. Розглядався текст довжиною 289387 символів. На основі цього тексту отримано частоти зустрічання українських літер (рис.2).

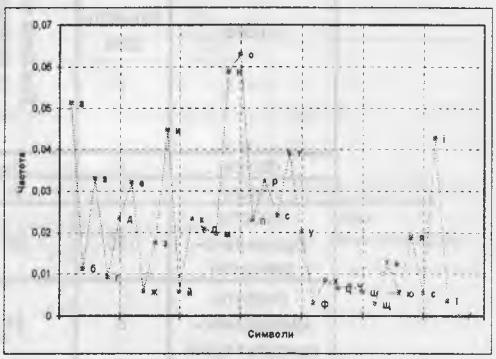

Рис.2. Частота зустрічання українських символів (вибірка - 289387 символів)

Для пар українських символів була зроблена вибірка, в яку війшли пари, частота зустрічання яких перевищувала значення 0.003. Графік частоти пар українських символів, які найбільше зустрічаються, наведено на рис.3.

Для практичної реалізація запропонованого алгоритму на основі результатів (рис.2 і рис.3) була побудована база даних в СУБД *MySql*. Побудована база даних містила одну таблицю (розмір таблиці становив 12316 записів), яка складалась з чотирьох полів, а саме:

- *Code* - цілого (довжиною 11 символів);
- *Produce* - символьного (довжиною 75 символів);
- *Realized* - символьного (довжиною 50 символів);
- *FreqSymb* - цілого (довжиною 4 символи);
- *FreqPair* - цілого (довжиною 4 символи).

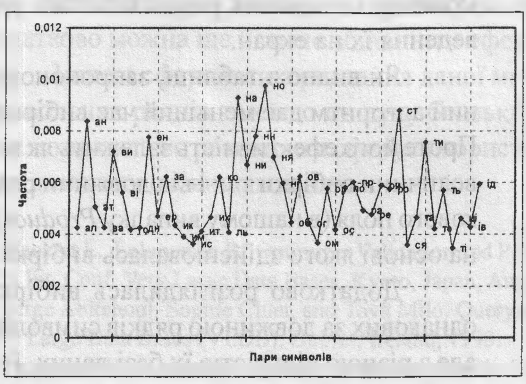

Рис.3. Частота найбільш вживаних пар українських символів (вибірка - 289387 символів)

З метою спрощення практичної реалізації вибірка проводилась по полю *Produce*. Інші поля (за винятком двох останніх) не були ключовими у вибірці.

Саме два останніх поля таблиці містили попередньо пораховані на основі запропонованої методики характристики P і G. Дані характеристики, які є додатними дійсними числами, були приведені до цілого числа з проміжку [0; 9999] шляхом їх помноження на додатковий ваговий коефіцієнт рівний 10000. Тобто в базу даних як характеристики P і G фактично заносились такі характерисики

$$P = \text{intr}(kP); \qquad G = \text{intr}(kG),$$

де $k = 10000$ - додатково введений ваговий коефіцієнт.

Це зроблено з метою пришвидшення пошуку, який є найбільш швидкий на цілих числах. Кількість розрядів у ваговому коефіцієнті вибрано не випадково. Для даної бази вже при такому ваговому коефіцієнті було досягнуто 100% правильності вибірки,

Результати вибірки символьних рядків з різною довжиною з бази даних, яка містить 12316 записів

Слово	Кількість слів	Довжина слова (в символах)	Час доступу, мс		Характеристики	
			Неоптимізований доступ	Оптимізований доступ	посимвольна	сполучна
факси	2	5	40	30	2923	0
горіхи	1	6	81	50	3334	165
роботи будівельно-ремонтні	125	26	150	140	3524	179
послуги еміграційно-консультаційні	5	34	50	40	3050	161

тобто вже забезпечувалась абсолютна точність вибірки. При меньшому ваговому коефіцієнті доведеться задіювати повторну вибірку, вже на меньшій кількості записів. Останній випадок поки що не досліджувався.

У випадку вибірки без врахування коефіцінтів P і G поля *FreqSymb* і *FreqPair* не вибирались.

Результи вибірки наведені на рис.4 і у таблиці. Час вибірки замірявся від початку посилання запиту на вибірку (зв'язок з базою був попередньо встановлений) і до моменту отримання результатів без виведення їх на екран.

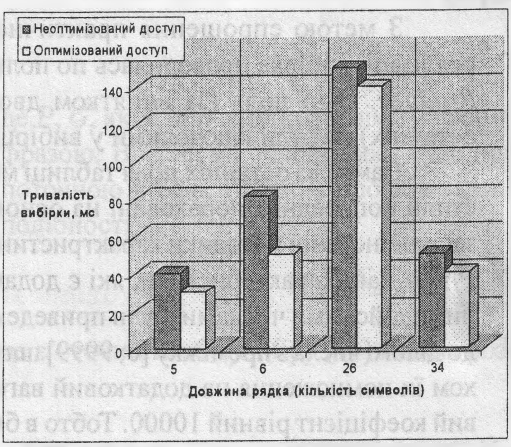

Рис.4. Тривалість вибірки рядків з різною довжиною

Як видно з таблиці, запропонований алгоритм дає меньший час вибірки. Проте його ефективність залежить як від величини вибірки так і від довжини рядкового поля (в нашому випадку *Produce*), на основі якого здійснювалась вибірка.

Додатково розглядалась вибірка однакових за довжиною рядків символів, але з різною кількістю їх базі даних. Результати вибірки наведені на рис.5. Як видно з рисунка, ефективність алгритму

суттєво залежить від обсягу вибірки і починає суттєво зростати при збільшенні вибірки. У протилежному випадку фіксовано зростає пришвидшення вибірки, але без прогресивного росту.

Висновки

Як показали дослідження запропонований в роботі алгоритм пришвидшення пошуку є ефективним при його використанні з великими базами даних (понад 10 000 записів). У меньших за обсягом базах він працює на рівні звичайної вибірки з семантично та логічно оптимізованим запитом. Окрім цього, ефективність алгоритму суттєво залежить від довжини символьного рядка та від обсягу вибірки. А тому в середніх базах (від 1000 до 10000 записів) при вибірці рядків, довжина яких є меншою 5 символів, також можна використовувати звичайні способи вибору інформації. Іншим недоліком алгоритму є експериментальне підбирання мінімальної розрядності вагового коефіцієнта в тому випадку, коли необхідно забезпечити 100-відсотково правильність вибраних результатів. Зменшення величини розрядності вагового коефіцінта може суттєво зменшити тривалість вибірки, особливо у реляційних та розподілених базах даних.

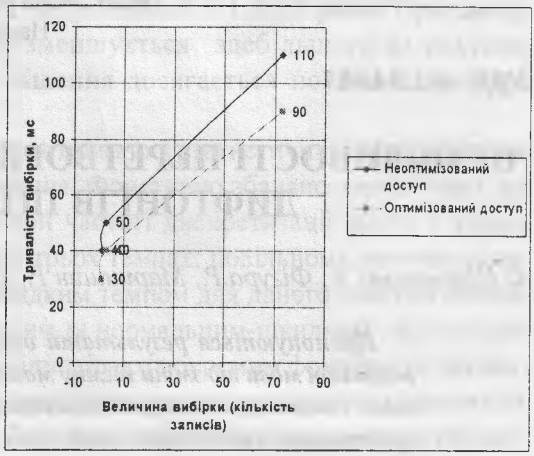

Рис.5. Результати вибірки слів довжиною 25 символів

Запропонована методика покращання часу вибірки є ефективною у великих і надвеликих базах даних, а також у базах з достатньо складними логічними зв'язками. Окрім цього, ефективність алгоритму вже починає виявлятись при роботі з символьними рядками з достатньо великою довжиною та при великих обсягах вибірки у середніх та малих за розмірами БД. У випадку, коли введені характеристики треба будувати для цілого запису, а не для окремого поля, як у наведеному прикладі, то цим додатково можна ще більше збільшити ефективність алгоритму.

Іншою сферою використання даної методики є внесення в неї елементів аналізу символьних рядків з метою пошуку близьких за змістом чи споріднених рядків. А це вже може суттєво розширити сферу використання даної методики, особливо при роботі у надвеликих сховищах даних.

1. Mackert L., Lohman G. R* Optimizer Validation and Performance Evaluation for Distrib-uted Queries // Proc. 12th Int. Conf. Very Large Data Bases, Kyoto, Japan, Aug. 1986. Los Altos, Calif., 1986.- С. 149-159
2. Serge Abiteboul, Sophie Cluet, and Tova Milo. Querying and updating the file. In Proc. of the Int. Conf. on Very Large Data Bases (VLDB), Dublin, Ireland, 1993.
3. Рашкевич Ю., Пелешко Д., Пасєка М., Стецюк А. Структурний аналіз систем управління навчанням// Вестник Херсонского государственного технического университета. – Херсон, 2002. – № 1(14).– С.464-470.
4. Рашкевич Ю., Пелешко Д., Пасєка М., Стецюк А. Проектирование WEB-ориентированных распре-

деленных учебных систем// Управляющие системы и машины. – К.:2002. – 3/4. – С.72-79.
5. Peleshko D., Pasyeka M. SQL-queries optimization. MS'2001 International Conference on Modeling &Simulation Proccedings, Lviv, Ukraine, 2001. – P. 184-186.

З. Шиманські*, Р.Фігура*, Р.Марцишин**
*Вища школа підприємництва та управління (Лодзь, Польща),
**Національний університет "Львівська Політехніка"

УДК 681.84.087

ОСОБЛИВОСТІ ПЕРЕТВОРЕННЯ ЧАСОВОЇ СТРУКТУРИ ДИФТОНГІВ ПОЛЬСЬКОЇ МОВИ

Пропонуються результати дослідження залежностей довжин дифтонгів польської мови від зміни темпу мовлення. Визначено структуру дифтонгів, отримано статистичні характеристики дифтонгів та їх структурних частин, обґрунтовано необхідність побудови функцій темпорального перетворення для зміни темпу мовлення.

In this paper offered Polish language diphthongs lengths dependence on the change of speech tempo research results. The diphthongs structure is here defined, diphthongs statistical description and there structural parts is obtained, a necessity of temporal transformation function production for the change of speech tempo is substantiated.

Вступ

Перетворення часового масштабу мовних сигналів відіграє важливу роль як в процесах розпізнавання та синтезу мови, так і в системах кодування та передачі мови каналами зв'язку, навчанні, системах мовної пошти тощо [1,2,3,4].

Процес перетворення часового масштабу має свої особливості для різних мов, які визначаються в першу чергу фонетикою звуків. Це викликає необхідність при перетворенні часового масштабу мови враховувати не тільки закономірності для основних класів звуків, але й фонетичні особливості кожного звуку. Через складність структури особливий інтерес в задачах аналізу та перетворення мовних сигналів викликають дифтонги, які звучать по-різному в різних мовах. Для польської мови такими дифтонгами є ą (wąs) та ę (ręki), які не мають аналогів в інших мовах [6]. На сьогодні відсутня інформація про дослідження структури та особливості перетворення дифтонгів в задачах нормалізації та зміни часового масштабу мовних сигналів. Тому метою даної