

УДК 658.012

Б.І. Мороз, Л.В. Кабак, О.В. Трофімов

МЕТОДИ Й АЛГОРИТМИ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ ЗБЕРІГАННЯ ОБ'ЄКТІВ БАЗ ДАНИХ

Введення

Нині відбувається стрімкий розвиток засобів обчислювальної техніки, підвищується доступність інформації, обсяг і швидкість її обробки стають вирішальними факторами розвитку й удосконалення митної служби України. Єдина автоматизована інформаційна система митної служби України являє собою програмний комплекс, функції якого є складовою в підтримці надійного зберігання інформації в базах даних, виконанні специфічних для митної служби додатків, перетворення інформації за допомогою обчислень, виконаних над великими масивами даних.[1] Обсяг інформації, з яким доводиться мати справу таким системам, дуже великий, а сама інформація має досить складну структуру.

Основним засобом, що застосовується в ЄАІС для реалізації централізованого керування даними, які зберігаються в базах даних, а також для забезпечення доступу до них і підтримки їх у стані, що відповідає стану предметної області, в ЄАІС використовується система керування базами даних (СКБД) Oracle.[2]

1 Постановка завдання

СКБД Oracle є реляційною базою даних (БД).

Таблиця - це основна одиниця пам'яті даних у базі даних. При створенні таблиць визначаються параметри, які істотно впливають на швидкість роботи СКБД. Одним з найважливіших таких параметрів є швидкість виконання операції DML (data manipulation language).[2] Для керування пам'яттю при створенні таблиці використовуються наступні параметри:

PCTFREE - розмір у відсотках вільної пам'яті в блоках;

PCTUSED - розмір у відсотках використовуваної пам'яті в блоках;

INITIAL - розмір, у байтах, першого екстента, що розподіляється при створенні сегмента;

© Мороз Б.І., Кабак Л.В., Трофімов О.В., 2009

NEXT - розмір, у байтах, наступного інкрементального екстента, що

розподіляється для сегмента;

MAXEXTENTS - загальне число екстентів, включаючи початковий, котре може бути розподілене для сегмента;

MINEXTENTS - загальне число екстентів, що повинне бути розподілене для сегмента при його створенні. [3]

Для прискорення роботи БД необхідно розробити метод знаходження оптимальних значень цих параметрів при створенні таблиць, а також можливості корекції заданих параметрів на основі даних, одержуваних на основі статистичної обробки інформації про DML-операції, виконаних над даною таблицею.

2 Розробка методів для оптимізації параметрів пам'яті таблиць

Перед тим як почати вирішення поставленого завдання, розглянемо, який вплив на роботу БД робить кожний з наведених параметрів. Параметр PCTFREE при створенні таблиці за замовчуванням встановлюється рівним 10% і може приймати будь-яке значення від 0% до 99 %.

Низьке значення PCTFREE має наступні позитивні ефекти: резервується менше місця для відновлень існуючих рядків таблиці, є можливість більш повно заповнювати блок вставками, заощаджується пам'ять, тому що всі дані таблиці або індексу зберігаються в меншій кількості блоків (більше рядків на один блок). Але при низькому значенні PCTFREE є й негативні ефекти: необхідність часткової реорганізації блоків у міру заповнення їх новими або оновленими даними, що збільшує вартість обробки й необхідну пам'ять, оскільки відновлення рядків або записів індексу приводять до росту рядків і переповненню пам'яті, виділеної в блоках даних під рядки.

Відмітимо, що зазначені фактори є несуттєвими для рідко змінюваних сегментів даних, тому для таких таблиць доцільно використовувати низьке значення PCTFREE.

Високе значення PCTFREE має наступні ефекти: резервується більше місця для відновлень існуючих рядків таблиці; може знадобитися більше пам'яті для тієї ж самої кількості вставлених даних (вставляє менше рядків на один блок); зменшується вартість обробки, тому що блоки рідко вимагають реорганізації; поліпшується продуктивність відновлень. Високе значення PCTFREE може бути

прийнятним, наприклад, для сегментів, що піддаються частим відновленням. Однак високе значення PCTFREE істотно збільшує обсяг пам'яті, що таблиця займає на диску.

При визначенні PCTFREE необхідно враховувати тип даних стовпця таблиці або індексу. Для даних змінної довжини (наприклад, VARCHAR) відновлення можуть приводити до росту рядків, оскільки нові значення можуть мати розмір, що перевищує розмір замінюваних ними даних. Якщо має місце багато відновлень, при яких розмір даних збільшується, то PCTFREE варто збільшити; якщо відновлення істотно не впливають на розміри рядків, PCTFREE може бути низьким. [3]

Розроблювальний метод дозволяє знайти задовільний компроміс між щільним упакуванням даних і гарною продуктивністю відновлень.

Тепер розглянемо параметр PCTUSED. Коли пам'ять у блоці даних виявляється заповненою на PCTUSED, у цей блок не вставляються нові рядки. ORACLE намагається утримати блок даних заповненим принаймні на PCTUSED. Це відсоток пам'яті в блоці, вільної для даних після відрахування накладних витрат із загальної пам'яті блоку.

Умовчання для PCTUSED дорівнює 80 відсотків, однак можна задавати будь-яке ціле значення від 0 до 99 включно. Важливо лише, щоб сума PCTFREE і PCTUSED не перевищувала 100. Низьке значення PCTUSED має наступні ефекти: у середньому, блоки утримуються менш заповненими, чим при високому відсотку PCTUSED; зменшується вартість обробки, що вимагається при виконанні операцій UPDATE і DELETE для переміщення блоку у вільний список, коли зайнята ним пам'ять падає нижче PCTUSED; збільшується не використувана пам'ять у базі даних. Високе значення PCTUSED має наступні ефекти: у середньому, блоки утримуються більше заповненими, ніж при низькому значенні PCTUSED; поліпшується ефективність використання пам'яті; збільшується вартість обробки, що вимагається при виконанні операцій UPDATE і DELETE. [3]

Значення параметрів INITIAL, NEXT, MAXEXTENTS MINEXTENTS відповідають за резервування місця на носії інформації й істотно на продуктивність не впливають. [3 - 6]

Метод 1. Метод вибору значень PCTUSED і PCTFREE

Позначимо значення PCTUSED як U а значення PCTFREE як F .

Обмежимо значення PCTUSED і PCTFREE.

$$U+F \leq 100 \quad (1)$$

Однак якщо відповідно до формули (1) сума дорівнює 100, то блоки даних виявляться недозаповненими приблизно на величину середнього рядка, тому з метою економії пам'яті варто покласти

$$U+F < 100 \quad (2)$$

Тоді для досягнення ідеального компромісу між використовуваною пам'яттю й продуктивністю сума повинна бути менше 100% на величину γ яка дорівнює відсотку пам'яті у вільному блоці, що займає середній рядок, і сума PCTUSED і PCTFREE розраховується відповідно до формули

$$U+F = 100 - \gamma \quad (3)$$

Позначимо розмір блоку БД у байтах як S_b , а розмір середнього рядка в байтах як \bar{R} . Оскільки, як відомо, накладні витрати в блоці становлять приблизно 100 байт, то величину γ необхідно розраховувати по формулі

$$\gamma = \frac{\bar{R}}{S_b - 100} \times 100 \quad (4)$$

При цьому спостерігається наступний ефект: чим менше різниця між 100 і сумою PCTUSED і PCTFREE, тим вище утилізація пам'яті за рахунок деякого підвищення вартості обробки. Виходячи з формул (3) і (4)

$$U + F = 100 - \frac{\bar{R}}{S_b - 100} \times 100 \quad (5)$$

З формули (5) ми одержимо оптимальне значення для суми PCTUSED і PCTFREE. Тепер необхідно визначитися з їхніми конкретними значеннями. Для визначення значень PCTUSED і PCTFREE розглянемо типові сценарії роботи БД.

Сценарій 1.

Виходячи з аналізу статистичних даних, ми одержали інформацію, що дії, вироблені над певною таблицею, в основному включають пропозиції UPDATE які змінюють розмір рядків на λ відсотків

$$\lambda = \frac{R_{\max} - R_{\min}}{\bar{R}} \times 100 \quad (6)$$

У формулі (6) \bar{R} - розмір середнього рядка в байтах.

Тоді необхідне значення PCTFREE визначити як

$$F = \lambda + \mu \quad (7)$$

де μ - коефіцієнт "запасу" пам'яті.

Виходячи з вимог пропонованих до СКБД, щоб було потрібно менше часу на обробку запитів при високій активності відновлень, тобто для поліпшення продуктивності, необхідно вибирати параметр μ приблизно рівним 4-5%.

Сценарій 2.

Типова робота включає пропозиції INSERT і DELETE, а пропозиції UPDATE у середньому не збільшують розміри рядків. У цьому випадку параметр PCTFREE встановлюємо в розмірі $F \approx 5\%$, щоб було потрібно менше часу на обробку запитів при високій їх активності, тобто для поліпшення продуктивності.

Сценарій 3.

Таблиця дуже велика; тому основною проблемою є пам'ять. Типова робота включає тільки читаючі транзакції - запити типу SELECT. При більших обсягах таблиці найбільш важливим параметром є місце на диску. Тому в цьому випадку значення параметра PCTFREE необхідно встановити рівним $F \approx 5\%$, а значення PCTUSED встановити максимально можливим.

Дослідження методу

Для дослідження розробленого методу використаємо базу даних телефонів яка має 240 000 записів. Поля в яких зберігаються дані прізвище ім'я та по батькові а також назва вулиці приймаємо як тип даних Varchar2(200). Тип даних Varchar2 був обраний з тією метою щоб при зміні даних у стовпцях таблиці мінялася довжина рядка, що приведе до реорганізації блоків у міру заповнення їхньої вільної пам'яті новими або оновленими даними. Для тестування використаємо дві таблиці першу з параметрами встановлюваними за замовчуванням (PCTUSED=90 і PCTFREE=5) при створенні таблиці другу з параметрами певними по розробленому методі (PCTUSED = 70 і PCTFREE=25). Кожну таблицю поміщаємо у два різних табличних простори, які перебувають на тому самому носії для чистоти проведення експерименту. Після цього оцінюємо обсяг, що займають

таблиці в табличному просторі так перша таблиця займає 26 Мбайт а друга 31 Мбайт.

Тестування операцій вибірки показало, що швидкість вибірки практично однакова. Як ми вже відзначали, що установки параметрів PCTUSED і PCTFREE істотно не впливають на швидкість операцій вибірки. Для тестування операцій UPDATE використаємо процедуру, що збільшує довжину імені й адреса в 2 рази. Як і передбачалася операція UPDATE на другій таблиці зайняла в 2,5 рази менше часу, ніж у першій таблиці так, як у випадку з першою таблицею перерозміщення рядків між блоками відбувалося набагато частіше. Однак як показує досвід у БД таке глобальне збільшення довжини рядків відбувається дуже рідко й платою за швидкість є розміри таблиці, тому при виборі параметрів PCTUSED і PCTFREE необхідно визначитися із частотою виконання операцій UPDATE, а також з розмірами на скільки вони збільшують розміри рядків і чи приводить дана процедура до реорганізації пам'яті блоків. Для оптимізації вибору параметрів PCTUSED і PCTFREE розроблений наступний метод.

Метод 2. Метод автоматичного коректування параметрів PCTUSED і PCTFREE

Метод ефективний якщо заздалегідь невідомий сценарій роботи БД із певною таблицею.

Крок 1. По формулі (5) визначаємо сумарне значення PCTUSED і PCTFREE. Створюємо таблицю з мінімальним параметром PCTFREE рівним 5% а відповідно PCTUSED - 90%.

Крок 2. Для відстеження операцій INSERT та UPDATE, проведених над певною таблицею, створюємо допоміжну таблицю, у яку будемо заносити інформацію про номер блоку й нову довжину рядка у цьому блоці. Створюємо тригери, які будуть вносити в цю таблицю інформацію.

```
create or replace trigger tr1 before update on demo /* назва таблиці*/
```

```
for each row
```

```
declare
```

```
v_oldsize number(6,2);
```

```
V_newsize number(6,2);
```

```
V_delta number(6,2);
```

```

begin
  select (vsize(:old.name)+VSIZE(:old.adres)+VSIZE(:old.name_CITI))
into v_oldsize from dual; /*визначаємо розмір рядка до операції
update */
  select(VSIZE(:new.name)+VSIZE(:new.adres)+VSIZE(:New.name_C
ITI)) into V_newsize from dual; /* визначаємо розмір рядка після
операції update */
  V_delta:=(V_newsize- v_oldsize)/ v_oldsize*100; /* визначаємо
відсоткове відхилення розміру рядка*/
  insert into demo_aud (delta,row_st) values (V_delta,
DBMS_ROWID.ROWID_BLOCK_NUMBER(:old.rowid));
  /* заносимо в таблицю відсоткове відхилення і номер блока */
end tr1;

```

Крок 3. Створюємо процедуру, що через заданий адміністратором проміжок часу буде оцінювати середнє збільшення розміру рядків, що привело до реорганізації блоків. Приймемо гіпотезу про нормальний розподіл довжини рядка, що додається (або змінюється).

Обчислюємо оцінки математичного сподівання довжини рядка та його дисперсії за формулами (8) та (9)

$$\bar{l} = \frac{1}{n} \sum_{i=1}^n l_i \quad (8)$$

$$S = \sqrt{\frac{\sum_{i=1}^n (\bar{l} - l_i)^2}{n - 2}}, \quad (9)$$

де n – кількість рядків (нових або змінених), l_i , $i=1, \dots, n$ – довжина кожного з таких рядків.

Задаючись певним рівнем ймовірності α , обчислюємо мінімальне значення l , що задовольняє нерівності за формулою (10)

$$\int_1^{\infty} \exp\left\{-\frac{(x - \bar{l})^2}{2\sigma^2}\right\} dx \leq \alpha \quad (10)$$

Реорганізація буде відбуватися у випадку, коли отримане значення l буде більше аналогічного значення, обчисленого під час останнього застосування алгоритму.

Крок 4. Використовуючи динамічний PL/SQL коректуємо параметри таблиці.

Висновки

Наукова новизна розробленого методу полягає в тому, що вперше розроблений метод, який дозволяє автоматично оптимальним чином визначити параметри зберігання таблиць, а саме PCTUSED і PCTFREE, у результаті чого істотно підвищилася продуктивність сервера, а також знизити обсяг пам'яті, займаної таблицями на носіях інформації.

Отримані результати можуть бути впроваджені як елемент ЄАІС митної служби України, що дозволить істотно підвищити якість функціонування сервера БД системи ЄАІС митної служби України завдяки розвитку існуючих і розробці нових методів оптимізації параметрів пам'яті таблиць, що забезпечує скорочення часу й зменшення обсягів пам'яті займаної таблицями БД сервера інформаційно-пошукової системи.

ЛІТЕРАТУРА

1. Копосов С.А. Розвиток Єдиної автоматизованої системи (ЄАІС) Митної служби України на сучасному етапі// Матеріали науково-методичної конференції "Актуальні проблеми підготовки фахівців для митної служби на сучасному етапі". - Дніпропетровськ, 1998. - С. 45-48.
2. Дейт К. Введення в системи баз даних. - Спб.: Видавничий будинок "Вільямс", 2000. - 846 с.
3. Oracle Corporation, Database Administrator's Guide. - Art. Managing Users and Resources, 2001, June. - P. 24-1-24-33.
4. Oracle Corporation, SQL Reference, 2001, June. P. 1-1-2-90.
5. Oracle Corporation, Servlet Engine Developer's Guide. - Art. Writing PL/SQL Servlets, 2001, July. - P. 9-1-9-14.
6. Oracle Corporation, Application Developer's Guide - Large Objects (LOBs), art. LOBs Case Studie. - P. 14-1-14-14.

Одержано 22.02.2009р.