

УДК 658.012

М.О. Дудко, канд. техн. наук

(Україна, Дніпропетровськ, Державний ВНЗ «Національний гірничий університет»)

О.В. Трофімов, канд. фіз.-мат. наук

(Україна, Дніпропетровськ, академія митної служби України)

Г.І. Сарічева, Ю.В. Петрова, І.В. Вернер

(Україна, Дніпропетровськ, Державний ВНЗ «Національний гірничий університет»)

МЕТОДИ Й АЛГОРИТМИ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ ЗБЕРІГАННЯ ОБ'ЄКТІВ БАЗ ДАНИХ

Введення

Гірнича промисловість потребує єдиної інформаційної бази, в якій би з метою зменшення кількості аварій аналізувалися дані з будівництва та реконструкції шахт. Обсяг інформації, з яким доведеться мати справу таким системам, дуже великий, тобто інформація буде мати досить складну структуру [1]. Основним засобом для реалізації централізованого керування даними, які зберігаються у базах, а також для забезпечення доступу до них і підтримки їх у стані, що відповідає предметній області, використовується система керування базами даних (СКБД) Oracle [2].

1. Постановка завдання

СКБД Oracle є реляційною базою даних (БД).

Таблиця – це основна одиниця пам'яті у базі даних. При створенні таблиць визначають параметри, які істотно впливають на швидкість роботи СКБД. Одним із таких найважливіших параметрів є швидкість виконання операції DML (data manipulation language) [2]. Для керування пам'яттю при створенні таблиці використовують такі 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;

NEXT – розмір у байтах наступного інкрементального екстента, що розподіляється для сегмента;

MAXEXTENTS – загальне число екстентів, включаючи початковий, яке може бути розподілене для сегмента;

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

Для прискорення роботи БД слід розробити метод знаходження оптимальних значень цих параметрів при створенні таблиць, а також розглянути можливості корекції заданих параметрів на основі даних, одержуваних під час статистичної обробки інформації про DML-операції.

2. Розробка методів для оптимізації параметрів пам'яті таблиць

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

Мале значення PCTFREE зумовлює такі позитивні ефекти: резервується менше місця для відновлень існуючих рядків таблиці, є можливість повніше заповнювати блок вставками, зберігається пам'ять, тому що всі дані таблиці або індекси зберігаються в меншій кількості блоків (більше рядків на один блок). Але при малому значенні PCTFREE існують й негативні ефекти: необхідність часткової реорганізації блоків у міру заповнення їх новими або оновленими даними, що потребує більше пам'яті й процесорного часу для виконання операцій обробки даних, оскільки відновлення рядків або записів індексу призводить до збільшення рядків і переповнення пам'яті, виділеної в блоках даних під рядки.

Відзначимо, що зазначені фактори є несуттєвими для мало змінюваних сегментів даних, тому для таких таблиць доцільно використовувати невелике значення PCTFREE.

Велике значення PCTFREE зумовлює такі ефекти: резервується більше місця для відновлень існуючих рядків таблиці; може знадобитися більше пам'яті для тієї самої кількості вставлених даних (вставляє менше рядків на один блок); витрачається менше часу на обробку даних, оскільки блоки майже не потребують реорганізації; поліпшується продуктивність відновлень. Велике значення параметра PCTFREE може бути прийнятним, наприклад, для сегментів, що піддаються частим відновленням. Однак велике значення PCTFREE істотно збільшує обсяг пам'яті, що таблиця займає на диску.

При визначенні параметра PCTFREE необхідно враховувати тип даних стовпця таблиці або індекс. Для даних змінної довжини (наприклад VARCHAR) відновлення можуть спричинити зростання рядків, оскільки нові значення можуть мати розмір, що перевищує розмір замінованих ними даних. Якщо

спостерігається багато відновлень, при яких розмір даних зростає, то параметр PCTFREE варто збільшити; якщо відновлення істотно не впливають на розміри рядків, то параметр PCTFREE може бути малим [3].

Розроблювальний метод дозволяє виявити задовільний компроміс між економічним розташуванням даних у оперативній пам'яті і високою продуктивністю відновлень.

Тепер розглянемо параметр PCTUSED. Коли пам'ять у блоці даних виявляється заповненою на величину PCTUSED, то у цей блок нові рядки не вставляються. Параметр ORACLE намагається утримати блок даних заповненим принаймні на величину PCTUSED. Це відсоток вільної пам'яті у блоці для даних (тут не враховується розмір службових даних, що зберігається у заголовку блока).

Умовчування для PCTUSED дорівнює 80 %, однак можна задавати будь-яке ціле значення від 0 до 99 % включно. Важливо лише тільки, щоб сума величин PCTFREE і PCTUSED не перевищувала 100 %. Мале значення PCTUSED зумовлює такі ефекти: у середньому блоки виявляються менш заповненими, ніж при значних відсотках PCTUSED; зменшується складність обробки, що потрібно під час виконання операцій UPDATE і DELETE для переміщення блока у пустий список, коли зайнята ним пам'ять стає нижче за PCTUSED; збільшується невикористовувана пам'ять у базі даних. Велике значення PCTUSED зумовлює такі ефекти: у середньому блоки виявляються більше заповненими, ніж при малому значенні PCTUSED; підвищується ефективність використання пам'яті; збільшується час обробки, потрібний для виконання 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]. Розглянемо декілька методів.

1. Метод вибору значень PCTUSED і PCTFREE

Позначимо параметри PCTUSED як U , PCTFREE як F .

Обмежимо значення PCTUSED і PCTFREE.

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

Однак, якщо згідно з формулою (1) сума дорівнює 100, то блоки даних виявляються недозаповненими приблизно на величину середнього рядка, тому з метою економії пам'яті варто покласти, щоб

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

Тоді для досягнення ідеального компромісу між використовуваною пам'яттю й продуктивністю сума має бути менше 100% від величини γ , яка дорівнює відсотку пам'яті у вільному блоці, що займає середній рядок. Тому сума параметрів PCTUSED і PCTFREE розраховується таким чином:

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

Позначимо розмір блока БД у байтах як S_b , а розмір середнього рядка як \bar{R} . Оскільки, як відомо, розмір службової інформації у заголовку блока становить приблизно 100 б, то величину γ слід обчислювати за формулою

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

При цьому спостерігається такий ефект: чим менше різниця між числом 100 і величиною суми PCTUSED і PCTFREE, тим вище утилізація пам'яті за рахунок деякого підвищення часу обробки. Виходячи з формул (3) і (4), запишемо

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

З виразу (5) одержимо оптимальне значення для суми PCTUSED і PCTFREE. Тепер треба визначитися з їхніми конкретними значеннями. Для цього розглянемо типові сценарії роботи БД.

Сценарій 1.

Виходячи з аналізу статистичних даних, ми одержали інформацію, що дії, виконані з величинами в таблиці, в основному включають пропозиції UPDATE, які змінюють розмір рядків на λ відсотків, тобто

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

де \bar{R} – розмір середнього рядка в байтах.

Тоді необхідне значення PCTFREE розраховується так:

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

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 слід вибирати параметр μ приблизно у межах 4 – 5%.

Сценарій 2.

Типова робота включає пропозиції INSERT і DELETE, а пропозиції UPDATE у середньому не збільшують розміри рядків. У цьому разі параметр PCTFREE приймаємо розміром $F \approx 5\%$, 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 запити типу SELECT, що читаються в транзакції. При більших обсягах таблиці найважливішим параметром є місце на диску. Тому в цьому разі значення параметра PCTFREE необхідно встановити розміром $F \approx 5\%$, а значення PCTUSED – максимальним.

Дослідження методу.

Для цього використаємо базу даних телефонів, яка має 240 000 записів. Поля, в яких зберігаються прізвище, ім'я та по батькові, а також назва вулиці, приймаємо як тип даних Varchar2(200). Такий тип 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, встановленими за замовчуванням (PCTUSED=90 і PCTFREE=5), другу з параметрами, що відповідають розробленому методу (PCTUSED = 70 і PCTFREE=25). Кожну таблицю поміщаємо у два різних простори, які перебувають на тому самому носії. Після цього оцінюємо обсяг, який займають таблиці в просторі. Так, перша таблиця займає 26, а друга 31 Мб.

Тестування операцій вибірки виявило, що швидкість вибірки практично однакова. Як ми вже відзначали, встановлені параметри PCTUSED і PCTFREE істотно не впливають на швидкість операцій вибірки. Для тестування операцій UPDATE використаємо процедуру, що збільшує величину імені й адреси в 2 рази. Як і передбачалося, операція UPDATE в другій таблиці зайняла в 2,5 рази менше часу, ніж у першій, оскільки у випадку з першою таблицею перерозміщення рядків між блоками відбувалося набагато частіше. Однак, як засвідчив досвід, таке глобальне збільшення довжини рядків відбувається дуже рідко, тому при виборі параметрів PCTUSED і PCTFREE необхідно визначитися з частістю 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.

2. Метод автоматичного коректування параметрів PCTUSED і PCTFREE

Метод ефективний, якщо заздалегідь невідомий сценарій роботи БД з певною таблицею.

Крок 1. За формулою (5) визначаємо сумарне значення PCTUSED і PCTFREE. Створюємо таблицю з мінімальним параметром PCTFREE, тобто 5% і відповідно PCTUSED – 90%.

Крок 2. Для відслідкування операцій INSERT та UPDATE, проведених над певною таблицею, створюємо допоміжну таблицю, в яку будемо заносити інформацію про номер блока 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_CITI)) 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, яка через заданий адміністратором проміжок часу буде оцінювати середнє збільшення розміру рядків, що привело до реорганізації блоків. Прийнемо гіпотезу, що довжина рядка, що збільшується або змінюється, є випадковою величиною з нормальним законом розподілу.

Обчислюємо статистичні оцінки для величин математичного сподівання довжини рядка та його дисперсії за такими формулами :

$$\bar{l} = \frac{1}{n} \sum_{i=1}^n l_i ; \tag{8}$$

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

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

Задаючи певний рівень імовірності α , знаходимо мінімальне значення l , що задовольняє нерівності

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

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

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

Висновки

Сучасні автоматизовані системи керування технологічними процесами (АСК ТП), що застосовуються у гірничодобувній промисловості, є системами реального часу і мають задовольняти досить жорстким критеріям надійності, швидкодійності, здатності до обробки великих масивів даних, що, в свою чергу, потребує ретельного підходу до вибору сервера СКБД та його відповідного налаштування.

Наукова новизна розробленого методу полягає в тому, що вперше був запропонований спосіб автоматичного визначення параметрів зберігання таблиць СКБД ORACLE, а саме PCTUSED і PCTFREE, у результаті чого істотно підвищилася продуктивність сервера, знизився обсяг пам'яті, що потрібно для зберігання таблиць бази даних на носіях інформації.

Отримані результати можуть бути впроваджені як елемент БД гірничої промисловості, а це дозволить істотно підвищити якість та надійність функціонування сервера БД існуючих систем, а також буде сприяти розробці нових методів оптимізації параметрів пам'яті таблиць, що забезпечить зменшення часу й обсягів пам'яті, займаної таблицями БД сервера АСК ТП.

Список літератури

1. Благодарный А.И. Программное обеспечение высоконадежной АСУ реального времени для предприятий горнодобывающей промышленности / А.И. Благодарный, Л.С. Каратышева, Г.П. Чейдо // Горн. пром. – 2009. – №2 – С.58 – 65.
2. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – М.: Вильямс, 2005. – 1328 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.

Рекомендовано до друку д-ром техн. наук, проф. Самусею В.І.