

ПЕЦКОВ Р.О., аспірант кафедри «Автоматизація та комп'ютерно-інтегровані технології транспорту» (Державний економіко-технологічного університет транспорту)

Вдосконалення програмних компонентів аналітичних моніторингових систем

Розглянуті питання вдосконалення програмних компонентів аналітичних моніторингових систем впровадженням збережених процедур, які є окремими програмними модулями, що об'єднують в собі декілька запитів і зберігаються в базі даних. Описуються питання розподілу навантаження на систему, перенесенням частини логічних операцій в збережені процедури, основні поняття і синтаксис, використовуваний для написання або виклику. Описується можливість одночасного звернення до процедури визначеної кількості користувачів з різних додатків і запобігання несанкціонованого втручання в базу даних, контролюючи доступ до інформації, що зберігається, відповідно до поставлених вимог адміністратором системи.

Ключові слова: моніторинг, трансформаторне обладнання, збережена процедура.

Постановка проблеми

Незважаючи на те, що швидкості передачі даних в останні кілька років сильно ростуть, передача даних залишається все ще повільною і ненадійною.

Таким чином, в деяких випадках для економії трафіку і збільшення надійності хорошим рішенням є зосередження обробки інформації на стороні сервера без передачі даних по мережі.

Аналіз останніх досліджень

На початку 2000 років в системах СКБД з'явилося кілька суттєвих нововведень: підтримка збережених процедур, функцій і тригерів. Вони успішно дозволяють перенести на сторону бази даних деяку частину виконуваних дій, тим самим у край спрощуючи логіку програми. Ці можливості можна використовувати і в продуктивних цілях.

В основному збережені процедури звикли застосовувати в бізнес-логіці, але використання частих запитів і великих об'ємів передачі даних започаткували їх використання у все більших масштабах. Збережені процедури, надають можливість розподілити навантаження на систему.

Формулювання цілей

- Провести аналіз використання збережених процедур,

- зробити опис загальної структури збережених процедур, що можуть застосовуватись в аналітичних моніторингових системах,

- синтезувати алгоритм взаємодії збереженої процедури з програмним забезпеченням.

Виклад основного матеріалу дослідження

Звернення до баз даних, що ідуть через мережу, можуть в деяких випадках призводити до зниження продуктивності. Збережені процедури допомагають поліпшити продуктивність прикладних програм і зменшити обсяг звернень до бази даних.

Збережена процедура (stored procedure) - це іменованій набір команд Transact-SQL, що зберігається безпосередньо на сервері і представляє собою самостійний об'єкт бази даних.

Оскільки збережені процедури виконують оператори SQL на сервері то продуктивність системи, в цілому, підвищується.

Але існує багато поглядів на їх використання. Розглянемо усі за та проти.

- ЗА

- Розподіл логіки з іншими додатками. Збережені процедури інкапсулюють функціональність; це забезпечує зв'язність доступу до даних і управління ними між різними додатками.

- Ізоляція користувачів від таблиць бази даних. Це дозволяє давати доступ до збережених процедур, але не до самих даних таблиць.

- Забезпечує механізм захисту. Відповідно до попереднього пункту, якщо ви можете отримати доступ до даних тільки через збережені процедури, ніхто інший не зможе стерти ваші дані через команду SQL DELETE.

- Покращення виконання, як наслідок скорочення мережевого трафіку. За допомогою збережених процедур безліч запитів можуть бути об'єднані.

- Використання збережених процедур реалізує принцип модульного проектування, так як процедури дозволяють розбивати великі завдання на самостійні, більш дрібні та зручні в управлінні частини.

- ПРОТИ

- Підвищення навантаження на сервер баз даних у зв'язку з тим, що більша частина роботи виконується на серверній частині, а менша - на клієнтській.
- Доведеться вивчити синтаксис PL/SQL для написання своїх збережених процедур.
- Неможлива міграція з однієї СУБД на іншу (DB2, SQL Server та ін.) Може призвести до проблем.

Збережені процедури відіграють ключову роль в підвищенні швидкодії при роботі в мережі з архітектурою «клієнт-сервер». На рис. 1 показано приклад виконання послідовності операторів SQL на клієнті та виконання тієї ж послідовності операторів

SQL, оформлених у вигляді збереженої процедури. В цьому випадку клієнт звертається до сервера тільки для виконання команди запуску збереженої процедури. Сама процедура, що зберігається виконується на сервері. Обсяг інформації що пересилається по мережі різко скорочується.

Крім того, збережені процедури допомагають зосередити всі робочі алгоритми в одному місці. Без збережених процедур користувачеві довелося б вводити весь набір команд усякий раз, коли він хоче виконати якусь дію, а в даному випадку ви змінюєте збережену процедуру і ці зміни відразу ж стають доступні для всіх клієнтських програм, що використовують її (рис. 2).

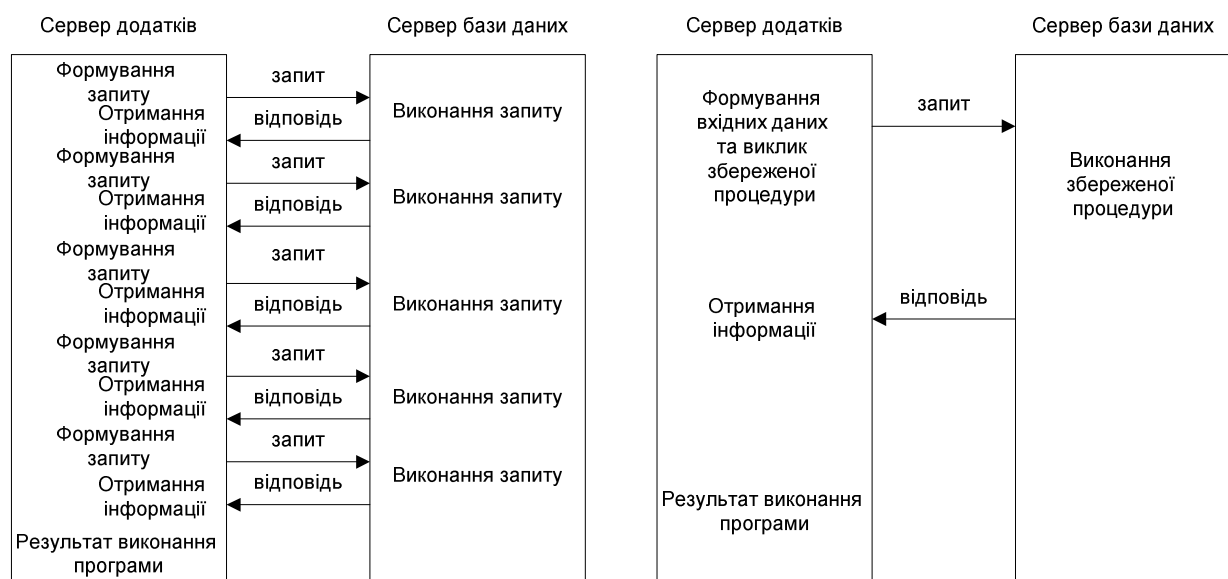


Рис. 1. Обмін інформацією між клієнтом та сервером бази даних в стандартному режимі та за допомогою збереженої процедури

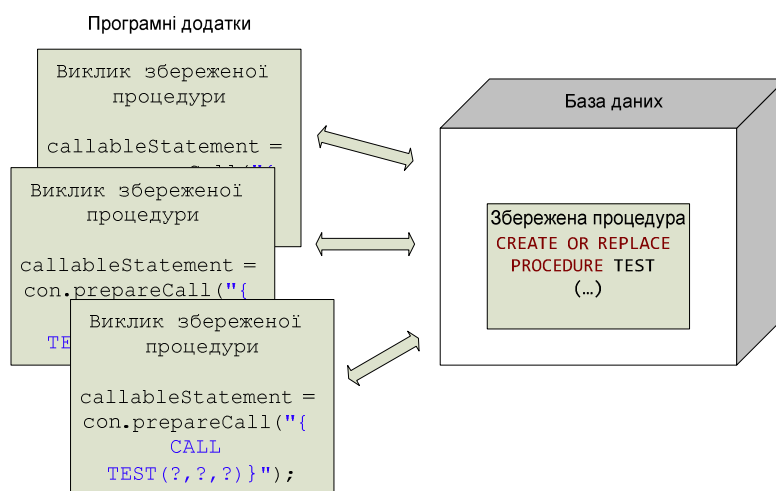


Рис. 2. Виклик однієї збереженої процедури різними програмами

Для використання збережених процедур потрібно написати два програмних модулі: збережену процедуру, працюючу на сервері баз даних, і клієнтську прикладну програму, що працює на робочій станції клієнта або на проміжному сервері (наприклад, на Web-сервері). Клієнтська програма викликає збережену процедуру за допомогою одного з доступних методів.

Збережена процедура слідує певним угодам для обміну даними з клієнтськими програмами, що додає надійності в обміні інформацією та управлінням даними.

Створення збереженої процедури складається з декількох етапів:

- Визначення типу створюваної збереженої процедури. Необхідно вирішити, чи буде процедура тимчасовою чи користувацькою.

- Запланувати права доступу. При створенні збереженої процедури слід враховувати, що вона матиме ті ж права доступу до об'єктів бази даних, що і створив її користувач.

- Визначити параметри збереженої процедури. Подібно процедурам, що входять до складу більшості мов програмування, збережені процедури можуть мати вхідні і вихідні параметри різних типів. SQL Server дозволяє визначити для однієї збереженої процедури до 1024 параметрів.

Інструкція Transact-SQL на створення збереженої процедури така:

```
CREATE OR REPLACE PROCEDURE TEST (NAME IN VARCHAR2(128), VARCOUNT OUT INTEGER)
AS
BEGIN
  --
  #####
  DECLARE DATOP DATE;
  DECLARE TIMOP TIME;
  DECLARE SQLCODE INT DEFAULT 0;
  DECLARE ERR_COD INT DEFAULT 0;

  IF ERR_COD = 0 THEN
    IF (LENGTH(RTRIM(NAME)) > 0) THEN
      A1:BEGIN
        DECLARE CURSOR1 CURSOR WITH RETURN FOR
          SELECT * FROM NAME.TABLE
          WHERE NP = NAME
          AND DATN1 BETWEEN DATN AND (DATK + 10 DAY)
          ORDER BY DATN
          WITH UR;

        OPEN CURSOR1;
        SET SQLCODE_BUF=SQLCODE;
        WHILE SQLCODE_BUF = 0 DO FETCH CURSOR1 INTO BUF;
        SET SQLCODE_BUF=SQLCODE;
        IF SQLCODE_BUF = 0 THEN
          SET ANS = BUF;
        END IF;
        END WHILE;
        CLOSE CURSOR1;
      END A1;
    END IF;
  END IF;
  #####
END;
```

де TEST - ім'я створюваної процедури. Використовуючи префікси sp_, # та ##, можна визначити створювану процедуру відповідно як системну або тимчасову. Як видно з синтаксису

команди, не допускається вказувати ім'я власника, якому належатиме створювана процедура, а також ім'я бази даних, в якій повинна бути розміщена процедура. Таким чином, щоб розмістити створювану збережену

процедуру в конкретній базі даних, необхідно виконувати команду CREATE PROCEDURE в контексті цієї бази даних. REPLACE дає можливість при тестуванні та багаторазовому deploy збереженої процедури перезаписувати її в базі.

NAME IN VARCHAR2(128), VARCOUNT OUT INTEGER - параметри вхідні, вихідні та їхній тип. При завданні параметрів користувач повинен вказувати їх значення в тому ж порядку, в якому вони були перераховані при створенні процедури. Присвоїти параметру значення за замовчуванням, просто опустивши його при перерахуванні, не можна. Тим не менш, кінцеві параметри можуть бути без проблем опущені, якщо для них визначено значення за замовчуванням.

DEFAULT - значення, яке буде приймати відповідний параметр за замовчуванням. Таким чином, при виклику процедури можна буде не вказувати явно значення відповідного параметра. У цьому випадку буде використано значення, задане за допомогою параметра default.

AS - це ключове слово свідчить про початок власне тіла збереженої процедури, тобто набору команд Transact-SQL, за допомогою яких і буде

реалізовуватися дія. У тілі процедури можуть використовуватися практично всі команди Transact-SQL, оголошуватися транзакції, встановлюватися блокування і вироблятися виклики інших збережених процедур. Для виходу з збереженої процедури можна використовувати команду RETURN.

Збережена процедура не з'єднується з базою даних, а використовує з'єднання вже встановлене клієнтом і оперує виключно потрібними їй параметрами вона використовує передані їй параметри для уточнення алгоритму своєї роботи і може повертати набір значень, рядків або змінювати значення в базі даних, відповідно до поставленої їй задачі.

У прикладних програм, що використовують збережені процедури, є ряд переваг, головною з яких є захист бази даних від не санкціонованого доступу.

Слід підкреслити, що збережені процедури дозволяють організувати контроль доступу до інформації, що зберігається в базі даних відповідно до поставлених вимог.

Приклад коду на мові java.

Метод, що відображає виклик збереженої процедури:

```
public static String VIDPOV (String XML){
    String otvXML = "";
    Connection con = getConnect();
    CallableStatement callableStatement;
    try {
        callableStatement = con.prepareCall("
                                                { CALL MESTA.TRAIN_LIST(?,?)});
        callableStatement.setString(1, XML);
        callableStatement.setString(2, otvXML);
        callableStatement.registerOutParameter(1, Types.CHAR);
        callableStatement.registerOutParameter(2, Types.CHAR);
        callableStatement.execute();
        otvXML = callableStatement.getString(2);
        con.commit();
        con.close();
    }
    catch (SQLException ex)
    {
        String Err = "Помилка";
        int SqlCode = ex.getErrorCode();
        String sss = ex.getMessage();
        String sss1 = ex.getSQLState();
        System.out.println(Err + ": " + SqlCode + " Message:
            "+sss+ " SQLState: "+sss1);
    }
    return otvXML;
}
```

Доброю практикою є обмеження користувача, під яким «ходить» в базу клієнтська програма в правах таким чином, що він не має прав на читання і зміни ніяких об'єктів. Лише виконує дозволені йому функції.

Таким чином можна жорстко контролювати які дії доступні клієнту, зменшується ймовірність порушення цілісності даних через помилку клієнтського додатку.

Висновки з даного дослідження і перспективи розвитку

Провівши аналіз використання збережених процедур, дійшли висновку, що гнучкість налаштувань прав доступу до даних, що зберігаються в базі та обмеженість SQL коду в рамках робочого алгоритму збереженої процедури, відкриває можливість контролю та доступу тільки до конкретних значень без можливості корегування запитів. Також виявлено що завдяки встановленню процедур на базу одноразово та можливість її виклику безліччю програм і безліччю разів це скорочує час обробки програмного коду завдяки зменшенню кількості запитів до бази даних. Використання універсальних збережених процедур оптимізує роботу системи в цілому.

Також відкриваються великі перспективи завдяки можливостям визначеними функціями в збереженій процедурі та вказаними діями користувача.

Література

1. <http://www.ibm.com/>, дата звернення 01.06.14
2. <http://www.mysql.com/>, дата звернення 15.07.14
3. MySQL 5 / Кузнецов М.В., Симдянов И.В. -СПб.: БХВ-Петербург, 2010.-1024с.
4. Пушников А. Ю. Введение в системы управления базами данных, Уфа.: изд-е Башкирского ун-та, 2000. с. 1-15, 45-47.

Petkov R.O. The perfection of software components of analytical monitoring systems. The main issue that has been considered in this scientific article is the perfection of software components of analytical monitoring systems by means of implementation of stored procedures, which are separate software modules, combining several queries with logical operations on the data obtained and stored in the database after a single installation. The article describes the issues, concerning the distribution of the load on the system by means of the transfer of the maximum possible number of logical operations to stored procedures as well as basic concepts and syntax which is used to write stored procedures in PL / SQL and call the procedure test sample in java language in which one should specify the name and parameters of the procedure for referencing the database through previously established connection. It also describes the possibility to access the procedure by a certain number of users from different applications at the same time and prevent unauthorized interference into the database, controlling users' access to stored information, giving them the rights in accordance with the requirements set by the system administrator.

Key words: monitoring, transformer equipment, stored procedure.

Петков Р.О. Совершенствование программных компонентов аналитических мониторинговых систем. Главным вопросом, который рассмотрен в этой научной статье, есть совершенствование программных компонентов аналитических мониторинговых систем путем внедрения хранимых процедур, которые являются отдельными программными модулями, объединяющими в себе несколько запросов с логическими действиями над полученными данными, и хранятся в базе данных после одноразовой установки. Описываются вопросы распределения нагрузки на систему переносом максимально возможного числа логических операций в хранимые процедуры, также описываются основные понятия и синтаксис, который используется для написания хранимых процедур на PL/SQL и вызова тестового образца процедуры на языке java, в котором нужно указывать название и параметры процедуры для обращения по ранее установленному коннекту к базе. Описывается возможность одновременного обращения к процедуре определенного количества пользователей из разных приложений и предотвращения несанкционированного вмешательства в базу данных, контролируя доступ пользователей к хранищейся информации, наделяя их правами в соответствии с поставленными требованиями администратора системы.

Ключевые слова: мониторинг, трансформаторное оборудование, хранимая процедура.

Рецензент д.т.н., професор Тимченко Л.І. (ДЕТУТ)

Поступила 26.06.2015г.

Petkov R.O., postgraduate of department "Automation and Computer-Integrated Technologies of Transport", State Economic and Technological University of Transport, Kyiv, Ukraine.

Петков Р.О., аспірант кафедри «Автоматизація та комп'ютерно-інтегровані технології транспорту», Державний економіко-технологічний університет транспорту, Київ, Україна.