

При розгляді питання поширення оновлень основною проблемою реплікації даних є те, що оновлення будь-якого логічного об'єкту повинно розповсюджуватись на всі збережені копії об'єкту. Труднощі виникають через те, що деякий вузол, який містить даний об'єкт, може бути недоступним саме в момент оновлення. В такому випадку очевидна стратегія негайного розповсюдження оновлень на всі копії може виявитися неприйнятною, оскільки передбачається, що оновлення буде провалено, якщо одна з цих копій буде недоступна в даний момент часу. Існують випадки, коли при використанні такої стратегії дані дійсно будуть менш доступними порівняно з їх використанням в нереплікаційному вигляді.

Ще одна проблема – це проблема керування відновленням. Керування відновленням в розподілених системах звичайно побудоване на протоколі двохфазної фіксації. Підтримка двохфазної фіксації необхідна в будь-якому середовищі, в якому одна транзакція може взаємодіяти з декількома автономними адміністраторами ресурсів. Однак вона особливо важлива в розподіленій системі, оскільки адміністратори ресурсів, тобто локальні СУБД, діють на різних вузлах і послідовно, і автономно.

Остання розглянута нами проблема – це керування паралельністю. Управління паралелізмом в більшості розподілених систем, як і в багатьох нерозподілених системах, засновано на блокуванні. Однак в розподіленій системі запити на перевірку, установку і зняття блокувань, є повідомленням, що несе за собою додаткові накладні видатки. Розглянемо, наприклад, транзакцію  $T$ , котра потребує оновлення об'єкту, який має реплікації на  $n$  віддалених вузлах. Якщо кожен вузол керує блокуваннями для об'єктів, збережених на цьому вузлі, то для простішого способу управління паралелізмом потрібно було б принаймі  $5n$  повідомлень.

Отже, з вище наведеного можна зробити висновок, що основною проблемою інформаційних мереж є їх повільність, проте ефективність роботи розподіленої СУБД можна покращити завдяки вирішенню вище перерахованих проблем.

#### **Список використаних джерел**

1. Карпова Т. Базы данных. Модели, разработка, реализация. – С-Пб.: Питер, 2004.
2. Дейт К.Дж. Введение в системы баз данных. – К.: Диалектика, 2005.

УДК 004.75

## **ОРГАНІЗАЦІЯ WEB-ДОСТУПУ ДО РОЗПОДІЛЕНОЇ БАЗИ ДАНИХ ДЛЯ МОДЕЛЮВАННЯ ДИСКРЕТНИХ ДИНАМІЧНИХ СИСТЕМ**

**Співак І.Я., Брославський В.П.**

Тернопільський національний економічний університет

Архітектура "клієнт-сервер" є фактичним стандартом для багатокористувацьких СУБД, прикладні програми, реалізовані в її рамках, мають розподілений характер. Тобто, частина функцій прикладної програми реалізована в програмі-клієнті, інша - в програмі-сервері, причому для їх взаємодії буде визначений деякий протокол.

Основний принцип технології "клієнт-сервер" полягає в розподілі функцій стандартного інтерактивного застосування на чотири групи, що мають різну природу. Перша група - це функції введення і відображення даних. Друга група об'єднує суто прикладні функції, що характерні для цієї предметної області. До третьої групи відносяться фундаментальні функції зберігання і управління інформаційними ресурсами (базами даних, файловими системами і так далі). А функції четвертої групи - службові, що відіграють роль зв'язку між функціями перших трьох груп.

Виділяються чотири підходи, реалізовані в наступних моделях [1]:

- модель файлового сервера (File Server - FS);
- модель доступу до віддалених даних (Remote Data Access - RDA);
- модель сервера бази даних (DataBase Server - DBS);
- модель сервера додатків (Application Server - AS).

FS-модель є базовою для локальних мереж. Відповідно до цієї моделі один з комп'ютерів в мережі вважається файловим сервером і надає послуги з обробки файлів іншим комп'ютерам. На інших комп'ютерах в мережі функціонує додаток, в кодах якого поєднані компонент представлення і

прикладний компонент. До технологічних недоліків моделі відносять високий мережевий трафік, вузький спектр операцій маніпулювання даними, відсутність адекватних засобів безпеки доступу до даних тощо. Усі перераховані недоліки - наслідок внутрішніх властивих FS-моделі обмежень, що визначаються її характером.

RDA-модель істотно відрізняється від FS-моделі характером компонента доступу до інформаційних ресурсів. Це, як правило, SQL-сервер. У RDA-моделі коди компонента представлення і прикладного компонента поєднані і виконуються на комп'ютері-клієнті. Останній підтримує як функції введення і відображення даних, так і суто прикладні функції. Доступ до інформаційних ресурсів забезпечується або операторами спеціальної мови (мови SQL, якщо йдеться про бази дані) або викликами функцій спеціальної бібліотеки (якщо є відповідний інтерфейс прикладного програмування - API).

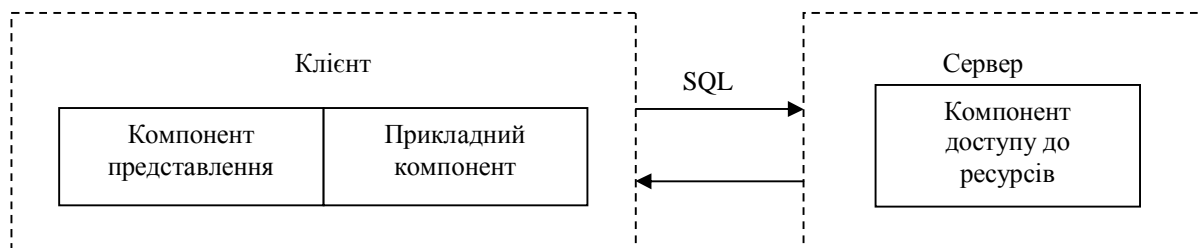


Рисунок 1 - Модель доступу до віддалених даних

Основна перевага RDA-моделі полягає в уніфікації інтерфейсу "клієнт-сервер" у вигляді мови SQL. Саме тому для реалізації доступу до розподіленої бази даних було вибрано саме цю модель.

Разом з RDA-моделлю все більшої популярності набуває DBS-модель. Остання реалізована в деяких реляційних СУБД (Informix, Ingres, Sybase, Oracle, InterBase). Її основу складає механізм процедур, що зберігаються, - засіб програмування SQL-сервера. Процедури зберігаються в словнику бази даних, розділяються між декількома клієнтами і виконуються на тому ж комп'ютері, де функціонує SQL-сервер. Мова, на якій розробляються процедури, що зберігаються, є процедурним розширенням мови запитів SQL і унікальною для кожної конкретної СУБД. У DBS-моделі компонент представлення виконується на комп'ютері-клієнті, тоді як прикладний компонент оформлений як набір процедур, що зберігаються, і функціонує на комп'ютері-сервері БД. Там же виконується компонент доступу до даних, тобто ядро СУБД. Переваги DBS-моделі: можливість централізованого адміністрування прикладних функцій, і зниження трафіку, можливість розподілу процедури між декількома застосуваннями, економія ресурсів комп'ютера. До недоліків можна віднести обмеженість засобів, що використовуються для написання процедур, що зберігаються, які є різноманітними процедурними розширеннями SQL. Сфера їх використання обмежена конкретною СУБД, у більшості СУБД відсутні можливості відладки і тестування розроблених процедур, що зберігаються.

На практиці часто використовується змішані моделі, коли підтримка цілісності бази даних і деякі прості прикладні функції виконуються процедурами, що зберігаються (DBS-модель), а складніші функції реалізуються безпосередньо в прикладній програмі, яка працює на комп'ютері-клієнті (RDA-модель).

У AS-моделі процес, що виконується на комп'ютері-клієнті, відповідає за інтерфейс з користувачем. Звертаючись за виконанням послуг до прикладного компонента, цей процес відіграє роль клієнта додатка. Прикладний компонент реалізований як група процесів, що виконують прикладні функції і називається сервером додатка. Усі операції над інформаційними ресурсами виконуються відповідним компонентом, по відношенню до якого AS відіграє роль клієнта.

RDA - і DBS-моделі спираються на дволанкову схему розподілу функцій. У RDA-моделі прикладні функції надані програмі-клієнті, в DBS-моделі відповідальність за їх виконання бере на себе ядро СУБД. У першому випадку прикладний компонент зливається з компонентом представлення, в другому - інтегрується в компонент доступу до інформаційних ресурсів. У AS-моделі реалізована триланкова схема розподілу функцій, де прикладний компонент виділений як найважливіший відокремлений елемент додатка, для його визначення використовуються універсальні механізми багатозадачної операційної системи, і стандартизовані інтерфейси з двома іншими компонентами.

Віртуальний доступ віддаленому користувачеві до проблемно-орієнтованого програмного забезпечення і високопродуктивної обчислювальної системи забезпечує простий доступ до ПЗ без

встановлення його на комп'ютері користувача. У цій роботі використовуються технології: додатки на основі MatLab web-server, що дозволяє досліджувати широкий клас моделей.

Компонент MatLab web-server (MWS) призначений для створення додатка MatLab, що використовує інтерфейс WWW для посилання запитів в MatLab, з наступним виведенням результатів в web-браузері. Тобто, користувач заходить на спеціально підготовлену сторінку в інтернеті та вводить дані поставленого завдання. Вони поступають в програму MatLab, яка, виходячи з введених даних, проводить необхідні обчислення, відправляє результат, який візуалізується безпосередньо на сторінці web-браузера. Таким чином, користувач, маючи web-браузер, використовує цей компонент для можливості побудови доступної, універсальної і зручної системи рішення математичних завдань, яка не накладатиме на користувача ніяких апаратних і програмних обмежень, оскільки усі необхідні операції виконуватимуться на сервері.

MatLab web-server (MWS) є сервіс взаємодії MatLab з web-сервером. У якості web-сервера ми використали Apache, особливо популярного в середовищі Unix.

Інтерактивні демонстраційні версії в основному складаються з двох документів HTML і одного М. MatLab файла. Перший документ HTML потрібний для збору даних від користувача з використанням форм HTML. Потім дані посилаються серверу, де витягаються з документів HTML і передаються програмі MatLab (М. файл), який виконує усі необхідні числові обчислення. Документ HTML, що містить результати у вигляді тексту або зображення, що формується web-сервером MatLab і посилається назад користувачеві, де відображається в його web-браузері. [2]

Кожна демонстрація складається з програми, написаної в середовищі MatLab, результатом виконання якої можуть бути чисельні або графічні матеріали, які MatLab web-server вставляє в html-файл і відправляє його на комп'ютер користувача. Для передачі вхідних даних також використовується html-форма, дані з якої витягаються сервером і передаються в програму. Таким чином, користувач може багаторазово використовувати демонстраційну програму, задаючи різні вхідні дані і спостерігаючи за результатом. При цьому наявність встановленого пакету MatLab на комп'ютері користувача не потрібно. Потрібний звичайний браузер, наприклад, Internet explorer. На web-сервері має бути дозволений запуск cgi-додатків, встановлений MatLab і MatLab web-server.

Розглянута технологія віддаленого доступу передбачає, що клієнт є "споживачем" готової моделі, яка має бути заздалегідь розроблена програмістом і встановлена на сервері

#### **Список використаних джерел**

1. Кузнецов С.Д. "Основы современных баз данных", К., 1999. – 257 с.
2. Лазарев Ю. Ф. Моделирование процессов и систем в Matlab. Учебный курс. – СПб.: Питер; Киев: Издат. группа BHV, 2005. – 512 с.

УДК 027.7

## **ВДОСКОНАЛЕНА МОДЕЛЬ ОБСЛУГОВУВАННЯ ПОТОКУ КОРИСТУВАЧІВ БІБЛІОТЕКИ**

**Хмурич І.М.**

Тернопільський національний економічний університет

### **Вступ**

Високий освітній рівень населення є однією з передумов успішності нації, оскільки відбувається підвищення добробуту її громадян та конкурентоздатності на міжнародному рівні. Доступність інформації належить до основних людських та громадянських прав в інформаційному суспільстві, і є невід'ємною умовою демократії.

Для розвинутого інформаційного суспільства та для системи освіти бібліотека відіграє виняткову роль інституції просвітництва та загальної грамотності. Традиційна грамотність лежить в основі медійної грамотності, яка передбачає можливість пошуку відповідної інформації як з друкованих так і з електронних джерел, можливість одержувати та порівнювати інформаційні ресурси різних джерел, а також уміння застосовувати здобуті знання.

Бібліотечні послуги враховують потреби різних вікових груп. Населення має довіряти бібліотеці. Протягом усього свідомого життя людина постійно вдається до послуг бібліотеки. Бібліотека так само як культура, наука і мистецтво належить до безперечних цінностей громадянського суспільства.