

УДК 519.68

Ю. Стех, А. Керницький

Національний університет “Львівська політехніка”, кафедра САПР

АНАЛІЗ ТЕХНОЛОГІЇ КЛІЄНТ-СЕРВЕР У СУЧАСНИХ СИСТЕМАХ УПРАВЛІННЯ БАЗАМИ ДАНИХ

© Стех Ю., Керницький А., 2002

Виконано порівняльний аналіз моделей технології клієнт-сервер у сучасних системах управління базами даних. Наведено переваги і недоліки кожної моделі.

Comparing analysis of technology Client-server models in modern systems of data base control is done. Advantages and disadvantages of every model are shown.

Технологія клієнт-сервер – це технологія взаємодії комп’ютерів у мережі. При такій технології комп’ютери не є рівноправними. Комп’ютери, які керують тим чи іншим ресурсом, прийнято називати сервером цього ресурсу, а комп’ютери, які використовують ці ресурси – клієнтами.

У мережі один і той же комп’ютер може виконувати роль як клієнта, так і сервера. Наприклад, в інформаційній системі, яка складається з персональних комп’ютерів, великої ЕОМ і міні-комп’ютера під управлінням UNIX, останній може виступати як в ролі сервера бази даних, що обслуговує запити клієнтів - персональних комп’ютерів, так і в ролі клієнта, котрий скерує запити до великої ЕОМ.

Використання технології клієнт-сервер означає, що прикладні програми, які реалізовані в її рамках, будуть мати розподілений характер.

Частина функцій прикладної програми буде реалізована в програмі-клієнті, інша – в програмі-сервері. Для їх взаємодії буде визначений певний протокол.

Основний принцип технології клієнт-сервер полягає у розподілі функцій стандартної інтерактивної аплікації на чотири групи:

- функції введення-відображення даних;
- прикладні функції, котрі характерні для даної предметної області;
- фундаментальні функції зберігання і управління інформаційними ресурсами;
- службові функції, які відіграють роль зв’язок між функціями перших трьох груп.

Відповідно до цього виділяють такі логічні компоненти:

- компонент подання, який реалізує функції першої групи;
 - прикладний компонент, який підтримує функції другої групи;
 - компонент доступу до інформаційних ресурсів, який підтримує функції третьої групи.
- Функції четвертої групи реалізує протокол взаємодії.

Відмінності в реалізації технології клієнт-сервер визначаються чотирма факторами:

- тим, в які види програмного забезпечення інтегрований кожен з цих компонентів;

- які механізми програмного забезпечення використовуються для реалізації функції всіх трьох груп;

- як логічні компоненти розподілені між комп'ютерами в мережі;
- які механізми використовуються для зв'язку компонентів між собою.

Виділяють чотири моделі технології клієнт-сервер:

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

FS-модель є базовою для локальних мереж персональних комп'ютерів. Вона широко використовувалася розробниками баз даних (БД), котрі застосовували такі системи як FoxPRO, Clipper, Clarion, Paradox. Суть моделі полягає у наступному. Один із комп'ютерів у мережі вважається файловим сервером і надає послуги з обробки файлів іншим комп'ютерам. Файловий сервер працює під керуванням мережевої операційної системи і відіграє роль компонента доступу до інформаційних ресурсів. На інших комп'ютерах у мережі функціонують аплікації, у кодах котрих об'єднані компоненти представлення і прикладний компонент (рис. 1). Протоколом обміну є набір низькорівневих викликів, які забезпечують аплікації доступ до файлової системи на файл-сервері. У таких системах на декількох персональних комп'ютерах виконується як прикладна програма, так і копія системи управління базами даних (СУБД), а бази даних містяться у відокремлених файлах, які знаходяться на файловому сервері. Коли прикладна програма звертається до бази даних, СУБД скеровує запит на файловий сервер. У цьому запиті вказані файли, де знаходяться потрібні дані. У відповідь на запит файловий сервер направляє по мережі необхідний блок даних. СУБД, отримавши його, виконує над даними дії, які були визначені в прикладній програмі.

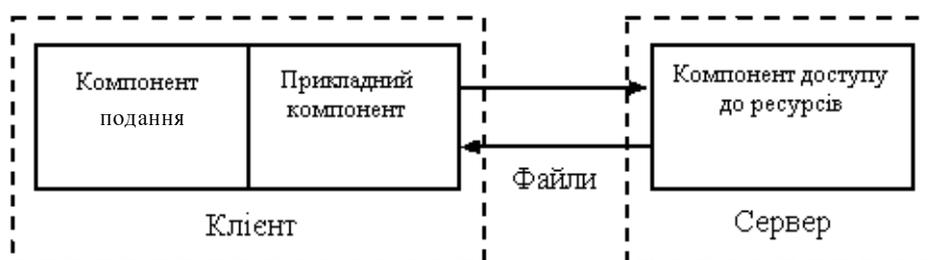


Рис. 1. Модель файлового сервера

До недоліків моделі FS належать:

- високий мережевий трафік; наприклад, якщо в результаті запиту клієнт повинен отримати 2 записи із таблиці об'ємом 100тис. записів, тоді всі 100тис. записів будуть скопійовані з файл-сервера на клієнтський комп'ютер; в результаті збільшиться не лише мережевий трафік, але і вимоги до апаратних потужностей клієнтських комп'ютерів;

- оскільки база даних є набором файлів на мережевому сервері, то доступ до таблиць регулюється тільки мережевою операційною системою, що робить таку БД по суті без-

захисною перед випадковим або навмисним спотворенням, знищенням або крадіжкою інформації, яка у ній зберігається;

- бізнес-правила у системах файл-сервер реалізуються в програмі клієнта, що в принципі не виключає проектування суперечних бізнес-правил у різних програмах; змістовна цілісність при цьому може порушуватися;
- недостатньо розвинутий апарат транзакцій локальних СУБД є потенційним джерелом помилок як при одночасному внесенні змін в один і той же запис, так і при реалізації відкочування результатів серії, об'єднаних за змістом в одне ціле операцій над БД, коли деякі з них завершилися успішно, а деякі – ні; це може порушити змістовну цілісність посилань БД.

Технологічніша RDA-модель суттєво відрізняється від FS-моделі характером компонента доступу до інформаційних ресурсів (рис. 2). Це, як правило, SQL-сервер. У RDA-моделі коди компоненти подання і прикладного компонента суміщені і виконуються на комп'ютері-клієнті. Останній підтримує як функції введення і відображення даних, так і чисто прикладні функції. Доступ до інформаційних ресурсів забезпечується або операторами спеціальної мови (наприклад, мова SQL), або викликами функцій спеціальної бібліотеки інтерфейсу прикладного програмування (API).

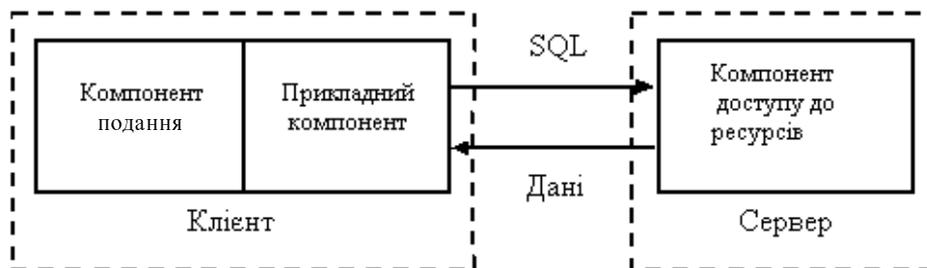


Рис. 2. Модель доступу до віддалених даних

Клієнт посилає запити до інформаційних ресурсів БД по мережі віддаленому комп'ютеру. На цьому комп'ютері функціонує ядро СУБД, яке обробляє запити і повертає клієнту результат, оформлений як блок даних. При цьому ініціатором маніпулювання з даними виступають програми, які виконуються на комп'ютерах-клієнтах. Ядру СУБД відводиться пасивна роль – обслуговування запитів і оброблення даних.

RDA-модель позбавлена недоліків, які властиві як системам з централізованою архітектурою, так і системам з файловим сервером.

Перенесення компонента подання і прикладного компонента на комп'ютери-клієнти суттєво розвантажує сервер БД, мінімізуючи загальну кількість процесів операційної системи. Процесор або процесори сервера БД повністю завантажуються операціями обробки даних, запитів і транзакцій. Різко зменшується завантаження мережі, оскільки по ній передаються від клієнта до сервера не запити на ввід-вивід (як у системах з файловим сервером), а запити на мові SQL, об'єм яких значно менший.

Однак задовільне адміністрування аплікацій в RDA-моделі практично неможливе через суміщення в одній програмі різних за своєю природою функцій (функції подання і прикладних функцій).

DBS-модель реалізована у деяких сучасних реляційних СУБД (Informix, Ingres, Sybase, Oracle). Її основу становить механізм процедур, що зберігаються – засіб програмування SQL-сервера. Процедури зберігаються у словнику бази даних, розподіляються між декількома клієнтами і виконуються на тому ж комп'ютері, де функціонує SQL-сервер. Мови, на яких розробляються процедури, що зберігаються, являють собою процедурне розширення мови запитів SQL і унікальні для кожної конкретної СУБД.

В DBS-моделі компонент подання виконується на комп'ютері –клієнті, тоді як прикладний компонент оформлений як набір процедур, що зберігається і функціонує на комп'ютері-сервері БД. Там же виконується компонент доступу до даних, тобто ядро СУБД (рис. 3).

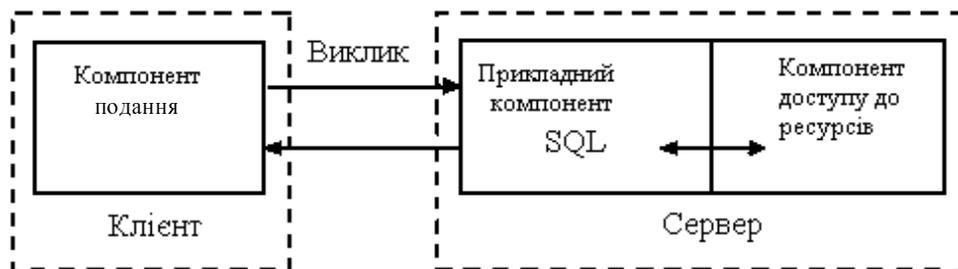


Рис. 3. Модель сервера бази даних

Перевагами DBS-моделі є:

- можливість централізованого адміністрування прикладними функціями;
- зниження трафіку (замість SQL-запитів по мережі направляються виклики процедур, що зберігаються);
- можливість розподілу процедури між декількома аплікаціями;
- економія ресурсів комп'ютера за рахунок багаторазового використання плану виконання процедури.

До недоліків моделі можна віднести обмеженість засобів, які використовуються для написання процедур, що зберігаються.

Сфера їх використання обмежена конкретною СУБД. У більшості СУБД відсутні засоби налагодження і тестування розроблених процедур, що зберігаються.

В AS-моделі процес, який виконується на комп'ютері-клієнті, відповідає за інтерфейс з користувачем (тобто реалізує функції першої групи). Звертаючись за виконанням послуг до прикладного компонента, цей процес відіграє роль клієнта аплікації (Application Client - AC). Прикладний компонент реалізований як група процесів, які виконують прикладні функції і називається сервером аплікації (Application Server - AS). Всі операції над інформаційними ресурсами виконуються відповідним компонентом, відносно до якого AS відіграє роль клієнта (рис. 4).

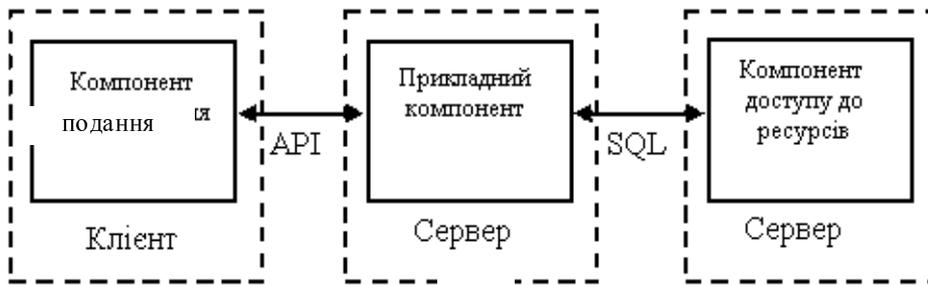


Рис. 4. Модель сервера аплікацій

RDA- і DBS-моделі спираються на дволанкову схему розділення функцій. У RDA-моделі прикладні функції надані програмі-клієнту, а в DBS-моделі вони реалізовані в ядрі СУБД. В AS-моделі реалізована триланкова схема розподілення функцій. Прикладний компонент виділений як найважливіший ізольований елемент аплікації. Для його визначення використовуються універсальні механізми багатозадачної операційної системи. Інтерфейси взаємодії прикладного компонента з двома іншими компонентами стандартизовані. AS-модель є фундаментом для моніторів оброблення транзакцій (Transaction Processing Monitors - TPM), які виділяються як особливий вид програмного забезпечення.

1. *Beynon-Davies P. Database Systems. MacMillan, 1996.*
2. *Боровски С. Oracle 7. Вычисления клиент/сервер. – М., 1996.*
3. *Connoly T., Begg C., Strachan A. Database Systems. A practical approach to design, implementation and management. Addison-Wesley, 1996.*
4. *Сигнор Р., Стегман М.О. Использование ODBC для доступа к базам данных. – М., 1995.*
5. *Хендерсон К. Delphi 3 и системы клиент/сервер. – К., 1997.*
6. *Хендерсон К. Руководство разработчика баз данных в Delphi 2. – К., 1996.*
7. *Шумаков П.В., Фаронов В.В. Delphi 5 Руководство разработчика баз данных. – М., 2000*