

СТВОРЕННЯ СЕРВЕРНОЇ СКБД, ЩО ПІДТРИМУЄ СІТКОВУ МОДЕЛЬ ДАНИХ, НА БАЗІ ЛОКАЛЬНОЇ

Розглянуто концепції та підходи до створення клієнт-серверної СКБД, що підтримує сіткову модель даних, на базі локальної моделі, для реалізації якої використовується СКБД dbVista. Наведено схеми реалізації серверної СКБД на платформі Windows NT, де використовуються поіменовані канали та сервіси операційної системи. Показано універсальність запропонованого підходу, що дає змогу за такими самими принципами без великих витрат створити такого типу серверну СКБД на іншій платформі. Розроблена СКБД є одним з основних компонентів АС Деймос (проектування корпусу судна), в якій раніше використовувалася локальна СКБД dbVista. Експлуатація протягом дев'яти місяців серверу, до якого приєднані 25 клієнтів, засвідчила його надійність і прийнятні часові характеристики.

Загальна постановка проблеми та її зв'язок з науково-практичними задачами. Бази даних є ядром сучасних систем автоматизації (СА), тому система керування базою даних (СКБД) є важливим компонентом будь-якої СА і ефективність роботи якої багато в чому залежить від того, наскільки ефективно обрана СКБД підтримує найбільш раціональну для даної предметної сфери структуру та моделі даних, наскільки мобільним і сумісним з функціональними програмами є її програмне забезпечення, а також наскільки ефективно і зручно можна побудувати технологію обробки інформації при експлуатації СА. Об'єктом дослідження цієї статті є СА в інформаційних технологіях проектування (ІТП), тому названі критерії ефективності СКБД розглядатимуться під цим кутом зору. Для систем автоматизованого проектування (виробів, технологічних процесів тощо) найбільш адекватною є сіткова модель даних [1; 2]. Мобільність програмного забезпечення СКБД та сумісність з функціональним програмним забезпеченням (ФПЗ) систем автоматизованого проектування (САПР) на сучасному етапі досягається внаслідок використання об'єктно-орієнтованої методології як при створенні СКБД, так і при розробці ФПЗ [1; 3-5].

Для організації ефективної роботи колективу проектувальників необхідна технологія розподіленої обробки інформації, яка, зокрема, певною мірою забезпечена архітектурою «клієнт-сервер» [1; 6; 7]. Системи з такою архітектурою забезпечують багатокористувацький доступ до центральної бази даних, дають змогу наблизити структуру обчислювального комплексу до струк-

тури організації, підвищити надійність і потужність системи автоматизації. Таким чином, враховуючи інформаційні технології проектування, вибір або розробка СКБД, яка підтримує сіткову модель даних, об'єктно-орієнтовану методологію створення бази даних та прикладного програмного забезпечення і технологію розподілених баз даних хоча б на рівні архітектур «клієнт-сервер», є актуальною, оскільки на вітчизняному ринку програмних продуктів практично немає таких СКБД, за винятком, мабуть, db Raima, про що буде зазначено нижче.

Огляд публікацій та аналіз невирішених питань. Є багато досліджень [1; 6; 7 та ін.], в яких викладено основні концепції розподілених баз даних, типові структури і функції відповідних СКБД, але вони переважно стосуються реляційних баз даних. Існують і відомі потужні «клієнт-серверні» СКБД (Oracle, Sybase, Informics, MS SQL Server тощо), за допомогою яких можна організувати розподілену базу даних, але всі вони також підтримують лише реляційну модель даних. Як зазначалося вище, в інформаційних технологіях проектування (системи автоматизованого проектування, геоінформаційні системи та ін.) ефективніше й природніше використовувати сіткову модель даних [2]. Отже, проаналізуємо детальніше цю гілку еволюції баз даних.

У 80-х роках минулого століття фірма «Raima» розробила та випустила на ринок першу версію СКБД - dbVista [8], яка, на відміну від популярної і поширеної на той час реляційної моделі, підтримувала сіткову модель даних, що й забезпечило їй належне місце в САПР

Крім того, що сіткова модель давала змогу природніше, а отже, й ефективніше відображати предметну сферу, вона мала ще й таку позитивну характеристику, як добра імплементація у програми, написані мовою Сі. Застосування цієї СКБД надало можливість досить швидко і за мінімальних витрат розробити першу версію САПР - Деймос [9]. У наступних версіях системи Деймос було запропоновано й реалізовано об'єктно-орієнтовану методологію та поширену модель розробки програмного забезпечення САПР [4; 101.

На початку 1990-х років, коли переваги об'єктно-орієнтованого програмування (ООП) стали загально визнаними, фірма «Raima» однією з перших відреагувала на цей виклик і випустила у продаж свою версію Raima DataBase Manager 4.5 (RDM). Крім наведених, у зазначеній СКБД була можливість під час генерації структури БД одержати об'єкти для використання в програмах користувача (утиліта Object Manager). Об'єктами в даному випадку вважаються класи мовою С++ [11]. Наприкінці 1999 р. RDM викупила фірма «Centura», і нова версія цієї СКБД від нового власника отримала назву «Velocis». І знову в нову версію було додано передові інформаційні технології, в тому числі методологію ООП та клієнт-серверну модель. Зрозуміло, що сучасна версія Velocis має всі елементи, що роблять побудовані на її базі СА сучасними та ефективними. Однак використання СКБД Velocis, по-перше, є майже неможливим для підприємств, які не мають досить значних коштів на її придбання і супроводження. По-друге, розробник такої СА стає залежним від сторонньої фірми «Centura» в подальших версіях системи. По-третє, втрачається мобільність СА.

При використанні тільки локальної dbVista на сучасних комп'ютерах із великою потужністю виникала ще одна проблема: вони, як правило, мають для відкритих файлів КЕШ-пам'ять великого обсягу. Для двох користувачів СКБД операційною системою відкриваються свої незалежні КЕШ, і запис інформації на диск не завжди виконується в порядку звернення користувачів до бази даних, що може призвести до порушення цілісності БД. Тому автори статті запропонували власну версію мобільної клієнт-серверної СКБД на основі попередніх розробок.

Метою досліджень цієї праці є створення ефективної мобільної клієнт-серверної СКБД, що підтримує сіткову та об'єктну моделі даних,

на базі dbVista, текст програмних кодів якої є відкритим.

Результати досліджень. Із зазначеного вище можна дійти висновку, що для створення ефективного мобільного ПЗ для СА необхідно побудувати мобільну СКБД як клієнт-серверну. При цьому в клієнтську частину будуть включені задачі формування та передачі запиту, а в серверну - прийом запиту, його обробка та повернення результату. Внаслідок того, що СКБД повинна бути багатокористувацькою, на сервері необхідно реалізувати багатопоточність (для кожного запиту клієнта окремий потік) та синхронізацію цих потоків. Відомо, що в реляційних СКБД для зменшення трансакцій і трафіку мережі та, відповідно, для збільшення ефективності роботи використовують збережені процедури. Аналогом цього в dbVista є dli-бібліотеки користувача. ПЗ клієнта за допомогою Network Communication Processor (NCP), окрім передачі запиту на виконання окремих базових функцій для роботи з СКБД, що зберігаються в бібліотеці підтримки dbVista, може передавати запит на виконання або функції серверної розширеної бібліотеки роботи з БД, або функції dll-бібліотеки користувача. Ці запити передаються за допомогою Named Pipe, який створюється для кожного клієнта при реєстрації на сервері, де запит приймається серверним NCP та передається диспетчеру. Останній або виконує безпосередньо виклик базових функцій для роботи з СКБД, або створює окремий потік і виконує функцію розширеної бібліотеки чи dll-бібліотеки. Ця функція звертається до базових функцій для роботи з СКБД. Коли потокова функція закінчується, вона повертає результат обробки запиту клієнтові через відкритий для нього Named Pipe. До серверу можна приєднувати декілька dll-бібліотек користувача — при формуванні запиту в клієнтському ПЗ вказуються не тільки номер функції, параметри запиту, а ще й ім'я dll-бібліотеки. Допрацьована за наведеними проектними рішеннями СКБД dbVista набула певних властивостей серверної, і надалі називатимемо її ServerDB. На рис. 1 наведено загальну схему функціонування розробленої СКБД із виділеним сервером обробки запитів клієнтів. Багатокористувацький режим підтримується тепер за рахунок організації на сервері багатопоточності (Multithreading): кожен запит обробляється в окремому потоці (thread), враховуючи системні засоби та властивості СКБД dbVista. Для цього в ній використовується так званий Lock Manager — утиліта, що виконує функції закриття та відкриття

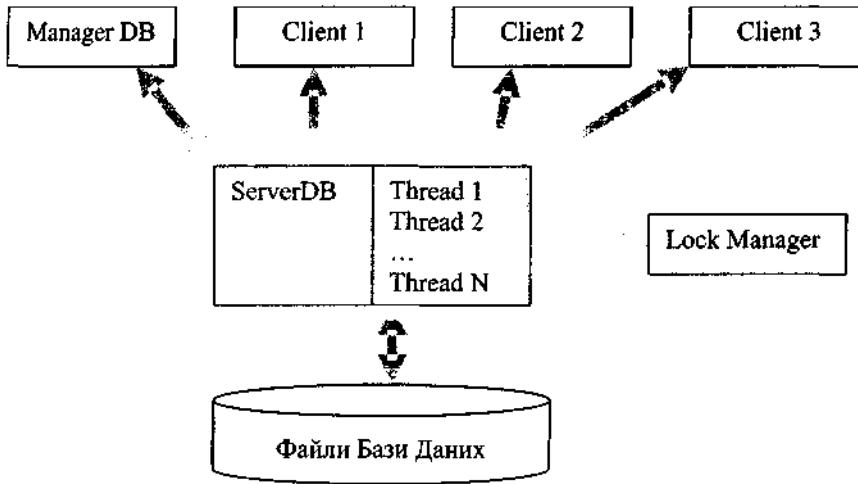


Рис. 1. Компоненти обробки запиту в СКБД АС Деймос

окремих файлів, записів, ключів та наборів даних.

СКБД dbVista, як і більшість інших СКБД, — це файли з даними, ключами та словником. При відкритті для кожної БД створюються КЕШ-буфери, які мають назву «сторінки». Кількість сторінок (за замовчанням 16) та розмір кожної з них (за замовчанням 16 Кб) може змінюватися спеціальною функцією (d_setpages). Коли база даних відкрита в багатокористувацькому режимі, всі зміни слід виконувати тільки за до-

могою транзакцій, що забезпечує підтримку цілісності бази даних db Vista. Для цього передбачено програмні засоби для створення (dtrbegin) та завершення (d_trend) транзакцій. Якщо під час транзакції виникне аварійна ситуація, то для аварійної зупинки транзакції повинна спрацювати функція (djabort). Після виконання функції dtrbegin СКБД dbVista створює тимчасову БД, в яку вносяться всі змінні сторінки. І лише функція dtrend ці сторінки переписує в БД та інформує LockManager про

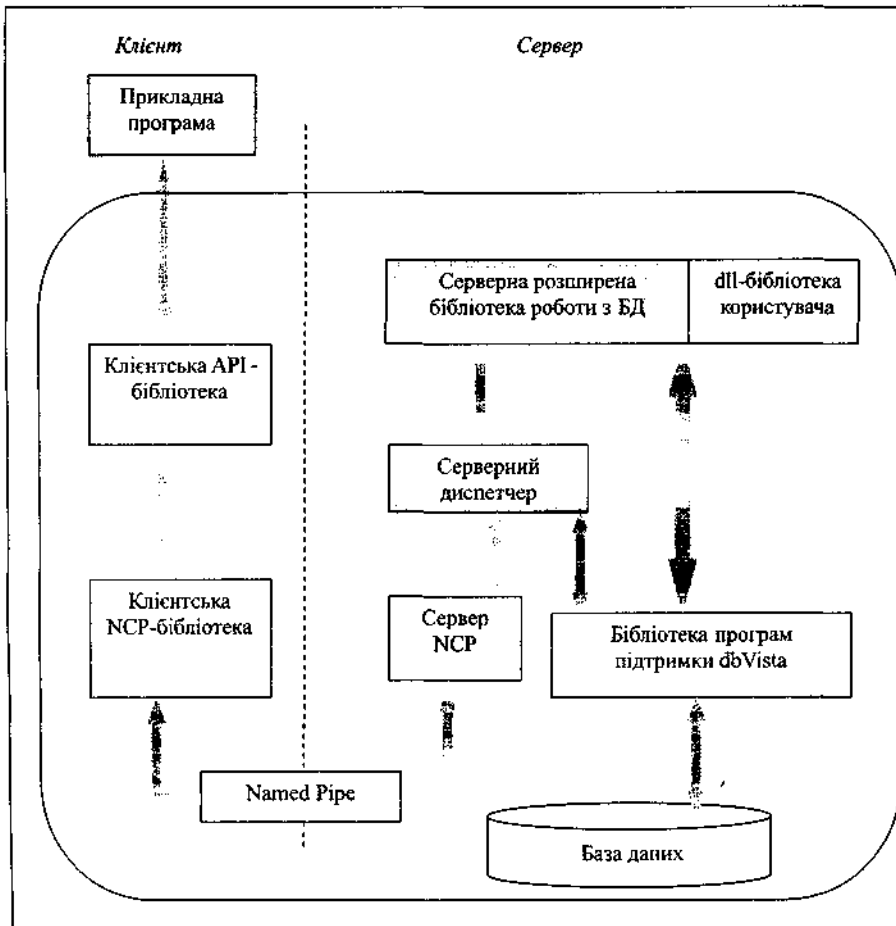


Рис. 2. Компоненти системи обробки запиту в серверній СКБД

необхідність звільнити всі закриті записи, набори, ключі чи файли.

У разі порушення цілісності БД (вимкнення світла, аварійного вимкнення серверу тощо) мають працювати утиліти перевірки цілісності (dbcheck) та автоматичного відновлення (dbrecover). До комплекту dbVista входить утиліта ручного корегування бази даних (dbedit) — за її допомогою можливе втручання в структуру БД. Нарис. 2 наведено компоненти обробки запиту в СКБД, що відповідає описаній вище технології функціонування.

У результаті проведення досліджень за зазначеними вище напрямками та розробки відповідного програмного забезпечення:

1) створено сервер бази даних ServerDB на базі програмного забезпечення локальної СКБД dbVista;

2) створено dll-бібліотеки користувачів САПР Деймос, які виконують функції збережених процедур у багатьох серверних СКБД;

3) створено механізми обміну інформацією між клієнтським та серверним програмним забезпеченням;

4) запропоновано механізм створення та обробки окремих потоків для виконання клієнтських запитів;

5) створено клієнтську dll-бібліотеку для формування та передачі запитів на сервери БД;

6) створено менеджер БД у вигляді спеціального клієнта БД.

Створена над локальною СКБД надбудова зробила програмне забезпечення САПР Деймос ефективнішим, зокрема, зменшився трафік у мережі. Крім того, внаслідок використання клієнт-серверної моделі та методів ООП значно підвищилася мобільність - без змін у ПЗ СА можливо перейти на будь-яку іншу СКБД. Для цього лише потрібно приєднати до серверу нову dll-бібліотеку користувача з новими Application Program Interface (API) до БД. Таким же чином можливо поступово розширювати інформаційний та функціональний простір САПР — сервер підтримує кілька відкритих БД, що дає змогу створювати та приєднувати нові dll-бібліотеки для роботи з ними. Поліпшилося управління СКБД — для серверу було запрограмовано менеджер БД. До його функцій належить резервне копіювання БД, відстеження роботи користувачів тощо. Якщо при використанні СКБД dbVista для копіювання необхідним було завершення всіх відкритих сеансів роботи САПР у мережі, то тепер сервер сам тимчасово припиняє їх із відповідним повідомленням, виконує копіювання, а після цього поновлює роботу

клієнтів. Окрім того, полегшилося керування користувачами: якщо з будь-якої причини клієнт заблокував запис, ключ чи файл, і сеанс завершено аварійно, то за допомогою створеного менеджера досить легко встановити такого клієнта та розблокувати БД.

Наведемо деякі характеристики бази даних, що підтримується розробленою серверною СКБД. Внаслідок того, що адреса запису складає 4 байти і перший байт є номером файла в базі даних, вона може мати до 256 файлів. Для представлення індексів виділено 3 байти, тому кількість слотів (записів) у файлі може сягати 16 777 215. Запису з номером 0 не існує - під цим номером зберігається службова інформація. Зауважимо, що кількість слотів може бути збільшено або зменшено за рахунок використання бітів з байта з номером файла. Отже, максимальна кількість всіх записів усіх файлів бази даних становить 4 294 967 040. Довжина кожного слота обмежена 4 байтами — таким чином, теоретично, максимальний можливий розмір запису становить 4 294 967 295. Однак адресний простір для процесу в операційній системі Windows складає 4 ГБ, і для ПЗ, що працює в режимі користувача, з цих 4 ГБ виділяється половина. Отже, теоретично, розмір одного запису може становити 2 ГБ. Зрозуміло, що внаслідок того, що запис з точки зору мови C++ є структурою (struct), її розмір обмежений розмірами оперативної пам'яті та розподілом адресного простору на стек (Stack) та купу (Heap).

Кожен із записів має: заголовок (1 байт), в якому міститься службова інформація, адресу запису (4 байти), час створення, якщо використовується (4 байти), час останнього модифікування (4 байти). Далі йдуть 12-байтні блоки опису підлеглих наборів, які містять три поля — кількість підлеглих членів, адресу першого та адресу останнього членів набору. Кількість наборів задається 4 байтами, тобто максимально може сягати 4 294 967 295.

Кількість потоків, як і кількість клієнтів, обмежено типом даних short мови C++, вона становить 65 535. Але ж зрозуміло, що це теоретичні підрахунки. Навантаження серверу зростає з кількістю клієнтів та складністю запитів, від яких залежить час виконання та існування потоку, який його виконує. При проектуванні АС з використанням наведеної архітектури, по-перше, було закладено можливості динамічної зміни часу очікування виконання запиту (timeout), а по-друге, коли час спливає, користувач отримує повідомлення про це та запит на повторне звернення до функцій БД. Реальна робота в ме-

режі з 25 робочими станціями АС Деймос, в якій була реалізована наведена схема роботи з БД, засвідчила задовільні часові результати — користувачі почувалися комфортно. При виконанні тестів з 35 клієнтами та одночасним виконанням запиту на запис складної складеної поверхні (запис більше 1 МБ) 8 клієнтів отримали повідомлення про тимчасове перевантаження серверу, але ж були здатні повторити запит. Зрозуміло, що в реальній роботі одночасне звернення до серверу зі «складними» запитами на запис великої кількості інформації від багатьох клієнтів мало ймовірно. Організаційна технологія роботи в НДІ «Центр», де зазначена система перебуває в експлуатації в проектувальному підрозділі корпусів суден, передбачає резервне копіювання наприкінці кожного робочого дня, але ж за останні 9 місяців роботи АС Деймос відкат (відміна операцій транзакції та повернення до попереднього стану БД через копіювання) трапився всього 3 рази, та й і то не з причин збою СКБД.

Перспективи подальших досліджень. Отже, клієнт-серверна надбудова над СКБД dbVista робить привабливим її використання в СА в ІТП. Крім того, вона дає змогу робити симбіоз кількох СКБД, що підтримують різні моделі даних, тобто з'являється можливість створення розподілених баз даних. Одним із напрямів розвитку цієї надбудови є формування інтерфейсів з реляційними СКБД, завдяки чому вона стане більш універсальною та привабливою не тільки для застосування в СА проектування, а й для програмування задач в СА технічної підготовки виробництва, де в більшості випадків краще використовувати реляційні СКБД. Зокрема, база даних САПР Деймос використовується в таких задачах, як формування різних технологічних документів: специфікації, плазові книги тощо. Нині у СА Деймос використовується інтерфейс з реляційними СКБД за допомогою технології OLEDB та ADO [12] без застосування запропонованої надбудови. Проте використання надбудови та сучасних стандартів AD02 або XML [13] є перспективним.

1. Конноли Т., Бега К., Странам А. Базы данных: проектирование, реализация и сопровождение. Теория и практика. - 2-е издание. — Москва — Санкт-Петербург — К.: Вильямс: 2000.- 1112с.
2. Быков Д. П., Фісун М. Т. Про організацію структури бази даних у САПР корпусу судна // УСІМ.— 2002.— № 6.— С 37-41.
3. Мюллер Р. Дж. Базы данных и UML. Проектирование.- М.: Изд-во «ЛОРИ», 2002.- 420 с.
4. Быков Д. П., Фісун М. Т. Застосування об'єктно-орієнтованої методології у САПР корпусів суден // Збірник наукових праць УДМТУ- Николаїв: УДМТУ- 2002.- С 126—132.
5. Быков Д. П., Фісун М. Т. Про використання об'єктно-орієнтованої СКБД в системах автоматизованого проектування // Вестник Херсонского государственного технического университета, — № 1 (19).
6. Дейт К. Дж. Введение в системы баз данных.— К.—М.: Диалектика, 1998.- 787 с.
7. Гарсия-Молана Г., Ульман Дж. Д., Уидом Дж. Системы баз данных. Полный курс— М.: Изд. дом «Вильямс», 2003.- 1088 с.
8. User's Guide.— Fifth Edition, revised for use with db_VISTA, Bellevue (USA), Raima Corporation, 1988.- 279 p.
9. Дубів І. І. Система деталювання й моделювання корпусу судна ДЕЙМОС: основні принципи та загальна структура // Труды УДМТУ.- Вип. 43.- С 32-39.- Николаїв: УДМТУ, 2001.
10. Быков Д. П., Фісун М. Т. Пошарова модель розробки програмного забезпечення САПР суден // Наукові записки НаУКМА.- Т. 22.- У 3 ч.- Ч. III: Природничі науки.- К.: Вид. дім «КМ Академія», 2003.- С. 477-483.
11. Дьюхарст С, Старк К. Программирование на С++.— К.: НИПФ «ДиаСофт», 1993.- 182 с
12. Олафсен Юджын, Скрайбнер Кенн, Уайт К. Девид и др. MFC и Visual C++ 6. Энциклопедия программиста: Пер. с англ. / Юджин Олафсен, Кенн Скрайбнер, К. Девид Уайт и др.- СПб.: ООО «ДиаСофтЮП», 2003.- 992 с.
13. Ахьян Р., Горев А., Макашарыпов С. Эффективная работа с СКБД.- Санкт-Петербург — Москва — Харьков — Минск: Юпитер, 1997.— 562 с.
14. Холзнер С. Книга XML. Энциклопедия.- 2-е изд.— СПб.: Питер, 2004.- 1101 с.

D. P. Bykov, M. T. Phisoon

CREATION CLIENT-SERVER'S DBMS, WHICH SUPPORTS NETWORK DATA MODEL, ON THE BASE OF LOCAL

In this article conceptions and methods of creation client-server's DBMS, which supports network data model, on the base of local, on the place of what DBMS dbVista is used, are looked through. Schemes of realization of server DBMS on the platform of Windows NT, where Named Pipes and servers of operation system are shown. Universality of this method is shown which helps without big wastes to create server DBMS of such type on the other platform. Developed DBMS is one of the main parts of AS Deymos (design of the hull of ship) in which local DBMS db Vista was used before. Use of server with connected 25 clients for 9 months showed its reliability and good time characteristic.