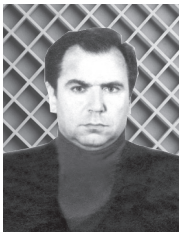
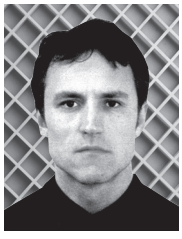
УДК 681.518

РОЗРОБЛЕННЯ КОНЦЕПЦІЇ ДОСТУПУ ДО БАЗ ДАНИХ ТА ЕЛЕКТРОННИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ УкрІНТЕІ ЧЕРЕЗ ІНТЕРНЕТ


В. І. Воронков, *канд. техн. наук,*
В. М. Куранда,
А. В. Круглий

Удосконалення процесів інформаційного забезпечення сфер науки, освіти, виробництва, державного управління і бізнесу на базі інформаційно-телекомунікаційної системи УкрІНТЕІ потребує проведення досліджень стану і можливостей цієї системи, результатом яких має стати визначення шляхів, засобів і механізмів її вдосконалення та розвитку.

Система НТІ за час свого існування накопичила величезні інформаційні ресурси, переважна більшість яких існує у вигляді баз даних (БД) і електронних інформаційних ресурсів (ЕІР). В УкрІНТЕІ створено унікальну систему БД і ЕІР з науково-технічної інформації:

• База даних науково-дослідних, дослідно-конструкторських робіт і дисертацій України.

• Міжнародна база даних AGRIS/CARIS.

• Корпоративна бібліографічна база даних «Зведений електронний каталог».

• Корпоративна база даних «Підприємства України: адреси і номенклатура продукції».

• База даних «Науково-технічні досягнення України».

• База даних «Технології України».

• База даних «Науково-технічні заходи».

• База даних «Наукові періодичні фахові видання МОН України».

• Електронний каталог «Наукові фахові видання України».

Згідно з тематичним планом УкрІНТЕІ на 2009–2010 рр. передбачено подальше вдосконалення системи інформаційно-аналітичного забезпечення шляхом організації доступу до БД/ЕІР УкрІНТЕІ через Інтернет.

Проблема полягає в тому, що забезпечення ефективного доступу до існуючих БД потребує визначення технологічних і організаційних рішень, пов'язаних з підтримкою актуальності баз даних і забезпеченням доступу до них. Практика використання мережевих технологій для доступу до БД надає широкий спектр технологічних рішень, по-різному пов'язаних між собою. Вибір конкретних рішень при забезпеченні доступу залежить від специфіки конкретної СУБД і від ряду інших чинників, зокрема: наявність спеціалістів, спроможних із мінімальними витратами засвоїти визначену гілку технологічних рішень; існування інших БД; WWW – доступ до яких має здійснюватися з мінімальними додатковими витратами і т.д.

Мета статті – дослідження проблеми і визначення необхідного варіанта рішення щодо створення для зовнішніх користувачів механізмів організації доступу до баз даних системи НТІ.

Теоретичною та методологічною основою дослідження стали праці таких учених, як В. М. Глушков, С. С. Лавров, Д. Кнут, Б. Шнейдерман та ін.

У публікаціях багатьох авторів висвітлено дослідження практичних аспектів управління базами даних. Зокрема, у роботах Д. Кнута [1] досліджено найважливіші алгоритми, що використовуються в інформатиці; Л. І. Курзанцевої [2] – проблеми побудови моделі користувача інтелектуального інтерфейсу комп'ютерних систем; О. Ждановича [3] –досвід застосування електронних баз даних у діалоговому режимі. Робота А. В. Мельничина [4] присвячена дослідженню ефективності основних методів пошуку інформації у файлах БД, побудові оптимальних схем методів і розробці нових підходів до пошуку інфор-

мації у файлах великих БД.

В авторських дослідженнях проведено аналіз основних сценаріїв, за якими може здійснюватися WWW – доступ до існуючих баз даних, і основну увагу було приділено наведеним нижче.

1. Одноразове або періодичне перетворення вмісту БД у статичні документи. У цьому варіанті вміст БД переглядає спеціальна програма, що створює множину файлів – зв'язкових HTML-документів. Отримані файли можуть бути перенесені на один або декілька WWW-серверів. Доступ до них здійснюється як до статичних гіпертекстових документів сервера.

Цей варіант характеризується мінімальними початковими витратами. Він ефективний на невеликих масивах даних простої структури і рідкісного відновлення, а також за наявності знижених вимог до актуальності даних, наданих через WWW. Крім цього, очевидним є повна відсутність механізму пошуку, хоча можливе розвинуте індексування.

У ролі перетворювача може виступати програмний комплекс, який автоматично або напівавтоматично генерує статичні документи. Програма-перетворювач може бути самостійно розробленою програмою або інтегрованим засобом класу генераторів звітів.

2. Динамічне створення гіпертекстових документів на основі вмісту БД. У цьому варіанті доступ до БД здійснюється спеціальною CGI-програмою, що запускається WWW – сервером у відповідь на запит WWW – клієнта. Ця програма, оброблюючи запит, переглядає вміст БД і створює вихідний HTML – документ, що повертається клієнту.

Це рішення є ефективним для великих баз даних із складною структурою і в разі необхідності – для підтримки операцій пошуку. Показаннями також є часте відновлення і неможливість синхронізації перетворення БД у статичні документи з відновленням вмісту. У цьому варіанті можна змінювати СУБД за допомогою WWW – інтерфейсів.

До недоліків цього методу можна віднести тривалий час опрацювання запитів, необхідність постійного доступу до основної бази даних, додаткове завантаження засобів підтримки БД, пов'язане з опрацюванням запитів від WWW – сервера.

Для реалізації такої технології необхідно використовувати взаємодію WWW – сервера з програмами CGI (Common Gateway Interface), що запускаються на сервері. Вибір програмних засобів достатньо широкий – мови програмування, інтегровані засоби типу генераторів звітів. Для СУБД із внутрішніми мовами програмування існують варіанти використання цієї мови з метою генерації документів.

3. Створення інформаційного сховища на основі високопродуктивної СУБД із мовою запитів SQL. Періодичне завантаження даних у сховище з основних СУБД. У цьому варіанті пропонується використання технології, що одержала назву «інформаційне сховище» (ІС). Для опрацювання різноманітних запитів, у тому числі і від WWW – сервера, використовується проміжна БД високої продуктивності, інформаційне наповнення якої здійснюється спеціалізованим програмним забезпеченням на основі вмісту основних баз даних: етап 1 – перевантаження даних; етап 2 – опрацювання запитів.

Даний варіант вільний від усіх хиб попередньої схеми. Крім того, після встановлення синхронізації даних інформаційного сховища з основними БД можливим є перенесення користувальницьких інтерфейсів на інформаційне сховище, що має істотно підвищувати надійність і продуктивність, сприяти організації розподілених робочих місць.

Незважаючи на громіздкість такої схеми, для задач забезпечення WWW – доступу до вмісту декількох баз даних накладні витрати істотно зменшуються.

Основою для підвищення продуктивності опрацювання WWW – запитів і різкого збільшення швидкості розробки WWW – інтерфейсів є використання внутрішніх мов СУБД інформаційного сховища для створення гіпертекстових документів.

З метою завантаження вмісту БД в інформаційне сховище можуть використовуватися всі рішення (мови програмування, інтегровані засоби), а також спеціалізовані засоби перевантаження, що поставляються з SQL-сервера і продукти підтримки інформаційних сховищ.

Класичну архітектуру **моделі пошукового механізму**, яку найчастіше реалізують на корпоративних сайтах та інформаційних порталах, зображено на рис. 1.

Існують клієнтська обчислювальна машина під керуванням ОС Windows і web-сервер під управлінням UNIX-подібностей ОС. На боці клієнта запущено типовий браузер, такий як Netscape, на боці сервера – web-сервер, котрий обслуговує запити від браузера, передаючи запити презентаційному шару, що розуміє CGI. Презентаційний шар передає запити до пошукового механізму в разі виклику послуги пошуку або відображує наповнення (content) сайта. Під час роботи адміністратора презентаційний шар також може передавати запити на ініціалізацію механізму індексації нового контенту, який ще не індексовано. Це необхідно, оскільки доки текст не індексовано, пошук у ньому за допомогою пошукової машини неможливий.

Ідея полягає в такому. Є мегабайти текстової інформації, і швидкість пошуку документів, що містять задані ключові слова, займає дуже багато процесорного часу. Припустімо, у 10 Мб текстової інформації пошук ключового слова триватиме протягом 10 с. Наприклад, заходить відвідувач на сайт, задає ключові слова, викликає послугу пошуку і чекає 10 с., поки сервер не видасть йому результат. Припустімо, сталося так,

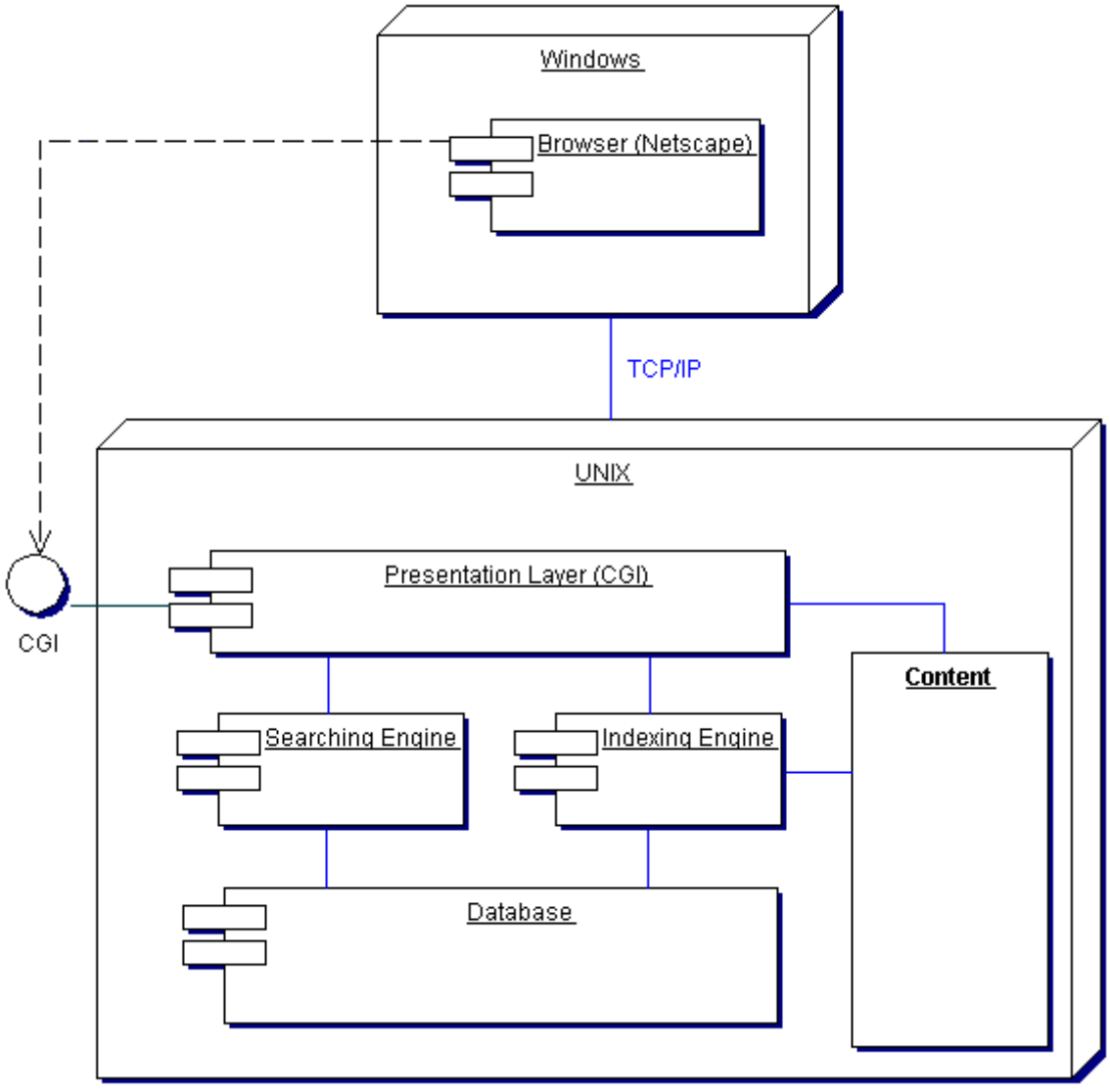


Рис. 1. Схема моделі архітектури пошукового механізму

що одночасно запит на пошук зробили п'ять осіб. Отже, час відповіді збільшиться вп'ятеро. Тобто в середньому по 50 с. користувач чекатиме відповіді від сервера. Це неприйнятно, надто коли в БД сотні мегабайтів текстової інформації і час реакції системи буде катастрофічно великим. Необхідно в разі пошуку ключових слів у текстовій інформації час відповіді скоротити до мілісекунд, а для цього – використати інший підхід.

ER-модель пошукового механізму

Існує така важлива характеристика баз даних, як дуже малий час вибірки конкретного запису з-поміж мільйонів інших. Це досягається за рахунок створення так званого індексу до таблиці на якесь із полів цієї таблиці. Зазвичай індекси реалізуються із застосуванням алгоритму збалансованого двійкового дерева (рис. 2).

У таблиці «document» зберігаються імена файлів або URL'и сторінок і кожному такому записові сто-

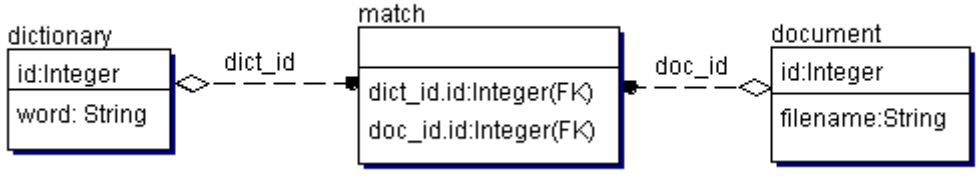


Рис. 2. Класична структура бази даних для вирішення проблеми забезпечення дуже малого часу вибірки конкретного запису

рінок поставлений у відповідність унікальний ключ «id». У таблиці «dictionary» зберігаються всі слова, які можуть зустрічатися в наших документах, і кожному слову відповідає унікальний «id». Природно, створюються індекси на полі word у таблиці «dictionary» і на полі «id» у таблиці «document», при цьому існує відношення багатьох до багатьох. Це необхідно, оскільки в таблиці «match» зберігається відповідність слова і документа. Інакше кажучи, в таблиці «match» зберігається інформація про те, які слова є в кожному документі. На таблицю «match» створюють індекс на поле «dict_id».

Індексний механізм

Перш ніж документи стануть доступними для пошуку, їх необхідно проіндексувати. Обсяг індексної інформації, який отримали з тексту, може вдвічі перевищувати обсяг власне тексту. А може й ще більше, якщо оптимально не використовуватиметься пам'ять. Алгоритм має такий вигляд.

1. Отримуємо документ для індексування.
2. Реєструємо його в таблиці «document», запам'ятовуємо отриманий його унікальний «id» і називатимемо його «doc_id».
3. Розбиваємо документ на окремі слова.
4. Визначаємо унікальні «id» цих слів з таблиці «dictionary» і називатимемо їх «dict_id».
5. Потім заносимо записи з нашим одним «doc_id» і різними «dict_id» (для кожного слова в документі) до таблиці «match».

Пошуковий механізм

Після проведення індексації документів треба з'ясувати, які запити відправляти до бази для пошуку цих документів за ключовими словами. Припустімо, є пошукова фраза «ріка Дніпро». Користувачеві необхідно отримати всі документи, що містять ці два слова. Спочатку треба звернутися до таблиці dictionary і знайти унікальні id цих слів, далі називатимемо їх $dict_id1 і $dict_id2. Потім необхідно послати такий запит до таблиці match, який видасть тільки ті номери документів, що містять ці два слова. Ось приклад цього запиту: SELECT doc_id FROM match where dict_id =$dict_id1 group by doc_id INTERSECT SELECT doc_id FROM match where dict_id=$dict_id2 group by doc_id. У тому разі, коли користувач введе три слова, доведеться додати ще раз INTERSECT і третю частину SQL запиту. За отриманими в результаті запиту doc_id можна здобути інформацію про ім'я файла документа з таблиці document.

Комплексне функціонування

Загальне уявлення про механізм взаємодії з пошуковою системою показано на рис. 3.

Як видно з рис. 3, є три потоки управління. Перший обслуговує запити користувача, другий виконує пошукові запити, третій займається індексуванням

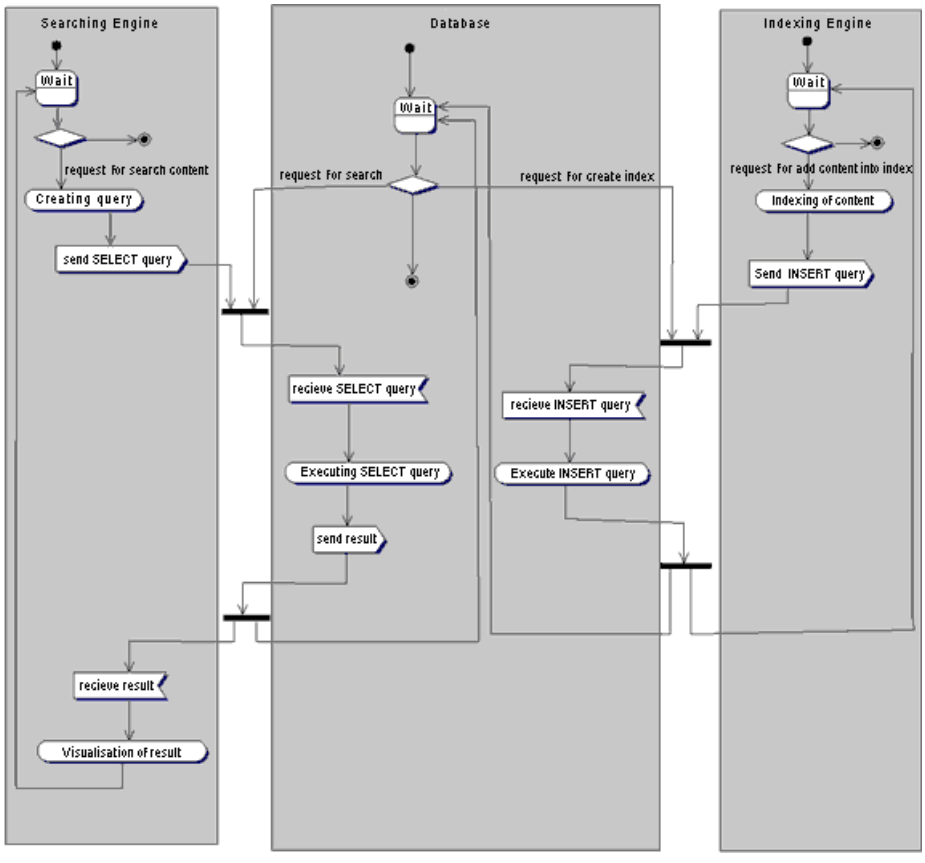


Рис. 3. Механізм взаємодії з пошуковою системою

нових документів, що надходять до системи. Перший потік – це скрипт на Perl, Servlet, ASP або PHP, який з ключових слів користувача формує пошукові SQL – запити; другий – це система управління базою даних, яка підтримує цілісність даних, індексний механізм і обслуговує SQL – запити; третій – це теж скрипт, що працює з новими документами, індексує їх і надсилає запити до бази даних на внесення нової індексної інформації.

Схема доступу до баз даних та електронних інформаційних ресурсів УкрІНТЕІ (рис. 4) складається з елементів:

1. *Шлюз* (Gateway):

а) виконує функцію маршрутизації:

• для віддалених користувачів Інтернет-порталу УкрІНТЕІ доступ до порталу www.uintei.kiev.ua за

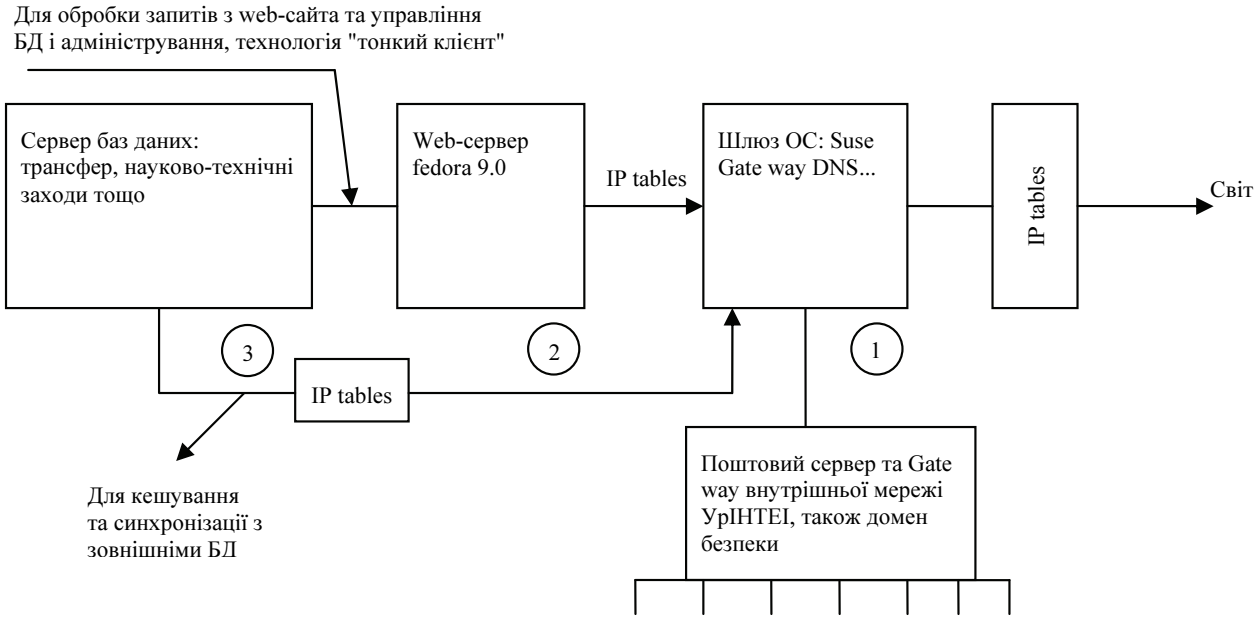


Рис. 4. Організація доступу до баз даних та електронних інформаційних ресурсів УкрІНТЕІ через Інтернет

допомогою протоколу HTTP крізь порт 80;

• для адміністраторів сайта, порталу, адміністраторів баз даних, та веб-сервісів (віддалених і локальних) – з локальної мережі крізь захищене з'єднання;

б) надає сервіс доменних імен (DNS):

• для сайта www.uintei.kiev.ua, всім користувачам Інтернету;

• для підсайтів УкрІНТЕІ, наприклад store.uintei.kiev.ua;

• для адміністраторів сайта, порталу, баз даних, та налаштувань веб-сервісів, наприклад dbadmin.uintei.kiev.ua;

в) є первинною системою безпеки мережі.

2. *Веб-сервер* – виконує роль посередника між віддаленим користувачем та інформаційними ресурсами, а також виконує функції системи захисту цілісності інформації для відкритого доступу. Складається з:

• веб-сервера Apache;

• програмного забезпечення (інтерпретатори мов програмування php, C, Perl … т.і, СУБД MySQL – для бази даних сайта та портала, а також необхідні компоненти (API) для доступу до багатьох різнорідних інформаційних ресурсів);

• Firewall та CronTab – захист та резервне копіювання і збереження інформації.

3. *Сервер баз даних* – програмний комплекс збереження, обробки та транспортування даних (рис. 5)

Глобальна БД індексів (ГБДІ) разом із програмним комплексом управління є ядром пошукового механізму сервера БД. Принцип роботи для порталу БД та EIP УкрІНТЕІ буде виглядати так:

Пошуковий запит від клієнта браузера на віддаленому комп'ютері обробляється веб-сервером та передається як SQL запит до ГБДІ сервера БД. У відповідь веб-сервер отримує перелік та опис записів (також ID бази даних, де є цей запис, та ID потрібного запису в цій БД) які задовольняють умовам пошуку. Веб-сервер генерує динамічну веб-сторінку, на якій користувач бачить список знайдених позицій у вигляді гіперпосилань. Вибравши одне з них, користувач натискає на гіперпосилання, веб-сервер формує запит до відповідного (або одного уніфікованого) веб-сервісу (обробника), який отримає з визначеної БД визначений запис.

Якщо користувач виконує пошук у конкретній локальній БД або БД в локальній мережі УкрІНТЕІ, щоб мінімізувати навантаження на ГБДІ, запит одразу йде до веб-сервіса (обробника) або (якщо БД підтримує

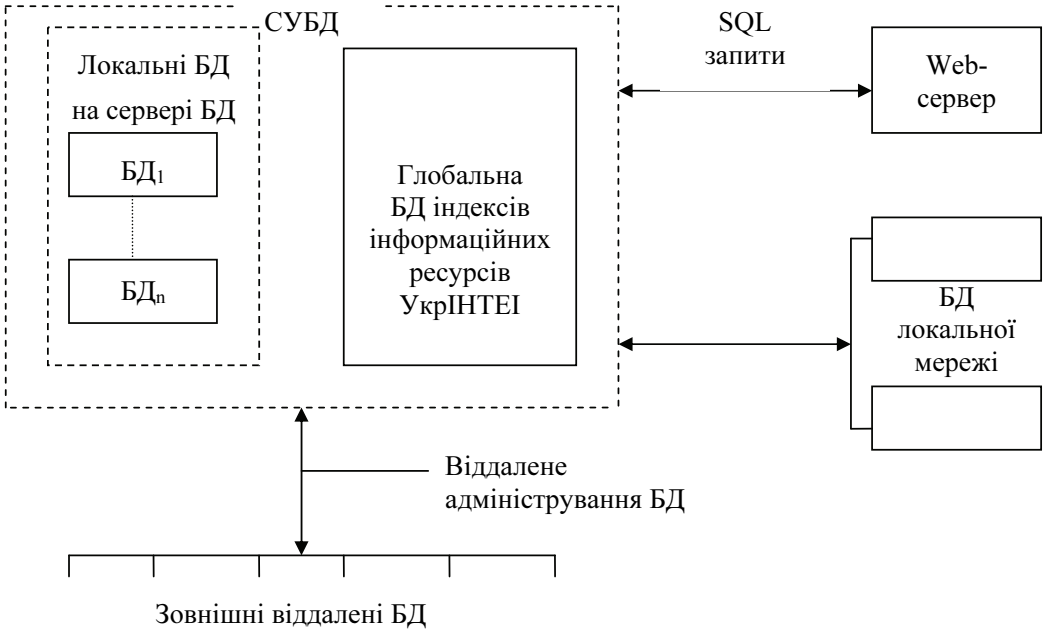


Рис. 5. Схема сервера баз даних

SQL запити) в напрямі до визначеної БД.

З віддаленими БД, розташованими поза мережею УкрІНТЕІ, існує ряд ускладнень:

• ODBC БД розташована на комп'ютері локальної мережі, де на сервері не прозорий проксі (схована за NAT). Вихід – встановлення на сервері веб-сервісу – клієнта для віддаленого пошуку інформації в БД;

• БД має формат що не підтримує одночасний розподільний доступ (трансакції). Вихід – перепроектування БД;

• надто вузький канал зв'язку «кволий Інтернет». Вихід – розташування БД на сервері БД УкрІНТЕІ і віддалене наповнення та адміністрування БД за допомогою технології тонкий клієнт через захищене (HTTPS) з'єднання.

Огляд СУБД для баз даних та ЕІР УкрІНТЕІ

Серед програмних продуктів для створення баз даних найбільшого поширення набули СУБД Microsoft Access, Microsoft SQL Server, Oracle, INGRES, Informix, DB2, Sybase, Paradox та ін. Ці програмні продукти надають користувачу широкий набір засобів для проектування і підтримки баз даних різного масштабу і призначення [5].

Ми будемо розглядати тільки дві СУБД:

1. Для локальних БД мережі УкрІНТЕІ та невеликих і простих за структурою віддалених БД – MySQL як визнаного лідера серед freeware програмного забезпечення.

MySQL – це багатопотоковість, підтримка декількох одночасних запитів, оптимізація зв'язків із приєднанням багатьох даних за один прохід, записи фіксованої і змінної довжини, ODBC драйвер у комплекті з вихідником, гнучка система привілеїв і паролей, до 16 ключів у таблиці, кожний ключ може мати до 15 полів. Також є підтримка ключових полів і спеціальних полів в операторі CREATE, підтримка чисел довжиною від 1 до 4, рядків змінної довжини і міток часу, інтерфейс з мовами C і Perl. Швидка система пам'яті, що ґрунтується на потоках, утиліта перевірки і ремонту таблиці, всі дані зберігаються у форматі ISO8859_1. Усі операції роботи з рядками не реагують на регістр символів в оброблюваних рядках, псевдоніми застосовні як до таблиць, так і до окремих колонок у таблиці, усі поля мають значення за умовчанням. INSERT можна використовувати на будь-якій підмножині полів. Слід відзначити також легкість керування таблицею, включаючи додавання і видалення ключів і полів. Можна виконувати команди SQL безпосередньо з командного рядка системи Unix або з інтерактивного режиму MySQL. СУБД MySQL має бібліотеку C API. Її можна використовувати для запитів до бази даних, вставляння даних, створення таблиць тощо. C API підтримує всі функції MySQL. Мова Perl підтримується відразу двома способами:

• портований інтерфейс з Perl з mini-SQL, розроблений Андреасом Коенігом;

• є модуль Perl DBD.

Також доступний 32-бітовий ODBC драйвер для MySQL. Він дає змогу робити запит і отримувати дані з інших джерел з підтримкою ODBC. Окрім технічних деталей можна додати, що MySQL працює як на Unix, так і на платформі Windows 95/98, він дуже простий і зручний у роботі.

2. СУБД для сервера баз даних УкрІНТЕІ – Cache – один із лідерів корпоративних СУБД, що чудово зарекомендував себе в УкрІНТЕІ.

Cache' Direct Access – прямий доступ до даних, забезпечує максимальну продуктивність і повний

контроль з боку програміста. Розробники додатків отримують можливість працювати безпосередньо зі структурами зберігання. Використання цього типу доступу висуває певні вимоги до кваліфікації розробників. Розуміння структури зберігання даних у Cache' дає змогу оптимізувати зберігання даних додатка і створювати надшвидкі алгоритми обробки даних.

Cache' SQL – реляційний доступ, що забезпечує максимальну продуктивність реляційних додатків з використанням вбудованого SQL. Cache' SQL відповідає стандарту SQL 92. Крім цього, розробник може використовувати різні типи тригерів і збережуваних процедур. Завдяки цьому Cache' успішно конкурує з реляційними СУБД. Навіть без використання прямого і об'єктного доступу додатка на Cache' працюють імовірніше за рахунок високої продуктивності сервера багатомірних даних.

Cache' Objects – об'єктний доступ. Для досягнення максимальної продуктивності розробки при використанні Java, EJB, C++, а також VB та інших ActiveX-сумісних засобів розробки, таких як PowerBuilder і Delphi. У Cache' реалізована об'єктна модель відповідно до рекомендацій ODMG (Група керування об'єктними базами даних – Object Database Management Group). У Cache' повністю підтримуються успадкування (у тому числі і множинне), інкапсуляція та поліморфізм. У разі створення інформаційної системи розробник отримує можливість використати об'єктно-орієнтований підхід до розробки, моделюючи предметну область у вигляді сукупності класів об'єктів, в яких зберігаються дані (властивості класів) і поведінка класів (методи класів). Cache', підтримуючи об'єктну модель даних, дає змогу природним чином використати об'єктно-орієнтований підхід як під час проектування (у Rational Rose) предметної області, так і під час реалізації додатків в ОО-засобах розробки (Java, C++, Delphi, VB). Постреляційна СУБД Cache' конкурує з об'єктними СУБД, значно перевершуючи їх за такими показниками, як надійність, продуктивність і зручність розробки.

Використання Cache SQL Gateway

У Cache реалізовано спеціальний механізм Cache SQL Gateway, що дає змогу звертатися до зовнішніх джерел даних за допомогою ODBC. Таким чином, цей механізм надає можливість програмам мовою Cache Object Script здійснювати звертання до зовнішніх реляційних баз даних. На відміну від Cache ODBC (JDBC тощо), які надають доступ до даних Cache для зовнішніх реляційних джерел, SQL Gateway – це засіб для здійснення взаємодії з реляційними джерелми даних у протилежному напрямку.

Використання Веб-сервісів у Cache

Веб-сервіс являє собою набір логічно пов'язаних функцій (методів), які можна програмно викликати через Internet (або Intranet). Таким чином, програми, написані різними мовами програмування, що функціонують на різних серверах під керуванням різних платформ, можуть звертатися до будь-якої програми, котра працює на іншому сервері (тобто до веб-свервісу), і використовувати відповідь, отриману від неї на своєму веб-сайті або додатку.

Веб-сервіси являють собою особливий вид веб-додатків для створення рівня бізнес-логіки і зв'язку різнорідних додатків на основі використання спільних стандартів, а також відкритих протоколів обміну і передавання даних. В основі технології веб-сервісів лежить мова XML eXtensible Markup Language – розширювана мова розмітки. Обмін даними між додатками здійснюється за допомогою стандартного протоколу HTTP і деяких інших Internet протоколів.

Висновки

Таким чином, Концепція дає бачення сучасного становища побудови систем доступу до баз даних та EIP через Інтернет, напрямів і перспектив її розвитку, методів і принципів її функціонування, забезпечення зручного та оперативного доступу до них через Інтернет на базі нових інформаційних технологій. Розроблення цього документа є реальним кроком на шляху до створення механізмів доступу до баз даних системи НТІ, визначення правил ефективного контролю і обмеження (у разі потреби) доступу до файлів баз даних, засобів безпеки програмного забезпечення

ЛІТЕРАТУРА

1. Д. Кнут Мистецтво програмування. Вибрані фрагменти, з українським перекладом (http://webdilo.korysne.info/ua/content/46.html)

2. Курзанцева Л. И. Об адаптивном интеллектуальном интерфейсе «пользователь – система массового применения» // Комп'ютерні засоби, мережі та системи : Зб. наук. пр. – К.: Ін-т кібернетики ім. В. М. Глушкова НАН України, 2008.– №7. – 7 стр. (http://www.nbuv.gov.ua/portal/natural/Kzms/2008/2008_st13.pdf)

3. Жданович О. Досвід застосування електронних баз даних в історичних дослідженнях (http://www.nbuv.gov.ua/portal/Soc_Gum/Igdu/2009_11/20.pdf)

4. Мельничин А. В. Моделювання та оптимізація доступу до інформації файлів баз даних: дис... канд. наук: 01.05.03 – 2009. (http://www.lib.ua-ru.net/diss/cont/353835.html)

5. Світличний О. О., Плотницький С. В. Основи геоінформатики (http://geoknigi.com/book_view.php?id=590)