

ЗАСТОСУВАННЯ *JAVA* – СЕРВЛЕТІВ ДЛЯ *OFFLINE*- ЗАВАНТАЖЕННЯ РЕСУРСІВ З МЕРЕЖІ *INTERNET*

На основі технології сервлетів розроблена програма для *offline*-завантаження ресурсів з мережі *Internet*.

The program for *offline*-loading of resources from the *Internet* network is developed on the basis of servlet technology.

Ще зовсім недавно *Internet* в Україні був доступний обмеженому колу громадян. Тепер ситуація змінилася – збільшилася кількість *Internet*-провайдерів, поліпшилася якість зв'язку. Але, на жаль, вартість послуг *Internet* усе ще досить висока для більшості наших співвітчизників. А для деяких бюджетних організацій якісний і дешевий зв'язок із мережею *Internet* залишається лише мрією, яка найближчим часом навряд чи збудеться. Багато хто вже звик працювати в умовах інформаційного голоду. Але що робити навчальним закладам масштабу Чернівецького національного університету, для яких підготовка сучасних фахівців залежить від можливості повноцінного спілкування майбутніх інженерів і їх викладачів із безмежним джерелом різноманітної інформації, яким є *Internet*. Особливо це стосується підготовки спеціалістів в області інформаційних технологій.

У цих умовах актуальним залишається питання про більш оптимальне використання можливостей існуючих з'єднань із мережею *Internet*. Одним з методів більш повного використання *Internet*-підключень в умовах великої кількості користувачів і невеликих швидкостей ліній, є більш ефективне використання часу, коли навантаження на лінії зв'язку і на *Internet*-сервери мінімальні (нічний час, вихідні й святкові дні). Цей час можна використовувати для неінтерактивного завантаження користувачам *Web*-сторінок і файлів.

В даний час з метою економії часу і грошей при завантаженні файлів із *Web*-сайтів можна скористатися існуючими спеціальними програмами – диспетчерами завантаження. Найвідоміші з них – *GetRight*, *Gozilla*, *JetCar*, *Bullet Proof FTP*. Ці програми, як правило, працюють без догляду

користувача. Після вибору сервера зі списку серверів, з якими існує зв'язок, із нього копіюються зазначені файли на комп'ютер користувача. Деякі з цих програм дозволяють вказувати зручний час завантаження файлів із *Internet*. Використання цих програм – ідеальне рішення для домашнього користувача, який має комутоване з'єднання, із провайдером *Internet*. Але для такої організації, як Чернівецький національний університет, із його розгалуженою структурою комп'ютерних мереж та специфікою роботи комп'ютерних лабораторій та *Internet*-центра застосування даних програм неефективно з таких причин:

- комп'ютери (у випадку комп'ютерних лабораторій і обчислювальних центрів) постійно змінюють користувачів, при цьому неможливо точно визначити належність завантажених файлів визначеному користувачеві;
- комп'ютери велику частину часу доби виключені і недоступні користувачам;
- неможливість централізованого адміністрування і керування процесом завантаження *Web*-ресурсів.

Виходом із цієї ситуації є впровадження в *Internet*-центрі університету спеціального сервісу по завантаженню файлів з *Internet*. Для цього розроблений пакет програм *OSWD* (*Offline System Web Download*).

Ідея неінтерактивного завантаження полягає в тому, що клієнт отримує необхідний йому ресурс з Інтернету не в поточному сеансі роботи з браузером, а після заповнення спеціальної форми запиту, відсилки її на сервер. На сервері запит реєструється в базі даних запитів і при першій можливості виконується. Замовлений ресурс клі-

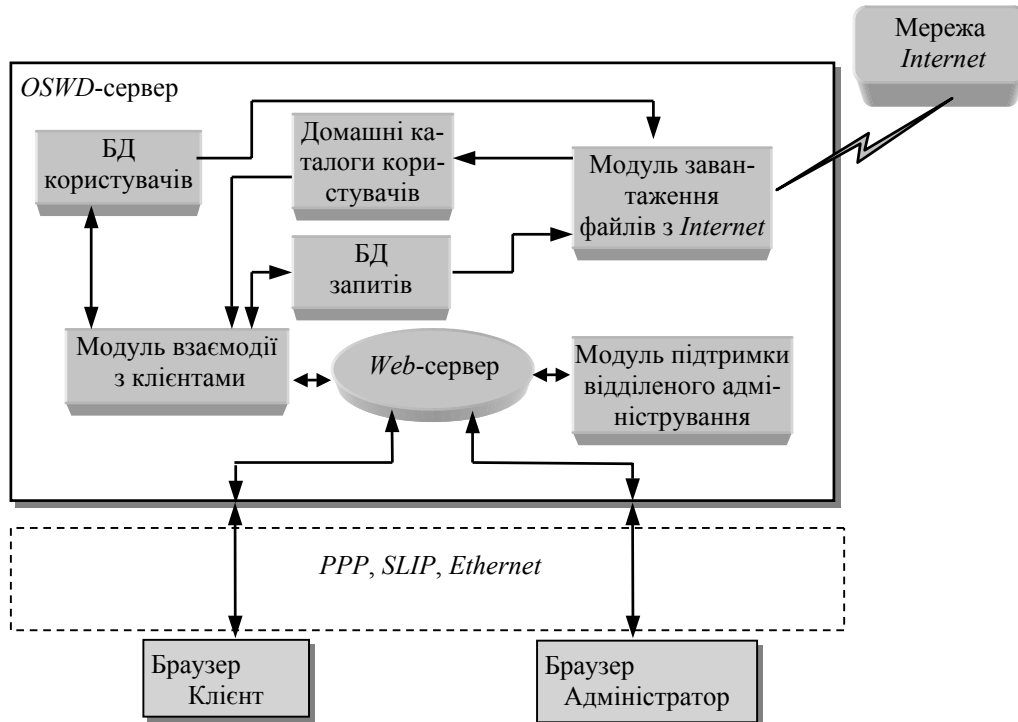


Рис.1. Структурна схема системи OSWD.

ент отримує через певний час, необхідний серверу для завантаження цього ресурсу із мережі Інтернет, і у вигляді архівного файлу, що зменшує навантаження на мережу під час пересилки інформації між сервером і клієнтом.

Пакет програм розроблений із застосуванням технології клієнт-сервер, тобто вся функціональність розділена між клієнтською та серверною частинами. У нашому випадку це виглядає як взаємодія клієнтських програм на машинах користувачів, що дають запити на завантаження файлів із Internet та серверної частини, яка централізовано здійснює обробку запитів, що надходять і завантажує інформацію з Web-серверів у придатний час.

Структурна схема

Структурна схема системи OSWD Suite представлена на рис.1.

Серверна частина складається з набору сервісних програм, що виконують такі функції:

- реєстрація та введення бази даних користувачів;
- установка з'єднань і підтримка діалогу з клієнтами;
- обробка запитів, що надійшли, і завантаження файлів із Internet;
- реалізація деяких функцій проксі-сервера, а саме:
 - заборона на завантаження певних ресурсів чи

файлів із визначених Web-сайтів;

- кешування інформації, що завантажується;
- підтримка пріоритетів для категорій користувачів (адміністратори, викладачі, студенти та ін.);
- підтримка віддаленого адміністрування системи з будь-якого комп'ютера, який має зв'язок із сервером.

Клієнтська частина являє собою звичайний браузер, що зазвичай встановлений на кожній робочій станції. Отже усувається проблема розповсюдження клієнтської програми та захист від підробленої копії.

Клієнт взаємодіє з сервером через Web-браузер, у вікні якого відображається html-сторінка, яка заключає в собі функціональність клієнтської частини. Сторінка динамічно генерується відповідними сервлетами і JSP-сторінками (Java Server Page), що входять до складу Web-сервера. Вони дозволяють користувачеві при наявності облікового запису зареєструватися на OSWD-сервері за допомогою введення ідентифікаційного імені користувача та його пароля, а після процедури аутентифікації виконувати такі дії:

- переглянути стан попередніх запитів;
- перенести завантажені файли з OSWD-сервера на локальну машину;
- сформувавати та відправити серверу новий запит;
- знищити на сервері запит або архівний файл із завантаженою інформацією.

При відсутності облікового запису користувач може створити його. Для цього необхідно заповнити невелику форму-анкету і відіслати її на сервер. Анкета заповнюється в присутності довірливої адміністратором *OSWD*-системи особи, яка вводом свого паролю надає чинності реєстрації користувача. Це забезпечує реєстрацію користувача тільки один раз та захист від несанкціонованого втручання.

При правильному заповненні форми сервер автоматично створює обліковий запис даного користувача і заносить його в базу даних користувачів.

Користувацький інтерфейс програми інтуїтивно-зрозумілий і дозволяє без попередньої підготовки приступити до роботи.

Для зв'язку клієнтів із сервером використовується стек протоколів *TCP/IP*, оскільки цей набір протоколів підтримує об'єднання окремих локальних мереж із різноманітним мережевим обладнанням. Як і в більшості додатків *Internet* як реалізація транспортного рівня використовується протокол *TCP*, тому що він забезпечує надійний зв'язок на основі логічного з'єднання.

Сеанс зв'язку з кожним користувачем виконується в окремому програмному потоці *Thread*. Цим забезпечується стабільність роботи сервера й незалежність окремих користувацьких сесій один від одної. Переваги технологій *servlet* і *JSP* полягають у тому, що окремі програмні потоки, в яких виконується взаємодія з окремим клієнтом системи, організується автоматично *Web*-сервером, що має модуль підтримки *servlet* і *JSP*. Крім того, на відміну від *CGI*-скриптів, що являють собою окремий процес, який щоразу запускається при звертанні до нього, байт-код сервлетів та скомпільованих *JSP* постійно знаходиться в пам'яті, чим забезпечується швидкість обслуговування запитів.

Кожний зареєстрований користувач має на сервері домашній каталог, в якому зберігаються файли – результати запитів. Максимальний розмір каталога обмежується (5-20 МВ) і визначається пріоритетом користувача.

Модуль завантаження файлів з *Internet*, використовуючи базу даних запитів, створює із них пріоритетну чергу, порядок слідування запитів в якій визначається пріоритетом користувача і часом створення запиту. Моментом початку та закінчення роботи модуля керує спеціальний програмний блок – планувальник часу завантаження. При активізації ним модуля завантаження, пріоритетна черга циклічно сканується. Запити, які

ще не були реалізовані, інкапсулюються в окремі об'єкти спеціального класу, що мають властивість виконуватися в окремому програмному потоці, та запускаються на виконання. Завдяки цьому стало можливим виконання декількох запитів одночасно. У відповідності до запиту відповідні файли завантажуються з мережі Інтернет у домашні каталоги користувачів та архівуються зовнішнім архіватором.

Технічна реалізація

Програма написана на популярній сьогодні мові програмування *Java*. Вибір був зроблений на користь *Java* із таких причин:

- це цілком об'єктно-орієнтована мова, що дозволяє розробникам програмного забезпечення створювати складні додатки в першу чергу для використання в *Internet*;

- створені додатки не залежать від платформи;

- у *Java* вбудована підтримка потоків, засоби для роботи в мережі й безліч інших можливостей.

Серверна частина створена за допомогою *Java SDK 1.3* фірми *Sun* і може працювати на будь-якій платформі, для якої існує *JVM* (наприклад, *Windows 2000 Server*, *SCO UnixWare 7.01*, *Linux*, *FreeBSD 4.0*, *Solaris* та ін.). Для роботи сервера необхідне постійне підключення до мережі *Internet* і проінстальована *JVM*.

Особливих вимог до апаратного забезпечення комп'ютерів, на які встановлюються програми, не існує. Для клієнтської комп'ютерів може використовуватися будь-який *IBM*-сумісний комп'ютер, на якому може працювати *Web*-браузер.

Серверну частину необхідно встановлювати на комп'ютер із процесором не менше PIII-350 і 128 МВ оперативної пам'яті. Також потрібно враховувати необхідність у вільному дисковому просторі для каталогів користувачів (5-10 МВ на кожного користувача).

Опис організації вхідних та вихідних даних

Вхідні дані для даного додатка – інформація про користувачів та про їх запити на завантаження файлів із мережі Інтернет. Всі користувацькі дані формуються під час сеансу взаємодії клієнта системи з сервером в інтерактивному режимі. Інформація зберігається у звичайних текстових файлах. Назва файлу з БД користувачів – *users.db*, *queries.db* – назва файлу з БД користувацьких запитів.

Формат файлів визначається специфікацією відповідних класів: *OSUser* та *Simple Query*. Вихідними даними додатка *OSWD* для користува-

чів є архівні файли (наприклад, ".zip"), в яких знаходяться результати завантаження. Тип архівного файла визначається типом зовнішнього архіватора, що підключається до системи.

Основним файлом настройки, який забезпечує роботу OSWD-сервера, є файл сценарію завантаження *schedule.cfg*, в якому зберігається інформація про час активації та дезактивації модуля завантаження. Цей файл створюється і поновлюється автоматично під час сеансу віддаленого адміністрування адміністратора системи із сервером.

Висновки

Швидкість та ефективність роботи програми залежить від багатьох чинників. Головне – це швидкість і завантаженість мережі, що з'єднує комп'ютер, на якому встановлено OSWD-сервер, з Інтернет-провайдером. Додаткові фактори, що посередньо впливають на продуктивність процесу завантаження – це розмір оперативної пам'яті та потужності центрального процесора сервера, на якому встановлена система OSWD.

Програма була розроблена і протестована в середовищі *Windows 2000 Server* з *Web*-сервером

Apache та додатком для запуску сервлетів та *JSP*-сторінок – *jakarta-tomcat*.

Додаток при регулярному використанні дозволяє значно підвищити ефективність використання існуючих з'єднань із мережею Інтернет. І поки в нашій державі не буде побудована могутня інфраструктура глобальних мереж (*ISDN*, *xDSL* та ін.) і не буде достатньої кількості конкуруючих між собою провайдерів, що надають швидкісний і дешевий доступ у *Internet*, розробка подібних програмних продуктів буде актуальною для великих розгалужених організацій, подібних Чернівецькому Національному університету.

СПИСОК ЛІТЕРАТУРИ

1. Вебер Д. Технология Java в подлиннике.: Пер. с англ. СПб.: С-Петербург, 2000.
2. Линн Кайл. Flash 5. Web-профессионалам: Пер. с англ. - К.: Издательская группа BHV, 2001.
3. Морган М. Java 2. Руководство разработчика.: Пер. с англ. М.: Издательский дом "Вильямс", 2000.