Міністерство освіти і науки України

Чернівецький національний університет

імені Юрія Федьковича

Факультет комп’ютерних наук

Кафедра комп’ютерних систем та мереж

**НОВИЙ ПІДХІД МОДЕЛЮВАННЯ ЗАВАНТАЖЕНОСТІ SQL-СЕРВЕРІВ**

**482.362.70915-05 81 51-3**

**(курсова робота)**

2007

**Анотація**

В пояснювальній записці подано загальний аналітичний огляд засобів аналізу трафіку, SQL-серверів та Web-серверів, які будуть використовуватися в даній роботі, опис алгоритмів функціонування класів, що були використані при розробці. Приведено повний опис вхідних та вихідних даних програми, очікуваних техніко-економічних показників.

Пояснювальна записка містить 19 сторінок друкованого тексту та 2 рисунка.

**Зміст**

Вступ

1. Призначення та область застосування

2. Технічні характеристики

2.1 Постановка задачі на розробку програми

2.2 Опис алгоритму і функціонування програми

2.3 Опис організації вхідних та вихідних даних

2.4 Опис організації вибору технічних і програмних засобів

3. Очікувані техніко-економічні показники

Список літератури

# Вступ

Тема роботи “Новий підхід моделювання завантаженості SQL-серверів”.

Завдання на роботу 482.362.70915-05 51-3 затверджене на кафедрі КСМ факультету комп’ютерних наук Чернівецького національного університету. Протокол засідання кафедри КСМ №1 від 28 серпня 2006 року.

**1. Призначення та область застосування**

З самого початку розвитку обчислювальної техніки утворилися два основних напрямки її використання. Перший напрямок – застосування обчислювальної техніки для виконання чисельних розрахунків, які доводилось робити занадто довго або взагалі неможливо виконати вручну. Становлення цього напрямку сприяло інтенсифікації методів чисельного рішення складних математичних задач, розвитку класу мов програмування, орієнтованих на зручний запис чисельних алгоритмів.

Другий напрямок – це використання засобів обчислювальної техніки в автоматичних або автоматизованих інформаційних системах. У самому широкому змісті інформаційна система являє собою апаратно-програмний комплекс, функції якого складаються в підтримці надійного збереження та обробки інформації в пам’яті комп’ютера, виконанні специфічних для додатків перетворень інформації або обчислень, наданні користувачам зручного і простого інтерфейсу.

Звичайно об’єми інформації, з якими приходиться мати справу таким системам, досить великі, а сама інформація має досить складну структуру. Класичними прикладами інформаційних систем є банківські системи, системи резервування авіаційних або залізничних квитків, місць у готелях та ін.

Інформаційні системи головним чином орієнтовані на збереження, вибір і модифікацію постійно існуючої інформації. Структура інформації найчастіше дуже складна, і хоча структури даних різні в різних інформаційних системах, між ними часто буває багато загального. На початковому етапі використання обчислювальної техніки для керування інформацією проблеми структуризації даних вирішувалися індивідуально в кожній інформаційній системі. Прагнення виділити й узагальнити загальну частину інформаційних систем, відповідальну за керування складно структурованими даними являлось першою спонукальною причиною створення СКБД.

В даній роботі здійснена спроба реалізувати модуль відображення завантаженості мережі, який відображає в графічному вигляді завантаженість локальної мережі між Web та SQL серверами. Обчислення даних завантаженості проводиться в період проведення визначеного користувачем тесту. Модуль можна використовувати в системі тестування для найрізноманітнішого відображення інформації у вигляді графіків, діаграм тощо. Модуль забезпечує більш зручний збір інформації про проведений системою тест.

# 2. Технічні характеристики

## 

## 2.1 Постановка задачі на розробку програми

В даній дипломній роботі необхідно розробити сукупність класів, які б забезпечували об’єднання двох елементів(системи тестування та програми-аналізатора трафіку), для подальшого графічного відображення завантаженості мережі. При постановці задачі було проведено деякі обмеження, які включали в себе:

* + Структура, принципи функціонування, взаємозв’язок між класами системи тестування SQL-серверів відома.
  + Для отримання даних про вхідний та вихідний трафік використати одну з зовнішніх безкоштовних програм.
  + Здійснювати відображення вхідного, вихідного та загального трафіку в текстовому та графічному вигляді.
  + Обмеження при виборі Web-сервера, яке призначене для огляду тих Web-серверів, котрі найбільш повно реалізують можливості мови програмування Java.

Аналіз даних обмежень наведено в наступних розділах.

## 2.2 Опис алгоритму і функціонування програми

Опис алгоритму роботи структури класів буде включати опис наступних елементів:

* Алгоритм роботи класу MyChart2.
* Алгоритм роботи класу ChartServlet.
* Алгоритм роботи класу ParseData.

Клас MyChart призначений для встановлення початкових параметрів для відображення, генерації об’єкту малюнку з вміщеними в нього даними та переправлення його для подальшої обробки.

У відповідності до цього, в даному класу існує 4 функції, які призначені для реалізації вищевказаних функцій.

Функція produceChart одна з основних, і призначена для визначення інтервалу часу, протягом якого проводився деякий тест, та подальшого виклику функції createChart.

Функція createChart призначена для створення графічних елементів вікна, які необхідні для графічного відображення сітки координат, а також самої кривої. Додаткові функції setTime та setColor призначені для встановлення внутрішніх змінних та полів класу, які необхідні для роботи класу.

Початок

Дані про час

Обробка

даних про час

Встановлення параметрів

createChart

Повернення графіка

Кінець

Рис. 2.1 Алгоритм роботи функції produceChart.

Під час обробки даних про час відбувається розклад даних про точку початку та точку закінчення проходження тесту. В подальшому проводиться встановлення параметрів малювання графіку таких, як колір лінії та кількість ліній на графіку.

Після встановлення всіх необхідних параметрів відбувається виклик функції createChart, яка проведе генерацію графіку по заданих даних, після чого, графік буде повернуто у точку виклику. Вказана точка виклику знаходиться в класі ChartServlet.

Даний клас призначений для виконання послідовних дій для отримання графіка завантаженості, та виводу даного графіку у вигляді малюнку на HTML-сторінці. Послідовність дій, які виконує даний сервлет включає в себе:

1. Визначення з параметрів запиту тип графіку, який необхідно вивести користувачу.
2. Запуск класу, що отримує початкові параметри для відображення для заданого типу графіку з xml-файлу.
   1. створення екземплярів класів, які необхідні для розбирання xml-файла;
   2. створення класів для читання атрибутів;
   3. пошук атрибутів для вхідного типу класу;
   4. повернення атрибутів викликаю чому об’єкту.
3. Обробка початкових параметрів.
4. Створення екземпляру класу MyChart2.
5. Збереження отриманого графіку в тимчасовій директорії у вигляді файлу.
6. Передача файлу користувачу.
7. Знищення тимчасового файлу графіку.

Для спрощення внесення змін або обновлення параметрів відображення графіків, добавлення нових типів графіків тощо, використовується xml-файл. Приклад вказання параметрів для типу графіка, що буде відповідати за відображення вхідного завантаження мережі буде мати вигляд:

**. . .**

<chart

id="MyChart2"

class="Additional.MyChart2"

width="780"

height="480"/>

**. . .**

Проте, вищеописані класи працюють з вже існуючою інформацією про координати точок графіка. Ці створення цих часових залежностей і призначений клас ParseData. Робота класу заключається в обробці даних лог-файлу, який створюється зовнішньою програмою аналізу трафіку. При цьому в класі здійснюється обробка тільки тих рядків файлу, які відповідають нашому заданому періоду тестування. Це дозволяє проводити обробку лог-файлу одночасно з внесенням змін в нього самою програмою-аналізатором.

При проведенні аналізу переданих даних, клас ParseData проводить також аналіз даних, які вказують на те, які дані були передані - вхідні (вхідний трафік) чи вихідні (вихідний трафік). Після цього відбувається перерозподіл даних згідно їх приналежності до однієї з трьох структур:

1. серії вхідного трафіку;
2. серії вихідного трафіку;
3. серії загального трафіку.

Початок

Початкові дані серії

Відкрити лог -файл

Прочитати рядок даних

Отримати дані дати

x>t2

Закінчення циклу

t1<x<t2

Формування точки графіку

Кінець

Рис. 2.2. Алгоритм роботи методу equalsDate класу ParseData.

## 2.3 Опис організації вхідних та вихідних даних

При виборі організації вхідних даних розглядалися різні способи їх представлення. Однак найоптимальнішим вибором вхідних даних буде розміщення цих даних у вигляді конфігураційних файлів типу xml. Це дозволить підвищити простоту зміни конфігурації відображення графіків. При цьому робота програми залишиться сталою, і зменшиться вплив зовнішніх даних на сам процес тестування та формування результатів проведених тестів.

Ще один тип вхідних даних – вихідні дані програми-аналізатора мереженого трафіку. Ці дані представляються у вигляді текстового файлу з наступним форматом рядків:

1. дата створення запису - 11.06.2007;
2. час створення запису - 22:30:31:;
3. хто надіслав дані та номер порту – comp1:445;
4. кому передано дані і номер порту - myadmin.atpi.group:2630;
5. кількість інформації - 1514 bytes;
6. версія протоколу - protocol: 6.

Як бачимо, рядок файлу містить всю необхідну інформацію про передані чи отримані дані. Аналізуючи вміст даного рядка з допомогою об’єкту StringTokenizer, ми виділяємо наступні необхідні для нас елементи:

1. дата створення запису;
2. час створення запису;
3. хто надіслав дані;
4. кількість інформації;

Перші два поля необхідні для визначення чи належать дані цього рядка до періоду проведення тесту. Третє поле необхідне для визначення типу трафіку: вхідний чи вихідний. Четверте – для отримання координати Y деякої точки.

Крім того, оскільки модуль входить в склад системи проведення тестування, то до вхідних даних належать також файли конфігурації SQL-серверів та файли з тестовими запитами.

Організація вихідних даних включає в себе як результат тестування, так і сервісні повідомлення системи Web-додатків. Результати тестування будуть відображатися користувачу в наступному вигляді:

1. Назва сервера, для якого проводилось тестування.
2. Тип виконаного тесту.
3. Кількість потоків.
4. Середній час обробки одного запиту (в мілісекундах).
5. Загальний час проведення тесту.
6. Графічні зображення завантаженості мережі.

Така організація вихідних даних дозволить користувачу отримати достатньо повну інформацію про проведення одного тестування. При цьому враховується той факт, що при кожному тестуванню необхідно встановлювати всі початкові параметри в їх початкове значення. До вихідних даних можна також віднести файли з графіками та текстові файли, які користувач може створити з допомогою використання кнопок, що призначені для збереження отриманих даних вказаних типів.

## 2.4 Опис організації вибору технічних і програмних засобів

Вибір технічних засобів, які використовувалися для виконання дипломної роботи обумовлений технічним завданням та вимогами до технічних параметрів засобів, на яких встановлюється відповідні СКБД.

Для коректної роботи розроблених Web-додатків необхідна наявність наступного програмного забезпечення:

1. Сервера веб-додатків Jakarta Tomcat 5.0.
2. Встановлених SQL-серверів, які будуть проходити процес тестування.
3. Операційних систем типу Windows NT.
4. Будь-якого браузера.
5. J2SE Development Kit Update 2.
6. J2SE Runtime Environment Update 2.
7. JDBC-драйвери відповідних SQL-серверів.
8. Редактор текстових файлів Notepad.

Вимоги до обладнання, на якому буде функціонувати Web-сервер:

1. Процесор з частотою 2000MHz і вище.
2. Об’єм оперативної пам’яті: 512 Мб.
3. Об’єм жорсткого диску: від 8 Гб.
4. Мережна карта стандарту Ethernet/Fast Ethernet.

Така мінімальна конфігурація Web-сервера вибрана на основі спостережень за використанням пам’яті та файлу підкачки. При тестуванні в екстремальних режимах (50 та вище паралельних потоків) відбувається значне збільшення часу виконання тестування за рахунок недостачі ОЗП для віртуальної машини Java, що приводить до великого збільшення файлу підкачки. Таке збільшення файлу підкачки в свою чергу призводить до збільшення часу тестування, оскільки читання з диска є набагато повільнішим ніж читання з оперативної пам’яті.

Для роботи сервера баз даних необхідна наступна конфігурація:

1. Процесор з частотою 2000 MHz і вище.
2. Об’єм оперативної пам’яті від 1 Гб.
3. Об’єм жорсткого диску від 10 Гб.
4. Мережна карта стандарту Ethernet/Fast Ethernet.

Такі вимоги сформульовані після аналізу вимог вказаних SQL-серверів до складу технічних засобів, на яких вони можуть працювати. При цьому було обрано спільний показник з мінімальних вимог до кожного із серверів, щоб поставити всі сервера в однакові умови. Найвищий показник у вимогах до технічних засобів був у SQL-сервера Oracle.

# 3. Очікувані техніко-економічні показники

Проведемо оцінку витрат праці на розробку програмного забезпечення для комплексу, при цьому виходимо з того, що розмір вихідного тексту запису алгоритму і даних в основному визначає затрати праці  та час розробки  програмного продукту:

 (1)

де  — кількість вихідних команд в тисячах.

В якості вихідної команди приймаємо рядок програми. Загальний об'єм вихідного тексту програм складає приблизно 900 рядків. Тоді:



Продуктивність праці розробників програмного забезпечення визначається наступним чином:

 (2)

Тоді ми отримуємо продуктивність праці розробників: 305,5.

Час необхідний для розробки програмного продукту, можна визначити за формулою (3):

 (3)

де  — строк розробки програмного продукту;

 — коефіцієнт вірності постановки завдання;

 — час розробки алгоритму;

 — час настройки та тестування;

 — час на підготовку тексту;

 — час на розробку документації.

Останні величини обчислюються за наступними формулами:

 (4)

 (5)

 (6)

 (7)

 (8)

 (9)

де К — залежить від ступеня підготовки програміста;

 — кількість рядків програми (в тис.).

Підставляємо значення у формули (4-9) враховуючи, що К=0,8 (стаж роботи до 2-х років).













Тоді час, який необхідний для розробки програмного продукту дорівнює:



Визначимо собівартість години роботи ПК. Для цього розраховуються поточні витрати на експлуатацію комп'ютера. До їх складу включаються витрати на електроенергію і амортизаційні відрахування на реновацію від вартості комп'ютера та інше.

Витрати на електроенергію визначають множенням витрати електроенергії за одну годину на вартість 1 кВт/год електроенергії і на час роботи комп'ютера за рік. Час роботи комп'ютера за рік визначається множенням кількості робочих днів у рік на час роботи комп'ютера за день:

 (10)

де  — середня кількість робочих днів у рік.

Середня кількість робочих днів у рік буде рівна:



Тоді, час роботи ПК за рік дорівнює:



Витрати енергії визначаються за формулою:

 (11)

де  — витрати електроенергії за одну годину;

 — вартість 1 кВт/год електроенергії;

 — час роботи комп’ютера за рік.

Тоді витрати енергії складають:



Амортизаційні відрахування визначаються множенням вартості комплексу на норму амортизаційних відрахувань 10%:



Приблизна річна заробітна плата обслуговуючого персоналу складає:



Відрахування на соціальне страхування, складають 3% від загальної заробітної плати за рік:



Вартість витрачених матеріалів складає 2% від вартості обчислювальної техніки:



Утримання на ремонт приміщень, в яких знаходяться засоби обчислювальної техніки, складає 3% від вартості обчислювальної техніки:



Кількість комплексів, що реалізовують зв’язок: 

Собівартість години роботи на комп'ютері визначається наступним співвідношенням:

 (12)

Підставляємо значення в останню формулу й отримаємо:



Рахуємо прямі витрати на виконання дипломної роботи, які визначаються наступним добутком:

 (13)

де  — собівартість години роботи на комп’ютері;

 — час необхідний для розробки програмного продукту.

Підставляємо значення й одержуємо:



Накладні витрати, що включають витрати на освітлення, опалення та ін. приймаються в розмірі 40-50% від суми прямих витрат:



Загальні витрати на виконання дипломної роботи:

 (14)

Підставляємо дані й одержуємо:



Ціна на програмний продукт визначається наступним співвідношенням:

 (15)

де В — витрати на виконання дипломного проекту;

P — рівень рентабельності, в нашому випадку P = 30;

K — коефіцієнт, що залежить від науково-технічного рівня, в нашому випадку К = 0,9.

Підставляємо ці значення й отримаємо ціну програмного продукту, яка дорівнює:



**Список літератури**

1. Вебер Д. Технология Java™ в подлиннике: Пер. с англ. – СПб.: BHV, 2000. – 1104.
2. Эккель Б. Философия Java. Библиотека програмиста. – СПб: Питер, 2001. – 880.
3. Патрик Нортон, Герберт Шилдт Полный справочник по Java.: Пер.с англ. – К.: Диалектика, 1997. – 592.
4. Gosling James, Joy Bill, Steele Guy The Java Language Specification

ISBN 0-201-63451-1- 435.

1. Lindholm Tim, Yellin Frank The Java Virtual Machine Specification ISBN 0-201-63452-X – 856.
2. JavaTM 2 SDK, Standard Edition Documentation version 1.4.2 – 965.
3. Холл М. Сервлеты и JavaServer Pages. Библиотека програмиста. – СПб.:Питер, 2001. – 496.
4. JavaServer Pages™ Specification version 2.0. - Sun Microsystems, 2003. – 765.
5. Грофф Дж., Вайнберг П. SQL: Полное руководство: Пер.с англ. – 2-е изд., К.: BHV, 2001. – 816.
6. Бобровски Стивен. Oracle7 и вычисления клиент/сервер. – М.: ЛОРИ, 1995. - 670.
7. Уинкуп С. Microsoft SQL Server 6.5 в подлиннике: пер. С англ. - СПб.: BHV, 1998. - 145.