

WWW - СЕРВЕР APACHE С ПОДДЕРЖКОЙ JAVA SERVLET

Бул макалада жаңы түзүлгөн MIME модулу каралган. MIME модулу берилген маалымат базасы менен иштөөгө ылайыкташкан, ал эки секунда ичинде 8300 берилген маалыматты карап чыгуу мүмкүнчүлүгү бар.

В данной статье рассмотрен созданный модуль MIME, который используется в работе с базами данных, хорошей чертой которого является обработка 8300 записей за две секунды.

In given article considered created module MIME, which is used in work with the databases and which has good advantage is processing of 8300 records during two seconds.

Глобальная информатизация общества приводит к тому, что потребность в информации, растет с каждым новым пользователем сети. При этом задачей специалистов в области информационных технологий обеспечить пользователей полной и достоверной информацией путем простого и удобного для пользователей доступа к накопленным массивам данных, целью моей магистерской работы являлось обеспечение доступа к базам данных в сервере за минимально короткие сроки. Известно что, самый распространенный Web-сервер в мире - это Apache. По данным компании Netcraft (15) общее число Web-узлов, работающих под его управлением, к концу 1998 г. достигло 2 млн. (55% общего числа узлов) и постоянно растет. Для сравнения: на долю серверов Microsoft приходится 25%, Netscape -7%. Будучи бесплатной открытой программой, предназначенной для бесплатных же Unix-систем (FreeBSD, Linux и др.), Apache по функциональным возможностям и надежности не уступает коммерческим серверам, а широкие возможности конфигурирования позволяют настроить его для работы практически с любой конкретной системой. Существуют локализации сервера для различных языков. Для того чтобы использовать Apache во взаимодействии с Java, необходимо было установить продукт Jserv, содержащий необходимые библиотеки Java-классов, затем изменить конфигурационный файл Apache Configuration, добавив в него строку: Module jserv_module mod_jserv.o - для Unix-системы.

При реализации сценария встал вопрос о выборе качественной платформы для создания информационного хранилища. Реляционная система управления базами данных фирмы Oracle является лидером на рынке СУБД. По производительности, надежности хранения данных, развитию семейства интерфейсов, объему серверных платформ продукты Oracle возглавляют многочисленные рейтинги. Гибкость использования, развитые средства управления доступом и распределенная архитектура делают сервер Oracle чрезвычайно привлекательным для технологии информационных хранилищ, а возможность работы на свободно - распространяемых Unix-платформах расширяет его возможности в некоммерческой среде. Существенным ограничением использования Oracle в сфере науки и образования является достаточно высокая цена и низкое бюджетное финансирование.

Рассмотрим принцип работы сервера. В общем виде схема работы WWW-сервера выглядит так:

1. Пользователь сети запускает пакет программного обеспечения, называемый браузером, в функции которого входит:

- Установление связи с сервером
- Получение требуемого документа

- Отображение полученного документа
 - Реагирование на действия пользователя - доступ к новому документу
 - После запуска браузер по команде пользователя или автоматически устанавливает связь с заданным WWW - сервером и передает ему запрос на получение заданного документа
2. WWW сервер ищет запрашиваемый документ и возвращает результаты браузеру
3. Браузер, получив документ, отображает его пользователю и ожидает его реакции:
- Возможные варианты:
 - Ввод адреса нового документа
 - Печать, поиск, другие операции над текущим документом
 - Активизация (нажатие) специальных зон полученного документа, называемых связями (link) и ассоциированными с адресом нового документа.
 - В первом и третьем случае происходит обращение за новым документом.

Адрес документа указывается в виде специальной строки, называемой URL. Для протокола HTTP, используемого при взаимодействии WWW клиента и WWW сервера, URL состоит из следующих компонент:

- 1) Наименование протокола, по которому работает сервер (http).
- 2) Имя машины - сервера в Internet или ее IP - номер.
- 3) Порт TCP, обращение к которому обрабатывает сервер.
- 4) Место (путь) документа на машине – сервере

http://www.cnit.nsu.ru:80/welcome.html

Здесь http означает протокол работы с WWW - сервером

- o ':' - разделитель
- o "www.cnit.nsu.ru" - имя машины - сервера в Internet
- o "80" - номер tcp - порта
- o /welcome.html - путь до документа на машине - сервере

Из общей схемы работы видно, что функции WWW сервера заключаются в следующем:

- 1) Установление соединения с клиентским ПО по протоколу tcp.
- 2) Принятие запроса на документ по протоколу http.
- 3) Поиск документа в локальных ресурсах.
- 4) Возврат результатов поиска по протоколу http.

Таблица 1

Наиболее распространенных платформ для WWW – сервера

Компьютер	Операционная Система
IBM PC	Unix (UnixWare, Open Server, Solaris, BSD, Linux и т.д.) Microsoft Windows NT/2000 IBM OS/2 Novell NetWare
Sun SparcStation и SparcServer	SunOS Solaris
Silicon Graphics серверы и рабочие станции	IRIS

В простейшем случае гипертекстовый документ представляет собой совокупность файлов. Представление этих файлов как единого документа производится браузером. По каждому файлу документа браузер делает запрос к WWW - серверу. Таким образом, сервер не имеет представления о структуре и составе документов, он отвечает только за выдачу локальных файлов по запросам. На различных платформах, в различных операционных системах пути файлов выглядят по разному.

Например:

D:\WWW\INDEX.HTM - в Windows,
/u/data/www/html/index.html - в Unix - системах,
USR:WWW/HTML - в NetWare и т.д.

Путь файла, указываемый в URL, имеет стандартный вид:

/<имя_каталога>/ ... /<имя_каталога>/<имя_файла>

Таким образом, в функции WWW - сервера входит преобразование адреса из внешнего единого формата в платформенно ориентированный внутренний формат. Появляется ряд понятий, специфичных для такого преобразования, необходимых для него.

Это каталог реальной файловой системы сервера, от которого идет вычисление пути, указанного в URL. Например, если исходным каталогом документов является **D:\WWW**, то на запрос к этому серверу документа по URL

http://<имя_сервера>/index.htm

будет возвращен файл

D:\WWW\index.htm

В случае, когда необходимо осуществить обращение к конкретному каталогу или файлу, находящемуся вне иерархии Исходного каталога документов, используется механизм синонимов. Синоним позволяет явно определить соответствие между путем, указанным в URL, и путем локальной файловой системы.

Например:

Синонимом для **/Harvest** объявляется **/projects/www/harvest** или синонимом для **/test/myfile.html** объявляется **C:\MYDIR\FILE.HTM**

В первом случае все обращения к файлам каталога **/Harvest** будут обрабатываться в каталоге **/projects/www/harvest**. Второй пример показывает работу синонима с конкретным файлом файловой системы.

Для каждого сервера определено имя так называемого индексного файла. Обычно этот файл содержит ссылки на другие файлы данного каталога. Содержимое индексного файла выдается сервером в случае, если в URL указан каталог без конкретного файла.

Для многопользовательских операционных систем (таких как Unix) ПО WWW - сервера позволяет каждому пользователю предоставлять доступ к своему собственному набору гипертекстовых документов вне основной иерархии (Исходного каталога документов, Синонимов и т.д.). Этот набор документов должен находиться в собственном (т.н. "домашнем") каталоге пользователя.

Например:

На сервере **Indy.cnit.nsu.ru** создан пользователь с именем **fancy** и "домашним" каталогом **/home/fancy**. Собственные гипертекстовые документы он хранит в каталоге **/home/fancy/public_html**. При обращении по URL **http://Indy.cnit.nsu.ru/~fancy/start.html**, WWW - сервер будет искать документ **start.html** в каталоге **/home/fancy/public_html**.

Протокол MIME - многоцелевое расширение электронной почты, был создан как способ передачи нетекстовой информации: изображений, звука, видео в письмах электронной почты. Механизм оказался удачным, и его перенесли и в on-line сервисы, в том числе WWW. Здесь MIME используется для передачи документов от сервера к клиенту.

В общем виде MIME основывается на передаче вместе с основными данными дополнительной информации, описывающей что это и в каком виде передается. Эта

дополнительная информация называется заголовком MIME. Базовой частью заголовка является строка, описывающая тип передаваемого сообщения. Формат строки:

Content-Type: <тип_MIME>

Перечень типов MIME (т.е. видов передаваемых данных) постоянно пополняется и может быть дополнен даже пользователем для описания своего собственного вида данных. Формат типа MIME:

<Тип> / <Подтип> [; <параметры>]

Где <Тип> - определяет общий тип данных:

Audio, Application, Image, Message, Multipart, Text, Video.

<Подтип> - указывает на специфический формат данных типа <Тип>

Например:

text/html - текстовые данные в формате HTML

image/gif - графические данные в формате gif

<Параметры> - список параметров, необходимых для интерпретации данных.

Для ведения специфичной обработки файлов различных типов и форматов на клиентской и серверной частях поддерживаются списки соответствий типов MIME и расширений файлов. Формат записи такого списка:

<Тип>/<Подтип> <расширение1> ... <расширениеN>

Эти списки сопоставляют всем файлам, имеющим определенные расширения, определенные типы MIME.

Например:

image/gif gif giff

text/html html htm

В первой строке всем файлам с расширением gif и giff присписывается тип содержимого image/gif. Если для типа содержимого image/gif определены специальные правила обработки (например, отображение на экране в определенной области), то так будут обрабатываться все файлы с расширениями gif и giff. Ниже приведен фрагмент программы по предназначениям:

// Цикл чтения файла по маркерам

```
while (fPosMarker != dbfile.length()) {
    try { mC++;
        dbfile.seek(fPosMarker);
        dbfile.read(Jumper);
        String jBuf = new String(Jumper);
        JIndex = Integer.parseInt(jBuf,10);
        int b = 0;
```

// Поиск конца словаря

```
while ( b != MD){
    dbfile.seek(fPosMarker+24+MIndex);
    b = dbfile.read();
    MTemp++;
    MIndex = MTemp;
}
MTemp= MTemp - 1;
```

// чтение Словаря из файла в отдельный массив

```
byte Dic[] = new byte[MTemp];
dbfile.seek(fPosMarker+24);
dbfile.read(Dic);
```

Главной задачей моей магистерской работы было создание программного интерфейса к существующей библиографической базе данных. Применение передовых технологий программирования позволили разработать программу, позволяющую производить поиск интересующей информации в базе данных не только по отдельно взятым ключевым словам, но и полному названию документа. Тем не менее, работа

требует дальнейшей доработки для организации постоянного доступа читателей к библиографическим ресурсам библиотекам города через Интернет.

Список литературы

1. Глушаков С.В. Базы данных: Учебный курс [Текст] / С.В. Глушаков, Д.В. Ломотыков. – К.: Абрис, 2000. - 504с.
2. Джейсон Мейнджер. Java: основы программирования: Пер. с англ. [Текст] / Джейсон Мейнджер. - К.: Издательская группа BHV, 1997. - 320с.
3. <http://www.java.sun.com>
4. http://httpd.apache.org/dist/httpd/binaries/win32/old/apache_1_3_14_win32_r2.exe
5. <http://java.apache.org/jserv/dist/ApacheJServ-1.1.2-2.exe>
6. <http://www.apache.org>