РАЗРАБОТКА ПОИСКОВОЙ СИСТЕМЫ, КАК КОМПОНЕНТЫ ИНТЕГРИРОВАННОЙ СЕТЕВОЙ СРЕДЫ

Потапенко В.А.
кафедра ЭВМ ДонГТУ
eline@cs.dgtu.donetsk.ua

Abstract

Potapenko V. The development of search system, as component of integrated network environment. *The purposes and tasks, methods and development tools of search system for use in the local network of DonSTU are described.*

Введение

Постоянный рост информационного наполнения сети ДонГТУ ставит задачи по обеспечению возможности удобного и быстрого поиска требуемой информации. В связи с этим возникла острая необходимость в организации программных средств для решения задач, связанных с доступом к внутренним ресурсам. Одним из самых распространенных и эффективных способов решения такого типа проблем является построение поисковой системы.

В информационном сообществе метод организации специальных средств для обеспечения поиска нужной информации не нов. Уже существуют и успешно работают различные службы, задача которых быстро и качественно указать пользователю адрес, где можно получить требуемую информацию. Приступая к построения собственной поисковой системы, рассмотрим уже имеющийся опыт в решении данной проблемы.

1. Обзор существующих поисковых систем

Поисковые службы различаются по количественным (охват, глубина поиска), и по качественным (возможность использования формальных логических запросов, фильтрация результата) характеристикам [2]. Условно их можно разделить на поисковые машины и каталоги (директории, рубрикаторы). Первые исследуют пространство IP –адресов с помощью специальных программ, называемых «пауками» (spider) или роботами, и индексируют найденные страницы. К самым мощным и популярным поисковым машинам обычно относят AltaVista (www.altavista.com), HotBot (www.hotbot.com) и Northern Lite (www.nlsearch.com), из русскоязычных – Яndex (www.yandex.ru) и Rambler (www.rambler.ru). Каталоги работают совершенно иначе: новые Web-узлы изучаются экспертами и относятся к соответствующим тематическим категориям. Многие каталоги также обеспечивают поиск в своей базе данных. В качестве примера можно привести Yahoo! (www.yahoo.com) и «Ау!» (www.au.ru).

В общем случае поисковая служба представляет собой довольно сложный программно-аппаратный комплекс. Например, AltaVista реализована в виде распределенной вычислительной системы из более чем двадцати компьютеров (естественно не персональных), на которых выполняется специализированное программное обеспечение.

Отличие в работе различных поисковых систем таково, что результаты поиска далеко не всегда пересекаются хотя бы на 20% [2]. Отчасти это объясняется различными алгоритмами исследования Internet, в первую очередь, компромиссом между качеством и скоростью обработки каждой Web-страницы.

Одни поисковые службы относятся к полнотекстовым: они ищут ключевые слова и в заголовке, и мета-тэгах, и в теле страницы; другие ограничиваются только заголовком и мета-тегами. То же относится и к глубине исследования узлов: одни обрабатывают только заглавную страницу, другие – все ссылки до определенного уровня, третьи – Web–узел целиком. Кроме того, некоторые службы имеют специализацию (явную или неявную) и уделяют больше внимания узлам, посвященным определенной теме.

Существенные проблемы имеются в вопросе поиска русскоязычных ресурсов, что является актуальным для сети ДонГТУ, где большинство информации представлено на русском языке. При этом используются несколько кодировок, и далеко не всегда страницы полностью дублируются. Некоторые русскоязычные службы ищут информацию сразу во всех кодировках, но далеко не все. Другая большая проблема связана с особенностями русского словообразования: для большинства поисковых машин слова «игра» и «игры» воспринимаются как разные, а общее число словоформ может быть довольно большим. Возможность поиска по подстроке (когда в качестве ключевого слова указывается «игр*») также не всегда дает приемлемый результат: во-первых, наверняка такая подстрока найдется и в совершенно посторонних словах (например, «игрек»), во-вторых, все равно выпадают слова вроде «игорный». Лучшие русскоязычные службы умеют искать не просто ключевые слова, а все их словоформы.

2. *Организация поисковой системы для сети ДонГТУ*

Рассмотрим структуру организации и работы разрабатываемой поисковой системы. На рис. 1 представлена схема работы типичной поисковой системы со стороны пользователя.

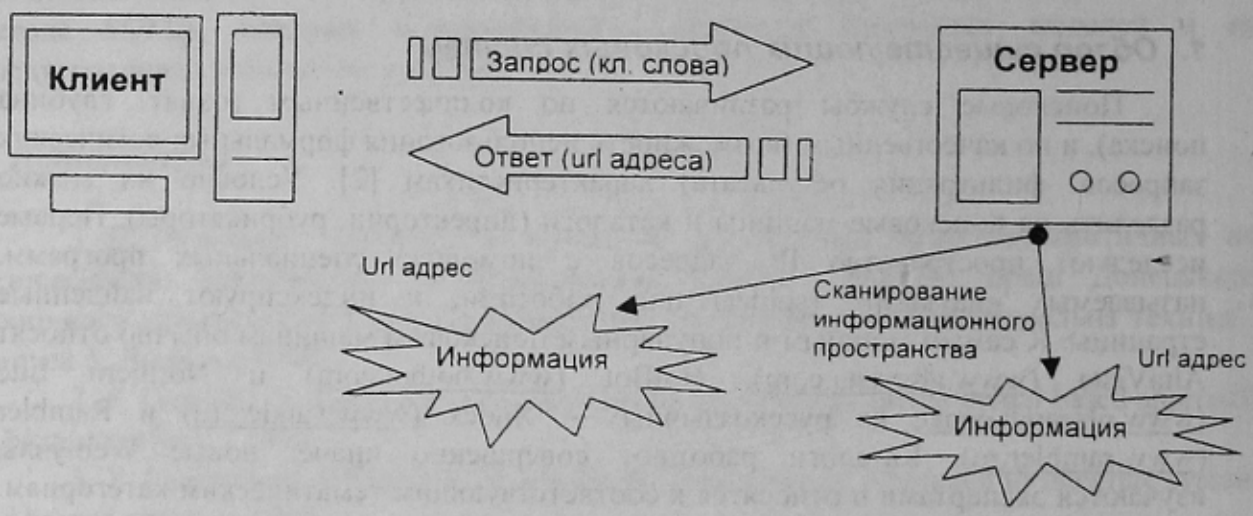

Рисунок 1. - Общая схема работы поисковой системы

Клиентом в данном случае выступает пользователь, на компьютере которого запущен интернет броузер. В поле ввода броузера вводится ключевое слово (или слова, если их несколько) и осуществляется запрос на сервер. На стороне сервера запускается специальная программа, которая осуществляет обработку запроса и выдачу информации обратно клиенту. В зависимости от типа поисковой системы (поисковая машина или каталог) информационное наполнение сервера может осуществляться по-разному. В нашем случае предполагается комбинированная организация системы. Информационное

наполнение сервера будет осуществляться как вручную, что характерно для систем типа «каталог», так и автоматически, при помощи робота, задача которого будет заключаться в сканировании адресов сети ДонГТУ и формировании базы данных полученной информацией.

Далее подробно рассмотрим организацию каждого компонента поисковой системы. Начнем с «клиента». Минимально необходимый HTML код для осуществления запроса на сервер приведен ниже:

```
<form name="RequestForm"
      method = "get/post"
      action="http://www.cs.dongtu.donetsk.ua/cgi-
bin/engine.pl
      onSubmit="JavaScriptFunction">
Поиск: <input type="text" size=70 name="RequestString"><br>
<input type="submit" value="Search">
</form>
```

Приведенный код начинается с объявления начала объекта `form`. Объекты `form` дают возможность пользователю осуществлять ввод информации в целый ряд полей гипертекстовой страницы и затем либо обрабатывать эту информацию локально, либо передавать ее для дальнейшей обработки на сервер [4]. Формы могут содержать поля ввода, многострочные поля ввода, селекторные кнопки, контрольные переключатели, меню или поля списка и кнопки. Обработчик событий `onSubmit` позволяет позволяет пользователю получить подтверждение о том, что данные формы были переданы. Данные формы передаются на сервер с помощью тега `<input type="submit">`. Значения GET и POST атрибута `method` пределяют метод, который применяется для передачи данных на сервер. При использовании метода POST содержащаяся в форме информация передается в виде текста через стандартный поток ввода, а при использованиее метода GET – через переменную среды QUERY_STRING, которая определена на сервере. QUERY_STRING специально предназначена для того, чтобы обеспечить получение информации сервером. Атрибут `action` определяет скрипт, который будет заниматься обработкой пользовательского запроса.

Обмен данными между прикладной программой и Web - сервером осуществляется при помощи CGI (Common Gateway Interface) [3]. С помощью CGI можно создавать CGI – программы, называемые *шлюзами*, которые во взаимодействии с такими прикладными системами, как система управления базой данных, электронная таблица, деловая графика и др., позволяют динамически выдавать на экран пользователя информацию. Для передачи данных об информационном запросе от сервера к шлюзу, сервер использует командную строку и переменные окружения. Эти переменные окружения устанавливаются в тот момент, когда сервер выполняет программу шлюза. Информация шлюзом передается в следующей форме:

имя=значение&имя1=значение1&..,

где «имя» – имя переменной (из оператора FORM, например), и «значение» – ее реальное значение. В зависимости от метода, который используется для запроса, эта строка появляется или как часть Url (в случае метода GET), или как содержимое HTTP запроса (метод POST). В последнем случае, эта информация будет послана шлюзу в стандартный поток ввода. Общая схема взаимодействия пользователя с приложением посредством CGI интерфейса, обобщающая все вышесказанное, изображена на рис. 2.

Роль CGI приложения в данном случае является обработка введенной пользователем информации и формирование результата поиска. На данном этапе поиск осуществляется в уже сформированной базе данных.

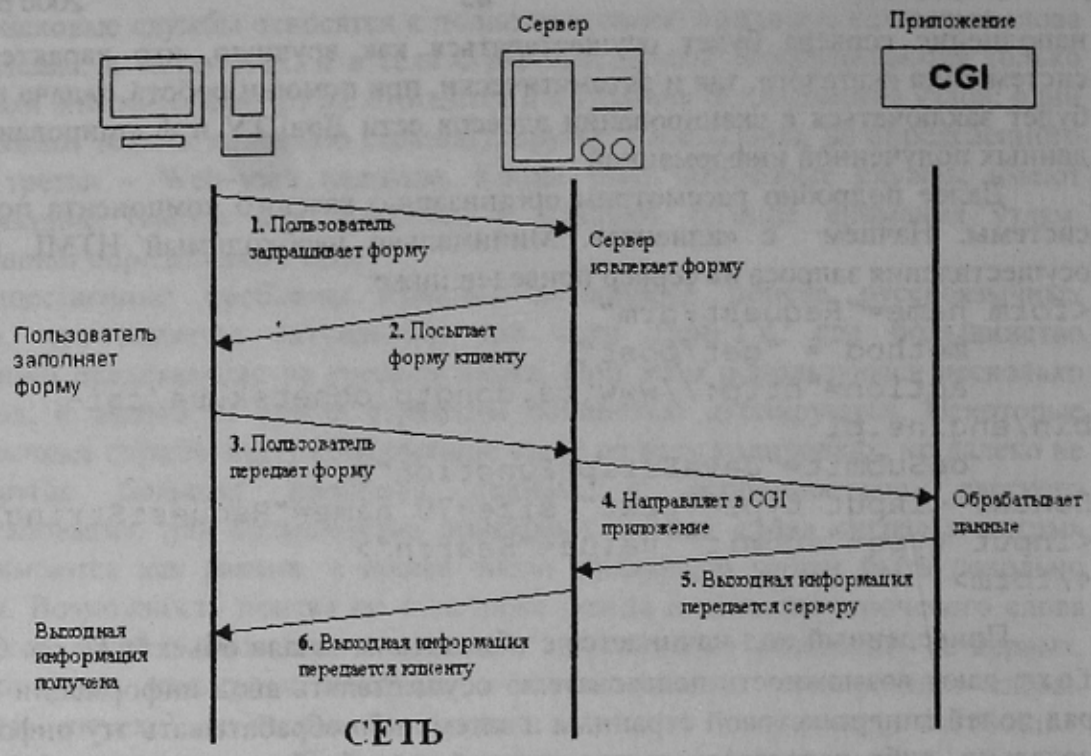

Рисунок 2. - Схема взаимодействия пользователя с приложением посредством CGI интерфейса

Ключевым моментом при разработке поисковой системы является организация оптимальной структуры базы данных. В настоящее время существует три так называемые классические модели данных, являющиеся основами построения реально функционирующих, эффективных СУБД [6,7]: иерархическая, сетевая и реляционная модели. В связи с учетом критериев эффективности, предъявляемых к моделям данных, наибольшее распространение получила реляционная модель. Данная модель позволяет хранить данные во многих таблицах, связанных друг с другом индексами [5,1]. Поскольку данные могут быть разнесены по нескольким таблицам, реляционные базы данных работают быстрее.

Проектирование любой базы данных включает следующие этапы:

- определение объектов (источников данных), которые должны быть внесены в базу данных;
- выявление связей между объектами;
- определение основных свойств объектов;
- выявление связей между свойствами объектов;
- определение отношений между таблицами базы данных, на основе связей между объектами данных, содержащихся в них;
- определение операций, выполняемых при создании и изменении информации в таблицах, включая обеспечение целостности данных;
- выявление индексов, необходимых для ускорения выполнения запросов;
- учет вопросов безопасности – какие полномочия и каким пользователям предоставлять;
- разработка процедур создания резервных копий и восстановления исходных файлов.

Каждый этап проектирования зависит от результатов предыдущих, и чем тщательнее он будет продуман, тем меньше итераций потребуется в дальнейшем. Среди целей наиболее важными представляются следующие:

- возможность хранения в базе данных всех необходимых данных;
- исключение избыточности данных;
- сведение к минимуму числа хранимых в базе данных таблиц;
- нормализация таблиц для упрощения решения проблем, связанных с обновлением и удалением данных.

Поскольку разрабатываемая поисковая система будет сочетать в себе собственно поисковый механизм и рубрикаторы, то предполагается использовать две основные таблицы, содержащие информацию для каждого режима работы. Для поискового механизма вид таблицы будет следующим:

Таблица 1. – Формат таблицы для поискового механизма

Ключ	URL	Список ключевых слов	Описание ссылки	Дополнительная информация
U27513	http://www.cs.dgtu.donetsk.ua	Кафедра ЭВМ, факультеты, сотрудники и т.д.	Персональная страница кафедры ЭВМ	Системный администратор Молдованов А.В. mold@cs.dgtu.donetsk.ua
…	…	…	…	…

Поле «*Ключ*» является уникальным идентификатором строки таблицы. На данном этапе оно не используется, однако наличие данного ключа крайне рекомендовано в литературе по СУБД [1,5,6,7]. Поле «*URL*» представляет собой уникальный идентификатор ресурса конкретной конкретного узла или страницы. В качестве примера приведен адрес домашней страницы кафедры ЭВМ. Одним из важных элементов таблицы является поле «*Список ключевых слов*». Формирование данного поля возможно двумя путями: в первом случае информация заносится исходя из заявок пользователей, во втором автоматически программой роботом, сканирующей страницы в поисках ключевых слов. Поле «*Дополнительная информация*» является необязательным атрибутом и предназначено, скорее, для сопровождения данного ресурса. Основными полями данной таблицы являются «*URL*» и «*Список ключевых слов*». Остальные поля дополняют основные и их количество может измениться.

Таблица, которая описывает каталог, имеет более сложную структуру. Ее формат приведен ниже:

Таблица 2. - Формат таблицы для формирования каталога

Ключ	Ключ поиска	Название раздела	Специальная ссылка	Доп. информация
L02568	Факультет ВТИ	Сотрудники факультета ВТИ	http://www.dgtu.donetsk.ua/cgi-bin/search.cgi?key=СотрудникФВТИ	…
L54125	Факультет ВТИ	Студенты факультета ВТИ	http://www.dgtu.donetsk.ua/cgi-bin/search.cgi?key=СтудентФВТИ	…
L47856	Сотрудник ФВТИ	Иванов А.Ю.	http://www.cs.dgtu.donetsk.ua/~ivanov	…

Поле «*Ключ*», как в предыдущей таблице пока не задействовано. Столбец «*Ключ поиска*» является специальным ключом для скрипта, формирующего

каталог в зависимости от выбранного пользователем элемента уровнем выше. Например, находясь на текущем уровне «Факультет ВТИ» пользователю предоставлен выбор: либо выйти на сотрудников ФВТИ, либо на студентов. В том случае, если пользователь выбрал в броузере ссылку на сотрудников ФВТИ, то запустится скрипт формирующий следующий уровень вложенности каталога. В данном случае он будет состоять из ссылки на сотрудника Иванова А.Ю. Если пользователь кликнет по ссылке на Иванова А.Ю., то попадет уже на его персональную страницу, а не на следующий уровень вложенности. Данный механизм работы обеспечивается полем «*Специальная ссылка*». В одном случае управление передается скрипту, формирующему следующий уровень вложенности, в другом ссылка указывает на конкретный URL.

Заключение

Вопросы построения поисковой системы возникают в любых, достаточно информационно наполненных, порталах. Рынок программных средств предоставляет определенные решения в данном вопросе. Тем не менее, каждое из предлагаемых средств имеет как сильные, так и слабые стороны. В каждом конкретном случае имеется своя специфика (например, наличие нескольких кодировок в русскоязычном интернете), что требует пересмотра всего механизма работы. В связи с этим разработка поисковой системы для ДонГТУ ведется с максимальным учетом всех особенностей топологии и информационного наполнения университетской сети с целью обеспечения наиболее полного и качественного доступа к ее ресурсам.

Литература

1. Тихомиров Ю.В. Microsoft SQL Server 7.0. СПб.: БХВ, 1999. – 720 с.
2. Дериев И. Поиск информации в Internet. Компьютерное обозрение №17-18, 1999. – С. 10-14.
3. Рэндал Шварц, Том Кристиансен Изучаем Perl. К.: Издательская группа БХВ, 1999. – 290 с.
4. Джейсон Мейнджер Java: основы программирования. К.: Издательская группа БХВ, 1997. – 320 с.
5. Джим Бойс, Скотт Фаллер, Ред Гилген Использование Microsoft Office 97, профессиональный выпуск. К.: Вильямс, 1998. – 1120 с.
6. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987. – 340 с.
7. Чоговадзе Г.Г., Качибая В.В., Сургуладзе Г.Г. Теория реляционных зависимостей и проектирование логической схемы баз данных. Тб.: Издательство Тбилисского университета, 1988. – 270 с.

Поступила в редакційну колегію 1.03.2000 р.