

Web Lecture Script를 위한 Java Web Server 구현

황호선*, 김은영, 김혜연, 한지선, 조동섭
이화여자대학교 컴퓨터학과

Implementation of Java Web Server for Web Lecture Script

Hyo-Sun Hwang, Eun-Young Kim, Hye-Yeon Kim, Ji-Seon Han, Dong-Sub Cho
Dept. of Computer Science & Engineering, Ewha Womans University

Abstract - 본 논문에서는 Web Lecture Script를 정의하고 이를 효과적으로 운영하기 위한 Java Web Server를 설계 구현하였다. Web Lecture Script란 강의록, 문제출제, 문제 평가, 숙제 제출, 자료실, 토론방 등의 가상대학 환경을 편리하게 구축하기 위해서 데이터베이스 및 파일의 접근을 Web 환경의 script만을 사용하여 구현할 수 있도록 정의한 것이다. Lecture시스템은 사용자에게 정형화된 포맷이 많이 쓰이기 때문에 script로 정형화된 형태의 인터페이스가 적합하며, 실제적인 구현이 Java로 이루어지기 때문에 확장성의 범위가 크다.

Web Server는 이러한 Lecture Script를 사용자에게 제공하기 위한 Script Engine을 포함하고 있어야 하는데, 이러한 Script Engine이 Web Server와 어떻게 상호작용하는가에 따라 성능이 좌우된다. 상호 작용하는 방법에는 Script Engine이 Web Server 자체에 포함된 경우, 독립적인 프로세서를 띄워서 처리하는 방법, 스크립트만을 처리하는 별도의 서버를 두어 처리를 전환하는 방법이 있다. 따라서 Lecture Script처리 Engine을 여러 방법으로 구현하고 성능을 비교하여 보다 효과적인 서버를 제안하고자 한다.

1. 서 론

최근 인터넷 사용의 증가와 함께 브라우저만을 이용한 교육 프로그램이 많이 사용되고 있다. 현재 가상대학 시스템, 원격 교육 시스템, 평생교육 시스템 등이 이러한 예이며 앞으로 이 분야에 대한 수요는 점차 늘어갈 것으로 보인다. 그러나 가상대학 시스템으로 사용되고 있는 대부분의 웹 서버들은 정형화된 인터페이스로 사용자와 관리자에게 일방적으로 제공되는 기능만을 사용하게 되어 있으며, 특정 OS에 종속적이다. 또한 사용자에게 보다 능동적인 수업효과를 제공하기 위해 멀티미디어 데이터의 사용이 증가되고 있기 때문에 서버의 기능이 커지고 있다. 앞으로도 인터넷 환경에서 교육 프로그램이 점점 증가하고 이를 이용하는 학생 수 또한 증가할 것으로 보이기 때문에 Web Server를 보다 효과적으로 구현할 필요성이 증가되고 있으며, 보다 빠르게 원하는 형태의 교육 시스템을 구축하고자 하는 요구가 증가할 것으로 보인다.

2. 본 론

2.1 Lecture Script

Script란 응용 프로그램이나 유틸리티 프로그램에 대한 명령어의 조합으로 구성된 일종의 간단한 프로그램으로 대개 루프나 if-then과 같은 간단한 제어 구조를 사

용하여 작성된다. 또한 구조가 간단하기 때문에 사용자가 사용하기 편하다. Lecture Script란 인터넷환경에서 교육용 시스템을 보다 편리하게 구축하기 위해 정의한 Script언어이다. 이 언어는 기존의 HTML(HyperText Markup Language)에 원하는 기능의 script를 추가시키는 방법으로 구현되므로 사용자가 간편하게 구현, 작성할 수 있도록 해준다. 또한 교육용 시스템이기 때문에 다른 시스템과 비교할 때 데이터베이스에 접근하거나 파일 다운로드(Download)나 업로드(Upload)가 많기 때문에 이와 관련된 문법이 많다.

2.1.1 Lecture script 문법 - etf

새로 정의한 script언어를 etf(Ewha Tag Format)라 정한다. 다음은 version-1에서 구현한 script의 문법으로 매크로, 파일 첨부, 루프에 관한 구문이 있다. 초기 버전은 기본적인 기능만을 구현하였고 이후 버전에서는 데이터베이스에 대한 구문을 추가할 계획이다.

▶ 매크로

기능 : 서버에 미리 정의되어 있는 일련의 값으로 치환시킬 수 있는 기능으로 "매크로의 매크로"와 같은 이중 매크로도 처리 가능. 서버에 macro.prf파일에 정의되어 있음

문법 : <%%\$매크로변수>

▶ 파일 첨부

기능 : etf script 파일에 다른 text나 html파일을 추가시킬 수 있는 기능으로 주로 파일의 머리말이나 꼬리말을 추가시킬 때 사용가능.

문법 : <%AttachFile=파일 이름>

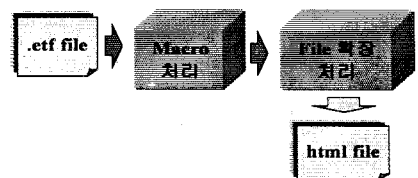
▶ Loop

기능 : 특정 부분을 반복할 수 있는 기능.

문법 : <%Loop=횟수> ~<%EndLoop>

2.1.2 Script 구현

스크립트의 처리 형태는 [그림1]과 같이 나타낼 수 있다.



[그림 1] Script 처리

1) 본 논문은 98년도 정보통신부 '산·학·연 공동기술개발사업'의 지원을 받았음.

1) Script가 포함된 etf파일 읽어들이

- 2) 매크로 정보가 들어있는 hash 테이블을 바탕으로 각 매크로를 치환
- 3) AttachFile, Loop 루틴을 처리

2.2 Java

Java 언어는 미국 선(Sun)마이크로시스템즈 사가 개발한 인터넷용의 프로그래밍 언어로 기존의 프로그래밍 언어와는 달리 네트워크 전용으로 만들어졌기 때문에 서로 다른 기종간의 네트워크 환경에서 강력한 기능을 발휘한다. 컴파일 시에 특정 하드웨어나 운영체제에 상관 없이 자바 가상 머신이 존재하는 곳에서는 어디서나 수행될 수 있는 바이트코드(bytecode) 형태로 목적 코드가 생성되기 때문에, 자바로 응용 프로그램을 작성할 경우 자바 가상 머신을 내장하고 있는 환경이라면 어디에서든 실행시킬 수 있다. 이러한 특징 때문에 자바는 사용자의 웹 페이지를 저장하는 일 뿐만 아니라 대규모의 응용 프로그램을 작성하는 일에도 적합하다.

2.3 서버-클라이언트(Server-Client)

간단하게 클라이언트는 서비스를 요청하는 프로세스이고 서버는 클라이언트의 서비스 요구에 응답하는 프로세스이다. 네트워크 환경에서 서버-클라이언트 모델의 구축 방법은 여러 클라이언트의 요구를 동시에 응답해 줄 수 있는가에 따라 반복 서버(iterative server)/ 병행 서버(concurrent server), TCP를 사용하는가 UDP를 사용하는가에 따라 연결형 서버(connection server)/비연결형 서버(connectionless server)로 분류할 수 있다. 다음은 이러한 두 가지 관점에서의 서버종류와 그 특징이다.

▶ 반복적, 비연결형 서버

클라이언트의 요구에 비하여 처리하는 시간이 작을 때 주로 이용된다. 반복적 서버는 종종 연결에 대한 상태를 유지하지 않으며 이 서버의 특징의 동작을 이해하기 쉽고 서버가 잘못된 사건이 잘 발생하지 않는다는 점이다.

▶ 반복적, 연결지향형 서버

클라이언트의 요구 서비스를 신속하게 처리할 수 있는 서버로써, 많이 사용되지 않으나 신뢰성 있는 전송을 필요로 하는 경우에 쓰인다. 연결을 설정하고 유지하는데 부담이 많기 때문에 평균적인 응답시간이 길어진다는 단점이 있다.

▶ 병행적, 비연결형 서버

클라이언트로부터 새로운 연결요구가 있을 때마다 새로운 종속 프로세스를 생성하는데 드는 비용이 병행수행을 함으로써 얻는 이익보다 크기 때문에 잘 쓰이지 않는다.

▶ 병행적, 연결지향 서버

가장 일반적인 서버형태로써 신뢰성 있는 전송을 보장해 주면서 여러 개의 클라이언트 요구를 동시에 처리할 수 있다. 기본적으로 여러 개의 병행 프로세스들을 이용하여 여러 클라이언트의 요청을 처리하지만, 효율을 높이기 위해 사전에 프로세스들을 미리 생성해 놓고 새로운 클라이언트의 요청이 도착하면 새 클라이언트에게 서비스할 프로세스를 하나 할당하므로 운영체제의 부담이 줄어들게 구현할 수 있다.

2.4 웹 서버(Web Server)

웹 서버란 브라우저(클라이언트)가 요구하는 웹 페이지(서비스)를 제공할 수 있는 환경을 구축하기 위해 사

용되는 서버로 대표적인 병행적, 연결지향 서버이다. 가장 많이 사용되는 웹 서버로는 apache server를 예로 들 수 있는데 이러한 서버들은 OS의 부담을 줄이기 위해 프로세서를 사전 할당하여 필요할 때 할당하는 방식을 취하고 있다.

그러나 자바에서는 시스템의 부담을 줄이기 위해 프로세스가 아닌 쓰레드(Thread) 개념을 가지고 서버를 구축할 수 있다. 쓰레드란 현재 작동 중인 한 태스크(task, 작업)에서 작동할 수 있는 태스크보다 더 작은 하나의 작업 단위를 말하는 것으로 작은 서브 모듈을 말한다. 일반적으로 이것은 동일한 문맥을 사용하는 작업의 단위를 의미하는데, 하나의 프로세스 안에서 각각의 쓰레드는 동일한 문맥을 사용하기 때문에 보다 빠르게 작업을 교체하여 실행시킬 수 있다는 특징을 가지고 있다.

자바의 쓰레드 개념의 웹 서버는 클라이언트의 connection이 요청될 때 새로운 프로세스를 생성하는 것이 아니라 connection에 대한 쓰레드를 생성시킴으로써 시스템의 부담을 줄일 수 있다.

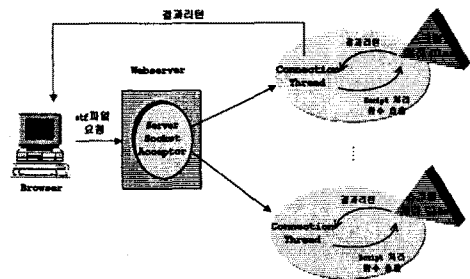
2.5 웹 서버에서의 script 처리

자바 웹 서버에서의 script처리는 다음 세 가지 형태로 구현될 수 있다.

2.5.1 서버 타입 1 - 함수 호출 서버

Script 처리를 위한 기능이 서버 자체에 있는 경우를 말한다. 즉 브라우저에서 파일의 요청을 받은 web server는 connection을 처리하기 위한 thread를 생성하는데, 이 thread에서 직접 script를 처리하여 브라우저에게 보내는 것이다. [그림2]은 이와 같은 서버의 개념도이며, 처리 순서는 다음과 같이 나타낼 수 있다.

- 1) Browser에서 etf파일 요청
- 2) Web Server에서 connection thread 생성
- 3) Script 처리를 위한 function을 호출
- 4) 결과 값(html)을 브라우저에게 반환



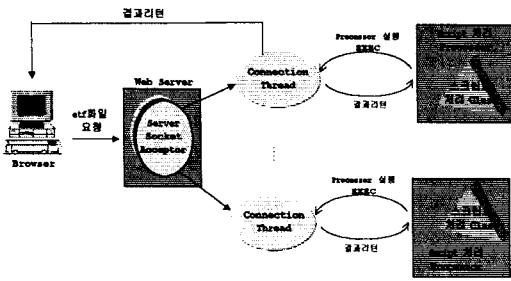
[그림 2] 함수호출 서버 개념도

2.5.2 서버 타입 2 - 프로세서 생성 서버

기존의 Web Server에서 CGI 프로그램을 처리하는 방법은 해당 프로그램을 독립적인 프로세서로 실행시켜서 그 결과를 브라우저에 보내준다. 서버 타입 2에서도 이러한 방식과 비슷하다. script를 처리해주는 독립적인 프로그램을 두고 web server가 이 프로그램을 실행시켜서 결과를 브라우저에게 보내는 것이다. [그림3]는 서버의 개념도이며, 처리 순서는 다음과 같다.

- 1) Browser에서 etf파일 요청
- 2) Web Server에서 connection thread 생성
- 3) Script 처리를 위한 processor생성

- 4) processor에서 결과 값(html)을 얻어옴
- 5) thread가 결과 값(html)을 브라우저에게 반환



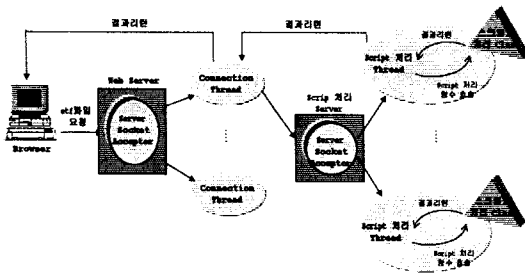
[그림 3] 프로세서 생성 서버 개념도

2.5.3 서버 타입 3 - 독립기능 서버

Script 처리를 자바의 서블릿(servlet)과 비슷한 방식으로 처리한다. 기존 웹 서버들은 서블릿을 실행시키기 위해 서블릿 처리를 위한 서블릿 엔진을 실행시킨다. 이 서블릿 엔진은 웹 서버에 등록된 서블릿들에 대한 정보를 알고 있다가 웹 서버의 요청이 들어오면 해당 서블릿을 실행시키고 그 결과를 웹 서버에게 돌려준다. 이때 웹 서버와 서블릿은 TCP-IP로 연결이 되기 때문에 서블릿 엔진의 다른 사이트의 웹 서버의 요청을 받아 실행시킬 수도 있다.

우리가 구현하고자 하는 서버는 스크립트만을 처리하는 Script 처리 Engine을 두고 서버가 필요할 때마다 이 Engine에 처리를 요청하고 받는 형태를 취할 수 있다.

- 1) Browser에서 etf파일 요청
- 2) Web Server에서 connection thread 생성
- 3) Script Server에 TCP-IP접속
- 4) script server에서 connection에 대한 thread 생성
- 5) script server의 thread가 .etf파일 처리
- 6) 결과 값(html)을 web server의 thread에게 반환
- 7) web server의 thread가 결과 값(html)을 브라우저에게 반환



[그림 4] 독립기능 서버 개념도

2.6 실험 결과 및 비교

각각의 서버들을 실행시키면서 처리 시간을 분석하여 다음과 같은 결과를 얻을 수 있었다.

▶ connection thread 생성

서버 1에서 생성되는 connection thread는 쓰레드 차체에 script처리 루틴을 포함하기 때문에 서버 2와 서버3에서 생성되는 것보다 부하가 클 것이라는 예상과

는 달리 세 타입 모두 connection thread 생성에 대해 동일한 시간이 걸렸다. 따라서 서버에서 thread를 생성하는 것은 서버의 성능을 크게 좌우하지는 않았다.

▶ script처리

처리 시간순으로 보면 서버1이 가장 빠르고 서버2가 제일 느렸다. 따라서 쓰레드에 비해 새로운 프로세서를 생성시키는 것은 서버에 많은 부담을 줄 수 있다는 것을 보여준다. 서버3은 서버 1과 같은 형태로 script를 처리하기 위한 쓰레드를 생성시키기 때문에 처리 속도는 빠르다. 하지만 connection thread가 script server에 TCP-IP로 접근하기 때문에 네트워크 상황에 따라 성능이 크게 좌우될 수 있기 때문에 실제적으로는 다른 두 서버 형태의 중간 정도의 실행시간을 갖는다.

▶ 분산 환경

서버3의 경우는 웹 서버와 TCP-IP로 연결되기 때문에 같은 사이트(site)에 있을 수도 있고 다른 사이트에 있을 수도 있다. 따라서 네트워크 상황이 좋다면 웹 서버에만 집중될 수 있는 처리량을 다른 서버로 분산시킬 수 있기 때문에 대용량 서버를 구현할 때 좋은 방법이 될 것이다.

3. 결 론

Lecture script를 정의하고 각각의 기능을 함수 호출 서버, 프로세서 생성 서버, 독립기능 서버의 세 가지 타입으로 구현하여 성능을 비교하였다. 비교 결과 script의 매크로처리 같은 간단하고 자주 사용되는 기능은 서버1의 형태로 구현해야 하며, 자주 사용되며 시스템 부하가 큰 데이터베이스 처리부분은 서버3의 형태로 구현하는 것이 바람직하다. 서버2의 형태는 자주 사용되지 않거나 분리된 기능을 구현하는 것이 서버의 성능을 높이는 방법이다.

4. 향 후 연 구

Lecture script에 데이터베이스나 파일 시스템을 체크할 수 있는 새로운 기능을 추가시켜 이것이 웹 서버에 어느 정도의 부하를 주는지 분석하여 이것을 효과적으로 구현하기 위한 웹 서버를 구현하고자 한다.

(참 고 문 헌)

- [1] Jason Hunter, "How to get started with server-side Java", <http://www.javaworld.com/javaworld/jw-04-1998/jw-04-step.html>
- [2] Rinaldo S. DiGiorgio, "Monitor your Web server in real time", <http://www.javaworld.com/jw-04-1996/jw-04-javadev.html>
- [3] Scott Clark, "The Java Web Server", http://www.webdeveloper.com/java/java_ji_java_web_server.html
- [4] JServ Server, <http://java.apache.org/>
- [5] Mike Cohn의 5인, "DEVELOPER'S REFERENCE JAVA", Sams.net, 1997
- [6] Rovert Orfali and Dan Harkey, "Client/Server Programming with JAVA and CORBA", John Wiley and Sons Inc., 1997