Apache Overview HOWTO

Daniel Lopez Ridruejo

ridruejo (at) apache.org

도운이

장석문

linuxodyssey (at) hanmail.net

v0.7, 2002-02-28

이 문서는 당신에게 아파치 웹 서버와 관련 프로젝트들에 관한 관점을 제공한다. 이 글은 지금 완성된 내용의 세부적인 것과, 앞으로 이루어질 일들에 관해 설명할 것이다.

1. 들어가면서

이 문서는 당신에게 아파치와 관련 프로젝트들에 관한 관점을 제공한다. 아파치는 인터넷에서 가장 일반적으로 사용되는 (http://www.netcraft.com/survay/) 서버이다. 윈도를 사용해 오던 새로운 아파치 사용자들은 종종 아파치가 할 수 있는 많은 일들과 쉽게 추가, 확장할 수 있음을, 그리고 더 일반적이라는 장점들을 놓치곤 한다. 이 문서는 아파치가 할 수 있는 일들을 간략히 소개하는 것이 주안점을 두고 작성될 것이다. 이와 같은 정보는 많은 소스들과 프로젝트 관련 웹페이지, 컨퍼런스, 메일링 리스트, 그리고 아파치 웹 사이트와 나의 지식에서 얻어진 것이다. 이런 내용에 대한 권한은 각각의 내용의 저작자에게 있다.

Disclaimer: 나는 커벌런트 (http://www.covalent.net) 사에서 일하고 있다. 우리는 아파치 웹 서버를 지원하는 서비스와 제품을 공급한다. 그리고 나는 그것들 중 일부에 참여하고 있으며, 우리의 그런 일이 오픈 소스 프로젝트와 유사하게 전개되도록 하고 있다.

만일 당신이 이 문서에서 오타나 잘못된 부분을 발견한다면 그와 같은 부분을 수정할 수 있도록 알려 주기 바란다.

2. 아파치

아파치는, 넷크래프트 (http://www.netcraft.com/survey)에 의하면, 시장의 60% 이상을 차지라며 인터넷 웹 서버 시장을 선도하고 있다. 몇몇의 괄목할만한 사실이 그와 같은 아파치의 성공을 대변하고 있다. :

- 아파치의 라이센스 (http://www.apache.org/LICENSE.txt)는 BSD와 마찬가지로 상업적인 목적과 비상업적인 목적 양쪽에 있어 오픈 소스를 따르고 있다.
- 기술적으로 유리한 다양한 개발과 경험을 지닌 재능있는 개발자들의 커뮤니티 (http://www.apache.org/foundation/index.html)
- 모듈 아키텍처. 아파치 사용자들은 아파치에 자신의 함수나 특별한 환경을 적용하기 용이하다.
- 가능: 아파치는 Unix (and Linux), Windows, BeOs, mainframes... 등에서 동작한다.
- 안정성과 보안성

Oracle (http://www.oracle.com), Red Hat (http://www.redhat.com) 그리고 IBM (http://www.ibm.com) 등의 많은 상업 벤더들은 아파치를 기반으로 한 솔루션 제품군들을 내놓고 있다. 참고로 커벌런트 (http://www.covalent.net)는 아파치를 위한 애드온 모듈과 24x7 을 공급하고 있다.

다음의 웹 사이트는 아파치와 여타의 것들을 사용한 것이다. 그런 사이트들에서 아파치가 잘 적용되었다면, 당신에게도 같지 않겠는가. ^^

- 아마존(Amazon) (http://www.amazon.com)
- 야후!(Yahoo!) (http://www.yahoo.com)
- W3 컨소시엄 (http://www.w3c.org)
- 파이낸셜 타임즈(Financial Times) (http://www.ft.com)
- 네트워크 솔루션(Network solutions) (http://www.networksolutions.com)
- MP3.com (http://www.mp3.com)
- 스탠포드 대학(Stanford) (http://www.stanford.edu)

아파치 웹 사이트 (http://www.apache.org)에서 :

아파치 프로젝트는 안정적이며 상업적인 수준에다가 두각을 나타내며 소스 코드를 자유롭게 공개할 수 있는 웹 서버를 지향하며 공동으로 만들어지고 있는 소프트웨어이다.

아파치 프로젝트는 단순한 웹 서버에서 자바와 XML 등의 서버 사이드 기술을 포함하는 서버로 발전하고 있다. 아파치 소프트웨어 재단에 관해서는 다음 섹션의 프로젝트들을 통해 말할 것이다.

관련된 이야기

- W09: 아파치 웹 서버 입문
- F16: 상업적인 OSS 제품들의 라이센싱 문제

3. 아파치 소프트웨어 재단

아파치 소프트웨어 재단은 아파치 오픈 소스 소프트웨어 프로젝트를 위한 조직적, 법률적, 경제적인 지원을 위해 존재한다. 일반적으로 아파치 그룹으로 알려진 바와 같이, 재단은 멤버십 기반으로 모인 조직으로, 개인적인 지원가들의 안정적인 지원이 계속되어 아파치 프로젝트를 유지하게 하려는 목적의 법인으로 시작한 것이 아 니었다. 지적인 공헌을 가능하게 하는 것은 쉬운 일 처럼 들리지만, 이것은 오픈 소스 소프트웨어 프로젝트에 대한 관여가 계속되는 동안의 법률적인 지원이 필요한 일이다.

ASF의 의장인 로이 T. 필딩(Roy T. Fielding)은 이렇게 말했다. : 아파치 소프트웨어 재단의 목적은, 아파치와 같은, 인터넷을 통해 생성되고 유지되고 웹 상의 하부조직이 표준으로 발전되는 방식의 협동적인 소프트웨어 개발 프로젝트를 지원하기 위한 데 있다.

당신은 이곳에서 (http://www.apache.org/foundation/) 재단에 대한 더 많은 정보를 얻을 수 있다.

4. 아파치를 이용한 웹 어플리케이션 개발

아파치에 콘텐츠를 제공하는 몇몇의 방법이 있다.

관련된 이야기

- W07: 웹 어플리케이션 기술 개관

4.1. 정적인 컨텐트

아파치는 HTML 파일이라던가 이미지와 같은 고정 컨텐츠를 보낼 수 있다. 만일 당신이 원하는 것이 이것이 전부라면, 아파치는 당신이 원하는대로 옳게 작동할 것이다. 저사양의 펜티엄에서 작동하는 리눅스와 아파치는 고정 컨텐츠에 있어 10Mbps의 전송 속도를 낼 수 있다. 만일 당신이 아파치를 사용하는 데 있어 초보라면, 5절 의 섹션을 참고하도록 한다.

4.2. 다이나믹 콘텐츠(Dynamic content)

대개의 웹 사이트에서, 정보는 항상 바뀌며, 페이지는 계속하여 새로운 것을 보여 주어야만 한다. 서버사이드 프로그래밍의 모든 것 - 프로그래밍 언어, 도구, 프레임워크등-은 개발자들이 서로 다른 소스들-데이터 베이스, 디렉토리 서비스, 고객의 레코드, 여타의 웹 사이트-에서 정보를 요구하고 수정할 수 있게 하며, 콘텐츠를 사용자에게 쉽게 전달하게 한다.

4.3. CGI 스크립트

CGI는 Common Gateway Interface라는 말이다. CGI 스크립트는 사용자가 특정 웹페이지를 호출했을 때 실행되는 외부 프로그램이다. CGI는 웹 서버에서 정보(폼에서 받은 변수, 브라우저의 타입, 클라이언트의 IP 주소 등)를 받고,이 정보들을 이용하여 클라이언트에 웹 페이지를 보낸다.

Pros: 이것이 외부적인 프로그램일 동안에는, 이것은 어떤 언어로도 코딩될 수 있다. 이 스크립트는 다른 웹 서버들에서도 사용할 수 있다. CGI 프로토콜은 단순하며, 표준 출력으로서 결과값을 돌려준다. 이와 같은 것의 기술적인 세부 사항에 관해서는 많은 양의 온라인 문서들이나 책을 참고할 수 있다.

Cons: 프로세스의 생성과 초기화에는 시간이 필요하다. CGI는 서버에 외부적이며 어떤 상황에서의 모든 요청에 대해 생성, 혹은 파괴를 계속한다. 프로세스가 외부의 라이브러리를 불러오거나 외부 데이터베이스와 연결한 상태에서 지연은 중요한 부분이다. 많은 히트 수를 기록할 때도 마찬가지이다. CGI는 외부적인 이유에 의해 낭비 혹은 세션 관리를 달성한다.

CGI는 종종 과부하를 불러 왔으므로, 스크립트 언어는 자연스러운 선택이 되었다. 펄(Perl) (http://www.perl.com/)은 CGI 프로그래밍 언어의 선택으로 일반적인 것이 되었다. 이것은 텍스트 처리나 문자열 핸들링에 대한 효과적인 지원을 위해 대가를 지불한다. CGI 스크립트와 라이브러리 들은 많은 양을 자유롭게 사용 할 수 있었다. 좋은 출발점은 the Open Directory CGI section (http://dmoz.org/Computers/Programming/Internet/CGI/)이 될 것이다.

4.4. 사이트 생성기

당신의 사이트가 많은 내용을 담고 있다면, 당신은 컨텐츠 내용을 동적으로 상황에 맞게 보이고 싶어할 것이다. 오프라인 컨텐츠 생산자는 둘 중 하나를 선택해야 한다. 그 해결책은 형식적인 외관에서 내용을 분리해 내는 것이다. HTML 생성기는 웹사이트를 생성하기 위해 컨텐트와 표현할 자료들을 읽고 웹사이트에 만들어진 정적인 파일로 내보낸다. 생성자는 정기적으로 혹은 컨텐트의 변경에 의해 동작한다.

16.5절의 다음 버전은 배치 모드를 지원할 것이다. 그 밖의 옵션에 관해서는 웹 사이트 메타 언어 (http://www.engelschall.com/sw/wml/)를 참고하자.

4.5. 서버 외부의 처리

웹 서버는 다른 프로그램으로 동적인 요청을 보낼 수 있다. 이런 프로그램은 요청이 들어올 때 까지 조용히 있는다. 요청은 처리되고 다시 웹서버로 돌아가 클라이언트로 되돌려 보내진다. 이와 같은 일들은 CGI 스크립트를 통해 이루어진다. 이런 접근의 예는 4.6절, 4.7절 등등을 확인해 보자.

4.6. Fast CGI

이 표준은 CGI 프로토콜의 빠른 접근을 위해 만들어졌다. 핵심적인 해결책은 하나 이상의 요청에 대해 단일하게 생성된 프로세스들이 처리하는 것이다. 아파치 모듈에는 Tcl, Perl, 기

타 등등을 위해 FastCGI 프로토콜과 라이브러리를 포함하고 있다. 자세한 정보는 이곳 (http://www.fastcgi.com)을 참고할 것!

관련된 이야기

- F18: FastCGI -- 잊혀진 보물

4.7. 자바 서블릿

자바 서블릿을 제공하기 위해 자바 가상머신(웹서버와 분리된)이 요청을 처리한다. 외부적인 자바 가상 머신 프로세스들은 요청한다. JVM은 같은 컴퓨터 혹은 서로 다른 컴퓨터에서 동작될 수 있다. 이것은 얼마나 많은 어플리케이션 서버들이 작동하느냐의 문제이다. 일반적인 표준 라이브러리들은 서버사이드 프로세싱을 위해 포함된다. 15.9절와 15.6절은 이 기능을 제공하기위한 아파치 프로젝트이다. 자바 어플리케이션 서버 프로젝트의 관계는 25.2절 에서 찾을수 있다.

관련된 이야기

- W16: 자바 기반의 웹 어플리케이션 아키텍처의 추천

4.8. 임베디드 인터프리터

서버 외부의 처리 문제는 서버 스스로 인터프리터를 내장하는 것으로 귀결된다. 이것은 모듈의 종류를 대략 2가지로 나눈다. 모듈은 요청에 대해 응답하거나 요청을 수정하고 클라이언트에게 결과값을 HTML 페이지로 만들어 보여준다. 가장 일반적인 접근은 17.3절과 18절을 통한 것이다.

5. 성능과 대역폭의 관리

저수준의 작동은 웹 서버 내의 요소만을 감안한다.(유연성과 안전성이 최우선으로 고려된다.)

웹서버가 정적인 콘텐츠를 공급할 때의 노역의 해결은 퍼포먼스의 개선을 통해 이루어야 한다고들 한다. 만일 당신이 호스팅 사업을 하고 있다면, 아파치는 당신이 일반적으로 측정하고 제어할 수 있는 대역폭 안에서 서비스를 공급할 것이다. 이런 상황에의 제어는 종종 응답 처리 등의 속도가 떨어지는 현상으로 나다난다. 이것은 과부하를 막자는 것으로 끝난다.

- **mod_mmap**: 현재의 아파치 버전에 포함되어 있다. 이것은 가끔 변경되지만 자주 요청되는 것에 대한 정적 리스트를 위한 메모리를 관리한다.

- **Mod_bandwidth (http://www.cohprog.com/mod_bandwidth.html)**: 특정 디렉토리**,** 특정 크기의 파일**,** 혹은 외부 **IP**에 대하여 서버 와이드 혹은 연결수당 대역폭 한계를 설정한다**..**
- 대역폭 공유 모듈 **(http://www.topology.org/src/bwshare/README.html)**: 클라이언트의 IP에 의한 균형, 조절을 담당하여 대역폭을 공급한다. 이것은 활발히 계속되고 있다.
- **Mod_throttle (http://www.snert.com/Software/Throttle/index.shtml)**:가상 호스트나 유저에 대한 대역폭 제어
- **Mod_throttle_access (http://www.fremen.org/apache/)**: 만일 당신이 slashdotted (http://everything2.com/index.pl?node_id=13464)한다면 유용하다. 자원 기반의 제어를 허용한다.

6. 가상 호스팅(virtual hosting)

아파치는 특별한 기능을 공급하는 모듈을 추가함으로서 대규모의 가상 호스팅을 지원한다.

- mod_dynvhost (http://funkcity.com/0101/)
- mod_pweb (http://www.joytec.de/mod_pweb.html)
- mod_v2h (http://www.fractal.net/mod_v2.htm)

추가하자면,아파치 2.0은 보안 문제를 해결하면서도 다른 도메인을 다른 사용자 id로 사용하게 하는 기능을 허용한다.

7. 로드 밸런싱

아파치는 증가분에 대비하여 서버 사이의 요청을 분배하는 몇몇의 모듈을 가지고 있다.

- **Reverse Proxy + mod_rewrite**: mod_rewrite (http://www.apache.org/docs/mod/mod_rewrite.html)를 사용할 수 없다면 당신이 아파치에서 사용할 수 있는 것은 없다. ^^;;; 이런 기술은 백엔드 서버를 위한 프록시와 같이 프론트엔드 서버를 동작하게 한다. 더 많은 정보는이곳에서 (http://www.apache.org/docs/misc/rewriteguide.html) 얻을 수 있다.

  Mod_backhand: 한 웹서버에서 다른 곳으로 **HTTP** 요청이 빈틈없이 리디렉션 되는 것을 허용한다**.** 이와 같은 리디렉션은 대상이 되는 컴퓨터에게 적은 리소스와**,** 좋은 결과를 공급하고**,** 웹 상의 요청의 로드 밸런싱에 대한 리퀘스트를 제공한다**..** 더 많은 정보는 이곳 (http://www.backhand.org/)을 참조하자.

관련된 이야기

- TH06: mod_backhand: 내부적인 설명

8. 보안 트랜잭션

아파치 서버를 위한 보안 트랜잭션을 위한 몇몇 방법들이 있다. 이것은 아파치 서버를 사용하는 전자 상거래나 민감한 정보들이 오가는(예컨대 신용카드 번호라던가) 여타의 일들을 가능하게 한다.

- Mod_ssl (http://www.modssl.org)과 Apache-SSL (http://www.apache-ssl.org) 은 오픈 소스의 산물이다. 이것들은 유럽에서 기원하였으며, RSA 조항을 따르지 않는다.
- 레드햇 (http://www.redhat.com)은 아파치에서 파생된 보안 서버를 제공한다. 레드햇은 C2Net, StrongHold, 그리고 그 밖의 보안 서버를 사용하게 한다.
- 커벌런트 (http://www.covalent.net)는 Covalent SSL 모듈이 플러그인으로 내재되어 있는 보안 버전의 아파치를 판매한다.

신용 카드 트랜잭션

아파치는 신용 카드 트랜잭션에 대한 명확한 방법들을 갖고 있다. :

- Cypay (http://www.cypay.com/) 신용카드 모듈. 템플릿 기반으로 세금 계산을 할 수 있다.
- Covalent credator (http://www.covalent.net), 다양한 금융 거래를 지원하며 잘못된 종료를 제어하고, PHP, Perl, Java를 지원한다.

9. SNMP

SNMP은 Simple Network Management Protocol을 의미한다. 이것은 네트워크 서버와 장비 전반에 대한 관찰과 관리를 허용한다. 아파치에서 사용되는 SNMP 모듈은 웹서버의 많은 다양한 전개에 대한 관리를 돕고, 서비스의 질을 측정하고, 존재하는 곤리 프레임워크 상에서 통합한다.

- 아파치 1.3을 기반으로 한 오픈 소스 Mod SNMP (http://www.simpleweb.org/software/packages/mod-snmp/)
- 커벌런트 SNMP (http://www.covalent.net)는 최근의 SNMPv3 을 지원하고 HP-Openview, Tivoli 등과 함께하는 상업적인 SNMP 모듈을 공급한다.

10. 인증 모듈

많은 상황에서 사용자 인증이 사용된다. 아파치는 기본적인 인증 지원을 포함하지만, 보안 프레임워크나 데이터 베이스, 기타 등등(NT 도메인 컨트롤러, 오라클, MySQL, 포스트그레스 SQL 등등)에 접속하기 위한 추가적인 인증 모듈이 존재한다.

LDAP 모듈은 기업의 디렉토리 서비스를 가능하게 하는 재미있는 녀석이다.

이와 같은 모듈들을 이곳에서 (http://modules.apache.org) 찾을 수 있다.

11. 아파치의 GUI 환경

아파치는 텍스트 설정 파일을 이용하여 설정한다. 여기에는 장점과 단점이 있다. ssh (http://www.openssh.com)를 사용하는 한 어디에서도 설정이 가능 하다는 것은 장점이지만, 손으로 설정 파일을 수정하는 것은 공부가 필요한 일이다. 오픈 소스의 그래픽 유저 인터페이스적인 도구를 이용하여 이 작업을 더 편하게 할 수 있다.

- 코만치 (http://www.comanche.org) : 이것은 Unix/Linux, Windows, 그리고 Mac 에서 실행되는 크로스플랫폼이다.
- gui.apache.org (http://gui.apache.org): 아파치의 GUI 환경 프로젝트.
- Webmin (http://www.webmin.com/webmin/): 쓸만한 웹 기반 환경.

12. 아파치 모듈 작성

아파치는 다른 많은 성공적인 오픈소스 프로젝트와 마찬가지로 모듈 아키텍처를 갖고 있다. 이것은, 당신이 전체적인 모든 코드를 이해하지 않더라도 함수 차원의 수정과 추가가 가능하다는 뜻이다. 아파치의 소스 코드에 접근한다는 말은 당신이 필요한대로 모듈을 추가하거나 당신의 것을 집어넣어 당신만의 서버를 만들 수 있다는 뜻이다.

아파치의 확장은 C나 다양한 다른 언어를 사용한 모듈을 통해 할 수 있다. 이와 같은 모듈은 아파치에 다른 언어인 Perl이나 Tcl 등으로 작성한 함수 등을 추가하게 한다.

C로 모듈 작성하기: 아파치는 C로 짜여졌으며, 아파치에 속해 있는 모듈들도 그러하다. 아파치 모듈을 작성하기 시작하는 가장 좋은 길은 Doug MacEachern과 Lincoln Stein이 쓴 Writing Apache modules with Perl and C (http://www.modperl.com)를 읽는 것이다. 이것은 잘 쓰여졌고 읽기에 쉬우며, 아파치와 펄의 고수 두 명이 함께 쓴 책이다. 위의 링크를 따라가면 이 책의 몇 장이 나와 있는 웹 사이트로 가게 될 것이다. 만일 당신이 책을 살 돈이 없거나 친구에게 이 책을 빌릴 여건도 되지 않는다면, 다른 방법이 있다. 당신은 온라인 상의 아파치 모듈 작성 튜토리얼들을 읽을 수 있다. 아파치 그룹 회원인 Ken Coar의 tutorial and slides online (http://web.golux.com/coar/slides/)을 참고하도록 하자. 아파치 구조의 전반적인 개관은 이곳에서 (http://www.grad.math.uwaterloo.ca/~oadragoi/CS746G/a1/apache_conceptual_arch.html) 찾아볼 수 있다. 아파치 웹 사이트는 API 의 주석 (http://www.apache.org/docs/misc/API.html)이 있

어, 당신의 시작을 확실하게 도와 줄 것이다. 또한 당신은 아파치에 속해 있는 모듈들의 소스 코드를 볼 수 있다. 아파치는 이와 같은 것들을 위해 아주 단순한 것들을 포함하고 있다.

타 언어로 아파치 모듈 작성하기: 다양한 아파치 모듈들이 제 3의 언어로 내부적인 아파치 API에 접근하는 것을 가능하게 한다. 그런 것들 중 가장 유명한 것은 17.3절이다.

만약 당신이 아파치 모듈을 작성하는 방법에 관하여 어떠한 궁금증이 있다면, 아파치 모듈 메일링 리스트 (http://modules.apache.org)에 가입하도록 하라. 당신의 문제를 해결하려 하기 전에 먼저, 이전에 논의된 메시지들을 모두 받아 보도록 하자. 누군가가 당신과 같은 문제에 관한 유용한 해결책을 달아 두었을지도 모른다.

만일 당신이 아파치의 핵심적인 부분을 개발하는 데 관심이 있다면, 아파치 개발자 사이트 (http://dev.apache.org)를 참고할 것을 권한다.

13. 아파치 서적

아파치 관련 서적에 대한 유용한 목록은 다음 (http://www.apache.org/info/apache_books.html)에서 구할 수 있다.

개인적으로 권하는 관련 서적들은 다음과 같다. :

- 당신이 아파치의 내부에 흥미가 있다면 : Perl 과 C로 아파치 모듈 작성하기 (http://www.modperl.com)
- 당신이 아파치를 시작하려 한다면, 책 제목만 보고 시덥지 않은 책을 고르지는 말아야 한다. 아래의 책은 좋은 정보를 갖고 있는 훌륭한 연습이 될 것이다. 아파치 서버 (for dummies) (http://apache-server.com/ForDummies)

14. WebDAV

WebDAV (http://www.webdav.org) 웹사이트에서 : **WebDAV 는 "Web-based Distributed Authoring and Versioning"의 약자이다. 이것은 HTTP 프로토콜을 확장하는 것으로, 수정이 용이하고 원격 웹 서버의 파일을 관리하는 것을 허용한다.**

이것은 MS FrontPage 프로토콜에 대응하는 열린 체제이지만, 몇몇 점에서 더 발전해 있다. 이것은 다른 프로토콜들이 이것을 바탕으로 빌드되는 것을 허용한다. (Subversion website (http://subversion.tigris.org/)를 그 예로서 확인할 수 있다.)

15. 자바 프로젝트

역사적인 이유로 인해, 자바 프로젝트는 java.apache.org와 jakarta.apache.org 양쪽에서 찾아볼 수 있다. 결과적으로 시간이 지나면 모든 자바 프로젝트들은 자카르타 쪽으로 옮겨 가게 될 것이다.

자카르타 프로젝트의 결론은 열려 있고 공동으로 개발되는 자바 플랫폼 기반의 상업적인 수준의 서버 솔루션을 공급하는 것이다.

아파치 커뮤니티에서의 자바는 양적인 면과 질적인 면 모두에서 매우 동적이고 활기찬 것이다.

15.1. 앤트(Ant)

Ant는 make에 대한 자바 환경이라고 생각할 수 있다.이것은 자바 관련 프로젝트들과 함께 큰 성공을 거두고 있다. 개발자들은 셸 명령 대신에 자바를 사용할 수 있다. 이것은, 공용성과 실행가능성의 증가를 뜻한다. Makefile 대신 Ant는 XML 파일을 사용한다. ANT에 관해 더 알기를 원한다면 이곳을 (http://jakarta.apache.org/ant/index.html) 방문해보자.

관련된 이야기

- F19: 자바 코드를 빌드하는데 사용하는 Ant

15.2. ORO 와 Regexp

ORO은 자바를 지원하는 정규식을 공급하는 완성된 패키지이다. 이것은 Perl 5의 정규식을 지원하고 뭉쳐진 표현들과 그 밖의 것들을 지원한다. 이것들 모두는 아파치 라이센스 하에 있다. 당신은 ORO에 관해 이곳 (http://jakarta.apache.org/oro/index.html)에서 정보를 얻을 수 있다. 당신은 또 다른 가벼운 정규식 패키지인 Regexp (http://jakarta.apache.org/regexp/)도 입수할 수 있다.

15.3. 슬라이드

슬라이드는 고수준의 컨텐츠 관리 도구이다. 이것은 제멋대로 놓여 있거나 혹은 외생의 것이서나, 정리된 데이터일 수도 있는 바이너리 컨텐츠에 있어 계층적으로 공급할 수 있다. 추가적으로 슬라이드는 락과 버전 서비스를 통해 보안의 측면에도 도움이 된다.

당신이 만약 WedDAV (http://www.webdav.org)를 이용하고 있다면, 슬라이드는 그것을 확장적으로 사용할 수 있다. 간단히 말하면, 슬라이드는 단일된, 단순한 방법으로 리소스와 정보에 접근할 수 있도록 하는 것이다. 또한 데이터베이스나 파일 시스템 등등에서 사용할 수 있으며, WebDAV 환경 혹은 슬라이드 자체 API 중 원하는 쪽으로 접근할 수 있다.

당신은 슬라이드 홈페이지 (http://jakarta.apache.org/slide/index.html)에서 더 많은 것을 배울 수 있다.

15.4. Struts

Struts는 웹 개발을 위해 Model-View-Controller(MVC)의 디자인 패러다임을 적용하려 하는 아파치 프로젝트이다. 이것은 서블릿 (http://java.sun.com/products/servlet)과 자바 서버 페이

지(JSP) (http://java.sun.com/products/jsp) 기술로 빌드된다. 모델이 되는 부분은 어플리케이션의 내부적인 상황을 보여주는 자바 서버 오브젝트이다. 엔터프라이즈 자바 빈즈는 종종 여기에 사용된다. 보여지는 부분은 정적인 HTML/XML과 Java로 이루어진 JavaServer Pages (JSP)이다. JSP는 개발자들에게 그들이 정의한 새로운 태그를 사용하는 것을 허용한다. 콘트롤러 부분은 클라이언트로부터 받은 요청(GET/POST)을 처리하는 서블릿으로 구성되어있고 모델 위에서 적절한 JSP를 제공하여 뷰를 갱신한다. Struts 프로젝트 페이지 (http://jakarta.apache.org/struts/index.html)에서 더 많은 정보를 얻을 수 있다. .

15.5. 태그 라이브러리

자바 서버 페이지 기술은 개발자들에게 자신들의 태그를 기능적으로 추가하는 것을 허용하고 있다. 태그 라이브러리 프로젝트는 공통적인 표현들을 정리하는 것이으로,SQL 데이터베이스 접근에 사용되는 등의, 공통된 유틸리티들은 위한 태그들을 포함하고 있다.(date같은)

태그 라이브러리에 대하여 더 많이 알고 싶다면 이곳 (http://jakarta.apache.org/taglibs/index.html)을 방문하여 보라. 패키지 안에 더 많은 문서를 포함하고 있다.

15.6. Tomcat

Tomcat는 자카르타 프로젝트의 중요 프로젝트이다. 이 것은 자바 서블릿 2.2와 자바서버페이지(JSP) 1.1기술들의 공식 참조 구현이다.

Tomcat 홈페이지 (http://jakarta.apache.org/tomcat/index.html)에서 더 많은 것을 알수 있다.Tomcat 프로젝트는 Sun Microsystems로부터 코드를 기증받아서 시작되었다.

15.7. Velocity

Velocity는 자바 기반의 템플릿 엔진이다**.** 이는 소스코드**, HTML,** 리포트등을 만들기 위한 단독 유틸로 사용할 수도 있고 템플릿 서비스를 제공하기 위해 다른 시스템과 연동 될 수도 있다**.** Velocity는 자바 코드와 HTML 템플릿을 분리하기 위한 Model View Controller 패러다임을 가지고 있다.

Velocity에 대하여 더 알고 싶다면 이곳 (http://jakarta.apache.org/velocity/index.html)을 방문하라. 15.18절같은 다른 프로젝트의 일부이다

15.8. Watchdog

watchdog프로젝트는 서블릿과 자바서버페이지(JSP) 명세서를 위한 검증 시험을 제공한다. 더 많은 정보는 이곳 (http://jakarta.apache.org/watchdog/index.html)에서 볼 수 있다.

15.9. JServ

Apache JServ (http://java.apache.org/jserv/index.html)는 현 시점에서 유지중이다**.** 이는 새로운 공식 릴리즈가 없을 것이라는 것을 의미한다 단지 요청된 패치를 시험하고 있다**.** 새로운 기능은 추가되지 않는다**.** 자바 서블릿 엔진이나 자바서버페이지**(JSP)**에 관한 최근의 구현을 찾고자 한다면 자카르타 프로젝트에서 가능한 **Java 15.6**절을 사용할 것을 고려해야 한다**.**

15.10. JSSI

JSSI는 자바로 구현한 SSI이다. SSI는 클라이언트에 페이지가 보내지기 전에 처리해야 할 것을 파일에 포함한 태그이다.예를 들면 현재 시간같은 것이다. 더 많은 정보는 이곳 (http://java.apache.org/jservssi/index.html)에서 확인할 수 있다.

15.11. Apache JMeter

Apache JMeter는 기능적인 행동이나 성능을 측정하기 위해 디자인된 **100%** 순수 자바로 작성된 데스크탑 프로그램입니다**.** 원래는 웹 프로그램을 시험하기위해 작성되었으나 지금은 함수들을 시험하는 것으로 확장되었습니다**.**

이 것은 정적,동적 리소스를 시험하거나 즉각적인 가시적 피드백을 얻는데 사용할 수 있다.

이곳 (http://java.apache.org/jmeter/index.html)에서 스크린샷과 많은 공부거리를 볼수 있다.

15.12. Server Pages Foundation Classes

SPFC는 서버기반 프로그램 개발시 일반적인 문제를 해결하기 위한 라이브러리 셋트이다.다음의 두 가지의 사안에 관심을 가진다.

- **HTML**과 **Java**의 혼합: HTML 코드를 생성하거나 자바 코드와 통합될수 있는 클래스 라이브러리를 제공한다.
- **HTTP**는 비연결성 프로토콜이다:SPFC는 세션기능을 제공해서 웹사이트를 여행한 사용자의 기록을 유지할수 있다. 프로그램 개발자는 페이지 생성에 대하여 특별히 세부사항을 걱정할 필요가 없다. 더 많은 보편적인 응용프로그램 관례에 대하여 생각할 수 있다. SPFC를 더 알고 싶으면 이곳 (http://java.apache.org/spfc/index.html) 에 가면 알수 있다.(역자주: 번역 당시 위 URL은 삭제된 뒤였다.)

15.13. Element Construction Set

Element Construction Set (ECS)는 **JAVA API**이며 다양한 마크업 언어를 위한 요소를 생성한다**.HTML 4.0**과 **XML**을 직접지원한다**.** 그러나 임의의 마크업 언어를 위한 태그를 만들도록

확장할 수 있다.

HTML과 자바 코드를 혼합한 깔끔한 해결을 이끌도록 자바 함수 호출을 이용하여 마크업 태그를 생성하는 것을 허용한다. ECS project page (http://java.apache.org/ecs/index.html)에서 더 많은 것을 배울 수 있다. (역자주: 이 주소는 http://jakarta.apache.org/ecs/index.html로 변경되었다.)

15.14. Avalon

펄이나 BSD시스템에 익숙하다면 아발론은 CPAN (http://www.cpan.org)나 자바-아파치 기술의 Ports 모음과 동등하다. 일반 소스 저장소를 위한 가이드라인을 제공하지 않는다.게다가 한가지 단계만 있다: 이는 자바로 작성된 서버 응용프로그램을 위한 일반적인 프레임워크를 작성, 디자인, 발전, 유지하기위한 노력이다. 이는 서버측 자바 프로젝트들을 통합하고 각각을 만드는것을 쉽게 하는 의미를 제공한다.

15.15. JAMES (Java Apache Mail Enterprise Server)

다른 아파치 서버측 기술의 총체로서 JAMES는 현재 가능한 프로토콜(**SMTP,POP3,IMAP,HTTP**)에 기반한 완벽하고 포터블한 기업형 메일 엔진 솔루션으로 디자인된 **100%** 순수 자바서버이다.

더 많은 정보가 필요하면 이 곳 (http://jakarta.apache.org/james/index.html)을 방문하라.

15.16. PicoServer

순수 자바로 작성된 경량급 HTTP/1.0 서버. 프로젝트는 중단된것처럼 보이고 가능한 소스도 없다. 웹 사이트와 CVS는 사용할 수 없다.

15.17. Jetspeed

Jetspeed (http://jakarta.apache.org/jetspeed/site/index.html)는 자바로 작성된 web기반 포탈이다. 서로 다른 자료 소스(XML, SMTP, iCalendar)를 집합하는 것을 허용하는 모듈 API를 가지고 있다.

관련된 이야기:

- TH11: JetSpeed를 이용한 엔터프라이즈 정보를 작성하기

15.18. Turbine

Turbine은 실험적인 자바 개발자들이 빠르게 보안 웹 응용프로그램을 작성하는 것을 허용한다.. Turbine은 자바 실행코드를 실행할 플랫폼과 재사용 가능한 컴포넌트, 아파치 라이선스하에 있는 모든 것을 함께 가져온다. 포함된 다음의 특성들:

- 템플릿 시스템으로의 통합
- MVC 형식 개발
- 접근제어리스트
- 지역화 제공
- 등등...

관심있는 사람은 Turbine 홈페이지 (http://jakarta.apache.org/turbine/)를 방문할 수 있다.

15.19. Jyve

Jyve project (http://java.apache.org/jyve/index.html)turbine을 기반을 작성되었다. 이것은 web 기반 FAQ 시스템을 제공하는 프로그램이다.

15.20. Alexandria

Alexandria는 통합문서관리시스템이다. CVS나 JavaDoc같은 오픈 소스 프로젝트에 공통적인 기술을 가져온다. 최종 목표는 코드 문서화와 공유를 촉진하기 위해 소스 코드와 문서의 통합이다. 더 많은 정보는 이곳 (http://java.apache.org/alexandria/index.html)에서 볼 수 있다. (역자주:이 주소는 http://jakarta.apache.org/alexandria/index.html (http://jakarta.apache.org/alexandria/)로 변경되었다.

관련된 이야기:

- W06:Alexandria의 소개

15.21. Log4j

이 패키지는 자바 응용프로그램이 사용할 수 있는 로깅 프레임워크를 제공한다. 이는 바이너리를 변경하지 않고 실행시에 가능하게 할 수 있다. 또한 성능을 위해 설계되었다. in mind. 이것에 관한 내용은 http://jakarta.apache.org/log4j/docs/index.html에서 찾을 수 있다.

16. XML 프로젝트들

Apache XML 프로젝트사이트에 있으며 목적은 아래와 같다:

- 열린 협동적인 양식으로 개발하기 위한 상업적 능력의 표준 기반 **XML** 솔루션을 제공하기 위하여
- **IETF**나 **W3C**같은 표준에 대하여 구현전망으로부터 피드백을 제공하기 위해**.**
- 아파치 프로젝트안에서 **XML**관련 활동의 초점이 되기 위해서

이 프로젝트의 홈페이지는 http://xml.apache.org에 있다. 이는 각각의 서브프로젝트를 위한 포괄체이다.

16.1. XML의 소개

이는 XML의 빠른 소개이다. XML에 대하여 더 많이 알고 싶다면 XML 홈페이지 (http://www.xml.com)에서 시작하라. XML은 태그와 속성을 사용하여 구조화된 객체를 설명하는 마크업 언어이다(HTML을 생각하라) 내용은 가시화와는 분리되어있지만 디스플레이 방식(셀폰,HTML,텍스트)을 선택하거나 변경할 수 있다. XML표준은 단지 태그와 속성이 정렬되는 방법을 설명한 것이지 의미하는 이름을 설명한 것은 아니다. 아파치(그룹)에서는 아래의 절에 설명된 도구들을 제공한다.

16.2. Xerces

Xerces프로젝트는 자바, C, 펄을 포함하는 다양한 언어를 위한 XML파서를 제공한다. 펄 바인딩은 C++소스에 기반한다. Xerces의 TCL 바인딩은 Steve Ball이 만든TclXML (http://tclxml.sourceforge.net/) 의 2.0버전에 있다. SourceForge (http://sourceforge.net/projects/tclxml) 프로젝트 페이지를 통해서 가능하다. XML파서는 XML 문서를 표제 접근하는 데 사용하는 도구이다. 아래는 Xerces에 의해 제공되는 표준들에 대한 설명이다:

- DOM (http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/level-one-core.html): DOM이란 문서기반모델(Document Object Model)을 의미한다. XML문서들은 중첩된 태그들에 의해 계층적인 구조로 되어있다. XML문서들은 나무구조 비슷한 인터페이스로 접근할수 있다. 처리과정은 아래와 같다.

  - 문서 분석
  - 구조 작성
  - 노드 추가/삭제/변경
  - 구조 정렬

- SAX (http://www.megginson.com/SAX/index.html):XML을 위한 단순 API. 이는 스트림기반의 API이다. 이는 계산된(encountered) 요소들로부터 콜백을 얻을수 있다는 것을 의미하며 이 콜백들은 예를 들어 DOM 트리를 생성하는데 사용할 수 있다.
- XML 주소공간 (http://www.w3.org/TR/REC-xml-names/)
- XML Schema: XML 표준은 문서를 작성하는 신텍스를 제공한다. XML Schema는 XML 문서(semantics)의 **contents**를 정의하기 위한 도구를 제공한다. 이는 문서안에서 특정요소가 10과 20사이의 정수이이여야 한다고 정의하는 것을 허용한다.

Xerces XML 프로젝트의 초기 코드는 IBM에서 제안되었다. 이에대한 자세한 정보는 Xerces Java (http://xml.apache.org/xerces-j/index.html),Xerces C (http://xml.apache.org/xerces-c/index.html)and Xerces Perl (http://xml.apache.org/xerces-p/index.html)홈페이지에서 찾을수있다.

16.3. Xalan

Xalan은 Java나 C++을 위한 XSLT 파서이다. XSL은 XML을 위한 스타일시트언어이다.여기서 T는 변환을 의미한다. XML은 구조적인 자료(정보)를 저장하는데 좋다. 때때로 이 자료를 사용자에게 표시하거나 다른 변환을 적용할 필요가 있다. Xalan은 원본 XML문서를 받아서 스타일시트를 이용한 변환정보를 읽은 후 HTML, 보통 텍스트나 또다른 XML 문서로 출력한다. Xalan에 대해서Xalan Java (http://xml.apache.org/xalan-j/index.html) and Xalan C++ (http://xml.apache.org/xalan-c/index.html)프로젝트 홈에서 더 많은 것을 공부할수 있다..

16.4. FOP

웹사이트 에서, **FOP**는 형식화된 객체 트리를 읽고 이를 **PDF**문서로 변환하는 자바 프로그램이다**..** 그래서 FOP는 Xalan이 HTML이나 텍스트를 이용하여서 하는 것과 비슷한 방법으로, XML문서를 읽고 PDF를 출력한다. FOP에 대해서는 이곳 (http://xml.apache.org/fop)에서 더 자세히 알수 있다.

16.5. Cocoon

Cocoon은 이해가능한 출판물을 제공하기 위해 Xerces, Xalan과 FOP같은 다른 아파치 XML 기술들에 효력이 있다. 웹사이트에 설명한대로 내용과 로직과 표현방식을 분리한다:

- **XML** 생성: **XML** 파일은 내용 소유자에 의해 만들어진다**. XML** 컨텐트는 특별히 선택된 **DTD/namespace** 보다는 오히려 처리하는데 특별한 지식을 요구하지 않는다**.** 이 계층은 항상 사람에 의하여 보통의 텍스트 편집기나 **XML**관련 도구**/**편집기를 이용하여 직접 수행된다**.**
- **XML** 처리 발생기:논리적인 부분은 내용 파일과 분리되어있다**.**

- **XSL** 변역: 만들어진 문서는 **XSL** 스타일시트를 적용하거나 특별한 형식**(HTML, PDF, XML, WML, XHTML)**로 형태화함으로서 변역된다**.**

cocoon에 대하여 더 알고 싶다면 Coon 홈페이지 (http://xml.apache.org/cocoon/index.html)를 방문하라.

16.6. Xang

Xang프로젝트의 목적은 개발자들이 웹을 위한 상업적 성능을 가진 **XML**관련 응용프로그램을 만들기 쉽게 만드는데 있다**.** 프로그램 구조는 자바스크립트같은 것으로 쓰여진 계층적인 XML파일안에 정의되어있다. 이 파일은 (XML 파일, 자바 플러그인등등이 될 수 있는) 자료를 접근하는 방법을 정의한다. Xang 엔진은 HTTP 요청들을 적절한 취급자(핸들러)로의 사상을 처리한다. Xang에 대하여 자세히 알고 싶다면프로젝트 홈페이지 (http://xml.apache.org/xang/)를 방문하라.

16.7. SOAP

아파치 **SOAP**(단순 객체 접근 규약)은 **W3C**에 제출된**SOAP submission (http://www.w3.org/TR/SOAP)**의 구현이다**.** 이 것은 **IBM**의 **SOAP4J**의 구현에 기반하며 대체한다**.**.

W3C 초벌 명세서에 의하면**: SOAP**는 분산 환경에서 정보의 교환을 위한 경량급 규약이다**.** 이 것은 다음의 세부분으로 구성된 **XML** 기반 규약이다**.**:

- 메시지 표현방법과 처리방법을 위한 하부 구조를 정의한 구조,
- 프로그램에 정의된 자료형의 객체를 표현하는 번역규칙 집합,
- 그리고 원격 프로세스 호출과 응답을 나타내기 위한 규정.

SOAP는 XML기반 원격 프로시저 호출이나 CORBA 시스템을 생각할 수 있다. 이것은 HTTP와 XML기반이다. 이것은 다른시스템과 비교하는게 더 자세하고 느리다는 것을 의미한다. 다시 말해서 대부분의 언어는 HTTP와 XML을 위한 모듈을 가지고 있기 때문에 다양한 언어(C, Java, Perl, Python, Tcl, etc.)를 위한 클라이언트와 서버의 개발을 상호운영하거나 디버깅하는 것을 쉽게 한다. 더 많은 것을 배우려면아파치 SOAP 홈페이지 (http://xml.apache.org/soap/)을 방문하라.

관련된 이야기

- W02: Rub-a-dub-dub-dubya: SOAP and the Web

16.8. Batik

Batik은 다양한 목적(보기,생성, 변형)을 위해서 **Scalable Vector Graphics (SVG) (http://www.w3.org/TR/SVG/)**에서 이미지를 사용하기 원하는 응용프로그램을 위한 자바기반 도구모음이다**.**

이는 XML 중심이고 W3C 명세서에 따른다. 그래픽관련 구성요소를 제공하여 다른 아파치 프로젝트와는 다른 전형적이 아니다. Batik는 사용자 태그를 통해 하부구조를 확장하는 고리를 제공하고 SVG로부터 JPEG나 PNG같은 다른 형식으로 변환하는 것을 허용한다.

Batik 홈페이지 (http://xml.apache.org/batik/index.html)

관련된 이야기

- W14: Batik 프로젝트 소개

16.9. Crimson

Crimson은 다양한 인터페이스를 통한 XML 1.0을 지원하는 선택적이고 자바 기반의 XML 파서입니다. 이 것은 Sun 프로젝트에 포함되어있는 파서이며 Xerces 2가 발표되기 전까지 임시 단계입니다.

Crimson 홈페이지 (http://xml.apache.org/crimson/index.html)

관련된 이야기

- TH08: XML을 처리하는 자바 API(JAXP) 1.1

16.10. 다른 XML 프로젝트

아파치 XML 그룹하에 있지 않는 아파치와 XML 기반의 프로젝트 들이 있다.

- mod_xslt (http://modxslt.sourceforge.net/).이것은 XML/XSL기반 문서를 전송하는 C기반의 모듈이다. 또한 GPL라이선스를 따른다.
- AxKit (http://axkit.org)는 mod_perl and Apache를 위한 XML기반의 응용서버이다.이는 컨텐트와 표현의 분리를 허용한다.

관련된 이야기

- TH04: AxKit - 아파치를 위한 XML 문서 서버

17. Perl

Perl과 Apache는 강력하고 널리 알려진 조합이다. 여기 두가지 기술을 이용한 여러 프로젝트들이 있다.

17.1. Embperl

Embperl은 HTML 페이지 안에 Perl을 포함하는 것을 허용한다. 이 페이지들은 클라이언트에 보내지기 전에 서버에서 처리된다. 18절와 비슷하다. 더 많은 것은이곳 (http://perl.apache.org/embperl/index.html)에서 알수 있습니다.

17.2. Mason

Mason 프로젝트 (http://www.masonhq.com/)는 재사용가능한 컴포넌트 모델을 사용하기 위하여 HTML안에서 Perl을 포함한다. 이것은 캐시나 템플릿을 사용하는 것을 허용한다.

17.3. Mod_Perl

Mod_perl (http://perl.apache.org/)은 가장 노련하고 성공 가능한 아파치 프로젝트의 하나이다. 이는 아파치내에 Perl 처리기를 내장하고 Perl이 웹서버를 접근하는 것을 허용한다. 이는 Perl로 쓰여지거나 Perl과 C을 혼용하여 쓰여진 모듈을 허용한다. 아파치 1.3에서 서버가 멀티프로세스 기반이기 때문에 하나의 처리기는 각각의 자식프로세스에 포함되어있어야 한다. 아파치 2.0은 최근의 Perl처럼 멀티 스레드 방식이다. Mod_perl의 다음 판에서는 이러한 이점을 최대한 이용하여 코드, 자료, 세션상태등을 공유하는 것을 허용할 것이다. 이는 더 빠르고 작아지는 결과를 만들어낸다.

를 같이 보기 바랍니다.

18. PHP

PHP (http://www.php.net)로 부터: **PHP**는 여러 플랫폼을 지원하는 서버측 **HTML** 스크립트 언어이다**.** PHP는 Perl, Python이나 Tcl같은 스크립트 언어이다. 아파치를 위한 가장 유명한 모듈 (http://www.securityspace.com/s_survey/data/man.200102/apachemods.html)이며 이는 다양한 이유에서이다:

- 배우는 곡선이 매우 낮다.
- 많은 문서
- 다양한 데이터베이스 지원
- 모듈 방식

PHP는 모듈방식 설계이다. 여기 제공되는 모듈이 있다:

- Oracle, ODBC, MySQL, mSQL, PostgreSQL, MS-SQL 서버등을 위한 데이터베이스 연결 더 많은 것은 PHP website (http://www.php.net)을 확인하시오.
- XML 지원
- 파일 전송: FTP
- HTTP
- 디렉토리 서비스: LDAP
- 메일 지원: IMAP, POP3, NNTP
- PDF 문서 생성
- CORBA

등등..... 단지 필요한 모듈만 컴파일하고 사용할 필요가 있다.

PHP는 다른 웹서버나 외부 CGI처럼 아파치와 같이 사용할 수 있다. 이 것은 플렛폼에 영향을 받지 않고 여러 Unix와 Windows에서 실행할 수 있다.

Window를 사용한다면 아마도 동적 서버 페이지(ASP)와 MS-SQL 서버와함께 인터넷 정보서버(IIS)를 사용하고 있을 것이다. 이 셋을 위한 Unix에서의 일반적인 모습은 PHP와 MySQL을 이용한 아파치 서버이다. PHP가 동작하는 것:

- IIS 마이크로소프트 IIS와 아파치
- MySQL 과 MS-SQL 서버
- Unix와 Windows에서

Microsoft중심의 프로그램으로부터 더 보안에 충실하고 , 안정적이고, 고성능 (FreeBSD (http://www.freebsd.org), Solaris (http://www.sun.com),Linux (http://www.linux.com) or OpenBSD (http://www.openbsd.com)같은) Unix 기반의 프로그램으로의 가장 좋은 방법이 있다.

19. Python

Python은 Perl이나 Tcl과 비슷한 스크립트 언어이다. 아파치 웹서버에 Python을 포함하는 다양한 모듈:

- Mod Python (http://www.modpython.org)
- Mod Snake (http://modsnake.sourceforge.net): 아파치 1.3.x 와 2.0이상에서 둘다 동작한다.

두 모듈은 만약 Python으로 새로운 아파치 모듈을 작성할 계획을 만들거나 기존의 Python CGI를 더 빠르게 동작시킬 계획이 있을 때 유용하다. Mod Snake 18절 처럼 Python을 HTML안에 포함하는 것을 허용한다.

관련 문건:

- F08: mod_snake: Python을 이용한 생산성 향상

20. Tcl

The Tcl Apache project (http://tcl.apache.org)는 아파치 웹서버에 Tcl이 통합되었다. Tcl은 경량급 확장가능한 스크립트 언어이다. Tcl에 대하여 더 알고 싶다면 이곳 (http://tcl.activestate.com/)을 방문하여보라. 현재 아파치 Tcl하에 있는 여러 모듈이 있다:

- Mod_dtcl (http://tcl.apache.org/mod_dtcl/)은 18절처럼 HTML 페이지에 Tcl을 포함하는 것을 허용한다.
- Neowebscript (http://tcl.apache.org/neowebscript/)은 유사하게 접근한다
- Mod_tcl (http://tcl.apache.org/mod_tcl/mod_tcl.html)은 mod_perl과 유사하게 접근하며 아파치 1.3.x와 2.x 둘다 동작한다.

다른 Tcl 아파치 프로젝트는 WebSH (http://websh.com/)에서 볼 수 있다.

21. 다른 언어를 위한 모듈들

이 문서는 Perl, Python, PHP같은 유명한 서버측 언어를 위한 모듈만 설명되어있다. 다른 언어 모듈(Javascript, Haskell 등등)은 Apache modules directory (http://modules.apache.org)에서 발견할 수있다.

22. Apache 2.0

아파치의 현재 판(1.3 계열)은 프로세스 기반이다. 이는 동시 요청에 응답하려고 자신을 여러번 복제한다.자식들은 서로서로 고립되어 있다. 이는 믿을 수 있다.: 만약 모듈이 잘못되면, 부모 프로세스는 자식을 제거한다 이는 서버 전체가 아니라 제공하던 요청만 영향을 받는다. Threads는 경량급 프로세스와 유사하다. Threads는 공통 자료를 공유할 수 있다. Thread가 잘못되면 다른 threads를 믿을 수 없게 하고 서버 전체가 종료 된다.다시 말해서 thread 모델은 더 빠르고 마른 웹서버를 허용한다. 아파치 2.0은 두 가지 중에서 최선책을 찾아내었다. 사용자가 프로세스의 수와 프로세스당 thread의 수를 정의할 수 있게 하였다. 아파치 2.0은 아파치의 이식가능성을 증가하기 위해 APR(Apache Portable Runtime)을 소개하였다. 마지막으로 층을 이룬 I/O는 아파치 개발에 모듈방식의 층을 만들었다.

23. Netscape (iPlanet) web servers로부터의 이식

작업의 크기는 사용자 모듈을 NSAPI로부터 아파치 API로 변경하는 것으로 귀속된다. 거의 모든 서버측 기술(Java, Perl, CGIs) 은 없거나 적은 변경만으로도 이식가능하다. Netscape는 LDAP 서버와 단단히 통합되어있다. 또한 Module for Apache (http://modules.apache.org)에서 LDAP 모듈에 흥미를 가지게 될 수 도 있다.

24. Microsoft IIS로부터의 이식

사람들이 IIS로부터 아파치로 이식하는 일반적인 이유는 안정성과 성능 그리고 보안성을 포함한다. 이는 부분적이다. 왜냐하면 아파치를 사용하는 많은 사람들은 Unix계열(Solaris,FreeBSD, 리눅스)의 OS에서 사용하고 있기 때문이다. 운이 좋게 아파치는 다중플랫폼이고 Unix와 Windows에서 동작을 한다. 또한 인식가능한 이식방법을 제공한다.

일반적인 Windows기반의 Coldfusion이나 동적 서버 페이지같은 웹 개발환경은 Unix용이나 호환가능한 환경을 가지고 있다.(일부는 상용이고 일부는 무료로 가능하다):

- Coldfusion for Linux (http://www.macromedia.com/Products/coldfusion/)
- Perl ASP module (http://www.apache-asp.org/)
- Halcyon ASP (http://www.halcyonsoft.com/)

Windows용 아파치는 또한 ISAPI 인터페이스를 지원한다.

Windows환경(IIS + ASP + MS-SQL 서버)로부터 완벽한 오픈소스 프로그램으로 가기를 원한다면 동등한(매우 대중적인) 조합은 아파치 + PHP + MySQL (http://www.mysql.com) 이나 PostgresSQL (http://www.postgresql.org)이다.PHP에 대하여 더 배울 수 있다. 18절

Windows를 위한 지원은 새 2.0 아파치에서 매우 향상 되었으나 현재 이 글을 쓰고있는 상태에서는 beta상태이다.

25. Links

더 많은 아파치 관련 자료들

25.1. 웹사이트

- 아파치 (http://www.apache.org)
- Apache modules directory (http://modules.apache.org)
- Apache today (http://www.apachetoday.com)

- Slashdot의 Apache 섹션 (http://slashdot.org/index.pl?section=apache)

25.2. 자바 응용프로그램 서버

여기 아파치에 포함되거나 아파치와 잘 동작한다고 알려진 오픈소스 응용프로그램서버가 있다.

- Resin (http://www.caucho.com/): Servlets, JSP, XSL
- Enhydra (http://www.enhydra.com/): Java/XML application server.
- JBoss (http://www.jboss.org/): Enterprise Java Beans container, J2EE

26. 저자와의 접촉

ridruejo at apache.org (ridruejo@apache.org)에서 저자를 만날수 있습니다. 제안 이나 수정은 환영합니다. 다만 아파치 설치에 대한 처리법을 물어보기 위해 메세지를 보내지 말아 주십시요. 나는 대역을 가지고 있지 않습니다. 그리고 당신의 메일은 대체로 무시될 것입니다. 지원이 필요하다면:

- error 로그를 살펴보거나, 문서를 읽고 특히 FAQ (http://www.apache.org/docs/misc/FAQ.html)를 읽고.
- 이곳 (http://groups.google.com)comp.infosystems.www.servers.unix으로 가서 비슷한 문제를 검색하세요.
- 아직도 해결이 되지 않았다면 할수 있는 한 많은 정보 - 관련된 error_log entries와 수행했던 단계들 - 를 뉴스그룹에 올리십시요. 이는 누군가가 당신의 문제에 응답할 기회를 증가시킨다.

상업적 지원을 원한다면 , 아파치를 위한 전문가적 지원을 제공할 (물론 유로로)커벌런트사 (http://www.covalent.net)와 접촉할 것을 고려하라. 리눅스에서 아파치를 사용한다면, 당신의 리눅스 배포 회사는 아파치를 포함한 설명을 지원해줄것이다.

26.1. 번역

이 문서의 번역에 동참하고 싶으신 분은 SGML 소스를 이용해야 합니다. 추가정보를 원하면 이곳 (http://www.linuxdoc.org)을 확인하십시오. 가장 최신 버전을 확인하고 싶다면 저자에게 메일을 보내면 됩니다.