



웹기반 내장형 응용

김 인 홍[†] 이 정 배^{††} William Weinberg^{†††}

◆ 목 차 ◆

- | | |
|----------------|--------|
| 1. 내장형 웹서버 GUI | 3. CGI |
| 2. 기본 HTTP 서버 | |

최근 모든 내장형(embedded) 응용들이 급속히 웹 기반으로 바뀌어 가고 있다. 사무자동화, 과학, 의료, 멀티미디어, 보안시스템 등 망에 접속된 모든 최종사용자들은 같은 공간 안에 있든, 같은 건물 내에 있든, 혹은 다른 도시에 위치하든 인터넷이나 인트라넷에의 접속장치를 사용하게 된다.

예를 들어, 대형 출판물사업에서도 원고의 접수나 작업일정을 원격으로 처리하기 시작했고 새로운 의료장치들은 의사들이 원격지나 오지에 있는 환자의 상태를 웹을 통해 진단할 수 있게 되었다. 자동화된 건물이나 원격보안 시스템은 LAN이나 WAN 상에서 건물내의 모든 출입문을 상태를 감시할 수가 있게 되는 등 그 활용성은 무한하다.

Agranat, Spyglass 그리고 프리웨어인 Apache 등 많은 내장형 서버 솔루션이 이미 HTTP 서버를 도입하고 있다. 내장형 어플리케이션에서의 HTTP 서비스, CGI 와 API, memory footprint, HTML 및 applet compression, OS threading model, security 그리고 licensing등에 대해 알아본다.

1. 내장형 웹서버 GUI

웹을 기반으로 하는 내장형 시스템의 새로운 사용자 접속환경의 변화는 "netification" 이라는 단어로 표현될 수가 있겠다. 손잡이니, 다이얼이니 하는 조정장치들 에서 부터 X-window를 기반의 고성능 모니터 환경들이 이제는 더 이상 필요치 않고 내장형 시스템 자체에 소규모의 HTTP 서버를 장착함으로써 웹 기반 인터페이스는 간소화될 수 있다. 이러한 "pico server" 들은 인터넷이나 인트라넷 상에서, static 혹은 dynamic HTML 이나 자바 애플릿에서 부터 넷스케이프, 익스플로러 등 표준 웹 브라우저까지 두루 활용할 수 있게 해준다. 이러한 client-server 구도는 사용자 인터페이스의 개발을 용이하게 하고 platform 바뀔 때마다 front-end를 새로이 작성해야 하는 등 종래의 비용부담을 현저하게 줄여준다.

2. 기본 HTTP 서버

기본적인 HTTP 서버는 다음의 간단한 알고리즘을 갖는다. 즉,

- ◆ Open a socket
- ◆ Listen for requests
- ◆ Based on the type of request, blast back

† 정 회 원 : 하이콤 리서치 대표

†† 종신회원 : 부산외국어대학교 부교수

††† 정 회 원 : Lynx Real-Time Systems Technical Marketing Manager

content or accept client input

◆ Listen again...

주어진 내장형 환경에서 가용한 TCP/IP 스택 API에 따라, connection의 시도를 인식하기 위해 polling을 하거나 UNIX signal을 사용한다. connection이 수용되면, 서버는 기본 HTTP 방식을 서비스하게 된다.

HTTP는 다음의 기본적인 방식을 지원한다.

- GET get a specified document
 resolve a URL
 generate a set of header
- HEAD returns the HTTP header
- POST receive form content

프로토콜 측면에서만 볼때는 이로써 충분하겠지만, 내장형 시스템에서는 HTTP 1.1에서 추가적으로 지원하는 여러가지 기능들 중 PUT 과 persistent connection이 중요하다고 하겠다. PUT 기능을 이용하여 content uploading, 즉 필요한 파일을 FTP를 이용하듯이 서버에 올리고, persistent connection이 가능하게 되면 클라이언트가 connection을 open 하여 서버로 하여금 복수 요청된 서비스를 처리하게 하는 것이다. 따라서 reopen과 close를 반복해야 하는 overhead를 줄일 수가 있겠다.

3. CGI

CGI(Common Gateway Interface)는 호스트의 resource에의 상호접속을 위한 일련의 서비스이다. 그러나 CGI는 content-orient 된 웹 디자이너나 내장형 시스템 프로그래머에게는 그리 우호적이지 못하다. CGI는 기본적으로 필드명을 패스하고

HTML로부터 content를 서버에 전달하는 데에 초점을 둔다. 허나 일반적인 어플리케이션들은 서치 엔진에 query를 전하거나, 내장형 환경에서 configuration parameter 나 control 정보를 규명하는 것이 중요하다. HTTP로 작성되어 사용자측에 다이얼로그 박스 및 버튼 형태의 인터페이스를 제공하는 방식이 일반화되었다. Shell, standard I/O 및 파일 시스템이 지원되는 UNIX-like OS하에서는 Apache 와 같은 서버를 이용하면 간단명료하게 구현할 수가 있으나 deep-내장형 시스템의 제한적인 resource상에서는 그 기능을 제대로 갖출 수가 없다. 다른 방법은 시스템 RAM에 완벽한 HTTP image를 설치하고 byte-by-byte로 socket에 전송하는 방식을 고려할 수 있겠으나 이는 현저한 resource낭비를 초래한다.

여기서 소개될 pico 서버들은 C 언어 interface를 통해 scaleable CGI 솔루션을 제공하므로써 이러한 문제들을 해결하고 있다. 즉, 정형적인 데이터의 비정형적인 스트림을 받아서 parsing하는 대신 이미 포매팅 및 type conversion이 완료된 형태로 C struct로 구현할 수 있게 하는 것이다. 또한 이 preprocessing 기법은 웹 페이지의 내용이나 태그들을 축약하여 최적화해 주게 된다.

이러한 C-level 인터페이스는 시간을 절약하게 하고 CGI 프로그램을 간략화 해 준다. 그러나 이들은 각기 고유의 API를 가지며 pico 서버들간이나 오픈 서버로의 호환성문제가 따른다.

Resource Utilization

차후 상세히 설명하게 될 Pico 서버들은 표준 32-bit Architecture 하에서 적절한 프로그램 기억용량만으로 그 기능을 충족할 수가 있다. 코드 footprint는 최소한의 기능을 갖추는 데는 약 10K(CISC기준임, RISC의 경우 30-100% 증가됨), 모든 feature들을 구현키위해서는 100K 남짓하다

고 본다. Full-feature의 경우에는 security 기능, SNMP 서비스 및 여러 가지 신호기술들이 포함된다. 데이터 공간은 버퍼의 크기나 허용된 HTTP 커넥션의 수(일반적으로 커넥션당 1~2K정도로 봄)에 따라 다양하게 요구되어 진다. 반면 Apache의 경우 그 configuration에 따라 300K 정도에서부터 1MB이상의 크기를 갖는다.

Pico 서버들은 콘텐츠가 차지하는 공간에 대해서도 다양한 compression 기법을 통해 부담을 줄여준다.

Page Compression

이미지, 자바 애플릿과 같은 일체형으로 구현된 기능이나 HTML 페이지들(dynamic이든 static이든)은 상당한 시스템 resource를 소모한다. 파일 시스템에 read/write하거나, media의 전환, 기억장치나 디스크공간 등 만 해도 엄청나다. 따라서 어떤 형태로든 웹 콘텐츠의 압축은 매우 중요한 이슈가 된다. HTML(ASCII) source나 텍스트 파일, GIF 이미지, 사운드, 자바 애플릿의 바이트 코드 등 일체형 데이터의 저장 및 공급은 전통적인 압축기법으로도 쓸만하다. 사실 ZIP 형태의 압축 알고리즘을 이용하여 ROM, RAM, 하드디스크, CD-ROM 따위로부터 안정적으로 압축된 page나 데이터를 운용할 수가 있겠으나 이 기법은 computing power를 요구하며 짧은 텍스트 파일이나 아니면 JPEG과 같은 압축 이미지 형태나에 따라 그 효과가 크게 다를 것이다.

그 대안으로 static 혹은 dynamic HTML 을 압축하는데에 directory/token 방식을 도입하는 것이다. 즉, HTML 태그 pair(예를 들어 <BODY> and </BODY>)를 short 혹은 byte value로 저장하고 운용시에만 확장하는 방법이다. 좀더 복잡한 attribute의 예로서, <BODY BGCOLOR="FFFF40" TEXT="000000" LINK="#0000EE" VLINK="#551A8B">와 같은 경우엔 template로 압축할 수가 있겠다.

Allegro의 RomPager는 한층 더 나아가서 token들을 일반 텍스트 스트링으로 배정하고 페이지를 운용할 때 스트링사전으로부터 전체 스트링을 가져오는 기법을 활용하고 있다. 이는 다국적 환경에도 스트링사전을 치환하기만 하면 간단하게 적용될 수가 있겠다.

페이지 출력을 버퍼링하지 않는 dynamic HTML은 dynamic value의 계산을 할 때나 페이지를 운용할 때나 매우 복잡하므로 압축방식을 선택하기 전에 그 trade-off를 면밀히 검토해야 하겠다.

Connection Management 및 System Scheduling

Apache와 같은 범용 HTTP 서버는 scheduling에 있어서 유리하며 UNIX 혹은 WindowsNT등에서 그 시스템 자체를 모델링한다. Resource를 관리하고 성능을 개선키 위해서 Apache서버는 새로운 커넥션을 다루기 위해 child process를 우선적으로 생성하여 load-increase/load-decrease 방식으로 유지/소멸 시킨다. 따라서 UNIX/POSIX 프로세서 모델에 의존적이며 static task 모델에 적용하기는 어렵다.

Deep-내장형 시스템에서는 보다 강력한 real-time executive나 OS가 없는 경량급 HTTP서버를 적용하여 그 자체의 내장형 scheduler를 이용하여 multiple 커넥션을 지원한다. 대부분의 웹 디바이스 Pico 서버들은 native OS의 threading 방식으로 multiple 커넥션을 처리하는 바, 전체 서버가 단일 Realtime OS나 thread 공간 내에서 동작한다. 복수개의 client로 부터 동시 요청이 요구될 경우, FIFO(pending된 커넥션을 TCP/IP queue에 저장) 방식을 사용하거나 서버 자체의 내부 threading에 의해 동작케 한다.

Porting

아주 간단한 내장형 웹 서버라 할지라도 내장형 시스템상에서 동작되기 위해서는 다양한 기본 interface를 갖추어야 한다. 즉,

- ◆ 서버가 어떻게 구성되었는가?
- ◆ 어디에 페이지나 웹 콘텐츠가 저장되었는가?
- ◆ 콘텐츠가 어떤 경로로 서버로부터 client에 전달되는가?

서버를 구현하는 것이 하찮은 일로 보일 지도 모르나, 새로운 환경에 Apache 서버를 탑재하고 재구성하는 일은 그리 간단하지가 않다.

Virtual File Systems

Native/hosted HTTP 서버 시스템상에서의 content 저장은 비교적 간단하다. URL로 부터 파일을 open 하여 socket에 쏟아 부으면 그만이다. 내장형 시스템에서, 운용되는 페이지나, 그래픽, 사운드, 자바 애플릿들은 대체로 ROM 이나 flash 메모리에 상주하며 그 일부 또는 전부가 RAM에 옮겨진다.

호환성을 갖기 위해 "Virtual File System"을 가정하거나 저장장치에 block-level 인터페이스를 도입한다. 만약 UNIX/POSIX API가 지원되면 대부분의 pico 서버들은 source level 호환성을 가지며 단순 recompiling만으로 탑재가 가능할 수도 있겠다.

파일 시스템이 약하거나 고유의 RTOS 자체 파일 시스템의 경우 pico 서버가 제공하는 block read/write 인터페이스를 RAM/ROM /flash interface routine에 맵핑하거나, 기존 내장형 파일 시스템의 system call을 개조하여 구현한다.

Protocol Stack과 API

대부분의 상용 RTOS나 실용제품들은 일종의 TCP/IP stack을 선택사항으로 제공하여 그들 operating system을 보완하고 있다. 여기서 소개되는 HTTP 서버들은 모두 TCP/IP 인터페이스 파일이 제공되며 Berkeley-like socket 서비스를 기본으로 갖고 있다. 만일 API가 BSD 4.3/4.4를 완전 충족하지 못하더라도 사용자가 간단히 주어진 네트워킹 API에 맞출 수가 있다.

Arganat 에서도 EmStack이라 불리는 embeddable TCP/IP를 제공하여 상용 시스템이나 자체 stack을 갖지 못한 고유 시스템에 적용할 수 있다.

Commercial Options

최근 수종의 상용 서버들이 각기 나름대로의 장단점을 갖고 출시되고 있으며, 대부분이 작은 크기의 footprint, page compression, security/ authentication 기능, 그리고 custom 파일 시스템이나 TCP/IP stack에의 탑재기능을 제공한다. 가격이나 licensing 비용은 다양하나 대부분 singlecopy licensing, buy-out 그리고 source code를 제공하고 있다. 다음은 각 공급 사별 제품에 대해 요약한다.

Agranat - EmWeb 과 EmStack

arganat의 EmWeb은 dynamic content 출력과 CGI-type 입력등 실로 독특한 접근방식을 사용하고 있다. 텍스트 스트링을 return value로 갖는 C code나 function call을 포함하는 고유한 tag를 설정하여 HTML 스트림을 client에 운용케하는 HTML-to-C preprocessor를 이용하여 SSI나 고비용의 HTML 이미지 페인팅을 하지 않아도 되게 한다. 통신 프로토콜을 갖추지 못한 low-end application이나 사용자를 위해서는 EmStack 이라는 TCP/IP stack을 제공한다. 그러나 내장형 시스템 개발자나 설계자는 IP, PPP, EtherNet 혹은 자체고유의 전송방식에 맞도록 물리계층의 드라이버를 어떤 형태로든 만들어야 한다.

Allegro - RomPager 와 RomMailer

최소의 footprint/풍부한 기능을 구가한 Allegro는 라우터, hub등 통신장비나, 고성능 프린터등을 목표시스템으로 하고 있다. RomPager의 특징점은 HTML 압축방식과 다국적언어 지원기능을 들 수 있다. 또한 off-load storage기능을 갖고 그래픽이

나 자바 애플릿등을 포함하는 대형 파일을 위한 GET request 의 redirection을 허용한다.

RomMailer는 내장형 시스템에서 "e-mail home" 기능을 제공하여 시스템 비정상상태를 관리자에게 알리거나 traffic report등을 주기적으로 발송하게 할 수 있다.

Magma Lava Family

Lava는 거의 최초로 출시된 서버이며 현재 6개의 OS 플랫폼과 9개의 프로세서에 동작되고 있으며 두 개의 유력한 RTOS 공급사들이 이를 자사 제품에 탑재하고 있다. 얼마전까지 Magma 는 HTML에 초점을 맞추고 있었으나 지금은 자바 애플릿, Lava-App, LavaPower, LavaPC 등 다양한 솔루션을 공급하며 SSL(Secure Socket Layer)를 최근 발표했다.

Quiotix QEWS

Quiotix 내장형 웹서버 즉, QEWS는 preprocessor인 HTML compiler를 제공한다. Virtual file system 에서 URL 및 URL tree handler를 사용하며 call이나 entry point를 자동생성하여 HTML 로부터 input value를 추출한다.

또한 QEWS는 호환성이 뛰어나 새로운 RTOS에의 탑재가 매우 용이하다.

SpyGlass MicroServer

SpyGlass는 이미 잘 알려진 회사이나 MicroServer보다는 Device Mosaic으로 유명하다.

MicroServer는 다양한 서비스를 제공하는 바, 그 footprint는 비교적 크다. 그러나 authentication, security, 내부 thread HTTP engine, Application Development Interface(ADI), 그리고 다양한 모듈과

개발툴들이 제공된다.

또한 소규모 서버를 위한 single-user 솔루션으로 약 10K 크기의 제품도 공급한다.

Web Devices(구 CNIT) Pico Server

웹 디바이스사의 Pico Server 는 security 및 authentication 부분을 완성한 첫 제품이라 할 수 있다. encrypted PUSH 기능이 독특하며 Netscape 서버에 일반 CGI나 plug-in 인터페이스를 선택적으로 적용할 수 있게 해 준다. Pico 서버는 복수 사용자의 동시서비스 요청 시 native OS threading 에서 현저한 성능향상을 기대할 수 있다.

웹 디바이스사는 고유의 내장형 executive 기술인 PICOS와 scaleable 내장형 browser를 함께 공급하고 있다.

Apache

Apache 웹 서버는 가장 세상에 널리 알려지고 사용되어 지고 있으며, 다양한 OS를 지원하고 있다. Internet 사용자에게 기능적으로나 구조적으로 이해되고 있어 많은 자료들이 출판되었다. 또한 무엇보다도 무료로 제공받을 수 있다.

그러나 모든 내장형 시스템에 호환하지는 않으며, low-end 혹은 deep-내장형 시스템에서는 적당치 않은 솔루션이다. 그 기능을 보면,

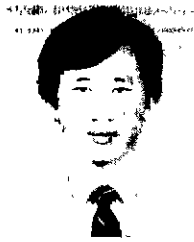
- ◆ Disk 기반의 파일 시스템
- ◆ UNIX/POSIX 및 WindowsNT API 지원
- ◆ Virtual address 공간 POSIX-style 프로세서 모델
- ◆ Berkeley Socket 네트워킹
- ◆ UNIX-style security
- ◆ Standard I/O를 통한 conventional script와 programmed CGI



김인홍

1983년 경북대학교 전산공학과 (학사)
1985년 한국과학기술원 전산학과 (석사)
1984년-1989년 한국전자통신연구원 연구원

1989년-1990년 삼성종합기술원 선임연구원
1990년-1995년 Maspar Computer Corp(U.S) Senior Engineer
1995년-1997년 Jinnes Technologies Inc.(U.S.) 이사
1997년-현재 하이콤 리서치 대표



이정배

1981년 경북대학교 전자공학과 전산전공 (학사)
1983년 경북대학교 전자공학과 전산전공 (석사)
1995년 한양대학교 전자공학과 (박사)

1982년-1991년 한국전자통신연구원 선임연구원
1996년-1997년 U.C.Irvine 객원교수
1991년-현재 부산외국어대학교 부교수
관심분야 : 컴퓨터 네트워크, 실시간 자바, 실시간 통신 프로토콜, 인터넷 응용

'98 추계 학술발표대회 논문모집

- ◎ 일 시 : 1998년 10월 16일(금) ~ 17일(토)
- ◎ 장 소 : 경희대학교(수원)
- ◎ 내 용 : 초청강연, 튜토리얼, 논문발표, 정기총회
- ◎ 문의전화 : (02)593-2894 팩스 (02)593-2896
- * 자세한 내용은 발송한 팜플렛 참조

논문마감 : 9월 4일 (금)까지