Web Servers

پروتکل HTTP

وب سرورها روزانه میلیاردها صفحه وب را پخش می‌کنند. آن‌ها وضعیت آب و هوا را به شما می‌گویند، به وسیله آن‌ها می‌توانید خریدهای خود را به صورت آنلاین انجام دهید و به شما اجازه می‌دهند دوستان دبیرستانی را که مدت‌هاست گم کرده‌اید پیدا کنید. وب سرورها، اسباب کار شبکه جهانی وب هستند. در این فصل ما موارد زیر را پوشش خواهیم داد:

- بررسی انواع مختلف سرورهای وب نرم افزاری و سخت افزاری
- تشریح نحوه نوشتن یک برنامه ساده برای تشخیص وب سرور در پرل
- توضیح گام به گام نحوه پردازش تراکنش‌های HTTP توسط سرورهای وب

Web Servers Come in All Shapes and Sizes

یک وب سرور درخواست‌های HTTP را پردازش می‌کند و پاسخ‌های متناسب را ارائه می‌دهد. اصطلاح «وب سرور» می‌تواند به نرم‌افزار وب سرور یا دستگاه یا رایانه خاصی که برای سرویس دادن به صفحات وب اختصاص داده شده است اشاره کند.

وب سرورها در انواع، شکل‌ها و اندازه‌های مختلفی عرضه می‌شوند. سرورهای وب ساده ۱۰ خطی اسکریپت Perl، موتورهای تجاری ایمن ۵۰ مگابایتی و سرورهای کوچک روی کارت (servers-on-a-card) وجود دارند. اما تفاوت‌های عملکردی هرچه که باشد، تمامی سرورهای وب، درخواست‌های HTTP را دریافت نموده و محتوا را به کلاینت‌ها ارائه می‌دهند.

Web Server Implementations

سرورهای وب، HTTP و مدیریت ارتباط TCP مربوطه را پیاده سازی می‌کنند. آن‌ها همچنین منابع ارائه شده توسط وب سرور را مدیریت می‌کنند و ویژگی‌های مدیریتی را برای پیکربندی، کنترل و بهبود وب سرور ارائه می‌دهند.

منطق وب سرور، پروتکل HTTP را پیاده سازی می‌کند، منابع وب را مدیریت نموده و قابلیت‌های مدیریت وب سرور را فراهم می‌کند. منطق وب سرور، مسئولیت به اشتراکت گذاشتن مدیریت اتصالات TCP با سیستم عامل را به عهده دارد.

سیستم عامل موجود در وب سرور نیز جزئیات سخت افزاری سیستم کامپیوتری زیربنایی را مدیریت می‌کند و پشتیبانی از شبکه TCP/IP، سیستم‌های فایل برای نگهداری منابع وب و مدیریت فرآیند برای کنترل فعالیت‌های محاسباتی جاری را فراهم می‌نماید.

وب سرورها به اشکال مختلف در دسترس هستند:

- شما می‌توانید وب سرورهای نرم افزاری همه منظوره را بر روی سیستم‌های کامپیوتری استاندارد نصب و اجرا کنید.
- اگر زحمت نصب نرم‌افزار را نمی‌خواهید، می‌توانید یک ابزار وب سرور بخرید، که در آن نرم‌افزار از پیش نصب شده و از پیش پیکربندی شده روی یک کامپیوتر، اغلب در یک شاسی زیبا قرار دارد.
- با توجه به معجزات ریزپردازنده‌ها، برخی از شرکت‌ها حتی وب سرورهای تعبیه شده (Embedded) را ارائه می‌دهند که در تعداد کمی از تراشه‌های کامپیوتری پیاده سازی شده اند و آن‌ها را به کنسول‌های مدیریتی عالی برای دستگاه‌های مصرف کننده تبدیل می‌کند.

General-Purpose Software Web Servers

وب سرورهای نرم افزاری همه منظوره بر روی سیستم‌های کامپیوتری استاندارد و دارای شبکه فعال می‌شوند. می‌توانید نرم‌افزار منبع باز (مانند Apache یا W3C's Jigsaw) یا نرم‌افزار تجاری (مانند وب سرورهای مایکروسافت و iPlanet) را انتخاب کنید. نرم افزار وب سرور تقریباً برای هر رایانه و سیستم عامل موجود است.

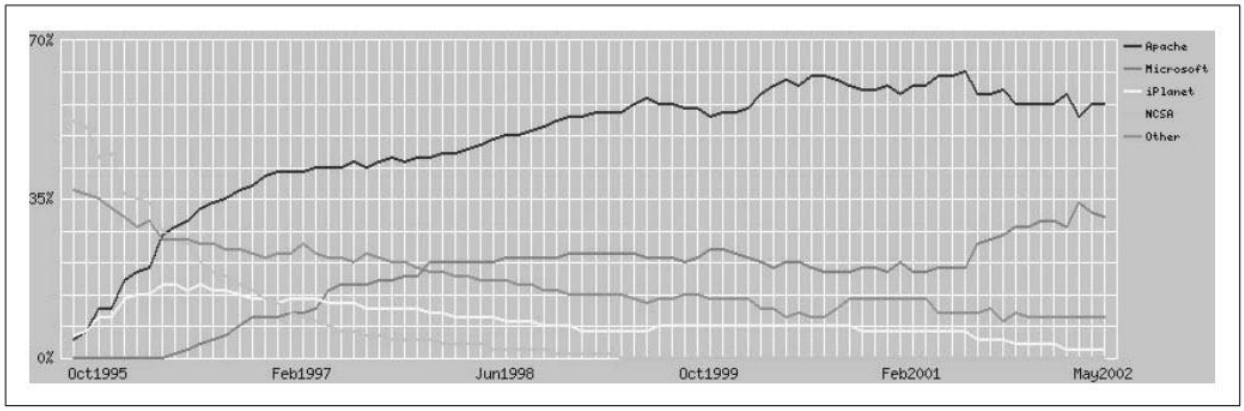
در حالی که ده‌ها هزار نوع مختلف از برنامه‌های وب سرور وجود دارد (از جمله وب سرورهای سفارشی و با هدف خاص)، اکثر نرم افزارهای وب سرور از تعداد کمی از سازمان‌ها تهیه می‌شوند.

در فوریه ۲۰۰۲، نظرسنجی نت کرافت (http://www.netcraft.com/survey/) نشان داد که سه فروشنده بر بازار وب سرورهای اینترنتی عمومی تسلط دارند:

نرم افزار رایگان آپاچی تقریبا ۶۰ درصد از تمام وب سرورهای اینترنتی را تامین می‌کند.

۳۰ درصد مابقی در اختیار وب سرورهای مایکروسافت است.

۳ درصد از این بازار هم در اختیار Sun iPlanet است.



با توجه به اینکه نت کرافت معمولاً در مورد تسلط نرم افزار آپاچی اغراق آمیز صحبت می‌کند، آمار مربوط به آپاچی را کمی اغراق در نظر بگیرید.

مطالعات بر روی سرورهای پروکسی ازISP های بزرگ نشان می‌دهد که تعداد صفحات ارائه شده از سرورهای Apache بسیار کمتر از ۶۰ درصد است اما همچنان از مایکروسافت و Sun iPlanet فراتر می‌رود.

علاوه بر این، سرورهای مایکروسافت و iPlanet در داخل شرکت‌های بزرگ از Apache محبوب‌تر هستند.

Web Server Appliances

Web Server Appliances راه حل‌های نرم افزاری/سخت افزاری از پیش بسته بندی شده هستند. فروشنده یک سرور نرم افزاری را روی یک پلت فرم کامپیوتری انتخاب شده توسط فروشنده نصب می‌کند و نرم افزار را پیکربندی می‌کند. چند نمونه از Web Server Appliances عبارتند از:

- Sun/Cobalt RaQ web appliances (http://www.cobalt.com)
- Toshiba Magnia SG10 (http://www.toshiba.com)
- IBM Whistle web server appliance (http://www.whistle.com)

راه‌حل‌های Appliance نیاز به نصب و پیکربندی نرم‌افزار را برطرف نموده و اغلب مدیریت را تا حد زیادی ساده می‌کنند. با این حال، وب سرور اغلب انعطاف‌پذیری و ویژگی‌های غنی‌تری دارد و سخت‌افزار سرور به راحتی قابل استفاده مجدد یا ارتقاء نیست.

Embedded Web Servers

سرورهای Embedded، وب سرورهای کوچکی هستند که برای تعبیه در محصولات مصرفی (به عنوان مثال، چاپگرها یا لوازم خانگی) در نظر گرفته شده اند. وب سرورهای Embedded به کاربران این

امکان را می‌دهد که دستگاه‌های مصرف کننده خود را با استفاده از یک رابط مرورگر وب مناسب مدیریت کنند.

برخی از وب سرورهای Embedded را می‌توان حتی در کمتر از یک اینچ مربع پیاده سازی کرد، اما آن‌ها معمولاً یک مجموعه ویژگی حداقلی را ارائه می‌دهند. دو نمونه از وب سرورهای Embedded بسیار کوچک عبارتند از:

- IPic match-head sized web server (http://www-ccs.cs.umass.edu/~shri/iPic.html)
- NetMedia SitePlayer SP1 Ethernet Web Server (http://www.siteplayer.com)

A Minimal Perl Web Server

اگر می‌خواهید یک سرور HTTP با امکانات کامل بسازید، باید کارهای زیادی انجام دهید. هسته وب سرور آپاچی بیش از ۵۰۰۰۰ خط کد دارد و ماژول‌های پردازش اختیاری، این تعداد را بسیار بزرگتر می‌کنند.

همه این نرم افزار برای پشتیبانی از ویژگی‌های HTTP/1.1 مورد نیاز است. ویژگی‌هایی از قبیل: پشتیبانی از منابع غنی، میزبانی مجازی(Virtual Hosting)، کنترل دسترسی، ورود به سیستم، پیکربندی، نظارت و ویژگی‌های عملکرد.

همچنین شما می‌توانید یک سرور HTTP با حداقل عملکرد را در کمتر از ۳۰ خط پرل ایجاد کنید. بیا نگاهی به آن بیندازیم.

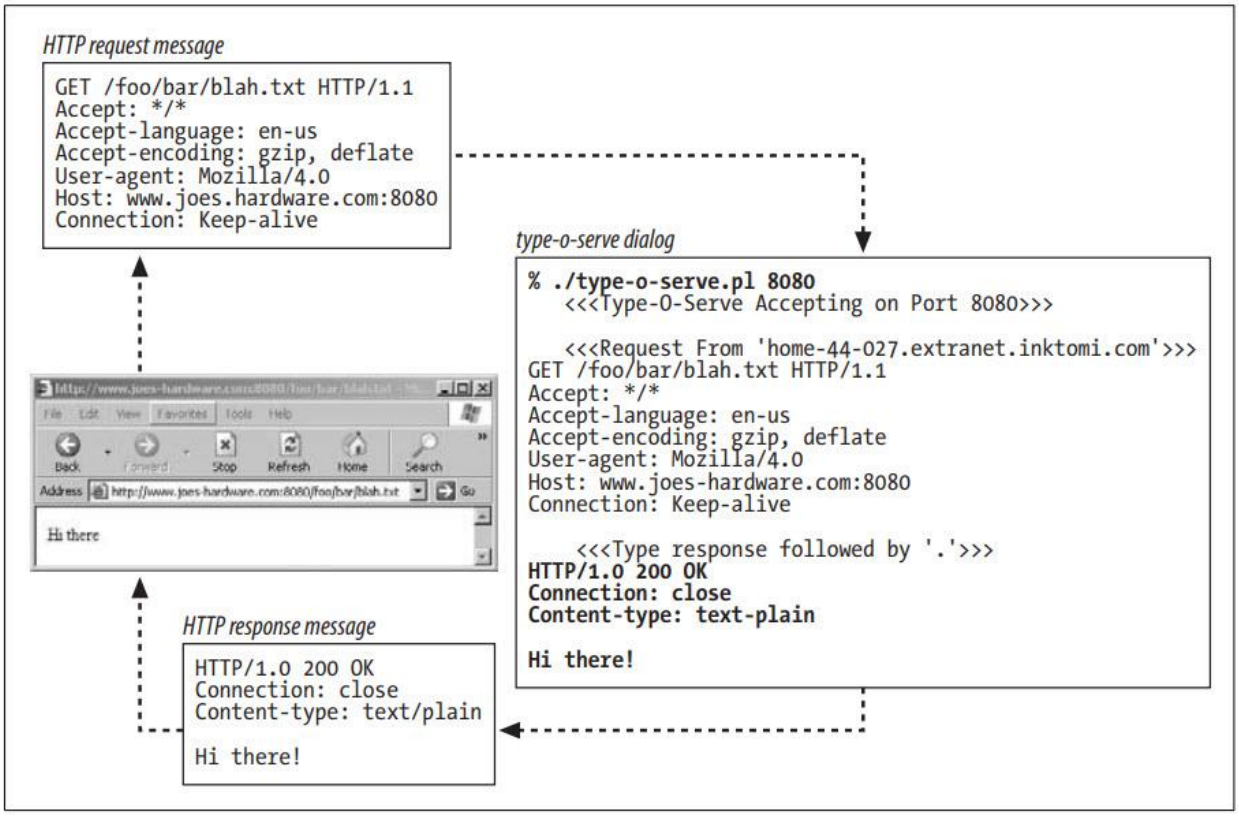
تصویر زیر، یک برنامه Perl کوچک به نام type-o-serve را نشان می‌دهد. این برنامه یک ابزار تشخیصی مفید برای تست تعامل با کلاینت و پروکسی‌ها است. مانند هر وب سرور، type-o-serve منتظر اتصال HTTP است. به محض اینکه type-o-serve پیام درخواست را دریافت می‌کند، پیام را روی صفحه چاپ می‌کند. سپس منتظر می‌ماند تا یک پیام پاسخ را تایپ کنید (یا Paste کنید) که برای کلاینت ارسال می‌شود. به این ترتیب، type-o-serve وانمود می‌کند که یک وب سرور است، پیام‌های درخواست HTTP دقیق را ضبط می‌کند و به شما امکان می‌دهد هر پیام پاسخ HTTP را ارسال کنید.

این ابزار ساده type-o-serve اکثر قابلیت‌های HTTP را پیاده‌سازی نمی‌کند، اما ابزار مفیدی برای تولید پیام‌های پاسخ سرور است. می‌توانید برنامه type-o-serve را از لینک زیر دانلود نمایید.

http://www.http-guide.com/tools/type-o-serve.pl

```
#!/usr/bin/perl

use Socket;
use Carp;
use FileHandle;

# (1) use port 8080 by default, unless overridden on command line
$port = (@ARGV ? $ARGV[0] : 8080);

# (2) create local TCP socket and set it to listen for connections
$proto = getprotobyname('tcp');
socket(S, PF_INET, SOCK_STREAM, $proto) || die;
setsockopt(S, SOL_SOCKET, SO_REUSEADDR, pack("l", 1)) || die;
bind(S, sockaddr_in($port, INADDR_ANY)) || die;
listen(S, SOMAXCONN) || die;

# (3) print a startup message
printf("    <<<Type-O-Serve Accepting on Port %d>>>\n\n",$port);

while (1)
{
    # (4) wait for a connection C
    $cport_caddr = accept(C, S);
    ($cport,$caddr) = sockaddr_in($cport_caddr);
    C->autoflush(1);

    # (5) print who the connection is from
    $cname = gethostbyaddr($caddr,AF_INET);
    printf("    <<<Request From '%s'>>>\n",$cname);

    # (6) read request msg until blank line, and print on screen
    while ($line = <C>)
    {
        print $line;
        if ($line =~ /^\r/) { last; }
    }

   # (7) prompt for response message, and input response lines,
   #     sending response lines to client, until solitary "."
   printf("    <<<Type Response Followed by '.'>>>\n");

   while ($line = <STDIN>)
   {
       $line =~ s/\r//;
       $line =~ s/\n//;
       if ($line =~ /^\./) { last; }
       print C $line . "\r\n";
   }
   close(C);
}
```

شکل زیر نشان می‌دهد که چگونه ممکن است مدیر فروشگاه سخت افزار Joe از type-oserve برای آزمایش ارتباط HTTP استفاده کند:



ابتدا، مدیر سرور، type-o-serve را با Listen نمودن یک پورت خاص راه اندازی می‌کند. از آنجایی که فروشگاه سخت افزار Joe در حال حاضر دارای یک وب سرور در محیط Production بوده که بر روی پورت ۸۰ Listen شده است، مدیر سرور، type-o-serve را در پورت ۸۰۸۰ (شما می‌توانید هر پورت بدون استفاده را انتخاب کنید) با دستور راه اندازی می‌کند:

```
% type-o-serve.pl 8080
```

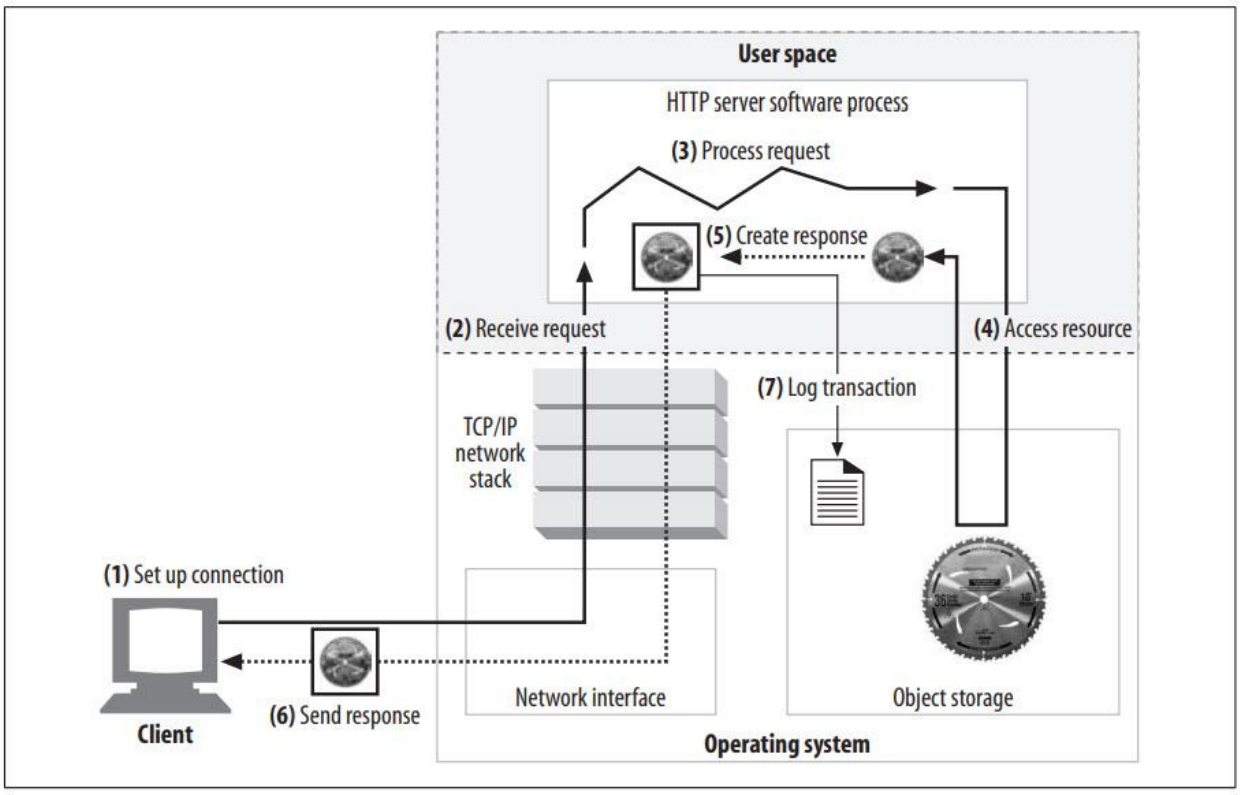
هنگامی که type-o-serve اجرا می‌شود، می‌توانید یک مرورگر را به این وب سرور هدایت کنید. در شکل بالا، ما به آدرس http://www.joes-hardware.com:8080/foo/bar/blah.txt مراجعه می‌کنیم.

برنامه type-o-serve پیام درخواست HTTP را از مرورگر دریافت می‌کند و محتوای پیام درخواست HTTP را روی صفحه چاپ می‌کند. سپس برنامه type-o-serve منتظر می‌ماند تا کاربر یک پیام پاسخ ساده را تایپ کند و سپس یک نقطه در یک خط خالی قرار دهد.

type-o-serve پیام پاسخ HTTP را به مرورگر می‌فرستد و مرورگر متن پیام پاسخ را نمایش می‌دهد.

What Real Web Servers Do

سرور پرل که در بخش قبلی نشان داده شد، یک وب سرور نمونه بی اهمیت است. وب سرورهای تجاری پیشرفته بسیار پیچیده‌تر هستند، اما آن‌ها چندین کار رایج را انجام می‌دهند:



Set up connection —یک اتصال کلاینت را می‌پذیرد، یا اگر کلاینت ناخواسته است، آن را می‌بندد.

Receive request - یک پیام درخواست HTTP از شبکه را می‌خواند.

Process request - پیام درخواست را تفسیر نموده و اقدام لازم را بر روی آن انجام می‌دهد.

Access resource – دسترسی به منبع مشخص شده در پیام

Construct response - پیام پاسخ HTTP را با هدرهای مناسب ایجاد می‌نماید.

Send response - پاسخ را برای مشتری ارسال می‌کند.

Log transaction – لاگ‌های مربوط به تراکنش انجام شده را در یک فایل گزارش قرار می‌دهد.

هفت بخش بعدی نشان می‌دهد که چگونه وب سرورها این وظایف اساسی را انجام می‌دهند.

Step 1: Accepting Client Connections

اگر یک کلاینت قبلاً یک اتصال دائمی به سرور داشته باشد، می‌تواند از آن اتصال برای ارسال درخواست خود استفاده کند. در غیر این صورت، مشتری باید یک اتصال جدید به سرور باز کند (برای بررسی فناوری مدیریت اتصال HTTP به فصل ۴ مراجعه کنید).

Handling New Connections

هنگامی که یک سرویس گیرنده درخواست اتصال TCP به وب سرور را ارسال می‌کند، وب سرور اتصال را برقرار می‌کند و تعیین می‌کند که کدام کلاینت در طرف دیگر اتصال است و آدرس IP را از اتصال TCP استخراج می‌کند. هنگامی که یک اتصال جدید برقرار و پذیرفته شد، سرور اتصال جدید را به لیست اتصالات وب سرور موجود خود اضافه می‌کند و برای مشاهده داده‌های موجود در اتصال آماده می‌شود.

وب سرور در رد کردن و بستن فوری هر گونه اتصال آزاد است. برخی از وب سرورها اتصالات را می‌بندند زیرا آدرس IP کلاینت یا نام میزبان غیرمجاز است یا یک سرویس گیرنده، مخرب شناخته شده است.

Client Hostname Identification

اکثر وب سرورها را می‌توان با استفاده از "Reverse DNS" برای تبدیل آدرس‌های IP کلاینت به نام میزبان کلاینت، پیکربندی کرد. وب سرورها می‌توانند از Hostname کلاینت برای کنترل دسترسی دقیق و ثبت گزارش استفاده کنند. البته هشدار داده می‌شود که جستجوی نام میزبان می‌تواند زمان بسیار زیادی طول بکشد و تراکنش‌های وب را کند کند. بسیاری از وب سرورهای با ظرفیت بالا یا Hostname Resolution را غیرفعال نموده و یا آن را فقط برای محتوای خاصی فعال می‌کنند.

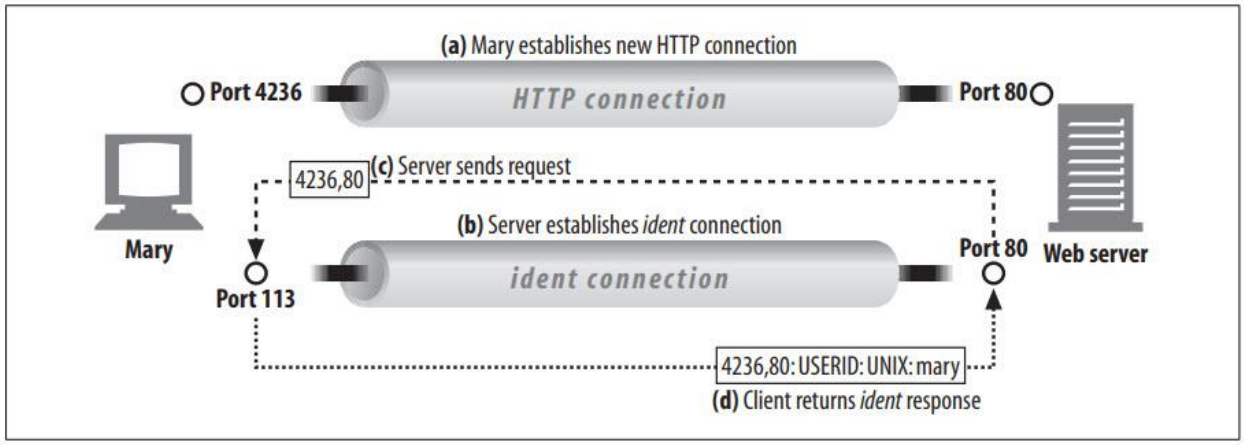
می‌توانید جستجوی Hostname را در آپاچی با دستور پیکربندی HostnameLookups فعال کنید. به عنوان مثال، دستورالعمل‌های پیکربندی آپاچی در مثال زیر، Hostname Resolution را فقط برای منابع HTML و CGI فعال می‌کند.

```
HostnameLookups off
<Files ~ "\.(html|htm|cgi)"$>
 HostnameLookups on
</Files>
```

Determining the Client User Through ident

برخی از سرورهای وب نیز از پروتکل IETF ident پشتیبانی می‌کنند. پروتکل ident به سرورها اجازه می‌دهد تا بفهمند که چه نام کاربری اتصال HTTP را آغاز کرده است. این اطلاعات به صورت خاص برای Logging وب سرور مفید است. فیلد دوم Log Format حاوی نام کاربری ident هر درخواست HTTP است.

اگر یک کلاینت از پروتکل ident پشتیبانی کند، کلاینت به درگاه TCP 113 برای درخواست‌های ident گوش می‌دهد. شکل زیر نحوه عملکرد پروتکل ident را نمایش می‌دهد.



در بخش a از شکل بالا، کلاینت یک اتصال HTTP را باز می‌کند. سپس سرور اتصال خود را به پورت سرور ident کلاینت (۱۱۳) برقرار می‌کند، یک درخواست ساده می‌فرستد و نام کاربری مربوط به اتصال جدید را می‌پرسد و پاسخ حاوی نام کاربری را از کلاینت دریافت می‌کند.

ident می‌تواند در داخل سازمان‌ها کار کند، اما به دلایل زیادی در سراسر اینترنت عمومی به خوبی کار نمی‌کند، از جمله:

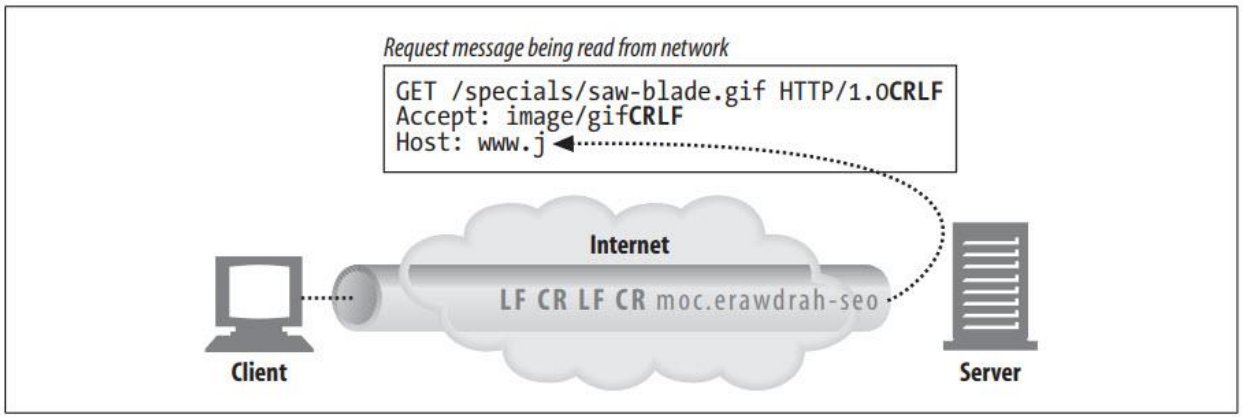
- بسیاری از رایانه‌های شخصی مشتری، نرم افزار دیمون identd Identification Protocol را اجرا نمی‌کنند.
- پروتکل ident به طور قابل توجهی تراکنش‌های HTTP را به تاخیر می‌اندازد.
- بسیاری از فایروال‌ها اجازه ترافیک ident ورودی را نمی‌دهند.
- پروتکل ident ناامن است و ساخت آن آسان است.
- پروتکل ident آدرس‌های Virtual IP را به خوبی پشتیبانی نمی‌کند.
- نگرانی‌های مربوط به حریم خصوصی در مورد افشای نام‌های کاربری مشتری وجود دارد.

می‌توانید به وب سرورهای آپاچی بگویید که از جستجوی شناسایی با Apache's IdentityCheck در دستورالعمل استفاده کنند. اگر هیچ اطلاعات شناسایی در دسترس نباشد، آپاچی فیلدهای لاگ

ident را با خط تیره (-) پر می‌کند. فایل‌های گزارش با Log Format رایج معمولاً حاوی خط فاصله در فیلد دوم هستند زیرا هیچ اطلاعات ident ای در دسترس نیست.

Step 2: Receiving Request Messages

همانطور که داده‌ها به Connection ها می‌رسد، وب سرور داده‌ها را از Connection شبکه می‌خواند و قطعات پیام درخواست را تجزیه می‌کند.



هنگام تجزیه پیام درخواست، وب سرور:

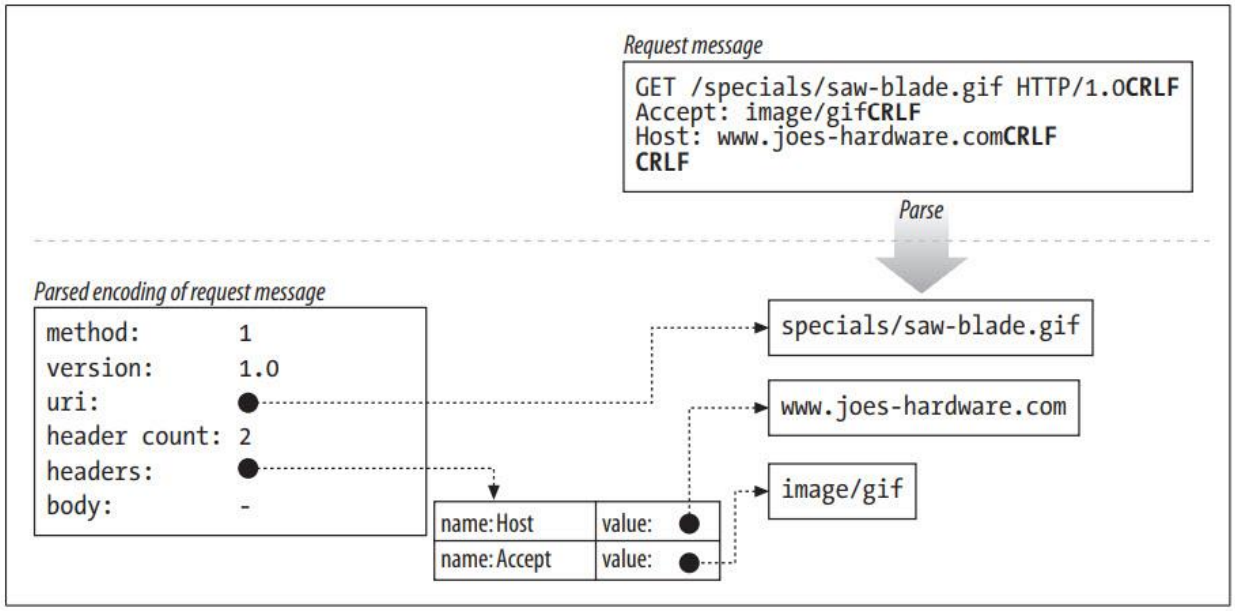
- در Request Line به دنبال متد درخواست، شناسه منبع مشخص شده (URI) و Version Number می‌باشد، که هر کدام با یک فاصله از هم جدا می‌شوند و به یک carriage-return line-feed یا CRLF ختم می‌شوند.
- هدرهای پیام را می‌خواند (هر کدام که به CRLF ختم می‌شوند)
- خط خالی انتهای هدر را که به CRLF ختم می‌شود (در صورت وجود) تشخیص می‌دهد.
- متن درخواست (Request Body) را در صورت وجود می‌خواند (طول مشخص شده توسط هدر Content-Length)

هنگام تجزیه پیام‌های درخواست، وب سرورها داده‌های ورودی را به طور نامنظم از شبکه دریافت می‌کنند. Connection شبکه می‌تواند در هر نقطه‌ای متوقف شود. وب سرور باید داده‌ها را از شبکه بخواند و به طور موقت داده‌های جزئی پیام را در حافظه ذخیره کند تا زمانی که داده‌های کافی برای تجزیه و درک آن را دریافت شود.

Internal Representations of Messages

برخی از وب سرورها نیز پیام‌های درخواست را در ساختارهای داده داخلی ذخیره می‌کنند که دستکاری پیام را آسان می‌کند. به عنوان مثال، ساختار داده ممکن است شامل نشانگرها و طول هر قطعه از پیام

درخواست باشد و هدرها ممکن است در یک جدول جستجوی سریع ذخیره شوند تا مقادیر خاص هدرهای خاص به سرعت قابل دسترسی باشند.



Connection Input/Output Processing Architectures

وب سرورهای با کارایی بالا از هزاران اتصال همزمان پشتیبانی می‌کنند. این اتصالات به وب سرور اجازه می‌دهد تا با کلاینت‌ها در سراسر جهان ارتباط برقرار کند که هر کدام دارای یک یا چند اتصال به سرور هستند. برخی از این Connection ها ممکن است درخواست‌ها را به سرعت به وب سرور ارسال کنند، در حالی که سایر اتصالات درخواست‌ها را به آرامی یا به ندرت چک می‌کنند و برخی دیگر بیکار بوده و بی سر و صدا منتظر برخی از فعالیت‌های آینده هستند.

وب سرورها دائماً به دنبال درخواست‌های وب جدید هستند، زیرا درخواست‌ها می‌توانند در هر زمانی وارد شوند.

همانطور که در شکل زیر نشان داده شده است:

وب سرورهای تک رشته ای یا Single-threaded (بخش a شکل)

وب سرورهای تک رشته ای هر بار یک درخواست را تا زمان تکمیل پردازش می‌کنند. هنگامی که تراکنش کامل شد، اتصال بعدی پردازش می‌شود. پیاده سازی این معماری ساده است، اما در حین پردازش، همه Connection های دیگر نادیده گرفته می‌شوند. این روش مشکلات عملکرد جدی ایجاد می‌کند و فقط برای سرورهای کم بار و ابزارهای تشخیصی مانند type-o-serve مناسب است.

وب سرورهای چند فرآیندی(Multiprocess) و چند رشته ای (Multithreaded) (بخش b شکل)

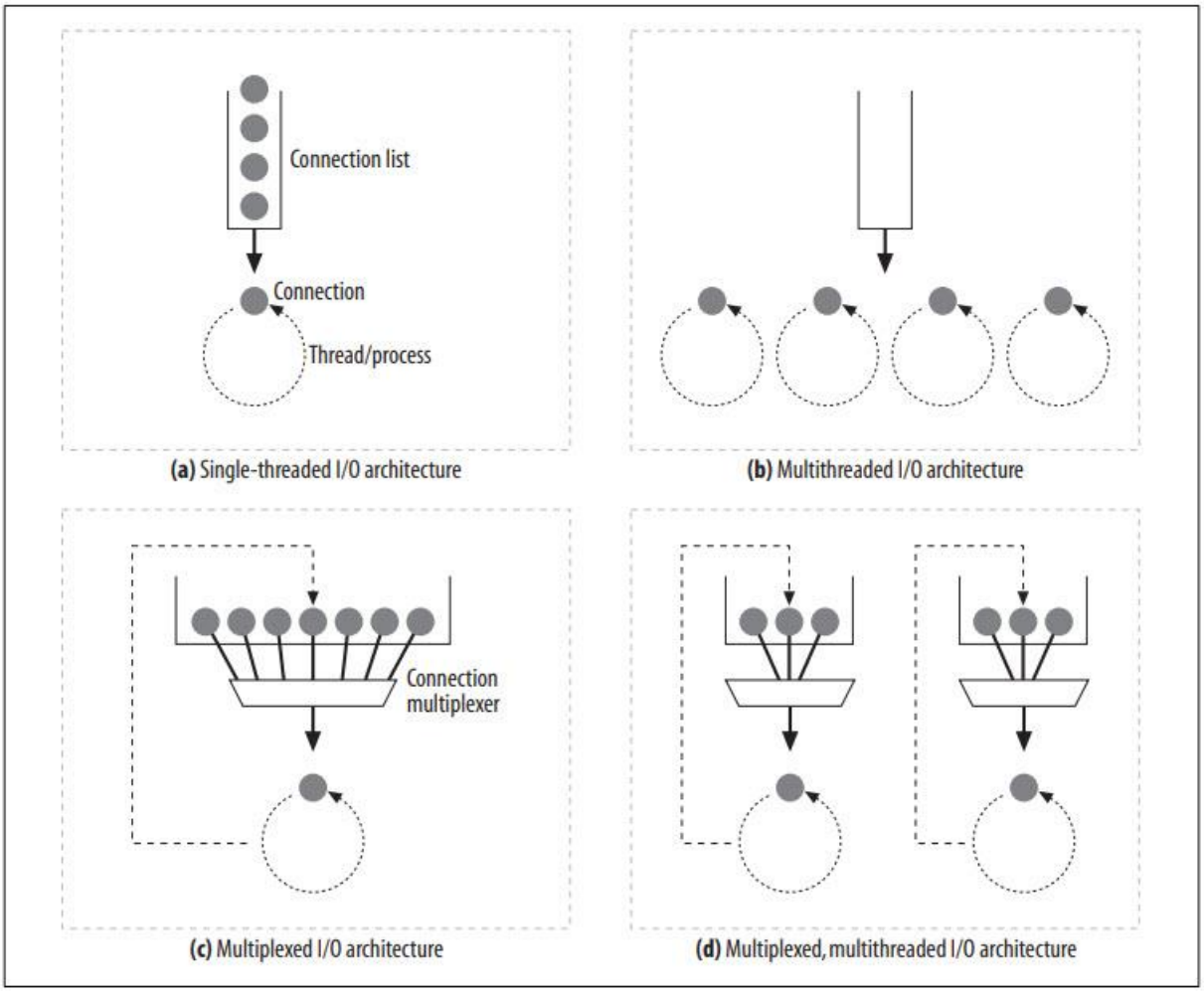
وب سرورهای چند فرآیندی و چند رشته ای چندین فرآیند یا رشته های با کارایی بالاتر را به پردازش درخواست‌ها به طور همزمان اختصاص می‌دهند. برخی از سرورها برای هر اتصال، یک thread/process اختصاص می‌دهند، اما زمانی که یک سرور صدها، هزاران یا حتی ده‌ها هزار اتصال همزمان را پردازش می‌کند، تعداد فرآیندها یا رشته‌های حاصل ممکن است حافظه یا منابع سیستم زیادی را مصرف کند. بنابراین، بسیاری از وب سرورهای چند رشته ای محدودیتی برای حداکثر تعداد موضوعات/فرآیندها قائل می‌شوند.

سرورهای ورودی/خروجی چندگانه یا Multiplexed I/O (بخشc شکل)

برای پشتیبانی از تعداد زیادی اتصال، بسیاری از سرورهای وب از معماری‌های چندگانه استفاده می‌کنند. در یک معماری مالتی پلکس، همه Connection ها به طور همزمان برای فعالیت، مشاهده می‌شوند. هنگامی که یک Connection، وضعیت را تغییر می‌دهد (به عنوان مثال، هنگامی که داده‌ها قابل دسترس می‌شوند یا یک خطا رخ می‌دهد)، مقدار کمی پردازش روی Connection انجام می‌شود. هنگامی که پردازش کامل شد، Connection برای تغییر وضعیت بعدی به لیست Open Connection بازگردانده می‌شود. کار روی یک Connection تنها زمانی انجام می‌شود که کاری برای انجام دادن وجود داشته باشد. موضوعات و فرآیندها در انتظار Connection های بیکار نیستند.

وب سرورهای چند رشته ای چند رشته ای یا Multiplexed Multithreaded (بخش d شکل)

برخی از سیستم‌ها، چند رشته ای و مالتی پلکسی را برای استفاده از چندین CPU در پلتفرم رایانه ترکیب می‌کنند. رشته‌های متعدد (اغلب یک در هر پردازنده فیزیکی) هر کدام Connection های باز (یا زیر مجموعه‌ای از اتصالات باز) را مشاهده می‌کنند و مقدار کمی کار روی هر اتصال انجام می‌دهند.



Step 3: Processing Requests

هنگامی که وب سرور درخواستی را دریافت کرد، می‌تواند درخواست را با استفاده از متد، منبع، هدرها و بدنه اختیاری پردازش کند.

برخی از متدها (به عنوان مثال، POST) به داده‌های Entity Body در پیام درخواست نیاز دارند. متدهای دیگر (مثلاً OPTIONS) به بدنه درخواست اجازه می‌دهند اما به آن نیاز ندارند. چند متد دیگر (به عنوان مثال، GET) داده‌های Entity Body را در پیام‌های درخواست ممنوع می‌کند.

Step 4: Mapping and Accessing Resources

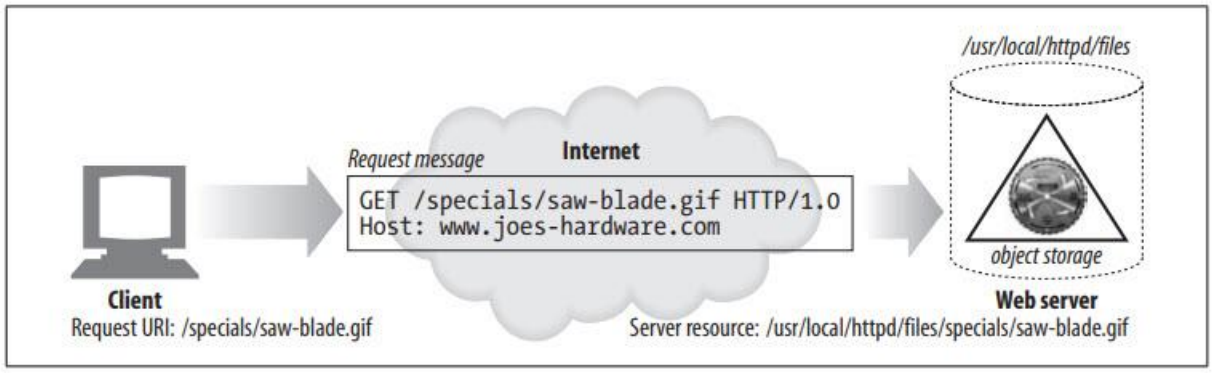
وب سرورها سرورهای منبع یا Resource Servers هستند. آن‌ها محتوای از پیش ساخته شده، مانند صفحات HTML یا تصاویر JPEG، و همچنین محتوای پویا را از برنامه‌های کاربردی تولید کننده منابع در حال اجرا بر روی سرورها ارائه می‌دهند.

قبل از اینکه وب سرور بتواند محتوا را به کلاینت تحویل دهد، باید منبع محتوا را با Map نمودن URI از پیام درخواست به محتوا یا تولید کننده محتوای مناسب در سرور وب شناسایی کند.

Docroots

وب سرورها انواع مختلفی از نگاشت منابع یا Resource Mapping را پشتیبانی می‌کنند، اما ساده ترین شکل نگاشت منابع، از URI درخواست برای نامگذاری فایل در سیستم فایل وب سرور استفاده می‌کند. به طور معمول، یک پوشه خاص در سیستم فایل وب سرور برای محتوای وب رزرو شده است. این پوشه ریشه سند یا docroot نام دارد. وب سرور URI را از پیام درخواست می‌گیرد و به ریشه سند اضافه می‌کند.

در شکل زیر، درخواستی برای /specials/saw-blade.gif می‌رسد. وب سرور در این مثال دارای ریشه سند /usr/local/httpd/files است. بدین ترتیب وب سرور فایل /usr/ local/httpd/files/specials/saw-blade.gif را برمی‌گرداند.



برای تنظیم ریشه سند برای وب سرور آپاچی، یک خط DocumentRoot را به فایل پیکربندی httpd.conf اضافه کنید:

```
DocumentRoot /usr/local/httpd/files
```

سرورها مراقب هستند که اجازه ندهند URL های نسبی از یک docroot پشتیبان تهیه کنند و سایر قسمت‌های سیستم فایل را در معرض دید قرار دهند. برای مثال، اکثر وب سرورهای بالغ به این URI اجازه نمی‌دهند فایل‌های بالای ریشه سند سخت‌افزار Joe را ببینند:

```
http://www.joes-hardware.com/../
```

Virtually hosted docroots

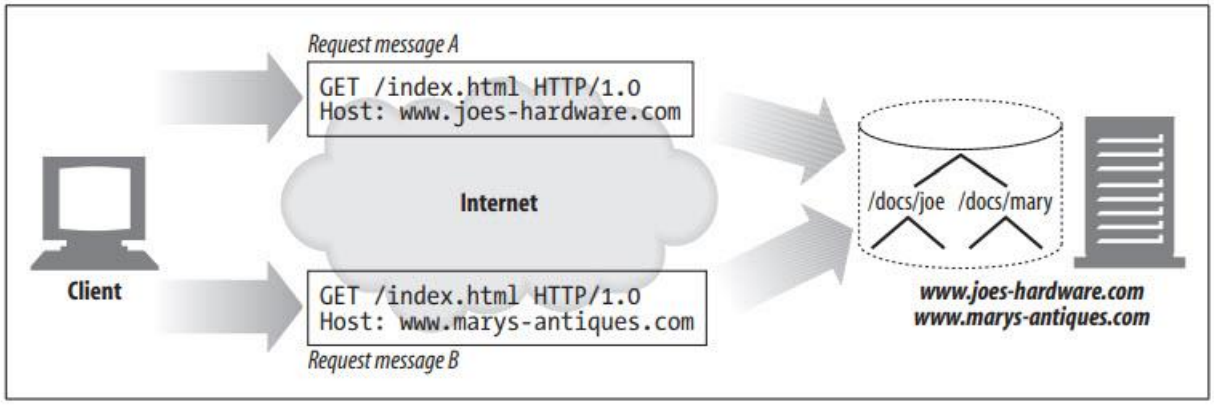
وب سرورهایی که به صورت مجازی میزبانی می‌شوند، چندین وب سایت را در یک وب سرور میزبانی

می‌کنند که هر سایت ریشه سند مجزای خود را در سرور دارد. یک وب سرور مجازی (Virtually Hosted Web Server)، ریشه سند صحیح را برای استفاده از آدرس IP یا نام میزبان در URI یا هدر Host شناسایی می‌کند. به این ترتیب، دو وب‌سایت میزبانی شده روی یک وب سرور می‌توانند محتوای کاملاً متمایز داشته باشند، حتی اگر URIهای درخواست یکسان باشند.

در شکل زیر، سرور میزبان دو سایت است:

www.joes-hardware.com

www.marysantiques.com



سرور می‌تواند وب سایت‌ها را با استفاده از هدر Host یا از آدرس‌های IP متمایز کند.

- هنگامی که درخواست A می‌رسد، سرور فایل را برای /docs/joe/index.html، Fetch می‌کند.
- هنگامی که درخواست B می‌رسد، سرور فایل را برای /docs/mary/index.html، Fetch می‌کند.

پیکربندی docroots به صورت مجازی برای اکثر سرورهای وب، ساده است. برای وب سرور محبوب Apache، باید یک بلوک VirtualHost برای هر وب سایت مجازی پیکربندی کنید و DocumentRoot را برای هر سرور مجازی اضافه کنید.

```
<VirtualHost www.joes-hardware.com>
 ServerName www.joes-hardware.com
 DocumentRoot /docs/joe
        TransferLog /logs/joe.access_log
        ErrorLog /logs/joe.error_log
```

<VirtualHost>

<VirtualHost www.marys-antiques.com>

ServerName www.marys-antiques.com

DocumentRoot /docs/mary

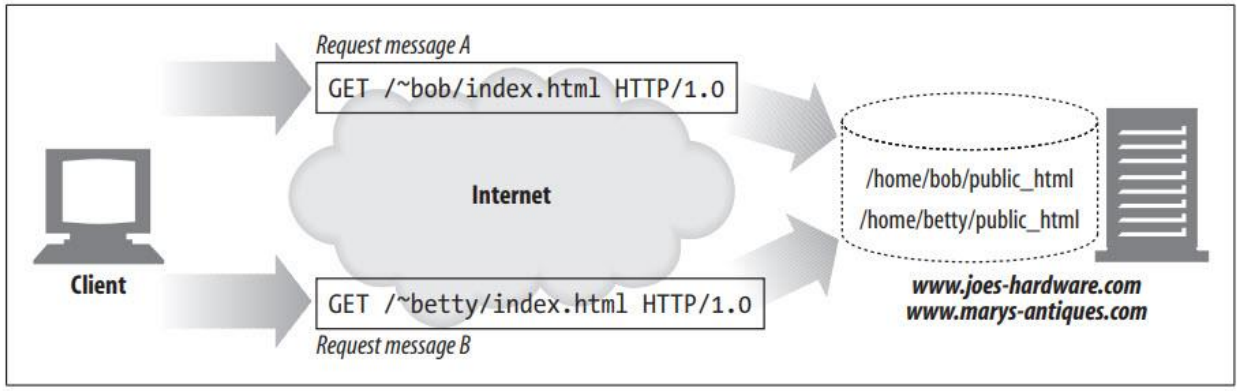
TransferLog /logs/mary.access_log

ErrorLog /logs/mary.error_log

</VirtualHost>

User home directory docroots

یکی دیگر از استفاده‌های رایج از docroots، به افراد وب سایت‌های خصوصی در یک وب سرور می‌دهد. یک قرارداد معمولی URIهایی را که مسیرهای آن‌ها با یک اسلش و tilde (~/) شروع می‌شود، به همراه یک نام کاربری به یک ریشه سند خصوصی برای آن کاربر Map می‌کند. Docroot خصوصی اغلب پوشه ای به نام public_html در دایرکتوری اصلی آن کاربر است، اما می‌توان آن را به شکلی متفاوت پیکربندی کرد.



Directory Listings

یک وب سرور می‌تواند در جایی که مسیر به یک دایرکتوری Resolve می‌شود، درخواست‌هایی را برای URL های مربوط به دایرکتوری دریافت کند. اکثر وب سرورها را می‌توان به گونه ای پیکربندی کرد که هنگام درخواست یک سرویس گیرنده URL دایرکتوری، اقدامات مختلفی را انجام دهند:

- یک پیام خطا بازگرداند.
- یک خطای خاص، پیش فرض و Index File به جای دایرکتوری بازگرداند.
- دایرکتوری را اسکن کند و یک صفحه HTML حاوی محتویات را برگردانید.

اکثر وب سرورها به دنبال فایلی به نام index.html یا index.htm در داخل یک دایرکتوری برای نمایش آن دایرکتوری می‌گردند. در صورتی که کاربر یک URL را برای یک دایرکتوری درخواست نماید و دایرکتوری شامل یک فایل با نام index.html (یا index.htm) باشد، سرور محتوای آن را باز می گرداند.

در وب سرور آپاچی، می‌توانید مجموعه‌ای از نام فایل‌ها را که به عنوان فایل‌های فهرست پیش‌فرض تفسیر می‌شوند، با استفاده از دستورالعمل پیکربندی DirectoryIndex پیکربندی کنید. دستورالعمل DirectoryIndex همه نام‌های فایلی را که به عنوان فایل فهرست، ارائه می‌شوند، به ترتیب ترجیحی فهرست می‌کند. خط پیکربندی زیر باعث می‌شود Apache در پاسخ به درخواست URL، فایل‌های فهرست شده را جستجو نموده و محتوای آن را بازگرداند:

```
DirectoryIndex index.html index.htm home.html home.htm index.cgi
```

اگر زمانی که کاربر URI دایرکتوری را درخواست می‌کند، هیچ فایل فهرست پیش‌فرضی وجود نداشته باشد و اگر فهرست‌های دایرکتوری غیرفعال نباشند، بسیاری از سرورهای وب به طور خودکار یک فایل HTML فهرست فایل‌های موجود که در آن نام فایل‌ها، اندازه آن‌ها و تاریخ‌های اصلاح هر فایل را برمی‌گردانند. این موضوع اما به افراد سودجو اجازه می‌دهد تا فایل‌هایی را که در یک سرور وب قرار دارند، شناسایی نماید که در حالت عادی معمولاً شناسایی آن‌ها امکان پذیر نیست.

شما می‌توانید تولید خودکار فایل‌های فهرست دایرکتوری را با دستور Apache زیر غیرفعال کنید:

```
Options –Indexes
```

Dynamic Content Resource Mapping

سرورهای وب همچنین می‌توانند URI ها را به منابع پویا Map کنند (یعنی به برنامه‌هایی که بر اساس تقاضا محتوا تولید می‌کنند). در واقع، یک کلاس کامل از وب سرورها به نام Application Server، سرورهای وب را به برنامه‌های کاربردی پیشرفته متصل می‌کنند. وب سرور باید بتواند تشخیص دهد که چه زمانی یک منبع، یک منبع پویا است، برنامه تولید کننده محتوای پویا در کجا قرار دارد و چگونه برنامه را اجرا کند. اکثر وب سرورها مکانیسم‌های اساسی برای شناسایی و Mapping از منابع پویا ارائه می‌دهند.

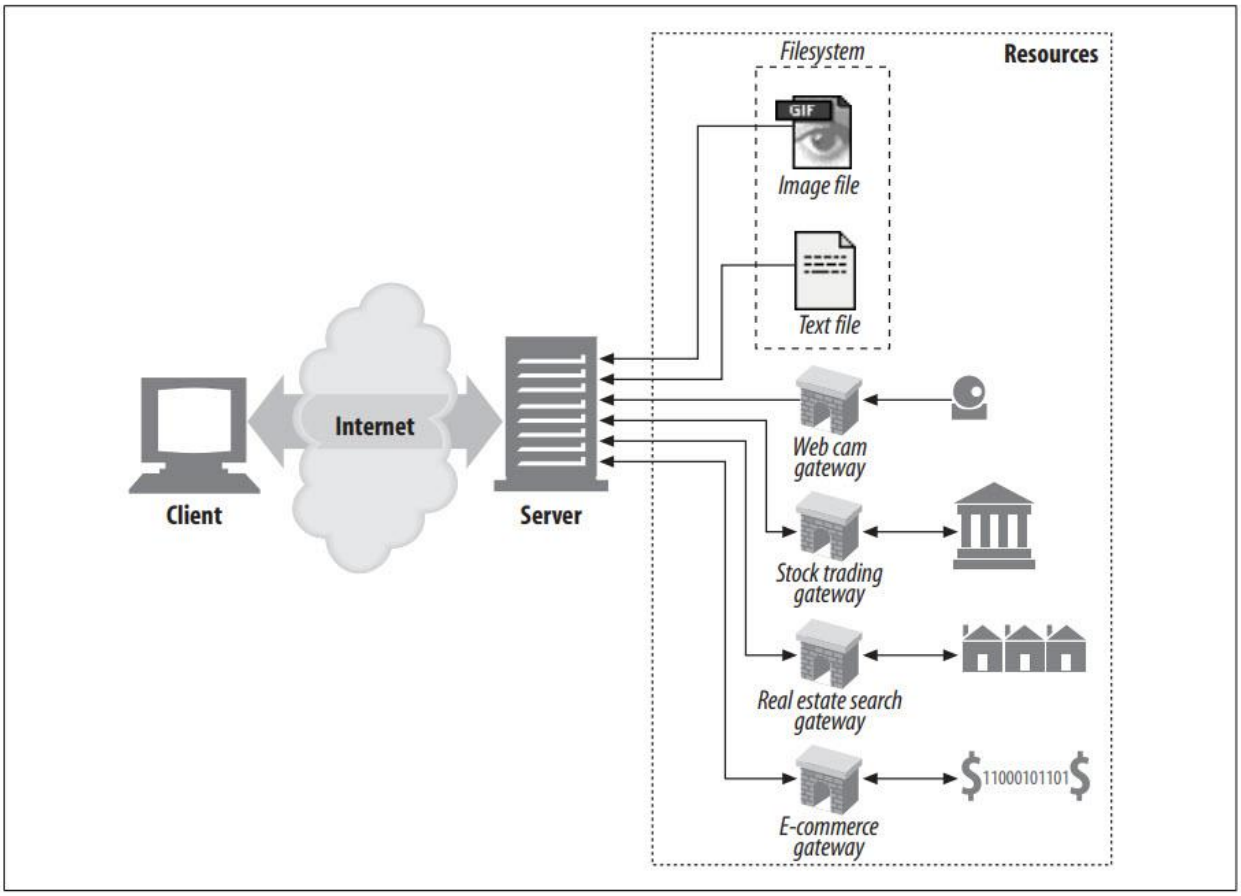
آپاچی به شما این امکان را می‌دهد که اجزای مسیر URI را در دایرکتوری‌های برنامه اجرایی Map کنید. هنگامی که یک سرور درخواستی برای یک URI با یک جزء مسیر اجرایی دریافت می‌کند، سعی می‌کند یک برنامه را در یک فهرست سرور مربوطه اجرا کند. برای مثال، دستورالعمل پیکربندی آپاچی زیر مشخص می‌کند که همه URI هایی که مسیرشان با /cgi-bin/ شروع می‌شود، باید برنامه‌های مربوطه موجود در دایرکتوری /usr/local/etc/httpd/cgi-programs/ را اجرا کنند:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-programs/
```

آپاچی همچنین به شما امکان می‌دهد فایل‌های اجرایی را با پسوند فایل خاص علامت گذاری کنید. به این ترتیب، اسکریپت‌های اجرایی را می‌توان در هر دایرکتوری قرار داد. دستورالعمل پیکربندی آپاچی زیر مشخص می‌کند که تمام منابع وب که به .cgi ختم می‌شوند باید اجرا شوند:

```
AddHandler cgi-script .cgi
```

CGI یک رابط اولیه، ساده و محبوب برای اجرای برنامه‌های سمت سرور است. سرورهای کاربردی مدرن از محتوای پویای قدرتمندتر و کارآمدتر در سمت سرور پشتیبانی می‌کنند، از جمله آن‌ها می‌توان به Active Server Pages مایکروسافت و Java servlet اشاره نمود.



Server-Side Includes (SSI)

بسیاری از وب سرورها نیز از Includeهای سمت سرور پشتیبانی می‌کنند. اگر منبعی به عنوان حاوی server-side include علامت گذاری شود، سرور محتویات منبع را قبل از ارسال به مشتری پردازش می‌کند.

محتویات برای الگوهای مشخص خاصی اسکن می‌شوند (اغلب در کامنت‌های خاص HTML موجود است)، که می‌توانند نام متغیرها یا اسکریپت‌های Embedded باشند. الگوهای ویژه با مقادیر متغیرها یا خروجی اسکریپت‌های اجرایی جایگزین می‌شوند. این یک راه آسان برای ایجاد محتوای پویا است.

Access Controls

وب سرورها همچنین می‌توانند کنترل‌های دسترسی را به منابع خاصی اختصاص دهند. هنگامی که درخواستی به یک منبع کنترل شده می‌رسد، وب سرور می‌تواند دسترسی را بر اساس آدرس IP کلاینت کنترل کند، یا می‌تواند یک چالش رمز عبور برای دسترسی به منبع صادر کند.

Step 5: Building Responses

هنگامی که وب سرور منبع را شناسایی کرد، عملی که در متد درخواست توضیح داده شده را انجام می‌دهد و پیام پاسخ را برمی‌گرداند. پیام پاسخ حاوی کد وضعیت پاسخ، هدرهای پاسخ و بدنه پاسخ در صورت ایجاد کد است.

Response Entities

اگر تراکنش یک بدنه پاسخ ایجاد کند، محتوا همراه با پیام پاسخ ارسال می‌شود. اگر بدنه‌ای وجود داشت، پیام پاسخ معمولاً شامل موارد زیر است:

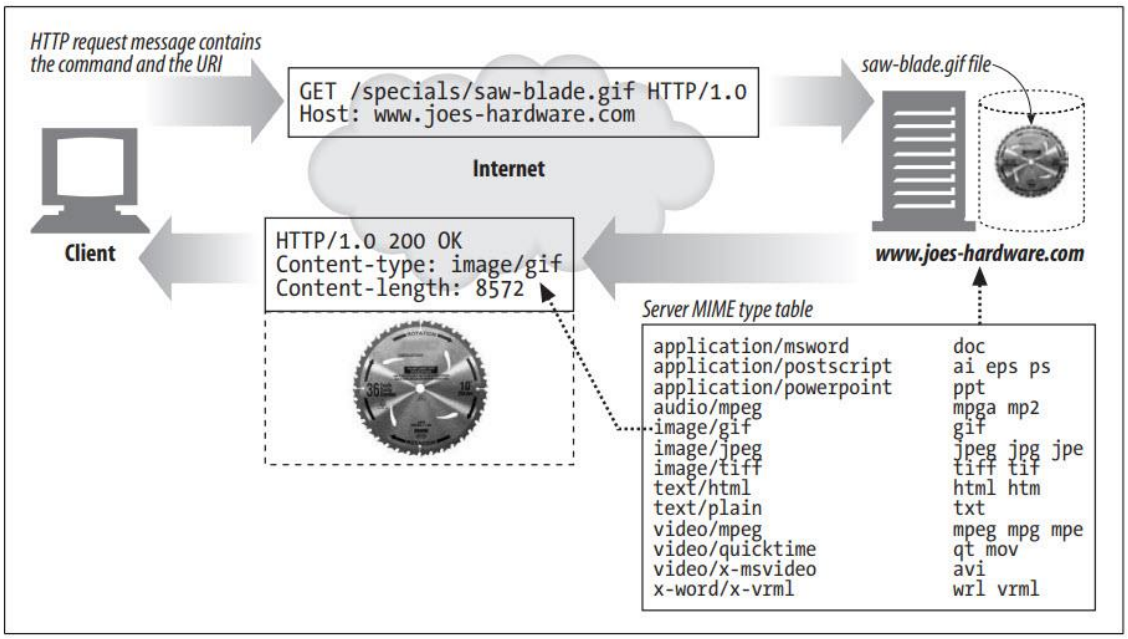
- یک هدر Content-Type که نوع MIME بدنه پاسخ را توصیف می‌کند.
- یک هدر Content-Length که اندازه بدنه پاسخ را توصیف می‌کند.
- محتوای واقعی پیام

MIME Typing

وب سرور وظیفه تعیین نوع MIME بدنه پاسخ را بر عهده دارد. راه‌های زیادی برای پیکربندی سرورها برای مرتبط کردن انواع MIME با منابع وجود دارد:

mime.types

وب سرور می‌تواند از پسوند نام فایل برای نشان دادن نوع MIME استفاده کند. وب سرور یک فایل حاوی انواع MIME را برای هر پسوند اسکن می‌کند تا نوع MIME را برای هر منبع محاسبه کند. این مدل extension-based، رایج ترین حالت در این بخش است.



Magic typing

وب سرور آپاچی می‌تواند محتویات هر منبع را اسکن کند و محتوا را با جدولی از الگوهای شناخته شده (به نام magic file) مطابقت دهد تا نوع MIME را برای هر فایل تعیین نماید. این فرآیند می‌تواند کند باشد، اما راحت است، به خصوص اگر فایل‌ها بدون پسوند استاندارد نامگذاری شوند.

Explicit typing

سرورهای وب را می‌توان به گونه‌ای پیکربندی کرد که فایل‌ها یا محتویات دایرکتوری خاصی را مجبور به داشتن نوع MIME، صرف نظر از پسوند یا محتویات فایل کنند.

Type negotiation

برخی از وب سرورها را می‌توان به گونه ای پیکربندی کرد که یک منبع را در قالب‌های مختلف سند ذخیره کند. در این مورد، وب سرور را می‌توان به گونه‌ای پیکربندی نمود که «بهترین» قالب مورد استفاده (و نوع MIME مربوطه) را توسط فرآیند مذاکره با کاربر تعیین کند. ما در این مورد در فصل‌های آینده بحث خواهد شد.

سرورهای وب همچنین می‌توانند پیکربندی شوند تا فایل‌های خاصی را با انواع MIME مرتبط کنند.

Redirection

وب سرورها گاهی اوقات به جای پیام‌های موفقیت آمیز، پاسخ‌های تغییر مسیر را برمی‌گردانند. یک وب سرور می‌تواند مرورگر را برای انجام درخواست، به جای دیگری هدایت کند. پاسخ تغییر مسیر با یک کد بازگشتی 3xx نشان داده می‌شود. هدر Location در پاسخ، شامل یک URI برای مکان جدید یا مکان ترجیحی محتوا است. تغییر مسیر برای موارد زیر مفید است:

Permanently moved resources

ممکن است یک منبع به مکان جدیدی منتقل شده باشد، یا نام آن تغییر کرده باشد و یک URL جدید به آن بدهد. وب سرور می‌تواند به کلاینت بگوید که نام منبع تغییر کرده است و کلاینت می‌تواند قبل از Fetch نمودن منبع از مکان جدید، هر Bookmark و موارد دیگر را به روز کند. کد وضعیت 301 Moved Permanently برای این نوع تغییر مسیر استفاده می‌شود.

Temporarily moved resources

اگر منبعی به طور موقت منتقل یا تغییر نام داده شود، سرور ممکن است بخواهد کلاینت را به مکان جدید هدایت کند. اما، از آنجایی که تغییر نام موقتی است، سرور از کلاینت می‌خواهد که در آینده با

URL قدیمی بازگردد و هیچ Bookmark ای را به روز نکند. کدهای وضعیت 303 See Other و 307 Temporary Redirect برای این نوع تغییر مسیر استفاده می‌شود.

URL augmentation

سرورها اغلب از تغییر مسیرها برای بازنویسی URLها و همچنین برای Embed Context استفاده می‌کنند. هنگامی که درخواست به سرور می‌رسد، سرور یک URL جدید حاوی اطلاعات وضعیت Embedded تولید می‌کند و کاربر را به این URL جدید هدایت می‌نماید. این یک روش مفید برای حفظ وضعیت در سراسر تراکنش‌ها است.

کدهای وضعیت 303 See Other و 307 Temporary Redirect برای این نوع تغییر مسیر استفاده می‌شود.

Load balancing

اگر سروری که Overloaded شده، درخواستی دریافت کند، سرور می‌تواند کلاینت را به سروری با Load کمتر هدایت کند. کدهای وضعیت 303 See Other و 307 Temporary Redirect برای این نوع تغییر مسیر استفاده می‌شود.

Server affinity

سرورهای وب ممکن است یکسری اطلاعات محلی برای کاربران خاصی داشته باشند. یک سرور می‌تواند کلاینت را به سروری هدایت کند که حاوی اطلاعاتی درباره کلاینت است. کدهای وضعیت 303 See Other و 307 Temporary Redirect برای این نوع تغییر مسیر استفاده می‌شود.

Canonicalizing directory names

هنگامی که یک کلاینت یک URI برای نام دایرکتوری بدون اسلش آخر درخواست می‌کند، بیشتر سرورهای وب، کلاینت را با اضافه کردن اسلش به یک URI هدایت می‌کنند تا پیوندهای نسبی به درستی کار کنند.

Step 6: Sending Responses

سرورهای وب با مشکلات مشابهی در ارسال داده در سراسر اتصالات مانند دریافت مواجه می‌شوند. سرور ممکن است اتصالات زیادی به بسیاری از کلاینت‌ها داشته باشد، برخی غیرفعال هستند، برخی داده‌ها را به سرور ارسال می‌کنند و برخی داده‌های پاسخ را به کلاینت‌ها برمی‌گرداند.

سرور باید وضعیت اتصال را ردیابی کند و اتصالات دائمی را با دقت خاصی مدیریت کند. برای اتصالات غیرمداوم، از سرور انتظار می‌رود که هنگام ارسال کل پیام، سمت اتصال خود را ببندد.

برای اتصالات دائمی، اتصال ممکن است باز بماند، در این صورت سرور باید برای محاسبه صحیح هدر Content-Length بسیار محتاط باشد، در غیر این صورت کلاینت راهی برای دانستن زمان پایان پاسخ نخواهد داشت (به فصل ۴ مراجعه کنید).

Step 7: Logging

در نهایت، هنگامی که یک تراکنش کامل شد، وب سرور یک ورودی را در فایل Log یادداشت می‌کند که تراکنش انجام شده را توصیف می‌کند. اکثر وب سرورها چندین فرم قابل تنظیم از Logging را ارائه می‌دهند.

For More Information

http://www.w3c.org/Jigsaw/

http://www.ietf.org/rfc/rfc1413.txt

Apache: The Definitive Guide

Ben Laurie and Peter Laurie, O’Reilly & Associates, Inc.

Professional Apache

Peter Wainwright, Wrox Press.