

به نام خدا

## تحقیق درس مهندسی اینترنت – فاز سوم

### Web Server

زینب عباسی مزار – ۸۶۲۱۳۰۷۳

Web Server هم می‌تواند به سخت‌افزار (کامپیوتر) و هم به نرم‌افزار (برنامه کاربردی کامپیوتر) اشاره کند. نرم‌افزار به دریافت مطالب از اینترنت کمک می‌کند. مهم‌ترین مورد استفاده از web serverها میزبانی web site است، اما موارد استفاده دیگری مانند نگهداری اطلاعات برای برنامه‌های کاربردی مهم را نیز انجام می‌دهند.

### کلیات

وظیفه اصلی یک وب‌سرور این است که بر اساس درخواست clientها، صفحات وب را به آنها برساند که شامل documentهای HTML و همه محتویاتی که می‌تواند در یک document جا بگیرد، مانند تصاویر، stylesheetها و کدهای JavaScript، می‌شود.

یک client، معمولاً یک web browser یا یک web crawler، ارتباط را با ایجاد یک درخواست HTTP برای دریافت یک منبع مشخص آغاز می‌نماید و سرور با ارسال آن منبع و یا پیغام خطا به درخواست client پاسخ می‌دهد. منبع یادشده معمولاً یک فایل موجود روی حافظه ثانویه سرور است، اما این شیوه نگهداری الزامی نیست و به نحوه پیاده‌سازی سرور بازمی‌گردد.

از آنجاکه وظیفه اصلی وب‌سرور، ارائه محتوا است، باید بتواند با استفاده از HTTP، مطالبی را از clientها نیز دریافت نماید؛ همان‌طور که در هنگام دریافت اطلاعات در فرم‌ها چنین کاری انجام می‌شود.

عموم وب سرورها از اسکریپت نویسی server-side مانند PHP و Apache HTTP Server پشتیبانی می-کنند. این بدان معناست که می توان در فایل های جداگانه ای با اسکریپت نویسی رفتار سرور را تعیین کرد، در حالی که اصل نرم افزار سرور بدون تغییر باقی مانده است. معمولاً از این مسئله برای ساخت document های HTML دینامیک در مقابل صفحات استاتیک استفاده می شود که اولی نیاز به دسترسی به database ها دارد، در حالی که دومی بسیار سریع و ساده ساخته و نگهداری می شود.

وب سرورها همیشه در اینترنت استفاده نمی شوند، بلکه می توان آن ها را در دستگاه هایی مانند printer, webcam و router ها نیز یافت؛ در این حالت، وب سرور تنها به یک شبکه محلی خدمت رسانی می کند. می توان در صورت نیاز از وب سرور برای سیستم monitoring یا مدیریت یک دستگاه استفاده کرد؛ و این یعنی به غیر از web browser که در اکثر os های امروزی موجود است، به نرم افزار اضافه دیگری نیاز نیست.

## ویژگی های کلی

- ✓ Virtual Hosting برای پشتیبانی از چندین وب سایت که از یک IP Address استفاده می کنند.
- ✓ Large File Support برای ایجاد قابلیت پشتیبانی از فایل هایی که حجم آن ها بیش از 2MB است، در OS های 32bit.
- ✓ Bandwidth Throttling برای محدود کردن سرعت پاسخ دهی، برای این که شبکه اشباع نشود و از طرفی بتوان به تعداد بیش تری client پاسخ داد.
- ✓ Server-side Scripting برای ساخت صفحات وب دینامیک؛ در عین حال پیاده سازی وب سرور و وب سایت هم چنان جدا از یکدیگر انجام می شود.

## ترجمه مسیر

وب سرورها قادرند مؤلفه های مسیر یک URL را به موارد زیر تبدیل کنند:

✓ یک فایل سیستمی محلی (برای درخواست‌های استاتیک).

✓ یک برنامه داخلی یا خارجی (بای درخواست‌های دینامیک).

برای یک درخواست استاتیک، مسیر URL مشخص شده توسط client به directory اصلی وب سرور بستگی

دارد. به URL زیر که می‌تواند توسط client درخواست شود، توجه کنید:

```
http://www.example.com/path/file.html
```

user agent این client، این URL را به یک ارتباط با `www.example.com` با درخواست HTTP 1.1 زیر

تفسیر می‌کند:

```
GET /path/file.html HTTP/1.1
```

```
Host: www.example.com
```

وب سرور موجود در `www.example.com` مسیر داده شده را به مسیر directory اصلی خود اضافه می‌کند.

در یک سرور `apache`، معمولاً این مسیر، `/home/www` و در ماشین‌های UNIX معمولاً `/var/www` است. نتیجه

فایل سیستمی محلی است:

```
/home/www/path/file.html
```

سپس وب سرور فایل را می‌خواند و اگر وجود داشت، پاسخی را به مرورگر client ارسال می‌نماید. پاسخ،

محتویات فایل را شرح خواهد داد و اصل فایل را دربر خواهد داشت. اگر فایل وجود نداشته باشد یا قابل دسترسی

نباشد، پیغام خطا ارسال خواهد شد.

## محدودیت‌های Load کاربران

یک وب سرور (بخش نرم‌افزاری) محدودیت‌هایی را در زمینه میزان load ایجاد می‌نماید، زیرا تنها می‌تواند به

تعداد محدودی از Client های concurrent برای هر IP Address، پاسخ‌گو باشد (این تعداد معمولاً بین ۲ تا

۸۰,۰۰۰ و به‌طور پیش‌فرض از ۵۰۰ تا ۱,۰۰۰ است) و تنها می‌تواند ماکزیمم تعداد مشخصی از درخواست‌ها را

در هر ثانیه پاسخ دهد؛ که این تعداد به موارد زیر بستگی دارد:

- ✓ تنظیمات اصلی وب سرور.
  - ✓ نوع درخواست HTTP ارسال شده.
  - ✓ شکل کلی محتوا(دینامیک یا استاتیک).
  - ✓ این که محتوای مورد درخواست، در cache موجود است یا خیر.
  - ✓ محدودیت‌های نرم‌افزاری و سخت‌افزاری سیستم‌عاملی که وب سرور روی آن کار می‌کند.
- اگر یک وب سرور از این محدودیت‌ها عبور کند، پاسخ‌گوی درخواست‌های رسیده نخواهد بود.

## وب سرورهای User-Mode و Kernel-Mode

یک وب سرور را می‌توان هم در هسته سیستم‌عامل و هم در فضای کاربری(همانند دیگر برنامه‌های کاربردی) پیاده‌سازی کرد.

یک وب سرور In-Kernel(مانند TUX در GNU یا Linux و یا Microsoft IIS در Windows)، معمولاً سریع‌تر کار می‌کند، زیرا به‌عنوان یک بخش از سیستم می‌تواند به‌طور مستقیم از منابع سخت‌افزاری مورد نیاز خود، مانند حافظه non-paged، بخش‌های زمانی CPU، adapterهای شبکه یا بافرها، استفاده کند.

وب سرورهایی که در حالت user به کار برده می‌شوند، باید از سیستم برای استفاده بیش‌تر از حافظه و یا امکانات CPU اجازه بگیرد. نه تنها درخواست‌های مربوط به kernel زمان زیادی را صرف می‌کنند، بلکه همواره نیز به این‌گونه درخواست‌ها پاسخ داده نمی‌شود؛ زیرا سیستم، منابع خود را برای استفاده خود رزرو کرده‌است و نمی‌تواند منابع سخت‌افزاری را با همه برنامه‌های کاربردی در حال اجرا به اشتراک بگذارد.

هم‌چنین، برنامه‌های کاربردی نمی‌توانند به بافرهای داخلی سیستم دسترسی داشته‌باشند، زیرا باعث ایجاد بافرهای کپی‌شده بی‌فایده است که مانع دیگری را برای سرورهای user-mode به وجود می‌آورد. نتیجتاً تنها راهی که یک سرور user-mode بتواند با استفاده از آن از نظر کارایی با سرورهای kernel-mode برابری کند، این است که کیفیت کد آن بالا برود، مشابه کدی که در سرورهای اجراشده در kernel استفاده می‌شود. این

مسئله، موضوع بسیار مهمی در windows به شمار می رود؛ زیرا در windows، سربار کار برای حدود ۶ برابر Linux است.

## **تأثیرات overload:**

وب سرورها می توانند در هر زمانی به دلایل زیر دچار overload شوند:

✓ ترافیک وب بیش از حد مجاز: هزارها و حتی میلیون ها client در یک بازه زمانی کوتاه به یک سایت متصل شوند؛ به این مسئله، Slashdot Effect گفته می شود.

✓ حملات DDoS.

✓ Worm های کامپیوتری که گاهی به علت آلوده کردن میلیون ها کامپیوتر، ترافیک غیرعادی ایجاد می کنند.

✓ ویروس های XSS می توانند به سبب آلوده کردن میلیون ها browser و یا وب سرور، ترافیک بالایی را ایجاد نمایند.

✓ Bot های اینترنتی: گاهی overload به سبب عدم کنترل یا عدم فیلتر وبسایت های بزرگی که منابع محدودی دارند، ایجاد می گردد.

✓ پایین آمدن سرعت اینترنت (شبکه): در این حالت، درخواست های client با سرعت کمتری پاسخ داده خواهد شد و تعداد connection ها افزایش می یابد؛ به نحوی که سرور دچار overload می شود.

✓ در دسترس نبودن بخشی از وب سرورها (کامپیوترها): این مسئله می تواند به دلیل تعمیرات یا به روزرسانی ضروری، خرابی سخت افزاری یا نرم افزاری، خرابی back-end یا database و غیره رخ

دهد؛ در چنین حالت‌هایی، وب‌سرورهای دیگر ترافیک بالایی را تحمل خواهند کرد و overload پیش خواهد آمد.

### **نشانه‌های overload:**

نشانه‌های overload در یک وب‌سرور عبارتند از:

- ✓ درخواست‌ها با تأخیر نسبتاً طولانی (از ۱ ثانیه تا چند صد ثانیه) پاسخ داده می‌شوند.
- ✓ کدهای خطای 500، 502، 503 و 504 از خطاهای HTTP (گاهاً پیام‌های خطای نامربوط مانند خطای 404 یا 408) به clientها بازمی‌گردد.
- ✓ Connectionهای TCP، پیش از آن‌که هیچ مطلبی به سمت client فرستاده شود، رد و یا reset می‌شوند (دچار وقفه می‌گردند).
- ✓ در برخی موارد نادر، تنها بخش‌هایی از مطلب فرستاده می‌شود (البته به این مسئله باید به‌عنوان bug نگاه شود).

### **تکنیک‌های مقابله با overload:**

برای فائق آمدن بر محدودیت‌های مربوط به load و جلوگیری از overload، بسیاری از وب‌سایت‌ها از تکنیک‌های متدوالی مانند تکنیک‌های زیر استفاده می‌کنند:

✓ مدیریت ترافیک شبکه با استفاده از:

- *Firewall*: برای بلوکه کردن ترافیک‌های رسیده از IP source های ناشناخته یا با pattern نامناسب.
- *HTTP Traffic Manager*: برای drop کردن، redirect کردن و یا بازنویسی درخواست‌هایی که pattern نامناسبی دارند.

• *Traffic Shaping* یا *Bandwidth Management*: برای اینکه نقطه peak ترافیک را در

شبکه کاهش دهند.

✓ استفاده از تکنیک‌های **Web Cache**.

✓ استفاده از domain name های مختلف برای ارائه اشکال مختلف محتوا (دینامیک یا استاتیک) با

استفاده از وب سرورهای مختلف:

• <http://images.example.com>

• <http://www.example.ccom>

✓ استفاده از domain name ها و یا کامپیوترهای مختلف برای جداسازی فایل‌های بزرگ از فایل‌های

کوچک و متوسط؛ ایده اصلی در این رابطه این است که با استفاده از تنظیمات مختلف برای هر

اندازه از فایل‌ها بتوان به راحتی فایل‌های کوچک و متوسط را cache کرد و در عین حال به خوبی

بتوان فایل‌های بزرگ (بالتر از 10 تا 100 MB) را ارائه داد.

✓ استفاده از چندین وب سرور نرم‌افزاری در هر کامپیوتر که هر کدام محدود به کارت شبکه خود بوده

و IP Address خود را دارند.

✓ استفاده از چندین وب سرور سخت‌افزاری که با یکدیگر یک گروه را ساخته‌اند؛ این گروه را می‌توان

به شکل یک وب سرور بزرگ دید.

✓ افزودن امکانات سخت‌افزاری بیش‌تر (نظیر RAM، Disk و غیره) به هر کامپیوتر.

✓ تنظیم پارامترهای OS در مورد توانایی‌های سخت‌افزاری.

✓ استفاده از برنامه‌های کامپیوتری بهینه‌تر برای وب سرورها.

✓ استفاده از راه‌کارهای دیگر برای مواجهه با مشکلات، خصوصاً اگر از محتوای دینامیک استفاده می‌-

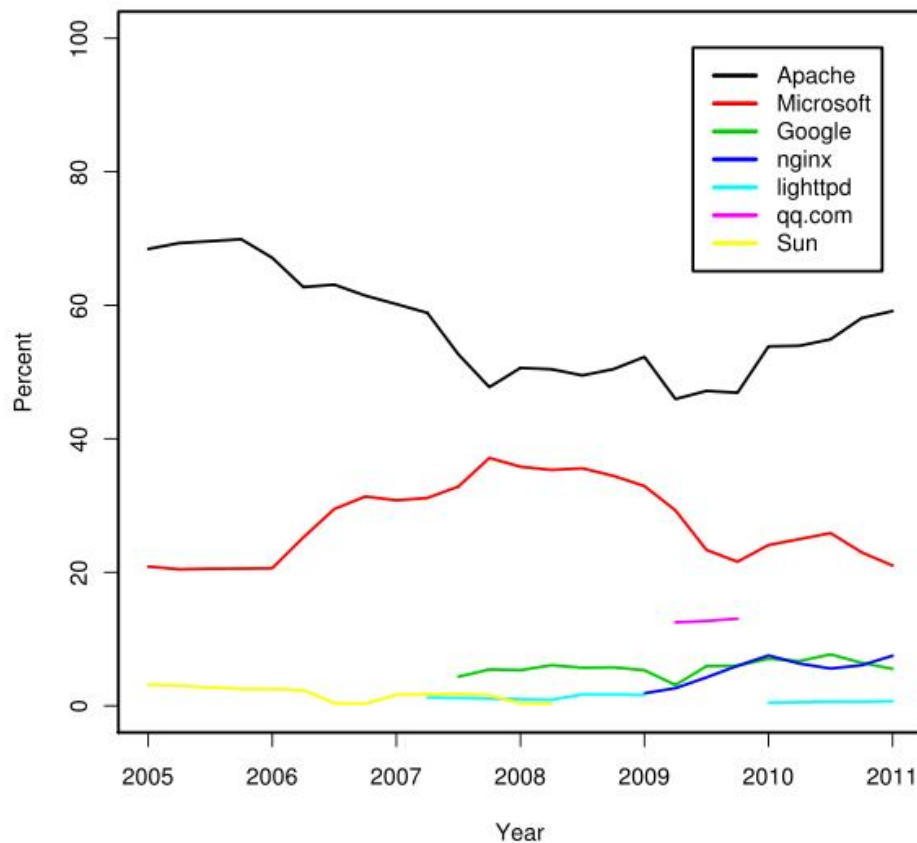
شود.

## Market Structure

در جدول و نمودار زیر، آخرین آمارها در مورد نحوه اشتراک بازار بین وب سرورهای مختلف آمده است.

<i>Vendor</i>	<i>Product</i>	<i>Web Sites Hosted</i>	<i>Percent</i>
<i>Apache</i>	<i>Apache</i>	179,720,332	60.31%
<i>Microsoft</i>	<i>IIS</i>	57,644,692	19.34%
<i>Igor Sysoev</i>	<i>nginx</i>	22,806,060	7.65%
<i>Google</i>	<i>GWS</i>	15,161,530	5.09%
<i>lighttpd</i>	<i>lighttpd</i>	1,796,471	0.60%

Usage share of web servers



Source: Netcraft