

websites
&



webservers



Op dit lesmateriaal is een Creative Commons licentie van toepassing.

© 2014 - 2024 Remie Woudt en Alborz Salimian Rizi

remie.woudt@gmail.com

a.salimian@broklede.nl

De afbeeldingen op het voorblad zijn van Xufan Zhang

My "artistic" creation is Hilbert curve, a curve which is able to cover the whole plane, or well known as a "2D curve". I define 4 types of "U" shapes which together form the curve. The key point is to draw the bridges across all curves. (Actually, it is the bridges that form the whole curve.) To add some taste, I incorporated random colors and "case-dependent" line thickness.

Bron: <http://www.cs.princeton.edu/courses/archive/fall10/cos126/art/index.php>

Inhoud

1

Websites & webservers

1.1 Webservers

Afbeelding 1: Gebruik webservers

Opdracht 1

1.2 IP adressen en DNS

Opdracht 2

Opdracht 3

Afbeelding 2: Een kaart van het internet

1.3 Client-server systemen

Opdracht 4

1

Websites & webservers

Wat heb je zoal nodig als je een website wilt gaan maken en op internet zichtbaar wilt laten zijn. In dit hoofdstuk gaan we ons vooral daarover buigen.

1.1 Webservers

Een **webserver** is een computerprogramma. Gewoonlijk draait dat programma op een computer die verbonden is met internet. Ook zo'n computer wordt meestal webserver genoemd maar in feite is dat onjuist. Het kan namelijk ook heel goed voorkomen dat op zo'n computer meerdere servers draaien. Bijvoorbeeld ook een database server.

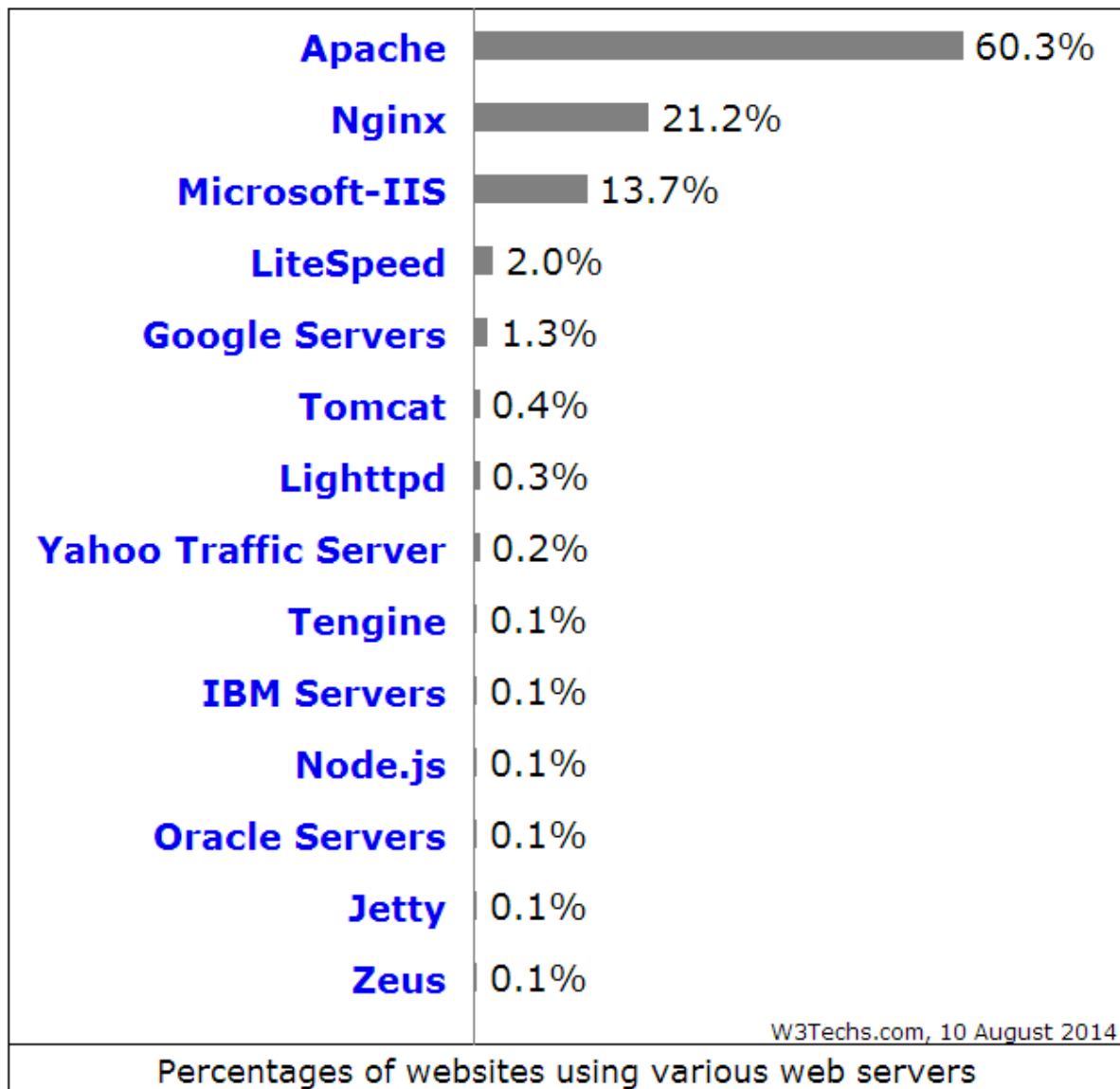
Als op een computer maar één serverprogramma draait noemen we dat een **dedicated server**. Maar soms wordt een serverprogramma, en dat geldt dan vooral voor heel druk bezochte servers, ook over meerdere computers verdeeld. In dat geval spreken we over een **clustered server**. Je mag aannemen dat dat bijvoorbeeld het geval is bij websites als Google of Facebook.

Als we hier spreken over het internet kan het ook een intern netwerk zijn. In zo'n geval noemen we dat een **intranet**.

De webserver krijgt, meestal via het internet, verzoeken om bepaalde documenten te laten zien. De webserver stuurt deze dan naar die computer die dat verzoek doet. Zo'n verzoek kan natuurlijk niet zomaar een leuk briefje zijn want dan zou de webserver er niets van begrijpen. Zo'n verzoek moet voldoen aan het Hypertext Transfer Protocol, afgekort **HTTP**. En dat is iets wat je vast wel eens in de browser hebt zien staan of hebt ingetypt. Een browser, zoals Internet Explorer, Mozilla Firefox, Safari, Opera of Google Chrome, weet hoe een verzoek volgens het HTTP-protocol eruit moet zien en zij kunnen het dan ook op die manier opstellen. Zo'n verzoek hoeft trouwens niet via het internet te lopen. Het kan ook binnen één en dezelfde computer plaatsvinden. Als je de onderdelen SQL, Databases en PHP gevolgd hebt dan heb je met een webserver gewerkt op een USB-stick. Daarbij maakten we gebruik van **Nginx** als webserver.

De documenten die aan de webserver worden gevraagd moeten uiteraard te verwerken zijn door de browser. De meeste documenten zullen zogenaamde HTML documenten zijn en daar is de browser uitstekend voor toegerust. Alhoewel de browsers over de interpretatie nog wel eens willen verschillen. Zoek de verschillen maar eens tussen <http://doc-type.nl/> in Internet Explorer en Google Chrome. De interpretatie van JavaScript is nog een stuk lastiger!

Er zijn veel aanbieders van webserver. De meest gebruikte is Apache. Op de tweede plaats komt Nginx. Beide programma's zijn gratis te downloaden in tegenstelling tot Internet Information Services (IIS) van Microsoft. In onderstaande afbeelding krijg je een beeld van het marktaandeel van webserver.



Afbeelding 1: Gebruik webserver

De snelst groeiende webserver is Nginx (spreek uit als engine X) en deze wordt dan ook geroemd om zijn stabiliteit en het grote aantal verzoeken dat hij kan verwerken. Een andere goede, net als Nginx zeer geschikt voor de wat kleinere computers (bijvoorbeeld een RaspberryPi) is Lighttpd. Kijk [hier](#) voor een overzicht van het gebruik van webserver per grootte van de website.

De pagina's die door een webserver naar de browser worden gestuurd kunnen uit verschillende bronnen komen. Het kunnen gewone pagina's zijn zoals wij die ook met HTML gemaakt hebben maar ze kunnen ook uitvoer zijn van andere

computerprogramma's die bijvoorbeeld een database bevragen om zo een HTML pagina te genereren.

Eén van de mogelijkheden van een webserver om met andere computerprogramma's te communiceren is **CGI** (Common Gateway Interface) of de snellere variant **FastCGI**. Veelgebruikte scripttalen die gebruik maken van CGI zijn Perl, Python en Ruby.

Bij CGI wordt, elke keer als een CGI-programma wordt opgevraagd, een nieuw proces gestart. Bij niet al te veel verzoeken gaat dat prima maar wanneer het aantal verzoeken toeneemt bestaat de mogelijkheid dat er capaciteitsproblemen kunnen ontstaan. Een oplossing hiervoor is FastCGI dat het eenmaal gestarte proces open houdt en wacht op een nieuw proces. Ook technologieën zoals **PHP**, **Active Server Pages** (ASP) van Microsoft en **ColdFusion** (nu van Adobe) maken gebruik van soortgelijke oplossingen. In de Apache webserver is het inmiddels mogelijk om op die manier CGI-programma's te laten werken door het installeren van modules als mod-perl, mod-python of mod-wsgi (ook voor Python).

Zoals hierboven al genoemd zijn er speciale talen die vooral gebruikt worden op webserver. Daarvan is PHP de meest gebruikte. PHP is in 1994 ontworpen door Rasmus Lerdorf, werkzaam bij IBM. Oorspronkelijk was het de afkorting van Personal Home Page (of om helemaal precies te zijn: Personal Home Page/Forms Interpreter, PHP/FI) maar tegenwoordig is de betekenis "PHP: Hypertext Preprocessor". Het grappige aan deze naam is dat het een, wat men wel noemt, recursief acroniem is. Als je namelijk de eerste drie letters voluit gaat schrijven ontstaat de naam binnen de naam. Een dergelijke constructie, een functie die zichzelf aanroept, komt wel regelmatig voor in een computerprogramma en noemt men recursief programmeren.

In je browser zul je een PHP pagina niet herkennen omdat PHP van de pagina die hij genereert gewoon HTML maakt en dit daarna naar de browser stuurt. PHP lijkt veel op C maar in tegenstelling tot C is het mogelijk, met name in PHP5, om object georiënteerd te programmeren.

De andere hier genoemde talen zijn over het algemeen talen die niet specifiek voor het maken van websites zijn gemaakt maar daar wel veel voor gebruikt worden. Zo kun je ook heel goed gewone computerprogramma's schrijven met Python. Voor al deze talen geldt dat ze bij toepassingen met webserver HTML documenten genereren.

De meeste webserver draaien op een computer waarop **Linux** als het operating systeem is geïnstalleerd. Dit wordt in de hand gewerkt doordat bij veel standaard Linux distributies Apache gewoon mee geleverd wordt. Een grote site als Marktplaats begon met zogenaamde **LAMP software**, Linux, Apache, MySQL en

PHP/Perl/Python. Al gauw hadden ze daarvoor zo'n 50 computers nodig die met elkaar verbonden waren om de enorme vraag bij te kunnen houden.

Opdracht 1

Zowel voor Ruby als voor Python zijn **frameworks** gemaakt om snel websites te kunnen bouwen.

- Onderzoek wat een framework is en noem voor zowel Ruby als voor Python minimaal één framework.

1.2 IP adressen en DNS

Als een webserver verbonden is met het internet betekent dat automatisch dat hem een uniek **IP-adres** is toegekend. Een IP-adres is een combinatie van 4 cijfergroepen en dat ziet er dan bijvoorbeeld uit als: **195.78.84.31**.

In theorie kan een IP adres bestaan uit 4 maal de getallen 0 t/m 255. Maar er zijn een aantal gereserveerd voor andere doeleinden.

Zo zijn de adresreeksen **10.0.0.0 – 10.255.255.255**, **172.16.0.0 – 172.31.255.255** en **192.168.0.0 – 192.168.255.255** gereserveerd voor interne netwerken.

In de begintijd dacht men dat er ruim voldoende IP adressen had en zo zijn er hele reeksen door grote bedrijven gereserveerd. IBM bijvoorbeeld is eigenaar van de gehele reeks **9.0.0.0 – 9.255.255.255**.

Opdracht 2

- Hoeveel IP adressen heeft IBM?

Een IP adres is er voor twee doeleinden, identificatie en adressering.

Een computer is binnen het internet te herkennen aan zijn IP adres. Als er een webpagina wordt opgevraagd wordt er met het verzoek om de pagina tevens informatie over het IP adres meegestuurd zodat het adres bekend is waar de opgevraagde pagina naar toe moet.

Websites worden normaliter aangeroepen met hun domeinnaam. Maar op het internet zijn **DNS servers** (Domain Name Servers) aanwezig die weten welk IP adres bij welke domeinnaam hoort en ze kunnen op die manier de juiste webserver identificeren.

Het aantal IP adressen raakt op. Met de nodige kunstgrepen zoals grote interne netwerken lukt het het nog even vol te houden maar het is duidelijk dat er op zeer korte termijn wat aan gedaan moet worden. De oplossing is er ook al en wordt ondertussen ook al gebruikt: **IPv6**. (Het huidige systeem wordt wel **IPv4** genoemd.) Bij IPv6 bestaat een IP adres uit 8 groepen van 4 hexadecimale getallen. Een IPv6 adres kan er dus zo uit gaan zien:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

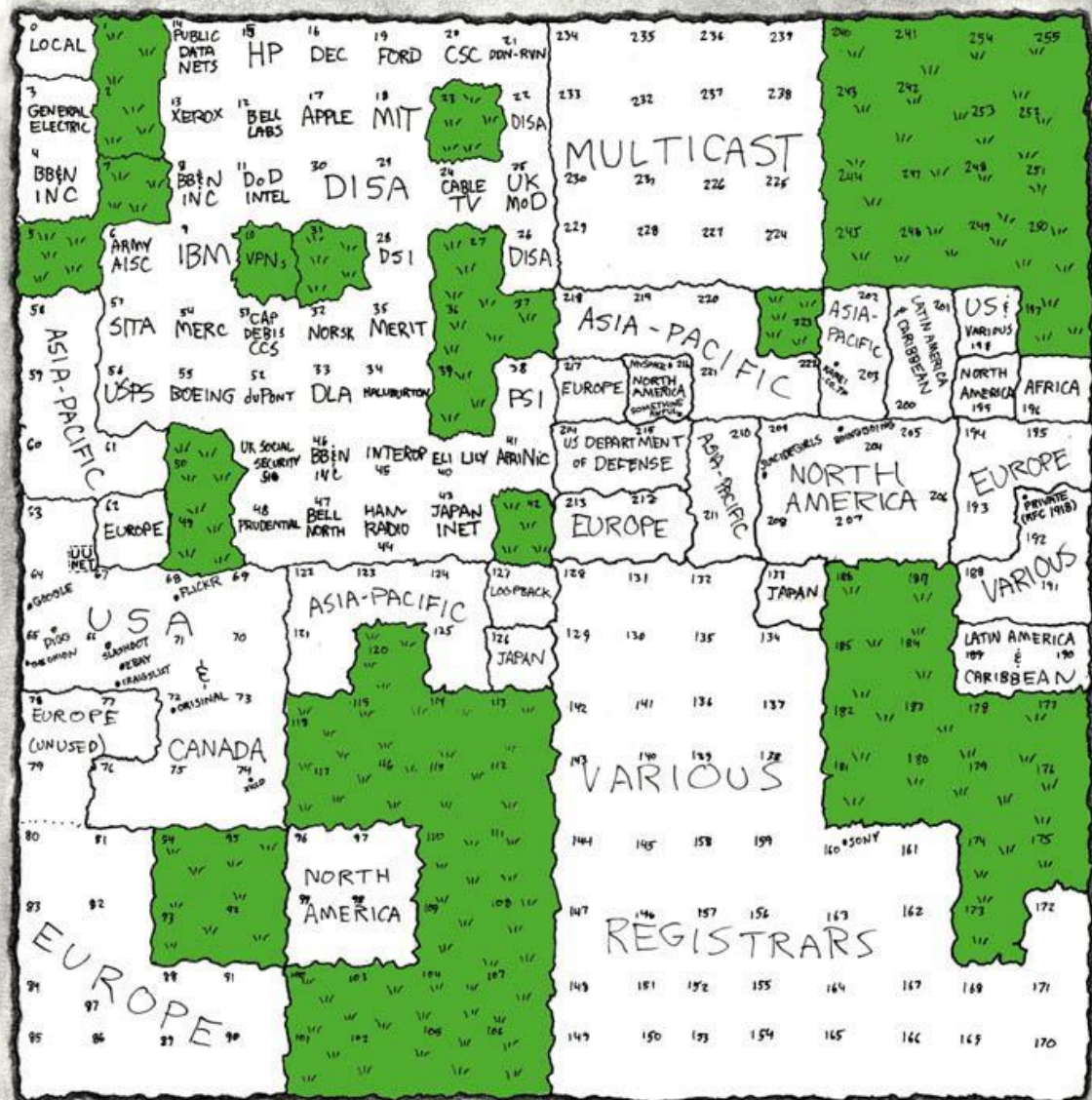
Daar heeft men nog wel een vereenvoudigde schrijfwijze voor bedacht, zoals het weglaten van eventuele voorste nullen in ieder groepje en door groepjes van nullen ook weg te laten. Dit IPv6 systeem zou ons de beschikking geven over $3,4 \times 10^{38}$ unieke adressen. Met IPv4 hebben we slechts de beschikking over $4,3 \times 10^9$ adressen. Met het nieuwe IPv6 hebben we per m² aardoppervlak de beschikking over 666.581.710.174.193.485.144.050 IP adressen. En daar kunnen we wel weer even mee vooruit.

Opdracht 3

- Bestudeer ook de website:
<http://computer.howstuffworks.com/web-server.htm>

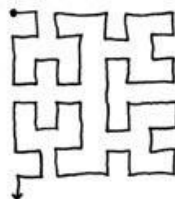
MAP OF THE INTERNET

THE IPv4 SPACE, 2006



THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING--ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990'S BEFORE THE RIR'S TOOK OVER ALLOCATION.

0	1	14	15	16	19 →
3	2	13	12	17	18
4	7	8	11		
5	6	9	10		



 = UNALLOCATED BLOCK

Afbeelding 2: Een kaart van het internet

1.3 Client-server systemen

Een browser wordt wel client genoemd. Als je met een browser contact maakt met een webserver heb je dus een **client-server** systeem. Ook spreken we van e-mail clients en die maken dan contact met een mail server.

Over het algemeen kunnen we zeggen dat een client-server systeem een systeem is waarbij de client een verzoek indient bij de server en de server geeft antwoord op dat verzoek.

Het is overigens niet zo dat client-server systemen zich alleen maar op het internet bevinden. Ook binnen een netwerk kun je gebruik maken van een dergelijk systeem. Wanneer je met een workstation, dit kan een gewone PC zijn maar ook een terminal, verbinding maakt met bijvoorbeeld een database op een centrale server is er ook sprake van een client-server systeem.

De tegenhanger van een client-server systeem is het **peer-to-peer** systeem. Daarbij gaat het om gelijkwaardige computers die via een netwerk of via het internet contact met elkaar maken en bijvoorbeeld gegevens of bestanden uitwisselen. Een bekend voorbeeld daarvan is **bittorrent**. Alhoewel we dan wel spreken van een bittorrent client omdat deze verbinding maakt met een bittorrent server, vindt de uiteindelijke bestandsuitwisseling toch gewoon tussen de aangesloten computers zelf plaats en niet tussen de client en de server.

Opdracht 4

- Noem nog een client-server systeem dat niet in de tekst wordt genoemd.