

# Caching op het Internet

Michiel Hochstenbach<sup>1</sup>      Rob van der Mei<sup>2</sup>

Mei 2000

## 1 Inleiding

Elk jaar wordt ergens in Nederland een "Studiegroep Wiskunde met de Industrie" gehouden. Hier kunnen bedrijven wiskundig getinte problemen presenteren, waar een team van wiskundigen zich vervolgens over buigt. In november 1999 is de TU Eindhoven plaats van handeling. Een greep uit de problemen:

- Akzo Nobel vraagt zich af hoe je een gewenste kleur verf kunt benaderen met een aantal basiskleuren.
- Het NLR wil graag met behulp van foto's in een windtunnel de exacte positie van het model weten.
- Tresa is geïnteresseerd in een model van een paneel, om daarmee te kunnen voorspellen hoe het paneel buigt onder bepaalde omstandigheden.
- KPN Research onderzoekt waar in een netwerk het beste caches kunnen worden geplaatst.

Onze groep heeft zich met het laatste probleem beziggehouden. Hieronder volgt een korte impressie.

## 2 Het WWW

Het World Wide Web (WWW) heeft de laatste jaren een enorme ontwikkeling doorgemaakt. Volgens recente onderzoeken zijn er meer dan 1 miljard webpagina's, waarvan 86% Engels- en 0,54% Nederlandstalig is. Ook zijn er al zo'n 600 miljoen email-adressen.

Hoewel ze vaak als synoniemen worden gebruikt, is er een verschil tussen de begrippen WWW en Internet. Met het Internet worden alle computers bedoeld die informatie met elkaar kunnen uitwisselen. Dit is mogelijk doordat er gebruik wordt gemaakt van vaste communicatie-protocollen, regels voor bijvoorbeeld het uitwisselen van documenten. Deze protocollen zijn niet geheim, zodat iedereen in principe zijn computer aan het internet kan

---

<sup>1</sup> Universiteit Utrecht, [www.math.uu.nl/people/hochsten](http://www.math.uu.nl/people/hochsten), email: [hochsten@math.uu.nl](mailto:hochsten@math.uu.nl)

<sup>2</sup> KPN Research, email: [R.D.vanderMei@kpn.com](mailto:R.D.vanderMei@kpn.com)

koppelen. Bovendien wordt het Internet decentraal beheerd, dit houdt in dat er (in tegenstelling tot bijvoorbeeld telefoonnetwerken) er niet één instantie is die alles beheert en controleert. Deze twee ingrediënten zijn belangrijke succesfactoren van het Internet.

Het WWW is een service die gebruik maakt van deze infrastructuur en de gebruiker toegang geeft tot een enorme hoeveelheid informatie. Aan het eind van de jaren '80 zijn dit vooral tekstdocumenten, maar gaandeweg komen daar onder andere plaatjes, geluid en video bij. De typische spinneweb-structuur ontstaat door de uitvinding van hyperlinks, waardoor vanuit een document gemakkelijk een ander kan worden opgeroepen.

Door het enorme dataverkeer komt de snelheid van de elektronische snelweg wel in het geding. Net als in het gewone verkeer zijn er ook op Internet spitsuren, bijvoorbeeld na het avondeten. Het downloaden van bestanden kan dan erg lang duren. Bovendien kunnen overbelaste 'Web-servers' onvoorspelbaar gedrag vertonen, wat gevolgen kan hebben voor de kwaliteit van de dienstverlening. Omdat er verwacht wordt dat het Web-gebruik voorlopig gigantisch blijft groeien, dreigt dit probleem in de toekomst alleen nog maar groter te worden. 'Caching' kan een mogelijke oplossing bieden.

### 3 Wat is caching?

Caching is het 'in de buurt' van de gebruiker opslaan van veel gevraagde informatie. Wanneer een gebruiker een document bij een bepaalde Web-server opvraagt, dan kan caching de data op de computer van de gebruiker, of op een andere computer 'onderweg' (bijvoorbeeld die van zijn Internet 'provider') opslaan. Wanneer nu deze gebruiker dezelfde informatie weer opvraagt, kan deze veel sneller worden opgehaald dan de eerste keer.

Caching heeft veel voordelen. Niet alleen heeft de gebruiker het opgevraagde document sneller, het vermindert tevens het dataverkeer op het Internet, waar het overige verkeer weer van kan profiteren. Bovendien raakt de computer waar het origineel staat minder snel overbelast, en kunnen providers toe met minder capaciteit voor hun gebruikers.

Documenten kunnen op verschillende plaatsen worden gecached. Bijvoorbeeld op de computer van de gebruiker, waardoor in een 'Web-browser' de 'Back' en 'Forward' knop snel het document weer oproepen. Hiervan heeft een 'buurgebruiker' echter geen profijt. De meest effectieve manier van cachen is daarom vaak het gebruik maken van zogenaamde 'proxy servers', waar kopiën van veelgevraagde documenten worden bewaard. Proxies kunnen op verschillende plaatsen in het netwerk worden ingezet. Overigens is het ook een interessante (en veel onderzochte) vraag welke documenten we het beste kunnen opslaan in de cache, maar daar gaan we hier verder niet op in.

### 4 Kansverdeling van de vraag naar documenten

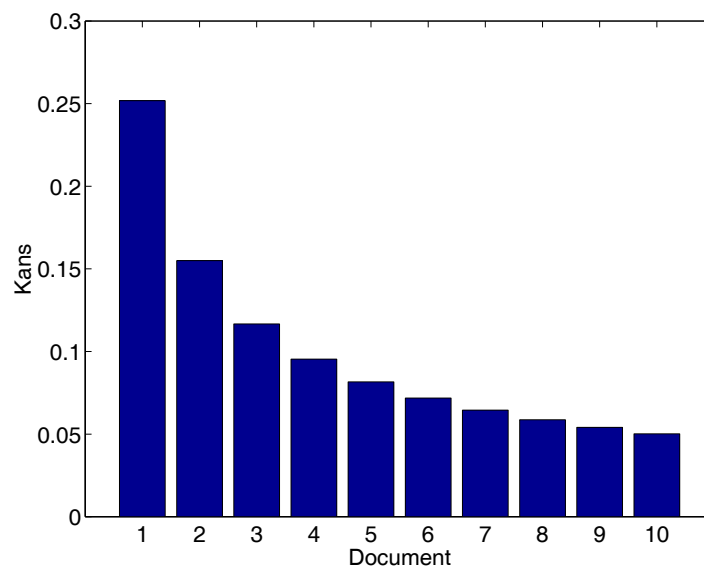
Er zijn allerlei tellingen gedaan om de populariteit van documenten op het Web te onderzoeken. Stel dat er  $N$  documenten op een bepaalde server aanwezig zijn, die we naar afnemende populariteit ordenen. Laat nu  $f_j$  het deel van de aanvragen zijn dat het  $j^e$  document opvraagt,  $f_j$  wordt ook wel de relatieve hit-frequentie van document  $j$  genoemd. Dan geldt dus

$$f_1 \geq f_2 \geq \dots \geq f_N > 0 \quad \text{en} \quad f_1 + f_2 + \dots + f_N = 1.$$

Uit vele metingen blijkt nu dat de relatieve hit-frequenties bij benadering aan de volgende relatie voldoen:

$$f_j = j^{-\alpha}/S, \quad \text{voor } j = 1, \dots, N,$$

waarbij  $\alpha$  een parameter is die per server kan verschillen, en  $S$  de bijbehorende normalisatieconstante is ( $S = \sum_{j=1}^N j^{-\alpha}$ ). Deze verdeling wordt in de kansrekening ook wel de Zipf- of Zeta-verdeling met parameters  $\alpha$  en  $N$  genoemd. De  $\alpha$  geeft een maat voor de spreiding van de hit-frequenties. Een hoge waarde voor  $\alpha$  betekent dat vrijwel alle aanvragen op een server naar een paar zeer populaire documenten gaan, terwijl een lage waarde van  $\alpha$  betekent dat de populariteit van de verschillende documenten op een server meer verdeeld is. Uit de metingen blijkt dat  $\alpha$  vaak tussen de 0.6 en 0.8 ligt. In Figuur 1 is de Zipf-verdeling met 10 documenten en parameter  $\alpha = 0.7$  afgebeeld.



Figuur 1: De Zipf-verdeling met 10 documenten en parameter  $\alpha = 0.7$ . Merk op dat op een dubbel-logarithmische schaal de grafiek een rechte lijn met richtingscoëfficiënt  $-\alpha$  is.

De naam Zeta-verdeling is afkomstig van de functie

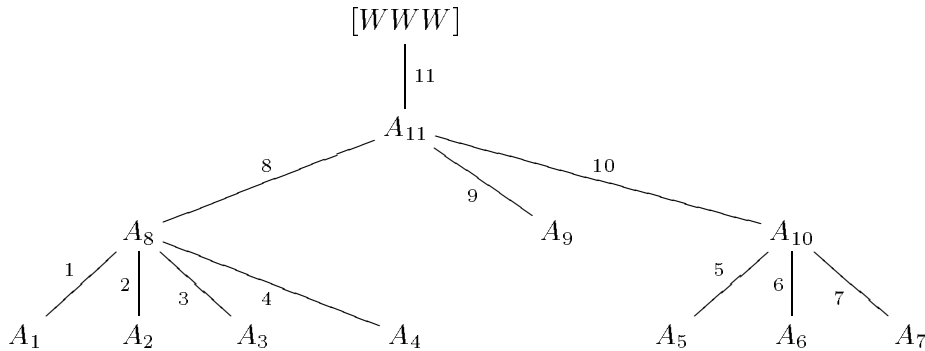
$$\zeta(s) := 1 + 2^{-s} + 3^{-s} + \dots,$$

die bekend staat als de Riemann Zeta-functie, genoemd naar de wiskundige G.F.B. Riemann. De Zeta-verdeling werd gebruikt door de econoom Pareto om de verdeling van inkomens in een gegeven land te beschrijven. G.K. Zipf gebruikte deze verdeling voor verschillende andere toepassingen en maakte daarmee de Zipf/Zeta-verdeling populair.

Het lijkt geen toeval te zijn dat we op de Zipf-verdeling zijn uitgekomen. Uit andere onderzoeken blijkt namelijk dat niet alleen de vraag naar pagina's op een server Zipf-verdeeld is, ook de vraag van de medewerkers van een bedrijf naar documenten op het Web. Maar bijvoorbeeld ook het voorkomen van woorden in een boek (met 'de' en 'en' als koplopers in Nederlandse boeken) en de gevraagde boeken in een bibliotheek. En, wat in de toekomst steeds belangrijker gaat worden, ook de vraag naar pagina's van verschillende bedrijven. Want met behulp van Internet kan veel omzet gemaakt worden. Bovendien kan een druk bezochte pagina profiteren van grote reclame-inkomsten.

## 5 Waar plaatsen we cache?

We bekijken een netwerk met een boomstructuur, zoals bijvoorbeeld in Figuur 2: elke  $A_i$



Figuur 2: Een voorbeeld van de boomstructuur van een netwerk met hoogte 3 en maximaal 4 kinderen.

('knoop') stelt een computer voor, en elk lijnstuk tussen twee knopen een kabelverbinding. We willen nu een aantal caches plaatsen in het netwerk met als doel (bijvoorbeeld) de gemiddelde wachttijd van de gebruikers te minimaliseren. Als er een bepaald budget  $B$  is om dit te doen, dan is een interessante vraag hoeveel en waar de caches het best geplaatst kunnen worden.

We nemen aan dat we op elke knoop cache kunnen plaatsen en dat de kosten om een cache op knoop  $i$  te plaatsen en te onderhouden lineair van de grootte van de cache afhangen:

$$k_i(y) = \begin{cases} a_i + b_i y & (y > 0) \\ 0 & (y = 0) \end{cases},$$

waarbij de vaste kosten  $a_i$  de variabele kosten  $b_i y$  duidelijk overheersen. Dit betekent dat we liefst een zo klein mogelijk aantal caches willen, waarbij de grootte van de cache van ondergeschikt belang is.

Aanvankelijk wilden we dit probleem als studiegroep aanpakken met stochastische methoden, waar in de vorige sectie een begin mee is gemaakt. We kunnen dan onder meer de staart van een Zipf-verdeling tegenkomen (als een gebruiker zijn meest gevraagde documenten uit een locale cache kan halen, van zijn vraag naar documenten blijft dan effectief de 'staart' over), of de som van een aantal Zipf-verdelingen (de totale vraag van verschillende gebruikers, waarbij ieder natuurlijk zijn eigen favoriete documenten heeft), of weer combinaties daarvan. Ook hebben we gekeken naar een analyse met behulp van Markov ketens. Maar omdat de tijd beperkt was, hebben we het probleem voornamelijk 'deterministisch' (niet stochastisch) benaderd.

We zijn er namelijk vanuit gegaan dat we de beschikking hebben over een verzameling historische data, meetgegevens over het internet-vraaggedrag van de gebruikers. Dit is in het algemeen een realistische veronderstelling.

We nummeren de verbindingen en geven met  $c_i$  de transmissie-capaciteit van de  $i^e$  verbinding aan, dat is de hoeveelheid informatie (bijvoorbeeld uitgedrukt in bits per seconde) die de verbinding maximaal aankan. De gemiddelde hoeveelheid data die op het drukste moment van de dag daadwerkelijk door de  $i^e$  verbinding gaat noteren we met  $x_i$ .

Uit de literatuur is bekend dat er een relatie bestaat tussen de capaciteit, de hoeveelheid data en de gemiddelde wachttijd  $t_i$ , dit is de gemiddelde tijd die de data erover doet om over

de  $i^e$  verbinding getransporteerd te worden. Voor elke verbinding is er een constante  $\gamma_i > 0$  zodat

$$t_i \approx \gamma_i \cdot \left(1 - \frac{x_i}{c_i}\right)^{-1}.$$

We lezen hier ondermeer uit af dat als de hoeveelheid data door de verbinding de capaciteit van de verbinding nadert, de wachttijd snel groter wordt.

Willen we nu dat de gemiddelde wachttijd op elke verbinding kleiner is dan een constante  $T$ , dan is dit equivalent met de eis dat

$$(1) \quad x_i < \left(1 - \frac{\gamma_i}{T}\right) \cdot c_i$$

Met andere woorden: de hoeveelheid data op een verbinding moet kleiner zijn dan een constante maal de capaciteit van die verbinding. Voor het bepalen van de plaatsen voor de caches, gebruiken we nu de historische data.

We beginnen bij de onderste laag van de boom. Als de hoeveelheid data  $x_1$  voldoet aan (1), dan is er geen cache nodig bij computer  $A_1$ . Als  $x_1$  te hoog is, dan plaatsen we het grootste document dat er vanuit computer  $A_1$  wordt gevraagd, in de cache van  $A_1$  (of het meest gevraagde, of een combinatie van beide). Is de hoeveelheid data  $x_1$ , waarvan we nu dit document hebben afgehaald, nog steeds te groot, dan plaatsen we meer documenten in de cache, net zolang totdat  $x_1$  wel voldoet aan (1). Merk op dat dit ook invloed heeft op  $x_8$ . Dit doen we consequent door de hele boom heen, van onder naar boven werkend.

Aan het einde tellen we de vaste kosten (de  $a_i$ 's) op van de caches die geplaatst moeten worden. Gaat dit al over ons budget heen, dan is het snelheids criterium is gezien het budget niet haalbaar. Hebben we daarentegen nog centen over ( $B < \sum k_i$ ), dan kunnen we het netwerk nog sneller maken door in (1), in plaats van de factoren  $(1 - \gamma_i/T)$ , kleinere factoren te nemen. We kunnen ons ook concentreren op verbindingen die bedrijfskritisch zijn, waar de verhouding  $x_i/c_i$  het grootst is, of die extreme piekuren hebben. In wiskundige termen: we kunnen een soort 'steepest descent' methode toepassen, in de richting waar de gradiënt van de doelfunctie het meest negatief is (caches plaatsen op knopen die een belangrijke invloed hebben op de gemiddelde wachttijd).

Tenslotte kunnen we de documenten die we in ons algoritme in de caches hebben geplaatst, gebruiken om bij benadering de grootte van de cache te bepalen. In de praktijk zal een cache meestal een vaste grootte hebben, en beheerd worden met bijvoorbeeld de 'Least Recently Used'-methode.

Hiermee hebben we in het kort een aanpak geschetst op de vraag van KPN Research hoe we ons budget goed kunnen besteden om caches te plaatsen in een netwerk.

## 6 Referenties

Zie [www.useit.com/alertbox/zipf\\_referring.html](http://www.useit.com/alertbox/zipf_referring.html) voor meer info over de Zipf-verdeling en de toepassing op het Internet. Wie geïnteresseerd is in meer informatie of referenties kan contact opnemen.