Veel projecten willen graag meeliften op het succes

DNS is ook een database

Rick van Rein

Hoewel het op het eerste gezicht een vreemde stelling lijkt, is het Domain Name System (DNS) een speciale vorm van een database. RDBMS en DNS gaan zelfs steeds meer op elkaar lijken.

Het domeinnaamsysteem DNS kan worden beschouwd als een wereldomspannende hiërarchische database. Zonder relationele data en dus ook zonder SQL, maar wel met veel aspecten van replicatie en redundantie die pas veel later voor relationele databases werden ontwikkeld.
Een DNS query vindt, in het meest algemene geval, in een paar stappen plaats. Het begint bij een root name server. Stel, we zoeken bakker.orvelte.nep, dan vragen we dat eerst aan een root name server, bijvoorbeeld a.root-servers.net met het vaste IP-nummer 198.41.0.4. De root-name server kent orvelte.nep niet, maar weet wel de name server voor .nep domeinen te noemen. Dat is een name server voor het top-level domein nep. Die vragen we nogmaals naar bakker.orvelte.nep, maar ook een top-level domain name server weet dat niet. Dus worden we doorgestuurd naar de name servers die zijn geconfigureerd voor het domain ('jut en jul', zie kader). Die weten doorgaans wel te vertellen welk IP-nummer bij bakker.orvelte.nep hoort.

De reden dat DNS een hiërarchische database wordt genoemd is vanwege de zoekstructuur, die van algemeen naar steeds concreter gaat, of van de wortel van een boom naar de takken. Er is een strikte hiërarchie van domeinnamen. Eigenlijk is die centrale herkomst heel vreemd voor het internet, maar het is wel een heel praktisch en goed werkend systeem, en dat is wel weer typisch iets voor het internet.

Verdeel en heers

Wat vooral opvalt is dat het beheer van al de name servers die meedoen in een DNS-query, door verschillende partijen beheerd kunnen worden. Zo zijn de root name servers in handen van partijen die nauw verwant zijn aan de Internet Engineering Task Force; in Nederland staat bijvoorbeeld een host bij RIPE, die ook de Europese verdeling van IP-nummers regelt. RIPE verwijst bijvoorbeeld naar de name servers voor .nl domeinen, maar die worden weer door een andere partij beheert, namelijk SIDN (waar .nl domeinen worden geregistreerd, dat wil zeggen opgenomen in de name servers die voor .nl domeinen de toon zetten). En vanuit SIDN wordt weer verwezen naar allerlei name servers bij allerlei providers, desnoods aan de andere kant van de wereld. Het verdelen van het beheer over al die data maakt van DNS een bonte mengeling van namen, die niet per se consistent van structuur is. Toch werkt het prima. Het werkt zelfs razendsnel, want in milliseconden is het IP-nummer onder elke domeinnaam te vinden. DNS werkt vaak zo goed dat niemand er erg in heeft dat het bestaat!

Een gedistribueerde zoekindex, die in een paar niveaus doorverwijst en toch zo snel antwoord geeft, dat is ronduit indrukwekkend. Er is dan ook veel werk in gaan zitten om het netwerkverkeer te optimaliseren. Er wordt zo veel mogelijk gebruik gemaakt van het UDP protocol, dat onbetrouwbaar maar razendsnel is.
De onbetrouwbaarheid bestaat eruit dat soms pakketjes verloren gaan, en dan moet (na een timeout) opnieuw dezelfde vraag verzonden worden. Dat doet DNS dan ook, zonder zeuren, want als het via TCP zou moeten dan zou dat ook gebeuren, alleen in een andere netwerklaag, maar vooral zou er heel veel gecommuniceerd moeten worden om een verbinding op te zetten en af te breken. DNS huppelt liever zo flitsend over het netwerk als UDP, dan zorgvuldig en traag zoals TCP.

> Het datamodel van DNS ligt even vast als de recordtypes terwijl een RDBMS een lokaal datamodel gebruikt

Het is dankzij deze keuze voor flitsende communicatie dat DNS zo snel is dat het nauwelijks opvalt. Het ligt niet voor de hand om iets dergelijks ook met een RDBMS uit te halen, maar toch is het zo gek nog niet; veel toepassingen gebruiken PHP boven een database, en doen voor een pagina een paar kleine query's. Dergelijke toepassingen zouden flink winst kunnen boeken met een lichter mechanisme. Dat blijkt ook wel uit de gevoelde noodzaak om 'connection pools' in te richten, ter voorkoming van de lange opstarttijd van een verbinding met een SQL interpreter.

Delegatie van query's

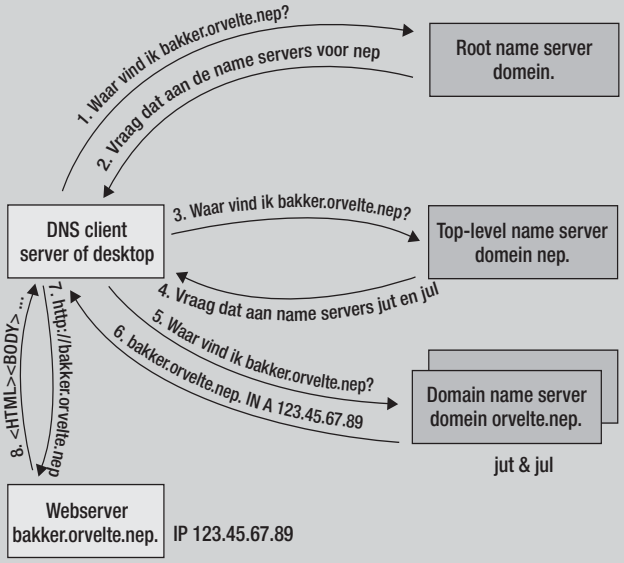
In het inleidende voorbeeld sprak de computer die een IP-nummer zocht rechtstreeks de root name server aan. Dat was eigenlijk een beetje al te simpel gesteld. In werkelijkheid worden zulke verzoeken gebundeld via een caching name server in de nabijheid van de nieuwsgierige server- of desktop-computer.
Een caching name server is de name server die elke internet provider door haar klanten laat installeren. De verzoeken van de aldus geconfigureerde host gaan dan via die caching name server naar de root name server en wat daar zoal onder valt.
De taak van de caching name server is vooral om records te onthouden, en zo de hoeveelheid verkeer te minimaliseren. Daartoe geeft elk record aan hoe lang het mag worden gecached.
De caching name server hoort bijvoorbeeld dat de DNS voor orvelte.nep de komende dag nog wel op jut en jul blijft draaien, en door die tijd af te laten tikken kan de caching name server die kennis vasthouden voor eventuele verdere verzoeken. Surft een gebruiker dus van bakker.orvelte.nep via een reclamelink naar veldwachter.orvelte.nep, dan weet de caching name server dat hij rechtstreeks op jut of jul af kan stappen.
Het voordeel dat DNS hierbij heeft ten opzichte van een RDBMS is dat de data vrijwel nooit veranderen. Dat is altijd een situatie waarin caching voordelig werkt. Maar het is wat flauw om hiermee DNS als fundamenteel anders dan een RDBMS af te doen. DNS zet de data namelijk wel een beetje naar zijn hand. Als het IP-nummer achter bakker.orvelte.nep verandert, dan is er een tijdje een inconsistent beeld op het internet. Hoewel jut en jul het nieuwe adres uitdragen, zullen er nog caches zijn die het oude IP-nummer uitdragen, en de tijd aftellen tot ze jut of jul weer lastigvallen voor bakker.orvelte.nep.
De beheerder van de DNS voor het orvelte.nep domein kiest zelf hoe lang de cache een opgezocht resultaat vasthoudt. Gebruikelijk is bijvoorbeeld een dag; daarbij houden caches de data enige tijd vast, terwijl de tijd dat er een inconsistent beeld heerst over de domeinnaam redelijk beperkt blijft.
De beheerder weet dat gedurende de inconsistente periode ondersteuning voor de (web)servers voor bakker.orvelte.nep moet draaien op zowel het oude als het nieuwe IP-nummer. En de beheerder weet ook dat dat een dag moet gaan duren.
Dit kan nooit in zijn algemeenheid op RDBMS'en worden toegepast. De oplossing voor inconsistentie is specifiek voor het kennisdomein van de opgeslagen data (IP-nummers kun je best een tijdje dubbelop bezetten) maar is niet algemeen mogelijk. Toch stemt het tot nadenken. In specifieke gevallen is hier best

Korte introductie DNS

Voor wie niet precies op de hoogte is met DNS: het is het systeem dat vooral domeinnamen in IP-nummers vertaalt. Browsers en allerlei andere tools maken daar gebruik van om mensen te laten werken met goed te onthouden namen, terwijl de computer kan werken met de IP-nummers die direct toepasbaar zijn in bijvoorbeeld routeringsbeslissingen.
DNS kan echter veel meer aan dan alleen deze eenvoudige query's. Allerlei soorten data kunnen onder een domeinnaam gehangen worden, door gebruik te maken van daartoe geëigende recordtypes. Daarbij schuilt er in de hiërarchie meer dan alleen hostnamen; ook IP-nummers zitten erin verweven, telefoonnummers en zo nog wat gekkigheid. Al deze dingen kunnen dus als zoeksleutel worden gebruikt.
Een applicatie die een IP-nummer nodig heeft stuurt een query naar een name server, en krijgt een antwoord terug zoals

```
orvelte.nep. IN A 123.45.67.89
orvelte.nep. IN NS jut.digiboeren.nep.
orvelte.nep. IN NS jul.digiboeren.nep.
```

De eerste regel verraadt het antwoord. IN A is het recordtype een IP-nummer op het internet, orvelte.nep. de gezochte naam. De extra informatie in de IN NS records geeft aan dat jut en jul de name servers zijn die dienst doen voor het gezochte domein. Merk op dat er meerdere antwoorden mogelijk zijn, ook binnen hetzelfde recordtype.
Er komen steeds meer recordtypes bij. Zo was het gebruikelijk om alleen inkomende mail-servers voor een domein te vermelden, maar spam en phishing hebben het interessant gemaakt om ook de uitgaande servers expliciet te maken. Dat kan volgens het Sender Policy Framework van spf.pobox.com; vooralsnog in textuele annotaties in DNS (via TXT recordtypes) maar idealiter zou SPF een eigen recordtype krijgen. En zo zijn er nog heel veel meer projecten die graag meeliften op het succes van de database die DNS heet.



Afbeelding 1: Het DNS protocol verricht razendsnelle wonderen. Het opzoeken gebeurt in milliseconden dankzij het UDP protocol.

winst mogelijk; zie ook eerdere artikelen in DB/M over gesplitste databases.

Soorten servers

In DNS zijn niet alle servers gelijkwaardig. Sommige servers hebben recht van spreken (dat wordt met het vreselijke woord *authoritative* aangeduid) – dit houdt in dat ze geen napraters zijn, geen caching name servers. De DNS-servers die zijn aangemeld voor een domeinnaam hebben recht van spreken; jut en jul uit het kader hebben dat bijvoorbeeld voor orvelte.nep.
De servers met recht van spreken zijn weer onderverdeeld in primaire en secundaire name servers. Een primaire name server is degene waarop de systeembeheerder wijzigingen invoert.
Een secundaire volgt de primaire, als het goed is zelfs vrijwel onmiddelijk wanneer een wijziging wordt doorgevoerd. Overigens kan het voorkomen dat de primaire name server niet gepubliceerd staat in de er hiërarchisch bovenliggende name server; dan spreken we van een verborgen primaire server. Daarmee scheidt men beheer van on-line paraatheid.

> DNS werkt vaak zo goed
> dat niemand er erg in heeft
> dat het bestaat

De communicatie tussen primaire en secundaire servers lijkt in veel op een cluster dat in single-master mode samenwerkt. In zo'n opstelling gebeuren alle wijzigingen op de master, en de andere servers in het cluster opereren als slave, en bedienen alleen read-only query's.
Toch valt er weer een belangrijk verschil op te merken. DNS-servers staan bij voorkeur over de wereld verspreid; zo kan de dichtstbijzijnde server worden verkozen boven een server op een ander continent. Bovendien is het vrijwel onmogelijk om DNS uit de lucht te schieten met een denial-of-service aanval; DNS is zo belangrijk dat een heel domein off-line lijkt wanneer DNS onbereikbaar is. Microsoft heeft dat ooit met schade en schande moeten ondervinden, omdat ze twee DNS-servers voor microsoft.com naast elkaar in de gang hadden staan, achter dezelfde uplink. Deze wereldwijde verspreiding is werkelijk uniek aan DNS. Geen relationeel database-product, open source of commerciëel, kan dit waarmaken. De oorzaak is het superstrakke transactionele concept, dat in dit blad al vaker onder de loep is genomen.

Recordtypes

Door de unieke eigenschappen van DNS wordt het steeds vaker gebruikt, of wellicht misbruikt, om allerhande informatie in op te slaan. Bijvoorbeeld iemands certificaat waarmee digitale handtekeningen gezet worden. Of een lijst van contactgegevens, opgeslagen onder iemands telefoonnummer. Er zit een duidelijk ontwikkeling in DNS naar meer dynamiek in opgeslagen data.
Vroeger werd DNS vooral gebruikt voor het vinden van de volgende domeingerelateerde query's:
- hostnaam naar IP (A);
- IP naar hostnaam (PTR);
- domeinnaam naar name server (NS);
- domeinnaam naar mail server (MX);
- domeinnaam naar aliasnaam (CNAME).

Deze query's worden gedaan door de zoeksleutel (zoals de domeinnaam bakker.orvelte.nep) te versturen met het gewenste recordtype (zoals A voor vertaling van hostnaam naar IP).
De aangesproken name server zocht dit dan op en antwoordde met een A record of met een verwijzing in de vorm van CNAME of NS records.

Tegenwoordig zijn veel uitgebreidere, hippere records in de mode. Bijvoorbeeld het ENUM record, dat onder de telefoonnummer-hiërarchie wordt geplaatst. Telefoonnummers vallen in omgekeerde volgorde onder e164.arpa; bijvoorbeeld +31.12.3456789 wordt vertaald in DNS-sleutel 9.8.7.6.5.4.3.2.1.1.3.e164.arpa. Doordat de cijfers elk een nieuw DNS-niveau inluiden is het mogelijk om het beheer tussen elk cijferpaar op te splitsen. Voor ons land heeft SIDN het beheer van 1.3.e164.arpa aangevraagd, dus binnenkort moeten we (records onder) ons telefoonnummer in deze index kunnen registreren.
De hoeveelheid informatie die men in ENUM kwijt wil is ontstellend. Men wil mailadressen kwijt, alternatieve telefoonnummers, en natuurlijk ook IP-nummers en protocollen voor VoIP, en dat alles in varianten, met keuzes en in overdaad. Een wereldwijd gedistribueerde database met compleet visitekaartje, razendsnel geïndexeerd op telefoonnummers.
Onafhankelijk hiervan, maar er goed mee te combineren, wordt gewerkt aan zekerstelling van de antwoorden die DNS geeft. Dat wordt gedaan door middel van digitale ondertekening. Ook hiervoor worden nieuwe recordtypes voor in het leven geroepen.

Hoewel DNS in principe een database is met een aantal vast-omlijnde recordtypes, wordt er flink aan de weg getimmerd om dat aantal uit te breiden. Dit is lastig voor een wereldwijd verspreide database; zelfs de al jaren bestaande en inmiddels breed geaccepteerde SRV records worden nog niet overal ondersteund, gewoon omdat er ergens nog oude kinken in de nieuwe kabels zitten. Met zo veel interactie tussen servers kan dat nu eenmaal gebeuren.
Op dit punt verschillen DNS en een RDBMS echt radicaal. Het datamodel van DNS ligt even vast als de recordtypes, die hoewel hip nog altijd via standaardisatieprocessen worden gedefinieerd, terwijl een RDBMS een lokaal datamodel gebruikt. DNS en LDAP liggen wat dat betreft dichter bij elkaar dan RDBMS en DNS.

Moderne ontwikkelingen

Zoals geschetst is er wat onderscheid te maken in de soorten name servers. Er zijn caches, die vraagspecifieke acties uitvoeren. Die spelen een server-rol ten overstaan van nabije desktops en

servers, en een client-rol ten overstaan van name servers elders in de wereld. Er zijn daarnaast ook name servers die recht van spreken hebben over bepaalde domeinen, en daarvan zijn er weer primaire en secundaire.
DNS is bepaald geen wonderkind qua beveiliging. Het protocol laat zich vrij gemakkelijk overnemen, maar ook de de facto standaardsoftware BIND heeft zoveel ontwikkelingen doorgemaakt dat er regelmatig lekken worden gemeld over deze software. BIND is een soort manusje van alles, het experimenteerplatform voor nieuw te ontwikkelen DNS-standaards.
Er zijn steeds meer producten die inspelen op een behoefte aan een eenvoudiger product. Van elke name server valt wel te kiezen of dat een cache is, een primaire of secundaire name server. BIND is voor al die gevallen te configureren, maar zoals gezegd is dat een kolos. Er komen steeds meer name servers in omloop die slechts één van de genoemde DNS-functies vervullen. Dat betekent eenvoudiger software die dus minder risico op bugs kent en het betekent efficiëntere code die niet met (te) algemene gevallen rekening hoeft te houden.
Deze ontwikkeling is logisch, gezien de steeds complexer wordende DNS-structuur, en de voortdurende wens tot efficiënte afhandeling. Voor mail servers zou eenzelfde benadeing ook nuttig zijn – een mail server is inkomend of uitgaand, niet beide – en met moderne tools zoals Postfix is dat aardig te benaderen. Voor databases zou het evenzo van nut kunnen zijn om kordate keuzes te maken over welke functies op diverse nodes in een cluster draaien, zonder de complexiteit van het gehele cluster in elke node te moeten representeren.

DNS is werkelijk een database

Het moge duidelijk zijn dat DNS een ander soort database is dan een RDBMS, maar dat het wel degelijk een database is. DNS is zelfs een heel indrukwekkende database, want er is geen andere die zo gedistribueerd is, zowel in server-samenhang als in serverbeheer.

Rick van Rein
Dr. ir. H. van Rein (rick@openfortress.nl) is ontwikkelaar en beheerder bij OpenFortress Digital signatures.

Update

BO zet centrale Benelux-organisatie op

Business Objects voert een organisatiewijziging door waarmee op 1 januari 2007 een centraal aangestuurde Benelux-organisatie een feit zal zijn. Hiermee verwacht Business Objects de relaties met haar klanten en partners beter te kunnen beheren. Een voordeel van de organisatiewijziging is dat specialistische productkennis, bijvoorbeeld op het gebied van Enterprise Performance Management (EPM) en Enterprise Information Management (EIM) voor meer klanten in de regio beschikbaar komt. Daarnaast verwacht Business Objects voordelen te behalen uit operationele efficiency van de back-office processen.
De organisatieverandering zal aan het einde van Business Objects' fiscale jaar op 1 januari 2007 zijn afgerond. De organisatiewijziging heeft vooralsnog geen gevolgen voor de beide Business Objects-kantoren. Het Nederlandse kantoor in Bilthoven bestaat sinds 1995, en bedient 900 klanten. Het kantoor in Brussel is in 1997 opgericht, en bedient 600 klanten.
Zie www.businessobjects.com

Tweede DAMA Nederland seminar met Stu Carty

Op 27 oktober 2006 organiseert DAMA Nederland haar tweede bijeenkomst waar kennis kan worden gemaakt met DAMA Nederland. Key-note speaker is Stu Carty, president en oprichter van Gavilan Research Associates (GRA), en een toegewijd metadata-expert met 20 jaar ervaring in de 'enterprise software'. GRA is een consultancy organisatie die bedrijven helpt bij het evalueren en selecteren van organisatiebrede technologische oplossingen. GRA is een wereldwijde autoriteit op het gebied van MDM-applicaties en leverancier/product-onderzoeken.

Carty heeft gewerkt voor metadata-organisaties zoals onder andere Informatica, Data Advantage Group en R&O Software. Hij heeft meer dan 1000 presentaties en workshops op dit vakgebied verzorgd. Daarnaast heeft hij een honderdtal organisaties succesvol geholpen bij evaluatie, selectie en implementatie van MDM-oplossingen.
Door onder andere Basel II en de Sarbanes-Oxley Act is voor diverse organisaties de tijd aangebroken om precies te kunnen aangeven hoe bepaalde gegevens tot stand zijn gekomen. Gebruik van 'data over data' maakt het onder andere mogelijk om de ontstaansgeschiedenis van verschafte gegevens vast te leggen en inzichtelijk te maken.
De belangstelling én het belang voor metadata management groeit daarom. 'Metadata management 2007' is een interactieve discussie met vakgenoten én business managers over de laatste trends op metadatagebied, corporate (business) issues en (integrale) software-oplossingen. Het seminar is bedoeld voor DBA's, (Informatie) Architecten, Metadata Project Managers, Datawarehouse-managers en BI-managers.
Het DAMA-seminar vindt plaats op 27 oktober 2006 in Hotel AC, De Meern van 13.30 uur tot 17.30 uur.
Voor DAMA-leden is dit seminar gratis. Voor overige deelnemers bedraagt de toegang 40,- euro.

Kijk op www.damanederland.nl voor meer informatie.