

Weer thuis

Na 6 weken “doe-vakantie”. Ik was net begonnen aan een rigoureuze verbouwing van de OBA website, toen ik een telefoontje kreeg.

Ik ben door mijn rug gegaan. Zou jij...

En toen liep alles anders dan gepland.

Ik moest mijn ex-lief helpen en de website bleek nog niet zo stabiel als ik gehoopt had.

In plaats van uit bed rollen en 3 seconden later in een koele werkkamer de laatste mails lezen, werd het na het verschonen van de kattenbakken in een bloedheet rommelhok met de laptop proberen haperende software op afstand weer werkend te krijgen.

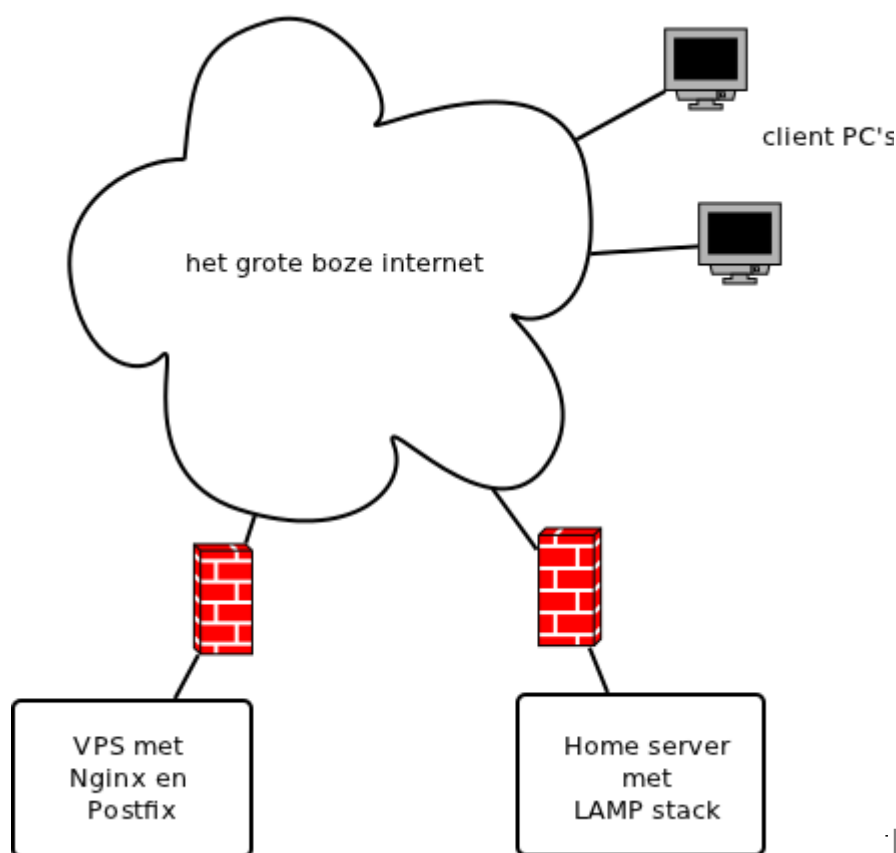
Dat is gelukt, en mijn zorgtaak is geleidelijk steeds meer een vakantie geworden.

6 Weken met iemand in een kleine ruimte geeft wel de nodige spanningen.

Als je werkt en 20 bent en elkaar nog niet zo lang kent, lijkt het allemaal ideaal.

Maar we zijn geen 20 meer, en er zijn zoveel dingen gebeurd en gezegd en de problemen van alledag zijn best wel ingewikkeld.

Wat heb ik veranderd aan de website ?



Voorheen draaide de OBA website op een VPS (een virtual private server).

Dat wil zeggen, na de

3-e verhuizing.

Begonnen op wordpress.com, daarna op de hoster mijndomein.nl en daarna op een VPS.

Sinds we [PuSH](#) gebruiken staat er in mijn huis een heel klein servertje dat niets anders deed dan wachten op publicaties en die via een XMLRPC koppeling op OBA zetten. OBA is gewoon wordpress, waar we zelf een aantal plugins voor gemaakt hebben.

De performance van wordpress op die VPS was beroerd. Het lijkt heel wat, 512 MB, maar je moet bedenken dat je op een VPS zelf niet kan instellen hoeveel geheugen gebruikt wordt voor buffers.

Wat bleek : het stond op 40 %, dus er was maar 300 MB voor Linux, een webserver, mailservers en database-server en al die andere kleine rotprogrammaatjes die je niet kan missen.

Als je via XMLRPC een update stuurde, kreeg je in 10 % van de gevallen niet binnen 60 seconden antwoord. Soms 5 keer achter elkaar (!).

Bezoekers moesten ook steeds langer wachten, soms 2 minuten.

Wat ik ook probeerde (caching plugins, webserver Apache MPM, Lighttpd, Nginx), de problemen bleven.

Bij een VPS mag je boven de afgesproken memory-limit uit komen, maar er zijn geen garanties dat dat geheugen er is. Daardoor loop je tegen segmentation-faults aan, en crashen processen geheel onverwacht, waardoor bezoekers een wit scherm zien, of functies zonder aanwijsbare oorzaak gewoon uitvallen.

PHP is de boosdoener, dacht ik.

Ik zag dat de OBA voorpagina 950 kB groot was. PHP produceert 175 kB (30 kB gecomprimeerd) HTML. De rest bestond uit plaatjes. Maar die 30 kB kost wel de meeste tijd.

PHP is leuk, ik zag ergens een 13-jarige een vraag stellen over PHP, zo simpel is het.

(lijkt het).

Maar het is ook een performance-killer, nog erger dan Java.

Laat je eigen server die PHP files verwerken en verstuur de output naar de VPS, dan kan die de plaatjes serveren, dacht ik.

Die 30kB versturen kost wat tijd, maar die tijd win je waarschijnlijk ruimschoots terug doordat je eigen servertje zwemt in het geheugen. (providers schermen met "snel internet", maar de upload-snelheid is altijd veel lager)

Dus bestelde ik een 320 Gb schijfje om het voorheen diskless servertje om te bouwen naar een echte server en zette PuSH tijdelijk op een andere server.

Wat ik bedacht had, bleek al door diverse mensen in de praktijk gebracht :

Nginx als proxyserver.

De eerste poging gebruikte Nginx op beide servers. Nginx heeft niet zoals Apache-prefork een ingebouwde PHP interpreter, maar maakt gebruik van FastCGI.

Daarbij draait PHP als een losse daemon, waarbij een supervisor-proces de verzoeken afkomstig van de webserver verdeelt over de beschikbare processen. 10 stuks leek me in eerste instantie wel genoeg.

Oeps.

Er ging nogal wat mis. Een proxyserver cached alles. Mensen konden zien wat de vorige gebruiker voor e-mail adres had ingevuld. De voorpagina voor mobiele gebruikers werd getoond aan mensen met een bureaumachine. Niet-ingelogde gebruikers zagen "hallo admin!" Ik moest een configuratie voor nginx maken die rekening hield met commenters, ingelogde gebruikers, niet ingelogde gebruikers, mobiele gebruikers etc.

Wordpress moest ook geleerd worden dat het achter een proxy zat.

Op een kwade dag ging ik na een paar heerlijke uurtjes in de tuin even kijken of de website het nog deed.

Geen beeld.

Zag je eerst nog vaak : "gateway error", nu was er helemaal niets.

Het script dat nginx herstart was niet goed. Ik had het van internet geplukt, het zou waarschijnlijk goed werken op een dedicated-server, maar niet op ons VPS waar je nooit zeker weet hoe lang iets duurt.

En PHP-FastCGI werkte gewoon niet goed. Er zit een bug in precies die versie die bij Ubuntu 12.04 hoort, en de oplossing zou "ooit" komen.

Hij crasht om de haverklap en nog erger : foutlogging werkt niet. Zonder logging ben je blind.

Ik zag maar 1 oplossing :

Apache prefork gebruiken aan de achterkant. Dat werkte. Als een zonnetje. Geen gateway-errors meer, want geen gateway.

Hadden we op de VPS maar ruimte voor 3 Apache-processen, nu mogen er 18 starten. Dat gebeurt weleens als feedwordpress aan het werk is, en de beheerder is ingelogd, en er komen via XMLRPC

updates en Google is allerlei niet gecachete pagina's aan het indexeren....

Hoe lang moet je iets cachen ?

De OBA voorpagina wordt voortdurend ververs. Als je te lang cachet, gaat het real-time karakter verloren, terwijl bij kort cachen de performance slechter wordt. Voor wordpress bestaan caching plugins die de voorpagina uit de cache gooien als er een bericht wordt toegevoegd, maar nginx is een hardware cache waar wordpress geen controle over heeft. Als je de webserver en de proxyserver op één machine draait, is er wel een plugin die de nginx-cache leeg kan gooien, maar aangezien OBA op 2 servers draait, kan dat niet.

De gekozen oplossing is :

Dubbel cachen. Hoe minder PHP hoeft te doen, hoe beter.

De proxyserver cached PHP pagina's 1 minuut en wordpress cached 1 week, of tot er iets verandert op de voorpagina.

Statische content, zoals plaatjes, javascript, css etc. cachen we in de proxyserver natuurlijk veel langer.

Het servertje waar de website op draait is niet veel sneller dan een tablet-pc. Maar veel geheugen hebben is een zegen. Je kan meer SQL-queries cachen en de xcache (daar zit gecompileerde PHP-code en variabelen in) mag ook groter zijn. Het resultaat is dat de voorpagina nu meestal tussen 1 en 2,5 seconden laadt.

Mail

Je kan vanuit een home server geen mail versturen. Je kan er natuurlijk een mailserverprogramma op zetten, maar KPN (mijn internetprovider) houdt inkomende en uitgaande mail via poort 25 tegen. (een anti-spam maatregel) Dat was dus een probleem. WordPress gaat er van uit dat er zoiets bestaat als PHP-mail, en daarmee kun je alleen binnen je eigen netwerk van 4 PC's mail versturen. De oplossing is, om een ander poortnummer te gebruiken. Mail gaat via een andere poort naar onze VPS, wordt dan in de firewall omgezet naar poort 25 en gaat door naar de mailserver. Een wordpress plugin zorgt er voor dat PHP-mail vervangen wordt door SMTP via een afwijkend poortnummer.

Perf. grade	Requests	Load time	Page size
97/100	164	868ms	613.1kB

Your website is faster than 91% of all tested websites

Pingdom.com

Is een handige website waar je kan zien hoe snel je website laadt in de optiek van een onafhankelijke waarnemer.

Op je eigen PC meet je ook je eigen netwerksnelheid en heb je te maken met browser-caching. Zoals je ziet is de pagina inmiddels wat kleiner, omdat Ina de Twitter-widget verwijderd heeft. Die widget had volgens mij geen invloed op de snelheid, omdat de pagina toch wel opgebouwd werd als er nog een informatie van Twitter ontvangen was.

Wat je ook ziet is een hoge performance-grade. Pingdom beoordeelt de maatregelen die je genomen hebt om je pagina snel te laten laden. Een bekende performance-killer die strafpunten oplevert is Gravatar. We hadden 100 Gravatars op de voorpagina staan, maar ik heb een plugin gemaakt die gravatars vervangt door lokale kopieën, vandaar.

Voor Mihai :



The image shows a performance analysis report for the website **www.mihai.nl**. The report includes a thumbnail of the website, a performance score of 64/100, 253 requests, a load time of 5.38s, and a page size of 2.3MB. It also states that the website is slower than 72% of all tested websites. There are buttons for 'DOWNLOAD HAR', 'Tweet', and 'Email'.

Perf. grade	Requests	Load time	Page size
64/100	253	5.38s	2.3MB

Your website is **slower than 72%** of all tested websites

[DOWNLOAD HAR](#) [Tweet](#) [Email](#)

