Use varnish, xmlsitemap, cron and bash to warm the cache for fast pages

Published on Dominique De Cooman (https://dominiquedecooman.com)

Dominique De Cooman

```
curl -A 'Mozilla/5.0 (X11; Linux x86_64)
  echo $line
done
wget --quiet http://$URL/nl/sitemap.xml
  curl -A 'Mozilla/5.0 (X11; Linux x86_64)
  echo $line
done
wget --quiet http://$URL/fr/sitemap.xml
  curl -A 'Mozilla/5.0 (X11; Linux x86_64)
  echo $line
```

Warming the cache gives your users extremely fast pages all the time. Using varnish, cron and bash you can give your clients very fast pages.

Extremely fast pages I mean pages with the main request that loads under 0,5sec. Something like this.

![Performance Chart](new-relic-stats.png)

The main http request is served in under 150ms which is almost not even noticeable. Loading all the other images, css and js takes around 1.28s. Further optimising could be done with sprites and CDN but for my blog I'm happy with the results for now.

```
1.1s (onload: 1.23s)
```

Varnish

Varnish is widespread technology that is easy to setup. You can find countless tutorials on how to set this up on the net. Here is a really good article: [http://andrewdunkle.com/how-install-varnish-drupal-7](http://andrewdunkle.com/how-install-varnish-drupal-7)

Cache warming

Warming your caches is pretty easy. With the xml sitemap module, a bash script and well configured cache you can do this.

Cache settings
Go to admin/config/development/performance

<table>
<thead>
<tr>
<th>CACHING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache pages for anonymous users</td>
</tr>
<tr>
<td>Cache blocks</td>
</tr>
</tbody>
</table>

Minimum cache lifetime

- 1 day

Cached pages will not be re-created until at least this much time has elapsed.

Expiration of cached pages

- 1 day

The maximum time an external cache can use an old version of a page.

Configuring the cache on a blog site to last for a day is a good idea since the content doesn't change that much. If you have a frequently commented blog then you could lower this.

Xml sitemap

Install the [http://drupal.org/project/xmlsitemap](http://drupal.org/project/xmlsitemap) By setting this up you should be able to get a map of your site urls. We will explore this to crawl the pages with our script.

Script warm-cache.sh

```bash
#!/bin/bash
ALIAS='@ddc.production'
URL='dominiquedecooman.com'

wget --quiet http://$URL/sitemap.xml --no-cache --output-document - | egrep -o "http://$URL[^<]+" | while read line; do
time curl -A 'Cache Warmer' -s -L $line > /dev/null 2>&1
    echo $line
done

wget --quiet http://$URL/fr/sitemap.xml --no-cache --output-document - | egrep -o "http://$URL[^<]+" | while read line; do
time curl -A 'Cache Warmer' -s -L $line > /dev/null 2>&1
    echo $line
done

time curl -A 'Cache Warmer' -s -L $line > /dev/null 2>&1
    echo $line
```

Page 2 of 4
Use varnish, xmlsitemap, cron and bash to warm the cache for fast pages

The script crawls all the sitemaps for each language.

Cron

Configure the cron with crontab -e

```
0 2 * * * /var/checkouts/hostingplatform/cache-warming.sh
```

This will crawl the site everyday ensuring that the cache is warm. Since it lasts for a day we know the cache is always warm. This will ensure users a fast page all the time.

Here is an image of our cache being warmed (from New Relic [8]). This slows the server for a brief time span during a time when the least users are on the site, only to make the site perform for the rest of day at lightning speed.

Want this too?

Like this: Do you want your website to perform like this? Please contact me http://dominiquedecooman.com/forms/contact-get-advice [9]