

**Loudon &
Company**

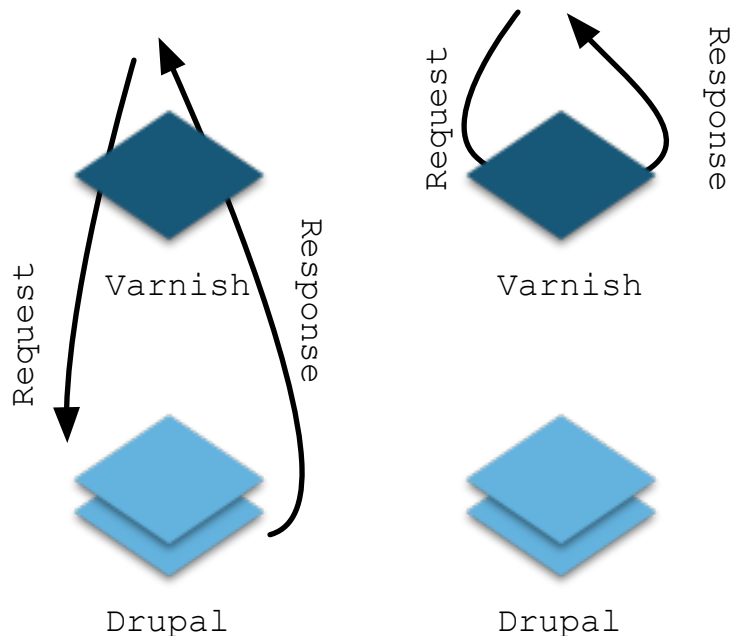
Drupal+Varnish Mini-Book

<http://loudonco.com>

How Fast?

Benchmarks vary, but Varnish can offer **10X-100X improvements** over Drupal's page cache

Initial Request (or cache MISS) **Subsequent Request (cache HIT)**



- * Think of **Varnish as an ultra fast, in-memory key-value store: URL to HTML**

- * It **works with native Drupal 7**, no hacking core :)

- * Serves **anonymous users at massive scale**

- * **Best-practices + lots of community work make installation easy** and standard across Drupal sites

- * **Stable and production-ready:** used on **high-profile sites** (D.O, whitehouse.gov, Grammy.com) and **used by Acquia and Pantheon**

Cookies - Varnish doesn't cache requests with cookies: strip them out in the Varnish VCL or use AJAX

Ecommerce - Uses sessions and cookies to track carts: use an AJAX cart

SSL - Varnish is HTTP only: put Stunnel or Nginx in front

Authenticated Users - They require sessions and cookies: ESI offers hope here but at significant complexity

Mobile - Cached pages can't execute theme-switching PHP code: either move the logic to the Varnish VCL or use Responsive Design

Shared Server - Root access is required to install/configure Varnish: either Boost instead or host with Pantheon or Acquia

Verify the Cache!

Loudon &
Company

TIP: Is This URL Cached?

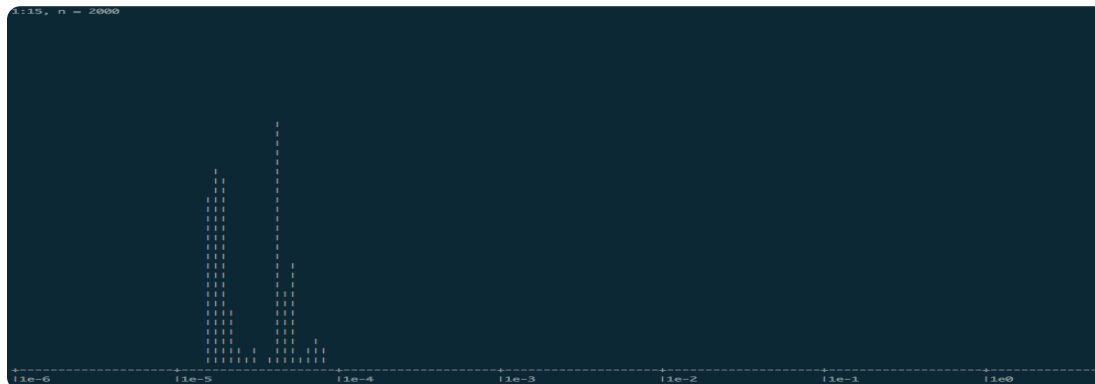
```
curl -sI http://example.com/my-url | grep -i HIT
```

```
# Get a quick list of your most popular pages
awk '{print $7}' apache-or-nginx.log | grep -Ev '(.jpg|css|js|png|woff)' | sort \
| uniq -c | sort -rn | awk '{print "http://mysite.com"$2}' | head -50 | tee urls.txt

# Run a simple bash script on urls.txt (the output from above)
# this will run curl -I against the URLs and check for cache HITs
./varnish_check.sh

# Top requests going to the backend (ie MISSES)
varnishtop -i txurl

# See MISSES and HITs live (you want pipes like example below, hashes are MISSES!)
varnishhist
```



TTL

Cache expires a set number of seconds after it was cached

Easy to understand

Easy to setup

Not very efficient, in practice Varnish caches the same HTML many times

Can cause confusion among less technical editors/staff when the new content "isn't showing up"

Content-based

Varnish purges content on changes in Drupal

Ideal: transparent to editors, staff, and users as well as maximum efficiency

Requires Drupal configuration and modules to:

- 1) Recognize content changes
- 2) Enable communication between Drupal and Varnish to purge the cached content

See Varnish module

See Purge module

See Expire module

Checklist

Loudon &
Company

Install Varnish 3 via apt-get or yum

Adjust the Varnish daemon to:

- 1) Store in RAM (preferred)
- 2) Listen on port 80

Download and tweak the Lullabot or Four Kitchens VCL—restart the service after changes.

(See Resources, you may need to e.g. adjust the backend host—AKA Drupal—IP or port)

Configure Drupal:

- 1) Update the reverse proxy section in settings.php
- 2) Enable “Cache pages for anonymous users” and set the max TTL in the admin GUI at /admin/config/development/performance

Install expiration/caching modules if desired

Check that your pages are caching! You may need to change modules, recreate pages, use AJAX, etc.

PROFIT

Four Kitchens VCL - Uses a whitelist for Cookies, adds HIT or MISS header, grips content, and provides 404, 500, and 301 caching (404 caching is crucial, Drupal bootstraps these!)

VCL Device Detection - If Responsive Design is not possible, device detection requires VCL changes, passing headers back to Drupal, and (custom) code to handle the headers in Drupal

Lullabot VCL - Very similar to the 4K VCL, but doesn't have 404 caching. Provides an example of multiple backend servers and a nicely detailed accompanying blog post.

Varnish CLI - a quick overview

Varnish Request Diagram - good for deep customization/ESI

Varnish Ecommerce - uses authcache on the bleeding edge!