The importance of Drupal Cache

Luis F. Ribeiro
Ci&T Inc. 2013
Introduction

• Caio Ciao Luppi

• Software Architect at Ci&T Inc.

• More than 4 years of experience with Drupal Development

• Experience with Application Development and Support using Drupal 6 and Drupal 7 Platforms
Established in 1995, Ci&T provides value-driven, nearshore, web and mobile application services and software product engineering. The company's high-performance teams employ agile methodologies and Lean principles to deliver innovative, value-generating projects for companies of all sizes by leveraging the latest mobile, cloud and web technologies. A strong commitment to innovation has put Ci&T on the forefront of discovering new ways to deliver value and eliminate waste to help clients reduce costs and improve quality, speed and business agility.

Ci&T's Drupal developers have more than five years of experience and have successfully launched hundreds of customized Drupal sites around the world, both in Drupal 6 and 7.

With more than 300 Drupal developers and a strategic partnership with Acquia, Ci&T aims to meet customers' evolving global IT usage and development demands by leveraging the best of the Drupal.

A truly global company, Ci&T is headquartered in Campinas, Brazil, with strategic locations across Latin America, North America, Europe and Asia. Ci&T's US headquarters is in King of Prussia, PA. Ci&T's delivery centers are located in Brazil, Argentina and China, offering nearshore outsourcing services to clients around the globe, including Citibank, Coca-Cola, Dow Jones, Johnson & Johnson, MetLife, Panasonic, Pfizer, Saint-Gobain and Yahoo! For more information, visit www.ciant.com.

Drupal contributions

gpanini, ScrumMaster
On Drupal.org for 4 years 12 weeks

frubim, Chief Technology Architect
On Drupal.org for 4 years 11 weeks

rafaelpedrosanto, ScrumMaster
On Drupal.org for 4 years 8 weeks

luiz290788, Software developer
On Drupal.org for 4 years 2 days

handrus, Software Architect
On Drupal.org for 3 years 44 weeks

ker, PHP/Drupal Developer
On Drupal.org for 3 years 20 weeks

ttakamoto, Software Architect
Why use Cache?

- Faster response time will increase the user experience
- Optimizing / rewriting code is often expensive or not an option
- Lower data center costs
- Reduce the CMS Workflow overhead
Why use Cache?
Cache Options

Drupal Modules/Distros
- Page Cache
- Boost Cache
- Authenthicated Cache
- PressFlow

Other Cache Options
- Varnish
- APC
- CDN (Akamai, Amazon CloudFront, Level 3)
- Memcache
Page Cache

- Out of the box Drupal page caching mechanism
- Saves rendered pages directly into the database
- By default only caches anonymous requests
- Hook_boot and hook_exit are always called – unless you’re using the aggressive cache option
Memcache

- High-performance, distributed memory object caching system
- Reduces database hits
- Cached data is stored in web server memory
- It works awesomely if the database is in a different server
Memcache
Memcache – Install

High-level steps to install memcache

1. Install memcached daemon
2. Install and configure PHP memcache extension
3. Install and enable memcache Drupal module
4. Configure Drupal to use memcache
Boost

- Module solution that provides static page caching

- Significant performance improvement for unauthenticated traffic

- Supports shared, VPS and dedicated hosting

- Requires cron & clean urls to work properly

- Have to edit .htaccess and robots.txt files
Authenticated User Page Caching (Authcache)

- Most popular solution for authenticated caching

- Caches final rendered HTML per role and makes AJAX calls to get user specific content

- Ability to use different types of caching (database, memcache, etc...)
Authenticated User Page Caching (Authcache)

Non-Cached Page Request

- HTTP GET Request
- PHP/Drupal Bootstrap (FULL)
- Connect to database & perform queries
- Render HTML
- gzip (compress) HTML
- Time: ~200ms*
- Display HTML on browser

Cached Page Request

- HTTP GET Request
- PHP/Drupal Bootstrap (EARLY_PAGE_CACHE)
- Retrieve cached gzipped HTML
- Time: 1-2ms*
- Display HTML on browser
  - Modify HTML?
  - Perform Ajax?
  - Send JSON back to browser
- PHP/Drupal Bootstrap (SESSION)
- Connect to database & perform queries

* Millisecond times are for demonstrative purposes only. Actual times depend on webserver setup and page complexity.

http://drupal.org/project/authcache
Authcache Disadvantages

- Requires changes on how user-customized content is displayed

- Increases development time and testing

- Not suited for heavy user-centric content

- Hooks for contributed modules may need to be added for compatibility
- Drupal distribution with integrated performance, scalability and availability enhancements

- Drupal 6 and Drupal 7 versions

- Provides integration with Varnish and allows for MySQL replication

- The key enhancements from Pressflow 6 were added to Drupal 7.
I DON'T ALWAYS USE CACHE

BUT WHEN I DO, I USE VARNISH
Varnish

- HTTP reverse proxy solution
- Default caching solution used at Acquia servers
- Focused exclusively on HTTP, unlike other proxy servers that often support FTP, SMTP, etc
- Supports **load balancing**
CDN – Content Delivery Network

- Geographically distributed servers

- Delivers the fastest possible download for all users based on location

- Focused on files (resources) referenced by the HTML, such as CSS, JavaScript, fonts and images

- **Free CDNs:** CloudFlare, Coral Content Distribution Network

- **Commercial CDNs:** Akamai Technologies, Amazon CloudFront, Level 3 Communications, etc
What if we use them all...

PREPARE YOURSELF

THIS CACHING IS GETTING SERIOUS
Front-end Improvements

• Sprites

• Compressing images

• Minify and aggregate JS and CSS files

• Using tool that will help you to analyze and improve front-end performance (YSlow, Web Page Analyzer, Firebug, etc)
Benchmarking / Load Test

- **Load testing tools**
  - Loadrunner
  - Jmeter
  - Etc...

- **Analyze results**
  - Response Time
  - Throughput
Load Test – CDN & Page Cache Example

50s
BEFORE

13s
AFTER

0 to 500 Vusers

0 to 500 Vusers
Load Test – CDN & Page Cache Example

**BEFORE**

4,500,000

**AFTER**

9,500,000
Problems / Risks

- Authenticated users

- Caching what’s not supposed to be cached
  - Dynamic/random content
  - Personalized content
  - Etc...

- Actual content and cache can be out of sync
What to do when nothing works...

CACHE

Y U NO WORK
Troubleshooting

- Clear cache (Drupal, Varnish, CDN, etc...) and try again
- Force a new request using random query strings (?asdasd)
- Disable cache for authenticated users
- Use helper tools/modules/modes (debug mode, memcache interface, etc...)
Questions?
Obrigado!