Acquia™

Performance for Site Builders

Erik Webb
@erikwebb
Senior Technical Consultant
Acquia

Professional Services

Acquia™

Agenda

- ✓ Introduction
- ✓ Evaluating Modules
- ✓ What to Look For
- ✓ Types of Caching
- ✓ Configuring Drupal
- ✓ Performance-related Tools

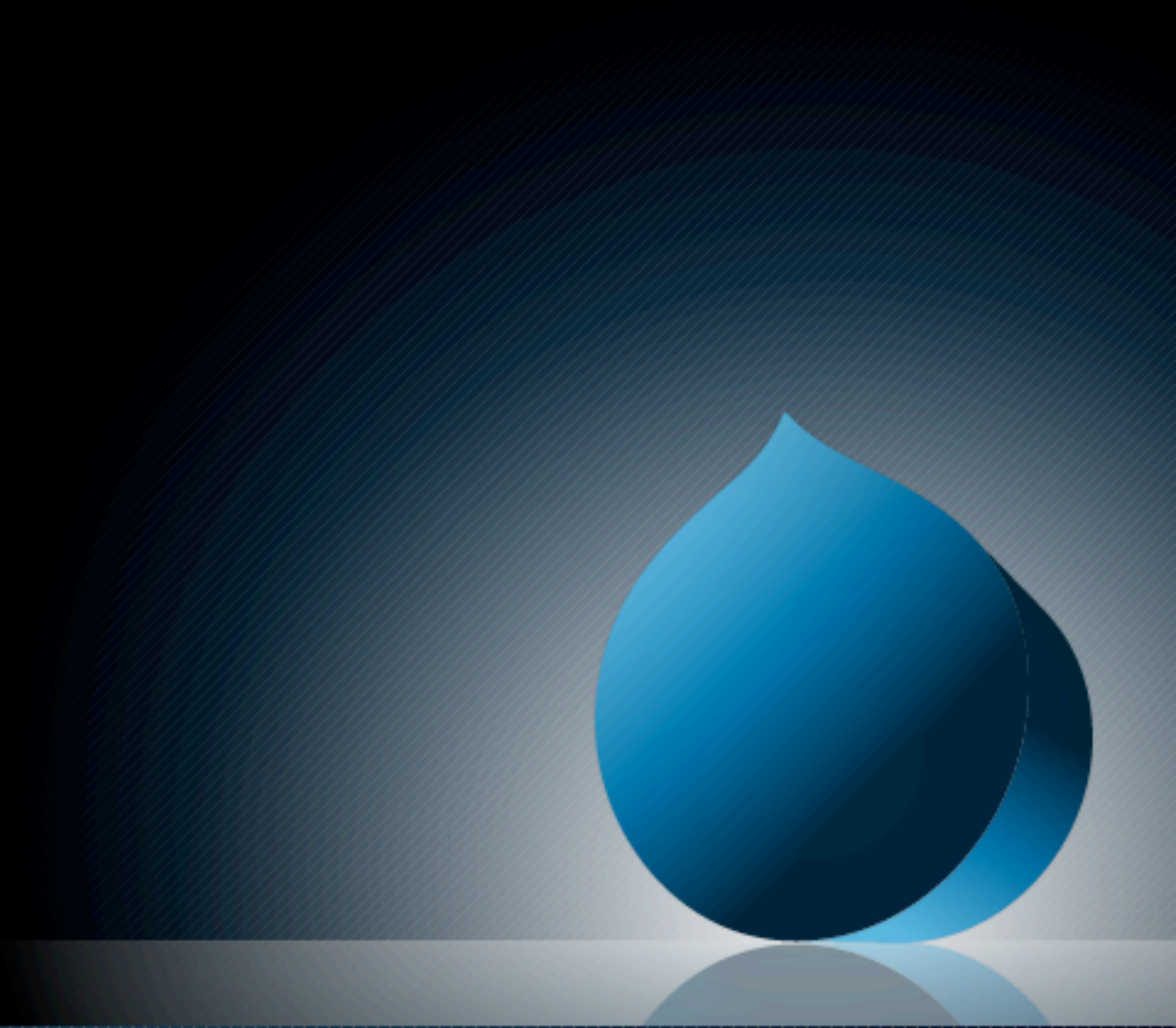
About Me

- ✓ Senior Technical Consultant
- ✓ Focus on Performance, Infrastructure, and Scalability

- ✓ 5+ years with Drupal
- ✓ 10+ years with LAMP
  - ✓ Red Hat Certified Engineer

- ✓ Worked previously at Georgia Tech, IBM

We’re hiring!



Bad Performance Advice

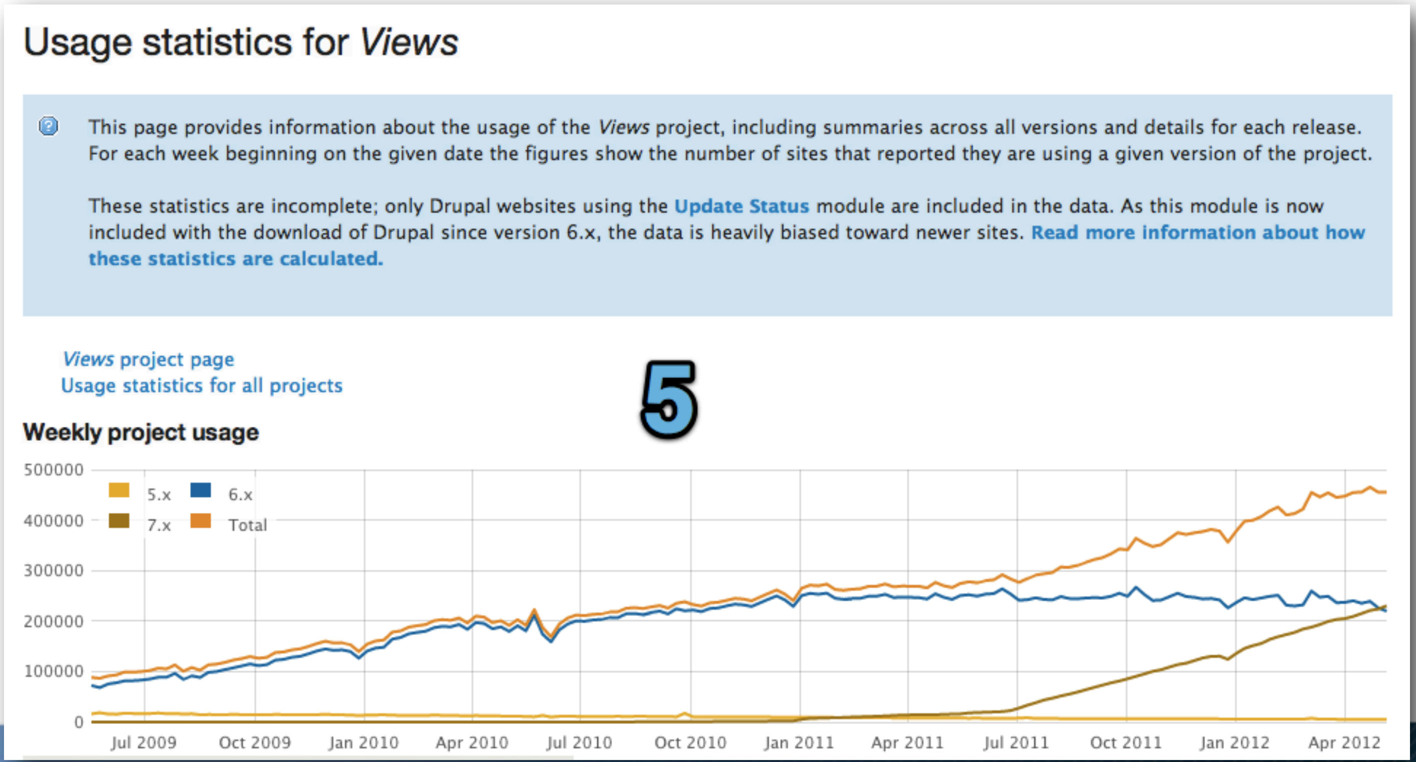
- ✓ *Drupal is slow.*
- ✓ *If it runs out of memory, give it more.*
- ✓ *Don’t use CCK/Views/Panels/whatever.*
- ✓ *If you don’t install X, your site will be slow.*
- ✓ *You need multiple servers.*
  - ✓ *You should have MySQL slave servers.*
- ✓ *Varnish will solve all of your problems.**

http://www.diesoft.net/blog/2012/04/10/first-aid-drupal

Evaluating Modules

General Evaluation

1. Supported version(s)
2. Maintainer reputation
3. Total usage
4. Number of open issues
5. Usage change over time



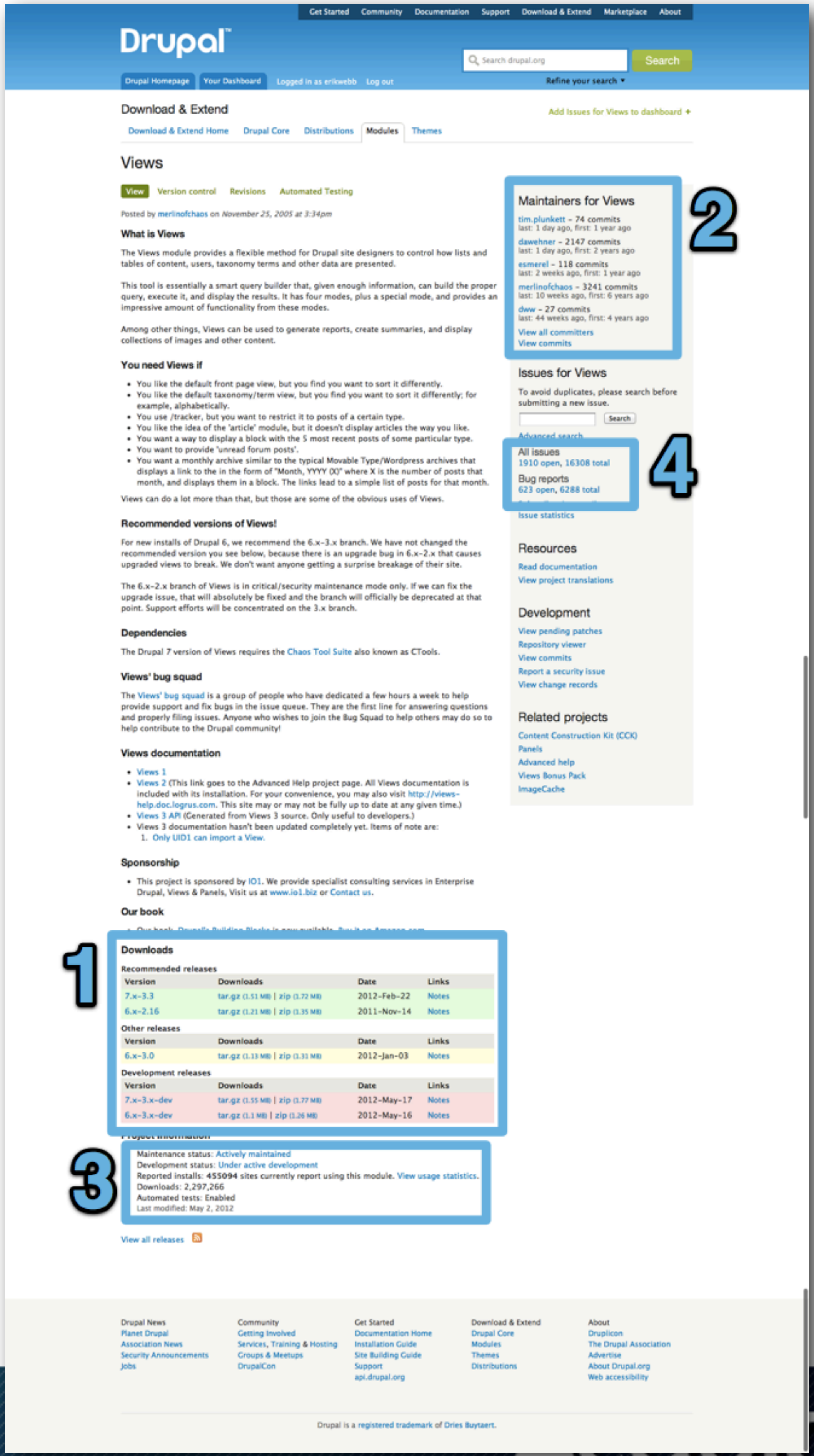
Usage statistics for *Views*

This page provides information about the usage of the *Views* project, including summaries across all versions and details for each release. For each week beginning on the given date the figures show the number of sites that reported they are using a given version of the project.

These statistics are incomplete; only Drupal websites using the **Update Status** module are included in the data. As this module is now included with the download of Drupal since version 6.x, the data is heavily biased toward newer sites. **Read more information about how these statistics are calculated.**

Views project page
Usage statistics for all projects

Weekly project usage



Download & Extend

Views

Posted by merlinofchaos on *November 25, 2005 at 3:34pm*

What is Views

The Views module provides a flexible method for Drupal site designers to control how lists and tables of content, users, taxonomy terms and other data are presented.

This tool is essentially a smart query builder that, given enough information, can build the proper query, execute it, and display the results. It has four modes, plus a special mode, and provides an impressive amount of functionality from these modes.

Among other things, Views can be used to generate reports, create summaries, and display collections of images and other content.

You need Views if

- You like the default front page view, but you find you want to sort it differently.
- You like the default taxonomy/term view, but you find you want to sort it differently; for example, alphabetically.
- You use /tracker, but you want to restrict it to posts of a certain type.
- You like the idea of the 'article' module, but it doesn't display articles the way you like.
- You want a way to display a block with the 5 most recent posts of some particular type.
- You want to provide 'unread forum posts'.
- You want a monthly archive similar to the typical Movable Type/Wordpress archives that displays a link to the in the form of "Month, YYYY (X)" where X is the number of posts that month, and displays them in a block. The links lead to a simple list of posts for that month.

Views can do a lot more than that, but those are some of the obvious uses of Views.

Recommended versions of Views!

For new installs of Drupal 6, we recommend the 6.x-3.x branch. We have not changed the recommended version you see below, because there is an upgrade bug in 6.x-2.x that causes upgraded views to break. We don't want anyone getting a surprise breakage of their site.

The 6.x-2.x branch of Views is in critical/security maintenance mode only. If we can fix the upgrade issue, that will absolutely be fixed and the branch will officially be deprecated at that point. Support efforts will be concentrated on the 3.x branch.

Dependencies

The Drupal 7 version of Views requires the Chaos Tool Suite also known as CTools.

Views' bug squad

The Views' bug squad is a group of people who have dedicated a few hours a week to help provide support and fix bugs in the issue queue. They are the first line for answering questions and properly filing issues. Anyone who wishes to join the Bug Squad to help others may do so to help contribute to the Drupal community!

Views documentation

- Views 1
- Views 2 (This link goes to the Advanced Help project page. All Views documentation is included with its installation. For your convenience, you may also visit http://views-help.doc.logrus.com. This site may or may not be fully up to date at any given time.)
- Views 3 API (Generated from Views 3 source. Only useful to developers.)
- Views 3 documentation hasn't been updated completely yet. Items of note are:
  1. Only UID1 can import a View.

Sponsorship

- This project is sponsored by IO1. We provide specialist consulting services in Enterprise Drupal, Views & Panels, Visit us at www.io1.biz or Contact us.

Our book

Downloads

Recommended releases

Version	Downloads	Date	Links
7.x-3.3	tar.gz (1.51 MB) | zip (1.72 MB)	2012-Feb-22	Notes
6.x-2.16	tar.gz (1.21 MB) | zip (1.35 MB)	2011-Nov-14	Notes

Other releases

Version	Downloads	Date	Links
6.x-3.0	tar.gz (1.13 MB) | zip (1.31 MB)	2012-Jan-03	Notes

Development releases

Version	Downloads	Date	Links
7.x-3.x-dev	tar.gz (1.55 MB) | zip (1.77 MB)	2012-May-17	Notes
6.x-3.x-dev	tar.gz (1.1 MB) | zip (1.26 MB)	2012-May-16	Notes

Maintenance status: Actively maintained
Development status: Under active development
Reported installs: **455094** sites currently report using this module. View usage statistics.
Downloads: 2,297,266
Automated tests: Enabled
Last modified: May 2, 2012

View all releases

Maintainers for Views

tim.plunkett - 74 commits
last: 1 day ago, first: 1 year ago
dawehner - 2147 commits
last: 1 day ago, first: 2 years ago
esmerel - 118 commits
last: 2 weeks ago, first: 1 year ago
merlinofchaos - 3241 commits
last: 10 weeks ago, first: 6 years ago
dww - 27 commits
last: 44 weeks ago, first: 4 years ago
View all committers
View commits

Issues for Views

To avoid duplicates, please search before submitting a new issue.

Advanced search

All issues
1910 open, 16308 total
Bug reports
623 open, 6288 total

Issue statistics

Resources

Read documentation
View project translations

Development

View pending patches
Repository viewer
View commits
Report a security issue
View change records

Related projects

Content Construction Kit (CCK)
Panels
Advanced help
Views Bonus Pack
ImageCache

Drupal News	Community	Get Started	Download & Extend	About
Planet Drupal	Getting Involved	Documentation Home	Drupal Core	Druplicon
Association News	Services, Training & Hosting	Installation Guide	Modules	The Drupal Association
Security Announcements	Groups & Meetups	Site Building Guide	Themes	Advertise
Jobs	DrupalCon	Support	Distributions	About Drupal.org
		api.drupal.org		Web accessibility

Drupal is a registered trademark of Dries Buytaert.

Performance Evaluation

- ✓ Record baseline before installation
- ✓ Record usage immediately after installation
- ✓ Use ongoing memory monitoring to correlate

- ✓ Use tag “Performance” in issue queue
  - ✓ Typically improvements
  - ✓ Weeds out “My site is slow” issues
  - ✓ Example: http://drupal.org/project/issues/search/views?issue_tags=Performance

Questions to Ask?

- ✓ When does this module “run”?
  - ✓ Examples: Login, Content update, Periodically/cron
- ✓ How does this module scale?
  - ✓ Examples: Per node, per user, per request
- ✓ What happens if this module fails?
  - ✓ *If this module fails, no user can login.*
  - ✓ *If this module fails, no content will have functioning slideshows.*
- ✓ Does my site care about performance?
  - ✓ Is my site visited entirely by anonymous users?
  - ✓ Is this site internal and low-traffic only?
- ✓ Do I really need this module?

What to Look For

Identifying the Problem

- ✓ When does it occur?
  - ✓ All pages? Anonymous and/or authenticated?
  - ✓ Only when saving content? Only when logging in?
  - ✓ Under heavy load? Random times during the day?
- ✓ When did it start?
  - ✓ Avoid the “it feels faster/slower” problem
  - ✓ Record performance numbers
  - ✓ Maintain release notes (or retain logs)
- ✓ Who is to blame?
  - ✓ Test against regression between features
  - ✓ Take note of any infrastructure changes

Where Problems Occur

- ✓ Page building modules
  - ✓ Views and Panels
- ✓ External web services
  - ✓ User logins
  - ✓ Any 3rd-party integration
- ✓ Overall complexity
  - ✓ Total number of modules
  - ✓ Views within Panels within Panels within...
- ✓ Misconfigured components
  - ✓ Default is uncached (for developers)
  - ✓ Understand what is being cached

Managing Performance

- Keep records of performance over time
  - Be analytical, don't *feel*
  - Note any milestones of activity or feature development
  - Correlate improvements and regressions
- Establish a performance metric
  - Set a level of acceptability
  - *Example: 80% of pages should return in 500ms or render in 3s*
- Adopt a "Definition of Done" (DoD)
  - Agile concept - aspects needing satisfaction before completion
  - Performance is part of QA
- Don't hide behind infrastructure
  - Slow Drupal is cheap, hardware is not

Types of Caches

Application-level Caching

- ✓ *Move along, nothing to see here.*
- ✓ Not configurable
- ✓ Should never result in “staleness”
- ✓ Can only be enhanced by improving backend

- ✓ *Examples: Filter, Menu, Path, Filter (not FORM!)*

Component-level Caching

- ✓ Stores user-facing components
- ✓ Best speedup for authenticated users
  - ✓ Limited effectiveness without more configuration
  - ✓ Mostly disabled by default
- ✓ Varying degrees of contents, HTML to serialized objects
  - ✓ Some implementations more effective than others
- ✓ *Examples: Block, Views, Panels*

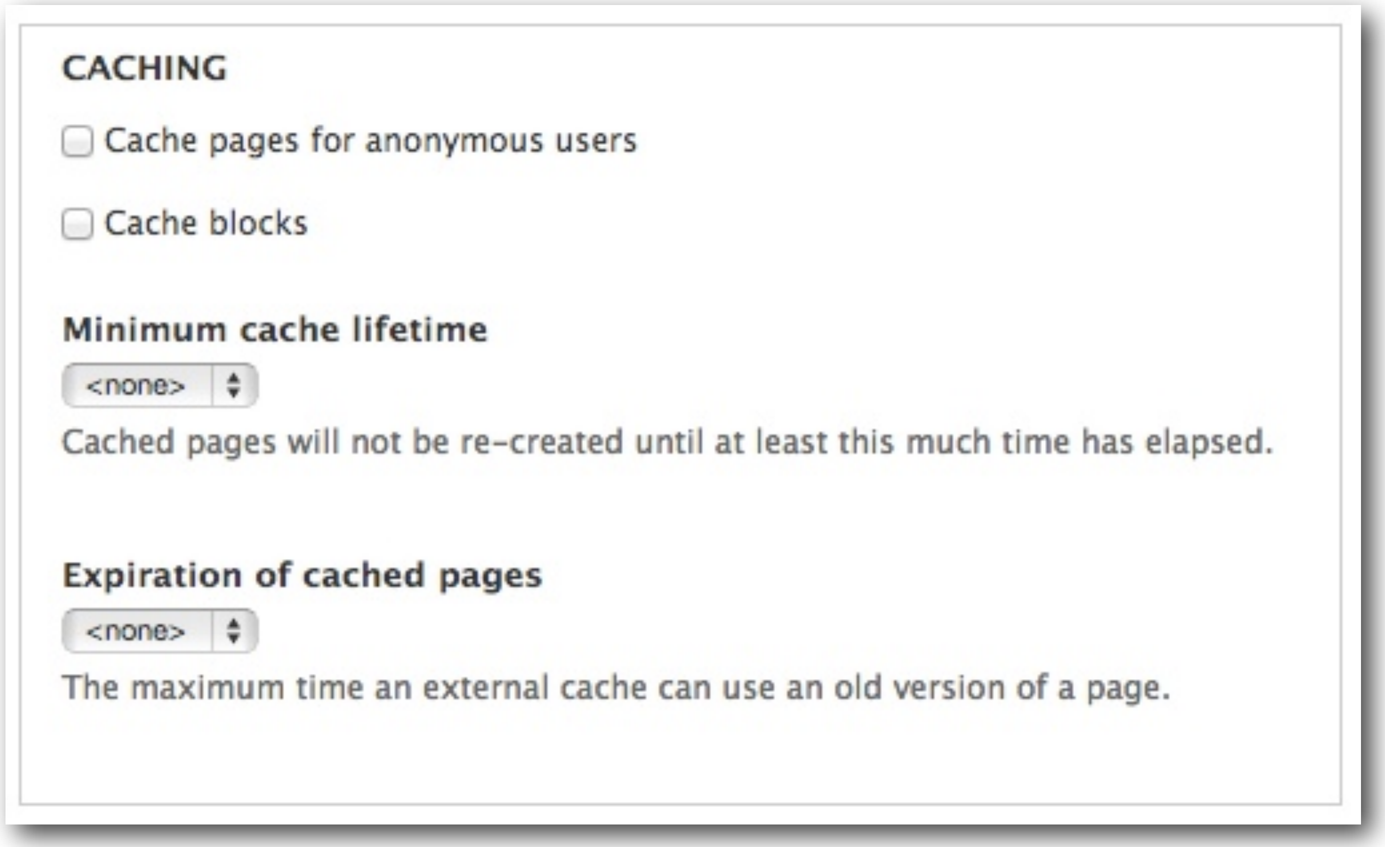
Page-level Caching

- ✓ Most efficient possible cache
  - ✓ Combine with reverse proxy
- ✓ Only applicable for anonymous users*
- ✓ Stored as full HTML
- ✓ page_cache_fastpath() in D6
  - ✓ Not supported by default cache backend
  - ✓ Bypasses database connection and full bootstrap

Configuring Drupal

Performance page

- ✓ Use block caching!*
  - ✓ Disabled by node access
  - ✓ Biggest speedup for auth users
- ✓ Content changes clear block and page cache
  - ✓ Use “Minimum cache lifetime”
- ✓ Using a reverse proxy?
  - ✓ Use “Expiration of cached pages”
- ✓ Aggregation/compression only on production
  - ✓ $conf[‘preprocess_css’] = 1;



CACHING

☐ Cache pages for anonymous users

☐ Cache blocks

Minimum cache lifetime

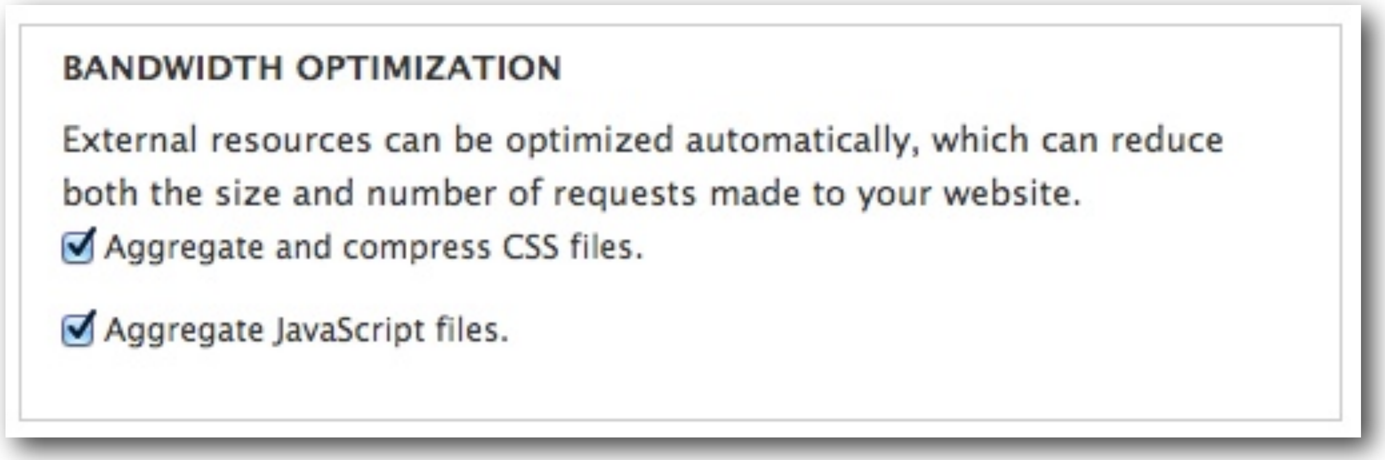
<none>

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

Expiration of cached pages

<none>

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



BANDWIDTH OPTIMIZATION

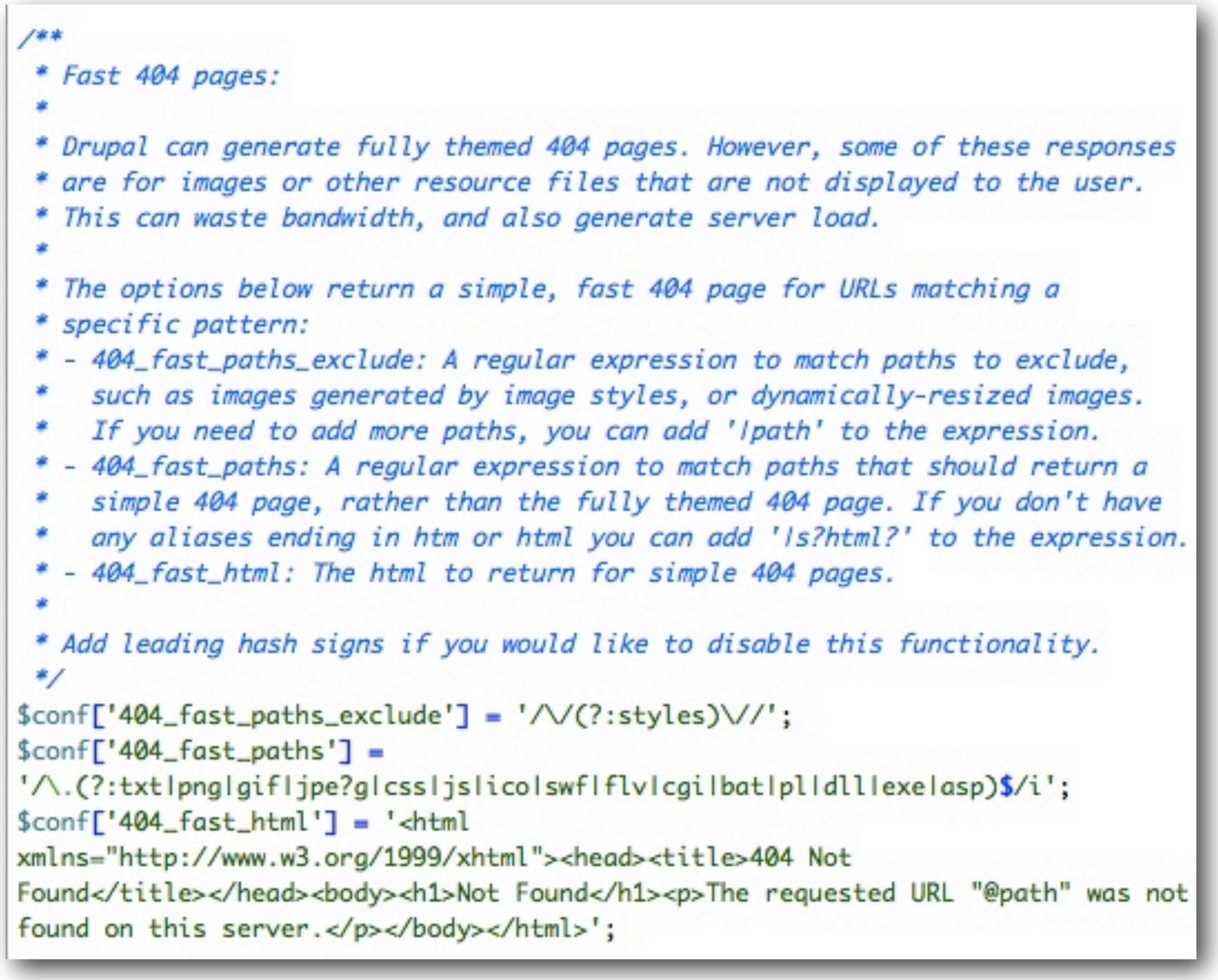
External resources can be optimized automatically, which can reduce both the size and number of requests made to your website.

☑ Aggregate and compress CSS files.

☑ Aggregate JavaScript files.

Fast 404

- ✓ Added in Drupal 7.9 (currently being backported to D6)
  - ✓ See http://drupal.org/node/76824
- ✓ Configured in settings.php
- ✓ Avoid performance hit from 404 errors



```
/**
 * Fast 404 pages:
 *
 * Drupal can generate fully themed 404 pages. However, some of these responses
 * are for images or other resource files that are not displayed to the user.
 * This can waste bandwidth, and also generate server load.
 *
 * The options below return a simple, fast 404 page for URLs matching a
 * specific pattern:
 * - 404_fast_paths_exclude: A regular expression to match paths to exclude,
 *   such as images generated by image styles, or dynamically-resized images.
 *   If you need to add more paths, you can add '|path' to the expression.
 * - 404_fast_paths: A regular expression to match paths that should return a
 *   simple 404 page, rather than the fully themed 404 page. If you don't have
 *   any aliases ending in htm or html you can add '|s?html?' to the expression.
 * - 404_fast_html: The html to return for simple 404 pages.
 *
 * Add leading hash signs if you would like to disable this functionality.
 */
$conf['404_fast_paths_exclude'] = '/\/(?:styles)\//';
$conf['404_fast_paths'] =
'/\.(?:txt|png|gif|jpe?g|css|js|ico|swf|flv|cgi|bat|pl|dll|exe|asp)$/i';
$conf['404_fast_html'] = '<html
xmlns="http://www.w3.org/1999/xhtml"><head><title>404 Not
Found</title></head><body><h1>Not Found</h1><p>The requested URL "@path" was not
found on this server.</p></body></html>';
```

Other Notes

- ✓ Understand what Drupal does and does not cache
  - ✓ Helps understand when to troubleshoot
- ✓ Don’t forget the frontend!
- ✓ Do not enable “UI modules” on production
  - ✓ Unneeded memory usage
  - ✓ Examples: Field UI, Rules Admin, Views UI
- ✓ Avoid Database Logging (if you have an alternative)
  - ✓ Examples: Syslog, log4php
- ✓ Unnoticed PHP errors slow down execution
  - ✓ Increase PHP logging on non-production environments

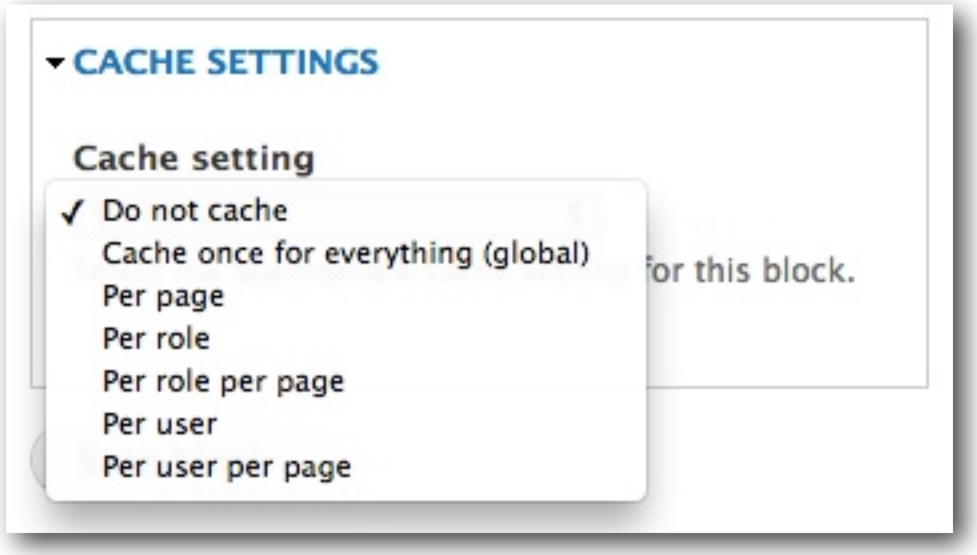
Performance-related Tools

Drupal Modules

- ✓ Devel
  - ✓ Execution time and memory usage
  - ✓ Query logging
- ✓ Boost
  - ✓ Flat file page caching
  - ✓ Designed for shared hosting (infrastructure neutral)
- ✓ Memcache
  - ✓ Replace database caching with Memcached
  - ✓ In-memory cache, reduces DB load

Drupal Modules

- ✓ Entity Cache
  - ✓ Drupal 7 only
  - ✓ Stores created objects a.k.a. “entities” (users, nodes, comments, etc.)
- ✓ Path Cache
  - ✓ Pressflow (D6) or Drupal 7
- ✓ Block Cache Alter
  - ✓ Maximize effectiveness of block caching
  - ✓ Fine-grained control per block



CACHE SETTINGS

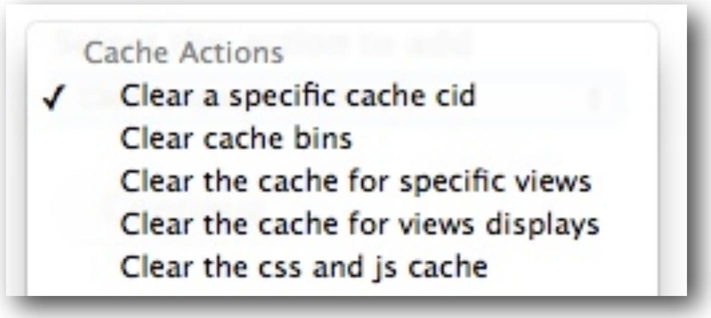
Cache setting

- ✓ Do not cache
- Cache once for everything (global)
- Per page
- Per role
- Per role per page
- Per user
- Per user per page

for this block.

Drupal Modules

- ✓ Views Litepager
  - ✓ Slow pagers on Views with large DB tables
- ✓ Views Content Cache
  - ✓ Store saved Views based on content changes rather than expiration
  - ✓ Example: Clear a View display when a new "Article" node is created
- ✓ Cache Actions
  - ✓ More generalized approach than Views Content Cache
  - ✓ Works with Drupal cache, CSS/JS aggregation, Views, and Panels
  - ✓ Requires the Rules module


Cache Actions
✓ Clear a specific cache cid
Clear cache bins
Clear the cache for specific views
Clear the cache for views displays
Clear the css and js cache

3rd-Party Tools

- ✓ Web optimization tools
  - ✓ Yahoo! Smush.it
  - ✓ SpriteMe
- ✓ Web testing tools
  - ✓ WebPagetest.org
  - ✓ Google PageSpeed Online
- ✓ Browser-based
  - ✓ Firebug/Web Inspector
  - ✓ YSlow!
  - ✓ Google PageSpeed
- ✓ SaaS products
  - ✓ New Relic
  - ✓ Yottaa

SpriteMe

New Relic

yottaa

Infrastructure Overview

Apache/Web Server

- ✓ Handles web requests for PHP
- ✓ Most common bottleneck
  - ✓ Application should be memory-bound
  - ✓ Least performance considerations
- ✓ Serves static files alongside PHP scripts
- ✓ Scalable: Horizontal and vertical
- ✓ Alternative: Nginx

PHP/Application “Server”

- ✓ Usually runs as apart of Apache (mod_php)
  - ✓ Most common configuration by far
- ✓ Use Alternative PHP Cache (APC)
  - ✓ Saves interpreted PHP files in memory
- ✓ Can run as separate process - PHP-FPM (5.3.3+)
  - ✓ Scale independent of Apache
  - ✓ Better privilege separation

MySQL/Database Server

- ✓ Sole datastore for Drupal
- ✓ "Natural" LAMP bottleneck
  - ✓ Hard to solve problem
- ✓ Most tunable component

- ✓ Scalable: Vertical
- ✓ Alternatives: Percona Server and MariaDB

Caching Server

- Two main advantages
  - Faster access than MySQL
  - Reduce overall load on MySQL
- Significant for authenticated users
- Easily configured through Drupal or PHP
- Requires PHP extensions

- Scalable: Horizontal and vertical
- Examples: Memcached, Redis

Varnish/Reverse Proxy

- ✓ Store entire pages for quick retrieval
- ✓ Extremely configurable
  - ✓ Load balancing and traffic management
  - ✓ Varnish Configuration Language (VCL)
- ✓ Scalable: Horizontal* and vertical

Questions?

Where to find me?

erikwebb.net
@erikwebb on Twitter
erikwebb on LinkedIn
erikwebb on SlideShare