

BIORAFT

ENTERPRISE SAFETY, COMPLIANCE & TRAINING SOFTWARE

High Performance Drupal



A Panel Discussion

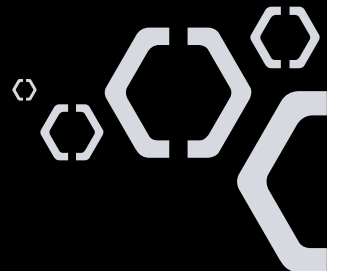
Panelists

- Erik Peterson (eporama)
- Seth Cohn (sethcohn)
- Micky Metts (freescholar)
- Patrick Corbett (pcorbett)
- Michelle Lauer (miche)



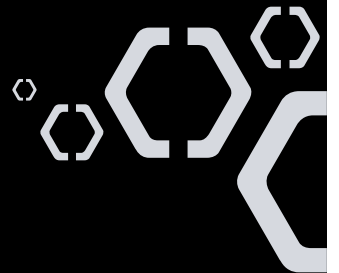
Overview

- 14 Rules
- Establish a baseline
- Set goals
- Make a list of areas to tackle



14 Rules by Steve Souders

1. Make Fewer HTTP Requests
2. Use a Content Delivery Network
3. Add an Expires Header
4. Gzip Components
5. Put Stylesheets at the Top
6. Put Scripts at the Bottom
7. Avoid CSS Expressions



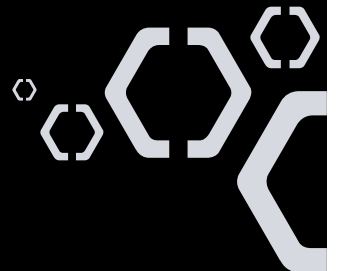
14 Rules by Steve Souders

8. Make JavaScript and CSS External
9. Reduce DNS Lookups
10. Minify JavaScript
11. Avoid Redirects
12. Remove Duplicate Scripts
13. Configure Etags
14. Make AJAX Cacheable



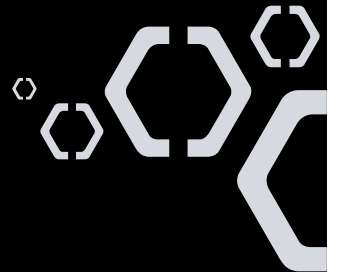
Establish a baseline - Tools

- **Google Analytics** (generates detailed statistics about a website's traffic)
- **Google PageSpeed** (generates a report about a website's page speed)
- **Yslow** (analyzes web pages and why they're slow based on Yahoo!'s rules for high performance web sites)



Establish a baseline - Tools

- **Xdebug** (a PHP extension which provides debugging and profiling capabilities)
- **XHProf** (reports memory usage, CPU times and number of calls for each function)



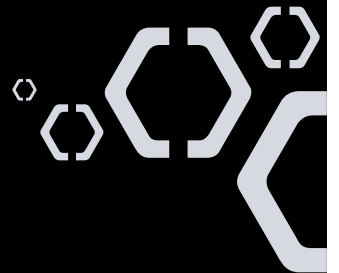
Establish a baseline - Tools

- **Devel module** (displays queries, count and time)
- **Performance module** (provides performance statistics logging for a site, such as page generation times, and memory usage, for each page load.)



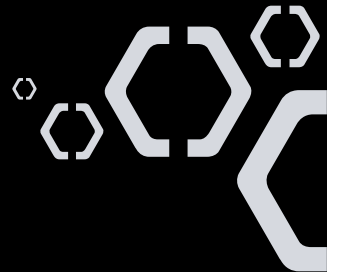
Establish a baseline - Analyze

- What do you analyze?
 - The front page
 - Each content type
 - Each View
 - High traffic/popular pages
 - Slowest pages
- Who do you masquerade as?
 - Anonymous
 - Different roles



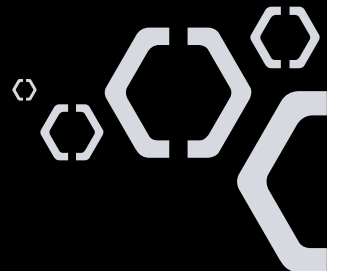
Set goals

- What is considered average/normal?
- What are your minimum requirements?
 - Different for anonymous vs logged in users?



Areas to tackle

- Drupal configurations
- Module development
- Caching & swappable storage
- Reverse proxy caching
- MySQL



Drupal Configurations - Performance

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.

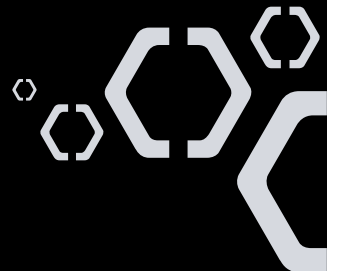


Drupal Configurations - Cron

Run cron every

3 hours ▼

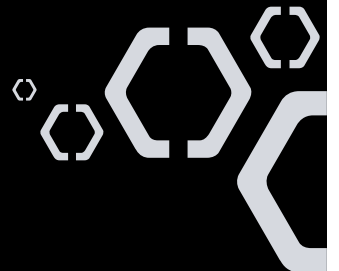
More information about setting up scheduled tasks can be found by [reading the cron tutorial on drupal.org](http://drupal.org).



Drupal Configurations – Image Styles

STYLE NAME	SETTINGS	OPERATIONS
Thumbnail (100x100)	Default	edit
Medium (220x220)	Default	edit
Large (480x480)	Default	edit

Picture Module – Backported from Core D8 to Contrib D7
Delivers alternate image sources based on device capabilities to prevent wasted bandwidth and optimize display for both screen and print.



Drupal Configurations - Views

Query results

The length of time raw query results should be cached.

Rendered output

The length of time rendered HTML output should be cached.



Drupal Configurations – Devel Settings

Never should be enabled on a production site.

Rebuild the theme registry on every page load

While creating new templates and theme_ overrides the theme registry needs to be rebuilt.

Display page timer

Display page execution time in the query log box.

Display memory usage

Display how much memory is used to generate the current page. This will show memory usage when `devel_init()` is called and when `devel_exit()` is called.

Display redirection page

When a module executes `drupal_goto()`, the query log and other developer information is lost. Enabling this setting presents an intermediate page to developers so that the log can be examined before continuing to the destination page.

Display \$page array

Display \$page array from `hook_page_alter()` in the messages area of each page.



Drupal Configurations - dblog.module (aka watchdog)

▼ FILTER LOG MESSAGES

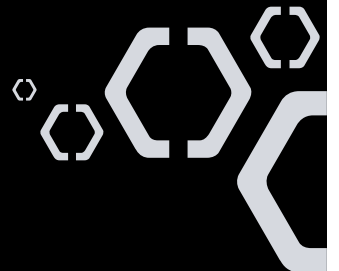
Type

actions
content
cron
php
system
user

Severity

emergency
alert
critical
error
warning
notice
info
debug

Filter



Drupal Configurations - syslog.module

Switching to *syslog* allows messages to be logged by the operating system rather than the database.

Error messages to display

- None
- Errors and warnings
- All messages

It is recommended that sites running on production environments do not display any errors.

Database log messages to keep

1000

The maximum number of messages to keep in the database log. Requires a [cron maintenance task](#).



Drupal Configuration – File System

Public file system path



A local file system path where public files will be stored. This directory must exist and be writable by Drupal. This directory must be relative to the Drupal installation directory and be accessible over the web.

Private file system path

An existing local file system path for storing private files. It should be writable by Drupal and not accessible over the web. See the online handbook for [more information about securing private files](#).

Temporary directory

A local file system path where temporary files will be stored. This directory should not be accessible over the web.

Default download method

Public local files served by the webserver.

This setting is used as the preferred download method. The use of public files is more efficient, but does not provide any access control.



Module Development – CSS/JS

Included files load on every page

```
example.info  
name = example  
description = "Example module"  
core = 7.x  
scripts[] = js/example.js
```



Module Development – CSS/JS

Check context before including

```
/**
 * Implements hook_node_view().
 */
function example_node_view($node, $view_mode,
$langcode) {
  if ($view_mode == 'full') {
    $path = drupal_get_path('module', 'example') .
      '/js/example.js';
    $node->content['foo']['#attached']['js'][$path]
      = array('every_page' => TRUE);
  }
}
```



Module Development - Querying & Loading Entities

Old way:

```
$query = db_query("
  SELECT nid
  FROM {node}
  WHERE status = 1
");
$mynodes = array();
while ($result = db_fetch_array($query)) {
  $mynodes[] = entity_view(
    entity_load($result['nid']));
}
```



Module Development - Querying & Loading Entities

Better way: Database agnostic so can query alternative storages. Calls single query using "IN" rather than looping.

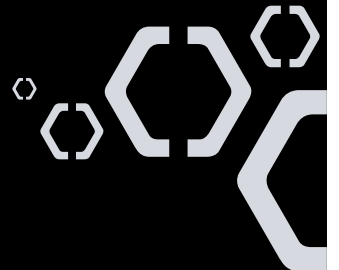
```
$query = new EntityFieldQuery();  
$query->entityCondition  
  ('entity_type', 'node');  
$result = $query->execute();  
$entities =  
  entity_load_multiple($result);  
return  
  entity_view_multiple($entities);
```



Module Development – Static Caching

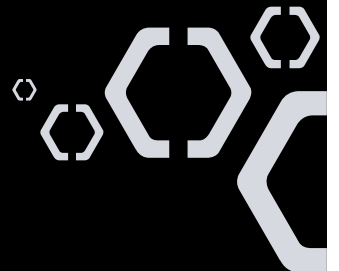
For the life of a page (ex: display node content in 3 places)

```
function my_module_function($reset =
FALSE) {
    static $my_data;
    if (!isset($my_data) || $reset) {
        // Do your expensive calculations
        // here, and populate $my_data
    }
    return $my_data;
}
```



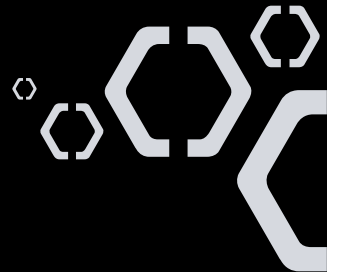
Module Development – Static Caching

- `drupal_static()`
Returns an empty value the first time called, but any changes to the variable will be preserved when the function is called again. That means that our function can check if the variable is already populated, and return it immediately without doing any more work.
- `drupal_static_reset()`
When modules need absolutely fresh data, they can call `drupal_static_reset()` to clear out any temporarily cached information.



Module Development – Static Caching

```
function my_module_function() {
    $my_data =
        &drupal_static(__FUNCTION__);
    if (!isset($my_data)) {
        // Do your expensive
        // calculations here, and
        // populate $my_data
    }
    return $my_data;
}
```



Module Development – Persistent Caching

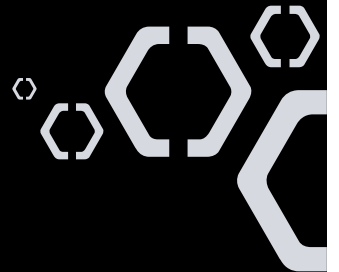
Saves in the database or works with alternative caching systems.

- `cache_set($cid, $data, $bin = 'cache', $expire = CACHE_PERMANENT)`
- `cache_get($cid, $bin = 'cache')`
- `cache_clear_all($cid = NULL, $table = NULL, $wildcard = FALSE)`



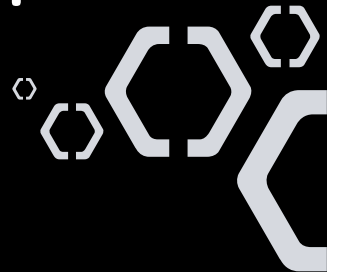
Module Development – Persistent Caching

```
function my_module_function() {
  $my_data = &drupal_static(__FUNCTION__);
  if (!isset($my_data)) {
    if ($cache = cache_get('my_module_data')) {
      $my_data = $cache->data;
    }
    else {
      // Do your expensive calculations here, and populate
      // $my_data
      cache_set('my_module_data', $my_data, 'cache');
    }
  }
  return $my_data;
}
```



Module Development – `variable_set()`

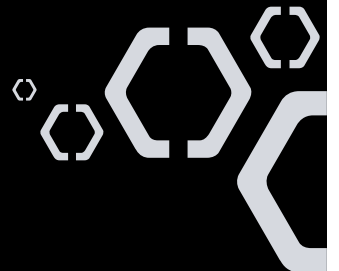
- “If `variable_set()` ends up being called very frequently, this can result in either a ‘cache stampede’ where dozens of requests are constantly updating the cache entry, or a ‘lock stampede’, where processes are constantly waiting for a new variable cache entry that may be invalidated again before they’re able to retrieve it from the cache.”



Caching & Swappable Storage

Drupal's Cache API uses the database storage implementation by default. Consider the following contrib projects:

- Memcache API and Integration (memcache)
- Redis (redis)
- MongoDB (mongodb)
- APC – Alternative PHP Cache (apc)
- File Cache (filecache)
- Boost (boost)



Caching & Swappable Storage

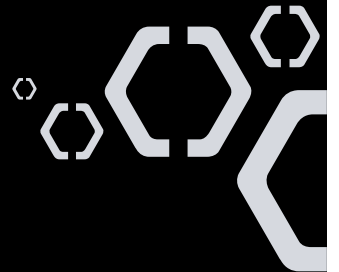
- Overview of caching options:
<https://drupal.org/node/326504>
- Comparison table of caching options:
<https://groups.drupal.org/node/21897>



Reverse Proxy Caching

Serve the entire page request without having to call back to Apache and PHP.

- Varnish
- Nginx



MySQL

- Use InnoDB vs MyISAM
- Increasing `max_connections` “only allows a server to thrash itself to death more effectively”
 - Instead, find and fix slow queries
- `innodb_buffer_size`: Set to between 1/2 and 2/3 your memory space
- `innodb_log_file_size/innodb_log_files`: Increase to 256MB for frequently updated sites



Conclusion

- Establish a baseline
- Set Goals
- Fix/Tweak
 - Drupal Configurations
 - Module Development
 - Swappable Storage
 - Reverse Proxy Cache
 - MySQL



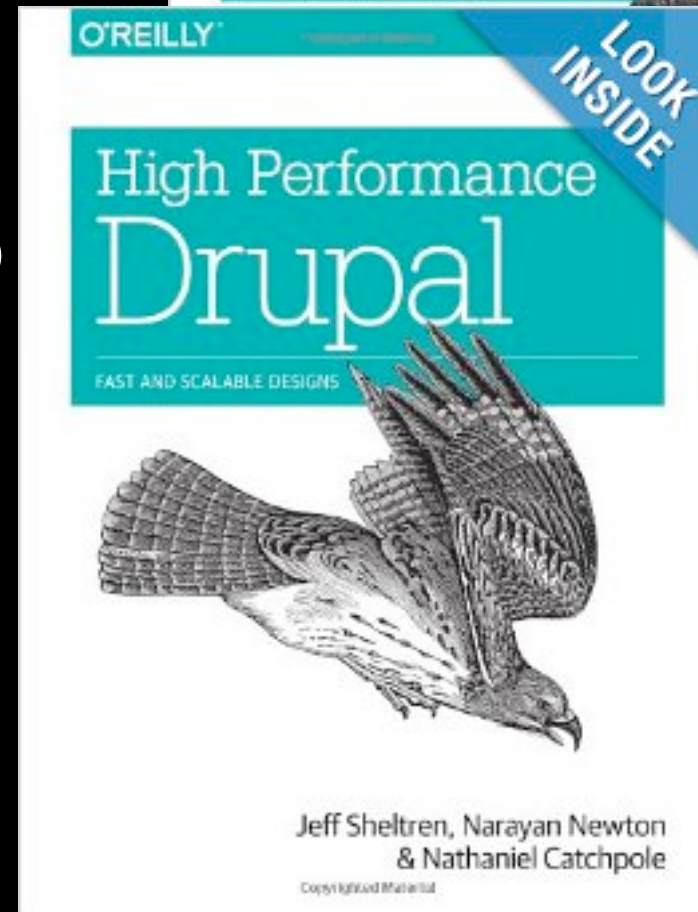
Books

High Performance Drupal

Jeff Sheltren,
Narayan Newton,
Nathaniel Catchpole

High Performance Web

Steve Souders



Thanks!

- Thank you to all of our panelists!
- Get well soon Chris Wells!

BICORRAFT

