WordCamp Austin 2014

Cluster Fudge

Recipes for Running WordPress
in the Cloud

Grant K Norwood

@grantnorwood
#clusterfudge
Slides and code at: bit.ly/cluster-fudge

- *(Talk about the keynote …)*
- **Thank you** all for coming to this talk today, **I'm excited to share** a few things with you that I've learned while running my favorite CMS in **cloud-hosted environments**.
- **My name is Grant Norwood**, and I make websites and other things for the Michael & Susan Dell Foundation.
- **This talk has been a couple years in the making**, though I only finished my slides last night at midnight :)
- **My goal is to share** with you the **lessons I've learned**, the **technologies I use**, and **tips and tricks** that I hope will help you guys get started and **ultimately succeed** with your own websites in clustered, cloud-based environments.
- **Please raise your hands if you have questions!** But let's not derail too much. We can always take things offline after the session.

Are we talking about food or functions()?

- Both!
- This talk is about the tools we need to scale WordPress horizontally.
- Why do we need to deploy our sites to clustered environments?
  - Performance
  - Redundancy & fault tolerance
  - Modern WordPress sites are full-blown web applications!
- 3 different recipes, using the same available ingredients, from food trailer to fancy, to suit everybody's tastes & budgets.

- **Food is a great metaphor** for everything!
- **Whether you are in a public or private cloud environment**, I think this will be helpful, and you'll learn something new that you can use.
- **3 different recipes** from the same available ingredients.
- **The more familiar we become with these ingredients**, the more we know how to use them for any recipe we need to make in the future.
- I'll talk about how **a local Austin chef, a legend-in-the-making, approaches food**, and how it is **analogous to how we might approach building cloud-based solutions** for our WordPress sites.

6 minutes

It takes about as long to respond to your website going down as it does to boil water.

Let me begin by sharing a story: 6 Minutes of Downtime

- When I was forced to remind myself that I needed to **explicitly design for performance**, and **test that performance**, and determine the **realistic load** that my environment could handle.
- It takes as long to respond to your site being down as it does to boil water.
- Over a year ago, **when Michael Dell shared the link** to our 2013 Founders' Letter on Twitter, Facebook, Google+
  - **Which tells the stories of the outcomes** and impact of our foundation's recent work …
  - With **personal messages** from Michael & Susan, and our Executive Director …
  - He **drove hundreds of visitors** to our site in just seconds, and **thousands within the next 20 minutes**.
- **msdf.org** went down for **6 minutes!**
- Single server with 4gb memory, 4 vCPUs, web & db, W3 Total Cache in disk caching mode (no SSD drives!).

- *(Click)* Michael Dell's has **just over 1 million followers**, plus people who share, re-post and retweet, etc.
  - 344k - Google+
  - 624k - Twitter
  - 55k - Facebook
- This configuration handled the **long-tail component** of those shares very well, and **in the past had always handled the traffic** thrown at it.
- It was the burst of more concurrent visitors than we'd ever had that broke our site.
- **One comment stood out to me**, and I can imagine that I looked like I had just seen a ghost. *(Click)*

Turning down the heat …

30,000ms - Time for URL monitoring alerts to trigger

60,000ms - Time for me to read my text message and call my hosting provider

60,000ms - Time for my support team to diagnose the problem

180,000ms - Time to kill connections and reboot

Our one-server configuration could scale up, but not out.

That's a problem!

- **Our pot was boiling over**, and I needed to turn down the heat on our server.
- **I get notified by SMS and email** when sites go down or respond slowly.
- Using **ms** because it makes the impact sound greater, because it was a big deal.

Ingredients

- **Reeling from this personal failure**, I decided I needed to build a new clustered environment.
  - Specifically designed for **performance**, for **redundancy**.
  - A solution to scale out more when needed, and scale back when not peak.
- **I had this list of ingredients to choose from**, which could be combined to create a recipe/solution that fit my application perfectly.
- We will **talk a little about each of these** and how they support the requirements of your website & environment, **from simpler environments to more complex solutions** with lots of moving parts.
- Working with **web servers**, like Apache & nginx, **databases** like MySQL, and the other **technologies that help accelerate & synchronize your site** across clustered servers.

Paul Qui

Born in Manila, Philippines, moved to Virginia at 10 years old. Texas Culinary Academy in 2003.

Began his career working for free at Uchi, rising to be executive chef at Uchiko. Owner/founder of East Side King and Qui Austin.

Winner of Top Chef 2009, James Beard Award 2012

A modern approach to food, with no boundaries and accessible to all, from his trailers to his brick and mortars.

- We can **approach creating various recipes for clustered solutions similar to how Paul Qui approaches food**.
  - Let me tell you a little about Paul Qui …
- As a culinary student, **took himself to Uchi night after night** to study and enjoy the cuisine.
- **Tyson Cole**: "Hey, a guy just quit, you wanna work here?"
- **First job as a cook was at Uchi**, at the lowest station doing tempura frying.
  - **Kept rising**: to making rice and sushi, to becoming the sous-chef, then chef de cuisine.
  - When Uchiko, a sister restaurant of Uchi, opened **Qui was named the executive chef**.
- Out of the Uchi family came **East Side King**, a food trailer Qui set up with two other Uchi cooks.
  - Food trailers are almost a staple here in Austin, because they are **accessible and affordable**.
- He later opened his own restaurant bearing his name, Qui. Offering more complex combinations of flavors and ingredients, but still affordable.
- Qui has been at the root of restaurants that range from food trailers, to fancy but affordable, to more upscale like Uchi & Uchiko.
- **His food can be enjoyed by everybody, no matter their taste or the size of their wallet.**

— (http://www.austin360.com/news/entertainment/dining/uchikos-key-player-1/nRtqP/)

Whatcha want for dinner?

East Side King

A quick, cheap, and easy meal from a nearby trailer.

You're basically just load-balancing the technologies you already know.

Qui

More complex flavors, but still affordable.

Add more layers, more caching, more performance, but requires more management.

Uchiko

Treat yourself.
You get what you pay for.

Customize everything, or have the chef make his favorite dish for you.

- All 3 of these are enhanced hosting configurations, more than normal single-server, or shared hosting environments.

East Side King

Good starter solution, with more performance and redundancy than a single server.

Ingredients

Linux	*APC*
Apache	*lsyncd*
MySQL	*memcached*
PHP	*W3 Total Cache*

Prepare your LAMP stack. Stir in APC, lsync, and W3 Total Cache configured for Memcached. Add SSD drives or loads of memory to taste.

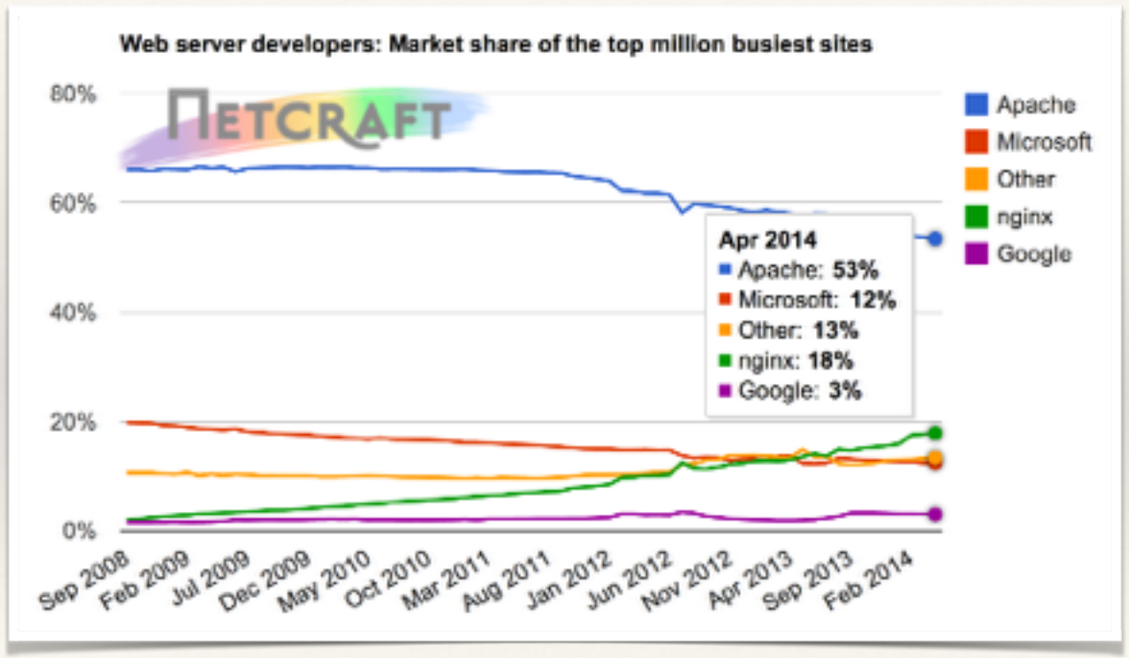
LAMP Stack

- Linux
  - **Tip:** Choose LTS releases to make patching multiple servers faster & easier. Supported for 5 years, versus 9 months for regular release.
  - Do you want to be a server administrator or a developer? Maintaining your clustered environment can take away from time spent building new things.
- Apache
  - Apache web server began in early 1995 after work on the NCSA HTTPd server code stalled.
  - The most popular HTTP server in use since April 1996.
  - Easy to administer with lots of documentation, and high compatibility with other software.
- MySQL
- PHP

http://httpd.apache.org/ABOUT_APACHE.html

- Linux
  - **Choose LTS releases** to make patching multiple servers faster & easier. **Supported for 5 years, versus 9 months** for regular release.
  - **Story:** When patching my various servers for the Heartbleed vulnerability, I had just a couple servers that were on non-LTS versions of Ubuntu.
    - One older version was not vulnerable (running Ubuntu 10.10).
    - The other ran Ubuntu 13.04, which was no longer receiving security updates since January. I had to upgrade the entire OS (*do-release-upgrade*) versus the simpler *apt-get update && apt-get upgrade*.
  - Ubuntu's latest 14.04 LTS just released last week, 12.04 LTS still supported through April 2017

Apache HTTP Server

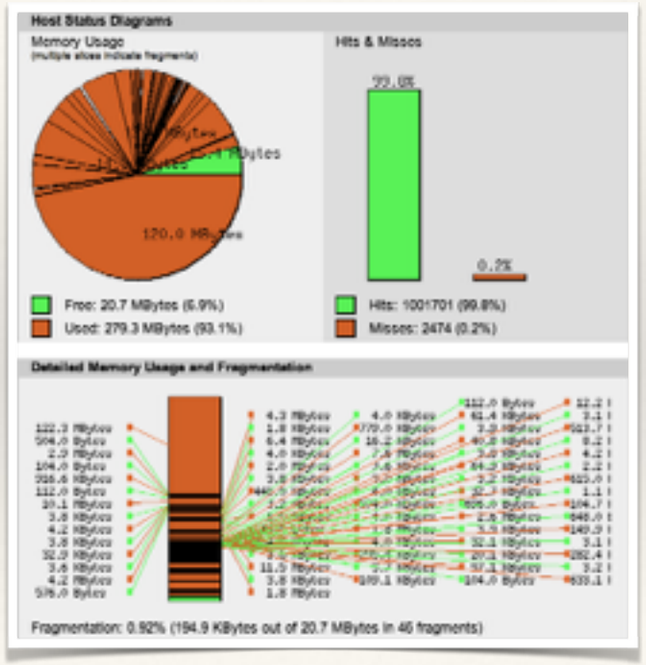


http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html

- Powers over half (53%) of the top 1 million busiest sites.
- Nginx is rising!
  - We'll take a look at that in our next solution.

APC

- The Alternative PHP Cache (APC) is a free and open opcode cache for PHP.
- Caches and optimizes compiled intermediate code so your application runs 3x-4x faster.
- With PHP 5.5, Zend open sourced their cache offering, under the new name of Zend OpCache — it is now included with PHP itself, in addition to being available for older PHP versions going all the way back to PHP 5.2!
- **Tip:** Zend OpCache will likely be the recommendation going forward.



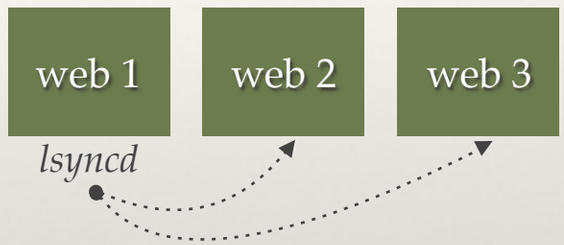
https://www.activecollab.com/blog/3-zend-opcache-memcached.html

- As **Zend OpCache is included in the PHP core**, it will have to be maintained with every new PHP release.
  - **Should be no more weird problems caused by incompatibilities between a new PHP release and the opcode cache.**
  - Development of APC has slowed down and, at least for now, APC is not compatible with PHP 5.5, and not stable for PHP 5.4.
- **Zend OpCache will likely be the recommended path going forward.**

- https://www.activecollab.com/blog/3-zend-opcache-memcached.html

lsyncd

- Lsyncd watches a local directory trees event monitor interface (inotify or fsevents).
- Aggregates and combines events for a few seconds and then spawns one (or more) process(es) to synchronize the changes.
- Config file uses Lua syntax.
- Use SSH keys to allow the master server to log into slaves securely & without a password.



https://github.com/axkibe/lsyncd

You have your database and your files. And there are a number of different types of files:

- Core
- Plugins
- Theme code
- User-generated

We need a way to sync these files between web nodes in the cluster.

Lua language allows easy setting of config options, as well as more complicated logic within configuration files.

lsyncd

lsyncd-sample.lua

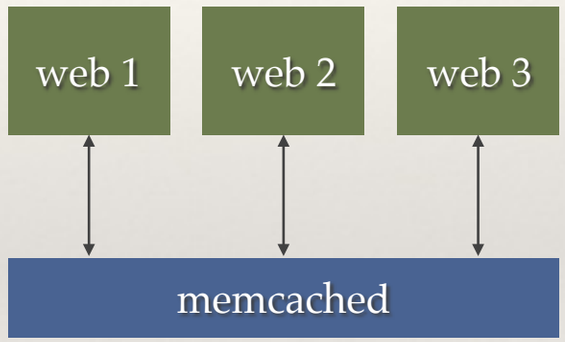
- **Tip:** Set your directory structure up where all vhosts are synced from a single root, or add as many sync blocks as you need.
- **Tip:** Consider excluding directories with temp files.
- **Tip:** Use private IP addresses to improve speed by leveraging the private network.

https://github.com/grantnorwood/cluster-fudge

- Set logging and status files, as well as the interval in settings block.
- Each target (or slave) requires a sync block.
- Each sync block has options for:
  - Type of sync (rsync, direct copy, bash, etc)
  - Source and target directories
  - Rsync options for compression, file permissions, and such.

memcached

- Short-term memory for your WordPress website. W3 Total Cache can leverage memcached for caching:
  - Pages
  - Database query results
  - Objects / Transients
- **Tip:** Disable PHP sessions if you can. If you must have PHP sessions, use memcached for fast, shared session handling.
- **Tip:** As with lsyncd, use private IP addresses to speed things up.
- Supports auto-expiration of values.



https://github.com/grantnorwood/cluster-fudge

- **PHP sessions can be slow**, and the session cookies can cause problems for caching, especially if those cookies are present with static files.
- **We can use Varnish to strip cookies from static files** and other certain requests, but we'll cover that later on.

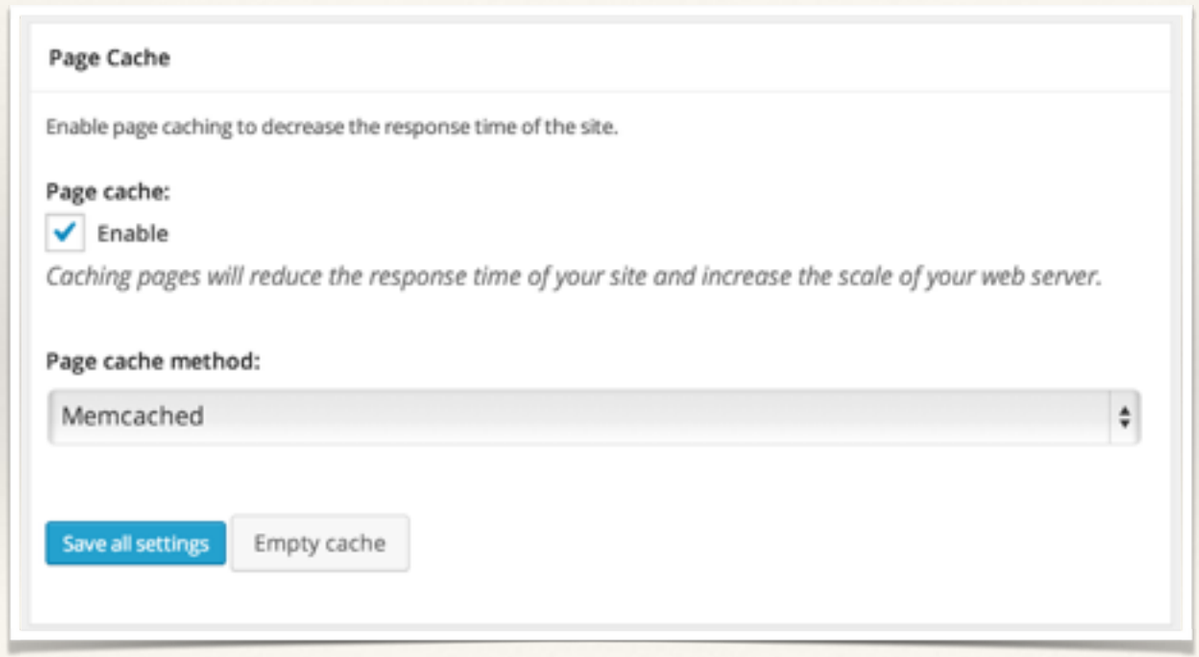
W3 Total Cache

- Author Frederick Townes is a forum ninja, and provides fantastic support.
- Supports many types of cache storage.
- **Tip:** Try to avoid caching to disk, even if you're on SSDs. It can get wonky on clustered sites.
- **Tip:** Use memcached for fast, shared memory storage of pages, queries, and objects.
- **Tip:** Use the built-in compatibility check to ensure things will work correctly.

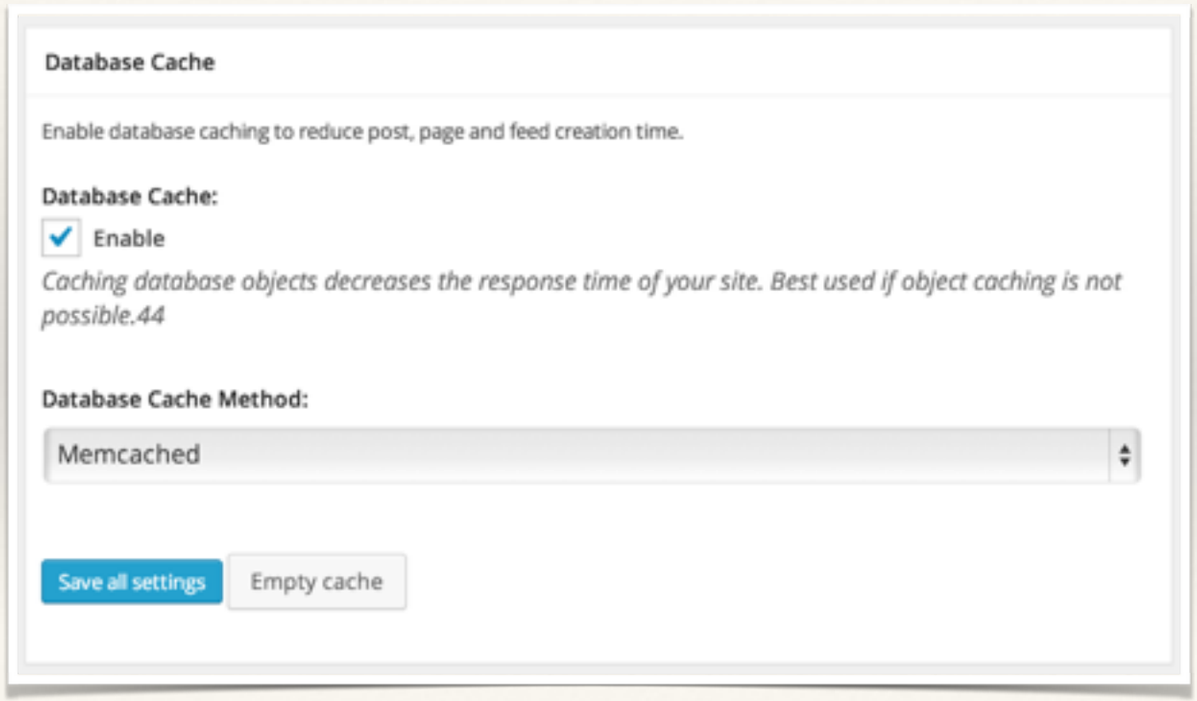
http://www.w3-edge.com/weblog/2013/09/w3-total-cache-pro/

- I choose W3 Total Cache because of the flexibility, the support, and frankly because I see others (like yoast.com) use it.
- The Pro version gives you fragment and transient caching, but I'm unsure of the price.
- Types of caching:
  - Disk-based (slowest, not for clustered servers)
  - APC
  - Memcached

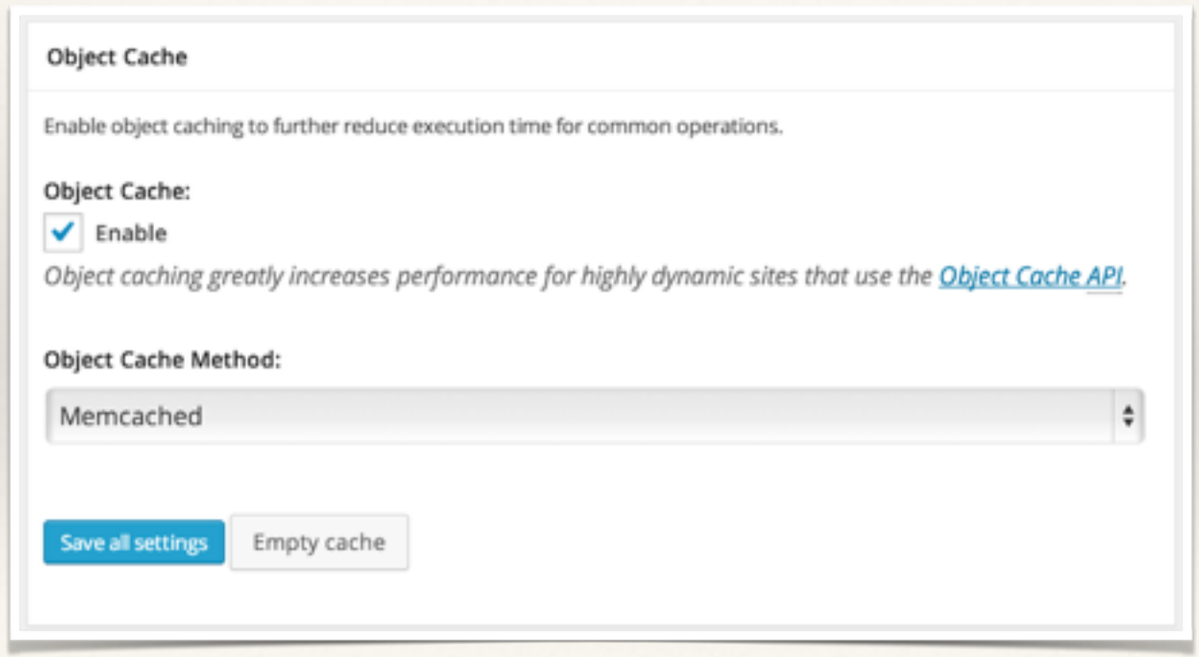
W3 Total Cache



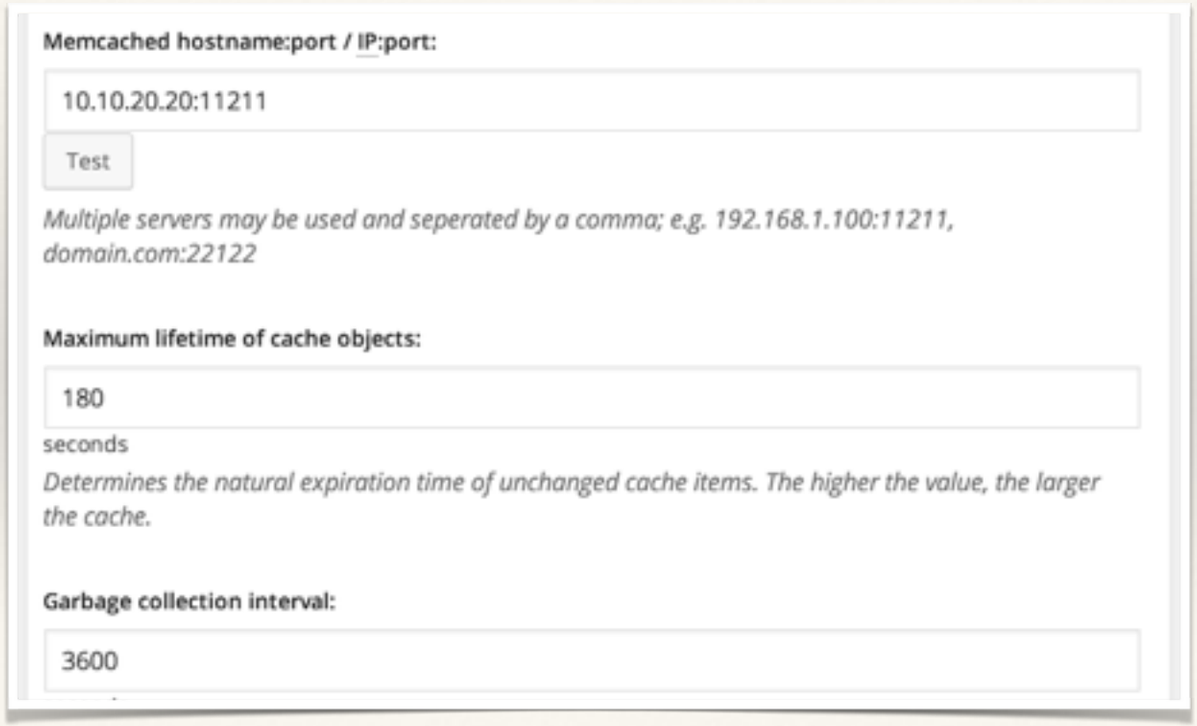
W3 Total Cache



W3 Total Cache

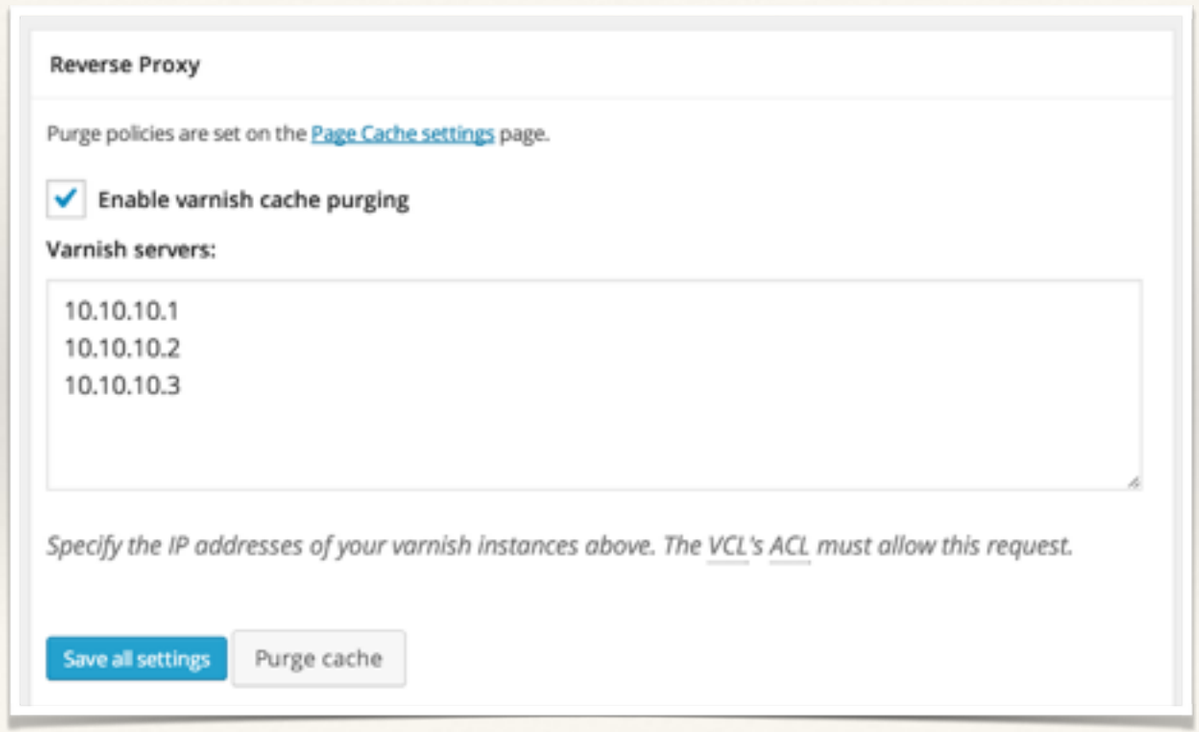


W3 Total Cache



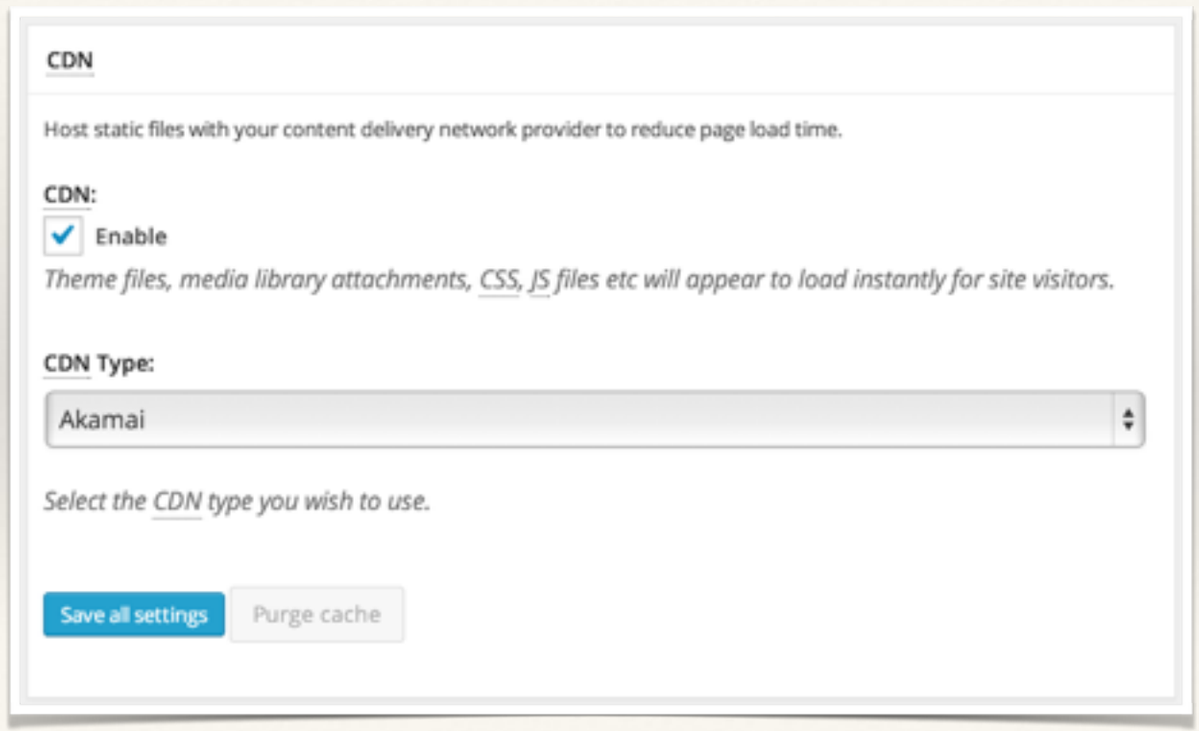
- **Set the memcached host or IP and port** for the page cache, database cache, and object cache.

W3 Total Cache

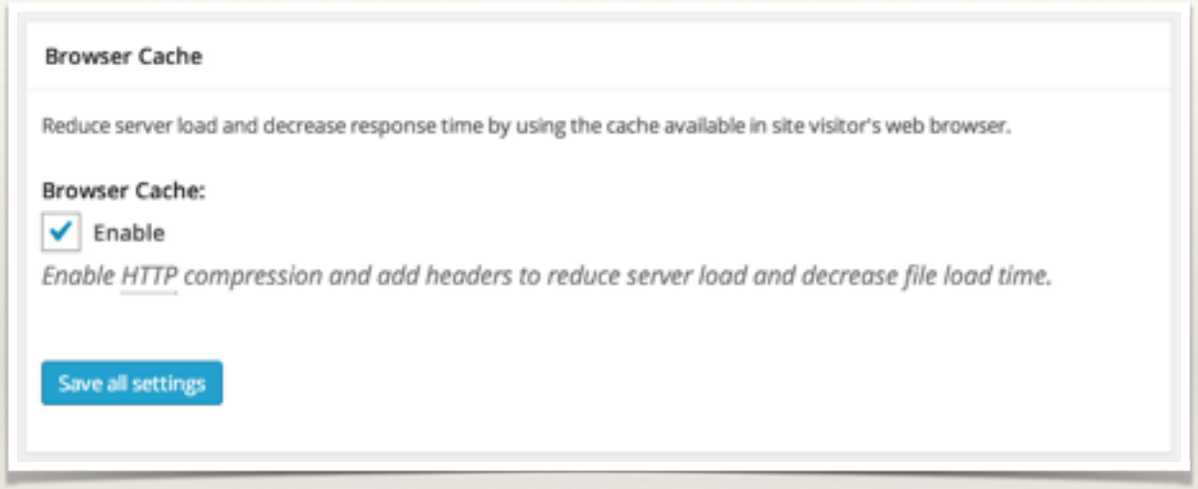


- **Varnish sits in front of your web server as a reverse-proxy**, so WordPress may not have the chance to serve the latest dynamic content.
- **W3TC can purge single pages in Varnish**, or even purge the entire site.
- We'll talk more about Varnish in the next solution.

W3 Total Cache



W3 Total Cache



- **Tip:**
  - Set far-future expiration for static files (1 day, 1 week, 1 month, 20 years …)
  - Set far lower expiration for your dynamic content (30 secs, 5 mins, 1 hour)

- Lots of settings for setting cache-control, pragma, expires, e-tag, and other headers.
- Every application is different, depending on the type of content, and how often it is refreshed/updated.

Qui

More gooey layers, more complex flavors, more robustness at an affordable price.

Ingredients

Linux	*APC*
nginx	*lsyncd*
MySQL	*memcached*
PHP	*W3 Total Cache*
	Varnish

The LEMP stack is an improvement on the LAMP stack in many regards, with some minor sacrifices in compatibility. More layers means troubleshooting can be more difficult.

- Improvement in performance with a more efficient web server, and a layer of front-end caching in front of the web server.

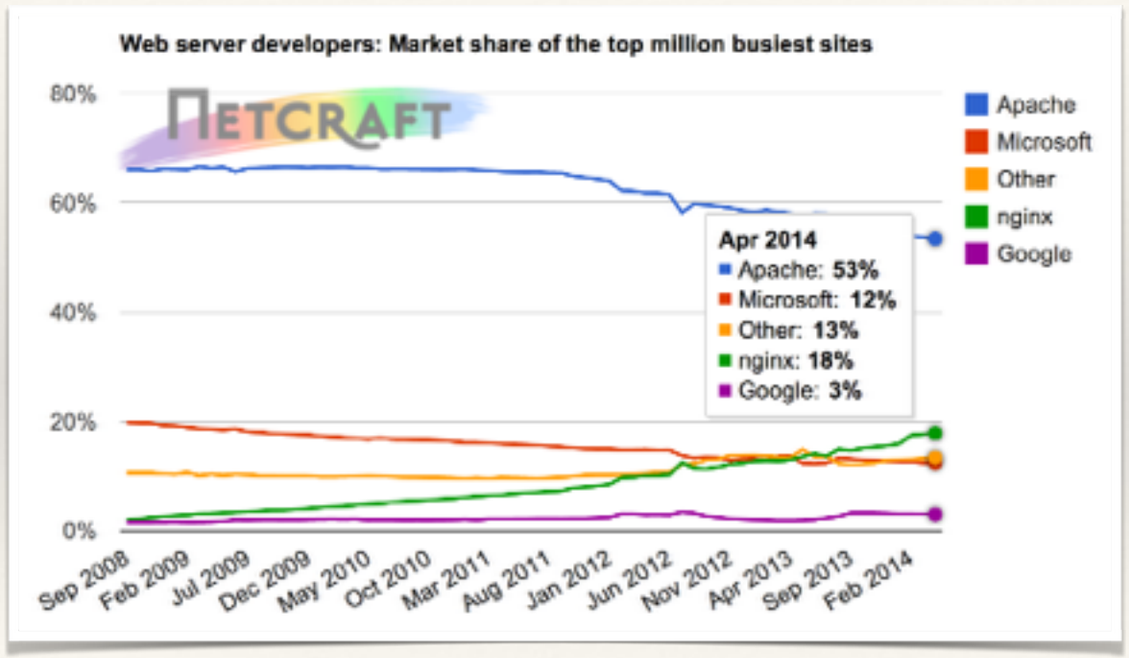
nginx

- About
  - Uses event-driven architecture instead of process-based architecture to make more efficient use of resources.
  - Asynchronous == non-blocking
  - Can be an HTTP server or reverse-proxy. (Reverse-proxying to Apache can get a bit spicy for my taste …)
- Compared to Apache
  - nginx does not create a process for every new request, Apache does.
  - nginx serves static files faster than Apache because of lower memory overhead.
  - nginx can be tougher to configure, and documentation is not as prevalent as the available docs for Apache.

http://www.thegeekstuff.com/2013/11/nginx-vs-apache/

- Can be an HTTP server or reverse-proxy.
- But reverse-proxies can get a bit too complex …
  - **For example, using nginx as a reverse-proxy for Apache may not guarantee .htaccess rules are followed**, especially if nginx or Varnish is caching pages and redirects before Apache can handle them.

nginx



http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html

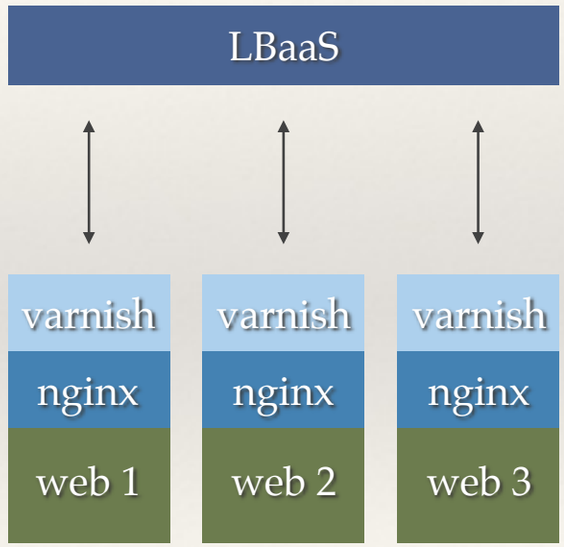
- nginx is powering 18% of the top million busiest sites and growing steadily!

Varnish

A web application accelerator also known as a caching HTTP reverse proxy.

Typically speeds up delivery with a factor of 300 - 1000x, depending on your architecture.

Put Varnish in front of your web server and configure it to cache the HTTP content.



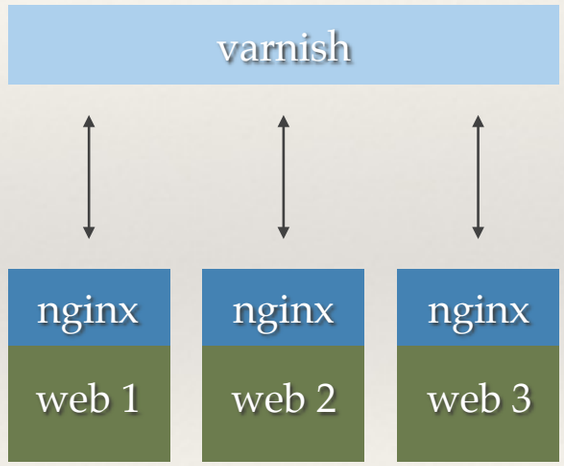
https://www.varnish-cache.org/

- Here, a LBaaS performs load-balancing, and each web node has Varnish installed in front of nginx.

Varnish

Or, how about simply using Varnish as the load-balancer.

Leveraging a dedicated LB takes a bit more effort to manage, but will help insulate you from the rare attacks on shared resources that ruin it for everybody.



https://www.varnish-cache.org/

- **Story:** I woke up one morning to multiple text messages coming in rapidly as my URL monitor was showing msdf.org was down. I whipped my laptop out, SSH'd into my servers as I called my hosting provider, and expected to see my servers on fire. But there was no activity, no server load, relatively few active processes.
  - The LBaaS was being attacked by Anonymous because SendGrid, another Rackspace customer who shared our LB, was being DDoS'd pretty good after a social media controversy.
  - After about 20 mins Rackspace had mitigated the attack pretty well.
  - We weren't the target, but another customer who shared our cloud resources were.
  - **The cloud is still shared hosting!**

Varnish

https://www.varnish-software.com/static/book/VCL_Basics.html

- Varnish can be complicated!

Varnish

Best Practices

- Really cache your homepage, it gets the most traffic. (Try stripping cookies!)
- Never cache wp-admin, wp-login, or registration pages. Consider not caching front-end pages when a user is logged in.
- Route all your wp-admin traffic to the master.
- Exclude preview URLs from cache.
- Cache all static assets, strip cookies.
- Do not cache POST requests.
- Exclude certain AJAX requests used for WP comments, login, and registration.
- Sessions (and their cookies) can ruin cacheability!
- You can achieve higher hit rates if you set higher TTLs on objects, and then simply purge dynamic content as soon as it's been updated. (W3 Total Cache!)

https://github.com/grantnorwood/cluster-fudge

- There is a delicate balance between higher hit rates and accommodating for constantly dynamic content.

Uchiko

Often expensive and time-consuming, but you're on the leading edge of clustered WordPress!

Ingredients

Linux	*lsyncd*
nginx	*memcached*
Clustered MySQL	*W3 Total Cache*
PHP	*Varnish*
APC	***???***

Building on the last recipe, we can experiment by swapping out layers and modules with whatever is new and cool. Finally add some database redundancy.

- I've never left a good sushi restaurant in less than an hour, for less than $100!

Clustered MySQL

- Want to setup and maintain your own MySQL cluster? Not me!

Some DBaaS Options

- Amazon RDS
- Rackspace Cloud Databases
- Azure Cloud via ClearDB

Clustering MySQL is most important for highly transactional sites.

SPDY

- HTTP is old! Modern web apps have more requests, more resources, more dynamic content, more personalized content, CDNs, and so much more.
- SPDY aims to correct the challenges of HTTP 1.1
  - Header compression
  - Multiplexing
  - Server push

SPDY

Header compression

- Today, headers are in plain text and uncompressed.
- Headers often include redundant info. The browser agent string never changes, and cookies are updated rarely, but they are sent with every request/response. What a waste!
- HTTP supports compression of the content only.
- In some cases, like small scripts or icons, headers can be as large as, or larger than, the response content!
- SPDY compresses request & response headers to save bytes on the wire.

SPDY

Multiplexing

- Better parallel processing of requests & responses.
- Every request requires a response before the next request can be sent.
  - Head of line blocking: If a response is slow, all other requests are delayed.
  - HTTP 1.1 introduced pipelining, which allows multiple requests to be sent at the same time.
    - Still requires responses to be received in the same order.
- SPDY enables true multiplexing.
  - Priority can be set by the client for server responses, indicating which resources should be returned first.
  - Per host only. If your browser requests resources from 10 different hosts, 10 different connections will still be created.

Multiplexing

- **For example**, primary scripts and stylesheets might be prioritized higher than images, and print stylesheets might be prioritized even lower.

SPDY

Server Push

- HTTP requires that every request/response be initiated from the client.
- With SPDY, when updated content like news articles or sports scores are available, they are pushed to the client over the already established connection.
- “Server Hint” helps to avoid the server pushing files to the client that already exists in the browser’s cache.
- Still an experiment to learn how all this is best used in the real world.

The client must still establish the connection in the first place, but after that the server can push to the client without the client constantly polling the server.

SPDY

Pros & Cons

- SPDY leverages HTTPS (port 443) to be compatible with existing firewalls and network devices.
- **Browser support:** Chrome, FireFox, Opera, Android 4.x, IE11, Amazon Silk - some support different versions of SPDY than others.
- **Server support:** Nginx (built-in, experimental support), Apache has mod_spdy
- HTTP2 is in development and based on SPDY (coming soon!)

The client must still establish the connection in the first place, but after that the server can push to the client without the client constantly polling the server.

While SPDY was developed entirely by Google, HTTP2 will be driven by the larger working group.

HipHop

- Specifically, HHVM (HipHop Virtual Machine)
- Open-sourced by Facebook, 2x to 20x performance increase over PHP-FPM.
- Compiles Hack and PHP into an intermediate bytecode. This bytecode is then translated into x64 machine code dynamically at runtime by a JIT compiler.
- Facebook's production version is completely run on HHVM since Q1 2013.
- FastCGI support for Apache, Nginx, Lighttpd, etc. Used in place of php5-fpm.
- **Bonus:** MySQLi support added/fixed in 3.0.1! WordPress 3.9 contains several fixes specifically for HHVM.
- **Bug:** WordPress cookies not fully working yet, but should be production-ready very soon!
- **Tip:** Requires a 64-bit OS.

http://hhvm.com/
https://kinsta.com/blog/hhvm-and-wordpress/

Get Your Cloud On!

A couple of my favs …

- Rackspace
  - http://developer.rackspace.com/devtrial/
  - Get USD $300 in free cloud services - that's up to USD $50 per month credit for six months on your Rackspace Cloud account.
  - Lots of full-featured, leading edge products!
- Digital Ocean
  - https://www.digitalocean.com
  - Start with $10 credit with promo code: **DEPLOY2DO** *(Expires April 30, 2014)*
  - Really, really cheap cloud hosting!

WordCamp Austin 2014

Thank you.

Slides and code at: bit.ly/cluster-fudge

Grant K Norwood
@grantnorwood
http://grantnorwood.com

Links

Further Reading

http://kaanon.com/blog/work/making-wordpress-shine-varnish-caching-system-part-1

Credits

Creative Commons Images

"Chocolate Fudge Cubed" by ZakVTA

"A Watched Pot..." by Brandon Warren