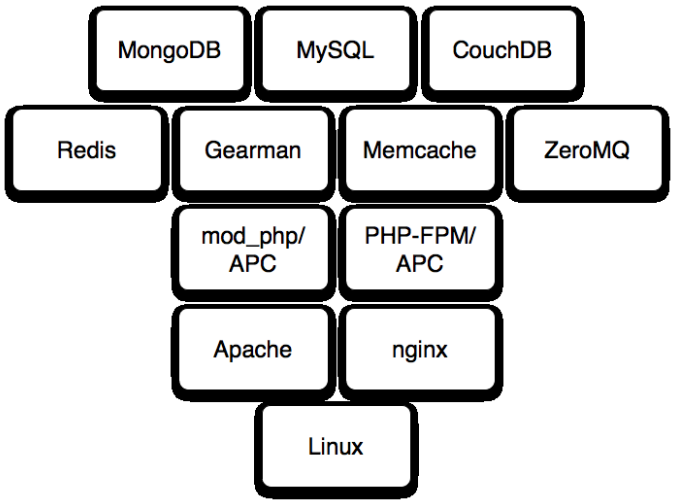
Rasmus Lerdorf

“Brain dump”

The modern LAMP stack:



PHP is the bottleneck, and the web server doesn't really matter – no significant difference betw. nginx/lighthttpd/apache

Lots of ignoring errors – error handling is not optimized in PHP b/c it should be an infrequent event

- .15ms to handle a non-logged E_NOTICE!
- error_reporting = -1

strace

- Look for ENOENT (file doesn't exist UNIX error)
- “Smarty isn't”
- Look for excessive STATs due to insufficient realpath_cache_size (esp. in PHP 5.2)
- Don't do stupid wrapping of PHP functions

Profiling

- C-level (entire stack) – callgrind
- XHDProf
- PHP-level – xdebug
- Includes – pecl_inclued (don't over-do it)
- Static analysis – hiphop (undeclared variables, methods, constants, assigning void return, etc – good complement to unit tests)

Architecture

- Use a different sub-domain for static assets
- Keep cookies small (exceed MTU = multiple TCP packets)

- Out-of-band processing (vs threads) – use gearman
- Don't use a relational db for a storage engine for non-relational data
- Avoid FKs if possible to avoid deadlocks
- Cache your ORM; don't query more than necessary
- Don't overload apache (8 core CPU = 25-30 apache clients; never more than 50)

Continuous Integration (CI)

- Confident that anyone can deploy, anytime
- Unit tests to prevent breakage
- Deploy must be atomic
  - Don't svn up, get pull, tar/cp/scp
  - Use symlink
- Clear caches
- Don't break running requests
- Don't miss requests
- Don't leave site broken
- Log
- Strategy 1:
  - rsync over existing files
  - version static files
  - inode will be different for php files
  - APC will continue serving old cached PHP files, provides buffer
  - Clear cache with apache max_requests_per_child
  - Clear cache with graceful apache restart
- Strategy 2:
  - symlink swap
  - eases rollback
  - store current release in shared memory
  - check current release, if it doesn't match, clear realpath_cache/opcode cache
  - need version-agnostic layer

Q&A

- Events will not be a focus in any future version of PHP, however can make it easier (not hard now; see http://php.net/libevent)
- Build-time tools that do static analysis and class maps – like
- Don't get pull into docroot; use symlink
- Deploy tools: Capistrano (ruby), weploy (PHP)

http://talks.php.net/show/phpcon2011

Drew McLellan

“The Original Hypertext Preprocessor”

Every line of code potentially impacts the user
Functionality is not enough
250 PHP-based CMS systems
“Wordpress learning curve very high – pages vs posts vs custom post types”

“Support will kick your [tail]” - Jason Fried, SxSW
Customers are inexperienced
Each sale – support threshold @ 30 minutes time
Find ways to reduce support requests

Every request should be unique (no FAQs)
Fix areas of confusion rapidly
Support your own software – programmers should see issues firsthand
Massively important to face own flak

Priority: 1) Security, 2) Confusion
Prevent failures from recurring

- “**Note**: if you use a visual editor such as Dreamweaver, make sure you paste this code into the *source code view*.”
- “Log into your perch account and add the following as your live or testing domain: **perchdev.swing.org**”

Support response speed is critical!

Schema-less structured data
XML-style tags for compatibility with Dreamweaver
Table of k-v pairs or blob of JSON
K-V: requires housekeeping, inefficient for small values (MEDIUMTEXT for boolean value), fast queries
JSON: cannot be filtered in SQL, JSON not in PHP < 5.2 (PEAR libraries are quite slow)
Stack data keys (revisions): [{ rev: n, key: value }, { rev: n, key: value }, …]

Experience > Theory

Platform – all bets are off
Cheap hosting is cheap (full/unwritable/unconfigured session save path)
Migrating between filesystems, case sensitivity, esp. w/MySQL (Windows: case-insensitive, Mac: case-preserving, UNIX: case-sensitive)

Opinionated vs Dictatorial (“WYSIWYG is evil”)
Ships with markdown text editor by default
It's not our place to tell people how to work
Plugin system

Help customers look good in front of their clients
Build customer confidence
Every field can be annotated by designer, help text blocks
Draft, preview, undo
Confidence promotes software use

“No you don't want to do that; what we've built is sufficient”
Accept when users are having problems
Really great devs solve problems
Wake-up call: can't say no every time

http://lanyrd.com/2011/phpcomcon
http://grabaperch.com
@drewm

Paul Reinheimer

The WonderProxy story

Born out of need to test geoip applications

Mistakes – “crouch and hope you don't get hit”

- No account de-activation
- NIH – wrote paypal IPN code instead of re-using own code
- Mixing Linux distros
- Server account renewals
- Afraid to look at profitability numbers

Blog about problems we encounter – competitors find posts
Blog about problems customers have – bring customers instead of competitors
Partner with complementary services

Laura Beth Denker

“Is it Handmade Code If You Use Power Tools”
@elblinkin

NEED CONFIDENCE in your code

Before...

- Weekly deployments
- Deployment happened on branches
- Release Manager rotations
- Buildbot (python tests, not well suited to server coding, slow, flaky)
- Jenkins CI to replace Buildbot
- Move to PHPUnit
- Deployinator, homegrown one button deployments (hourly deployments)

Test pyramid – shaped like Sorry board game piece

- Functional (ball) – human testing
- Integration (neck) – database
- Unit (base) – annotate based on what happens when components are removed – drastic speed difference
  - @group
  - caches
  - databases
  - network tests (external 3rdparty APIs)
  - sleep
  - time
  - smoke, curl, regex
  - flaky

Don't use random data in unit tests
Test each case in control structures

DBUnit for multiple databases

- Sharded architecture

Test against expectations
Use PHPUnit Mock Helpers

trunk/qa/princess/prod

Cross-browser testing
homegrown tester, soon to be released open-source
uses Clojure compiler to check JS

Andrei Zmievski

"What happened to Unicode in PHP"

What is Unicode?

- Industry standard for consistent text handling
- Universal character set
- Not I18N

I18N

- Character set
- Date/time formats
- Currency formats
- Collation (sorting, contractions)

Mojibake – garbage characters

“I | Unicode, You | Unicode”

Helgi = Islthorp, aka “Mr. Security Override”
Joel

Weakest link should not be PHP

- Native
- Complete
- No dependency mishmash
- No missing locales
- No bias

ICU library

PHP6

- PHP5 + Unicode
- String types – unicode, binary
- input/output encoding
- code points
- capitalization
- transliteration
- pecl_intl – Collator, NumberFormatter, MessageFormatter

Postmortem

- + raised awareness
- + right technology (ICU had everything needed)
- + unit tests
- + pecl_intl
- + code segregation
- - choice of UTF-16 (primary driver was ICU, but UTF-8 would have been easier)
- - core stuff lagged (PDO, filter, etc)
- - lack of mind share
- - boring
- - delayed new features
- - mea culpa (left Yahoo and got distracted)

PHP6 trunk moved to a branch

http://zazzle.com/andreiz

http://www.slideshare.net/andreizm/the-good-the-bad-and-the-ugly-what-happened-to-unicode-and-php-6

Terry Chay

Closing Keynote

http://phpdoc.info/chayism/