CWX
CODEWORKS 2010

A 100k Users.. Now What?

SEATTLE • PORTLAND • AUSTIN • BALTIMORE • ORLANDO

D. Keith Casey Jr
Chief Stuff Breaker/Blue Parabola

Overview

- Basic triage and debugging
- Stack-wide Performance Tips
  - PHP
  - Web Server
  - MySQL
  - Browser
- More Tools & Help

So.. who are you again?

- D. Keith Casey, Jr.
  - Chief Stuff Breaker
  - My Job:
    - 60-70% development
    - 30-40% community building, etc
    - 50-60% babysitting Cal

Our premise

- Your site became successful overnight, what used to a humorously pitiful amount of traffic was just mentioned by [choose one: Drudge, Mike Arrington, The Oprah, Justin Bieber, those vampire guys in Twilight]
  - You have to scale but you don't have a plan in place, the budget on hand, or the calm, serene boss

DON'T PANIC

Step 0: Plan

- If you have a plan for scaling – *including signals on when you should* – this is much easier
- If you wait until you have an outage, you're most likely toast.. for now
- If you're on “the cloud,” scaling is usually faster and easier, not always cheaper

Step 1: Triage!

- Find the biggest problem first
- Eliminating the 5% problem is great but what if we can reduce the 40% problem by half?
- But we don't know which problem is 5% and which is 40% until we collect data

Step 1: Triage – Collect Info

- Collect Database Logs
- Collect Server Logs
- Collect PHP & Application Error Logs
- Collect Firewall/Router Logs *(usually later)*
- Collect User Feedback *(usually later)*

Step 2: Analyze

- Find your bottlenecks
  - What's your server load?
  - Are certain pages significantly slower than others?
  - Is your database responsive?
  - Is the firewall dropping connections?
  - Are your images loading but not the page?
  - Are your pages loading but not the images?

Step 3: Hypothesize

- When you find a problem area, figure out what “fixing it” - *both short and long term* – looks like
- We don't have the time to guess what will fix things and just “*see what happens*”

Step 4: Make the Change

- Once you have the fix figured out, test & apply it when appropriate
- *Good God, have a plan to roll back just in case*

Step 5: Measure Again

- If what you predicted didn't happen, why not?
  - If the problem is worse, roll back
  - If the problem is better, how much better?
  - If the problem is solved, stop kidding yourself
- goto 'step1';

Database Optimizations

- The Slow Query Log
  - Find your Slow Query Log, read it
  - No, seriously. That's it.
  - Hint: You might have to turn it on in your my.cnf and tell it where to write
- EXPLAIN
  - Incredibly powerful way to get MySQL's perspective on your select queries
  - Another session by itself: http://www.slideshare.net/ligaya/explain

Database Indexes (short term)

- Indexes are for the database, not you
- They make your queries faster by “filtering” or “caching” a smaller set of information before the query is run
- Remember orthogonality:
  - They make your queries faster by “filtering” or “caching” a smaller set of information before the query is run

So how useful are they?

- In dotProject, prior to v2.1:
  - 20 projects with 25-30 tasks ~30s
  - 100 projects with 400 tasks ~6 minutes

- In web2project v1.0 (with indexes):
  - Added indexes on all foreign key relationships
  - First scenario... 0.44s (~70x faster)
  - Second scenario... 12s (~30x faster)

Database Denormalization (long term)

- Entity-Attribute-Value Model
  - **Table**: products
  - **Fields**: product_id, product_name
  - **Sub-table**: product_info
  - **Fields**: info_id, pi_field, pi_value
- This is a problem for datatype validation

So is it useful?

- In web2project v1.0 (with indexes):
  - Added indexes on all foreign key relationships
  - First scenario... 0.44s (~70x faster)
  - Second scenario... 12s (~30x faster)
- In web2project v1.2.2 (with aggregation):
  - Added caching on task count, hours worked, and percent complete
  - First scenario.. 0.48s (irrelevant)
  - Second scenario... 0.6s (20x faster, 600x faster than original code)

echo vs print()

- The age old question: echo vs print..

Which is faster?

echo vs print()

- The age old question: echo vs print:
- 0.011095046997070 vs 0.011945962905884 which means echo is 850µs (0.85ms) faster!!
  - mysql_connect() → 100µs (0.1ms)

echo vs print()

- The age old question: echo vs print:
- 0.011095046997070 vs 0.011945962905884 which means echo is 850µs (0.85ms) faster!!

  - mysql_connect() → 100µs (0.1ms)
  - HTTP Get Request → 35,000µs (35ms)
  - ping yahoo.com → 75,000µs (75ms)
  - DNS lookup → 200,000µs (200ms)
  - *JUST STOP ARGUING!!!111eleventyone*

Source: http://www.crazy-media.se/echo-vs-print/
Source: http://slidesha.re/1oANCZ

PHP Caching

- APC is best known for its byte code caching abilities but it’s also capable of storing data in shared memory
- Useful when data does not need to be propagated across multiple machines
- Provides a single point of memory configuration

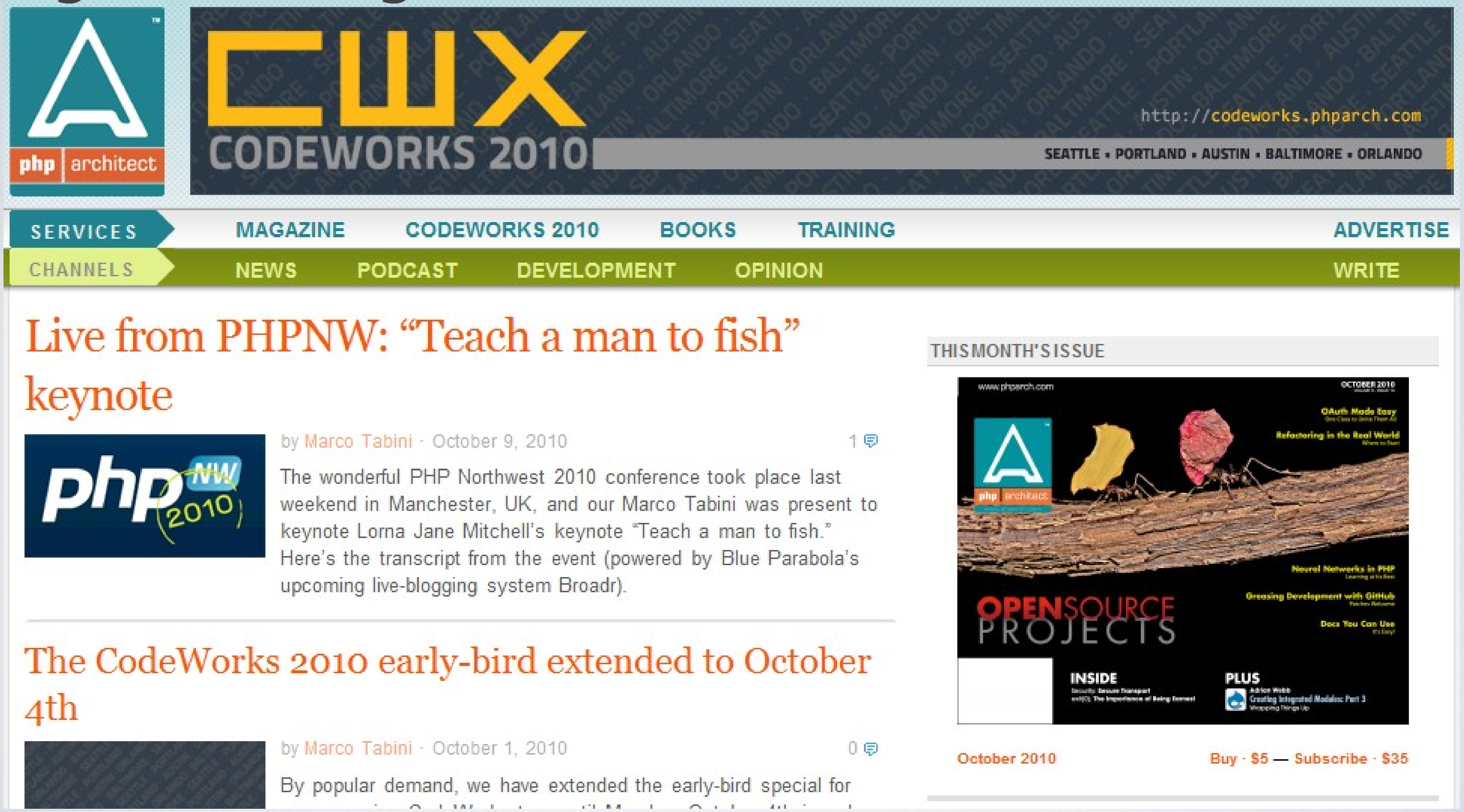
- APC coming in core in PHP 5.4
- WinCache – http://www.iis.net/download/wincacheforphp

Page Caching

- Quick and Dirty:
  - Generate the page, write it to a data store
  - Varnish, memcache, ram disk, database(?)
  - Drupal, WordPress, etc, etc
  - How do you handle invalidation?

  - Two big issues to solve:
    - *How often does your page change?*
    - *Does it change all at once?*

Page Caching - Problem!



Live from PHPNW: "Teach a man to fish" keynote

by Marco Tabini · October 9, 2010 1

The wonderful PHP Northwest 2010 conference took place last weekend in Manchester, UK, and our Marco Tabini was present to keynote Lorna Jane Mitchell's keynote "Teach a man to fish." Here's the transcript from the event (powered by Blue Parabola's upcoming live-blogging system Broadr).

The CodeWorks 2010 early-bird extended to October 4th

by Marco Tabini · October 1, 2010 0

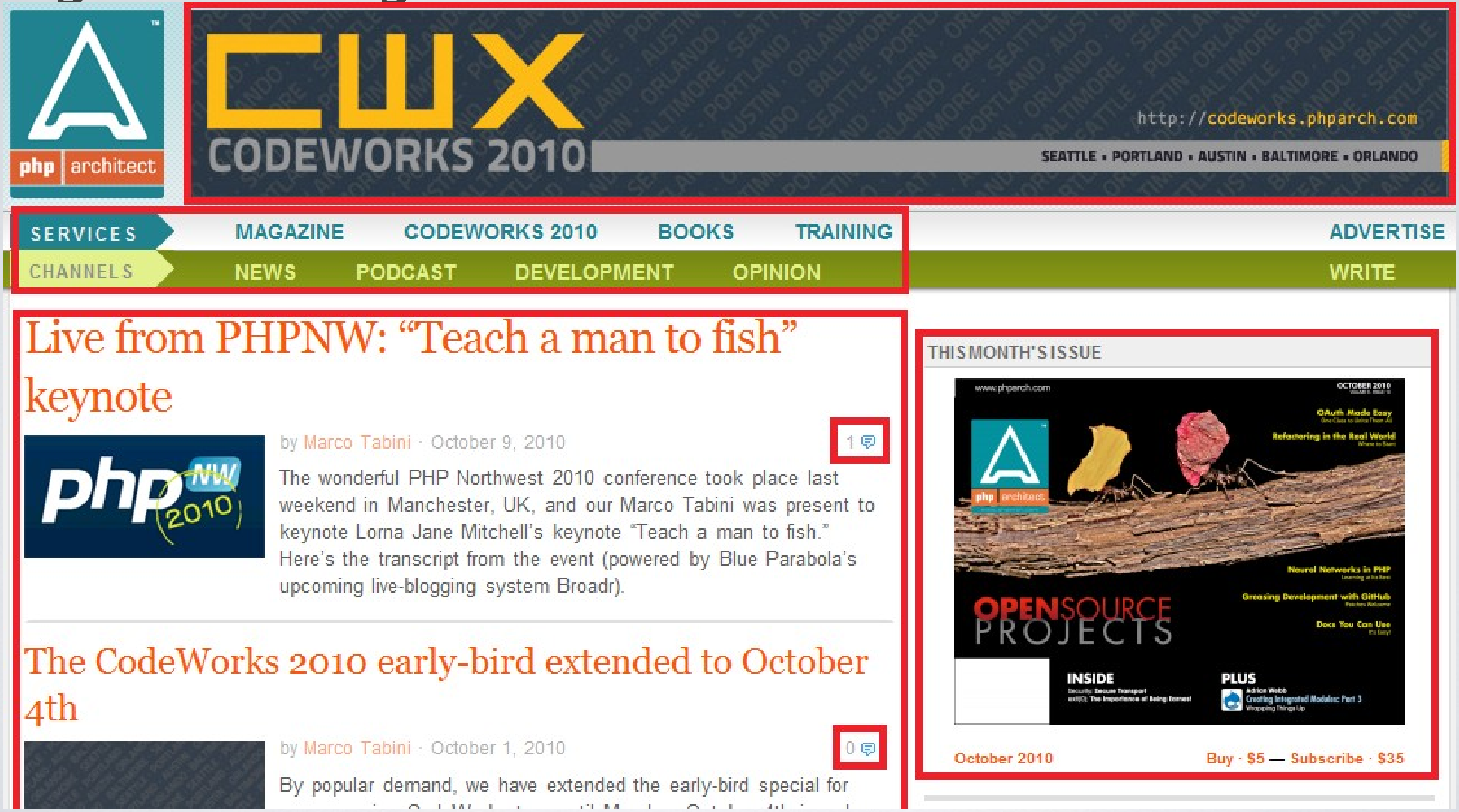
By popular demand, we have extended the early-bird special for

THIS MONTH'S ISSUE

October 2010

Buy · $5 — Subscribe · $35

Page Caching - Solution!



Live from PHPNW: "Teach a man to fish" keynote

by Marco Tabini · October 9, 2010 1

The wonderful PHP Northwest 2010 conference took place last weekend in Manchester, UK, and our Marco Tabini was present to keynote Lorna Jane Mitchell's keynote "Teach a man to fish." Here's the transcript from the event (powered by Blue Parabola's upcoming live-blogging system Broadr).

The CodeWorks 2010 early-bird extended to October 4th

by Marco Tabini · October 1, 2010 0

By popular demand, we have extended the early-bird special for

THIS MONTH'S ISSUE

October 2010 Buy · $5 — Subscribe · $35

Partial Page Caching

- Quick and Dirty:
  - Generate and save individual blocks of info
  - Varnish, memcache, ram disk, database(?)
  - Drupal, WordPress, etc, etc
- Two big issues to solve:
  - *How often do your blocks change?*
  - *How do you handle invalidation?*

Memcache

- Memcached is an advanced memory manager, supporting expiration of data, and sharing across a network
- aka *The Big Bucket in the Sky*
  - Stand alone application
  - Requires a trivial PHP extension
  - Available across the network is key

Apache (or IIS) Tweaks

- Every file provided by your server must be provided by your server
- Consider moving static files – flat html, css, js, or images – to other physical servers or different server packages – lighttpd, etc
- Consider moving large(r) files – images, audio, video, etc – to a Content Delivery Network
- Compress output with gzip – but check first!

Browser/Front End Tweaks

- Every HTTP request takes resources (time, bandwidth, etc), so make fewer of them
- Put CSS at the top of the page, Javascript at the bottom, and remove duplicate CSS & Javascript
- Don't scale images in the browser!

Tools: Browser.. Server?

- Page Speed
  - Firefox Plugin building on Yslow
  - Examines the page loading process from beginning to end from the ***client's*** perspective
  - Performs a series of 22 tests/analyses including tests on Content, Cookies, CSS, images, Javascript, and Server w/ priorities
  - mod_pagespeed for Apache2
  - Rumors from Microsoft on IIS..

What about the User?

- Google Page Rank
  - Matt Cutts – Google Search Guy – confirmed that search rankings will be affected by site performance/responsiveness
  - Overall Visitor Experience
    - Users get bored, distracted, and annoyed easily
    - Especially when they're paying

Resources:

- Database Design for Mere Mortals - http://www.amazon.com/Database-Design-Mere-Mortals-Hands/dp/0201694719
- Anything from Jay Pipes – http://bit.ly/5yqTdf
- “Why you should replace ENUM with Something Else”
- MySQL Performance Blog - http://www.mysqlperformanceblog.com/
- SQL Antipatterns Strike Back - http://www.slideshare.net/billkarwin/sql-antipatterns-strike-back
- IIS – WinCache / http://web.ms/php
- Apache Performance Tuning – http://httpd.apache.org/docs/2.0/misc/perf-tuning.html
- MySQL Tools: Explain, Slow Query Log – http://www.mysqlperformanceblog.com/

Questions?

D. Keith Casey Jr, Chief Stuff Breaker

keith@blueparabola.com

Twitter/Skype/AIM/IRC: caseysoftware