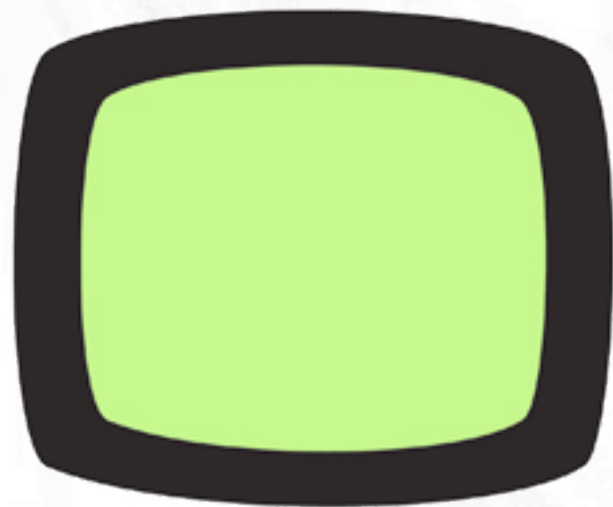


The Eye of the Storm: Scaling the Pickens Plan website after huge media attention



THE **b**IVINGS GROUP
strategy.technology.impact

John Bafford
Senior Director, Programming Services
The Bivings Group

DC PHP 5/13/2009

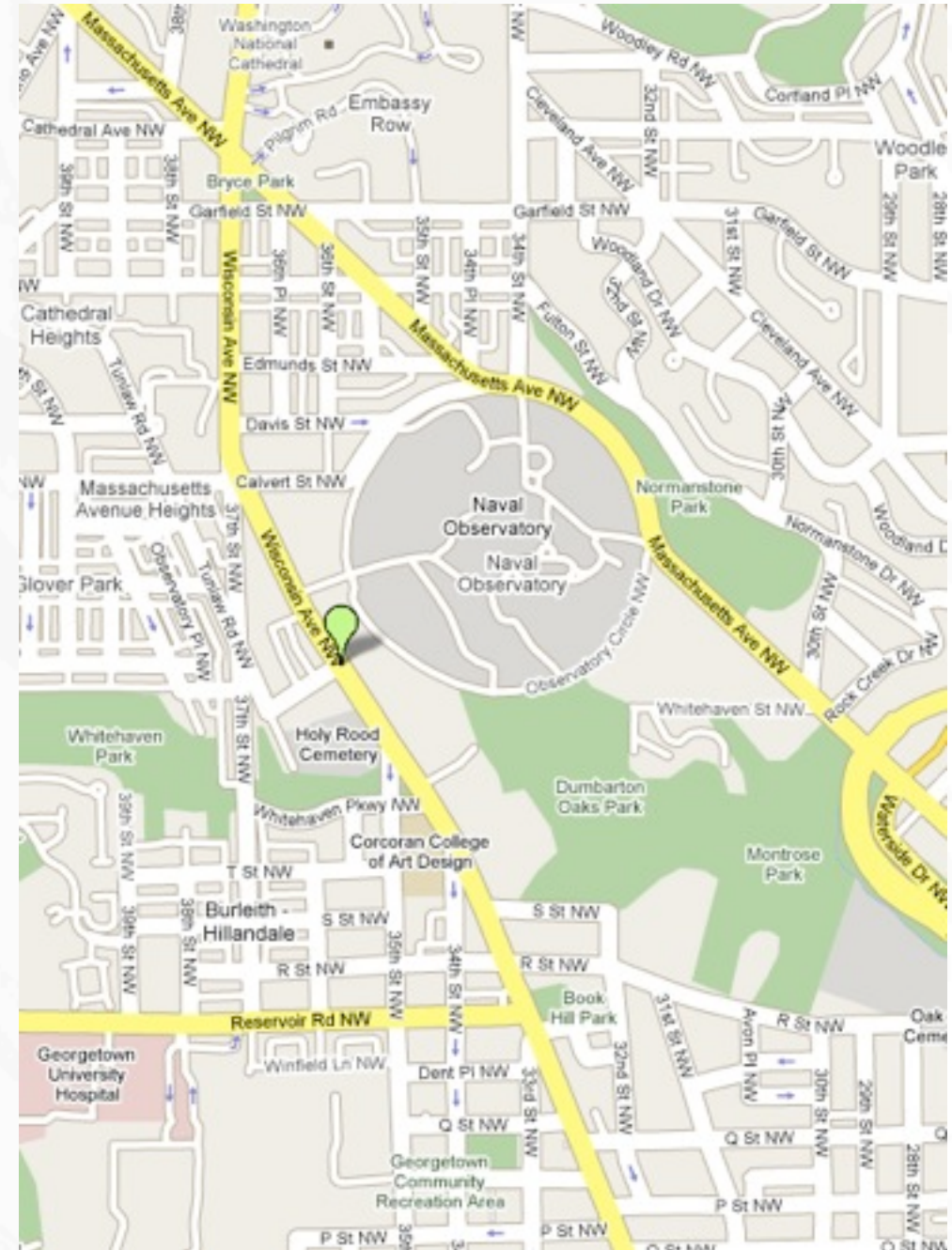
Who am I?

- John Bafford
Senior Director, Programming Services
The Bivings Group
- <http://bafford.com>
- Twitter: @jbafford
- PHP/mysql programmer since 1999
- Programmer since 1987

THE BIVINGS GROUP

strategy.technology.impact

- Internet Communications
 - WordPress and Drupal
- Media Monitoring
 - ImpactWatch
 - <http://impactwatch.com>
- Clients include HP, AMD, BlackBerry, Pickens Plan
- Twitter: @bivings
- <http://bivings.com>



What We'll Discuss

- Pickens Plan website
- Background
- What caused the site to fail
- How we fixed it
- Lessons Learned

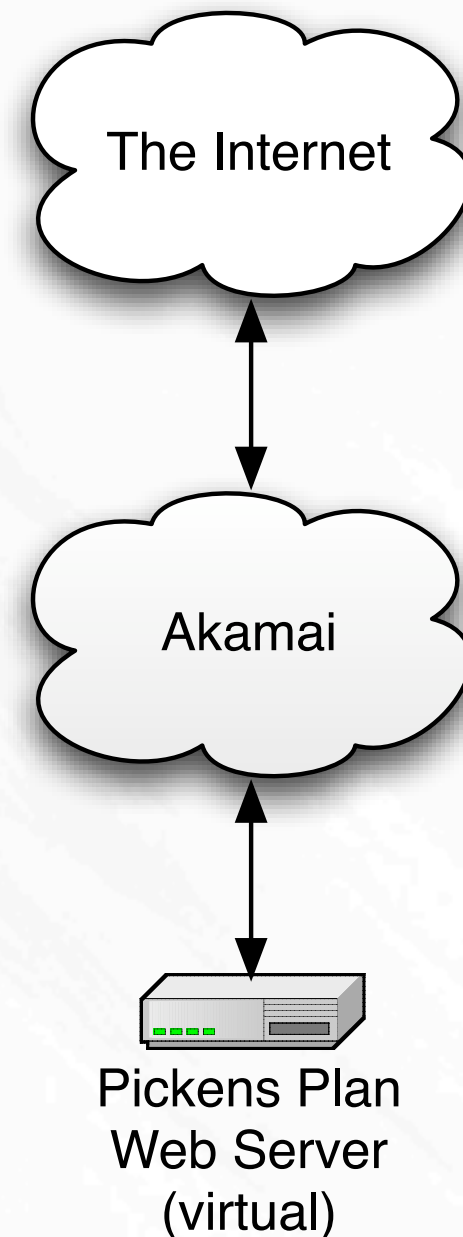
Background

- Marketing website for Pickens Plan
- Took over in September 2008
- Flat HTML/PHP
 - No CMS
- WordPress blog
- Ning community site
- <http://www.pickensplan.com>



Background: Website Architecture

- Hosted on virtual private server
- LAMP
- One server for both dev and live
- Akamai Content Distribution Network

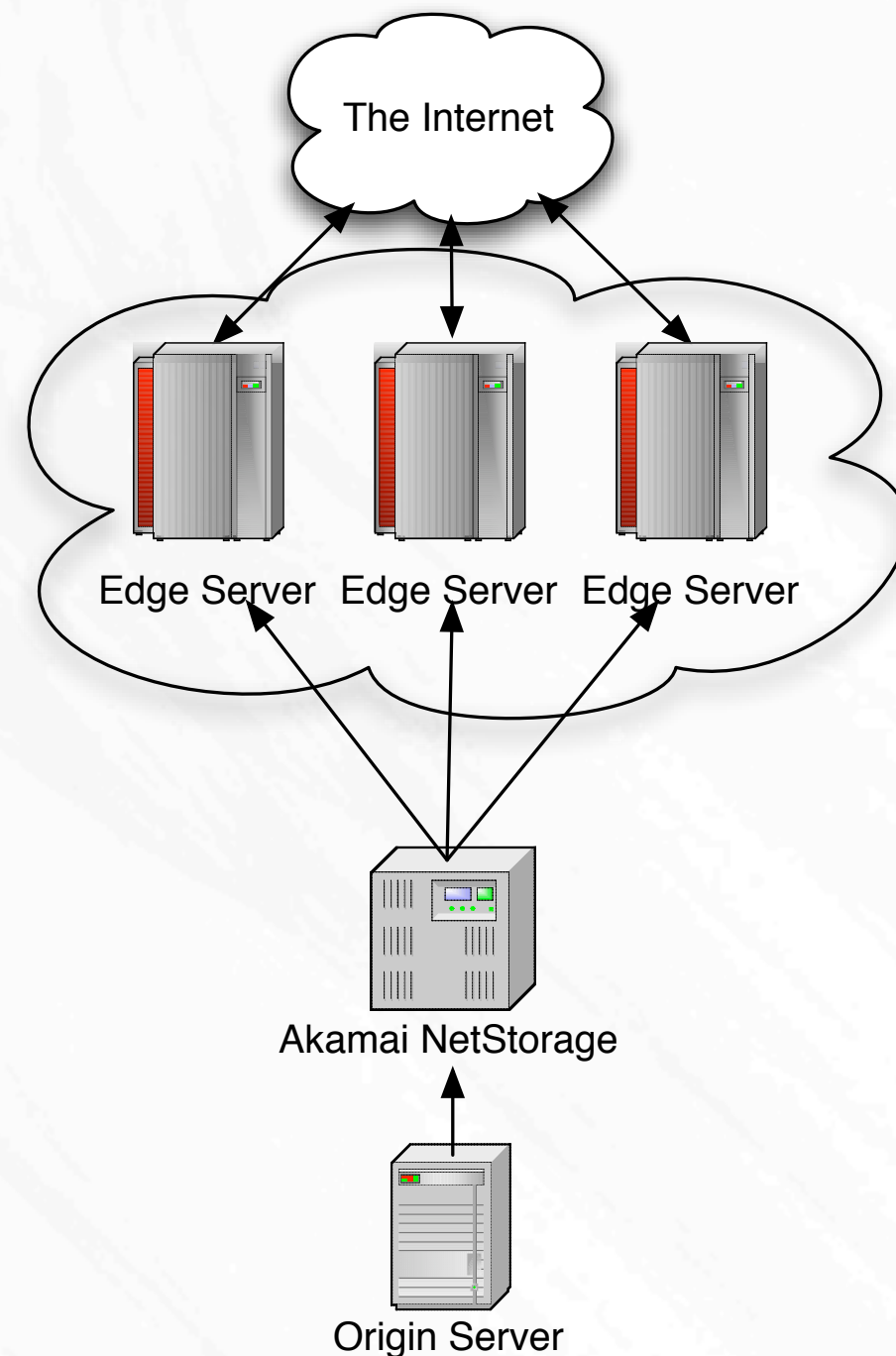


Akamai Overview

- Akamai has worldwide network of servers
 - These servers store or cache your data
- Website DNS points to Akamai
 - Akamai returns closest mirror to end-user
- End-user gets faster download from geographically local server
- Reduced traffic to origin server
- We used two services: NetStorage and EdgePlatform

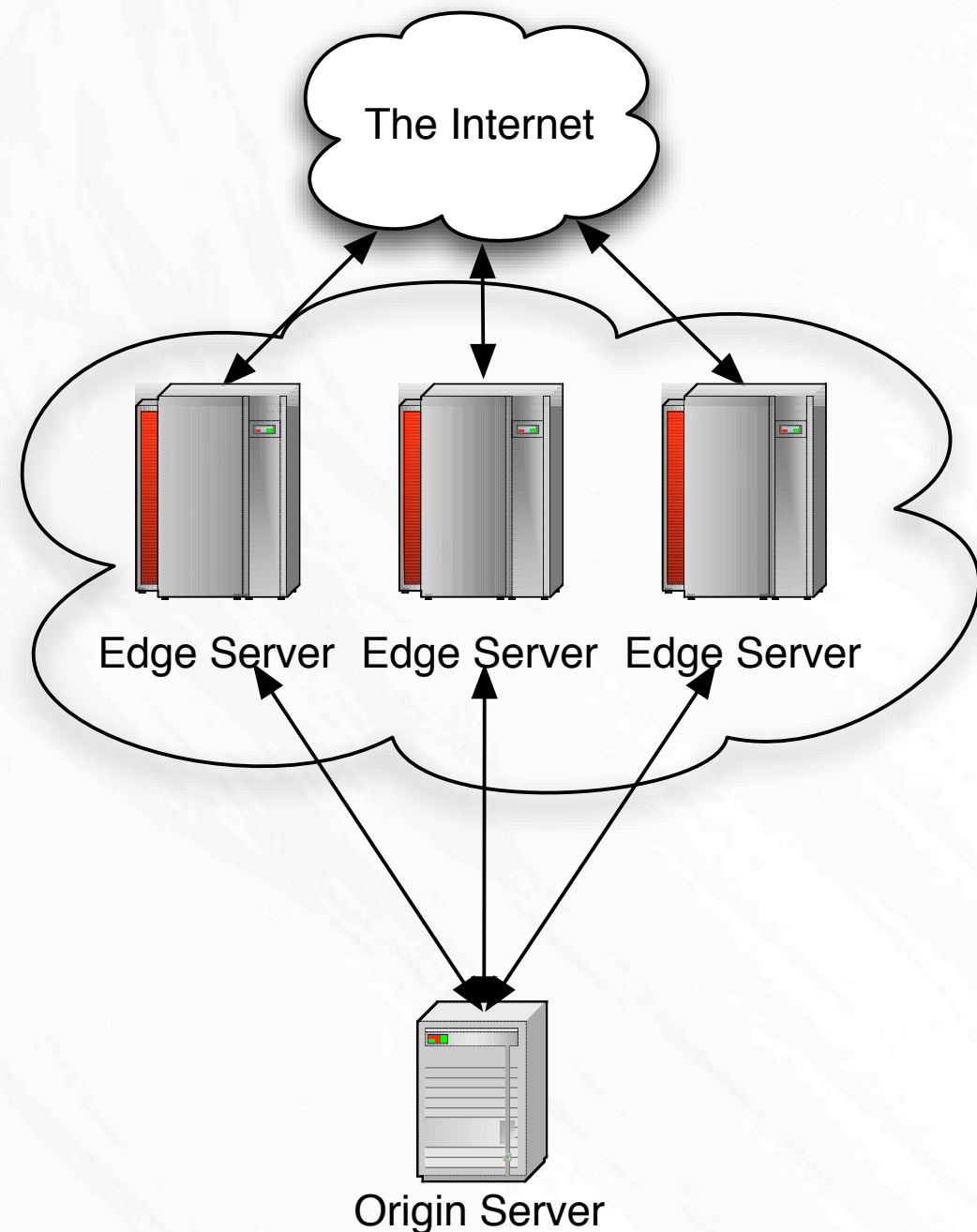
Akamai NetStorage

- Store media and infrequently changing files directly on Akamai's Servers
- Requests never come to primary server, saving bandwidth and CPU
- Served directly from Akamai's global network



Akamai EdgePlatform

- Automatically caches web pages and images
- Easier to manage than NetStorage
- Reduces (but does not eliminate) traffic to origin server



Disaster Strikes!

- Friday September 26 — First Presidential Debate
- Live chat on website after debate
 - Advertised on national TV!
- 91.2 page views / second (2600 requests / second) at Akamai
- Apache stopped responding, but load was low and server was otherwise responsive

What Happened?

- Hosting plan was “insufficient”
- Live chat software blamed (but innocent)
- Server had plenty of memory and CPU
- But VPS had kmemsize limit of 64 MB
 - Hosting providers tend to keep this low intentionally
- Server was more than capable of handling traffic, if only it had the kernel memory to keep track of all of its connections...
- Lots of problems contributed

Contributing Problems: PHP Opcode Cache

- Wasn't installed!
- Provides incredible performance boost by caching PHP opcodes
 - PHP does less work on each page request
 - Requests completed faster
 - More pages served in less time
- Every PHP site should have one installed
- We like APC
- <http://pecl.php.net/apc/>

Contributing Problems:

WP-Super-Cache

- Wasn't installed!
- Dramatically increases performance of WordPress by caching most content
- Should be installed on every WordPress site
- <http://wordpress.org/extend/plugins/wp-super-cache/>

Contributing Problems: Poor Caching

- Home page had a section displaying recent blog posts
 - Wasn't cached!
- Every access to the home page caused an access to the blog RSS feed

Contributing Problems: User Access Tracking Software

- Mint was installed to generate access statistics
 - Like Google Analytics, but runs on your own server
- Requires two accesses to the server to generate non-cacheable dynamic JavaScript
- Third access caused by misconfiguration

Other Problems

- Every page had 404 errors in HTML templates
 - 404 pages not cached for very long by Akamai
- Splash page implemented oddly
 - Apache parsing .html as PHP
 - Akamai not caching .html files
- Every page generated multiple PHP warnings, which were not displayed, but logged to Apache's error_log

Other Problems

- Webserver and MySQL were on same machine
 - Less server resources available to Apache
- We were not on hosting provider's contact list
 - They were very nice in telling us what was going on, but they refused to make any changes.

Connection Statistics

Total Apache Connections	6
Concurrent Apache Connections	2
Data Transferred	95k
MySQL Connections	4
Concurrent MySQL Connections	2

The Challenge:

We only had four days

How To Fix?

- Initial reaction: upgrade to full private server
- Expected even more traffic on VP and 2nd and 3rd Presidential Debates
- Wanted to make sure this couldn't possibly happen again
- Client decided they want a load-balanced setup.
 - Won't be ready for VP debate.

**“14 days? We need the new
servers in 14 hours!”**

What We Did — Easy Stuff

- Upgrade to full virtual server
- Install APC and WP-Cache
 - APC and WP-Cache have issues, need to disable APC for WP-Cache
 - <http://markjaquith.wordpress.com/2006/02/13/adventures-with-wp-cache2-apc/>
- Replace Mint with Google Analytics
- Cache blog entries on home page
- Fix 404 errors

What We Did — Hard Stuff

- Changed splash page to be entirely JavaScript driven
- Change Akamai configuration to allow caching for html and php files
 - And change PHP code to tell Akamai how long to cache PHP files

Caching Pages With Akamai

```
function SetCacheHeaders()
{
    if(!headers_sent())
    {
        if(($_SERVER['REQUEST_METHOD'] == 'GET') && (!empty($_CacheAnyways) ||
empty($_GET)) && !session_id())
        {
            $dir = dirname(__FILE__);

            $mtime = max(filemtime(__FILE__), filemtime($_SERVER['SCRIPT_FILENAME']),
filemtime($dir . '/widgets.php'), filemtime($dir . '/footer.php'));

            $expiresTime = 900; //15 minutes
            header('Last-Modified: ' . gmdate('D, d M Y H:i:s', $mtime) . ' GMT');
            header('Expires: ' . gmdate('D, d M Y H:i:s', time() + $expiresTime) . ' GMT');
            header('Edge-control: cache-maxage=15m,!no-store,!bypass-cache');
        }
    }
}
```

Performance Testing

- With all changes made:
 - 225 hits/sec to index page
 - 100 hits/sec to WordPress
- Used ab (ApacheBench)
 - ab has problems with testing with large numbers of concurrent connections
 - Had to run tests from four different machines on two networks
- Numbers even better with Akamai

Connection Statistics

	Before	After
Total Apache Connections	6	1
Concurrent Apache Connections	2	1
Data Transferred	95k	35k
MySQL Connections	4	1
Concurrent MySQL Connections	2	1

Connection Statistics

	Before	After
Total Apache Connections	6	0
Concurrent Apache Connections	2	0
Data Transferred	95k	0k
MySQL Connections	4	0
Concurrent MySQL Connections	2	0

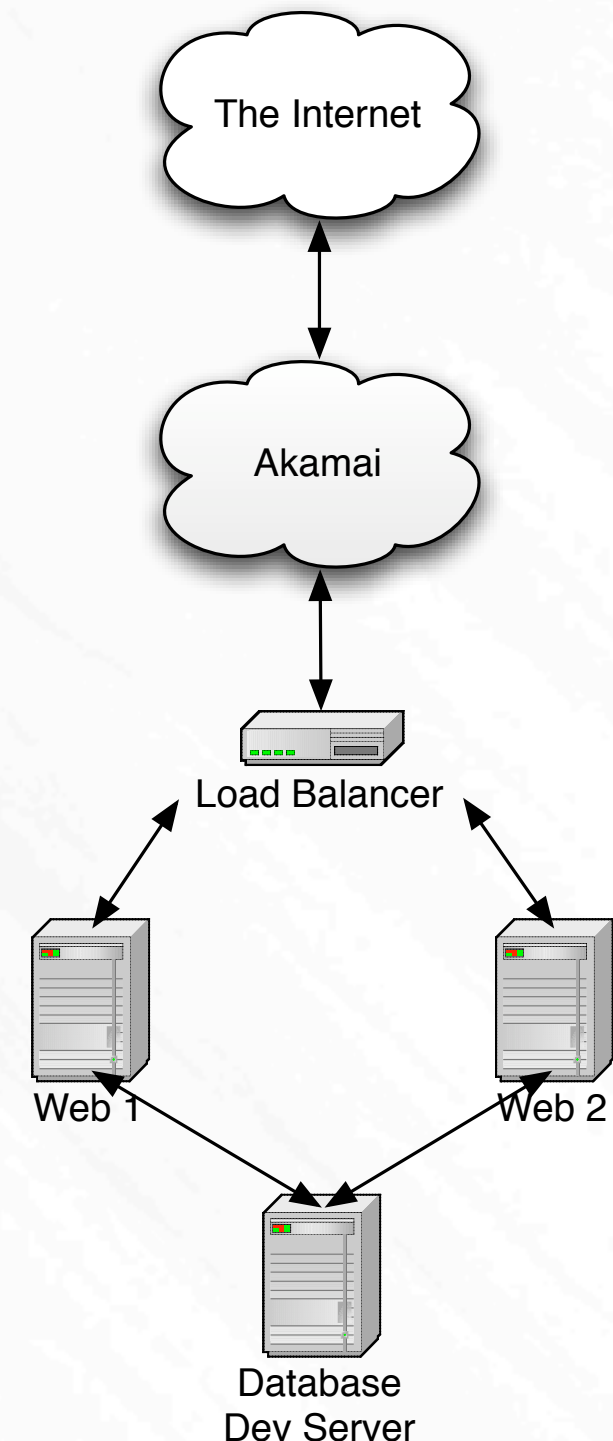
VP Debate Results

- 9.7 page views/sec (1/10 traffic from 1st Presidential Debate)
- Server load 1/100 of 1st debate
 - 10x performance improvement at the server

Success!

Setting up the Load Balancer

- Requires hosting provider to provision three new machines
 - Two web servers
 - Load balancer
 - Existing web server becomes dev web server and database server
- We got them to do it in four days (sort of...)



“My database server is in California. What do you mean my new web servers are in Virginia?!”

Two Servers Adds Complexity

- Needed new system for site deployments
 - Manually uploading files to the live site is much harder with two live sites
- Use rsync to copy to the two servers — faster and safer
 - Web-based script to trigger rsync
- We now have a staging website
 - Bonus: Dev and staging are not on the live servers

Deployment Improvements

- One-click deployments
- Don't need to give access to live servers to anyone
- Can see list of changed files before deploying
- Possible improvements we didn't implement:
 - Email alerts on deployment
 - Automatic svn commits on deployment
 - Use svn to provide diffs before deploying
 - Use svn to provide rollback functionality

Two Servers Adds Complexity

- File-based sessions don't work with multiple web servers
- Database-backed session handler
- Lots of examples for creating session handlers on the internet, but few get database locking correct

Two Servers Adds Complexity

- Load balancer configuration
 - Non-Sticky sessions: No affinity between a user and a web server
 - Accesses go to least-utilized (or random) web server
 - Sticky sessions: User sessions stay on one web server
 - Problem when your “user” is the Akamai Edge network

Two Servers Adds Complexity

- WordPress file upload
 - Images need to be on both web servers to work
 - WordPress stores full URL of uploaded content
 - Lets us take advantage of Akamai NetStorage
 - Modified code to upload files to Akamai
 - Bonus: images on blog never accessed from web servers
 - If you don't have Akamai, could upload to static content server instead

Second Presidential Debate

- October 7
- Live text chat starting at 7 pm
- Main Event: video chat with T. Boone Pickens at 10:30 pm after debate
- We have everything set up and working, right? Nothing can go wrong!
- We decided to switch live chat providers

New Chat Software

- We decided to change live chat vendors to remove moderation and incorporate video
 - Apparently, trying to moderate thousands of people is hard
- New chat vendor was writing new code at 5 pm
 - For 7 pm chat!
- Chat software went live with no significant testing or review
 - Made us nervous

Murphy's Law In Action

- Second-guessing ourselves and poor quality of chat code led to provisioning third web server
- Live Chat software had some problems
 - Not completely set up until 6 pm
 - Exposed previously unknown bug in chat server
- Random 404 errors
 - Wait ... about 1/3 of the accesses ...
 - Hosting provider added (unconfigured) third web server into load balance pool before they were told to do so

Chat Success

- 207,000 unique visitors
- 40,000 email signups
- Load on web servers was barely measurable
- The properly configured servers did not go down

The client was happy!

(with us, anyway)

Third Presidential Debate

- Client decided to forego the live chat.
- Live video broadcast after the event.

Lessons Learned

- People need sleep!
 - Hard to think clearly on hour #60
 - Mistakes and misunderstandings are more likely.

Lessons Learned

- Have confidence in your abilities
 - Last minute second-guessing won't help.
 - Pick what you think you need, and work with that. Don't try to scale past that at the last minute.
 - If you think you will have to, you should probably look at allowing your website to gracefully degrade instead.

Lessons Learned

- Leave time for testing!
 - (Deploying untested code is bad.)
- If you're doing a live event, it should be done and tested the day before.
 - Live events have many failure modes besides just the code

Lessons Learned

- For some sites, CDNs allow sites to scale far in excess of what one web server can handle
 - Adds some complexity, but it can be worth it

Lessons Learned

- Expect last minute changes
 - It's not possible to cheat Murphy

Questions?

Thank You!

Resources

- APC: <http://pecl.php.net/apc/>
- WP-Super-Cache
 - <http://wordpress.org/extend/plugins/wp-super-cache/>
 - <http://markjaquith.wordpress.com/2006/02/13/adventures-with-wp-cache2-apc/>
- Pickens Plan: <http://www.pickensplan.com>
- Bivings: <http://bivings.com> (@bivings)
- John Bafford: <http://bafford.com> (@jbafford)