Hey, my web app’s slow! Where’s the problem?

Charlie Arehart
Independent Consultant

charlie@carehart.org / @carehart

Another day, another slow site…

- Video available at http://fast.wistia.net/embed/iframe/swl2mt99vb

A range of problems/solutions

- Understanding the challenges
- Finding the problems: from the client perspective
- Finding the problems: from the server perspective
- Solving the problems: several ways
- Testing your solutions
- Watching for problems over time

About Charlie Arehart

Independent consultant

- 30+ in Enterprise IT
- Adobe Forum MVP, CAB member
- Frequent speaker to conf's worldwide
- Organizer, Online ColdFusion Meetup (coldfusionmeetup.com), 2800+ members
- Living in south central KY

Web home at www.carehart.org

- 100+ presentations, 80+ articles, 400+ blog entries
- UGTV: recordings of 600+ presos by 300+ speakers
- CF411.com: 2000+ tools/resources, 150+ categories
- Hosting courtesy of EdgeWeb Hosting

Housekeeping

- Slides available online
  - carehart.org/presentations
- My meta resource site
  - CF411.com
- Closely related talks this week at CF Summit 2015
  - My DB Skills Killed She server – Dave Ferguson
  - 10 Common CF Server Challenges… - Charlie Arehart

Understanding the challenges

Why worry about site speed? (duh)

- Simple usability
- User impatience: they may seek alternatives
- Search engines now ranking based on speed
- Bandwidth (on your server): someone’s paying
- Cloud deployments: you’re definitely paying!

Who am I talking to?

- Any kind of web app developer, any platform
- Whether designing for desktop or mobile
  - Hybrid/responsive, static/dynamic, ajax, etc
  - Less if using flat or single page models, of course
- Speed problems can affect anyone
  - For many reasons…

Where might a problem be?

- Could be on client, server, or network in between
- On the client?
  - Could be your app/code
  - Could be a temporary OS problem for the user
    - CPU, memory, disk, etc
  - Could be the user’s browser
  - And so on

Where might a problem be? (cont)

- On the server?
  - Could be a problem in app server
  - Could be in web svr, db svr, etc
  - Could be a temporary OS problem (cpu/disk/memory)
  - Could be something in your code (app svr, db svr code)
  - Could be issue with config
  - Could be a need to optimize for web delivery

Where could the problem be? (cont)

- In the network?
  - From client, within their network (wifi, wired, mobile)
  - On server, could be comm between servers (app/db)
  - Could be calls to 3rd party sites (from client or server)
  - Could be infrastructure in between
    - Network/mobile provider
    - Proxies
    - Firewall/load balancer
    - Colleagues/family/neighbors overusing bandwidth

Where could the problem be? (cont)

- Can be maddening!
- But problems can be found and solved
  - with right perspective, tools, techniques, resources
- Goal today
  - Give you that perspective
  - Show you such tools: some already at hand
    - May be familiar to some; seems not to most
  - Point you to resources for techniques and more

Finding the problems: from the client perspective

Browser-provided tools

- Most browsers offer “dev tools” with similar features
  - Right-click on browser whitespace, choose "inspect element"
  - Or use ctrl-shift-I in most browsers, or F12 in some
  - Safari: use Preferences>Show Developer Menu
- See “network” tab, especially
  - Load times, of both your and 3rd party site requests
  - Details on each resource requested (text, graphics, css, etc.)
  - Options to focus on certain file names, types only
  - Option to clear/preserve display of files between refreshes
  - Clearing/disable of browser caching
  - Dockability of window, other configurability

Browser-provided tools (cont)

- Chrome network tab also offers
  - “Capture screenshots”
  - Bandwidth throttling simulation
- Most also offer javascript profiling and much more
- Features have evolved over the years
  - Browser-specific observations follow
- Chrome/Opera
  - Related tabs: timeline, profiles
  - Audits tab makes recommendations
  - …

Browser-provided tools (cont)

- Firefox
  - Related tabs: performance (to profile js)
  - Right-click any requested file, choose "start performance analysis" to see graphs of cached vs not cached
    - or click on number of requests in bottom right, to see same
- Internet Explorer (especially as of 11)
  - Related tabs: Profiler, UI Responsiveness, Memory
  - See also Emulation tab
  - See also tools>performance dashboard
- …

Browser-provided tools (cont)

- Similar functionality in standalone apps
  - Fiddler, Charles, ServiceCapture, many others
  - See my list at cf411.com/genericproxies

- Let’s leave browser tools; focus on client machine

System (OS) Monitoring Tools

- Problem could be on your/client’s machine
  - Several tools built-into OS can help
  - Can observe other things running on machine
  - May be stealing cpu, memory, disk i/o, registry i/o, bandwidth
- Windows:
  - Task Manager and its cousin, Resource Monitor
  - See also Performance Monitor, Process Monitor (downloadable)
- *nix
  - top/plist
- OS X
  - Activity Monitor

Network testing tools

- Again, problem could be in network bandwidth of your/client’s machine
- Several free services can test that
  - speedtest.net
  - speedof.me
  - bandwidthplace.com
  - See my list at cf411.com/networkbandwidthtest

- Great if you can be at client machine, what if not?
  - Will cover later with “user experience monitoring” tools

Page/site testing tools

- Several tools help evaluate web app performance
  - As would be experienced by client
- Some downloadable client-based tools
  - Yslow (yahoo), speedtracer (google), others
  - See my list at cf411.com/pageperftest_client
- Many are services
  - Most are free, some are paid
  - These test from their servers to your app (simulate client)
  - Most evaluate, grade, and can recommend tweaks
  - Let’s consider a few, with demos

Page/site testing tools (cont)

- webpagetest.org
- tools.pingdom.com/fpt/
- Google pagespeed
  - has feature to provide compressed content from your site
- Several more
- See my list at cf411.com/pageperftest_server

Finding the problems: from the server perspective

Could be any of many things

- Again: app server, web server, OS problems, DB svr
  - Or third-party server called from your server app
- Key is to have tools to monitor on server
  - And many such tools can alert on problems
- Let’s consider each major component

Is it in your app server?

- Several tools can monitor different app servers
  - Newrelic
  - Appdynamics
  - See my list at cf411.com/appmon
- Is it in ColdFusion? CF-specific monitor tools
  - FusionReactor, SeeFusion
  - CF Enterprise’s Server Monitor
  - See my list at cf411.com/cfmon

Is it in your web server?

- Several built-in web server monitoring tools
  - Apache: mod_status
  - IIS: Worker Process Monitor
  - Nginx: ngx_http_stub_status_module
    - Nginx Plus: "Live Activity Monitoring"
- Other downloadable or service-based tools
  - Appdynamics
  - IISTracer
  - LeanSentry
  - See my list at cf411.com/wsmon

Is it an OS resource problems?

- See same OS tools discussed before for clients
  - Good for point-in-time evaluation
- See downloadable and/or service-oriented tools
  - To observe things over time, trigger alerts, etc
  - Nagios, NewRelic, Zabbix
  - See my list at cf411.com/sysmon

Is it in your database server?

- So many database engines, so many tools
  - Can monitor running queries, history, slowness
  - Most have no overhead, yet few seem to consider them
  - See my list at cf411.com/dbmon
- Consider also monitoring the OS resources there
  - Again, problem “with DB” may really be about its server
- Remember also:
  - Network problems could exist between app svr & db svr
  - Would show as “slowness” of query processing

Another dimension to consider

- User Experience Monitoring Tools
  - With these you add small js code to your web app
  - Gets sent to browser with your content
  - Reports back “time on network”, “page render time”
- See tools like
  - NewRelic Browser and Mobile Monitoring features
  - FusionReactor User Experience Monitoring (UEM)

Solving the problems: several ways

So many problems,
so many solutions (so little time now)

- Again, some tools seen offer specific recommendations
- Many other site speed resources to help you
  - developer.yahoo.com/performance/
  - yslow.org, stevesouders.com
  - developers.google.com/speed/
  - developers.google.com/speed/docs/insights/rules
  - iispeed.com/pagespeed/optimizations
  - www.perf.rocks
  - And others

Some sample blog entries

- yeoman.io/blog/performance-optimization.html
- stevesouders.com/blog/2013/05/13/moving-beyond-window-onload/
- cdnify.com/blog/top-10-gulp-tasks-for-optimising-front-end-performance/
- zoompf.com/blog/2013/10/easy-ways-to-speed-up-your-wordpress-site
- rigor.com/blog/2015/02/identify-solve-3rd-party-javascript-problems

The common recommendations

- Most tools/resources focus on speeding up web app content delivery, recommending that you:
    - Minify Javascript/css
    - Remove unused js/css
    - Remove inline css
    - Combine jss and css files into one
    - Compress/minify things not already compressed
        - binaries (pdf, svg, ico, bmp, etc)
        - other text (html, rss/atom, etc)
- Good recommendations, *as long as it’s the problem*

Caching (client, server)

- Can also implement many forms of caching
  - Reduce number of page requests, speed them up
- Some caching is client-oriented
  - Controlled from server code (see previous resources)
  - May also be configurable in client
- Some caching is server-oriented
  - In app server, or via ehCache, memcached, redis, etc
  - See my list at cf411.com/distcache

Caching (client, server, network)

- Can also cache content “on network”
  - So requests from clients are handled “before” server
- Content Delivery Networks (CDN's)
  - Akamai, Cloudfront, Lightspeed, etc
  - See my list at cf411.com/cdn
- Reverse proxies
  - squid, varnish, etc

Automation

- Some tools can ease optimization task
  - Some are tools you use as developer
  - Some are server-side automation
- Some frameworks perform optimizations
- Server-side optimization
  - mod_pagespeed, iispeed, etc
- See also services like Cloudflare, Squixa
- Development automation tools may optimize also
  - grunt, maven, ant, rake, make, etc

Testing your solutions

- So you’ve implemented solutions. How to test?
  - First, lather/rinse/repeat page testing solutions earlier
  - Recall also bandwidth throttling simulation tools
- Also consider load testing
  - Jmeter, loadimpact, loadstorm, others
  - Some can simulate different clients, bandwidths
  - See my list at cf411.com/loadtest

Watching for problems over time

Many tools to help

- You’ve done your optimizing, testing
  - How can you know if problems crop up later, over time?
- Many tools to help
  - pingdom.com
  - siteuptime.com
  - monitor.us
  - newrelic.com/synthetics
  - zoompf.com/alerts
  - See my list at cf411.com/sitemon

In Conclusion

What we’ve learned

- Range of performance challenges
  - On client, network, server
  - And many parts within those
- Range of tools/resources to spot/solve problems
- Tools to watch for ongoing problems over time
- Gave lots of tools, links
  - but also direction, *perspective*
- Hope you feel empowered to go solve problems!
- Feedback welcome, on twitter, email, etc