Loudon &
Company

Better Decision-Making: Drupal Performance

May 17th, 2014

The problem space

Performance, Scaling, & Optimization focuses on:

Latency - delay, e.g. length of time for a single request

- Experts say 2 second page loads [1]
- Measure with tools.pingdom.com or webpagetest.org
- Example issue: search page is slow every time I use it

Throughput - number of reqs per time, often RPS or RPM

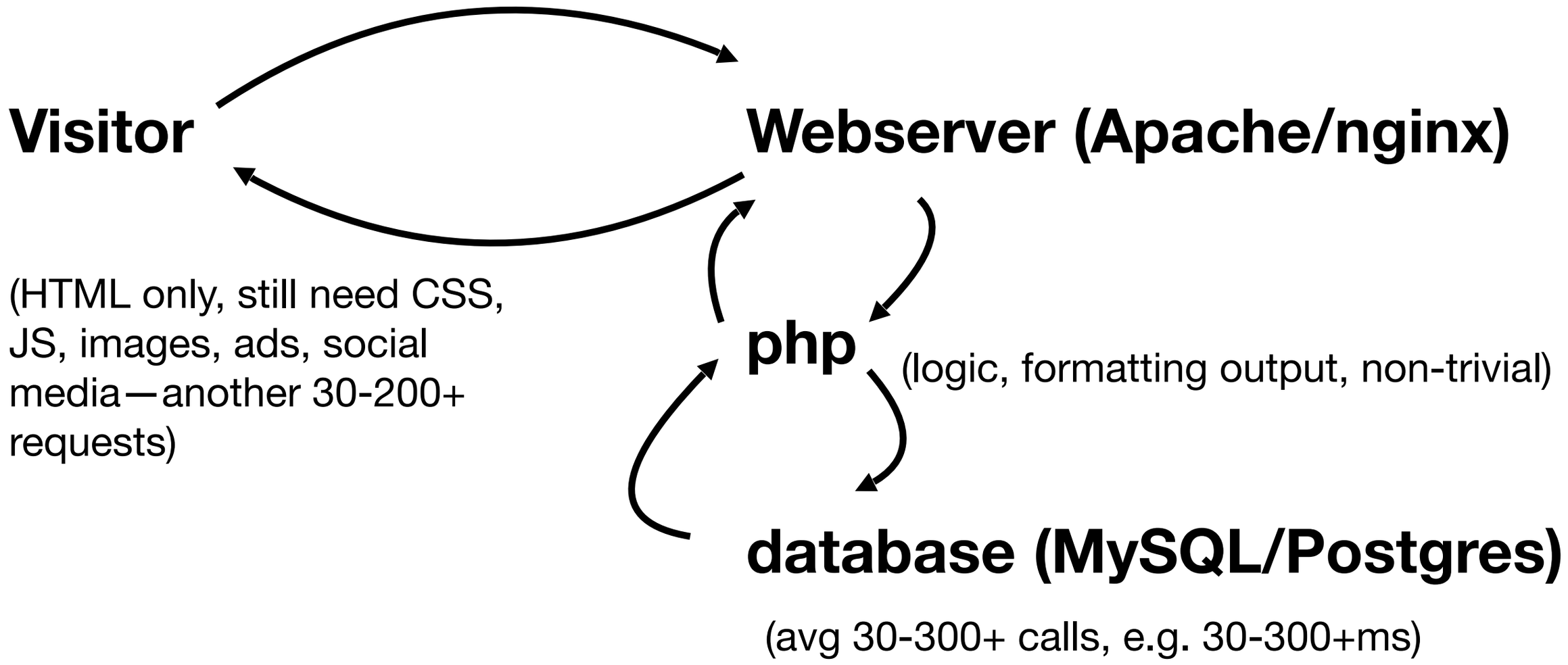
- Experts say YMMV and “here be dragons” :/
- Measure* with Apache Bench, Nouvola, BlazeMeter, JMeter, etc
- Example issue: site is slow on Friday nights

* Apache Bench is fine for quick ’n dirty anonymous, other tools are for auth users.

But it’s really all about (reducing) work

What work* do servers do?

(why is it slow?)



* Does not include DNS (10-300ms) or factor in network time

Constraint to just hiring it out

Consultants are expensive*, why?

- Sr. devs only, (Drupal + server knowledge)
- Unique: your data + biz needs + code + server
- Iterative and time-consuming (make changes and test/ verify fixes + troubleshooting!)
- It’s “art”, “black magic”; limited public knowledge: discussions often devolve into BMW vs Mercedes not Porsche vs Econoline Van => lack prioritization

* Assume $50-200/hr and a $1,000 minimum

What to do?

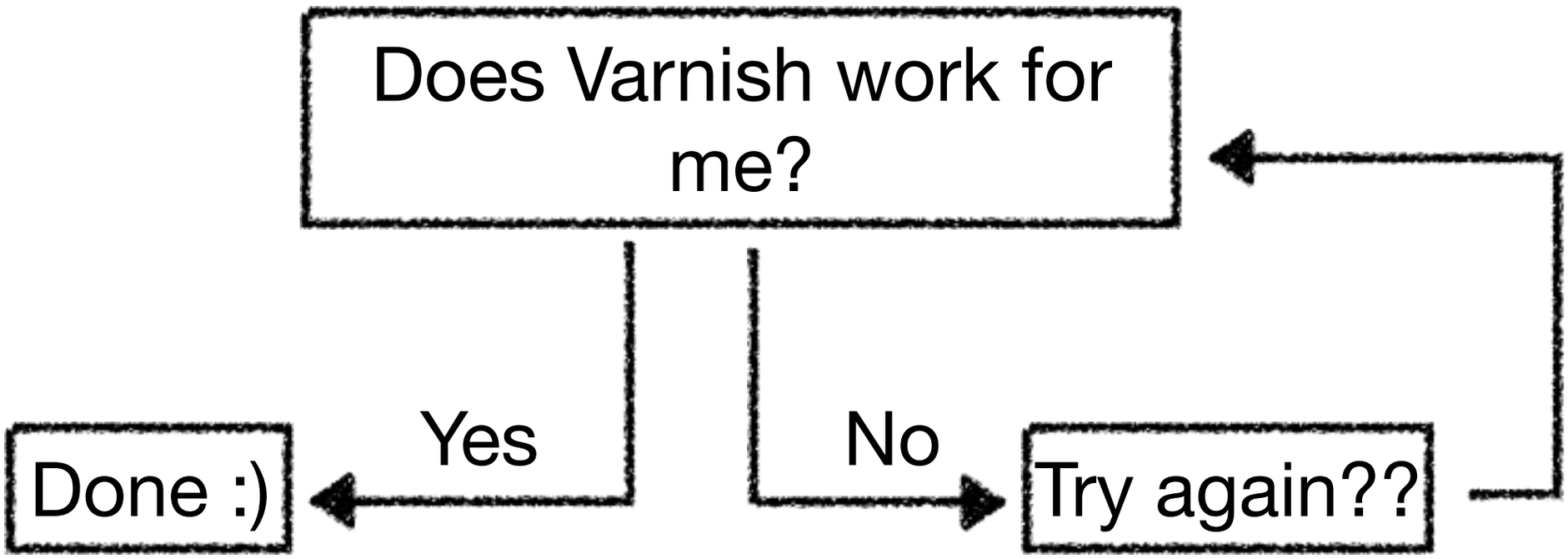
First steps

- **DIY** (low-hanging fruit) => modules: site_audit, css/js aggregation, page caching, css sprites, entity_cache, views_content_cache, block_cache_alter

- **Bigger/better/more hardware** [2]

- **Host w/ Acquia or Pantheon** (“fast” is baked-in). Great value! But expect— $5k+/yr, can be $20k+/yr => do your due diligence.

Need more?



I’m only kind of joking. See the Varnish mini-book [3] for caveats/workarounds. Why? Because you skip all the php/MySQL work and serve directly from RAM or disk? (The boost module would also work in the same way.)

Authenticated users?

Common recommendations

- Memcache - replaces database caching for faster lookups, eases DB bottleneck— requires sys admin
- Authcache module - offers great potential, but can have major architecture changes and require a dev
- APC/Opcache - makes PHP faster, sys admin required
- CDN - faster files (CSS, JS, images): good for global/ wide audiences; files aren't your bottle-neck*

* It *could* be the problem, but files are 10-100X faster than php

Context: latency case-study

- Simple example, not meant to be definitive or scientific*; just so we are not talking in a vacuum

- Site on-hand, Drupal 7, no custom modules, 120 enabled modules—built by FTE experienced devs (but not me); looks well-built; estimated cost $40k

- Sample page, not complex. It's an "about" page, but could be a blog post or similar—a couple of sidebar blocks with "featured content", etc

- Dev server: 2GB RAM/2 CPU, many other processes

* Stay tuned to http://loudonco.com/blog, I may do something on this front near-term.

Anonymous users

The happy path

Page Cache: 1.3-2.3s, ~90 reqs, **HTML @ 300ms**

No Page Cache: 2.9-3.9s, ~90 reqs, **HTML @ 1.5-2s**

No CSS/JS Agg: 1.5-3s, ~135 reqs, **HTML @ 300ms**

Varnish: 1.3-2s, ~90 reqs, **HTML @ 150ms**

Authenticated users (warm cache)

logged-in: 2.5-3.1s*, 70 reqs, **HTML @ 1.5-2s**

block caching: 2-3s, 70 reqs, **HTML @ 1.5-2s**

advagg: 2-3s, 58 reqs, **HTML @ 1.5-2s**

views caching: 2-3s, 70 reqs, **HTML @ 1.5-2s**

memcache: 2-3s, 70 reqs, **HTML @ 1.5-2s**

apc: 2-3s, 70 reqs, **HTML @ 1.5-2s**

bootstrap.php [4]: **0.8-1.3s,** *you can't be here for 2s page!*

* This is actually pretty good, I would probably live w/ 3-5s latency but see [1].

Takeaways

- Identify performance goals upfront— latency vs throughput.

- Prioritize against business needs! Don't chase minor performance gains—be holistic and strategic.

- Make sure you have the easy stuff covered! Page caching, css/js aggregation.

- No magic bullet—a big cut percent-wise (100-200ms) is *not going to be noticeable*. Oftentimes, many techniques are needed to achieve the desired result.

Additional Notes

- Many technologies are for *scaling* not reducing noticeable latency for an individual visitor.

- You can offload slow Drupal work to other processes/services—Disqus, Apache Solr, etc

- Caching is awesome/complex—entire books on it.

- DIY Resources:
  - Drupal High Performance Group, ask questions :)
  - Drupal Performance book, (for devs/sys admin only?)

- Thanks! Please send feedback to tim@loudonco.com :)

Footnotes

1 Kissmetrics page load time infographic

2 Popular and affordable hosting from Digitial Ocean

3 Drupal + Varnish mini-book (free)

4 Script to measure bootstrapping, see code comments. Gives a baseline of authenticated user without menu routing, php logic, or theming overhead for HTML