

Server Architecture for High-Performance Drupal

Robert Ristroph

rgristroph@gmail.com

@robgr

<http://www.drupalcampphoenix.com/high-performance-server-architecture>

Outline

- What is performance ? Scaling ?
- Overview of the standard architecture
- Economics masquerading as tech
- Avoid Drupal or PHP as much as possible
- Unsolved Problems
- The Future – Connecting Systems

What are you Optimizing ?

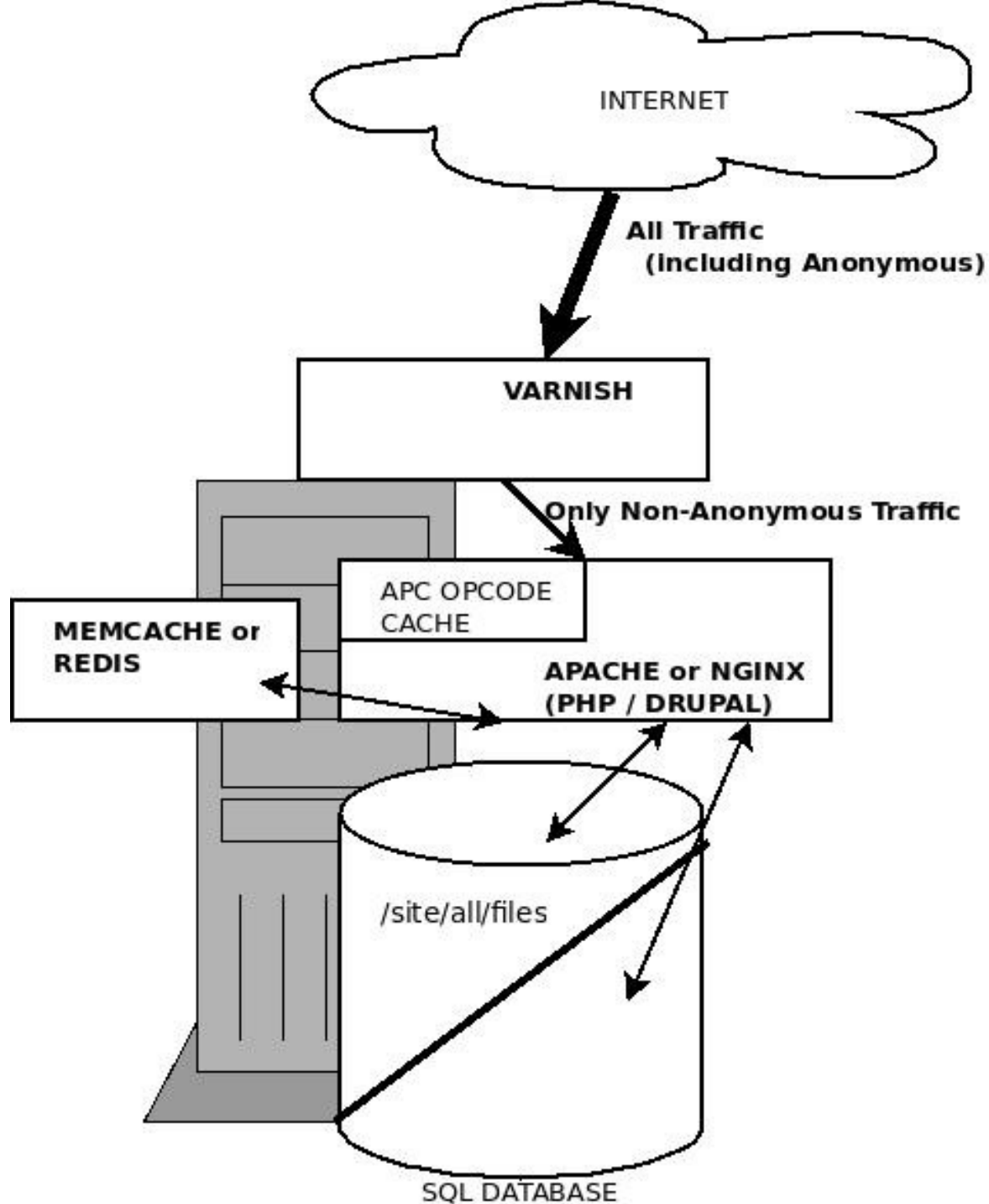
- One site fast vs. many sites cheaply
- One page view fast vs. many page views at once
- Page load time vs. availability, accuracy, age
- User created content vs. read-only
- All questions of how to spend money and time
- You need to be able to optimize for different things for different projects

Drupal and Scaling

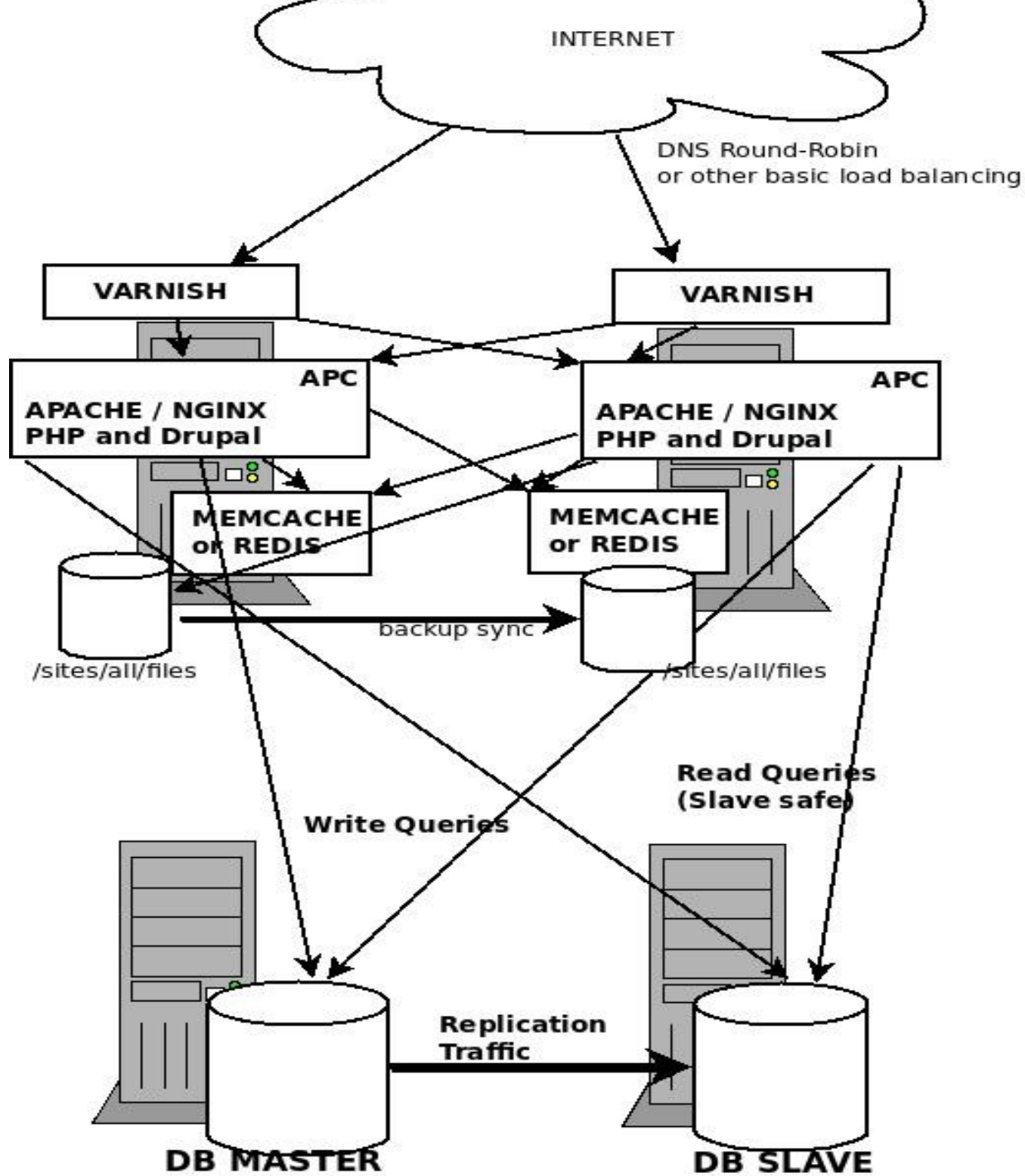
- Good standard PHP practices, caching strategies, etc
- Codebase fairly good, quality of contributors high, they know and care about scaling tradeoffs
- But flexibility / extensibility balance speed and size, often win
- D7 uses more memory than D6

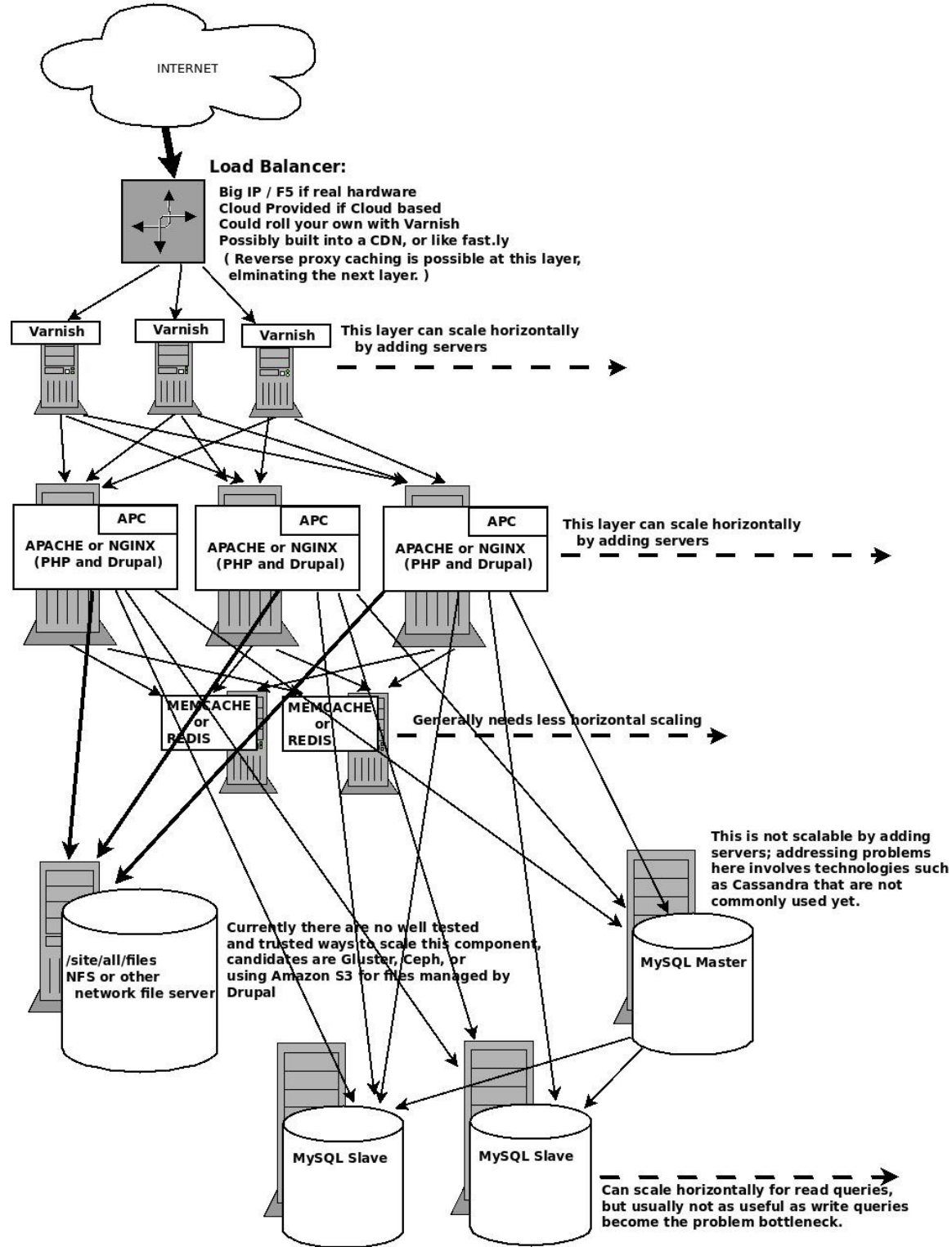
Architecture Diagrams

- One Server
- Small Cluster
- Big Cluster



A Single Server with a High-Performance Drupal Stack





Large Scale Drupal Cluster in Cloud or Hosted

Similar to High-Perf Wordpress

- Hart Hoover (@hhoover, www.harthoover.com) presentation from Texas Linux Fest 2012
- <http://www.slideshare.net/harthoover>
Rackspace / WordPress expters
- WPENGINE (high-performance WP hosting)
also runs an architecture very similar

“Scaling PHP” by Steve Corona

- <https://www.scalingphpbook.com/>
- Steve Corona lead the scaling of twitpic during it's insane growth
- Not Drupal specific at all
- The particular application avoids some of the harder problems – filesystem can be split, transactions not linked

Pain Point: Server Config Management

- Many servers – consistency, quick deployment
- Different roles for different servers
- Problems that occur only when they hit a particular server are hard to debug
- Chef
- Puppet
- Ansible (<http://ansible.cc>) - simpler for small clusters ?

Pain Point: Complicated Cache Rules

- Varnish – excluding cookies, per-page rules, handling languages
- Flushing an entry from Drupal
- Handling a cold start under load
- SSL – exclude from cache, decrypt in pound (<http://www.apsis.ch/pound/>) or maybe Varnish

Pain Point: Clustered Filesystems

- Common problem for everyone hosting big clustered Drupal
- Drupal presumes filesystem is fast
- Filesystem used for concurrency and locking, not just persistence
- GlusterFS – closest to standard, receiving attention from RedHat
- Stream wrappers to S4
- Ceph, many other candidates
- Pantheon wrote a cassandra backed FS

Evolution

- Nginx (replacing Varnish?)
- Redis instead of Memcache
- ESI with Varnish
- NoSQL for specific uses

Unsolved

- Clustered filesystem
- Large amount of DB write traffic
 - Especially if transactional, say a flood of people buying a limited item

Current Cutting Edge

- Architectures that are a “matrix” of servers
- Heroku, Pantheon, OpenShift (RedHat)
- Use nginx, systemd, cgroups
- Complicated routing of requests
- Distributed / Clustered filesystems still an issue

Beyond This Architecture

- Big Drupal sites more and more connect to APIs, and provide APIs
- Drush commands / DrupalQueue for offline processing
- More “services” based architectures
- More small, quick components implemented in node.js or similar
- More OpenStack / OpenShift private infrastructure, we will have to learn how to optimize that

Next

- We should work on an automated performance testing framework
- PHP itself – particularly size of code in RAM
- APC – seems like it could use more attention, maybe steal ideas from JVMs
- Re-examine Postgres?

A Note on Debugging

- Solving scaling problems can seem like a black art if you cannot replicate the problem
 - Wget spider scripts, apache ab, LoadStorm
 - It's not hard, if you can replicate then you can debug as you would any problem
 - Ok, simulating some kinds of logged in transactions at scale can be hard . . .

Questions ?

Robert Ristroph

rgristroph@gmail.com

@robgr

[http://www.drupalcampphoenix.com
/high-performance-server-
architecture](http://www.drupalcampphoenix.com/high-performance-server-architecture)

Pressflow

- Scaling and Performance enhancements that could not be folded back into Drupal
- User interface and API compatible
- Gave up PostgreSQL, other features
- Allows reverse proxy (Varnish)
- All improvements are in Drupal 7 now
- Not much need for PF7, perhaps it will hold backported D8 features in the future