# Overview of Drupal Performance


## Robert Ristroph

Email & AIM: robert@fourkitchens.com

Google Chat: rgristroph@gmail.com

Twitter: @robgr


Slides: http://2012.badcamp.net/program/sessions/high-performance-drupal-server-architecture

FOUR KITCHENS

# Outline

- What is performance / scaling ?
- Overview of standard big architecture
- A big bag of tricks – economics issues masquerading as tech
- Punchline: Avoid actually using Drupal or PHP as much as possible
- Unsolved problems
- The Future – Connecting Systems

# Scaling What ?

- One site fast vs. many sites cheaply
- Anonymous vs. logged in traffic
- User created content vs. read-only
- Page load time vs. availability, accuracy, or age of content (high availability)
- Flexibility and Complexity
- You need to be able to tune for different metrics
- Monitor over time, archive results

# Inside Drupal

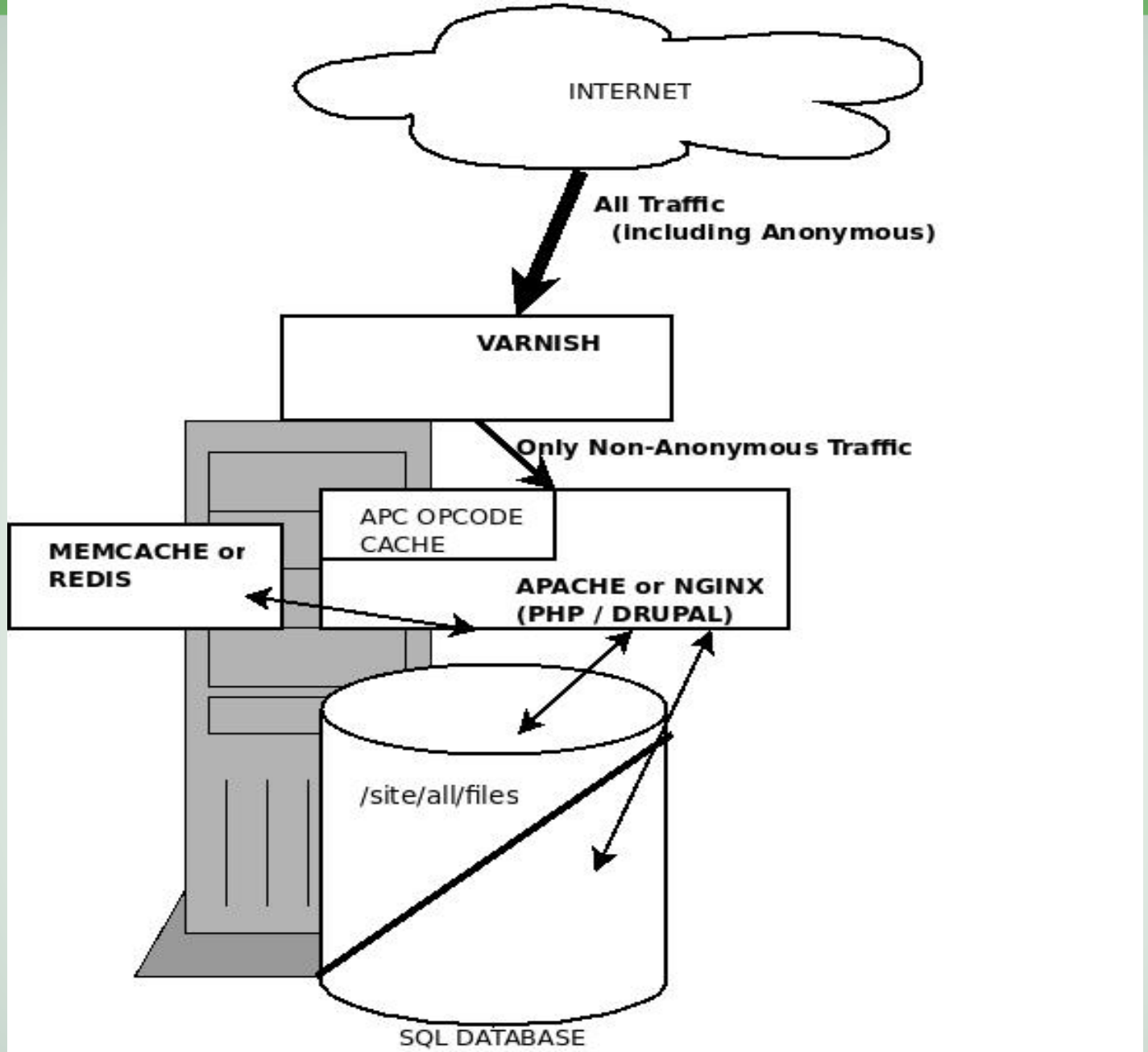
- Standard PHP practices (static caches, etc)
- Codebase fairly good, quality of community contributors is high
- Flexibility, extensibility, other factors balance speed and size and often win
- Drupal 7 uses more memory than Drupal 6
- Pressflow – more scalable fork of Drupal (mainly 6)

# Pressflow

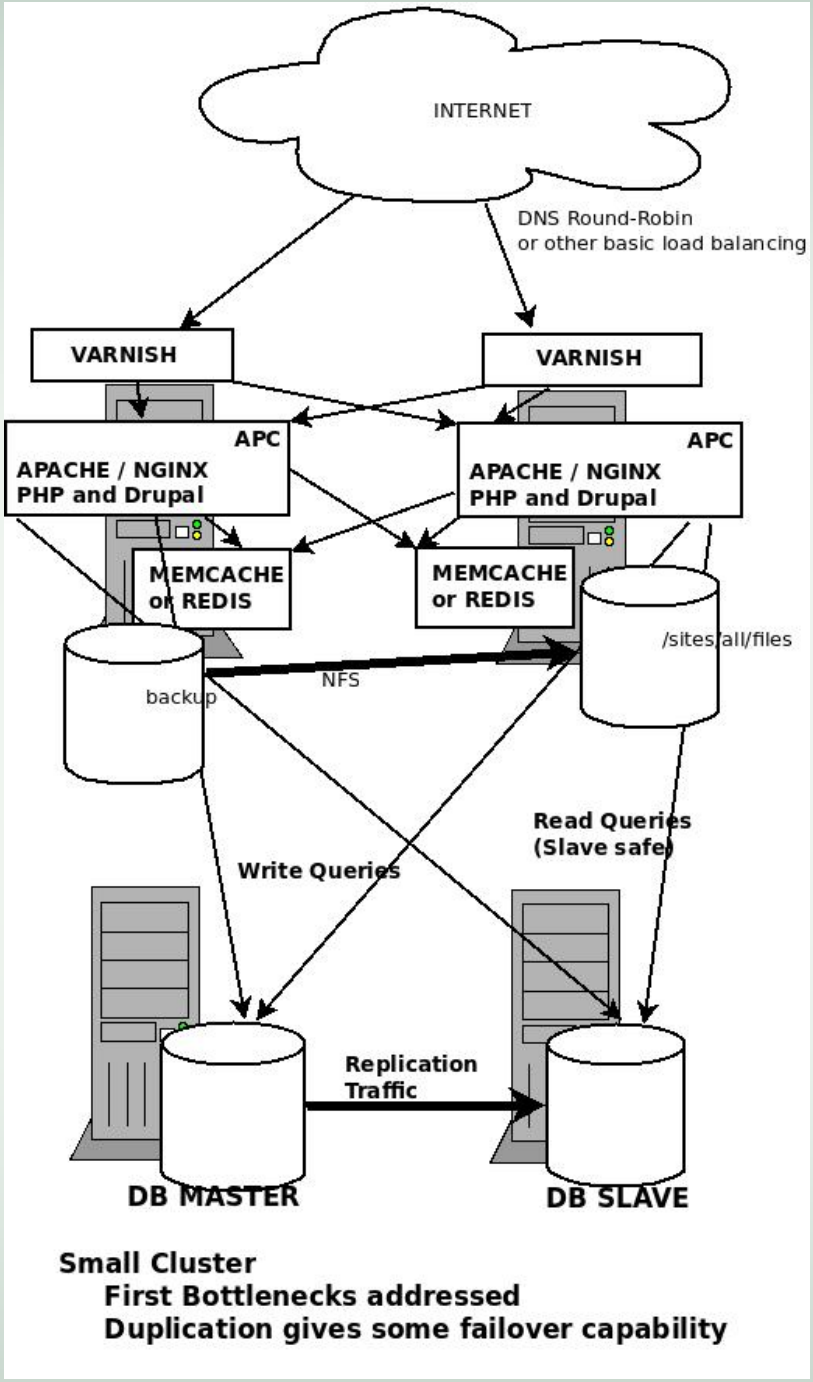
- Scaling & performance enhancements of Drupal 6
- User interface and API compatible
- Gave up PostgreSQL, some other features
- Allows for reverse proxy (Varnish)
- Not much need for PF7 (yet), may end up holding more backported D8 than improved D7

# Architecture Diagrams

- One server
- Small Cluster
- Big Cluster

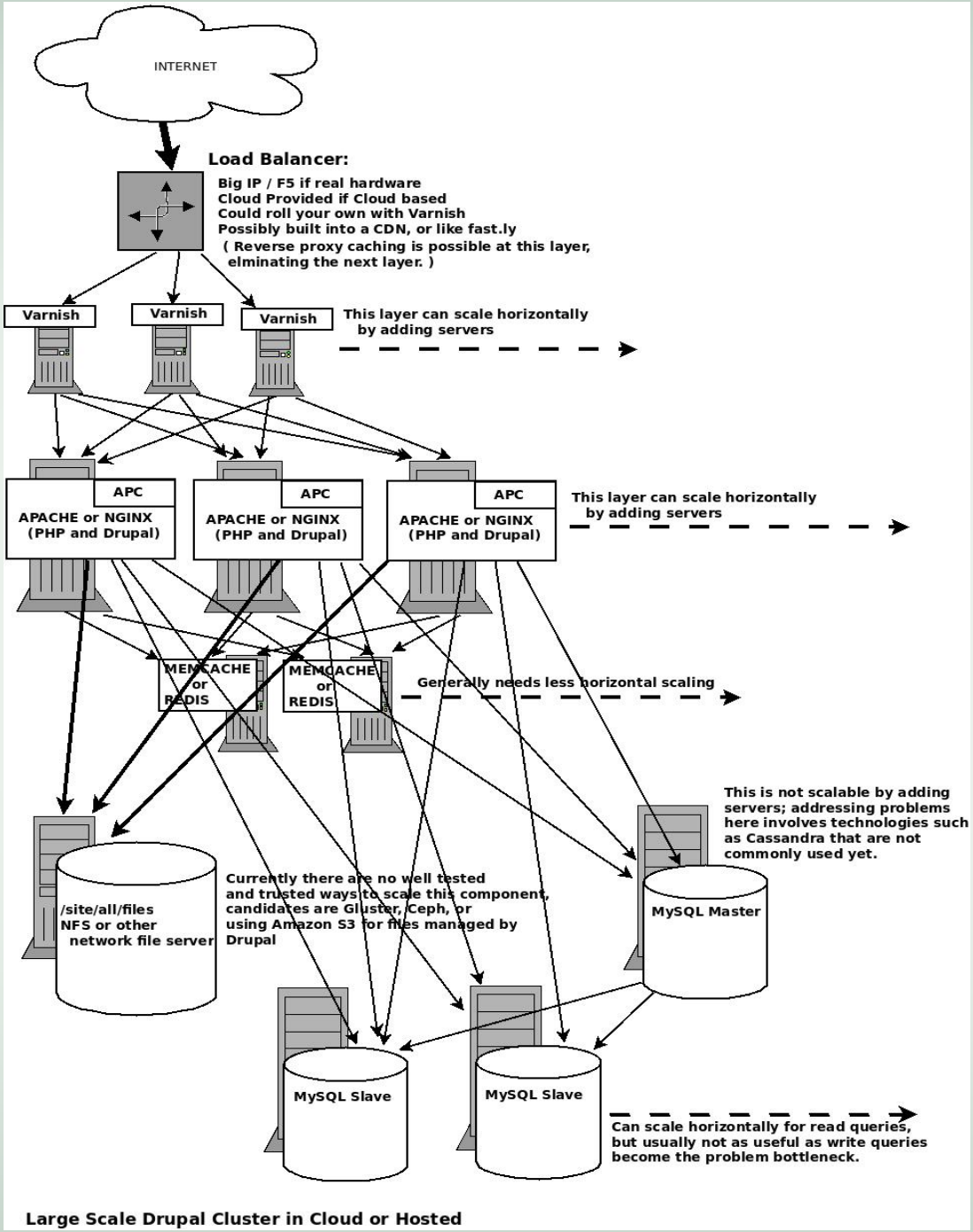



**A Single Server with a High-Performance Drupal Stack**




**Small Cluster**

- **First Bottlenecks addressed**
- **Duplication gives some failover capability**




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/architecting-wordpress-on-the-rackspace-cloud
- Same architecture was arrived at by the Rackspace / WordPress experts
- WPEngine (high-perf WP hosting) also runs a roughly similar architecture

# Pain Point: Configuration Management

- Many servers in different roles require configuration  management
- If you get a behavior that only happens when the request happens to hit a particular server, it's hard to debug
- Chef (most popular, seems to be the future)
- Puppet
- Ansible ( http://ansible.cc/ ) may be simpler for small clusters

# Pain Point: Complicated Cache Rules

- Varnish – excluding cookies, particular pages, handling languages
- Flushing an entry from Drupal
- Handling a cold start of everything
- SSL – exclude from cache, or decrypt in pound ( http://www.apsis.ch/pound/ ) (or maybe varnish)

# Pain Point: Clustered Filesystems

- A pain point with everyone hosting big clustered Drupal
- Drupal presumes filesystem is fast
- Filesystem used for concurrancy, not just persistence
- GlusterFS – closest to a standard
- Stream wrappers to cloud (S3)
- Ceph and many other candidates
- Pantheon wrote a cassandra backed FS

# This Stack is Evolving

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

# Still Unsolved:

- Clustered filesystem
- Large amount of DB write traffic
  - NoSQL solutions

# Current Cutting Edge

- Architecture that is more a “matrix” of servers
- Heroku, Pantheon, OpenShift (RedHat)
- Use nginx, systemd, cgroups

# Beyond This Architecture

- Big Drupal sites more and  more connect to outside APIs, provide APIs
- Drush commands, node.js for background processing (DrupalQueue)
- More “services” based architectures
- Future has more OpenStack private cloud infrastructure, we will have to learn how to optimize that

# Next Things

- We need an automated performance testing framework
- Big sites have a lot of write load; Cassandra ?
- PHP itself – size of code in memory
- APC – perhaps could use more attention
- Re-examine Postgres ?

# Next Things

- We need an automated performance testing framework
- Big sites have a lot of write load; Cassandra ?
- PHP itself – size of code in memory
- APC – perhaps could use more attention
- Re-examine Postgres

# A Note on Debugging

- To many developers, solving performance problems is a black art because they can't replicate the problem
  - Wget spider scripts, apache ab, LoadStorm, (session tomorrow morning)
  - It's not hard, if you replicate you can debug as you do everything else
- Try out XHProf module

we make BIG websites

# FOUR KITCHENS