Site Performance, Optimization and Scalability

Alan Dixon http://blackflysolutions.ca/

Khalid Baheyeldin http://2bits.com

DrupalCamp Toronto 2011

Agenda

- Introduction

- Symptoms and Tips

- Hardware and Software

- Drupal

- Reverse Proxies and CDNs

- Other

- Discussion

About Khalid

- 25+ years in software development and consulting
  - Mainframe, UNIX, Linux, LAMP
- Drupal since 2003, develop, contribute, consult
- Several contributions to Drupal core
  - Site maintenance, hook_watchdog()
- Over 37 modules contributed on drupal.org
  - Userpoints, Nagios, SecondLife, Favorite Nodes, Flag Content, Nodevote, ...

About Khalid (cont'd)

- Volunteer on and off with various teams (webmasters, infrastructure, security, ...)

- Member of the **Advisory Board** and **Nomination Committee** of the **Drupal Association**

- Co-Founder of the **Waterloo Region Drupal Users Group** (Southern Ontario, Canada)

About 2bits.com

- Founded in 1999, Drupal since 2003

- Services

  - Drupal / LAMP performance optimization and tuning

  - Server provisioning for performance and uptime

  - Manage large sites for clients

  - Drupal custom module development

- Clients from all inhabited continents on the planet

- In depth articles, and testimonials at http://2bits.com

Symptoms

- Site "locks up" under load ...

    - Excessive swapping under load

        - reduce memory usage

        - add memory

    - Slow queries tying up the disks with temp tables

- Bloat

    - Many services running (DirectAdmin, Courier, FTP, Tomcat, Apache modules)

Symptoms

- Aggressive crawlers
  - URL structure is messed up (e.g. &quicktabs_1=2 added to lots of pages)
    - Check Google Webmaster, see if you have over a million pages indexed!
  - Robots.txt (Most crawlers ignore it …)
  - Google Webmaster to ignore parameter
  - Vhost changes
  - Use "go access" tool to analyze logs

- Large database tables
  - Left over stuff from when you used cache_* as the cache tables. Found 50,000+ rows, and over 574MB. Mostly 404s!
  - Watchdog? Accesslog? History?
  - Harder to cache in memory
- Locking on the table level?
  - MyISAM -> InnoDB

Tips

- Boost

  - Use it on small sites only, and shared hosts

  - For VPS or dedicated, use memcache, and/or varnish

  - Can be a bottleneck itself (e.g. When posting a comment, cache is cleared, and the disk is tied up for 20 seconds)

Tips

- Network calls

  – Avoid network calls from PHP as much as possible

  – On a site where each node displayed 4 links (Twitter, Facebook, LinkedIn, ...etc.), and used service_links with TinyURL.com!

  – View with 50 articles it took 68 seconds!

Tips

- Site slow when cron runs?

  – Use Elysia cron to stagger different cron hooks, and make them less frequent

  – Set it to run every minute

- Site sends lots of mail?

  – Use Queue Mail module to batch them in cron

Tips

- Taxonomy Browser module
    - Use a form_alter to limit the number of terms to search on.

- Fast 404s
    - Save resources by a simple modification to settings.php

Tips

- Node Access modules

    - Organic Groups, TAC Lite, ...etc.

    - Extra joins (slow queries?)

    - Watch for how many joins, and how many rows

- Free tagging?

    - Watch for how many rows are in term_data and term_node tables.

    - Extra joins again (slow queries?)

Tips

- Use syslog module instead of dblog
    - Reduces load on the database

- PHP 5.3?
    - Turn off notices
    - error_reporting = E_ALL & ~E_NOTICE

Technologies

- Mercury/Pantheon
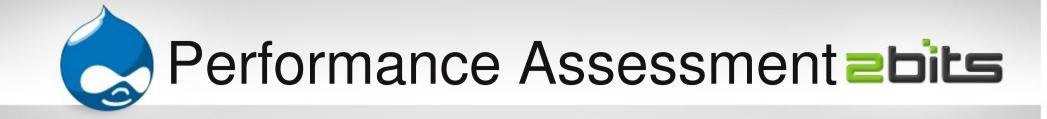- Pressflow
- AdvCache
- AuthCache
- Nginx/Lighttpd
- PHP-fpm
- CDN
- Varnish/Squid
- MongoDB/Cassandra
- Amazon Cloud
- Server cluster/farm
- Solid State Disks

Surprise!

None of the above ....

Performance Assessment

- A regular service that we do for clients facing issues, or about to launch a high traffic site.

  - Load testing

  - Bottleneck identification

  - Analysis

  - Recommendations

- Can be "Targeted" to one or a few symptoms, rather than all that we can find.

Refactoring code

- Removed all unnecessary modules

  – Less code to load/execute

  – Less memory to consume

  – Less database queries

  – Less maintenance and security concerns too

- Replaced CCK with a custom module (one content type with a few fields)

- Teach client that the theme is not for business logic, just presentation

Hardware

- As many cores as you can (64-bits)

  - Better many slower cores than only two fast ones

- The more RAM the better (8 or 16GB for large sites with dedicated servers)

- Separate Disks (root, logs, Drupal, database)

  - Don't put 12 VMs on a single physical machine (disks are still shared!)

Software

- Apache

  - MPM Worker (threaded, far less memory for static content)

  - Apache fcgid (less memory, less MySQL connections)

- PHP

  - FastCGI (stable with fcgid)

  - APC 3.0.19 (or later, installed via pecl)

Software

- Munin for monitoring, and historical resource usage
    - Why did the graph change?
    - What does that mean?

- Awstats for statistics
    - And comparison with Google Analytics

Drupal

- How many modules?

- 3.4 million page views on a single server: Only 42 modules enabled, and 2 themes

- 100,000 pv/d and 200+ modules, and lots of issues

Simplicity vs. Complexity

"Simplicity is a prerequisite for reliability"
-- Edsger Djikstra 1975

"Complexity is a disease"
-- Dries Buytaert, May 2006

Chronic Featuritis

- Defined as:
  - "The pathological condition of the site's owner(s) having the burning desire to cram every feature available for download as a module on drupal.org, and asking for more features and new modules, in the hope that the site will be the next Facebook or Twitter ...

KISS

Keep It Simple Stupid ...

It pays off in the long term ...

Monitoring

- Munin

    - CPU, disk, memory, load, I/O, Apache, MySQL, memcache, ...etc.

    - We added scripts for number of logged in users, and anonymous users

Resources

- Many articles on performance, tuning and optimization http://2bits.com