Drupal Performance and Scalability

What you need to know ...

Khalid Baheyeldin

http://2bits.com

London (Ontario) Drupal Users Group

About Khalid

- 27 years in software development and consulting
- First computer: Sinclair ZX Spectrum
- Experience: Mainframe, UNIX
- Open Source: Linux, LAMP
- Full time open source developer, contributor and consultant

About Khalid (cont'd)

- Drupal since 2003
- Core features as well as 37+ contrib modules
- Member of the Advisory Board of the Drupal Association
- Co-Founder of the Waterloo Region Drupal Users Group
- Drupal talks at DrupalCons, DrupalCamps, DUGs
- Also Ontario Linux Fest, KW LUG
- Google Summer of Code (2005 to 2010)

About 2bits.com

- Founded in 1999 as a partnership, incorporated since 2007
- Drupal since 2003
- Services
  - Specialize in Drupal scalability and performance
    - Performance Assessment (various flavours)
    - Hosting provisioning, tuning and management
    - Drupal custom module development
- International clientele
- Extensive in depth articles and testimonials at http://2bits.com

Performance vs Scalability

2bits

- Performance:
  - Less time to serve a request (faster)
  - For example: 300 ms vs. 700 ms
- Scalability:
  - More capacity and throughput (more requests per hour/day)
  - Handling more simultaneous requests

High Availability

2bits

- Reduction of down time
- Different methods, depending on requirements and tolerance
  - By horizontal scaling, and a load balancer
    - Susceptible to data center outages (RackSpace and Amazon!)
  - By fail over across data centers
    - A few minutes of down time, but immune to data center wide outages

High Availability (cont'd)

2bits

- Measured in 9's
  - Two 9's (99%) = 3.65 *days* per year
  - Five 9's (99.999%) = 5.26 minutes per year
- “The more 9's you want in your uptime, the more 0's you need add to your budget”

Define your Goals

2bits

- Performance?
- Scalability?
- High Availability?
- All of them?
- For $20?

Hardware

- CPU
  - The more cores the better ...
- Memory
  - Minimum 512MB, 1G or more recommended (for memcache), yet more is better (8GB for dedicated)
- Disk
  - Try separating operating system, web root, and database on different disk spindles
- Stay on a single server for as long as you can, simpler environment, less headaches

Hardware Caveats

- Virtualized Wrong!
  - 12 Virtual instances on one hardware server, all sharing same disk!
  - Boost used for caching: 12 seconds (20 seconds to 8 seconds)
- Underpowered (Amazon EC2 low end instances)
- Virtualization technology (Xen better than Virtuozzo)

“Cloud” caveats

2bits

- Usual “Cloud” pros and cons
- Disks cannot be virtualised (bad neighbors? Can't be sure)
- Variability in performance (good, good, sloooow, good, ...)
- Gets expensive as you grow
- Amazon AWS downtime (3 major incidents in 2012, so far!)

Linux Distro

2bits

- Use a stable Long Term Release (LTS) release
- Whatever your staff knows best (CentOS?)
- Be a minimalist and use the repositories
- Ubuntu 12.04 LTS is quite good
  - No need to install APC using PECL, package available
  - Except for: Varnish oddities
  - Install from Varnish project's repository (article on our site)
  - Or, compile from source (less desirable)

Web Server

2bits

- Apache MPM prefork, with PHP as mod_php5
  - Most common configuration
  - Has to be front ended by something else for static files (Varnish or nginx), to avoid large number of processes
- Apache MPM Worker (threaded)
  - PHP has to be in Fast CGI mode or PHP FPM (our current favorite)
- Nginx
  - Use microcaching (caching for 5 seconds)

MySQL

2bits

- Make sure that the query cache is enabled
- Use 5.5 if you can
  - Can log slow queries less than 1 seconds (e.g. > 100 ms)
- InnoDB file per table enabled, before you import the database
- Start with defaults, then tune from there
- Temporary tables to tmpfs helps

PHP

- 5.3.10 minimum for Drupal 8 (Ubuntu 12.04 version)
- mod_php5
- FastCGI
  - APC code cache is **not** shared, duplicated per process
  - mod_fcgid, preferred
- PHP FPM
  - APC code cache **is** shared
  - With Apache, have to install and configure mod_fastcgi
  - With nginx

APC

- Op-Code Caching, but *never* as a data cache!
- Always install and enable it!
- Share the code base for multiple sites
- Shared cache only with mod_php5 and PHP FPM
- Each process has its own code cache if using FastCGI
- Make sure you don't have fragmentation, otherwise it will slow down your site
- Version in Ubuntu Server LTS 12.04 is good enough (no need to use PECL)

Drupal Best Practices

2bits

- Separation of code (modules), data (database) and presentation (theme)
- Ignore them at your own peril, performance (and maintainability) will suffer
- One client had all the application logic in the theme layer, followed by exit(0);
  - Nothing was cached!

Fast 404 for static files

2bits

- By default Drupal handles 404s, even if the request is for a static file (.jpg, .png, .css, ,js, ...etc.)
- This is a full bootstrap of Drupal
- For Drupal 7.x, there is a fast 404 for static files that exits early, avoiding doing the bootstrap
- Enable it in settings.php
  - Adjust these $conf variables (usually not needed)
    - `404_fast_paths`
    - `404_fast_paths_excludes`
  - Uncomment: `drupal_fast_404();`

Redirects

- Redirects are not cached, and requires Drupal to boot to some extent, hence expensive (few 100s of ms)
- If a site constantly re-writes the URLs, it will suffer
- An example
  - Site using purl + context + spaces in a way that
  - e.g. /tv/title becomes /entertainment/title
  - 30% of requests were redirects
  - Killed the site when heavy traffic came by (link from Yahoo's front page, or Drudge Report)
- Try to push redirects to the web server (.htaccess or vhost)

Network calls

2bits

- Emailing 20,000 users? Getting a short URL dynamically?
- A PHP process will block until it gets a response from the other host
- Can add 100s of ms per request
- Can produce “pile ups”
- Avoid them if possible, or use sparingly
- Try to cache the info, if feasible
- Use queues

Media Streaming

2bits

- Ties up resources, specially for slow clients
- A few of these, and you have several web server and PHP processes tied up
- Again, can produce “pile ups”
- Use a third party video service, or CDN
- Or, from a separate machine

Number of Modules

2bits

- Watch out for the number of enabled modules, the less the better
  - Performance
  - Maintainability
  - Complexity
- Actually, number of PHP files loaded per request
  - Via: `include, require, include_once, require_once`
  - Themes and theme files count too ...
  - Modules that do their own extensive includes (views, ctools)

Module caveats

- Modules that disable the page cache
  - Examples: Flag, Invite, Biblio, CAPTCHA
  - `$conf['cache'] = 0;`
  - `$_GLOBALS['cache'] = FALSE;`
- Modules that create session variables
  - Drupal 7.x, Pressflow 6.x, but *not* Drupal 6.x
  - `$_SESSION['foo'] = 'bar';`

Caching

- Caching method
  - Core provided (database tables)
  - Memcached
  - External: Varnish
- Page Caching
  - For anonymous users
- Block caching
- Views Caching is your friend (both query & output)
- Custom Caching
  - Your module does cache_set() and cache_get()

Memcache

2bits

- Memory caching daemon
- Well tried and tested
- Recently simpler (single bin)
- Can be replicated on two machines
- Often sufficient on its own
- Has the benefit that hook_boot() and hook_exit() are executed
- 20-30 milliseconds for a cached request

Boost

2bits

- Useful for mostly static sites, with anonymous visitors
- Creates bottlenecks if the site is busy (lots of commenting) or having lots of logged in visitors
  - Disk contention, when invalidating caches for nodes, ...etc.

Varnish

- Reverse Caching Proxy
- Front ends your web server, and caches content
- Caches static files by default, needs extra configuration for Drupal
- Kinda finicky, until you get it right
- Consider any .vcl configuration as a starting point
- Keep default.vcl unchanged, and add changes to custom.vcl
- You can see 10 milliseconds response time for a cached request
- Must specify “Page cache maximum age” (Pressflow 6.x)or “Expiration of cached pages” (Drupal 7x.)

CDNs

2bits

- Content Delivery Networks
- Another caching level, outside your infrastructure
- Useful for locality (physically closer to site visitor, faster response)
- CloudFlare is a commonly used one, with a free tier
- Akamai, Amazon CloudFront
- Masks IP addresses in the web server's log (or Varnish)

DDoS

2bits

- Crawlers
- Spam comments
- Harvesters (Beijing Express)
- Misbehaving news clients (Genieo)

Pressflow caveats

- Pressflow 6.x
- Cache set to “External” without Varnish/Squid

Monitoring

- Munin for resource usage
  - Memory, CPU
  - Varnish hits/misses
  - Memcache hits/misses
- Go Access for web server log analysis
  - Which IP hit my site hard
  - Bogus User Agents

Complexity is Bad!

2bits

- Complexity is bad for:
  - Performance
  - Maintainability
  - Troubleshooting
- KISS principle

Diagnosis before Treatment

2bits

- Any treatment that is based on no diagnosis, or the wrong diagnosis, is doomed to fail, and may hurt more than help
- Would you trust a doctor recommending surgery without any clinical examination, lab tests? On your first visit?
- Don't go for the shotgun approach
  - Maybe this, maybe that, oh, this guy says here it may be the other ... etc
  - Time consuming and expensive

Performance Services

- Performance Assessment
  - Site slow?
  - Outages?
  - High Resource Usage?
- Copy of site in our labs

Performance Services

- Server Provisioning
  - Install, Configure, Tune your server
  - Specifically for Drupal (APC, caching, ...etc.)
  - With Monitoring tools

Discussion

Questions?