Drupal Performance

Tips and Tricks

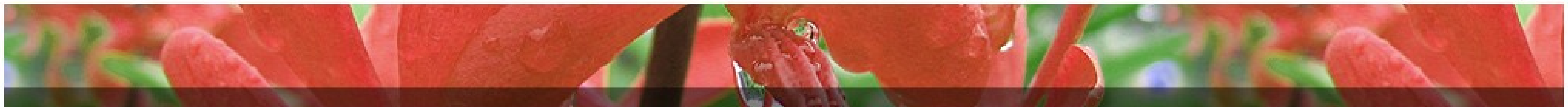
Khalid Baheyeldin

http://2bits.com

Drupal Camp Toronto 2014

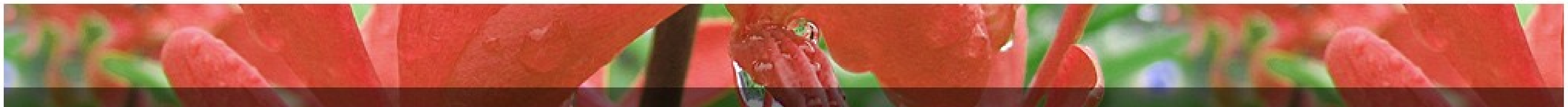
About Khalid

- 29 years in software development and software consulting
- First computer: Sinclair ZX Spectrum
- Experience: Mainframe, UNIX
- Open Source: Linux, Web, PHP, MySQL, Drupal
- Full time open source developer, contributor and consultant



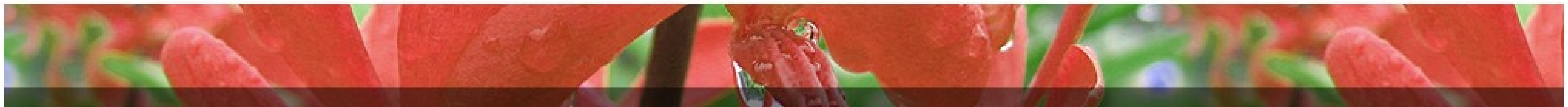
About Khalid (cont'd)

- 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
- Google Summer of Code (2005 to 2010)
- Also, Ontario Linux Fest, KW LUG, Toronto LUG



About 2bits.com

- Founded in 1999 as a partnership, incorporated since 2007
- Drupal since 2003
- Specializes in Drupal scalability and performance
    - Site Performance Assessment
    - Hosting selection, provisioning, tuning and management
    - Custom Drupal module development
- International clients (USA, Canada, Europe, South America, China, ...)
- Extensive in depth articles and testimonials at http://2bits.com

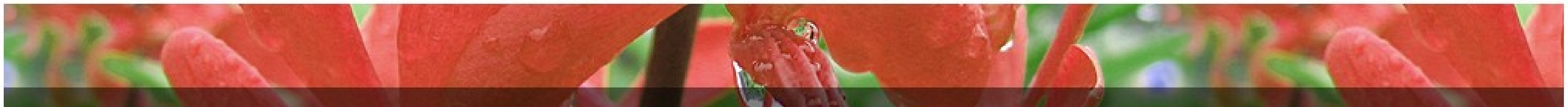


Some Misconceptions

- Drupal is bloated/slow/resource hog ...
- Caching is useful only when users are mainly not logged in
- Memcached is not useful when most users are logged in.
- I need many cloud servers for the site to have reasonable performance

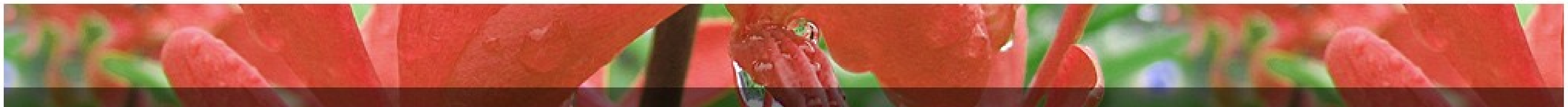
Performance

- How fast (or slow) a single request is
- For example: a backend page load can be:
  - ~ 0.5 seconds (great!)
  - ~ 1.2 seconds (average)
  - ~ 8 seconds (bad)



Scalability

- The ability to successfully handle increased traffic (i.e. X requests over Y timespan)
- Can be:
  - Sudden, due to a link from another high traffic site (famous person tweeted it)
  - Due to time of day, day of week
  - Seasonal (shopping, university registration, summer vacation, ...etc.)
- Most sites have predictable patterns



Performance vs. Scalability

- Good performance is usually a prerequisite for good scalability
- Analogy:
  - Cashier check-out: more total customers, if less per unit of time
  - More service desks, or faster cashiers
- But, speed alone is not enough, there is:
  - Resource capacity (CPU, memory, ...)
  - Limitations (number of I/O operations per seconds, kernel/stack lock contention, ...)

Anonymous Visitors

- Visitors are not logged in to Drupal
- Can serve the same page to different users
- Straight forward with Page Caching and cache_lifetime
- Varnish + memcached/Redis
- ~ 10 to 40 milliseconds
- Watch out for:
  - Modules that use hook_init() and hook_exit(), e.g. Statistics. Will not work with Varnish

Page Cache

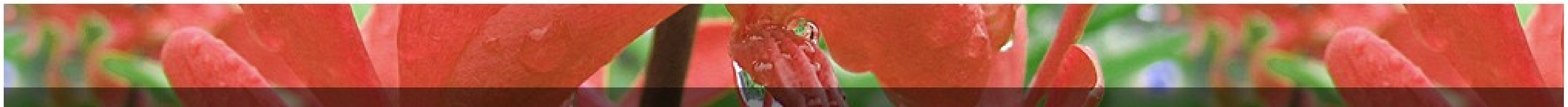
- Some modules have variations of these:
  - `$conf['cache'] = FALSE;`
  - `$GLOBALS['conf']['cache'] = CACHE_DISABLED;`
- Attempt to disable page cache
- Examples: flag, biblio, invite, ...
- Means some pages will not be cached for anonymous users

Memcache

- Make sure you have PHP memcache 2.2.6 or later
- Add this to php.ini, if you use more than one memcached server
  - memcache.hash_strategy = "consistent"

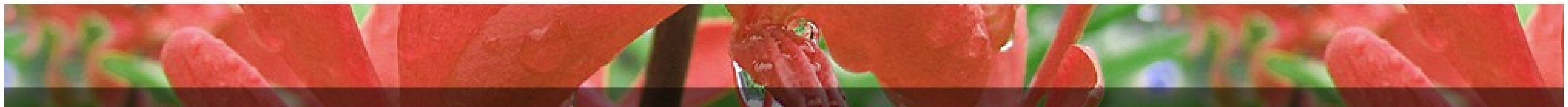
Logged In Visitors

- Visitors are logged in to Drupal (called “Authenticated Users” in Drupalese)
- Cannot serve the same page to different users
- Limited caching, just not for pages
  - Variables
  - Bootstrap (list of enabled modules)
- Memcached or Redis still recommended



Logged In Visitors

- Ideal situation is ~ 150 ms (rare these days)
- ~ 250 ms to 750 ms achievable
- Many (most?) sites are more than that
- “It all depends” on site specifics



Logged In Visitors

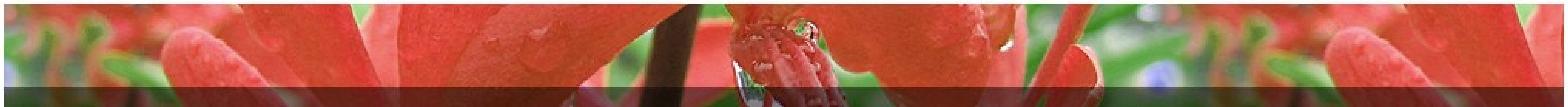
- Because:
  - Caching to the database (default) not memcached
  - Number of enabled modules (open buffet binge)
  - Also, “which modules” and “how they are used”?
  - Number of blocks/panels/...etc, and what is inside each of them
  - Writing on every page load (e.g. Statistics module, dblog)

Modules

- Less is more
  - For performance, but also maintainability, security
- Disable stuff like:
  - devel, table_wizard, schema (development)
  - memcache, memcache_admin (not needed)
  - views_ui, context_ui
  - admin_menu
  - statistics
  - dblog (replace with syslog if possible)

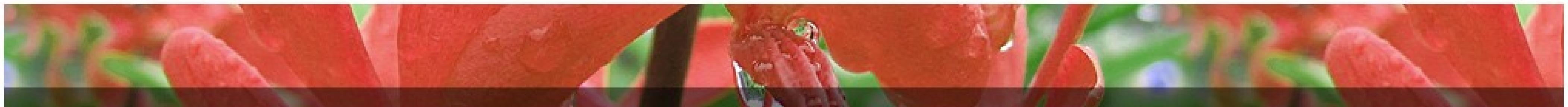
Views

- Make sure views are cached, goes a long way, and many sites forget to do this ...
- Consider reducing number of items displayed in a view
  - Users are not going to read through a list of 50 nodes, 10 or 15 are sufficient
  - Reduces rendering time
  - Reduces memory usage per page



Features

- If you are using features, do not go overboard
  - e.g. A site had 92 features!
- Many of them have a .module that is empty, or only does a require_once() to a .inc.
- Could be combined
  - “mega feature” (cf. University of Waterloo)
  - Creative post-processing script

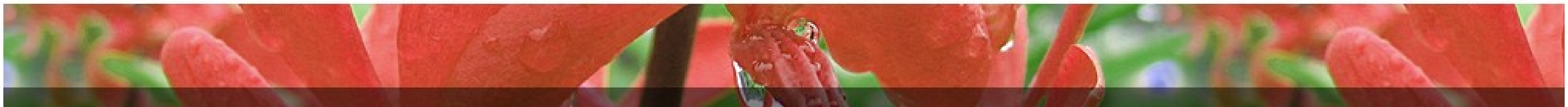


Cron

- Can execute heavy stuff
- Indexing of new content
- Clearing of cache
- Other time consuming and resource intensive tasks, such as sending emails, processing work from a queue, ...

Elysia Cron

- On Drupal 7, set the internal cron to “Never”, rather than the default, and run cron externally
- Regular cron execute all cron hooks on every single run
- Elysia Cron allows fine grained tuning of cron
- How often cron hook for run, per module, and when
- For example: Can defer slow hooks to after midnight, and others every 10 minutes



Network calls

- Happen for Solr search
- Certain social networking sites (depending on module used and how it is configured)
- Bad for performance, specially when you have multiple calls per page load
- Can add up to 300 milliseconds
- Frequent issue with sites we investigate
  - Generating tinyurl short URLs for each node in a list of 50 on a page!

Fast 404

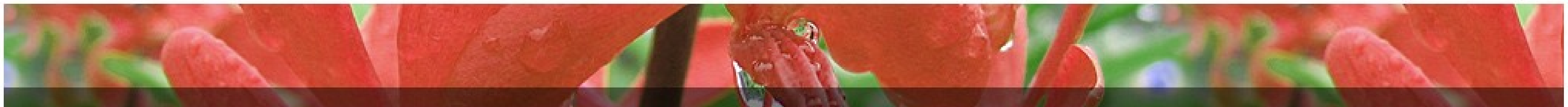
- Built in feature in Drupal 7
- Prevents Drupal from booting for static files (.jpg, gif, .png, .css, .js, ...)
- Uncomment the function in `settings.php`

LAMP Stack

- Ubuntu Server 12.04.4 (until 14.04.1 comes out in July)
  - PHP 5.3.10 as FastCGI FPM
  - Apache 2.2 MPM-Worker (threaded), or nginx if you prefer that.
  - MySQL 5.5
- Use the repository software, except PHP APC (3.1.13) and PHP memcache, install from PECL

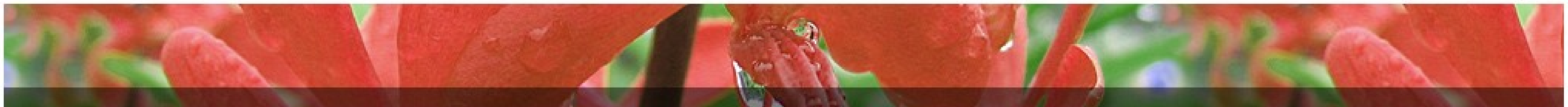
Hosting

- “Cloud” is in fashion, but has 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 alone)
  - Suitable for simple or low traffic sites, or all anonymous traffic (Varnish does the caching)



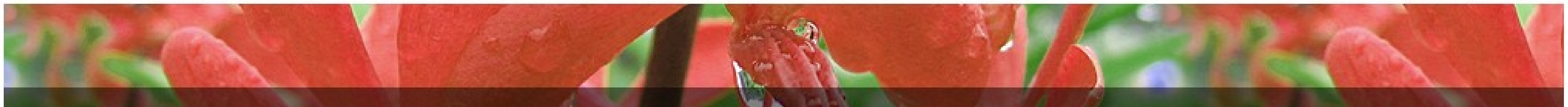
Dedicated

- Often overlooked, or underrated
- Offers the best performance for larger sites
- Does not have to be expensive
- $260 a month gets you a decent Canadian server



Multiple Servers

- Sometimes needed for complex high traffic sites, and for redundancy
- A way of having multiple spindles (db, web)
- Make sure they are over 1Gbps network
- If using multiple web heads, you need a way to propagate changes to the “files” directory.
  - NFS (slow)
  - Rsync from cron. Works but there is a lag
- Do NOT network share the entire web root

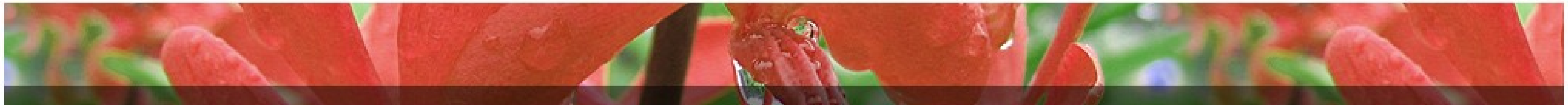


Hardware Bottlenecks

- CPUs are fast, and can fit many of them in a box (16 core, 32 threads)
- Memory is plentiful (64GB or 128GB servers available for $260 - $410 a month)
- Disks are the remaining bottleneck (mechanical)
- Can use many of them (multiple spindles instead of a single one)

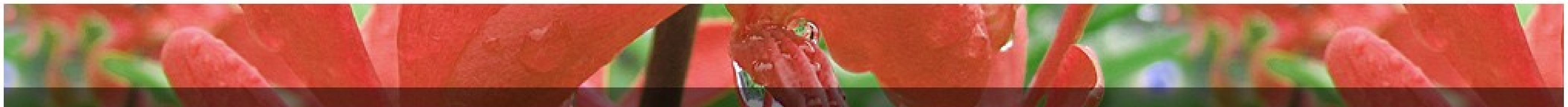
Solid State Disks

- IOPS (I/O operation per second)
  - 7200 RPM SATA disks: 100 IOPS
  - 15K RPM SAS: 210 IOPS
  - Desktop SSD: 50,000 to 80,000 IOPS
  - Enterprise SSD (PCI-e): 300,000 to 1,200,000 IOPS
- Cons
  - Marketing figures (maximums, idealized tests)
  - Watch for limitations elsewhere (SATA chipset)
  - Finite number of read/write cycles, wear levelling
  - Expensive



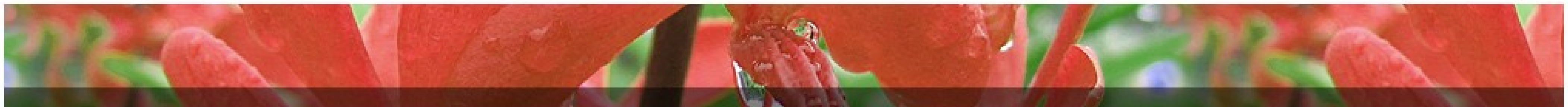
Example 1

- Outsourced service (paperless automation)
- Access via SSL only, all users logged in
- Pre-launch testing on two dedicated servers
- Testing 400 simultaneous logged in users, each sending a request every 15 seconds.
- Average response time: 385 milliseconds
- 45,674 requests in 15 minutes (= 4.38 million page view per day)



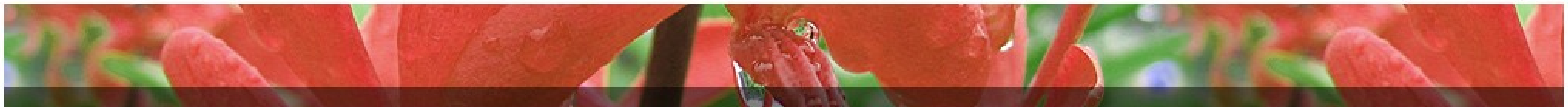
Example 2

- A live site
- Single older dedicated server
- Managed 613 simultaneous logged in users (upon sending the email newletter)
- ~ 750 to 850 milliseconds response time
- Main bottleneck was a single module that was rewritten, since it is crucial to them



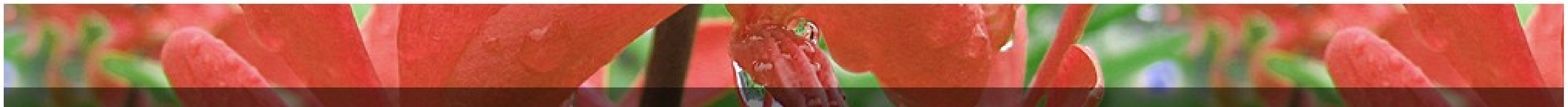
Example 3

- High traffic site with mainly non logged in visitors
- Single dedicated server
- 12,000 simultaneous non-logged in users during peak hours, 8,000 average, plus 50-90 logged in users
- No Varnish, just memcached
- 74.7 million page views per month peak, 53 million when traffic is low
- 3.4 million page views per day peak



Need Help?

- If your site has any of these symptoms
  - Site slow?
  - Suffering outages?
  - High resource usage?
- Services
  - Site Performance Assessment
  - Hosting selection, install, configure, tune



Questions?

Questions? Comments?