The Hotrodder's Guide to Maximum Performance LAMP

Photo by Brett Levin (cc-by)

Briefly About Me

- FOSS
  - Lead Masochist, Southeast Linuxfest (http://www.southeastlinuxfest.org)
  - Co-host, DistroCast Podcast (http://www.distrocast.org)
  - Gentoo Enthusiast

- Non-FOSS
  - Liberty Activist
  - Proprietor, Galt's Gulch Industries, LLC (http://www.galtsgulchindustriesllc.com)
    - College Sports … CSNbbs.com
      - Go Georgia Tech / South Carolina – Upstate!
  - Ham Radio Operator (W3GTS)

Twitter: @jeremysands

http://www.jeremysands.com

FB/G+/etc links on website

What This Talk Is NOT

- A how to guide
- Advocacy for one particular solution
  - Distro
  - Database
  - Web Server
  - Cache System

What This Talk IS

- A roadmap
  - Forged over 15 years of constantly breaking everything at all levels of the stack from a relentlessly expanding college sports website
- Provocation to plan your future
  - Fail forward, not backward
    - Always easier, cheaper, less painful to know where make a change in advance than scramble
- Interrupt me at any time with questions or clarification. Especially if I rudely blindside you with an acronym or term you don't grok.

LAMP … ish

- Linux (or FreeBSD, or NetBSD, or OpenBSD)
  - Distro (Gentoo, Debian, RHEL, SLES, CentOS, etc)
- Apache (or nginx, or lighttpd, or some scary others)
- MySQL (or Percona, or MariaDB, or Postgres, or NoSQL?)
- PHP (or Python, or JavaScript, or Java, or Perl, or Go)

- In general, the lower down that list you're doing work, the bigger the performance gains you will see.

The Hotrodder's Philosophy

- Maximum speed, minimum cost
  - Simplify
    - Query optimization, code optimization
  - Reduce weight
    - Cache, only run what you absolutely need
  - Streamline
    - Database optimization, web server optimization
  - (As a last resort) Add power
    - NEEDS MOAR XEONS!

Major Infrastructure Leaps

- 'Wal-Mart' style shared hosting
  - GoDaddy, Namecheap, super entry level VPS, etc
  - Spending < $10/mo
- Entry level VPS, dedicated
  - Linode, Server4U, 'Insert name of local datacenter here'
  - Spending < $30/mo
- Mid-Range VPS, dedicated … entry level colocation
  - Linode, Server4U, 'Insert name of local datacenter here'
  - Spending < $90/mo

Major Infrastructure Leaps 2

- High End VPS, dedicated … mid to high end colocation … entry level 'cloud'
  - Spending < $200/mo
- Scaling to infinity and beyond … high end multiple server colocation, multiple server VPS/dedicated, multiple or seriously large 'cloud' nodes
  - Spending $250 to however deep your wallet is per month

Major Infrastructure Leaps 3

- Every time you make a leap, costs jump big time
- Worth it to examine every level of the stack before you make the jump
  - Can save you hundreds of dollars up front, thousands of dollars in the long run. When you start seriously scaling, a 20% difference starts becoming entire servers worth of difference.

Infrastructure Considerations

- VPS Hosting
  - Pro: Cheaper than dedicated servers on the low end … total control
  - Con: Hypervisor/Virtualization overhead (10-15%), high end price
- Dedicated Hosting:
  - Pro: Cheaper up front than colocation, cheaper than VPS from mid range on up.
  - Con: Some of your software choices have been irrevocably made for you, in particular distro of Linux

Infrastructure Considerations 2

- Colocation
  - Pro: The most bang for your buck from mid to high end on up.
  - Con: Expensive up front
- Cloud
  - Pro: Crazy easy to scale very large very quickly
  - Con: The most expensive option by far, worry about vendor lock-in on APIs and the potentially unpretty process of intervendor cloud to cloud migration.

Operating System Considerations

- Monolithic Kernel (Linux) vs Mach Kernel (*BSD)
  - Speed (Linux) vs Security (*BSD)
    - Unless you have really over the top security needs, it is probably a better option to stick with Linux
- Distros
  - Debian: Stable and stale … or Unstable and risky
    - Ubuntu: More package support, less longevity, risky upgrades
  - Gentoo: Front end work … but back end empowerment and time savings
  - RHEL: Enterprise level quality … but decidedly not free as in beer
    - CentOS: Enterprise level quality … but no enterprise level support from the source
  - SLES: Enterprise level quality … but better than RHEL? Also not free as in beer.

Web Server Considerations

- Apache
  - Pro: Versatile, ease of use, immense Google-fu (ModRewrite)
  - Con: Slow and bloaty stock … PHP FPM security difficulties*
    - * Default config allows FPM to execute ANYTHING … even non-PHP files.   Can your users upload files?   Uhhh ohhhhh.   This is a one liner fix in nginx … it's complicated and computationally expensive to fix in apache.
- Nginx
  - Pro: Fast, lightweight
  - Con: Not as versatile … config file rage
- Lighttpd
  - Pro: Faster than stock apache, more versatile than nginx
  - Con: Smaller community, weak Google-fu

Web Server Considerations 2

- Node.js
  - Web server?   We don't need no stinkin' webserver!
  - More of a 'do it yourself' webserver … so it can be astonishingly lightweight and fast.   In use by Yahoo!, Wal-Mart, etc.
  - It just seems so very … very … very wrong to have an entire web server running on Javascript.

Database Considerations

- MySQL
  - Do not use anything other than >=5.5.   LARGE speed improvements over 5.4
  - MyISAM vs InnoDB … Use sphinx for full text search on InnoDB tables!!!
  - MySQL 5.7 looks pretty sweet …..
- Percona
  - Drop In MySQL replacement based on MySQL
  - Typically faster than stock MySQL
- MariaDB
  - Open source fork of MySQL started due to Oracle's aquisition of MySQL
  - Diverging quickly from MySQL starting with MariaDB 10.0.   MariaDB 10.x is still, in practice, binary compatible, but the current trajectory suggests this will not be the case in the future.

Database Considerations 2

- Postgres
  - A 'true' database (ORDBMS) … MySQL/Percona/MariaDB are RDBMS. RDBMS stores data based on relationships of data. ORDBMS stores data based on object oriented techniques.
- NoSQL
  - ERROR: E_NOTADATABASE
  - Pure key/value stores only
  - Aren't you brave?
  - See Richard Hipp's talk on NoSQL from SELF on YouTube
- TL;DR: http://www.howf****dismydatabase.com
- In general if you are single server or have a read heavy load on indexed tables, MySQL and Percona and MariaDB will give you the most power by far. If you are running a large DB cluster or have a write heavy load, Postgres may be a better performer for you.

Programming Language Considerations

- Spoiler alert: they're all somewhat unsatisfactory
- PHP
  - Pro: Easy to use … with a big enough hammer can be coaxed to perform decently well
  - Con: Version pain (4.x vs 5.3 vs 5.4 vs 5.5 etc) … easy to get pwned (an alarming number of PHP web apps demand chmod 777)
  - PHP 7 could be a big improvement, but adoption and maturity isn't there yet
- Python
  - Pro: Easy and very powerful to develop for … Tornado and Django are awesome
  - Con: Not a particularly elegant integration … doesn't work on all webservers … likely has a lower performance ceiling than PHP … Python JIT (Psyco) incompatible with GRSEC hardened kernel
- Java
  - I'm afraid I can't let you do that Dave...
  - That is A LOT of overhead to swallow before you even get into Tomcat … just please don't
  - Count the number of Java exploits in the last few years. Scream in horror, run away quickly.

Programming Language Considerations 2

- Javascript
  - Pro: Pretty slick to get the end user to do the work of rendering the page.
  - Con: And if that end user has a garbage machine?   And do you really trust the Ukranian website to run sanitary code on your local machine?   Many people block javascript entirely.
- Go
  - Pro: Very fast, a great compromise between development and performance
  - Con: Compiled down means strictly defined, not a wealth of choice with frameworks
- Other (Perl, etc)
  - Best to avoid (more complex, slower, poor web server support, etc)
- Computer Language Benchmarks Game (google it)

Mining For Performance Gains

- Profiler (PHP xdebug)
- PHP FPM
- Opcode and Variable caching (PHP xcache, pecl-apc … now baked in)
- Python JIT Compiler (psyco)
- Database Slow Query Logger
- Database Explain
- Database HugeTLB (RAM CACHE ALL THE THINGS)
- Memcached
- Varnish (Want to reduce your load? Stop dynamically processing entire pages)
- New Relic (Non-FOSS … but incredibly powerful)

What Works For Me

- Which as you should know by now, may not work best for you depending on the formerly mentioned considerations

- Background on use:
  - Read heavy database loading
  - Open source PHP web application
  - High concurrent load (1,000+ users all day every day)

What Works For Me 2

- Infrastructure
  - 1U Twin Servers … deploy minimal grow over time … high density, big grunt … Colocated
- OS
  - Gentoo Linux
    - Slot version installing … rolling release … glsa … hardened profile … USE flag minimalism
    - Currently deployed server in year 7+ of service (started mid kernel 2.6!)
- Database
  - MySQL (InnoDB, Sphinx, HugeTLB)
- Web Server
  - Nginx
- Language
  - PHP FPM (required by webapp … with pecl-apc, memcached, and xdebug)
- Varnish

Best Practices

- Only change one thing at a time … get a good A-B comparison
- Make the painful major changes (OS, Language) early
- Reserve capital for an upgrade warchest whenever possible. Being forced down an 'upgrade' path due to lack of frontend funds can become a vicious circle
- The biggest gains are to be made in your queries, your code, and aggressive caching.
- Always have more RAM than you think you actually need so you can aggressively cache

Best Practices 2

- MONITOR ALL THE THINGS
  - Real Time
    - Top, iotop, mytop, varnishstat, memcache-top, apachetop
  - Over Time
    - Cacti
      - Percona Cacti Templates!!!!
    - Many many others
    - Look for cyclical loading (brutal cron jobs, rarely run cron-like queries, etc)
      - Push painful stuff you have to run to around 4AM ET
- BLOCK ALL THE TROLLS … bots use resources too (and often a lot more!)
  - Ip2location's apache/iptables CIDR blocklist generator
  - Go ahead and block Russia, Ukraine, China, Myanmar, India … perhaps others.
  - Spot trolls using iftop

Painful Protips

- Mysterious high load
  - Kernel memory mayhem (HugeTLB)
  - Out of entropy
- Kernel versions MATTER if you have have HugeTLB
- Disable swap on big databases … don't get pwned by a swap happy kernel when you have enough memory
- Read the Percona MySQL Configuration Basics … my.cnf is a good starter, but not adequate. Documentation is overwhelming, particularly for MyISAM configurations.

Questions?

SouthEast | LinuxFest

LINUX IN THE GNU/SOUTH

www.southeastlinuxfest.org

@jeremysands
www.jeremysands.com
Links to G+/FB/etc on website

www.csnbbs.com