What's the Difference?

Comparing the major players in the Web Servers Space, and PHP5.6 / PHP7.0

Presented By: Joshua Knapp

Take photos, make comments on twitter with #WCRS2017

About \\Me

Joshua Knapp

Twitter:
@shadowdao
Email:
josh@cybercove.io

- Worked **with** hosting for 18+ years
  - First website was built with NotePad on Windows 98 (Hosted on AngelFire)
- Worked with Linux for 17+ years
  - Knoppix was my first Distro I installed by myself
- Worked **in** the hosting Industry for the last 10 years
  - Started as a Tier 1 Server Technician
  - Developed and implemented virtualization, spam protection, and automation solutions for Hosting Providers
- Worked with WordPress for the last 10 years

Take photos, make comments on twitter with #WCRS2017

The Colosseum

- 8 Core vCPU VM (X5670 @2.93GHz)
- 8GB of RAM
- CentOS7, Fully Updated
- Located in Irvine, California

- nghttp2/h2load (to fully take advantage of http/2)
- MariaDB (10.1) for Database, with INNODB Optimization and RAMDISK for tmp space
- PHP-FPM and lsphp are configured to use sockets

The Command

h2load -c 200 -m 200 --duration=60 --warm-up=5 https://stacker.cybercove.io

h2load - The application

-c - Concurrent Connections

-m - Max Connections

--duration - How long, in seconds, should the test run?

--warm-up - How long, in seconds, should we work up to the max number of connections?

The Fights

Web Servers

All 3 Web servers will have http/2 and SSL provided by Let's Encrypt.

All 3 have been performance tuned to give them the best chance.

We will use SuperCache to offload some of the connection limits of PHP and MySQL.

PHP Versions

Both PHP versions will be running PHP-FPM for Apache and Nginx, and lsphp for OpenLiteSpeed.

No Caching Will be Enabled to get the true performance of PHP.

Who are our contenders?

VS

VS

VS

Apache HTTPD (2.4.29)

- Most Used Web Server on the internet
- Native URL Rewrites
- Most Compatible Web Server
- .htaccess File
- Longest standing contender
- Easiest to Configure
- All Features are **Free**

- Due to compatibility, file access times are slower than the other contenders in this fight.
- Fcgi Proxy was developed late in the game, and performance can be unpredictable.

Nginx (1.13.6)

- A web server that is gaining popularity for its performance and ability to act as a cache/proxy natively
- Faster (than Apache) file access times
- Faster (than Apache) time to first byte

- Does not support .htaccess
- rewrites need to be in the config
- Flaky Connection Retry for fcgi calls
- Configurations for caching is more complicated

OpenLiteSpeed (1.4.27)
(OLS)

- Relative newcomer
- Fastest direct file access from disk
- Lowest memory footprint
- Built-in connection queuing
- Smart Keep-Alive
- Web GUI to config and manage
- Supports Apache Style Rewrite (in GUI)

- Some of the coolest features are only available in the paid for version
- OLS doesn't support .htaccess
  - Litespeed Enterprise does...
- Has a learning curve, and documentation isn't kept up with the latest version

PHP 5.6 & PHP 7.0

PHP 5.6

- Released on 28 August 2014
- Widely used for compatibility
- End of 5.x line of PHP
- End of Life 31 December 2018

PHP 7.0

- Released on 3 December 2015
- Start of the 7.x line
- Large number of performance improvements.
- Several functions have become deprecated or removed in this version.

First Battle: PHP 5.6 vs 7.0

Running h2load against all 3 web servers, 6 times, 3 with PHP 5.6 and 3 with PHP 7.0, then taking the average from the results. No Caching will be used to handle these requests.

We will look at *time to finish requests*, *number of requests total*, *requests per second*, and *request success*.

Apache 2.4
PHP 5.6 and PHP 7

PHP 5.6

Time to Finish: 65.07
Number of Total Requests: 1702
Requests per second: 28.37
Request Success Rate: 100%

PHP 7.0

Time to Finish: 65.05
Number of Total Requests: 2658
Requests per second: 44.30
Request Success Rate: 100%

Winner: PHP 7

Nginx 1.13
PHP 5.6 and PHP 7

PHP 5.6

Time to Finish: 172.29
Number of Total Requests: 82541
Requests per second: 1317.33
Request Success Rate: ~1% (791 Total)

PHP 7.0

Time to Finish: 99.70
Number of Total Requests: 146856
Requests per second: 2102.23
Request Success Rate: ~1% (1640 Total)

Winner: PHP 7

OLS 1.4.27
PHP 5.6 and PHP 7

PHP 5.6

Time to Finish: 65.06
Number of Total Requests: 1809
Requests per second: 30.15
Request Success Rate: 100%

PHP 7.0

Time to Finish: 65.05
Number of Total Requests: 2714
Requests per second: 45.23
Request Success Rate: 100%

Winner: PHP 7

Overall Winner: PHP 7

So why use PHP 5.6?

PHP 7 is great, as long as your themes and plugins support it.

Switching without testing can lead to 500 errors, infamous white screen of doom, and functions that stop working.

If you are starting a new site, start with PHP7 and build from there.

You do want to test and switch soon, because PHP 5.6 is EOL December 31, 2018.

Main Event: Apache, NGINX, OLS

Running h2load against all 3 web servers, 3 times, with PHP 7.0 (since it was the clear winner), then taking the average from the results. Caching **will** be used to handle these requests.

We will look at *time to finish requests*, *number of requests total*, *requests per second*, and *request success*.

The Results

Apache

Time to Finish: 65.07
Number of Total Requests: 2467
Requests per second: 41.12
Request Success Rate: 100%

Nginx

Time to Finish: 65.02
Number of Total Requests: 97480
Requests per second: 1308.52
Request Success Rate: ~19% (18430)

The Results (part 2)

OLS

Time to Finish: 65.06
Number of Total Requests: 36666
Requests per second: 611.10
Request Success Rate: 100%

Takeaway

Everyone is going to have things to say about the results.

PHP 7 is on average 40% faster than PHP 5.6

I am aware there are enterprise versions of Nginx and LiteSpeed. I wanted to use the free versions to stay on an even level with Apache.

Nginx is extremely fast web server/proxy, but in my tests, I could not get it to control the rate of connections very well. This lead to some very skewed numbers. Nginx Plus has some better features to control the issue we ran into, but wouldn't be far for OLS or Apache.

Who do you think won?

Feel free to contact me on twitter @shadowdao

Or Email at josh@cybercove.io