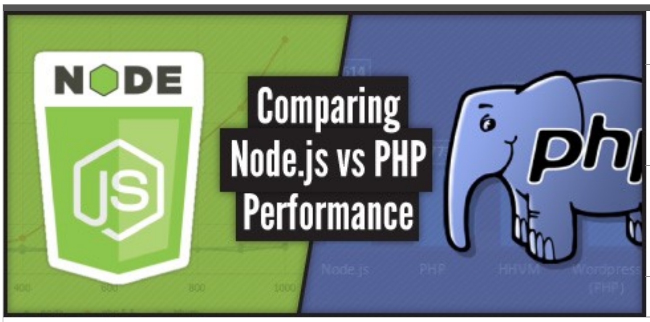
(HTTP://WWW.HOSTINGADVICE.COM) Home (Http://www.hostingadvice.com) > Blog (Http://www.hostingadvice.com/blog/)

Comparing Node.js vs PHP Performance



BLOG POST

By: Roberto Sanchez
Posted: May 5, 2016

Follow "The Hosting Blog" daily as hosting experts, programmers, and tech managers go in depth on a variety of tech-related topics.

Like 148 G+1 5

In the online development world, Node.js (https://nodejs.org/) and PHP (http://php.net/) are some of the most popular platforms used.

PHP was first released in 1995, followed later by Node.js in 2009, and they are both mainly used as a server-side scripting language for developing web-based applications. However, Node.js is also starting to be used as a desktop application development platform.

PHP began as a dynamic language to update HTML pages with sytax similar to Perl, and started off as a rather basic programming language. Node.js used the already well-established JavaScript language and created a new development platform on top of the V8 JavaScript engine. Over time, PHP got more and more advanced programming features, such as an object-oriented programming model, namespaces, package management, and exceptions.

In this article, we will take a look at the performance differences between these two platforms, some popular PHP extensions to improve performance, such as the Zend JIT runtime and HHVM PHP runtime project, and some third-party benchmarks of these extensions.

The Basics of PHP and Node.js Performance

One area where there has been a healthy competition of sorts between PHP and Node.js is in the area of performance.

PHP - Interpreted Compilation

The standard PHP Zend runtime uses an interpreted compilation (http://en.wikipedia.org/wiki/Interpreter_(computing)) strategy, leading to less optimized code execution than a good JIT-(Just-in-Time)-based runtime.

Node.JS - Just-In-Time Compilation Using V8

Node.js uses Just-in-Time compilation (http://en.wikipedia.org/wiki/Just-in-time_compilation) with the V8 JavaScript Engine. This strategy for dynamically-typed languages has proven itself, but does often increase startup time a little.

Increase PHP Performance: OPcaching, JIT Compilation, and HHVM

Several popular extensions have been developed — designed to improve PHP performance.

OPcaching in PHP

Zend PHP 5.5+ also has a caching engine called OPcache (http://php.net/manual/en/book.opcache.php), which stores precompiled script bytecode in shared memory, thereby removing the need for PHP to load and parse scripts on each request. This is not as performant as JIT compilation but is a stable feature that is working now. You can enable PHP 5.5 OPcache (http://www.hostingadvice.com/how-to/enable-php-5-5-opcache-ubuntu-14-04/) if for some reason it's not enabled by default (as it should be with PHP 5.5 or higher).

PHP Runtime Alternatives: HHVM & Hack

HHVM (http://hhvm.com/) is an open-source virtual machine designed for running applications

Follow the Experts

1,566 (https://www.facebook.com/HostingAdviceCom) 463 (https://twitter.com/Hosting_Advice)

We Know Hosting

240 (https://plus.google.com

HostingAdvice.c...

$ 8 1,9 7 4

spent on testing web hosting services!

From the Blog

(http://www.hostingadvice.com/blog/how-jimdo-founders-lead-by-example/)

Jimdo Founders Focus Building Sites Sans Co (http://www.hostingadv/blog/how-jimdo-found lead-by-example/)
Laura Bernheim, 5/19/16

(http://www.hostingadvice.com/blog/load-impact-cto-talks-load-testing/)

Load Impact CTO: Why Testing Matters (http://www.hostingadv/blog/load-impact-cto-t load-testing/)
Alexandra Leslie, 5/18/1

(http://www.hostingadvice.com/blog/how-hosting-com-assures-hipaa-compliance/)

HOSTING Assures HIP Compliance with Secu Managed Cloud Hostin (http://www.hostingadv/blog/how-hosting-com-assures-hipaa-compliance/)
Toby Sembower, 5/17/16

more blog posts (http://www.hostingadvice.com/blog/) »

Hosting Reviews

Best Web Hosting 2016 (http://www.hostingadvice.com/reviews/)

Cheap Web Hosting (http://www.hostingadvice.com/reviews/cheap/)

Dedicated Server Hosting (http://www.hostingadvice.com/reviews/dedicated/)

VPS Hosting (http://www.hostingadvice.com/reviews/vps/)

WordPress Hosting (http://www.hostingadvice.com/reviews/wordpress/)

Website Builder (http://www.hostingadvice.com/reviews/website-builder/)

Hosting How To's

Turn a PHP Array Into a String (Using *implode* 101) (http://www.hostingadvice.com/how-to/php-array-to-string/)
PJ Fancher, 5/16/16

written in PHP and/or the Hack language (http://hacklang.org/).

HHVM uses a Just-in-Time (JIT) compilation approach to achieve better performance. Node.js also does JIT compilation using the V8 JavaScript runtime.

To make the most of the HHVM JIT compilation, it is important to put your code into functions, especially for the more time-consuming code parts.

The Hack language allows for performance improvements mainly due to these language features:

- Async programming (http://docs.hhvm.com/manual/en/hack.async.php): Run separate blocks of code in parallel (using different threads)
- Type annotations (http://docs.hhvm.com/manual/en/hack.annotations.php): Improve performance with a more predictable type environment

JIT Compilation in Future PHP Releases

There has been discussion (https://www.mail-archive.com/internals@lists.php.net/msg76332.html) and development to use JIT compliation (https://github.com/zendtech/php-src/tree/zend-jit/ext/opcache/jit) in a future release of the PHP Zend runtime. JIT for PHP provides more than five times speedup on *bench.php* and as high as a 30 times improvement on some other specific benchmarks like Mandelbrot.

Multi-Tasking in Node.js and PHP

Node.js and PHP differ in their approach to concurrency, with Node.js using a non-blocking event loop (running in a single process) and the standard PHP Zend (https://en.wikipedia.org/wiki/Zend_Framework) runtime using a blocking process.

For this reason, multiple PHP processess are often launched from the web server to be able to keep serving web requests. In Apache, you can set the maximum number of PHP processess allowed to run, and this works fine for many types of web applications.

Now, when the work being done in these processess tends to be long(er)-running, the server can quickly run out of available PHP processess. This will cause the HTTP request response times to increase, perhaps even to the point of requests timing out.

Web servers like Apache and NGINX have ways to improve the PHP multi-tasking performance to some extent, but these are more limited than true language support for multi-tasking.

There are several projects being worked on to make PHP asynchronous processing (multi-tasking) work better. The most popular is the HHVM project (http://hhvm.com/) released in 2014 by Facebook; another project tackling PHP concurrency is pthreads (http://pthreads.org/).

Node.js vs PHP Performance Benchmarks

So what might these PHP extensions bring us in terms of performance compared to standard PHP or Node.js?

Let’s have a look at a few different benchmarks people have done.

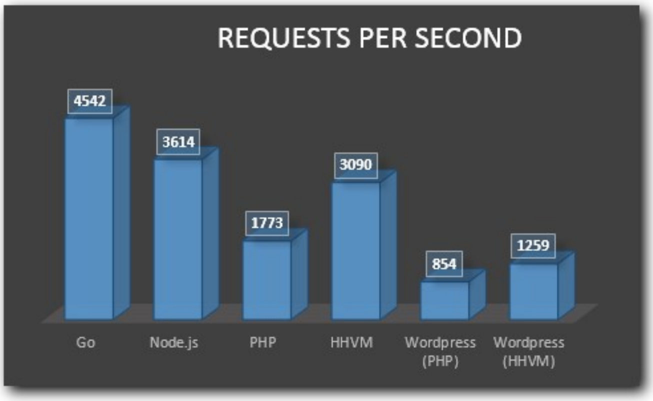
Benchmark 1: Simple HTTP Requests

Click on the image to go to the benchmark website and see the full setup details.

Linux: Add Users to Group (http://www.hostingadvice.com/how-to/linux-add-user-to-group/)
Roberto Sanchez, 4/27/16

Squarespace vs. WordPress (http://www.hostingadvice.com/how-to/squarespace-vs-wordpress/)
Toby Sembower, 4/14/16

more how to's (http://www.hostingadvice.com/how-to/) »



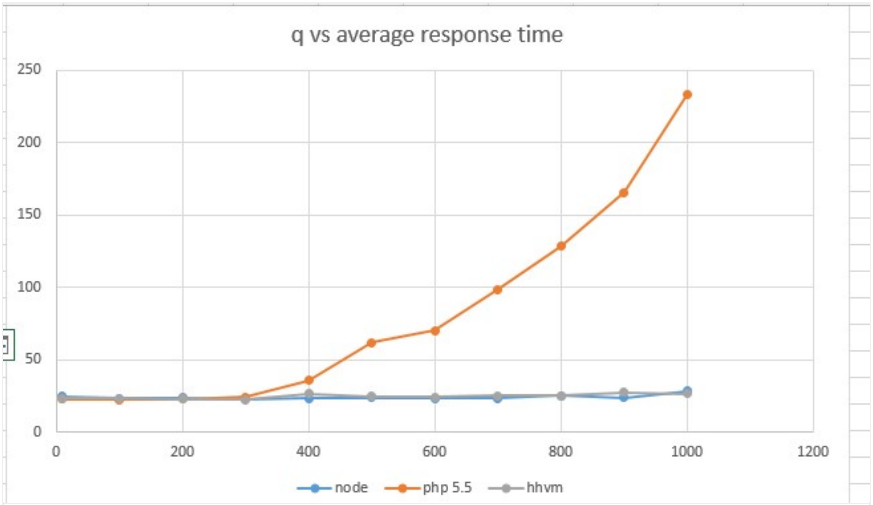
Node.js vs. PHP performance and requests handled per second

Notes:

- This HTTP-controlled “hello world” benchmark used Zend PHP 5.6.6 with OPcache enabled.
- The test was done using the Apache ab benchmarking tool (http://www.hostingadvice.com/external/?site=762f3dd1b6d29f265e08d3a14c0abacae2ded72d9bf10c577ad5246c9ccecb090489b56582bcc07b2658d7d3ea19e6f725924dac3ffbd806).
- The performance of Zend PHP is less than half of Node.js.
- The performance of HHVM is about 74% better than Zend PHP.
- The performance of HHVM is very close to Node.js, but Node.js is still about 17% faster.
- The performance increase when using WordPress with HHVM is about 47%.

Benchmark 2: HTTP + CPU tasks

Click on the image to go to the benchmark website and see the full setup details.



Node.js vs. PHP Performance with CPU heavy task of running a bubble sort

Notes:

- This HTTP bubble-sort (https://en.wikipedia.org/wiki/Bubble_sort)-job benchmark used Zend PHP 5.5.7 with OPcache enabled.
- The PHP response times degrade rapidly after depleting the available PHP process pool (using a standard maximum of 250 processes on the web server).
- The HHVM performance is similar to Node.js up to 1,000 sort elements, but with 10,000 sort elements, Node.js was about twice as fast HHVM (not visible on this chart).

Benchmark 3: CombSort Strict CPU Test

The following CombSort sorting benchmark (http://kokizzu.blogspot.be/2015/02/numeric-combsort-benchmark-updated.html) is a strict CPU test.

The benchmark results:

	CPU time	System time	RAM
PHP 5.6.4	102.69s	104.20s	2497508 KB
HHVM 3.5.0	12.56s	14.83s	362488 KB
Node.js v0.10.35	2.64s	2.64s	92240 KB

HHVM is seven times faster than plain PHP (by system time), but Node.js is more than five times faster than HHVM in this number-crunching test.

In terms of RAM usage, HHVM is much more efficient than PHP, but Node.js is even better.

Final Thoughts on Node.js and PHP Performance

For most PHP applications, HHVM clearly provides a significant performance boost thanks to the JIT compilation. The same will likely be true when the Zend runtime JIT feature is stable and released. How much the PHP JIT compilation improves upon the non-JIT PHP is also a matter of code organzation (especially the time-consuming code parts).

At best, HHVM is on par with Node.js, but in some cases, Node.js will clearly perform much better (e.g., for CPU-bound tasks). There are also more async code modules available for Node.js.

There is ongoing discussion about the future of the Zend and HHVM PHP runtimes (http://www.sitepoint.com/hhvm-vs-zend-engine-php-6/). HHVM also looks to be getting even faster by using LLVM (http://phoronix.com/scan.php?page=news_item&px=HHVM-2015-Plans-3.6) later this year.

If you are interested in monitoring your PHP application performance better, perhaps have a look at these performance dashboard solution providers:

- New Relic (http://newrelic.com/php): Allows you to track application issues like Apdex score (https://en.wikipedia.org/wiki/Apdex), response times, requests per minute, and errors. After which, you can drill down to the code parts that are causing certain performance issues
- DataDog (https://www.datadoghq.com/): Offers real-time application metrics with an interactive dashboard

Questions or Comments? Ask Roberto!

Ask a question and Roberto **will respond** to you. We strive to provide the best advice on the net and we are here to help you in any way we can.

30 Comments HostingAdvice.com 1 Login

Recommend Share Sort by Best

Join the discussion…

Alexandre Morgaut A$AP #2fleek • a year ago

I'd be happy to see benchmarks to agree with you
I agree any comparison can be biased depending on how benchmarks are done, so if you have seen other results, please share them with us.

Note that results can occasionally depend on the OS, the server architecture, the storage type (HDD/SSD), the web server configuration, ...

It could also be interesting to see updated comparisons with the Zend PHP 7 engine & the last io.js fork which use a more recent V8 engine (now nodejs/node before merge back with joyent/node).

Note also that scripts performances in both languages can depend on algorithm optimisation, like usage of asm.js patterns and/or Typed Arrays

• Reply • Share ›

frankel0 HA Expert Alexandre Morgaut • a year ago

We would also love to see more benchmarks. In house, we use both HHVM and NodeJS and they serve different purposes for us. We tend to go with NodeJS scripting when parallelization is of importance and go with PHP/HHVM for ease of use and we actually like PHP.

4 • Reply • Share ›

Jefferson González Alexandre Morgaut • a year ago

Is always fun to read articles like this but I did my own testing and here are the results. Spoiler: PHP was faster. (CPU: AMD FX(tm)-8320 Eight-Core Processor, RAM: 12G)

Node v0.12.4, code:

```
-----------------------------------------------------------------------
var http = require('http');
var server = http.createServer(function (request, response) {
response.writeHead(200, {"Content-Type": "text/plain"});
response.end("Hello World\n");
});
server.listen(8000);
```

PHP v5.6.9 with opcache, fpm, hiawatha webserver, code:

```
-----------------------------------------------------------------------
print "Hello World";
```

I did benchmarks with siege, and ran them on the same machine first.

see more

2 • Reply • Share ›

bmarkovic Jefferson González • 8 months ago

You do realize that opcache made sure nothing but the very string "Hello World" gets serverd by your PHP setup (print pretty much defaults to a few lines of machine code returning the string through, in your case, fastCGI pipe) whereas node actually had to do webserving itself (things like session handshake dealing with content type negotiations etc.). Much as no production PHP setup would nowadays be running (hopefully) without opcache, no node setup would serve simple static stuff (which a string is) through node itself. What you've actually shown here is that hiawatha was a better web server than node. Any web server (even Apache) is better than node, static content that needs no processing has always been served traditionally by nginx in the node world.

1 • Reply • Share ›

ralstonrats bmarkovic • 2 months ago

No, the opcache caches the compiled code, not its output.

• Reply • Share ›

bmarkovic ralstonrats • a month ago

Which for printing a literal string is nothing but that very string and very few lines of code sending characters down a pipe.

• Reply • Share ›

Was this helpful? Tell Us Thanks.

148 Like **Like this article on Facebook**

Tweet **Tweet this article on Twitter**

5 G+1 **Share this article on Google+**

About the Author

Roberto Sanchez

G+ Folgen

Roberto Sanchez got his start with web hosting when he was still in high school, with a Perl script to conjugate verbs on a shared host. Roberto has been involved with web development ever since, experimenting with many web languages and hosting types. Roberto is on the web development team at Digital Brands Inc., where he applies his knowledge of PHP and WordPress to improve performance for the firm's various web properties. Roberto also leverages NodeJS to develop exciting products at Digital Brands Inc. and is eager to share his experience with HostingAdvice.com users.

(http://www.hostingadvice.com)

Our mission: to inform and educate our visitors about the world of Web Hosting.

Home (http://www.hostingadvice.com/)
| **About (http://www.hostingadvice.com/about/)**
| **Contact (http://www.hostingadvice.com/contact/)**
| **Privacy (http://www.hostingadvice.com/privacy/)**
| **Terms (http://www.hostingadvice.com/terms/)**
| **Sitemap (http://www.hostingadvice.com/sitemap/)**

Contributing Experts

Latest Advice

— Sections — ▾

— Best Hosting — ▾

— Host Reviews — ▾

Follow Us

(https://www.facebook.com/HostingAdviceCom)

(https://twitter.com/Hosting_Advice)

(https://plus.google.com/+Hostingadvice/)