AHOSTING

Apache vs LiteSpeed for WordPress

A transparent performance study of dynamic WordPress execution and warm page-cache delivery in representative AHosting environments.

51-54%	~11 ms	0
lower dynamic median TTFB on LiteSpeed	warm cached TTFB on both platforms	failed load-test requests

Prepared by AHosting Technical Operations
Test date: 23 August 2026
Document version: 1.0

Executive Summary

We tested two AHosting-controlled WordPress lab sites representing our Apache and LiteSpeed shared-hosting environments. The most important finding was not that one web server wins every test. It was that page caching changes the result dramatically.

- On dynamically generated WordPress requests, the LiteSpeed/LSAPI environment produced approximately 51 to 54 percent lower median TTFB across the home page, a post and search.
- At 20-way dynamic concurrency, LiteSpeed completed a median 6.04 requests per second versus 3.08 on Apache, with p95 latency of 3.60 seconds versus 6.50 seconds.
- With both fast page-cache paths enabled, cached home and post TTFB was effectively tied at approximately 11 milliseconds.
- No failed requests were recorded across 1,134 dynamic and 12,000 cached load-test requests.

Practical conclusion: LiteSpeed provided the stronger and more consistent dynamic WordPress path in these AHosting environments. Correct full-page caching made both platforms extremely fast for cacheable pages.

What was identical

Both sites used WordPress 7.1, PHP 8.2, Twenty Twenty-Five 1.5, the same content fixture and equal account-level CPU, memory, process and I/O limits. Cloudflare was DNS-only, with no CDN page cache in front of either origin.

What was not identical

The host hardware, PHP execution path and database placement differed. These are representative product environments, not a laboratory isolation of the web-server binary. Both hosts had substantial unused memory and showed no resource pressure during testing.

Test Environment

Condition	Apache lab	LiteSpeed lab
Representative role	Apache development/shared stack	LiteSpeed production/shared stack
Web server	Apache 2.4.67	LiteSpeed Enterprise 6.3.6
CPU threads	48	40
Host memory	31.5 GB	64.0 GB
Available memory near test	About 28 GB	About 61.5 GB
PHP execution	PHP 8.2 PHP-FPM	PHP 8.2 LSAPI with OPcache
Page cache	WP Super Cache 3.1.1	LiteSpeed Cache 7.9
Fast cached path	Expert / mod_rewrite	Server-integrated LSCache
Database	Local MariaDB 10.11	Remote MySQL 8.0.41
Sequential protocol	HTTP/1.1	HTTP/2

Identical WordPress fixture

- 50 posts, 8 pages and 60 comments
- 10 locally hosted 1600 x 900 JPEG images
- Identical permalink structure and application content
- Search indexing disabled on temporary lab domains
- Equal limits: 100% CPU, 20 entry processes, 1 GB RAM, 100 processes, 1 MB/s I/O and 1,024 IOPS

Methodology

Testing ran on 23 August 2026 from AHosting's Michigan jump host, close to both origins. Sequential tests used three independent rounds, alternating platform order. Every route had 30 observations per platform and cache mode. Dynamic requests used unique URLs. Warm-cache requests were primed and cache hits were verified before measurement.

Load tests used HTTPS and HTTP/1.1 keep-alive. The cached test used three rounds of 500 requests at concurrency 1, 5, 10 and 20. For the dynamic test, both page-cache plugins were temporarily disabled and each concurrency stage ran for 10 seconds in three rounds. Cache configurations were restored and revalidated immediately afterward.

Dynamic WordPress Results

Lower TTFB is better. Values below are based on 30 observations per route and platform.

Route	Apache median	LiteSpeed median	Reduction	Apache p95	LiteSpeed p95
Home	361.5 ms	165.2 ms	54.3%	2,972.9 ms	207.2 ms
Post	376.5 ms	175.7 ms	53.3%	2,583.4 ms	217.0 ms
Search	385.2 ms	189.2 ms	50.9%	2,996.7 ms	197.8 ms

Interpretation

The LiteSpeed environment was approximately twice as fast at the median and substantially more consistent at p95. This result belongs to the complete stack, including LSAPI, OPcache, PHP process behaviour, database placement and host conditions. It should not be attributed to the web-server binary alone.

Dynamic concurrency

Concurrency	Apache req/s	LiteSpeed req/s	Apache p95	LiteSpeed p95
1	2.97	6.62	361 ms	184 ms
5	2.84	6.83	1,794 ms	782 ms
10	3.08	6.59	3,402 ms	1,598 ms
20	3.08	6.04	6,501 ms	3,598 ms

At concurrency 20, LiteSpeed completed about 96 percent more dynamic requests per second and reduced p95 latency by about 45 percent. No failed requests were recorded.

Warm Page-Cache Results

Apache used WP Super Cache Expert/mod_rewrite delivery. LiteSpeed used server-integrated LSCache. Verified cache hits were primed before measurement.

Cacheable route	Apache median	LiteSpeed median	Apache p95	LiteSpeed p95
Home	11.1 ms	10.9 ms	14.4 ms	11.7 ms
Post	10.9 ms	10.8 ms	12.0 ms	11.8 ms

Once both platforms served static page-cache files through their fastest paths, the practical latency difference was negligible. Search is excluded because the plugins used different default search-cache policies.

Cached concurrency

Concurrency	Apache req/s	LiteSpeed req/s	Apache p95	LiteSpeed p95
1	655.23	260.60	2 ms	4 ms
5	1,263.26	911.47	5 ms	7 ms
10	1,266.65	1,149.32	10 ms	10 ms
20	1,262.94	1,190.48	18 ms	19 ms

Apache's Expert cache path was slightly faster in this specific HTTP/1.1 transfer test. Both systems nevertheless delivered roughly 1,200 or more cached requests per second at higher concurrency with p95 below 20 ms. The test was increasingly influenced by connection and transfer behaviour rather than WordPress execution.

What the Results Mean

1. LiteSpeed's strongest advantage appears when WordPress must execute

Logged-in sessions, carts, personalised pages, cache misses and selected application endpoints still reach PHP and the database. LiteSpeed provided materially better dynamic headroom and tail latency in these tests.

2. Correct caching matters more than the server logo

An uncached WordPress site can become PHP-bound quickly. A correctly configured Apache cache reduced cacheable-page TTFB from hundreds of milliseconds to approximately 11 ms. LSCache produced similarly fast cached responses.

3. Integration reduces operational friction

WP Super Cache required Expert mode and Apache rewrite rules to reach its fastest path. LiteSpeed Cache uses a server-integrated path with WordPress-aware purge controls. The plugin and policy must still be enabled and verified.

Recommended product interpretation

For highly dynamic WordPress and WooCommerce workloads, the tested LiteSpeed/LSAPI stack provided meaningful headroom. For cacheable publishing sites, both platforms performed very well when a fast full-page cache path was correctly configured.

Approved public claim

In representative AHosting lab environments, LiteSpeed reduced median dynamic WordPress TTFB by 51 to 54 percent. When both platforms used a correctly configured full-page cache, cacheable pages were effectively tied at approximately 11 ms from the origin-proximate test location.

Limitations and Reproducibility

- The hosts did not use identical CPU counts, memory capacity or database placement.
- The study compared representative AHosting product environments, not isolated web-server binaries on identical hardware.
- The load generator was in Michigan and results should not be treated as global end-user latency.
- ApacheBench used HTTP/1.1 keep-alive and did not measure browser rendering, HTTP/2 multiplexing or Core Web Vitals.
- The fixture does not represent every theme, plugin stack, WooCommerce store or traffic mix.
- The tests were controlled capacity probes, not endurance tests or service-level guarantees.
- Requests per second are HTTP throughput, not simultaneous human visitors.

Raw evidence retained internally

final2-michigan-bypass-combined-20260823.csv

final-expert-michigan-warm-combined-20260823.csv

final-michigan-dynamic-load-20260823.csv

final-michigan-cached-load-20260823.csv

Test owner

AHosting Technical Operations
Measured: 23 August 2026
Recommended review date: 23 February 2027

Claim boundary

This report does not claim that LiteSpeed is always faster than Apache, that 1,200 requests per second equals 1,200 simultaneous visitors, or that benchmark results guarantee Core Web Vitals, SEO rankings, traffic capacity or uptime.