AHOSTING

Apache vs LiteSpeed for WooCommerce

A controlled comparison of cached catalog delivery and dynamic cart, checkout and add-to-cart performance under equal WooStart account limits.

55-64%	2.2-2.7x	0
lower median latency on transactional paths	dynamic request throughput	failed requests and LVE faults

Prepared by AHosting Technical Operations
Test date: 24 August 2026
Document version: 1.0

Executive Summary

AHosting tested the same WordPress and WooCommerce fixture on controlled Apache and LiteSpeed lab sites. Both accounts used the published WooStart CloudLinux limits. The largest practical difference appeared on dynamic commerce paths that cannot be handled as ordinary public page-cache hits.

- LiteSpeed reduced median empty-cart latency by 63.8 percent, from 1,183 ms to 428 ms.
- LiteSpeed reduced median filled-cart latency by 60.8 percent, from 722 ms to 283 ms.
- LiteSpeed reduced median filled-checkout latency by 59.1 percent, from 696 ms to 285 ms.
- LiteSpeed reduced median add-to-cart latency by 55.1 percent, from 683 ms to 307 ms.
- The cached home page was effectively tied; LiteSpeed led the cached product page by 33.3 percent at the median.
- Every published scenario completed with zero failed requests, and neither account recorded an LVE resource fault during the final measurement window.

Practical conclusion: the tested LiteSpeed Enterprise, LSAPI and LiteSpeed Cache stack provided materially lower latency and more transactional throughput for WooCommerce cart, checkout and add-to-cart activity under equal WooStart account limits.

Approved Public Claim

Under equal WooStart account limits, LiteSpeed reduced median WooCommerce cart, checkout and add-to-cart response times by approximately 55 to 64 percent compared with the tested Apache stack.

Test Environment

Condition	Apache lab	LiteSpeed lab
Application	WordPress 7.1 / WooCommerce 11.0.1	WordPress 7.1 / WooCommerce 11.0.1
PHP	PHP 8.2 with PHP-FPM	PHP 8.2 with LSAPI/LSPHP
Page cache	WP Super Cache 3.1.1	LiteSpeed Cache 7.9
Catalog	50 identical synthetic products	50 identical synthetic products
Host CPU	Xeon Silver 4214R / 48 threads	Xeon Silver 4114 / 40 threads
Host memory	30 GiB; 27 GiB available	62 GiB; 60 GiB available
Test source	Michigan network	Michigan network

Equal WooStart Account Limits

Resource	Both accounts
CPU	300% - up to 3 cores
Physical memory	3 GiB
Virtual memory	Unlimited
Entry processes	40
Processes	200
Disk I/O	20 MiB/s
IOPS	2,048

The nodes were not hardware-identical. The Apache node had more and newer CPU threads; the LiteSpeed node had more total memory. Both nodes had ample available memory, and CloudLinux constrained each benchmark account to the same product-level ceilings. This is a comparison of two production-style hosting stacks under equal account limits, not an isolation of the web-server binary.

Measured Results

Median response latency is the user-facing comparison below. Lower latency and higher requests per second are better.

Scenario	Load	Apache	LiteSpeed	Latency change	Req/s A / LS
Cached home	200 / c10	9 ms	10 ms	Effectively tied	1,037.24 / 1,012.68
Cached product	200 / c10	12 ms	8 ms	33.3% lower	797.19 / 1,196.82
Product with session	60 / c3	5 ms	5 ms	Tied	538.45 / 622.38
Empty cart	100 / c5	1,183 ms	428 ms	63.8% lower	4.11 / 10.93
Filled cart	60 / c3	722 ms	283 ms	60.8% lower	3.92 / 9.85
Filled checkout	60 / c3	696 ms	285 ms	59.1% lower	4.10 / 10.00
Add to cart	30 / c2	683 ms	307 ms	55.1% lower	2.73 / 6.09

Transactional Interpretation

- Empty cart throughput was 2.66 times higher on LiteSpeed.
- Filled cart throughput was 2.51 times higher on LiteSpeed.
- Filled checkout throughput was 2.44 times higher on LiteSpeed.
- Add-to-cart throughput was 2.23 times higher on LiteSpeed.

Cached delivery made both stacks very fast. The stronger commercial result is the dynamic WooCommerce path: cart, checkout and add-to-cart still require application and database work, where LiteSpeed led consistently in this test.

Methodology and Claim Boundaries

The final tests ran on 24 August 2026 from AHosting's Michigan jump host. Both installations used the same WooCommerce version, product count, product data and product identifier. Public pages were warmed before cache tests. A real anonymous WooCommerce session containing a product was used for filled cart and checkout tests. No payment, email or order submission occurred.

Validation

- HTTP 200 was confirmed for the filled cart and checkout session before load measurement.
- All published scenarios reported zero failed requests.
- CloudLinux reported zero CPU, memory, entry-process, process, I/O and IOPS faults during the final measurement window.
- A preliminary run using unequal default LVE limits was rejected and is not included in the published results.

Limitations

- The host hardware, PHP execution path and cache implementations were not identical.
- The synthetic 50-product catalog does not reproduce every production store, theme, plugin set, database size or traffic pattern.
- ApacheBench measures HTTP response behaviour, not browser rendering or Core Web Vitals.
- The Michigan test source is origin-proximate and does not represent worldwide visitor latency.
- Results describe the tested configurations and are not a universal guarantee for every Apache or LiteSpeed deployment.
- Requests per second are protocol throughput, not a direct count of simultaneous human shoppers.

Recommended Product Use

Use WooStart for smaller and growing stores whose dynamic workload remains within the published shared-account limits. Monitor sustained CPU, memory, entry-process, I/O, database and background-job demand. Move stores with continuously high checkout concurrency, large imports or custom workers to WooPower or another environment with reserved resources.