**ЛЕКЦІЯ 4. Серверні технології. Веб-сервери: Apache, Nginx. Сервери додатків. Конфігурація та налаштування серверів**

Веб-технології активно розвиваються, і роль серверних технологій у клієнт-серверній архітектурі є ключовою. Сервери забезпечують обробку запитів клієнтів, управління даними та передачу відповідей. У даній лекції ми детально розглянемо веб-сервери, зокрема Apache та Nginx, а також сервери додатків, які відіграють важливу роль у сучасній інфраструктурі веб-додатків. Ми обговоримо, як їх правильно налаштовувати та конфігурувати для забезпечення ефективної роботи.

**1. Серверні технології**

Серверні технології — це інструменти, що дозволяють надавати користувачам послуги через мережу, зазвичай через Інтернет. Основними компонентами цих технологій є веб-сервери та сервери додатків.

**Що таке вебсервер**

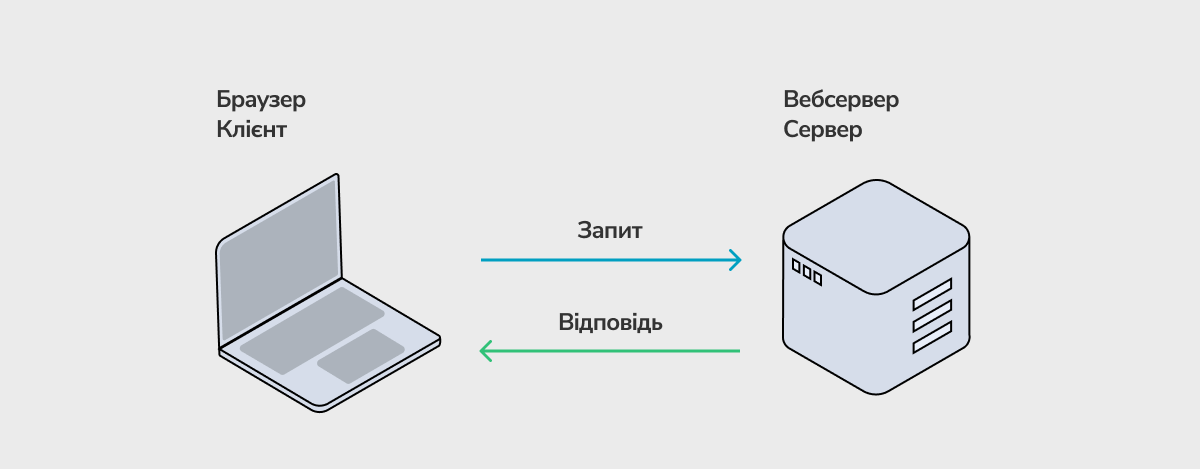
Якщо на сторінці будь-якого сайту ми зайдемо в «Інструменти розробника» (F12 в Google Chrome), то побачимо, що сторінка складається з певної кількості запитів. Щоб опрацювати ці запити, з нашого, клієнтського, боку працює браузер — він запити подає, а з боку сервера працює вебсервер, який, власне, на ці запити відповідає.

Вебсервер — це програмне та апаратне забезпечення, яке використовує протокол HTTP (Hypertext Transfer Protocol) та інші протоколи для відповіді на запити клієнтів, зроблені через браузер.

Основне його завдання — показувати вміст сайту, водночас зберігаючи, обробляючи та доставляючи його сторінки користувачам.

**Як працює вебсервер**

Схематично базову роботу вебсервера можна зобразити так:



Ліворуч зображено клієнтську сторону, або сторону браузера. Праворуч — сторона сервера, де працює вебсервер. Наприклад, ми запитуємо головну сторінку, а вебсервер має відпрацювати й повернути нам те, що очікується.

Всередині вебсерверу в цей момент може відбуватися багато чого. І це «багато чого» можна поділити на легкі (статичні) та важкі (динамічні) запити.

Якщо запит може вирішуватися віддачею якогось простого файлу, наприклад, картинки, його можна назвати статичним. Зі списку файлів, які лежать у директорії ми беремо один конкретний файл і нічого більше не змінюємо. Коли ми отримуємо сторінку, таких статичних запитів може бути дуже багато, включно з картинками, шрифтами, сss-ми, js-ми, які лежать на диску.

Але може статися і так, що запитувана сторінка тягне за собою купу програмного коду. Тут маємо справу вже з динамічним, важким запитом. Якраз у цьому випадку у вебсервері виконується чимало складних процесів — йдуть запити в базу даних, вираховуються шаблони та інші джерела даних. Коротко кажучи, відбувається звичайна робота динамічного сайту.

За вебсервером у нас може бути цілий блок вебдодатків, або серверів додатків. Без них відповісти на динамічний запит практично неможливо.

**Чим вебсервер відрізняється від сервера**

**Сервер** — це фізичне центральне сховище, де зберігаються інформація та комп’ютерні програми.

**Вебсервер** — це програмне забезпечення, призначене для запуску необхідного обладнання, яке може задовольняти запити клієнтів і доставляти їм вебсторінки.

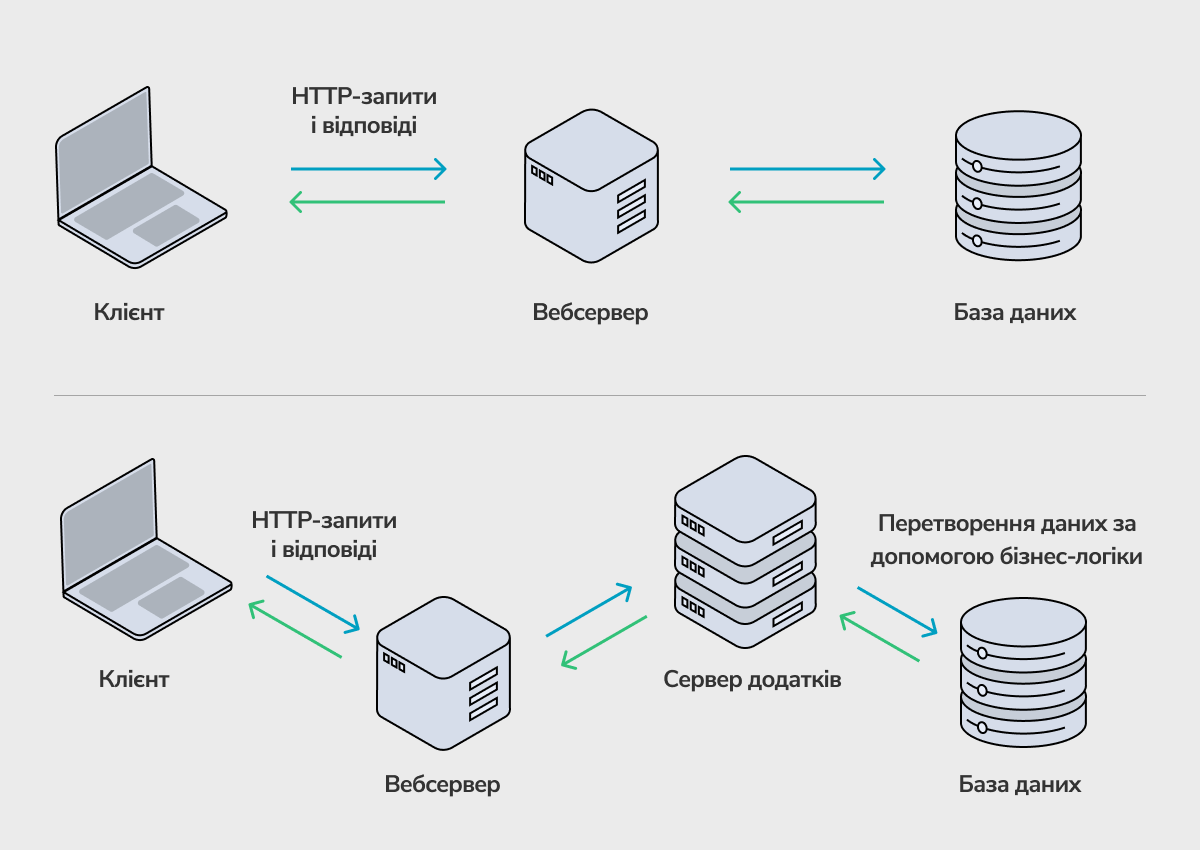
Клієнт — будь-який пристрій, який може отримати доступ до інтернету, наприклад, смартфон, розумний годинник, ноутбук або ПК, і запитувати дані з вебсервера, як правило, за допомогою браузера, наприклад, Chrome або Firefox.

Основні різновиди серверів, які використовуються для доставлення сайтів кінцевому користувачу – **вебсервер**і **сервер додатків.**

**Вебсервер** приймає і виконує запити від клієнтів на отримання статичного вмісту, тобто HTML-сторінок, файлів, зображень і відео, з сайту. Він обробляє лише HTTP-запити та відповіді.

**Сервер додатків** надає клієнтам бізнес-логіку, яка генерує динамічний контент. Це програмний фреймворк, який перетворює дані для забезпечення спеціалізованої функціональності. Сервери додатків покращують інтерактивні частини сайту, які можуть виглядати по-різному залежно від контексту запиту.

Ілюстрація нижче підкреслює різницю в їхній архітектурі.



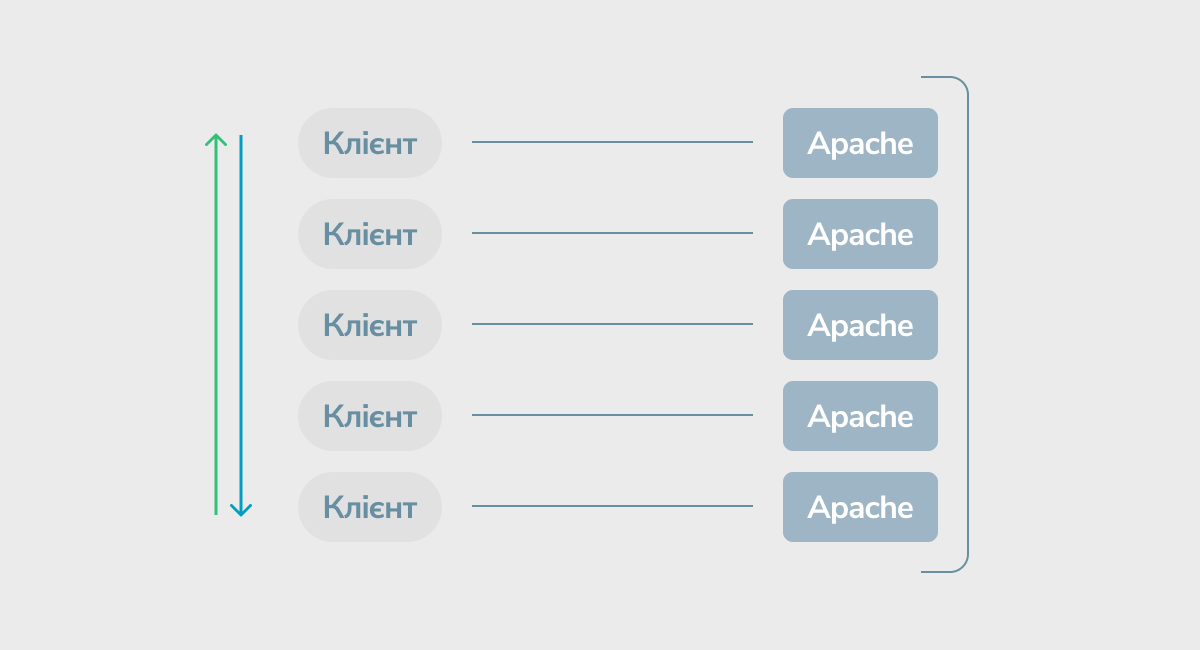
**Які різновиди вебсерверів бувають**

Сьогодні є три найпопулярніші вебсервери: Apache, NginX та LiteSpeed.

**Apache.**Це досить відомий набір ПЗ із відкритим вихідним кодом для створення вебсайтів та вебдодатків.

Створений ще в 1995 році Apache досить довгий час домінував на ринку. Він попередньо встановлений у всіх основних дистрибутивах Linux, що завжди полегшувало його запуск і роботу.

Apache досить складний продукт. Має багато модулів і споживає значну кількість ресурсів. Щоб обслуговувати кілька клієнтів, на кожне підключення створюється окремий робочий процес. А через те, що кожний із них вимагає певної кількості ресурсів, встановлюються обмеження на кількість зʼєднань.



Як тільки кількість зʼєднань масштабується, ми впираємося в розмір оперативної памʼяті й витрачаємо купу ресурсів просто на керування великою кількістю робочих процесів.

Хоча є в Apache й очевидно потрібні штуки, як-от кешування зворотного проксі-сервера та балансування навантаження.

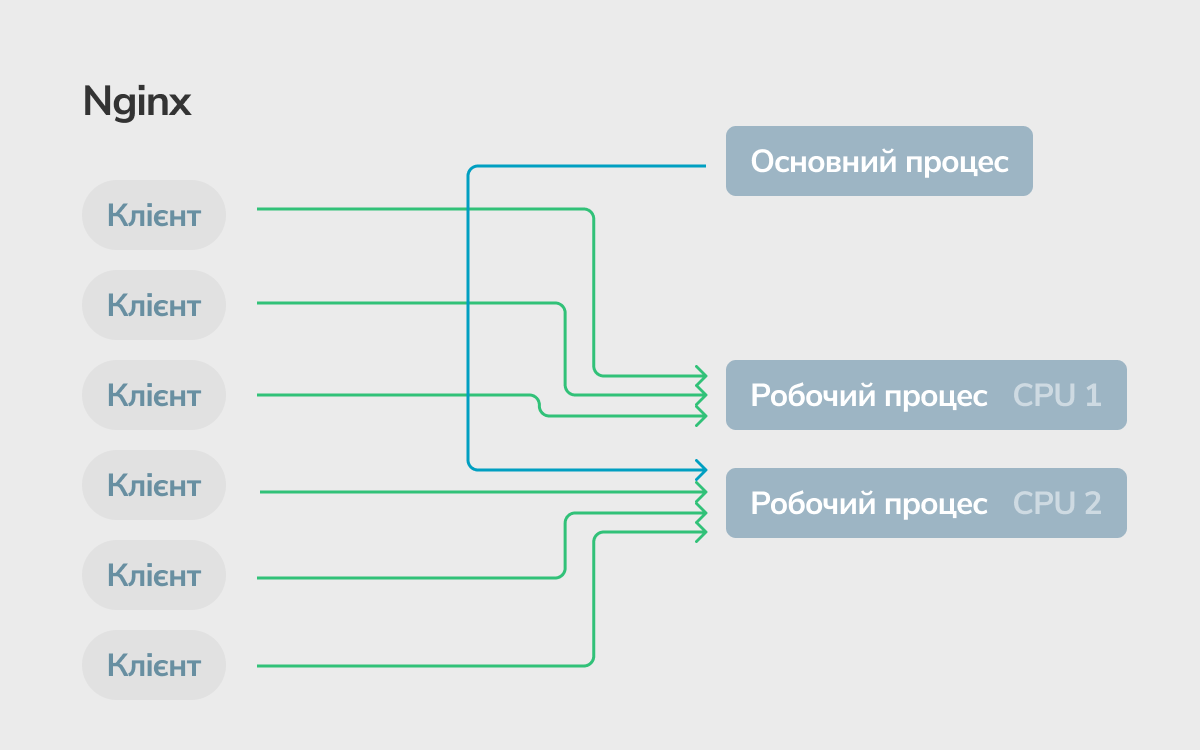
**Nginx**. Зʼявився дещо пізніше — за 10 років після виходу Apache і революційно запропонував вирішення «проблеми з 10К» — зміг обробити 10 000 одночасних зʼєднань.

Працює Nginx у такий спосіб. Є перший основний процес, у нього є кілька робочих процесів (worker-ів), їх може бути один, два. Як правило, їх створюють за кількістю ядер.

І коли виникає ситуація, коли до сервера підключається велика кількість користувачів, то нам достатньо цих двох воркерів, щоб обслугувати тисячі процесів. Саме тому один Nginx спокійно переварює 10К зʼєднань.

Технологія не потребує постійного знаходження процесу і він «прокидається» тільки тоді, коли до нього приходять дані. Виходить, що запити обробляються в безперервному циклі й Nginx споживає набагато менше пам’яті, ніж Apache.

Теоретично цей вебсервер може обробляти учетверо більше запитів на секунду, оскільки обслуговує лише статичні файли. Із цієї причини його спочатку використовували як балансувальник навантаження або зворотній проксі. А вже з розвитком ПЗ та розширенням кодової бази Nginx зміг повністю замінити Apache, а не просто працювати разом із вебсервером.

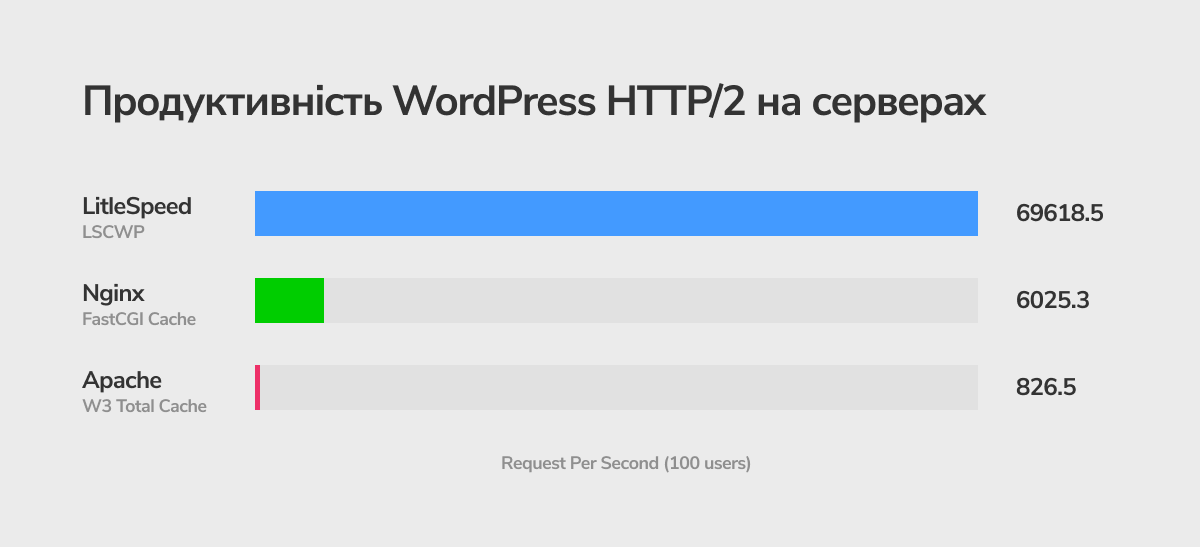


**LiteSpeed**. Високопродуктивний вебсервер від компанії LiteSpeed Technologies. Він наймолодший, однак уже встиг нажити собі славу найшвидшого та найбільш економного в плані серверних ресурсів. А саме цього часто не вистачає сайтам зі стрибками трафіку.Завдяки тому, що користувачі хостингу на вебсервері LiteSpeed мають можливість встановлювати такі плагіни, як LSCache, і використовувати кешування на рівні сервера, швидкість завантаження сайтів стає ще більшою.

З LSWS використовується також серверний API, який забезпечує дуже хорошу продуктивність динамічного контенту. Це неабияк оптимізує обмін даними між сервером і сторонніми додатками. До того ж LSAPI сумісний з PHP, RUBY, PYTHON і має відкритий вихідний код.

За останні кілька років LiteSpeed Web Server здобув широку, можна сказати навіть культову популярність серед хостингових компаній, і все завдяки оптимізованій архітектурі, а також можливостям інтуїтивно зрозумілого налаштування безпеки й кешування з боку самого користувача.

За результатами тестування від LiteSpeed Technologies вебсервер LiteSpeed обробляє запити у 12 разів швидше за Nginx і в 84 рази швидше за Apache.



**Який вебсервер вибрати**

Насправді питання з вибором вебсервера вирішується досить просто — вам потрібно лише обрати компанію, яка надає високопродуктивний хостинг. Хостинг із чистим Apache сьогодні вже навряд можна знайти. Усі провайдери зараз використовують Apache + Nginx, Ngnix + phpfpm або Litespeed вебсервер.

Ngnix підійде для обслуговування тільки статики, тоді як для міксованого вмісту краще працюватиме зв’язка Ngnix + php-fpm. А от LightSpeed — технологія з акцентом на масштабованість і буде більш актуальною для великого корпоративного сектору. Крім того, якщо є можливість на сайт встановити плагін Litespeed Cache, то тоді точно треба вибирати вебсервер Litespeed.

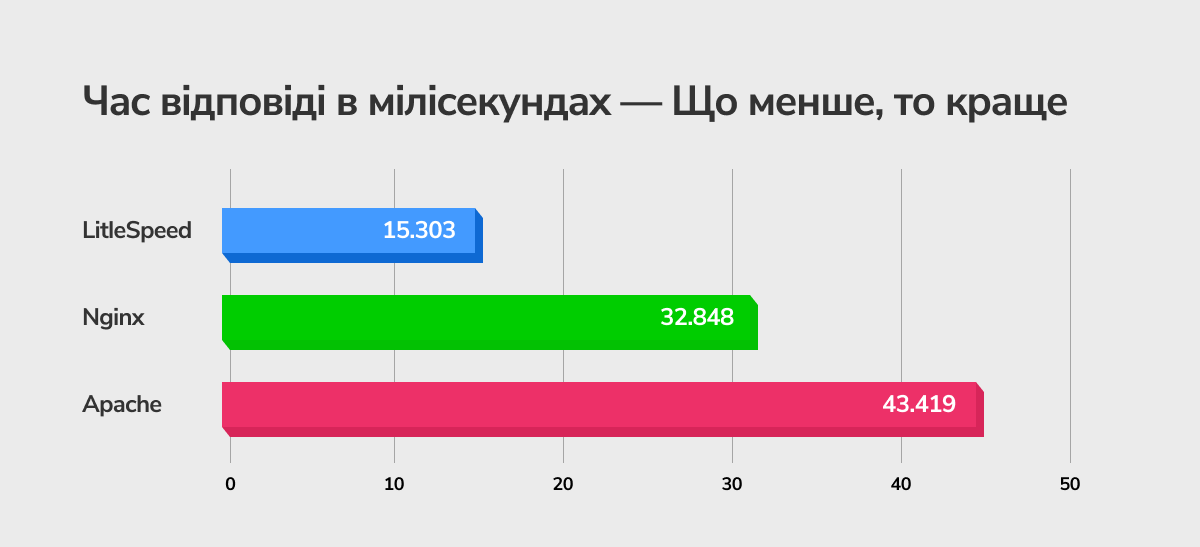
Розглянемо на прикладі українського хостинг-провайдера [HostPro](https://hostpro.ua/ua/wordpress-hosting.html?utm_source=partner&utm_medium=sendpulse&utm_campaign=blog_what_is_a_web_server).

Більшість їхніх клієнтів запускають свої сайти на PHP-додатках, як-от WordPress або Joomla, а одним із ключових компонентів LiteSpeed є lsphp — процес, який забезпечує з’єднання PHP-додатків із вебсервером. LSPHP неймовірно ефективно виконує свою роботу, особливо в порівнянні з аналогічним процесом на Apache.

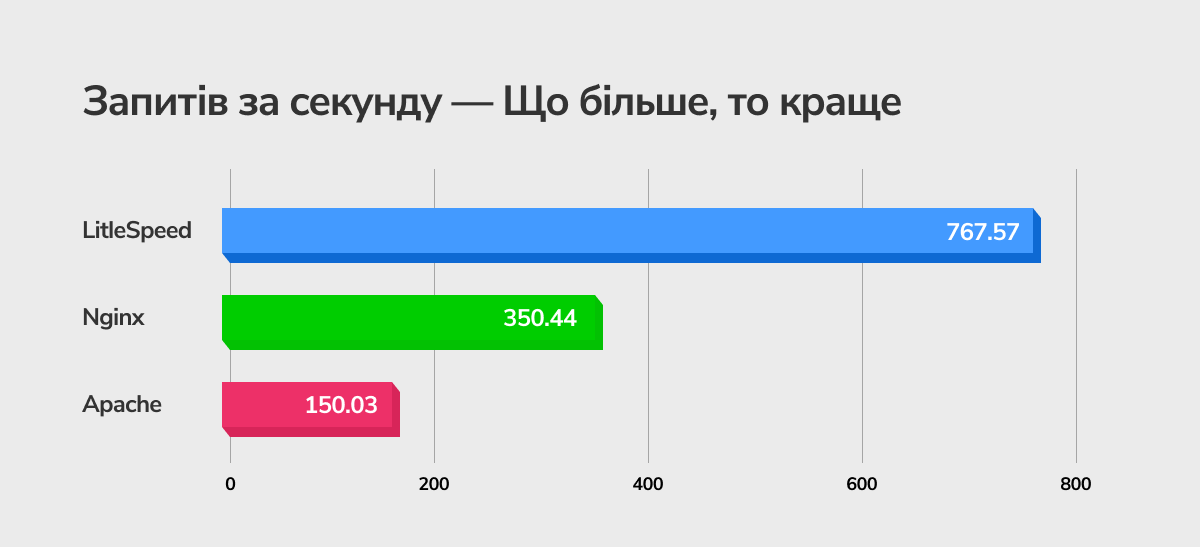
Та на теперішньому етапі розвитку хостингу і вебсерверів, саме LiteSpeed вважається ідеальним для сайтів із великою кількістю відвідувачів або з великою кількістю зображень і відео, оскільки він буде швидко завантажувати сторінки незалежно від обсягу трафіку.

Ще однією перевагою використання LiteSpeed є його масштабованість — це означає, що він може легко обробляти великі обсяги трафіку і водночас не сповільнюватись і не виходити з ладу через перевантаження. Це робить його чудовим вибором для компаній, які хочуть, щоб їхній вебсайт залишався доступним навіть у години пік або під час раптового збільшення кількості відвідувачів.

І наостанок кілька корисних результатів тестування на кожному з трьох вебсерверів. Перше — час відповіді.



Друге — кількість запитів.



Результати тестування сайтів з LiteSpeed говорять самі про себе, але варто врахувати плюси й мінуси інших вебсерверів для сайту.

Так, LiteSpeed пропонує найвищу продуктивність при мінімальних ресурсах, однак Nginx водночас забезпечує більшу гнучкість у налаштуванні вашого сервера. І, звісно, Apache залишається теж досить популярним варіантом завдяки своїй надійності та сумісності з різними операційними системами.

Але коли мова заходить про успіх вашого вебсайту, швидкість завантаження сторінок для клієнтів і відвідувачів справді має вирішальне значення. А оскільки LiteSpeed має чітку і виразну перевагу в продуктивності над конкурентами, краще обирати [хостера](https://hostpro.ua/ua/cms-hosting.html?utm_source=partner&utm_medium=sendpulse&utm_campaign=blog_what_is_a_web_server), який пропонує LiteSpeed своїм клієнтам за замовчуванням.

**2. Веб-сервери: Apache та Nginx**

**2.1 Apache HTTP Server**

**Apache** — один із найпоширеніших веб-серверів у світі, розроблений Apache Software Foundation. Його основні переваги включають гнучкість, розширюваність за допомогою модулів та активну підтримку спільноти.

**Особливості Apache**:

* 1. **Модульна структура**: Apache дозволяє додавати або вимикати функціональність через модулі (наприклад, для підтримки SSL, переписування URL).
  2. **Віртуальні хости**: Apache підтримує хостинг кількох веб-сайтів на одному сервері.
  3. **Конфігурація через текстові файли**: Apache використовує файли конфігурації, такі як httpd.conf і .htaccess, для налаштування параметрів сервера.
  4. **Велика кількість модулів**: Можливість додавати додаткові функції через модулі, як-от підтримка PHP, перетворення URL та інші.

**Переваги Apache**:

* Велика спільнота підтримки.
* Гнучкість у налаштуванні.
* Підтримка як статичного, так і динамічного контенту.

**Недоліки**:

* Більш високе використання ресурсів у порівнянні з іншими серверами (наприклад, Nginx) при великій кількості одночасних запитів.

**2.2 Nginx**

**Nginx** (вимовляється як "енджінкс") — високопродуктивний веб-сервер, який також виконує роль зворотного проксі-сервера. Його основна відмінність від Apache полягає в тому, що він побудований для ефективної обробки великої кількості одночасних з'єднань.

**Особливості Nginx**:

* 1. **Архітектура подій**: Nginx використовує неблокуючу модель на основі подій, що дозволяє обробляти тисячі з'єднань одночасно, не сповільнюючи продуктивність.
  2. **Зворотний проксі та балансування навантаження**: Nginx часто використовується для розподілу запитів між кількома серверами додатків або веб-серверами.
  3. **Підтримка статичного контенту**: Nginx швидко обробляє статичні файли, такі як HTML, CSS, JavaScript і зображення.

**Переваги Nginx**:

* Висока продуктивність при великій кількості запитів.
* Мала кількість використовуваних ресурсів.
* Простота налаштування для статичного контенту та зворотного проксі.

**Недоліки**:

* Менша гнучкість у порівнянні з Apache щодо налаштування динамічного контенту.
* Складність у конфігуруванні для новачків.

**3. Сервери додатків**

**Сервер додатків** — це програмне забезпечення, яке дозволяє виконувати додатки на сервері. На відміну від веб-серверів, які лише обслуговують статичний або динамічний контент, сервери додатків забезпечують виконання бізнес-логіки веб-додатків. Вони можуть взаємодіяти з базами даних, здійснювати обробку запитів клієнтів і забезпечувати роботу серверної частини веб-додатків.

**Приклади серверів додатків:**

* **Tomcat** — сервер додатків для роботи з Java-додатками.
* **Node.js** — сервер додатків для JavaScript, що дозволяє виконувати серверний код на JavaScript.
* **JBoss** — сервер додатків для корпоративних Java-додатків.

**Особливості серверів додатків:**

1. **Підтримка різних мов програмування**: Сервери додатків підтримують різні мови програмування та фреймворки, такі як Java, JavaScript, PHP.
2. **Інтеграція з базами даних**: Сервери додатків часто використовують бази даних для зберігання та обробки даних.
3. **Обробка бізнес-логіки**: Вони відповідають за виконання логіки додатку на стороні сервера, що дозволяє забезпечити більш складні функціональні можливості.

**4. Конфігурація та налаштування серверів**

**Конфігурація серверів** — це процес налаштування параметрів сервера для оптимізації його продуктивності та забезпечення безпеки. Налаштування включає управління доступом, конфігурацію портів, налаштування доменних імен та інші аспекти.

**Конфігурація веб-сервера Apache:**

1. **Файл конфігурації**: Основний файл конфігурації Apache називається httpd.conf, але найчастіше використовується /etc/apache2/apache2.conf.
2. **Налаштування віртуальних хостів**: Функція віртуальних хостів дозволяє хостити кілька веб-сайтів на одному сервері. Файл конфігурації для кожного віртуального хоста знаходиться в /etc/apache2/sites-available/.
   * Створіть конфігураційний файл для кожного сайту:

bash

sudo nano /etc/apache2/sites-available/mywebsite.conf

* + Додайте наступні рядки для конфігурації:

bash

<VirtualHost \*:80>

ServerAdmin admin@mywebsite.com

ServerName mywebsite.com

DocumentRoot /var/www/html/mywebsite

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

* + Активуйте віртуальний хост:

sudo a2ensite mywebsite.conf

sudo systemctl reload apache2

**Конфігурація Nginx:**

1. **Файл конфігурації**: Основний файл конфігурації Nginx зазвичай знаходиться в /etc/nginx/nginx.conf.
2. **Налаштування віртуальних хостів (серверних блоків)**:
   * Створіть конфігураційний файл для кожного сайту в директорії /etc/nginx/sites-available/.

arduino

sudo nano /etc/nginx/sites-available/mywebsite

* + Додайте конфігурацію для серверного блоку:

perl

server {

listen 80;

server\_name mywebsite.com;

root /var/www/html/mywebsite;

index index.html;

location / {

try\_files $uri $uri/ =404;

}

}

* + Активуйте серверний блок:

bash

sudo ln -s /etc/nginx/sites-available/mywebsite /etc/nginx/sites-enabled/

sudo systemctl reload nginx

**5. Порівняння Apache та Nginx**

**Apache** є більш гнучким сервером, який підходить для обслуговування як статичних, так і динамічних ресурсів. Він має безліч модулів, що дозволяють гнучко налаштовувати сервер під потреби користувача.

**Nginx** демонструє високу продуктивність при обслуговуванні великої кількості одночасних запитів і особливо підходить для статичних ресурсів. Він також чудово виконує функції зворотного проксі-сервера.

**Висновки**

Розуміння різниці між веб-серверами та серверами додатків є критичним для побудови ефективних клієнт-серверних архітектур. Apache та Nginx — це два основні веб-сервери, що мають свої переваги та недоліки, а сервери додатків дозволяють виконувати бізнес-логіку веб-додатків. Правильна конфігурація та налаштування серверів мають вирішальне значення для досягнення високої продуктивності та безпеки в сучасному веб-середовищі.