

4. Визначення оптимальних стратегій: Вони допомагають визначити оптимальні рішення для досягнення конкретних економічних цілей, таких як зростання ВВП, зменшення безробіття тощо.

5. Оцінка ефективності політик: Математичні моделі допомагають визначити, як різні господарські політики та реформи можуть вплинути на економіку, а також оцінювати, які з них є найбільш ефективними.

У загальному контексті, математичні моделі допомагають створювати аналітичний фреймворк для розуміння складних економічних процесів та прийняття обґрунтованих рішень для забезпечення стабільного та стійкого економічного розвитку.

Математичні моделі – це ключ до інноваційного розвитку та створення стійкого майбутнього.

ЛІТЕРАТУРА

1. «Macroeconomics» by N. Gregory Mankiw
2. «Introductory Econometrics: A Modern Approach» by Jeffrey M. Wooldridge
3. «Dynamic Macroeconomic Theory» by Thomas J. Sargent
4. «Econometric Models, Techniques, and Applications» by Michael D. Intriligator
5. «Climate Change Modeling: An Assessment of the State-of-the-Art» by E. Roe and M. Baker.
6. «Dynamic Stochastic General Equilibrium Models: Assessing Their Properties and Presenting an Application to New Zealand» by O. Lancee and G. C. Lim.
7. «Mathematical Models in Epidemiology» by F. Brauer and C. Castillo-Chavez.

ОПТИМІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ WEB-СЕРВЕРА NGINX

Лотюк Ю. Г.

кандидат педагогічних наук, доцент

Приватного вищого навчального закладу

*«Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем'ячука»*

Web-сервер на відміну від звичайного файлового сервера потребує особливої підтримки, як у вигляді спеціалізованого програмного забезпечення, так і фізичного обладнання. Тому що Web-сервер орієнтований на звернення від браузерів, антивірусів, цифрової побутової техніки IoT. На відміну від файлового сервера, де запити надходять лише від користувачів.

Web-сервер має ефективно працювати в режимі активного навантаження. Під час вибору Web-сервера важливо оцінити сумісність з операційною системою не лише його компонентів, але і програмного забезпечення, щоб сервер працював максимально швидко та безперебійно.

Окрім роботи з Web-сторінками Web-сервер веде статистику запитів, взаємодіє з сервісами електронної пошти, виконує функцію авторизації чи аутентифікації, перевіряє сертифікати безпеки тощо.

Щоб підібрати оптимальне поєднання апаратного та програмного забезпечення, маємо врахувати кількість запитів в одиницю часу, обсяг жорсткого диска для зберігання даних, його фізичний розмір та місце зберігання бази даних [1]. Актуальною є задача правильного вибору та подальшого налаштування Web-сервера для надійної та безперебійної його роботи.

Дослідивши процес обміну інформацією Web-сервера з браузерами, компонентами IoT, гаджетами тощо, та проаналізувавши його, були створені файли конфігурації Web-сервера для зручного, простого у використанні та швидкого обміну даними.

Досліджувався Web-сервер, розгорнутий на базі операційної системи Linux, його налаштування, підключені модулі, задіяні ресурси, налаштування параметрів операційної системи для забезпечення роботи Web-сервера, процес формування відповідей Web-сервером, їх оптимізація, стиснення, підлаштування під пристрій з якого прийшов запит [5].

Сервер Nginx - це спеціалізоване програмне забезпечення, яке керує обробкою запитів користувачів (HTTP-запити від клієнтів мережі – web-браузерів) і генерує відповіді у вигляді HTML-коду, які web-браузери клієнтської сторони інтерпретують у формат Web-сторінок, що легко читаються [3].

На даний момент Web-сервер Nginx є одним з найнадійніших серверних програм завдяки керованій подіями асинхронній архітектурі [2]. Основою Nginx є асинхронні алгоритми. Сервер генерує робочі процеси, завдяки чому йому вдається одночасно обробляти величезну кількість запитів, що надходять. Усі робочі процеси виконуються незалежно один від одного. У межах кожного робочого процесу виконуються робочі з'єднання. Обробка з'єднань відбувається лише в тому випадку, якщо було згенеровано нову подію. Таким чином, маємо тришарову архітектуру Web-сервера: 1. Робочі з'єднання – структурні одиниці робочого процесу; 2. Запити надходять від робочих з'єднань до робочих процесів; 3. Всі дані надходять до головного процесу, що надає кінцеві результати обробки [3].

Nginx - однопотокний сервер, що не генерує процеси для кожного нового з'єднання. Цим обумовлена рівномірність використання ресурсів фізичного сервера (процесор, ОЗП) навіть під час обробки великої кількості запитів.

Продуктивність. Nginx показує високу швидкість обробки підключень статичного контенту. За цим показником він обходить найближчого конкурента Apache вдвічі. Продуктивність при роботі з динамічними сайтами обох програмних продуктів приблизно однакова.

На відміну від Apache швидше відбувається обробка запитів, пов'язаних із статичним контентом. Однак Nginx не містить алгоритмів для самостійної обробки запитів до динамічних даних. Для цього використовується зовнішній процесор, який виконує функції обробки та повертає підсумковий результат Nginx.

Деякою незручністю є відсутність механізму динамічного підключення таких модулів, як шифрування, проксіювання, поштові функції тощо. Це підвищує безпеку, проте знижує гнучкість рішення порівняно з Apache.

Web сервер – це перша ланка в роботі будь-якого Web сайту. Він приймає запит від клієнта, формує відповідь та надсилає його назад клієнту (рис. 1). Коли кількість таких запитів зростає, швидкість роботи Web-сервера буде падати.

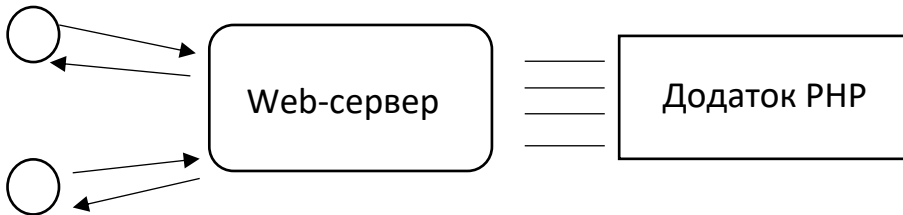


Рис.1. Робота Web сервера

Для підвищення ефективності роботи Web-сервера необхідно або провести стиснення запитів, або зменшити кількості запитів.

Усі сучасні браузери підтримують роботу зі стисненням Gzip. Web-сервер стискає вміст відповіді перед відправкою його клієнту, а браузер розпаковує його в момент отримання. Так можна заощадити до 70 % розміру файлу. Для клієнтів це означатиме вищу швидкість роботи Web сайту [5].

Для зменшення кількості запитів на сервер, потрібно мінімізувати кількість зовнішніх запитів зі сторінки, наприклад шляхом склеювання css/js, [5]. У файлі налаштувань mime.types потрібно визначити, який тип передається компресору.

Ефективність роботи Web-сервера підвищуємо шляхом внесення оптимальних параметрів у файл конфігурації nginx.conf, який знаходиться за адресою /usr/local/etc/nginx/nginx.conf.

Серверні процесори фактично являють собою декілька окремих процесорів, пов'язаних дуже швидкою шиною. Вони спільно використовують ресурси і утворюють NUMA (Non-Uniform Memory Access, архітектура з неоднорідним доступом до пам'яті).

Для правильного використання NUMA потрібно розглядати кожен її вузол як окремий сервер.

Для сервера потрібно перевіряти: кількість вузлів; обсяг пам'яті кожного вузла, кількість процесорів на кожному вузлі, відстань між вузлами. На (рис.2) та (рис.3) показано приклад перевірки для процесора ноутбука.

```
[user@fedora ~]$ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0
node 0 size: 1917 MB
node 0 free: 73 MB
node distances:
node 0
0: 10
[user@fedora ~]$
```

```
[user@fedora ~]$ numastat -n -c
Per-node numastat info (in MBs):
                Node 0 Total
                -----
Numa_Hit         7757  7757
Numa_Miss         0     0
Numa_Foreign     0     0
Interleave_Hit   5     5
Local_Node       7757  7757
Other_Node       0     0
[user@fedora ~]$
```

Рис. 2. Аналіз вузлів NUMA у Linux **Рис. 3.** Перевірка кількості вузлів та обсягу пам'яті у кожному з вузлів

Дослідження проводилися для одиночного Web-сервера. У результаті дослідження створені файли конфігурації Nginx та файли конфігурацій ОС Linux, які забезпечують найбільш ефективну роботу Web-сервера в парі з операційною системою.

Досліджені технічні параметри роботи Web-сервера Nginx для мобільних гаджетів, браузерів та елементів IoT, ці результати використані для збільшення продуктивності сервера.

ЛІТЕРАТУРА

1. Оптимальна конфігурація веб-серверу. Internet publications. URL: <http://www.nas.gov.ua/> (дата звернення: 23.05.2023).
2. Веб-сервер NGINX для Linux. Internet publications. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-nginx/> (дата звернення: 28.08.2023).
3. Ігорь Грегорченко // Оптимизация Web сервера. Internet publications. URL: <https://highload.today/optimizatsiya-web-servera/> (дата звернення: 28.08.2023).
4. Ігорь Грегорченко // Детальное описание настройки Nginx для оптимальной работы. Internet publications. URL: <https://highload.today/index-php-2009-04-24-nastroyka-nginx/> (дата звернення: 28.08.2023).
5. Макс Матюхин // Оптимизация веб-серверов для повышения пропускной способности и уменьшения задержки. Internet publications. URL: <https://highload.today/optimizatsiya-web-servera/> (дата звернення: 28.0.2023).

АНАЛІЗ І КЛАСИФІКАЦІЯ КВАНТОВИХ МЕРЕЖ ДЛЯ ПОБУДОВИ ІНТЕРНЕТУ

Павлюк Д. Ю.

*здобувач третього (освітньо-наукового) рівня вищої освіти
Державного університету
«Житомирська політехніка»*

Актуальність дослідження в області квантових мереж обумовлена постійним