

УДК 004:338.48

Р.О. Вакульчук, К.В. Мельник, В.М. Мельник
Луцький національний технічний університетАНАЛІЗ ПРОДУКТИВНОСТІ РОБОТИ WEB-СЕРВЕРІВ APACHE ТА NGINX ПРИ ПІКОВИХ
НАВАНТАЖЕННЯХ

В статті розглядається питання ефективного використання ресурсів веб-серверів та якісного обслуговування користувачів сайтів. Було проведено порівняння параметрів роботи двох популярних веб-серверів Apache та Nginx. Проаналізовано їх сильні та слабкі сторони. Проведено оптимізацію налаштувань Apache.

Ключові слова: веб-сервер, інтернет, навантаженість, ресурси.

Табл. 1. Рис. 5. Літ. 8.

Постановка проблеми. Оскільки сучасні сайти не просто відображають певну текстову інформацію на сторінці, а містять чимало мультимедійної інформації, яка зберігається у об'ємних базах даних та динамічно змінюється. Тому сучасні сайти потребують чимало ресурсів і повинні розміщуватись на потужних та надійних серверах. Однак велику частину навантажень на сервері генерує спам.

Спам та інші електронні зловживання вже давно в центрі уваги досліджень комп'ютерної безпеки. Проте, останні роботи не піддаються пасивним методам вимірювання. Дослідники стали посередниками або активними учасниками спам поведінки; методологій, які лежать в стику правових, етичних і людських взаємин (наприклад, IRB) [3].

Через величезну кількість користувачів і сайтів веб-сервер отримує велике навантаження, яке позначається на його працездатності і надійності. Через велике навантаження веб-сервер може або постійно не справлятися з обслуговуванням потоку поданих йому заявок, або відчувати короточасні пікові перевантаження. З виниклими проблемами можна боротися двома способами: у першому випадку простим, але дорогим способом боротьби з перевантаженістю буде підвищення потужності обладнання (сервера). У другому випадку дешевше буде в моменти перевантаженості обслуговувати в першу чергу найбільш важливі, які мають високий пріоритет заявки. В обох випадках часто можна усунути перевантаженість веб-сервера, реорганізувавши обчислювальний процес. У теорії масового обслуговування доведено (формула Поллачека-Хінчина), що при одному і тому ж коефіцієнті завантаження обладнання середній час перебування заявки в системі тим менше, чим менше вхідний потік заявок і час обслуговування відрізняються від детермінованих. Інакше кажучи, чим менше час обслуговування і проміжки між надходженням заявок відрізняються від середніх, тим менше часу очікування користувачем сайту відповіді на його запит. Неоднорідність потоку заявок визначається типом веб-сайту. Наприклад, веб-сайт для перегляду розкладу електричок виконує запити приблизно однаковою трудомісткістю. Запити до пошукової системи Google можуть мати трудомісткість, що розрізняється в сотні разів. Підбираючи суміш заявок на виконання з числа, що стоять у черзі, можна значно рівномірніше завантажити устаткування комп'ютера (процесор, диски, оперативну пам'ять і т.д.), тим самим, збільшивши продуктивність системи. Теорія дає лише загальний напрямок вироблення стратегії управління обчислювальним процесом. Практичні способи оптимізації завантаження обладнання для конкретної обчислювальної системи, на якій встановлено веб-сервер, можна підібрати лише емпіричним шляхом. Експерименти з працюючим веб-сервером можуть порушити його роботу або вивести всю систему з ладу, що є недоцільним. Тому для вивчення властивостей веб-сервера краще використовувати імітаційне моделювання [7], або перевіряти завантаженість на віртуальних серверах.

Тому важливо вибрати не тільки потужний комп'ютер, але й відповідне надійне програмне забезпечення для його роботи. У роботі розглянуті та протестовані найпопулярніші web-сервери.

Аналіз останніх досліджень і публікацій. Значний внесок у розвиток цих питань внесли праці Бондара О.С. [9], П. Сешарді[4], Р. Бовена [2], Ендрю Вест [3], Манасової К. С. [7] та інші.

За статистикою Netcraft на жовтень 2012 року одним з найпоширеніших серверів є Apache (рис.1), тому всі дослідження та використання методів оптимізації будуть проводитись саме для цього сервера [1]. Для порівняння ефективності роботи даного серверу наводиться характеристика роботи сервера Nginx, як одного із найпоширеніших.

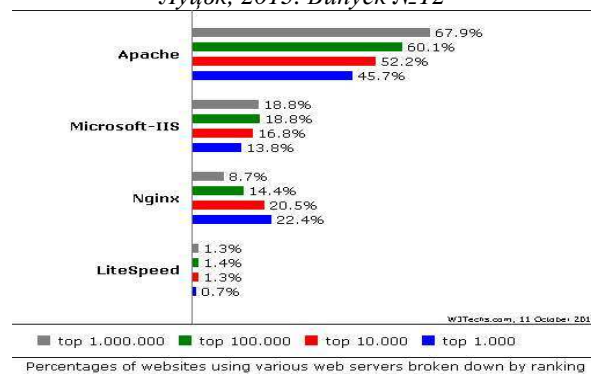


Рис. 1. Співвідношення використання різних веб-серверів на найпопулярніших сайтах світу [№ 1]

Як відомо спочатку Apache був розроблений в якості патча до вже існуючого web сервера NCSA httpd у 1995 році. Подальшим розвитком сервера займалися Б. Белендорф (Apache HTTP Server), Р. Тау (змінив серверну архітектуру). Перша версія Apache була випущена ще у 1995 році. З того часу були розроблені версії для більшості операційних систем та комп'ютерних архітектур.

Кожен адміністратор веб-ресурсу повинен вміти не лише встановити сервер, а і налаштувати його роботу під відповідне навантаження користувачами. Адже якщо перехід між сторінками сайту буде дуже повільним, то швидше за все, користувач не дочекається закінчення завантаження і покине сайт, що є неприйнятним для рейтингу сайтів, та їхнього комерційного успіху.

Подробиці установки і настройки сервера описані в монографії Р. Бовена [2], однак автор детально не описує в ній методи оптимізації.

Зауважимо також, що оптимізації можна досягти не тільки шляхом вибору і настройки вище описаних компонент, але і шляхом редагування стандартної конфігурації сервера.

Таким чином, збільшенню швидкості роботи web сервера сприяє вибір *MPM (Multi-Processing Modules)* модулів і оптимізація конфігурації під конкретну задачу. Стандартні приклади конфігурації, без оптимізації, розглянуті в електронному ресурсі [5].

Сервер Apache кожен запит обробляє у своєму процесі або потоці. При компіляції web сервера є можливість вибору одного з декількох MPM, які відповідають за прослуховування портів, прийом запитів і роздачу запитів дочірнім процесам або потокам, в яких ці запити будуть оброблені.

П. Сешарді проводив дослідження суперскалярних мікропроцесорів для побудови корпоративних web серверів. Останнє десятиліття спостерігається розвиток динамічно прогнозуючих суперскалярних мікропроцесорів. Вони стали комерційно успішними не зважаючи на те, що навантаження від сайтів інтернет-магазинів, великих баз даних та подібних обрахунків не є спеціалізованими для даного типу мікропроцесорів [4].

Невирішені частини проблеми. Попри широкий спектр досліджень, наявність великої кількості довідкових матеріалів та інструкцій актуальним залишається питання оптимального налаштування web-серверів, визначення їхньої відмовостійкості та пікової кількості користувачів, які можуть одночасно обслуговуватись сервером.

Метою дослідження є визначення критичної точки навантаження на сервер, дослідження часу відповіді при піковому навантаженні і визначення показника відмов.

Основні результати дослідження. Для тестування навантаження серверів у роботі використовується сайт, розроблений на CMS Drupal 7. При цьому, на однакових за характеристиками серверах з однаковою версією PHP та MySQL, розміщено копію даного сайту та проведено тестування навантаження з вимірюванням швидкості відповіді сервера та видачі сторінок. При тестуванні використовується Apache Jmeter 2.9. Даний продукт дозволяє генерувати віртуальних користувачів, які відвідують сервер по певних сторінках, з вказанням кількості користувачів, які перебувають на сайті одночасно та кількості переглянутих сторінок. Для звітування надається широкий вибір інструментів, які показують максимальні і мінімальні значення часу завантаження, дають повний табличний звіт а також можливість побудови графіків та багато інших варіантів.

Було проведено початкове тестування сервера Nginx. При одночасних запитах від 100 користувачів час відповіді сервера значно зріс, і на сторінках, які передають більший об'єм

інформації, сервер видав помилки після очікування в 60с. Результати тестів при змінних навантаженнях від 50 до 500 користувачів на хвилину зведено у таблиці 1.

Таблиця 1. Результати тестування веб-сервера Nginx.
 Авторська розробка

Кількість запитів	Час відповіді	Кількість відмов	Перша відмова
50	458	0	0
100	449	0	0
150	545	0	0
200	2085	0	0
250	9676	0	0
300	18126	0	0
350	24059	2	237
400	27460	42	191
450	27866	92	168
500	28668	142	148

Для встановлення функціональної залежності між кількістю запитів та часом їх обробки проводиться кореляційно-регресійний аналіз, на основі якого можна зробити прогнозування відмовостійкості. На рис. 2-4 наведено результати проведеного аналізу, а саме рівняння регресії та коефіцієнт кореляції. Великий коефіцієнт кореляції вказує на функціональну залежність між досліджуваними величинами.

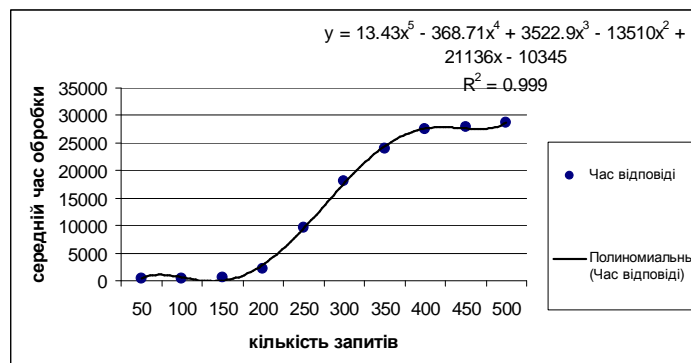


Рис. 2 Залежність середнього часу обробки запиту від кількості запитів для сервера Nginx
 Авторська розробка

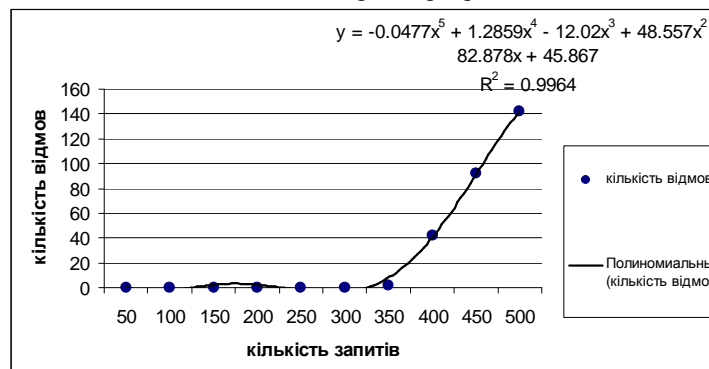


Рис. 3 Залежність кількості відмов від кількості запитів для сервера Nginx. Авторська розробка



Рис. 4 Залежність кількості відмов від часу обробки для сервера Nginx
 Авторська розробка

Аналогічне дослідження проведено для веб-сервера Apache (рис.5-6). Штриховою лінією представлені результати для сервера Nginx з метою їх порівняння.

Розглядаючи методи оптимізації роботи web-сервера Apache Бондар О.С. згадує такі легкодоступні та корисні методи як: стиснення трафіку (параметр Deflate у конфігурації сервера); KeepAlive і KeepAliveTimeout параметр, який дозволяє робити декілька запитів, для одного TCP підключення (завантаження зображень на сторінці); використання кешування сторінок на стороні клієнта, або на сервері; встановлення максимальної кількості одночасно підключених до сервера користувачів (параметр MaxClients у налаштуваннях [9]).

Оскільки, при тестуванні сервер отримує запити від кількох сотень віртуальних користувачів, то параметр MaxClients було збільшено зі 150 до 300. По замовчуванню опція KeepAlive увімкнена, KeepAliveTimeout зменшено з 300 до 60, оскільки довге завантаження сторінки не є бажаним для користувачів. Ці зміни вносяться в основний конфігураційний файл web-сервера Apache, який розташовується /etc/apache2/apache.conf. Результати тестів при змінних навантаженнях без оптимізації та з оптимізацією зведено у таблиці 2.

Таблиця 2. Результати тестування веб-сервера Apache.
 Авторська розробка

Кількість запитів	Стандартна конфігурація			Оптимізований варіант		
	Час відповіді	Кількість відмов	Перша відмова	Час відповіді	Кількість відмов	Перша відмова
50	779	0	0	438	0	0
100	464	0	0	433	0	0
150	505	0	0	437	0	0
200	1798	0	0	741	0	0
250	5254	182	69	3344	0	0
300	8340	300	6	5290	228	73
350	11021	350	5	7904	341	10
400	13261	400	4	12039	393	8
450	15974	450	5	8284	442	9
500	19378	500	4	13690	495	6

Як видно з таблиці 2, збільшення параметра MaxClients та зменшення параметра KeepAliveTimeout дозволяє збільшити швидкість видачі сторінок клієнтам майже в 2 рази, особливо це помітно при частоті запитів від 200 до 500 запитів за хвилину. А відмови у роботі сервера після оптимізації з'являються при частоті 300 запитів за хвилину.

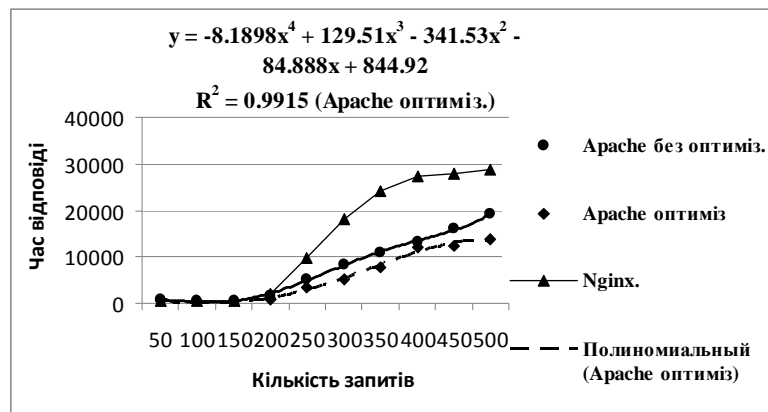


Рис. 5 Залежність середнього часу обробки запиту від кількості запитів для серверів Nginx та Apache
Авторська розробка

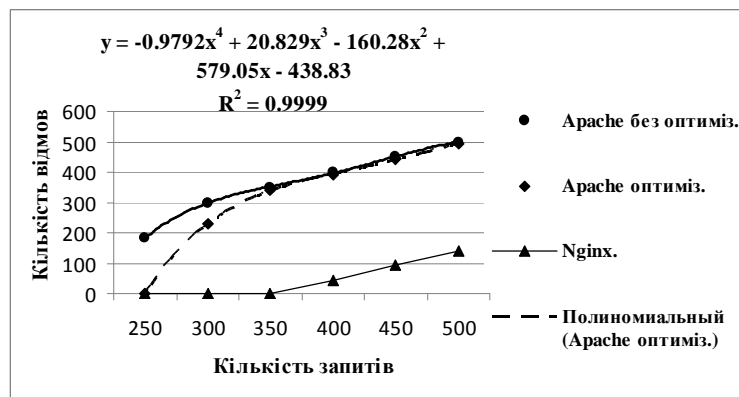


Рис. 6 Залежність кількості відмов від кількості запитів для сервера Nginx. Авторська розробка

Як видно з рис.5 при великій частоті запитів швидкість видачі сторінок клієнтам сервера Apache швидше ніж Nginx, однак для сервера Nginx показник відмов в сотні разів менший.

Висновки. Проведені дослідження дозволяють вибрати оптимальні конфігурації веб серверів і передбачувати умови їх критичного навантаження. Показано, що за допомогою основних параметрів конфігураційного файлу Apache можна збільшити швидкість роботи сервера та його пропускну спроможність. Знайдені рівняння регресії можна використовувати для розрахунку моменту настання критичного навантаження для запобігання збоїв у роботі сервера.

1. October 2012 Web Server Survey. [Електронний ресурс] – Режим доступу: <http://news.netcraft.com/archives/2012/10/02/october-2012-web-server-survey.html#more-6650>
2. R. Bowen. Apache Cookbook: Solutions and Examples for Apache Administrators. – 2008. – 310.
3. Andrew G. West, Pedram Hayati, Vidyasagar Potdar, and Insup Lee. Spamming for Science: Active Measurement in Web 2.0 Abuse Research [Електронний ресурс] – Режим доступу: http://repository.upenn.edu/cgi/viewcontent.cgi?article=1524&context=cis_papers
4. Pattabi Seshadri and Lizy K. John University of Texas at Austin. Characterization of Web Server Workloads on Three Generations of IBM PowerPC Microarchitectures [Електронний ресурс] – Режим доступу: <http://www.research.ibm.com/acas/projects/00seshadri.pdf>
5. Приемы конфигурирования web сервера Apache 2. [Електронний ресурс] – Режим доступу: http://web.ixit.ru/pdf/apache/apache_config.pdf
6. J. Marantz, I. Grigorik A. Make the web faster with mod_pagespeed. [Електронний ресурс] – Режим доступу: <http://googledevelopers.blogspot.com/2012/10/make-web-faster-with-modpagespeed-now.html>
7. Манасова Каринэ Сергеевна. Семейство имитационных моделей для исследования характеристик веб-серверов. [Електронний ресурс] – Режим доступу: <http://sibac.info/index.php/2009-07-01-10-21-16/2121-2012-04-19-09-06-51>
8. Александр Астапенко. Насколько быстры сайты в разных странах мира? [Електронний ресурс] – Режим доступу: <http://webperformance.ru/2012/05/02/wpo-around-the-world/>
9. Бондар О.С. Методи оптимізації передачі контенту Web-серверами. Інформаційна безпека, №2 (8), 2012, с. 23-26.