

НАВАНТАЖУВАЛЬНЕ ТЕСТУВАННЯ ДЛЯ ПЕРЕВІРКИ СТАБІЛЬНОСТІ РОБОТИ WEB-РЕСУРСУ

В даний час, з постійно зростаючим числом інтернет користувачів, постає необхідність створення систем здатних працювати під колосальними навантаженнями без видимих уповільнень роботи і збоїв. Будь-якого роду проблеми пов'язані з недостатньою продуктивністю, можуть стати причиною відмови клієнтів від використання Вашого web-ресурсу, навіть якщо функціонально він їх влаштував. У зв'язку з цим, проведення якісного тестування навантаження повинне стати обов'язковим, для забезпечення стабільності роботи web-ресурсу.

Для вирішення проблеми продуктивності існують різні види тестування, кожен з яких спрямований на діагностику окремої частини системи. Для інформаційних систем мережі Інтернет одним із найбільш значущих є тестування продуктивності web-ресурсу. В свою чергу тестування продуктивності поділяється на такі типи [1,2]:

□ тестування навантаження — форма тестування, яка визначає працездатність системи для того, щоб оцінити поведінку додатку (сайту) із заданим наперед очікуваним навантаженням.

□ стресове тестування — тестування, що дозволяє перевірити наскільки додаток і система в цілому працездатні в умовах стресу, а також оцінити здатність системи до регенерації, тобто до повернення до нормального стану після припинення впливу стресу.

□ об'ємне тестування — проводиться для визначення продуктивності додатка в умовах різкого збільшення користувачів. Основним параметром, на якому слід зосередитися під час об'ємного тестування, є час відгуку.

□ тестування стабільності (надійності) — це перевірка працездатності web-ресурсу при тривалому (багатогодинному) тестуванні з середнім рівнем навантаження.

□ тестування на гнучкість — це тестування, яке проводиться з метою виміру можливості системи адаптуватися до певних умов.

Кожне з них направлено на перевірку різних аспектів функціонування досліджуваного web-ресурсу.

Перелічені типи тестування дозволяють вимірювати продуктивність на основі статистичних даних, які одержуються із результатів планових вимірювань з подальшою чисельною обробкою. Це являє собою ефективний спосіб з аналізу функціонування продуктивності, що дозволяє систематично знаходити обмеження, помилки і підозрілі події поведінки, не тільки мережевих компонентів, а й самої логіки роботи web-ресурсу.

Для створення будь-якого web-ресурсу зазвичай використовують трирівневу архітектуру. Така архітектура включає в себе наявність ланцюжка «сервер бази даних — web-сервер — клієнтський комп'ютер і браузер». Сервер бази даних підтримує СУБД і виконує всю роботу з даними. У базі даних зберігається різного роду інформація, а також службова інформація, яка включає дані про користувачів, і дані про роботу самого web-ресурсу. Веб-сервер — виконує функції HTTP-сервера, сценарного інтерпретатора і обробляє дані із баз даних. Браузер виконує функції за поданням інформації на комп'ютері клієнта. Сукупність таких технологій дозволяють реалізувати інтеграцію між різними системами.

При тестуванні продуктивності web-ресурсу зазвичай використовують навантаження WEB-сервера з великою кількістю звернень віртуальних користувачів. На web-сервер досліджуваного web-ресурсу, що тестується відправляються тестові запити з великою інтенсивністю. Він відповідає на кожен запит, і ці відповіді зберігаються у зручному для користувача вигляді, а в подальшому аналізуються. Для більшої достовірності кожен з віртуальних користувачів має «ходити» по сайту за індивідуальним сценарієм та мати особисті параметри. Наприклад, встановлюється таймер для одного користувача, щоб визначити, скільки часу займе виконання запиту, коли тисячі інших посилають запити на той же самий сервер в той же самий час. Також в процесі тестування можна імітувати короткочасні піки навантаження, коли кількість відвідувачів стрибкоподібно збільшується або ж зменшується. Такий тест дає змогу визначити час відгуку WEB-сервера при змінних навантаженнях, а також максимальну кількість транзакцій, яку сервер може виконати за певний період часу.

В процесі дослідження були проведені проміжні навантажувальні тести продуктивності web-ресурсу www.fizmat.tnpu.edu.ua.

У загальному вигляді процес навантажувального тестування складається з таких етапів [4]:

- аналіз вимог і збір інформації по системі, що тестується;
- конфігурація тестового стенду для навантажувального тестування;
- розробка моделі навантаження;
- вибір інструмента для навантажувального тестування;
- створення та налагодження тестових скриптів;
- проведення тестування;
- аналіз результатів;
- підготовка, відправлення та публікація звіту по проведеному навантажувальному тестуванню.

Деякі етапи можуть бути змінені, або ж опущені в залежності від вашого типу тестування, а також вимог які надаються. В даному дослідженні всі етапи було пройдено без змін.

Для проведення тестування навантаження WEB-ресурсів, використовується спеціальне програмне забезпечення, що дозволяє створювати необхідне навантаження на сервері і заміряти контрольні параметри. На сьогоднішній день існує багато інструментів, а також онлайн сервісів для навантажувального тестування. Для навантажувального тестування можна використовувати такі засоби:

- безкоштовні інструменти: Yandex tank, Apache Jmeter, Tsung та ін.;
- онлайн сервіси: <https://loadimpact.com/>, <http://loaddy.com/>, <http://www.alertra.com/> та багато інших.

Після ознайомлення з можливостями перелічених інструментів, для вирішення поставленої проблеми було обрано Apache JMeter.

Apache JMeter (jakarta.apache.org/jmeter) є Java-додатком з відкритим кодом, призначеним для навантажувального тестування не тільки веб-додатків і їх окремих компонентів (скрипти, сервлети, Java об'єкти й ін.), але також FTP-серверів, баз даних (з використанням JDBC) та мережі. Функціональність розширюється за допомогою плагінів. Підтримується SSL (через Java Secure Sockets Extension). Можливе проведення тестів, як з використанням графічного інтерфейсу, так і з командного рядка. За рахунок підтримки кросплатформеності, JMeter працює в різних *nix-системах, в Windows від 98 і деяких інших ОС. Розповсюджується під Apache License [3].

У JMeter передбачені механізми авторизації віртуальних користувачів, підтримуються користувальницькі сеанси, шаблони, кешування і подальший offline аналіз результатів тесту, ці функції дозволяють сформулювати новий запит, ґрунтуючись на відповіді сервера на попередній. Є також можливість проводити розподілені тести.

Для початку роботи необхідно запустити ApacheJMeter.jar або в консолі jmeter.bat (Windows) або jmeter.sh (*nix). JMeter має вбудований проксі-сервер, який призначений для запису сесій, але можна використовувати і зовнішній. Перед початком тестування необхідно скласти тестовий план, що описує серію завдань, які необхідно виконати JMeter. Він повинен містити одну або кілька груп потоків (Thread Groups) та інші елементи:

- Логічні контролери (Logic controllers);
- Типові контролери (Sample generating controllers);
- Слухачі (Listeners);
- Таймери (Timers);
- Відповідності (Assertions);
- Конфігураційні елементи (Configuration elements).

На першому етапі додаємо групу потоків (Edit - Add - Thread Group). В її налагодженнях вказуємо назву, кількість виконуваних потоків що запускаються, тобто віртуальних користувачів (Number of threads), час затримки між запуском потоків (Ramp-Up Period), кількість циклів виконання завдання (Loop Count), тут же можна визначити виконання завдання за розкладом (Scheduler).

Далі, обираючи створену групу, необхідно додати зразок запиту (Sampler), вибравши його зі списку. Для навантажувального тестування, або перевірки працездатності сервера треба вибрати один із елементів конфігурації. Одним з таких елементів конфігурації є HTTP Request

(Add -Sampler - HTTP Request), де вказуємо назву, IP-адресу і порт веб-сервера, протокол, метод передачі даних (GET, POST), параметри переадресації, передачу файлів на сервер. Нажимаємо кнопку Run і тест розпочинається [Рис. 1].

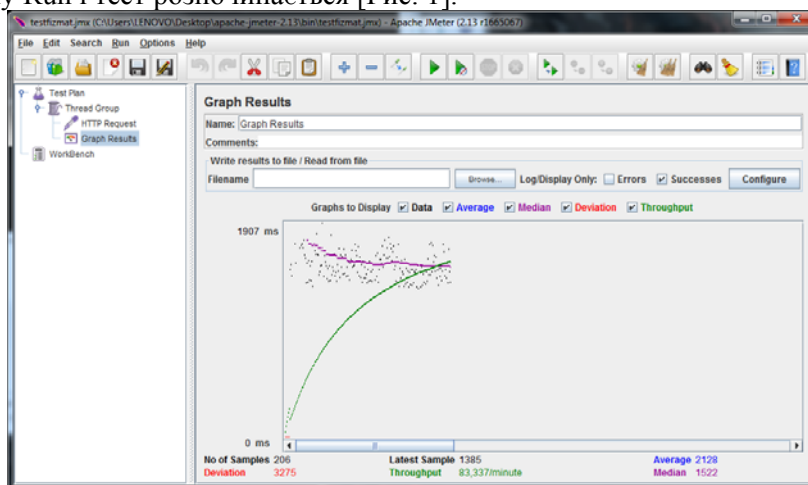


Рис. 1. Load Testing

В результаті використання даного інструмента, ми можемо отримати метрики, які дозволяють оцінити продуктивність нашого web-ресурсу в реальних умовах. Проаналізувавши метрики, ми також можемо сформулювати рекомендації, щодо оптимізації та покращення нашого web-ресурсу. Тестування — процес циклічний: аналіз результатів можна використати, як основу при плануванні наступного циклу досліджень.

ЛІТЕРАТУРА:

1. Нагрузочное тестирование или тестирование производительности: [Електронний ресурс]. — <http://www.protesting.ru/testing/types/loadtesttypes.html>. — Назва з екрану.
2. Основы тестирования производительности приложения: [Електронний ресурс]. — <http://ru.agileload.com/agileload/blog/2012/11/09/application-performance-testing-basics>. — Назва з екрану.
3. Под предельной нагрузкой: обзор программ нагрузочного тестирования веб-серверов: [Електронний ресурс]. — <https://xakep.ru/2008/04/21/43327/>. — Назва з екрану.
4. Этапы проведения нагрузочного тестирования: [Електронний ресурс]. — <http://www.protesting.ru/automation/load/loadteststages.html>. — Назва з екрану.

Черемшинська О.

Науковий керівник - доц. Лотоцький В. А.

ЛІНІЙНІ ПОПЕРЕЧНИКИ КЛАСІВ $B_{p,\theta}^\Omega$ ПЕРІОДИЧНИХ ФУНКЦІЙ БАГАТЬОХ ЗМІННИХ У ПРОСТОРІ L_p

У роботі встановлюються порядкові оцінки лінійних поперечників класів $B_{p,\theta}^\Omega$ у випадку $1 < p \leq 2$, $p' < q < \infty$, $\theta \in (q, \infty$

Нехай \mathbf{R}^d , $d \geq 1$, d -вимірний евклідов простір з елементами $x = (x_1, \dots, x_d)$, $y = (y_1, \dots, y_d)$, $(x, y) = x_1 y_1 + \dots + x_d y_d$, і $L_p(\pi_d)$ — простір p -періодичних за кожною змінною і сумовних у степені p , $1 \leq p < \infty$ (відповідно суттєво обмежених при