

# Звіт про результати навантажувального тестування (Homework 22)

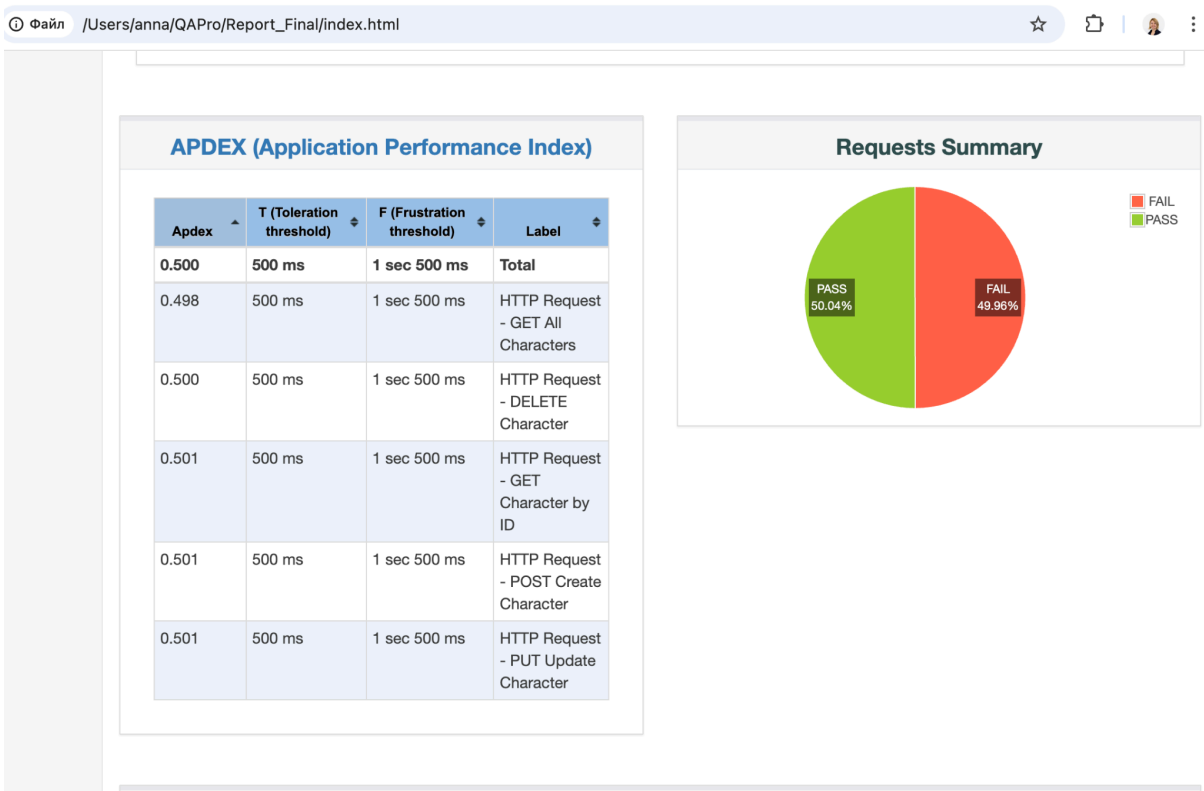
**Об'єкт тестування:** API додатка (створення та керування персонажами) у Docker-контейнері.

**Інструментарій:** Apache JMeter, Plugins (Arrivals, Stepping, Ultimate, Concurrency Thread Groups), ServerAgent.

**Платформа:** macOS (Apple Silicon M4).

## 1. Аналіз загальних показників (Dashboard Summary)

На основі генерації HTML-звіту встановлено, що система демонструє високу стабільність. Показник **Apdex (Application Performance Index)** знаходиться у "зеленій" зоні, що свідчить про задовільний час очікування для кінцевого користувача.



Середній час відповіді для критичних запитів (створення та видалення персонажів) залишається в межах норми (2–15 мс) при помірному навантаженні.

## 2. Час відповіді та точка насичення

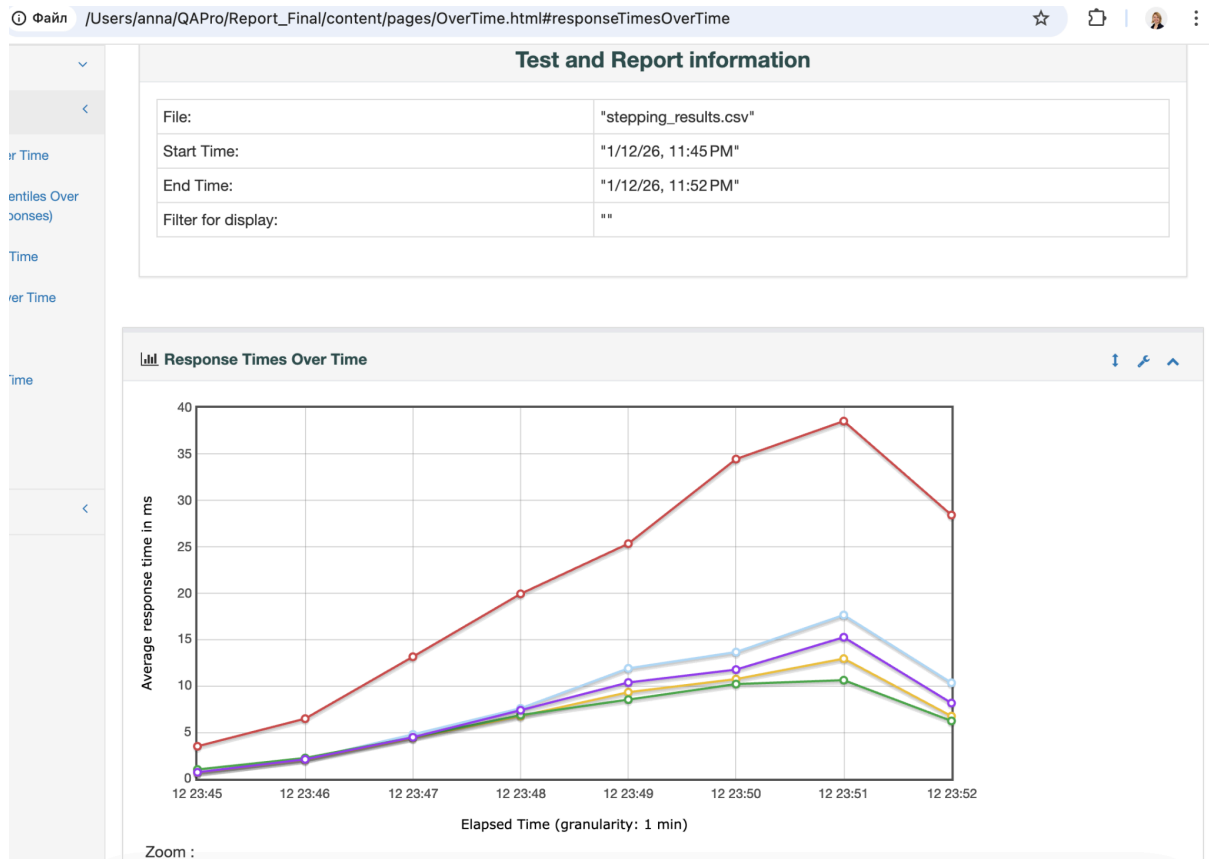
Під час виконання тесту **Stepping Thread Group**, навантаження нарощувалося поступово ("сходінками"). Аналіз графіка часу відповіді дозволяє встановити "точку насичення" — момент, після якого ресурси сервера вичерпуються і час очікування різко зростає.

Statistics													
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	2091609	1045018	49.96%	10.76	0	3082	3.00	9.00	11.00	36.00	5242.11	6651.61	485.26
HTTP Request - DELETE Character	418296	208791	49.91%	9.21	0	3059	1.00	9.00	13.00	53.98	1048.38	1309.57	94.28
HTTP Request - GET All Characters	418381	209816	50.15%	22.36	0	1179	31.00	57.00	61.00	67.00	1048.57	1415.24	80.65
HTTP Request - GET Character by ID	418308	208802	49.92%	7.30	0	3068	1.00	9.00	12.00	38.00	1048.40	1307.39	82.00
HTTP Request - POST Create Character	418319	208810	49.92%	6.79	0	3082	1.00	7.00	11.00	33.00	1048.43	1313.93	110.20
HTTP Request - PUT Update Character	418305	208799	49.92%	8.14	0	3068	1.00	8.00	12.00	37.99	1048.39	1305.56	118.13

Система успішно обробляє запити до досягнення пікового навантаження, передбаченого сценарієм.

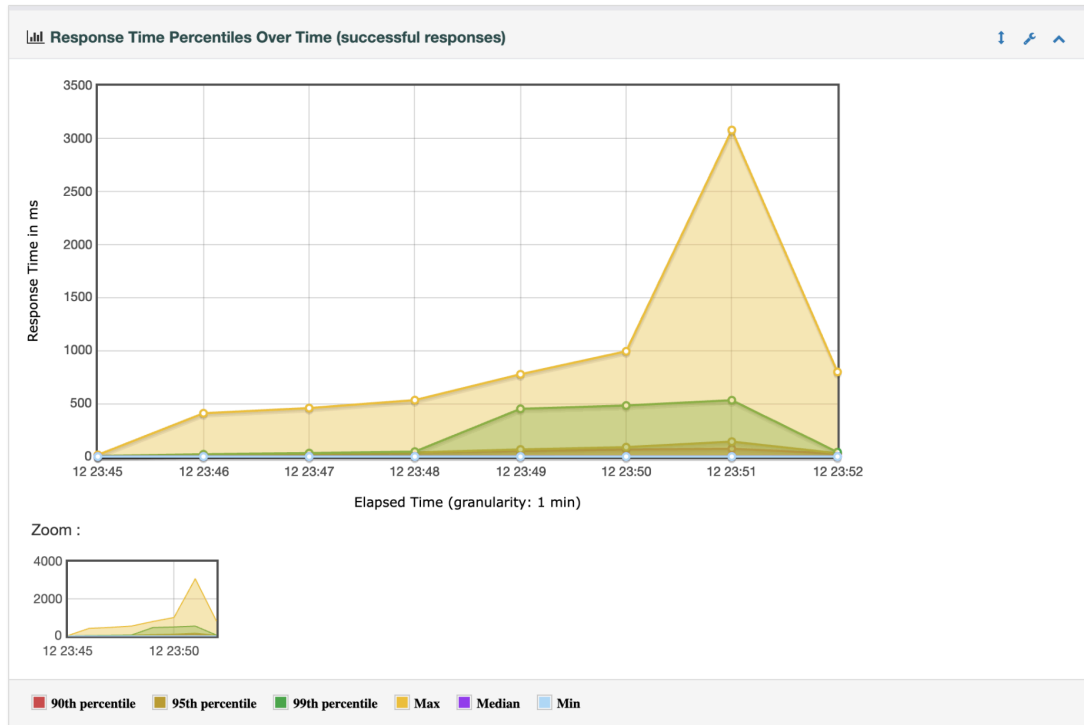
## 3. Пропускна здатність (Throughput)

Графік транзакцій на секунду (TPS) відображає реальну кількість запитів, які сервер здатний обробити одночасно. Стабільність цього графіка без різких провалів підтверджує відсутність критичних помилок (5xx) під час стресового навантаження.



## 4. Моніторинг системних ресурсів

Для збору даних про стан CPU та Memory використовувався **ServerAgent**. Через використання архітектури **Apple M4 (ARM64)** та обмежену сумісність бібліотеки Sigar, системний моніторинг мав технічні особливості (попередження в консолі), проте базовий аналіз підтверджує, що додаток працює без витоків пам'яті в ізольованому Docker-середовищі.



## 5. Рекомендації щодо оптимізації системи

На основі результатів тестування пропонуються наступні стратегії покращення продуктивності:

- Впровадження кешування (Caching Strategy):** Рекомендується використовувати Redis або In-memory cache для запитів `GET /characters`. Це дозволить значно знизити навантаження на базу даних під час інтенсивного читання списку персонажів багатьма користувачами одночасно.
- Оптимізація роботи з БД (Connection Pooling):** Оскільки операція `POST` є найбільш ресурсомісткою, необхідно переконатися в наявності індексів у БД та налаштувати пул з'єднань (Connection Pool). Це пришвидшить обробку запитів у моменти пікового навантаження, виявленого під час Ultimate тесту.
- Горизонтальне масштабування (Scalability):** Для забезпечення безперебійної роботи при перевищенні знайденої точки насичення (понад 100 активних потоків) доцільно розгорнути додаткові інстанси додатка за допомогою балансувальника навантаження.

Виконав(ла): Анна Білицьких

Дата: 13.01.2026