Лабораторна робота №8

Тема: Оптимізація трафіку та балансування навантаження з Nginx

Мета:

- Ознайомитися з принципами балансування навантаження в багатокористувацьких системах.
- Навчитися налаштовувати Nginx як засіб для розподілу трафіку між декількома сервісами.
- Провести базову оптимізацію конфігурації для покращення продуктивності та стійкості системи.

Хід роботи

1. **Ознайомлення з принципами балансування навантаження**
   - Дослідити типи балансування (round robin, least connections, IP hash).
   - Вивчити, як Nginx використовується як балансувальник навантаження (reverse proxy).
2. **Розгортання тестового середовища**
   - Підготувати кілька простих веб-сервісів (можна у вигляді Docker-контейнерів).
   - Переконатися, що сервіси працюють і доступні по різних портах.
3. **Налаштування Nginx**
   - Налаштувати nginx.conf для балансування навантаження між сервісами.
   - Реалізувати щонайменше два режими балансування (наприклад, round robin та least connections).
   - Додатково: увімкнути кешування або стиснення трафіку для підвищення продуктивності.
4. **Тестування ефективності**
   - За допомогою інструментів типу curl, ab, wrk, hey протестувати розподіл запитів.

- Проаналізувати, як балансування впливає на навантаження серверів і час відповіді.

5. **Оформлення звіту**
   - Документувати структуру тестового середовища.
   - Надати фрагменти конфігурації Nginx та скрипти запуску (якщо використовувався Docker).
   - Зробити порівняння між різними стратегіями балансування.

Основні команди та приклади (без повного вирішення)

Приклад базової конфігурації Nginx для round robin:

```
http {
    upstream backend {
        server 127.0.0.1:8081;
        server 127.0.0.1:8082;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://backend;
        }
    }
}
```

Команда для перевірки навантаження:

```
ab -n 100 -c 10 http://localhost/
```

Висновок

У ході виконання роботи студент ознайомиться з можливостями Nginx як інструменту балансування навантаження, налаштує різні режими розподілу трафіку, проведе базову оптимізацію запитів і зробить висновки щодо ефективності конфігурації в умовах багатокористувацьких систем.