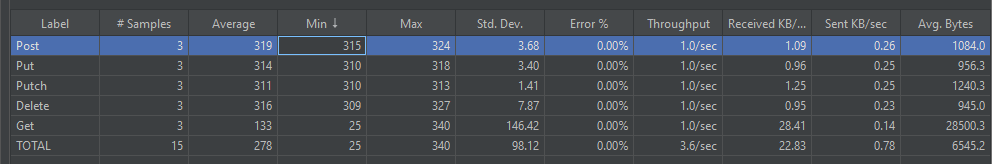
**Домашнє завдання №24**

**Основи навантажувального тестування за допомогою Apache Jmeter**

**Перший рівень**

Тестування системи одним користувачем, що надсилав всіх 4 запити кожні 10 секунд протягом 3 ітерацій пройшло успішно. Система впоралась з таким навантаженням. До кожного запиту були створені асерти на розмір відповіді в байтах, на отриманий результат з кодом 200 і на те, що отриманий результат буде мати XML дані. Всі асерти також були успішними.

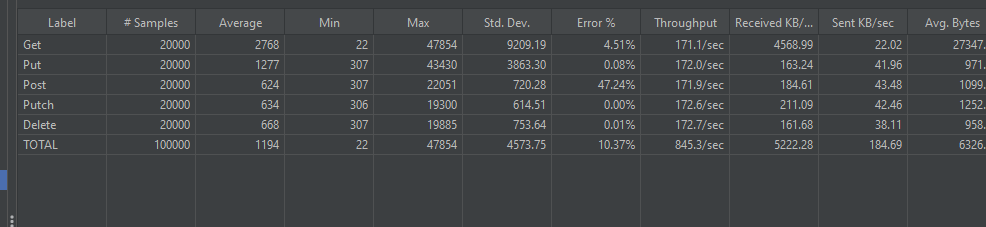


В таблицы видно мінімальний, максимальний та середній час за 3 ітерації, що потратила система на виконання кожного з видів запиту. Також можемо побачити відсоток помилок для кожного запиту. У нас він дорівнює 0 для кожного, адже всі запити були виконані успішно. Як видно в таблиці найбільше часу було потрачено на запит POST, а найзатратнішим по кількості байт був запит GET.

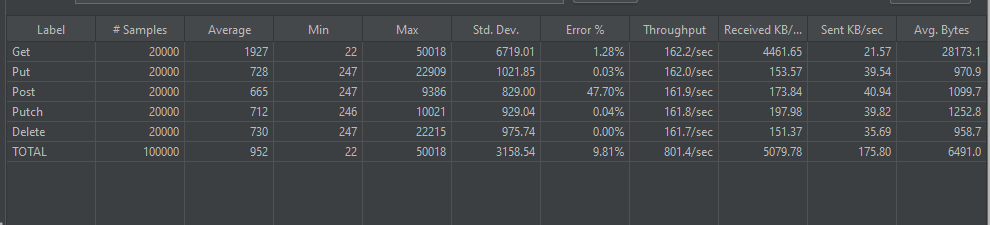
**Другій рівень**

При свторенні стресс тесту найчутливішим виявився запит POST. Він не витримав навантаження вже починаючи з 100 користувачів, які кожні 10 секунд робили запит потягом 10 ітерацій. Найстресостійкішим виявився запит Putch. Він витримав навантаження 2000 користувачів, які кожні 5 секунд надсилали запити до системі на протязі 10 ітерацій.

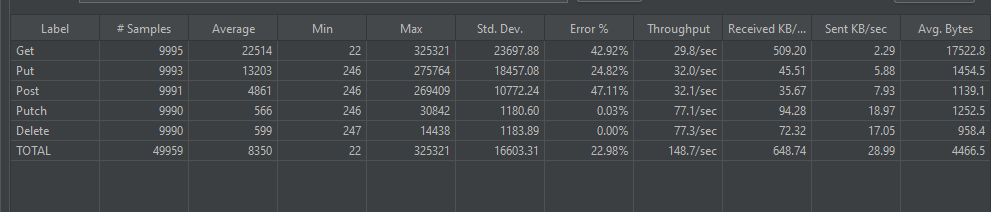
Результати стресс тесту. 2000 користувачів, які кожні 5 секунд надсилали запити до системі на протязі 10 ітерацій.



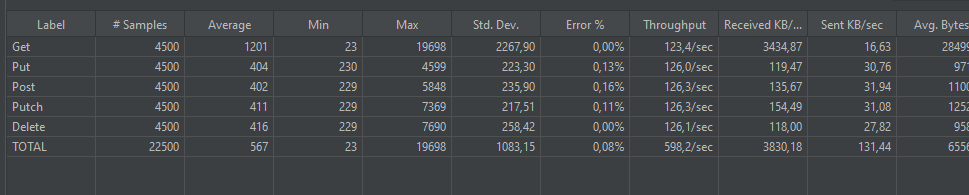
Результати стресс тесту. 2000 користувачів, які кожні 2 секунд надсилали запити до системі на протязі 10 ітерацій.



Результати стресс тесту. 5000 користувачів, які кожні 2 секунд надсилали запити до системі на протязі 2 ітерацій.



Результати стресс тесту. 1500 користувачів, які кожну секунду надсилали запити до системі на протязі 3 ітерацій.

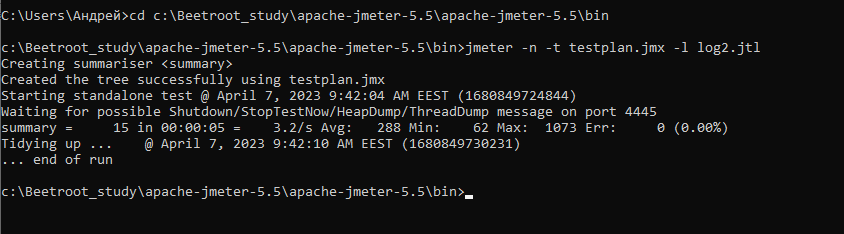


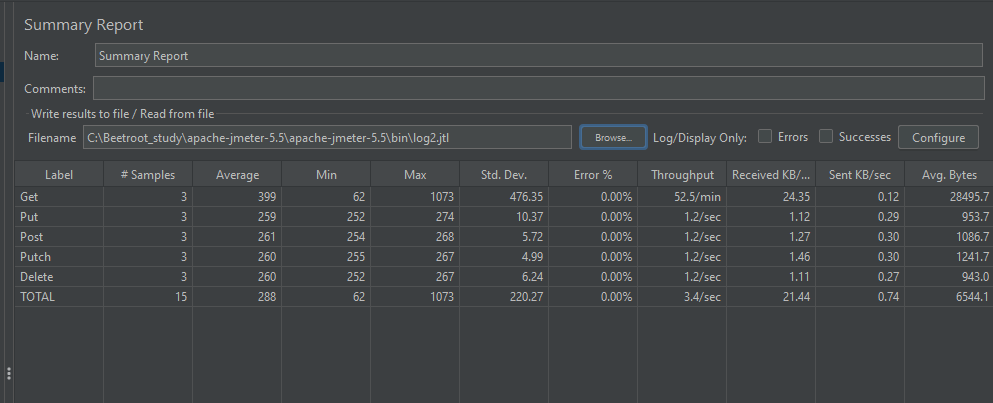
**Третій рівень**

**:лиственное_дерево:**Mighty Beet — різнобічно опануй тематику уроку.

|  |
| --- |
| 1. Виконай завдання двох попередніх рівнів.  2. Створи команду для запуску вказаного тест-плану через Bash CLI. Запусти її та завантаж результати в Summary Report. |

2. Створи документ зі скриншотом виконання команди в командному рядку та скриншотом Summary Report.





3. Документ додай у відповідний репозиторій в GitHub.