В качестве входных данных возьмем следующие величины которые являются общими для POST и GET

(См. рисунок 1-2)

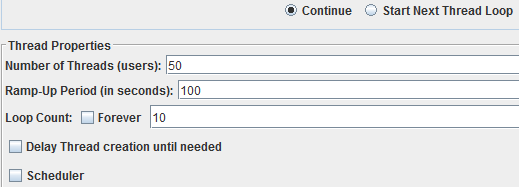


Рисунок 1

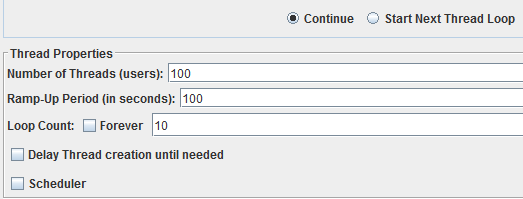


Рисунок 2

Где:

* Number of Threats (users) – количество пользователей (потоков)
* Ramp-Up Period (in seconds) – время которое выделено для пользователей (потоков)
* Loop Count - сколько раз выполнит сценарий каждый из потоков.

Т.е. у нас есть 25 потоков которые стартую по очереди через каждые 2 сек (50/25) и повторяются 10 раз. Запустив с такими параметрами вы получите исполнение своего сценария 250 раз. В конце работы первый поток отработает первый т.к. он первый начал работу

Т.к. количество пользователей будет уменьшаться следовательно нагрузка (как и время отклика) также уменьшится см (рисунок 3-4). На рисунках 5-6 показано графическое отображение запросов в секунду и времени отклика в зависимости от количества активных пользователей.

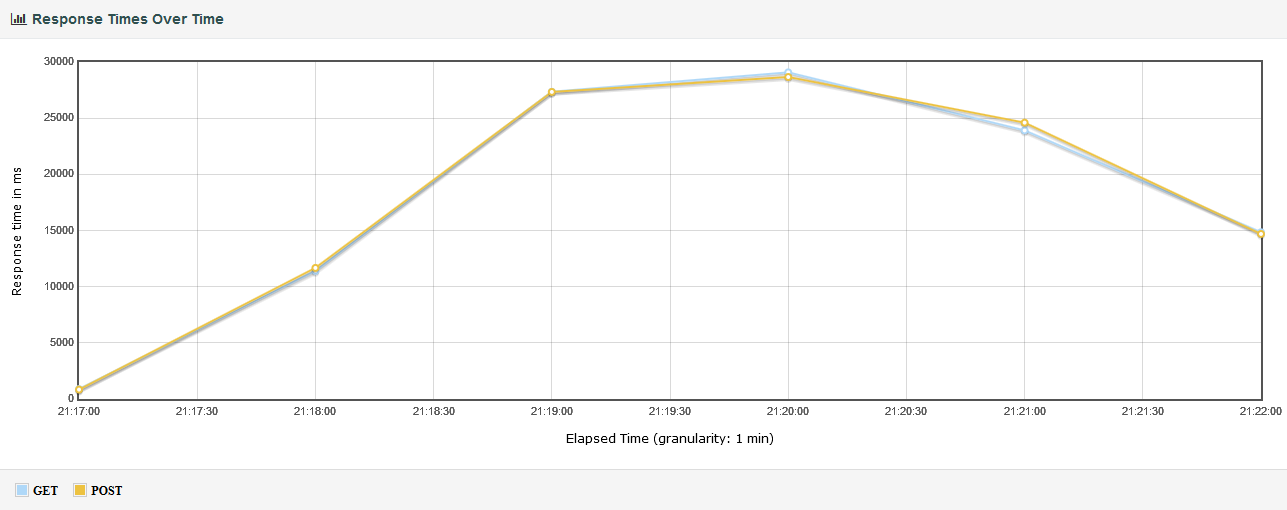


Рисунок 3-Графическое отображение времени отклика

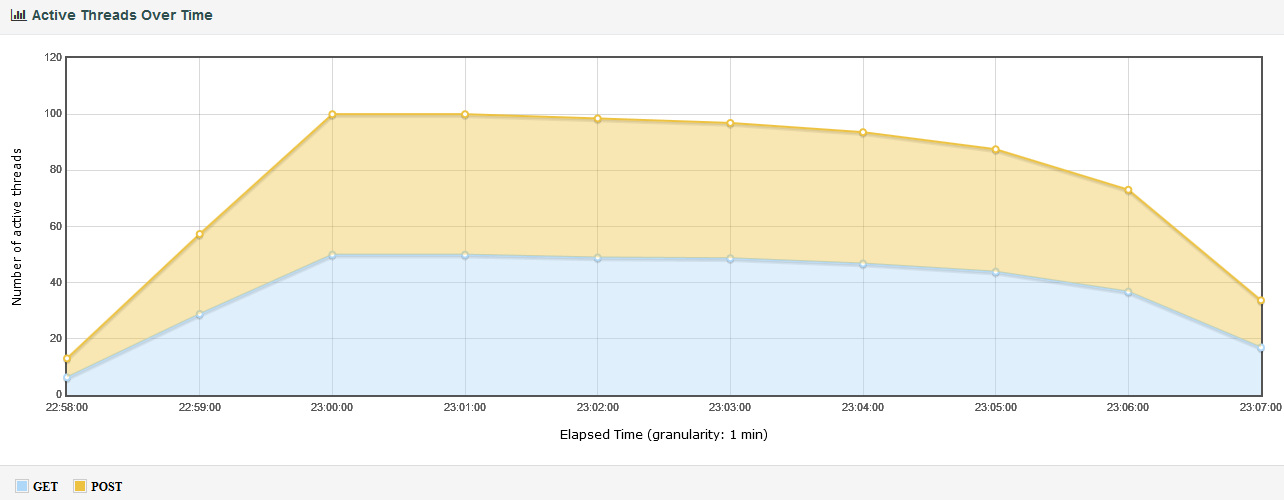


Рисунок 4 - Графическое отображение активных пользователей

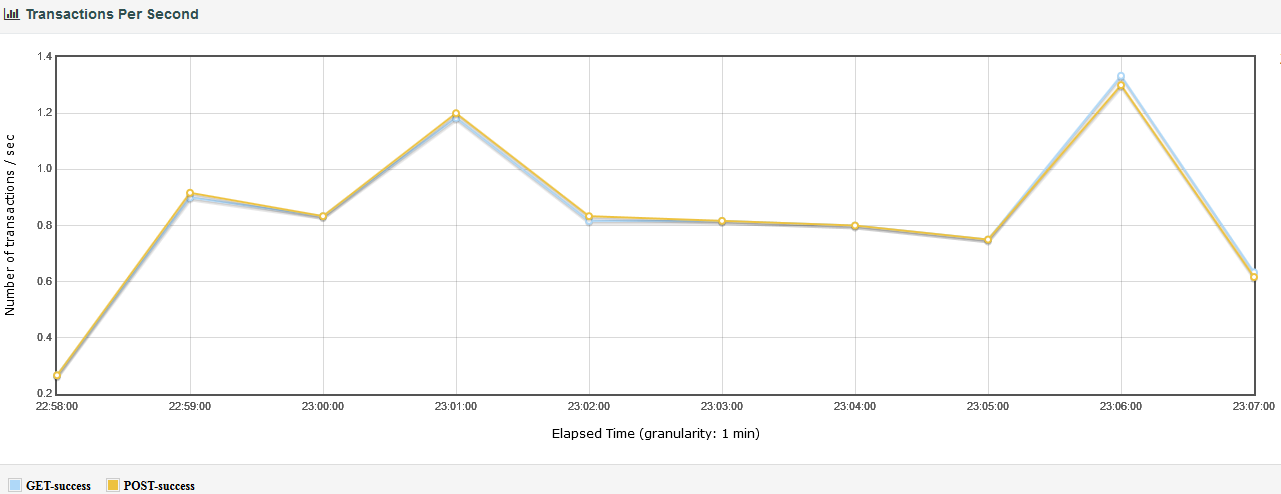


Рисунок 5 - Графическое отображение запросов в секунду

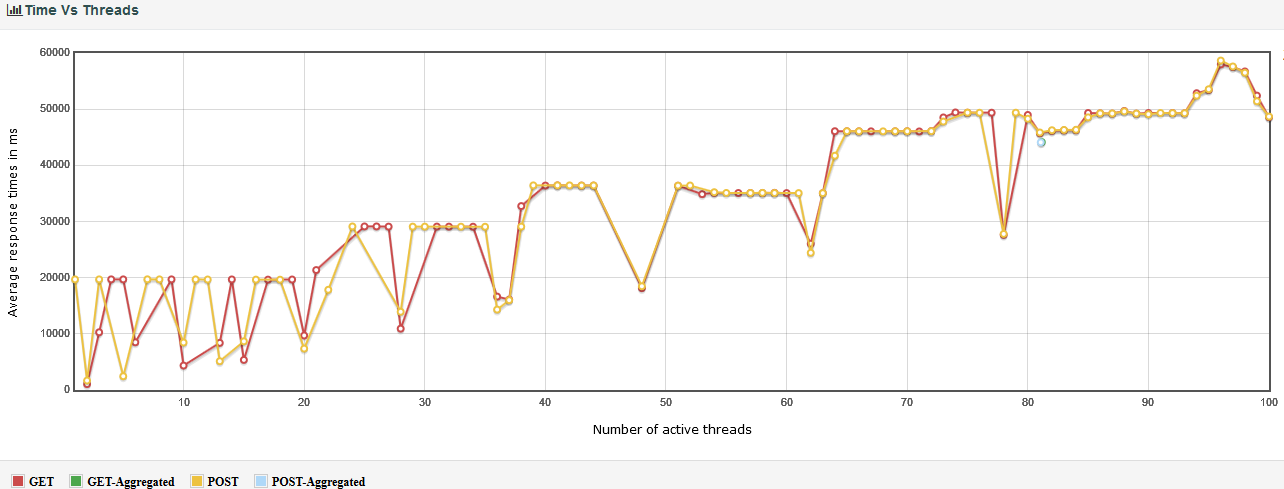


Рисунок 6 - Графическое отображение времени отклика в зависимости от количества активных пользователей

**Number of Threads (users):** 50

**Ramp-Up Period (in seconds):**100

**Loop Count:** 10

При таких значениях уже есть **нехватка производительности** (см. рис 4).

Среднее время отклика **Average** растет, а скорость обработки **Throughput** не изменяется (на сервере операции становятся в очередь). Также на графике отображено **максимальное значение производительности 55 запросов в минуту (55/minute)** Как результат в сутки сайт сможет обработать не больше 79 200 запросов.

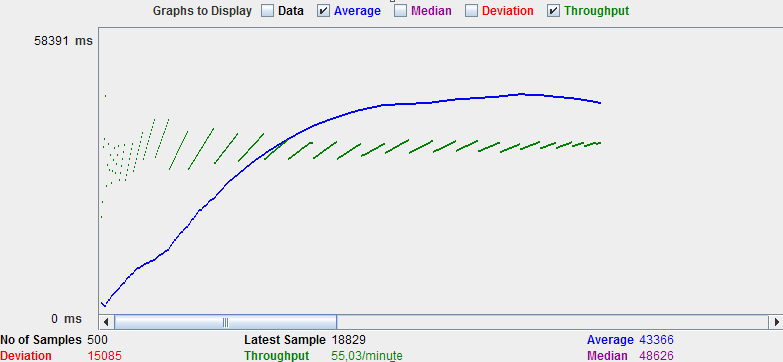


Рисунок 4

Если увеличить нагрузку еще больше

**Number of Threads (users):** 100

**Ramp-Up Period (in seconds):** 200

**Loop Count:** 10

Получим следующее рис.5

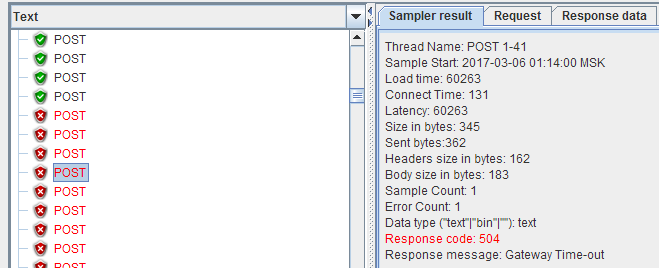


Рисунок 5

**504 ошибка** также указывает на нехватку производительности (ресурсов). Другими словами, запросы выстраиваются в очередь и когда лимит времени на получение ответа заканчивается, они сбрасываются ни с чем.

**Решение:**

1. Можно добавить ресурсов серверу, в частности оперативной памяти, и увеличиваем количество процессов httpd (Apache)

2.Если ресурс расположен на виртуальном хостинге, то можно попросить их администраторов решить такую ситуацию либо вообще сменить хостинг-провайдер