

## Homework\_Lesson15\_WebServers\_2

Цель: выполнить тестирование производительности вашего веб-сервера, чтобы определить, как он работает при большом объеме запросов и нагрузке.

Задания:

1. Установите и настройте инструменты тестирования производительности, такие как Apache JMeter или Gatling. (лучше JMeter)
2. Определите основные метрики производительности, которые нужно измерить, например, время отклика, задержку, пропускную способность и загрузку процессора.
3. Создайте тестовые сценарии: просмотр страницы, с наличием определённого слова. Протестировать на 100,1'000,10'000 подключений.
4. Задайте параметры для тестовых сценариев, например, количество пользователей, количество запросов в единицу времени, длительность теста и т.д.
5. Запустите тест и отслеживайте производительность сервера в режиме реального времени, собирая данные о времени отклика, задержке, пропускной способности и загрузке процессора при различных нагрузках.
- \*\*\* 6. Проанализируйте результаты тестирования, чтобы определить, как сервер работает при различных нагрузках. Определите максимальную нагрузку, которую сервер может выдержать, и выявите узкие места в инфраструктуре.

Выполнение задания:

1. Настройка окружения для тестирования веб-сервера и установка JMeter.

- 1.1 Поднятие новой VM на Ubuntu + настройка Nginx с сайтом.

Создана VM VM\_Ubuntu с двумя адаптерами сети. NAT + Bridged adapter (IP: 192.168.100.14). На ней установлен ОС Ubuntu версии 22.04. После завершения установки ОС устанавливаем необходимые пакеты веб-сервиса Nginx.

<code>\$ apt-get update</code>	<i>#обновляем репозиторий</i>
<code>\$ apt-get install nginx</code>	<i>#установка Nginx</i>
<code>\$ systemctl start nginx</code>	<i>#стартуем Nginx</i>
<code>\$ systemctl status nginx</code>	<i>#проверяем работоспособность Nginx</i>
<code>\$ cd /etc/nginx/sites-available/</code>	<i>#переходим в каталог конфиг. сайта</i>
<code>\$ touch siteforjmeter.conf</code>	<i>#создаем необходимый нам *.conf</i>
<code>\$ nano siteforjmeter.conf</code>	<i>#вносим свою конфигурацию</i>

```
server {  
    listen 80;  
    listen [::]:80;  
    server_name siteforjmeter.by www.siteforjmeter.by;  
  
    root /var/www/html;  
    index siteforjmeter.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Создаем символическую ссылку в директорию ~/sites-enabled.

```
$ sudo ln -s /etc/nginx/sites-available/siteforjmeter.conf /etc/nginx/sites-enabled/  
$ sudo systemctl restart nginx      #обязательный рестарт (либо reload)
```

Создаем html-страницу “siteforjmeter.html” и прокладываем в директорию /var/www/html:

```
<!DOCTYPE html>  
  <html lang="ru">  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>NGINX</title>  
</head>  
<body>  
  <h1>ФИО: Макаров Александр</h1>  
  <h2>Тема урока: Вебсерверы. Часть 2</h2>  
  <h3>JMeter</h3>  
</body>  
</html>
```

Добавляем строку для локального DNS:

```
$ nano /etc/hosts  
192.168.100.14      siteforjmeter.by
```

Для проверки работоспособности сайта вводим команду:

```
$ curl -v tms.by
```

```
root@makarov-VirtualBox:/etc/nginx/sites-available# curl -v siteforjmeter.by  
* Trying 192.168.100.14:80...  
* Connected to siteforjmeter.by (192.168.100.14) port 80 (#0)  
> GET / HTTP/1.1  
> Host: siteforjmeter.by  
> User-Agent: curl/7.81.0  
> Accept: */*  
>  
* Mark bundle as not supporting multiuse  
< HTTP/1.1 200 OK  
< Server: nginx/1.18.0 (Ubuntu)  
< Date: Fri, 27 Dec 2024 12:00:53 GMT  
< Content-Type: text/html  
< Content-Length: 350  
< Last-Modified: Fri, 27 Dec 2024 11:14:16 GMT  
< Connection: keep-alive  
< ETag: "676e8c08-15e"  
< Accept-Ranges: bytes  
<  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>NGINX</title>  
</head>  
<body>  
  <h1>ФИО: Макаров Александр</h1>  
  <h2>Тема урока: Вебсерверы. Часть 2</h2>  
  <h3>JMeter</h3>  
</body>  
* Connection #0 to host siteforjmeter.by left intact  
</html>root@makarov-VirtualBox:/etc/nginx/sites-available#
```

Сайт в рабочем состоянии. На всякий случай добавляем в Windows в файл C:\Windows\System32\drivers\etc\hosts строку “192.168.100.14 siteforjmeter.by”.

1.2 Переходим к установке JMeter на UNIX, можно будет развернуть ещё и в Windows.

Для Linux необходимы пакеты jdk:

```
$ sudo apt install openjdk-11-jdk
$ java --version
```

Через браузер в URL “[https://jmeter.apache.org/download\\_jmeter.cgi](https://jmeter.apache.org/download_jmeter.cgi)”, качаем дистрибутив JMeter:

*apache-jmeter-5.6.3.zip*

## Apache JMeter 5.6.3 (Requires Java 8+)

### Binaries

[apache-jmeter-5.6.3.tgz sha512 pgp](#)  
[apache-jmeter-5.6.3.zip sha512 pgp](#)

Либо в терминальном режиме с помощью команды wget:

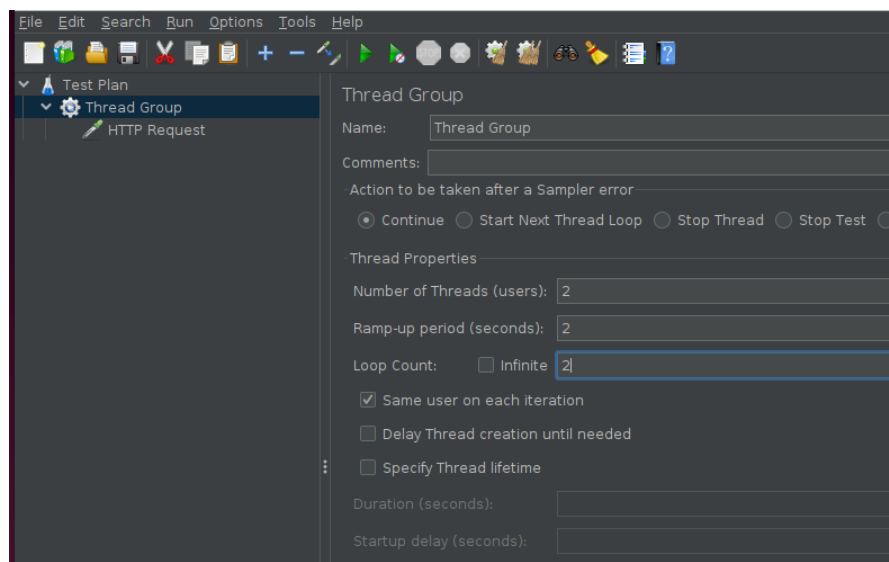
```
$ sudo wget https://apache-mirror.rbc.ru/pub/apache/jmeter/binaries/apache-jmeter-5.6.3.zip
```

В директории /home/[username]/Downloads/ появится архив *apache-jmeter-5.6.3.zip*

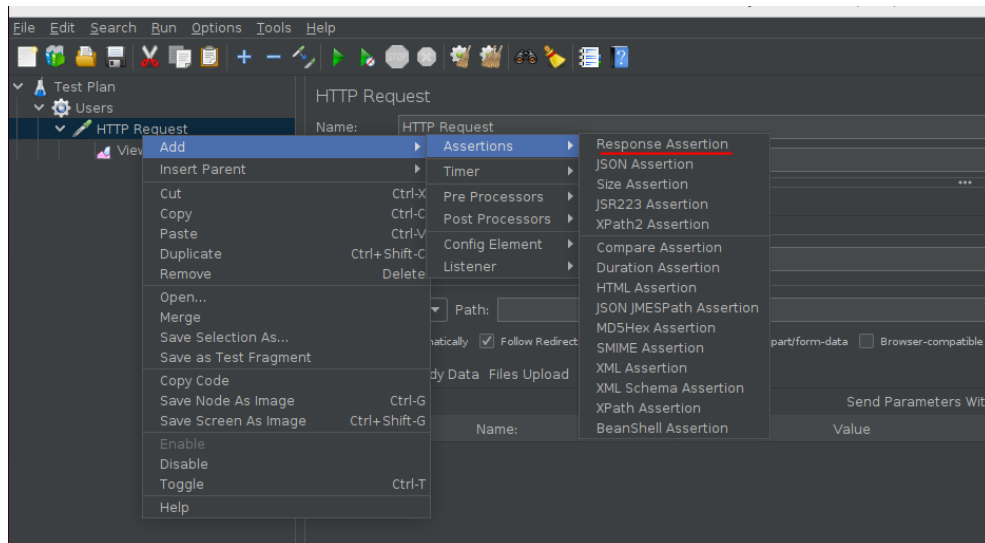
```
$ unzip apache-jmeter-5.6.3.zip      #распаковываем архив
$ cd ./apache-jmeter-5.6.3/bin      #переходим в каталог ./bin
$ ./jmeter                          #запускаем jmeter
```

2. Определение основных метрик производительности, которые нужно измерить, например, время отклика, задержку, пропускную способность и загрузку процессора.

1) Threads Group (Users) - определяет количество пользователей (поток), можно задавать нарастание и количество итераций или циклов для теста.



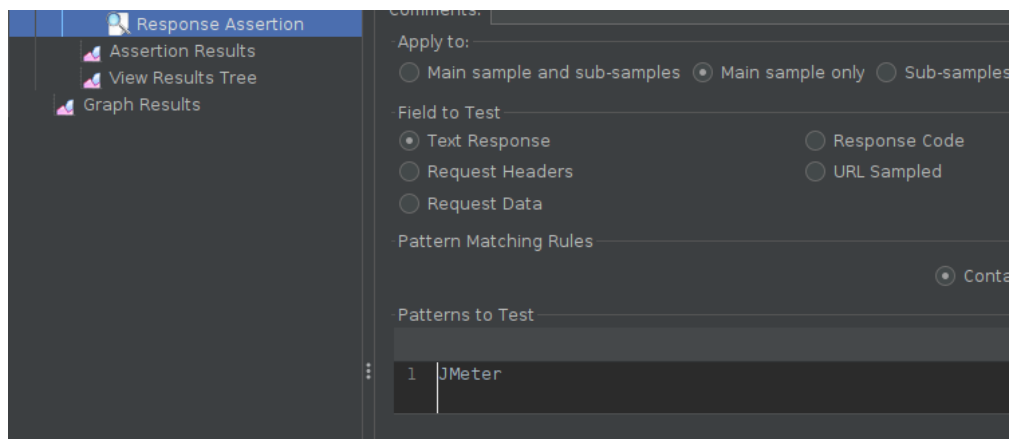
2) Response Assertion - позволяет делать проверку текста и не только. Также можно делать проверку на статус ответа “200” и т.д.



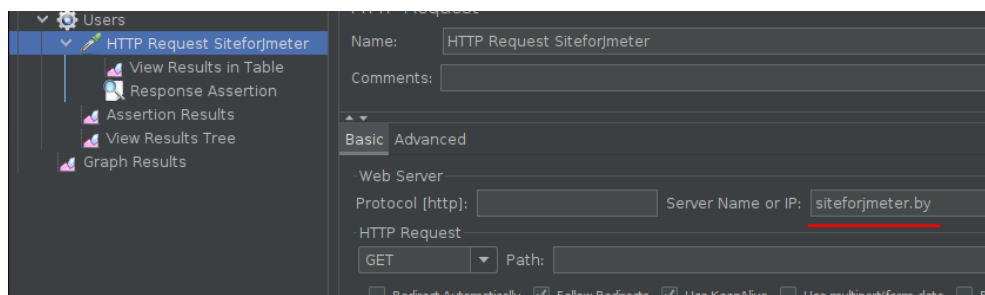
3-4) Проводим тестирование на 1000 подключений с поиском слова JMeter.

“Add -> Assertion -> Response Assertion”

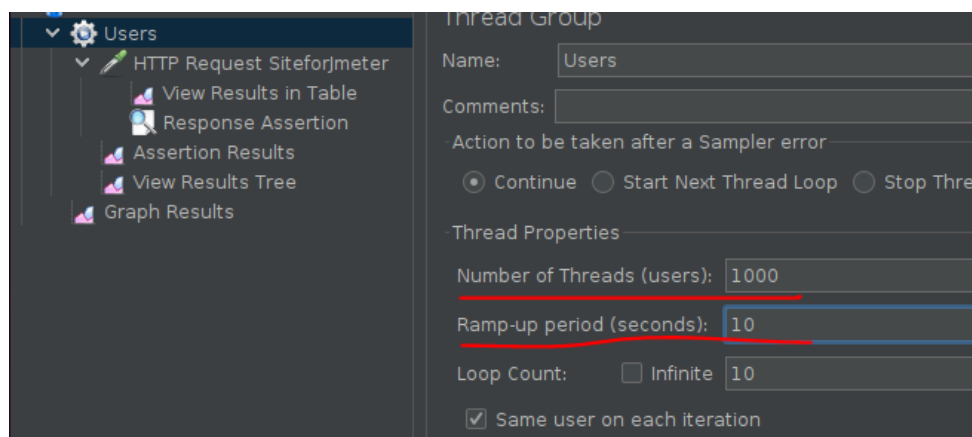
Добавляем паттерн с поиском слова JMeter на нашем созданном сайте siteforjmeter.by.



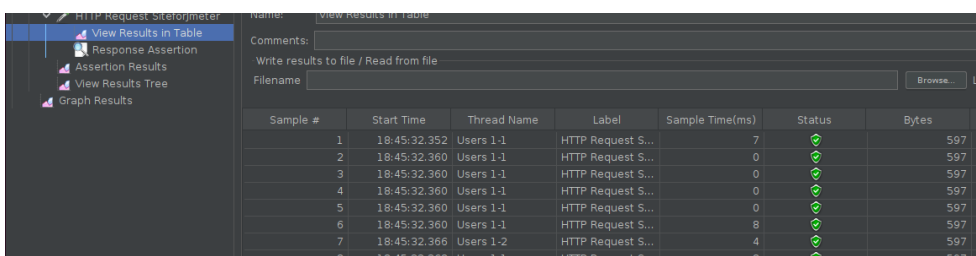
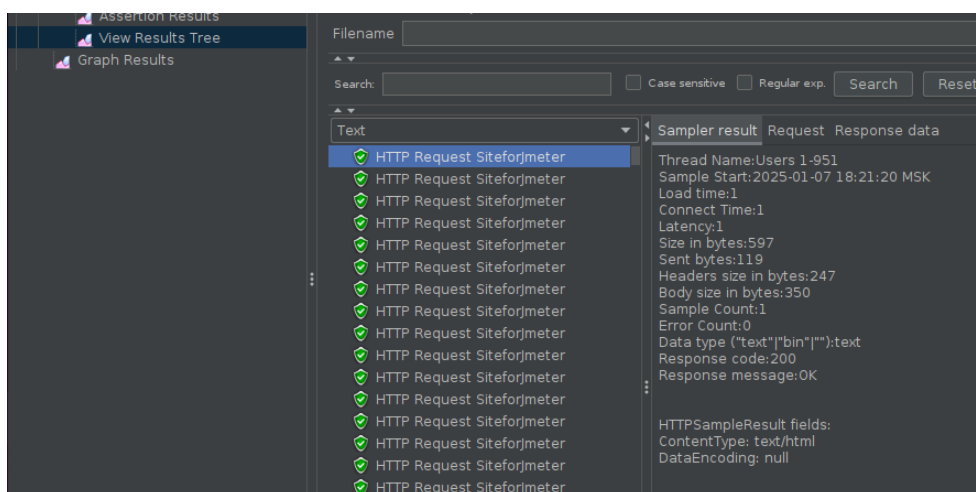
В вкладке “HTTP Request” прописываем имя нашего сайта.



Threads Group прописываем количество users для тестирования сайта. Добавляем 1000 users (Number of Threads) с периодом нарастания пользователей в 10 секунд (Ramp-up period).

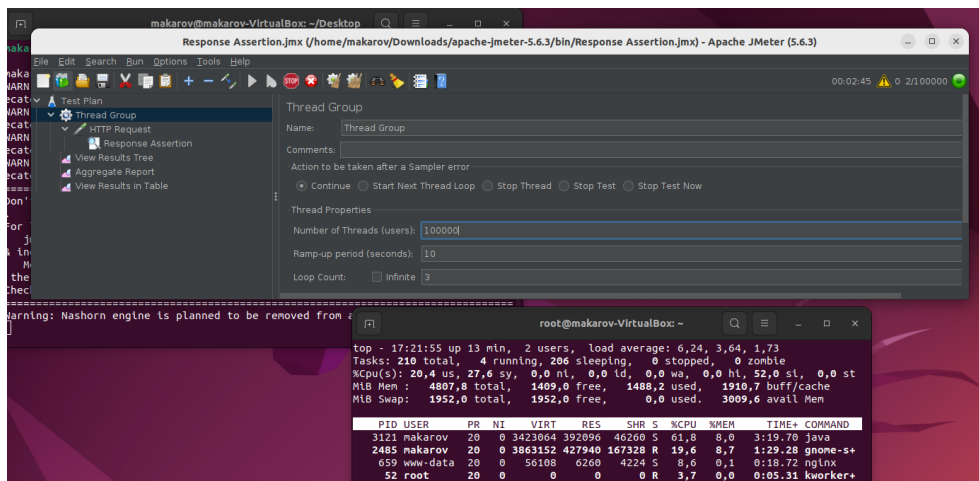


Также, для просмотра результатов добавляем Listener. “Add -> Listener -> View Results Tree” и запускаем тестирование, нажимаем на “Start”. При успешной проверки, примерно будет так:

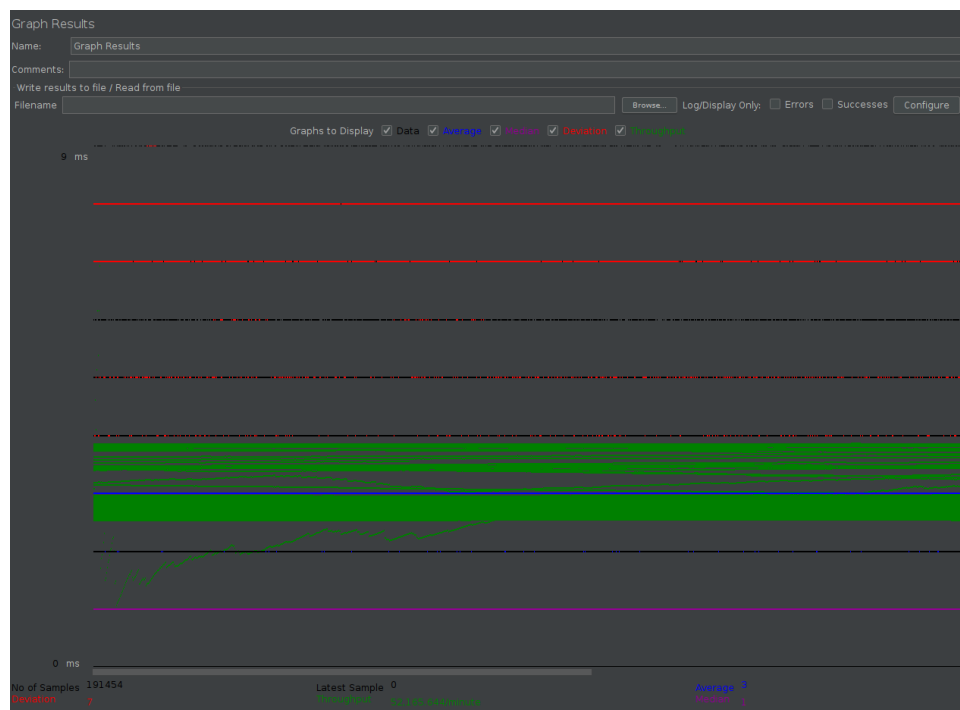


5) Запуск теста для отслеживания производительности сервера в режиме реального времени.

100000 users с периодом нарастания в 10 секунд. С поиском слова JMeter на самом сайте. Сам тест проводился 5 минут 28 секунд. В режиме реального теста, вывели top для наблюдения нагрузки на сервер, загрузка CPU пользователями примерно ~ от 15 - 35%, в то время как система (sy) ~ 20 - 55%.



Графический вариант:



В результате нагрузочного теста, проблем не обнаружено. Как вариант можно попробовать увеличить поток пользователей, и проверить нагрузку устойчивость сервера.