

ИНСТРУМЕНТЫ ДЛЯ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Туровец Н.О.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Алефиренко В.М. – канд. техн. наук, доцент, доцент кафедры ПИКС

Аннотация. В статье рассмотрены основные инструментальные средства для проведения нагрузочного тестирования на информационную систему. Выбрано наиболее подходящее средство тестирования. Установлено, что при выполнении в правильном порядке всех составляющих плана тестирования возможно получить ожидаемые результаты.

Ключевые слова: информационная система, нагрузочное тестирование, выбор, инструментальные средства, план тестирования.

Для того, чтобы начать нагрузочное тестирование информационной системы необходимо выбрать инструмент, с помощью которого будет все это происходить. На сегодняшний день существует множество средств тестирования. К числу таких относятся доступные инструменты или программное обеспечение, и они могут стоить от нуля до десятков тысяч долларов.

В настоящее время на рынке программного обеспечения существует большое количество различных инструментов для проведения тестирования производительности, начиная от программных средств с открытым исходным кодом и заканчивая приложениями для автоматизированного тестирования премиум-класса. Однако, имея такое множество альтернатив, иногда трудно выбрать лучший инструмент. Рассмотрим несколько инструментов нагрузочного тестирования [1].

Micro-Focus Loadrunner (ранее известный как *HP Loadrunner*) – это довольно сложный и универсальный инструмент тестирования производительности программного обеспечения, который обнаруживает проблемы с производительностью прежде всего в корпоративных приложениях, используемых крупными компаниями для зарабатывания денег. Однако из-за своей цены этот инструмент больше подходит для средних и крупных организаций. *LoadRunner* может применяться для тестирования программного обеспечения *ERP*, устаревших системных приложений, а также технологий *Web 2.0*. *Loadrunner* позволяет проводить комплексную оценку производительности своей системы. *LoadRunner* очень полезен при обнаружении пробелов в производительности, если будет проводиться обновление системы. Также он предоставляет пользователям функции для прогнозирования затрат по увеличению производительности приложений [1].

BlazeMeter – компания-производитель одноименного программного обеспечения для тестирования, предоставляющая пользователям тестирование производительности и нагрузочное тестирование как услугу. Служба содержит инновационную и всеобъемлющую платформу непрерывного тестирования. Веб-интерфейс приложения эффективен для создания статических нагрузочных тестов и использования сценариев *JMeter* для выполнения динамических нагрузочных тестов. Он предоставляет различные корпоративные функции для бесплатной платформы. То есть пользователи могут получить доступ ко многим расширенным функциям, таким как мониторинг производительности приложений (*APM*), создание отчетов в режиме реального времени, распределенное тестирование и интеграция с инструментами разработчика для непрерывной интеграции [2].

Выбор будет зависеть от потребностей, имеющихся ресурсов, бюджета тестового проекта и финансирования проекта. Если возникли проблемы с проектом, такие как ограничен-

ный бюджет, и вам нужно быстрое и надежное тестирование, стоит выбрать *Apache JMeter*. Этот инструмент, соответствует следующим критериям тестирования [3]:

1. Нулевая стоимость приобретения – просто нужно загрузить бинарные файлы по *URL*-адресу.

2. Простота в обучении – базовые знания *HTML* достаточны, чтобы овладеть данным продуктом.

3. Универсальность – может использоваться для тестирования не только веб-приложений.

4. Масштабируемость – модульная конструкция позволяет объединять компоненты для поддержки масштабного тестирования. Тестирование также может выполняться более чем на одном устройстве.

5. Расширяемость – *API (Application Programming Interface)* доступен для настройки.

6. Хорошая поддержка – онлайн-руководство пользователя, пользовательские форумы, веб-заметки и т. д.

Apache JMeter – это десктопное приложение, предназначенное для нагрузочного тестирования, тестирования и измерения производительности и функционального поведения клиент-серверных приложений, таких как веб-приложения или *FTP*-серверы. На сегодняшний день это один из наиболее широко используемых свободно распространяемых продуктов с открытым исходным кодом. Оно основано исключительно на *Java* и использует графический *API Swing*. *Apache JMeter* работает, как «клиентская сторона» «клиент/серверного» приложения. Оно измеряет время отклика и все другие ресурсы, такие как загрузка процессора, использование памяти и использование ресурсов. В этом отношении *Apache JMeter* можно эффективно использовать для функционального тестирования. Кроме того, в нем есть инструменты, поддерживающие регрессионное тестирование аналогичных типов приложений. Хотя изначально он был разработан для тестирования веб-приложений, он был расширен для поддержки других тестовых функций.

Apache JMeter был впервые разработан Стефано Маццокки из *Apache Software Foundation*. Он написал его прежде всего для проверки производительности *Apache JServ*, который позже был заменен проектом *Apache Tomcat*. С тех пор *Apache JMeter* был разработан и расширен для нагрузочного тестирования *FTP*-серверов, серверов баз данных и т. д. На сегодняшний день он используется в качестве инструмента нагрузочного тестирования веб-приложений [3].

Различные компании, в том числе *AOL*, использовали *Apache JMeter* для нагрузочного тестирования своих веб-сайтов, а *SharpMind of Germany* использовала приложение для функционального и регрессионного тестирования своих приложений и клиентов [3].

Рассмотрим некоторые особенности *Apache JMeter*:

1. Будучи программным обеспечением с открытым исходным кодом, оно свободно доступно.

2. Оно имеет простой и понятный графический интерфейс.

3. *Apache JMeter* может проводить тестирование нагрузки и производительности для различных типов серверов – *Web – HTTP, HTTPS, SOAP*, База данных через *JDBC, LDAP, JMS, Mail – POP3* и т. д.

4. Это независимый от платформы инструмент. В *Linux / Unix* *Apache JMeter* может быть вызван нажатием на скрипт оболочки *JMeter*. В *Windows* его можно запустить, запустив файл *jmeter.bat*.

5. *Apache JMeter* хранит свои планы испытаний в формате *XML*. Это означает, что вы можете создать план тестирования с помощью текстового редактора.

6. Его полная многопоточная структура позволяет одновременную выборку многими потоками и одновременную выборку различных функций отдельными группами потоков.

Apache JMeter создаёт имитацию группы пользователей, которые выполняют определённые действия в системе, эти действия отправляются на сервер, и возвращает результаты такие как: производительность, функциональность сервера, графики и т. д.

На рисунке 1 показан принцип работы Apache JMeter.

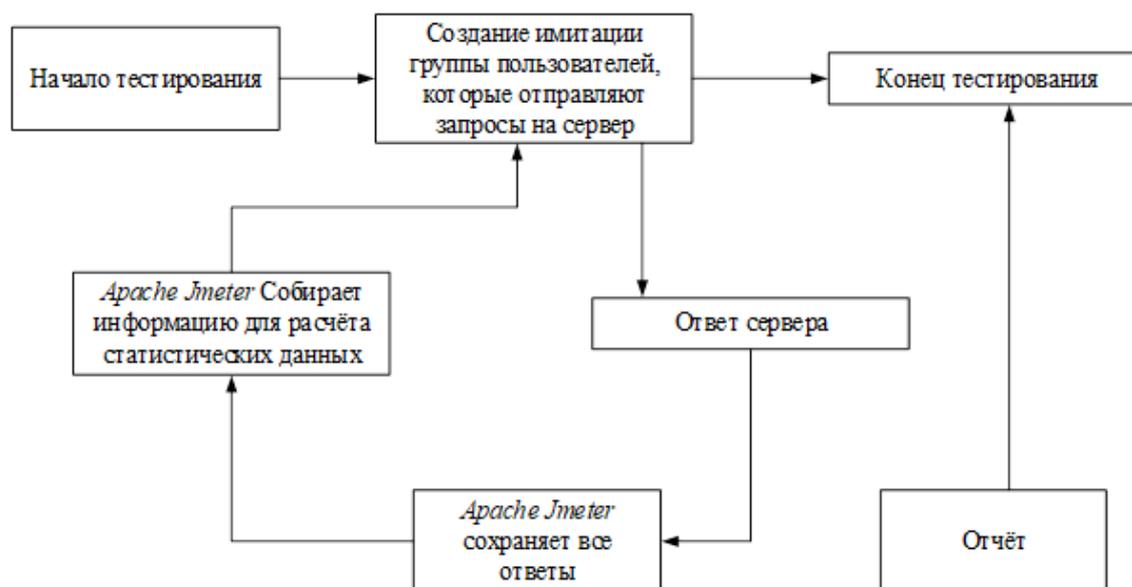


Рисунок 1 – Принцип работы Apache JMeter

Apache JMeter можно использовать для проверки производительности на статических и динамических ресурсах, таких как статические файлы, сервлеты, *FTP*-серверы, объекты *Java*, базы данных, и др. Чтобы протестировать и измерить надежность *HTTP*- или *FTP*-сервера тестировщики должны обеспечить моделирование нескольких и разных типов нагрузок на эти системные объекты. *Apache JMeter* может помочь им сделать именно это и в большем масштабе. Кроме того, его графические инструменты позволяют лучше анализировать производительность при больших нагрузках.

Рассмотрим интерфейс и некоторые возможности *Apache JMeter*.

После установки и запуска данного инструмента для проведения нагрузочного тестирования появляется графический интерфейс *JMeter*, представленный на рисунке 2.

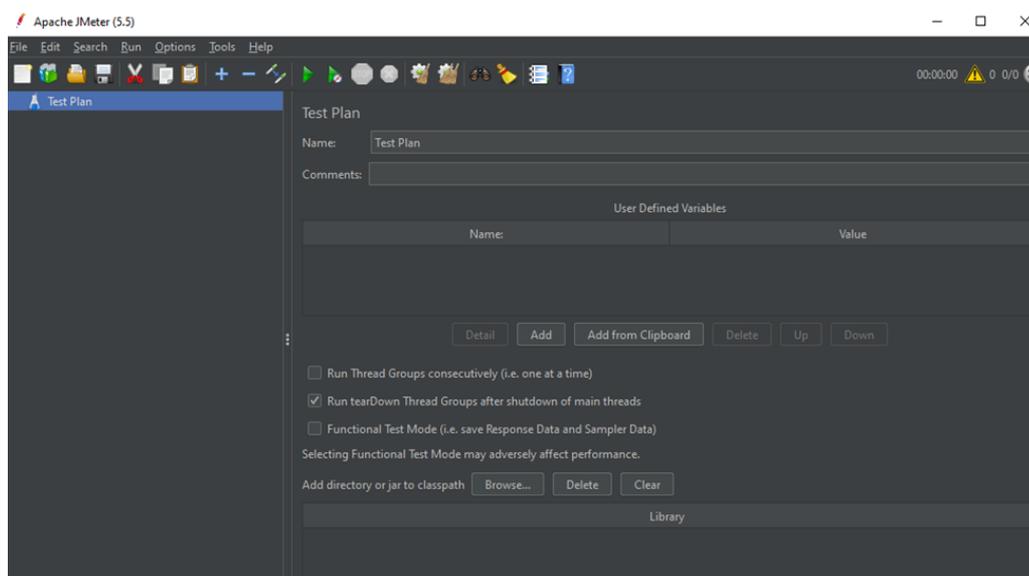


Рисунок 2 – Графический интерфейс Apache JMeter

Эта страница является главной для сборки и начала проекта.

Рассмотрим понятие плана тестирования (тест план). Тест план (*Test Plan*) – это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, страте-

гии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения [4].

Типичный план тестирования будет состоять из одной или нескольких групп потоков, логических контроллеров, слушателей, таймеров, утверждений, а также элементов конфигурации:

1. Группа потоков (*Thread Group*) – каждый поток имитирует одного пользователя. Все элементы тест плана должны находиться в группе потоков.

2. Слушатели (*Listeners*) – предоставление доступа к информации, собранной *Apache JMeter* о тестовом случае во время работы *Apache JMeter*.

3. Контроллеры (*Controllers*) – сэмплеры говорят *Apache JMeter* отправлять запросы на сервер, в то время как логические контроллеры позволяют настраивать его логику.

4. Таймеры (*Timers*) – позволяют *Apache JMeter* делать задержки между каждым запросом, который делает поток.

5 Утверждения (*Assertions*) – позволяет «проверить», что система возвращает результаты, которые ожидаются.

6. Элементы конфигурации (*Configuration Elements*) – тесно взаимодействуют с сэмплером, они могут добавить или изменить запросы.

Таким образом, *Apache JMeter* предоставляет видимые инструменты (или элементы), которые позволяют собирать информацию о системе, которую подвергается тестированию, и сохранять и просматривать результаты в различных форматах. Визуальная информация, которую предоставляет *Apache JMeter*, упрощает процесс успеха/неудачи нашего тестирования. Поскольку *Apache JMeter* позволяет сохранять части (или ветви) плана тестирования индивидуально, есть возможность использовать их повторно, объединяя с другими планами тестирования.

Список литературы

1. Туровец Н.О. Методы тестирования интегрированных информационных систем / Н.О. Туровец, В.М. Алефиренко // Science Time. – 2022. – №3 (99). С.46–54.
2. PerfomanceLab [Электронный ресурс]. – Режим доступа: <https://www.performance-lab.ru/blog/luchshie-instrumenty-dlya-nagruzochnogo-testirovaniya>
3. From Technologies to Solutions, *Apache JMeter* / Packt Publishing. – Birmingham Mumbai, 2008.
4. ПроТестинг [Электронный ресурс]. – Режим доступа: <http://www.protesting.ru/testing/plan.html>

UDC 004.415.533

TOOLS FOR LOAD TESTING OF INFORMATION SYSTEMS

Turovets N.O.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Alefirenko V.M. – PhD, associate professor, associate professor of the Department of ICSD

Annotation. The article considers the main tools for carrying out load testing on an information system. The most appropriate testing tool has been selected. It has been established that when all components of the test plan are executed in the correct order, it is possible to obtain the expected results.

Keywords: information system, load testing, choice, tools, test plan.