

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет вычислительной техники

Кафедра ИВС

ОТЧЕТ ОБ ОСНОВНЫХ РЕЗУЛЬТАТАХ ПОДГОТОВЛЕННОЙ
НАУЧНО-КВАЛИФИКАЦИОННОЙ РАБОТЫ

на тему:

Адаптивный метод имитационного моделирования сетевых нагрузок для
аппаратно-программных комплексов

Направление подготовки

09.06.01 Информатика и вычислительная техника

Направленность (профиль) 05.13.18 Математическое моделирование,
числительные методы и комплексы программ

Форма обучения Очная

Выполнил аспирант



Стельмах Д.С.

(подпись, дата)

Научный руководитель

Косников Ю.Н.

профессор, д.т.н.

(подпись, дата)

Отчет размещен
в ЭИОС 27.09.2020
Зав. кафедрой ИВС  Г.В. Бобрышева

Пенза, 2020

Содержание

Введение.....	3
Анализ имеющихся решений.....	6
Предлагаемые решения.....	14
Результаты работы.....	15
Список опубликованных работ по теме исследования.....	16
Список использованных источников.....	17
Список используемых сокращений.....	19

Введение

Использование компьютерных сетей для решения самого широкого круга вычислительных задач началось в 80-х годах прошлого столетия. Но и сегодня мы наблюдаем активное внедрение сетевых технологий во все новые сферы деятельности. Информационную инфраструктуру современной организации нельзя представить без огромного количества программных комплексов самых разных направлений: бухгалтерский учет, система поддержки принятия решений, различные КРІ системы, базы данных и т.п. Все эти средства нуждаются в надежной и качественной транспортной платформе. С растущими информационными потребностями общества все больше требований предъявляется к качеству работы вычислительных сетей (ВС), которые становятся все более сложноорганизованными. Надежность и производительность ВС часто являются критически важными факторами в деятельности организаций самых различных масштабов. В то же время ВС остаются уязвимыми для аппаратных и программных сбоев.

Спрогнозировать реальную нагрузку на вычислительный комплекс на этапе запуска очень сложно, но еще сложнее проверить систему на эту самую нагрузку. Сколько было примеров не удачных стартов, больших программных продуктов просто из-за того, что система была не готова к такому пользователей и трафика? Но даже если вычислительных мощностей хватало для работы, появлялись «узкие места», например сервер авторизации не справлялся с количеством запросов, мощностей хватало, а именно программная часть не была рассчитана на столько пользователей. Именно для этого и существуют системы нагрузочного тестирования.

Актуальность работы

Спрогнозировать реальную нагрузку на вычислительный комплекс перед этапом запуска очень сложно, но еще сложнее комплексно правильно нагрузить систему. Известно множество примеров не удачных стартов, больших программных продуктов из-за того, что система была не готова к такому количеству пользователей и трафика. Даже если вычислительной мощности хватало для работы, появлялись «узкие места», например сервер авторизации не справлялся с количеством запросов. Вычислительных ресурсов было достаточно, но программная часть сервера не была рассчитана на такое количество одновременных запросов пользователей. Для тестирования каждого ПАК требуется индивидуальный подход, и разработка систем тестирования под каждый комплекс будет слишком трудоемкой. В тоже время существуют готовые общие решения для тестирования, но они либо проверяют нагрузку только сетевое оборудование и его конфигурацию или проверяют устойчивость популярных программных средств. Чтобы провести комплексное тестирование приходится использовать сразу несколько готовых решений и кооперировать их работу. Создать программный комплекс что будет производить нагрузку не только на вычислительные ресурсы, но и будет проверять возможность отказа программных средств цель данной работы. Сложность создания такого программного комплекса в индивидуальных свойствах каждого отдельно взятого ПАК. Комплексных решений на текущий момент нет. Рассмотрим существующие методы и программные решения в данном вопросе.

Цель работы является разработка и программная реализация нагрузки на ПАК учитывая структуру и топологию сети, а также процессы передачи и обработки трафика.

1. метод анализа структуры ВС, в который входит: определение узлов сети и их аппаратные характеристики, используемые протоколы сетевого взаимодействия, маршрутизацию;

2. алгоритм анализа сетевого трафика ВС
3. алгоритм генерации трафика имитирующего свойства трафика реальной сети;
4. оценка эффективности внедренных технических мер и определение степени их влияния на качество ВС.
5. Построение архитектуры и реализация программной реализации ядра средства тестирования.

Объектом исследования является процесс поведения трафика вычислительного комплекса и генерация его генерация.

Предметом исследования является создание комплекса адаптивных процедур программного моделирования сетевой нагрузки и последовательности их применения.

Методы исследования:

В основе теоретических исследований лежат методы программного моделирования, методы анализа сетевого трафика и изучение его свойств. Реализация программного комплекса требует применения методов объектно-ориентированного программирования. Результаты вычислительных экспериментов анализируются с помощью методов математической статистики.

Научная новизна:

1. Предложен алгоритм анализа сетевого трафика, отличающийся от существующих тем что так же анализирует временное поведения трафика и определяет его показатель Херста;
2. Предложен алгоритм генерации сетевого трафика учитывающий показатель Херста, заложена возможность генерировать модель трафика реальной сети.
3. Предложен математический аппарат, определяющий надежность узлов вычислительной сети и позволяющий определить точку отказа узла;

4. Комплексное применение перечисленных решений образует адаптивный метод программного моделирования сетевых нагрузок на аппаратно-программные комплексы.

Практическая полезность состоит в программном комплексе для решения задач моделирования взаимодействия ВС и ПО, основанный на алгоритмах анализа и генерации трафика реальной сети.

Положения, выносимые на защиту:

1. Разработан алгоритм анализа сетевого трафика, определяющий не только состав пакетов, но и анализирующий поведения трафика тестируемой сети. Так же алгоритм определяет показатель Херста для трафика тестируемой сети;
2. Разработан алгоритм генерации сетевого трафика, с возможностью генерировать реалистичную модель поведения трафика. Позволяющий создать адаптивный метод тестирования сетевых нагрузок.
3. Разработан математический аппарат, вычисляющий узкие места сетевой инфраструктуры.

Анализ имеющихся решений

Проведем сравнительный анализ существующих продуктов нагрузочного тестирования ВС: JMeter, LoadRunner Gatling, Taurus, The Grinder, Rational Performance Tester. Изучим возможности этих систем и оценим их следующим показателям:

- 1) генерация отчетов;
- 2) специализированность ПО (насколько программный комплекс ориентирован именно на тестирование ВС);
- 3) потребность в дополнительном программном обеспечении;
- 4) возможность расширяемости за счет дополнений;
- 5) сложность внедрения системы и работы с ней;
- 6) независимость от платформы;

- 7) развитие и поддержка системы;
- 8) стоимость программного средства.

Все эти системы разработаны для тестирования популярных программных решений, например JMeter изначально разрабатывался для тестирования web-приложений. Сейчас он, помимо этого, используется для функционального тестирования и тестирования баз данных. JMeter обеспечивает поддержку нескольких модулей нагрузки, которым управляет один контролер.

Рассмотрим особенности JMeter:

- 1) Помимо обычных отчетов результаты исполнения тестов могут отображаться в различном виде: таблицы, диаграммы, графики. Так же модуль Apache Tomcat позволяют видеть результаты тестов в режиме реального времени;
- 2) JMeter поддерживает протоколы: HTTP, FTP, SOAP, LDAP, JDBC, JMS;
- 3) JMeter не требует установки дополнительного программного обеспечения;
- 4) JMeter интегрирован с бесплатными библиотеками Jenkins, Maven и Gradle.
- 5) Довольно прост в освоении. Для ознакомления с JMeter не требуется много времени;
- 6) JMeter является Java-приложением, он и присутствует всех основных платформах;
- 7) JMeter проект поддерживается в настоящее время;
- 8) Бесплатный продукт.

Необходимо отметить отдельно следующие слабые места продукта:

- 1) Отсутствует возможность тестирования HTTPS-соединений;
- 2) Отсутствует возможность работы с AJAX;
- 3) Отсутствует гибкость настройки отчетов;

4) Ориентированность на тестирование web-приложений.

The Grinder продукт для нагрузочного тестирования, созданный на Java.

Основным языком для написания тестов является Jython, специальная реализация Java в Python. Тестовые сценарии также могут быть написаны на языке Clojure. The Grinder состоит из двух частей:

- 1) The Grinder Console – это интерактивная среда разработки, которая используется для создания и редактирования тестов. В ней так же возможен мониторинг результатов тестирования в режиме реального времени;
- 2) The Grinder Agents– специальные генераторы нагрузки.

Проведем анализ The Grinder по выбранным параметрам:

- 1) Простые графики производительности для всех соединений в одной вкладке;
- 2) Есть возможность загрузки Java API в качестве тестов из сервера HTTP, SOAP и REST сервисов;
- 3) Потребности в дополнительном программном обеспечении отсутствуют;
- 4) Доступен API для создания собственных дополнений;
- 5) Использование требует специальные знания языков Jython или Clojure;
- 6) The Grinder так же написан на Java. Запускается на всех основных платформах;
- 7) Поддерживается в данный момент;
- 8) Бесплатный продукт.

Стоит отметить следующие недостатки:

- 1) Случаются сбои в сценариях при использовании больших файлов для загрузки;
- 2) Grinder не может динамически регулировать уровни нагрузки;
- 3) Отсутствуют инструменты для дополнительного анализа результатов.

Gatling – инструмент для проведения нагрузочного на ВС, созданное на языке Scala. Представляет из себя два модуля: первый для создания тестов, а второй для их запуска. Структурно можно разделить четыре части:

- 1) HTTP protocol configuration помогает определить начальный URL, с которого начнутся тесты. Кроме того, дает возможность определены дополнительные параметры;
- 2) Headers definition предоставляет заголовки для отправляемого на сервер запроса;
- 3) Scenario definition определяет набор действий пользователя, которые выполняются на тестируемое приложение;
- 4) Simulation definition используется для установки количества пользователей, для одновременного запуска сценария в течение заданного периода времени.

Теперь проведем разбор по выделенным параметрам:

- 1) Имеет информативные графические отчеты, выделяющие важную информацию о тестируемой системе;
- 2) Специализирован на проверки web-приложений;
- 3) Не требует дополнительного программного обеспечения;
- 4) Gatling легко интегрируется с Jenkins, а также может запускать дополнительные тесты с помощью плагинов Maven и Gradle;
- 5) API изменялась радикальным образом. Поэтому, могут возникнуть проблемы с запуском тестов из старых версий. А также возникает путаница в документации сервиса;
- 6) Поддерживает операционную систему Windows;
- 7) Продукт сейчас поддерживается;
- 8) Является бесплатным приложением.

Отдельно стоит отметить минусы данного ПО: не поддерживается распределение нагрузки между разными узлами ВС; возможность горизонтального масштабирования отсутствует.

LoadRunner, разработан компанией Hewlett Packard, является одним из самых популярных инструментов для проведения тестов нагрузки на ВС. Можно разделить на следующие компоненты:

- 1) Virtual User Generator модуль для создания сценариев тестирования. Позволяет создавать индивидуальные схемы поведения пользователей для множества протоколов и приложений;
- 2) Controller является компонентом для управления нагрузкой;
- 3) Agents Machines – система распределения сценариев пользователей;
- 4) LoadRunner Analysis and Monitoring модуль анализа результатов исполнения сценариев тестирования;
- 5) LoadRunner Books Online большая онлайн библиотека по работе с приложением.

Проведем анализ по выбранным параметрам:

- 1) Есть подробные отчеты о работе тестов;
- 2) Поддерживает большое количество типов протоколов и программ;
- 3) Не требует дополнительного ПО;
- 4) Нет возможности расширения функционала с помощью дополнений;
- 5) Общая сложность использования, так же необходима постоянная техническая поддержка про время выполнения тестов;
- 6) Полностью работает только в Windows;
- 7) Продукт поддерживается;
- 8) Высокая стоимость продукта.

Так же можно выделить отдельные недостатки: невозможно запустить несколько сценариев из командной строки; большой объём занимаемого места на диске.

Taurus – открытая платформа, созданная компанией BlazeMeter. Главная идея создания Taurus, объединение и усиление возможностей бесплатных инструментов: JMeter, Grinder, и Gatling.

Рассмотрим параметры продукта:

- 1) Помимо обычных отчетов так же есть функция просматривать отчёты в режиме реального времени во время исполнения тестов;
- 2) Имеет довольно широкий спектр тестов и легко совмещается с JMeter, Grinder, и Gathling, существенно расширяя их производительность;
- 3) Требуется установки библиотек Microsoft Visual C++ и Python 3.7+;
- 4) Расширяется путем установки дополнительных систем тестирования;
- 5) У Taurus простая настройка и обновления. Но конфигурации оборудования придется выполнять вручную;
- 6) Работает на всех основных платформах;
- 7) Продукт поддерживается.
- 8) Бесплатный продукт.

Так же можно дополнительно выявить недостатки: Taurus поглощает много ресурсов; не поддерживаются долговременные тесты на выносливость.

Rational Performance Tester (RPT) – инструмент нагрузочного тестирования, созданный в компании IBM. Разработан для тестирования web и серверных приложений. RPT создает процесс взаимодействия пользователя с конечной службой.

Разберем возможности по параметрам:

- 1) Генерирует отчеты о тестах, так же поддерживает отчёты в режиме реального времени.
- 2) Специализирован на приложениях использующие такие протоколы как: Citrix, HTTP, SAP, Siebel, SIP, TCP.
- 3) Не требует дополнительного ПО
- 4) Нет возможности расширения функционала
- 5) Имеет высокую сложность в освоении

- 6) Приложение поддерживает операционные системы: AIX, Mac OS X, Windows
- 7) Продукт поддерживается
- 8) Имеет высокую стоимость

Отдельно стоит отметить, что в RPT отсутствует возможность уменьшения и наращивания ресурсов, отсутствует возможность планирования и не поддерживает тестирование приложений на Java Applet

В основе этих программных средств находятся различные методы тестирования ВС. Проведенный анализ существующих программных реализаций демонстрирует, что на текущий момент нет оптимальных или универсальных продуктов, способных эффективно тестировать все сетевые аспекты ВС. С другой стороны, исследованы различные методы и подходы для решения задач тестирования ВС. Эти обстоятельства и вынуждают разработчиков искать новые пути решения и создавать новые программные продукты.

Таблица 1 Сравнительные характеристики средств тестирования ВС

Продукт	JMeter	LoadRunner	Gatling	Taurus	The Grinder	Rational Performance Tester
Генерация отчетов о работе	+	+	+	+	+	+
Специализированность ПО	+	+	+	+	+	+
Потребность в дополнительном программном обеспечении	-	-	-	+	-	-
Возможность расширяемости за счет дополнений	+	-	+	+	+	-
Сложность внедрения системы и работы с ней	-	+	+	-	+	+
Независимость от платформы	+	-	-	+	+	-
Развитие и поддержка системы	+	+	+	+	+	+
Стоимость программного средства	Бесплатно	Стоимость рассчитывается от конкретной системы	Бесплатно	Бесплатно	Бесплатно	От 84000 На базовую конфигурацию на 12 месяцев

Предлагаемые решения

По результатам проведенного анализа был выдвинут ряд требований к разрабатываемой системе тестирования, а именно

- Разделение системы тестирования на подсистемы: подсистема анализа трафика ПАК, подсистема управлением генерации трафика, модульная система генерации трафика, клиентская система имитации нагрузки. Данное деление обеспечивает возможность комплексного управления сегментами тестирования;
- Модульное построение системы тестирования обеспечивает возможность реконфигурации либо расширения изделия;
- Модули поддержки устройств приводят частный протокол взаимодействия с устройством к типовому протоколу взаимодействия данного типа, что обеспечивает взаимозаменяемость применяемых устройств сохраняя единый интерфейс взаимодействия для ПО верхнего уровня;
- Генерация трафика с учетом особенности ВС, путем анализа существующего трафика, выведения его параметра Херста, для приближения свойств сгенерированного трафика к реальному;
- Режим отчетов реального времени с детальным текстовым описанием отказа с возможными рекомендациями по восстановлению работоспособности;
- Возможность работы на всех доступных операционных системах и отсутствие требований к дополнительному программному обеспечению
- Простая установка и настройка продукта вне зависимости от сложности сети, путем первичного анализа ВС и гибкой настройки
- Простой в освоении API для разработчиков и интеграторов.

Результаты работы

На текущий момент было разработано ПО, для проведения сетевых нагрузок производящие анализ сетевого трафика, с изучением его свойств и поведения, что дало возможность сгенерировать трафик приближенный по своим свойствам к показателям тестирования ВС реальными пользователями. Пакеты создаются со свойством самоподобия если на этапе анализа вычислительного комплекса было выявлено высокое значение показателя Херста, если же показатель Херста был не достаточно высок то используется подход более близкий к модели поведения реальных макетов тестируемой системы в виде модель движения Пуассона или модель длиннохвостого трафика. Разработанное ПО представлено в формате клиент-серверного приложения и выполняет функции: генерации трафика между узлами ВС, собирает статистику по пропускной способности сети, производит анализ трафика сети. Есть возможность расширения дополнительными модулями для решения других задач ВС.

Для тестирования использовалась виртуальная ВС, прототипом которой легла существующая сеть. По результатам работы программы удалось обнаружить некоторые узлы, замедляющие работу ВС. Но при проведении тестирования выяснилось, что тестирование данным методом «рабочей» сети, могут возникнуть сбои в работе клиентских приложений. В связи с этим алгоритм требует доработки в плане более гибкой настройки, для исключения конфликтов со сторонним ПО. Так же проработка более детальных отчетов и отчетов в реальном времени. Требуются продолжения исследований в выборе методов генерации трафика и при низком уровне самоподобия.

Разработка ПО проводилась средствами разработки QT Creator на языке программирования C++. Это связано с простотой использования и большим выбором библиотек для данного продукта, а также возможность создания кроссплатформенных приложений.

Список опубликованных работ по теме исследования

1. Бабич А.М., Акимов М.В, Стельмах Д.С. Структура и протоколы обмена данными модифицируемого программного комплекса // Вопросы Радиоэлектроники, 2018. №12. С. 90-95 DOI 10.21778/2218-5453-2018-12-90-95 УДК 004.89

Список использованных источников

- [1] Barton R.R., Designing simulation experiments // Proceedings of the 2002 Winter Simulation Conference. USA: IEEE. 2002. p.45-51.
- [2] Calvert K.L., Zegura E.W., Doar M. Modeling Internet Topology // IEEE Communications Magazin. 1997. p. 160-163.
- [3] Leland W.E., Taqqu M.S., Willinger W., Shermann R. Proof of a Fundamental Result in Self-Similar Traffic Modeling // ACM SIGCOMM'95, San-Fransisco. 1995.
- [4] Leland W.E, Taqqu M.S., Willinger W., Wilson D.V. On the Self-Similar Nature of Ethernet Traffic. // ACM SIGCOMM Computer Communications Review, Vol. 23, 1993,
- [5] Астафьева Н.М. Вейвлет - анализ: основы теории и примеры применения. Москва, УФН. 1996, 1145 с.
- [6] Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. Москва, Наука, 1987, 354с.
- [7] Дж. Скотт Хогдал. Анализ и диагностика компьютерных сетей. Москва, Лори. 2001. 353с.
- [8] Дубинин В.Н., Зинкин С.А. Проектирование вычислительных систем на основе сетевого формализма. Пензенский государственный университет, 1998 // Электронный ресурс: <http://alice.stup.ac.ru/~dvn/complex/title.html>.
- [9] Замятина Е.Б.. Современные теории имитационного моделирования: Специальный курс. Пермский государственный университет. Учебное пособие, 2007.119с.
- [10] Костров В. О. Разработка метода расчета пропускной способности мультисервисных сетей связи с дифференцированным обслуживанием потоков сообщений // Дисс. кандидата технических наук. Москва, 2003

- [11] Миков А.И., Замятина Е.Б. Система имитации с удаленным доступом // Материалы третьей междисциплинарной конференции с международным участием (НБИТТ-21), 2004 г. Петрозаводск, с.73.
- [12] Миков А.И., Замятина Е.Б. Технологии имитационного моделирования больших систем. Труды Всероссийской научной конференции «Научный сервис в сети Интернет». Москва, МГУ. 2008, с. 199-204.
- [13] Пастухов А.С., Матвеев С.Б. Исследование самоподобия GPRS трафика. ГОУ ВПО «МГУС», г. Москва. //Электронный ресурс: www.mgus.ru/files/electronicJournal/number5/pastuhov_matveev.doc.
- [14] Тарасов В.Н., Коннов А.Л., Ушаков Ю.А. Анализ и оптимизация локальных сетей и сетей связи с помощью программной системы OPNET Modeler // Электронный ресурс: <http://elibrarv.ru/download/28319375.pdf>.
- [15] Шелухин О.И., Тенякшев А.М., Осин А.В. Фрактальные процессы в телекоммуникациях. Под ред. Шелухин О.И. Москва, Радиотехника. 2003. 480с.
- [16] Ярославцев А.Ф. Методы и программные средства гибридного моделирования мультисервисных сетей большой размерности // Дисс. докт. технических наук. Новосибирск, 2006.

Список используемых сокращений

ВС - вычислительные сети.

ПО - программное обеспечение.

ПАК – программно-аппаратный комплекс