

# Система автоматизации регрессионного тестирования программного обеспечения с закрытым исходным кодом, разрабатываемая для мейнфреймов, работающих под управлением операционной системы z/OS

Иконников Виктор Викторович,  
Лебедев Александр Анатольевич  
ФГОУ ВПО (НИУ) "Южно-Уральский  
Государственный Университет" (г. Челябинск)  
05.13.01  
viktor.ikonnikov@gmail.com

*Аннотация. В статье описывается процесс разработки фреймворка автоматизации регрессионного тестирования программного обеспечения с закрытым исходным кодом, разрабатываемой для мейнфреймов, работающих под управлением операционной системы z/OS. В статье описана структура системы, а так же приведены ключевые аспекты реализации системы на объектно-ориентированном языке высокого уровня Python.*

**Ключевые слова:** автоматизация тестирования, регрессионное тестирование, z/OS, мейнфрейм, закрытый исходный код

## Test Framework for Black Box Regression Testing of z/OS Software

Ikonnikov Viktor Viktorovich,  
Lebedev Alexandr Anatilyevich,  
State Educational Institution of Higher Professional  
Education "South Ural State University" (Chelyabinsk)

*Abstract. The article describes the process of developing test framework for software with closed sources, developed for mainframes running under z/OS operating system. Article describes the system structure and key aspects of implementation the system on object-oriented high-level language Python.*

**Key words:** testing automation, regression testing, z/OS, mainframe, black box testing

### Введение

**В** настоящий момент на крупных промышленных предприятиях Европы, США и Японии используются мейнфреймы от компании IBM (80 % рынка мейнфреймов), работающие под управлением операционной системы z/OS.

Мейнфрейм — это вычислительная система, ориентированная на бесперебойное исполнение исключительно больших, смешанных рабочих нагрузок при высоком уровне коэффициента использования системы, обеспечивающей защиту информации, мак-

симальную производительность и высокую пропускную способность.

Развитие рынка мейнфреймов в России отстает от среднемировых показателей, однако, в последние несколько лет наблюдается рост интереса российских компаний к разработке программного обеспечения для мейнфреймов.

Программное обеспечение подобного класса, которое в первую очередь должно отвечать критериям надежности и стабильной работы в условиях непрерывности технологических процессов [1]. Достижение заданных

показателей качества возможно лишь при правильном построении процесса разработки ПО. Одним из важнейших этапов разработки ПО является тестирование [2]. В соответствии с международным стандартом ISO9126 тестирование позволяет исследовать такие качественные характеристики ПО как надежность, сопровождаемость, практичность, эффективность, мобильность и функциональность.

Наиболее эффективным способом исследования надежности разрабатываемого ПО считается поиск регрессионных ошибок, а наиболее универсальной стратегией регрессионного тестирования ПО является тестирование "черного ящика", т. е. тестирование в соответствии с заранее заданными требованиями без обращения к исходному коду программы [3].

На данный момент существует множество систем автоматизации тестирования ПО с закрытым исходным кодом, однако, большинство из них ориентировано на тестирование программного обеспечения, разрабатываемого для PC-совместимых компьютеров. Сложность программного обеспечения для мейнфреймов, его глубокая интеграция с операционной системой z/OS, ориентированность на консольный интерфейс, а также отсутствие универсальных систем автоматизации регрессионного тестирования ПО с закрытым исходным кодом заставляет разработчиков вкладывать значительные средства в разработку собственных систем, ориентированных на конкретный продукт [4].

### Постановка задачи

Необходимо разработать кроссплатформенную распределенную систему, позволяющую автоматизировать процесс тестирования программного обеспечения, разрабатываемого для мейнфреймов, работающих под управлением операционной системы z/OS.

Необходимо реализовать набор инструментов, позволяющих упростить процессы управления тестированием и ручной обработки результатов тестирования. Данные инс-

трументы должны быть реализованы как веб-фронтэнд клиент-серверного ядра системы. В веб-интерфейсе необходимо реализовать набор средств, предназначенных для управления процессом разработки тестов и эффективного взаимодействия QA-групп. В основу реализуемых инструментов управления процессами разработки и тестирования ПО с закрытым исходным кодом будет заложена современная методология управления развитием информационных систем Scrum.

Для расширения стандартного функционала системы необходимо предусмотреть прикладной интерфейс программирования API, позволяющий переопределять стандартные модули системы, а также реакцию системы на определенный класс событий.

Для ускорения процесса тестирования система должна обладать возможностью перераспределения нагрузки между мейнфреймами с тестируемым программным обеспечением.

Необходимо реализовать интерфейс взаимодействия с подсистемами ввода заданий на мейнфрейм JES2 и JES3 по стандартному протоколу FTP.

Базовые модули системы автоматизации, а также прикладной интерфейс программирования API необходимо реализовать на объектно-ориентированном высокоуровневом языке программирования Python. Модули обработки результатов тестирования будут реализованы на языке C с использованием библиотеки Pугех.

### Структура системы

Система автоматизации спроектирована как распределенное клиент-серверное приложение и конструктивно может быть разделена на три части: сервер, клиент и веб-интерфейс. На приведенном ниже рис. 1 приведена упрощенная структура связей между компонентами системы.

Клиент используется для запуска регрессионного тестирования и представляет собой копию тестирования, запущенного в данный момент на одном из мейнфреймов.

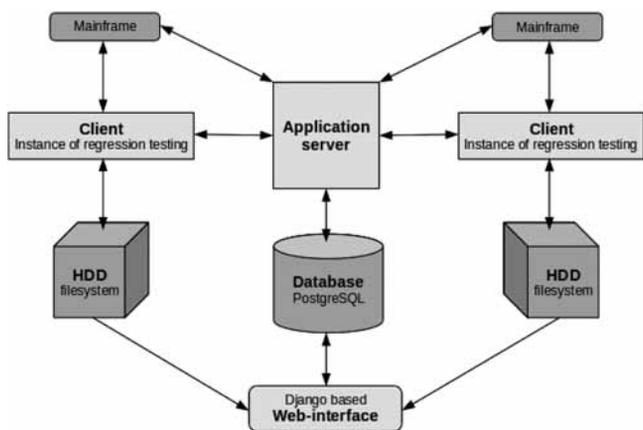


Рис. 1. Структура системы автоматизации тестирования

Выполняемые клиентом функции:

1. Чтение и проверка конфигурационных файлов.
2. Проверка целостности структуры проекта, генерация тестов.
3. Объединение и слияние тестов для различных версий тестируемого программного продукта.
4. Запуск и сохранение результатов регрессионного тестирования.
5. Обработка результатов тестирования.
6. Оповещение тестеров о возникающих ошибках и вариантах их решения.
7. Регистрация событий на сервере.

Основное предназначение сервера — контроль работы клиентов. Выполняемые сервером функции:

1. Регистрация действий, осуществляемых клиентом.
2. Выдача разрешения клиенту на выполнение тех или иных действий.
3. Осуществление контроля загруженности мейнфреймов.
4. Отмена запущенных задач при возникновении конфликтов.
5. Оповещение администратора при возникновении сбоев.

Контроль нагрузки осуществляется путем опроса программы мониторинга системных ресурсов мейнфрейма МХІ и анализа полученных данных.

В основе архитектуры клиентской части системы автоматизации регрессионного тес-

тирования будет заложен один из современных принципов построения программного обеспечения — принцип Модель—Вид—Контроллер (MVC), что позволяет разделить модель данных приложения, пользовательский интерфейс и управляющую логику на три отдельных компонента, так что модификация одного из компонентов будет оказывать минимальное воздействие на другие компоненты [5].

В архитектуре разрабатываемой системы представление отвечает за генерацию исходных тестов и отчетов о тестировании по их абстрактному описанию и шаблонам. Основное предназначение модели — сохранение и анализ результатов тестирования, а также реакция на внутрисистемные события. Контроллер осуществляет настройку окружения как для отдельного тесткейза, так и для всего тестирования в целом, поддержку целостности и работоспособности окружения, управление ходом выполнения тестирования, получение и интерпретацию результатов тестирования, а также информирование модели о необходимости реакции на внутрисистемные события.

Использование архитектуры MVC, а также выделение интерфейса прикладного программирования API, позволяющего конечному пользователю переопределить стандартные функции и реакцию системы на распространяемые в ней события, позволяет расширить базовый функционал системы, делает ее гибкой и универсальной системой, способной решать широкий класс задач в области автоматизации тестирования.

Разрабатываемая система автоматизации регрессионного тестирования представляет собой фреймворк, реализуемый на высокоуровневом языке объектно-ориентированного

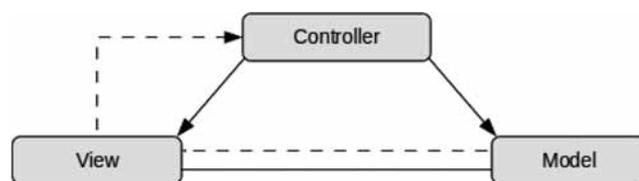


Рис. 2. Архитектура Модель—Вид—Контроллер

программирования Python. Использование языка Python, в основу которого заложены такие парадигмы программирования, как динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных, позволяет значительно ускорить процесс разработки и отладки программного обеспечения. Стоит отметить, что язык Python является одним из наиболее популярных языков программирования, используемых техническими специалистами, занимающимися тестированием.

Взаимодействия между клиентами и сервером реализовано на базе технологии удаленного вызова процедур RPC и ее реализации на языке Python — библиотеки Pyro (Python Remote Objects).

При решения задачи классификации знаний для комплексного анализа результатов регрессии на основе накопленного опыта используется метод опорных векторов для случая линейной классификации при заданном формальном описании ожидаемого результата. Модули обработки результатов тестирования реализованы на языке C с использованием библиотеки Pyrex.

Для сохранения результатов тестирования и пользовательских настроек используется система управления базами данных PostgreSQL.

Веб-фронтэнд системы реализован на базе веб-фреймворка Django. Для подсветки логов, получаемых с мейнфреймов, а также синтак-

сиса языка управления запуском пакетных заданий JCL, использована библиотека.

Для взаимодействия системы с тестируемым программным обеспечением реализован программный интерфейс, предоставляющий доступ к подсистеме ввода заданий на мейнфрейм JES2/3 по протоколу FTP.

## Внедрение

Система автоматизации используется QA-инженерами компаний ООО "Прикладные технологии" (РФ) и Rocket Software Incorporated (США) для распределенного тестирования программного продукта Advanced Allocation Optimizer. С момента введения в эксплуатацию было проведено около 350 регрессионных тестирований целевого ПО, в общей сложности запущено около 700 000 тестов. На данный момент система работает на кластере из 2 серверов и 7 мейнфреймов. Время наработки на отказ в текущей конфигурации системы составляет 10 000 ч.

## Список литературы

1. *Дастин Э., Рэшка Дж., Пол Дж.* Автоматизированное тестирование программного обеспечения. Издательство "Лори", 2003.
2. *Тамре Л.* Введение в тестирование программного обеспечения. Вильямс, 2003.
3. *Бейзер Б.* Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004.
4. *Винниченко И.* Автоматизация процессов тестирования. СПб.: Питер, 2005.
5. *Беликова Н.В., Галич М.Г.* Использование паттерна проектирования Model—View—Controller при разработке Web-приложений: Материалы III международной научно-практической конференции "Объектные системы-2011". Ростов-на-Дону, 2011