

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ВЕРИФИКАЦИОННОГО ТЕСТИРОВАНИЯ РАСЧЕТНЫХ ПРОГРАММНЫХ МОДУЛЕЙ ПАКЕТА ПРОГРАММ «ЛОГОС»

А. Ю. Логвин¹, Н. С. Аверина, Т. Н. Серова, А. В. Шемякин

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.
¹ООО «ЦКО», п. Сатис Нижегородской обл.

Введение

Процесс тестирования является неотъемлемой частью жизненного цикла программного обеспечения (ПО). Крупными компаниями, ориентированными на ПО, проводятся исследования влияния ошибок, совершенных на различных стадиях разработки ПО, на общие финансово-экономические показатели успешности реализации проекта. Рядом аналитических агентств, как за рубежом, так и в России, проводились оценки затрат на выявление и устранение ошибок. Стоимость тестирования программного продукта может составлять до 45 % от его общей стоимости. Если не так давно процесс тестирования носил бессистемный характер, проводилось в основном отладочное тестирование и устранение ошибок, полученных от пользователей уже в ходе эксплуатации ПО, то сейчас к тестированию применяется системный подход. «Тестирование выделилось в отдельную область знаний в составе разработки ПО, но быстро пришло понимание того, что тестирование вручную неэффективно, поскольку требует больших трудовых ресурсов и много времени. Первые средства автоматизации тестирования практически представляли собой библиотеки, которые можно было использовать для написания тестов, что требовало от тестировщика умения программировать на уровне разработчика. Современные средства автоматизированного тестирования позволяют создавать автоматизированные тесты с минимальным участием человека.

На рынке коммерческих средств функционального тестирования сегодня представлены главным образом продукты следующих компаний: HP (QuickTest Professional, WinRunner), IBM (Robot, Functional Tester), Borland (SilkTest) и AutomatedQA (TestComplete), представляющие собой средства разработки приложений. [1] Несмотря на то, что вышеперечисленные производители ведут гибкую политику продаж лицензий и применяют динамическое ценообразование, стоимость продуктов высока, составляет 5 000-10 000 USD за лицензию.

В связи с тем, что пакет программ «ЛОГОС» является ресурсоёмким высокопроизводительным продуктом, для его тестирования должны применяться средства, которые позволят производить запуск тестов на вычислительном кластере. Эталонные резуль-

таты представляют собой большие наборы данных, что затрудняет использование большинства свободных распространяемых продуктов, применяемых для тестирования ПО без доработки.

Принимая во внимание вышесказанное, для обеспечения автоматизированного верификационного тестирования расчетных программных модулей пакета программ «ЛОГОС» использование вышеперечисленных программных продуктов нецелесообразно и было принято решение разрабатывать собственную систему автоматизированного верификационного тестирования.

Основные подходы к разработке

Для обеспечения гибкости применения системы тестирования она разрабатывалась как кроссплатформенное web-приложение. Это помогло использовать продукт удаленно, доступ к нему возможен с различных устройств под управлением различных операционных систем, с которыми может работать web-браузер. Пользователь таких систем не привязан к месту расположения аппаратной части, где установлена система тестирования, и может использовать его из любой точки мира, где есть рекомендуемое интернет-соединение.

Важным аспектом разработки системы автоматизированного тестирования является необходимость отслеживать изменения тестовых задач и эталонных данных тестов, которые являются эталоном для сравнения полученных результатов. В ходе разработки системы тестирования было принято решение организовать хранилище эталонных данных с использованием программного обеспечения Mercurial, а в базе данных разрабатываемой системы (база верификационных тестов) хранить только путь к эталонным данным и контрольную сумму для обеспечения целостности данных.

Структура системы тестирования

Система автоматизированного тестирования реализована на языке программирования Python с использованием web-фреймворка Django. Она построена по клиент-серверной архитектуре. Сервер приложений и база данных функционирует в ОС

Linux, для взаимодействия с пользователем используется любой веб-браузер, входящий в состав базового программного обеспечения всех современных операционных систем (Linux, Windows и т. д.). Функционально система состоит из четырех блоков: 1) интерфейс пользователя 2) сервер управления, 3) счетное поле для запуска тестов, 4) База данных. Блоки представлены на рис. 1.

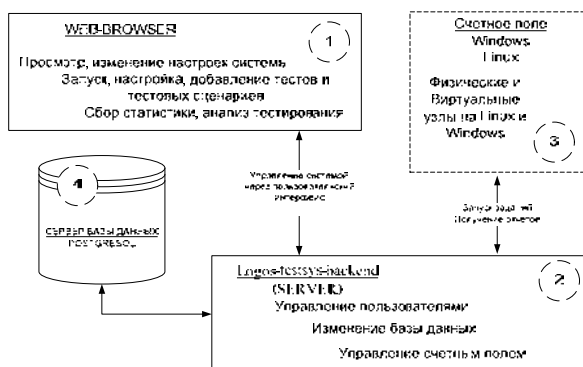


Рис. 1. Функциональная схема системы тестирования

В состав системы тестирования входят следующие взаимодействующие между собой компоненты:

- веб-сервер *Nginx*;
- СУБД *PostgreSQL*;
- NoSQL база данных *Memcached*;
- брокер сообщений *RabbitMQ*;
- подсистема управления;
- подсистема запуска периодических задач;
- подсистема обнаружения новых тестов;
- подсистема обнаружения новых версий тестируемого ПО;
- подсистема запуска тестов и анализа результатов.

Веб-сервер *nginx* предназначен для перенаправления HTTP-запросов из браузера подсистеме управления *logos-testsys-backend*.

СУБД *Postgresql* предназначена для долговременного хранения настроек системы, тестов, результатов запуска тестов, а также авторизационных данных пользователей системы.

NoSQL база данных *memcached* предназначена для хранения временных системных данных, а также для синхронизации процессов доступа к разделяемым ресурсам с целью избежания коллизий, возможных при одновременном изменении их содержания.

Брокер сообщений *RabbitMQ* предназначен для регистрации и хранения задач, выполняющихся в неблокирующем режиме, с целью максимально эффективного использования серверных ресурсов, выделенных системе.

Подсистема *logos-testsys-backend* представляет собой ядро, связывающее компоненты системы в одно целое.

Программные и технические средства

Для функционирования системы тестирования необходимы аппаратные средства, имеющие следующие минимальные характеристики:

- центральный процессор, совместимый с X86-64 архитектурой;
- оперативная память не менее 1 Гб;
- свободное дисковое пространство не менее 1 Гб;
- коммуникационные среды Ethernet.

Система тестирования работает под управлением следующих операционных систем:

- Windows 7;
- WindowsServer 2008 R2;
- ScientificLinux 6.x.

Версии сторонних программных продуктов, пакетов и библиотек, используемых при работе системы тестирования:

- PostgreSQLServer 9.4.14
- Mercurial
- RabbitMQ 3.2.4
- MPI
- Slurm
- Python 2.7.6 [1]
- pip 1.5.4
- nginx 1.12.1
- uwsgi 1.9.17.1
- celery 4.1.0
- Django 1.11.6 [2]
- django-admin-list-filter-dropdown 1.0.1
- django-ordered-model 1.4.3
- django-constance 2.0.0
- django-daterange-filter 1.3.0
- django-nested-admin 3.0.20
- django-picklefield 1.0.0
- django-taggit 0.22.1
- paramiko 2.3.1
- ryparsing 2.2.0
- pywinrm 0.2.2
- PyYAML 3.12
- requests 2.18.4
- tabulate 0.8.1
- taggit-selectize 2.3.0

Функциональные возможности

Разработанная система автоматизированного тестирования предназначена для проведения тестирования программного обеспечения, не обладающего графическим пользовательским интерфейсом, в частности счетных модулей пакета программ «ЛЮГОС» и анализа результатов под различными операционными системами и окружениями.

Система тестирования реализует управление следующими компонентами:

- 1) тесты;
- 2) решатели – непосредственно тестируемые компоненты «ЛЮГОС»;
- 3) пользователи и организации;
- 4) инфраструктура системы тестирования.

Работа в системе тестирования осуществляется через веб-интерфейс. После входа в систему пользователю предоставляется основное рабочее окно, рис. 2.

В этом окне пользователь имеет доступ ко всем функциям системы, для которых у него есть права доступа. Права доступа устанавливаются в специальном разделе. Их может изменять пользователь с правами администратора системы (суперпользова-

тель). На рис. 3 показано диалоговое окно управления доступом пользователей.

Основными шагами процесса тестирования, реализованного в описываемой системе, являются:

- 1) загрузка эталонных данных от разработчиков;
- 2) загрузка новых версии тестируемого ПО, в частности решателей «ЛОГОС»;

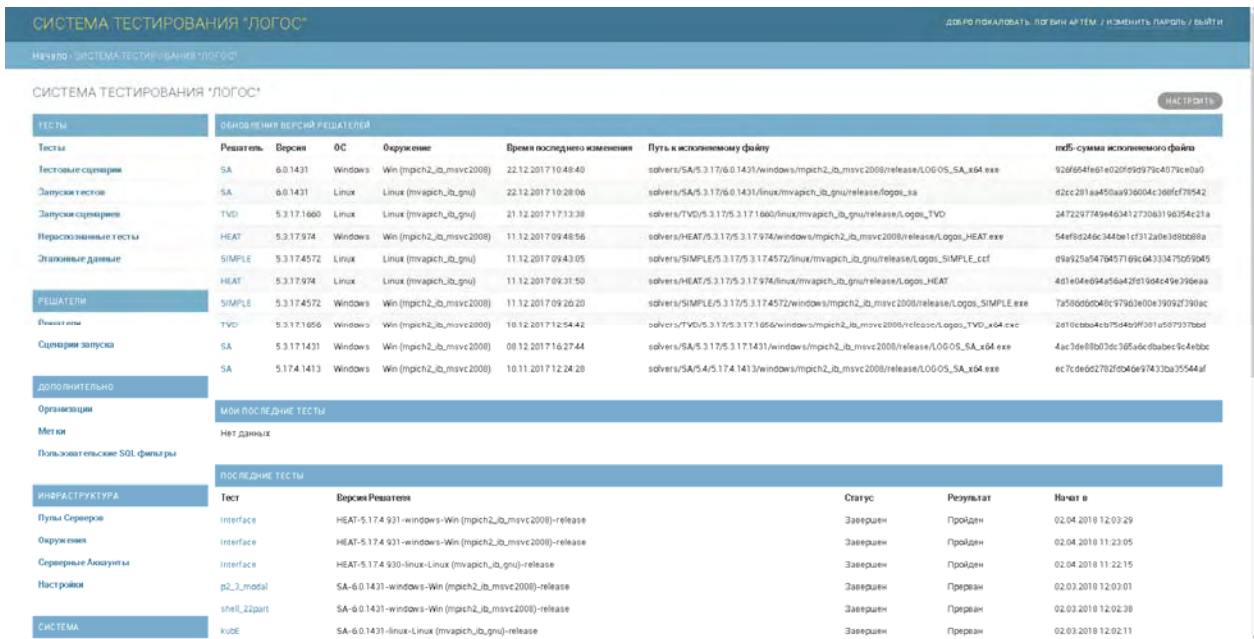


Рис. 2. Основное рабочее окно системы

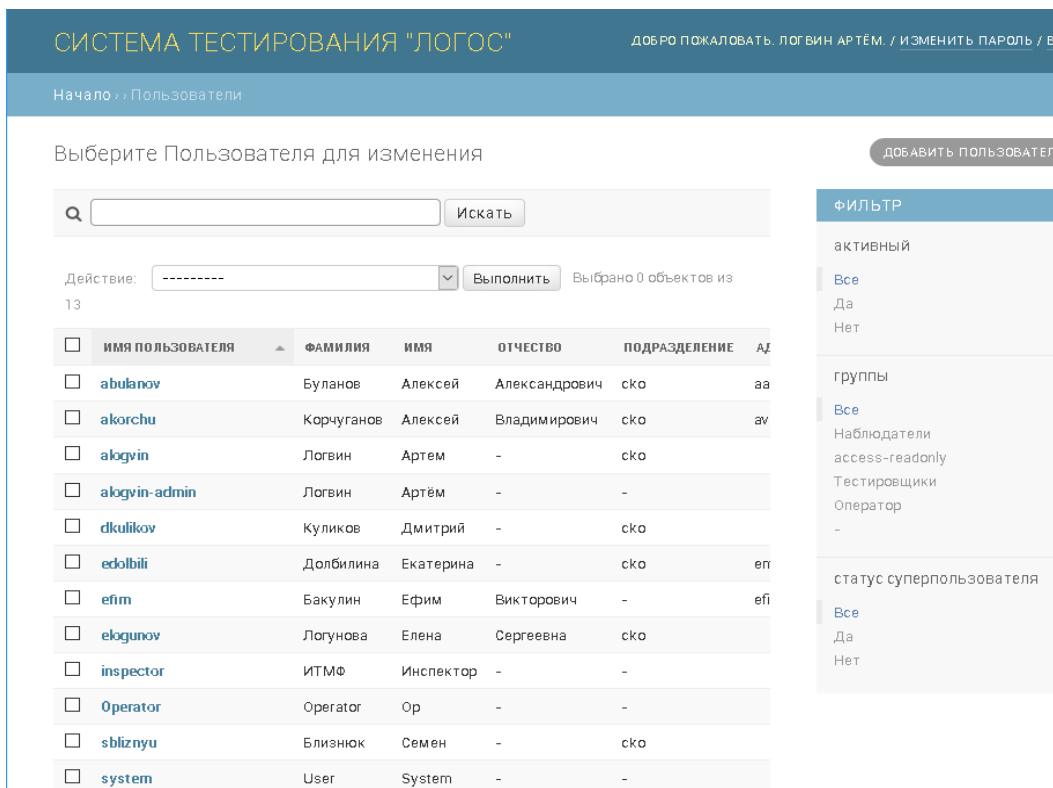


Рис. 3. Диалоговое окно управления доступом пользователей

3) запуск тестов с указанными параметрами в автоматическом или, при необходимости, ручном режиме;

4) автоматическое сравнение полученных результатов с эталонными данными;

5) вывод сводной информации тестирующему с возможностью просмотра деталей выполнения теста.

Основная задача системы – запуск тестов и анализ результатов под различными операционными системами и окружениями. Для этого предусмотрена возможность изменять и удалять компоненты счетного поля, переменные системного окружения и серверные аккаунты. Компонентами системного окружения являются настройки библиотеки запуска параллельных задач – MPI и планировщик заданий SLURM (только под ОС Linux.). На рис. 4 представлены соответствующие разделы меню. Эти настройки служат для добавления, удаления и изменения серверов под управлением ОС Linux или Windows на которых будут запускаться тестовые задачи.

Добавление и изменение решателей производится в разделе меню «Решатели». Решатели, как уже говорилось, представляют собой основной тестируемый компонент «ЛЮГОС». В этом разделе пользователь может добавить или изменить сценарий запуска тестов. Так как система разрабатывалась максимально гибкой, то сценарии запуска тестов можно модернизировать на основании простых шаблонов, написанных на сценариях bash или bat под Linux и Windows соответственно. Пример тестового сценария запуска теста и верификации данных представлен на рис. 5.

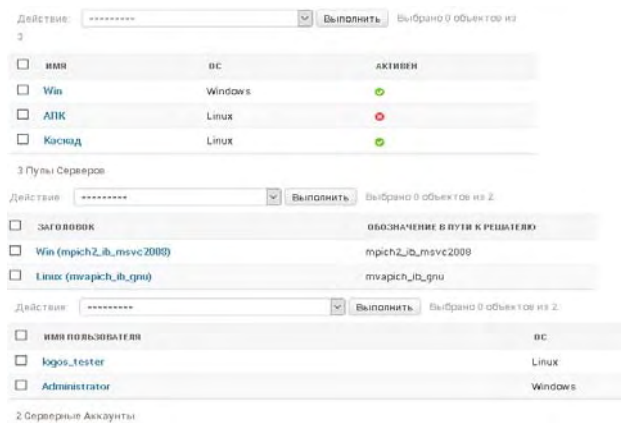


Рис. 4. Разделы меню управления настройками серверов



Рис.5. Пример тестового сценария

В базу верификационных тестов можно добавлять новые тесты по заданным критериям, также можно осуществлять поиск и выбор тестов, удалять и изменять тесты. Обнаружение в системе контроля версий Mercurial новых тестов и входных данных, предназначенных для проверки корректности работы решателя, выполняется Системой автоматически.

Система также предоставляет возможность формировать набор тестов из имеющихся в Базе верификационных тестов, и сохранять их для последующего использования.

Также Система осуществляет автоматическое сканирование хранилища на наличие новых версий решателя.

Корректность проведенных тестов подтверждается на основе эталонных данных по заданным критериям сравнения, которые предоставляют разработчики тестируемого ПО. Результаты тестов сравниваются с эталонными данными и система сигнализирует об успешности или не успешности прохождения того или иного теста.

В системе предусмотрено протоколирование действий пользователей и ведется протоколирование процесса тестирования, также формируется краткий сводный отчет по результатам тестирования.

Кроме того, система тестирования позволяет устанавливать различные метки для тестов, добавлять и удалять организации, привлекаемые к тестированию и настраивать различные уведомления.

Функция периодических задач позволяет производить автоматический запуск фоновых задач по определенному расписанию. К этим задачам относятся:

- автоматическое обнаружение новых тестов;
- автоматическое обнаружение новых версий решателей.

В процессе разработки система автоматизированного верификационного тестирования расчетных программных модулей пакета программ «ЛЮГОС» была развернута на кластере «Каскад» в ООО «ЦКО», произведен тестовый запуск системы. Работоспособность подтверждена на кластере «Каскад», состоящем из 80 вычислительных узлов. Тестирование проводилось с использованием набора тестов, предоставленных РФЯЦ-ВНИИЭФ для выполнения данных работ. На рис. 6 показана статистика запуска тестов.

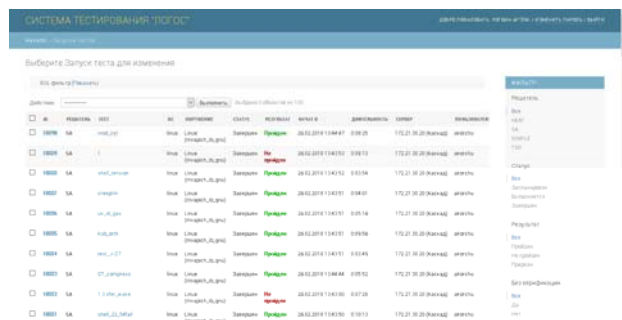


Рис. 6. Окно запуска тестов

В процессе разработки система тестирования проходила процесс верификации путем сравнения результатов запуска тестов из базы верификационных тестов в ручном режиме и с помощью данного программного средства на вычислительных мощностях ООО «ЦКО». Результаты верификации: см. таблицу.

Результаты верификации

Кол-во тестов	Счетный модуль (решатель)	Результат сравнения ручного и автоматизированного тестирования
96	ЛОГОС-Прочность (Logos-SA)	совпадает
68	ЛОГОС-Гидродинамика (Logos-SIMPLE)	совпадает
38	ЛОГОС-Тепло (Logos-HEAT)	совпадает
3	ЛОГОС-Аэродинамика (Logos-TVD)	совпадает

Заключение

Разработанная система автоматизированного верификационного тестирования расчетных про-

граммных модулей пакета программ «ЛОГОС» была развернута на кластере «Каскад» в ООО «ЦКО» и успешно прошла процесс верификации.

В ходе сопровождения, тестирования и верификации системы автоматизированного верификационного тестирования расчетных программных модулей пакета программ «ЛОГОС» проведены все испытания на соответствие необходимым требованиям Технического задания на разработку. Система успешно интегрирована в работы математического отделения ИТМФ РФЯЦ-ВНИИЭФ по разработке счетных модулей пакета программ «ЛОГОС». В ходе использования системы тестирования в ИТМФ был сформирован список доработок, которые планируется реализовать в 2018 г.

Литература

1. Pythondocumentation. [Электронный ресурс] Режим доступа: <https://docs.python.org/2.7/>
2. Djangodocumentation. [Электронный ресурс] Режим доступа: <https://docs.djangoproject.com/en/1.11/>
3. Schedmddocumentation. [Электронный ресурс] Режим доступа: <https://slurm.schedmd.com/>