

# Math-Net.Ru

Общероссийский математический портал

А. В. Самочадин, О. И. Сужаев, Д. А. Тимофеев, П. А. Рогов, Инструментальные средства нагрузочного тестирования для систем централизованного управления мобильными устройствами,  
*Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление*, 2014, выпуск 6, 33–42

<https://www.mathnet.ru/ntitu83>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<https://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.9.173

12 мая 2025 г., 21:12:00





УДК 004.7

*А.В. Самочадин, О.И. Сужаев, Д.А. Тимофеев, П.А. Рогов*

## **ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ ДЛЯ СИСТЕМ ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ МОБИЛЬНЫМИ УСТРОЙСТВАМИ**

*A.V. Samochadin, O.I. Suzhaev, D.A. Timofeev, P.A. Rogov*

### **TOOLS FOR LOAD TESTING MOBILE DEVICE MANAGEMENT SYSTEMS**

Статья посвящена разработке средств нагрузочного тестирования систем централизованного управления мобильными устройствами (систем MDM). Важное требование к системам MDM – возможность работы в условиях больших пиковых нагрузок, когда к системе одновременно обращаются несколько сотен или тысяч мобильных устройств. Проверка функционирования системы в таких условиях с помощью реальных мобильных устройств оказывается дорогой и трудоемкой. Решением является использование программных средств, моделирующих одновременную работу многих мобильных устройств. В статье сформулированы требования к программным средствам нагрузочного тестирования MDM и предложена архитектура таких средств, позволяющая моделировать большое количество разнотипных мобильных устройств и допускающая настройку на новые платформы или новые версии мобильных операционных систем.

**НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ; АРХИТЕКТУРА; УПРАВЛЕНИЕ МОБИЛЬНЫМИ УСТРОЙСТВАМИ.**

This paper deals with the problem of load testing mobile device management (MDM) systems. MDM systems should cope with peak loads generated by hundreds and thousands of mobile devices at a time. It is too hard and expensive to test the capabilities of an MDM system with so many real devices operating simultaneously. A much more convenient approach is to use load testing software that emulates numerous mobile devices. We state a set of requirements for the software designed for load testing MDM server, and propose the architecture of a load testing tool. This architecture allows emulating the main mobile platforms (Android, iOS, Windows Phone) and can be modified to support new platforms or operating system versions.

**LOAD TESTING; ARCHITECTURE; MOBILE DEVICE MANAGEMENT.**

Рост использования мобильных устройств в корпоративной среде приводит к активному внедрению средств централизованного управления мобильными устройствами (mobile device management – MDM). Система MDM представляет собой программное обеспечение для работы с корпоративными системами при помощи мобильных устройств, основными функциями которого являются управление распространением программного обеспечения, управление политиками, безопасностью и предоставляемыми сервисами [1].

В корпоративной среде используются как принадлежащие организации мобиль-

ные устройства, так и собственные устройства сотрудников. При этом использование личных устройств в служебных целях (подход Bring Your Own Device) становится все более популярным. В частности, компания Gartner [2] прогнозирует, что к 2017 г. 50 % компаний будут использовать для ведения бизнеса мобильные устройства своих сотрудников. Такой подход избавляет работодателей от необходимости приобретать мобильные устройства, но в то же время препятствует формированию унифицированной среды. Требуя от сотрудника использовать личные устройства, компания имеет чрезвычайно ограниченную возмож-

ность влияния на выбор производителя устройства, используемой операционной системы, установленных на устройстве приложений. Организация корректного взаимодействия большого количества разнообразных мобильных устройств становится основной функцией корпоративной системы MDM.

При разработке программного обеспечения MDM важной задачей является проверка корректности функционирования системы в различных режимах работы. Проверка корректности включает функциональное, интеграционное и нагрузочное тестирование. Функциональное тестирование позволяет убедиться в том, что программное обеспечение правильно реализует все определенные в спецификации возможности. Цель интеграционного тестирования – проверка корректной совместной работы компонентов системы. Задача нагрузочного тестирования состоит в исследовании способности тестируемой системы выполнять свои функции при повышении интенсивности запросов к ней.

При проведении нагрузочного тестирования важно учитывать, что взаимодействие мобильного устройства и системы MDM представляет собой протяженный во времени процесс. Так как устройства пользователей не однотипны, процессы взаимодействия двух разных устройств с сервером MDM будут иметь разные свойства. Например, при реализации MDM в разных мобильных платформах используются различные протоколы, каждый из которых имеет свои временные характеристики и требует передачи разного объема данных. Как следствие, для корректного моделирования нагрузки на систему MDM необходимо учитывать особенности функционирования клиентских устройств. Сформировать необходимую нагрузку, используя реальные устройства, в ходе нагрузочного тестирования невозможно: в крупных организациях за короткий промежуток времени к серверу MDM могут обращаться сотни и тысячи устройств. Для снижения стоимости испытаний формирование тестовой нагрузки на систему выполняется с помощью специализированных программных средств, мо-

делирующих работу большого количества одновременно функционирующих мобильных устройств.

Поскольку нагрузочное тестирование широко применяется при разработке распределенных приложений, в частности, web-сервисов, существует много инструментов для формирования тестовой нагрузки (например, [3–8]). Эти инструменты по заданным правилам обращаются к сетевым сервисам, имитируя действия большого числа пользователей. Однако при тестировании системы MDM необходимо учитывать ряд особенностей, не позволяющих использовать широко распространенные инструменты нагрузочного тестирования. В данной статье анализируются требования к инструментальным средствам тестирования систем MDM и описывается архитектура такого средства.

#### **Требования к средствам нагрузочного тестирования MDM**

Для передачи данных между сервером MDM и мобильными устройствами обычно используется протокол HTTP [9, 10]. Для основанных на этом протоколе систем, таких как web-сайты или web-сервисы, разработано множество инструментальных средств нагрузочного тестирования. Однако организация нагрузочного тестирования систем MDM требует учета их специфических особенностей, что оказывается невозможным при использовании большей части существующих инструментов.

Инструментальные средства нагрузочного тестирования можно разделить на два класса: средства для тестирования систем без состояния и систем с изменяемым состоянием. К системам без состояния относятся простые web-сайты: результат обращения к странице сайта не зависит от предшествующих обращений. Для нагрузочного тестирования таких систем достаточно обеспечить одновременное поступление на сервер большого количества запросов, что достигается путем параллельного формирования запросов к фиксированному набору страниц или сервисов. К этому классу относятся системы нагрузочного тестирования `http_load` [3], `httperf` [6], `ab` [7].

Используя средства управления состоянием протокола HTTP [11], можно реализовать более сложное поведение системы, при котором результат обращения к сервису определяется не только параметрами запроса, но и его предысторией. Тестирование систем с изменяемым состоянием требует использования моделей, описывающих изменение состояния системы в зависимости от действий пользователя [12]. Поскольку при нагрузочном тестировании целью является не проверка функциональной корректности, а исследование поведения системы под нагрузкой, вместо формальных моделей системы удобно использовать заданные пользователем сценарии, которые определяют необходимую последовательность действий. К инструментам такого класса относятся, в частности, системы JMeter [4], Grinder [5], SoapUI [8].

Системы MDM являются системами с изменяемым состоянием. Простые средства нагрузочного тестирования, не поддерживающие задаваемые пользователем сценарии, могут иметь только ограниченное применение при тестировании систем MDM. Чтобы определить степень применимости инструментов, которые используют сценарии, необходимо сопоставить общие свойства web-сервисов, обладающих состоянием, и специфические свойства систем MDM.

Одним из важных свойств систем MDM является специфический набор запросов, который в общем случае зависит от набора реализованных в системе MDM сервисов и от используемых мобильных платформ. Реализованный на мобильном устройстве пользователь клиент MDM не просто формирует запросы, но и выполняет значительный объем вычислений: генерирует запросы на подпись сертификатов, шифрует данные, выполняет команды управления устройством. От выполнения этих действий зависят временные характеристики процесса взаимодействия устройств с сервером MDM. Кроме того, корректное формирование набора отправляемых данных может быть необходимо для правильного изменения состояния сервера MDM. Даже когда реализация такого набора возможностей средствами языка сценариев конкретного

инструмента нагрузочного тестирования возможна, она требует значительных усилий.

Еще одним, причем наиболее существенным, отличием системы MDM от традиционных web-сервисов является то, что не только пользователь (клиент MDM на мобильном устройстве), но и сервер MDM может выступать инициатором сеанса взаимодействия. При тестировании необходимо исследовать показатели работы системы MDM в режиме, когда взаимодействие осуществляется по инициативе сервера. Существующие системы нагрузочного тестирования такой возможности не предоставляют.

Таким образом, для организации нагрузочного тестирования сервера MDM необходимо инструментальное средство, обладающее следующими свойствами.

- Средство тестирования должно обеспечивать необходимую нагрузку на сервер MDM, моделируя одновременную работу достаточно большого количества (несколько тысяч) мобильных устройств.
- Запросы к серверу должны формироваться с учетом реалистичных сценариев взаимодействия мобильных устройств и системы MDM. Необходимо моделировать поведение всех основных мобильных платформ: Android, iOS, Windows Phone.
- Должна быть реализована возможность тестирования сервера MDM в режиме, когда он выступает в качестве инициатора сеанса взаимодействия.

Представленные на рынке системы такого класса не универсальны. Например, компания BluePoint Security предоставляет специализированный инструмент для нагрузочного тестирования систем MDM BluePoint Enterprise Edition MDM Load Simulator [13]. Он обладает нужными свойствами, однако может использоваться только вместе с системой MDM BluePoint Enterprise Edition для ее настройки и оценки производительности. Как следствие, представляется необходимой разработка собственного решения.

Перечислим более детально требования к инструментальным средствам нагрузочного тестирования систем MDM, сформу-

лированные на основе анализа существующих систем и особенностей взаимодействия сервера MDM и мобильных устройств.

- Возможность начинать сессию управления устройством по инициативе сервера.
- Возможность обрабатывать управляющие команды сервера и генерировать корректные с точки зрения протокола MDM ответы.
- Возможность корректно обрабатывать неверные запросы и команды, поступающие от сервера.
- Возможность моделировать регистрацию устройств в системе MDM.
- Возможность собирать информацию о времени отклика системы MDM при одновременно выполняемых операциях регистрации и управления устройствами.
- Возможность использовать для тестирования заранее заданные сценарии.
- Возможность оперативно изменять сценарии тестирования.

Инструмент нагрузочного тестирования должен моделировать поведение реальных мобильных устройств, выполняющих протокол взаимодействия с сервером MDM. Сценарии предназначены для выбора проверяемых наборов сервисов.

Разнообразие и быстрое развитие мобильных платформ определяют дополнительные требования к инструментальным средствам, связанные с возможностью адаптации к быстро изменяющемуся рынку мобильных платформ.

- Одновременная поддержка различных версий мобильных операционных систем.
- Возможность добавлять поддержку новых версий мобильных операционных систем и новых мобильных платформ.

#### **Архитектура инструментального средства нагрузочного тестирования**

Мы предлагаем архитектуру инструментального средства нагрузочного тестирования систем MDM, позволяющую выполнить сформулированные требования. На рис. 1 на примере одной мобильной операционной системы представлены структурная схема инструмента тестирования и его взаимодействие с сервером MDM. Компоненты, моделирующие поведение других

операционных систем, отличаются только реализацией.

Инструментальное средство состоит из следующих компонентов:

модуль тестирования, отвечающий за сбор информации о поведении системы, выявление ошибок и запуск необходимых сценариев тестирования;

модуль управления, отвечающий за выполнение сценария тестирования и проверку корректности состояния симулятора в процессе тестирования;

симуляторы мобильных операционных систем, описывающие поведение каждой конкретной версии операционной системы при взаимодействии с сервером MDM. Симулятор содержит три компонента: модуль состояния симулятора, модуль регистрации устройства и модуль команд.

Взаимодействие мобильного устройства с сервером MDM возможно в двух режимах: режиме регистрации устройства на сервере MDM и режиме выполнения команд. При регистрации данные устройства заносятся в базу данных MDM, создаются необходимые сертификаты, а на устройство устанавливается приложение-агент, с которым в дальнейшем будет взаимодействовать сервер MDM. Набор функций агента зависит от мобильной платформы. Операционные системы iOS и Windows Phone 8 уже содержат реализации протоколов MDM, позволяющие зарегистрировать устройство на сервере и выполнять базовые команды (например, блокировку устройства по команде с сервера MDM). В то же время некоторые операции, такие как определение местоположения устройства, не поддерживаются встроенным клиентом MDM и выполняются агентом. При необходимости агент запрашивает у пользователя разрешение на выполнение определенного класса операций, таких как чтение данных сервиса местоположения или доступ к списку контактов. Операционная система Android не содержит стандартной реализации протокола MDM, поэтому агент выполняет все необходимые для взаимодействия с сервером операции.

Таким образом, порядок регистрации устройства на сервере MDM будет своим

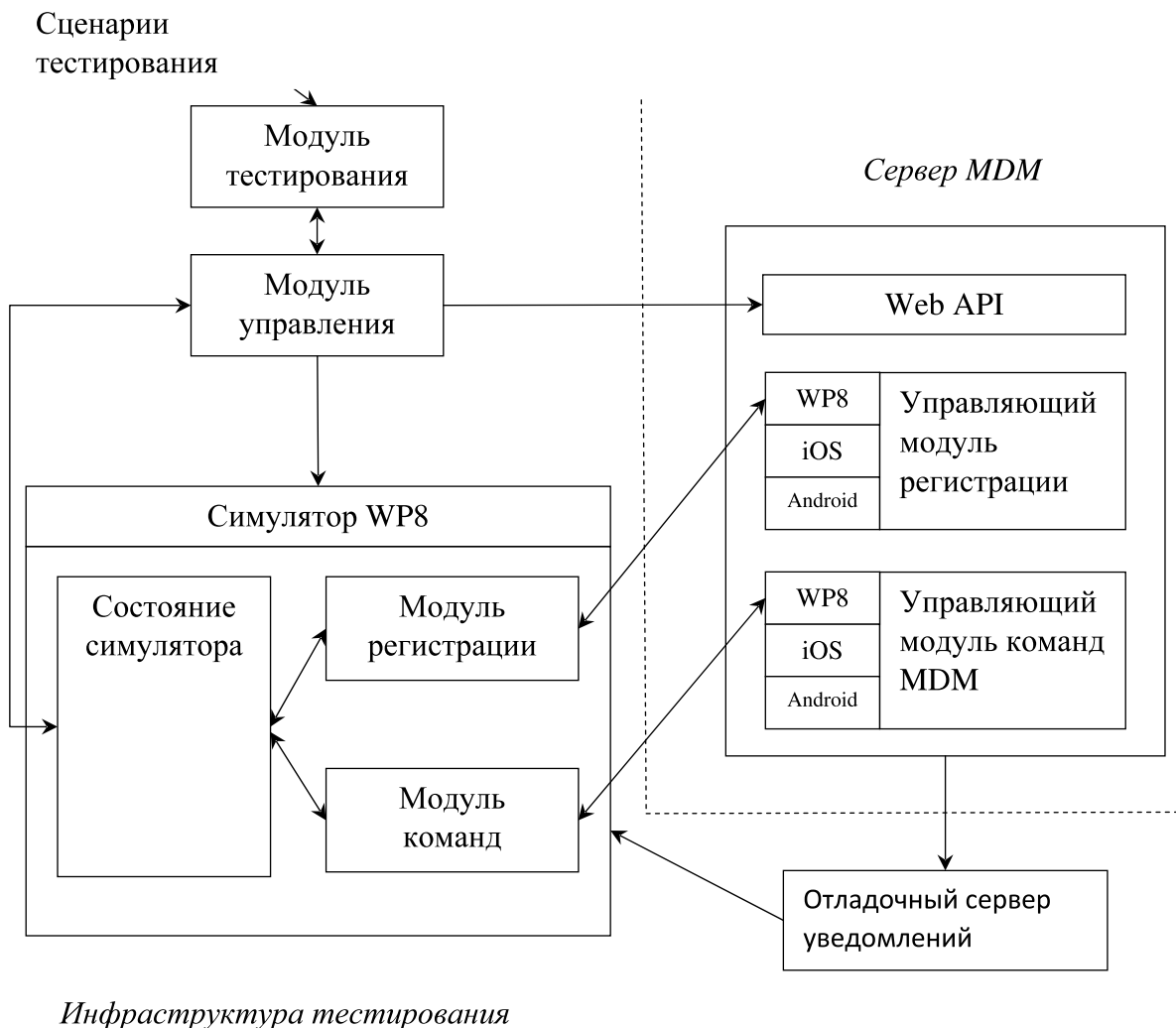


Рис. 1. Архитектура средств тестирования MDM

для каждой мобильной платформы. В режиме выполнения команд взаимодействие с устройствами производится уже по общему протоколу управления устройствами, однако содержание и, в некоторых случаях, способы представления запрашиваемых и передаваемых устройством данных различаются для разных мобильных платформ. В связи с этим в режиме команд сервер MDM также вынужден учитывать ряд специфических для платформы деталей. Таким образом, сервер MDM включает два основных модуля, отвечающих за взаимодействие с устройствами: управляющий модуль регистрации устройств и управляющий модуль команд MDM. Каждый из них включает

часть, не зависящую от конкретной мобильной платформы на клиентском устройстве, и реализации платформенно-зависимой части протокола для всех поддерживаемых мобильных платформ. Как следствие, при тестировании сервера MDM также требуется разработка отдельного симулятора для каждой поддерживаемой мобильной платформы. При этом модули команд в симуляторах разных платформ, отличаясь в деталях реализуемого протокола, оказываются в значительной степени похожими между собой, что позволяет вынести общую часть в повторно используемую библиотеку.

Еще одним важным компонентом сервера MDM является программный интерфейс

Web API. С его помощью как мобильные устройства, так и другие корпоративные сервисы могут взаимодействовать с сервером MDM при организации мобильных сервисов. Например, одна из реализуемых с помощью Web API функций состоит в публикации документов, которые затем передаются сервером MDM на мобильные устройства пользователей. Сценарии тестирования могут включать обращения к Web API для проверки функционирования и временных характеристик таких сервисов.

Система тестирования представляет собой программную библиотеку, которая может быть интегрирована с различными инструментами автоматизации тестирова-

ния. Это позволяет использовать систему тестирования в разных окружениях (ручное тестирование, автоматическое тестирование при ежедневных сборках проекта), а также использовать возможности системы не только для нагрузочного, но и для интеграционного тестирования.

Для тестирования сервера MDM необходимо разработать набор сценариев и определить конфигурацию, в которой будут проводиться тесты: набор используемых мобильных платформ и количество моделируемых устройств для каждой из них. За создание необходимой конфигурации отвечает модуль тестирования. При интеграции с инструментами автоматизации тестиро-

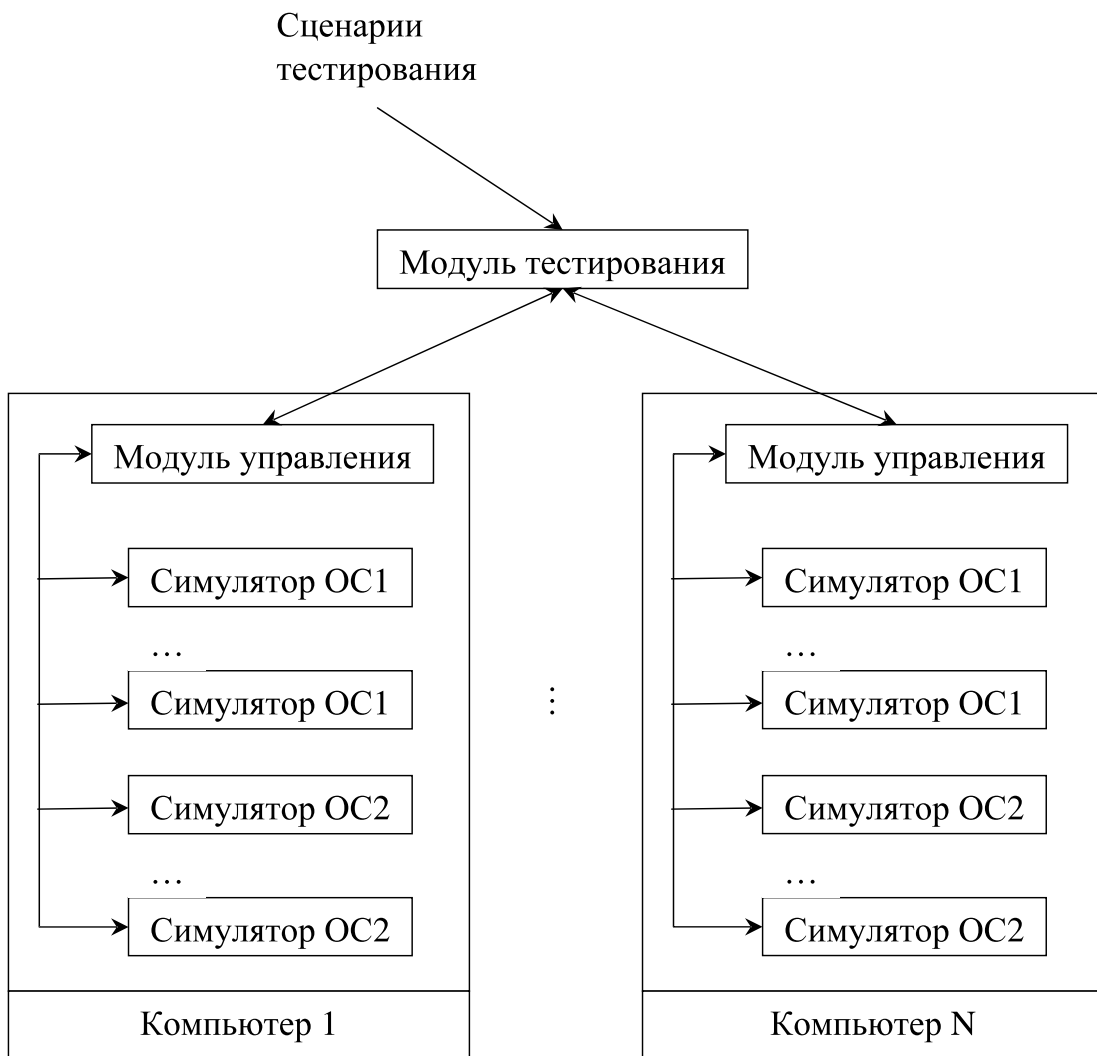


Рис. 2. Структура распределенной системы тестирования

вания может потребоваться создание нескольких вариантов модуля тестирования на основе общего библиотечного кода. В проекте в основном используется модуль тестирования, интегрированный со средой модульного тестирования NUnit.

В зависимости от планируемой нагрузки, симуляторы мобильных платформ могут быть запущены на одном или нескольких компьютерах. При запуске на одном компьютере симуляторы выполняются в виде многих потоков операционной системы в рамках процесса, создаваемого модулем тестирования, и взаимодействуют с модулем тестирования непосредственно. При использовании нескольких компьютеров процессы, выполняемые на каждом из них, взаимодействуют с модулем тестирования по сети. Структура распределенной системы тестирования представлена на рис. 2.

Модуль управления отвечает за выполнение сценариев тестирования. Он включает в себя интерпретатор языка сценариев и средства для слежения за состоянием симуляторов.

Симулятор мобильной платформы является центральным компонентом системы тестирования. Каждый экземпляр симулятора имеет собственное состояние, которое включает протокол функционирования симулятора, набор данных, описывающих моделируемое устройство, и данные, формируемые в процессе взаимодействия с сервером MDM (например, набор сертификатов). Модуль управления передает каждому экземпляру симулятора команды, которые необходимо выполнить в соответствии со сценарием, такие как «зарегистрироваться в системе MDM» или «запросить документ с заданным идентификатором», и проверяет соответствие состояния симулятора ожидаемому.

Модуль регистрации симулятора отвечает за выполнение процедуры регистрации моделируемого мобильного устройства в системе MDM. В процессе регистрации формируются уникальные элементы состояния симулятора, которые используются при дальнейшем взаимодействии симулятора и сервера MDM. Модуль команд фор-

мирует и посылает серверу MDM запросы, соответствующие описанным в сценарии действиям.

Важным требованием к системе является возможность тестирования сервера MDM в условиях, когда инициатором взаимодействия является не мобильное устройство, а сам сервер MDM. При взаимодействии с реальными мобильными устройствами для этого используется механизм push-уведомлений. Чтобы оповестить устройство о необходимом действии, сервер MDM отправляет сообщение на сервер уведомлений, а мобильное устройство периодически обращается к этому серверу, чтобы получить новые уведомления. Все три поддерживаемые мобильные платформы используют централизованные серверы уведомлений, каждый из которых поддерживается производителем платформы. Таким образом, сервер MDM взаимодействует с серверами уведомлений, принадлежащими компаниям Apple, Google и Microsoft. Взаимодействие между сервером уведомлений и мобильным устройством происходит по закрытым, уникальным для каждой платформы, протоколам. Даже при условии реализации такого протокола в симуляторе, проблематичной представляется регистрация на сервере уведомлений виртуальных мобильных устройств, которые создаются системой тестирования на короткий срок. По этим причинам вместо настоящих серверов используется отладочный сервер уведомлений, рассчитанный на взаимодействие только с симуляторами мобильных устройств. Сервер MDM взаимодействует с отладочным сервером как с официальным сервером уведомлений соответствующей мобильной платформы. Симулятор мобильного устройства периодически обращается к отладочному серверу, проверяя наличие уведомлений, и выполняет связанные с полученными уведомлениями действия.

Предложенная архитектура реализована и используется для организации нагрузочного, а также интеграционного тестирования разрабатываемой системы MDM. В настоящее время моделируются мобильные платформы Android (версии Jelly Bean 4.1.1,



4.2.2 и KitKat 4.4.2, 4.4.3), iOS версий 6.1, 7.0, 7.1 и Windows Phone 8.1.

Количество одновременно моделируемых мобильных устройств не ограничено. Эксперименты показывают, что на одном компьютере могут быть смоделированы около тысячи устройств. Более высокая нагрузка обеспечивается путем параллельного выполнения системы тестирования на нескольких соединенных в локальную сеть компьютерах.

В статье сформулированы требования к средствам инструментальной поддержки нагрузочного тестирования систем централизованного управления мобильными устройствами и предложена архитектура такого средства.

Описанная система тестирования имеет модульную архитектуру, что позволяет добавлять поддержку новых версий мобильных операционных систем, повторно используя уже реализованные общие функции, а также интегрировать систему тестирования с различными инструмента-

ми автоматизации сборки и тестирования программ. Реализация отладочного сервера уведомлений обеспечила возможность тестирования сервера MDM в сценариях, предполагающих выполнение действий по инициативе сервера. При этом для всех моделируемых мобильных платформ используется единый протокол взаимодействия между сервером уведомлений и симуляторами устройств, что упрощает добавление новых мобильных платформ.

Поскольку каждый симулятор работает независимо от других, система легко масштабируется для достижения необходимых показателей нагрузки путем запуска на нескольких компьютерах.

Разработка программного обеспечения для систем централизованного управления мобильными устройствами проводится в рамках совместного проекта компании IBS (Москва) и Санкт-Петербургского государственного политехнического университета. Работа выполняется при финансовой поддержке Министерства образования и науки РФ, госконтракт № 02.G25.31.0024 от 12.02.2013.

#### СПИСОК ЛИТЕРАТУРЫ

1. Gartner IT-Glossary, 2013 [электронный ресурс] / URL: <http://www.gartner.com/it-glossary> (дата обращения: 06.11.2014).
2. Gartner. Gartner Predicts by 2017, Half of Employers will Require Employees to Supply Their Own Device for Work Purposes, 2013 [электронный ресурс] / URL: <http://www.gartner.com/newsroom/id/2466615> (дата обращения: 07.11.2014).
3. http\_load – multiprocessing http test client [электронный ресурс] / URL: [http://www.acme.com/software/http\\_load/](http://www.acme.com/software/http_load/) (дата обращения: 07.11.2014).
4. Apache JMeter [электронный ресурс] / URL: <http://jmeter.apache.org> (дата обращения: 07.11.2014).
5. The Grinder, a Java Load Testing Framework [электронный ресурс] / URL: <http://grinder.sourceforge.net> (дата обращения: 07.11.2014).
6. The httpperf HTTP load generator [электронный ресурс] / URL: <https://code.google.com/p/httpperf/> (дата обращения: 07.11.2014).
7. ab – Apache HTTP server benchmarking tool [электронный ресурс] / URL: <http://httpd.apache.org/docs/current/programs/ab.html> (дата обращения: 07.11.2014).
8. SoapUI. The Swiss-Army Knife of Testing [электронный ресурс] / URL: <http://www.soapui.org/About-SoapUI/what-is-soapui.html> (дата обращения: 10.11.2014).
9. [MS-MDM]: Mobile Device Management Protocol, 2014 [электронный ресурс] / URL: <http://msdn.microsoft.com/en-us/library/dn392112.aspx> (дата обращения: 10.11.2014).
10. Schuetz D. The iOS MDM Protocol. BlackHat USA 2011 [электронный ресурс] / URL: [https://media.blackhat.com/bh-us-11/Schuetz/BH\\_US\\_11\\_Schuetz\\_InsideAppleMDM\\_WP.pdf](https://media.blackhat.com/bh-us-11/Schuetz/BH_US_11_Schuetz_InsideAppleMDM_WP.pdf) (дата обращения: 10.11.2014).
11. Barth A. RFC 6265. HTTP State Management Mechanism. IETF, 2011 [электронный ресурс] / URL: <http://www.rfc-editor.org/rfc/rfc6265.txt> (дата обращения: 10.11.2014).
12. Utting M., Legeard B. Practical Model-Based Testing: A Tools Approach. Morgan Kaufmann, 2006.
13. BluePoint Enterprise Edition MDM Load Simulator [электронный ресурс] / URL: <http://www.bluepointsecurity.com/presentationlayer/pages/enterpriseeditionmdmloadtester.aspx> (дата обращения: 09.11.2014).

## REFERENCES

1. *Gartner IT-Glossary, 2013*. Available: <http://www.gartner.com/it-glossary> (Accessed 06.11.2014).
2. *Gartner. Gartner Predicts by 2017, Half of Employers will Require Employees to Supply Their Own Device for Work Purposes, 2013*. Available: <http://www.gartner.com/newsroom/id/2466615> (Accessed 07.11.2014).
3. *http\_load – multiprocessing http test client*. Available: [http://www.acme.com/software/http\\_load/](http://www.acme.com/software/http_load/) (Accessed 07.11.2014).
4. *Apache JMeter*. Available: <http://jmeter.apache.org> (Accessed 07.11.2014).
5. *The Grinder, a Java Load Testing Framework*. Available: <http://grinder.sourceforge.net> (Accessed 07.11.2014).
6. *The httpperf HTTP load generator*. Available: <https://code.google.com/p/httpperf/> (Accessed 07.11.2014).
7. *ab – Apache HTTP server benchmarking tool*. Available: <http://httpd.apache.org/docs/current/programs/ab.html> (Accessed 07.11.2014).
8. *SoapUI. The Swiss-Army Knife of Testing*. Available: <http://www.soapui.org/About-SoapUI/what-is-soapui.html> (Accessed 10.11.2014).
9. *[MS-MDM]: Mobile Device Management Protocol, 2014*. Available: <http://msdn.microsoft.com/en-us/library/dn392112.aspx> (Accessed 10.11.2014).
10. *Schuetz D. The iOS MDM Protocol. BlackHat USA 2011*. Available: [https://media.blackhat.com/bh-us-11/Schuetz/BH\\_US\\_11\\_Schuetz\\_InsideAppleMDM\\_WP.pdf](https://media.blackhat.com/bh-us-11/Schuetz/BH_US_11_Schuetz_InsideAppleMDM_WP.pdf) (Accessed 10.11.2014).
11. *Barth A. RFC 6265. HTTP State Management Mechanism. IETF, 2011*. Available: <http://www.rfc-editor.org/rfc/rfc6265.txt> (Accessed 10.11.2014).
12. *Utting M., Legeard B. Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann, 2006.
13. *BluePoint Enterprise Edition MDM Load Simulator*. Available: <http://www.bluepointsecurity.com/presentationlayer/pages/enterpriseeditionmdmloadtester.aspx> (Accessed 09.11.2014).

---

**САМОЧАДИН Александр Викторович** – профессор кафедры распределенных вычислений и компьютерных сетей Института информационных технологий и управления Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: [samochadin@gmail.com](mailto:samochadin@gmail.com)

**SAMOCHADIN, Alexander V.** *St. Petersburg Polytechnic University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: [samochadin@gmail.com](mailto:samochadin@gmail.com)

**СУЖАЕВ Олег Игоревич** – студент кафедры распределенных вычислений и компьютерных сетей Института информационных технологий и управления Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: [oleg.suzhaev@gmail.com](mailto:oleg.suzhaev@gmail.com)

**SUZHAEV, Oleg I.** *St. Petersburg Polytechnic University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: [oleg.suzhaev@gmail.com](mailto:oleg.suzhaev@gmail.com)

**ТИМОФЕЕВ Дмитрий Андреевич** – старший преподаватель кафедры распределенных вычислений и компьютерных сетей Института информационных технологий и управления Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: [dtim@dcn.icc.spbstu.ru](mailto:dtim@dcn.icc.spbstu.ru)

**TIMOFEEV, Dmitry A.** *St. Petersburg Polytechnic University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: [dtim@dcn.icc.spbstu.ru](mailto:dtim@dcn.icc.spbstu.ru)

**РОГОВ Петр Александрович** – заместитель начальника организационного отдела научной части  
*Санкт-Петербургского государственного политехнического университета.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: petr.rogov@spbstu.ru

**ROGOV, Petr A.** *St. Petersburg Polytechnic University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: petr.rogov@spbstu.ru