

ОБ ЭФФЕКТИВНОСТИ ВЕБ-ДОСТУПА К ПРИЛОЖЕНИЯМ С ПОМОЩЬЮ ТЕХНОЛОГИИ BROADWAY

В. Ю. Коваленко, Д. А. Костюк

Кафедра электронных вычислительных машин и систем,
Брестский государственный технический университет

Брест, Республика Беларусь

E-mail: vl.kovalenko1989@outlook.com

Рассмотрены практические особенности удалённого доступа с использованием браузера к стандартным настольным либо мобильным приложениям с помощью технологии broadway. Описывается принцип работы, оценена эффективность данного подхода в сравнении с типовым удалённым доступом по протоколу VNC. Описано практическое применение подхода в авторской разработке.

ВВЕДЕНИЕ

Использование веб-интерфесов в интранет-приложениях, решающих локальные задачи предприятия, за последнее время приобрело широчайшее распространение и стало общеупотребительным. Универсальная доступность везде (в том числе и на мобильных платформах) делает веб-браузер удобной точкой входа для пользователя приложения или сервиса, а решать проблемы производительности помогает то, что основная вычислительная нагрузка ложится на сервер и/или другие узлы сети, а ресурсы клиента тратятся на обеспечение взаимодействия с приложением, то есть на управление и доступ к ресурсам.

Одним из частных случаев такого удалённого интерфейса на основе веб-технологий является обеспечение доступа к программному обеспечению либо сервису, запущенному в контейнере виртуальной машины (ВМ) [1–2]. Технически ВМ запускается в фоновом (headless) режиме, обеспечивая взаимодействие с виртуализованной системой через клиент удалённого доступа по протоколу VNC, SPICE или RDP. Для доступа к ВМ из веб-среды можно использовать клиент удалённого доступа, написанный на языке JavaScript – например, noVNC [1]. В ряде случаев последнее решение безальтернативно, т. к. обеспечивает веб-доступ в том числе и к стандартным приложениям для настольных или мобильных компьютеров без создания для них специализированных веб-интерфейсов [3].

I. ВЕБ-ТЕХНОЛОГИИ В ИНТЕРФЕЙСЕ ТРАДИЦИОННЫХ ПРИЛОЖЕНИЙ

Одновременно наблюдается продвижение веб-технологий в классические интерфейсы с противоположной стороны: современные графические оболочки и библиотеки виджетов всё активнее применяют каскадные таблицы стилей и разметку на основе XML. Отдельно в этом ряду стоит библиотека GTK+ 3, у которой в версиях начиная с 3.8 присутствует полноценный HTML-бэкенд broadway, позволяющий организо-

вать веб-доступ к любому графическому приложению, написанному с ее использованием.

Broadway отрисовывает интерфейс GTK-приложений через HTML5. В его состав входит собственный небольшой сервер, написанный на C, служащий для отдачи HTML-страниц клиенту. Общая схема работы показана на рис. 1.

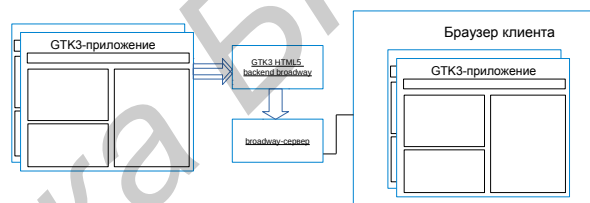


Рис. 1 – Схема работы broadway

Если операционная система – хост-система сервера или гостевая в контейнере ВМ – позволяет использовать библиотеку GTK нужной версии, появляется возможность заменить часть звеньев цепочки удалённого доступа более производительным кодом, интегрированным в средства организации интерфейса.

II. ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

Для оценки целесообразности такой замены нами выполнено сравнение на тестовом сервере с ОС GNU/Linux Ubuntu 15.04, процессором AMD FX-8230 и 16 GB ОЗУ. Трафик клиенту noVNC предоставлял VNC-сервер x11vnc. В качестве показателя загрузки сравнивался параметр load average, выдаваемый утилитой htop. Сценарий тестирования включал работу с документами, файловым менеджером и текстовым редактором. Результаты можно видеть на рис. 2.

То, что загруженность сервера с broadway ниже, чем с VNC, можно объяснить несколькими факторами: отказом от выделенного VNC-сервера, отсутствием трансляции протокола VNC в веб-представление и обратно, более оптимальным (за счет узкой специализации) сервером, отдающим веб-страницу. Также заметно, что выигрыш от использования broadway увеличивается при увеличении числа рабочих мест.

На приведенной диаграмме показана суммарная нагрузка клиента и сервера. Фактически и вычислительная нагрузка на клиент (потребление процессора веб-браузером) и потребление ресурсов сети у обоих подходов приблизительно одинаковы: оба параметра невелики и сравнимы по величине. Однако *broadway* имеет дополнительное преимущество – полное отсутствие артефактов изображения, периодическое появление которых типично для VNC. Также использование *broadway* приводит к уменьшенному потреблению ОЗУ клиентом: экономия порядка 30–40 МБ на одно виртуальное рабочее место.

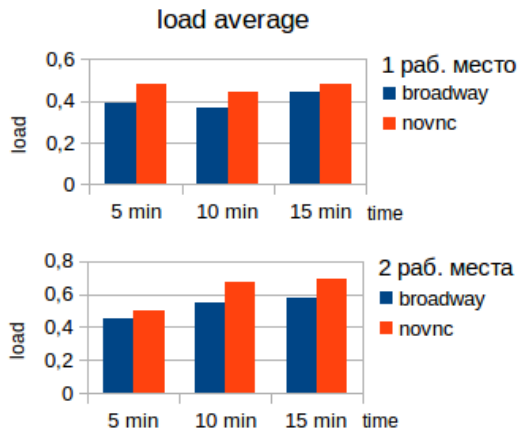


Рис. 2 – Работа *broadway*

III. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ

Описываемое решение было применено нами на практике в составе демонстрационно-тестовой фермы для разработки Android-приложений [3]. Поводом для ее создания послужил следующий комплекс причин. Сегодня платформа Android – одна из наиболее фрагментированных. Разработчику необходимо держать на своей машине набор образов для эмулятора с различными версиями и запускать их по очереди. При этом низкая производительность эмуляции, связанная с разницей процессорных архитектур, при множественном запуске эмулятора негативно сказывается на производительности и эффективности разработки. При коллективной же разработке требуется общий доступ к экземпляру тестового приложения, а также к демонстрационным промежуточным сборкам, из-за чего методы локальной эмуляции оказываются крайне неудобны. Решением перечисленных недостатков является развертывание тестовой фермы на сервере и доступ к ней по сети.

Разработанная архитектура фермы представлена на рисунке 3. Для удобства развертывания набор эмуляторов работает внутри контейнера VM на сервере локальной сети – VirtualBox либо VMWare, в зависимости от уже существующей инфраструктуры предприятия, в которую

должна быть интегрирована ферма. Число вложенных VM (эмуляторов) варьируется в процессе эксплуатации. В качестве вложенной VM вместо штатного эмулятора Android SDK используется QEMU или VirtualBox (в зависимости от внешней VM, с которой должна работать аппаратно-ускоренная вложенная виртуализация).

На стороне клиента доступ к эмуляторам (как на машинах разработчиков, так и менеджеров) осуществляется через веб-браузер, который взаимодействует с *broadway*. Промежуточный клиент на GTK3 запускается в headless-режиме, то есть только для сетевого доступа. Для этого в системе, по одному на клиента, запускаются демоны *broadwayd*, каждый из которых занимает отдельный сетевой порт. Эталонные образы виртуальных машин с Android-x86 различных версий сконфигурированы и настроены для максимально быстрого запуска. Для тестирования разрабатываемого ПО один из них клонируется. Развертывание и отладка программы в клонированной VM выполняется через ADB, а взаимодействие с ней – через веб-интерфейс. Через него же осуществляется загрузка тестируемого ПО, восстановление и управление образами.

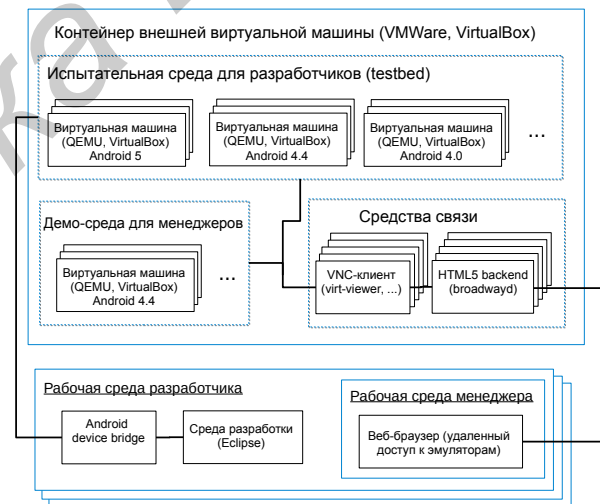


Рис. 3 – Архитектура фермы

1. Касцюк Д.А., Луцок П.А., Уласенка С.С., Жалудок В.А. Уживання віртуальних машин у складі ілюстрованих оглядач гісторії програмного забезпечення // Третя міжнародна науковопрактична конференція FOSS Lviv 2014: Збірник наукових праць / Львів, 24– 27 квітня 2014 р. – С. 51–54.
2. Коваленко В.Ю., Костюк Д.А., Кричок А.Г. Подход к автоматизации распределенного тестирования графических приложений // Информационные технологии и системы 2014 (ИТС 2014): материалы международной научной конференции. Минск, БГУИР, 29.10.2014. – С. 104–105.
3. Каваленко У.Ю., Касцюк Д.А. Демонстраційна-тестова ферма програм для платформи Android з веб-інтерфейсом // Третя міжнародна науковопрактична конференція FOSS Lviv 2015: Збірник наукових праць / Львів, 23– 26 квітня 2015 р. – С. 42–45.