

Дмитрий Костюк

Брест, Брестский государственный технический университет

Особенности использования виртуализованных окружений, внедренных в презентационные материалы

Аннотация

Рассматривается подход к использованию виртуализации для повышения наглядности и интерактивности учебных материалов за счет непосредственного встраивания окон виртуальных машин в слайды презентаций либо электронные учебные пособия.

Медийная насыщенность демонстрационных материалов и электронной документации обычно ограничивается копиями экранов и, возможно, вставками анимационных фрагментов. «Живая» демонстрация обеспечивается частым переключением между окном, отображающим слайды или страницы документа, и окнами демонстрируемых программ. При этом для более простого развертывания демонстрируемое ПО может помещаться в контейнер виртуальной машины (ВМ). Импортируемое виртуализованное окружение может содержать любую готовую конфигурацию системного и прикладного ПО, а механизм снимков (snapshots) позволяет быстро выполнить откат ВМ к нужному моменту работы для повторной демонстрации ключевых элементов либо пропуска длительных процедур.

Однако установленное ПО, в отличие от копий экрана, не составляет единое целое с сопроводительными и поясняющими материалами. Частое переключение между окнами ВМ и презентационной программы ухудшает непрерывность восприятия информации во время лекции, а если ВМ является приложением к электронной документации или учебному пособию, лишние действия приходится выполнять конечному пользователю.

В плане вычислительной мощности современных ноутбуков и десктопов замена копий экрана программы «живым» выводом изображения ВМ, интегрированным непосредственно в поясняющие материалы — вполне осуществимая задача. Современные ВМ способны работать в «невидимом» (headless) режиме, предоставляя вместо графического окна программу-сервер для удаленного доступа (например, по

протоколу VNC). Клиентское же ПО является сравнительно несложным, а некоторые клиенты написаны на скриптовых языках либо в виде легко встраиваемых апплетов.

Задача создания подобных интегрированных виртуальных окружений решалась нами в рамках подготовки учебно-демонстрационных материалов по истории графического интерфейса. Изначально созданное решение было использовано в качестве лекционного материала, а затем адаптировано к роли действующей электронной экспозиции (выбранная тематическая направленность делает интерактивность демонстрируемого объекта особенно выигрышной).

Анализ исходной задачи показал, что возможность внедрить вывод окна внешнего приложения в документ на платформе GNU/Linux в свободных пакетах для подготовки и показа презентаций недоступна без создания модулей расширения. В результате было принято решение в пользу презентационных фреймворков на базе HTML5. Рост популярности этого формата для показа презентаций, наблюдающийся в последние годы, вызван богатыми возможностями языка для создания визуальных эффектов и простотой редактирования контента, при использовании готового фреймворка, отвечающего за оформление, показ и смену слайдов. На текущий момент в свободном доступе находится более десяти готовых решений, различающихся богатством визуальных эффектов и стилей оформления, а также возможностями сложной компоновки материала.

В результате были выбраны следующие компоненты:

- виртуальные машины KVM либо QEMU с аппаратной поддержкой виртуализации;
- VNC-клиент noVNC, написанный на javascript и HTML5 (<http://kanaka.github.com/noVNC/>);
- фреймворк reveal.js для показа слайдов (<http://github.com/hakimel/reveal.js>).

Как и другие подобные фреймворки, reveal.js обладает обширным функционалом, включая показ встроенных слайдов, упрощенную разметку содержимого, возможность экспорта в PDF. Для более эффектного отображения требуется браузер с поддержкой трехмерных CSS-преобразований, однако предусмотрен и упрощенный режим показа. Подготовка слайдов выполняется на HTML, в разметке markdown, либо в визуальном онлайн-редакторе, доступном по адресу <http://www.rvl.io>.

Возможность включения HTML-фреймов в содержимое слайдов была использована для вставки страниц, отображающих на элементе canvas передаваемое ВМ изображение. KVM и QEMU не поддерживают веб-сокеты, поэтому трафик передавался VNC-клиенту через прокси websockify, изначально разработанный в рамках проекта noVNC для работы с серверами, поддерживающими лишь TCP-соединение.

Выбор средств виртуализации продиктован желанием использовать полностью свободный набор ПО, возможностью эффективной работы большого числа ВМ, а также широким спектром платформ, поддерживаемых проектом QEMU. Эксперименты показали, что достаточно производительность среднего процессора для ноутбуков либо десктопов, имеющего аппаратную поддержку виртуализации (отчасти низкие требования вызваны тем, что в каждый момент предполагается активное использование только виртуальной машины, отображаемой на текущем слайде). Требуемый объем ОЗУ более критичен и определяется нуждами конкретных гостевых ОС. Однако при большом числе однотипных гостевых систем он может быть уменьшен использованием KVM в связке с технологией Kernel Samepage Merging (KSM), позволяющей объединять одинаковые страницы памяти для различных приложений (одно из типовых применений — как раз сервера виртуальных машин). KSM существенно снижает потребление памяти на системах, на которых затруднительным является одновременный запуск всех ВМ, необходимых для демонстрации, однако требует их предварительного запуска, и возможно, не в один этап (т. к. дедупликация страниц выполняется в фоновом режиме). Таким образом, этот способ снижения системных требований более удобен для постоянно действующей экспозиции, чем для лекционных материалов.

Запуск всей системы, включая нужные ВМ, прокси и браузер, выполняется с помощью тривиального управляющего скрипта (рис. 1).

В целом, получаемое презентационное окружение, несмотря на определенную ресурсоемкость, оказывается вполне работоспособным на типичном оборудовании, а также обладает рядом преимуществ: добавляет в материалы необходимую интерактивность и при этом не требует от пользователя или лектора переключения программ, фрагментирующего процесс преподнесения и усвоения материала.

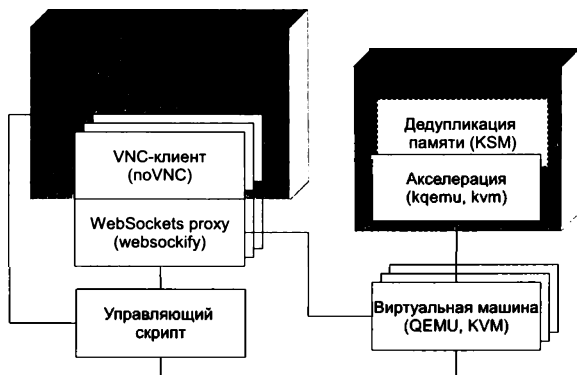


Рис. 1: Структура системы