

*А.М. Бабичев**

РАЗРАБОТКА СИСТЕМЫ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ СЕТЕВЫХ КЛИЕНТОВ НА ОСНОВЕ WEB-ТЕХНОЛОГИЙ

В настоящее время получили большое распространение различные портативные устройства, такие как ноутбуки, нетбуки, интернет-планшеты и смартфоны. Они обеспечивают постоянное присутствие владельца в сети и помогают ему работать, даже находясь вдалеке от своего рабочего места. Однако у них имеются свои недостатки, как например низкая производительность по сравнению со стационарными рабочими станциями, различные программные платформы, не всегда совместимые друг с другом и т.д. Для устранения этих недостатков можно использовать так называемые «облачные технологии».

Развитие мобильной техники и облачных технологий способствует использованию их для контроля за рабочими процессами вне рабочей

* Работа выполнена под руководством д-ра техн. наук, проф., проректора по информатизации ФГБОУ ВПО «ТГТУ» В.А. Подольского, канд. пед. наук, доцента Н.А. Иньковой.

зоны. На зарубежном и отечественном рынках существуют программные продукты, осуществляющие удаленное управление компьютерными системами, однако эти программы не предоставляют достаточной гибкости в эксплуатации, так как требуют обязательной установки на компьютер администратора, а также часто адаптированы только для одной платформы и не имеют возможности запуска с мобильных устройств.

Система удаленного администрирования сетевых клиентов на основе WEB-технологий разрабатывается с прицелом на использование на различных мобильных устройствах, таких как планшетные компьютеры и смартфоны на iOS и Android, ноутбуки и нетбуки под управлением Windows, Linux или MacOS. Большинство существующих систем не обеспечивают достаточной гибкости в использовании, так как требуют обязательной установки дополнительных компонентов [1]. Такую гибкость могут обеспечить облачные технологии, доступные из веб-браузера, который обязательно прилагается в базовом наборе любой современной операционной системы.

Конечно, существуют системы, имеющие возможность удаленного управления через браузер. Однако эти системы используют технологию Adobe Flash для вывода данных в браузер. Исследования показывают, что данная технология имеет ряд недостатков, таких как крайне высокое энергопотребление при ее использовании, а также низкопроизводительная или отсутствующая реализация данной технологии на некоторых операционных системах, таких как Apple iOS. Поэтому было решено выбрать для разработки администраторской части язык JavaScript в связке с серверным языком PHP, связь между которыми осуществляется с использованием технологии Ajax.

Для начала работы в системе пользователь должен запустить на своем компьютере клиентскую часть системы. При запуске клиентская часть системы генерирует 8-значный идентификатор, состоящий из случайных чисел, и такой же пароль. Они шифруются и передаются в подсистему авторизации и записываются в базу данных для последующей проверки. Затем их следует передать администратору, который будет осуществлять управление компьютером и должен ввести этот идентификатор и пароль в веб-интерфейс подсистемы авторизации.

Если идентификатор и пароль совпадают, то происходит переход на веб-страницу, представляющую собой подсистему визуализации и выводящую в браузер изображение с рабочего стола клиента. Подсистема визуализации написана с использованием связки языков JavaScript и PHP. Необходимость такой связки обуславливается тем, что в языке JavaScript отсутствует прямая реализация работы с файловой системой, поэтому при попытке вывода файла изображения в веб-интерфейс этот файл оказывается заблокированным для перезаписи новыми изображениями, поступающими из подсистемы снятия изображений. Для решения этой проблемы используется специальный PHP-скрипт.

При запросе пользователя web-сервер просматривает документ, выполняет найденные в нем PHP-инструкции, а результат их выполнения возвращает пользователю. Сам скрипт находится на сервере и его содержимое просмотреть невозможно. При активации скрипта серверная программа выполняет все команды этого скрипта, не затрагивая статическую часть файла, и результат возвращается программатраузеру. Таким образом, PHP-скрипт считывает данные из файла изображения и формирует изображение, готовое к выводу в браузер, оставляя оригинальный файл открытым для записи.

Одновременно с этим начинает работать подсистема снятия изображения, которая снимает изображение с рабочего стола и передает на сервер, после чего оно выводится в подсистеме визуализации.

Для передачи команд от администратора клиенту используется подсистема ввода-вывода. Администратор через веб-интерфейс передает клиенту специально записанные и зашифрованные команды, которые затем интерпретируются и эмулируются на клиентской машине. Подсистема ввода-вывода через короткие промежутки времени считывает данные из базы данных, в которую записываются данные о командах мыши и клавиатуры, подаваемых через веб-интерфейс с помощью технологии Ajax.

Наибольшая часть передаваемых данных представляет из себя графические файлы, содержащие изображение с экрана сетевого клиента. В качестве таких файлов выступают файлы, перекодированные в формат GIF. Этот формат был выбран исходя из нескольких факторов. Современные веб-браузеры поддерживают только определенное количество форматов, основными из которых являются BMP, GIF, JPG и PNG. Каждый из этих форматов имеет недостатки и преимущества, так BMP передает изображение без потери качества и не требует перекодировки, однако исходные файлы имеют большой размер и долго передаются по сети; PNG также передает изображение без искажений и занимает значительно меньше места чем BMP, но кодирование занимает значительное время; JPG передает изображение с искажениями и также требует перекодировки, при этом размер файла получается сравним с размером PNG-файла. GIF был выбран, потому что передает изображение с минимальными искажениями, затрагивающими только цветовую палитру, но при этом кодировка занимает меньше времени, чем у форматов PNG и JPG, а размер файла GIF приблизительно равен размеру файлов этих форматов. В графическом формате GIF используется алгоритм сжатия Лемпеля-Зива-Велча (Lempel-Ziv-Welch, LZW) [2].

Главная задача состоит в определении оптимального размера исходящих от клиента данных, таких как графическое отображение удаленного рабочего стола. Размер графических данных напрямую влияет на время отклика клиента и на быстродействие системы в целом. Время отклика вычисляется по формуле:

$$T = X \times Y \times Q / S ,$$

где X и Y – размеры изображения по ширине и высоте, Q – качество изображения, S – скорость передачи данных в сети. Сочетание размеров изображения и качества дает размер изображения в байтах, соответственно соотношение размера (байт) и скорости (байт/с) даст искомое время отклика. Оптимальное значение T , округленное до целых, должно составлять 12 миллисекунд.

Для ускорения передачи данных исходное изображение с рабочего стола ужимается до размеров окна браузера принимающей стороны и разбивается на несколько небольших частей, размер и количество которых выбирается в зависимости от разрешения исходного изображения и скорости сети. После этого производится процедура вычисления приоритета пересылки данных частей. В первую очередь посылается та часть, в которой в данный момент находится курсор мыши. После этого идет сравнение частей на изменение с предыдущими переданными частями. Для этого части ужимаются до 10 % от собственного размера и производится сравнение их по уровням яркости, так как малейшее изменение части изображения будет заметно даже на уменьшенной версии этой части и отразится на ее общем уровне яркости. После того, как будут переданы измененные части изображения и часть изображения с находящимся на ней курсором мыши, передаются все остальные части.

Разрабатываемая система позволяет управляющему персоналу находиться вне территории управляемого объекта и дает возможность работать с этими объектами людям с ограниченными возможностями, а также всем тем, кто по каким-либо причинам не может присутствовать на данных объектах. Система позволит управлять одновременно несколькими сетевыми клиентами в рамках решения одних и тех же задач, что значительно увеличит производительность труда, даст возможность значительного увеличения заработной платы диспетчера и экономии денежных средств.

СПИСОК ЛИТЕРАТУРЫ

1. Подольский, В.Е. Повышение эффективности региональных образовательных компьютерных сетей с использованием элементов структурного анализа и теории сложности : монография / В.Е. Подольский, С.С. Толстых. – М. : Машиностроение, 2006. – 175 с.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005, 2006.