

ВЕБ-ПЛАТФОРМА ДЛЯ ИГРОВЫХ И ПРИКЛАДНЫХ ПРИЛОЖЕНИЙ

П.Д. Ключов, А.Е. Телепнев
Научный руководитель: И.А. Ботыгин
Томский политехнический университет,
E-mail: pdk3@tpu.ru, aet12@tpu.ru

Введение

На сегодняшний день, существует множество онлайн-сервисов для распространения компьютерных игр и программ [1]. Данные сервисы требуют обязательной установки приложения на пользовательское устройство, помогая лишь в выборе приложений. Также, следует упомянуть о существовании облачных веб-сервисов, которые инициализируют запуск игр и приложений на своем облачном сервисе, предоставляя пользователю лишь отображение используемой программы. Исходя из сказанного выше, актуальной задачей является создание веб-платформы, которая исключит установку приложения и позволит распределить нагрузку между серверной и клиентской частями. В данной работе речь пойдет о реализации веб-платформы игровых и прикладных приложений.

Архитектура веб-платформы игровых и прикладных приложений

Построение архитектуры веб-платформы начинается с детерминирования составляющих частей платформы. Для данной платформы были определены следующие части (рисунок 1):

- Серверная часть сайта: отвечает за перенаправление на запрашиваемые пользователем веб-страницы, взаимодействие с игровыми серверами, распределение нагрузки между логическими серверами;
- Клиентская часть сайта: обрабатывает действия пользователя и визуализирует результаты запросов к серверной части сайта и игровым серверам;
- Логические сервера: отвечают за обработку и синхронизацию данных в соответствии с логикой используемого приложения;
- Клиентская нативная часть приложения: обрабатывает пользовательские данные, взаимодействуя с конкретным логическим сервером.

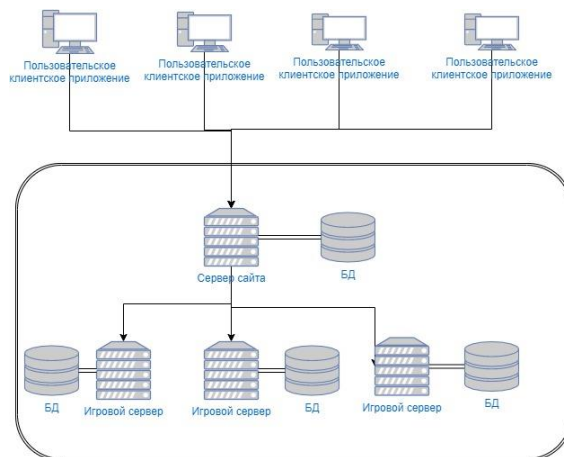


Рис. 1. Архитектура разрабатываемого проекта

Этапы реализации веб-платформы

Для успешной реализации данного проекта были выделены следующие этапы:

1. Реализация нативного игрового приложения;
2. Распределение функционала разработанного приложения между клиентской и игровой серверной частями;
3. Реализация веб-платформы и API взаимодействия серверной части платформы с логическими серверами;
4. Дополнение коллекции игровых и прикладных приложений.

Результатом реализации первого этапа будет являться спроектированная основная часть будущего логического сервера и клиентского нативного приложения.

В результате выполнения второго этапа функционал разработанного нативного приложения распределяется между клиентской и серверной составляющей, а также реализуется взаимодействие между ними.

При выполнении третьего этапа, все части разрабатываемой веб-платформы составляют функционирующую распределенно-вычислительную систему.

Выбор средств разработки и формата передачи данных

Логический сервер выполняет основную вычислительную нагрузку, вследствие чего быстрдействие сервера играет ключевую роль. Язык программирования C++ позволяет выполнять низкоуровневые операции, что положительно

сказывается на скорости выполнении программного кода [2].

Среда разработки Qt Creator позволяет вести разработку на языке программирования C++. Его особенностью является слот-сигнальная система, которая позволила спроектировать архитектуру приложения на первом этапе в соответствии с архитектурой клиент-серверного приложения, исключая лишние зависимости модулей проекта [3].

Клиентская часть сайта взаимодействует в одних случаях с логическим сервером через серверную часть сайта, а в других – напрямую. Все основные ресурсозатратные математические расчеты происходят на логическом сервере, а результаты рендеринга отображаются конкретно на странице пользователя. Программная библиотека React JS, позволяет выполнять рендеринг страницы и игровой карты, как на серверной части, так и на клиентской [4]. Помимо данной возможности создания изоморфного приложения данная библиотека позволяет ускорить разработку сайта, предоставляя разработчику большой функционал.

Для передачи данных между модулями распределенной вычислительной системы был выбран формат JSON. Данный формат позволяет хранить данные в строковой форме и удобно обрабатывается в разных языках программирования [5].

Результаты выполнения первого и второго этапов реализации веб-платформы

В качестве первого нативного приложения было реализовано игровое приложение «Танчики». Далее, игра была перенесена на клиент-серверное приложение, структура которого включает в себя следующие части (рисунок 2):

- Пользовательское клиентское приложение: обрабатывает пользовательские действия и выполняет рендеринг игровой карты;
- Игровой сервер (логический сервер): выполняет основные математические вычисления в соответствии с игровой логикой приложения;
- Клиентское приложение «Система ботов»: инициализирует ботов, выполняет анализ игровой карты для дальнейшего управления ботами.

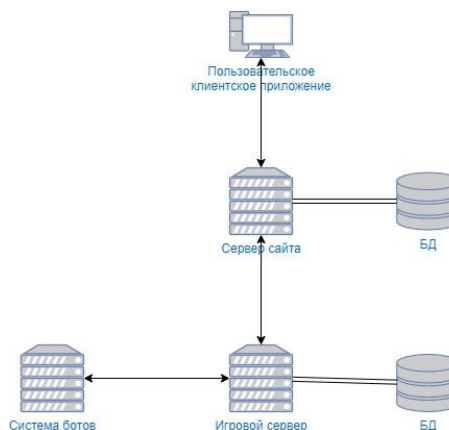


Рис. 2. Структура клиент-серверного приложения «Танчики»

Результатом распределения нагрузки сервера является выделение системы ботов как отдельное клиентское приложение. Таким образом, система ботов взаимодействует с игровым сервером, как пользователь, передавая информацию о действии ботов, на основе проанализированной игровой ситуации.

Заключение

Итогом выполненной работы является спроектированная архитектура будущей веб-платформы и игровое клиент-серверное приложение «Танчики». Приложение реализовано на основе архитектуры распределенной вычислительной системы.

В дальнейшем планируется запуск веб-платформы, модификация клиент-серверного приложения «Танчики», дополнение коллекции игровых и прикладных программ веб-платформы.

Список использованных источников

1. Startpack. Поиск и подбор облачных сервисов. [Электронный ресурс]. – URL: <https://startpack.ru/> (дата обращения 14.01.2020).
2. Progopedia. C++. [Электронный ресурс]. – URL: <http://progopedia.ru/language/c-plus-plus/> (дата обращения 11.09.2019).
3. Qt. Qt Creator. [Электронный ресурс]. – URL: https://www.qt.io/product?utm_campaign=Navigation%202019&utm_source=Nav%202019 (дата обращения 12.09.2019).
4. ReactJS. React. [Электронный ресурс]. – URL: <https://ru.reactjs.org/> (дата обращения 13.09.2019).
5. Json. Введение в JSON. [Электронный ресурс]. – URL: <https://www.json.org/json-ru.html> (дата обращения 15.09.2019).