РАЗРАБОТКА МАСШТАБИРУЕМОЙ ВЫСОКОНАГРУЖЕННОЙ СИСТЕМЫ АДАПТИВНОЙ ПОТОКОВОЙ ПЕРЕДАЧИ МЕДИАДАННЫХ

П. Д. Гордей, А. А. Петров

fpm.petrovAA@bsu.by;
Научный руководитель — Е. А. Левчук, кандидат технических наук, доцент;
Н. А. Карпович, старший преподаватель

Статья посвящена проектированию и разработке высоконагруженной и масштабируемой системы с использованием вычислительных ресурсов облачной платформы AWS, способной по требованию передавать преобразованный по протоколу HLS медиапоток и воспроизводить его с помощью разработанного кросс-браузерного медиапроигрывателя.

Ключевые слова: модель клиент-сервер; микросервисная архитектура; облачная платформа; потоковая передача данных; адаптивный битрейт; Media Source Extension (MSE); HTML5 медиапроигрыватель.

ВВЕДЕНИЕ

В основе всех современных мультимедийных сервисов лежит принцип потоковой передачи данных от сервера на клиент. Простейшая система потоковой передачи состоит из сервера потокового медиа и клиента, в роли которого как правило выступает медиапроигрыватель.

Потоковое вещание начинается с кодирования медиаданных, которое представляет собой процесс сжатия медиафайла, что позволяет уменьшить размер и оптимизировать его передачу по сети. Для работы системы адаптивной потоковой передачи необходимо предварительно закодировать медиафайлы с различными настройками скорости передачи потока данных по каналу (битрейт) (200, 500 и 1000 кбит/с) и сохранить на сервере потоковой передачи или в системе доставки контента.

Чтобы получить информацию о различных версиях медиафайлов и соответствующих им битрейтах для управления процессом загрузки и воспроизведения используется специальный манифест файл. С помощью него современные протоколы потоковой передачи могут регулировать качество передаваемых медиаданных в зависимости от пропускной способности сети, тем самым реализуя адаптивность потоковой передачи.

Популярные мультимедийные платформы, использующие потоковую передачу медиаданных, обслуживают десятки тысяч активных пользователей, что делает актуальным вопрос масштабируемости системы. В статье рассматриваются основные принципы и технологии,

используемые при проектировании таких систем и медиапроигрывателей для них.

СЕРВЕРНАЯ ЧАСТЬ И ИНФРАСТРУКТУРА

Вся система спроектирована и разработана согласно архитектурному стилю REST, который определяет набор принципов и ограничений, использующихся для взаимодействия отдельных компонентов распределенной системы. При таком подходе основной концепцией является модель клиент-сервер.

Серверная часть в свою очередь спроектирована с использованием микросервисной архитектуры на основе шаблона API-шлюз, где ключевыми службами являются сервисы данных и авторизации. В качестве шлюза выступает ресурс облачной платформы Amazon – Application Load Balancer, обладающий механизмом маршрутизации запросов между микросервисами, а также функцией равномерного распределения нагрузки между вычислительными серверами AWS EC2 внутри каждого из сервисов. Машины каждого микросервиса располагаются в собственной группе автоматического масштабирования (Amazon Autoscaling group), что позволяет увеличивать либо уменьшать количество используемых серверов в зависимости от текущей нагрузки. При этом процессы масштабирования двух микросервисов являются полностью независимыми друг от друга, гарантируя вместе с этим оптимальное использование ресурсов. Использование отдельных баз данных для каждого сервиса делает их самостоятельными, что увеличивает гибкость и отказоустойчивость всей системы.

Для преобразования исходного видеофайла в формат, который поддерживает потоковую передачу данных, использовался сервис Amazon MediaConvert. При загрузке файла в хранилище медиаданных (Amazon S3) срабатывает внутренний механизм событий, что позволяет передать метаданные события создания объекта для формирования нужной конфигурации конвертации в управляемый сервис Amazon Lambda. Уже преобразованный видеопоток передается на сторону клиента при помощи сети доставки контента (Amazon CloudFront), основной функцией которого является кеширование данных на промежуточных узлах глобальной инфраструктуры AWS, что позволяет уменьшить время отклика и увеличивает производительность всей системы в целом.

КРОСС-БРАУЗЕРНЫЙ МЕДИАПРОИГРЫВАТЕЛЬ

Для воспроизведения потокового видео в веб-браузере обычно требуется медиапроигрыватель, который должен быть совместим с используемой потоковой технологией.

Архитектура разработанного медиапроигрывателя основана на принципе модульного подхода к проектированию программных систем. Важно отметить, что данных подход применяется не только к логике проигрывателя, но и к слою представления, что обеспечивает независимость компонентов системы, иерархическое использование структур и разделение пользовательского интерфейса на отдельные компоненты представления.

Для реализации поддержки адаптивного медиапотока с использованием различных технологий воспроизведения уровень логики проигрывателя и уровень технологии обработки и воспроизведения медиапотока должны быть независимыми и зависеть от абстракций, предоставляемых программными интерфейсами. Это достигается путем размещения реализации технологии воспроизведения в отдельном модуле, используя систему внешних расширений. Такой подход позволяет управляющему уровню проигрывателя и уровню технологии воспроизведения не быть связанными друг с другом. Вместо этого они зависят от абстракций, определяющих общие интерфейсы и ожидаемое поведение. Такая архитектура облегчает замену или добавление новых технологий воспроизведения без внесения изменений в логику проигрывателя и позволяет выбирать оптимальную технологию воспроизведения в зависимости от контента и факторов.

В контексте разработки медиапроигрывателя возникает проблема обеспечения гибкости и важности изменения как цветовой схемы и базовых стилей медиапроигрывателя, так и его разметки, а также замены или добавления новых элементов управления. Одним из решений является создание элементов управления проигрывателя в виде отдельных заменяемых компонентов пользовательского интерфейса, которые передаются базовому компоненту проигрывателя – макету. Макет является корневым элементом пользовательского интерфейса и содержит базовую разметку, нативный HTML медиа-элемент проигрывателя и блочный элемент, который будет служить сеткой для дочерних компонентов.

Сетка определяется с помощью CSS свойства «display: grid», что позволяет гибко управлять разметкой проигрывателя, изменяя только некоторые CSS свойства, такие как «grid-template-area», «grid-template-columns», «grid-template-rows» и «gap». Для задания именной области в сетке элементу управления нужно указать в корневом элементе CSS свойство «grid-area» со значением, соответствующим имени элемента управления. Таким образом, путем определения шаблона сетки можно изменять разметку без необходимости переписывания CSS стилей, их

полной замены или создания пользовательского интерфейса проигрывателя с нуля.

ЗАКЛЮЧЕНИЕ

По итогам выполнения исследования был достигнут приемлемый уровень масштабируемости системы, позволяющий развернуть необходимое количество дополнительных вычислительных серверов в течение пяти минут с момента, как уровень нагрузки на центральное процессорное устройство превысил 50%. Также использование сети доставки контента позволило уменьшить среднее время отдачи видеопотока с нескольких секунд до сотых долей секунды.

Разработанный медиапроигрыватель представляет собой ряд пакетов, содержащих EcmaScript модули, которые были опубликованы в пакетном менеджере npm, входящего в состав Node.js. Исходя из проведенного анализа существующих решений и разработанных пакетов, можем сделать вывод, что в сравнении с существующими решениями, разработанные пакеты являются более легковесными и позволяют модифицировать не только стилизацию, но и разметку пользовательского интерфейса проигрывателя под нужды разрабатываемого сервиса. Однако стоит понимать, что разработанный проигрыватель обладает лишь базовыми функциональными возможностями и не имеет такого количества функций и плагинов, как у существующих решений. Несмотря на это, наличие удобного программного интерфейса для расширения функциональных возможностей позволяет проигрывателю стать хорошим решением, учитывая требования проекта к набору функциональности.

Таким образом, представленные подходы к разработке мультимедийных сервисов и медиапроигрывателей для них являются универсальными и могут быть использованы при разработке любых других систем для достижения подобного результата.

Библиографические ссылки

1. *Burnett, Steve M.* AWS for Beginners: The Ultimate Guide to the Fundamentals of AWS / *Steve M.* Burnett — Independently published, 2021. 54 p.
2. *Wolenetz, M.* Media Source Extensions / M. Wolenetz, M. Watson // W3C [Electronic resource]. 2022. Mode of access: https://www.w3.org/TR/media-source-2. Date of access: 30.05.2023.