

## Исследование бессерверной инфраструктуры Yandex.Cloud для построения событийно-ориентированных приложений

Назаренко В. В., Морщанина Л.В.

(Университет ИТМО)

Научный руководитель – кандидат педагогических наук Государев И.Б.

(Университет ИТМО)

**Введение.** Изначально веб-приложения предполагали использование простой клиент-серверной модели, где клиент посылает HTTP-запрос серверу, а сервер подготавливает ответ на запрос и отправляет его клиенту. Но данная модель предполагает, что клиентским запросом всегда следует ответ сервера, и это не работает в обратную сторону. Со временем требования к веб-приложениям начали возрастать, и, как следствие, потребовалась возможность обновлять данные в режиме реального времени.

**Основная часть.** Для решения задач обновления данных в режиме реального времени в 2011 году был разработан WebSocket [1]. WebSocket — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером, используя постоянное соединение [2]. В отличие от HTTP, WebSocket поддерживает постоянное и двунаправленное соединение между клиентом и сервером, что позволяет существенно уменьшать задержки и создавать более быстрые приложения.

В качестве преимуществ данного протокола следует отметить следующее:

1. Двунаправленная связь сервера и клиента. Протокол поддерживает отправку сообщений как от клиента серверу, так и от сервера клиенту
2. Мгновенные обновления. WebSocket позволяет обмениваться данными без обновления страницы и отправки повторных запросов.
3. Открытие соединения один раз. После открытия соединения оно используется в обе стороны, без необходимости повторного открытия при каждом запросе. У такого соединения скорость выше, чем у HTTP [3]. WebSocket не только снижает количество запросов к серверу, но и позволяет клиенту получать события обновления от сервера асинхронно при открытом соединении.

Тем не менее, свои ограничения в работу с WebSocket-соединениями вносит природа бессерверной инфраструктуры, основанная на событийно-ориентированном подходе. Данный подход не позволяет поддерживать WebSocket-соединения постоянно открытыми, так как бессерверные функции стартуют только при обращении к ним и имеют ограниченное время жизни – timeout. Несмотря на данные ограничения, существуют реализации, которые позволяют эффективно интегрировать WebSocket-соединения в бессерверную инфраструктуру.

В данном исследовании рассматривается практическая реализация асинхронного взаимодействия между клиентом и множеством бессерверных функций на платформе облачных вычислений Yandex.Cloud. Под множеством бессерверных функций понимается совокупность бессерверных функций, обеспечивающих выполнение операций с WebSocket соединением и логику приложения. Логика взаимодействия с клиентом выстроена согласно концепции публикации/подписки [4], где клиент подписывается на получение определенных событий в соединении, а бессерверные функции публикуют в соединение только те события, на которые клиент подписался.

В рамках исследования проектируется бессерверное веб-приложение, позволяющее обмениваться данными с клиентом в режиме реального времени. Данное приложение реализовано с помощью шлюза в Yandex.Cloud [5] и бессерверных функций. При открытии соединения по протоколу WebSocket через Api Gateway, бессерверная функция регистрирует

соединение в базе данных, записывая время открытия, статус соединения и уникальный идентификатор и другую метаданную. Через открытое соединение клиенты подписываются на обновления выбранных событий, отправляя через шлюз список событий, при изменении которых они хотят получать обновления. Используя множество бессерверных функций и взаимодействия между ними через очереди сообщений, приложение реализует механизм отправки сообщений клиентам в открытые соединения согласно зарегистрированным подпискам при получении сообщений от внешнего источника.

Производительность системы измеряется метриками, согласно которым можно оценить время обработки полученного сообщения и доставки его в открытые WebSocket-соединения.

**Выводы.** Таким образом, в результате исследования была спроектирована и реализована система асинхронного взаимодействия между клиентом и множеством бессерверных функций на платформе бессерверных вычислений Yandex.Cloud. Функциональность системы позволяет клиентам осуществлять подписку на обновления конкретных событий, а бессерверным функциям публиковать такие обновления в открытые с клиентами WebSocket-соединения. В результате исследования собраны метрики скорости доставки сообщения подписчикам при различном количестве одновременного открытых WebSocket-соединений и поступающих сообщений. Данная метрика позволит оценить скорость доставки обновлений клиентам через разработанную систему при разной нагрузке. Для оценки подготовлены данные 10 измерений, средняя скорость обработки и доставки одного сообщения составляет 2,15 с на сообщение. Однако следует отметить, что данная метрика изменяется в зависимости от общего количества поступающих сообщений.

#### **Список использованных источников:**

1. WebSockets for fun and profit [Электронный ресурс]. – 2019. – URL: <https://stackoverflow.blog/2019/12/18/websockets-for-fun-and-profit/> (дата обращения 20.01.2024).
2. WebSocket [Электронный ресурс]. – 2022. – URL: <https://ru.wikipedia.org/wiki/WebSocket> (дата обращения 20.01.2024).
3. WebSocket: что это, когда следует использовать и какие преимущества дает [Электронный ресурс]. – 2021. – URL: <https://cloud.vk.com/blog/websocket-kogda-sleduet-ispolzovat-i-preimushhestva> (дата обращения 25.01.2024).
4. Что такое отправка сообщений по модели «издатель – подписчик» (Pub/Sub)? [Электронный ресурс]. – 2023. – URL: <https://aws.amazon.com/ru/what-is/pub-sub-messaging> (дата обращения 25.01.2024).
5. Работа с API-шлюзом по протоколу WebSocket [Электронный ресурс]. – 2024. – URL: <https://cloud.yandex.ru/ru/docs/api-gateway/tutorials/api-gw-websocket> (дата обращения 28.01.2024).

Назаренко В.В. (автор)

Подпись

Морщинина Л.В. (автор)

Подпись

Государев И.Б (научный руководитель)

Подпись