



Столыпинский
вестник

Научная статья

Original article

УДК 004.057.4

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕАЛИЗАЦИЙ СЕТЕВОГО СТЕКА ДЛЯ
СЕРВЕРА ОБРАБОТКИ ВИДЕОПОТОКА**

**COMPARATIVE ANALYSIS OF NETWORK STACK IMPLEMENTATIONS FOR
VIDEO PROCESSING SERVER**

Сысойкин Егор Михайлович, студент 2-ого курса магистратуры кафедры ИУ5
ФГБОУ ВО «Московский государственный технический университет имени Н. Э.
Баумана (национальный исследовательский университет)» г. Москва,
egor.sysoykin@mail.ru

Шпак Игорь Денисович, студент 2-ого курса магистратуры кафедры ИУ5 ФГБОУ
ВО «Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)» г. Москва, shid18u875
@student.bmstu.ru

Горячкин Борис Сергеевич, кандидат технических наук, доцент, ФГБОУ ВО
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)» г. Москва, bsgor@mail.ru

Sysoykin E. Mikhailovich, a second-year graduate student at the Department of IU5,
Bauman Moscow State Technical University (National Research University), Moscow,
egor.sysoykin@mail.ru.

Shpak I. Denisovich, a second-year graduate student at the Department of IU5, Bauman Moscow State Technical University (National Research University), Moscow, shid18u875@student.bmstu.ru.

Goryachkin B. Sergeyeovich, Ph.D., Associate Professor, Bauman Moscow State Technical University (National Research University), Moscow, bsgor@mail.ru

Аннотация. Статья посвящена сравнительному исследованию различных подходов к реализации клиент-серверной архитектуры для систем автоматического распознавания объектов на изображениях и видео. В статье описан пример существующей системы, основанной на классической архитектуре, где сервер выполняет обработку изображений с использованием специализированных алгоритмов и нейронных сетей, а клиент отправляет данные на обработку и получает результаты распознавания. Основная проблема, выявленная в предыдущих исследованиях, заключается в низкой производительности системы при обработке видеопотоков в реальном времени, обусловленной ограничениями существующих методов взаимодействия между клиентом и сервером. В этой статье проведено сравнение различных методов взаимодействия с целью определения наиболее эффективного варианта для обработки видеопотоков с высокой скоростью и эффективностью. Основной метрикой оценки является количество кадров, обрабатываемых в единицу времени. В результате анализа было выяснено преимущество использования протокола WebSocket, который позволяет передавать каждый новый кадр в рамках одного соединения.

Abstract. The article is dedicated to a comparative study of various approaches to implementing client-server architecture for systems aimed at automatic object recognition in images and videos. The article describes an example of an existing system based on classical architecture, where the server processes images using specialized algorithms and neural networks, while the client sends data for processing and receives recognition results. The main issue identified in previous research lies in the low performance of the system when processing real-time video streams, due to limitations of existing methods of interaction between the client and server. This article compares different interaction

methods to determine the most efficient option for processing video streams with high speed and efficiency. The primary evaluation metric is the number of frames processed per unit of time. As a result of the analysis, it was found that using the WebSocket protocol has an advantage, as it allows for transmitting each new frame within a single connection.

Ключевые слова: видео, стриминг, обработка видео, сетевые протоколы, передача видеофайлов

Keywords: video, streaming, video processing, network protocols, video file transmission

Введение и описание предметной области

В настоящее время развиваются системы автоматического распознавания объектов на изображениях и видео, которые находят применение в разных предметных областях. Большинство подобных систем основаны на классической клиент-серверной архитектуре, в которых сервер функционирует как вычислительный узел, обеспечивающий обработку изображений с использованием специализированных алгоритмов и нейронных сетей. Для взаимодействия с клиентом сервер реализует определенный программный интерфейс (API), который обеспечивает передачу данных. Клиент, в свою очередь, отправляет изображения участников на обработку, после чего получает, содержащий результаты распознавания. Скорость подобных систем определяется производительностью сервера, и скоростью передачи данных между клиентом и сервером.

Проблема и основные цели

В нашей предыдущей статье [1] был предложен подход к распознаванию номеров участников при помощи нейронной сети, который, несмотря на возможность обработки видео, демонстрирует низкую производительность и не позволяет работать с потоковыми данными в режиме реального времени. Одной из причин низкой производительности является используемая реализация сетевого взаимодействия между клиентом и сервером, основанная на классическом подходе с использованием REST API, поскольку требуется установка нового HTTP-

соединения для обработки каждого нового кадра видео. Таким образом цель настоящей статьи заключается в анализе и сравнении нескольких возможных вариантов реализации взаимодействия клиент-сервер для задачи обработки видео и оценке их производительности, с целью подбора наиболее оптимального для нашей задачи. Основной метрикой оценки является количество кадров, обрабатываемых в единицу времени, что отражает способность системы к обработке видеопотоков с высокой скоростью и эффективностью. Проведение такого сравнительного анализа позволит выявить оптимальные подходы к реализации клиент-серверной архитектуры для обработки видео и определить наиболее эффективные стратегии с точки зрения производительности.

Описание сравниваемых подходов

Текущее решение основывается на передаче каждого кадра в отдельном HTTPs запросе. Проблема такого подхода - постоянные затраты на создание нового соединения.

Теоретическое время установки HTTPs соединения можно рассчитать, исходя из времени, затрачиваемого на установку TCP-соединения, установку TLS-соединения и передачу самого запроса и ответа. В общем случае [2] оно включает в себя время, требуемое для выполнения следующих этапов:

1. Установка TCP-соединения;
2. Установка TLS-соединения;
3. Отправка данных кадра по HTTPs;
4. Получение ответа от сервера по HTTPs и закрытие соединения.

Более того, фактическое время установки HTTP соединения может значительно варьироваться в зависимости от множества факторов, таких как загрузка сети, производительность сервера и клиента, а также наличие промежуточных узлов маршрутизации. Передача видео посредством HTTPs может быть описана схемой на рис.1.

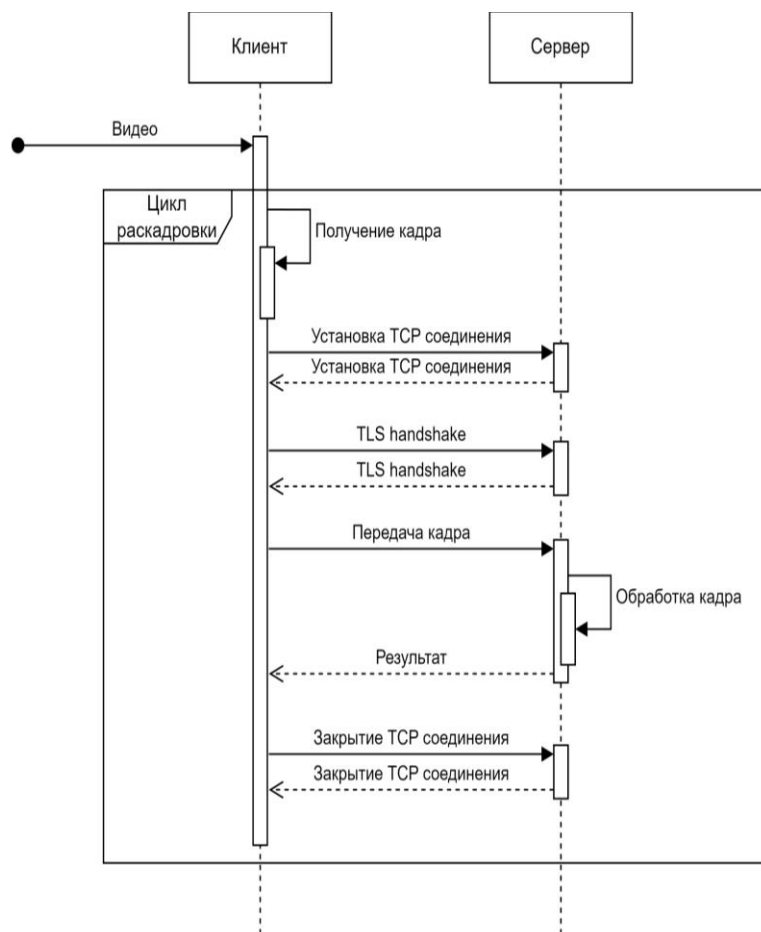


Рис. 1 - Диаграмма последовательности обработки видео при использовании HTTPS-POST-соединения

Альтернативные подходы к реализации сетевого взаимодействия, позволяют уменьшить накладные расходы на установку и разрыв HTTP-соединений для каждого кадра видео. Они основываются на использовании протоколов, которые поддерживают долгосрочное соединение между клиентом и сервером, а также возможность передачи данных в режиме реального времени без лишних задержек. Два наиболее распространенных примера таких протоколов - HTTP/2 и WebSockets [3], также для потоковых данных в формате видео часто применяется протокол HLS [4].

1. HTTP/2: это новая версия протокола HTTP, предназначенная для оптимизации передачи данных в Интернете. Одной из ключевых особенностей HTTP/2 является поддержка мультиплексирования, что позволяет устанавливать одно TCP-соединение между клиентом и

сервером и передавать множество запросов и ответов параллельно. Это существенно уменьшает задержки при обмене данными и улучшает производительность системы. HTTP/2 также поддерживает серверные push-уведомления, что позволяет серверу инициировать передачу данных клиенту без предварительного запроса.

2. WebSockets: это протокол двусторонней связи поверх TCP, который обеспечивает постоянное соединение между клиентом и сервером. После установки соединения через HTTP, WebSockets позволяют обмениваться данными в реальном времени без лишних накладных расходов. Этот протокол особенно подходит для случаев, когда требуется передача потоковых данных, таких как видео или аудио, и обеспечивает низкую задержку и высокую пропускную способность.
3. HLS (HTTP Live Streaming): этот подход подразумевает использование HTTP/HTTPs протокола для передачи видео, при этом поток разбивается на куски, т.е. в рамках одного HTTP запроса передается несколько кадров видео. Такое решение строго стандартизировано. Для возврата клиенту результата в таком случае придется использовать отдельное соединение.

Реализация асинхронного сетевого взаимодействия с использованием HTTP/2 или WebSockets позволит значительно увеличить производительность системы обработки видео, уменьшив задержки при передаче данных между клиентом и сервером и обеспечивая более эффективное использование сетевых ресурсов. Этот подход особенно полезен в задачах, где требуется обработка больших объемов потоковых данных. В рамках данной работе, было принято решение включить в сравнение реализацию с использованием WebSockets, в таком случае схема взаимодействия будет описываться схемой на рис. 2.

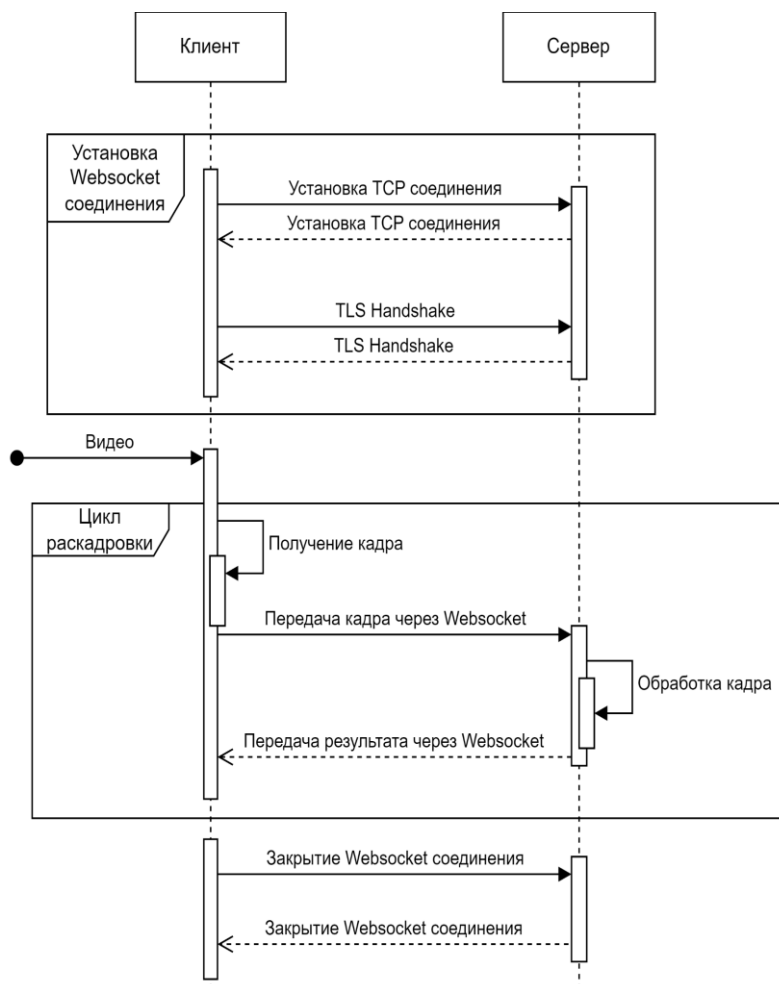


Рис. 2 - Диаграмма последовательности обработки видео при использовании WebSocket-соединения

Экспериментальная часть

Для расчета теоретических временных затрат на передачу видео, с использованием каждого из сравниваемых вариантов будем опираться на схемы взаимодействия с рис.1 и 2, а для расчета фактических значений выполним эксперимент по передаче видео между используемым сервером, и выделенным тестовым клиентом. Таким образом, для передачи видео посредством HTTP количество кадров в секунду будет описываться следующей формулой (1):

$$F = \frac{1}{(T_{TCP} + T_{TLS} + T_{Transmit} + T_{inference} + T_{Term})} \quad (1), \text{ где}$$

F - Количество кадров в секунду,

T_{TCP} - Время на установку TCP соединения, ср. время для нашей среды,

T_{TLS} - Время на установку TLS соединения, ср. время для нашей среды,

$T_{Transmit}$ - Время на передачу кадра, ср. время для нашей среды,

$T_{inference}$ - Время на обработку кадра сервером, ср. время для нашей среды,

T_{Term} - Время на закрытие соединения, ср. время для нашей среды

Также аналогично для WebSocket подсчитывается формулой (2):

$$F = \frac{1}{(T_{Transmit} + T_{inference})} \quad (2), \text{ где}$$

F - Количество кадров в секунду

$T_{Transmit}$ - Время на передачу кадра, ср. время для нашей среды

$T_{inference}$ - Время на обработку кадра сервером, ср. время для нашей среды

В таблице 1 приведены обозначения переменных из формул 1 и 2, а также их численные значения для используемой конфигурации сервера, включающей в себя два сервера, расположенные в Москве (не в локальной сети). Пропускная способность канала 500 Мбит/с.

Таблица 1 – Результаты тестирования

Переменная	Наименование	Значение
$T_{Transmit}$	Время на передачу кадра, м с	2
$T_{inference}$	Время на обработку кадра сервером, мс	25
T_{Term}	Время на закрытие соединения, м с	1
T_{TLS}	Время на установку TLS соединения, мс	17,5
T_{TCP}	Время на установку TCP соединения, мс	12,5

Исходя из вышеуказанных значений получаются следующие теоретические значения для времени количества кадров: для варианта с использованием HTTP - **16** кадров в секунду, для варианта с WebSocket - **27** кадров в секунду.

Вывод

В ходе выполнения экспериментов было подтверждено, что разница между сравниваемыми протоколами HTTP и WebSocket заключается во времени, необходимом на установку и разрыв соединения. Полученные значения сходятся с расчетами, и подтверждают, из чего можно сделать вывод о корректности представленных формул. Кроме того, полученные значения подтверждают гипотезу, о том, что использование протокола, который постоянно поддерживает соединение позволяет добиться большей производительности при обработке потоковых данных.

Литература

1. Сысойкин Е.М., Шпак И.Д., Антонов А.И. Распознавание номеров участников соревнований с помощью ансамбля моделей глубокого обучения // Нейроинформатика-2022: Сборник научных трудов XXIV Международной научно-технической конференции, Москва, 17–21 октября 2022 года. – Москва: Национальный исследовательский университет "МФТИ", 2022. – С. 213-220.
2. Harini Kolamunna, Are wearable devices ready for HTTPS? Measuring the cost of secure communication protocols on wearable devices / Jagmohan Chauhan, Yining Hu, Diego Perino, Kanchana Thilakarathna, Dwight Makaroff, Aruna Seneviratne // - URL: DOI 10.48550/arXiv.1608.04180
3. Łukasz Kamiński, Comparative review of selected Internet communication protocols / Łukasz Kamiński, Maciej Kozłowski, Daniel Sporysz, Katarzyna Wolska, Patryk Zaniewski, Radosław Roszczyk // - URL: DOI 10.48550/arXiv.2212.07475

4. Ефимов Е.Н., Голушко А. П., Моргунова О. Н., Маслова О.В. АНАЛИЗ ТЕХНОЛОГИЙ ПЕРЕДАЧИ ПОТОКОВОГО ВИДЕО // МОЛОДЕЖЬ. ОБЩЕСТВО. СОВРЕМЕННАЯ НАУКА, ТЕХНИКА И ИННОВАЦИИ. 2019. – 19. – 223-224с

References

1. Sysoykin E.M., Shpak I.D., Antonov A.I. (2022) Raspoznavanie nomerov uchastnikov sorevnovanii s pomoshch'yu ansamblya modelej glubokogo obucheniya [Recognition of competition participants' numbers using an ensemble of deep learning models]. Neuroinformatics 2022, 17–21 October 2022, pp. 213-220.
2. Jagmohan Chauhan, Yining Hu, Diego Perino, Kanchana Thilakarathna, Dwight Makaroff, Aruna Seneviratne. (2016) Are wearable devices ready for HTTPS? Measuring the cost of secure communication protocols on wearable devices, doi:10.48550/arXiv.1608.04180
3. Łukasz Kamiński, Maciej Kozłowski, Daniel Sporysz, Katarzyna Wolska, Patryk Zaniewski, Radosław Roszczyk. (2022) Comparative review of selected Internet communication protocols, doi:10.48550/arXiv.2212.07475
4. Efimov E.N., Golushko A.P., Morgunova O.N., Maslova O.V. (2019) ANALYSIS OF STREAMING VIDEO TRANSFERRING TECHNOLOGIES // MOLODEZH. OBSHCHESTVO. SOVREMENNAYA NAUKA, TEKHNIKA I INNOVATSII, no.19, pp. 223-224

© Сысойкин Е.М., Шпак И.Д., Горячкин Б. С. 2024 Научный сетевой журнал «Столыпинский вестник» №5/2024.

Для цитирования: Сысойкин Е.М., Шпак И.Д., Горячкин Б.С. СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕАЛИЗАЦИЙ СЕТЕВОГО СТЕКА ДЛЯ СЕРВЕРА ОБРАБОТКИ ВИДЕОПОТОКА // Научный сетевой журнал «Столыпинский вестник» №5/2024.