

# ВЫБОР ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ ДЛЯ ВЫСОКОНАГРУЖЕННЫХ ФИНАНСОВЫХ СИСТЕМ НА ОСНОВЕ WEBSOCKETS, SSE И GRPC

## CHOICE OF DATA TRANSMISSION TECHNOLOGY FOR HIGH-LOAD FINANCIAL SYSTEMS USING WEBSOCKETS, SSE, AND GRPC

*D. Zolotukhina*

*Summary.* The article presents a study of data transmission technologies WebSockets, SSE, and gRPC in the context of their application to high-load financial systems. A Java-based program was developed to simulate real-time transmission of financial quotes under various load levels. Key metrics, including latency, throughput, connection stability, and resource consumption, were measured. The obtained data allow for identifying the most suitable technology for different scenarios of financial applications.

*Keywords:* WebSockets, Server-Sent Events, gRPC, HTTP/2 protocol, Protocol Buffers, throughput, connection stability.

**Золотухина Дарья Юрьевна**

ФГБОУ ВО «Воронежский государственный  
университет»  
dar.zolott@gmail.com

*Аннотация.* В статье представлено исследование технологий передачи данных WebSockets, SSE и gRPC в контексте их применения для высоконагруженных финансовых систем. Разработана программа на языке Java, моделирующая передачу котировок в реальном времени с разными уровнями нагрузки. Проведены замеры ключевых метрик, включая задержку, пропускную способность, стабильность соединений и потребление ресурсов. Полученные данные позволяют определить наиболее подходящую технологию для различных сценариев финансовых приложений.

*Ключевые слова:* WebSockets, Server-Sent Events, gRPC, протокол HTTP/2, Protocol Buffers, пропускная способность, стабильность соединений.

### Введение

Современные финансовые системы требуют высокой надёжности и производительности в условиях растущего объёма данных и увеличения числа подключений. Передача финансовых данных в реальном времени, таких как котировки, уведомления о транзакциях и аналитические данные, является ключевым аспектом работы таких систем. Выбор подходящей технологии для обработки этих данных напрямую влияет на стабильность, масштабируемость и скорость реакции финансовых платформ.

Сложность задачи передачи данных в финансовых приложениях заключается в необходимости обеспечения низкой задержки и высокой пропускной способности при значительных нагрузках. Технологии, такие как WebSockets, Server-Sent Events (SSE) и gRPC, предоставляют различные подходы к решению этой проблемы, однако их производительность и ресурсоёмкость сильно варьируются в зависимости от сценариев использования.

WebSockets обеспечивают двунаправленную связь между клиентом и сервером через одно соединение, установленное по протоколу HTTP. Она позволяет обмениваться данными в реальном времени с минимальными задержками, что делает её подходящей для задач,

требующих интерактивного взаимодействия, таких как чаты, онлайн-игры или биржевые платформы. Особенностью WebSockets является постоянное соединение, которое остаётся открытым до завершения сеанса, что снижает накладные расходы на установление новых соединений [1]. Это преимущество особенно важно для приложений с большим количеством подключений, однако при росте числа клиентов увеличивается нагрузка на серверные ресурсы.

SSE использует однонаправленные потоки HTTP для передачи данных в реальном времени. SSE подходит для сценариев, где сервер должен отправлять обновления клиенту, например, для уведомлений, информационных панелей или трансляции котировок. В отличие от WebSockets, SSE обеспечивает более простую и лёгкую в реализации архитектуру, что делает её привлекательной для приложений с ограниченными требованиями к двунаправленной коммуникации. Однако использование протокола HTTP/1.1 ограничивает масштабируемость SSE в условиях высоких нагрузок, что может стать препятствием для работы с большим числом подключений [2].

gRPC — это высокопроизводительная технология удалённого вызова процедур (RPC), разработанная Google, которая использует протокол HTTP/2 и бинарный формат Protocol Buffers для передачи данных. Она

оптимизирована для минимизации задержек и максимальной пропускной способности, что делает её особенно эффективной для приложений с большими объёмами данных и высокими требованиями к стабильности. gRPC поддерживает потоковую передачу данных и двунаправленную коммуникацию, обеспечивая низкие накладные расходы даже в условиях интенсивного трафика [3]. Тем не менее её сложность в реализации и высокая ресурсоёмкость могут стать ограничением для некоторых систем.

Данное исследование направлено на систематический анализ указанных технологий с целью выявления их преимуществ и ограничений в условиях высоконагруженных финансовых систем. Результаты работы могут быть использованы для обоснованного выбора технологии, оптимально подходящей для задач финансовой индустрии.

### Материалы и методы

Для проведения исследования была разработана программа на языке Java, предназначенная для тестирования технологий передачи финансовых данных в реальном времени: WebSockets, SSE и gRPC. Программа представляет собой многопоточное приложение, которое имитирует сценарий трансляции котировок финансовых инструментов в условиях высокой нагрузки. Основная задача программы заключалась в обеспечении потоковой передачи данных в формате JSON клиентам, подключённым через различные технологии, с одновременным замером ключевых метрик производительности.

Программа была разделена на три отдельных модуля, каждый из которых реализует одну из тестируемых технологий. Для реализации WebSockets использовался фреймворк Spring Boot, обеспечивающий поддержку двунаправленного постоянного соединения. Клиенты подключались к серверу через endpoint /ws, а сервер передавал данные с минимальными накладными расходами. SSE также реализовывались на базе Spring Boot, где сервер отправлял обновления через однонаправленный поток HTTP/1.1 с использованием endpoint /sse. Реализация gRPC выполнена с использованием библиотеки gRPC-Java, обеспечивающей передачу данных через HTTP/2 и бинарный формат Protocol Buffers. Для каждого клиента открывался потоковый RPC-метод streamData, что минимизировало задержки и повышало пропускную способность. Схема архитектуры тестовой программы представлена на рисунке 1.

Передаваемые данные представляли собой синтетически сгенерированные котировки финансовых инструментов, включавшие уникальный идентификатор, временную метку, случайное значение цены в диапазоне от 1 до 1000 и статус («OK», «WARNING» или «ERROR»).

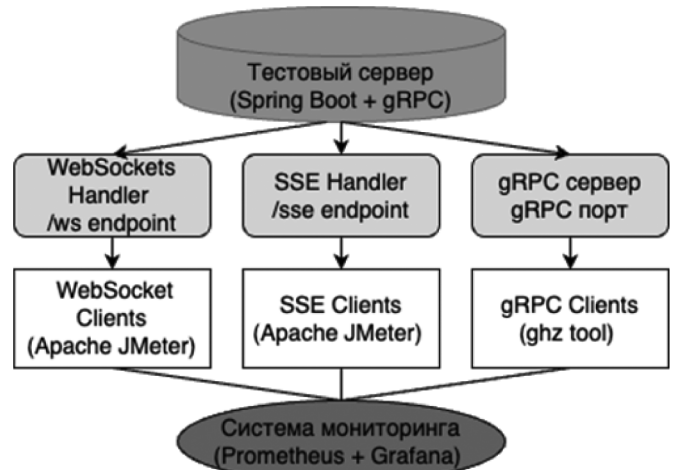


Рис. 1. Архитектура тестовой программы

Данные обновлялись каждые 100 миллисекунд для эмуляции биржевых сценариев реального времени. Генерация осуществлялась сервером программно и была унифицированной для всех тестируемых технологий.

Тестирование проводилось в условиях низкой, средней и высокой нагрузки. Низкая нагрузка предусматривала подключение 1000 клиентов с интервалом обновления данных в 100 миллисекунд. Средняя нагрузка увеличивала количество подключений до 5000, а интервал обновления сокращался до 50 миллисекунд. Высокая нагрузка включала 10 000 клиентов и интервал обновления в 10 миллисекунд. Для создания нагрузки и подключения клиентов использовались инструменты Apache JMeter[4] для WebSockets и SSE, а также ghz[5] для gRPC.

В рамках экспериментов замерялись следующие метрики:

1. Средняя и максимальная задержка (latency) для оценки времени доставки данных от сервера до клиента;
2. Пропускная способность (throughput), отражающая количество успешно переданных сообщений за единицу времени;
3. Стабильность соединений, измеряемая в процентах успешных соединений при увеличении нагрузки;
4. Потребление ресурсов (CPU и памяти), что критично для оценки масштабируемости технологий;
5. Потери данных, указывающие на количество недоставленных сообщений.

Тестовая среда включала сервер со следующими характеристиками:

- процессор: Intel Core i7-12700K (12 ядер, 20 потоков, 3.6 GHz);
- оперативная память: 32 GB DDR4;
- операционная система: Ubuntu 22.04;
- JVM: OpenJDK 17.

Мониторинг производительности осуществлялся с помощью Prometheus[6] и Grafana[7], а также встроенных инструментов JVM, таких как Java Flight Recorder и VisualVM.

Результаты

Результаты исследования представлены в таблице 1.

При низкой нагрузке минимальная средняя задержка была зафиксирована для gRPC и составила 7 мс. WebSockets показали задержку 12 мс, а SSE — 18 мс. С увеличением нагрузки до 10 000 клиентов средняя задержка возросла до 20 мс для gRPC, 65 мс для WebSockets и 80 мс для SSE.

На графике рисунка 2 представлены значения пропускной способности (Throughput) для технологий WebSockets, SSE и gRPC при трёх уровнях нагрузки: низкой, средней и высокой. Для низкой нагрузки максимальное значение Throughput продемонстрировала технология gRPC, составив 25 000 сообщений в секунду. WebSockets достигли 18 000 сообщений в секунду, а SSE — 14 000 сообщений в секунду. При средней нагрузке Throughput уменьшился для всех технологий. gRPC показал результат 15 000 сообщений в секунду, WebSockets — 10 000 сообщений в секунду, а SSE — 8 000 сообщений в секунду. При высокой нагрузке значение

Таблица 1.  
Производительность WebSockets, SSE и gRPC в зависимости от нагрузки

| Метрика                        | Нагрузка | WebSockets | SSE   | gRPC  |
|--------------------------------|----------|------------|-------|-------|
| Средняя Latency (мс)           | Низкая   | 12 ms      | 18 ms | 7 ms  |
|                                | Средняя  | 28 ms      | 35 ms | 10 ms |
|                                | Высокая  | 65 ms      | 80 ms | 20 ms |
| Throughput (сообщ/сек)         | Низкая   | 18000      | 14000 | 25000 |
|                                | Средняя  | 10000      | 8000  | 15000 |
|                                | Высокая  | 6000       | 5000  | 12000 |
| Процент разорванных соединений | Низкая   | 0.1 %      | 0.3 % | 0 %   |
|                                | Средняя  | 0.5 %      | 1 %   | 0.2 % |
|                                | Высокая  | 3 %        | 5 %   | 0.5 % |
| Потребление CPU (%)            | Низкая   | 15 %       | 10 %  | 20 %  |
|                                | Средняя  | 40 %       | 30 %  | 50 %  |
|                                | Высокая  | 80 %       | 70 %  | 90 %  |

Throughput достигло 12 000 сообщений в секунду для gRPC, 6 000 сообщений в секунду для WebSockets и 5 000 сообщений в секунду для SSE.

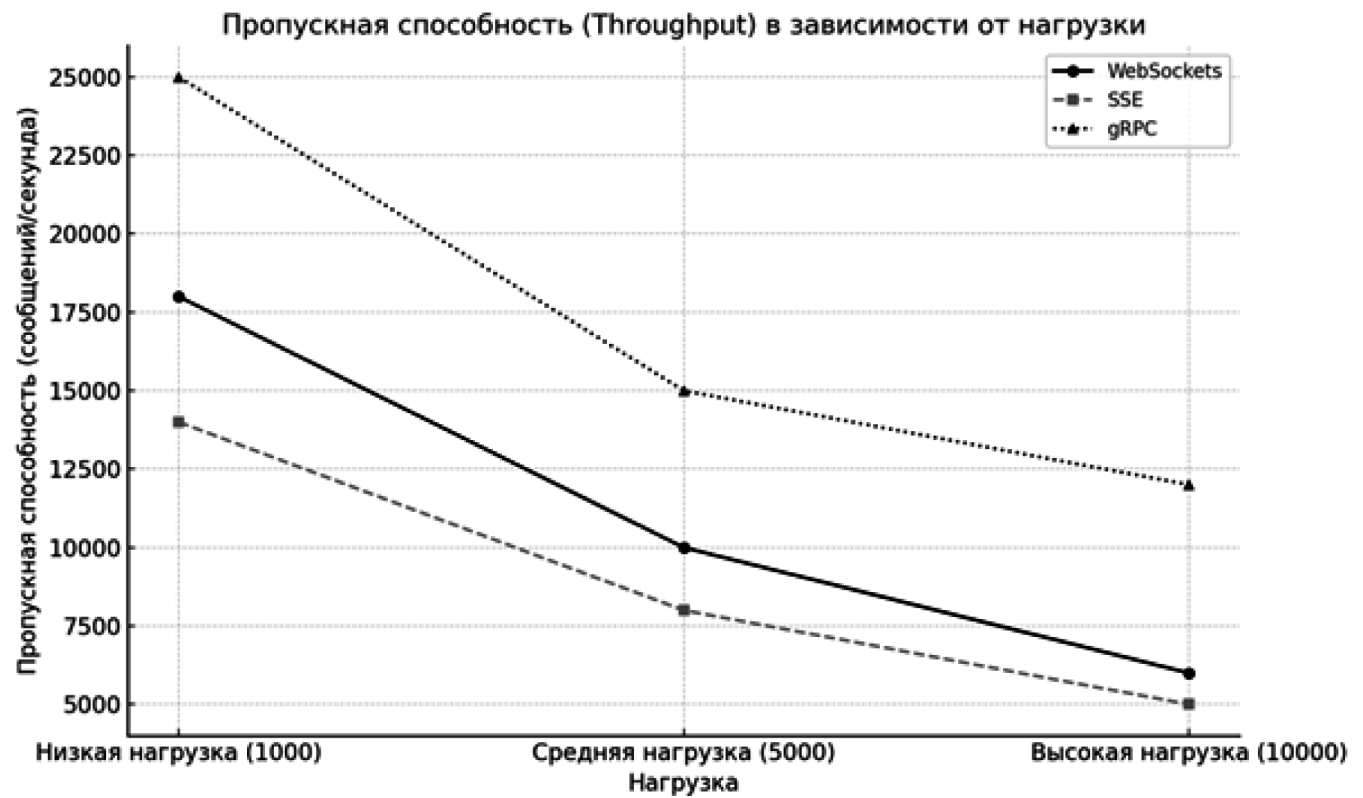


Рис. 2. Пропускная способность (Throughput) в зависимости от нагрузки

Процент разорванных соединений при низкой нагрузке составил 0 % для gRPC, 0.1 % для WebSockets и 0.3 % для SSE. При увеличении нагрузки до 10 000 клиентов процент разорванных соединений вырос до 0.5 % для gRPC, 3 % для WebSockets и 5% для SSE.

Потребление CPU при низкой нагрузке составило 15 % для WebSockets, 10 % для SSE и 20 % для gRPC. При высокой нагрузке CPU использовалось на уровне 80 % для WebSockets, 70 % для SSE и 90 % для gRPC.

### Обсуждение

Анализ результатов исследования позволяет выявить ключевые особенности и ограничения технологий WebSockets, SSE и gRPC при передаче финансовых данных в условиях различных уровней нагрузки. Основные выводы касаются показателей задержки, пропускной способности, стабильности соединений и ресурсоёмкости.

Прежде всего, использование gRPC продемонстрировало значительные преимущества в условиях низкой и средней нагрузки. Это связано с использованием HTTP/2 и бинарного протокола Protocol Buffers, которые обеспечивают более эффективное использование сетевых и процессорных ресурсов. Однако при высокой нагрузке потребление CPU у gRPC оказалось значительно выше, что может ограничивать его применение на ресурсозависимых системах. В то же время gRPC показал наименьший процент разорванных соединений, что особенно важно для критически важных финансовых приложений.

WebSockets, благодаря двунаправленной модели передачи данных, обеспечили хорошую производительность при всех уровнях нагрузки, однако их ресурсоёмкость существенно возросла при высокой нагрузке. Это может быть связано с увеличением числа активных соединений и повышенной нагрузкой на серверные ресурсы. Тем не менее WebSockets остаются универсальным решением для приложений, где важно поддерживать интерактивное взаимодействие с клиентами.

SSE показали худшие результаты среди тестируемых технологий, особенно при увеличении числа клиентов. Однонаправленная модель передачи данных через HTTP/1.1 ограничивает масштабируемость технологии, что объясняет низкие значения пропускной способности и повышенный процент разорванных соединений при высокой нагрузке. SSE может быть целесообразно использовать только для сценариев с ограниченным числом клиентов и невысокой частотой обновления данных.

### Выводы

Сравнительный анализ технологий WebSockets, SSE и gRPC, проведённый в ходе исследования, показал, что выбор технологии передачи финансовых данных зави-

сит от специфики задач, а также от характеристик нагрузки и требований к производительности.

Для приложений, обрабатывающих финансовые данные в реальном времени с большим числом подключений и строгими требованиями к задержкам и стабильности, наиболее подходящим выбором является gRPC. Использование HTTP/2 и бинарного формата Protocol Buffers обеспечивает низкие задержки и минимальные потери данных, что делает gRPC оптимальным для систем, таких как биржевые платформы, трансляция котировок или высокочастотный трейдинг. Однако высокая ресурсоёмкость gRPC при увеличении нагрузки требует оптимизации серверной инфраструктуры, например, за счёт горизонтального масштабирования или выделения специализированных серверных ресурсов.

WebSockets демонстрируют хорошие результаты в сценариях, где требуется двунаправленная передача финансовых данных, например, для предоставления клиентских уведомлений, реализации инструментов анализа или поддержки взаимодействия в трейдерских приложениях. Эта технология характеризуется высокой стабильностью при умеренной нагрузке, но при увеличении числа подключений требует оптимизации управления соединениями и распределения ресурсов, особенно для долгоживущих соединений.

SSE являются подходящим выбором для задач с низкой нагрузкой и минимальными требованиями к ресурсам. SSE могут использоваться для трансляции однонаправленных уведомлений о состоянии портфеля, изменении балансов или обновлении ограниченного набора котировок. Однако их ограниченная масштабируемость и более высокая задержка делают технологию менее эффективной для крупных финансовых систем. Перспективным направлением для SSE является интеграция с HTTP/2, что может существенно повысить их производительность.

Таким образом, выбор технологии передачи данных для финансовых приложений должен основываться на количестве подключений, требованиях к задержкам, стабильности и доступных серверных ресурсах. gRPC рекомендуется для высоконагруженных систем с большими объёмами данных, WebSockets подходят для задач с умеренной нагрузкой и двунаправленной связью, а SSE остаются разумным выбором для простых сценариев с ограниченным количеством подключений.

Полученные результаты и выводы имеют практическую ценность для разработки финансовых систем, таких как биржевые платформы, системы уведомлений, аналитические инструменты и приложения для управления портфелем. Эти технологии обеспечивают гибкость в выборе решений, адаптированных под конкретные потребности финансовой индустрии.

---

ЛИТЕРАТУРА

1. Журавлев Н.Е. Взаимодействие web-сервера и web-приложение через web-socket // Мировая наука. 2019. №12 (33). URL: <https://cyberleninka.ru/article/n/vzaimodeystvie-web-servera-i-web-prilozhenie-cherez-web-socket> (дата обращения: 20.01.2025).
2. What is Server-Sent Events (SSE) and how to implement it? [Электронный ресурс]. — Режим доступа: URL: <https://medium.com/deliveryherotechhub/what-is-server-sent-events-sse-and-how-to-implement-it-904938bfd73> (дата обращения: 20.01.2025).
3. Buzhin I.G., Derevyankin A.Y., Antonova V.M., Perevalov A.P., Mironov Y.B. Comparative analysis of the REST and gRPC used in the monitoring system of communication network virtualized infrastructure // T-Comm. 2023. №4. URL: <https://cyberleninka.ru/article/n/comparative-analysis-of-the-rest-and-grpc-used-in-the-monitoring-system-of-communication-network-virtualized-infrastructure> (дата обращения: 20.01.2025).
4. Туровец Н.О., Алефиренко В.М. Методы тестирования интегрированных информационных систем // Science Time. 2022. №3 (99). URL: <https://cyberleninka.ru/article/n/metody-testirovaniya-integrirovannyh-informatsionnyh-sistem> (дата обращения: 20.01.2025).
5. Introduction to ghz. [Электронный ресурс]. — Режим доступа: URL: <https://ghz.sh/docs/intro.html> (дата обращения: 20.01.2025).
6. Маратканов А.С., Суханов А.А., Воробьева А.А. Средства анализа и визуализации метрик работы приложения // International scientific review. 2019. №LIX. URL: <https://cyberleninka.ru/article/n/sredstva-analiza-i-vizualizatsii-metrik-raboty-prilozheniya> (дата обращения: 20.01.2025).
7. Средство визуализации данных Grafana. [Электронный ресурс]. — Режим доступа: URL: <https://github.com/grafana/grafana> (дата обращения: 20.01.2025).

---

© Золотухина Дарья Юрьевна (dar.zolott@gmail.com)

Журнал «Современная наука: актуальные проблемы теории и практики»