Wysokowydajny, uniwersalny framework RPC typu open source

gRPC w C++ z użyciem OTel

Autorzy:
Patryk Czuchnowski, Michał Pędrak, Andrzej Wacławik, Ivan Zarzhitski

Rok, Grupa:
2025, grupa 9:45 czwartek

Spis treści

1 Wprowadzenie

W ramach projektu, budujemy prostą aplikację typu klient-serwer opartą o gRPC [1] z włączoną instrumentacją danych telemetrycznych (śladów, metryk, logów prowadzonej komunikacji) przy wykorzystaniu OpenTelemetry [2]. Dzięki wykorzystaniu OTLP (czyli protokołu OpenTelemetry), zabrane dane będą mogły być eksportowane do narzędzi wizualizacyjnych takich jak Grafana [3].

Celem projektu jest stworzenie aplikacji, która nie tylko umożliwia komunikację pomiędzy klientem a serwerem za pomocą gRPC, ale również zapewnia pełną obserwowalność działania systemu, czyli zdolność do zrozumienia, co dzieje się wewnątrz niego, na podstawie zewnętrznych sygnałów (danych telemetrycznych). Dzięki integracji z OpenTelemetry, aplikacja będzie monitorowana pod kątem wydajności oraz dostępności, co pozwoli na szybsze wykrywanie ewentualnych problemów.

2 Podstawy teoretyczne i stos technologiczny

Framework gRPC to nowoczesny i wydajny system typu open source do zdalnego wywoływania procedur, który może działać w niemalże dowolnym środowisku. Umożliwia efektywne łączenie usług zarówno wewnątrz, jak i pomiędzy centrami danych, oferując możliwość podłączania modułów do obsługi równoważenia obciążenia, śledzenia danych telemetrycznych czy uwierzytelniania.

OpenTelemetry to standard i zestaw narzędzi typu open source, służący do zbierania, przetwarzania i eksportowania danych telemetrycznych z aplikacji, takich jak metryki, logi, ślady i profile. Jest on rozwijany przez Cloud Native Computing Foundation (CNCF) i ma na celu ujednolicenie sposobu obserwowalności systemów rozproszonych.

OTLP (OpenTelemetry Protocol) to standaryzowany protokół komunikacyjny używany przez OpenTelemetry do przesyłania danych telemetrycznych między aplikacjami do obserwowalności (takimi jak Grafana, Jaeger, Prometheus). Jest on binarnym protokołem opartym na gRPC lub HTTP/Protobuf [4].

W ramach projektu, wykorzystujemy technologię gRPC do komunikacji pomiędzy klientem a serwerem, OpenTelemetry do zbierania danych telemetrycznych. Protokół OTLP jest używany do eksportowania zebranych danych telemetrycznych do narzędzia Grafana, służącego do wizualizacji wyników.

3 Opis koncepcji studium przypadku

W ramach projektu, studium przypadku koncentruje się na budowie wydajnej aplikacji typu klient-serwer w środowisku rozproszonym, w której kluczową rolę odgrywają komunikacja między usługami oraz monitorowanie wydajności i dostępności systemu.

4 Architektura rozwiązania

Architektura rozwiązania opiera się na trzech głównych komponentach: kliencie, serwerze i systemie monitorowania. Wykorzystanie gRPC do komunikacji między tymi komponentami pozwala na szybkie i efektywne przesyłanie danych z minimalnym opóźnieniem. System jest zaprojektowany aby był łatwo skalowalny.

Klient jest aplikacją, która wysyła żądania do serwera. Po wysłaniu żądania, oczekuje on na odpowiedź od serwera. W trakcie działania klienta zbierane są dane telemetryczne, za pomocą OpenTelemetry. Te dane są później przesyłane do agenta lub kolektora w celu dalszej obróbki.

Serwer odpowiada na żądania wysyłane przez klienta. Przetwarza on żądania, wykonuje odpowiednie operacje i odsyła odpowiedzi do klienta. Podobnie jak klient, zbiera on dane telemetryczne o swoich działaniach, takich jak np. informacje o czasie odpowiedzi, liczba żądań lub napotkanych błędów.

Dzięki OpenTelemetry, zarówno klient, jak i serwer gromadzą dane telemetryczne, które są następnie przesyłane przez OTLP w celu eksportu do narzędzia Grafana, która umożliwia wizualizację i monitorowanie aplikacji w czasie rzeczywistym.

Odniesienia

[1] gRPC website. https://www.cncf.io/projects/grpc/.

[2] OpenTelemetry website. https://www.cncf.io/projects/opentelemetry/.

[3] Grafana tools website. https://grafana.com/.

[4] Protobuf website. https://protobuf.dev/.