

УДК 004.75:004.7

В. Дерягін; М. Дрогобицький; Н. Луцик доктор філософії, доц.

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

МЕТОДИ МОНІТОРИНГУ ТА ОПТИМІЗАЦІЇ ВЗАЄМОДІЇ МІКРОСЕРВІСІВ В ISTIO SERVICE MESH

UDC 004.75:004.7

V. Deriahin; M. Drohobyskyi; N. Lutsyk PhD., Assoc. Prof.

METHODS OF MONITORING AND OPTIMIZATION OF MICROSERVICES INTERACTION IN ISTIO SERVICE MESH

Розподілені мікросервісні системи потребують ефективних механізмів моніторингу та оптимізації міжсервісної комунікації для забезпечення надійності та продуктивності. Istio service mesh надає комплексний інструментарій для вирішення цих завдань без модифікації коду застосунків [1]. Зростання популярності мікросервісної архітектури у хмарних середовищах обумовлює необхідність впровадження спеціалізованих рішень для управління складною мережевою взаємодією між сотнями або тисячами окремих сервісів.

Моніторинг в Istio базується на автоматичному збиранні телеметрії через sidecar-прокси Envoy, які перехоплюють весь трафік між сервісами [2]. Система генерує три типи даних спостережуваності: метрики (latency, traffic volume, error rates), розподілене трасування запитів та логи доступу. Інтеграція з Prometheus забезпечує збір метрик, Grafana – їх візуалізацію, а Jaeger або Zipkin – аналіз трасування запитів через ланцюжки сервісів [1]. Ці інструменти дозволяють операторам ідентифікувати вузькі місця продуктивності та швидко локалізувати причини збоїв у розподілених системах.

Оптимізація взаємодії реалізується через механізми балансування навантаження та забезпечення відмовостійкості. Istio підтримує алгоритми Least Request, Round Robin, Random та Consistent Hash, що конфігуруються через DestinationRule [3]. Механізм Circuit Breaker автоматично обмежує запити до перевантажених сервісів при досягненні лімітів з'єднань (maxConnections) або очікуваних запитів (http1MaxPendingRequests), запобігаючи каскадним відмовам [1].

Outlier Detection доповнює оптимізацію автоматичним виключенням несправних екземплярів з пулу балансування на основі аналізу частоти помилок [3]. VirtualService дозволяє налаштовувати timeouts, retries та розподіл трафіку між версіями сервісів для A/B тестування та canary deployments [2].

Комбінація методів моніторингу та оптимізації в Istio забезпечує повну видимість стану системи та автоматичне реагування на проблеми продуктивності, підвищуючи надійність мікросервісних архітектур [4].

Література

1. Mendonça N.C., Box C., Manolache C., Ryan L., 2021. The Monolith Strikes Back: Why Istio Migrated From Microservices to a Monolithic Architecture. IEEE Software. Vol. 38, No. 5. P. 17–22.
2. Duque A.O., Klein C., Feng J., Cai X., Skubic B., Elmroth E., 2022. A Qualitative Evaluation of Service Mesh-based Traffic Management for Mobile Edge Cloud. IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid). P. 210–219.
3. Alshuqayran N., Ali N., Evans R., 2016. A Systematic Mapping Study in Microservice Architecture. 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA). P. 44–51.
4. Elkhatab Y., Povedano Poyato J., 2023. An Evaluation of Service Mesh Frameworks for Edge Systems. 6th International Workshop on Edge Systems, Analytics and Networking (EdgeSys '23). ACM, New York, NY, USA. DOI: 10.1145/3578354.3592867.