

Для визначення реальної потреби конкретної культури ET_0 використовується коефіцієнт K_c , що зберігається у базі даних у вигляді "профілів росту". Система автоматично обирає відповідний коефіцієнт K_c залежно від фази вегетації, яка визначається за кількістю днів від моменту посадки. Апаратна частина базується на мікроконтролері ESP32, датчиках температури та вологості повітря і водяній pompі.

Розроблено архітектуру системи, що дозволяє перейти від реактивного до прогностичного управління поливом. Наукова новизна полягає в економічному обґрунтуванні гібридної моделі збору даних. Запропонований підхід дозволяє використовувати точний метод Пенмана-Монтейта без необхідності встановлення дорогих локальних піранометрів та анемометрів, що робить рішення економічно доцільним для впровадження у малих господарствах та теплицях.

Запропонована прогностична система дозволяє перейти від "поливу за розкладом" до "поливу за реальною потребою". Це створює передумови для значного скорочення витрат води та енергії, уникнення стресу рослин, спричиненого як нестачею, так і надлишком вологи, та, як наслідок, підвищення врожайності.

Література

1. Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56. Rome: Food and Agriculture Organization of the United Nations.
2. Smith, J. Optimization of Water Usage in Automated Irrigation Systems. Journal of Agricultural Engineering. 2018. P. 201-214.
3. Brown, T., White, K. IoT and Data Analytics for Smart Irrigation. IEEE Transactions on Smart Agriculture. 2020. P. 145-160.
4. Chen, H., Wang, Y. Mathematical Models for Efficient Water Management. Journal of Environmental Systems. 2019. P. 98-112.
5. The Penman-Monteith Method C. 2–3. URL: https://www.researchgate.net/publication/241492864_The_Penman-Monteith_Method.

УДК 004.75

В.Л. Дерягін, М.В. Дрогобицький, Н.С. Луцик доктор філософії, доц.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ЗАСОБИ АВТОМАТИЧНОЇ ОПТИМІЗАЦІЇ ТРАФІКУ МІЖ МІКРОСЕРВІСАМИ В ISTIO SERVICE MESH

V.L. Deriahin, M.V. Drohobytskyi, N.S. Lutsyk Ph.D., Assoc. Prof.
**TOOLS FOR AUTOMATIC OPTIMIZATION OF TRAFFIC BETWEEN
MICROSERVICES IN ISTIO SERVICE MESH**

Сучасна мікросервісна архітектура значно підвищує швидкість та гнучкість розробки програмного забезпечення, проте водночас збільшує операційну складність розподілених систем [1]. Перехід від монолітних застосунків до мікросервісів створює нові виклики у сфері управління міжсервісною комунікацією, моніторингу та забезпечення безпеки. Service mesh є перспективним підходом до вирішення цієї проблеми шляхом впровадження виділеного інфраструктурного рівня над мікросервісами без необхідності модифікації їх реалізації [1]. Istio, побудований на основі високопродуктивного Envoy проху, є однією з найпоширеніших реалізацій

service mesh, що забезпечує автоматичне управління трафіком, безпеку та спостережуваність у хмарних середовищах [2].

Архітектура Istio базується на патерні sidecar proxy, де кожен мікросервіс супроводжується окремим контейнером Envoy, що перехоплює весь вхідний та вихідний трафік [2]. Така архітектура забезпечує повну прозорість мережових взаємодій для застосунків — розробникам не потрібно впроваджувати логіку управління трафіком безпосередньо у код сервісів. Це дозволяє централізовано застосовувати політики маршрутизації, балансування навантаження та забезпечення відмовостійкості без змін у коді застосунків. Компонент control plane (Istiod) динамічно конфігурує всі проксі відповідно до поточної топології mesh через xDS API, забезпечуючи автоматичне оновлення конфігурації при зміні стану кластера [3].

Ключовим механізмом оптимізації трафіку є система балансування навантаження, що підтримує алгоритми Least Request (маршрутизація до екземпляра з найменшою кількістю активних запитів), Round Robin (послідовний розподіл), Random (випадковий вибір) та Consistent Hash (забезпечення session affinity на основі HTTP-заголовків або cookies) [1]. Конфігурація здійснюється через Custom Resource Definition DestinationRule, що дозволяє визначати політики для окремих сервісів або їх підмножин (subsets). За замовчуванням Istio використовує алгоритм Least Request, який динамічно адаптується до поточного навантаження кожного екземпляра сервісу. Дослідження показують, що service mesh забезпечує ефективне управління трафіком навіть у складних розподілених середовищах з тисячами сервісів [4].

Механізм Circuit Breaker є критичним патерном для створення стійких мікросервісних застосунків, що обмежує вплив відмов та затримок окремих сервісів на загальну продуктивність системи [2]. Цей патерн запозичений з електротехніки та реалізує принцип швидкої відмови для захисту системи від перевантаження. Istio реалізує circuit breaker через параметри connection pool: maxConnections (максимальна кількість з'єднань), http1MaxPendingRequests (максимальна кількість очікуваних запитів) та maxRequestsPerConnection. При досягненні встановлених лімітів circuit breaker автоматично блокує нові запити, забезпечуючи швидку відмову замість накопичення черги запитів до перевантаженого сервісу. Після певного періоду circuit breaker переходить у напіввідкритий стан, пропускаючи обмежену кількість тестових запитів для перевірки відновлення сервісу [3].

Outlier Detection доповнює circuit breaker механізмом автоматичного виключення несправних екземплярів із пулу балансування [3]. Система аналізує частоту помилок (consecutive5xxErrors) та виключає проблемні екземпляри на визначений період (baseEjectionTime). Параметр maxEjectionPercent контролює максимальний відсоток екземплярів, що можуть бути виключені одночасно, забезпечуючи мінімальну доступність сервісу. Такий підхід забезпечує самовідновлення системи та мінімізує вплив несправних компонентів на загальну продуктивність.

Додаткові можливості оптимізації надає механізм rate limiting, який дозволяє обмежувати кількість запитів до сервісу за одиницю часу. Це особливо важливо для захисту критичних сервісів від перевантаження під час пікових навантажень або DDoS-атак. Istio підтримує як локальний rate limiting на рівні окремого проксі, так і глобальний через зовнішній сервіс, що забезпечує узгоджене обмеження швидкості запитів у масштабах всього mesh.

Важливим аспектом оптимізації є також механізм retry з експоненційним backoff, який автоматично повторює невдалі запити з наростаючими інтервалами між спробами. Це дозволяє компенсувати тимчасові збої мережі або короточасну недоступність сервісів без втручання на рівні застосунку. Конфігурація retry включає параметри

кількості спроб, таймаутів та умов, за яких запит вважається невдалим і потребує повторення.

Спостережуваність є невід'ємною складовою оптимізації трафіку в Istio. Система автоматично збирає метрики латентності, пропускну здатності та частоти помилок для кожного з'єднання між сервісами. Інтеграція з Prometheus та Grafana забезпечує візуалізацію цих метрик у реальному часі, а розподілене трасування через Jaeger або Zipkin дозволяє аналізувати шлях запиту через ланцюжок мікросервісів та ідентифікувати вузькі місця продуктивності.

VirtualService забезпечує гнучке управління маршрутизацією трафіку, включаючи розподіл між версіями сервісів для canary deployments та A/B тестування, налаштування timeouts та автоматичних retries з конфігурованою кількістю спроб [2]. Комбінація механізмів балансування навантаження, circuit breaker та outlier detection дозволяє будувати самовідновлювальні системи, де несправні компоненти автоматично ізолюються, а трафік перенаправляється до здорових екземплярів без втручання оператора.

Література

1. Li W., Lemieux Y., Gao J., Zhao Z., Han Y., 2019. Service Mesh: Challenges, State of the Art, and Future Research Opportunities. 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE). P. 122–127. DOI: 10.1109/SOSE.2019.00026.
2. Saleh Sedghpour M.R., Klein C., Tordsson J., 2022. An Empirical Study of Service Mesh Traffic Management Policies for Microservices. Proceedings of the 2022 ACM/SPEC International Conference on Performance Engineering (ICPE '22). ACM, New York, NY, USA. P. 17–27. DOI: 10.1145/3489525.3511686.
3. Karn R., Das R., Pant D., Heikkonen J., Kanth R., 2022. Automated Testing and Resilience of Microservice's Network-link using Istio Service Mesh. Proceedings of the 31st Conference of Open Innovations Association FRUCT. DOI: 10.23919/FRUCT54823.2022.9770890.
4. Elkhatib 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.

УДК 004.6:004.855

Л. П. Дмитроца, к.т.н., доцент; О. І. Шубалий, аспірант

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ СУЧАСНИХ ТРЕНДІВ АНАЛІТИКИ BIG DATA

L. P. Dmytrotsa Ph.D, Assoc. Prof.; O. I. Shubalyi, postgraduate student

(Ternopil Ivan Puluj National Technical University, Ukraine)

RESEARCH ON CURRENT TRENDS IN BIG DATA ANALYTICS

Від часу появи на початку 2000-х терміну "Big Data", в процесі активного розвитку цього напрямку Data Science, сучасні акценти досліджень у цій сфері змістилися від задач накопичення даних до пошуку шляхів покращення аналітичної обробки та якнайширшої інтеграції із штучним інтелектом. В той час як на старті "ери великих даних" для їх аналізу у більшій мірі застосовувалися класичні статистичні методи (регресія, баєсівський аналіз і т.п.) та традиційні реляційні бази даних, то