УДК 004.414

Боданов Є.С., здобувач,
Глазок О.М., к.т.н., доцент
Національний університет «Київський авіаційний інститут»

МАСШТАБУВАННЯ В KUBERNETES НА ОСНОВІ СПЕЦІАЛІЗОВАНОГО ІНТЕЛЕКТУАЛЬНОГО ОПЕРАТОРА

Сучасні багатокористувацькі платформи програмного забезпечення, що надають послуги за моделлю SaaS, стикаються з рядом викликів ефективності. Метою таких платформ є забезпечити неперервні, високопродуктивні послуги для різних категорій клієнтів, одночасно зберігаючи операційну ефективність та масштабованість. Зокрема, одна з проблем пов’язана з обмеженнями стандартного автоматичного масштабування Kubernetes, оскільки воно спирається на реактивні показники інфраструктури (CPU/RAM), яких недостатньо для суворого дотримання кількісних цілей щодо рівня обслуговування (SLO – Service Level Objective). Ціль рівня обслуговування (SLO) у Kubernetes – це вимірювана ціль для певного аспекту продуктивності сервісу, яку команда зобов'язується досягти. Прикладами можуть бути величина затримки, доступність, рівень надійності або коефіцієнт помилок. Типовий робочий процес включає вимірювання метрики за допомогою Prometheus, OpenTelemetry або іншого стеку моніторингу та порівняння метрики з визначеним SLO. Якщо SLO порушується, система запускає сповіщення, коригує ресурси або ініціює інше відповідне реагування на інциденти.

Автоматичне масштабування в Kubernetes було активною темою досліджень в останні роки. Існуючі підходи можна розділити на три основні категорії: розширення власних механізмів Kubernetes, навчання з підкріпленням та методи на основі часових рядів або машинного навчання.

У роботі [1] представлено фреймворк, який виконує автоматичне масштабування на основі затримки, довжини черги та використання ресурсів. Основним обмеженням цього підходу є його зосередженість на кожному робочому навантаженні: він оптимізує кожну послугу окремо та не враховує конкуренцію між орендарями різних рівнів за умови дефіциту ресурсів у кластері.

У роботі [2] автори пропонують методологію балансування затримки SLO та вартості інфраструктури. Автори описують Custom Pod Autoscaler, який обчислює бажану кількість реплік, використовуючи зважені коефіцієнти. Подібно до попередньої роботи, цей підхід виконує оптимізацію для кожного розгортання, ігноруючи

багатокористувацьку оренду та необхідність глобальної пріоритезації трафіку за умови насичення кластера.

У статті [3] пропонується багатокористувацька SaaS-архітектура, покращена на основі штучного інтелекту, яка забезпечує інтелектуальну ізоляцію ресурсів за допомогою динамічного масштабування. Обмеження включають залежність від точних прогнозів попиту, додаткову складність компонентів штучного інтелекту та оцінку, обмежену певними робочими навантаженнями та хмарними середовищами.

Стандартні засоби автоматичного масштабування ігнорують бюджетні обмеження та пріоритети клієнтів. Це призводить до затримки масштабування, порушень SLO та неконтрольованих витрат на інфраструктуру. У цій роботі було розроблено та впроваджено спеціалізований оператор Kubernetes для заміни стандартних механізмів масштабування інтелектуальним конвеєром прийняття рішень. Рішення інтегрує бізнес-контекст безпосередньо в цикл управління через спеціальне визначення ресурсів (CRD). Було впроваджено інтелектуальні механізми випередження та скидання навантаження для захисту високопріоритетних рівнів за умови дефіциту ресурсів, а також стратегію розміщення подів з урахуванням витрат для оптимізації витрат на інфраструктуру. Експеримент показує, що розроблена система зменшує пікову затримку приблизно в 3,8 раза (з 17,07 с до 4,5 с) порівняно зі стандартним базовим рівнем HPA. За критичної нестачі ресурсів кластера оператор надає пріоритет продуктивності вищого рівня за рахунок нижчих рівнів, що дозволяє провести контрольовану деградацію замість колапсу продуктивності всього кластера.

Список використаних джерел:

1. An SLO driven and cost-aware autoscaling framework for Kubernetes. / V. Punniyamoorthy, B. Kumar, S. Saha, L. Butra, M. Palanigounder, A. K.Agarwal, and K. Kannan. //arXiv preprint: arXiv:2512.23415, December 2025. URL: https://arxiv.org/pdf/2512.23415.

2. Marchese A., Tomarchio O. SLO and Cost-Driven Container Autoscaling on Kubernetes Clusters. /15th Int. Conf. on Cloud Computing and Services Science: proceedings. 2025. Pp. 72-79. DOI: 10.5220/0013482100003950.

3. Ravi Chandra Thota. Intelligent Orchestration for Performance-Tuned Multi-Tenant Cloud Systems /Int. J. of Scientific Research in Engineering and Management. December 2025. Vol. 09(12). Pp. 1-9. DOI: 10.55041/IJSREM55464.