KAPITEL 2 / *CHAPTER 2* [2]
PROACTIVE RESOURCE MANAGEMENT IN A KUBERNETES CLUSTER USING AI-BASED RESOURCE USAGE PREDICTION IN THE CLOUD ENVIRONMENT

DOI: 10.30890/2709-2313.2024-32-00-015

Вступ

Хмарні обчислення є ключовим компонентом сучасних технологічних рішень, оскільки вони забезпечують високу гнучкість, масштабованість і ефективність використання обчислювальних ресурсів. У цьому контексті платформа Kubernetes відіграє провідну роль, дозволяючи автоматизувати розгортання, управління та масштабування додатків. Однак, існуючі механізми управління ресурсами, такі як Horizontal Pod Autoscaler (HPA) та Vertical Pod Autoscaler (VPA), спираються на реактивний підхід, що має значні обмеження. Реактивна природа цих механізмів означає, що рішення приймаються лише після фіксації змін у навантаженні. Це призводить до низки проблем, зокрема затримок у масштабуванні, перевитрат ресурсів або їх недостатнього виділення, що, у свою чергу, може спричинити деградацію продуктивності додатків.

В умовах динамічних хмарних середовищ, де навантаження може змінюватися швидко й непередбачувано, виникає необхідність реалізації проактивного управління ресурсами. Такий підхід дозволяє передбачати зміни у використанні ресурсів і завчасно адаптувати систему до нових умов. Використання штучного інтелекту (ШІ) та методів машинного навчання відкриває нові можливості для прогнозування використання ресурсів у кластері Kubernetes. Зокрема, алгоритми прогнозування, такі як аналіз часових рядів (time-series analysis), надають змогу виявляти тренди у використанні ресурсів, що дозволяє оптимізувати витрати на інфраструктуру та підвищувати стабільність роботи додатків. Інтеграція таких алгоритмів у середовище Kubernetes забезпечує не лише економічну ефективність, а й значно покращує загальну продуктивність системи.

[2] ***Authors:*** *Tolbatov Andrii, Sokyrka Ie., Kukulevskyi I., Tolbatova Olena, Lozova Iryna*

Метою цієї роботи є дослідження підходів до проактивного управління ресурсами у кластері Kubernetes, заснованих на прогнозуванні використання ресурсів із застосуванням сучасних методів штучного інтелекту, зокрема у динамічних хмарних середовищах.

Для досягнення поставленої мети визначено такі завдання:

- Провести аналіз існуючих методів управління ресурсами у Kubernetes, включаючи HPA та VPA, з їх перевагами та обмеженнями.
- Розкрити концепцію проактивного управління ресурсами, визначивши її доцільність та потенційні переваги в умовах динамічних хмарних середовищ.
- Дослідити алгоритми прогнозування, такі як Prophet та LSTM, оцінюючи їхню відповідність задачам масштабування ресурсів.
- Оцінити вплив проактивного підходу на ефективність роботи кластеру Kubernetes, стабільність та економічність використання ресурсів.

Дослідження проактивного управління ресурсами спрямоване на вирішення ключових проблем реактивних механізмів, таких як затримка в адаптації до змін і нераціональне використання ресурсів. Використання ШІ у процесах управління Kubernetes створює можливості для значного підвищення точності прогнозування та ефективності масштабування, що відповідає сучасним вимогам до продуктивності хмарних систем.

2.1. Аналіз існуючих рішень та розкриття концепції проактивного управління ресурсами

У сучасних динамічних хмарних середовищах управління ресурсами є важливим аспектом для забезпечення продуктивності, стабільності та економічної ефективності. Kubernetes пропонує механізми автоматичного масштабування, такі як Horizontal Pod Autoscaler (HPA) та Vertical Pod Autoscaler (VPA), які значно спрощують управління ресурсами в кластері. Однак, вони

мають свої обмеження, що знижує їх ефективність у складних сценаріях з мінливим навантаженням [1, 10, 11, 12, 13, 14, 15].

2.1.1. Аналіз існуючих методів управління ресурсами у Kubernetes

Horizontal Pod Autoscaler (HPA) є стандартним інструментом Kubernetes для автоматичного масштабування кількості подів у межах одного додатка залежно від поточного використання ресурсів (CPU, пам'ять) [2]. HPA працює шляхом аналізу метрик, отриманих від Kubernetes Metrics Server, і прийняття рішення про додавання або зменшення кількості подів, якщо поточні значення метрик перевищують або знижуються відносно заданого порогу.

Таб. 1 - Переваги та недоліки HPA.

Переваги	**Недоліки**
Простота налаштування та використання	HPA відповідає на зміни лише після їх виникнення, що може призводити до затримок у масштабуванні.
Реактивна адаптація до короткострокових змін у навантаженні	Залежність від базових метрик, наприклад, CPU і пам'ять, які не завжди точно відображають поточний стан навантаження
Висока інтеграція з екосистемою Kubernetes	Немає можливості врахувати складні тренди або передбачити довготривалі зміни

Авторська розробка

Тепер розглянемо Vertical Pod Autoscaler (VPA), це інструмент, який автоматично налаштовує запити та ліміти ресурсів (CPU, пам'ять) для кожного поду залежно від його фактичного використання [3]. Він аналізує історичні дані, щоб запропонувати оптимальні параметри або автоматично оновлює їх.

Таб. 2 - Переваги та недоліки VPA.

Переваги	Недоліки
Знижує ризик недостатнього або надмірного виділення ресурсів.	Зміна ресурсних налаштувань вимагає перезапуску подів (сервісу), що може вплинути на безперервність роботи додатка.
Ефективно працює для додатків із стабільним і передбачуваним навантаженням.	VPA не завжди ефективний у ситуаціях із раптовими сплесками трафіку.
	Працює тільки на рівні налаштувань одного поду і не вирішує задачі масштабування кількості подів.

Авторська розробка

2.1.2. Обмеження існуючих рішень

Попри свої переваги, обидва методи базуються на реактивному підході, що значно обмежує їх ефективність у динамічних середовищах із непередбачуваним навантаженням. Основні обмеження полягають у наступному:

- Реактивний характер, тоюто рішення про масштабування приймаються лише після виникнення змін у навантаженні, що може призводити до затримок у реакції системи.
- Неврахування довготривалих трендів - метрики, які використовуються в HPA та VPA, не дозволяють врахувати сезонність або передбачити майбутнє зростання навантаження.
- Обмежений набір метрик - існуючі механізми враховують лише базові показники, такі як CPU і пам'ять, ігноруючи складні залежності між іншими метриками, наприклад, мережею або ввід/вивід.
- Неадаптивність до пікових навантажень - раптові зміни в навантаженні

(наприклад, рекламні кампанії або розпродажі) можуть викликати затримки у масштабуванні, що може вплинути на якість роботи додатків.

Ці обмеження створюють ризик недостатньо швидкої адаптації Kubernetes кластеру до змін у навантаженні, що критично для додатків у складних сценаріях та у динамічних хмарних середовищах.

2.1.3. Концепція проактивного управління ресурсами

Проактивне управління ресурсами — це підхід, який базується на прогнозуванні майбутнього використання ресурсів у кластері. Завдяки цьому рішення щодо масштабування або налаштування ресурсів приймаються не після, а до виникнення змін у навантаженні. Це можливо завдяки використанню алгоритмів машинного навчання та аналізу часових рядів та метрик.

Ключові принципи:

- Для прогнозування використовуються методи машинного навчання та аналізу часових рядів для визначення майбутніх змін у використанні ресурсів.
- Для завчасної адаптації виконуються дії для масштабування ресурсів до того, як система зазнає змін.
- Врахування складних залежностей є здатністю аналізувати взаємозв'язки між різними метриками та враховувати сезонні коливання чи аномалії.

Хмарні середовища, особливо з великим числом користувачів або непередбачуваними навантаженнями, потребують більш адаптивних рішень для управління ресурсами [10, 11, 12, 13, 14, 15]. Наприклад як у стримінгових сервісах вечірній трафік може зростати в кілька разів, і прогнозування дозволяє заздалегідь масштабувати інфраструктуру. А також у фінансових додатках, де піки навантаження можуть бути викликані раптовими подіями, завчасне масштабування допоможе уникнути простоїв.

Основними аспектами обґрунтування доцільності використання проактивного управління є:

- Економічна ефективність яка дозволяє уникнути перевитрат на ресурси,

виділяючи їх лише тоді, коли це необхідно.

- Підвищення стабільності - коли система встигає адаптуватися до змін у навантаженні, знижуючи ризик простоїв або збоїв.
- Інтеграція з штучним інтелектом дозволяє обробляти великі обсяги даних і враховувати складні залежності, що робить проактивний підхід більш точним і ефективним.

Таким чином, існуючі методи (HPA, VPA) забезпечують базову функціональність автоматичного масштабування, проте їхній реактивний підхід має суттєві обмеження, що обмежують їхню ефективність у складних сценаріях. Проактивне управління ресурсами, засноване на прогнозуванні, є перспективним рішенням для динамічних хмарних середовищ, яке дозволяє підвищити стабільність, ефективність та економічність роботи кластерів Kubernetes. Інтеграція таких підходів із методами штучного інтелекту відкриває нові можливості для забезпечення продуктивності сучасних систем.

2.2. Експериментальна оцінка моделей для прогнозування навантаження в Kubernetes кластерах

Для розробки та тестування прогнозів було використано підходи, запропоновані у підручнику Hyndman та Athanasopoulos [6], що є основним ресурсом для роботи з часовими рядами. Крім того, для глибшого розуміння архітектури та можливостей Kubernetes було звернено до книги "Kubernetes: Up and Running" [7].

Метою даної роботи є дослідження доцільності та можливості застосування алгоритмів масштабного прогнозування для передбачення навантаження в Kubernetes кластері. Через експеримент перевірити здатність алгоритмів Prophet і LSTM прогнозувати типові патерни навантаження компонентів кластеру.

У рамках дослідження для прогнозування навантажень на кластер Kubernetes було обрано дві моделі, які використовують різні підходи до роботи

з часовими рядами:

- Prophet - модель, для прогнозування сезонних патернів, що спеціалізується на аналізі часових рядів з трендами та сезонністю. Цей алгоритм було запропоновано Meta (раніше Facebook) і широко висвітлено у дослідженні [4]. Її особливість - це автоматичне розкладання даних на тренд, сезонність і залишки. Завдяки цьому Prophet добре працює навіть із даними, що мають аномалії чи пропуски. Ця модель дозволяє швидко налаштувати параметри без глибоких знань у сфері машинного навчання.
- LSTM (Long Short-Term Memory) - модель штучної нейронної мережі, яка відноситься до рекурентних архітектур, про що докладно розповідається в роботі Hochreiter та Schmidhuber [5]. LSTM ефективно працює з часовими рядами завдяки своїй здатності "пам'ятати" важливі залежності між значеннями на різних етапах часу.

Для тестування моделей було створені два різних набори даних з різними патернами, які імітують типові сценарії роботи кластеру Kubernetes:

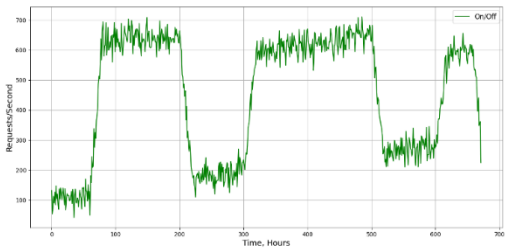

Рис. 1. Тестові дані на основі паттерну піків та нульової активності.

Авторська розробка

Цей тип навантаження характеризується різкими змінами між високими піками (On) і періодами майже нульової активності (Off). Така поведінка часто виникає в кластерах, де сервіси запускаються і зупиняються залежно від потреби, наприклад, під час обробки даних або виконання планових завдань.

У цьому випадку навантаження має чітко виражену сезонність: добову та тижневу. Паттерн змодельовано у вигляді синусоїдальних коливань із додаванням випадкових шумів, що відображає регулярну активність сервісів із періодичними піками та спаданнями.

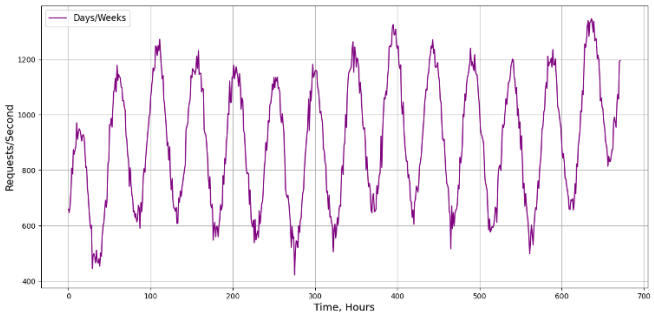

Рис. 2. Тестові дані на основі тижневого паттерну.

Авторська розробка

Мета вибору цих патернів полягала в оцінці продуктивності моделей у різних сценаріях навантаження: від різких змін до довготривалої сезонності. Всі моделі тренувалися на історичних даних довжиною 4 тижні без попередньої обробки, щоб забезпечити універсальність підходу.

Для порівняння точності моделей використовувалися Root Mean Squared Error (RMSE) та Mean Absolute Percentage Error (MAPE) метрики.

Root Mean Squared Error (RMSE) - це міра точності прогнозування, яка обчислює квадратний корінь середнього значення квадратів різниці між фактичними та прогнозованими значеннями. RMSE використовується для оцінки помилок моделі прогнозування, особливо для задач регресії та аналізу часових рядів та обчислюється за наступною формулою:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{x}_t - x_t)^2}{n}}, \quad (8)$$

де n - кількість спостережень у наборі даних., $y(i)$ - істинне значення, $\hat{y}(i)$ - передбачене значення.

Mean Absolute Percentage Error (MAPE) - дана метрика дозволяє виміряти точність прогнозування, яке обчислюється як середнє абсолютне відсоткове відхилення між фактичними значеннями та прогнозами. Розраховується за наступною формулою:

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{\hat{x}_t - x_t}{x_t}\right|, \quad (9)$$

де n - кількість спостережень у наборі даних., $y(i)$ - істинне значення, $\hat{y}(i)$ - передбачене значення.

Після тренування моделей набором даних на основі паттерну піків та нульової активності, отримали наступні результати:

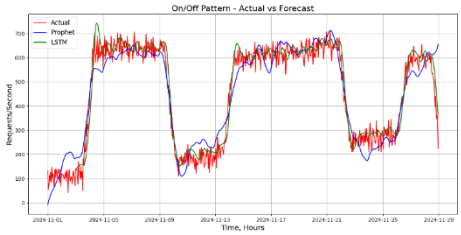

Рис.3. Прогнозування моделей на основі паттерну піків та нульової активності.

Авторська розробка

Патерн On/Off демонструє здатність моделей адаптуватися до різких змін у навантаженні. Метрики продуктивності для обох моделей наведені нижче:

Таб. 3. Результат продуктивності моделей Prophet та LSTM для патерну On/Off

Модель	**RMSE**	**MAPE (%)**
Prophet	243.65	13.89
LSTM	45.96	10.58

Авторська розробка

Як показано на Рис. 3, LSTM значно перевершує Prophet у цьому сценарії. Нижчий RMSE (45.96 проти 243.65) вказує на більшу точність у прогнозуванні різких змін. Крім того, MAPE для LSTM становить 10.58%, що є нижчим за MAPE Prophet (13.89%), підтверджуючи вищу точність.

Результат прогнозування навантаження досліджуваними моделями на основі патерну тижневих навантажень виглядає наступним чином:

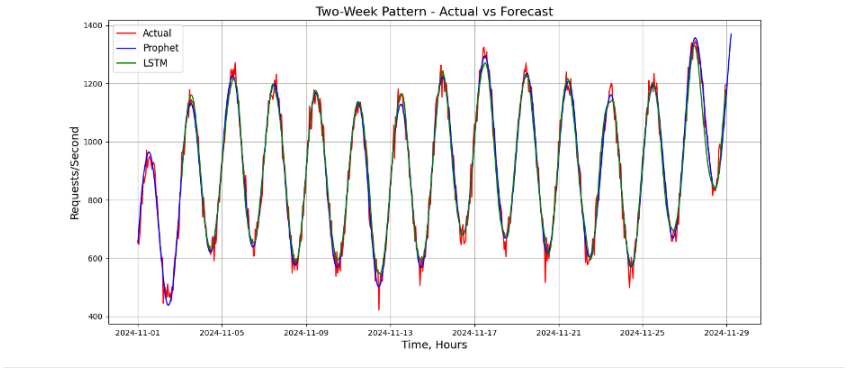

Рис. 4. Прогнозування моделей на основі тижневого паттерну.

Авторська розробка

Для більш плавного та сезонного патерну Prophet демонструє кращі результати за метрикою MAPE (2.98% проти 4.65%), що підтверджує її здатність враховувати сезонність. LSTM, хоча і має трохи вищий MAPE, показує менший RMSE (52.72 проти 104.09), що свідчить про меншу абсолютну помилку.

Таб. 4. Результат продуктивності моделей Prophet та LSTM для патерну тижневих навантажень.

Модель	**RMSE**	**MAPE (%)**
Prophet	104.09	2.98
LSTM	52.72	4.65

Авторська розробка

Для сценаріїв із різкими змінами навантаження, таких як On/Off Pattern, модель LSTM виявилася більш ефективною завдяки здатності враховувати нелінійні залежності у даних. У випадках із сезонними патернами, модель Prophet краще підходить для відносної оцінки прогнозу, але LSTM демонструє меншу абсолютну помилку. Вибір моделі для прогнозування слід здійснювати з урахуванням специфіки патерну навантаження та вимог до точності прогнозів.

Висновки

У цій роботі було проведено комплексне дослідження проактивного управління ресурсами у кластері Kubernetes із використанням методів штучного інтелекту для прогнозування навантажень. Дослідження підтвердило, що проактивний підхід є ефективним рішенням для забезпечення продуктивності та стабільності роботи додатків у динамічних хмарних середовищах. Аналіз існуючих механізмів управління, таких як Horizontal Pod Autoscaler (HPA) та Vertical Pod Autoscaler (VPA), виявив їхні суттєві обмеження, зокрема реактивний характер, що призводить до затримок у масштабуванні та ігнорування довготривалих трендів.

Запропоновано концепцію проактивного управління ресурсами, яка базується на прогнозуванні змін навантаження за допомогою алгоритмів машинного навчання. Цей підхід забезпечує завчасну адаптацію системи до змін, підвищує точність масштабування та мінімізує витрати на інфраструктуру. Експериментальне порівняння алгоритмів Prophet та LSTM показало, що LSTM є найбільш ефективною для сценаріїв із різкими змінами навантаження, тоді як Prophet демонструє кращі результати у випадках із сезонними коливаннями.

Оцінка моделей за допомогою метрик RMSE та MAPE підтвердила їх ефективність у різних сценаріях, що дозволяє адаптувати вибір алгоритму до специфіки завдань. Результати дослідження свідчать, що проактивне управління може бути інтегроване в Kubernetes через API для автоматизації масштабування на основі прогнозів, забезпечуючи підвищену стабільність і раціональне використання ресурсів.

Подальші дослідження будуть спрямовані на використання підходів підкріплювального навчання для вдосконалення прогнозів у реальному часі, інтеграцію додаткових метрик для підвищення точності оцінки стану системи та тестування розроблених рішень у масштабних реальних середовищах. Таким чином, впровадження проактивного управління ресурсами відкриває нові можливості для підвищення ефективності та економічності роботи хмарних середовищ, забезпечуючи стабільність і продуктивність сучасних обчислювальних систем.