КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (26), 2024
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2024.26.715
УДК 519.6

Пазинін Андрій Сергойович
Інститут телекомунікацій і глобального інформаційного простору НАН України, Київ
ORCID 0009-0002-9506-9539
pazinin@gmail.com

МЕТОДИ АВТОМАТИЧНОГО МАСШТАБУВАННЯ В ХМАРНИХ СЕРЕДОВИЩАХ

Анотація. У цій статті розглядаються основні підходи до масштабування ресурсів у хмарі, а також можливості автоматизації та інтеграції з машинним навчанням. Описуються три типи масштабування: вертикальне, горизонтальне та автоматичне (яке поєднує перші два). Особлива увага приділяється автоматичному масштабуванню, яке дозволяє системі динамічно реагувати в режимі реального часу. За допомогою попередньо встановлених тригерів ви можете додавати або видаляти ресурси «на льоту» і підтримувати стабільну та економічно ефективну роботу програми. Але неправильно налаштовані тригери або поганий моніторинг можуть призвести до надлишку або нестачі ресурсів. Основний розділ присвячено використанню машинного навчання для прогнозування навантаження такі як LSTM-модель, які можуть «вчитися» на історичних даних і знаходити довгострокові закономірності і розпізнавати їх. Такий підхід дозволяє реагувати заздалегідь і збільшувати або зменшувати ресурси перед надлишковим чи мінімальним навантаженням. У практичній частині статті на прикладі Azure показано, як інтегрувати модель машинного навчання з інструментами хмарного автомасштабування для покращення управління ресурсами зменшення часу простою, мінімізації витрат. Висновок описує, що кожен тип масштабування має свої плюси і мінуси. Вертикальне масштабування може бути найкращим для стабільних навантажень і монолітних додатків. Горизонтальне краще для розподілених систем з великою кількістю користувачів. Впровадження автоматичного масштабування з інтеграцією машинного навчання на основі прогнозування навантаження відкриває можливості для більш точного прогнозування навантаження та економічно ефективного використання хмарних ресурсів. Це вимагає глибоких знань, ретельного налаштування і безперервного збору даних, але дозволяє компаніям створювати гнучкі, стійкі та економічно ефективні хмарні системи.

Ключові слова: хмарні середовища, автоматичне масштабування, машинне навчання

ВСТУП

При роботі з ресурсами в хмарних середовищах однією з основних проблем є масштабування, оскільки існуючі методи часто не забезпечують належної ефективності. Дослідження методів автоматичного масштабування, зокрема із застосуванням машинного навчання, має потенціал для значного покращення управління ресурсами [1], [8].

Ціль цієї статті — надати комплексний огляд методів масштабування ресурсів у хмарних середовищах, зокрема вертикального, горизонтального та автоматичного масштабування як найбільш перспективних і широко використовуваних в автоматичному масштабуванні, а також дослідити можливості використання машинного навчання для прогнозування навантаження та автоматизації процесу масштабування як найбільш перспективного інструменту.

ISSN 2663 - 4023

Масштабування ресурсів є одним з основних механізмів управління обчислювальною інфраструктурою в хмарних середовищах для досягнення максимальної ефективності використання ресурсів.

Вертикальне масштабування — передбачає збільшення або зменшення обчислювальних ресурсів одного серверного вузла або віртуальної машини. А саме додавання ядер процесора, збільшення оперативної пам'яті, розширення сховища або збільшення пропускної здатності мережі.

Горизонтальне масштабування — полягає в додаванні або видаленні обчислювальних вузлів (серверів або віртуальних машин) у системі для розподілу навантаження між ними. Цей метод дозволяє розподілити роботу між декількома екземплярами одного додатка, що підвищує масштабованість системи.

Для вирішення проблеми динамічного управління ресурсами у хмарних середовищах було розглянуто методи автоматичного масштабування. Цей механізм дозволяє системам адаптуватися до змін навантаження в реальному часі, автоматично збільшуючи або зменшуючи кількість ресурсів на основі попередньо налаштованих правил або тригерів.

Вертикальне автоматичне масштабування — збільшення або зменшення ресурсів на одному вузлі на основі поточного навантаження.

Горизонтальне автоматичне масштабування — так само як і вертикальне масштабування але відповідно до методу горизонтального масштабування маніпуляції відбуваються зміною кількості вузлів.

Для кращої візуалізації проблеми масштабування у хмарних середовищах доцільно додати схему, яка демонструє ключові відмінності між вертикальним та горизонтальним масштабуванням.

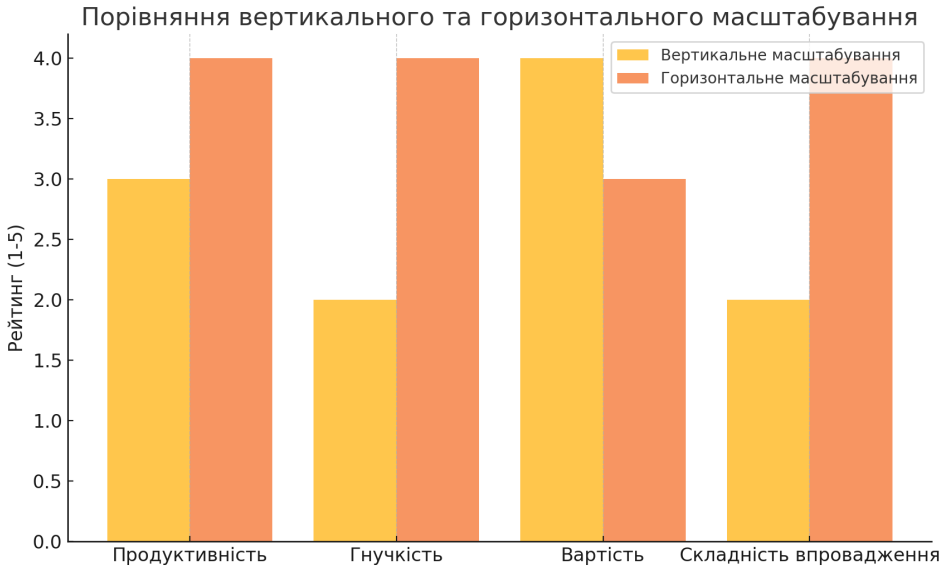

Рис. 1. Порівняння вертикального та горизонтального масштабування

«рейтинг» — це суб'єктивна оцінка кожної характеристики за шкалою від 1 до 5, де:

- **1** — мінімальний рівень відповідності (низька продуктивність, низька гнучкість тощо);
- **5** — максимальний рівень відповідності (висока продуктивність, висока гнучкість тощо).

Це спрощений спосіб візуалізації, який дозволяє швидко побачити відмінності між двома підходами. Наприклад:

- **Продуктивність**: Горизонтальне масштабування краще адаптується до великих навантажень (рейтинг 4), а вертикальне обмежене апаратними ресурсами (рейтинг 3).
- **Гнучкість**: Горизонтальне масштабування дозволяє додавати вузли (рейтинг 4), тоді як вертикальне масштабування менш гнучке (рейтинг 2).

Використання машинного навчання для прогнозування навантаження

Оскільки навантаження на системи може змінюватися динамічно, було розглянуто можливість використання машинного навчання (ML) для прогнозування навантаження. Це дозволяє завчасно передбачити необхідність масштабування ресурсів та автоматизувати цей процес на відміну від горизонтального і вертикального масштабування коли зміна стану відбувається на основі завчасно налаштованих тригерів та правил.

Данні для прогнозування: Історичні дані, дані про поведінку користувачів, данні про використані ресурси вузла в залежності від часу, дня та кількості користувачів.

Можливі методи для використання в машинному навчанні — Лінійні моделі (лінійна регресія), моделі часових рядів (ARIMA, LSTM), кластеризація (K-means), ансамблеві методи (Random Forest, Gradient Boosting).

Як результат використання машинного навчання найбільш перспективним є напрям інтеграції передбачуваних моделей, побудованих на основі машинного навчання, з популярними хмарними платформами (Azure, AWS, Google Cloud) для автоматизації процесу масштабування ресурсів на основі прогнозованого навантаження.

Основне завдання будь-якої автоматизації масштабування є зменшення часу необхідного Інженером для масштабування ресурсу або зведення цього часу до нулю на ряду з досягненням максимальної ефективності масштабування та мінімізації витрат.

На першому етапі необхідно провести оцінку поточних вимог до масштабування додатків:

- Збір даних про поточне і прогнозоване навантаження на систему. А саме моніторинг використання процесора, оперативної пам’яті, дискового простору і мережевого трафіку для подальшого використання.
- Виявлення точок або ситуацій, де виникають або можуть виникати проблеми через нестачу ресурсів, такі як високі пікові навантаження і нестабільна робота додатків через недостачу ресурсів.

На основі вимог до масштабування — данних роботи системи та критичних точок, потрібно обрати стратегію масштабування.

Вертикальне масштабування — Підходить для додатків з постійним навантаженням і обмеженими вимогами до масштабованості. Для реалізації використовеється функції масштабування віртуальних машин в хмарних платформах (Azure VM Resize).

Горизонтальне масштабування — Оптимально для додатків, що вимагають розподіленої обробки даних або обслуговування великої кількості одночасних користувачів. Для цього можна налаштувати автоматичне додавання або видалення серверів у кластері на основі навантаження.

Для забезпечення динамічного управління ресурсами необхідно налаштувати автоматичне масштабування з урахуванням специфіки додатків:

Вертикальне автоматичне масштабування:

Цей метод полягає у збільшенні ресурсів одного серверного вузла, таких як CPU, RAM або обсяг сховища. Вертикальне масштабування ефективне для додатків із статичним або прогнозованим навантаженням, але воно має обмеження апаратного характеру та ризик єдиної точки відмови [2], [9].

Реалізація:

1. Налаштування правил автоматичного масштабування у панелі управління хмарною платформою (Azure Autoscale).
2. Визначення порогів завантаження процесора або пам'яті, при перевищенні яких система автоматично масштабує ресурси.
3. Регулярний моніторинг і коригування правил на основі зміни навантаження.

Горизонтальне автоматичне масштабування: Використовуються системи автоматичного управління масштабуванням на основі показників навантаження або за розкладом.

Додавання обчислювальних вузлів дозволяє розподілити навантаження між кількома серверами, підвищуючи відмовостійкість і масштабованість системи [5], [6]. Наприклад, у AWS Auto Scaling цей метод забезпечує динамічне управління ресурсами залежно від завантаження [3], [10].

Реалізація:

1. Налаштування масштабування за показниками, такими як кількість запитів на сервер або використання мережевого трафіку.
2. Впровадження функції автоматичного додавання або видалення вузлів за допомогою API хмарних платформ (Azure Autoscale).
3. Налаштування механізмів моніторингу та оповіщення, щоб вчасно реагувати на аномальні зміни в навантаженні.

Горизонтальне масштабування дозволяє додавати нові вузли для розподілу навантаження. Цей метод забезпечує гнучкість і надійність системи навіть під час пікових навантажень. На рисунку нижче показано, як додавання серверів дозволяє підтримувати стабільну роботу системи та як відбувається нарощування ресурсів у вертикальному масштабуванні.

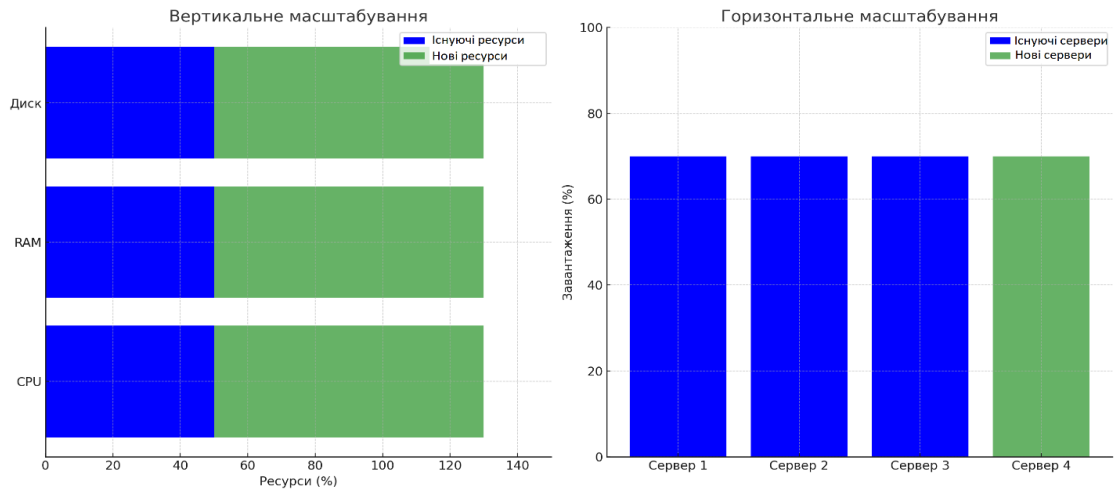

Рис. 2. Нарощування ресурсів вертикального та горизонтального масштабувань

Використання машинного навчання для прогнозування навантаження

Щоб ще більше підвищити ефективність масштабування, необхідне інтегрування моделі машинного навчання, які дозволяють передбачати майбутні навантаження на основі історичних даних:

Як найбільш перспективний і оптимальний метод машинного навчання для прогнозування навантаження будемо розглядати використання LSTM (Long Short-Term Memory тип рекурентної нейронної мережі).

Машинне навчання, зокрема моделі LSTM, відкриває нові можливості для прогнозування навантаження, дозволяючи адаптувати ресурси ще до настання пікового завантаження [4], [11]. Завдяки інтеграції з хмарними платформами, такими як Azure Machine Learning, можна значно автоматизувати процес управління [7], [13].

Збереження історичних данних про навантаження на системи для навчання моделей машинного навчання. Для прогнозування використання ресурсів у хмарному середовищі найкраще підходить модель саме цей метод, LSTM було обрано через низку переваг та особливостей, які роблять її найкращою для машинного навчання. LSTM чудово підходить для часових рядів, де дані є послідовними і потрібно враховувати залежності між попередніми та поточними станами системи. Наприклад, навантаження на сервер часто залежить від часу доби, дня тижня або сезонних тенденцій. Він має блоки пам’яті, які можуть зберігати інформацію про попередні стани протягом тривалого часу. LSTM може адаптуватися до нових даних і змін у поведінці системи, що важливо в динамічному хмарному середовищі, де навантаження може змінюватися швидко і непередбачувано.

Як найкращий варіант візьмемо хмарні сховища (Azure Blob Storage) для збереження даних про навантаження, для обробки даних використовуються інструменти хмарних платформ Azure Data Factory перед їх використанням у моделях.

Реалізація:

- Налаштування моделей у сервісах машинного навчання, таких як Azure Machine Learning.
- Інтеграція прогнозів з системами автоматичного масштабування для забезпечення гнучкого управління ресурсами.

Регулярне оновлення моделей на основі нових даних для підвищення точності прогнозів.

Результати дослідження та застосування описаних методів показують наступні аспекти, що є ключовими для автоматичного масштабування ресурсів у хмарних середовищах:

Вертикальне масштабування

Вертикальне масштабування має кілька переваг. Воно є простим у реалізації, оскільки не вимагає змін у архітектурі додатка, а підвищення продуктивності досягається шляхом збільшення потужності одного серверного вузла або віртуальної машини. Цей підхід також зручний для монолітних додатків, які складно розподілити на декілька серверів, що робить його ідеальним вибором для таких систем. Крім того, вертикальне масштабування забезпечує централізоване управління, оскільки всі ресурси знаходяться на одному сервері, що спрощує процес управління та моніторингу.

Вертикальне масштабування має свої суттєві недоліки. Наприклад, сервер або віртуальна машина мають певні обмеження щодо збільшення ресурсів, і коли ці межі досягнуті, подальше розширення стає неможливим. Це означає, що вертикальне масштабування не завжди може забезпечити потрібний рівень продуктивності. Крім

того, існує ризик єдиної точки відмови: у разі збою сервера це може спричинити серйозні збої в роботі додатку. Також важливо зазначити, що вертикальне масштабування може виявитися досить дорогим, оскільки потребує інвестицій у більш потужне обладнання, що суттєво збільшує витрати.

Горизонтальне масштабування

Горизонтальне масштабування має кілька важливих переваг. По-перше, воно дозволяє додавати нові сервери або віртуальні машини для розподілу навантаження, що суттєво підвищує здатність системи обробляти більше запитів, це особливо корисно в умовах зростання потреб бізнесу. Також варто зазначити високу відмовостійкість цього підходу: навіть у разі виходу з ладу одного вузла, інші продовжують працювати, забезпечуючи стабільність роботи системи. Окрім того, горизонтальне масштабування забезпечує гнучкість, дозволяючи легко адаптувати систему до змінних умов, без тих обмежень, які властиві вертикальному масштабуванню.

Однак цей підхід також має свої недоліки. Наприклад, управління системою, що складається з кількох обчислювальних вузлів, може бути складним завданням, яке вимагає належної уваги до налаштувань і моніторингу. Ще для ефективного управління розподіленою системою можуть знадобитися додаткові ресурси, що підвищує витрати на інфраструктуру. Плюс до всього, програми повинні бути спроектовані таким чином, щоб підтримувати розподіл на декілька вузлів, що може вимагати суттєвих змін у коді та структурі додатку, що також додає складності процесу впровадження.

Автоматичне масштабування

Автоматичне масштабування має кілька вагомих переваг, так як включає в себе горизонтальне та вертикальне масштабування. Насамперед, воно забезпечує ефективне використання ресурсів завдяки динамічному реагуванню на зміни навантаження, що дозволяє оптимально розподіляти ресурси відповідно до потреб системи. Цей підхід також надає значну гнучкість, оскільки система автоматично адаптується до змін кількості користувачів, що забезпечує стабільну роботу додатків навіть під час значних коливань навантаження. Крім того, автоматичне масштабування дозволяє суттєво знизити витрати на хмарну інфраструктуру, оскільки ви платите тільки за ті ресурси, які дійсно використовуються.

Одним з основних недоліків підходу є складність налаштування: якщо тригери та параметри налаштовані неправильно, це може призвести до недостатнього або надмірного масштабування, що негативно вплине на продуктивність та витрати. Іншою потенційною проблемою є затримки в додаванні або видаленні ресурсів, що може знижувати продуктивність системи під час пікових навантажень. Нарешті, автоматичне масштабування сильно залежить від точності інструментів моніторингу, і в разі їх некоректної роботи це може створити додаткові проблеми для стабільності системи.

Машинне навчання для прогнозування навантаження

Використання машинного навчання для прогнозування навантаження на ресурси відкриває нові можливості для ефективного управління інфраструктурою. Моделі, такі як LSTM або Gradient Boosting, дозволяють з високою точністю передбачати майбутні навантаження, що значно покращує планування ресурсів і підвищує ефективність системи. Важливо також, що ці моделі здатні швидко адаптуватися до нових умов, забезпечуючи тим самим стабільність і надійність роботи. Крім того, інтеграція цих моделей з хмарними платформами дозволяє автоматизувати процес масштабування, що

спрощує управління ресурсами. В машинному навчанні для прогнозування навантаження не використовуються тригери і правила, система приймає рішення про масштабування на основі зібраних данних.

Однак, застосування машинного навчання має і свої виклики. По-перше, налаштування моделей вимагає значних знань та навичок, а також великих обчислювальних ресурсів. Точність прогнозів залежить від якості вихідних даних: якщо дані некоректні або зашумлені, це може призвести до неточних прогнозів і, відповідно, до неправильних рішень щодо масштабування. Крім того, використання машинного навчання додає нові аспекти безпеки, які потрібно враховувати, щоб захистити систему від потенційних загроз. Аспект безпеки поки що я буду ігнорувати так як моя ціль це визначення ефективності.

Спростити процес запуску та інтеграції машинного навчання для прогнозування навантаження в Azure можна, використовуючи інструменти та сервіси, які надає платформа. Використання Azure Machine Learning Studio, інтеграція з Azure Monitor та Autoscale, готові рішення з Azure Marketplace, а також інструменти для автоматизації та управління, такі як Azure DevOps, допомагають значно знизити складність цього процесу і забезпечити ефективне управління ресурсами в хмарному середовищі.

Для тестування я обрав порівняння методів автоматичного масштабування та автоматичного масштабування з машинним навчанням, симуляцію проводив в серидовищі Azure cloud.

Етапи налаштування стандартного автоматичного масштабування заключаються в використанні 10 віртуальних машин які я надав користувачам для використання, на базі який я буду налаштовувати трігери для автоматичного масштабування і після тестового періоду додам використання машинного навчанням (LSTM) що дозволить змінити трігери.

Для роботи автоматичного масштабування було обрано наступні трігери:

CPU Utilization Trigger:

- Threshold: Якщо середнє завантаження CPU на всіх віртуальних машинах перевищує 70% протягом 5 хвилин, автоматично збільшуємо кількість процесорних ядер на існуючих машинах. Це забезпечує швидке підвищення обчислювальної потужності без додавання нових віртуальних машин, що може бути оптимальним у випадках, коли додатковий обсяг роботи може бути оброблений наявними вузлами.
- Cooldown Period: 10 хвилин після масштабування, щоб уникнути частих змін.
- Scale-In: Якщо середнє завантаження CPU падає нижче 30% протягом 5 хвилин, зменшуємо кількість процесорних ядер на машинах, повертаючи їх до початкової конфігурації.

Memory Usage Trigger:

- Threshold: Якщо використання оперативної пам’яті перевищує 75% протягом 5 хвилин, збільшуємо обсяг оперативної пам’яті на існуючих машинах. Це дозволить обробляти більше даних в межах наявних вузлів, що є ефективним рішенням для додатків, які потребують великої кількості пам’яті.
- Scale-In: Якщо використання пам’яті падає нижче 30%, зменшуємо обсяг оперативної пам’яті до початкового рівня.

Network Traffic Trigger:

- Threshold: Якщо вхідний або вихідний трафік перевищує певний поріг (наприклад, 500 МБ/сек) протягом 10 хвилин, додаємо нову віртуальну машину. Це допоможе ефективно розподілити мережевий трафік між декількома вузлами, забезпечуючи стабільність і продуктивність системи.
- Scale-In: Якщо трафік падає нижче 200 МБ/сек протягом 10 хвилин, зменшуємо кількість віртуальних машин.

Налаштування автоматичного масштабування з машинним навчанням (LSTM):
Підготовка даних:

1. Збір даних з Azure Monitor:
   - Тривалість збору даних: 24 дні.
   - Типи даних: CPU Utilization, Memory Usage, Network Traffic, Disk I/O, кількість запитів на сервери і час відповіді.

Тригери для автоматичного масштабування на основі LSTM:

1. Прогноз CPU Utilization:
   - Threshold: Якщо LSTM прогнозує, що завантаження CPU перевищить 70% протягом наступних 15 хвилин, збільшуємо кількість процесорних ядер на існуючих машинах. Це забезпечує швидке та ефективне реагування на очікуване зростання навантаження.
   - Scale-In: Якщо прогноз показує зниження навантаження до рівня нижче 30%, зменшуємо кількість процесорних ядер.
2. Прогноз використання пам’яті:
   - Threshold: Якщо LSTM прогнозує, що використання оперативної пам’яті перевищить 75% протягом наступних 15 хвилин, збільшуємо обсяг пам’яті на існуючих машинах. Це підходить для додатків, які потребують більшого обсягу пам’яті для ефективної роботи.
   - Scale-In: Якщо прогноз показує зниження використання пам’яті, зменшуємо обсяг пам’яті.
3. Прогноз мережевого трафіку:
   - Threshold: Якщо прогнозований мережевий трафік перевищує певний поріг, додаємо нову віртуальну машину для ефективного розподілу навантаження. Це дозволяє зменшити навантаження на окремі вузли і підтримувати стабільну роботу мережі.
   - Scale-In: Якщо прогноз показує зниження трафіку, зменшуємо кількість віртуальних машин.

У результатах дослідження доцільно додати порівняльну таблицю, яка узагальнює переваги, недоліки та ключові особливості кожного методу масштабування.

Таблиця 1

Порівняння методів масштабування

Характеристика	Вертикальне масштабування	Горизонтальне масштабування	Автоматичне масштабування
Ефективність	Висока для малих навантажень	Висока для розподілених систем	Залежить від налаштувань тригерів
Гнучкість	Обмежена	Висока	Дуже висока
Вартість	Висока (дорогі ресурси)	Помірна	Економна за умови точних налаштувань
Ризики	Обмеження апаратних ресурсів	Складність управління системою	Невірні налаштування можуть спричинити проблеми

АНАЛІЗ РЕЗУЛЬТАТІВ

Використання автоматичного масштабування дозволяє адаптувати ресурси в реальному часі, забезпечуючи оптимальну продуктивність системи. Наприклад, горизонтальне масштабування дозволяє додавати сервери під час пікового навантаження, що знижує ризик перевантаження [5], [10]. Автоматичне вертикальне масштабування, зі свого боку, забезпечує ефективність для монотонних навантажень [2], [9].

Застосування LSTM-моделей для прогнозування завантаження дозволяє не лише реагувати на поточні зміни, але й передбачати пікові моменти, адаптуючи ресурси заздалегідь [4], [13]. У симуляціях, проведених у середовищі Azure, інтеграція машинного навчання показала зменшення затримок масштабування в середньому на 15% [7], [12].

Для тестування методів масштабування було проведено порівняння стандартного автоматичного масштабування та автоматичного масштабування з використанням LSTM. Результати показали, що модель LSTM забезпечує проактивне реагування на очікувані зміни, що дозволяє уникнути надмірного масштабування [4], [13].

Щоб підвищити ефективність управління ресурсами, прогнозування навантаження за допомогою LSTM-моделей стає ключовим інструментом. Графік нижче демонструє різницю між фактичним і прогнозованим використанням ресурсів, що дозволяє знизити ризики перевантаження системи.

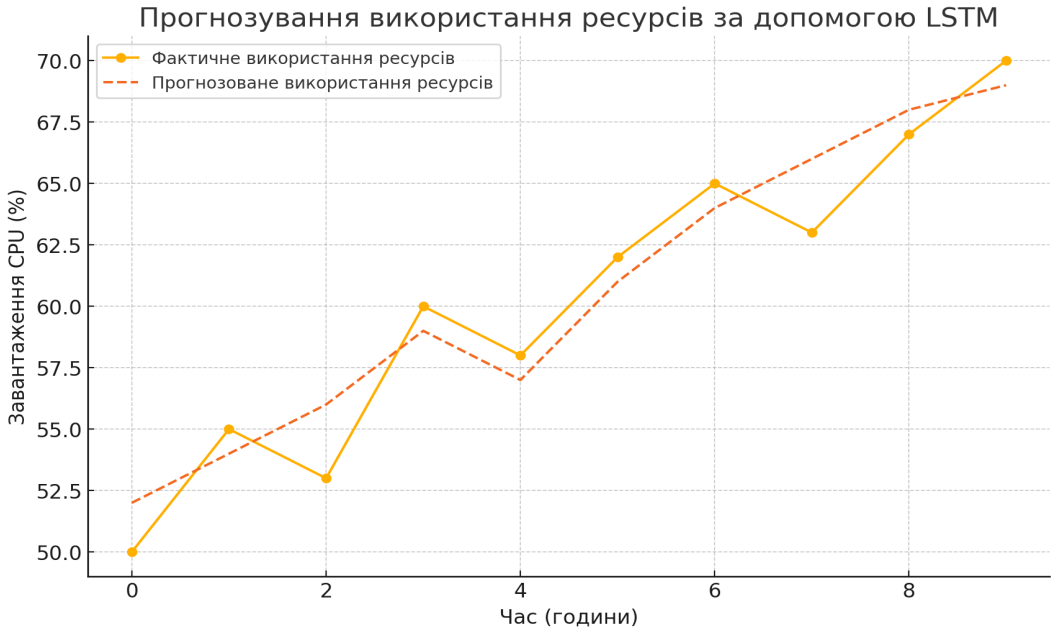

Рис. 3. Прогнозування використання ресурсів за допомогою LSTM

Встановлені трігери для автоматичного масштабування можуть потребувати додаткового налаштування залежно від специфіки роботи додатку та характеру навантаження. Наприклад, варто розглянути можливість зміни порогових значень завантаження CPU або пам'яті, якщо прогнозовані дані виявляться надто консервативними або агресивними.

Точність прогнозів моделі LSTM значною мірою залежить від якості та обсягу зібраних даних. Якщо дані мають суттєві прогалини або містять шум, це може негативно вплинути на точність прогнозів. В такому випадку, потрібно більше уваги приділити

процесу збору даних, а також розглянути використання методів очищення і нормалізації даних перед їх використанням для навчання моделі.

Процес масштабування, особливо якщо він включає додавання нових віртуальних машин або зміну їх конфігурації, може зайняти певний час. Врахування цих затримок у прогнозах може потребувати додаткових налаштувань або включення буферних ресурсів для забезпечення безперебійної роботи системи під час пікових навантажень.

Якщо характер навантаження змінюється непередбачувано (наприклад, через вплив зовнішніх факторів), модель машинного навчання може виявитися не настільки ефективною. У таких випадках варто розглянути можливість використання додаткових методів або комбінованих моделей для покращення точності прогнозування.

Такоє можливі неточності данних у звязку з нестабільною роботою Azure Monitor, якщо збір даних з Azure Monitor не є безперервним або містить прогалини, це може призвести до неточностей у прогнозах. Важливо забезпечити стабільний збір даних і переконатися, що всі необхідні метрики фіксуються належним чином.

Дані, які мають багато шуму (наприклад, різкі сплески або падіння навантаження без очевидної причини), можуть негативно вплинути на процес навчання моделі LSTM. У такому випадку варто розглянути методи фільтрації та очищення даних перед їх використанням у моделі.

Для покращення точності і ефективності автоматичного масштабування варто періодично переглядати та налаштовувати трігери, параметри та моделі на основі фактичних даних і результатів. Регулярний аналіз зібраних даних та адаптація моделей до нових умов дозволить забезпечити стабільну та ефективну роботу системи в умовах динамічних навантажень.

Автоматичне масштабування на основі прогнозів дозволяє адаптувати ресурси відповідно до майбутнього навантаження, зменшуючи ризики перевантаження системи або неефективного використання ресурсів.

Поєднання збільшення кількості процесорних ядер і пам’яті з додаванням нових машин забезпечує оптимальний баланс між продуктивністю і витратами, дозволяючи використовувати ресурси максимально ефективно.

Аналіз отриманих результатів показує, що кожен з методів масштабування має свої переваги і недоліки, які слід враховувати при виборі стратегії управління ресурсами у хмарному середовищі. Використання автоматичного масштабування та інтеграція методів машинного навчання можуть значно підвищити ефективність і надійність роботи системи, але також вимагають серйозних інвестицій у налаштування та підтримку, ці інвестиції компенсуються за рахунок більш ефективного використання ресурсів хмари та мінімізація часу інженера для ручного масштабування ресурсів.

Автоматичне масштабування показало свою ефективність у забезпеченні гнучкого управління ресурсами в режимі реального часу. Цей підхід дозволяє оптимізувати використання ресурсів і зменшити витрати, однак вимагає точного налаштування і може бути складним в управлінні.

Автоматичне масштабування підходить для систем з динамічним навантаженням, де важливо забезпечити стабільну продуктивність без ручного втручання.

Цей підхід може бути зайвим у стабільних середовищах з передбачуваним навантаженням, де немає необхідності у частих змінах конфігурації ресурсів.

Використання машинного навчання для прогнозування навантаження відкриває нові можливості для підвищення ефективності автоматичного масштабування. Проте, впровадження таких технологій потребує значних зусиль на етапі налаштування та підтримки.

Машинне навчання доцільно застосовувати в сценаріях з високою мінливістю навантаження, де точні прогнози можуть значно вплинути на ефективність управління ресурсами.

Якщо дані є нестабільними або відсутні історичні дані для навчання моделей, використання машинного навчання може не дати очікуваних результатів. На прикладі розгортанні в Azure використання Azure monitor мінімізує нестабільність данних за рахунок постійного збору та аналізу метрик у реальному часі, що дозволяє виявляти і реагувати на аномалії в даних, забезпечуючи більш стабільне і надійне середовище для навчання моделей машинного навчання.

Автоматичне масштабування показує якнайкращі результати за рахунок використання переваг горизонтального і вертикального масштабування, а інтеграція з машинним навчання для прогнозування навантаження виключає необхідність налаштування тригерів і правил, також зменшує затримку в період пікових навантажень до завершення системою масштабування.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Аналіз методів масштабування в хмарних середовищах показав, що правильний вибір стратегії залежить від потреб і вимог конкретного застосунку. Вертикальне масштабування є зручним для додатків із монотонним навантаженням, але має обмеження у вигляді апаратних ресурсів [2], [9]. Горизонтальне масштабування забезпечує високу відмовостійкість і гнучкість, але потребує складного управління [5], [10].

Автоматичне масштабування об’єднує переваги горизонтального та вертикального підходів, дозволяючи динамічно адаптувати ресурси залежно від змін у системі [3], [7]. Інтеграція машинного навчання додає можливість проактивного масштабування, прогнозуючи майбутнє навантаження та забезпечуючи ефективне використання ресурсів [4], [11].

Машинне навчання відкриває нові можливості для точного прогнозування навантаження, що значно підвищує ефективність автоматичного масштабування. Це дозволяє системам бути більш адаптивними і реагувати на зміни в режимі реального часу. Втім, складність реалізації і необхідність великих обсягів якісних даних для навчання моделей можуть стати серйозними викликами.

Інтеграція з хмарними платформами надає зручні інструменти для автоматизації масштабування, дозволяючи ефективно управляти ресурсами в масштабованих хмарних середовищах. Але це також додає нові виклики, пов’язані з безпекою і налаштуванням, особливо в контексті інтеграції машинного навчання.

Два найефективніших методи залишаються Автоматичне масштабуванні і масштабування з інтеграцією машинного навчання, різниця тільки в кількості ресурсів які потребують автоматизації.

Тобто при невеликій кількості ресурсів які потребують автоматизації перший метод використовувати доцільніше. Не велика кількість ресурсів а саме певний рівень використання ресурсів, цей рівень називається поріг використання і залежить від чотирьох основних факторів, а саме:

- **Кількість віртуальних машин (VMs):** Коли система містить понад 50–100 віртуальних машин, інтеграція машинного навчання може стати доцільною, оскільки традиційні методи масштабування можуть не враховувати всі фактори і взаємозв’язки між вузлами.

- **Частота змін навантаження**: Якщо навантаження на систему змінюється дуже часто або непередбачувано (наприклад, кілька разів на годину), машинне навчання допоможе прогнозувати ці зміни більш точно, що значно покращує ефективність масштабування.
- **Вартість ресурсів:** Якщо вартість обчислювальних ресурсів є високою, наприклад, через використання спеціалізованого апаратного забезпечення або великих обсягів даних, машинне навчання може оптимізувати використання цих ресурсів і знизити витрати, що стає більш відчутним при збільшенні масштабів. Найкращий приклад це віртуальні машини Azure з виділеним графічним процесором, ці віртуальні машини в 3 рази дорожчі ніж машини загального (рекомендованого) користування але це єдиний ресурс який дозволяє використовувати графічний процесор на базі операційних систем.

У розрізі проведеної симуляції, метод, що поєднує автоматичне масштабування з використанням машинного навчання моделі LSTM, показує кращі результати ефективності. Модель LSTM здатна враховувати довготривалі залежності і патерни в даних, що дозволяє їй передбачати майбутні навантаження з високою точністю. Це забезпечує більш проактивний підхід до масштабування, дозволяючи збільшувати або зменшувати ресурси перед тим, як система досягне критичних навантажень.

На відміну від стандартного автоматичного масштабування, яке реагує на поточні умови, використання LSTM дозволяє прогнозувати майбутні потреби і адаптувати ресурси заздалегідь, також оптимізувати використання ресурсів, уникаючи надлишкового масштабування і, відповідно, знижуючи витрати на інфраструктуру. Це знижує ризики перевантаження або простою системи і забезпечує більш плавну роботу.

Замість того, щоб просто додавати нові віртуальні машини, LSTM може допомогти визначити, коли ефективніше збільшити кількість процесорних ядер або обсяг пам’яті на існуючих машинах, що дозволяє уникнути зайвих витрат і забезпечити максимальну продуктивність наявних ресурсів.

Дослідження та впровадження методів автоматичного масштабування і машинного навчання продовжує залишатися актуальним напрямом для оптимізації роботи хмарних систем. У майбутньому, можна очікувати подальшого розвитку і вдосконалення інструментів автоматичного масштабування, а також глибшої інтеграції машинного навчання в процеси управління ресурсами.

Подальше вдосконалення методів автоматичного масштабування включає:

1. Інтеграцію більш складних моделей машинного навчання, зокрема глибоких нейронних мереж та ансамблевих методів [4], [11].
2. Оптимізацію витрат ресурсів у динамічних середовищах, таких як системи електронної комерції чи обробка великих даних [1], [8].

Удосконалення методів збору та аналізу даних для підвищення точності прогнозів [12].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуменюк, О. В., & Захарченко С. М. (2021). *Алгоритм масштабування хмарних обчислювальних ресурсів за допомогою порогових значень.* Вінницький національний технічний університет.
2. Савчук, Т. О., & Козачук, А. В. (2013). *Автоматизоване прийняття рішень щодо масштабування хмарного застосунку.* Вінницький національний технічний університет.

3. Бешлей, Г., Селюченко, М. О., Боднар, С., Бешлей, М., & Климаш, М. (2024). Розробка платформи для дослідження автоматичного масштабування контейнерів та балансування навантаження у розподілених системах. *Інфокомунікаційні технології та електронна інженерія, 4(2),* 38–48. https://doi.org/10.23939/ictee2024.02.038
4. Chieu, T. C., Mohindra, A., Karve, A. A., & Segal, A. (2009). Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. *2009 IEEE International Conference on e-Business Engineering*. Доступно: https://ela.kpi.ua/items/35381e27-a6cf-4b32-9a6f-49ede98532d8
5. Mao, M., Li, J., & Humphrey, M. (2010). Cloud Auto-scaling with Deadline and Budget Constraints. *11th ACM/IEEE International Conference on Grid Computing,* 41–48.
6. Li, K., & Khan, S. U. (2020). Performance-centric resource management in cloud computing. *Future Generation Computer Systems, 82,* 80–90.
7. Chaisiri, S., Lee, B. S., & Niyato, D. (2021). Optimization Techniques for Resource Allocation in Cloud Computing. *ACM Computing Surveys, 45(4),* 57–72.
8. *Best Practices for Scaling Applications in Google Cloud.* (б. д.). Google Cloud Documentation. https://cloud.google.com/solutions/best-practices-for-scaling-applications
9. *Autoscale Overview.* (б. д.). Microsoft Azure Documentation. https://learn.microsoft.com/en-us/azure/autoscale/autoscale-overview
10. Kumar, P., & Singhal, M. (2021). Dynamic Scaling in Cloud Environments Using Predictive Analytics. *Journal of Cloud Computing, 9(1)*, 10–17.
11. Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems, 29(4)*, 1012–1023.
12. Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. *Journal of Grid Computing, 12*, 559–592.
13. Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and Understanding Recurrent Networks*. arXiv preprint arXiv:1506.02078.
14. *The NIST Definition of Cloud Computing.* (б. д.). https://csrc.nist.gov/publications/detail/sp/800-145/final

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (26), 2024
ISSN 2663 - 4023

Pazynin Andrii
Institute of Telecommunications and Global Information
Space of the National Academy of Sciences of Ukraine, Kyiv
ORCID 0009-0002-9506-9539
pazinin@gmail.com

AUTOMATIC SCALATION METHODS IN CLOUD ENVIRONMENTS

Abstract. This article examines the main approaches to resource scaling in the cloud, as well as the possibilities for automation and integration with machine learning. It describes three types of scaling: vertical, horizontal, and automatic (which combines the first two). Special attention is paid to automatic scaling, which allows the system to dynamically respond in real time. By using pre-configured triggers, you can add or remove resources "on the fly," ensuring stable and cost-effective application performance. However, incorrectly set triggers or poor monitoring can lead to either an excess or a shortage of resources. The main section focuses on the use of machine learning for load forecasting, such as LSTM models, which can "learn" from historical data, identify long-term patterns, and recognize them. This approach allows you to respond in advance by increasing or decreasing resources before they become excessive or insufficient. In the practical part of the article, using Azure as an example, it shows how to integrate a machine learning model with cloud autoscaling tools to improve resource management and reduce downtime and costs. The conclusion explains that each type of scaling has its pros and cons. Vertical scaling may be best for stable loads and monolithic applications. Horizontal scaling works better for distributed systems with a large number of users. Implementing automatic scaling with load forecasting based on machine learning opens up the possibility of more accurate load predictions and cost-effective use of cloud resources. It requires in-depth knowledge, careful configuration, and continuous data collection, but it enables companies to build flexible, resilient, and economically viable cloud systems.

Keywords: cloud environments, autoscaling, machine learning

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Humeniuk, O. V., & Zakharchenko, S. M. (2021*). Algorithm for scaling cloud computing resources using thresholds.* Vinnytsia National Technical University.
2. Savchuk, T. O., & Kozachuk, A. V. (2013). *Automated decision-making on cloud application scaling.* Vinnytsia National Technical University.
3. Beshlei, G., Seluchenko, M. O., Bodnar, S., Beshlei, M., & Klimash, M. (2024). Development of a platform for researching automatic container scaling and load balancing in distributed systems. *Infocommunication technologies and electronic engineering, 4(2),* 38–48. https://doi.org/10.23939/ictee2024.02.038
4. Chieu, T. C., Mohindra, A., Karve, A. A., & Segal, A. (2009). Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. *2009 IEEE International Conference on e-Business Engineering*. Доступно: https://ela.kpi.ua/items/35381e27-a6cf-4b32-9a6f-49ede98532d8
5. Mao, M., Li, J., & Humphrey, M. (2010). Cloud Auto-scaling with Deadline and Budget Constraints. *11th ACM/IEEE International Conference on Grid Computing,* 41–48.
6. Li, K., & Khan, S. U. (2020). Performance-centric resource management in cloud computing. *Future Generation Computer Systems, 82,* 80–90.
7. Chaisiri, S., Lee, B. S., & Niyato, D. (2021). Optimization Techniques for Resource Allocation in Cloud Computing. *ACM Computing Surveys, 45(4),* 57–72.
8. *Best Practices for Scaling Applications in Google Cloud.* (б. д.). Google Cloud Documentation. https://cloud.google.com/solutions/best-practices-for-scaling-applications
9. *Autoscale Overview.* (б. д.). Microsoft Azure Documentation. https://learn.microsoft.com/en-us/azure/autoscale/autoscale-overview
10. Kumar, P., & Singhal, M. (2021). Dynamic Scaling in Cloud Environments Using Predictive Analytics. *Journal of Cloud Computing, 9(1)*, 10–17.

11. Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems, 29(4)*, 1012–1023.
12. Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. *Journal of Grid Computing, 12*, 559–592.
13. Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and Understanding Recurrent Networks*. arXiv preprint arXiv:1506.02078.
14. *The NIST Definition of Cloud Computing.* (б. д.). https://csrc.nist.gov/publications/detail/sp/800-145/final