

АЛГОРИТМ МАСШТАБУВАННЯ ХМАРНИХ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ ЗА ДОПОМОГОЮ ПОРОГОВИХ ЗНАЧЕНЬ

Вінницький національний технічний університет

Анотація

У роботі висвітлюється процес покращення масштабування хмарних обчислювальних ресурсів за допомогою алгоритму масштабування з використанням порогових значень. Дана робота розглядає два типи масштабування: горизонтальне – за рахунок збільшення кількості оперативної пам'яті та швидкості центрального процесору; вертикальне масштабування - за рахунок збільшення кількості віртуальних машин.

Ключові слова: горизонтальне масштабування, хмарні ресурси, вертикальне масштабування

Abstract

The work highlights the process of improving the scaling of cloud computing resources using a scaling algorithm with threshold values. This work considers two types of scaling: horizontal - by increasing the amount of RAM and the speed of the central processor; vertical scaling - by increasing the number of virtual machines.

Keywords: horizontal scaling, cloud resources, vertical scaling

Вступ

Розробка ефективних алгоритмів масштабування динамічних обчислювальних середовищ є актуальною задачею для сучасних науковців із усього світу.

Сьогодні, для продовження роботи над науковими відкриттями та розробками, необхідно аналізувати величезні масиви даних. Під час даного процесу виникає дві проблеми: час, необхідний для виконання аналізу та бюджет наукового дослідження.

Для того, щоб пришвидшити процес аналізу необхідно мати дуже потужні обчислювальні середовища, які мають велику вартість. І навпаки, для того щоб зменшити вартість аналізу отриманих результатів, можливо проводити аналіз з використанням менших потужностей. Відповідно, необхідно знайти баланс між цими двома характеристиками за допомогою ефективних алгоритмів масштабування динамічних обчислювальних ресурсів

Алгоритми масштабування пам'яті та ресурсів процесора.

Було розроблено два окремих алгоритми для масштабування пам'яті та ЦП. Принцип роботи обох алгоритмів однаковий. У разі використання алгоритму масштабування пам'яті кожна віртуальна машина використовує пам'ять (M_x). Значення кількості зчитувань із пам'яті порівнюється з максимальним пороговим значенням пам'яті (M_{mx}). Якщо M_x більше або дорівнює M_{mx} , тоді лічильник максимального використання пам'яті (MT_x) збільшується, якщо менше лічильник використання пам'яті (MT_m) скидається.

Роботу алгоритму масштабування пам'яті показано на Рисунку 1. Параметри алгоритму мають наступні значення:

- M_{mx} – Максимальне значення пам'яті

- M_{mn} – Мінімальне значення пам'яті
- T – Інтервал для зчитування значення пам'яті
- $TTMt$ – Порогове значення лічильника пам'яті.
- MTx – Максимальна кількість зчитувань значення пам'яті.
- MTm - Мінімальна кількість зчитувань значення пам'яті.
- Mx – Використання пам'яті віртуальною машиною.

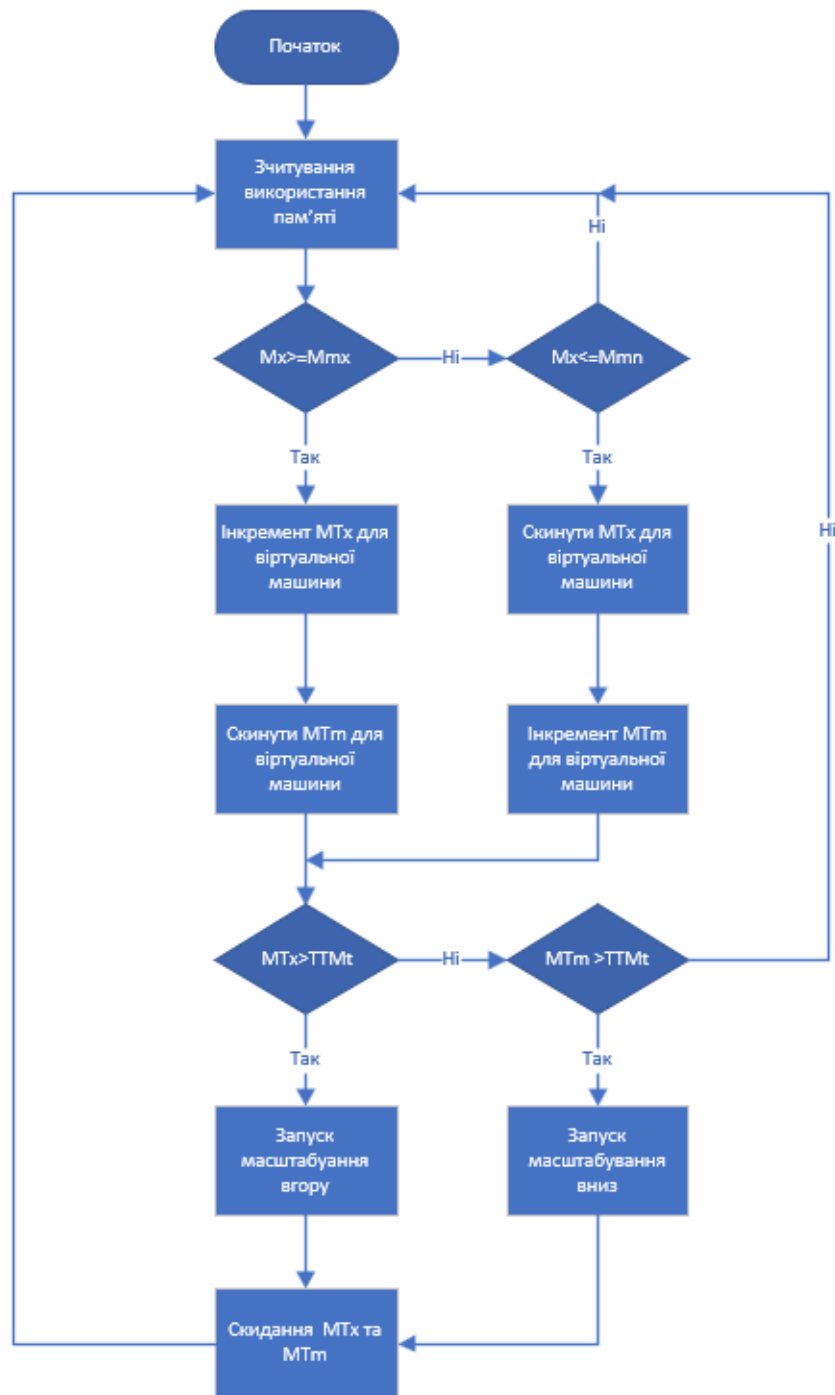


Рисунок 1 Алгоритм масштабування обсягу пам'яті

Робота алгоритму масштабування ресурсів центрального процесора показана на Рисунок 2.

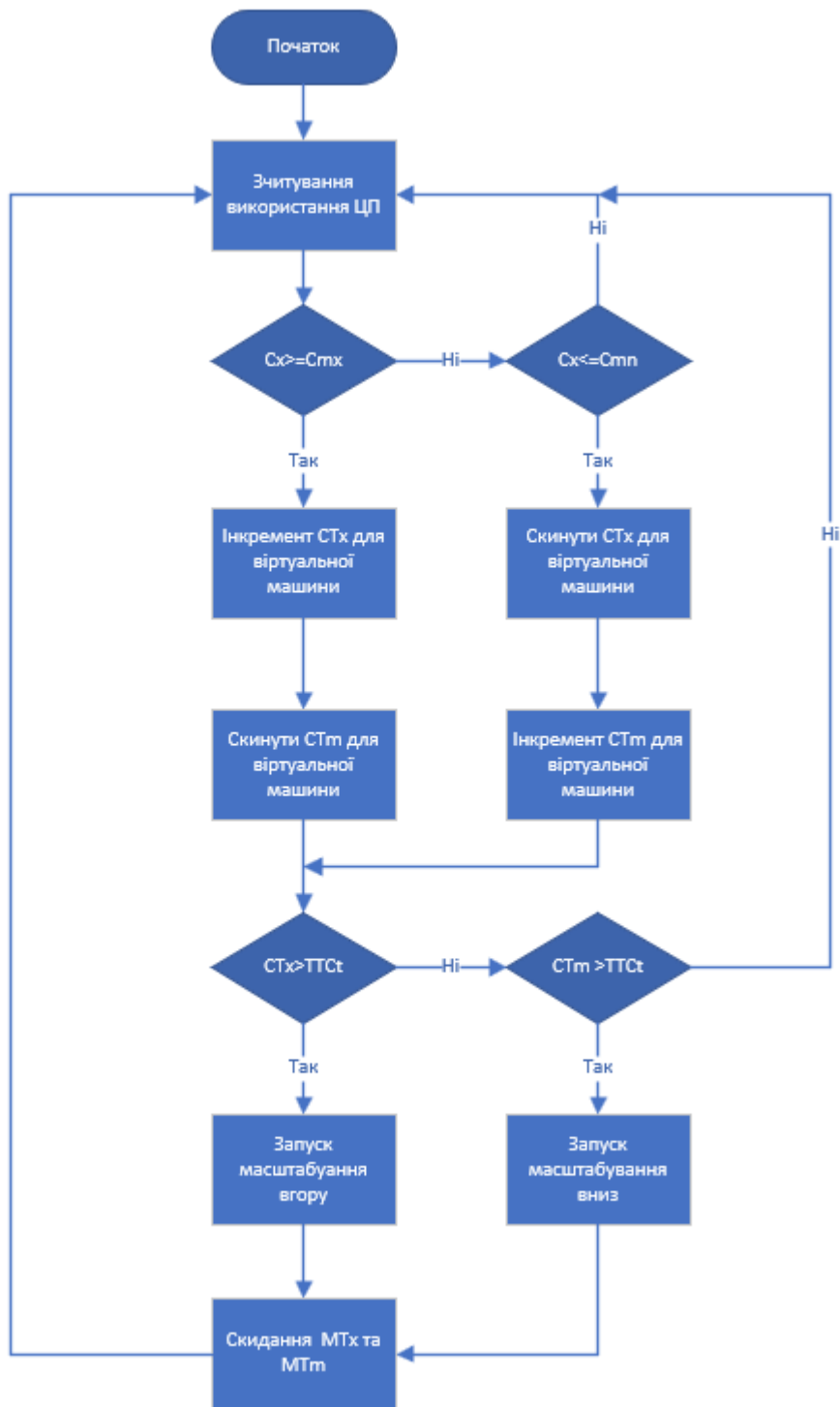


Рисунок 2 Алгоритм масштабування ресурсів центрального процесора

Параметри алгоритму мають наступні значення:

- C_{mx} – Максимальне значення ЦП
- C_{mn} – Мінімальне значення ЦП
- T – Інтервал для зчитування значення ЦП
- $TTCt$ – Порогове значення лічильника ЦП.
- CTx – Максимальна кількість зчитувань значення ЦП.
- CTm – Мінімальна кількість зчитувань значення ЦП.
- Cx – Використання ЦП віртуальною машиною.

Результати роботи алгоритму масштабування за пороговими значеннями

Хмарне середовище налаштовується за допомогою Xen Cloud Platform (ХСР). ХСР включає в себе Xen Hypervisor, стек інструментів Xen API, vSwitch тощо. ХСР — корпоративна програма з відкритим кодом, яка являє собою платформу віртуалізації серверів і хмарних обчислень. Багато з існуючих IaaS провайдерів використовують налаштований ХСР для створення інфраструктури віртуалізації. ХСР надає гіпервізор Xen із підтримкою ряду гостьових операційних систем, включаючи підтримку мереж і сховищ Windows і Linux, інструменти керування в єдиному перевіреному інтсаяційному образі, який також називають пристроєм ХСР. Він також підтримує динамічне масштабування віртуальної машини. ХСР API використовуються для отримання статистики пам'яті та використання ЦП віртуальних машин, а також для масштабування віртуальних машин

ХСР підтримує два типи пам'яті: статичну та динамічну. Кожен тип має мінімальне і максимальне значення. Максимальний обсяг статичної пам'яті визначає максимальний обсяг фізичної пам'яті, до якої гостьова операційна система може звертатися з моменту, коли гостьова система завантажується, до моменту, коли гостьова система вимкнеться. Неможливо змінити обсяг статичної пам'яті, коли віртуальна машина працює. У випадку динамічної пам'яті можливо збільшити/зменшити діапазон під час роботи віртуальної машини. ХСР надає функцію під назвою контролер динамічної пам'яті. Використовуючи такі API, надані ХСР, з'являється збільшувати/зменшувати динамічну пам'ять у межах допустимого діапазону, коли це необхідно.

Масштабування ЦП можна здійснити шляхом зміни ємності ЦП. Ємність ЦП додатково фіксує максимальну кількість ЦП, яку може споживати домен. Обмеження виражається у відсотках від одного фізичного ЦП: 100 – це 1 фізичний ЦП, 50 – половина ЦП, 400 — це 4 ЦП тощо... Значення за замовчуванням 0 означає відсутність верхнього обмеження. Вага ЦП віртуальної машини визначає, скільки центрального процесора виділено для цієї віртуальної машини. Домен із вагою 512 отримає вдвічі більше ЦП, ніж домен із вагою 256 на виділеному комп'ютері. Допустимі ваги варіюються від 1 до 65535, а за замовчуванням мають значення 256. В даному випадку для масштабування використовується ємність ЦП.

Хмарний сервер налаштовано на 4-ядерній машині з процесором Intel Xeon W3250 з 2,67 ГГц, 12 ГБ оперативної пам'яті DDR3, 1 ТБ Жорсткий диск з 7200 об/хв. Машина, яка використовується для моніторингу використання пам'яті та ЦП віртуальними машинами має процесор Intel Core2 Duo з тактовою частотою 2,66 ГГц, 1 ГБ оперативної пам'яті, жорсткий диск 500 ГБ, під'єднаний через локальну мережу Ethernet. Використовується операційна система CentOS 5.7.

Для створення навантаження на віртуальні машини використовувались як інтенсивні обчислювальні програми, так і програми, що інтенсивно використовують пам'ять. Після запуску програм використання процесору і пам'яті віртуальних машин збільшилось. Було встановлено наступні коефіцієнти: коефіцієнт масштабування для пам'яті як 1,25, а для ЦП – 2, лічильник використання встановлено на 3 хвилини, тобто якщо ресурс використання перевищує верхнє порогове значення протягом 3 хвилин безперервно, потім відповідним віртуальним машинам буде виділено більше ресурсів, як зазначено в коефіцієнті масштабування (масштабування в гору). Якщо використання ресурсу нижче нижнього порогового значення протягом 3 хвилин безперервно, тоді ресурс буде вилучено з віртуальної машини відповідно до коефіцієнту масштабування (Масштабування вниз).

Були встановлені наступні порогові значення: верхній поріг – 80%, нижній – 25%.

Необхідно враховувати наступні особливості:

- Масштабування вниз мінімізує витрату ресурсів.
- Під час масштабування в гору необхідно переконатися, що продуктивність програми не погіршується.

- Вибір правильних порогових значень важливий для коректної роботи алгоритму

Нижче порогове значення призводить до підвищення коливання ємності VM, а вище порогове значення робить алгоритм менш чутливими до зміни ресурсу використання VM.

На Рисунках 3 і 4 показано збільшення та зменшення пам'яті відповідно. Рисунки 5 і 6 показують збільшення та зменшення масштабу ЦП відповідно. Оскільки було встановлено підрахунок використання до 3 хвилин, можна спостерігати масштабування на 4-й хвилині в усіх випадках.



Рисунок 3 Збільшення розміру оперативної пам'яті

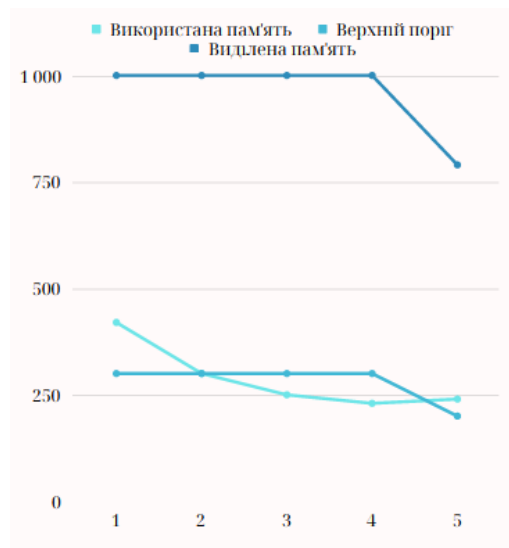


Рисунок 4 Зменшення розміру оперативної пам'яті

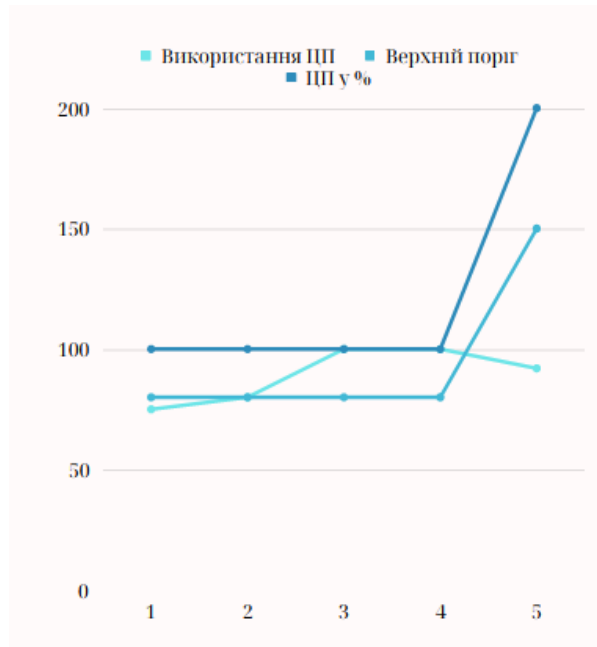


Рисунок 5 Масштабування ЦП вгору

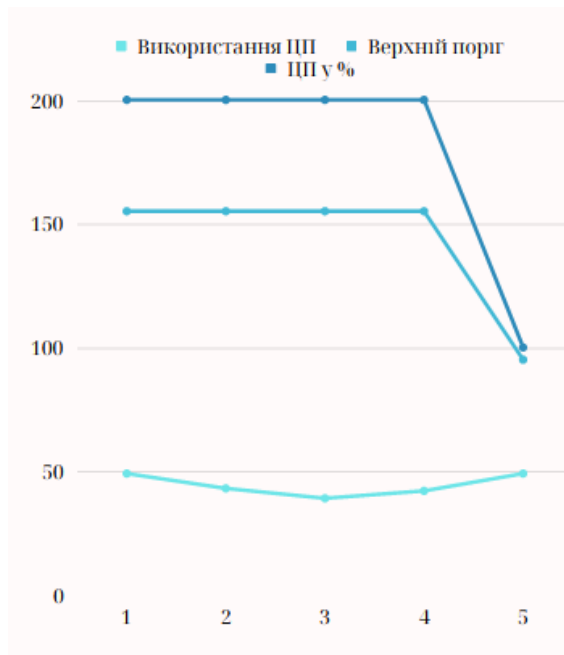


Рисунок 6 Масштабування ЦП донизу

Розширення моделі масштабування з використанням порогових значень за допомогою горизонтального масштабування

Після застосування алгоритму масштабування за допомогою порогових значень, досі існує імовірність того, що складна програмна система використає всі можливості вертикального масштабування і потрібно буде додавати додаткові екземпляри віртуальних машин і розділяти обчислювальні завдання поміж ними.

Відповідно до цього пропонується розширити алгоритм масштабування за допомогою порогових значень можливістю вертикального масштабування, як це показано на Рисунку 2.8. Якщо використання ресурсів віртуальної машини досягають максимального порогового значення протягом інтервалу $TMIsx$, якщо він більший максимального порогового інтервалу $TMIsxm$ - запускається горизонтальне масштабування і додається новий екземпляр віртуальної машини. Значення лічильнику інтервалу скидається. В іншому разі перевіряється значення мінімального порогового інтервалу $TMIsx$, якщо воно менше за мінімальний пороговий інтервал $TMIsxm$ – запускається горизонтальне масштабування і екземпляр віртуальної машини видаляється. Значення лічильнику інтервалу скидається.

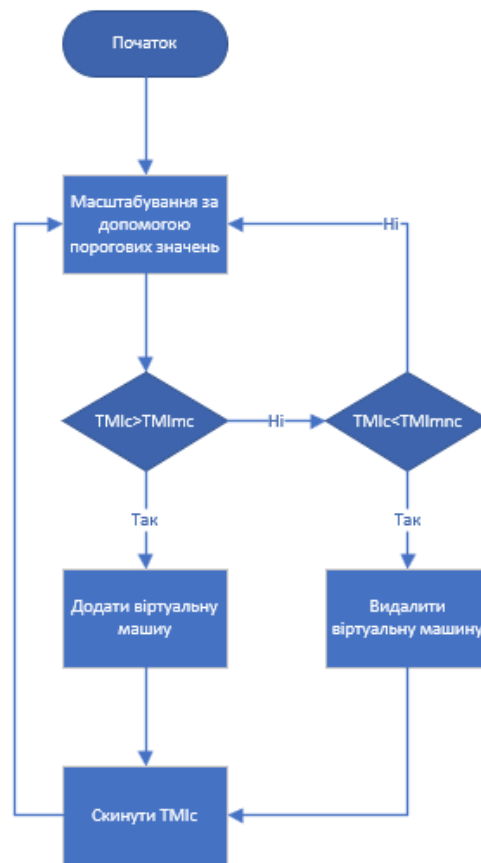


Рисунок 2.8 Вертикальне масштабування

Висновки

Застосування ефективних методів використання ресурсів може звести до мінімуму призвести втрату ресурсів. Автоматичне масштабування на основі порогового значення є одним із методів, при якому віртуальна машин динамічно масштабується відповідно до вимог програми до ресурсів, таким чином мінімізуючи використання ресурсів.

Вибір належних порогових значень є дуже важливим фактором успіху даного підходу. Нижче порогове значення призводить до частої зміни конфігурації віртуальної машини та більш високе значення знижує здатність віртуальної машини адаптуватися до нових вимог до ресурсів. Існує можливість використання декількох методів, щоб знайти оптимальні порогові значення. Наприклад, на основі історії, математичної моделі тощо.

Наразі система динамічного масштабування базується на пороговому значенні, коли порогові значення є статичними та попередньо визначеними.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Р. Мартін, Чиста архітектура. Харків, Україна : Фабула, 2021, 368 с. ISBN: 978-617-09-5286-8.
2. Ming Mao, Jie Li, Marty Humphrey (2011) T. S. J. Schwarz and E. L. Miller, "Cloud Auto-scaling with Deadline and Budget Constraints", Department of Computer Science University of Virginia Charlottesville, VA, USA 22904 {ming, jl3yh, humphrey}@cs.virginia.edu
3. Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla Segal "Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment", 2009 IEEE International Conference on e-Business Engineering.
4. XCP Design and Architecture [Електронний ресурс] – Режим доступу до ресурсу: http://wiki.xen.org/XCP_Design_and_Architecture

Гуменюк Олександр Володимирович – студент групи ІКІ-22м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: oleksandr.humeniuk.dev@gmail.com

Захарченко Сергій Михайлович – професор, кафедра обчислювальної техніки, Вінницький національний технічний університет, Вінниця