УДК 004.75

О. В. Гуменюк
С. М. Захарченко

ГОРИЗОНТАЛЬНЕ МАСШТАБУВАННЯ ХМАРНИХ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ ЗА ДОПОМОГОЮ ПОРОГОВИХ ЗНАЧЕНЬ

Вінницький національний технічний університет

Анотація

У роботі висвітлюється процес покращення горизонтального масштабування хмарних обчислювальних ресурсів за допомогою розробки процесу моніторингу використання ресурсів із впровадженням порогових значень. Даний процес допомагає підвищити економічну ефективність використання хмарних ресурсів. Демонструється принципова схема програмного комплексу для управління масштабуванням. Програмний комплекс підтримує моніторинг двох характеристик: обсягу оперативної пам'яті та ЦП і може масштабуватись вгору та вниз

Ключові слова: горизонтальне масштабування, хмарні ресурси, моніторинг

Abstract

The work highlights the process of improving the horizontal scaling of cloud computing resources by developing a process for monitoring the use of resources with the introduction of threshold values. This process helps to increase the economic efficiency of using cloud resources. The schematic diagram of the software complex for scaling control is demonstrated. The software complex supports monitoring of two characteristics: the amount of RAM and CPU and can scale up and down

Keywords: *horizontal scaling, cloud resources, monitoring*

Вступ

Економічна ефективність є однією з причин популярності хмарних сервісів. За рахунок ефективного використання ресурсів витрати можна ще більше зменшити, а втрати ресурсів можна мінімізувати. Вимоги до програмного забезпечення можуть відрізнятися в залежності від багатьох факторів (наприклад, навантаження на програму); користувач може запускати різні типи програм (від простого текстового редактора до складної програмної системи) у віртуальній машині. У таких випадках, якщо характеристики екземпляра віртуальної машини є статичними існує висока ймовірність невідповідності між даними характеристиками та вимогами програми до ресурсів. Якщо характеристики віртуальної машини більше, ніж вимоги до ресурсів програми, тоді ресурси будуть витрачені даремно; якщо параметри віртуальної машини менші за вимоги до ресурсів програми - продуктивність програми знизиться. Для вирішення цих проблем запропоноване автоматичне масштабування віртуальних машин на основі порогових значень, у яких віртуальні машини будуть динамічно масштабуватися на основі використання програмних ресурсів (ЦП і пам'ять).

Проектування системи масштабування пам'яті за пороговими значеннями

У хмарній парадигмі програмного забезпечення, інфраструктура та платформа надаються як послуги. Дана робота, розглядає інфраструктуру (віртуальні машини). Сервісні компанії надають віртуальні машини (VM) для кінцевого користувача. Користувач використовує екземпляр віртуальної машини для розміщення/запуску свого програмного забезпечення, і він заплатить певну суму відповідно до SLA

(Угода про рівень обслуговування)[1]. Багато організацій переходять до приватної хмари; ефективне використання ресурсів, зниження вартості та легке обслуговування є однією з причин для цього. Співробітники в організаціях отримують екземпляри віртуальних машин. Вони мають увійти в ці екземпляри, для того щоб використовувати їх. Незалежно від того, комерційна це хмара чи приватна, існує два можливих сценарії:

а) Користувач розміщує різні програми на віртуальній машині

Користувач може використовувати віртуальну машину для розміщення різних програм від простого текстового редактора до складних бухгалтерських програм. Якщо екземпляр віртуальної машини є статичним (зазвичай це так), користувач має вибрати екземпляр віртуальної машини таким чином, щоб відповідати максимальним потребам програми у ресурсах. У цьому випадку, якщо користувач використовує віртуальну машину для запуску програми, яка має максимальні потреби у ресурсах лише протягом 2 годин на день то протягом решти 22 годин ресурс буде витрачено даремно. Якщо вимога до ресурсів програми більша ніж кількість ресурсів віртуальної машини, то це призводить до деградації продуктивності програми [2].

б) Вимоги до програми змінюються з часом

Розглянемо програму бази даних, яка потребує більше ресурсів, коли транзакції відбуваються. Якщо транзакцій немає, це не потребує великих ресурсів. В випадку статичного екземпляру віртуальної машини, це призведе до втрати ресурсів. Для вирішення даної проблеми можливе перенесення програми з однієї віртуальної машини на іншу, але воно має багато недоліків. Це забирає багато часу, утомливо, економічно не ефективно і підвищує імовірність виникнення помилок. Якщо VM динамічно масштабується відповідно до вимог програми втрата ресурсів може бути мінімізована.

Для вирішення вищезазначених проблем було розроблено та перевірено поріг

на основі механізму автоматичного масштабування віртуальної машини, у якому віртуальна машина автоматично налаштовується відповідно до вимог програми. Під час автоматичного масштабування використання ресурсу на основі порогового значення віртуальної машини відстежується. Якщо значення перевищують попередньо визначені порогові значення, то характеристики віртуальної машини будуть збільшуватися або зменшуватися динамічно без її вимкнення відповідно до потреб, що мінімізує втрату ресурсів.

У даній роботі розглядається використання оперативної пам’яті та процесора віртуальної машини. Коли збільшується потреба програми в ресурсах, завантаження оперативної пам’яті та ЦП VM збільшується. У якийсь момент потреба в ресурсах програми стане більшою в порівнянні з потужністю віртуальної машини в результаті продуктивність програми деградує і зрештою вона перестане працювати. Щоб уникнути цієї проблеми, коли використання процесора і пам’яті віртуальної машини перевищує попередньо визначене максимальне порогове значення автоматично збільшується ємність оперативної пам’яті та процесора віртуальної машини.

Коли потреба програми в ресурсах зменшиться, відповідно зменшиться потреба в оперативній пам’яті і процесорі. Це призведе до втрати ресурсів віртуальної машини коли потужність використовується не повністю. Щоб уникнути втрати ресурсів, коли використання ЦП і пам'яті віртуальної машини перевищує заздалегідь визначене мінімальне порогове значення, необхідно обсяг оперативної пам’яті та процесора віртуальної машини. Моніторинг і масштабування оперативної пам'яті і ЦП віртуальної машини є двома незалежними завданнями [3].

Вимоги до програми можуть змінюватися з часом, а також користувач може запускати різні програми (які мають інші вимоги до ресурсів) на віртуальній машині. У цих випадках фіксована ємність віртуальної машини може призвести до втрати ресурсів або деградації продуктивності програми. Це можна вирішити шляхом динамічного масштабування віртуальної машини відповідно до вимог до розміщеної програми. Під час автоматичного масштабування ресурсу на основі порогового значення відстежується використання віртуальної машини. Якщо показники перевищують попередньо встановлені порогові значення, тоді ємність VM буде динамічно збільшуватися або зменшуватися відповідно до потреб без вимикання віртуальних машин, що мінімізує втрату ресурсів [4]. Загальний вигляд системи зображений на Рисунку 1

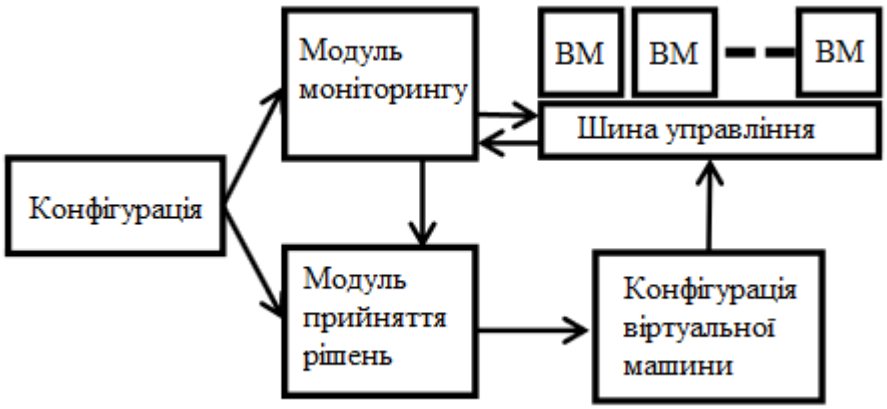

Рисунок 1 Загальний вигляд системи масштабування

Система масштабування складається із наступних компонентів:

- Модуль моніторингу - відстежує віртуальні машини; він зчитує використання ЦП і пам'яті і передає ці дані компоненту прийняття рішень. Він використовує шину управління для отримання значень ЦП і використання пам'яті віртуальних машин. За замовчуванням він відстежує всі активні віртуальні машини або є можливість контролю лише певних віртуальних машин, налаштувавши відповідні значення в конфігурації. Інтервал часу для надсилання запиту шині даних для отримання статистики віртуальних машин можна налаштувати в конфігурації.

Коли модуль моніторингу запускається, він зчитує всі значення конфігурації з конфігураційного файлу і відстежує віртуальні машини відповідно до даних значень.

- Модуль прийняття рішень - отримує статистику віртуальної машини з модуля моніторингу, а також читає порогові значення з файлу конфігурації, порівнює їх із статистикою віртуальної машини та вирішує, чи потрібно масштабувати віртуальну машину вгору/вниз, і передає це рішення модуль конфігурації віртуальної машини. Інформація, що передається до модуля конфігурації віртуальної машини містить ідентифікатор віртуальної машини, яку необхідно масштабувати, тип масштабування: оперативна пам’ять ЦП і кількість необхідних ресурів. Існує ймовірність того, що використання процесора та оперативної пам’яті віртуальної машини може перевищувати порогове значення протягом кількох секунд і знову повернутися до нормальних значень. Якщо модуль монітора отримує ці значення, він ініціює збільшення/зменшення масштабу оперативної пам’яті/ЦП. У наступній ітерації модуль монітора знову отримує нормальні значення та ініціює зменшення/збільшення розміру оперативної пам’яті/ЦП віртуальної машини, що призводить до непотрібних операцій масштабування віртуальних машин. Для уникнення цієї проблеми, запроваджено конфігураційні значення, які називаються cpuiteration (min і max) і memoryiteration (min і max). Будь-яке додатне ціле число від 0 до n може бути встановлено для обчислення та збереження. Модуль прийняття рішень ініціює збільшення/зменшення ресурсів, лише якщо використання оперативної пам’яті та ЦП віртуальної машини перевищує порогові значення в послідовній кількості ітерацій, вказаних у конфігурації.

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 1КІ-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