



Савчук Т. О.,
Козачук А. В.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ МАСШТАБУВАННЯ ХМАРНОГО ЗАСТОСУНКУ ЗІ ЗМІННИМИ ПІКАМИ НАВАНТАЖЕННЯ

В статті розв'язується задача розробки інформаційної технології масштабування хмарного застосунку на основі методів реактивного та проактивного масштабування. Описується взаємодія між ключовими компонентами технології, а також їхня внутрішня будова. Проведене дослідження ефективності використання розробленої технології показує, що її застосування є доцільнішим, ніж використання стандартного підходу на базі реактивних правил.

Ключові слова: хмарні обчислення, PaaS, масштабування хмарного застосунку.

1. Вступ

Однією з основних переваг використання хмарних обчислень є можливість швидкої адаптації до зміни кількості користувачів шляхом масштабування хмарного застосунку, що дозволяє виділяти обчислювальні ресурси лише тоді, коли в них є необхідність. Сучасні системи автоматичного масштабування хмарного застосунку діють на основі правил реактивного масштабування, що передбачає ініціалізацію процесу масштабування після того, як певна метрика, наприклад завантаженість процесора, досягла критичного значення. Даний підхід є ефективним в цілому, але у випадку, коли для хмарного застосунку характерні короткі та часті піки навантаження, можуть спостерігатися проблеми в роботі застосунку в проміжку часу між початком процесу масштабування та виділенням обчислювальних ресурсів. Цей недолік може бути усунений шляхом прогнозування інтенсивності використання хмарного застосунку та завчасному виділенню обчислювальних ресурсів.

Таким чином, актуальною є задача розробки інформаційної технології масштабування хмарного застосунку, що використовує прогноз його стану при прийнятті рішень щодо масштабування, скорочуючи час реакції на зміну навантаження та зменшуючи проміжок часу, протягом якого можуть спостерігатися збої в роботі хмарного застосунку.

2. Аналіз існуючих рішень

Розглянемо основні платформи та системи, що використовуються з хмарними застосунками та підтримують можливість масштабування. Кожна платформа розміщення хмарних застосунків має набір засобів отримання телеметрії та масштабування, які не сумісні з іншими платформами. Тому порівняння ефективності роботи технологій масштабування доцільно проводити в рамках однієї платформи. Найбільш популярними середовищами типу «Платформа як послуга» (PaaS) є Amazon AWS Elastic Beanstalk [1, 2], Google App Engine [3], Microsoft Azure [4, 5]. Усі перераховані платформи можуть забезпечувати високу доступність при роботі хмарного застосунку за рахунок географічного розподілення сер-

верів. При виборі конкретної платформи використовуються такі критерії, як вартість хостингу та додаткові можливості платформи. Враховуючи такі особливості, як простота розгортання та можливість програмного масштабування хмарних застосунків, що використовують технологію Microsoft Azure App Service (Web apps), інформаційну систему автоматизованого масштабування хмарного застосунку доцільно реалізовувати саме в контексті цієї технології.

До існуючих систем, які підтримують автоматичне масштабування хмарного застосунку, що розміщується на платформі Azure Web apps відносяться портал Microsoft Azure [6] та систему автоматизації CloudMonix [7]. Обидві системи дозволяють налаштовувати реактивні правила масштабування, а також підтримують масштабування за розкладом.

3. Об'єкт, мета та задачі дослідження

Об'єкт дослідження — процес масштабування хмарного застосунку.

Метою дослідження є розробка інформаційної технології масштабування хмарного застосунку, що дозволить підвищити ефективність його функціонування.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Розробити архітектуру інформаційної технології масштабування хмарного застосунку.
2. Провести дослідження ефективності роботи хмарного застосунку із застосуванням розробленої інформаційної технології та порівняти результати з аналогами.

4. Архітектура інформаційної технології

Вибір стратегії масштабування хмарного застосунку є комплексним процесом, що складається з декількох етапів, на яких відбувається уточнення режиму роботи хмарного застосунку, прогнозування кількості мережевих запитів, що надходять до застосунку, а також вибір оптимального варіанту масштабування. Перераховані етапи з'єднуються між собою інформаційними потоками, що містять вхідні та вихідні дані кожного з етапів вирішення задачі масштабування хмарного застосунку.

Для систематизації масштабування хмарного застосунку етапи цього процесу доцільно об'єднати у технологічний ланцюжок, що дозволяє розглядати їх як інформаційну технологію [8].

У відповідності з класичним визначенням [9, 10] інформаційної технології, інформаційну технологію масштабування хмарного застосунку можна представити як сукупність процесів класифікації поточного режиму роботи хмарного застосунку, прогнозування його стану за допомогою методів прогнозування часових рядів, прийняття рішення щодо необхідності проведення масштабування на основі методів реактивного та проактивного масштабування, об'єднані в технологічний ланцюжок з метою підвищення ефективності функціонування хмарного застосунку. Тоді інформаційна технологія масштабування хмарного застосунку складається з регулярного повторення наступних етапів:

1. Визначення поточного стану завантаження хмарного застосунку, а саме — кількість запитів за хвилину, завантаження процесора та пам'яті, середній час виконання мережевого запиту, кількість та тип віртуальних машин. Дані стан записується в базу даних статистичної інформації про роботу хмарного застосунку.

2. Поповнення службових структур даних, необхідних для прогнозування стану хмарного застосунку та прийняття рішення щодо вибору варіанту масштабування зібраною статистичною інформацією.

3. Прогнозування кількості мережевих запитів, яке складається з наступних під-етапів:

- 3.1. Проведення класифікації поточного режиму роботи хмарного застосунку.

- 3.2. Здійснення прогнозування кількості вхідних мережевих запитів за допомогою моделі прогнозування часових рядів, обраної в залежності від результатів класифікації.

4. Прийняття рішення про проведення масштабування, складається з таких під-етапів:

- 4.1. Перевірка правил реактивного масштабування хмарного застосунку.

- 4.2. Вибір варіанту масштабування за допомогою алгоритму проактивного масштабування.

- 4.3. Обробка результатів методів реактивного та проактивного масштабування за допомогою алгоритму прийняття рішень щодо масштабування хмарного застосунку.

5. Здійснення масштабування хмарного застосунку, якщо було прийняте відповідне рішення. Після завершення масштабування — оновлення інформації

про ваги дуг в графі станів інфраструктури хмарного застосунку.

Перед початком використання необхідно провести розгортання інформаційної технології, а саме створення структур даних та введення початкових даних про правила реактивного масштабування та граф можливих станів хмарного застосунку.

Загальна схема інформаційної технології масштабування хмарного застосунку зображена на рис. 1.

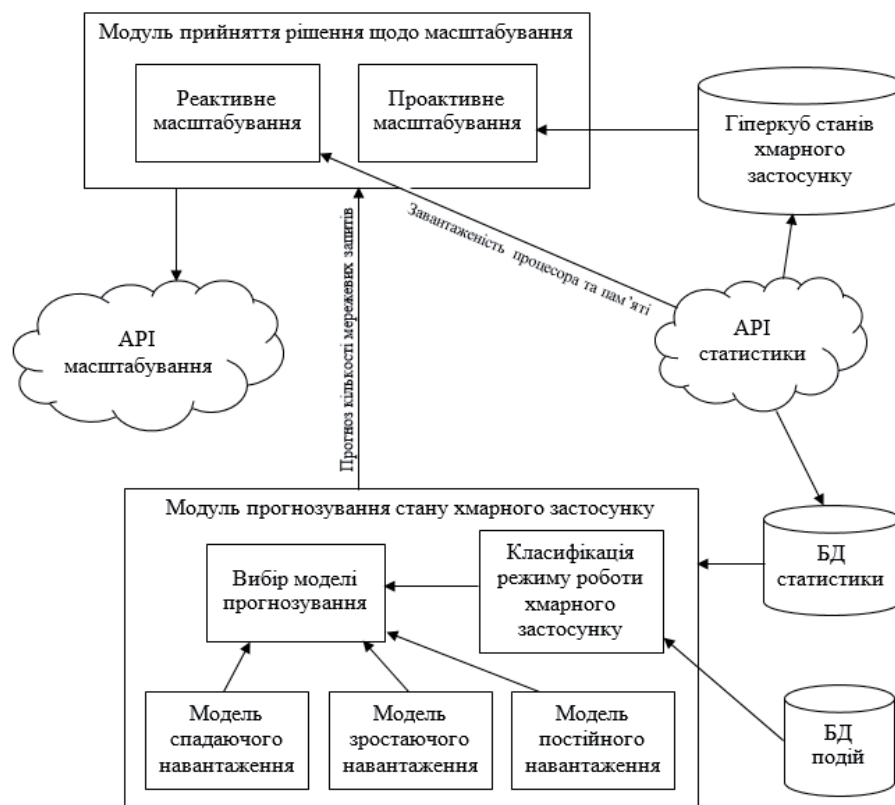


Рис. 1. Архітектура інформаційної технології масштабування хмарного застосунку

Інформаційна технологія масштабування хмарного застосунку повинна надавати можливість легкого розгортання для вже функціонуючих застосунків. Для цього необхідно максимально ізолювати роботу технології від внутрішньої логіки хмарного застосунку. Таку ізоляцію можна забезпечити шляхом виділення структур даних хмарного застосунку в окрему логічну одиницю, а також шляхом групування усіх етапів інформаційної технології у модуль масштабування, який є складовою хмарного застосунку та взаємодіє лише з API управління хмарним застосунком, але не з іншими його частинами.

Модуль масштабування хмарного застосунку може бути виділений в якості окремого хмарного застосунку, що підвищує його ізолюваність та позбавляє основний застосунок від паразитного споживання обчислювальних ресурсів. Проте такий підхід вимагає додаткових витрат на утримання окремого хмарного застосунку, а також може призвести до проблем з безпекою, так як потребує надання прав керувати одним хмарним застосунком з іншого. Схема взаємодії модуля масштабування з іншими частинами хмарного застосунку наведена на рис. 2.

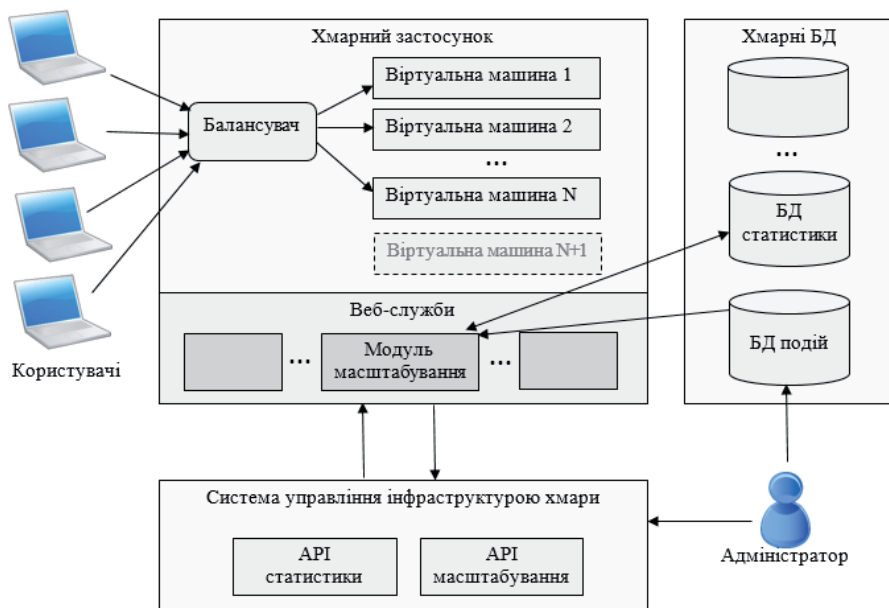


Рис. 2. Взаємодія модуля масштабування з іншими частинами хмарного застосунку

4.1. Модуль прогнозування кількості мережевих запитів до хмарного застосунку. Модуль прогнозування кількості мережевих запитів, що надходять до хмарного застосунку приймає в якості вхідних даних список подій, що впливають на роботу хмарного застосунку та часовий ряд, що представляє історію частоти мережевих запитів. Результатом роботи модуля є прогноз кількості мережевих запитів.

Модуль включає в себе такі складові:

- блок класифікації режиму роботи хмарного застосунку, що реалізує алгоритм класифікації роботи хмарного застосунку;
- реалізацію моделей прогнозування часового ряду кількості мережевих запитів до хмарного застосунку;
- механізм вибору методу прогнозування на основі режиму роботи хмарного застосунку;
- блок виконання прогнозування кількості мережевих запитів, що надходять до хмарного застосунку.

Потоки даних між компонентами модуля прогнозування кількості мережевих запитів наведені на рис. 3. Вхідними даними є часовий ряд мережевих запитів достатньої довжини для застосування прогнозування за допомогою регресії, а також час та тривалість попередньої та наступної подій, що можуть вплинути на роботу хмарного застосунку. Вихідні дані модуля являють собою число – прогноз кількості мережевих запитів.

Для коректного функціонування модуль також повинен включати сховище даних для таких налаштувань, як горизонт

прогнозування, список моделей прогнозування та критерії вибору тієї чи іншої моделі прогнозування в залежності від режиму роботи хмарного застосунку. Так як кількість даних для налаштування роботи модулю є невеликою та вони не потребують частого редагування, доцільно зберігати налаштування модуля у виділеному файлі файлової системи.

4.2. Модуль прийняття рішень щодо масштабування. Модуль прийняття рішень щодо масштабування хмарного застосунку складається з блоків, які забезпечують виконання алгоритму масштабування. В процесі функціонування модуль взаємодіє з такими об'єктами, як база даних правил реактивного масштабування, гіперкуб станів хмарного застосунку, що

містить інформацію про час виконання мережевого запиту при різних станах інфраструктури хмарного застосунку, функція розподілу імовірності втрати користувача в залежності від часу виконання мережевого запиту та відношення загальної кількості мережевих запитів до кількості користувачів.

Для забезпечення коректної неперервної роботи інформаційної технології вказані джерела даних повинні постійно оновлюватися на основі статистики роботи хмарного застосунку. Також на роботу модуля впливають налаштування пошуку інформації у гіперкубі станів хмарного застосунку, що регулюють визначення часу виконання мережевих запитів в умовах невизначеності.

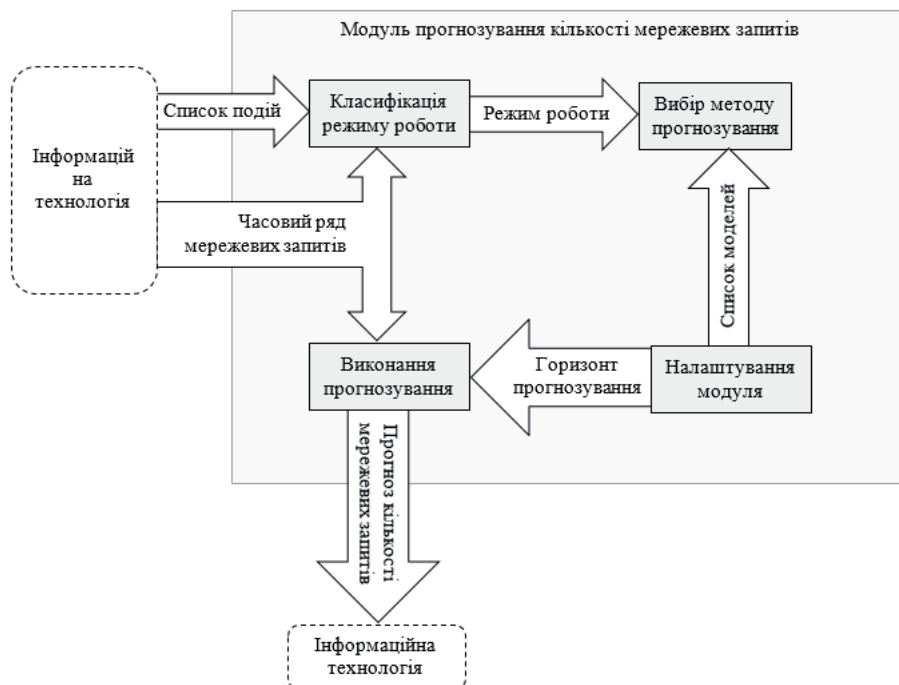


Рис. 3. Обмін даними між блоками модуля прогнозування кількості мережевих запитів

Модуль передбачає формування рішень щодо проведення масштабування блоками реактивного та проактивного масштабування з подальшим порівнянням отриманих значень та прийняттям остаточного рішення щодо масштабування на їх основі. Результатом роботи модуля є рішення щодо проведення масштабування $\langle S', \Delta N \rangle$, де S' – розмір віртуальної машини після масштабування, ΔN – приріст кількості віртуальних машин відносно стану інфраструктури хмарного застосунку до початку масштабування. Потіки даних між компонентами модуля масштабування хмарного застосунку наведені на рис. 4.

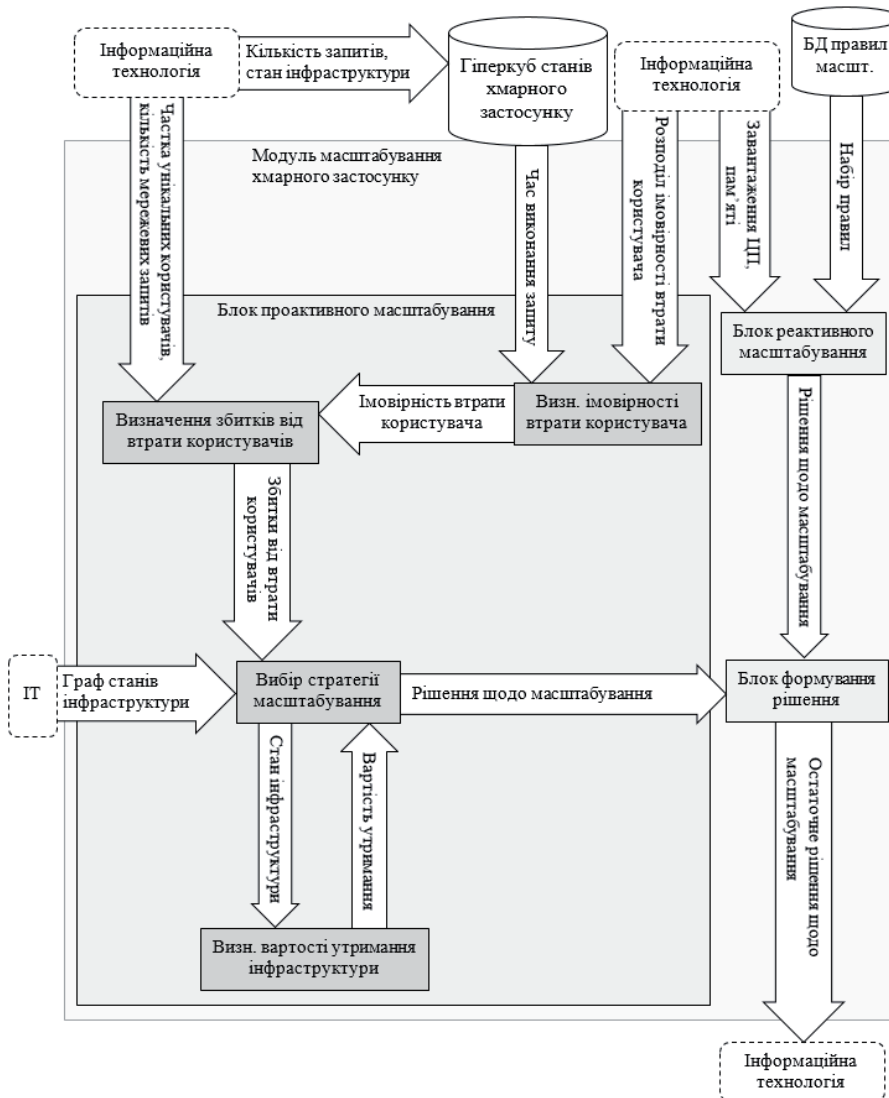


Рис. 4. Обмін даними між блоками модуля масштабування хмарного застосунку

4.3. Вибір цільової платформи використання. Розглянемо платформи розміщення хмарних застосунків, на базі яких може бути використана інформаційна система масштабування хмарного застосунку з періодичними піками навантаження. До найбільш популярних середовищ типу «Платформа як послуга» (PaaS) належать Amazon AWS Elastic Beanstalk [1, 2], Google App Engine [3], Microsoft Azure [4, 5]. Усі перераховані платформи можуть забезпечувати високу доступність при роботі хмарного застосунку за рахунок географічного розподілення серверів. При виборі конкретної платформи використову-

ються такі критерії, як вартість хостингу та додаткові можливості платформи. З точки зору використання інформаційної технології масштабування хмарного застосунку необхідно розглянути можливості платформи по постачанню актуальної статистичної інформації про стан роботи хмарного застосунку та можливостей програмного інтерфейсу (API) масштабування.

Порівняння швидкості та питомої вартості платформ [11] не виявило однозначного лідера – результати суттєво відрізнялись в залежності від конкретного тесту продуктивності, тому цей критерій не може бути використаний при виборі цільової платформи використання інформаційної системи масштабування хмарного застосунку за періодичними піками навантаження.

Google App Engine підтримує створення застосунків з використанням мов програмування Java, Python, PHP та Go. Розробка в межах цієї платформи ускладнюється необхідністю використання вузькоспеціалізованих рішень, які не працюють за межами платформи [12]. Платформа підтримує три режими масштабування: ручне, при якому кількість віртуальних машин задається користувачем в файлі конфігурації [13]; базове – при наявності запитів від користувачів активується віртуальна машина із застосунком, при відсутності запитів – хмарний застосунок переводиться в режим очікування; автоматичне – масштабування здійснюється за допомогою вбудованого модуля, що використовує час обробки мережевого запиту та частоту запитів для прийняття рішення щодо масштабування. У зв'язку з необхідністю використовувати жорстко прив'язані до платформи технології та незручністю автоматичного користувацького масштабування, використання Google App Engine в якості цільової платформи для інформаційної платформи масштабування є недоцільним.

Amazon AWS, зокрема служба Elastic Beanstalk дозволяє розміщувати застосунки, розроблені за допомогою Java, Node.js, PHP, Python, Ruby, та ASP.NET. При цьому платформа забезпечує автоматичне оновлення програмного забезпечення та балансування навантаження між віртуальними машинами. Розгортання хмарного застосунку може бути здійснене з репозиторія GIT або з таких середовищ розробки, як MS Visual Studio та Eclipse за допомогою спеціальних плагінів. Платформа Elastic Beanstalk підтримує автоматичне масштабування хмарних застосунків та масштабування за розкладом, проте не надає програмний інтерфейс для проведення

масштабування, що робить її непридатною для використання з запропонованою інформаційною технологією масштабування хмарного застосунку.

Більш низькорівневе рішення від Amazon – Elastic Compute Cloud (EC2) має гнучку систему об'єднання віртуальних машин в групи масштабування [14], проте EC2 ближча до IaaS, ніж до PaaS та вимагає виконання значних обсягів операцій для розгортання хмарного застосунку.

Платформа Microsoft Azure надає можливість розміщувати хмарні застосунки, використовуючи три технології: Azure App Service (Web apps), Azure Cloud Services (Web roles) та віртуальні машини [15]. Остання технологія відноситься до рівня IaaS та вимагає значних адміністраторських робіт по розгортанню та підтримці функціонування хмарного застосунку. Розглянемо більш детально різницю між Azure Web apps та Azure Web roles, обидві з яких є PaaS технологіями.

Web apps надає можливість швидкого розгортання хмарного застосунку та зручного запуску планових задач за допомогою Azure WebJobs [16], в той же час дана технологія не підтримує деякі застарілі можливості ASP.NET та складні налаштування середовища виконання коду. З точки зору масштабування, Web apps надає зручний графічний інтерфейс та можливість автоматизованого масштабування за допомогою спеціальної утиліти. При масштабуванні не потрібно проводити повторну конфігурацію та розгортання хмарного застосунку, що важливо при наявності піків навантаження. Microsoft Azure Web apps дозволяє розміщувати ASP.NET та PHP застосунки, а також надає доступ до Java та Ruby SDK.

Web roles дозволяє проводити складні налаштування середовища, віддалено під'єднуватися до серверів, автоматично запускати задачі при старті сервера та використовувати необмежену кількість віртуальних машин. В той же час масштабування хмарного застосунку вгору вимагає його повторного розгортання, що значно ускладнює процедуру.

PaaS платформи можна розділити на низькорівневі (Microsoft Azure Web roles, Amazon EC2), що дозволяють проводити детальне налаштування середовища, в тому числі – проводити гнучке масштабування через програмні інтерфейси, проте вимагають відносно великих затрат на розгортання та підтримку, та високорівневі (Google App Engine, Amazon Elastic Beanstalk, Microsoft Azure Web apps), що дозволяють проводити швидко розгортання та мають вбудований механізм автоматичного масштабування.

Враховуючи такі особливості, як простота розгортання та можливість програмного масштабування хмарних застосунків, що використовують технологію Microsoft Azure App Service (Web apps), інформаційну систему автоматизованого масштабування хмарного застосунку доцільно реалізовувати саме в контексті цієї технології.

5. Розробка модулів прогнозування та масштабування

Для реалізації алгоритму модуля прогнозування необхідно забезпечити можливість отримання часового ряду мережеских запитів, що надходять до хмарного застосунку, реалізувати алгоритми класифікації стану

роботи хмарного застосунку та прогнозування кількості мережеских запитів.

Розглянемо можливі джерела інформації про кількість мережеских запитів, які надходять до хмарного застосунку, що працює розгорнутий на базі Microsoft Azure Web apps:

- Microsoft Azure Application insight [17] – служба по збору телеметрії роботи хмарного застосунку, потребує модифікації коду хмарного застосунку для початку роботи. Статистика кількості мережеских запитів доступна через портал Microsoft Azure. Недоліком підходу є новизна технології та відсутність API збору статистики, що ускладнює задачу автоматичного аналізу статистики.

- Continuous export [18] – опція, що надає можливість експорту телеметрії Application Insight у blob сховище, до якого може мати доступ модуль прогнозування кількості мережеских запитів. Для цього підходу характерна значна затримка отримання даних через накладання затримок від збору телеметрії через Application Insight, накопичення даних перед експортом та читання експортованих даних. Дослідження показують, що затримка може сягати кількох хвилин, що перевищує інтервал між вимірюванням кількості мережеских запитів та може негативно вплинути на ефективність роботи інформаційної технології.

- Performance Counters – стандартний механізм збору статистики в ASP.NET при класичному, не хмарному, має обмежену область застосування у Azure Web apps через розподілену архітектуру сервісу [15], виміряти кількість мережеских запитів таким чином неможливо.

- Бібліотека Azure Web Sites Management [19] – доступна у вигляді NuGet-пакегу та дозволяє отримувати інформацію про поточний стан хмарного застосунку та здійснювати операції по керуванню хмарним застосунком через .net-середовище. Використання бібліотеки потребує додаткових дій щодо адміністрування хмарного застосунку, які полягають у генерації криптографічного ключа доступу на рівні хмарного застосунку для збірки, що буде використовувати бібліотеку. Після підключення ключа бібліотека дає можливість отримувати статистику кількості мережеских запитів з інтервалом до однієї хвилини.

Таким чином, для отримання статистичної інформації про роботу хмарного застосунку, що розміщується у Microsoft Azure варто використовувати Azure Web Sites Management, так як ця бібліотека забезпечує легкий доступ до актуальної і детальної статистичної інформації.

Для забезпечення більш тісної інтеграції з платформою Microsoft Azure варто використовувати технологію .NET для реалізації методів прогнозування часових рядів, так як саме ця технологія є основою для платформи. .NET не містить вбудованих алгоритмів прогнозування часових рядів, а існуючі CLR бібліотеки не реалізують необхідний набір функцій. Проте існують бібліотеки, що забезпечують можливість виклику з .NET-коду спеціалізованих середовищ аналізу даних, таких як Matlab [20] та статистичної мови програмування R [21]. Для реалізації алгоритмів прогнозування було обрано бібліотеку R.NET, так як вона є безкоштовною для комерційного використання.

Список подій, що можуть вплинути на завантаженість хмарного застосунку має досить просту структуру і у більшості випадків використовується у режимі читання, тому для його збереження використаємо плоский відсортований csv-файл, так як він простіший у реалізації в порівнянні з реляційними і словниковими базами даних. Налаштування модуля прогнозування кількості мережових запитів, а саме горизонт прогнозування та опис моделей прогнозування мовою R також доцільно зберігати у файловій системі.

Статистичні дані про стан хмарного застосунку варто зберігати у реляційній базі даних, так як можливе інтенсивне двостороннє використання даних у режимі читання та запису, а також часті операції селекції, виконання яких може бути прискорене завдяки використанню індексів. Платформа Microsoft Azure має тісну інтеграцію з Microsoft SQL Server, зокрема дозволяє розміщувати бази даних безпосередньо в хмарі, без додаткового шару у вигляді віртуальної машини [22], тому доцільно використовувати саме Microsoft SQL Server для розміщення реляційних баз даних. У базі даних статистичної інформації слід зберігати поточну мітку часу, кількість мережових запитів у хвилину, середній час виконання мережового запиту.

Перераховані дані доцільно зберігати у одній таблиці.

Модуль масштабування хмарного застосунку складається з блоків проактивного та реактивного масштабування, а також з блоку прийняття рішень, реалізованого відповідним алгоритмом.

Модуль реактивного масштабування включає в себе класи, що відповідають за опис правил масштабування, завантаження списку правил та формування логічних функцій, що дозволяють перевірити умови виконання правила, перевірки істинності умов застосування правил масштабування.

Для реалізації блока проактивного масштабування були розроблені класи, що забезпечують визначення імовірності втрати користувача, прогнозування часу виконання мережового запиту, керування списком подій, що пов'язані з роботою хмарного застосунку, визначення вартості утримування інфраструктури хмарного застосунку та втрати користувачів.

Весь програмний код, що реалізує модуль масштабування хмарного застосунку об'єднаний в .NET-збірку ScalingStrategies.dll.

Взаємодія цієї збірки з іншими компонентами системи масштабування хмарного застосунку зображена на рис. 5.

Таким чином, на базі технології .NET Framework було розроблено модулі прогнозування кількості мережових запитів та масштабування хмарного застосунку, що на основі даних про використання хмарного застосунку, отриманих за допомогою бібліотеки Azure Web Sites Management та бази даних подій, пов'язаних з піками навантаження дозволяють вибрати стратегію масштабування хмарного застосунку.

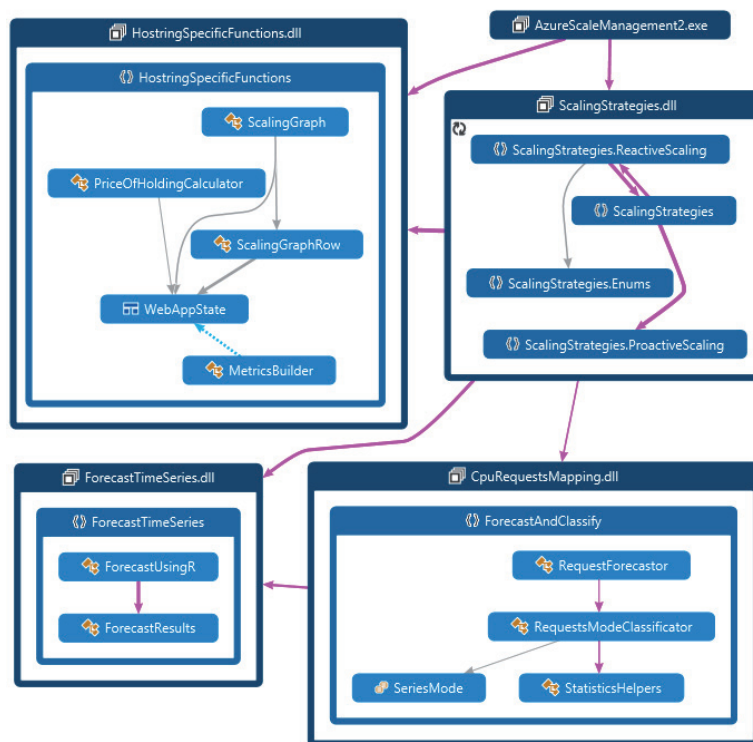


Рис. 5. Взаємодія компонентів системи масштабування хмарного застосунку

6. Дослідження ефективності застосування інформаційної технології масштабування хмарного застосунку

За допомогою симулятора роботи хмарного застосунку було проведено визначення ефективності роботи технології масштабування хмарного застосунку в умовах інтенсивних пікових навантажень та в умовах штатної роботи з періодичними піковими навантаженнями. Також була визначена ефективність масштабування за допомогою реактивних правил. Визначення ефективності проводилося шляхом розрахунку значення критерія ефективності функціонування хмарного застосунку (2.11), що відповідає сумарним витратам від втрат користувачів через занадто довге виконання запитів та затрат на підтримку інфраструктури хмарного застосунку. Для побудови профілів тестів навантаження використовувались дані відвідуваності сайту чемпіонату світу з футболу [23].

Порівняння ефективності функціонування хмарного застосунку із використанням різних технологій масштабування зображено на рис. 6.

Стратегія реактивного масштабування включала в себе такі правила:

- збільшення кількості віртуальних машин на 1, якщо завантаження ЦП більше 75 %;
- зменшення кількості віртуальних машин на 1, якщо завантаження ЦП менше 10 %;
- збільшення кількості віртуальних машин на 1, якщо завантаження пам'яті більше 90 %;
- збільшення розміру віртуальної машини на 1, якщо час виконання запиту більший за 5 с;
- збільшення розміру віртуальної машини на 1, якщо завантаження ЦП більше 70 %;
- зменшення розміру віртуальної машини на 1, якщо завантаження ЦП менше 15 %.

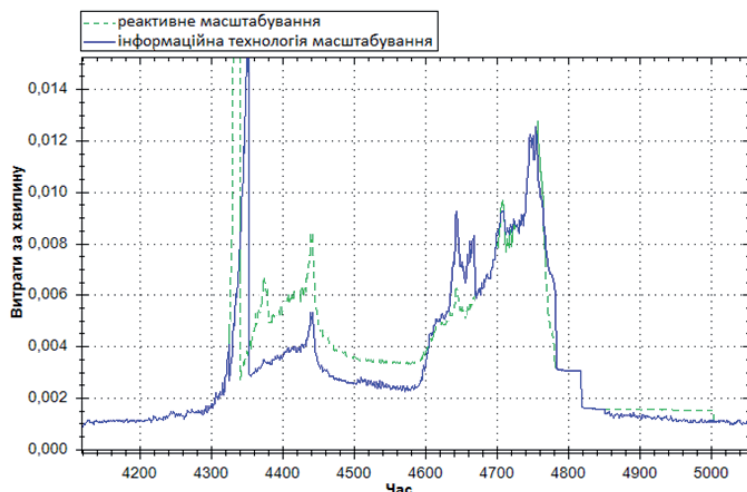


Рис. 6. Порівняння вартості утримання хмарного застосунку під час інтенсивних піків навантаження

та 2,75 (0,0039 за хвилину) для інформаційної технології масштабування хмарного застосунку.

Таким чином, використання інформаційної технології масштабування хмарного застосунку дозволяє зменшити експлуатаційні витрати на 8 % в режимі штатного використання та на 12 % в режимі інтенсивних піків навантаження. Результати інших тестів наведені в табл. 1.

Як видно з табл. 1, застосування інформаційної технології масштабування хмарного застосунку збільшує ефективність використання хмарного застосунку, при чому в найбільшій мірі ефективність зростає під час піків навантаження. В середньому, використання інформаційної технології масштабування збільшує ефективність роботи хмарного застосунку на 8 %, а при високій частоті піків навантаження — до 28 %.

Таблиця 1

Порівняння результатів тестів навантаження

Опис тесту навантаження	Інформаційна технологія масштабування		Реактивне масштабування		Приріст ефективності з використанням інформаційної технології
	Значення критерію ефективності	Середнє за 1 хв.	Значення критерію ефективності	Середнє за 1 хв.	
Усі дані	12,51	0,00118	13,6	0,00125	8 %
Подвійний пік	2,75	0,0039	3,12	0,0044	12 %
Подвійний пік № 2	1,27	0,00253	1,77	0,0035	28 %
Малий подвійний пік	0,758	0,00094	0,74	0,00093	-2 %
Штатний високонавантажений режим	2,34	0,00086	2,34	0,00086	0 %
Одиничний пік	1,25	0,00313	1,67	0,00417	25 %
Один день	2,14	0,00149	2,56	0,00178	16 %
Штатний режим з низьким навантаженням	2,24	0,00075	2,24	0,00075	0 %

Правила створювалися з врахуванням того, що завантаження процесора є головним фактором сповільнення обробки мережевих запитів. Налаштування інформаційної технології масштабування хмарного застосунку включали в себе цей же набір правил реактивного масштабування, а також передбачали вартість втрати одного користувача рівною \$0,00027 (недоотриманий дохід від показу реклами Google AdSense [24]), кількість елементів часового ряду мережевих запитів, що використовується для прогнозування — 125, крок часового ряду мережевих запитів — 1 хв, горизонт прогнозування — 15 хв (вибраний як максимальне значення часу операції масштабування), мінімальний час між операціями масштабування — 25 хв. Початковий стан інфраструктури хмарного застосунку складав одну малу віртуальну машину.

Для тесту, що відтворює штатні умови з періодичними піками навантаження сумарне значення критерію ефективності склало 13,6 (0,00125 за хвилину) для реактивного масштабування та 12,51 (0,00118 за хвилину) для інформаційної технології масштабування хмарного застосунку.

Для тесту, що відтворює інтенсивні пікові навантаження сумарне значення критерію ефективності склало 3,12 (0,0044 за хвилину) для реактивного масштабування

7. Висновки

В результаті проведених досліджень була розроблена інформаційна технологія масштабування хмарного застосунку з періодичними піками навантаження, що базується на використанні методів реактивного та проактивного масштабування, яке здійснюється шляхом прогнозування кількості мережевих запитів, що надходять до хмарного застосунку. На базі інформаційної технології була розроблена система автоматичного масштабування хмарного застосунку, сумісна з платформою Microsoft Azure web apps.

Порівняння ефективності роботи хмарного застосунку з використанням розробленої інформаційної технології та з використанням стандартного методу масштабування на основі реактивних правил показало, що використання розробленої інформаційної технології збільшує ефективність використання хмарного застосунку на 8 %.

Література

1. Amazon EC2 [Electronic resource] // Amazon Web Services. — Available at: \www/URL: <http://aws.amazon.com/es/ec2/>
2. Jinesh, V. Overview of Amazon Web Services [Electronic resource] / V. Jinesh, M. Sajee // Amazon Web Services. —

- January 2014. — Available at: \www/URL: <http://www.fronde.com/assets/Datasheets/AWS-Overview.pdf>
3. Sanderson, D. Programming Google App Engine with Python: Build and Run Scalable Python Apps on Google's Infrastructure [Text] / D. Sanderson. — O'Reilly Media, Inc., 2015. — 464 p.
 4. Virtual Machine and Cloud Service Sizes for Azure [Electronic resource] // Microsoft Azure. — June 24, 2015. — Available at: \www/URL: <https://msdn.microsoft.com/en-us/library/azure/dn197896.aspx>. — 10.08.2015.
 5. Wilder, B. Cloud Architecture Patterns: Using Microsoft Azure [Text] / B. Wilder. — O'Reilly Media, Inc, 2012. — 182 p.
 6. Microsoft Azure [Electronic resource]. — Available at: \www/URL: <http://portal.azure.com/>
 7. CloudMonix [Electronic resource]. — Available at: \www/URL: <http://cloudmonix.com/>
 8. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы [Текст] / В. А. Гвоздева. — М.: Форум, 2011. — 544 с.
 9. Лихачева, Г. Н. Информационные технологии [Текст]: уч.-практ. пос. / Г. Н. Лихачева, М. С. Гаспарян. — М.: Изд. центр ЕАОИ, 2007. — 189 с.
 10. Корнеев, И. К. Информационные технологии [Текст]: уч. / И. К. Корнеев, Г. И. Ксандопуло, В. А. Адамович. — Проспект, 2009. — 224 с.
 11. Wayner, P. Ultimate cloud speed tests: Amazon vs. Google vs. Windows Azure [Electronic resource] / P. Wayner // InfoWorld. — Feb 26, 2014. — Available at: \www/URL: <http://www.infoworld.com/article/2610403/cloud-computing/ultimate-cloud-speed-tests--amazon-vs--google-vs--windows-azure.html>
 12. Разважаев, А. Технологии «хмарного» обчислення для застосування в інформаційних центрах [Текст] / А. Разважаев, О. Соловійов // Наукові праці Національної бібліотеки України ім. В. І. Вернадського. — 2014. — Вип. 40. — С. 226–236.
 13. Toffetti, G. Web Engineering for Cloud Computing [Text] / G. Toffetti // Current Trends in Web Engineering. — Springer Science + Business Media, 2012. — P. 5–19. doi:10.1007/978-3-642-35623-0_2
 14. Bellenger, D. Scaling in Cloud Environments [Electronic resource] / D. Bellenger, J. Bertram, A. Budina, A. Koschel, B. Pfänder, C. Serowy, I. Astrova, S. G. Grivas, M. Schaaf // Recent Researches in Computer Science. — 2011. — P. 145–150. — Available at: \www/URL: <http://www.wseas.us/e-library/conferences/2011/Corfu/COMPUTERS/COMPUTERS-23.pdf>
 15. Dykstra, T. Azure App Service, Cloud Services, and Virtual Machines comparison [Electronic resource] / T. Dykstra // Microsoft Azure. — 08.10.2015. — Available at: \www/URL: <https://azure.microsoft.com/en-us/documentation/articles/choose-web-site-cloud-service-vm/>
 16. Dykstra, T. Azure WebJobs documentation resources [Electronic resource] / T. Dykstra // Microsoft Azure. — 09.22.2015. — Available at: \www/URL: <https://azure.microsoft.com/en-us/documentation/articles/websites-webjobs-resources/>
 17. Wills, A. C. Get started with Visual Studio Application Insights [Electronic resource] / A. C. Wills // Microsoft Azure. — 10.05.2015. — Available at: \www/URL: <http://azure.microsoft.com/en-us/documentation/articles/app-insights-get-started/>
 18. Fritz, A. Export telemetry from Application Insights [Electronic resource] / A. Fritz // Microsoft Corporation. — 11 Dec 2014. — Available at: \www/URL: <http://blogs.msdn.com/b/visualstudioalm/archive/2014/12/11/export-telemetry-from-application-insights.aspx>
 19. Gaster, B. Getting Started with the Windows Azure Management Libraries for.NET [Electronic resource] / B. Gaster // Website of Brady Gaster. — 22 October 2013. — Available at: \www/URL: <http://www.bradygaster.com/post/getting-started-with-the-windows-azure-management-libraries>
 20. MATLAB Compiler SDK [Electronic resource] // The MathWorks, Inc. — Available at: \www/URL: <http://www.mathworks.com/products/matlab-compiler-sdk/>
 21. R.NET [Electronic resource] // CodePlexProject Hosting for Open Source Software. — Available at: \www/URL: <http://rdotnet.codeplex.com/>
 22. Curino, C. Relational Cloud: A Database-as-a-Service for the Cloud [Electronic resource] / C. Curino, E. P. C. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, N. Zeldovich // 5th Biennial Conference on Innovative Data Systems Research, CIDR 2011, January 9-12, 2011, Asilomar, California. — Massachusetts Institute of Technology, 2011. — P. 235–240. — Available at: \www/URL: http://www.cidrdb.org/cidr2011/Papers/CIDR11_Paper33.pdf
 23. WorldCup98 [Electronic resource] // The Internet Traffic Archive. — Available at: \www/URL: <http://ita.ee.lbl.gov/html/contrib/WorldCup.html>
 24. Mohan, M. How Much Traffic Do You Need To Make \$100,000 With Google AdSense [Electronic resource] / M. Mohan // Minterest. — April 28, 2014. — Available at: \www/URL: <http://www.minterest.org/how-much-traffic-do-you-need-to-make-money/>

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ МАСШТАБИРОВАНИЯ ОБЛАЧНОГО ПРИЛОЖЕНИЯ С ПЕРЕМЕННЫМИ ПИКАМИ НАГРУЗКИ

В статье решается задача разработки информационной технологии масштабирования облачного приложения на основе методов реактивного и проактивного масштабирования. Описывается взаимодействие между ключевыми компонентами технологии, а также их внутреннее строение. Проведенное исследование эффективности использования разработанной технологии показывает, что ее применение является более целесообразным, чем использование стандартного подхода на базе реактивных правил.

Ключевые слова: облачные вычисления, PaaS, масштабирование облачного приложения.

Савчук Тамара Олександрівна, кандидат технічних наук, професор, кафедра комп'ютерних наук, Вінницький національний технічний університет, Україна.

Козачук Андрій Валерійович, асистент, кафедра комп'ютерних наук, Вінницький національний технічний університет, Україна, e-mail: kozachuk35@rambler.ru.

Савчук Тамара Александровна, кандидат технических наук, профессор, кафедра компьютерных наук, Винницкий национальный технический университет, Украина.

Козачук Андрей Валерьевич, ассистент, кафедра компьютерных наук, Винницкий национальный технический университет, Украина.

Savchuk Tamara, Vinnytsia National Technical University, Ukraine. Kozachuk Andriy, Vinnytsia National Technical University, Ukraine, e-mail: kozachuk35@rambler.ru