УДК 519.7.007.004.9

Т. О. САВЧУК, А. В. КОЗАЧУК

Вінницький національний технічний університет

АВТОМАТИЗОВАНЕ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО МАСШТАБУВАННЯ ХМАРНОГО ЗАСТОСУНКУ

Анотація. В статті запропоновано алгоритм автоматизованого прийняття рішень щодо масштабування хмарного застосунку на основі набору правил реактивного масштабування та використання оціночної функції ефективності застосування можливих варіантів масштабування. В результаті проведених досліджень була розроблена оціночна функція варіантів масштабування, що дозволяє отримати оцінку ефективності застосування інформаційної технології, яка базується на вартості утримання інфраструктури хмарного застосунку та на оцінці кількості користувачів, що перестануть користуватися хмарним застосунком через його перевантаженість. Розроблений алгоритм може бути використаний в якості складової інформаційної технології масштабування хмарного застосунку зі змінними піками навантаження.

Ключові слова: хмарні обчислення, масштабування хмарного застосунку.

Аннотация. В статье предложен алгоритм автоматизированного принятия решений по масштабированию облачного приложения на основе набора правил реактивного масштабирования и использования оценочной функции эффективности применения возможных вариантов масштабирования. В результате проведенных исследований была разработана оценочная функция вариантов масштабирования, позволяющая получить экономическую оценку эффективности на основаннии стоимости содержания инфраструктуры облачного приложения и оценке количества пользователей, которые перестанут пользоваться облачным приложением из-за его перегруженности. Прогноз количества потерянных пользователей производиться на основании расчета времени выполнения сетевых запросов. Разработанный алгоритм может быть использован в качестве составной части информационной технологии масштабирования облачного приложения с периодическими пиками нагрузки.

Ключевые слова: облачные вычисления, масштабирование облачных приложений.

Abstract. Article describes the algorithm of automatic decision making on scaling of a cloud application based on a set of rules of reactive scaling and on a function that estimates each possible way of scaling of a cloud application using calculation of lost user count and price of cloud hosting equipment. Article introduces evaluation function of scaling options that allows obtaining economic evaluation of the efficiency based on the cost of maintaining infrastructure in the hosting cloud and estimation of the number of users that stop using cloud applications due to overheating-intensity. The forecast number of orphaned users is made on the basis of calculating the network requests execution time. Developed algorithm can be used as part of a cloud application scaling technology.

Key words: cloud computing, scaling of a cloud application.

Вступ

Завантаженість сучасних веб-застосунків суттєво змінюється протягом доби. Однією з причин розміщення застосунків у хмарних хостингах є можливість проведення швидкого масштабування веб-застосунку в пікові періоди для уникнення перебоїв у його функціонуванні за рахунок виділення додаткових обчислювальних потужностей. В найпростішому випадку масштабування здійснюється в ручному режимі, при цьому оператору необхідно періодично перевіряти завантаженість хмарного застосунку та приймати рішення щодо доцільності проведення масштабування. Автоматизація процесу прийняття рішення щодо масштабування дозволяє швидше реагувати на зміну завантаженості обчислювальних потужностей на усуває необхідність в постійному ручному моніторингу роботи хмарного застосунка.

Постановка задачі

Нехай $\langle S, N, P \rangle$ – поточний стан інфраструктури хмарного застосунку, де S розмір віртуальної машини, N – кількість виділених віртуальних машин P – вартість утримання віртуальних машин протягом 1 хвилини, X – вектор прогнозів кількості мережевих запитів на наступні h хвилин, час після минулого масштабування $\tau_{масш}$, G – граф станів інфраструктури хмарного застосунку, $t_{спок}$ – проміжок часу після масштабування протягом якого нове масштабування не здійснюється, C – інформація про середній час виконання мережевого запиту при різних станах роботи хмарного застосунку.

Необхідно побудувати алгоритм, що на основі даних про стан хмарного застосунку повертає рішення щодо проведення масштабування хмарного застосунку $e = \langle S', \Delta N \rangle$, де S' – розмір віртуальної машини після масштабування, ΔN – приріст кількості віртуальних машин відносно стану інфраструктури хмарного застосунку до початку масштабування.

Аналіз існуючих рішень

Найпоширенішими підходами до автоматизованого масштабування хмарного застосунку є використання методу реактивного масштабування та масштабування за розкладом [1]. Недоліком цих методів є неможливість проведення масштабування до виникнення перезавантаження хмарного застосунку та можливі неточності розкладу, відповідно, що робить описані підходи неефективними при певних умовах функціонування хмарного застосунку.

Алгоритм прийняття рішення щодо масштабування хмарного застосунку

Введемо алгоритм прийняття рішень щодо масштабування хмарного застосунку, що базується на використанні методів реактивного або проактивного масштабування. Метод реактивного масштабування представляє собою набір правил, які виконуються коли існує гостра потреба в проведенні масштабування, проактивне масштабування здійснюється на основі прогнозу роботи хмарного застосунку, який може

бути неточним, тому при виборі остаточного рішення пріоритет надається результатам роботи методу реактивного масштабування.

Результатом роботи алгоритму є рішення про необхідність масштабування. При цьому масштабування може бути вертикальним (змінюється значення N) або горизонтальним (змінюється значення S). Позначимо результати масштабування хмарного застосунку парою $e = \langle S', \Delta N \rangle$. Додатні значення ΔN відповідають масштабуванню вгору, від'ємні – масштабуванню вниз. Результатом алгоритму може бути пара $\langle S, 0 \rangle$, що відповідає відсутності необхідності у масштабуванні. Вважатимемо, що такий результат еквівалентний пустій операції масштабування: $\langle S, 0 \rangle = null$.

Робота алгоритму прийняття рішення щодо масштабування хмарного застосунку базується на використанні під-алгоритмів, що представляють методи реактивного та проактивого масштабування. Вхідними даними цих алгоритмів є підмножини вхідних даних головного алгоритму, а результат їхньої роботи – рішення щодо проведення масштабування $\langle S', \Delta N \rangle$.

Алгоритм складається з наступних кроків:

Крок 1. Початок роботи.

Крок 2. Введення інформації про стан хмарного застосунку.

Крок 3. Присвоїти поточному рішенню щодо масштабування *e* результат роботи алгоритму реактивного масштабування.

Крок 4. Якщо *e* є пустим – виконати крок 5, інакше – крок 6.

Крок 5. Присвоїти поточному рішенню щодо масштабування *e* результат роботи алгоритму проактивного масштабування.

Крок 6. Вивести значення поточного рішення щодо масштабування *e*.

Крок 7. Кінець.

Схема алгоритму зображена на рис. 1.

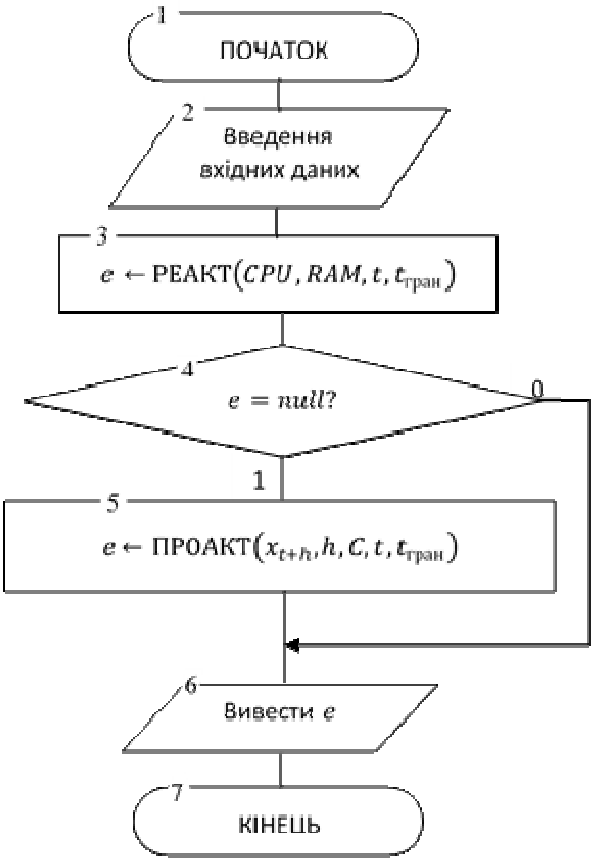

Рисунок 1 – Алгоритм прийняття рішень щодо масштабування хмарного застосунку

Метод реактивного масштабування

Класичним підходом до прийняття рішення щодо масштабування хмарного застосунку є застосування методу реактивного масштабування який передбачає визначення граничних показників використання пам'яті та процесора при досягненні яких відбувається масштабування [1]. Цей метод масштабу-

вання також називають статичним пороговим методом [2]. Ядро методу реактивного масштабуваня складає набір правил $r \in R$ вигляду

$$\text{якщо } X_r \text{ і } t \geq t_{\text{грак}},$$
$$\text{то здійстини масштабування } \langle f_r(S), \Delta N \rangle, t_{\text{грак}} \leftarrow t + \Delta t$$

де t – поточний момент часу, $t_{\text{грак}} = \tau_{\text{масш}} + \Delta t$ – момент часу в який закінчується період стабілізації після минулої операції масштабування, Δt – проміжок часу спокою після масштабування впродовж якого нові операції масштабування не здійснюють, необхідний для того щоб почати знімати показники завантаженості застосунку після зміни конфігурації хмарної інфраструктури, f – функція, що на основі поточного розміру віртуальної машини S повертає необхідний розмір віртуальної машини S', X_r – умова проведення масштабування, що може мати вигляд

завантаженість процесора > 80% або частка вільної пам'яті < 20%.

В більшості випадків [1] правила створюються попарно: для проведення масштабування вгору і вниз. При створенні правил слід враховувати обмеження хостингу хмарного застосунку: слідкувати, щоб кількість та розмір віртуальних машин були у дозволених межах, не використовувати переходи, що відсутні в графі станів інфраструктури хмарного застосунку, це призводить до ускладнення умов правил X_r.

При проведенні реактивного масштабування необхідно враховувати можливість виникнення осциляцій кількості віртуальних машин, для уникнення цього небажаного ефекту слід ретельно підбирати граничні показники правил масштабування. Для того, щоб метод не видавав команд масштабування у відповідь на одиничні піки навантаження в умовах правил X_r використовують усереднені показники за декілька останніх кроків часу, з цією ж метою деякі різновиди методу видають команду масштабування тільки за послідовного виконання умови X_r протягом кількох кроків часу.

В набір правил R можна додати правила з умовами, які виконуються в певний момент часу, без прив’язки до рівня використання системних ресурсів. Такі правила дають можливість провести масштабування напередодні подій, що збільшують завантаженість хмарного застосунку. Для мінімізації видатків можуть бути зроблені модифікації правил з врахуванням політики ціноутворення хостинга. Наприклад, при погодинній тарифікації немає сенсу здійснювати масштабування вниз протягом години після операції масштабування вгору.

Основними перевагами застосування методу реактивного масштабування до хмарного застосунку з періодичними піками навантаження є простота реалізації методу, його висока швидкодія та можливість початку експлуатації без збору додаткової інформації про інфраструктуру хмарного хостингу. До обмежень методу можна віднести реактивну природу, яка відображається в його назві – команда про проведення масштабування може бути віддана тільки тоді, коли виявлена нестача обчислюючих ресурсів, при цьому під час виділення нових ресурсів функціонування хмарного застосунку може відбуватися в нештатному режимі. Також, складною процедурою є встановлення порогових значень в правилах r, а також вибір дії масштабування $\langle S', \Delta N \rangle$ – через велику кількість комбінацій вибір оптимальної є нетривіальним.

Можливість використання методу без додаткових знань про функціонування хмарного застосунку в умовах конкретної інфраструктури хмарного хостингу дозволяють використовувати описаний метод як резерв для більш складних методів масштабування – при неможливості прийняття рішення щодо масштабування за допомогою власних ресурсів вони можуть використовувати результати роботи методу реактивного масштабування.

Метод прийняття рішення на основі прогнозу кількості мережевих запитів

За умови, що хмарний застосунок має стан $\langle S_i, N_i, P_i \rangle$ із графу станів інфраструктури хмарного застосунку, можливі *d(i)* варіанти проведення масштабування, де *d* – функція, що повертає степінь вершини графа. Позначимо множину можливих варіантів масштабування як E_i. Для вибору оптимального варіанту масштабування розрахуємо значення оціночної функції $G(e)$, $e \in E_i$ наступним чином:

$$G(e) = P_{\text{ІНФР}} + k * P_{\text{ВТРАТ}}, \quad (1)$$

Де $P_{\text{інфр}}$ – вартість утримання інфраструктури хмарного застосунку протягом наступних $t_{спок}$ хвилин, $P_{\text{втрат}}$ – вартість втрати одного користувача, k – кількість користувачів, втрачених через надмірну завантаженість хмарного застосунку протягом наступних $t_{спок}$ хвилин, e – дуга графу станів інфраструктури хмарного застосунку $\left(\langle S_i, N_i, P_i \rangle, \langle S_j, N_j, P_j \rangle, t_{ij}\right)$.

Вартість утримання інфраструктури визначається як сума вартості утримання до закінчення масштабування та після його закінчення, кожен з доданків розраховується як значення тарифу за одну хвилину P помножена на кількість хвилин $t_{спок} - t_{ij}$ та t_{ij}:

$$P_{ІНФР} = P_j t_{ij} + P_j\left(t_{СПОК} - t_{ij}\right). \quad (2)$$

Кількість користувачів, втрачених через надмірну завантаженість хмарного застосунку розраховується наступним чином:

$$k = \sum_{t=1}^{t_{спок}} v_t q_{ВТРАТ}(t). \quad (3)$$

де v_t – кількість користувачів хмарного застосунку в період часу $(t;t+1)$, $q_{втрат}(t)$ – імовірність того, що користувач перестане користуватися хмарним застосунком. Підставивши (2) та (3) в (1) отримаємо більш детальний вираз для обчислення оціночної функції:

$$G(e) = P_j t_{ij} + P_j\left(t_{СПОК} - t_{ij}\right) + P_{ВТРАТ} \sum_{t=1}^{t_{спок}} v_t q_{ВТРАТ}(t). \quad (4)$$

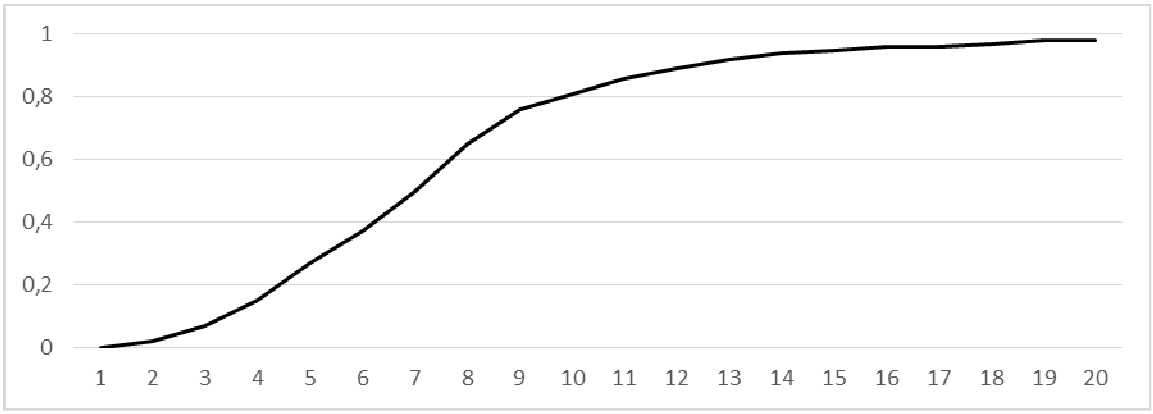

Рисунок 2 – Імовірність втрати користувача в залежності від часу запиту

Для визначення імовірності втрати користувача $q_{втрат}$ використаємо функцію розподілу імовірностей, що представляє залежність імовірності втрати користувача від часу виконання мережевого запиту $F(\tau)$ – імовірність втрати користувача зростає разом з часом очікування відповіді від хмарного застосунку. Дослідження [3-6] показують, що кількість користувачів веб-сторінок починає зменшуватись при $\tau \geq 2c$. Портал Pear1 hosting [3] при цьому наводить функцію розподілу імовірності втрати користувача, що показана на рисунку 2.

Позначимо випадкову величину, що відповідає часу виконання одного мережевого запиту хмарним застосунком за R. Статистичні характеристики R залежать від завантаженості хмарного застосунку в конкретний момент часу t та від стану інфраструктури хмарного застосунку. Визначимо імовірність втрати користувач через математичне сподівання часу виконання мережевого запиту $M(R)$:

$$q_{ВТРАТ}(t) = F\left(M\left(R_{j,t}\right)\right). \quad (5)$$

При умові, що $t_{ij} \leq h$ та з врахуванням (5) вираз (3) можна представити наступним чином:

$$k = \sum_{t=1}^{t_{ij}} v_t F\left(M\left(R_{i,t}\right)\right) + \sum_{t=t_{ij}+1}^{h} v_t F\left(M\left(R_{j,t}\right)\right) + \sum_{t=h+1}^{t_{спок}} v_t F\left(M\left(R_{j,t}\right)\right). \quad (6)$$

Враховуючи те, що прогноз кількість мережевих запитів робиться на h кроків вперед необхідно визначити значення v_t та R для проміжку часу $(h; t_{\text{СПОК}}]$. Використаємо прогноз на момент часу h для цього часового проміжку, тоді останній доданок в (6) може бути представлений добутком:

$$k = \sum_{t=1}^{t_{ij}} v_t F\left(M\left(R_{i,t}\right)\right) + \sum_{t=t_{ij}+1}^{h} v_t F\left(M\left(R_{j,t}\right)\right) + (t_{\text{СПОК}} - h) v_h F\left(M\left(R_{j,h}\right)\right), \quad (7)$$

Таким чином, вираз (4) може бути деталізований за допомогою співвідношення (7):

$$G(e)\,|\,t_{ij} \le h = P_j t_{ij} + P_j\left(t_{\text{СПОК}} - t_{ij}\right) +$$

$$+P_{\text{ВТРАТ}}\left(\sum_{t=1}^{t_{ij}} v_t F\left(M\left(R_{i,t}\right)\right) + \sum_{t=t_{ij}+1}^{h} v_t F\left(M\left(R_{j,t}\right)\right) + (t_{\text{СПОК}} - h) v_h F\left(M\left(R_{j,h}\right)\right)\right). \quad (8)$$

Для випадку коли $t_{ij} > h$ аналогічні перетворення приведуть до виразів (9) та (10):

$$k = \sum_{t=1}^{h} v_t F\left(M\left(R_{i,t}\right)\right) + (t_{ij} - h) v_h F\left(M\left(R_{i,h}\right)\right) + (t_{\text{СПОК}} - t_{ij}) v_h F\left(M\left(R_{j,h}\right)\right) \quad (9)$$

$$G(e)\,|\,t_{ij} > h = P_j t_{ij} + P_j\left(t_{\text{СПОК}} - t_{ij}\right) +$$

$$+P_{\text{ВТРАТ}}\left(\sum_{t=1}^{h} v_t F\left(M\left(R_{i,t}\right)\right) + (t_{ij} - h) v_h F\left(M\left(R_{i,h}\right)\right) + (t_{\text{СПОК}} - t_{ij}) v_h F\left(M\left(R_{j,h}\right)\right)\right). \quad (10)$$

Таким чином, на основі прогнозу кількості користувачів хмарного застосунку та середнього часу виконання мережевого запиту можна обчислити значення оціночної функції $G(e)$ для всіх можливих способів масштабування $e \in E_i$ хмарного застосунку, що має стан інфраструктури $\langle S_i, N_i, P_i \rangle$ в момент часу t. Спосіб масштабування з найменшим значенням оціночної функції використовується в якості обраного варіанта масштабування.

Формальний запис правила масштабування має наступний вигляд:

$$\text{якщо } t > t_{\text{гран}},$$
$$\text{то здійстини масштабування } e_{min} | G(e_{min}) = \min_{e} G(e)$$

$$\text{якщо } e_{min}.S' \neq S_i \text{ або } e_{min}.\Delta N \neq 0,$$
$$\text{то } t_{\text{гран}} \leftarrow t + \Delta t$$

де t – поточний момент часу, $t_{\text{гран}}$ – момент часу в який закінчується період стабілізації після минулої операції масштабування, Δt – проміжок часу спокою після масштабування впродовж якого нові операції масштабування не здійснюються.

Прогнозування часу виконання мережевого запиту

Для визначення значення оціночної функції ефективності масштабування хмарного застосунку (2.10) необхідно мати інформацію про кількість користувачів хмарного застосунку в хвилину v та про середній час виконання мережевого запиту $M(R)$. Побудуємо прогноз цих величин на основі прогнозу кількості мережевих запитів та статистичної інформації про функціонування хмарного застосунку.

Відношення кількості мережевих запитів до кількості унікальних користувачів при великій кількості користувачів є сталою величиною для конкретного хмарного застосунку. Тому прогноз кількості користувачів v_t може бути розрахована наступним чином:

$$v_t = x_t * c_{\text{корист}},$$

де t – проміжок часу для якого здійснюється прогнозування, x_t – прогноз кількості мережевих запитів за проміжок часу *t*, $c_{корист}$ – коефіцієнт, що показує частоту запитів від одного користувача за одиницю часу для конкретного хмарного застосунку. Значення коефіцієнта $c_{корист}$ визначається шляхом періодичного збору статистики про кількість унікальних користувачів та кількість мережевих запитів під часу роботи хмарного застосунку.

Для прогнозування часу виконання мережевого запиту на основі кількості мережевих запитів в одиницю часу необхідно мати інформацію про виділені обчислювальні потужності. Таку інформацію містить опис інфраструктури хмарного застосунку $\langle S, N, P \rangle$, а саме розмір віртуальних машин *S* та їх кількість *N*. Представимо можливі комбінації значень змінних x, S та *N* за допомогою гіперкуба станів хмарного застосунку[7, 8]. Кількість вимірів такого гіперкуба дорівнює трьом, в комірках зберігається масив, кожен елемент якого представляє собою час виконання окремого мережевого запису. В комірці також зберігається середнє значення елементів масиву, що дозволяє пришвидшити розрахунки. Для зменшення кількості комірок гіперкубу та полегшення прогнозування[9] необхідно провести дискретизацію змінної x, розбивши діапазон її значень на 3-10 відрізків.

Перевагою використання гіперкубу для зберігання даних про середній час мережевого запиту є можливість такого впорядкування даних, при якому виникає змога використання інформації з сусідніх комірок у випадку відсутності інформації в комірці, що описує стан хмарного застосунку. Графічне зображення гіперкубу станів хмарного застосунку наведене на рисунку 3, пунктирними стрілками показані напрями зростання середнього часу виконання мережевого запиту.

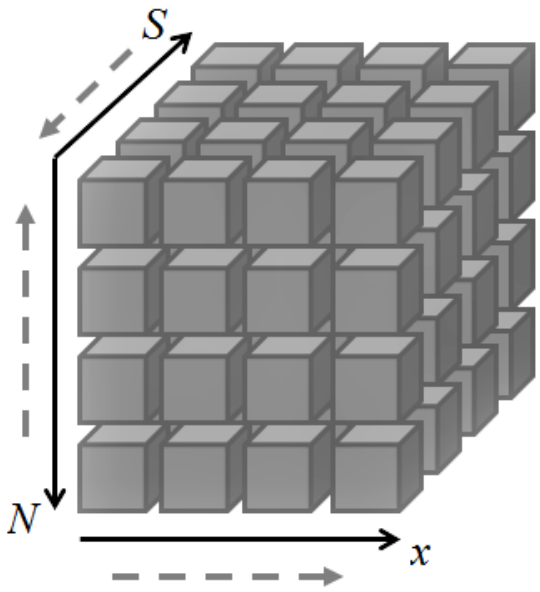

Рисунок 3 - Гіперкуб станів хмарноо застосунку

За умови, що зростання часу виконання мережевого запиту змінюється лінійно при пересуванні вздовж одного з вимірів гіперкубу, можна ввести алгоритм визначення часу виконання мережевого запиту, який дозволяє доповнити інформацію про пусті комірки гіперкуба за допомогою лінійної інтерполяції.

Крок 1. Початок роботи.

Крок 2. Введення інформації про розмірність гіперкубу станів хмарного застосунку k_S, k_N, k_x, множину значень його комірок C та цільовий стан iS, iN, ix, глибину пошуку *z*.

Крок 3. Якщо $C_{iS,iN,ix}$ має значення – виконати крок 4, інакше – крок 5.

Крок 4. Вивести $C_{iS,iN,ix}$, перейти до кроку 13.

Крок 5. Для *j* від 1 до z виконати кроки 6 – 11.

Крок 6. Якщо $C_{iS-j,iN,ix}$ існує та має значення і $C_{iS+j,iN,ix}$ існує та має значення – перейти до кроку 7, інакше – перейти до кроку 8.

Крок 7. Вивести $\left(C_{iS-j,iN,ix} + C_{iS+j,iN,ix}\right)/2$, перейти до кроку 13.

Крок 8. Якщо $C_{iS,iN-j,ix}$ існує та має значення і $C_{iS,iN+j,ix}$ існує та має значення – перейти до кроку 9, інакше – перейти до кроку 10.

Крок 9. Вивести $\left(C_{iS,iN-j,ix} + C_{iS,iN+j,ix}\right)/2$, перейти до кроку 13.

Крок 10. Якщо $C_{iS,iN,ix-j}$ існує та має значення і $C_{iS,iN,ix+j}$ існує та має значення – перейти до кроку 11, інакше – перейти до кроку 5.

Крок 11. Вивести $\left(C_{iS,iN,ix-j} + C_{iS,iN,ix+j}\right)/2$, перейти до кроку 13.

Крок 12. Вивести невизначений результат.

Крок 13. Кінець.

Схема алгоритму визначення часу виконання мережевого запиту зображена на рис. 4. За рахунок введення змінної *z* алгоритм дозволяє проводити прогнозування на основі комірок гіперкубу, значення яких є невизначеним.

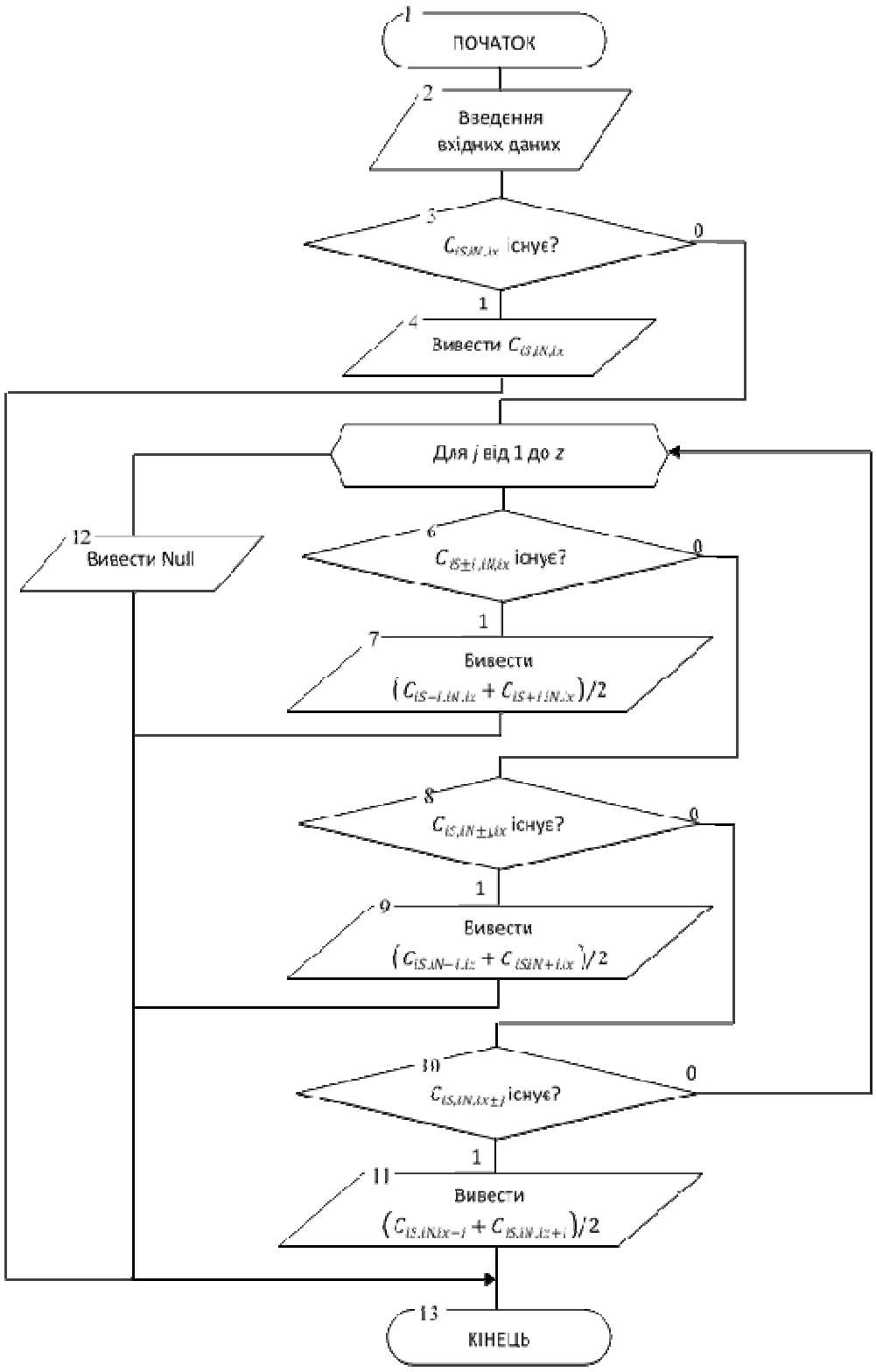

Рисунок 4 – Алгоритм визначення часу виконання мережевого запиту

Висновки

Таким чином, на основі інформації про стан хмарного застосунку було сформовано алгоритм, що на основі набору правил реактивного масштабування та оцінки ефективності здійснює прийняття рішення щодо проведення масштабування хмарного застосунку. Розроблена оціночна функція варіантів масштабування дозволяє отримати економічну оцінку ефективності, що базується на вартості утримання інфраструктури хмарного застосунку та на оцінці кількості користувачів, що перестануть користуватися хмар-

ним застосунком через його перевантаженість. Розроблений алгоритм може бути використаний в якості складової інформаційної технології масштабування хмарного застосунку.

Список літератури

1. Lorido-Botran T. Auto-scaling techniques for elastic applications in cloud environments/ T. Lorido-Botran, J. Miguel-Alonso, J. A. Lozano// Department of Computer Architecture and Technology, University of Basque Country, Tech. Rep. – EHU-KAT-IK-09, 2012 – № 12.
2. RightScale Cloud Management [Електронний ресурс] – режим доступа: http://www.rightscale.com/.
3. How a Slow Website Impacts Your Visitors and Sales [Електронний ресурс] – режим доступа: http://www.peer1.com/knowledgebase/how-slow-website-impacts-your-visitors-and-sales .
4. Nah F.F.H. A study on tolerable waiting time: how long are Web users willing to wait?// Behaviour & Information Technology, 2004 – №23(3) – С. 153-163.
5. How Loading Time Affects Your Bottom Line [Електронний ресурс] – режим доступа: https://blog.kissmetrics.com/loading-time/ .
6. Menasce D. Load testing of web sites.// Internet Computing, 2002 – № 6(4) – С. 70-74.
7. Савчук Т. О. Прогнозування стану техногенної ситуації за допомогою імітаційної моделі, що базується на багатовимірному рівнянні дифузії / Т. О. Савчук, А. В. Козачук// Вісник національного університету "Львівська політехніка". Інформаційні системи та мережі – 2014 – №783, – С. 473-478
8. Савчук Т. О. Визначення доцільності використання багатовимірного підходу до прогнозування стану техногенної ситуації/ Т. О. Савчук, А. В. Козачук// Вимірювальна та обчислювальна техніка в технологічних процесах – Хмельницький, 2014 - № 2(47) – С. 179-182
9. Data Mining with Microsoft SQL Server 2008. Jamie MacLennan, ZhaoHui Tang, Bogdan Crivat. 2008. – 672 pages. – ISBN: 978-0-470-27774-4.

Стаття надійшла: 09.06.2015.

Відомості про авторів

Савчук Тамара Олександрівна – к.т.н. проф. кафедри Комп'ютерних наук ВНТУ. Місто Вінниця, Хмельницьке шосе 95.

Козачук Андрій Валерійович – аспірант кафедри Комп'ютерних наук ВНТУ. Місто Вінниця, Хмельницьке шосе 95.