

Д. С. Гребенюк, В. В. Давидов

Національний технічний університет "Харківський політехнічний інститут", Харків, Україна

## МЕТОД ПЕРВИННОГО ВИДІЛЕННЯ ХМАРНИХ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ НА ОСНОВІ АНАЛІЗУ ІЄРАРХІЙ

**Анотація.** Предметом дослідження в статті є моделі та методи розмежування навантаження і ресурсів в системах хмарних обчислень, зокрема, які базуються на моделі надання послуг інфраструктури як сервісу. Метою роботи є оптимізація первинного виділення ресурсів в системах хмарних обчислень шляхом адаптації методу аналізу ієрархій для впровадження в хмарні обчислення. Це дозволить запускати нові віртуальні машини з мінімальним зниженням продуктивності вже функціонуючих екземплярів. У статті вирішуються наступні завдання: дослідження доцільності використання моделі аналізу ієрархій в хмарних обчисленнях; адаптація методу аналізу ієрархій для впровадження в системи хмарних обчислень; оцінка ефективності впровадженого методу. Для вирішення поставлених завдань були використані підходи і методи теоретичних досліджень, які засновані на наукових положеннях статичного, функціонального і системного аналізів. Отримані наступні результати: запропоновано використання методу аналізу ієрархій для розподілу ресурсів у системах хмарних обчислень. Наведено переваги та недоліки цього методу для використання у хмарних обчисленнях. Метод аналізу ієрархій був адаптовано для систем хмарних обчислень. Наведено алгоритм його використання. Проведена оцінка ефективності розробленого метода, який показав доцільність його використання при розподілу ресурсів у системах хмарних обчислень. **Висновки.** Вдосконалення методу первинного розмежування ресурсів в системах хмарних обчислень дозволило підвищити здатність цих систем запускати нові віртуальні машини з мінімальним зниженням продуктивності вже функціонуючих екземплярів.

**Ключові слова:** системи хмарних обчислень, аналіз ієрархій, виділення.

### Постановка проблеми

Розвиток сучасних інформаційних технологій потребує розвитку систем хмарних обчислень. Серед інтенсивного та екстенсивного напрямків розвитку було обрано екстенсивний, як напрям, що дозволяє масштабувати інформаційні ресурси, не упираючись в поточні обмеження комп'ютерних ресурсів (наприклад, ЦПУ, мережна пропускання здатність тощо), що можуть бути розташовані на одному хості. Екстенсивний напрям розвитку призвів до утворення віддалених кластерів, де зазначені ресурси загальні для всіх хостів. Обчислення на таких віддалених кластерах отримали назву «хмарні обчислення». Хмарні обчислення надають такі послуги [1]:

- Software-as-a-Service (SaaS) – програмне забезпечення, серверна частина якого знаходиться в системах хмарних обчислень;
- Platform-as-a-Service (PaaS) – віртуальний хост зі своєю ОС, на яку можна встановити декілька програмних забезпечень типу SaaS;
- Infrastructure-as-a-Service (IaaS) – віртуальний кластер, що включає до себе декілька віртуальних хостів та конфігурацію мережі між ними.

У ході життєвого циклу роботи системи типу IaaS настає проблема, коли частина кластеру має високий рівень навантаження використання ресурсів, а інша – простоє. Це призводить до необхідності балансування ресурсів для оптимальної витрати існуючих ресурсів у системах хмарних обчислень для підвищення ефективності їх використання.

**Аналіз літератури.** Існуючі в хмарних середовищах підходи до первинного виділення ресурсів можна розділити на три типи: виділення ресурсів адміністратором; планування ресурсів диспетчером хмари; планування ресурсів середовищем віртуалізації.

Специфіка розподілу ресурсів в хмарі IaaS бере свій початок в основних ідеях, які закладені в цій

моделі хмарного сервісу. Зокрема можна виділити такі: відсутність у адміністратора інформації про реальні потреби застосунків, які перебувають всередині віртуальних машин та можливості ручного призначення ресурсів екземплярів.

Таким чином, системи хмарних обчислень мають особливості, які створюють специфіку розподілу ресурсів в хмарі. Деякі елементи цієї проблематики є спільними з проблемами систем віртуалізації, а деякі - мають особливості, характерні для хмари. На процес розподілу ресурсів в хмарі впливають [2]: різні класи обладнання і поєднання ресурсів; нерівномірна завантаженість кожного з ресурсів; відсутність відомостей про реальні потреби застосунків у ресурсах; різні потреби застосунків у ресурсах; невідповідність запитаних ресурсів із реально споживаними; відсутність можливості ручного призначення ресурсів.

Таким чином, хмарні середовища мають ряд особливостей, не характерних для інших моделей надання ресурсів, що викликає необхідність удосконалення механізмів розподілу ресурсів для зазначених середовищ.

Аналіз літератури [2, 3] дозволив виявити особливості методу розподіленого виділення ресурсів (DRS). Метод DRS вирішує задачу розподілення робочих навантажень віртуальної машини по вузлах всередині кластера віртуалізації, а також відстежує доступні ресурси. Аналіз показав, що цей метод оцінює лише навантаження центрального процесора і не прогнозує зміну навантаження за іншими параметрами (наприклад, кількість операцій введення-виведення та завантаженість мережі). Саме тому метод DRS є достатньо примітивним і не підходить для застосування у системах хмарних обчислень, де адміністратор системи хмарних обчислень не має можливості передбачити всі небажані наслідки від перерозподілу ресурсів у такий спосіб.

Аналіз літератури [2, 3] дозволив виявити особливості методу динамічного розподілу ресурсів (DPM). Метод DPM оптимізує енергоспоживання на рівні кластера або вузла. Дослідження даного методу показало, що він вирішує задачу зниження енергоспоживання в середовищі віртуалізації, однак не вирішує задачу максимізації ефективного використання наявних ресурсів.

Проведений аналіз існуючих методів розподілу ресурсів у системах хмарних обчислень показав, що поточні методи розподілу ресурсів не задовольняють реальним потребам власників і споживачів хмарної послуги IaaS і не дозволяють оптимально використовувати наявні ресурси для віртуальних машин.

**Метою роботи** є підвищення ефективності використання обчислювальних ресурсів у системах хмарних обчислень на основі моделі IaaS шляхом розроблення алгоритму первинного виділення хмарних обчислювальних ресурсів. Дослідження поставленої задачі показало, що вона відноситься до класу багатокритеріальних задач прийняття рішень і може бути вирішена з використанням математичного апарату методу аналізу ієрархій [4].

### Результати досліджень

**Дослідження доцільності використання методу аналізу ієрархій.** Метод аналізу ієрархій – методологічна основа для вирішення завдань вибору альтернатив за допомогою їх багатокритеріального рейтингування, що розроблена Т. Сааті. Основне застосування методу – підтримка прийняття рішень за допомогою ієрархічної композиції завдання і рейтингування альтернативних рішень. Основні можливості зазначеного методу [5]:

1. Проведення аналізу проблеми. При цьому проблема прийняття рішення представляється у вигляді ієрархічно впорядкованих: а) головної мети (головного критерію) рейтингування можливих рішень; б) декількох груп (рівнів) однотипних факторів, які так чи інакше впливають на рейтинг; в) групи можливих рішень; г) системи зв'язків, що вказують на взаємний вплив чинників і рішень.

2. Проведення збору даних про проблему. Відповідно до результатів ієрархічної декомпозиції модель ситуації прийняття рішення має кластерну структуру. Набір можливих рішень і всі фактори, що впливають на пріоритети рішень, розбиваються на відносно невеликі групи – кластери. Розроблена в методі аналізу ієрархій процедура парних порівнянь [6, 7] дозволяє визначити пріоритети об'єктів, що входять до кожного кластера. Для цього використовується метод власного вектору. При цьому, складна проблема збору даних розбивається на ряд більш простих, вирішальних для кластерів.

3. Оцінка суперечливості даних і мінімізація її. Для реалізації даної можливості, в методі аналізу ієрархій розроблені процедури узгодження. Зокрема, є можливість визначити найбільш суперечливі дані, що дозволяє виявити найменш ясні ділянки проблеми.

4. Синтез проблеми прийняття рішення. Після того, як проведено аналіз проблеми та зібрано дані

по всім кластерам, за спеціальним алгоритмом розраховується підсумковий рейтинг – набір пріоритетів альтернативних рішень. Властивості цього рейтингу дозволяють здійснювати підтримку прийняття рішень. Наприклад, приймається рішення з найбільшим пріоритетом. Крім того, метод дозволяє побудувати рейтинги для груп факторів, що дозволяє оцінювати важливість кожного фактора.

5. Обговорення проблеми, що сприяє досягненню консенсуса. Думки, що виникають під час обговорення проблеми прийняття рішення, самі можуть в даній ситуації розглядатися в якості можливих рішень. Тому метод аналізу ієрархій можна застосувати для визначення важливості врахування думки кожного учасника обговорення.

6. Оцінка важливості врахування кожного рішення і важливості врахування кожного чинника, що впливають на пріоритети рішень. У відповідності з формулюванням завдання прийняття рішення, величина пріоритету безпосередньо пов'язана з оптимальністю рішення. Тому рішення з низькими пріоритетами відкидаються як несуттєві.

7. Оцінка стійкості прийнятого рішення. Прийняте рішення можна вважати обґрунтованим лише за умови, що неточність даних або неточність структури моделі ситуації прийняття рішення не впливають істотно на рейтинг альтернативних рішень.

*До переваг* зазначеного методу при вирішенні завдання первинного виділення ресурсів в системах хмарних обчислень у порівнянні з іншими методами багатокритеріального аналізу слід віднести [5]:

- відсутність загальних правил для формування структури моделі прийняття рішення, що дозволяє враховувати «людський фактор» при підготовці прийняття рішення. Таким чином, вдається наблизити роботу підсистеми розподілу ресурсів до дій системного адміністратора, який здатний враховувати неявні взаємозв'язки між різними параметрами функціонування системи хмарних обчислень;
- простота процедури розрахунків рейтингів, що дозволяє зробити процес вибору рішення максимально прозорим при налаштуванні підсистеми розподілу ресурсів;

- метод надає можливість враховувати думку не одного, а декількох експертів, що дозволяє отримати більш об'єктивну оцінку значимості показників;

- метод надає багато можливостей для виявлення і мінімізації протиріч у вихідних даних, що дозволяє ефективно оперувати великим числом показників в процесі прийняття рішення.

- метод дає не лише спосіб виявлення найкращого рішення, але і дозволяє кількісно виразити ступінь переваги за допомогою рейтингування, відображаючи природний хід людського мислення і даючи більше загальний підхід, ніж метод логічних ланцюгів. Це дозволяє більш повно задіяти отримані результати при їх подальшому використанні.

Метод аналізу ієрархій, проте, має і низку *недоліків*, які не є суттєвими при вирішенні задачі:

- в рамках методу аналізу ієрархій немає можливостей для перевірки достовірності даних. Це

важливий недолік, що частково обмежує можливість застосування методу. Однак при виборі оптимального хоста для розміщення екземпляра в хмарній системі в принципі не може бути об'єктивних даних з урахуванням її поганої спроможності до формалізації неявних залежностей. Якщо збір даних проведено за допомогою досвідчених експертів, і в даних немає істотних протиріч, то якість таких даних визнається задовільною;

– метод дає лише спосіб рейтингування альтернатив, але не має внутрішніх засобів для інтерпретації рейтингів, тобто вважається, що особа, яка приймає рішення, знаючи рейтинг можливих рішень, повинна в залежності від ситуації сама зробити висновок. Проте, одержуваних даних досить для прийняття рішення в рамках підсистеми розподілу ресурсів.

Таким чином, обґрунтовано доцільність застосування методу аналізу ієрархій для вирішення задачі первинного виділення ресурсів в системах хмарних обчислень.

**Адаптація методу аналізу ієрархій для впровадження в системи хмарних обчислень.** Метод аналізу ієрархій Сааті успішно застосовується в різних областях науки та техніки для вирішення подібних завдань [8], однак, в силу особливостей розв'язуваної задачі, вимагається виконання кроків, які враховують специфіку систем хмарних обчислень.

**Крок 1.** Система хмарних обчислень попередньо підрозділяється адміністратором на зони, що складаються зі схожих за параметрами хостів і схожих даних.

**Крок 2.** З урахуванням заданих обмежень для віртуальної машини (на число процесорів, обсяг пам'яті, ресурси зберігання та ін.) вибирається зона, що найбільш йому відповідає. Використання зон дозволяє врахувати обмеження віртуальної машини, що запускається, за характеристиками, щоб обрати його розташування вже там, де ресурсів завідомо буде достатньо для функціонування програм, які виконуються на екземплярі.

**Крок 3.** Виконується оцінка показників якості [9] кожного хоста і сховища даних, якими описується функціонування хоста і його систем, включених в обрану зону. До таких показників якості необхідно віднести наступні:

- продуктивність ( $P$ ): число ЦП ( $N_{CPU}$ ), число ядер ЦП ( $N_{core}$ ), тактова частота ЦП ( $V_{CPU}$ ), швидкість роботи мережі ( $V_{net}$ );
- надійність ( $P$ ) [10]: середній час між збоями ( $M_{fail}$ );
- навантаженість ( $S$ ): зайняте місце в ОЗП ( $L_{mem}$ ), завантаження ЦП ( $L_{CPU}$ ), навантаження на дискову підсистему ( $L_{disk}$ ), відгук дискової підсистеми ( $T_{disk}$ ), завантаження мережі ( $L_{net}$ ), число екземплярів, що вже виконуються ( $N_{vm}$ ).

При цьому, сукупність окремих показників, що впливають на комплексний показник якості, може бути зведена в 2 групи: детерміновані (пов'язані зі зміною обладнання); стохастичні (пов'язані з навантаженням і часом роботи).

Частина зазначених показників, таких як фізичний стан обладнання, якість його збірки і рівень

компетенції обслуговуючого персоналу, не можуть бути кількісно визначені, а отже, є нечіткими і можуть бути представлені як лінгвістичні змінні [11].

**Крок 4.** Виконується вибір найбільш відповідного хоста на основі аналізу альтернатив на основі отриманих значень показників якості [12]. Кожен показник має свою попередньо задану вагу, що визначена певним експертним шляхом для кожної зони, виходячи з узагальнених потреб клієнтів і наявного обладнання. В результаті, для розміщення віртуальної машини обирається хост, що забезпечує максимальну ймовірність оптимального розподілу ресурсів.

**Дослідження методу аналізу ієрархії на прикладі декілька-гілкової ієрархії.** Для дослідження методу аналізу ієрархій, що адаптований до систем хмарних обчислень, розроблено алгоритм, що оперує ієрархією з декількома гілками (рис. 1) [13].

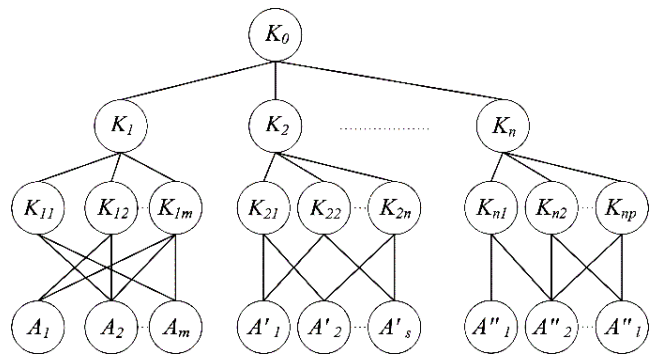


Рис. 1. Приклад ієрархії з декількома гілками

**Крок 1.** Обчислюються вектори пріоритетів альтернатив відносно критеріїв  $K_{ij}$ :

$$\left\{ W_{K_{11}}^A, W_{K_{12}}^A, \dots, W_{K_{1m}}^A \right\}; \left\{ W_{K_{21}}^{A'}, W_{K_{22}}^{A'}, \dots, W_{K_{2n}}^{A'} \right\}; \left\{ W_{K_{n1}}^{A''}, W_{K_{n2}}^{A''}, \dots, W_{K_{np}}^{A''} \right\}. \quad (1)$$

**Крок 2.** Будується  $r$  матриць  $[A_i]$ , в яких найменуваннями строк є альтернативи, а стовпців – критерії  $K_{ij}$ . При цьому, якщо альтернатива  $A_i$  не має зв'язку з критерієм  $K_{ij}$ , то в матриці  $[A_i]$  на перетині відповідних строки та стовпця ставиться значення 0.

**Крок 3.** Обчислюється  $r$  векторів пріоритетів альтернатив  $W_i^A (i=1, r)$  відносно критеріїв  $K_i$ :

$$W_i^A = [A_i][S_i][L_i]\bar{X}_i[B_i], \quad (2)$$

де  $[S_i]$  – матриця для нормування матриці  $[A_i]$ ;  $[L_i]$  – структурна матриця для зміни ваги альтернатив пропорційно до відношення  $R/N$  ( $R$  – кількість альтернатив, що знаходяться під критерієм  $K_{ij}$ ,  $N$  – загальна кількість альтернатив);  $[B_i]$  – діагональна матриця для отримання вектору  $W_i^A$ ;  $\bar{X}_i$  – вектор пріоритетів критеріїв  $K_{ij}$  відносно до критеріїв  $K_i$ .

**Крок 4.** Обчислюється вектор пріоритетів критеріїв  $X$  відносно фокуса ієрархії  $K_0$ .

**Крок 5.** Будується результуюча матриця  $[A_0]$ , де найменуваннями строк є усі альтернативи, що роз-

глядаються ( $\{A_i\}, i=1..m; \{A'_i\}, i=1..s; \{A''_i\}, i=1..t;$ ), а найменуваннями стовпців – критерії  $K_i$ . При цьому результуюча матриця  $[A_0]$  має такий вигляд:

$$[A_0] = \begin{bmatrix} W_1^A & 0 & \dots & 0 \\ 0 & W_2^A & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & W_r^A \end{bmatrix}. \quad (3)$$

**Крок 6.** Визначається результуючий вектор пріоритетів  $W_0^A$  усіх альтернатив, що розглядаються, відносно фокуса ієрархії  $K_0$  стосовно виразу (2).

Для встановлення відносної важливості елементів ієрархії використовується шкала відносин (табл. 1). Дана шкала дозволяє експерту ставити ступінь значущості у відповідність до ступенів

переваги одного порівнюваного об'єкта перед іншим. Після побудови ієрархії застосовується метод попарного порівняння, тобто будується множина матриць парних порівнянь. Для цього в ієрархії виділяються елементи двох типів: елементи-батьки і елементи-нащадки. Елементи-нащадки впливають на відповідні елементи вищого рівня ієрархії, що є по відношенню до перших елементами-батьками. Матриці парних порівнянь будуються для всіх елементів-нащадків, що відносяться до відповідного елемента-батька. Елементами-батьками можуть бути елементи, що належать будь-якому ієрархічному рівню, крім останнього, на якому розташовані, як правило, альтернативи.

Парні порівняння проводяться в термінах домінування одного елемента над іншим. Отримані судження виражаються в цілих числах з урахуванням дев'ятибальної шкали (табл. 1).

Таблиця 1 – Шкала відносин (за ступенем значущості параметрів функціонування системи хмарних обчислень)

Ступінь значущості	Визначення	Пояснення
1	Однакова значимість.	Два параметра вносять однаковий внесок у досягнення мети.
3	Деяке переважання значущості однієї дії над іншою (слабка значимість).	Існують міркування на користь переваги одного з параметрів, однак ці міркування недостатньо переконливі.
5	Істотна або сильна значимість.	Є надійні дані або логічні судження для того, щоб показати перевагу одного з параметрів.
7	Очевидна або дуже сильна значимість.	Переконливе свідчення на користь одного параметра перед іншим.
9	Абсолютна значимість.	Свідчення на користь переваги одного параметра над іншим надзвичайно переконливі.
2,4,6,8	Проміжні значення між двома сусідніми судженнями.	Ситуація, коли необхідно компромісне рішення.
Зворотні величини	Якщо параметру $i$ при порівнянні з параметром $j$ приписується одне з визначених вище ненульових чисел, то параметру $j$ при порівнянні з параметром $i$ приписується зворотне значення.	Якщо узгодженість було постульовано при отриманні $N$ числових значень для утворення матриці.

Заповнення квадратних матриць парних порівнянь здійснюється за наступним правилом. Якщо елемент  $E_1$  домінує над елементом  $E_2$ , то клітина матриці, відповідна рядку  $E_1$  і стовпчику  $E_2$ , заповнюється цілим числом, а клітина, відповідна рядку  $E_2$  і стовпчику  $E_1$ , заповнюється зворотним до нього числом. Якщо елемент  $E_2$  домінує над  $E_1$ , то ціле число ставиться в клітину, відповідну рядку  $E_2$  і стовпчику  $E_1$ , а дріб проставляється в клітину, відповідну рядку  $E_1$  і стовпчику  $E_2$ . Якщо елементи  $E_1$  і  $E_2$  рівнозначні, то в обидві позиції матриці ставляться одиниці.

Для отримання кожної матриці експерт виносить  $n(n-1)/2$  суджень (тут  $n$  – порядок матриці парних порівнянь).

Нехай  $E_1, E_2, \dots, E_n$  – множина з  $n$  елементів (альтернатив) і  $v_1, v_2, \dots, v_n$  – відповідно їх ваги, або інтенсивності. Порівняємо попарно ваги, або інтенсивність, кожного елемента з вагою, або інтенсивністю, будь-якого іншого елемента множини по відношенню до загальної для них властивості або цілі (по відношенню до елемента-батька). В цьому випадку матриця парних порівнянь  $[E]$  має такий вигляд:

$$[E] = \begin{bmatrix} E_1 & E_2 & \dots & E_n \\ E_1 & v_1/v_1 & v_1/v_2 & \dots & v_1/v_n \\ E_2 & v_2/v_1 & v_2/v_2 & \dots & v_2/v_n \\ \dots & \dots & \dots & \dots & \dots \\ E_n & v_n/v_1 & v_n/v_2 & \dots & v_n/v_n \end{bmatrix} \quad (4)$$

Матриця парних порівнянь має властивість зворотної симетрії, тобто:

$$a_{ij} = 1/a_{ji}, \quad a_{ij} = v_i/v_j. \quad (5)$$

Ранжування елементів, які аналізуються з використанням матриць парних порівнянь  $[E]$ , здійснюється на підставі головних власних векторів, що отримуються в результаті обробки матриць. Обчислення головного власного вектора  $W$  позитивної квадратної матриці  $[E]$  проводиться на підставі рівності:

$$EW = \lambda_{\max} W, \quad (6)$$

де  $\lambda_{\max}$  – максимальне власне значення матриці  $[E]$ .

Для позитивної квадратної матриці  $[E]$  правильний власний вектор  $W$ , який відповідає максимальному

власному значенню  $\lambda_{\max}$ , з точністю до постійного співмножника  $C$  можна обчислити за формулою:

$$\lim_{k \rightarrow \infty} [E]^k e / (e^T [E]^k e) = CW, \quad (7)$$

де  $e = \{1, 1, 1, \dots, 1\}^T$  – одиничний вектор;  $k$  – показник ступеня;  $C$  – константа;  $T$  – знак транспонування.

Обчислення власного вектора  $W$  за виразом (2) виконується до досягнення заданої точності:

$$e^T |W^{(l)} - W^{(l+1)}| \leq \xi, \quad (8)$$

де  $l$  – номер ітерації, такий, що  $l = 1$  відповідає  $k = 1$ ;  $l = 2, k = 2$ ;  $l = 3, k = 3$  і т. д.;  $\xi$  – допустима похибка.

З достатньою для практики точністю можна прийняти  $\xi = 0,01$  незалежно від порядку матриці.

Максимальне власне значення обчислюється як

$$\lambda_{\max} = e^T [E]W. \quad (9)$$

Таким чином, здійснюється побудова ієрархії, яка використовується в подальшому для вибору найкращого сервера для розміщення конкретного екземпляру [13-16]. Як приклад розглянемо первинне виділення ресурсів для нового екземпляра в системі хмарних обчислень, що складається з 5 серверів з 30 запущеними екземплярами. На момент запуску екземпляра параметри функціонування обчислювальної системи описуються значеннями, зазначеними в табл. 2.

Таблиця 2 – Параметри функціонування системи хмарних обчислень

Параметр	Сервер 1	Сервер 2	Сервер 3	Сервер 4	Сервер 5
<b>Продуктивність</b>					
$N_{CPU}$	2	1	2	1	1
$N_{core}$	12	4	10	2	6
$V_{CPU}$	2.2	3	3.2	2	2.5
$V_{mem}$	128	16	256	8	32
$V_{net}$	1000	100	10000	1000	1000
<b>Надійність</b>					
$M_{fail}$	1000	1000	500	1000	1000
Фізичний стан обладнання	1	1	0.95	1	0.8
Якість збірки обладнання	0.87	0.96	0.92	0.99	0.78
Рівень компетенції персоналу	1	1	0.8	1	1
<b>Завантаженість</b>					
$L_{mem}$	40	2	120	4	19
$L_{CPU}$	0.1	0.4	0.6	0.02	0.2
$L_{disk}$	0.01	0.05	0.6	0.05	0.35
$T_{disk}$	10	8	102	12	65
$L_{net}$	0.01	0.02	0.52	0.05	0.03
$N_{vm}$	12	4	10	2	2

Виходячи з параметрів продуктивності, попередньо виконано підрозділення обчислювальної системи на зони - в даному випадку зони високої продуктивності (сервери 1 і 3) і зони низької продуктивності (сервери 2, 4 і 5). Приклад параметрів віртуальної машини, що запускається:

$$N_{CPU} = 2; N_{core} = 4; V_{mem} = 16 \text{ Гб}; V_{disk} = 20 \text{ Гб};$$

Виходячи зі значень показників запускається екземпляр, здійснюється вибір зони. Очевидно, що для цього екземпляра обрано зону високої продуктивності. Для аналізу альтернатив використовується отримана раніше на підставі експертних оцінок матриця парних порівнянь  $[E]$ , що містить вагові коефіцієнти значущості параметрів, актуальні для даної системи хмарних обчислень. В результаті оцінки альтернатив здійснюється вибір сервера для розміщення екземпляра.

**Оцінка ефективності** первинного виділення ресурсів проводилася шляхом вимірювання показника нерівномірності використання ресурсів сервера  $p$ . На цьому сервері було проведено запуск екземпляра при використанні різних методів. Критерієм порівняння було обрано параметр нерівномірності  $N_R$  завантаження ресурсів  $r$  сервера  $p$ :

$$N_R = \sqrt{\sum_{i=1}^n ((r_i - \bar{r}_i) / \bar{r}_i)^2}, \quad (10)$$

де  $r_i$  – навантаження на поточний ресурс сервера  $p$ ;  $\bar{r}_i$  – середня поточна завантаженість усіх ресурсів сервера  $p$ .

У ході оцінки ефективності, для кожного методу було проведено 100 пробних запусків віртуальної машини з довільними характеристиками у працюючій системі хмарних обчислень без зміння будь-яких параметрів навантаження. Результати оцінки (значення параметру  $N_R$ ):

- алгоритм першочергового виділення ресурсів: 0.48;
- простий алгоритм: 0.43;
- випадковий алгоритм: 0.44;
- алгоритм DRS: 0.26;
- алгоритм, що розроблено у статті: 0.14.

Як видно з наведених даних:

– алгоритм DRS показав значно кращий результат за рахунок врахування такого параметра, як завантаження центрального процесора.

– розроблений алгоритм на основі методу аналізу ієрархій показав найкращий результат зменшення нерівномірності ресурсів хосту, оскільки враховує весь набір показників функціонування хосту стосовно характеристик екземпляра, що запускається.

Таким чином, розроблений алгоритм виділення ресурсів продемонстрував свою ефективність при

первинному запуску екземпляра у функціонуючій системі хмарних обчислень.

### Висновки

У рамках дослідження проведено аналіз існуючих методів розподілу ресурсів в системах хмарних обчислень. Зокрема, алгоритми першочергового розподілу ресурсів. Були виявлені їх особливості та недоліки. Доведено необхідність використання інших підходів до розподілу ресурсів систем хмарних обчислень. Так, було запропоновано використання методу аналізу ієрархії для розподілу ресурсів у

системах хмарних обчислень. Наведено переваги та недоліки цього методу для використання у хмарних обчисленнях. Метод аналізу ієрархій було адаптовано для систем хмарних обчислень. Наведено алгоритм його використання. Проведена оцінка ефективності розробленого методу, який показав доцільність його використання при розподілі ресурсів у системах хмарних обчислень.

Завданням для подальшого розвитку є більш детальний аналіз наведених недоліків обраного методу та мінімізація ризиків їх впливу на показник нерівномірності розподілу ресурсів.

### СПИСОК ЛІТЕРАТУРИ

1. Sala-Zárate María, Colombo-Mendoza Luis (2012). Cloud computing: a review of PaaS, IaaS, SaaS services and providers. *Lámpsakos*, No. 7, pp. 47-57. DOI: <https://doi.org/10.21501/21454086.844>.
2. Гребенюк Д.С. (2018). Аналіз методів розподілення ресурсів у середовищах віртуалізації. *Системи управління, навігації та зв'язку*, № 6 (52), с. 98-103. DOI: <https://doi.org/10.26906/SUNZ.2018.6.098>.
3. Gulati A., Holler A., Ji M., Shanmuganathan G., Waldspurger C., Zhu X. (2012). VMware distributed resource management: Design, implementation and lessons learned. *VMware Technical Journal*, No. 1, pp. 45-64.
4. Микрюков А.А., Князев А.А. (2009). Об одном методе решения задачи многокритериального выбора. Сборник трудов II международной научно-практической конференции «Информационные технологии в образовании, науке и производстве», Серпухов, с. 241-243.
5. Технологии принятия решений: метод анализа иерархий [Электронный ресурс] - Режим доступа: <http://citforum.ru/consulting/BI/resolution>
6. Саати Т.Л. (2015). Об измерении неосязаемого. Подход к относительным измерениям на основе главного собственного вектора матрицы парных сравнений, *Cloud Of Science*, Том 2, № 1, с. 5-39.
7. Саати Т.Л. (2016). Относительное измерение и его обобщение в принятии решений. Почему парные сравнения являются ключевыми в математике для измерения неосязаемых факторов, *Cloud Of Science*, Том 3, № 2, с. 171-262.
8. Ногин В.Д. (2004). Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев. *Журнал вычислительной математики и математической физики*, Том 44, № 7, с. 1261-1270.
9. Анохин А.М. (1997). Методы определения коэффициентов важности критериев. *Автоматика и телемеханика*, № 8, с. 3-35.
10. Bauer E., Adams R. (2012). *Reliability and Availability of Cloud Computing*. Wiley-IEEE Press, 352 p.
11. Беркетов Г.А., Микрюков А.А. (2008). Сравнительный анализ сложных систем с использованием нечетких критериев. *Информационные технологии в проектировании и производстве*, № 1, с. 80-83.
12. Михалевич М.В. (1994). Замечания к дискуссии Дж. Дайера и Т. Саати. *Кибернетика и системный анализ*, № 1, с. 97-102.
13. Зиков І. С., Кучук Н. Г., Шматков С. І. Синтез архітектури комп'ютерної системи управління транзакціями e-learning. *Сучасні інформаційні системи*. 2018. Т. 2, № 3. С. 60-66. DOI: <https://doi.org/10.20998/2522-9052.2018.3.10>
14. Nechausov A., Mamusu I., Kuchuk N. Synthesis of the air pollution level control system on the basis of hyperconvergent infrastructures. *Сучасні інформаційні системи*. 2017. Т. 1, № 2. С. 21-26. DOI: <https://doi.org/10.20998/2522-9052.2017.2.04>
15. Кучук Н.Г., Гавриленко С.Ю., Лукова-Чуйко Н.В., Собчук В.В. Перерозподіл інформаційних потоків у гіперконвергентній системі / С.Ю. Гавриленко. *Сучасні інформаційні системи*. 2019. Т. 3, № 2. С. 116-121. DOI: <https://doi.org/10.20998/2522-9052.2019.2.20>
16. Андрейчиков А.В., Андрейчикова О.Н. (2000). Анализ, синтез, планирование решений в экономике. Финансы и Статистика, Москва, 151 с.

Received (Надійшла) 15.06.2020

Accepted for publication (Прийнята до друку) 05.08.2020

### Initial cloud computing resource allocation method based on hierarchy analysis

D. Hrebeniuk, V. Davydov

**Abstract.** The subject of the study in the article are models and methods of load and resources reallocation in cloud computing systems, in particular, which are based on the infrastructure as a service model. **The goal of the work** is to optimize initial resource allocation in cloud computing systems by adapting the hierarchy analysis method for implementation in cloud computing. This will allow new virtual machines to be launched with minimal performance degradation for already running applications. The following **tasks** were solved in the article: the hierarchy analysis model in cloud computing using feasibility research; adapting hierarchy analysis method to cloud computing; implemented method effectiveness evaluation. To solve set tasks, theoretical research approaches and **methods** were used, which are based on the scientific positions of static, functional and systemic analyses. The following **results** were obtained – the hierarchy analysis method using to allocate resources in cloud computing systems is proposed. The advantages and disadvantages of this method for use in cloud computing are presented. The hierarchy analysis method has been adapted for cloud computing systems. The algorithm of its use is given. The developed method effectiveness estimation was carried out, which showed the feasibility of using it in cloud computing systems resource allocation. **Conclusions.** The improvement in the method of initial cloud computing resource allocation has increased the ability of these systems to start new virtual machines with a minimum decrease in the performance of already running applications.

**Keywords:** cloud computing systems, hierarchy analysis, allocation.