

*Секція: Математичні методи, моделі та інформаційні технології в економіці*

**Камінський Олег Євгенович**

*кандидат економічних наук,*

*доцент кафедри інформаційного менеджменту*

*Київський національний економічний університет*

*імені Вадима Гетьмана*

*м. Київ, Україна*

## **МЕТОДИ ОЦІНЮВАННЯ ПРОЕКТІВ ХМАРНИХ СЕРВІСІВ ЗА РІВНЕМ QoS**

Для розробки та проектування хмарних сервісів необхідним є прийняття значної кількості технічних і управлінських рішень. Завдання ускладнюється тим, що кількісний і якісний розвиток хмарної парадигми привів до потреби розглядати її як складну інформаційну систему, котра може охоплювати тисячі користувачів і містити сотні тисяч розподілених елементів, котра підтримує динамічне масштабування, котра використовує різні комунікаційні та обчислювальні сервіси та постійно взаємодіє з іншими інформаційними системами. Отже, очевидна актуальність вивчення економічного аспекту процесів порівняння, вибору і використання хмарних сервісів у діяльності підприємств і організацій. На рівні SaaS власники сервісів мають можливість контролювати якість обслуговування (QoS), для того щоб провадити оптимізацію відповідно до власних завдань [1]. Залежно від сфери використання та типу хмарних сервісів можуть використовуватися різні показники якості QoS, такі як час відгуку на запити користувача.

Задача вибору структурних елементів хмарного сервісу має наступний вигляд:

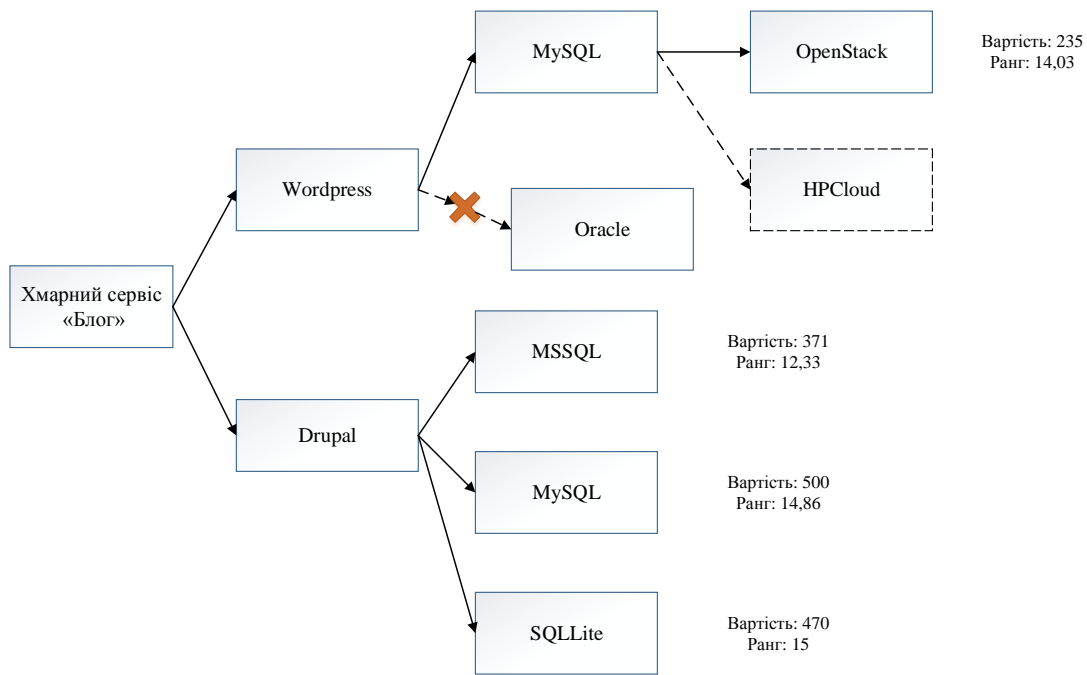
$$TCO(CT) \rightarrow \min,$$

$$CT \in MCT(M), f(CT) \geq f(P), v g_i \in VG(CT), \forall_i = \overline{1, Nvg}. \quad (1)$$

Потрібно вибрати структурні елементи мінімальної вартості, які дозволять реалізувати вказаний проект хмарного сервісу. Першим обмеженням  $MCT$  є набір допустимих композицій структурних елементів. Друге обмеження гарантує, що характеристики хмарного сервісу, які ми можемо отримати, вибравши певну групу структурних елементів, не можуть бути нижче, ніж специфікації сервісу, визначені технічним завданням. Третє обмеження встановлює, що функції сервісу можуть бути реалізовані за допомогою вибраного набору структурних елементів сервісу.

Характеристики проекту сервіса включають в себе функціональні та нефункціональні вимоги. Функціональні вимоги – це опис глобальних необхідних функцій сервісу на високому рівні абстракції. Нефункціональні вимоги- це коефіцієнти, задані користувачем для кожного параметра QoS, виходячи з його пріоритету. Сума всіх призначених значень не має перевищувати 10. Як цільову функцію ми використаємо модель сукупної вартість володіння хмарним сервісом TCO (Total Cost of Ownership). Модель TCO складається з розрахунку таких показників: вартість оренди хмарних ресурсів; вартість навчання персоналу; затрати праці на побудову сервісу; затрати праці на експлуатацію сервісу; затрати праці користувачів на роботу із сервісом.

Розглянемо приклад. Ми розробляємо кілька робочих проектів хмарного сервісу, що відповідають функціональним вимогам користувача. Ми розглядаємо обмеження структурних елементів перед створенням робочих проектів. На рис.1 наведено схему аналізу проектів сервісу, сформовану для попередньо описаного хмарного сервісу "движок блога".



**Рис. 1. Схема аналізу проектів розробки хмарного сервісу «Блог»**

Схема показує, що фреймворк WordPress для розробки блогу не може використовувати СУБД Oracle; оскільки опис сервіса WordPress визначає жорсткі обмеження щодо баз даних, та вимагає використання СУБД MySQL. Аналіз також показує, що фреймворк Drupal, як структурний елемент сервісу обов'язково потребує використання баз даних, але може використовувати різні СУБД, а не тільки конкретну, як у випадку з WordPress. Після ранжування робочих проектів хмарного сервісу, композиція структурних елементів з найвищим рейтингом розгортається на попередньо обраному IaaS. Для проектів хмарного сервісу ми обчислюємо рейтинг відповідності структурних елементів вимогами QoS. Ранг проектів сервісів розраховується на основі коефіцієнтів, пов'язаних із атрибутами QoS, і призначається користувачем на основі його пріоритетів. Доступні два сценарії ранжування коефіцієнтів.

Наприклад,  $C_i$  –  $i$ -й проект хмарного сервісу, а  $IQ_j$  –  $j$ -й індикатор QoS, а  $RN$  – ранг проекту:

$$RN(C_i, IQ_j) = \begin{cases} RN_{\text{верхнє}} \\ RN_{\text{нижнє}} \end{cases} \quad (2)$$

Сценарій №1: чим вище значення атрибута (наприклад, доступність послуги), тим кращим є цей проект сервісу. У цьому випадку ранг, пов'язаний із цим атрибутом QoS для даного сервісу, розраховується таким чином:

$$RN_{\text{верхнє}} = \frac{Z}{Z_{\text{max}}} * K; \quad (3)$$

де:

$Z$  - це значення атрибута QoS для даного проекту сервісу;

$Z_{\text{max}}$  - це максимальне значення атрибуту серед всіх проектів;

$K$  - це коефіцієнт, раніше закріплений за атрибутом замовником.

Сценарій №2: чим менше значення атрибута (наприклад, час відгуку), тим кращим є цей проект сервісу. У цьому випадку ранг, пов'язаний із цим атрибутом для заданого проекту, розраховується таким чином:

$$RN_{\text{нижнє}} = (1 - \frac{Z}{Z_{\text{max}}}) * K; \quad (4)$$

Припустимо, що показник  $RN(C_i)$  буде глобальним рейтингом щодо всіх показників QoS для  $i$ -го проекту хмарного сервісу. При виборі проекту сервісу перевага надається проектам з найвищим рейтингом.

$$RN(C_i) = \sum_{j=1}^n RN(C_i, IQ_j); \quad (5)$$

В табл.1 наведено параметри QoS, пов'язані з двома відповідними проектами хмарного сервісу, сформованими для попередньо описаного сценарію "движок блога".

Таблиця 1

### Розраховані значення коефіцієнтів QoS для проектів хмарного сервісу

Параметри QoS	Хмарний сервіс на фреймворку WordPress	Хмарний сервіс на фреймворку Drupal
Конфіденційність даних	9	6
Втрата даних	8,97	0,39
Функціональна придатність	10	10
Час відгуку	224	398

Використовуючи описаний вище спосіб розрахунку рангу коефіцієнтів, ми обчислюємо рейтинг, пов'язаний із проектами хмарного сервісу "движок блогу", відповідно на базі використання фреймворків WordPress і Drupal. Дані підрахунку свідчать, що проект сервісу, структурним елементом якого буде фреймворк Drupal має кращий рейтинг, ніж проект з використанням фреймворку WordPress.

$$RN_{wordpress} = \left(\frac{9}{9} * 4\right) + \left(\left(1 - \frac{8,97}{8,97}\right) * 4\right) + \left(\frac{10}{10} * 1\right) + \left(\left(1 - \frac{224}{398}\right) * 1\right);$$

$$RN_{wordpress} = 5,44;$$

$$RN_{drupal} = \left(\frac{6}{9} * 4\right) + \left(\left(1 - \frac{0,39}{8,97}\right) * 4\right) + \left(\frac{10}{10} * 1\right) + \left(\left(1 - \frac{398}{398}\right) * 1\right);$$

$$RN_{drupal} = 7,5;$$

Ми не розглядаємо ранг лише одного структурного елементу хмарного сервісу, а також складаємо ранг кожного елемента сервіса, що бере участь у робочому проекті. Визначається рейтинг структурних елементів сервісу, які знаходяться на рівнях PaaS та IaaS хмари, після чого визначається глобальний рейтинг всього проекту. В процесі розробки проекту та вибору технологій альтернативні варіанти порівнюються за QoS. Розгортається в хмарі відповідно проект з найвищим рейтингом. Використання комплексу економічних, математичних та імітаційних моделей для вирішення окреслених проблем допоможе розробникам приймати економічно обґрунтовані рішення щодо розробки й упровадження хмарних сервісів.

### **Література**

1. G. Singh, P. Jain and I. Gulati, "Process Model for Cloud Service Engineering", *International Journal of Computer Applications*, vol. 36, no. 8, (2011), pp. 24-29.