

М. В. ЄВЛАНОВ, В. В. ШУТЬКО

РОЗРОБКА БАЗОВОЇ МОДЕЛІ ВИБОРУ СТРАТЕГІЇ ХМАРНОЇ МІГРАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Хмарна міграція інформаційних системи наразі є одним з критичних перехідних етапів розвитку підприємства на шляху до цифрової трансформації. Розглянуто проблему вибору стратегії хмарної міграції інформаційної системи на передміграційній стадії як задачу досягнення цільових характеристик якості та продуктивності, прийнятого рівня витрат і керуваності впровадження в умовах обмежених вихідних даних. Визначено, що наявні підходи до планування міграції часто є різноплановими та фрагментарними. Запропоновано формалізацію вибору стратегії як порівняння альтернатив у спільному просторі ключових показників ефективності. Введено цільовий профіль показників як набір мінімально прийнятних порогових значень після міграції, а також ваги важливості, які відображають пріоритети зацікавлених сторін і дозволяють явно фіксувати, які вимоги є критичними. Визначено придатність кожної альтернативи мірою недосягнення цільових порогів за показниками, де прогнозований стан не відповідав вимогам. Враховано компроміс між досягненням цільового профілю та складністю реалізації, що дозволив контролювати баланс між очікуваними покращеннями і ресурсними витратами на перетворення інформаційної системи. Виконано експериментальну перевірку моделі на заданому наборі вхідних даних, що підтвердило детермінованість і відтворюваність результатів. За фіксованих початкових значень показників, правил нормування, порогових цілей, ваг важливості та параметрів компромісу модель формувала однозначне ранжування стратегій і дозволила ідентифікувати, які показники формували основний дефіцит та в яких напрямках необхідні зміни для його зменшення. Обґрунтовано, що така постановка підсилила прозорість та аргументованість вибору стратегії хмарної міграції. Подальші напрями дослідження визначено як калібрування параметрів моделі на емпіричних прикладах, аналіз чутливості результатів до ваг і порогів, урахування невизначеності прогнозів показників, а також розширення механізмів сценарного порівняння, щоб підвищити переносимість рекомендацій для різних доменів і типів інформаційних систем.

Ключові слова: інформаційна система, ключові показники ефективності, модель, стратегія, хмарна міграція.

М. IEVLANOV, V. SHUTKO

DEVELOPMENT OF A BASIC MODEL FOR CHOOSING AN INFORMATION SYSTEM CLOUD MIGRATION STRATEGY

Cloud migration of information systems is currently one of the critical transitional stages of enterprise development on the way to digital transformation. The problem of choosing a strategy for cloud migration of the information system at the pre-migration stage is considered as a task of achieving the target characteristics of quality and productivity, an acceptable level of costs and manageability of implementation in conditions of limited initial data. It is determined that the existing approaches to migration planning are often diverse and fragmented. The formalization of the choice of strategy as a comparison of alternatives in the common space of key performance indicators is proposed. The target indicator profile is introduced as a set of minimum acceptable thresholds after migration, as well as important weights that reflect the priorities of stakeholders and allow you to explicitly capture which requirements are critical. The suitability of each alternative as a measure of failure to achieve the target thresholds for indicators where the predicted state did not meet the requirements is determined. The compromise between the achievement of the target profile and the complexity of implementation is considered, which made it possible to control the balance between the expected improvements and resource costs for the transformation of the information system. Experimental verification of the model on a given set of input data was performed, which confirmed the determinism and reproducibility of the results. With fixed initial values of indicators, normalization rules, threshold goals, weights of importance and compromise parameters, the model formed an unambiguous ranking of strategies and made it possible to identify which indicators formed the main deficit and in which directions changes were needed to reduce it. It is substantiated that such a formulation strengthened the transparency and reasoning of the choice of a cloud migration strategy. Further areas of research are defined as calibration of model parameters on empirical examples, analysis of the sensitivity of results to weights and thresholds, considering the uncertainty of indicator forecasts, as well as expanding the mechanisms of scenario comparison to increase the portability of recommendations for different domains and types of information systems.

Keywords: cloud migration, information system, key performance indicators, model, strategy

Вступ. Хмарна міграція інформаційних систем (ІС) у сучасній цифровій трансформації розглядається не лише як технічне перенесення компонентів у середовище провайдера, а як комплексна зміна способу керування обчислювальними ресурсами, ризиками та економікою цифрової інфраструктури компанії. Саме тому найбільш критичним стає етап передміграційного планування, де формуються цілі міграції, очікувані показники результативності та обмеження бізнесу, а також закладаються припущення щодо майбутніх профілів навантаження й операційних витрат. Практика показує, що без формалізованого планування організації часто входять у міграцію без належно обраної стратегії через низку факторів, що створює високий ризик помилок у подальших технічних і фінансових рішеннях.

На передміграційній стадії однією з ключових прикладних проблем є оптимізація витрат ресурсів для планування (часових та фінансових витрат, Net Present Value, NPV), потрібних для підготовки доказової бази щодо доцільності міграції, одночасно зі збереженням прийнятної точності оцінок. Тут виникає суперечність: щоб обґрунтувати рішення, потрібно зібрати дані про наявну ІТ-інфраструктуру, взаємозалежності застосунків, пікові режими роботи та приховані статті витрат, але сам збір і валідація цих даних є дорогим та тривалим процесом, особливо на рівні портфеля застосунків. Відповідно, дослідження, що орієнтовані на портфельне оцінювання та врахування «прихованих витрат» прямо підкреслюють потребу у попередніх оцінках як механізмі зниження ризику перевищити бюджету та зриву строків [1].

© М. В. Євланов, В. В. Шутько, 2026

Другою проблемою є невизначеність продуктивності та масштабованості цільового середовища, яка на пряму впливає на вибір стратегії міграції. Якщо планування спирається на усереднені або нерепрезентативні профілі навантаження, організація ризикує або переплатити за надмірне резервування потужностей, або отримати деградацію продуктивності ІС після перенесення. У цьому контексті важливими є методи емпіричного бенчмаркінгу, які пропонують виконувати контрольовані експерименти для різних комбінацій типу «навантаження × ресурси» і перевіряти досягнення визначених угод про рівень сервісу (Service Level Agreement, SLA), але водночас керувати компромісом між відтворюваністю результатів і практичною вартістю (часом) таких експериментів. Це задає науково обґрунтовану основу для передміграційного аналізу та зіставлення варіантів розгортання ще до фактичного перенесення ІС [2].

Окремо постає проблема вибору стратегії хмарної міграції та доведення її раціональності для бізнесу. На практиці стратегія не зводиться до «техніки перенесення», а вимагає багатокритеріального компромісу між витратами, очікуваними вигодами, ризиками, організаційною готовністю та регуляторними вимогами. У цьому напрямі релевантними є гібридні фреймворки прийняття рішень, які поєднують кількісну економічну аргументацію (зокрема фінансові оцінки на кшталт чистої поточної вартості) з експертними процедурами, придатними для неповних або «нечітких» вимог на ранній стадії. Такий клас підходів важливий тим, що переводить обґрунтування стратегії з рівня інтуїтивних припущень у рівень відтворюваного прийняття рішень, де видно логіку вибору та джерела невизначеності [3].

Потенційна вигода від вирішення останньої проблеми полягає у зниженні невизначеності та витрат на прийняття рішень на передміграційній стадії. Так, формалізоване планування дозволило б скоротити час і фінансові витрати на підготовку доказової бази, підвищити відтворюваність оцінок витрат і прогнозів продуктивності. Зменшення ймовірності хибного вибору стратегії та, як наслідок, мінімізація ризиків перевитрат бюджету, зриву строків і накопичення технічного боргу після перенесення ІС було б другою перевагою. Актуальність дослідження зумовлена тим, що хмарні платформи стають базовою інфраструктурою цифрових сервісів, архітектури ІС ускладнюються, а регуляторні та безпекові вимоги посилюються, тому фрагментарний підхід до планування стає дедалі менш релевантним. Таким чином підкреслюється необхідність проведення наукових досліджень саме за цією проблемою.

Аналіз літературних джерел. У [1] Було запропоновано розширений фреймворк Cloud Computing Considerations for Companies (CCCC) для передміграційного планування на рівні портфеля

застосунків із додатковими фазами Application Portfolio Profiling (APP) та Application Portfolio Assessment (APA), де формувалися базові лінії навантаження і конфігурацій, виконувалася портфельна оцінка та надавалися рекомендації щодо розміщення у публічній/приватній хмарі (Technical Platform Recommendation, TPR) і варіанта перенесення «as-is» або з модернізацією; ухвалення рішення підтримувалося Cloud Decision Framework на основі Rule-based Reasoning (RBR) і Case-based Reasoning (CBR), а фінансове обґрунтування оформлювалося через Financial Viability Assessment (FVA) як «bottom-up» оцінювання витрат.

Перевагами запропонованого підходу було те, що він систематизував планування у вимірюваних параметрах портфеля, підвищував стійкість рішень за неповних вимог завдяки RBR/CBR та забезпечував більш повну картину витрат за рахунок FVA.

Однак, недоліками такого підходу було те, що він зосереджував рішення переважно на виборі платформи (публічна/приватна хмара) та загального варіанта перенесення («as-is» або з трансформацією), тоді як вибір між альтернативами різної глибини перетворень залишався слабо параметризованим і не завжди забезпечував прозору, відтворювану аргументацію компромісу між витратами, вигодами та ризиками. Оскільки RBR спиралося на експертну класифікацію критеріїв як «Required/Optional», рекомендації суттєво залежали від суб'єктивних трактувань критичності вимог, що ускладнювало стандартизацію та порівнюваність рішень. CBR, за браку релевантних прецедентів, могло пропонувати формально подібні аналогії, які не відображали ключових обмежень, знижуючи надійність висновків за неповних ранніх даних.

У [2] було запропоновано формалізувати оптимізацію ресурсів на передміграційній стадії як серію SLO-експериментів (Service Level Objective, ціль рівня обслуговування), де перевірялося, чи здатна ІС з фіксованою ресурсною конфігурацією витримати задану інтенсивність навантаження з одночасним виконанням усіх SLO. Для цього ІС розгорталася з певними ресурсами, генерувався сталий потік навантаження, моніторилися метрики з відкиданням інтервалу «розігріву», виконувалися повтори, а результати агрегувалися відповідно до конкретного SLO. Далі оцінювалися дві функції масштабованості – мінімальні достатні ресурси для кожного рівня навантаження («метрика попиту») та максимальна інтенсивність навантаження для кожної конфігурації («метрика ємності»), що використовувалося для оптимального підбору розміру та кількості хмарних ресурсів.

Перевагами запропонованого підходу було те, що він задавав кількісно відтворювану процедуру, де досяжність вимог інтерпретувалася безпосередньо через SLO/KPI як «виконується/не виконується», а доказовість і вартість оцінювання керувалися параметрами експериментів та стратегією пошуку.

Однак, недоліками такого підходу було те, що прискорені стратегії спиралися на припущення «більше ресурсів означає не гірше виконання SLO», яке могло порушуватися (зокрема для подія-орієнтованих мікросервісів), створюючи ризик хибних рекомендацій при автоматизованому звуженні пошуку. Додатково метод вимагав репрезентативного навантаження, коректних SLO та стабільного експериментального середовища, тому переносимість висновків між контекстами обмежувалася. Також підхід переважно обґрунтовував ресурсну достатність і альтернативи розгортання, але не давав формалізованого механізму переходу до варіантів глибших перетворень, коли SLO досяжні лише за високої ціни або нестабільної поведінки.

У [3] було запропоновано комплексну модель прийняття рішення для планування хмарної міграції, у якій бізнес-процес формалізувався, а функціональні та нефункціональні вимоги переводилися у вимірювані критерії, зокрема пов'язані з профілем навантаження та споживанням ресурсів. У моделі «шаблон використання» трактувався як визначальний чинник варіанта розгортання. Ресурсні потреби класифікувалися як передбачувані та непередбачувані, для короткочасних піків обґрунтовувалася доцільність використання еластичних ресурсів, а прогнозування часових рядів пропонувалося для побудови передміграційної «пісочниці» оцінювання майбутніх потреб. Оптимізаційна частина реалізовувалася через порівняння альтернативних сценаріїв у термінах сукупної вартості та ризик-втрат. Зменшення суб'єктивності підтримувалося поєднанням Delphi з методом аналізу ієрархій (Analytical Hierarchy Process, АНР – для попарного порівняння у структурі багатокритеріального вибору); практичність демонструвалася NPV-аналізом із висновком на користь приватного дата-центру в довгостроковій перспективі та гібридного розгортання для пікових режимів.

Перевагами запропонованого підходу було те, що він поєднував ресурсне планування з економічною аргументацією в єдиній процедурі та підтримував багатокритеріальне зіставлення альтернатив із частковим керуванням суб'єктивністю через Delphi і АНР.

Однак, недоліками такого підходу було те, що ключові «втрати» та приховані загрози (зокрема vendor lock-in, витрати на повторне розгортання) переважно ідентифікувалися та оцінювалися експертно, а отже залишалися організаційно-залежними і важко відтворюваними у стандартизованому вигляді. Навіть за наявності Delphi та АНР зберігалася невизначеність, оскільки АНР суттєво залежав від суджень і досвіду аналітиків, що могло змінювати ваги критеріїв і підсумкове ранжування сценаріїв при повторенні оцінювання іншою групою. Додатково під час оцінювання сценаріїв частина змінних могла ігноруватися або пом'якшуватися залежно від дизайну системи та типу розгортання, що підвищувало гнучкість, але

послаблювало стандартизацію, порівнюваність і переносимість результатів між різними кейсами, особливо коли потрібно узгоджено зіставляти альтернативи з різними профілями ризиків і вигод.

У [4] було запропоновано практико-орієнтований підхід до передміграційного планування перенесення on-premises веб-застосунків у хмару, у якому ключовою задачею вважалося точне оцінювання та подальша оптимізація витрат, а реалізацію було цілеспрямовано звужено до Amazon Web Services (AWS). У межах цього фокусу були сформовані AWS-орієнтовані моделі оцінювання вартості типових сервісів і ресурсів на основі документації з ціноутворення та калькуляторів AWS. Описано використання CloudWatch, Cost Explorer і Trusted Advisor для аналізу трендів витрат і патернів споживання, а оптимізацію формалізовано як мінімізацію загальної вартості (обчислення, сховище, перенесення даних, резервне копіювання, мережеві компоненти та «ефект зв'язків» між конфігураціями). Для пошуку близькооптимальних конфігурацій було застосовано гібридний алгоритм GA-SA (Genetic Algorithm – Simulated Annealing) з перевіркою на 17 еталонних benchmark-функціях.

Перевагами запропонованого підходу було те, що він задавав формалізовану, інтерпретовану модель витрат і механізм пошуку економічно кращих конфігурацій у межах конкретного хмарного провайдера.

Однак, недоліками такого підходу було те, що звуження до AWS знижувало переносимість результатів на мультимарні або альтернативні сценарії та робило висновки чутливими до змін у моделях ціноутворення й сервісних політиках AWS. Додатково ефективність GA-SA була продемонстрована на абстрактних benchmark-функціях, що не гарантувало аналогічної якості на реальних профілях навантаження та під реальними міграційними обмеженнями, тобто залишалася методологічна прогалина між синтетичним тестом і польовою перевіркою. Також модель була зорієнтована переважно на витратні компоненти, через що при зіставленні альтернатив із різною глибиною перетворень (lift-and-shift, re-platforming, re-architecting) вимагався додатковий набір критеріїв для узгодженого врахування очікуваних вигод, ризиків і нефінансових обмежень, а не лише мінімізації вартості.

У [5] було подано емпірично обґрунтовану «карту» міграції до мікросервісної архітектури в хмарі як довготривалого ітеративного процесу з двома взаємопов'язаними режимами змін: системним (стратегічним, організаційним і архітектурним) та технічним (операційним, реалізаційним, спринтовим), які відбувалися паралельно. Для обох режимів було виокремлено планування, виконання, побудова підтримувальних артефактів, а міграційні дії були структуровані у вигляді 14 активностей (6 – системні, 8 – технічні), конкретизованих через 53 «solution outcomes» як типові результати інженерної та

управлінської роботи. Емпірична база поєднувала 19 інтерв'ю з практиками (16 компаній) та ручний аналіз 215 дискусій із ресурсу StackOverflow із застосуванням технік «grounded theory» (обґрунтована теорія).

Перевагами запропонованого підходу було те, що він задавав узгоджений опис планування та координації міграції на системному й технічному рівнях у формі прикладних активностей і конкретних результатів, зменшуючи фрагментарність рекомендацій щодо переходу до мікросервісів.

Однак, недоліками такого підходу було те, що модель мала переважно описово-процедурний характер і не задавала кількісної, відтворюваної процедури зіставлення альтернатив за витратами, очікуваними вигодами та ризиками, через що її важко було використовувати для формалізованого обґрунтування вибору між варіантами різної глибини перетворень. Додатково репрезентативність інтерв'ю не гарантувалася через добір з персональних мереж і добровільну участь, що могло зміщувати набір активностей та «solution outcomes» у бік типових практик конкретних середовищ. Також використання даних зі StackOverflow потенційно підсилювало домінування інженерних перспектив, через що управлінські, організаційні та регуляторні обмеження могли бути недопредставлені, а отже частина критичних факторів, що впливають на вибір стратегії міграції, могла не бути зафіксована в моделі.

У [6] було запропоновано підхід до міграції критичних SAP-рішень (Systems, Applications, and Products in Data Processing, німецька компанія-розробник програмного забезпечення) і баз даних з опремис до Microsoft Azure. Підхід центрований на відмовостійкій цільовій архітектурі та поетапній організації переходу через фази оцінювання, планування, виконання та подальшої оптимізації. Передміграційне планування реалізувалося інструментально через Azure Migrate для аналізу залежностей, оцінювання продуктивності й попередньої оцінки вартості, із використанням Azure Site Recovery для безперервності/відновлення та Database Migration Service для перенесення СУБД з мінімальним простоем. Додатково формувався інвентар застосунків і БД із взаємозалежностями та еталон метрик «до/після» для верифікації. Вибір подавався як застосування lift-and-shift для швидкого перенесення без суттєвих модифікацій або re-architecting із рефакторингом під «cloud-native» можливості (включно з мікросервісними та «serverless» підходами), а післяміграційна оптимізація підтримувалася Azure Monitor, Azure Cost Management, Reserved Instances та Autoscale.

Перевагами запропонованого підходу було те, що він задавав практично відтворюваний покроковий сценарій, підкріплений конкретними Azure-інструментами, і акцентував вимоги високої доступності, відновлення, мультирегіональності та регуляторної відповідності.

Однак, недоліками такого підходу було те, що вибір обмежувався двома високорівневими сценаріями (lift-and-shift або re-architecting) без багатокритеріальної методики, вагових коефіцієнтів і формалізованих порогів прийнятності, через що логіка зіставлення альтернатив залишалася недостатньо прозорою та важко відтворюваною. Попри наявність окремих KPI-орієнтирів, не було запропоновано кількісної моделі, яка б явно пов'язувала характеристики ІС/компонентів із KPI та процедурою обґрунтування вибору, тому економічні й ризикові аргументи залишалися переважно описовими. Додатково специфічність до Microsoft Azure знижувала переносимість на мультихмарні стратегії або інші платформи, а твердження про економічний ефект подавалися як загальні оцінки без демонстрації на конкретному прикладі міграції.

У [7] було запропоновано міграційно-орієнтований механізм балансування навантаження в IaaS-середовищі (Infrastructure as a Service, інфраструктура як сервіс), у якому оптимізація використання ресурсів досягалася через прогнозування майбутніх навантажень і подальший підбір цільових міграцій віртуальних машин (VM). Рішення будувалося на інтеграції gene expression programming (GEP), що формувало регресійні моделі навантаження VM, та GA, який на основі прогнозів обирав перепризначення VM на фізичні хости (VMH) для вирівнювання навантаження після міграцій. Оптимізація формалізувалася функцією пристосованості, яка агрегувала мінімізацію вартості міграцій та максимізацію збалансованості за умови відсутності прогнозного перевантаження. Алгоритм викликався періодично, а поточний стан використовувався як відправна точка. Перевірка виконувалася в реальному середовищі за метриками Load-Balance status (LB) і Balancing Efficiency (BE).

Перевагами запропонованого підходу було те, що оптимізація спиралася на прогнозний стан інфраструктури після міграцій і явно враховувала вартість міграцій як керований параметр, що підвищувало контрольованість компромісу між стабільністю сервісу та якістю вирівнювання.

Однак, недоліками такого підходу було те, що він описував переважно операційну оптимізацію розміщення VM у межах IaaS і не давав критеріїв для зіставлення альтернатив із різною глибиною перетворень системи, оскільки «вартість» у моделі інтерпретувалася як штраф за міграцію VM, а не як повний економічний ефект або сукупні витрати переходу. Також фокус на CPU-навантаженні й централізованому балансуванні без урахування мережі, пам'яті чи графічного процесора обмежував придатність моделі для реальних багаторесурсних профілів, де рішення про доцільність і «раціональність» перенесення визначається комплексом ресурсних і нефункціональних обмежень. Додатково застосованість підходу була обмежена малим масштабом експериментів та використанням штучно згенерованих даних.

У [8] було запропоновано підхід CIRP (Continuous Integration of architectural Performance Models with parametric dependencies), орієнтований на підтримку передпроектних і передексплуатаційних рішень щодо конфігурації ІС на основі архітектурної моделі продуктивності. CIRP забезпечував сценарний аналіз «what-if» для різних варіантів навантаження, дизайну та розгортання, а оптимізацію ресурсів реалізовував через безперервне підтримання актуальності параметризованої моделі. Замість повного перерахунку після змін виконувалося інкрементальне калібрування лише «зачеплених» параметрів із застосуванням адаптивного моніторингу та самовалідації, яка контролювала точність прогнозів і ініціювала перекалібрування неточних фрагментів. Для уточнення параметричних залежностей використовувалися статистичні методи.

Перевагами такого підходу було те, що він дозволяв обґрунтовувати потреби в ресурсах і продуктивності на основі архітектурних прогнозів та зменшувати вартість спостережуваності завдяки інкрементальному калібруванню й самовалідації, зберігаючи придатність для порівняння альтернатив без суцільних експериментів.

Однак, недоліками такого підходу було те, що він фокусувався переважно на продуктивності та ресурсоспоживанні, тому зіставлення альтернатив різної глибини перетворень не задавалося як формалізована процедура і вимагало додаткового набору критеріїв (вартість, ризики, залежності), без якого аргументація могла вважатися суб'єктивною. Додатково застосування CIRP вимагало наявності артефактів архітектурного моделювання та коректного співставлення між моделлю і кодом, а також передбачало ручні рішення для усунення неоднозначностей у побудові System Model, що знижувало відтворюваність і ускладнювало масштабування на великі проекти. Нарешті, залежність від циклів налаштування і збору телеметрії могла обмежувати практичність у середовищах із жорсткими обмеженнями на експерименти або спостережуваність, що зменшувало надійність прогнозів і, відповідно, обґрунтованість альтернатив.

У [9] було запропоновано ієрархічну кількісну модель Cloud-QM, побудовану на основі стандартів ISO 25011 (SQuaRE) та розширену хмарно-специфічними атрибутами, де структуру моделі сформовано через Delphi-процедуру узгодження експертних суджень. Модель задавала стандартизоване оцінювання альтернативних хмарних продуктів за ієрархією «вимір якості – підвимір – метрика» та формалізованими KPI з правилами інтерпретації. В контексті ресурсів і еластичності були введені метрики Resource Behavior (мінімально потрібні RAM, CPU та дискова пам'ять) і Resource Allocation Adjustment (час масштабування та діапазон кроку виділення пам'яті на клієнта). Застосування моделі було продемонстровано через порівняння двох продуктів, а вибір інтерпретувався як обґрунтоване

ранжування цільового продукту з урахуванням бізнес-обмежень.

Перевагами такого підходу було те, що він забезпечував структуроване й порівняльне оцінювання альтернатив на основі стандартизованих KPI, знижуючи довільність інтерпретацій та підсилюючи відтворюваність висновків щодо ресурсних вимог, еластичності, переносимості та відповідності регуляторним умовам.

Однак, недоліками такого підходу було те, що валідацію було виконано лише на PaaS-кейсах, а Delphi-узгодження спиралося на невелику групу експертів, що обмежувало узагальнюваність і стійкість ієрархії метрик та їх інтерпретацій у ширших контекстах. Також у моделі не були зафіксовані ваги вимірів якості, через що відсутній формалізований механізм узгодження компромісів між групами критеріїв. Додатково «ресурсна оптимізація» мала переважно порівняльний характер і залежала від доступності коректних вихідних даних, тому за дефіциту або неоднорідності даних доказовість оцінок знижувалася. Нарешті, використання «грубих» шкал для частини характеристик (зокрема бінарних ознак) могло зменшувати чутливість моделі для близьких за властивостями альтернатив і ускладнювати диференціацію варіантів, коли потрібна точна аргументація вибору.

У [10] було запропоновано підхід до передміграційного планування для serverless-міграцій (Function as a Service, FaaS), де оптимізація ресурсів формалізувалася як задача прогнозування продуктивності та вартості з подальшим підбором конфігурацій. Для цього вводилася conditional stochastic Petri net (CSPN) і застосовувалося семплювання для отримання розподілів часу відгуку та вартості з фокусом на SLA для tail latency і high-percentile cost. Оптимізацію реалізовано як вибір пам'яті для кожної функції через задачі Best Cost under Performance Constraint (BCPC) та Best Performance under Budgeted Constraint (BPBC), а параметри підбиралися алгоритмом Depth-First Bottleneck Alleviation (DFBA) у циклі «профілювання → пошук вузького місця → алокація ресурсу → перепрофілювання», прив'язаному до моделі оплати AWS Lambda та послідовностей викликів із production-логів/trace-файлів або синтезованих даних.

Перевагами такого підходу було те, що він формалізував передміграційні рішення як задачі оптимізації з явними обмеженнями та враховував «хвостові» характеристики розподілів (tail latency, high-percentile cost), знижуючи ризик прийняття рішень лише за середніми метриками.

Однак, недоліками такого підходу було те, що його точність і доказовість критично залежали від репрезентативності логів/навантаження та достатності ітерацій профілювання, а на ранніх етапах міграції ці дані часто є неповними або нестабільними, що послаблювало надійність обґрунтування конфігурацій. Додатково оптимізаційна модель покривала «стратегію» переважно у вузькому сенсі

налаштування ресурсів FaaS (насамперед пам'яті) і не забезпечувала узгодженого вибору між альтернативними класами розгортання IaaS/PaaS/FaaS, коли потрібне зіставлення різних типів перетворень і супутніх ризиків. Також сильна прив'язка до актуальної моделі оплати AWS Lambda підвищувала чутливість до еволюції платформи: зміни в ціноутворенні або поведінці сервісу могли погіршувати переносимість і точність моделі, що вимагало регулярної адаптації методу для збереження коректності оцінок.

Таким чином, стан вирішення проблеми доцільно трактувати як частково зрілий на рівні окремих компонентів. По-перше, при переході від теорії до практики, розглянуті рішення є надто різноплановими та не зводяться до одного єдиного. По-друге, розглянуті підходи є в своїй суті фрагментарними, тобто вони подаються як окремі інструменти вирішення загальної проблеми і рідко інтегруються в 1 універсальний підхід. Виходячи з цього можна зробити висновок щодо необхідності подальших досліджень даної проблеми.

Мета статті. Метою статті є формалізація задачі оптимізації ресурсів на передміграційному етапі планування хмарної міграції ІС шляхом розроблення та обґрунтування KPI-орієнтованого підходу до планування, який забезпечуватиме прозоре для бізнесу зіставлення альтернатив і формалізований вибір стратегії хмарної міграції на основі вимірюваних показників ефективності з відтворюваністю результатів.

Для досягнення поставленої мети визначено такі задачі дослідження:

- розробити KPI-орієнтовану модель вибору стратегії хмарної міграції;
- провести експериментальне дослідження моделі.

Матеріали та методи дослідження. Об'єктом дослідження є стратегії хмарної міграції ІС. Предметом дослідження є процес вибору стратегії хмарної міграції ІС. Основна гіпотеза дослідження полягає у тому, що для відтворюваного вибору оптимальної стратегії хмарної міграції на передміграційному етапі недостатньо лише зафіксувати KPI «до/після» (як було наведено у [5], [6], [8]), а необхідно сформувати базову кількісну модель ІС, у якій характеристики навантаження, ресурсні профілі та архітектурні залежності системи переводяться у набір нормованих KPI й фінансових показників (витрати/ефект), а також доповнюються оцінками ризиків і обмежень. За такої побудови різні альтернативи міграції можуть бути зіставлені в єдиній процедурі як багатокритеріальний компроміс, що робить обґрунтування вибору прозорим і порівнюваним між кейсами.

У передміграційній фазі ключовою була потреба швидко й відтворювано порівняти альтернативи, маючи неповні відомості про майбутню архітектуру та

обмежений час на глибокі експерименти. Саме тому доцільним виявився вибір математико-алгоритмічної статичної моделі, що дозволило описати ІС «знімком» поточного стану, а наслідки стратегій задати як очікувані зміни цих KPI з подальшим формалізованим ранжуванням альтернатив. Такий підхід добре узгоджувався з логікою планування, де потрібні первинна оцінка придатності, узгодження з бізнес-цілями та попередня оцінка витрат і досяжності вимог, а не детальна часово-динамічна симуляція всіх процесів міграції. Отже, вибір математико-алгоритмічної статичної моделі в передміграційному контексті був виправданий тим, що вона забезпечувала швидке порівняння стратегій за бізнес-орієнтованими KPI. Це сприяло прозорості результату через внески метрик з відповідними вагами, і створювала основу для ранньої оцінки досяжності цільових рівнів та прийнятності витрат [1].

Одним з ключових припущень було те, що ІС можна описувати не через повний архітектурний рівень деталізації компонентів, а через набір вимірюваних KPI, які відображали результат її роботи з позиції бізнесу та експлуатації. Таке припущення було зумовлене практичними обмеженнями передміграційного аналізу: відтворення архітектури, залежностей і сценаріїв навантаження потребувало значних ресурсів, тоді як KPI зазвичай були доступні з моніторингу та звітності й могли використовуватися як стислий опис стану ІС. Вважалося, що обраний набір KPI є достатнім для порівняння стратегій, тобто наслідки міграції повинні проявлятися через зміну цих показників.

Окремо було прийнято, що різні за природою KPI можна зробити порівнюваними, якщо уніфікувати напрям інтерпретації «краще-гірше» та виконати нормування в межах узгоджених діапазонів. Такі діапазони трактувалися як практично обґрунтовані орієнтири, що задавалися історичними значеннями, бенчмарками або вимогами SLO/SLA. Це дозволило уникнути ситуації, коли метрики з більшими числовими масштабами «переважали» інші, та створити спільний простір для зіставлення альтернатив.

Для збереження зрозумілості та відтворюваності результатів було прийнято, що оцінки ефектів стратегій задавалися або на основі узагальнених спостережень з кейсів міграції, або як експертні оцінки з подальшим уточненням у міру накопичення даних. На початковому етапі невизначеність і варіативність середовища були враховані спрощено: аналіз виконувався в детермінованій постановці як порівняння очікуваних (середніх) ефектів, а ризики невиконання KPI відображалися через факт недосягнення порогів. Таке припущення було обрано як стартове, оскільки воно дозволяло отримати базовий консультативний інструмент і визначити критичні показники та пороги для прийняття рішення.

Розробка KPI-орієнтованої моделі вибору стратегії хмарної міграції ІС. Стан ІС до міграції

було подано у вигляді вектора-стовпця ключових показників ефективності, які відображали результати функціонування системи з позиції продуктивності, надійності, масштабованості та сукупних витрат володіння. Векторне подання було використано для одночасного врахування декількох критеріїв, що могли конфліктувати між собою, і для подальшого формалізованого порівняння альтернативних стратегій і відображалось як:

$$KPI^{base} = \begin{bmatrix} kpi_1^{base} \\ \dots \\ kpi_m^{base} \end{bmatrix},$$

де kpi_i^{base} – базове значення показника ефективності до міграції;

m – кількість показників ефективності ІС.

Мінімально прийнятний рівень якості після міграції було задано у вигляді порогового вектора-стовпця KPI^{target} .

$$KPI^{target} = \begin{bmatrix} kpi_1^{target} \\ \dots \\ kpi_m^{target} \end{bmatrix},$$

де KPI_m^{target} – цільове значення КРІ після міграції.

Різномірні КРІ було приведено до порівнюваного вигляду за рахунок нормування та приведення до вигляду «більше-краще». Нормовані значення метрик $nKPI_j^{base}$ та $nKPI_j^{target}$ типу «більше-краще» та «менше-краще» обчислювалися відповідно:

$$nKPI_j^{base} = \frac{kpi_j^{base} - kpi_j^{min}}{kpi_j^{max} - kpi_j^{min}}, \quad (1)$$

$$nKPI_j^{base} = \frac{kpi_j^{max} - kpi_j^{base}}{kpi_j^{max} - kpi_j^{min}}, \quad (2)$$

$$nKPI_j^{target} = \frac{kpi_j^{target} - kpi_j^{min}}{kpi_j^{max} - kpi_j^{min}}, \quad (3)$$

$$nKPI_j^{target} = \frac{kpi_j^{max} - kpi_j^{target}}{kpi_j^{max} - kpi_j^{min}}, \quad (4)$$

де kpi_j^{min} – мінімальне значення опорного діапазону; kpi_j^{max} – максимальне значення опорного діапазону j -ї метрики, $j = 1 \dots m$, m – кількість метрик; $nKPI_j^{base}, nKPI_j^{target} \in [0,1]$.

Умови, за яких планувалася міграція, було описано вектором-стовпцем контекстних факторів C . Він відображав властивості середовища та системи, що модулювали ефект стратегії, але не розглядалися як керовані змінні вибору:

$$C = \begin{bmatrix} c_1 \\ \dots \\ c_q \end{bmatrix},$$

де $context_q$ – q -й контекстний фактор;

q – загальна кількість контекстних факторів.

Множину альтернативних стратегій міграції S було задано дискретно:

$$S = \{s_1, s_2, \dots, s_n\},$$

де n – загальна кількість стратегій.

Базовий ефект BE_{s_i} стратегії $s_i \in S$, $i = 1 \dots n$ визначався як вектор-стовпець:

$$BE_{s_i} = \begin{bmatrix} b_1^{s_i} \\ \dots \\ b_j^{s_i} \end{bmatrix},$$

де $b_j^{s_i}$ – j -й базовий показник КРІ відповідної i -ї стратегії; $j = 1 \dots m$.

Контекстна поправка CE_{s_i} стратегії $s_i \in S$, $i = 1 \dots n$ визначалася як матриця розмірності $m \times q$:

$$CE_{s_i} = \begin{pmatrix} c_{11}^{s_i} & c_{12}^{s_i} & \dots & c_{1q}^{s_i} \\ c_{21}^{s_i} & c_{22}^{s_i} & \dots & c_{2q}^{s_i} \\ \dots & \dots & \dots & \dots \\ c_{m1}^{s_i} & c_{m2}^{s_i} & \dots & c_{mq}^{s_i} \end{pmatrix},$$

де $c_{mq}^{s_i}$ – коефіцієнт впливу q -го контекстного фактору на контекстну поправку m -го нормованого КРІ для i -ї стратегії, $c_{mq}^{s_i} \in [-1; 1]$.

Слід зазначити, що значення контекстної поправки в межах $[-1,1]$ трактувалися наступним чином: конкретна контекстна поправка «послаблює» чи «підсилює» конкретний очікуваний ефект стратегії. Нульові значення трактувалися як відсутність помітного ефекту.

Ефект кожної стратегії $\Delta nKPI_{s_i}$ було подано вектором-стовпцем, як очікуваний зсув нормованих КРІ:

$$\Delta nKPI_{s_i} = BE_{s_i} + CE_{s_i} \cdot C \quad (5)$$

У портфельній постановці або при розділенні системи на компоненти параметри BE і CE можуть задаватися окремо для кожної системи/компонента, тобто утворюється набір таких векторів і матриць.

Постміграційний профіль нормованих КРІ для кожної стратегії $s_i \in S$ було прогнозовано як суму базового стану та очікуваного зсуву:

$$nKPI_{s_i}^{post} = nKPI_j^{base} + \Delta nKPI_{s_i}, \quad (6)$$

з додатковим обмеженням для уникнення від'ємних значень:

$$nKPI_{s_i}^{post} = \min(1, \max(0, nKPI_{s_i}^{post})).$$

Бізнес-пріоритети було формалізовано через вектор-стовпець ваг W , який задавав відносну критичність кожної метрики:

$$W = \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix},$$

де w_n – n -та вага відповідної метрики; $n = 1 \dots m$; $w_n \geq 0$.

Для кількісної оцінки придатності стратегії $s_i \in S$ було використано штраф за недосягнення порогів – D_{s_i} як зважену суму дефіцитів, що виникали лише у разі, якщо прогнозоване значення було нижчим за цільове:

$$D_{s_i} = \sum_{j=1}^m w_j \cdot \max(0, nKPI_j^{target} - nKPI_{s_i,j}^{post}), \quad (7)$$

Витрати на реалізацію стратегії $s_i \in S$ було враховано через нормовану оцінку MC_{s_i} , яка відображала відносну складність, трудомісткість і тривалість виконання міграції. Окремо варто зазначити, що компроміс між мінімізацією витрат міграції та досягненням KPI було задано параметром β , $\beta \in [0,1]$. Даний параметр інтерпретується наступним чином: чим ближче значення параметру β до 0, тим більшу роль в фінальній оцінці грало досягнення цільових KPI. І навпаки, чим ближче параметр β до 1, тим більшу роль в фінальній оцінці грало мінімізація витрат/складності міграції. Таким чином, інтегральний бал T_{s_i} було визначено як:

$$T_{s_i} = \beta \cdot MC_{s_i} + (1 - \beta) \cdot D_{s_i}, \quad (8)$$

Таким чином стратегію $s_i \in S$ було визначено як альтернативу з мінімальним інтегральним балом у заданому контексті. Таким чином забезпечувалося одночасне врахування якості сервісу за KPI та обмежень на витрати міграції, а також можливість пояснення вибору через внески окремих метрик у дефіцит.

$$s = \arg \min T_{s_i}, \quad (9)$$

за умови виконання обмежень:

$$\begin{cases} nKPI^{base} \in [0,1] \\ nKPI^{target} \in [0,1] \\ c_{mq}^{s_i} \in [-1; 1] \\ w_n \geq 0 \\ \beta \in [0,1] \end{cases}$$

Апробація результатів дослідження.

Необхідність вибору стратегії хмарної міграції було сформульовано як задачу порівняння трьох альтернативних підходів LS (Lift-and-Shift), PF (Re-platforming) та EN (Re-engineering) за очікуваним профілем ключових показників ефективності. Для системи було задано базові виміряні та мінімально прийнятні цільові значення KPI після міграції [11]. Опис вхідних KPI та опорних меж наведено в табл. 1. Опис цільових порогів за ключовими показниками ефективності системи було наведено у табл. 2.

Таблиця 1 – Вхідні KPI, опорні межі нормування

Метрика	Напрямок	kpi^{min}	kpi^{max}	kpi^{base}
Response time, мс	менше-краще	120	400	220
Throughput, транз/с	більше-краще	300	1200	600
Resource utilization rate, %	менше-краще	40	95	75
Total cost of ownership, USD/pik	менше-краще	30000	650000	48000
Availability, %	більше-краще	99.0	99.99	99.5
Time to market, дні	менше-краще	3	21	14
Scalability, коеф.	більше-краще	1.0	3.0	1.6

Таблиця 2 – Цільові пороги метрик

Метрика	kpi^{target}
Response time	170
Throughput	800
Resource utilization rate	70
Total cost of ownership	420000
Availability	99.7
Time to market	7
Scalability	2.2

Після цього було задано умови застосування стратегій. У таблиці 3 наведено контекстні фактори нормованій шкалі [0;1].

Таблиця 3 – Контекстні фактори

Фактор	Позначення	Значення
Критичність системи	context_crit	0.8
Варіативність навантаження (міра нестабільності робочого профілю)	context_var	0.6
Складність інтеграцій	context_int	0.7

У таблиці 4 наведено розрахунок $nKPI^{base}$ і $nKPI^{target}$ шляхом нормування значень з табл. 1 за (1)–(4), причому обидва типи метрик були приведені до інтерпретації «більше – краще».

Таблиця 4 – Нормовані значення KPI для базового та цільового станів

Метрика	$nKPI^{base}$	$nKPI^{target}$
Response time	0.642857	0.821429
Throughput	0.333333	0.555556
Resource utilization rate	0.363636	0.454545
Total cost of ownership	0.485714	0.657143
Availability	0.505051	0.707071
Time to market	0.388889	0.777778
Scalability	0.300000	0.600000

Було задано матриці контекстних поправок з коефіцієнтами $c_{crit}, c_{var}, c_{int}$ для кожної стратегії. Додатково було задано параметри MC_{s_i} і β . В межах прикладу матриці CE_{s_i} було спрощено до вигляду $(c_{crit} \ c_{var} \ c_{int})$, тоді величина $CE_{s_i} \cdot C$ була однаковою для всіх KPI в межах стратегії. Результат наведено в таблиці 5.

Таблиця 5 – Коефіцієнти контекстної поправки та міграційна вартість

s_i	c_{crit}	c_{var}	c_{int}	$CE_{s_i} \cdot C$	MC_{s_i}	β
LS	-1	0.8	0.3	-0.11	0.2	0.3
PF	0.9	-0.8	-0.3	0.03	0.5	0.3
EN	0.9	-0.7	-0.2	0.16	0.9	0.3

Далі було розраховано BE_{s_i} для кожного KPI, тобто внесок стратегії без урахування контекстної поправки (табл. 6).

Таблиця 6 – Базовий ефект для нормованих КРІ

Метрика	BE_{LS}	BE_{PF}	BE_{EN}
Response time	0.052314	0.193086	0.282429
Throughput	0.072156	0.312133	0.461000
Resource utilization rate	-0.037945	0.160618	0.324636
Total cost of ownership	0.102314	0.235943	0.303857
Availability	0.067105	0.231325	0.365040
Time to market	0.127711	0.312133	0.461000
Scalability	0.116600	0.328800	0.511000

Слід звернути увагу на від'ємне значення показника Resource utilization rate для BE_{LS} . Це цілком допустиме значення, що інтерпретувалося наступним чином: в середньому/типовому сценарії без контекстної поправки стратегія LS прогнозувалася як така, що погіршує цей КРІ на нормованій шкалі.

Далі, використовуючи (5) та (6) було отримано повний прогнозований ефект та постміграційний профіль для кожної зі стратегій відповідно. Результати розрахунків за формулами було наведено у таблиці 7.

Таблиця 7 – Зсуви $\Delta nKPI$ та прогнозовані $nKPI^{post}$ для стратегій

Метрика	$\Delta nKPI_{LS}$	$nKPI_{LS}^{post}$	$\Delta nKPI_{PF}$	$nKPI_{PF}^{post}$	$\Delta nKPI_{EN}$	$nKPI_{EN}^{post}$
Response time	-0.057686	0.585171	0.223086	0.865943	0.442429	1.000000
Throughput	-0.037844	0.295489	0.342133	0.675466	0.621000	0.954333
Resource utilization rate	-0.147945	0.215691	0.190618	0.554254	0.484636	0.848272
Total cost of ownership	-0.007686	0.478028	0.265943	0.751657	0.463857	0.949571
Availability	-0.042895	0.462156	0.261325	0.766376	0.525040	1.000000
Time to market	0.017711	0.406600	0.342133	0.731022	0.621000	1.000000
Scalability	0.006600	0.306600	0.358800	0.658800	0.671000	0.971000

Від'ємні значення $\Delta nKPI_{LS}$ означали, що для стратегії LS прогнозований постміграційний рівень нормованого КРІ зменшувався відносно базового стану, тобто це інтерпретувалося як погіршення показника.

Після цього було розраховано зважені внески та проміжні дефіцити для кожної зі стратегій. Тут за d_{s_i} позначено $\max(0, nKPI_j^{target} - nKPI_{s_i,j}^{post})$ з (7), тобто проміжні розрахунки. Результат наведено у таблиці 8.

Таблиця 8 – Зважені внески та дефіцити

Метрика	w_j	d_{LS}	$w \cdot d_{LS}$	d_{PF}	$w \cdot d_{PF}$	d_{EN}	$w \cdot d_{EN}$
Response time	0.20	0.236258	0.047252	0.000000	0.000000	0.000000	0.000000
Throughput	0.15	0.260067	0.039010	0.000000	0.000000	0.000000	0.000000
Resource utilization rate	0.10	0.238854	0.023885	0.000000	0.000000	0.000000	0.000000
Total cost of ownership	0.20	0.179115	0.035823	0.000000	0.000000	0.000000	0.000000
Availability	0.25	0.244915	0.061229	0.000000	0.000000	0.000000	0.000000
Time to market	0.05	0.371178	0.018559	0.046756	0.002338	0.000000	0.000000
Scalability	0.05	0.293400	0.014670	0.000000	0.000000	0.000000	0.000000

З Для кожної за стратегій було обчислено показник дефіциту за (7). Результати наведено у таблиці 9.

Таблиця 9 – Підсумкові значення дефіцитів за стратегіями

Стратегія s_i	D_{s_i}
LS	0.240428
PF	0.002338
EN	0.000000

Нарешті, для кожної зі стратегій було обчислено інтегральний бал за (8). Результати розрахунків наведено в таблиці 10.

Таблиця 10 – Інтегральна оцінка T_{s_i}

s_i	MC_{s_i}	$\beta \cdot MC_{s_i}$	$(1 - \beta) \cdot D_{s_i}$	T_{s_i}
LS	0.2	0.060000	0.168299	0.228299
PF	0.5	0.150000	0.001636	0.151636
EN	0.9	0.270000	0.000000	0.270000

Згідно (9), цільова стратегія вибиралася як та, що має мінімальний інтегральний показник T_{s_i} . В даному випадку, $s = \arg \min T_{s_i} = PF$.

Таким чином, апробація показала, що модель забезпечувала відтворюваний вибір стратегії через формалізоване зіставлення альтернатив у просторі нормованих КРІ та явне врахування контексту й вартості міграції. За заданих порогових вимог, ваг пріоритетів та параметра компромісу $\beta = 3$ мінімальне

значення T_{s_i} було отримано для стратегії PF, що відповідало найкращому балансу між досягненням цільового профілю KPI та витратами на реалізацію. Стратегія LS була оцінена як менш придатна через накопичення дефіцитів за критичними KPI, тоді як EN забезпечувала нульовий дефіцит, але поступалася через вищу умовну вартість міграції в обраній постановці. Отриманий результат продемонстрував, що модель придатно відокремлювала ефект стратегії від управлінського компромісу між якістю та витратами і могла використовуватися як консультативний інструмент для попереднього ранжування стратегій.

Обговорення результатів дослідження. За результатами дослідження було розроблено KPI-орієнтовану модель передміграційного вибору стратегії, яка прогнозувала постміграційний профіль нормованих показників ефективності на основі базового стану ІС, очікуваного ефекту стратегії та контекстної поправки, а також формувала інтегральну оцінку альтернатив через поєднання штрафу за недосягнення цільових KPI, нормованої міграційної вартості з керованим параметром компромісу β та системи обмежень.

Апробацію розробленої моделі було виконано на прикладі, де задача вибору стратегії хмарної міграції формалізувалася як порівняння трьох альтернативних підходів LS, PF та EN за очікуваним профілем ключових показників ефективності. Вхідні дані містили базові значення KPI, опорні межі нормування та цільові пороги після міграції для заданого набору метрик. Умови, за яких планувалася міграція, було подано через контекстні фактори, що модулювали ефект стратегії, а порівняння альтернатив виконувалося у нормованому просторі KPI так, щоб «більше» завжди означало «краще» незалежно від первинного напрямку метрики.

Отримані результати показали, що різні стратегії формували принципово відмінні постміграційні KPI-профілі навіть за однакових цільових порогів. Для LS приріст нормованих метрик виявився недостатнім, щоб досягти заданих порогів за критичними KPI: постміграційне значення для response time становило 0.585171 при цільовому 0.821429, для throughput – 0.295489 при цільовому 0.555556, для availability – 0.462156 при цільовому 0.707071, а для scalability – 0.306600 при цільовому 0.600000. Негативні зсуви $\Delta nKPI$ для LS у таблиці 6 означали, що після врахування контекстної поправки сумарний ефект стратегії зменшував нормовану «якість» відповідних KPI відносно базового стану. У практичній інтерпретації це могло відповідати ситуації, коли перенесення «as-is» не дає достатнього вигаду, а в умовах підвищеної критичності, варіативності навантаження та складних інтеграцій частина метрик схильна до потенційної деградації через накладні витрати або обмежену оптимізацію конфігурацій та ресурсів. Водночас PF формувала профіль, який практично повністю відповідав порогам: response time

і досягав 0.865943, throughput – 0.675466, availability – 0.766376, а єдиний помітний «ззор» зберігався для time-to-market (0.731022 при пороговому 0.777778). Така картина могла бути типовим практичним очікуванням від PF, коли зміни обмежуються рівнем платформи та інфраструктурних механізмів і дають суттєве покращення продуктивності й надійності без повної перебудови прикладної логіки. В свою чергу, організаційно-зумовлені метрики, на кшталт time to market, значних покращень не отримали, оскільки в меншій мірі залежали від хмарної інфраструктури, як такої. Нарешті, EN забезпечувала найвищі значення нормованих KPI (зокрема 1.000000 для response time та 1.000000 для availability), що узгоджувалося з очікуваною перевагою глибшої переробки ІС з позиції досягнення цільового стану. Поява значень 1.000000 означала, що прогнозований KPI досягав найкращого можливого рівня в рамках прийнятої шкали, що могло відповідати сценарію суттєвих архітектурних покращень.

Ці відмінності стали кількісно явними на рівні дефіцитів відносно порогів та їх зважених внесків. Значення дефіцитів у таблиці 7 фактично відображали «відстань» від прогнозованого стану до порогу в нормованій шкалі. Для LS дефіцити накопичувалися одразу за кількома метриками з високою значущістю, включно з availability (дефіцит 0.244915 за ваги 0.25) та response time (дефіцит 0.236258 за ваги 0.20), що формувало вагомі внески у штраф. Додатково суттєвими були дефіцити за throughput (0.260067 за ваги 0.15), resource utilization (0.238854 за ваги 0.10) і time-to-market (0.371178 за ваги 0.05). У підсумку це дало $D_{LS} = 0.240428$, тобто значний сукупний недобір вимог у нормованому просторі KPI. Для PF сумарний дефіцит практично зник і становив 0.002338, причому він був зумовлений виключно метрикою time-to-market (дефіцит 0.046756 за ваги 0.05), тоді як EN забезпечив нульовий дефіцит, тобто повне досягнення порогів за всіма метриками. У реалістичній інтерпретації такий розподіл внесків міг модулювати ситуацію, коли стратегія або «провалює» кілька критичних метрик одночасно (тоді штраф швидко зростає), або має один локальний недолік в менш пріоритетній метриці (тоді сумарний дефіцит майже нульовий).

Ключовим результатом апробації стало те, що рекомендація моделі визначилася не лише «якістю» за KPI, а керованим компромісом між досягненням цільового профілю та витратами на реалізацію стратегії через параметр $\beta = 0.3$. За умовної нормованої вартості $MC_{LS} = 0.2, 0.5$ для PF і 0.9 для EN інтегральний бал T набув значень 0.228299, 0.151636 та 0.270000 відповідно, тобто мінімум був отриманий для PF. Отже, модель відтворювано обрала PF як стратегію з найкращим балансом між мінімізацією дефіцитів KPI та обмеженнями на ресурси міграції, тоді як LS прогнала через накопичені дефіцити, а EN – через завищену «вартісну» складову за заданого β .

Порівняно з дослідженнями [1], [3], [6], [9], запропонована модель мала перевагу в тому, що зводила передміграційне рішення до прозорої й відтворюваної процедури зі зрозумілою логікою пояснення результату через внесок окремих КРІ у дефіцит і через явний параметр компромісу β . На відміну від підходів, де вибір стратегії часто лишається наслідком інструментальної оцінки або провайдерно-специфічного планування ([4], [6], [10]), тут стратегія обиралася як мінімум інтегрального критерію в нормованому просторі метрик, що підвищувало переносимість моделі між організаціями та середовищами за умови коректного задання опорних меж нормування і порогів. Процедура також не вимагала дорогих серій експериментів або повного економічного моделювання на ранньому етапі (у порівнянні з [2]). Для первинного ранжування були достатні базові КРІ, цільові порогові ваги пріоритетів і узагальнена оцінка міграційної вартості, що відповідало задачі зменшення ресурсних витрат планування при збереженні формалізації.

Разом із тим апробація виявила й недоліки, що можуть бути пов'язані з математичними припущеннями та характером даних. По-перше, параметри ефектів стратегій, контекстної чутливості та вартості міграції у такій постановці задавалися умовно, тому зовнішня валідність висновків була обмеженою. По-друге, контекстна поправка для спрощення була прийнята однаковою для всіх КРІ всередині стратегії, що знижувало реалістичність, адже на практиці контекст по-різному впливає на доступність, ТСО та time-to-market.

У підсумку апробація підтвердила працездатність моделі як консультативного інструмента для раннього, пояснюваного ранжування стратегій у заданому контексті та продемонструвала, що вибір стратегії може бути обґрунтовано через поєднання дефіцитів КРІ відносно порогів і обмежень на ресурси міграції. Одночасно було окреслено необхідність подальшої калібрації параметрів на емпіричних даних і розширення моделі, щоб зменшити залежність від умовних оцінок і підвищити прикладну достовірність рекомендацій.

Висновки.

1. Було розроблено модель вибору стратегії хмарної міграції на основі профілю КРІ, де різні метрики було приведено до нормованого простору так, щоб після перетворення більше значення відповідало кращому стану незалежно від первинного напрямку метрики. Ефект кожної стратегії було формалізовано як очікуваний зсув нормованих КРІ, що складався з базового ефекту та контекстної поправки, а відповідність цільовим вимогам оцінювалася через дефіцит метрик відносно порогів із зважуванням за вектором бізнес-пріоритетів W . Витратну складову було введено через нормовану оцінку MC , а компроміс між мінімізацією витрат і досягненням КРІ було задано параметром β . Інтегральний критерій T визначено як комбінацію MC

і D , після чого рекомендовану стратегію задавали як альтернативу з мінімальним T у заданому контексті.

2. Було виконано апробацію моделі на прикладі порівняння стратегій LS, PF та EN за заданими базовими КРІ, опорними межами нормування й пороговими цільовими значеннями, а також контекстними факторами, що модулювали ефект стратегії у просторі нормованих КРІ. У результаті було отримано постміграційні нормовані КРІ-профілі для кожної стратегії та розраховано дефіцити відносно порогів і підсумкові значення D , які становили 0.240428 для LS, 0.002338 для PF та 0.000000 для EN. За $\beta=0.30$ інтегральна оцінка T склала 0.228299 (LS), 0.151636 (PF) та 0.270000 (EN), тому модель відтворювано визначила PF (Re-platforming) як найкращий компроміс між досягненням цільового профілю КРІ та витратами на реалізацію, тоді як LS поступалася через накопичення дефіцитів, а EN – через вищу умовну вартість міграції.

Список літератури

1. Ramchand K., Baruwal Chhetri M., Kowalczyk R. Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment. *Journal of Cloud Computing*. 2021. Т. 10, № 1. <https://doi.org/10.1186/s13677-020-00210-w> (дата звернення: 21.01.2026).
2. Henning S., Hasselbring W. A configurable method for benchmarking scalability of cloud-native applications. *Empirical Software Engineering*. 2022. Т. 27, № 6. <https://doi.org/10.1007/s10664-022-10162-1> (дата звернення: 21.01.2026).
3. Hosseini Shirvani M., Amin G. R., Babaeikiadehi S. A decision framework for cloud migration: A hybrid approach. *IET Software*. 2022. <https://doi.org/10.1049/sfw2.12072> (дата звернення: 21.01.2026).
4. An efficient model to estimate and optimise the cloud migration costs from on-premises web apps / V. Prakash та ін. *Discov Computing*. 2025. Т. 28, № 151. <https://doi.org/10.1007/s10791-025-09666-3> (дата звернення: 21.01.2026).
5. Michael Ayas H., Leitner P., Hebig R. An empirical study of the systemic and technical migration towards microservices. *Empirical Software Engineering*. 2023. Т. 28, № 4. <https://doi.org/10.1007/s10664-023-10308-9> (дата звернення: 21.01.2026).
6. Harikrishna M., Srinivasa R. T., Gopikrishna Y. Optimizing cloud migration: designing robust architectures for seamless transition from on-premises to azure for sap and database systems. *International Journal of Engineering Technology Research & Management (IJETRM)*. 2025. Т. 09, № 01. <https://doi.org/10.5281/zenodo.14782256> (дата звернення: 21.01.2026).
7. Migration-Based Load Balance of Virtual Machine Servers in Cloud Computing by Load Prediction Using Genetic-Based Methods / L.-H. Hung та ін. *IEEE Access*. 2021. Т. 9. С. 49760–49773. <https://doi.org/10.1109/access.2021.3065170> (дата звернення: 21.01.2026).
8. Continuous integration of architectural performance models with parametric dependencies – the CIPM approach / M. Mazkatli та ін. *Automated Software Engineering*. 2025. Т. 32, № 2. <https://doi.org/10.1007/s10515-025-00521-9> (дата звернення: 21.01.2026).
9. Şener U., Gökalp E., Eren P. E. CLOUD-QM: a quality model for benchmarking cloud-based enterprise information systems. *Software Quality Journal*. 2024. <https://doi.org/10.1007/s11219-024-09669-1> (дата звернення: 21.01.2026).
10. Fine-Grained Performance and Cost Modeling and Optimization for FaaS Applications / C. Lin та ін. *IEEE Transactions on Parallel and Distributed Systems*. 2022. С. 1–

15. <https://doi.org/10.1109/tpds.2022.3214783> (дата звернення: 21.01.2026).
11. Свланов М. В., Шутько В. В. Ключові показники ефективності інформаційної системи для хмарної міграції. *Інформаційно-комунікаційні технології та кібербезпека (ІКТК-2025): Матеріали міжнар. наук. конф.*, м. Харків, 4 груд. 2025 р. 2025. https://ice.nure.ua/wp-content/uploads/2025/12/91-Ievlanov_Shutko_Sektsiia-2.pdf (дата звернення: 22.01.2026).
- References (transliterated)**
1. Ramchand K., Baruwah Chhetri M., Kowalczyk R. Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment. *Journal of Cloud Computing*. 2021. Т. 10, № 1. <https://doi.org/10.1186/s13677-020-00210-w> (accessed: 21.01.2026).
2. Henning S., Hasselbring W. A configurable method for benchmarking scalability of cloud-native applications. *Empirical Software Engineering*. 2022. Т. 27, № 6. <https://doi.org/10.1007/s10664-022-10162-1> (accessed: 21.01.2026).
3. Hosseini Shirvani M., Amin G. R., Babaekiahehi S. A decision framework for cloud migration: A hybrid approach. *IET Software*. 2022. <https://doi.org/10.1049/sfw2.12072> (accessed: 21.01.2026).
4. An efficient model to estimate and optimise the cloud migration costs from on-premises web apps / V. Prakash et al. *Discov Computing*. 2025. Т. 28, № 151. <https://doi.org/10.1007/s10791-025-09666-3> (accessed: 21.01.2026).
5. Michael Ayas H., Leitner P., Hebig R. An empirical study of the systemic and technical migration towards microservices. *Empirical Software Engineering*. 2023. Т. 28, № 4. <https://doi.org/10.1007/s10664-023-10308-9> (accessed: 21.01.2026).
6. Harikrishna M., Srinivasa R. T., Gopikrishna Y. Optimizing cloud migration: designing robust architectures for seamless transition from on-premises to azure for sap and database systems. *International Journal of Engineering Technology Research & Management (IJETRM)*. 2025. Т. 09, № 01. <https://doi.org/10.5281/zenodo.14782256> (accessed: 21.01.2026).
7. Migration-Based Load Balance of Virtual Machine Servers in Cloud Computing by Load Prediction Using Genetic-Based Methods / L.-H. Hung та ін. *IEEE Access*. 2021. Т. 9. P. 49760–49773. <https://doi.org/10.1109/access.2021.3065170> (accessed: 21.01.2026).
8. Continuous integration of architectural performance models with parametric dependencies – the CIPM approach / M. Mazkatli та ін. *Automated Software Engineering*. 2025. Т. 32, № 2. <https://doi.org/10.1007/s10515-025-00521-9> (accessed: 21.01.2026).
9. Şener U., Gökalp E., Eren P. E. CLOUD-QM: a quality model for benchmarking cloud-based enterprise information systems. *Software Quality Journal*. 2024. <https://doi.org/10.1007/s11219-024-09669-1> (accessed: 21.01.2026).
10. Fine-Grained Performance and Cost Modeling and Optimization for FaaS Applications / C. Lin та ін. *IEEE Transactions on Parallel and Distributed Systems*. 2022. P. 1–15. <https://doi.org/10.1109/tpds.2022.3214783> (accessed: 21.01.2026).
11. Yevlanov M. V., Shut'ko V. V. Klyuchovi pokaznyky efektyvnosti informatsiynoyi systemy dlya khmarnoyi mihratsiyi [Key performance indicators of an information system for cloud migration]. *Informatsiyno-komunikatsiyini tekhnolohiyi ta kiberbezpeka (IKTK-2025): materialy mizhnar. nauk. konf. [Information and Communication Technologies and Cybersecurity (ICTC-2025): Proc. of the Int. Sci. Conf.]*, Kharkiv, 4 December 2025. 2025. Available at: https://ice.nure.ua/wp-content/uploads/2025/12/91-Ievlanov_Shutko_Sektsiia-2.pdf (accessed: 22.01.2026).

Надійшла (received) 24.01. 2026

Відомості про авторів / About the Authors

Євланов Максим Вікторович (Ievlanov Maksym) – доктор технічних наук, професор, професор кафедри інформаційних управляючих систем, Харківський національний університет радіоелектроніки, м. Харків, Україна; email: maksym.ievlanov@nure.ua; ORCID: <https://orcid.org/0000-0002-6703-5166>.

Шутько Віктор Валерійович (Shutko Viktor) – аспірант кафедри інформаційних управляючих систем, Харківський національний університет радіоелектроніки, м. Харків, Україна; email: viktor.shutko@nure.ua, ORCID: <https://orcid.org/0009-0001-0527-4401>.