УДК 004.75; 004.7 DOI: 10.18372/2073-4751.84.20905

Чижов Олександр,
orcid.org/0000-0002-3992-8522,
oleksandr.chyzhov@npp.kai.edu.ua
Фесенко Андрій,
orcid.org/ 0000-0001-5154-5324,
andrii.fesenko@npp.kai.edu.ua

МЕТОД ПЛАНУВАННЯ ПОТУЖНОСТЕЙ ДЛЯ КОНСОЛІДАЦІЇ РЕСУРСІВ У МАСОВОМУ ХМАРНОМУ ВЕБ-ХОСТИНГУ

Державний університет «Київський авіаційний університет»

Вступ

Сучасна індустрія веб-технологій налічує понад 1,2 млрд активних вебсайтів, переважна більшість яких належить до сегмента низько навантажених ресурсів [1]. Традиційна модель надання послуг хостингу (Shared Hosting) базується на статичному розподілі ресурсів, де за кожним клієнтом закріплюється фіксований ліміт обчислювальних потужностей. Попри простоту реалізації, такий підхід демонструє неефективність у використанні оперативної пам'яті (RAM) — одного з найдорожчих компонентів серверної інфраструктури. Проблема полягає у стохастичній природі трафіку. Середня інтенсивність навантаження на низько навантажений ресурс становить лише близько 0,114 зап/с, проте розподіл цих запитів у часі є вкрай нерівномірним [3]. При статичному підході провайдер змушений застосовувати стратегію надлишкового резервування, виходячи з потенційних піків навантаження окремих вузлів. Це призводить до виникнення значних обсягів зарезервованого ресурсу, який у масштабах сучасних дата-центрів обчислюється терабайтами [3].

Хоча для високонавантажених систем динамічне управління ємністю на базі контейнеризації вже стало галузевим стандартом, у сфері масового Shared-хостингу досі домінує статична архітектура. Основною перешкодою є складність математичного гарантування якості обслуговування при агрегації тисяч незалежних, слабких та нестабільних потоків трафіку в єдиний пул [3]. Провайдеру необхідні верифіковані методи, які б виключали ризик деградації сервісу через ефект накладання стохастичних піків. У роботі запропоновано метод планування ємності для консолідації ресурсів, що базується на математичному моделюванні агрегованих стохастичних навантажень.

Об'єктом дослідження є процес розподілу оперативної пам'яті у кластерних системах веб-хостингу при обслуговуванні масових низько навантажених ресурсів.

Метою роботи є розроблення методу планування ємності для консолідації ресурсів у масовому хмарному веб-хостингу, що використовує математичну модель агрегування навантажень для прийняття рішень щодо виділення ресурсів та забезпечує мінімізацію витрат оперативної пам'яті за рахунок консолідації трафіку в єдиний обчислювальний пул при суворому дотриманні цільового рівня надійності (SLA 99,9%).

Аналіз останніх досліджень та постановка задачі

Питання оптимізації ресурсів у хмарних середовищах є предметом

активних дискусій. Більшість сучасних досліджень фокусуються на алгоритмах прогнозування навантаження для окремих великих додатків (наприклад, Horizontal Pod Autoscaler у Kubernetes) [5] або на загальній оптимізації енергоспоживання великих дата-центрів [4]. У той же час, стратегічний аналіз ринку показує, що провайдери масового хостингу все ще шукають баланс між економічною ефективністю та гарантіями безпеки для мільйонів малих клієнтів.

Фундаментальні праці у сфері теорії масового обслуговування (ТМО) [6,7] заклали математичну основу для розрахунку ємності мереж, проте їх класичне застосування в хостингу тривалий час обмежувалося статичним плануванням. Сучасні автори розглядають архітектури Serverless та Function-as-a-Service (FaaS) як відповідь на потребу у гнучкості, але ці рішення часто виявляються занадто дорогими або технічно складними для перенесення класичних Legacy-сайтів (WordPress, Python/PHP додатки) [2].

Попередні дослідження автора показали, що використання формули Ерланга-Б дозволяє математично обґрунтувати необхідну кількість ресурсів для запобігання відмовам в обслуговуванні навіть в умовах аномальних сплесків трафіку [3]. Проте залишається недостатньо висвітленим питання кількісної оцінки інтегральної економії обчислювальних ресурсів при переході від інфраструктури статичних ізольованих серверів до динамічного хмарного пулу саме для сегмента низько навантажених веб-ресурсів [2,3]. Існуючі моделі автоскейлінгу здебільшого орієнтовані на високонавантажені системи [5,8] і не пропонують математичного інструментарію для розрахунку економічного ефекту від консолідації масових джерел трафіку з урахуванням накладних витрат на балансування та суворих вимог SLA.

До завдань цієї наукової роботи належить: розробка імовірнісної моделі агрегованого навантаження для множини низько навантажених джерел; формулювання критеріїв вибору кількості вузлів кластера на основі формули Ерланга для заданого SLA; проведення порівняльного аналіза інтегрального споживання пам'яті (RAM) між традиційною та хмарно-кластерною моделями.

Формалізація математичного апарату

Формалізація вхідного навантаження. Для математичного опису процесів у хмарному кластері розглянемо сукупність незалежних веб-ресурсів $W=\{w_1, w_2, \dots, w_n\}$, що обслуговуються системою. В межах обраної архітектури ми відходимо від детермінованого закріплення обчислювальних потужностей за конкретним сайтом чи невеликою групою сайтів. Натомість пропонується розглядати систему як єдиний обчислювальний пул, на вхід якого надходить агрегований стохастичний потік запитів:

$$\Lambda(t)=\sum_{i=1}^{n}\lambda_i(t)$$

де:

- $\lambda_i(t)$ - інтенсивність надходження запитів до i-го ресурсу в момент часу t.

Враховуючи специфіку низько навантажених сайтів, де інтервали між запитами є випадковими та незалежними, для моделювання вхідного потоку Λ(t) доцільно використовувати розподіл Пуассона. Це дозволяє застосувати апарат теорії масового обслуговування для оцінки поведінки всієї системи як цілісного об'єкта, що оперує сумарною інтенсивністю запитів, згладжуючи

локальні піки окремих джерел навантаження [6].

Параметри обчислювального вузла та стан системи. Для переходу від абстрактної інтенсивності запитів до фізичних параметрів обчислювальної системи необхідно визначити базову одиницю споживання ресурсів. Аналіз роботи сучасних інтерпретаторів скриптових мов (PHP-FPM, Python WSGI) показує, що критичним обмеженням при обслуговуванні веб запитів є обсяг оперативної пам'яті (RAM), що виділяється під кожну активну сесію [4]. Для переходу до фізичних параметрів інфраструктури введемо характеристики типового обчислювального вузла (сервера). Ємність одного вузла C_{node} визначається як кількість одночасних запитів, які він може обробити в межах доступної оперативної пам'яті:

$$C_{node} = \lfloor \frac{RAM_{total} - RAM_{os}}{m_{unit}} \rfloor$$

де:

- RAM_{total} - повний обсяг пам'яті сервера;
- RAM_{os} - резерв під операційну систему та системні процеси;
- m_{unit} - базовий квант ресурсу, що виділяється для утримання стану однієї активної сесії.

Для коректного застосування апарату теорії масового обслуговування, стан системи в момент часу t також характеризується середньою тривалістю утримання ресурсу одним запитом - квантом часу $t_{session}$. Таким чином, сукупне навантаження на систему в момент t, виражене в Ерлангах (A), визначається як добуток інтенсивності вхідного потоку на тривалість обслуговування:

$$A(t) = \Lambda(t) \cdot t_{session}$$

Поточний стан завантаження кластера визначається кількістю активних сесій $N(t)$, що генеруються агрегованим вхідним потоком. Необхідна кількість серверів $K_{req}(t)$ у момент часу t повинна бути такою, щоб сумарна ємність слотів покривала попит із заданим рівнем імовірності P_{block}:

$$M_{cluster}(t) = K_{req}(t) \cdot RAM_{total}$$

Тут $M_{cluster}(t)$ - це саме зарезервований (оплачений) обсяг оперативної пам'яті, яка завжди кратна обсягу пам'яті одного сервера.

Ймовірнісна модель якості обслуговування. Для забезпечення об'єктивності порівняльного аналізу вводиться припущення про ідентичність середнього навантаження для всіх одиниць множини W, що дозволяє абстрагуватися від індивідуальних особливостей контенту та зосередитися на стохастичних характеристиках сумарного потоку. Дане припущення дозволяє оцінити граничну ефективність моделі, не втрачаючи загальності висновків для масового сегмента.

У кластерній моделі балансувальник динамічно змінює $K(t)$, адаптуючись до поточного значення інтенсивності сумарного потоку $\Lambda(t)$. Для визначення $K(t)$ використовується обернена задача Ерланга - знайти мінімальне ціле K, при якому виконується умова:

$$P_{block}(\Lambda(t), K \cdot C_{node}) \le 1 - SLA$$

У традиційній моделі Shared Hosting кількість серверів S розраховується як сума вузлів, кожен з яких має однакову кількість веб-ресурсів і повинен самостійно забезпечувати цільовий рівень $P_{block} \le 1 - SLA$ для своєї групи сайтів при піковому навантаженні.

Показник теоретичної ефективності консолідації. Для порівняння двох архітектур оцінюється інтегральний обсяг зарезервованої

пам'яті за період T. У традиційній моделі загальний обсяг ресурсів є незмінним:

$$M_{shared}^{total} = S \cdot RAM_{total} \cdot T$$

де:

- S - кількість серверів, яка необхідна провайдеру для розміщення всієї множини сайтів W за традиційною схемою.
- RAM_{total} - повний обсяг пам'яті на одному сервері

У хмарно-кластерній моделі загальний обсяг ресурсів розраховується як інтегральна площа під сходинковою функцією зміни кількості вузлів, включно з витратами на балансування M_{LB}:

$$M_{cluster}^{total} = \int_0^T \left(K_{req}(t) \cdot RAM_{total} + M_{LB} \right) dt$$

де:

- $K_{req}(t)$ - мінімальна необхідна кількість серверів для обслуговування вхідного потоку у момент часу t;
- M_{LB} - постійна кількість пам'яті необхідної для обслуговування інтелектуального балансувальника.

Показник M_{LB} враховує не лише роботу керуючого програмного забезпечення, а й витрати на утримання в пам'яті таблиць маршрутизації та TLS-сертифікатів для всієї множини n об'єктів [9]. Теоретична максимальна ефективність економії ресурсів ξ_{max} розраховується як:

$$\xi_{max} = \left(1 - \frac{M_{cluster}^{total}}{M_{shared}^{total}} \right) \cdot 100\,\%$$

Цей показник демонструє відсоток економії за рахунок того, що в періоди низької активності $K_{req}(t) \ll S$, і компанія не платить за незадіяні фізичні сервери завдяки моделі "Pay-as-you-go" [10].

Запропонована математична модель використовується як основа методу планування ємності, який включає етапи збору статистики навантаження, агрегування потоків, визначення необхідної ємності кластера та прийняття рішень щодо консолідації ресурсів.

Методологія та параметри моделювання

Параметри експерименту та обґрунтування профілю навантаження. Для забезпечення точності розрахунків у моделі використовуються бінарні одиниці вимірювання обсягу пам'яті (GiB, MiB). Для проведення імітаційного моделювання було сформовано віртуальне середовище, що відтворює роботу типової інфраструктури хостинг-провайдера. Основними параметрами експерименту є:

- Загальна кількість веб-ресурсів: $n = 10000$ сайтів.
- Середня інтенсивність навантаження на один ресурс: $\lambda_{avg} \approx 0{,}114$ зап/с, що відповідає приблизно 50 000 відвідувачам на місяць, що відкривають від 4 до 6 сторінок на один сайт. Даний показник обрано як репрезентативний для сегмента low-traffic ресурсів (малий бізнес, персональні блоги), що складають основну масу клієнтів Shared-хостингу [3].
- Агрегована середня інтенсивність системи: $\Lambda = n \cdot \lambda_{avg} = 1140$ зап/с. Це значення використовується як базова лінія для накладання часових коефіцієнтів циклічності.
- Конфігурація обчислювального вузла: $RAM_{total} = 64$ GiB, з яких $RAM_{os} = 4$ GiB виділено під системні потреби [3].
- Базовий квант ресурсу: $m_{unit} = 256$ MiB на одну сесію.

- Середня тривалість обслуговування запиту $t_{session}$: 1 с.
- Витрати на балансування: $M_{LB}=24$ GiB.
- Цільовий рівень якості обслуговування: $SLA=99{,}9\,\%$.
- Період оцінки: $T=168$ годин (один тиждень).

Обґрунтування базового кванту ресурсу (m_{unit}). Вибір величини $m_{unit}=256$ MiB базується на результатах попередніх досліджень автора [3], де було проаналізовано ліміти споживання пам'яті процесами інтерпретаторів (PHP-FPM, Python WSGI) при обробці типових динамічних запитів. Даний обсяг відповідає верхній межі безпечного виконання середньостатистичного скрипта, що дозволяє уникнути помилок переповнення при обробці сесій. Використання фіксованого кванту дозволяє перейти від абстрактних одиниць навантаження до дискретного планування ємності фізичних вузлів.

Обґрунтування витрат на балансування (M_{LB}). Для забезпечення відмовостійкості хмарно-кластерної моделі сумарний обсяг пам'яті, зарезервований під потреби балансування, встановлюється на рівні 24 GiB. Ця величина базується на необхідності функціонування високодоступної пари вузлів у режимі Active-Passive або Active-Active. Це дозволяє системі відповідати заявленому SLA 99,9% навіть під час технічного обслуговування, оновлення серверного програмного забезпечення або виходу з ладу одного з компонентів [9]. Сумарний обсяг у 24 GiB (2 вузли по 12 GiB) є теоретично обґрунтованим і враховує наступні фактори для кожного вузла:

- **Обробка вхідного трафіку**. Резервування пам'яті під мережеві буфери стека TCP/IP та черги обробки пакетів для обслуговування агрегованого потоку від 10 000 джерел навантаження.
- **Керування сесіями SSL/TLS**. Утримання в оперативній пам'яті кешу сесій та дескрипторів для 10 000 SSL-сертифікатів, що є критичним для забезпечення низької затримки при встановленні захищених з'єднань.
- **Синхронізація станів**. Додаткові витрати ресурсів на підтримку актуальності таблиць сесій між основним та резервним вузлами балансування. Логіка динамічного масштабування, робота алгоритмів моніторингу працездатності робочих вузлів та прийняття рішень про зміну кількості активних серверів $K_{req}(t)$ у реальному часі.

Обґрунтування цільового рівня якості обслуговування (SLA). Для моделювання встановлено цільовий показник імовірності відмови в обслуговуванні на рівні $P_{block}\le 0{,}001$, що відповідає рівню доступності ресурсів 99,9%. Даний вибір обґрунтований наступними факторами:

- **Галузеві стандарти**. Рівень "трьох дев'яток" (Three Nines) є базовим галузевим стандартом для систем хостингу та хмарних сервісів класу малого та середнього бізнесу (SMB), що підтверджується специфікаціями провайдерів та аналітичними звітами [12].
- **Баланс надійності та вартості**. Згідно з дослідженнями в галузі теорії масового обслуговування [13], подальше зниження P_{block} (наприклад, до 10^{-4}) для низьконавантажених систем призводить до експоненціального зростання витрат на надлишкове резервування.
- **Академічна прецедентність**. Дане значення часто використовується в роботах, присвячених верифікації моделей Ерланга в телекомунікаційних та обчислювальних мережах, як

репрезентативний поріг відчутної деградації сервісу. Зокрема, у дослідженнях архітектур обчислювальних систем та мереж масового обслуговування ймовірність відмови на рівні 0,001 вважається критичною межею для надання стабільного сервісу в умовах агрегованого трафіку [14].

Обґрунтування періоду оцінки. Для моделювання обрано часовий інтервал $T = 168$ годин (один тиждень). Такий вибір є науково обґрунтованим, оскільки дозволяє охопити повний цикл людської активності, що безпосередньо впливає на навантаження веб-ресурсів:

- **Добова циклічність.** Денний пік з 07:00 до 00:00 та нічний спад з 00:00 до 07:00. Згідно з дослідженнями CISCO Visual Networking Index та звітами Akamai State of the Internet, амплітуда коливань трафіку для загальних веб-ресурсів може досягати 3-кратної різниці [15-17]. В моделі прийнято рівень нічного спаду в 30% від денного максимуму. Динаміка вхідного навантаження Λ(t) моделюється відповідно до стандартного добового циклу, де інтервал 00:00 – 07:00 визначено як період мінімальної активності. Це дозволяє перевірити ефективність алгоритму динамічного керування ємністю кластера в умовах глибокого спаду навантаження.
- **Тижнева циклічність.** Зниження активності у вихідні дні на 15%. Даний показник корелює з галузевими даними [15-17], які демонструють стабільне просідання запитів до інформаційних та бізнес-ресурсів у суботу та неділю.
- **Облік SLA.** Більшість метрик доступності (uptime) розраховуються саме на щотижневій або щомісячній основі, що робить цей період стандартом для перевірки дотримання SLA.

Алгоритм проведення розрахунків

Процес верифікації реалізується у п’ять етапів:

Етап 1. Дискретизація часового простору. Весь період дослідження $T = 168$ годин розбивається на часові інтервали $\Delta t = 5$ хвилин. Вибір такого кроку дискретизації обумовлений технічною інерційністю хмарних платформ: процес ініціалізації нового обчислювального вузла, що включає запуск віртуальної машини, завантаження операційної системи та старт служб веб-сервера, у середньому триває від 2 до 3 хвилин [18]. Встановлення інтервалу $\Delta t > \text{Delay}$ дозволяє моделі уникати помилок керування, пов’язаних із затримкою розгортання ресурсів.

Етап 2. Генерація стохастичного профілю навантаження. Для кожного дискретного моменту часу t_i розраховується миттєве значення інтенсивності запитів $\Lambda(t_i)$. Математична модель базується на поєднанні детермінованого тренду (добові та тижневі флуктуації) та стохастичної складової, що описується розподілом Пуассона. З огляду на велику кількість джерел навантаження ($n = 10000$), згідно з центральною граничною теоремою, розподіл Пуассона апроксимується нормальним розподілом. Для верифікації системи в межах заданого рівня якості $SLA = 99{,}9\,\%$, розрахункове значення навантаження у кожній точці моделюється з урахуванням максимально допустимого статистичного сплеску:

$$\Lambda(t_i) = \Lambda_{plan}(t_i) + Z_p \cdot \sqrt{\Lambda_{plan}(t_i)}$$

де:

- $\Lambda_{plan}(t_i)$ - детерміноване значення інтенсивності згідно з часовим профілем;
- $\sqrt{\Lambda_{plan}(t_i)}$ - стандартне відхилення σ для пуассонівського потоку;
- $Z_p \approx 3{,}09$ - квантиль нормального розподілу, що відповідає односторонньому довірчому інтервалу для імовірності 0,999.

Використання одностороннього інтервалу обумовлене фізикою процесу. Критичним для стабільності системи є лише перевищення інтенсивності трафіку, тоді як відхилення в бік зменшення трафіку не створює ризиків порушення SLA. Такий підхід дозволяє отримати стохастично зважену криву навантаження, яка враховує рідкісні пікові сплески і є базою для подальшого розрахунку необхідної кількості серверів.

Етап 3. Розрахунок параметрів статичної моделі. Визначається глобальний пік навантаження $N_{max} = max\left(N(t_i)\right)$ за весь період T. Для статичної моделі параметр N_{max} визначається як абсолютний максимум інтенсивності агрегованого потоку, зафіксований протягом усього періоду моделювання T. Такий підхід імітує стратегію планування ємності традиційних провайдерів, які змушені утримувати максимальну кількість активних вузлів S для гарантованого дотримання SLA у моменти випадкових пікових сплесків навантаження. Таким чином, значення S залишається константою для всього часового інтервалу $[0, T]$. Оскільки в цій моделі ресурси розподілені рівномірно і статично, S обирається як мінімальне ціле, що забезпечує $P_{block} \le 1 - SLA$ при N_{max}. Обчислюється сумарна зарезервована пам'ять: M^{total}_{shared}.

Етап 4. Моделювання роботи кластера. Для кожного кроку t_i виконується наступний цикл:

- Визначається поточна інтенсивність $\Lambda(t_i)$.
- Шляхом ітераційного розв'язання оберненої задачі Ерланга знаходиться мінімально необхідна кількість активних вузлів $K_{req}(t_i)$. Алгоритм знаходження K базується на послідовному збільшенні кількості доступних ресурсних слотів $M = K \cdot C_{node}$ до моменту виконання умови $P_{block} \le 0{,}001$.
- Фіксується миттєве значення зарезервованої пам'яті з урахуванням витрат на балансування.

Етап 5. Розрахунок ефективності. Після завершення циклу проводиться чисельне інтегрування отриманих значень за методом прямокутників. Сумарний обсяг зарезервованих ресурсів за тиждень $M^{total}_{cluster}$ розраховується як:

$$M^{total}_{cluster} = \sum_{i=1}^{n} M_{cluster}(t_i) \cdot \Delta t$$

де $n = 2016$, це кількість 5-хвилинних інтервалів у тижневому періоді, а $\Delta t = 5/60$ години. На основі отриманих інтегральних значень розраховується підсумковий коефіцієнт ξ_{max}, що демонструє теоретичну межу економічної ефективності.

Аналіз результатів

В основу аналізу покладено результати імітаційного моделювання 168-годинного циклу роботи системи з кроком дискретизації $\Delta t = 5$ хвилин (всього 2016 розрахункових точок). На кожному кроці враховувався стохастичний сплеск інтенсивності вхідного потоку, що відповідає односторонньому довірчому інтервалу для $SLA = 99{,}9\,\%$.

Параметри традиційної моделі Shared Hosting. Згідно з отриманими даними, глобальний пік агрегованого

навантаження склав 1616,09 RPS. Для забезпечення цільового показника $P_{block} \le 0{,}001$ у моменти таких піків, провайдер традиційного хостингу змушений утримувати $S = 8$ обчислювальних вузлів (ємністю 240 слотів кожен). Верифікація статичної моделі шляхом зворотного розрахунку за формулою Ерланга-Б при рівномірному розподілі сайтів (1250 на один вузол) показала, що при піковому навантаженні на вузол у 202,01 Ерланг імовірність відмови становить [2,3]

$$P_{block} = B(202{,}01\,;240) \approx 0{,}00089$$

Це значення задовольняє умову $P_{block} \le 0{,}001$.

Динаміка хмарно-кластерної моделі. На відміну від статичного підходу, кластерна модель демонструє високу адаптивність до коливань трафіку. Результати моделювання зміни ємності кластера залежно від інтенсивності трафіку наведено на рис. 1. Кількість активних вузлів $K(t)$ змінювалася дискретно у діапазоні від 3 одиниць, у періоди глибокого нічного спаду, до 8 одиниць у моменти денних максимумів. Завдяки алгоритму динамічного масштабування, на кожному 5-хвилинному інтервалі кількість виділених слотів пам'яті $M(t)$ була оптимізована під поточне значення $\Lambda(t)$, що дозволило утримувати імовірність блокування запитів на рівні, близькому до граничного значення 0,001, не допускаючи при цьому значного перевитрачання ресурсів у години низької активності.

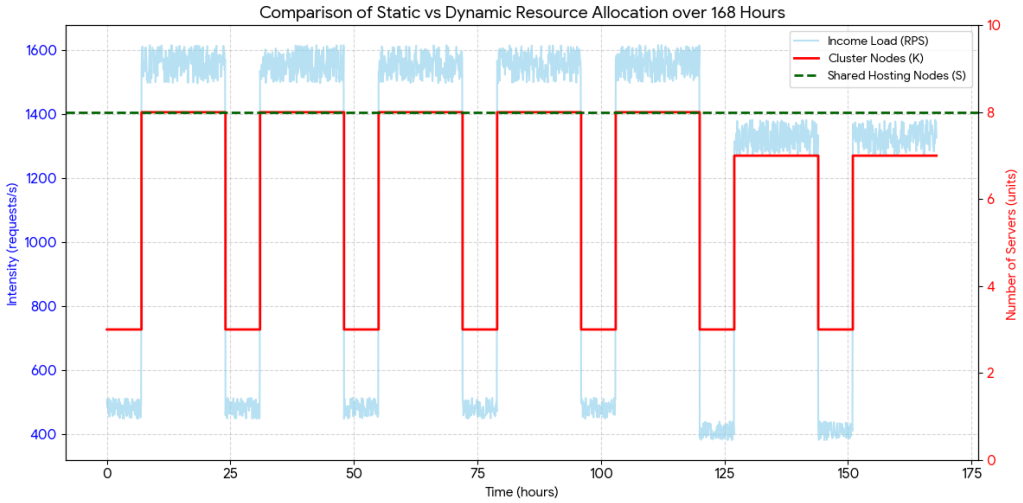

Рис. 1. Динаміка вхідного навантаження та відповідна зміна ємності кластера

Оцінка ефективності консолідації ресурсів

Для проведення порівняльного аналізу двох архітектур було розраховано сумарний інтегральний обсяг обчислювальних ресурсів, задіяних протягом усього періоду моделювання T. Згідно з формалізованим математичним апаратом, оцінка проводилася шляхом обчислення площі під кривою резервування пам'яті для кожної моделі.

Статична модель Shared Hosting. Оскільки кількість серверів $S = 8$ залишалася незмінною для забезпечення SLA у моменти пікового навантаження, сумарний обсяг зарезервованої пам'яті (M^{total}_{shared}) являє собою площу прямокутника з висотою 512 GiB (8×64 GiB). Результат інтегрування за весь період становить:

$M_{shared}^{total} = 8 \cdot 64\ \text{GiB} \cdot 168\ \text{год} = 86016\ \text{GiB} \cdot \text{год}$

Кластерна модель. Загальний обсяг ресурсів ($M_{cluster}^{total}$) розраховувався як інтеграл від сходинкової функції $K(t)$ з урахуванням постійних витрат на забезпечення високої доступності рівня балансування $M_{LB} = 24$ GiB. Чисельне інтегрування отриманих даних за методом прямокутників дало наступний результат:

$$M_{cluster}^{total} = \sum_{i=1}^{2016} \left(K_{req}(t_i) \cdot RAM_{total} + M_{LB} \right) \cdot \Delta t = 721$$

Підсумковий показник економії ресурсів, досягнутий за рахунок динамічного масштабування та консолідації трафіку, склав:

$$\xi_{max} = \left(1 - \frac{72192}{86016} \right) \cdot 100\,\% = 16{,}07\,\%$$

Важливо врахувати вплив вузла балансування та резервування дубльованого вузла, що створює постійне навантаження 24 GiB, яке в інтегральному обчисленні становить 4032 GiB · год або 5,6% від усього ресурсного бюджету кластера. Незважаючи на ці витрати, кластерна модель демонструє вищу ефективність завдяки здатності вивільняти ресурси у періоди низької активності. Слід підкреслити, що отримане значення $\xi_{max} = 16{,}1\,\%$ є консервативною нижньою межею і містить значний потенціал для подальшої оптимізації. Окрім зростання кількості сайтів n, що нівелює питому вагу витрат на балансування, додатковим важелем ефективності є оптимізація гранулярності обчислювальних вузлів.Використання менших за обсягом RAM вузлів (наприклад, 32 GiB замість 64 GiB) дозволить системі гнучкіше адаптуватися до коливань трафіку, мінімізуючи надлишкове резервування при округленні $K_{req}(t)$. Проте такий підхід вимагає врахування компромісу між дискретністю масштабування та сумарними витратами пам'яті на роботу операційних систем RAM_{os}, оскільки кожна додаткова фізична чи віртуальна одиниця збільшує питому частку системних витрат у загальному бюджеті кластера. Дослідження цього балансу є перспективним напрямком для пошуку глобального максимуму економічної ефективності хмарної інфраструктури.

Висновки

Запропонований метод планування ємності дозволяє здійснювати ефективну консолідацію ресурсів у масовому хмарному веб-хостингу, забезпечуючи заданий рівень надійності при зменшенні споживання оперативної пам'яті. Проведене дослідження підтвердило життєздатність переходу від статичного резервування ресурсів до динамічного хмарно-кластерного управління в сегменті низько навантажених веб-систем. Математичне моделювання на базі розподілу Пуассона та формули Ерланга дозволило отримати наступні результати:

- Доведено, що традиційна модель Shared-хостингу змушує провайдера утримувати близько 10-15% запасу пам'яті на кожному вузлі лише для покриття стохастичних піків. Кластерна модель успішно консолідує цей простій, перетворюючи його на доступний ресурс.
- Навіть за умови жорсткого дотримання SLA 99,9% та впровадження дубльованого рівня балансування, досягнуто реальну економію у 16%. Це підтверджує факт, що витрати на обслуговування хмарної інфраструктури значно нижчі ніж вигода від її гнучкості.
- Встановлено, що ефективність моделі прямо залежить від гранулярності обчислювальних вузлів та загальної кількості сайтів у пулі. Це відкриває шлях до створення самокерованих хостинг-

платформ, де вартість володіння ресурсом (TCO) динамічно знижується пропорційно зростанню кількості клієнтів.

Таким чином запропонований підхід є готовим математичним фундаментом для розробки алгоритмів інтелектуального масштабування в сучасних хмарних середовищах.

Література

1. Haan K. Top website statistics for 2025. Forbes Advisor. 2025. URL: https://www.forbes.com/advisor/business/software/website-statistics/ (дата звернення: 08.02.2026).

2. Chizhov, A., & Fesenko, A. (2025). Web hosting companies' client solutions: A study of a strategic standpoint [Special issue]. Corporate & Business Strategy Review, 6(1), 421–429. https://doi.org/10.22495/cbsrv6i1siart18

3. Чижов О., Фесенко А., Зюзюн В., Башикизи Д. Cloud shared hosting DDoS resistance and potential ways of protection // Proceedings of the Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2024), Kyiv, Ukraine, January 24–27, 2024. — CEUR Workshop Proceedings. — 2025. ISSN 1613-0073. — Vol-3925. — P. 13–23. — Режим доступу: https://ceur-ws.org/Vol-3925/paper02.pdf (дата звернення: 08.02.2026).

4. Чижов О., Фесенко А., Пустосвіт М., Німченко Т. Методи оптимізації розподілу навантаження на обчислювальний ресурс інфраструктури хмарного сервісу // Захист інформації. — 2023. — Т. 25, № 4. — С. 207–213. — DOI: 10.18372/2410-7840.25.18226.

5. Lorido-Botran J. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments / J. Lorido-Botran, J. Miguel-Alonso, J. A. Lozano // Journal of Grid Computing. — 2016. — Vol. 14, no. 2. — P. 217–250. — DOI: 10.1007/s10723-015-9325-2.

6. Kleinrock L. Queueing Systems. Volume I: Theory / L. Kleinrock. — New York : Wiley-Interdisciplinary, 1975. — 417 p.

7. Erlang A. K. Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephony Exchanges / A. K. Erlang // The Post Office Electrical Engineers' Journal. — 1918. — Vol. 10. — P. 189–197.

8. Hybrid resource provisioning for cloud applications / A. Gandhi, Y. Chen, D. Gmach et al. // ASPLOS '14: Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems. — 2014. — P. 543–558. — DOI: 10.1145/2541940.2541946.

9. Чижов О., Фесенко А. Intelligent load balancing management in cloud web hosting: evaluation criteria and methodology // Проблеми інформатизації та управління. — 2025. — Т. 83, № 3. — С. 138–147. — DOI: 10.18372/2073-4751.83.20551.

10. The NIST Definition of Cloud Computing : Recommendations of the National Institute of Standards and Technology / P. Mell, T. Grance. — Gaithersburg : NIST, 2011. — 7 p. — (Special Publication 800-145). — DOI: 10.6028/NIST.SP.800-145.

11. Google. Improve server response time (PageSpeed Insights) [Електронний ресурс]. – Режим доступу: https://developers.google.com/speed/docs/insights/Server – Дата звернення: 08.02.2026.

12. Tier Standard: Topology [Електронний ресурс] // Uptime Institute Professional Services. — 2022. — 18 p. — Access mode: https://uptimeinstitute.com/tier-standard-topology (дата звернення: 08.02.2026)

13. Ivers J., Troya J., Moreno-Torres M. [та ін.] Evaluating the trade-offs between cost and availability in cloud configurations // 2018 IEEE World Congress on Services (SERVICES). —

IEEE, 2018. — P. 45–52. — DOI: 10.1109/SERVICES.2018.00034.

14. Girard A. Routing and Dimensioning in Circuit-Switched Networks. — Boston : Addison-Wesley Longman Publishing Co., Inc., 1990. — 556 p. — ISBN 0-201-12792-X.

15. Cisco Systems, Inc. Cisco Annual Internet Report (2018–2023) White Paper [Електронний ресурс]. — 2020. — Режим доступу: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf (дата звернення: 08.02.2026).

16. Akamai Technologies. State of the Internet Research Reports [Електронний ресурс]. — Режим доступу: https://www.akamai.com/resources/state-of-the-internet (дата звернення: 08.02.2026).

17. Cloudflare. Cloudflare Radar: Global Internet Traffic Patterns and Insights [Електронний ресурс]. — 2024. — Режим доступу: https://radar.cloudflare.com/ (дата звернення: 08.02.2026).

18. Mao M. A Performance Study on the VM Startup Time in the Cloud / M. Mao, M. Humphrey // 2012 IEEE Fifth International Conference on Cloud Computing. — IEEE, 2012. — P. 423–430. — DOI: 10.1109/CLOUD.2012.103.

Чижов Олександр, Фесенко Андрій

МЕТОД ПЛАНУВАННЯ ПОТУЖНОСТЕЙ ДЛЯ КОНСОЛІДАЦІЇ РЕСУРСІВ У МАСОВОМУ ХМАРНОМУ ВЕБ-ХОСТИНГУ

У науковій статті запропоновано метод планування ємності для консолідації ресурсів у сегменті масового хмарного веб-хостингу, що базується на математичному моделюванні агрегованих стохастичних навантажень і теорії масового обслуговування. Метод забезпечує оптимізацію розподілу оперативної пам'яті для низько навантажених ресурсів за рахунок консолідації трафіку в єдиний обчислювальний пул при суворому дотриманні цільового рівня надійності (SLA 99,9%). Шляхом імітаційного моделювання тижневого циклу роботи системи показано, що запропонований метод забезпечує інтегральну економію ресурсів на рівні 16%, нівелює вплив стохастичних піків навантаження та знижує операційні витрати провайдера без деградації якості сервісу.

***Ключові слова**: хмарні обчислення, хостинг, формула Ерланга, низько навантажені веб-ресурси, динамічне масштабування, кластерізація, SLA, оптимізація ресурсів, масове обслуговування.*

Chyzhov Oleksandr, Fesenko Andriy

CAPACITY PLANNING METHOD FOR RESOURCE CONSOLIDATION IN MASS CLOUD WEB HOSTING

In this scientific article, a capacity planning method is proposed for resource consolidation in the mass cloud web hosting segment, based on mathematical modeling of aggregated stochastic loads and queueing theory. The method optimizes the allocation of RAM for low-traffic resources by consolidating traffic into a single computing pool while strictly maintaining the target reliability level (SLA 99.9%). Through simulation of the system's weekly operating cycle, it is shown that the proposed method provides an integrated resource saving of 16%, mitigates the impact of stochastic load peaks, and reduces the provider's operational costs without degrading service quality.

Keywords: cloud computing, hosting, Erlang formula, low-traffic web resources, dynamic scaling, clustering, SLA, resource optimization, queueing theory.

Стаття подана до редакції: 02/11/2025
Стаття прийнята до опублікування: 15/11/2025
Стаття опублікована: 30/12/2025