

## АНАЛІТИЧНА МОДЕЛЬ ВПЛИВУ ПАРАЛЕЛІЗМУ НА ПРОДУКТИВНІСТЬ СИСТЕМИ ВПОРЯДКОВАНОЇ ДОСТАВКИ ПОДІЙ З AT-LEAST-ONCE СЕМАНТИКОЮ

У сучасних розподілених інформаційних системах, що базуються на мікросервісних і подієво-орієнтованих архітектурах, широко використовуються асинхронні механізми обміну повідомленнями через брокери подій. Однією з важливих проблем таких систем є забезпечення впорядкованої обробки подій при використанні семантики доставки at-least-once, яка гарантує доставку повідомлення щонайменше один раз, але допускає його повторне надходження. За умов паралельної обробки подій виникають додаткові фактори, що впливають на продуктивність системи, зокрема витрати на перевірку ідемпотентності, повторну обробку повідомлень, буферизацію подій для відновлення їх правильного порядку, використання резервних каналів доставки та конкуренцію за спільні ресурси. Традиційні моделі аналізу продуктивності паралельних систем, такі як ідеальна модель масштабування та модель Амдала, не враховують специфічних особливостей подієво-орієнтованих систем доставки повідомлень. Метою роботи є розроблення аналітичної моделі впливу паралелізму на продуктивність системи впорядкованої доставки подій з семантикою at-least-once. У роботі запропоновано узагальнену математичну модель, яка враховує витрати ідемпотентної обробки, повторної доставки повідомлень, затримки буферизації подій, використання резервних каналів передачі та конкуренцію за спільні ресурси системи. Крім того, у моделі враховано нерівномірність розподілу навантаження між ключами подій, що дозволяє оцінювати ефективний рівень паралелізму та визначати межі масштабованості системи. На основі запропонованої моделі отримано аналітичну залежність пропускну здатності системи від кількості паралельних обробників і визначено коефіцієнт ефективності використання паралельних ресурсів. Проведено порівняльний аналіз із класичними моделями масштабування, що показав більш точний опис поведінки подієво-орієнтованих систем. Практична цінність роботи полягає у можливості використання моделі для прогнозування продуктивності та оптимізації параметрів конфігурації розподілених мікросервісних систем.

**Ключові слова:** подієво-орієнтована архітектура, паралельна обробка подій, пропускну здатність системи, мікросервісні системи, at-least-once delivery, продуктивність розподілених систем, інформаційні технології.

### Вступ

Сучасні інформаційні системи дедалі частіше реалізуються на основі подієво-орієнтованих архітектур (Event-Driven Architecture), які забезпечують слабе зв'язування компонентів, масштабованість і високу відмовостійкість розподілених програмних систем. У таких архітектурах взаємодія між сервісами здійснюється через передачу потоків подій за допомогою брокерів повідомлень або платформ потокової обробки даних. Використання поточкових обчислень дозволяє забезпечити обробку великих обсягів даних у режимі близькому до реального часу та є ключовим елементом сучасних систем аналітики, систем моніторингу, фінансових платформ, а також систем Інтернету речей.

Однією з фундаментальних проблем таких систем є забезпечення коректності доставки та обробки подій у розподіленому середовищі. Через наявність мережових затримок, відмов вузлів, повторних спроб доставки та асинхронності виконання процесів виникають ситуації, коли події можуть доставлятися кілька разів або надходити у неправильному порядку. У зв'язку з цим системи потокової обробки використовують різні семантики доставки повідомлень, зокрема at-most-once, at-least-once та exactly-once. Семантика at-least-once гарантує, що подія буде доставлена щонайменше один раз, проте допускає її повторну доставку, що потребує спеціальних механізмів для обробки дублікатів.

У більшості промислових систем саме семантика at-least-once є найбільш поширеною, оскільки вона дозволяє забезпечити високу надійність доставки повідомлень без значних накладних витрат, характерних для механізмів exactly-once. При цьому уникнення повторної обробки подій зазвичай досягається за рахунок використання ідемпотентних операцій та механізмів дедуплікації на стороні споживача.

Іншою важливою проблемою є збереження правильного порядку подій. У розподілених системах відсутній глобальний годинник, тому визначення порядку подій потребує

використання спеціальних алгоритмів, наприклад логічних часових міток або інших механізмів упорядкування.

Порушення порядку обробки подій може призводити до некоректного стану системи, особливо у випадках, коли події представляють послідовність змін одного й того самого об'єкта або транзакції.

У сучасних мікросервісних системах обробка подій зазвичай виконується паралельно декількома споживачами для підвищення пропускну здатності системи. Проте збільшення рівня паралелізму може призводити до зростання затримок обробки, появи додаткових накладних витрат на синхронізацію, буферизацію подій та повторну доставку повідомлень. Внаслідок цього виникає задача визначення оптимального рівня паралелізму, який забезпечує максимальну продуктивність системи при збереженні коректності порядку обробки подій.

Незважаючи на значну кількість досліджень у сфері потокової обробки даних та розподілених систем, питання формального моделювання впливу паралельної обробки на продуктивність систем впорядкованої доставки подій із семантикою *at-least-once* залишається недостатньо дослідженим. Більшість існуючих робіт зосереджена на архітектурних аспектах побудови систем потокової обробки або на механізмах забезпечення узгодженості даних, тоді як аналітичні моделі продуктивності таких систем розглядаються обмежено.

### **Мета та постановка задачі**

Отже, актуальною науковою задачею є розробка аналітичної моделі, що дозволяє оцінювати вплив рівня паралелізму на продуктивність і затримку систем впорядкованої доставки подій з урахуванням повторної доставки, ідемпотентної обробки та буферизації подій.

Метою роботи є розробка аналітичної моделі впливу паралелізму на продуктивність системи впорядкованої доставки подій з семантикою *at-least-once* та експериментальна перевірка адекватності запропонованої моделі.

### **Аналіз дотичних робіт**

Проблематика впорядкованої обробки подій у розподілених і паралельних системах перебуває в центрі уваги сучасних досліджень, оскільки саме порядок виконання подій, механізми їх узгодження та особливості масштабування істотно впливають на коректність функціонування і продуктивність програмних систем. Водночас аналіз наукових публікацій показує, що наявні роботи переважно зосереджені або на питаннях детермінованого впорядкування подій, або на архітектурних аспектах подієво-орієнтованих систем, або на загальних підходах до підвищення продуктивності паралельних обчислень, тоді як аналітичне моделювання впливу паралелізму на продуктивність систем впорядкованої доставки подій з семантикою *at-least-once* залишається недостатньо опрацьованим.

Одним із базових напрямів досліджень є формування детермінованого порядку подій у паралельних системах. Зокрема, у роботі [1] розглянуто проблему неупередженого детермінованого тотального впорядкування подій у паралельному дискретно-подієвому моделюванні. Автори показують, що навіть за одночасного настання подій питання коректного визначення порядку їх виконання є критичним для відтворюваності результатів моделювання. Подібний напрям продовжено в роботі тих самих авторів [10], де систематизовано підходи до детермінованого впорядкування подій у паралельних системах моделювання. Значення цих досліджень для даної роботи полягає в тому, що вони підтверджують фундаментальну важливість проблеми впорядкування, однак основний акцент у них зроблено на детермінізмі симуляційних систем, а не на аналітичному оцінюванні продуктивності поточкових мікросервісних архітектур.

Інший напрям представлений роботами, присвяченими оптимізації алгоритмів паралельного виконання подій. У праці [5] запропоновано алгоритм *Window Racer* для паралельного дискретно-подієвого моделювання, орієнтований на підвищення продуктивності за рахунок ефективного керування вікном обробки подій. У роботі [9]

досліджено динамічну синхронізацію часу з урахуванням поточного навантаження, що також спрямовано на підвищення ефективності паралельного виконання подій. Ці роботи демонструють, що зі зростанням паралелізму виникають додаткові накладні витрати на координацію та синхронізацію. Проте вони стосуються переважно систем паралельного дискретно-подієвого моделювання і не враховують таких характерних для мікросервісних систем чинників, як повторна доставка повідомлень, ідемпотентність та буферизація подій поза порядком.

Важливий клас досліджень стосується впорядкованого виконання транзакцій і подій у розподілених транзакційних середовищах. Так, у роботах [4, 6] розглядається *ordered scheduling* у *control-flow distributed transactional memory*, де впорядкування операцій використовується для забезпечення коректності виконання паралельних транзакцій. У публікації [8] досліджується обробка розподілених транзакцій у заздалегідь визначеному порядку, що є близьким до задачі збереження послідовності подій у системах доставки повідомлень. Ці роботи мають важливе методологічне значення, оскільки демонструють, що впорядкування операцій є необхідною умовою коректного функціонування розподілених систем. Водночас вони орієнтовані переважно на транзакційні механізми, а не на аналітичне моделювання пропускну здатності систем подієвої взаємодії з повторною доставкою повідомлень.

Окремо слід відзначити дослідження, присвячені масштабованим середовищам виконання подій. У роботі [2] представлено фреймворк *Devastator* для сучасного паралельного дискретно-подієвого моделювання, який орієнтований на високу продуктивність і масштабованість. Це дослідження підтверджує, що збільшення кількості паралельних виконавців саме по собі не гарантує лінійного зростання продуктивності, оскільки з'являються витрати на координацію, планування та підтримку коректного порядку подій. Проте, як і в інших роботах цього напрямку, головний акцент зроблено на архітектурних і системних аспектах реалізації платформи, а не на побудові узагальненої аналітичної моделі.

Значний інтерес становлять також роботи, що досліджують архітектурні рішення для подієво-орієнтованих систем високої доступності. Зокрема, у публікації [3] подієво-орієнтований підхід розглянуто в контексті резервування і відмовостійкості фінансових платформ. Автор підкреслює роль резервних каналів доставки та механізмів *failover* у забезпеченні безперервності обробки транзакційних подій. Робота обґрунтовує важливість урахування затримок у резервних каналах, проте в ній відсутній формальний аналітичний опис їхнього впливу на пропускну здатність системи залежно від кількості паралельних обробників.

Ще один близький напрям охоплює дослідження механізмів узгодженого багатоадресного доставлення повідомлень у розподілених системах. У роботі [7] запропоновано протокол *FlexCast*, орієнтований на ефективну атомарну *multicast*-доставку. Хоча ця робота стосується іншого класу систем, вона демонструє, що підтримка узгодженості та впорядкованості повідомлень у розподіленому середовищі неминуче супроводжується додатковими накладними витратами. Саме цей висновок є принципово важливим для задачі побудови моделі продуктивності систем *ordered delivery*.

Наявні дослідження формують теоретичне та прикладне підґрунтя для вивчення впорядкованої обробки подій у паралельних і розподілених середовищах. У цих роботах розглянуто питання детермінованого впорядкування подій [1, 10], алгоритмів паралельного дискретно-подієвого моделювання [2, 5, 9], впорядкованого виконання транзакцій [4, 6, 8], відмовостійкої подієво-орієнтованої архітектури [3] та протоколів узгодженого доставлення повідомлень [7]. Проте більшість із них або не орієнтована на мікросервісні системи з семантикою *at-least-once*, або не враховує сукупний вплив повторної доставки, ідемпотентної обробки, буферизації подій поза порядком, резервних каналів передачі та нерівномірності розподілу навантаження між ключами. Саме це зумовлює наявність наукової прогалини, яка полягає у відсутності узагальненої аналітичної моделі, здатної формально оцінювати вплив

рівня паралелізму на пропускну здатність системи впорядкованої доставки подій з семантикою *at-least-once*. Розроблення такої моделі є необхідним для прогнозування поведінки системи при зміні рівня паралелізму, оцінювання її межі масштабованості та визначення оптимальної конфігурації обробників.

#### Виклад основного матеріалу дослідження

Побудована аналітична модель дозволяє формально оцінити вплив рівня паралелізму на продуктивність системи впорядкованої доставки подій з урахуванням повторної доставки, ідемпотентної обробки, буферизації подій та конкуренції за ресурси. Для практичного використання моделі необхідно визначити послідовність її розрахунку, яка дозволяє перейти від первинних параметрів системи до інтегральних характеристик її продуктивності.

На початковому етапі визначаються базові параметри потоку подій та процесу обробки. До таких параметрів належать інтенсивність генерації подій  $\lambda$ , середній базовий час обробки однієї події  $T_p$ , а також кількість паралельних обробників  $k$ . На основі цих параметрів визначається базова швидкість обробки одного обробника:

$$\mu = \frac{1}{T_p}. \quad (1)$$

Ця величина характеризує продуктивність одного сервісу-обробника за відсутності будь-яких накладних витрат. Виходячи з цього, можна визначити теоретичну пропускну здатність системи при використанні  $k$  паралельних обробників:

$$X_{ideal}(k) = k\mu = \frac{k}{T_p}. \quad (2)$$

Дана величина відповідає ідеалізованому випадку, коли система масштабується лінійно і не має жодних додаткових витрат.

Наступним етапом є врахування семантики доставки подій *at-least-once*, яка допускає повторну доставку повідомлень. Нехай  $p_r$  – імовірність повторної доставки події. У цьому випадку середня кількість доставок однієї події визначається математичним сподіванням:

$$E[N_d] = \frac{1}{1 - p_r}. \quad (3)$$

Відповідно фактична інтенсивність потоку подій, яка надходить до обробників, зростає і визначається як:

$$\lambda_{eff} = \lambda E[N_d] = \frac{\lambda}{1 - p_r}. \quad (4)$$

Цей крок дозволяє перейти від номінального потоку подій до реального навантаження, яке виникає у системі з урахуванням повторних доставок.

Далі модель враховує витрати, пов'язані з ідемпотентною обробкою подій. Оскільки одна й та сама подія може бути доставлена кілька разів, перед її обробкою виконується перевірка ідентифікатора події. Нехай середній час такої перевірки становить  $T_i$ . Окрім цього, повторна доставка може спричинити додаткові витрати часу на повторну обробку повідомлень. Якщо позначити середній час такої обробки через  $\beta$ , то середній внесок повторної доставки до часу обробки визначається як:

$$T_r = \beta \frac{p_r}{1 - p_r}. \quad (5)$$

Після врахування витрат повторної доставки та перевірки ідемпотентності модель переходить до аналізу впливу паралельної обробки на порядок виконання подій. У системах, де події можуть оброблятися кількома обробниками одночасно, можливе порушення їх

початкового порядку. Для відновлення правильного порядку використовується буфер подій, що призводить до додаткових затримок. Для формалізації цього явища вводиться коефіцієнт варіації часу обробки:

$$CV = \frac{\sigma}{T_p}, \quad (6)$$

де  $\sigma$  – стандартне відхилення часу обробки. На основі цього коефіцієнта середню затримку буферизації можна оцінити як:

$$T_b = \alpha k CV, \quad (7)$$

де  $\alpha$  – коефіцієнт чутливості системи до порушення порядку подій.

Окрім цього, в реальних системах можуть виникати додаткові затримки, пов'язані з використанням резервних каналів передачі повідомлень. Якщо  $p_h$  – імовірність використання резервного каналу, а  $T_h$  – середня затримка передачі через нього, то середній часовий внесок такого механізму становить:

$$T_{h,eff} = p_h T_h. \quad (8)$$

Ще одним важливим фактором є конкуренція за спільні ресурси системи, яка виникає при збільшенні кількості паралельних обробників. Для опису цього ефекту введемо коефіцієнт конкуренції  $\gamma$ , який визначає середні витрати часу на синхронізацію доступу до ресурсів. Відповідні витрати можна записати як:

$$T_c = \gamma k. \quad (9)$$

Таким чином, сумарні накладні витрати системи визначаються як:

$$T_{overhead}(k) = T_i + T_r + T_b + T_{h,eff} + T_c. \quad (10)$$

Після підстановки отримуємо:

$$T_{overhead}(k) = T_i + \beta \frac{p_r}{1 - p_r} + \alpha k CV + p_h T_h + \gamma k. \quad (11)$$

Відповідно ефективний середній час обробки однієї події набуває вигляду:

$$T_{eff}(k) = T_p + T_{overhead}(k). \quad (12)$$

На основі цього значення визначається ефективна швидкість обслуговування одного обробника:

$$\mu_{eff}(k) = \frac{1}{T_{eff}(k)}. \quad (13)$$

Однак реальна продуктивність системи визначається не лише часом обробки, а й рівномірністю розподілу подій між ключами. У багатьох практичних сценаріях існують так звані «гарячі ключі», які генерують значно більшу кількість подій. Для врахування цього явища вводиться коефіцієнт нерівномірності:

$$s = \frac{\lambda_{max}}{\lambda_{avg}}, \quad (14)$$

де  $\lambda_{max}$  – максимальна інтенсивність потоку подій для одного ключа, а  $\lambda_{avg}$  – середня інтенсивність потоку подій.

З урахуванням цього коефіцієнта ефективна кількість обробників визначається як:

$$k_{eff} = \frac{k}{s}. \quad (15)$$

Після цього можна визначити узагальнену пропускну здатність системи:

$$X_g(k) = \frac{k_{eff}}{T_{eff}(k)}. \quad (16)$$

Підставивши попередні співвідношення, отримаємо остаточний вираз:

$$X_g(k) = \frac{k/s}{T_p + T_i + \beta \frac{p_r}{1-p_r} + \alpha kCV + p_h T_h + \gamma k}. \quad (17)$$

Отримана залежність дозволяє визначити реальну продуктивність системи з урахуванням усіх основних факторів, що впливають на обробку подій.

Для оцінювання ефективності використання паралельних ресурсів вводиться коефіцієнт ефективності паралелізму:

$$E_p(k) = \frac{X_g(k)}{X_{ideal}(k)}. \quad (18)$$

Цей показник дозволяє порівняти реальну продуктивність системи з її теоретичною максимальною продуктивністю.

На завершальному етапі виконується перевірка стійкості системи. Для цього обчислюється коефіцієнт завантаження:

$$\rho = \frac{\lambda_{eff}}{X_g(k)}. \quad (19)$$

Якщо виконується умова  $\rho < 1$ , система працює у стійкому режимі. В іншому випадку система є перевантаженою, і необхідно змінити параметри конфігурації.

Після перевірки стійкості система може бути оптимізована шляхом підбору кількості обробників. Оптимальний рівень паралелізму визначається як значення:

$$k^* = \arg \max_k X_g(k). \quad (20)$$

Отриманий алгоритм дозволяє послідовно оцінити всі складові продуктивності системи та визначити оптимальні параметри її конфігурації.

Для оцінювання ефективності запропонованої аналітичної моделі доцільно виконати її порівняння з базовими моделями масштабування, які традиційно використовуються для аналізу продуктивності паралельних і розподілених систем. У даному дослідженні як базові підходи використано ідеальну модель масштабування, а також класичну модель Амдала, що дозволяє оцінити граничні властивості паралельних обчислень. Порівняння з цими моделями дозволяє показати, яким чином врахування специфічних факторів подієво-орієнтованих систем, таких як повторна доставка повідомлень, впорядкування подій та нерівномірність розподілу навантаження, впливає на результати моделювання продуктивності.

Однією з найпростіших моделей оцінювання продуктивності паралельних систем є ідеальна модель масштабування, яка ґрунтується на припущенні про повну незалежність паралельних операцій та відсутність накладних витрат на синхронізацію і взаємодію між обчислювальними вузлами. У межах цієї моделі передбачається, що кожен додатковий обробник забезпечує пропорційне збільшення продуктивності системи. Нехай  $T_p$  – середній час обробки однієї події одним обробником, а  $k$  – кількість паралельних обробників. Тоді швидкість обробки одного обробника визначається як  $\mu = \frac{1}{T_p}$ , а пропускна здатність системи при використанні  $k$  обробників визначається співвідношенням:

$$X_{ideal}(k) = k\mu = \frac{k}{T_p}. \quad (21)$$

Особливістю цієї моделі є те, що вона не враховує жодних додаткових витрат, пов'язаних із передаванням повідомлень, синхронізацією або доступом до спільних ресурсів. Тому вона

використовується переважно як теоретична верхня межа продуктивності, яка показує максимально можливу швидкість обробки подій за умови повністю лінійного масштабування системи. У контексті подієво-орієнтованих мікросервісних архітектур ця модель має обмежене застосування, оскільки в реальних системах виникають накладні витрати, пов'язані з механізмами доставки повідомлень, обробкою дублікатів та підтримкою впорядкованості подій.

Більш реалістичним підходом до аналізу продуктивності паралельних систем є модель Амдала, яка враховує наявність послідовної частини алгоритму. Основна ідея цієї моделі полягає в тому, що не всі операції можуть бути виконані паралельно, тому максимальне прискорення системи обмежується часткою послідовних обчислень.

Нехай  $P$  – частка операцій, які можуть виконуватися паралельно, а  $1 - P$  – частка послідовних операцій. У такому випадку прискорення системи при використанні  $k$  обробників визначається формулою:

$$S(k) = \frac{1}{(1 - P) + \frac{P}{k}}. \quad (22)$$

Якщо вважати, що базовий час обробки однієї події одним обробником дорівнює  $T_p$ , тоді ефективний час обробки події при використанні  $k$  обробників можна записати як:

$$T_{amdahl}(k) = T_p \left( (1 - P) + \frac{P}{k} \right). \quad (23)$$

Відповідно пропускна здатність системи в рамках моделі Амдала визначається як:

$$X_{amdahl}(k) = \frac{1}{T_{amdahl}(k)} = \frac{1}{T_p \left( (1 - P) + \frac{P}{k} \right)}. \quad (24)$$

Особливістю цієї моделі є те, що вона дозволяє оцінити граничний ефект масштабування: навіть при необмеженому збільшенні кількості обробників швидкість системи не може перевищити значення  $\frac{1}{T_p(1-P)}$ .

Таким чином, модель Амдала добре описує системи, у яких продуктивність обмежується послідовними операціями. Проте в контексті систем доставки подій вона має певні обмеження, оскільки не враховує специфічних механізмів, характерних для подієво-орієнтованих архітектур, зокрема повторну доставку повідомлень, необхідність ідемпотентної обробки, буферизацію подій для підтримки їх порядку, а також нерівномірність розподілу навантаження між ключами подій.

На відміну від зазначених моделей, запропонована у роботі аналітична модель враховує комплекс факторів, що виникають у системах впорядкованої доставки подій з семантикою at-least-once. Зокрема, у ній формалізовано вплив повторної доставки повідомлень, витрат на перевірку ідемпотентності, затримок буферизації подій, використання резервних каналів передачі та конкуренції за спільні ресурси системи. Крім того, модель враховує нерівномірність розподілу навантаження між ключами подій, що є характерною особливістю реальних подієво-орієнтованих систем. У результаті запропонована модель дозволяє отримати більш точну оцінку пропускної здатності системи при різних рівнях паралелізму та визначити оптимальну кількість обробників, при якій досягається максимальна продуктивність системи.

Таким чином, використання ідеальної моделі масштабування та моделі Амдала як базових підходів для порівняння дозволяє продемонструвати відмінності між класичними моделями паралельних обчислень та запропонованою моделлю, яка орієнтована на аналіз продуктивності подієво-орієнтованих мікросервісних систем із впорядкованою доставкою подій.

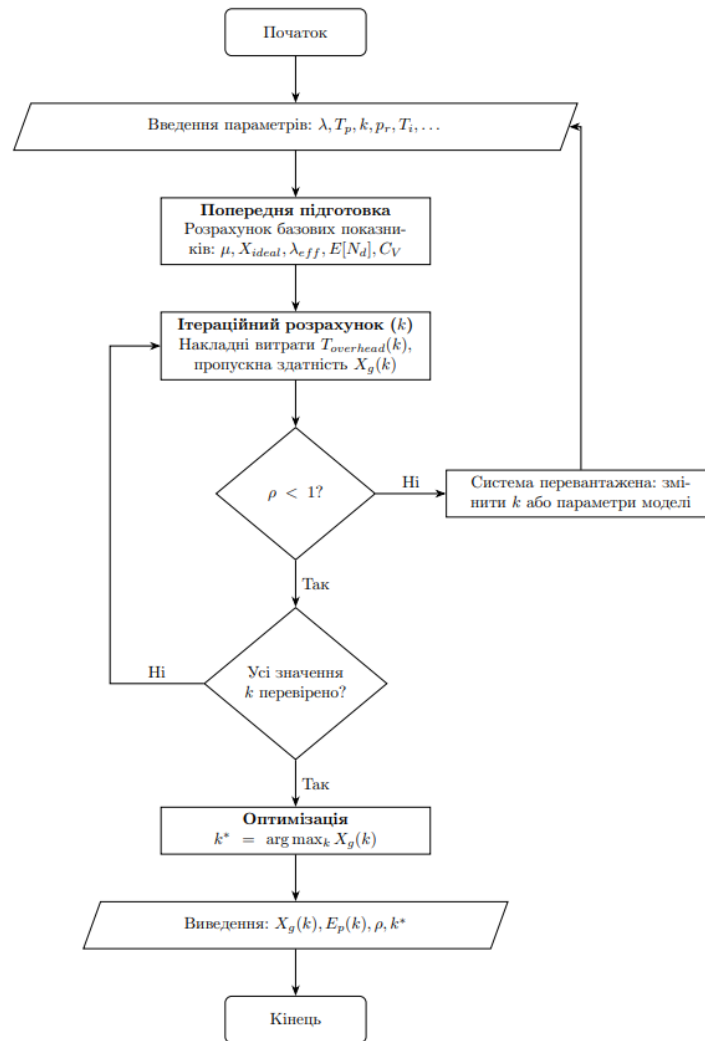


Рис. 1. Блок-схема алгоритму оцінювання продуктивності та пошуку оптимального рівня паралелізму

Таблиця 1

Порівняння запропонованої моделі з базовими підходами

Модель	Врахування повторної доставки	Врахування впорядкування	Врахування нерівномірності ключів	Реалістичність
Ідеальна	-	-	-	низька
Модель Амдала	-	-	-	середня
Запропонована модель	+	+	+	висока

Для оцінювання ефективності запропонованої моделі було проведено порівняльний аналіз із класичними моделями масштабування, зокрема ідеальною моделлю та моделлю Амдала. Результати порівняння наведено на рисунках 2, 3 та 4.

Як видно з графіків, ідеальна модель демонструє лінійне зростання пропускної здатності зі збільшенням кількості обробників. Модель Амдала показує наявність межі масштабування, обумовленої послідовною частиною обчислень.

Запропонована модель враховує додаткові накладні витрати, характерні для систем впорядкованої доставки подій, зокрема повторну доставку повідомлень, ідемпотентну обробку та буферизацію подій. У результаті отримана залежність більш точно описує

© Колодюк А.В., Аронов А.О. Аналітична модель впливу паралелізму на продуктивність системи впорядкованої доставки подій з At-Least-Once семантикою. Сучасний захист інформації, 1(65), 169–179. <https://doi.org/10.31673/2409-7292.2026.011987>

поведінку подієво-орієнтованих мікросервісних систем і дозволяє визначати реальну межу масштабованості.

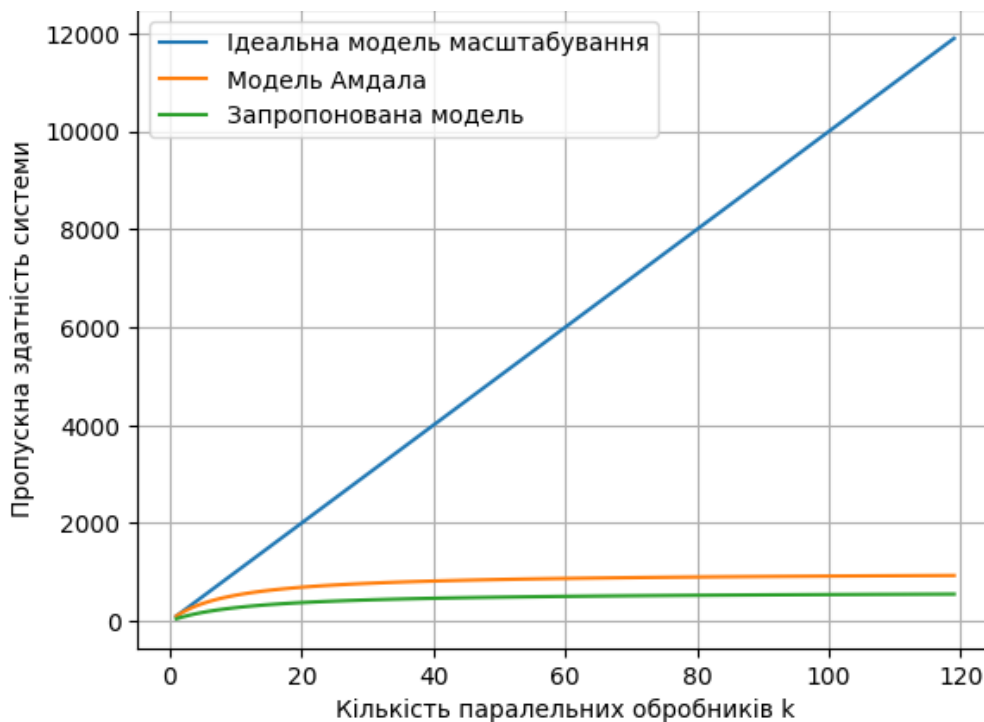


Рис. 2. Порівняння пропускної здатності різних моделей

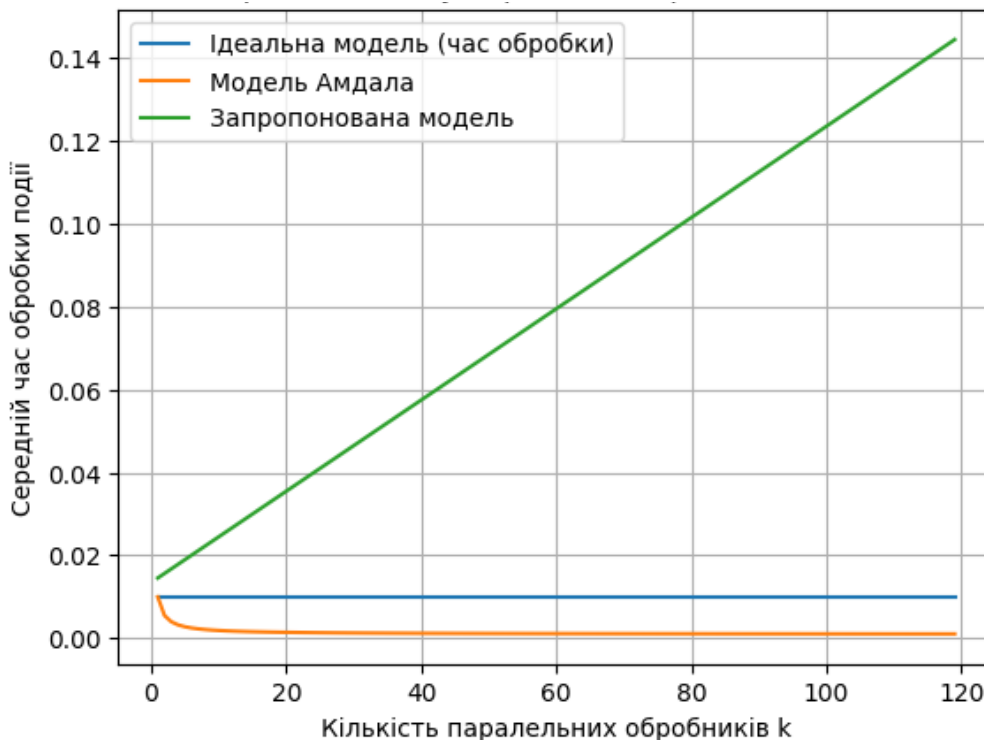


Рис. 3. Порівняння часу обробки для різних моделей

Для більш глибокого аналізу відмінностей між класичними та запропонованою моделями було побудовано графік втрати масштабованості (*scalability gap*), який визначається як різниця між ідеальною пропускною здатністю системи та реальною продуктивністю

© Колодюк А.В., Аронов А.О. Аналітична модель впливу паралелізму на продуктивність системи впорядкованої доставки подій з At-Least-Once семантикою. Сучасний захист інформації, 1(65), 169–179. <https://doi.org/10.31673/2409-7292.2026.011987>

відповідної моделі. Такий підхід дозволяє оцінити, наскільки кожна з моделей відхиляється від лінійного масштабування при збільшенні кількості паралельних обробників.

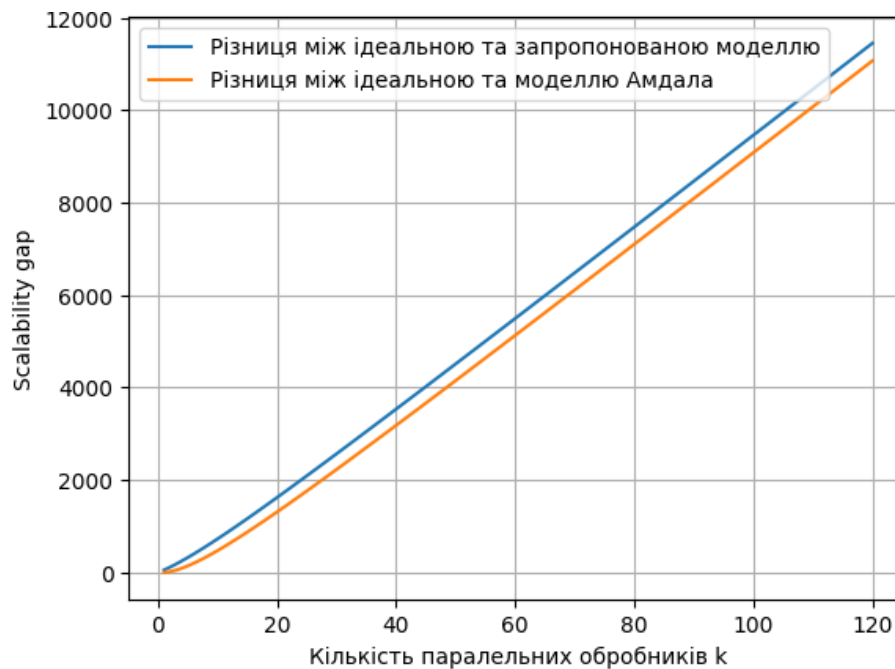


Рис. 4. Порівняння витрат масштабованості для різних моделей

Отримані результати показують, що зі зростанням рівня паралелізму втрата масштабованості збільшується для всіх моделей, однак для запропонованої моделі це зростання є більш вираженим. Це пояснюється тим, що на відміну від моделі Амдала, запропонований підхід враховує додаткові фактори, характерні для систем впорядкованої доставки подій з семантикою *at-least-once*, а саме повторну доставку повідомлень, перевірку ідемпотентності, буферизацію подій для підтримки їх порядку та конкуренцію за спільні ресурси.

Таким чином, графік втрати масштабованості підтверджує, що запропонована модель більш реалістично описує поведінку високонавантажених подієво-орієнтованих систем у порівнянні з класичними моделями масштабування.

### Висновки

У роботі розроблено узагальнену аналітичну модель продуктивності системи впорядкованої доставки подій з семантикою *at-least-once*, яка дозволяє формалізовано врахувати основні фактори, що впливають на ефективність паралельної обробки подій у розподілених системах. На відміну від спрощених підходів, запропонована модель одночасно враховує вплив повторної доставки повідомлень, витрат на ідемпотентну обробку, затримок буферизації, що виникають при порушенні порядку подій, використання резервних каналів передачі та конкуренції за спільні ресурси системи. Крім того, у моделі враховано нерівномірність розподілу подій між ключами, що дозволяє оцінювати ефективний рівень паралелізму у випадку наявності «гарячих» потоків подій.

Отриманий аналітичний вираз для пропускної здатності системи встановлює функціональну залежність між рівнем паралелізму та накладними витратами, які виникають у процесі впорядкованої обробки подій. Це дозволяє не лише оцінювати реальну продуктивність системи, але й визначати межу її масштабованості. Показано, що збільшення кількості паралельних обробників не завжди призводить до пропорційного зростання продуктивності, оскільки накладні витрати, пов'язані із синхронізацією та впорядкуванням подій, зростають разом із рівнем паралелізму.

Запропонована модель може використовуватися для прогнозування поведінки системи при зміні конфігурації обробників, оцінювання ефективності використання паралельних ресурсів та визначення оптимального рівня паралелізму, який забезпечує максимальну пропускну здатність за умов збереження стійкості системи. Отримані результати створюють аналітичне підґрунтя для подальшого експериментального дослідження продуктивності подієво-орієнтованих мікросервісних архітектур та оптимізації параметрів їх функціонування.

#### Перелік посилань

1. McGlohon N., Carothers C. Toward Unbiased Deterministic Total Orderings of Parallel Simulations with Simultaneous Events. Proceedings of the Winter Simulation Conference, 2021. DOI: 10.1109/WSC52266.2021.9715459.
2. Bachan J., Ye J. C., Jiang X. et al. Devastator: A Scalable Parallel Discrete Event Simulation Framework for Modern C++. Proceedings of the ACM, 2024. DOI: 10.1145/3615979.3656061.
3. Bhupatiraju R. V. Event-Driven Architecture for Payment Failover and Redundancy: A Framework for High-Availability Financial Transaction Processing. European Modern Studies Journal, 2025. DOI: 10.59573/emsj.9(5). 2025. 75.
4. Poudel P., Rai S. K., Guragain S. Ordered Scheduling in Control-flow Distributed Transactional Memory. Theoretical Computer Science, 2024. DOI: 10.1016/j.tcs.2024.114463.
5. Andelfinger P., Köster T., Uhrmacher A. M. The Window Racer Algorithm for Parallel Discrete Event Simulation. Proceedings of the ACM, 2023. DOI: 10.1145/3573900.3591115.
6. Poudel P., Rai S. K., Guragain S., Sharma G. Ordered Scheduling in Control-Flow Distributed Transactional Memory. Lecture Notes in Computer Science, Springer, 2023. DOI: 10.1007/978-3-031-24848-1\_5.
7. Batista E., Coelho P., Alchieri E. et al. FlexCast: An Efficient Atomic Multicast Protocol for Distributed Systems. Proceedings of the ACM, 2023. DOI: 10.1145/3590140.3629122.
8. Poudel P., Rai S., Sharma G. Processing Distributed Transactions in a Predefined Order. Proceedings of the ACM, 2021. DOI: 10.1145/3427796.3427819.
9. Eker A., Arafa Y., Badawy A. A. et al. Load-Aware Dynamic Time Synchronization in Parallel Discrete Event Simulation. Proceedings of the ACM, 2021. DOI: 10.1145/3437959.3459249.
10. McGlohon N., Carothers C. Deterministic Event Ordering Techniques for Parallel Discrete Event Simulation Systems. ACM Digital Library.
11. V. Zhebka, et al., Methods for predicting failures in a smart home, in: Digital Economy Concepts and Technologies Workshop, vol. 3665, 2024, 70–78.

Надійшла до редакції (Received): 27.02.2026

Прийнята до друку (Accepted): 17.03.2026

Опубліковано онлайн (Available online): 30.03.2026

<http://creativecommons.org/licenses/by/4.0/>

This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.