**DOI: 10.32347/2412-9933.2025.61.170-179**
УДК 004:728+827

**Соловей Ольга Леонідівна**
Кандидатка технічних наук, докторантка кафедри інформаційних технологій,
*https://orcid.org/0000-0001-8774-7243*
*Київський національний університет будівництва і архітектури, Київ*

# ПРОЦЕС ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ДЛЯ НАЛАШТУВАННЯ APACHE KAFKA-ПРОДЮСЕРА

***Анотація.*** *Предметом статті є методи підтримки прийняття рішень щодо визначення конфігурації Apache Kafka-кластера для забезпечення високої пропускної здатності повідомлень від Kafka-продюсера до Kafka-споживача, що є однією з ключових вимог для інформаційних технологій управління різнорідними даними проєктів міського будівництва. Метою роботи є розробка процесу підтримки прийняття рішень щодо визначення кількості розділів в Apache Kafka-топіку на основі аналізу чутливості в дискретній мережі Байєса. Процес пропонує рекомендації стосовно метрик продуктивності Apache Kafka-продюсера та їх інтервальних значень, які слід враховувати під час визначення кількості розділів. Для досягнення поставленої мети в роботі вирішені завдання: огляд і аналіз наявних формальних методів для визначення кількості розділів в Apache Kafka-топіку та визначення причин, через які ці методи можуть бути менш ефективними; розроблення структури мережі Байєса для включення в процес підтримки прийняття рішень щодо визначення кількості розділів у Apache Kafka-топіку; визначення математичних методів для обчислення параметрів мережі Байєса та кількісної оцінки чутливості апостеріорної ймовірності цільової функції до зміни значень параметрів мережі; оцінка запропонованого процесу на основі результатів перевірочних тестувань; аналіз діаграм метелика для ранжування параметрів мережі Байєса за силою впливу на цільову змінну. Для реалізації поставлених завдань використовувалися методи з теорій: ймовірності та статистики, системного аналізу, штучного інтелекту, інформаційних технологій. На основі отриманих результатів перевірочних тестувань запропонованого процесу доведено здатність використання процесу для формування рекомендацій щодо визначення розділів у Kafka-топіку. Крім того, запропонований процес визначає метрики продуктивності Apache Kafka-продюсера, значення яких слід постійно спостерігати для забезпечення продуктивності Kafka-продюсера після розгортання інформаційної технології, яка включає Kafka-кластер. Визначена на основі сформованих рекомендацій конфігурація Kafka-топіку зменшить ймовірність виникнення сценаріїв, коли ці конфігурації переглядаються після розгортання інформаційної системи, що призводить до ризику порушення порогу доступності даних.*

***Ключові слова: Kafka–топік; дискретна мережа Байєса; параметри мережі Байєса; аналіз чутливості; апостеріорна ймовірність***

## Вступ

Шаблон архітектури інформаційної технології для управління різнорідними даними проєктів міського будівництва складається з шарів, наведених на рис. 1, де “Шар прийому потокових даних» на основі платформи Apache Kafka забезпечує обробку потоків даних, які називаються подіями, у реальному часі з низькою затримкою [1].

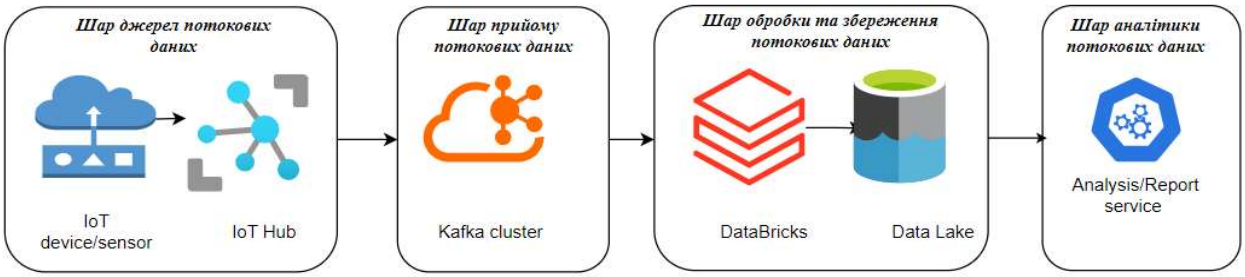


*Рисунок 1 – Шаблон архітектури інформаційної технології для управління даними проєктів міського будівництва*

 

Kafka-кластер складається із серверів, які часто називають брокерами, та сервісу ZooKeeper, який використовується Kafka для управління та координації серверів, включених в Kafka-кластер. Кожен сервер Kafka контролює топіки, до яких продюсери надсилають потокові дані, а споживачі підключаються для отримання даних. Kafka-топік може бути конфігурований з одним або багатьма розділами.

Коли Kafka-топік включає один розділ, процес потокової обробки даних складається з послідовних кроків: подія від продюсера потравляє в буфер сокета на сервері, звідки мережеві потоки перекладають її у спільну чергу запитів. Потоки вводу-виводу додають подію до журнала подій з унікальним ідентифікатором, який відповідає зміщенню події в журналі (offset). За замовченням, «сервер-лідер» підтверджує отримання події після завершення на всі визначені сервери. Після підтвердження від «сервера-лідера» подія потрапляє до черги для вихідних повідомлень, звідки мережевий потік відправляє її у буфер сокета для відправлення. У результаті продюсер отримує підтвердження, що подія доставлена, а споживач може її отримати. Отже, коли Kafka-топік включає один розділ, мережевий потік обробляє лише один запит від продюсера до підтвердження від «сервера-лідера», і лише після цього береться наступний запит [2 – 5].

Для оптимізації пропускної здатності Apache Kafka пропонує налаштовувати Kafka-топіки з більш ніж одним розділом, оскільки це забезпечує можливість продюсеру надсилати події паралельно, а сервер їх приймати.

При цьому кількість розділів має бути правильно визначена, виходячи з конкретних системних вимог. Надто мала кількість розділів може призвести до затримок для продюсера, оскільки він чекатиме, поки сервер прийме повідомлення. Це збільшує час між створенням повідомлення і його збереженням на сервері. Занадто велика кількість розділів на топік може призвести до додаткових витрат, пов'язаних з необхідністю створення n-сегментів (один сегмент на кожну секцію) та копіюванням даних для n-сегментів між серверами.

Висока пропускна здатність для повідомлень від Kafka-продюсера до Kafka-споживача є однією з ключових вимог для інформаційних технологій управління різнорідними даними проєктів міського будівництва [6]. Кількість розділів для Apache Kafka-топіку є одним із ключових параметрів конфігурації Kafka-кластера, який забезпечує високу продуктивність за умови, що кількість розділів визначено відповідно до потреб системи. Вибір кількості розділів для Apache Kafka-топіку на сьогодні базується на тестах кластера Kafka з різними параметрами конфігурації та обсягами потокових даних.

Через обмежену кількість формальних методів для визначення конфігурації Kafka-топіку часто виникають сценарії, коли ці конфігурації переглядаються після розгортання інформаційної системи. Такі перегляди потенційно можуть порушити поріг доступності даних, чого слід уникати [7].

## Мета статті

Метою статті є розробка процесу підтримки прийняття рішення щодо визначення кількості розділів на Apache Kafka-топіку на основі аналізу чутливості в дискретній мережі Байєса. Запропонований процес формуватиме рекомендації стосовно метрик продуктивності Apache Kafka продюсера та їх інтервальних значень, які слід враховувати під час визначення кількості розділів.

## Аналіз останніх досліджень і публікацій

У дослідженні [8] для визначення кількості розділів на Kafka-топіку було сформульовано задачу лінійного програмування з цільовою функцією для знаходження максимального значення кількості розділів $P \to max$ для Kafka-топіку з набором обмежень: обмеження $P - max(\frac{T}{T_p}, \frac{T}{T_c}, c) \geq 0$ – визначає, що кількість розділів $P$ має гарантувати, що пропускна здатність буде досягнута на одному розділі для продюсера та споживача ($Tp$ та $Tc$ відповідно) і має бути більшою за кількість споживачів $c$; обмеження $P \cdot r - b \cdot FH_{max} \leq 0$ – визначає, що кількість розділів $P$, помножена на коефіцієнт, який визначає кількість копій даних, які треба створити ($r$), має бути меншою або дорівнювати максимальній кількості відкритих обробників файлів $FH_{max}$, які можуть підтримуватися операційною системою на кожному сервері ($b$); обмеження $P \cdot r \cdot l_r - b \cdot L \leq 0$ – визначає, що кількість розділів $P$, помножена на коефіцієнт $r$ і коефіцієнт затримки, пов'язаний з часом, необхідним для копіювання ($lr$), має бути меншою або дорівнювати визначеному пороговому значенню затримки через копіювання $L$; обмеження $P \cdot u - b \cdot U \leq 0$ – визначає, що кількість розділів $P$, помножена на час недоступності $u$ під час збою сервера, має бути меншою або дорівнювати визначеному пороговому значенню недоступності $U$ на сервері ($b$).

На нашу думку, запропонована модель є ефективною, коли зв'язок між змінними Tp, Tc, b, $FH_{max}$, r, U, L та цільовими функціями P є лінійним, однак вона може бути менш ефективною там, де зв'язки нелінійні.

У роботі [9] було досліджено, як розмір буфера для передавання пакету даних впливає на

продуктивність Kafka-кластера, яка вимірювалася часом, що потрібен для переміщення повідомлення від виробника до споживача. Було запропоновано модель для прогнозування частоти затримок через різні мережеві умови з урахуванням розміру буфера (batch size) та обмежень на затримку даних (latency constraints).

У дослідженні [10] запропоновано модель для прогнозування продуктивності системи обміну повідомленнями Kafka на основі процесу Пуассона для наближення трьох розподілів: $PH_s$, $PH_f$ та $PH_r$, із відповідними параметрами $\lambda_s$, $\lambda_f$ та $\lambda_r$. $PH_s$ представляє закон розподілу часу, від моменту отримання сервером події від продюсера до моменту її збереження на диску; $PH_f$ позначає закон розподілу часу для споживчих, потрібний для зчитування подій; $PH_r$ – закон розподілу часу, який сервер використовує для копіювання подій.

На нашу думку, моделі, запропоновані у [9; 10], не враховують, як кількість розділів на Kafka-топіку може покращити продуктивність Kafka-продюсера.

## Виклад основного матеріалу

Запропонуємо процес для підтримки прийняття рішення щодо визначення кількості розділів на Apache Kafka-топіку, який включатиме п’ять підпроцесів (рис. 2). У цій роботі фокус буде зосереджено на параметрі «кількість розділів» Apache Kafka-топіку, який є одним із критичних параметрів для продуктивності Apache Kafka-продюсера. Для визначення процесу з рис. 2 опишемо підпроцеси 1 – 5.


*Рисунок 2 – Процес для підтримки прийняття рішення щодо визначення кількості розділів на Apache Kafka-топіку*

1. Визначення структури та параметрів мережі Байєса. Нехай множина $Y = \{y_j\}_{j=\overline{1,n}}$ визначає кількість розділів Apache Kafka-продюсера, де $n$ – максимальна кількість розділів; множина $X_{i=\overline{1,k}} = \{x_i\}$ описує метрики [11], за якими вимірюють продуктивність Apache Kafka-продюсера. Структура дискретної мережі Байєса, що відтворює зв’язок між X та Y, є ациклічний граф, де Y є вузлом, який залежить від змінних $x_i \in X$, що є батьківськими вузлами, незалежними один від одного, тобто вузли $x_i$ не мають прямих зв'язків між собою (рис. 3). Вплив батьківських вузлів з множини X на вузол Y виражається спільною ймовірністю за правилом Байєса: $P(y_1, y_2, ...y_n) = \prod_{i=1}^{n} y_i | X$. Отже, вузли $x_i$ є випадковими змінними мережі на рис. 3, значення яких визначаються параметрами $\theta_X$, які представляють умовні ймовірності для кожного вузла. Цільовою змінною мережі є вузол Y, значення якого визначаються параметрами $\theta_y$ за умови відомих параметрів $\theta_X$. Отже, для визначення мережі Байєса необхідно визначити значення параметрів $\theta_X$, $\theta_y$.

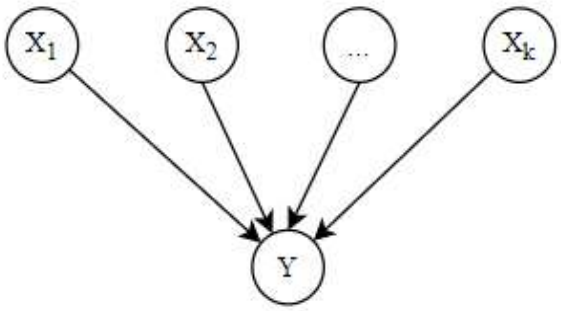


*Рисунок 3 – Схема мережі Байєса для підтримки прийняття рішення щодо оптимальної кількості розділів Apache-Kafka продюсера*

Параметри $\theta_X$, $\theta_y$ визначають апріорні умовні ймовірності для кожного батьківського вузла $x_i$ (1) та цільової змінної $Y$:

$$\theta_X = \left\{ \theta_{x_{11}}, \ldots, \theta_{x_{1N[s_{x_1}]}}, \theta_{x_{21}}, \ldots, \ldots, \theta_{x_{2N[s_{x_2}]}}, \ldots, \theta_{x_{k1}}, \ldots, \theta_{x_{kN[s_{x_k}]}} \right\}, \quad (1)$$

$$\theta_y = \left\{ \theta_{y_1|x_{11}}, \ldots, \theta_{y_1|x_{1N[s_{x_1}]}}, \theta_{y_2|x_{11}}, \ldots, \ldots, \theta_{y_2|x_{1N[s_{x_1}]}}, \theta_{y_m|x_{11}}, \ldots, \theta_{y_m|x_{1N[s_{x_1}]}} \right\}, \quad (2)$$

де $N[s_{x_1}], N[s_{x_2}], N[s_{x_k}]$ – кількість станів S, які визначені умовними ймовірностями для змінних $x_1$, $x_2, \ldots, x_k$.

За умови наявності набору даних $D$, параметри $\theta_X, \theta_y$ мережі Байєса, можна обчислити за функцією правдоподібності $L(\theta|D)$, яка виражає спільний розподіл імовірностей спостережуваних даних $D$ за умови, що вони були згенеровані моделлю з параметрами $\theta$. Для мережі на рис. 3 функція правдоподібності $L(\theta|D)$ обчислюється відповідно до виразу:

$$L(\theta|D)=\prod_{t=1}^{d}P\left(X(t),Y(t)|\theta\right)=$$

$$=\prod_{t=1}^{d}P\left(X(t)\,|\,\theta\right)P(Y(t)|X(t),\theta)=$$

$$=\theta_{x_{11}}^{N[x_{11}]}\cdot\ldots\cdot\theta_{x_{1N[s_{x_1}]}}^{N\left[x_{1N[s_{x_1}]}\right]}\cdot\ldots\cdot\theta_{x_{k1}}^{N[x_{k1}]}\cdot\ldots\cdot \quad (3)$$

$$\cdot\ldots\cdot\theta_{x_{kN[s_{x_k}]}}^{N\left[x_{kN[s_{x_k}]}\right]}\cdot\ldots\cdot\theta_{y_1|x_{11}}^{N[y_1|x_{11}]}\cdot\ldots\cdot\theta_{y_1|x_{1N[s_{x_1}]}}^{N\left[y_1|x_{1N[s_{x_1}]}\right]}\cdot\ldots\cdot$$

$$\cdot\ldots\cdot\theta_{y_m|x_{11}}^{N[y_m|x_{11}]}\cdot\ldots\cdot\theta_{y_m|x_{1N[s_{x_1}]}}^{N\left[y_m|x_{1N[s_{x_1}]}\right]},$$

де $N\left[y_1|x_{1N[s_{x_1}]}\right], N\left[y_m|x_{1N[s_{x_1}]}\right]$ – кількість разів параметр $\theta_{y_1|x_{1N[s_{x_1}]}}$ зустрічається у виразі (3).

Задача визначення параметрів мережі $\theta^*$ полягає у відшуканні розв'язку рівнянь (4), (5) у частинних похідних:

$$\theta_X^*=\left\{\frac{\partial L(\theta|D)}{\partial\theta_{x_{11}}}=0,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{x_{1N[s_{x_1}]}}}=\right.$$

$$=0,\frac{\partial L(\theta|D)}{\partial\theta_{x_{21}}}=0\ ,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{x_{2N[s_{x_2}]}}}= \quad (4)$$

$$\left.=0,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{x_{k1}}}=0,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{x_{kN[s_{x_k}]}}}=0\right\}.$$

$$\theta_Y^*=\left\{\frac{\partial L(\theta|D)}{\partial\theta_{y_1|x_{1N[s_{x_1}]}}}=0,\ \frac{\partial L(\theta|D)}{\partial\theta_{y_2|x_{11}}}=\right.$$

$$=0,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{y_2|x_{1N[s_{x_1}]}}}=0,\frac{\partial L(\theta|D)}{\partial\theta_{y_m|x_{11}}}= \quad (5)$$

$$\left.=0,\ldots,\frac{\partial L(\theta|D)}{\partial\theta_{y_m|x_{1N[s_{x_1}]}}}=0\right\}.$$

Отже, через відсутність знань щодо апріорних ймовірностей змінних $x_1$, $x_2$,…, $x_k$ необхідно створити набір даних D, на основі якого виконати обчислення параметрів $\theta_X, \theta_y$.

2. Збір та підготовка даних. Набір даних D отримаємо в результаті виконання сценарію для тестування системи з різними навантаженнями та конфігураціями продюсера.

Значення параметрів для конфігурації Apache Kafka-сервера:

- Унікальний ідентифікатор сервера (broker.id) – 0; 1; 2.
- Адреса, яку прослуховує сервер сокетів (listeners) – "localhost: 9092; localhost:9093; localhost:9094".
- Кількість мережевих потоків (num.network.threads) – 3.
- Кількість потоків вводу-виводу IO (num.io.threads) – 8.
- Розмір буфера сокета для прийняття потокових даних (socket.receive.buffer.bytes) – 102400.
- Розмір буфера сокета для надсилання потокових даних (socket.send.buffer.bytes) – 102400.
- Максимальний розмір запиту, який прийме сокет сервера (socket.request.max.bytes) – 104857600.
- Шлях до журналів логування (log.dirs) – localhost:2181.
- Адреса для встановлення з'єднання із сервісом ZooKeeper (zookeeper.connect) – //kafka_2.13-3.8.1/tmp/kafka-logs-0; //kafka_2.13-3.8.1/tmp/kafka-logs-1; //kafka_2.13-3.8.1/tmp/kafka-logs-2.
- Кількість мережевих потоків (offsets.topic.replication.factor) – 3.
- Кількість розділів на топіку (num.partitions) – Змінюється відповідно зі сценарієм для тестування від 1 до 9.

Значення параметрі для конфігурації Apache Kafka-продюсера:

- Список пар хостів і портів для встановлення з'єднання із серверами Kafka (BOOTSTRAP_SERVERS_CONFIG) – "localhost: 9092; localhost:9093; localhost:9094".
- Налаштування конфігурації для серіалізації ключа та значення повідомлення (KEY_SERIALIZER_CLASS_CONFIG, VALUE_SERIALIZER_CLASS_CONFIG) – org.apache.kafka.common.serialization.StringSerializer.
- Час (у мілісекундах), протягом якого продюсер чекає, перш ніж надіслати пакет повідомлень Kafka серверу (LINGER_MS_CONFIG) – Випадкове число з інтервалу [0..100].
- Рівень підтвердження, який виробник вимагає від брокерів Kafka для визнання запису успішним (ACKS_CONFIG) – Випадково обране значення "1" або "all".

Сценарій для тестування визначимо послідовністю кроків:

*Крок 1.* Кожну секунду протягом 50 секунд надсилатимемо повідомлення розміром 387 байт на Kafka-топік з одним розділом.

*Крок 2.* Кожної секунди фіксуємо в журнал логування значення метрик продуктивності Kafka-продюсера [11].

Повторюємо кроки 1 – 2 ще вісім разів, щоразу збільшуючи кількість розділів на Kafka-топіку на один.

У результаті виконаного сценарію для тестування отримаємо набір даних із значеннями

метрик продуктивності Kafka-продюсера і відповідним фактичним значенням кількості розділів Kafka-топіку. Загальний опис отриманого набору даних:

- Кількість спостережень – 451.
- Кількість змінних – 27.
- Цільова змінна – «Кількість розділів».
- Тип значень змінних – Безперервні та дискретні.
- Кількість змінних з однаковим значеннями – 12.
- Кількість змінних, значення яких корелюють – 5.

Для обчислення параметрів мережі Байєса треба вилучити змінні з однаковими значеннями, а також змінні, між якими визначена кореляція. Щоб описати змінні набору $D$ через множину станів $S = [s_{x_1}, \dots, s_{x_k}]$ потрібно виконати дискретизацію для безперервних значень змінних [12].

На рис. 4 представлено створену дискретну мережу Байєса зі змінними визначеними на основі набору D та їх можливими станами, а саме до змінних мережі належать: Node1 – linger.ms – час (у мілісекундах), протягом якого продюсер чекає, перш ніж надіслати пакет повідомлень Kafka серверу; Node2 – batch-size-avg – середня кількість байтів, надісланих на розділ за один запит; Node3 – batch-size-max – максимальна кількість байтів, надісланих на розділ за один запит; Node4 – record-queue-time-avg – середній час (у мілісекундах) перебування пакетів з даними у буфері надсилання; Node6 – record-queue-time-max – максимальний час (у мілісекундах) перебування пакетів з даними у буфері надсилання; Node7 – record-send-rate – середня кількість подій, надісланих за 1 секунду; Node8 – record-send-total – загальна кількість подій надісланих Kafka-продюсером на сервер; Node9 – record-size-avg – середній розмір події, яка надсилається за один запит в байтах; Node11 – records-per-request-avg – середня кількість подій, надісланих за один запит; Node12 – request-latency-avg – середній час затримки запитів (у мілісекундах). Цільова змінна «Node 5 – Partitions» визначена станами P1-P9, які відповідають кількості розділів на Kafka-топіку від 1-го до 9-ти. Ймовірності, що значення змінних будуть відповідати станам, визначимо після обчислення параметрів мережі.

3. Обчислення параметрів мережі на основі створеного набору даних D виконаємо відповідно до EM-алгоритму [13].

4. Аналіз впливу змінних станів мережі на цільову змінну. Апостеріорна ймовірність $P(Y|X)(p)$ цільової змінної Yчерез зміну параметрів $\theta_X^*$ є часткою двох лінійних функцій від параметра p, який є ймовірністю, що параметри мережі приймуть певні значення:

$$P(Y|X)(p) = \frac{a \cdot p + b}{c \cdot p + 1}, \qquad (6)$$

де $a$, $c$ – кутові коефіцієнти рівняння прямих; $b$ – зсув по осі OY.

Мірою чутливості цільової змінної Y для мережі на рис. 4 до зміни значень $x_i \in X$ є часткова похідна функції (6) за змінною $p$:

$$Dr = \frac{\partial(P(Y|X)(p))}{\partial p} = \frac{a - bc}{(c \cdot p + 1)^2}. \qquad (7)$$

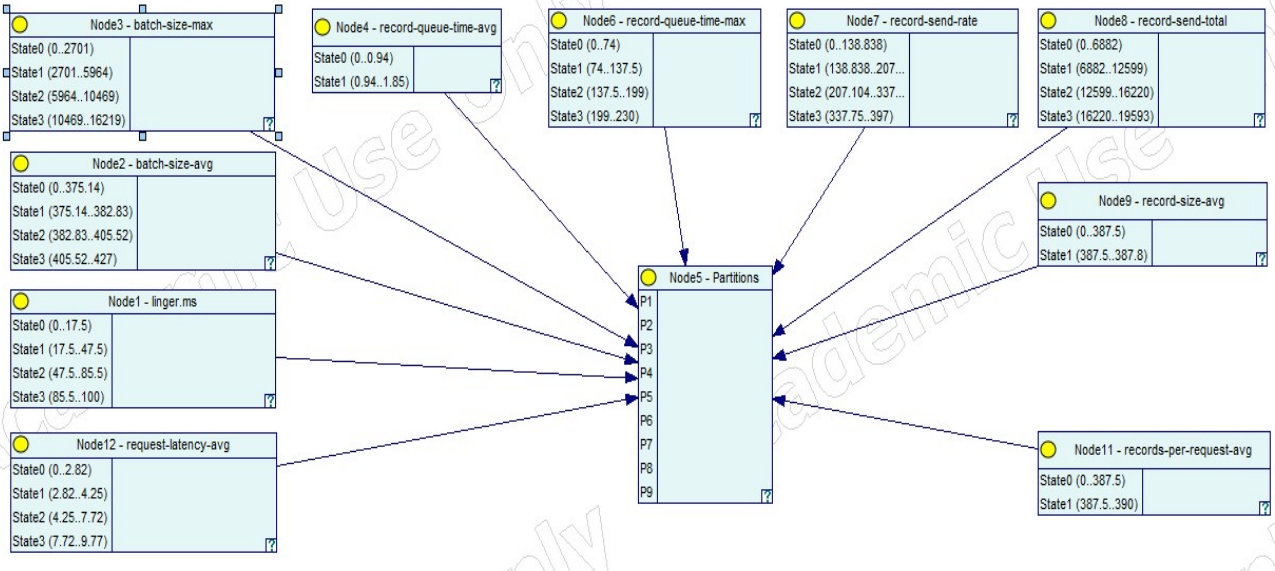


*Рисунок 4 – Структура мережі Байєса для підтримки прийняття рішення щодо оптимальної кількості розділів Apache Kafka-продюсера*

Сила випливу $I_{x_i}$ від зміни значень параметрів змінних $x_i \in X$ на апостеріорну ймовірність $P(Y|X)(p)$ цільової змінної Y визначається добутком ширини інтервалу $W_{x_i}$ зміни значень параметрів змінних $x_i \in X$ на модуль похідної $Dr$ і обчислюється за виразом (8). Відносне значення сили впливу ($Ir_{x_i}$, *%)* дорівнює відношенню сили випливу $I_{xi}$ на середнє значення середини інтервалу, який визначає зміну апостеріорної ймовірності і обчислюється за виразом (9).

$$I_{x_i} = W_{x_i} \cdot Dr, \tag{8}$$

$$Ir_{x_i} = \frac{I_{x_i}}{\left(p_{1_{y_i}} + p_{2_{y_i}}\right)/2} \cdot 100\%, \tag{9}$$

де $p_1$, $p_2$ – границі інтервалу зміни апостеріорної ймовірності (7), які обчислюються за умови зміни параметра $p$ у формулі (6) по всьому діапазону, тобто: якщо $p=0$, тоді $p_1=b$; якщо $p=1$, тоді $p_2=(a+b)/(c+1)$.

Змінні $x_i \in X$ мережі Байєса, для яких значення сила випливу $I_{x_i}$ та відносне значення сила випливу $Ir_{x_i}$, %, є найбільшими, визначимо як впливовими на рішення щодо кількості розділів Apache Kafka-продюсера.

Виконаємо перевірочне тестування запропонованого процесу і проведемо аналіз отриманих результатів.

## Аналіз результатів дослідження

На рис. 5 у відсотках представлено обчислені ймовірності того, що значення змінних мережі Байєса відповідатимуть визначеним станам. Наприклад, змінна Node2-batch-size-avg з ймовірністю 88% матиме значення з інтервалу [0..375.14].

Для цільової змінної визначено апостеріорні ймовірності 48% для кількості розділів один та два на Kafka-топіку.

На рис. 6 представлено мережу Байєса з оцінкою чутливості цільової змінної «Partitions» до зміни станів “батьківських” вузлів. Червоним кольором позначені змінні, стан яких має найбільший вплив на значення цільової змінної, а саме: Node2 – batch-size-av, Node3 – batch-size-max, Node4 – record-queue-time-avg, Node6 – record-queue-time-max, Node7 – record-send-rate, Node8 – record-send-total. Блідо-червоним кольором позначена змінна Node1 – linger.ms, вплив якої є слабкішим. Змінні, позначені сірим кольором, мають вплив тільки через взаємодію зі змінними, які позначені червоним кольором.

Для формування рекомендацій щодо метрик продуктивності Apache Kafka-продюсера, які слід моніторити для визначення кількості розділів Apache Kafka-продюсера, виконаємо кількісний аналіз чутливості за формулами (6), (7) на основі діаграми «метелика» для конфігурацій kafka-топіку з одним та двома розділами. Оскільки отримані апостеріорні ймовірності для інших конфігурацій kafka-топіку склали менш ніж 2%, виходячи з наявного набору даних, оцінка впливу на інші конфігурацію не є доцільною.

На рис. 6 представлені діаграми “метелика”, які візуалізують вплив змінних мережі Байєса на цільову змінну «Node – Partitions», коли її стан відповідає P1 – один розділ на kafka-топіку та P2 – два розділи на kafka-топіку відповідно. Довжина горизонтальних смуг вказує на ширину інтервалу зміни значень змінної «Node5 – Partitions» при зміні стану змінної, яка позначена над горизонтальною смугою або одночасній зміні станів багатьох змінних. Вертикальна лінія вказує на середнє значення параметрів $\theta_{y_1}, \theta_{y_2}$ за умови відсутності змін у параметрах θx.

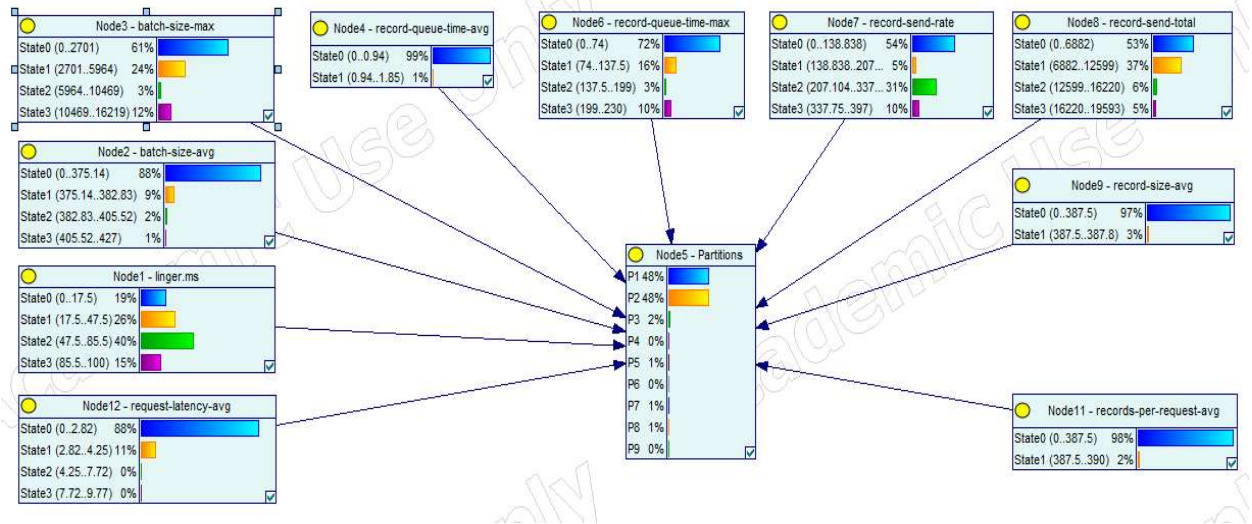


*Рисунок 5 – Мережа Байєса з визначеними ймовірностями для змінних*

Зелений колір справа від вертикальної лінії на горизонтальних смугах вказує на необхідність збільшити кількість розділів на kafka -топіку з одного до двох або з двох до трьох, якщо змінні, вказані над смугами, приймуть значення з визначених інтервалів. Червоний колір справа від вертикальної лінії на горизонтальних смугах вказує на необхідність зменшити кількість розділів на kafka-топіку, це означає, що в системі є певний запас для продуктивної роботи Kafka-продюсера з фактичною кількістю розділів на kafka-топіку, для конфігурації з одним розділом – зміни не потрібні. Горизонтальні смуги відсортовані відповідно до їх чутливості до змін у параметрах $\theta x$ . Кількісний аналіз діаграм "метелика" з рис. 7 наведено в табл. 1, 2.

*Отримані значення* сили випливу $I_{x_i}$ та відносної сили випливу $Ir_{x_i}$ , %, разом зі знаком похідної в табл. 1 вказують, що кількість розділів на kafka-топіку треба збільшити, якщо змінні зі списку 1 одночасно будуть визначатись вказаними в списку станами. Це твердження також справедливе для змінних зі списку 2. Вплив від змінних зі списку 3 та списку 3* є взаємно протилежним для станів P1 та P2, тому не може бути використаний для формування рекомендацій.

*Отримані значення* сили випливу $I_{x_i}$ та відносної сили випливу $Ir_{x_i}$ , %, разом зі знаком похідної в табл. 2 вказують на необхідність виконання корекцій: кількість розділів на kafka-топіку треба зменшити до одного, якщо змінні Node2 – batch-size-av, Node3 – batch-size-max, Node4 – record-queue-time-avg, Node6 – record-queue-time-max, Node7 – record-send-rate, Node8 – record-send-total приймуть вказані стани. Вплив від змінних зі списку 4 та списку 4* є взаємно протилежним для станів P1 та P2, тому не може бути використаний для формування рекомендацій.

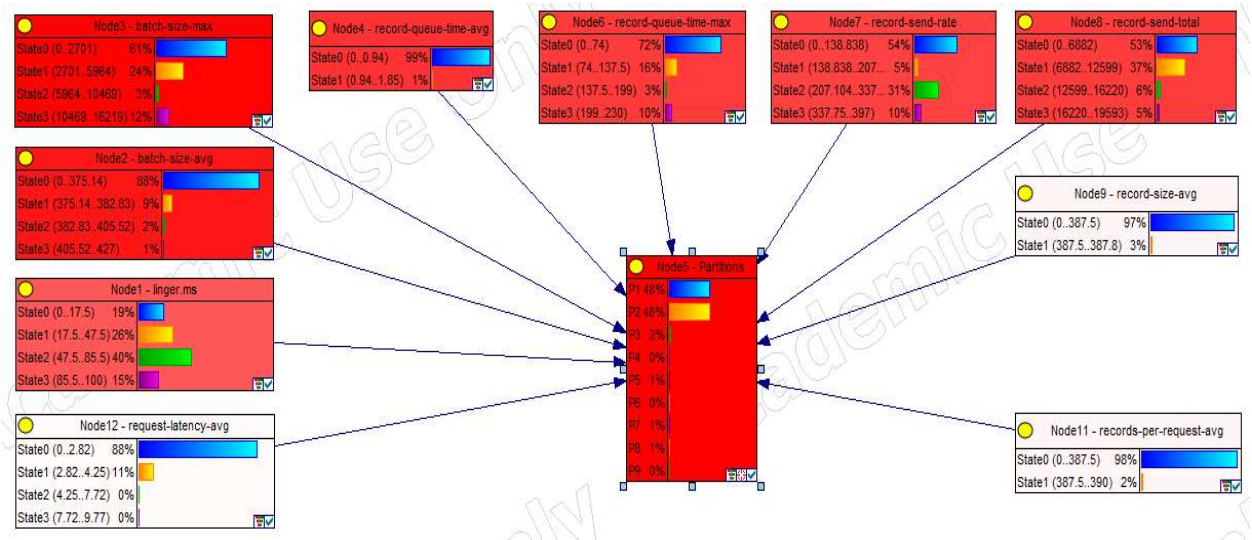


*Рисунок 6 – Мережа Байєса з визначеним впливом станів змінних на цільову змінну «Partitions»*


*Рисунок 7 – Діаграми «метелика» для візуального відображення чутливості цільової змінної «Node 5 – Partitions», коли її стан відповідає P1 – один розділ на kafka-топіку та P2 – два розділи на kafka-топіку відповідно*

*Таблиця 1 –* ***Кількісний аналіз діаграм "метелика" для цільової змінної «Node 5 – Partitions», коли її стан відповідає P1 – один розділ на kafka-топіку***

| Назва та стан змінної | Чутливість цільової змінної, *Dr* | Ширина інтервалу зміни значень змінних, $W_{x_i}$ | Ліва границя інтервалу чутливості цільової змінної, $p_1$ | Права границя інтервалу чутливості цільової змінної, $p_2$ | Середнє значення інтервалу, $[p_1..p_2]$ | Сила випливу, $I_{x_i}$ | Відносна сила випливу, $Ir_{x_i}$,% |
|---|---|---|---|---|---|---|---|
| Node2=State0 (0..375.14) | -0.024 | 0.177 | 0.475 | 0.479 | 0.477 | 0.004 | 0.907 |
| Список1 | 0.037 | 0.078 | 0.476 | 0.479 | 0.477 | 0.003 | 0.600 |
| Список2 | 0.026 | 0.099 | 0.476 | 0.478 | 0.477 | 0.003 | 0.533 |
| Node3=State0 (0..2701) | -0.018 | 0.122 | 0.476 | 0.478 | 0.477 | 0.002 | 0.468 |
| Список3 | 0.021 | 0.099 | 0.476 | 0.478 | 0.477 | 0.002 | 0.431 |
| Список3* | -0.021 | 0.099 | 0.476 | 0.478 | 0.477 | 0.002 | 0.431 |
| Node8=State0 (0..6882) | -0.019 | 0.105 | 0.476 | 0.478 | 0.477 | 0.002 | 0.410 |
| Node6=State0 (0..74) | -0.014 | 0.144 | 0.476 | 0.478 | 0.477 | 0.002 | 0.408 |
| Node7=State0 (0..138.838) | -0.018 | 0.108 | 0.476 | 0.478 | 0.477 | 0.002 | 0.404 |

Список1: Node5=P1 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State0 (0..138.838), Node8=State0 (0..6882), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

Список2: Node5=P1 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State0 (0..138.838), Node8=State1 (6882..12599), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

Список3:Node5=P1 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State2 (207.104..337.75), Node8=State0 (0..6882), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

Список3*:Node5=P2 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State2 (207.104..337.75), Node8=State0 (0..6882), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

Список4: Node5=P2 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State0 (0..138.838), Node8=State1 (6882..12599), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

Список4*: Node5=P2 | Node1=State2 (47.5..85.5), Node2=State0 (0..375.14), Node3=State0 (0..2701), Node4=State0 (0..0.94), Node6=State0 (0..74), Node7=State0 (0..138.838), Node8=State1 (6882..12599), Node9=State0 (0..387.5), Node11=State0 (0..387.5), Node12=State0 (0..2.82)

*Таблиця 2 –* ***Кількісний аналіз діаграм "метелика" для цільової змінної «Node 5 – Partitions», коли її стан відповідає P2 – два розділи на kafka-топіку***

| Назва та стан змінної | Чутливість цільової змінної, *Dr* | Ширина інтервалу зміни значень змінних, $W_{x_i}$ | Ліва границя інтервалу чутливості цільової змінної, $p_1$ | Права границя інтервалу чутливості цільової змінної, $p_2$ | Середнє значення інтервалу, $[p_1..p_2]$ | Сила випливу, $I_{x_i}$ | Відносна сила випливу $Ir_{x_i}$,% |
|---|---|---|---|---|---|---|---|
| Node2=State0 | -0.025 | 0.177 | 0.479 | 0.474 | 0.476 | 0.004 | 0.945 |
| Node3=State0 (0..2701) | -0.032 | 0.122 | 0.478 | 0.474 | 0.476 | 0.004 | 0.827 |
| Список1 | 0.037 | 0.078 | 0.475 | 0.478 | 0.476 | 0.003 | 0.601 |
| Node6=State0 (0..74) | -0.019 | 0.144 | 0.478 | 0.475 | 0.476 | 0.003 | 0.571 |
| Список4 | 0.026 | 0.099 | 0.475 | 0.478 | 0.476 | 0.003 | 0.534 |
| Список4* | -0.026 | 0.099 | 0.475 | 0.478 | 0.476 | 0.003 | 0.534 |
| Node8=State0 (0..6882) | -0.018 | 0.105 | 0.477 | 0.475 | 0.476 | 0.002 | 0.399 |
| Node7=State0 (0..138.838) | -0.017 | 0.108 | 0.477 | 0.475 | 0.476 | 0.002 | 0.398 |

Найбільший вплив на значення цільової змінної «Node 5 – Partitions» мають змінні Node6 – record-queue-time-max, Node7 – record-send-rate, Node8 – record-send-total. У разі зниження продуктивності Kafka-продюсера значення цих метрик належатимуть до інтервалів, що відрізняються від початкових, а це означає, що виникне необхідність у коригуванні параметрів Kafka-продюсера. Отже, за визначеними змінними слід постійно спостерігати для забезпечення продуктивності Kafka-продюсера після розгортання інформаційної технології, яка включає Kafka-кластер.

## Висновок

У пропонованій статті поставлено і розв'язано задачу розробки процесу підтримки прийняття рішень щодо визначення кількості розділів в Apache Kafka-топіку на основі аналізу чутливості в дискретній мережі Байєса. Процес пропонує рекомендації щодо метрик продуктивності Apache Kafka-продюсера та їх інтервальних значень, які слід враховувати під час визначення кількості розділів. Крім того, запропонований процес визначає метрики продуктивності Apache Kafka-продюсера, значення яких слід постійно спостерігати для забезпечення продуктивності Kafka-продюсера після розгортання інформаційної технології, яка включає Kafka-кластер.

Визначена на основі сформованих рекомендацій конфігурація Kafka-топіку зменшить ймовірність виникнення сценаріїв, коли ці конфігурації переглядаються після розгортання інформаційної системи, що призводить до ризику порушення порогу доступності даних.

## Список літератури

1. Babun, L., Denney, K., Celik, Z. B., McDaniel, P., & Uluagac, A. S. (2021). A survey on IoT platforms: Communication, security, and privacy perspectives. Computer Networks, 192, 108040.
2. Apache Kafka Available online: https://kafka.apache.org/
3. Kafka Producer. Available online: https://docs.confluent.io/platform/current/clients/producer.html
4. Kafka Consumer. Available online: https://docs.confluent.io/platform/current/clients/consumer.html
5. Apache Kafka Supports 200K Partitions Per Cluster. Available online: https://www.confluent.io/blog/apache-kafka-supports-200k-partitions-per-cluster/
6. Turner, C. J., Oyekan, J., Stergioulas, L., & Griffin, D. (2020). Utilizing industry 4.0 on the construction site: Challenges and opportunities. IEEE Transactions on Industrial Informatics, 17 (2), 746–756.
7. Mehmood, E., & Anees, T. (2020). Challenges and solutions for processing real-time big data stream: a systematic literature review. IEEE Access, 8, 119123–119143.
8. Raptis, T. P., & Passarella, A. (2022, July). On efficiently partitioning a topic in apache kafka. In 2022 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 1–8). IEEE.
9. Wu, H., Shang, Z., Peng, G., & Wolter, K. (2020, October). A reactive batching strategy of apache kafka for reliable stream processing in real-time. *In 2020 IEEE 31st International Symposium on Software Reliability Engineering* (ISSRE) (pp. 207–217). IEEE.
10. Wu, H., Shang, Z., & Wolter, K. (2019, August). Performance prediction for the apache kafka messaging system. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; *IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems* (HPCC/SmartCity/DSS) (pp. 154–161). IEEE.
11. Kafka producer metrics. Available online: https://kafka.apache.org/32/generated/producer_metrics.html.
12. Solovei, O. (2023, November). *Analysis of a fixed-width binning method. In 2023 2nd International Conference on Innovative Solutions in Software Engineering* (ICISSE) (p. 49).
13. Wasserkrug, S., Marinescu, R., Zeltyn, S., Shindin, E., & Feldman, Y. A. (2021, May). Learning the parameters of bayesian networks from uncertain data. *In Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 13, pp. 12190–12197).

*Стаття надійшла до редколегії 24.01.2025*

**Solovei Olha**
Doctoral student of the department of information technologies of design and applied mathematics, associate professor of the department of information technologies of design and applied mathematics,
*https://orcid.org/0000-0001-8774-7243*
*Kyiv National University of Construction and Architecture, Kyiv*

**DECISION SUPPORT PROCESS TO DEFINE APACHE KAFKA PRODUCER CONFIGURATION**

***Abstract.*** *The subject of the article is decision support methods for determining the configuration of an Apache Kafka cluster that ensures high throughput for messages from the Kafka producer to the Kafka consumer, which is one of the key requirements for information technologies managing heterogeneous data in urban construction projects. The purpose of the work is to develop a decision support process to determine the number of partitions in an Apache Kafka topic based on sensitivity analysis in a discrete*

*Bayesian network. The process offers recommendations regarding the performance metrics of the Apache Kafka producer and their interval values that should be considered when determining the number of partitions. To achieve the stated goal, the following tasks were solved: reviewing and analyzing existing formal methods for determining the number of partitions in an Apache Kafka topic and identifying the reasons why these methods may be less effective; defining mathematical methods and developing the structure of a Bayesian network for inclusion in the decision support process for determining the number of partitions in an Apache Kafka topic; evaluating the proposed process based on the results of verification tests. To accomplish these tasks, methods from probability and statistics, systems analysis, artificial intelligence, and information technology theories were employed. Based on the results of verification tests of the proposed process, its ability to generate recommendations for determining the number of partitions in a Kafka topic has been demonstrated. Furthermore, the proposed process identifies performance metrics of an Apache Kafka producer, the values of which should be continuously monitored to maintain the producer's performance after deploying an information technology system that includes a Kafka cluster. The configuration of a Kafka topic, determined based on the formulated recommendations, will reduce the likelihood of scenarios where these configurations are revised after deploying the information system, which could lead to the risk of breaching the data availability threshold.*

***Keywords: Kafka topic; discrete Bayesian network; parameters of a Bayesian network; sensitivity analysis; posterior probability***

**References**

1. Babun, L., Denney, K., Celik, Z. B., McDaniel, P., & Uluagac, A. S. (2021). A survey on IoT platforms: Communication, security, and privacy perspectives. Computer Networks, 192, 108040.
2. Apache Kafka Available online: https://kafka.apache.org/
3. Kafka Producer. Available online: https://docs.confluent.io/platform/current/clients/producer.html
4. Kafka Consumer. Available online: https://docs.confluent.io/platform/current/clients/consumer.html
5. Apache Kafka Supports 200K Partitions Per Cluster. Available online: https://www.confluent.io/blog/apache-kafka-supports-200k-partitions-per-cluster/
6. Turner, C. J., Oyekan, J., Stergioulas, L., & Griffin, D. (2020). Utilizing industry 4.0 on the construction site: Challenges and opportunities. *IEEE Transactions on Industrial Informatics*, 17 (2), 746–756.
7. Mehmood, E., & Anees, T. (2020). Challenges and solutions for processing real-time big data stream: a systematic literature review. *IEEE Access*, 8, 119123–119143.
8. Raptis, T. P., & Passarella, A. (2022, July). On efficiently partitioning a topic in apache kafka. *In 2022 International Conference on Computer, Information and Telecommunication Systems* (CITS), (pp. 1–8). IEEE.
9. Wu, H., Shang, Z., Peng, G., & Wolter, K. (2020, October). A reactive batching strategy of apache kafka for reliable stream processing in real-time. In 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), pp. 207–217. IEEE.
10. Wu, H., Shang, Z., & Wolter, K. (2019, August). Performance prediction for the apache kafka messaging system. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; *IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems* (HPCC/SmartCity/DSS) (pp. 154–161). IEEE.
11. Kafka producer metrics. Available online: https://kafka.apache.org/32/generated/producer_metrics.html.
12. Solovei, O. (2023, November). Analysis of a fixed-width binning method. In 2023 2nd International Conference on Innovative Solutions in Software Engineering (ICISSE) (p. 49).
13. Wasserkrug, S., Marinescu, R., Zeltyn, S., Shindin, E., & Feldman, Y. A. (2021, May). Learning the parameters of bayesian networks from uncertain data. *In Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 13, pp. 12190–12197).

---

**Посиланя на публікацію**

APA Solovei, Olha, (2025). Decision support process to define Apache Kafka producer configuration. *Management of Development of Complex Systems*, 61, 170–179, dx.doi.org\10.32347/2412-9933.2025.61.170-179.

ДСТУ Соловей О. Л. Процес підтримки прийняття рішення для налаштування Apache Kafka-продюсера. *Управління розвитком складних систем.* Київ, 2025. № 61. С. 170 – 179, dx.doi.org\10.32347/2412-9933.2025.61.170-179.