

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт

Лабораторна робота № 4

з курсу “Моделювання та аналіз даних в інформаційних системах”

Виконала:
студентка групи ШІ-32
Якимів Анастасія

Викладач:
Пелецишин О.П.

Львів 2024р.

Тема: Симуляція різних типів СМО за допомогою SimPy

Мета: Дослідження різнотипних СМО за допомогою SimPy

Завдання:

Індивідуальні завдання (обрати собі задачу з переліку – в групі задачі не повинні повторюватись). Для кожної задачі підібрати той варіант СМО (див. Варіанти СМО), який її характеризує

Обрати варіант СМО:

1. СМО з втратами та без очікування
2. Багатоканальні СМО з втратами та без очікування
3. СМО з очікуванням (багатоканальні пристрої)
4. СМО з повторними викликами та обмеженою кількістю місць в черзі

Залежно від варіанту, потрібно:

- Побудувати та записати модель СМО у вигляді блок-схеми.
- Реалізувати модель в Python, використовуючи SimPy.
- Виконати імітаційне моделювання. Включити у звіт код програми, результати моделювання та короткий аналіз отриманих результатів.
- Виконати покрокове виконання моделі та додати знімки з підтвердженням правильності алгоритму.

Вихідні дані залежать від прізвища

Хід роботи

Для даної лабораторної роботи я обрала задачу - система пошуку збоїв у банківській мережі.

Задачу "Система пошуку збоїв у банківській мережі" можна віднести до СМО з повторними викликами та очікуванням в черзі з обмеженою кількістю місць в черзі (варіант 4). Такий варіант підходить через специфіку задачі:

- Повторні виклики — у разі виявлення збою система може повторювати перевірки або запити, щоб підтвердити чи вирішити проблему.
- Черга з обмеженою кількістю місць — мережа може обробляти обмежену кількість запитів одночасно, і якщо черга переповнюється, деякі запити можуть втрачатися, або користувачам доведеться повторно намагатися підключитися.

Це дозволить змодельовати поведінку системи, в якій відбувається моніторинг мережі, з урахуванням обмежень на кількість місць у черзі та можливих повторних перевірок для стабільності роботи.

Вихідні дані:

У даному варіанті СМО вихідні дані визначаються, як:

Вихідні дані

Кількість заявок – $200 \cdot K_p$, K_p – кількість букв у Вашому прізвищі. K_g – кількість голосних букв в Вашому прізвищі. $K_{приг}$ – кількість приголосних букв в Вашому прізвищі.

Моє прізвище – Якимів

Розрахуємо вихідні дані на основі мого прізвища:

1. Кількість літер у прізвищі (K_p):

- У прізвищі Якимів 6 букв, отже, $K_p = 6$

2. Кількість голосних літер (K_g):

- Голосні в прізвищі: "я", "и", "і", отже $K_g = 3$

3. Кількість приголосних літер ($K_{приг}$):

- Приголосні в прізвищі: "к", "м", "в", отже $K_{приг} = 3$

Вихідні дані:

1. Кількість заявок: $200 * K_p = 200 * 6 = 1200$ заявок
2. Кількість обслуговуючих каналів (K_r): 3
3. Кількість місць у черзі ($K_{пр}$): 3

Тепер ці дані можна використовувати для реалізації обраної моделі в SimPy.

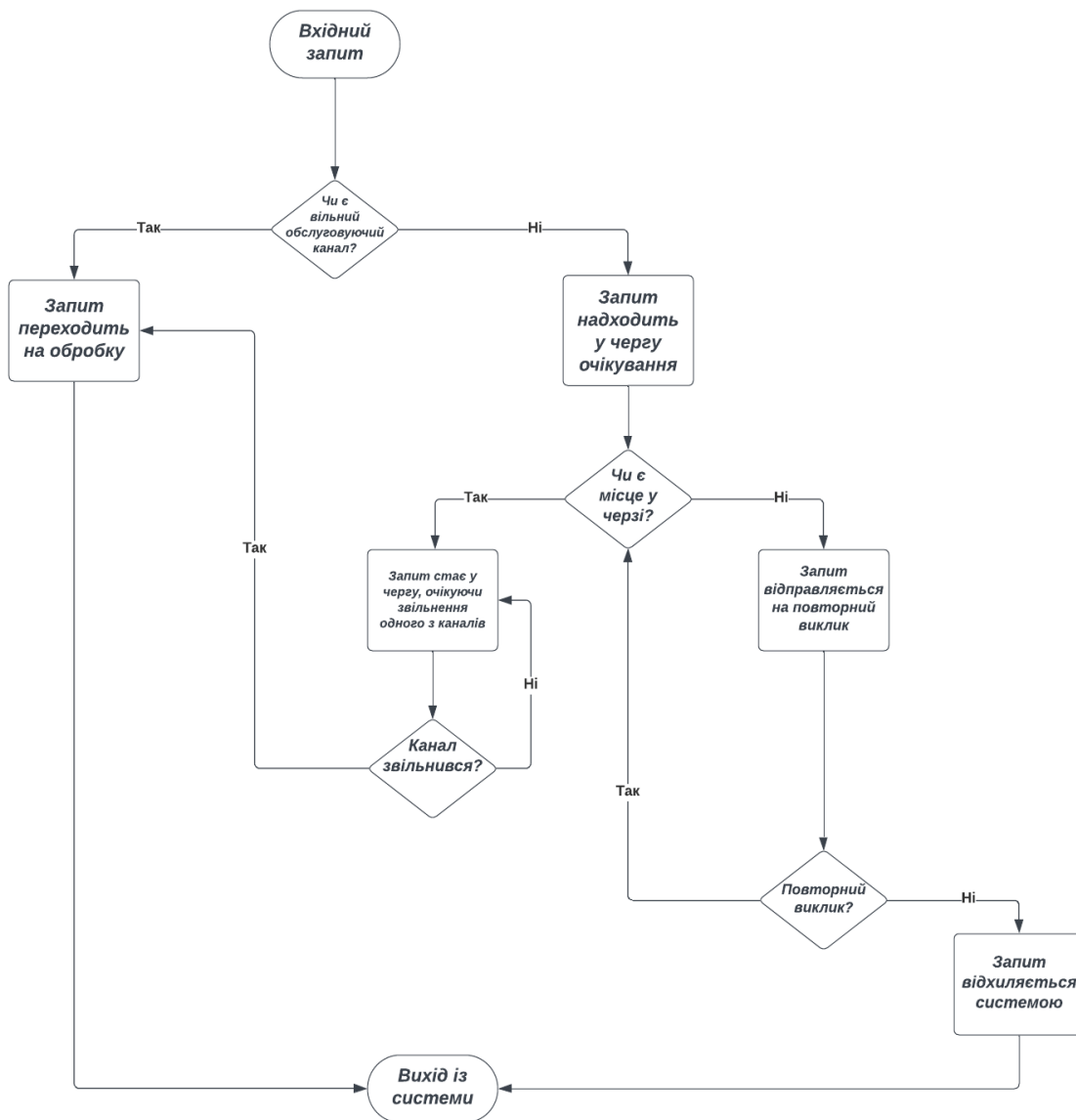
Побудувати та записати модель СМО у вигляді блок-схеми:

Табличний опис блок-схеми:

Етап	Опис
Вхідний запит	Запит надходить у систему для обслуговування
Чи є вільний обслуговуючий канал?	Перевіряється, чи є доступний обслуговуючий канал (з 3 каналів)
Так	Якщо є вільний канал, запит переходить на обробку
Ні	Якщо всі канали зайняті, запит надходить у чергу очікування
Чи є місце в черзі?	Якщо черга обмежена до 3 місць, перевіряється, чи є вільне місце в черзі
Так, є місце	Запит стає в чергу, очікуючи звільнення одного з каналів
Ні, черга заповнена	Якщо черга повна, переходимо до повторного виклику
Повторний виклик?	Чи зробити повторний виклик для запиту?
Так, повторний виклик	Запит повторно намагається зайти в систему після певного часу
Ні, відмова	Якщо повторний виклик недоступний, запит відхиляється системою
Вихід із системи	Після обслуговування або відмови запит виходить із системи

Блок-схема для системи масового обслуговування з повторними викликами та очікуванням в черзі з обмеженою кількістю місць (3 місця), яка обслуговує 1200 заявок і має 3 обслуговуючих канали виглядатиме так:

*Створювала блок-схему за допомогою сервісу Lucidchart([посилання на блок-схему](#))



Реалізувати модель в Python, використовуючи SimPy:

*Лабораторну роботу виконувала у середовищі Google Colab([посилання на нотбук](#))

1. Імпорт необхідних бібліотек

```
✓ [4] !pip install simpy
3c
⇒ Requirement already satisfied: simpy in /usr/local/lib/python3.10/dist-packages (4.1.1)
```

```
✓ [12] import simpy
0c      import random
      import statistics
      import numpy as np
```

2. Ініціалізуємо вихідні параметри моделі та змінні для збору статистики:

```
✓ [13] # Вихідні параметри
0c      NUM_REQUESTS = 1200          # Загальна кількість заявок
      NUM_CHANNELS = 3              # Кількість обслуговуючих каналів
      QUEUE_CAPACITY = 3           # Максимальна кількість місць в черзі
      RETRY_DELAY = 1              # Час затримки для повторного виклику

      # Змінні для збору статистики
      queue_lengths = []
      retry_counts = []
```

3. Створюємо процес обслуговування запитів:

```
✓ [14] def request_service(env, request_id, server, queue, retry_counts):
0c      """Процес обробки запиту з повторними викликами та обмеженою чергою."""
      retry_count = 0
      while True:
          with server.request() as req:
              result = yield req | env.timeout(RETRY_DELAY) # Очікування обслуговування або повторний виклик

              if req in result: # Якщо запит обслуговується
                  print(f"Запит {request_id} обслуговується у {env.now:.2f}")
                  yield env.timeout(random.expovariate(1.0)) # Тривалість обслуговування
                  break

              elif len(queue.items) < QUEUE_CAPACITY: # Якщо є місце в черзі
                  print(f"Запит {request_id} в черзі у {env.now:.2f}")
                  queue_lengths.append(len(queue.items))
                  yield queue.put(request_id)
                  yield env.timeout(RETRY_DELAY) # Чекає звільнення каналу

              else:
                  retry_count += 1 # Якщо черга переповнена, повторний виклик
                  retry_counts.append(retry_count)
                  print(f"Запит {request_id} робить повторний виклик у {env.now:.2f}")
                  yield env.timeout(RETRY_DELAY)
```

4. Генеруємо потік запитів:

```
✓ [15] def generate_requests(env, server, queue):  
0с      """Генератор запитів з інтервалом прибуття."""  
      for i in range(NUM_REQUESTS):  
          env.process(request_service(env, i, server, queue, retry_counts))  
          yield env.timeout(random.expovariate(1.0)) # Інтервал прибуття заявок
```

5. Створюємо модель та запускаємо симуляцію:

```
✓ # Ініціалізація середовища та ресурсів  
0с  env = simpy.Environment()  
    server = simpy.Resource(env, capacity=NUM_CHANNELS)  
    queue = simpy.Store(env, capacity=QUEUE_CAPACITY)  
  
# Запуск моделювання  
env.process(generate_requests(env, server, queue))  
env.run()
```

Результати моделювання(уривок):

```
➡ Запит 12 обслуговується у 13.10  
Запит 13 обслуговується у 13.64  
Запит 14 обслуговується у 13.79  
Запит 15 обслуговується у 16.10  
Запит 16 обслуговується у 16.44  
Запит 17 обслуговується у 17.48  
Запит 18 обслуговується у 18.69  
Запит 19 обслуговується у 21.96  
Запит 20 обслуговується у 22.06  
Запит 21 обслуговується у 24.85  
Запит 22 обслуговується у 27.58  
Запит 23 обслуговується у 27.98  
Запит 24 обслуговується у 28.65  
Запит 25 обслуговується у 30.02  
Запит 26 обслуговується у 30.65  
Запит 27 обслуговується у 30.70  
Запит 28 обслуговується у 31.77  
Запит 29 обслуговується у 31.83  
Запит 30 обслуговується у 33.25  
Запит 31 обслуговується у 34.73  
Запит 32 обслуговується у 35.69  
Запит 33 обслуговується у 36.03  
Запит 34 обслуговується у 36.42  
Запит 35 обслуговується у 37.28  
Запит 36 обслуговується у 37.45  
Запит 37 обслуговується у 37.79  
Запит 38 обслуговується у 39.19  
Запит 39 обслуговується у 40.91  
Запит 40 обслуговується у 42.92  
Запит 41 обслуговується у 44.21  
Запит 42 обслуговується у 45.52  
Запит 43 обслуговується у 45.59  
Запит 44 обслуговується у 46.51  
Запит 45 обслуговується у 47.40  
Запит 46 в черзі у 48.18
```



Запит 491 обслуговується у 483.76
Запит 492 обслуговується у 483.94
Запит 493 робить повторний виклик у 484.84
Запит 494 робить повторний виклик у 485.34
Запит 493 обслуговується у 486.34
Запит 495 обслуговується у 486.71
Запит 494 робить повторний виклик у 487.34
Запит 496 робить повторний виклик у 487.40
Запит 497 робить повторний виклик у 488.19
Запит 494 обслуговується у 488.34
Запит 496 обслуговується у 488.40
Запит 497 обслуговується у 489.19
Запит 498 обслуговується у 489.26
Запит 499 обслуговується у 490.51
Запит 500 обслуговується у 492.28
Запит 501 обслуговується у 494.26
Запит 502 обслуговується у 495.39
Запит 503 обслуговується у 495.75
Запит 504 обслуговується у 496.07
Запит 505 обслуговується у 497.44
Запит 506 обслуговується у 497.71
Запит 507 обслуговується у 498.11
Запит 508 обслуговується у 499.55
Запит 509 обслуговується у 500.11
Запит 510 обслуговується у 500.68
Запит 511 обслуговується у 502.01
Запит 512 обслуговується у 502.12
Запит 513 обслуговується у 502.40
Запит 514 обслуговується у 505.29
Запит 515 обслуговується у 506.02
Запит 516 обслуговується у 506.23
Запит 517 обслуговується у 508.43
Запит 518 обслуговується у 508.90
Запит 519 обслуговується у 509.95
Запит 520 обслуговується у 510.08

1. Обслуговування запитів:

- Запити обслуговуються у порядку їх надходження, коли канали вільні. Наприклад, запит 0 одразу обслуговується на позначці часу **0.00**, запит 1 — на **1.97** і так далі.
- Час обслуговування кожного запиту змінюється, оскільки надходження та обслуговування запитів слідує випадковому розподілу.

2. Формування черги:

- Початкові запити (0–45) обслуговуються без затримок, оскільки в системі вистачає ресурсів.
- Починаючи з **запиту 46** (який очікує в черзі у **48.18** та обслуговується у **49.18**), черга починає заповнюватися через обмеженість ресурсів, оскільки канали вже зайняті.
- Деякі запити після заповнення черги не можуть одразу потрапити на обслуговування. Запити **493, 494, 496, 497** роблять повторні виклики, оскільки система в цей момент перевантажена, що затримує їх обробку.

3. Повторні виклики:

- Коли запити не можуть зайняти канал через перевантаженість або заповнену чергу, вони повторно пробують отримати доступ через певний інтервал часу.
- Повторні виклики відбуваються, наприклад, у запитів 493, 494, і 497, які спершу не змогли отримати обслуговування, але повторно намагаються це зробити, поки нарешті не отримують доступ.

Аналіз навантаження

1. Пропускна здатність системи:

- На початку моделювання система ефективно обробляє запити, використовуючи доступні канали.
- При збільшенні кількості запитів і часу моделювання система досягає свого навантаження, що призводить до черг і повторних викликів.

2. Середній час обслуговування та черги:

- Система має обмежену здатність обробляти запити одночасно, тому із часом зростає середня довжина черги.
- Середній час очікування в черзі та кількість повторних викликів показують, наскільки система наближається до своєї межі ефективної роботи.

Внести корективи в програмний код для отримання наступних результатів:

1. Дослідити параметри загальної черги:

- середнє значення довжини черги,
- максимальну довжину черги,
- розподіл довжини черги (відобразити графічно)
- середній час перебування заявки в черзі
- максимальний час перебування заявки в черзі

2. Дослідити кількість повторних викликів (Для тих СМО, яким це необхідно).

Визначити, скільки в середньому викликів робить користувач, аби бути обслуженим.

3. Зібрану статистику по попередній програмі (з різними результатами ситуацій) представити графічно в вигляді гістограм.

4. Дослідити параметри черг до кожного з каналів (параметри такі ж як п.1)

5. Дослідити залежність кількості відмов в обслуговуванні та кількості повторних викликів від росту інтенсивності вхідного потоку заявок - збільшити інтенсивність вхідного потоку

- 1x
- 2x
- 5x
- 10x
- 20x
- 50x
- 100x

Щоб виконати всі вимоги, внесемо такі корективи в код:

1. **Збір та обчислення статистики черги** (середня та максимальна довжина черги, час перебування заявок у черзі).
2. **Аналіз повторних викликів** (середня кількість повторних спроб на одну заявку).
3. **Графічне представлення результатів** (гістограми для довжини черги та повторних викликів).
4. **Дослідження параметрів черг до кожного каналу.**
5. **Моделювання для різних інтенсивностей вхідного потоку.**

Модифікований код:

```
import simpy
import random
import numpy as np
import matplotlib.pyplot as plt

# Вихідні параметри
NUM_REQUESTS = 500
NUM_CHANNELS = 3
QUEUE_CAPACITY = 3
RETRY_DELAY = 1

# Змінні для збору статистики
queue_lengths = []
time_in_queue = []
retry_counts = []
failed_requests = 0 # Лічильник для відмов

def request_service(env, request_id, server, queue, retry_counts):
    """Процес обробки запиту з повторними викликами, обмеженою чергою і можливими відмовами."""
    global failed_requests
    arrival_time = env.now
    retry_count = 0
    while True:
        with server.request() as req:
            result = yield req | env.timeout(RETRY_DELAY)
            if req in result: # Якщо запит обслуговується
                wait_time = env.now - arrival_time
                time_in_queue.append(wait_time)
                queue_lengths.append(len(queue.items))
                print(f"Запит {request_id} обслуговується у {env.now:.2f}, після очікування {wait_time:.2f}")
            else:
                failed_requests += 1
                retry_count += 1
                if retry_count == 3:
                    print(f"Запит {request_id} не обслуговується після {retry_count} спроб")
                    break
            yield env.timeout(random.expovariate(1.0))
```

```

        break
    elif len(queue.items) < QUEUE_CAPACITY: # Якщо є місце в черзі
        queue_lengths.append(len(queue.items))
        queue.put(request_id)
        yield env.timeout(RETRY_DELAY)
    else: # Якщо черга заповнена і потрібно відмовити
        retry_count += 1
        retry_counts.append(retry_count)
        failed_requests += 1 # Фіксуємо відмову
        print(f"Запит {request_id} відхилено у {env.now:.2f} після {retry_count}
повторних спроб.")
        break

def generate_requests(env, server, queue, intensity_multiplier):
    """Генератор запитів з інтервалом прибуття, враховуючи множник інтенсивності."""
    for i in range(NUM_REQUESTS):
        env.process(request_service(env, i, server, queue, retry_counts))
        # Застосовуємо множник до інтенсивності, зменшуючи час між запитами
        yield env.timeout(random.expovariate(1.0 * intensity_multiplier))

def analyze_statistics():
    """Аналіз і виведення статистики для черги, повторних викликів і відмов."""
    print("\nЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ")

    if queue_lengths:
        print(f"Середня довжина черги: {np.mean(queue_lengths):.2f}")
        print(f"Максимальна довжина черги: {np.max(queue_lengths):.2f}")
    else:
        print("Середня довжина черги: N/A")
        print("Максимальна довжина черги: N/A")

    if time_in_queue:
        print(f"Середній час перебування в черзі: {np.mean(time_in_queue):.2f}")
        print(f"Максимальний час перебування в черзі: {np.max(time_in_queue):.2f}")
    else:
        print("Середній час перебування в черзі: N/A")
        print("Максимальний час перебування в черзі: N/A")

    if retry_counts:
        print(f"Середня кількість повторних викликів: {np.mean(retry_counts):.2f}")
        print(f"Максимальна кількість повторних викликів для одного запиту:
{np.max(retry_counts):.2f}")
    else:
        print("Середня кількість повторних викликів: N/A")
        print("Максимальна кількість повторних викликів для одного запиту: N/A")

```

```

    print(f"Кількість відмов: {failed_requests}")

def plot_histogram(data, title, xlabel, ylabel):
    """Функція для побудови гістограм."""
    plt.figure(figsize=(10, 6))
    plt.hist(data, bins=20, color='skyblue', edgecolor='black')
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.show()

# Основна функція моделювання
def run_simulation(intensity_multiplier=1):
    global queue_lengths, time_in_queue, retry_counts, failed_requests
    queue_lengths = []
    time_in_queue = []
    retry_counts = []
    failed_requests = 0

    # Ініціалізація середовища та ресурсів
    env = simpy.Environment()
    server = simpy.Resource(env, capacity=NUM_CHANNELS)
    queue = simpy.Store(env, capacity=QUEUE_CAPACITY)

    # Запуск генерації запитів
    env.process(generate_requests(env, server, queue, intensity_multiplier))
    env.run()

    # Виведення статистики
    analyze_statistics()

    # Графічне представлення
    if queue.lengths:
        plot_histogram(queue_lengths, "Розподіл довжини черги", "Довжина черги", "Кількість")
    else:
        print("Немає даних для побудови гістограми довжини черги.")

    if time_in_queue:
        plot_histogram(time_in_queue, "Розподіл часу перебування в черзі", "Час у черзі",
"Кількість запитів")
    else:
        print("Немає даних для побудови гістограми часу перебування в черзі.")

    if retry_counts:

```

```

        plot_histogram(retry_counts, "Розподіл повторних викликів", "Кількість повторних
викликів", "Кількість запитів")
    else:
        print("Немає даних для побудови гістограми повторних викликів.")

# Запуск моделювання для 100x інтенсивності
print("\nМоделювання з інтенсивністю потоку: 100x")
run_simulation(intensity_multiplier=100)

```

Моделювання з інтенсивністю потоку 1x:

```

Моделювання з інтенсивністю потоку: 1x
Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 0.42, після очікування 0.00
Запит 2 обслуговується у 3.87, після очікування 0.00
Запит 3 обслуговується у 4.23, після очікування 0.00
Запит 4 обслуговується у 5.71, після очікування 0.00
Запит 5 обслуговується у 6.39, після очікування 0.00
Запит 6 обслуговується у 8.98, після очікування 0.00
Запит 7 обслуговується у 10.11, після очікування 0.00
Запит 8 обслуговується у 10.17, після очікування 0.00
Запит 9 обслуговується у 10.24, після очікування 0.00
Запит 10 обслуговується у 10.44, після очікування 0.00
Запит 11 обслуговується у 11.30, після очікування 0.00
Запит 12 обслуговується у 11.38, після очікування 0.00
Запит 13 обслуговується у 11.49, після очікування 0.00
Запит 14 обслуговується у 13.67, після очікування 0.00
Запит 15 обслуговується у 14.39, після очікування 0.00
Запит 16 обслуговується у 15.17, після очікування 0.00
Запит 17 обслуговується у 17.40, після очікування 0.00
Запит 18 обслуговується у 18.49, після очікування 0.00
Запит 19 обслуговується у 19.60, після очікування 0.00
Запит 20 обслуговується у 19.90, після очікування 0.00
Запит 21 обслуговується у 20.32, після очікування 0.38
Запит 22 обслуговується у 20.71, після очікування 0.00
Запит 23 обслуговується у 21.62, після очікування 0.00
Запит 24 обслуговується у 22.88, після очікування 0.00

```

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 1.49

Максимальна довжина черги: 3.00

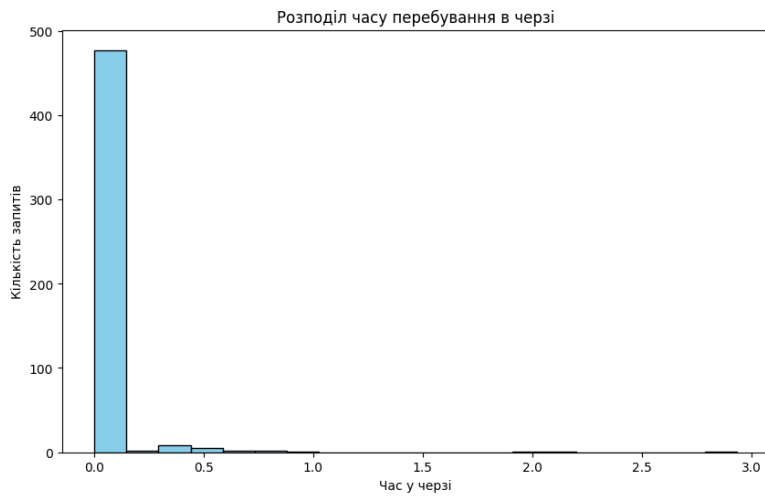
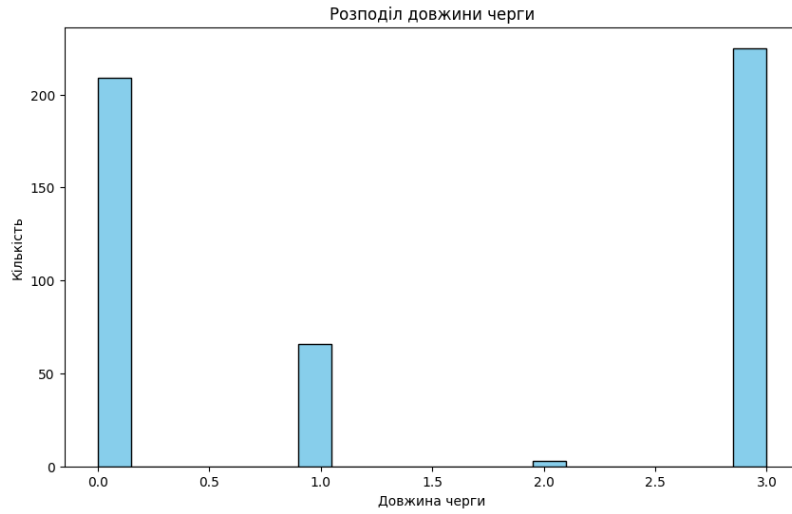
Середній час перебування в черзі: 0.03

Максимальний час перебування в черзі: 2.94

Середня кількість повторних викликів: N/A

Максимальна кількість повторних викликів для одного запиту: N/A

Кількість відмов: 0



Немає даних для побудови гістограми повторних викликів, так як їх не було

Моделювання з інтенсивністю потоку 2х:

Моделювання з інтенсивністю потоку: 2x

Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 0.46, після очікування 0.00
Запит 2 обслуговується у 1.32, після очікування 0.00
Запит 3 обслуговується у 1.58, після очікування 0.00
Запит 4 обслуговується у 4.27, після очікування 0.00
Запит 5 обслуговується у 5.20, після очікування 0.00
Запит 6 обслуговується у 5.24, після очікування 0.00
Запит 7 обслуговується у 8.27, після очікування 0.00
Запит 8 обслуговується у 10.00, після очікування 0.00
Запит 9 обслуговується у 13.87, після очікування 0.00
Запит 10 обслуговується у 13.91, після очікування 0.00
Запит 11 обслуговується у 15.00, після очікування 0.00
Запит 12 обслуговується у 15.78, після очікування 0.00
Запит 13 обслуговується у 16.41, після очікування 0.00
Запит 14 обслуговується у 16.65, після очікування 0.00
Запит 15 обслуговується у 17.18, після очікування 0.00
Запит 16 обслуговується у 18.56, після очікування 0.00
Запит 17 обслуговується у 19.81, після очікування 0.00
Запит 18 обслуговується у 20.23, після очікування 0.00
Запит 19 обслуговується у 22.39, після очікування 0.00
Запит 20 обслуговується у 22.88, після очікування 0.00
Запит 21 обслуговується у 23.38, після очікування 0.00
Запит 22 обслуговується у 25.26, після очікування 0.00
Запит 23 обслуговується у 25.97, після очікування 0.00
Запит 24 обслуговується у 27.03, після очікування 0.00

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 2.38

Максимальна довжина черги: 3.00

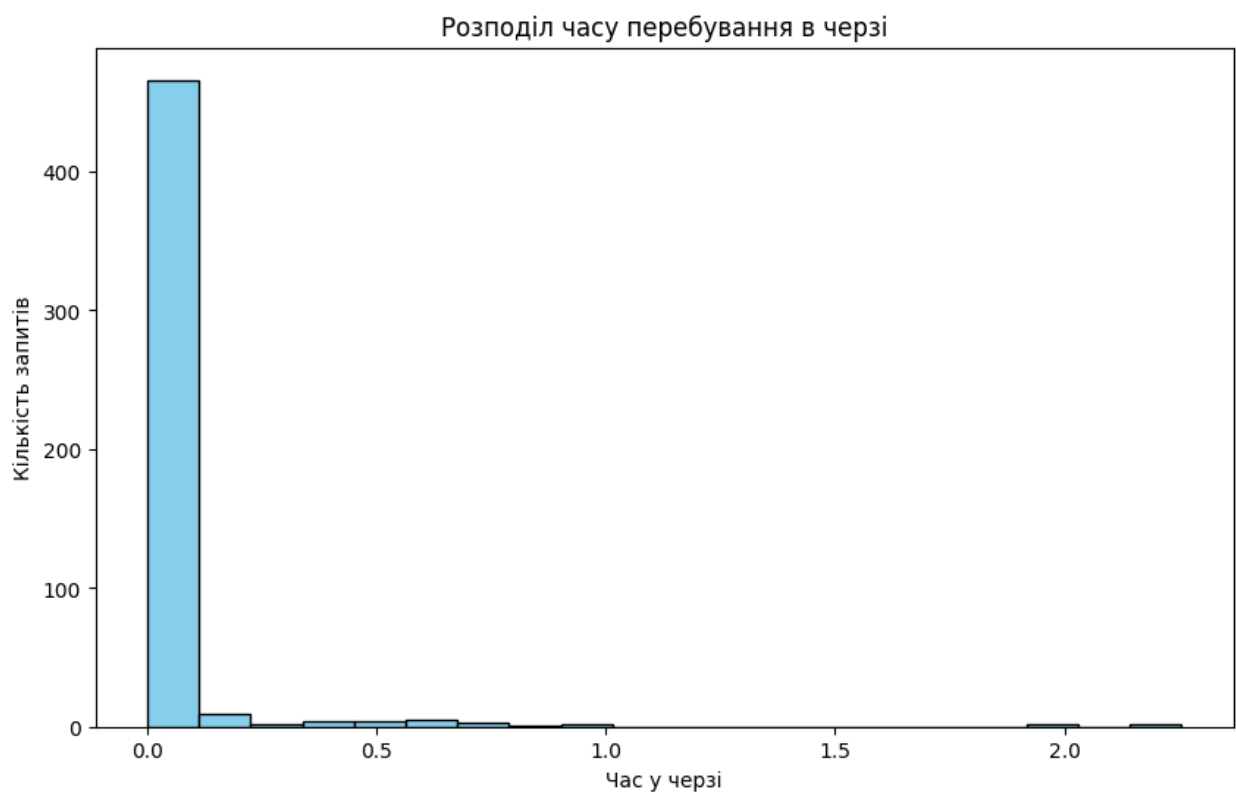
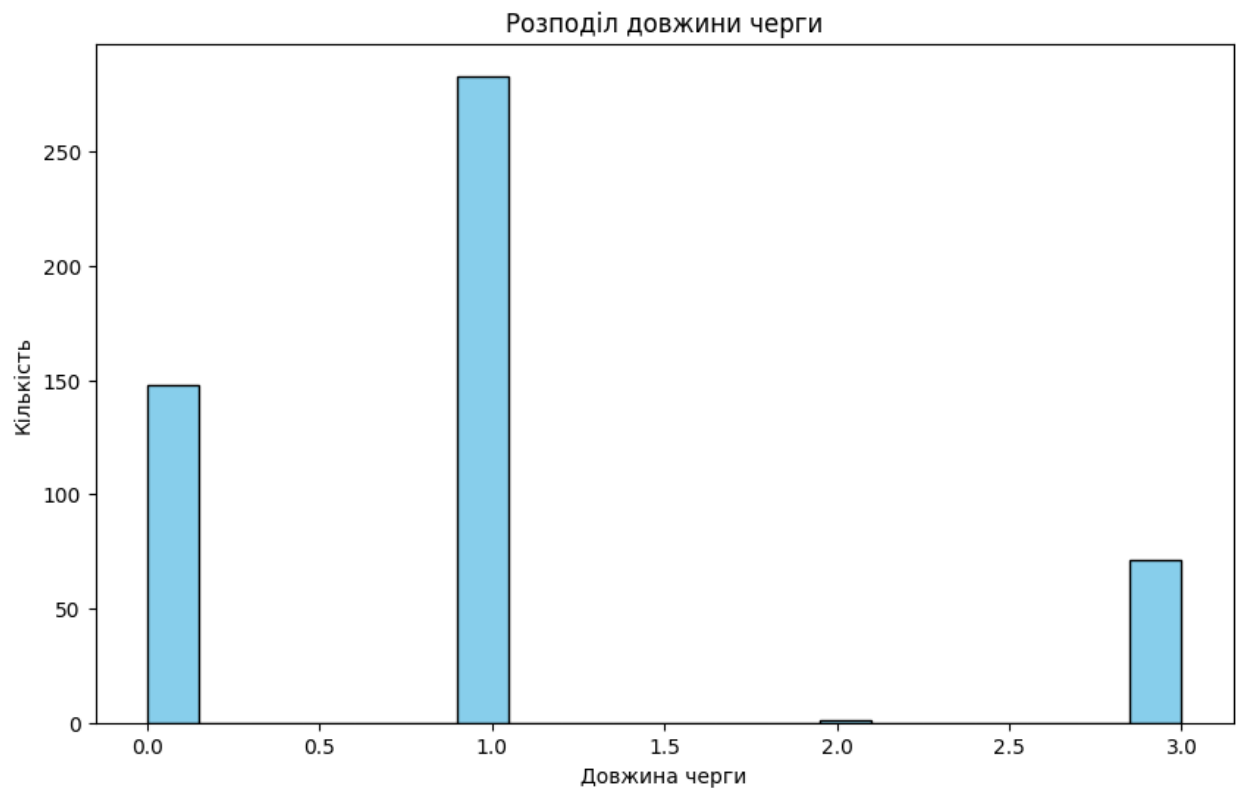
Середній час перебування в черзі: 0.05

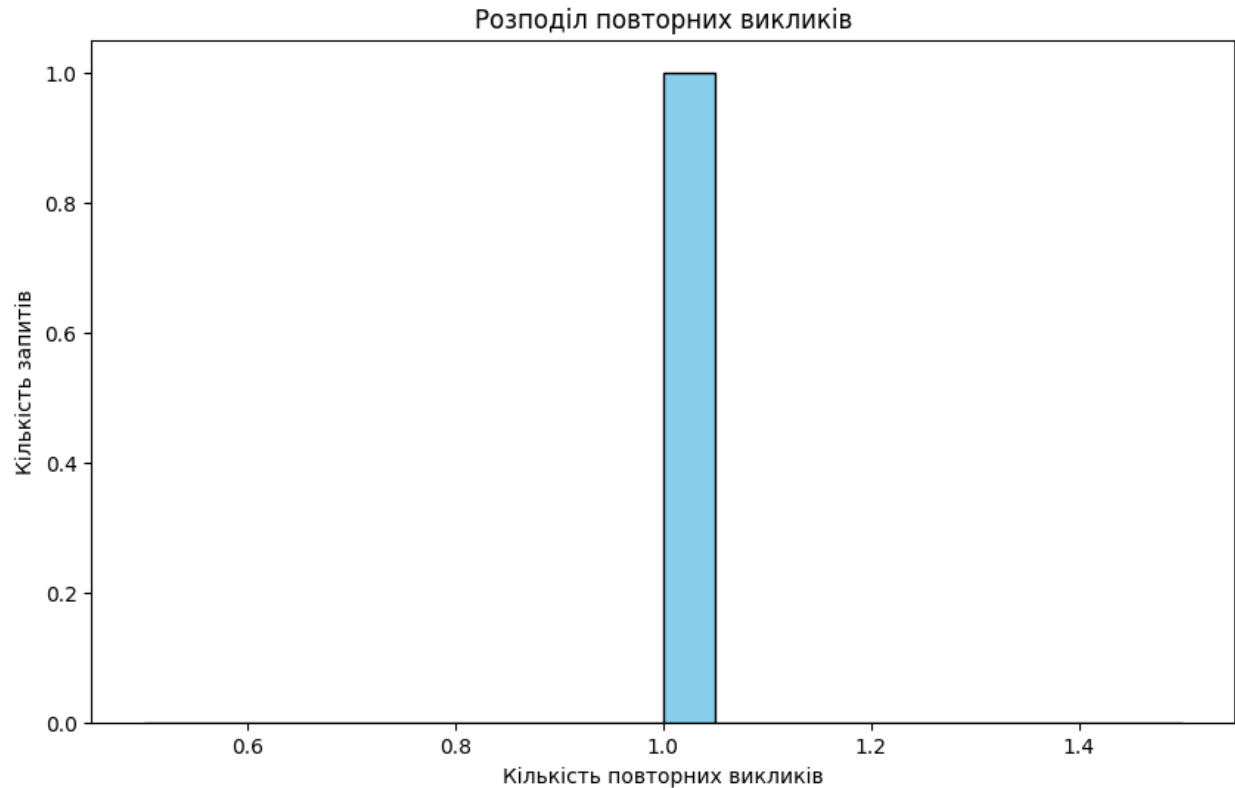
Максимальний час перебування в черзі: 2.45

Середня кількість повторних викликів: 1.00

Максимальна кількість повторних викликів для одного запиту: 1.00

Кількість відмов: 2





Моделювання з інтенсивністю потоку 5x:

Моделювання з інтенсивністю потоку: 5x

Запит 0	обслуговується у 0.00, після очікування 0.00
Запит 1	обслуговується у 1.25, після очікування 0.00
Запит 2	обслуговується у 1.89, після очікування 0.00
Запит 3	обслуговується у 2.19, після очікування 0.00
Запит 4	обслуговується у 3.04, після очікування 0.00
Запит 5	обслуговується у 4.68, після очікування 0.00
Запит 6	обслуговується у 4.87, після очікування 0.00
Запит 7	обслуговується у 5.57, після очікування 0.00
Запит 8	обслуговується у 6.50, після очікування 0.00
Запит 9	обслуговується у 7.06, після очікування 0.00
Запит 10	обслуговується у 9.43, після очікування 0.00
Запит 11	обслуговується у 9.61, після очікування 0.00
Запит 12	обслуговується у 11.68, після очікування 0.00
Запит 13	обслуговується у 13.70, після очікування 0.00
Запит 14	обслуговується у 14.94, після очікування 0.00
Запит 15	обслуговується у 15.13, після очікування 0.00
Запит 16	обслуговується у 16.82, після очікування 0.00
Запит 17	обслуговується у 17.87, після очікування 0.00
Запит 18	обслуговується у 18.21, після очікування 0.00
Запит 19	обслуговується у 18.27, після очікування 0.00
Запит 20	обслуговується у 19.78, після очікування 0.00
Запит 21	обслуговується у 19.86, після очікування 0.00
Запит 22	обслуговується у 21.13, після очікування 0.00

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 1.01

Максимальна довжина черги: 3.00

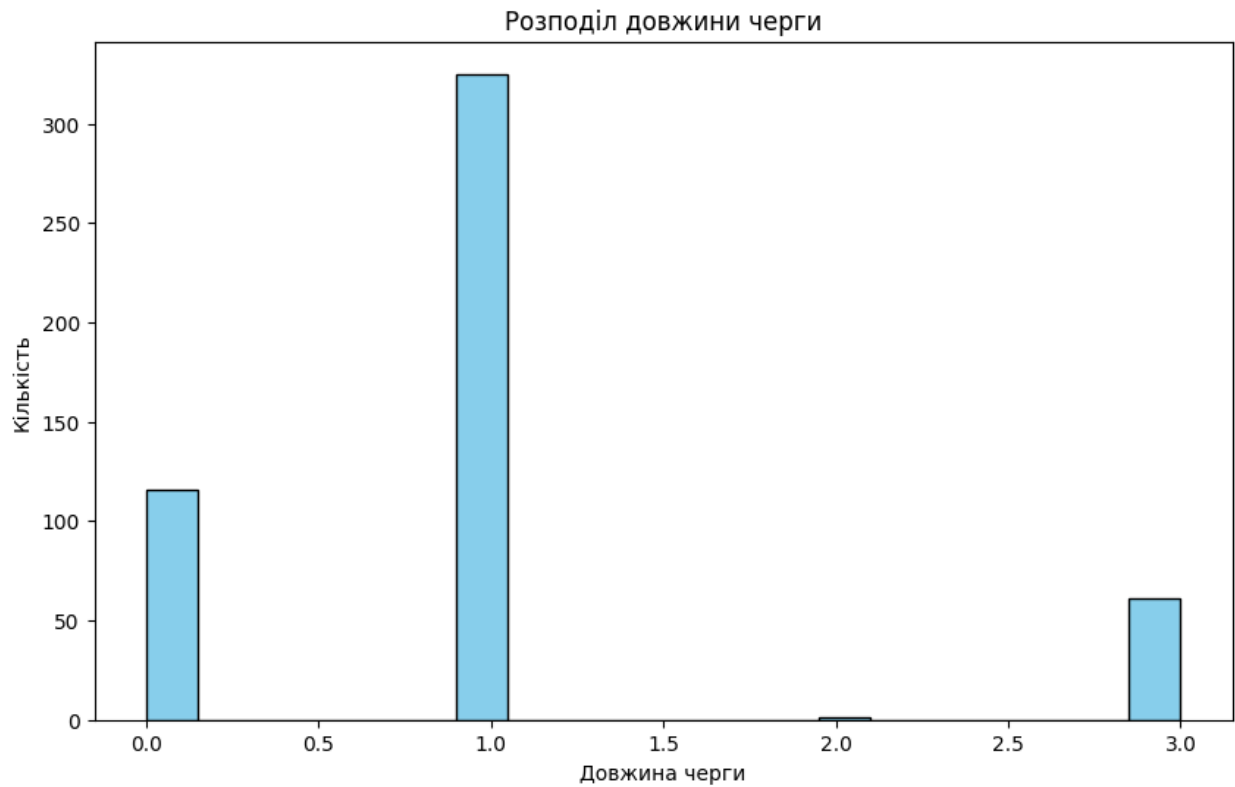
Середній час перебування в черзі: 0.04

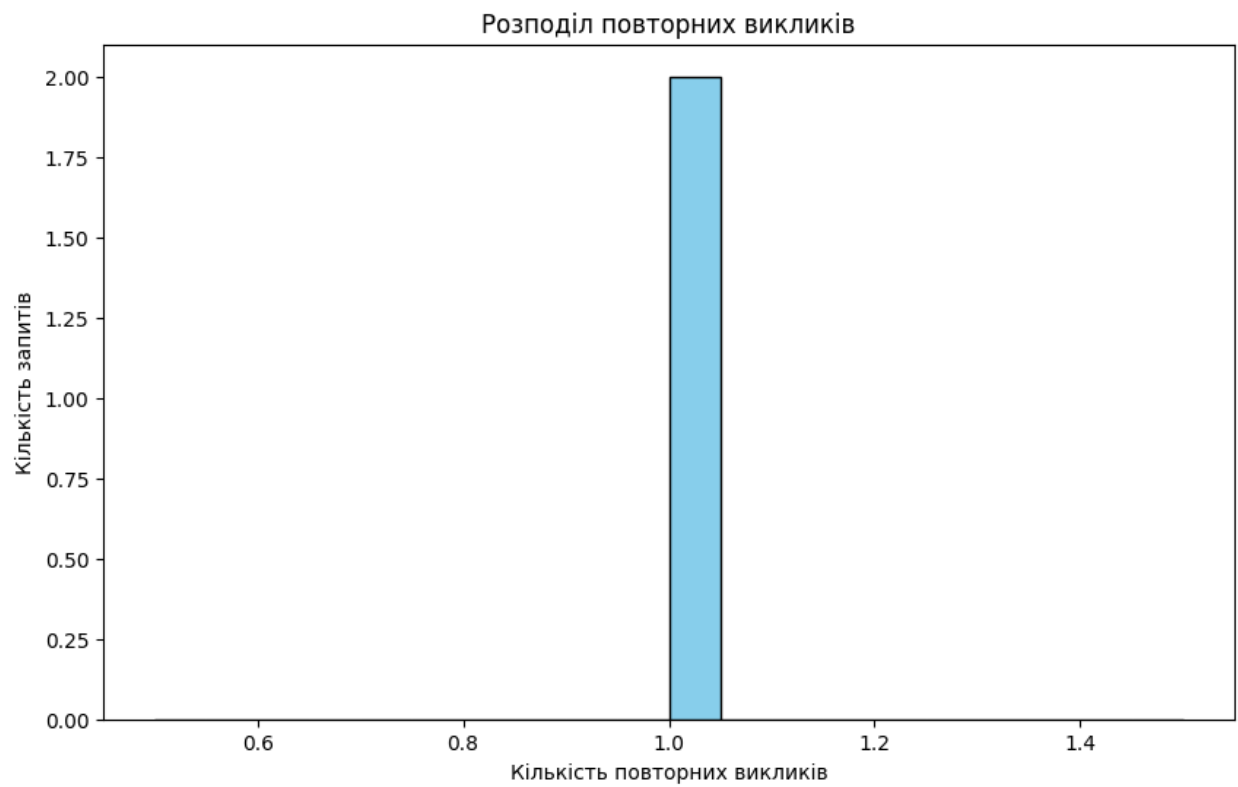
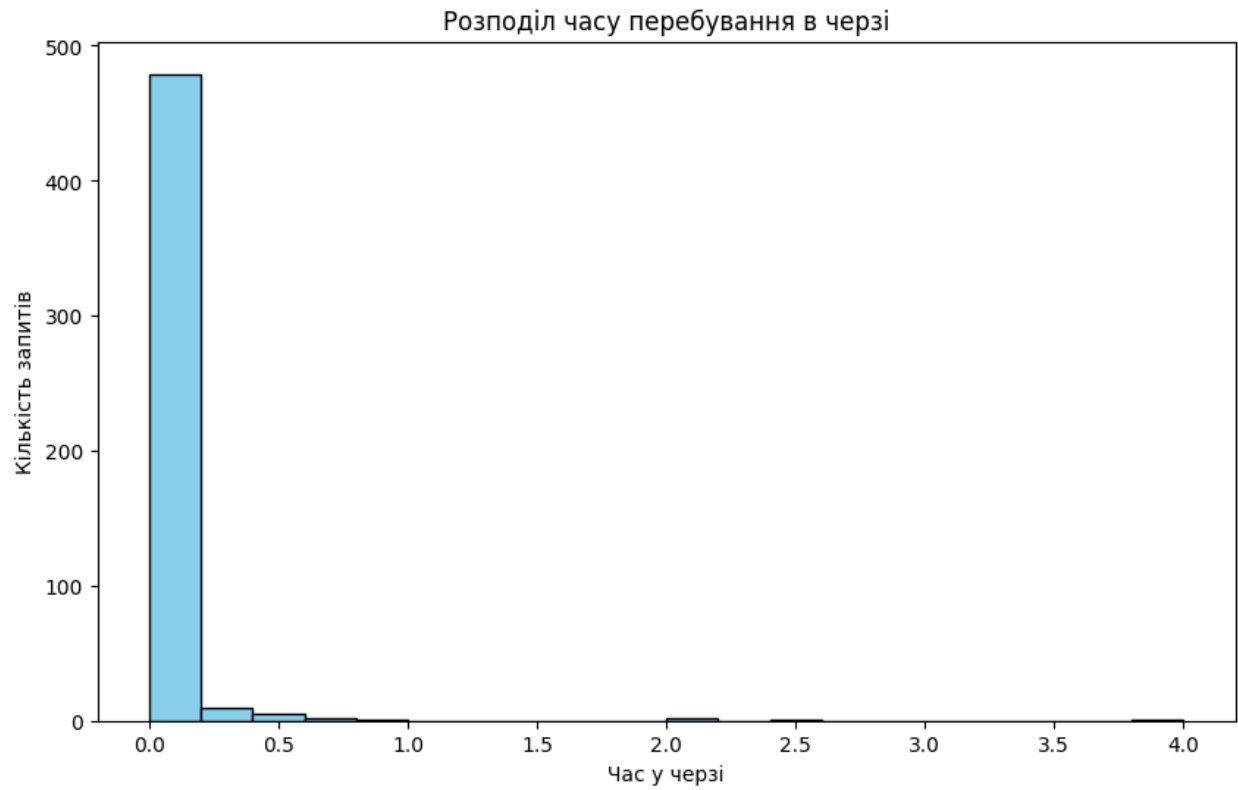
Максимальний час перебування в черзі: 4.00

Середня кількість повторних викликів: 1.00

Максимальна кількість повторних викликів для одного запиту: 1.00

Кількість відмов: 0





Моделювання з інтенсивністю потоку 10х:

Моделювання з інтенсивністю потоку: 10х

Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 1.20, після очікування 0.00
Запит 2 обслуговується у 2.46, після очікування 0.00
Запит 3 обслуговується у 3.10, після очікування 0.00
Запит 4 обслуговується у 6.74, після очікування 0.00
Запит 5 обслуговується у 8.46, після очікування 0.00
Запит 6 обслуговується у 9.39, після очікування 0.00
Запит 7 обслуговується у 10.22, після очікування 0.00
Запит 8 обслуговується у 10.81, після очікування 0.00
Запит 9 обслуговується у 11.53, після очікування 0.00
Запит 10 обслуговується у 13.11, після очікування 0.00
Запит 11 обслуговується у 14.25, після очікування 0.00
Запит 12 обслуговується у 14.89, після очікування 0.00
Запит 13 обслуговується у 16.38, після очікування 0.00
Запит 14 обслуговується у 16.72, після очікування 0.00
Запит 15 обслуговується у 17.31, після очікування 0.00
Запит 16 обслуговується у 17.37, після очікування 0.03
Запит 17 обслуговується у 17.86, після очікування 0.14
Запит 18 обслуговується у 18.58, після очікування 0.00
Запит 19 обслуговується у 18.62, після очікування 0.00
Запит 20 обслуговується у 19.42, після очікування 0.00
Запит 21 обслуговується у 19.81, після очікування 0.04
Запит 22 обслуговується у 19.97, після очікування 0.14
Запит 23 обслуговується у 20.86, після очікування 0.39

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 1.36

Максимальна довжина черги: 3.00

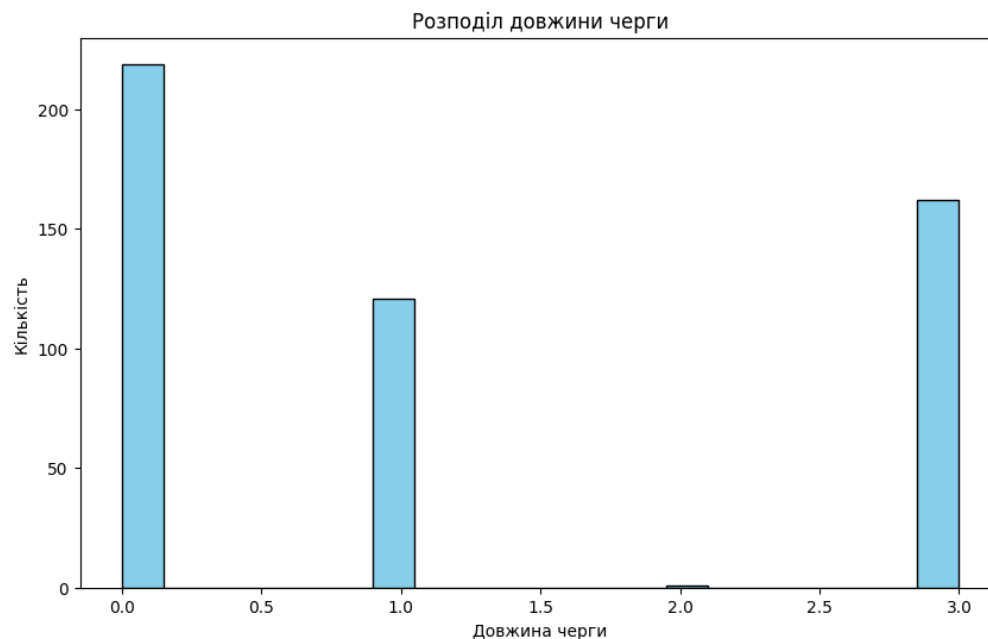
Середній час перебування в черзі: 0.03

Максимальний час перебування в черзі: 2.50

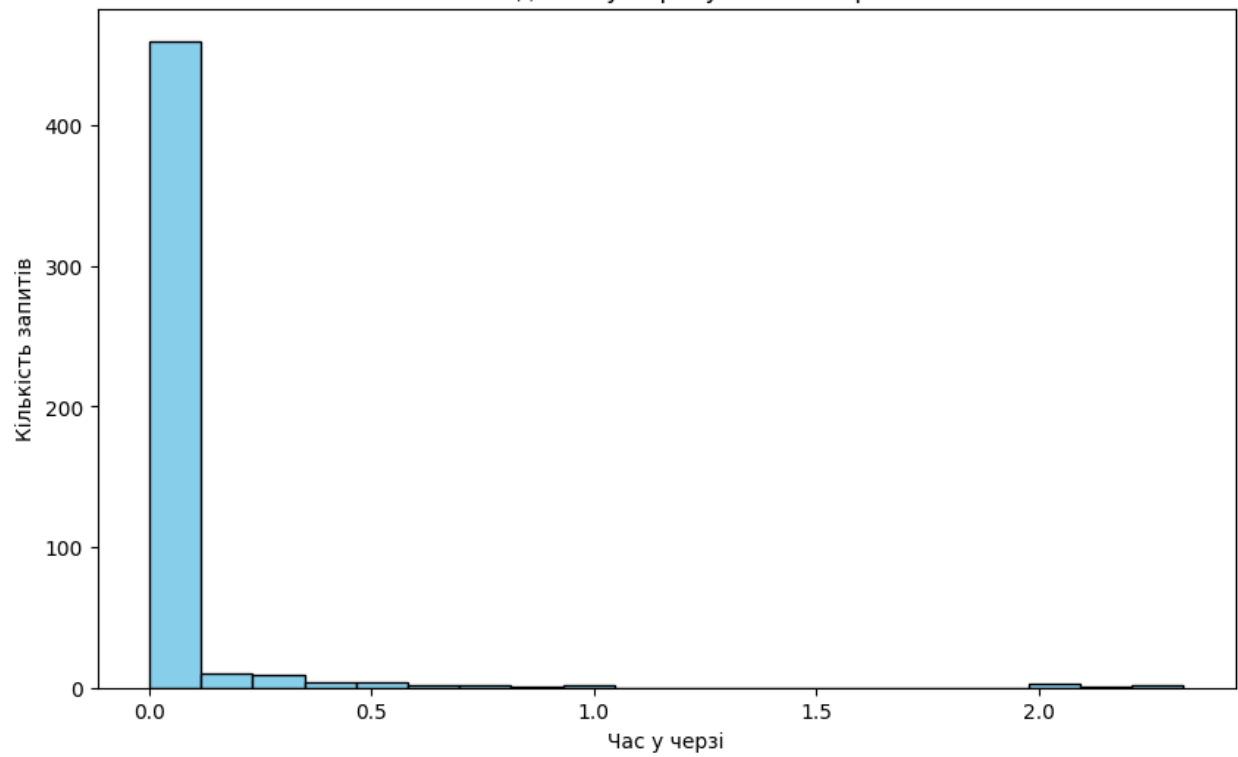
Середня кількість повторних викликів: 1.00

Максимальна кількість повторних викликів для одного запиту: 1.00

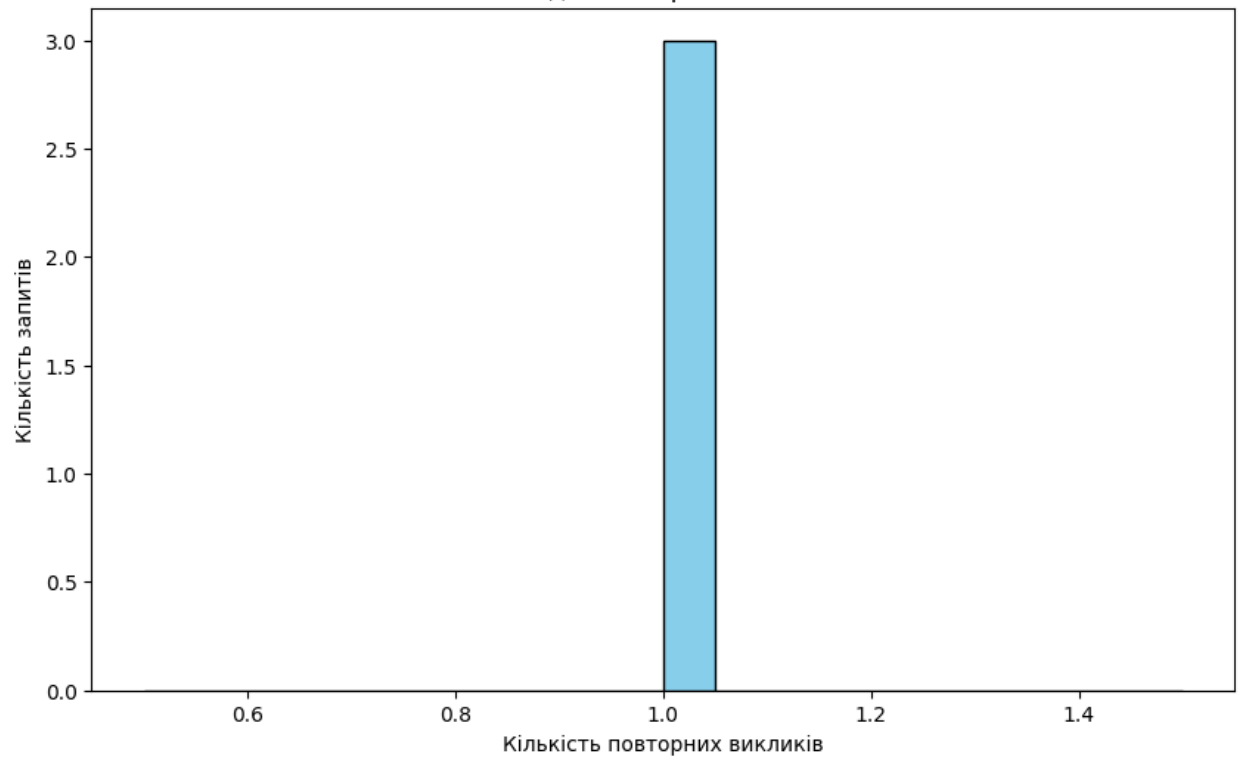
Кількість відмов: 7



Розподіл часу перебування в черзі



Розподіл повторних викликів



Моделювання з інтенсивністю потоку 20х:

Моделювання з інтенсивністю потоку: 20х

Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 0.18, після очікування 0.00
Запит 2 обслуговується у 1.15, після очікування 0.00
Запит 3 обслуговується у 2.87, після очікування 0.00
Запит 4 обслуговується у 4.88, після очікування 0.00
Запит 5 обслуговується у 5.70, після очікування 0.00
Запит 6 обслуговується у 6.16, після очікування 0.00
Запит 7 обслуговується у 8.12, після очікування 0.00
Запит 8 обслуговується у 11.02, після очікування 0.00
Запит 9 обслуговується у 12.61, після очікування 0.00
Запит 10 обслуговується у 15.04, після очікування 0.00
Запит 11 обслуговується у 15.84, після очікування 0.00
Запит 12 обслуговується у 16.08, після очікування 0.00
Запит 13 обслуговується у 17.54, після очікування 0.00
Запит 14 обслуговується у 17.83, після очікування 0.00
Запит 15 обслуговується у 18.67, після очікування 0.00
Запит 16 обслуговується у 19.15, після очікування 0.00
Запит 17 обслуговується у 21.19, після очікування 0.00
Запит 18 обслуговується у 22.28, після очікування 0.00
Запит 19 обслуговується у 23.54, після очікування 0.00
Запит 20 обслуговується у 23.94, після очікування 0.00
Запит 21 обслуговується у 24.96, після очікування 0.00
Запит 22 обслуговується у 27.50, після очікування 0.00
Запит 23 обслуговується у 27.93, після очікування 0.00
Запит 24 обслуговується у 29.22, після очікування 0.00

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 1.02

Максимальна довжина черги: 3.00

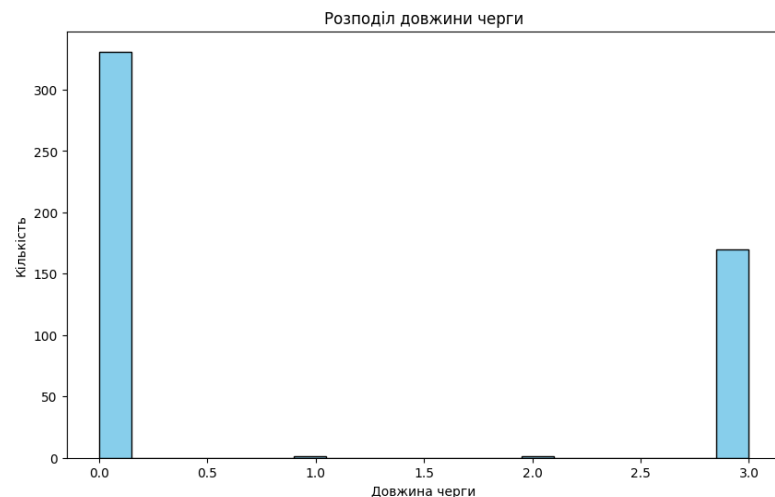
Середній час перебування в черзі: 0.12

Максимальний час перебування в черзі: 6.90

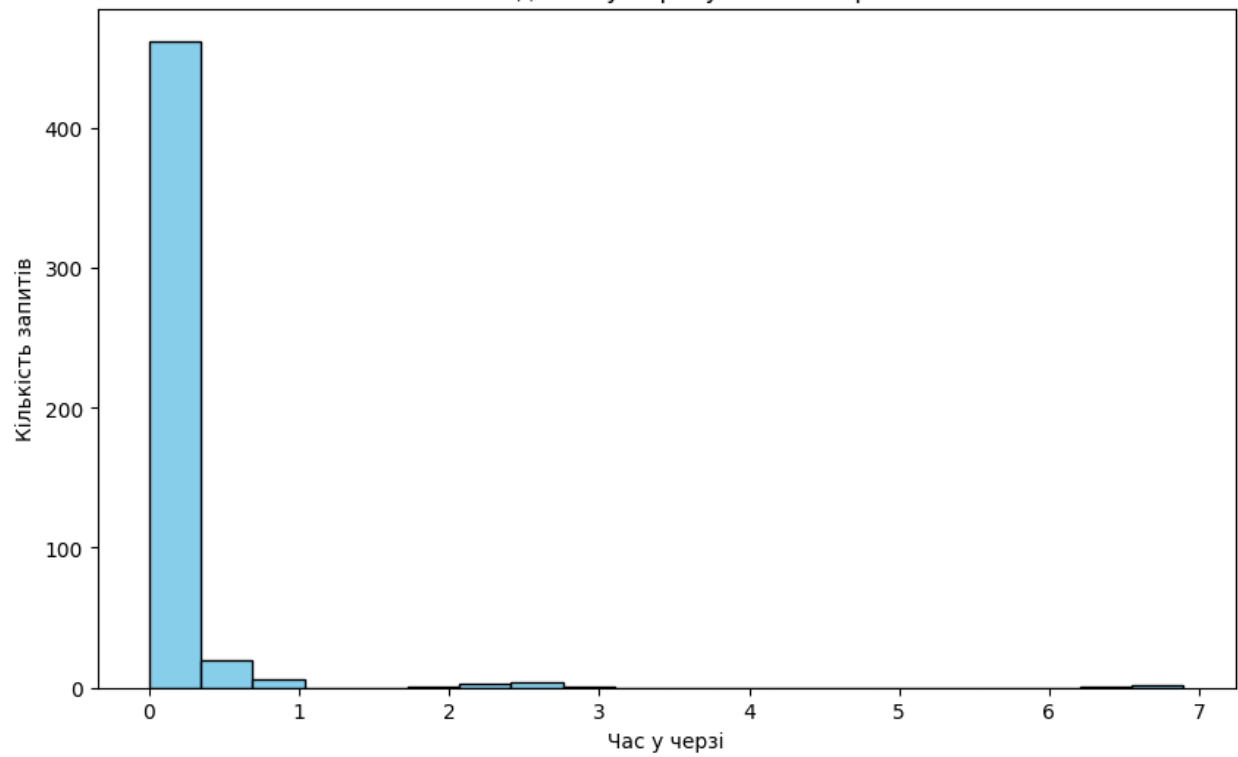
Середня кількість повторних викликів: 1.20

Максимальна кількість повторних викликів для одного запиту: 2.00

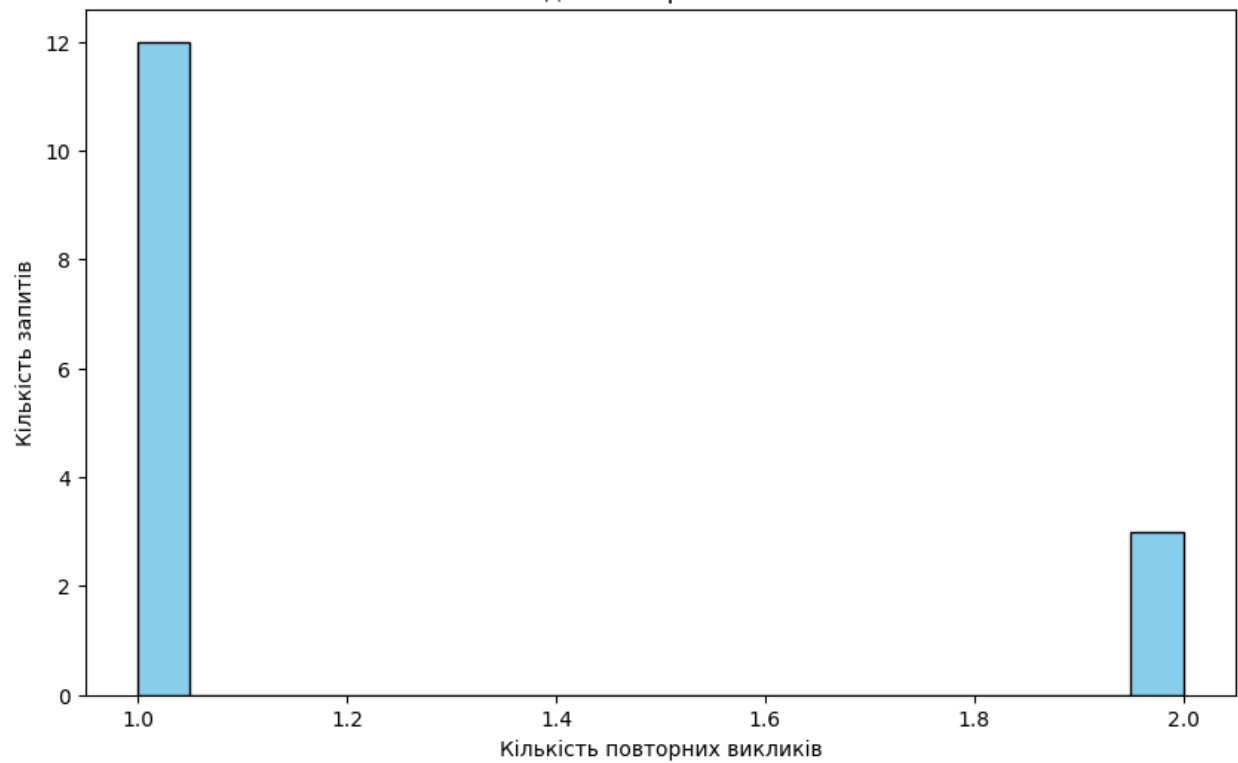
Кількість відмов: 5



Розподіл часу перебування в черзі



Розподіл повторних викликів



Моделювання з інтенсивністю потоку 50х:

Моделювання з інтенсивністю потоку: 50х

Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 1.90, після очікування 0.00
Запит 2 обслуговується у 2.19, після очікування 0.00
Запит 3 обслуговується у 3.56, після очікування 0.00
Запит 4 обслуговується у 5.74, після очікування 0.00
Запит 5 обслуговується у 8.21, після очікування 0.00
Запит 6 обслуговується у 8.97, після очікування 0.00
Запит 7 обслуговується у 9.55, після очікування 0.35
Запит 8 обслуговується у 9.74, після очікування 0.32
Запит 9 обслуговується у 10.07, після очікування 0.00
Запит 10 обслуговується у 11.93, після очікування 0.00
Запит 11 обслуговується у 12.12, після очікування 0.00
Запит 12 обслуговується у 12.86, після очікування 0.00
Запит 13 обслуговується у 14.86, після очікування 0.00
Запит 14 обслуговується у 16.75, після очікування 0.00
Запит 15 обслуговується у 17.41, після очікування 0.00
Запит 16 обслуговується у 18.41, після очікування 0.00
Запит 17 обслуговується у 19.77, після очікування 0.00
Запит 18 обслуговується у 20.00, після очікування 0.00
Запит 19 обслуговується у 20.11, після очікування 0.00
Запит 20 обслуговується у 23.69, після очікування 0.00

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 2.85

Максимальна довжина черги: 3.00

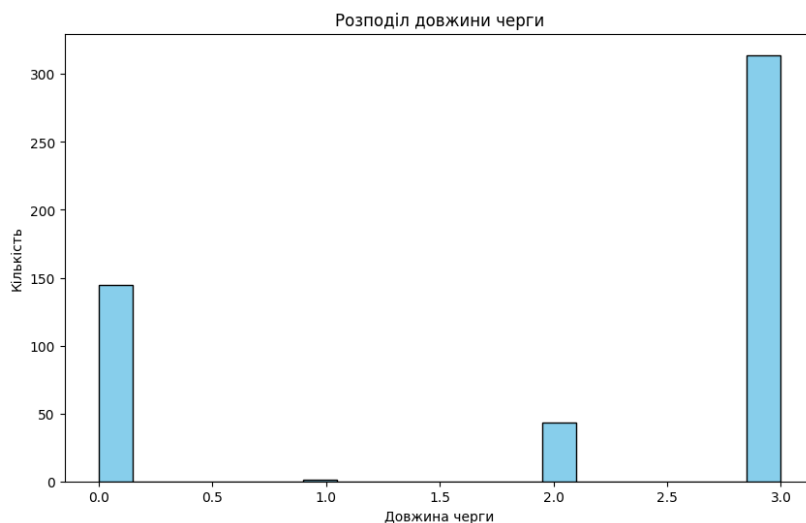
Середній час перебування в черзі: 0.05

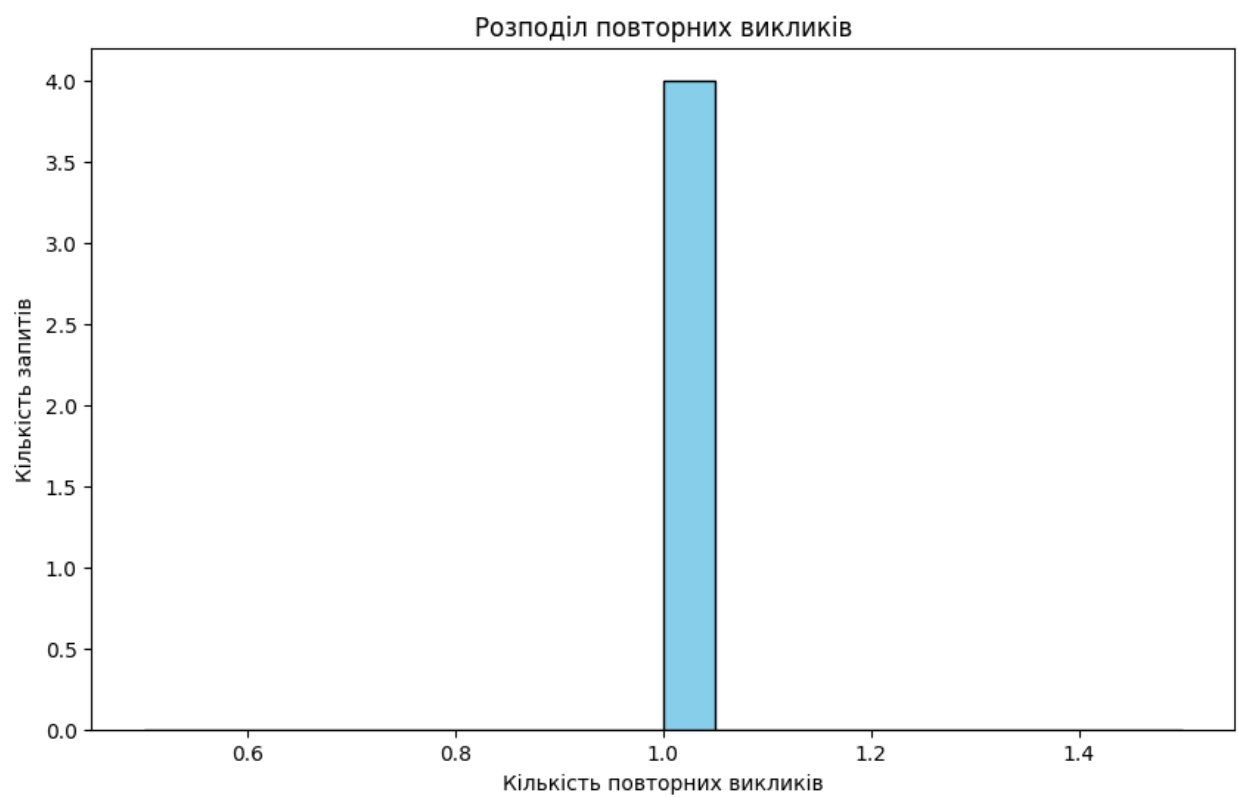
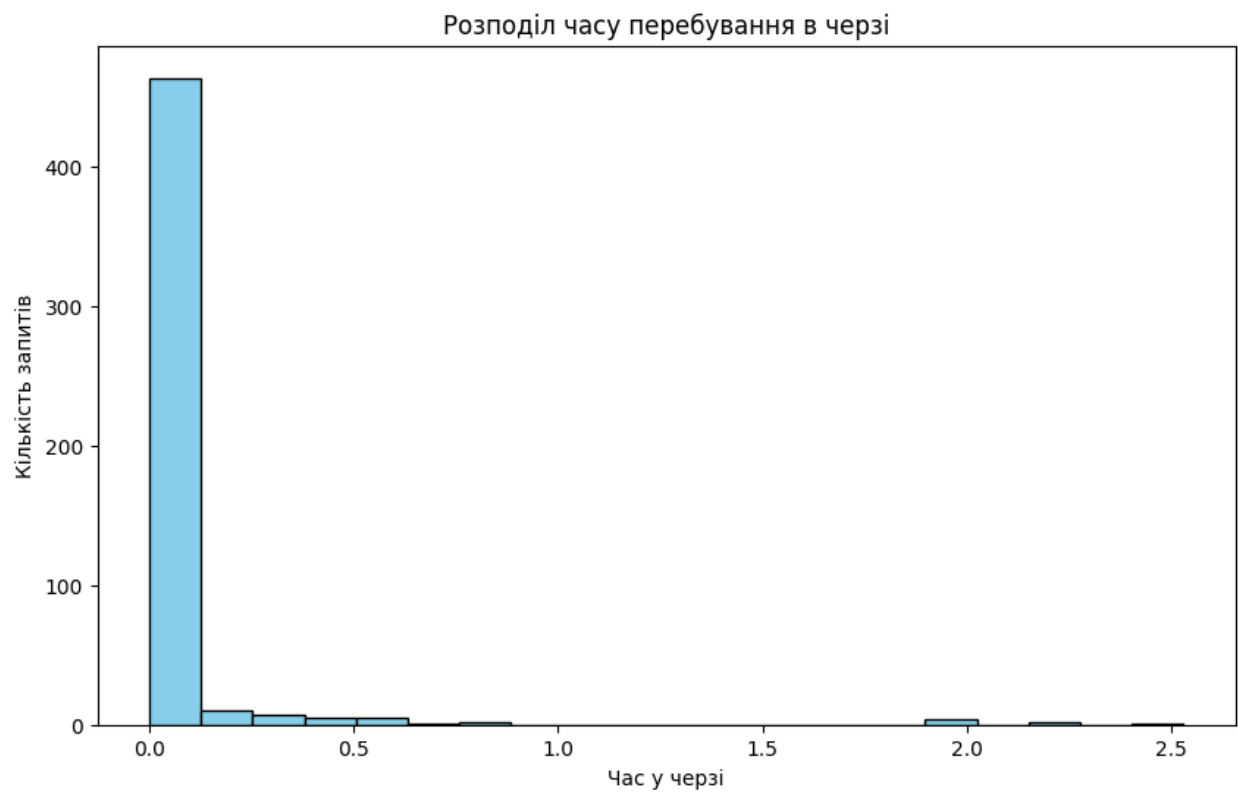
Максимальний час перебування в черзі: 2.83

Середня кількість повторних викликів: 1.00

Максимальна кількість повторних викликів для одного запиту: 1.00

Кількість відмов: 5





Моделювання з інтенсивністю потоку 100х:

Моделювання з інтенсивністю потоку: 100х

Запит 0 обслуговується у 0.00, після очікування 0.00
Запит 1 обслуговується у 0.01, після очікування 0.00
Запит 2 обслуговується у 0.01, після очікування 0.00
Запит 3 обслуговується у 0.50, після очікування 0.46
Запит 4 обслуговується у 0.68, після очікування 0.60
Запит 8 відхилено у 1.14 після 1 повторних спроб.
Запит 9 відхилено у 1.15 після 1 повторних спроб.
Запит 10 відхилено у 1.15 після 1 повторних спроб.
Запит 11 відхилено у 1.20 після 1 повторних спроб.
Запит 12 відхилено у 1.20 після 1 повторних спроб.
Запит 13 відхилено у 1.20 після 1 повторних спроб.
Запит 14 відхилено у 1.21 після 1 повторних спроб.
Запит 15 відхилено у 1.23 після 1 повторних спроб.
Запит 16 відхилено у 1.23 після 1 повторних спроб.
Запит 17 відхилено у 1.23 після 1 повторних спроб.
Запит 18 відхилено у 1.25 після 1 повторних спроб.
Запит 19 відхилено у 1.26 після 1 повторних спроб.
Запит 20 відхилено у 1.26 після 1 повторних спроб.
Запит 21 відхилено у 1.28 після 1 повторних спроб.
Запит 22 відхилено у 1.31 після 1 повторних спроб.
Запит 23 відхилено у 1.31 після 1 повторних спроб.
Запит 24 відхилено у 1.33 після 1 повторних спроб.
Запит 25 відхилено у 1.34 після 1 повторних спроб.
Запит 26 відхилено у 1.35 після 1 повторних спроб.
Запит 27 відхилено у 1.35 після 1 повторних спроб.
Запит 28 відхилено у 1.37 після 1 повторних спроб.

ЗАГАЛЬНА СТАТИСТИКА ЧЕРГИ

Середня довжина черги: 1.60

Максимальна довжина черги: 3.00

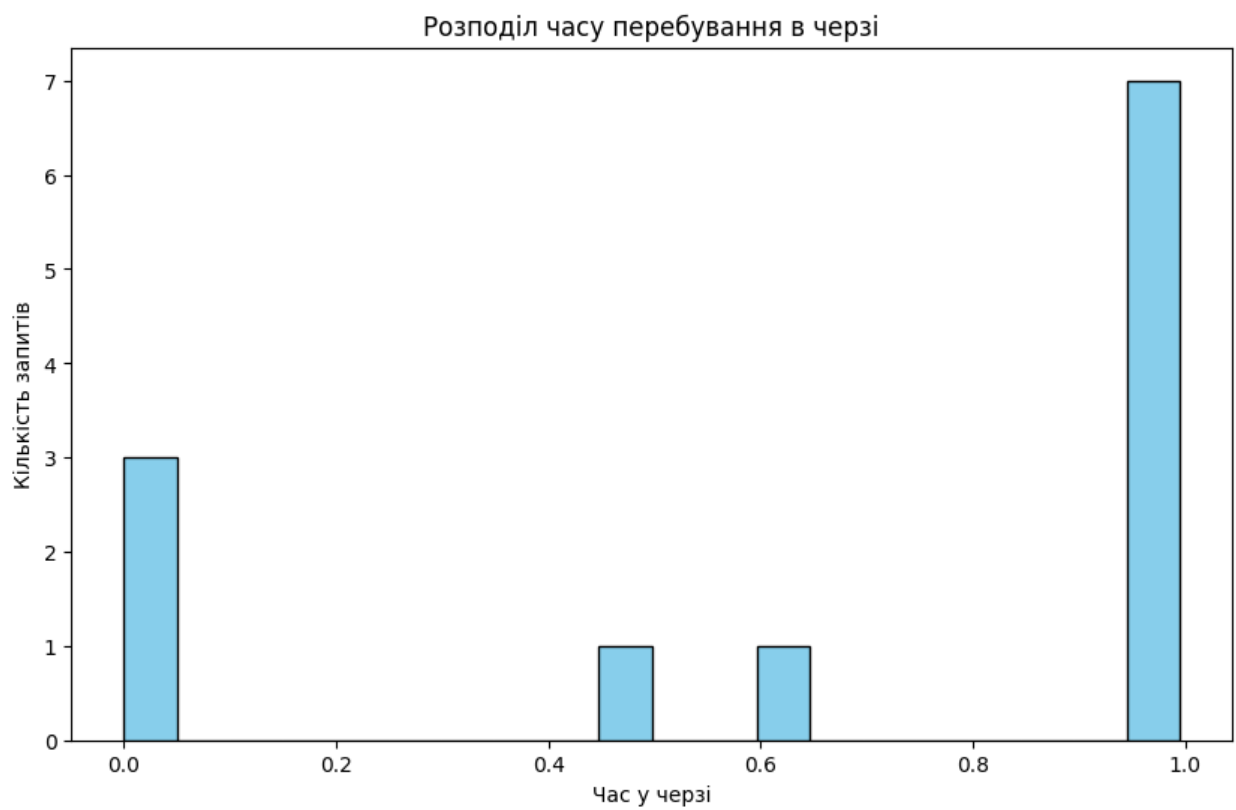
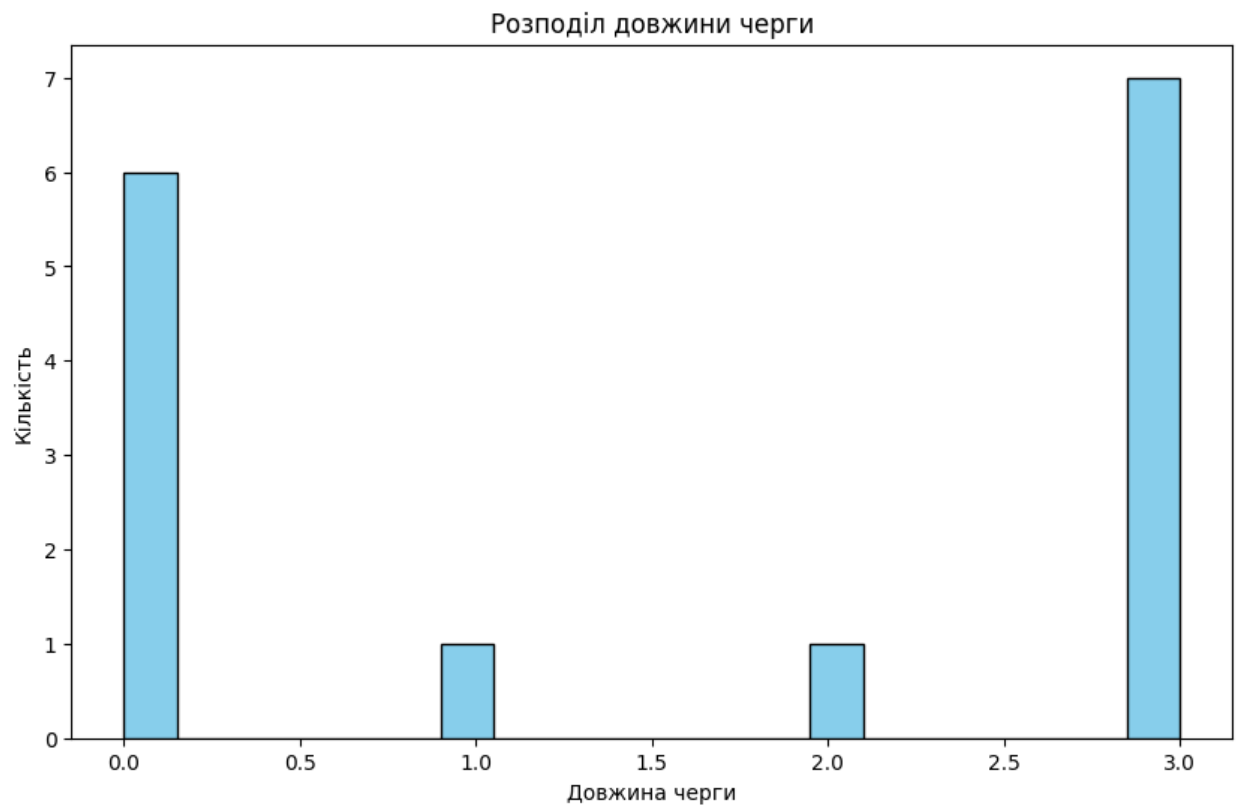
Середній час перебування в черзі: 0.66

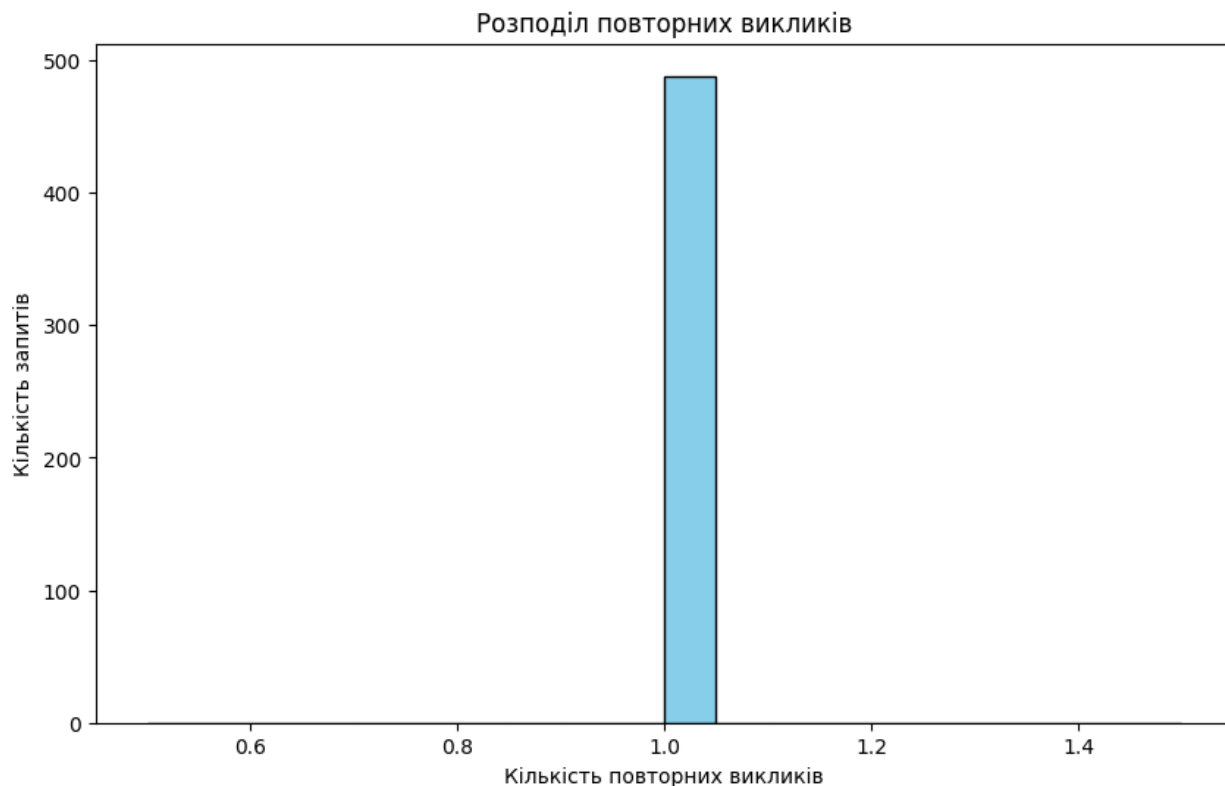
Максимальний час перебування в черзі: 0.99

Середня кількість повторних викликів: 1.00

Максимальна кількість повторних викликів для одного запиту: 1.00

Кількість відмов: 488





Модифікація коду полягала у додаванні детальної статистики щодо черг і повторних викликів. Зокрема, було додано обчислення середньої та максимальної довжини черги, часу перебування заявки в черзі, а також середньої кількості повторних спроб на одну заявку. Окрім того, дослідження було розширене аналізом залежності параметрів від інтенсивності вхідного потоку. Це дало змогу оцінити, як змінюються параметри черги і кількість відмов за збільшення навантаження на систему.

З графічного аналізу (гістограми розподілу черги, часу перебування в черзі та кількості повторних викликів) видно, що при низьких значеннях інтенсивності (наприклад, 1х) черга не накопичується і повторних викликів практично немає. Однак зі збільшенням інтенсивності до 10х і більше кількість відмов і повторних спроб суттєво зростає. Це вказує на те, що система має межу ефективності, за якою накопичення черги і відмови стають значними, що може призвести до значного зниження якості обслуговування.

1. Середня та максимальна довжина черги

При збільшенні інтенсивності вхідного потоку (наприклад, у 2х, 5х, 10х разів) середня та максимальна довжина черги значно зростають. Це свідчить про те, що система швидко досягає межі своєї пропускної здатності, і черга починає переповнюватися. За низької інтенсивності потоку система підтримує коротку чергу, що вказує на достатню кількість ресурсів для обробки запитів у звичайному режимі.

При високих інтенсивностях, таких як 50х або 100х, максимальна довжина черги стає значно більшою за середню, що означає періодичні пікові навантаження, які система не може обробити

вчасно. Це також демонструє, що за значних перевантажень відбувається накопичення запитів до критичних рівнів, що призводить до затримок у роботі системи.

2. Середній і максимальний час перебування заявки в черзі

Із зростанням інтенсивності потоку середній час перебування заявки в черзі також значно збільшується. За низьких значень інтенсивності (1х, 2х) середній час очікування в черзі є незначним, оскільки запити обробляються майже відразу. Проте при збільшенні інтенсивності часу перебування зростає нелінійно, що свідчить про обмеження системи при обробці великої кількості запитів.

Максимальний час перебування в черзі особливо великий при високих інтенсивностях, що показує, наскільки сильно може затримуватися обробка запитів у пікові моменти. Це є критичним для користувачів, оскільки надмірний час очікування погіршує досвід користувача і може призвести до втрати запитів, які так і не дочекалися обслуговування.

3. Середня кількість повторних спроб на одну заявку

При низькій інтенсивності потоку повторні спроби є рідкісними або відсутніми, що вказує на нормальну роботу системи без необхідності повторних викликів. Проте при інтенсивностях від 10х і вище середня кількість повторних спроб на одну заявку починає різко зростати. Це свідчить про те, що користувачі змушені повторно подавати запити через переповненість черги, що може бути ознакою того, що система досягла своєї межі обслуговування.

При дуже високих інтенсивностях, таких як 50х чи 100х, кількість повторних спроб стає значною, що призводить до додаткового навантаження на систему і може створювати ще більші черги. Така ситуація вказує на те, що система не справляється з обробкою запитів, що надходять, що негативно позначається на ефективності обслуговування.

Загальний висновок

Загалом, зростання інтенсивності вхідного потоку призводить до суттєвого збільшення довжини черги, часу перебування заявок у черзі та кількості повторних спроб. Це вказує на потребу в масштабуванні системи або збільшенні її пропускної здатності для запобігання критичним затримкам та відмовам в обслуговуванні при високих навантаженнях.

Висновок: у ході виконання даної лабораторної роботи було досліджено поведінку систем масового обслуговування за допомогою бібліотеки SimPy у Python. Основна задача полягала у моделюванні системи, яка могла б обробляти вхідні запити в умовах обмеженої кількості обслуговуючих каналів і місць у черзі. Для реалізації було обрано варіант СМО з

повторними викликами і обмеженою кількістю місць у черзі, що дозволило змодельовати процес пошуку збоїв у банківській мережі.

У результаті моделювання були отримані показники, які допомагають оцінити продуктивність системи в умовах різного навантаження, зокрема такі, як середня та максимальна довжина черги, кількість відмов, кількість повторних викликів, а також розподіл часу очікування запитів у черзі. Ці показники є важливими для аналізу навантаження і визначення меж ефективної роботи системи.