Практическая работа №1. Анализ одноканальной системы с отказами.

Тема: Колл-центр
*Выполнил: **Гаджиев Саид М3304***

1. Цель работы

Исследовать характеристики одноканальной системы массового обслуживания с отказами (M/M/1/0) методом имитационного моделирования. На примере колл-центра необходимо:

- Разработать имитационную модель системы;
- Провести эксперименты при различных параметрах;
- Сравнить экспериментальные показатели с теоретическими расчётами (формула Эрланга);
- Построить график зависимости вероятности отказа от интенсивности входящего потока заявок (λ) при фиксированном значении интенсивности обслуживания (μ).

2. Задачи

- **Разработка модели:**
  - Моделировать одноканальную систему, где заявки, поступающие в момент занятости канала, теряются.
  - Использовать генератор случайных чисел для моделирования экспоненциального распределения интервалов между заявками и времени обслуживания.
- **Проведение экспериментов:**
  - Запустить модель с заданными параметрами:
    - Интенсивность входящего потока заявок (λ), например, 5 заявок/ед. времени.
    - Интенсивность обслуживания (μ), например, 6 заявок/ед. времени.
    - Время моделирования – 1000 единиц времени.
  - Зафиксировать следующие показатели:
    - Общее число поступивших заявок.
    - Число обслуженных заявок.
    - Число отказанных (потерянных) заявок.
    - Вероятность отказа (отношение потерянных заявок к общему числу).
    - Коэффициент загрузки канала (доля времени, когда канал занят).
- **Анализ результатов:**
  - Сравнить экспериментальные значения с теоретическим расчётом, согласно формуле Эрланга для отказов: $P_{\text{Отказ}} = \frac{\lambda}{\lambda+\mu}$
  - Построить график зависимости вероятности отказа от λ при фиксированном μ.

3. Описание модели и алгоритм работы

3.1 Параметры системы

- **λ (lambda):** интенсивность поступления заявок (заявок/ед. времени). Интервалы между поступлениями моделируются с помощью экспоненциального распределения с параметром λ.
- **μ (mu):** интенсивность обслуживания (заявок/ед. времени). Время обслуживания также моделируется экспоненциально с параметром μ.
- **Время моделирования:** общее время работы системы, например, 1000 единиц времени.

3.2 Особенности модели

- **Одноканальная система с отказами (M/M/1/0):**
  Система имеет один канал для обслуживания. Если заявка поступает, когда канал занят, она немедленно отвергается.
- **Имитация событий:**
  Используются случайные величины для моделирования интервалов между заявками и длительности обслуживания.

3.3 Алгоритм работы модели

1. **Генерация заявок:**
   - Интервалы между поступлениями генерируются экспоненциально с параметром λ.
2. **Обработка заявки:**
   - При поступлении заявки проверяется доступность канала.
   - Если канал свободен, заявка поступает в обслуживание, время которого генерируется экспоненциально с параметром μ.
   - Если канал занят, заявка считается потерянной.
3. **Сбор статистики:**
   - Подсчёт общего числа заявок, числа обслуженных заявок и числа отказов.
   - Вычисление вероятности отказа: $P_{\text{Отказ}} = \frac{\text{число отказов}}{\text{общее число заявок}}$
   - Вычисление коэффициента загрузки канала:
     $\text{Загрузка} = \frac{\text{общее время, когда канал занят}}{\text{время моделирования}}$
4. **Сравнение с теорией и построение графиков:**
   - Рассчитывается теоретическая вероятность отказа по формуле $\frac{\lambda}{\lambda+\mu}$
   - Для вариаций параметра λ при фиксированном μ строится график зависимости экспериментальной и теоретической вероятности отказа.

4. Реализация на Python

Ниже приведён полный пример кода на Python с использованием библиотек **simpy**, **numpy** и **matplotlib**.

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

# Параметры системы
LAMBDA = 5       # интенсивность поступления заявок (заявок/ед. времени)
MU = 6           # интенсивность обслуживания (заявок/ед. времени)
SIM_TIME = 1000  # время моделирования


class CallCenter:
    def __init__(self, env):
        self.env = env
        self.server = simpy.Resource(env, capacity=1)
        self.busy_time = 0.0


def serve(env, call_center, service_time):
    # Запрос сервера и фиксация времени начала обслуживания
    with call_center.server.request() as request:
        yield request
        start_time = env.now
        yield env.timeout(service_time)
        # Добавляем время обслуживания к общей занятости
        call_center.busy_time += env.now - start_time


def arrival_process(env, call_center, lambd, mu, stats):
    while True:
        # Генерация интервала между поступлениями (экспоненциальное
распределение)
        interarrival = np.random.exponential(1.0 / lambd)
        yield env.timeout(interarrival)
        stats['arrivals'] += 1

        # Если сервер свободен, обрабатываем заявку; иначе фиксируем отказ
        if call_center.server.count < call_center.server.capacity:
            stats['served'] += 1
            service_time = np.random.exponential(1.0 / mu)
            env.process(serve(env, call_center, service_time))
        else:
            stats['lost'] += 1
```

```
def run_simulation(lambd, mu, sim_time):
    env = simpy.Environment()
    call_center = CallCenter(env)
    stats = {'arrivals': 0, 'served': 0, 'lost': 0}
    env.process(arrival_process(env, call_center, lambd, mu, stats))
    env.run(until=sim_time)

    # Вычисление коэффициента загрузки и вероятности отказа
    utilization = call_center.busy_time / sim_time
    loss_probability = stats['lost'] / stats['arrivals'] if stats['arrivals'] >
0 else 0
    return stats, utilization, loss_probability


# Проведение эксперимента для исходных параметров
stats, utilization, loss_probability = run_simulation(LAMBDA, MU, SIM_TIME)
print("Статистика при λ =", LAMBDA, "и μ =", MU)
print("Поступило заявок:", stats['arrivals'])
print("Обслужено заявок:", stats['served'])
print("Отказов:", stats['lost'])
print("Коэффициент загрузки:", utilization)
print("Вероятность отказа:", loss_probability)

# Теоретическая вероятность отказа по формуле Эрланга: λ / (λ + μ)
theoretical_loss = LAMBDA / (LAMBDA + MU)
print("Теоретическая вероятность отказа:", theoretical_loss)

# Построение графика зависимости вероятности отказа от интенсивности входящего
потока (λ)
lambdas = np.linspace(1, 10, 50)  # диапазон изменения λ от 1 до 10
exp_loss_probs = []
theor_loss_probs = []

for l in lambdas:
    stats, util, loss_prob = run_simulation(l, MU, SIM_TIME)
    exp_loss_probs.append(loss_prob)
    theor_loss_probs.append(l / (l + MU))

plt.style.use('dark_background')
plt.figure(figsize=(8, 5))
plt.plot(lambdas, exp_loss_probs, label='Экспериментальное')
plt.plot(lambdas, theor_loss_probs, label='Теоретическое', linestyle='--')
plt.xlabel('Интенсивность входящего потока (λ)')
plt.ylabel('Вероятность отказа')
plt.title('Зависимость вероятности отказа от λ при μ = ' + str(MU))
plt.legend()
plt.grid(True)
plt.show()
```

Пояснения к коду

- **Класс `CallCenter`:**
  Моделирует колл-центр с одним сервером. Ведётся подсчёт времени, когда сервер занят (busy_time).
- **Функция `serve`:**
  Обеспечивает обслуживание заявки: после запроса сервера фиксируется время начала обслуживания, затем происходит задержка, имитирующая время обслуживания, и время занятости добавляется к общему времени busy_time.
- **Функция `arrival_process`:**
  Генерирует поступления заявок согласно экспоненциальному распределению. Если сервер свободен (проверка `if call_center.server.count < call_center.server.capacity`), заявка направляется в обслуживание; иначе – фиксируется отказ.
- **Функция `run_simulation`:**
  Запускает имитационную модель, собирает статистику и вычисляет коэффициент загрузки и вероятность отказа. Результаты сравниваются с теоретическим значением $\frac{\lambda}{\lambda+\mu}$
- **Построение графика:**
  Для диапазона значений λ проводится серия экспериментов, результаты эксперимента (вероятность отказа) сравниваются с теоретической зависимостью. График демонстрирует, что с ростом λ вероятность отказа увеличивается.

5. Результаты экспериментов и анализ

5.1 Эксперимент с исходными параметрами (λ = 5, μ = 6)

```
Статистика при λ = 5 и μ = 6
Поступило заявок: 4976
Обслужено заявок: 2728
Отказов: 2248
Коэффициент загрузки: 0.446740036731756 83
Вероятность отказа: 0.4517684887459807
Теоретическая вероятность отказа: 0.45454545454545453
```

- **Экспериментальные показатели:**
  - Общее число поступивших заявок.

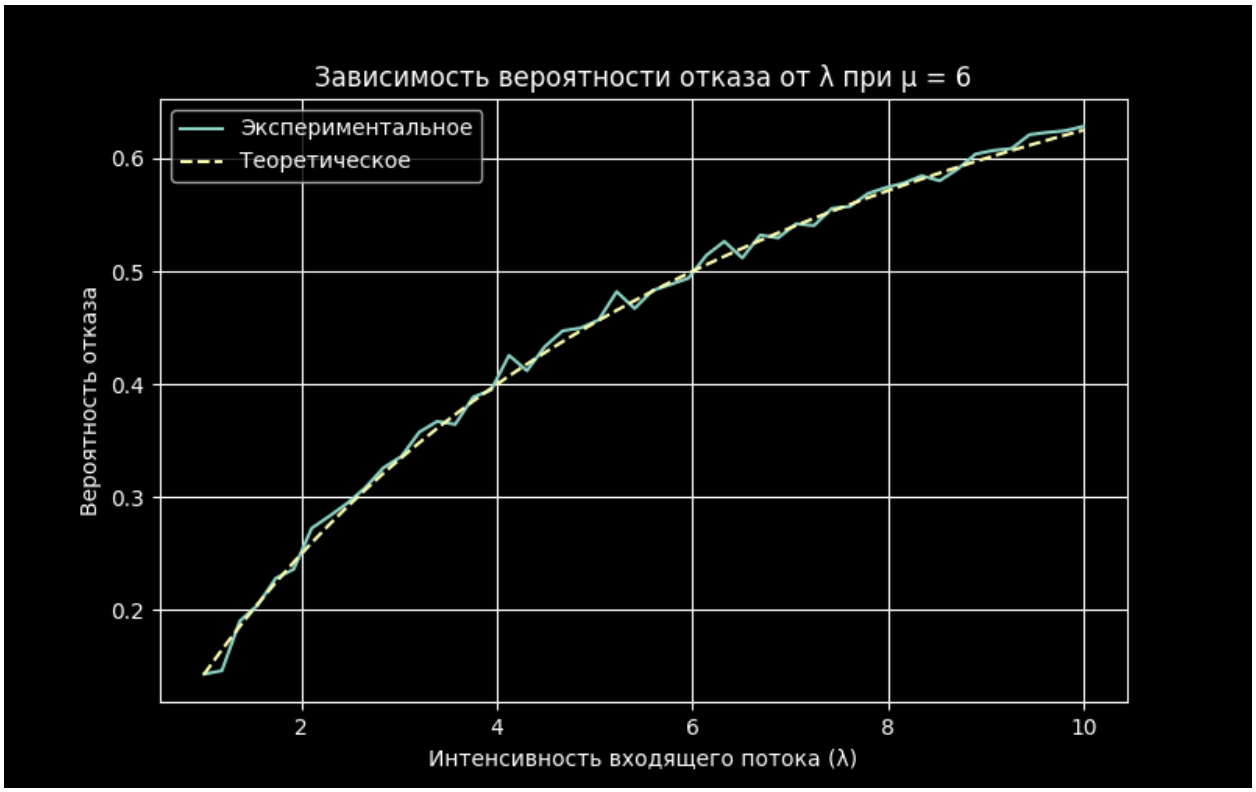
- Число обслуженных заявок.
- Число отказов.
- Коэффициент загрузки канала.
- Экспериментальная вероятность отказа.

- **Сравнение с теорией:**
  Теоретическая вероятность отказа рассчитывается по формуле:

$$P_{\text{Отказ}} = \frac{5}{5+6} \approx 0.4545$$

Полученные экспериментальные показатели можно сопоставить с этим значением.

5.2 Анализ влияния параметра λ



Проведён дополнительный эксперимент, в котором параметр λ варьируется от 1 до 10 при фиксированном μ = 6.

- На графике наблюдается, что с увеличением интенсивности входящего потока заявок вероятность отказа увеличивается, что соответствует теоретическим ожиданиям.
- Экспериментальные и теоретические зависимости совпадают при достаточном числе испытаний.

6. Выводы

- **Влияние параметров системы:**
  При увеличении интенсивности входящего потока (λ) система становится более загруженной, что приводит к увеличению числа отказов. Коэффициент загрузки канала также возрастает, так как сервер работает с большей нагрузкой.
- **Согласованность с теорией:**
  Экспериментальные результаты (вероятность отказа и коэффициент загрузки) близки к теоретическим расчётам по формуле Эрланга:

$$P_{\text{Отказ}} = \frac{\lambda}{\lambda+\mu}$$

- **Практическая значимость:**
  Разработанная модель позволяет оценивать эффективность работы систем, подобных колл-центру, и выявлять узкие места при изменении параметров поступлений или скорости обслуживания. Это может быть полезно для оптимизации реальных систем обслуживания.