Отчет по лабораторной работе №1

Тема: Полнодоступная двухсервисная модель Эрланга с одинаковыми интенсивностями обслуживания

Студент	Ким Реачна
Группа	НПИбд-01-20
Преподаватели:	Доцент Маркова Е.В.
Количество баллов:	______ баллов из 20 б.

Содержание

Список иллюстраций

1 Теоретические сведения

Исследуется сота сети связи емкостью C. Пусть пользователям сети предоставляются услуги двух типов. Запросы в виде двух пуассоновский потоков (ПП) с интенсивностями λ_1, λ_2 поступают в соту. Среднее время обслуживания запросов на предоставление услуг каждого типа μ_1^{-1}, μ_2^{-1} соответственно. Исследуются основные характеристики модели для случая $\mu_1 = \mu_2 = \mu$.

В классификации Башарина-Кендалла $MM|MM|C|0$.

Основные обозначения:

- C - пиковая пропускная способность соты;
- λ_1, λ_2 - интенсивность поступления запросов на предоставление услуги 1, 2-го типа [запросов/ед.вр.];
- μ^{-1} - среднее время обслуживания запроса на предоставление услуги 1, 2-го типа [запросов/ед.вр.];
- ρ_1, ρ_2 - интенсивность предложенной нагрузки,создаваемой запросами на предоставление услуги 1, 2-го типа;
- $X(t)$ - число запросов, обслуживаемых в системе в момент времени $t, t \geq 0$ (случайный процесс (СП), описывающий функционирование системы в момент времени $t, t \geq 0$);
- X - пространство состояний системы;
- n - число обслуживаемых в системе запросов;
- B_1, B_2 - множество блокировок запросов на предоставление услуги 1, 2-го типа;
- S_1, S_2 - множество приема запросов на предоставление услуги 1, 2-го типа.

Схема модели (рис. 1.1):

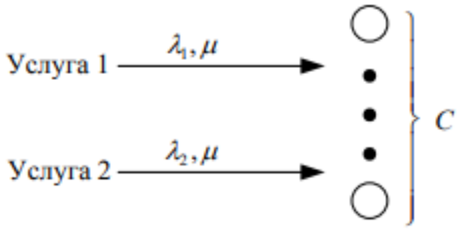

Рис. 1.1: Схема полнодоступной двухсервисной модели Эрланга с одинаковыми интенсивностями обслуживания

Пространство состояний системы (рис. 1.2):

$$X = 0, ..., C, |X| = C + 1 \tag{1.1}$$

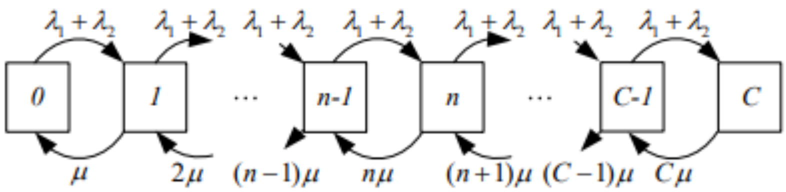

Рис. 1.2: Диаграмма интенсивностей переходов для полнодоступной двухсервисной модели Эрланга с одинаковыми интенсивностями обслуживания

Множество блокировок запросов на предоставление услуги i-типа, i=1,2:

$$B_1 = B_2 = C \tag{1.2}$$

Множество приема запросов на предоставление услуги i-типа, i=1,2:

$$S_i = \bar{B}_i = X \backslash B_i = \{0, 1, , , ., C - 1\} \tag{1.3}$$

Система уравнений глобального баланса (СУГБ):

$$\begin{cases} (\lambda_1+\lambda_2)p_0 = \mu p_1, \\ (\lambda_1+\lambda_2+n\mu)p_n = (\lambda_1+\lambda_2)p_{n-1} + (n+1)\mu p_{n+1}, n=\overline{1,C-1}, \\ C\mu p_C = (\lambda_1+\lambda_2)p_{C-1} \end{cases} \tag{1.4}$$

Система уравнений локального баланса (СУЛБ):

$$(\lambda_1+\lambda_2)p_{n-1} = n\mu p_n, n=\overline{1,C} \tag{1.5}$$

Стационарное распределение вероятностей состояний системы:

$$p_n = \left(\sum_{i=0}^{C} \frac{(\rho_1+\rho_2)^i}{i!}\right)^{-1} . \frac{(\rho_1+\rho_2)^n}{n!}, n=\overline{0,C} \tag{1.6}$$

Доказательство:

Используя СУЛБ, найдем стационарное распределение вероятностей состояний системы $p_n, n=\overline{1,C}$:

$$p_n[C] = p_n = p_{n-1}.\frac{\lambda_1+\lambda_2}{n\mu} = p_{n_1}.\frac{\rho_1+\rho_2}{n} = ... = p_0.\frac{(\rho_1+\rho_2)^n}{n!}, n=\overline{1,C}$$

Для нахождения вероятности p_0 воспользуемся условием нормировки $\sum_{n=0}^{C} p_n = 1$:

$$p_0 = p_0[C] = \left(\sum_{n=0}^{C} \frac{(\rho_1+\rho_2)^n}{n!}\right)^{-1}$$

Основные вероятностные характеристики (ВХ) модели:

- Вероятность блокировки по времени E_i запроса на предоставление услуги i-типа, $i=1,2$:

$$E_1 = E_2 = E[C] = \sum_{n \in B_i} p_n = p_C[C] \tag{1.7}$$

- Вероятность блокировки по вызовам B_i запроса на предоставление услуги i-типа, $i = 1, 2$:

$$B_i = \frac{\lambda_i}{\lambda_1 + \lambda_2}.E[C], \tag{1.8}$$

где $\frac{\lambda_i}{\lambda_1+\lambda_2}$ — вероятность того, что поступит запрос на предоставление услуги i -типа;

- Вероятность блокировки по нагрузке C_i запроса на предоставление услуги i-типа, $i = 1, 2$:

$$C_1 = C_2 = E[C] \tag{1.9}$$

- Среднее число $\bar{N}$ обслуживаемых в системе запросов:

$$\bar{N} = \sum_{n \in X} n p_n. \tag{1.10}$$

2 Численный анализ

Создание программы, реализующей расчет распределения вероятностей, вероятности блокировки, среднего количества обслуженных запросов для любых значений исходных данных. Программа должна выводить на экран:

- значение распределения вероятностей;
- значения вероятностей блокировки;
- значение среднего числа заявок.

Код написан на языке Python в Google Colab:

```
import math #для степенной и факторной функции
import matplotlib.pyplot as plt #для визуализации данных - для построения графика
import numpy as np #для массивных вычислений
```

Рис. 2.1: Импортируйте необходимую библиотеку python

Определение функции p_0 для вычисления вероятности того, что система находится в исходном состоянии:

```
def p_0(C, rho):
    # rho = rho1+rho2
    sum = 0
    for n in range(C+1):
        sum += math.pow(rho, n)/math.factorial(n)
    probab = 1/sum
    return probab
```

Рис. 2.2: Определение функции p_0 для вычисления вероятности

Определение функции p_n для вычисления вероятности того, что система находится в n состоянии:

```
[5]  def p_n(C, n, rho):
         p_nC = ((math.pow(rho, n) / math.factorial(n))) * p_0(C=C, rho=rho)
         return p_nC
```

Рис. 2.3: Определение функции p_n для вычисления вероятности

Определения функция p_C для вычисления вероятности блокировки по нагрузке и по времени $C_1 = C_2 = E$, в которой $n = C$:

```
[6]  def p_C(C, rho):
         # compute E, C1, C2
         return p_n(C=C, n=C, rho=rho)
```

Рис. 2.4: Определение функции p_C для вычисления вероятности

Определение функций mean_request для вычисления среднего число запросов в системе $\bar{N}$:

```
[8]  def mean_request(C, rho):
         expecation = 0
         for n in range(C+1):
             expecation += n * p_n(C=C, n=n, rho=rho)
         return expecation
```

Рис. 2.5: Определение функции для вычисления среднего число

Для расчета основных вероятностных характеристик модели были взяты следующие параметры:

$$C = 100, \mu_1 = \mu_2 = \mu = 0.8, \lambda_1 = 40, \lambda_2 = 60 \tag{2.1}$$

Заметить что:

$\rho_1 = \lambda_1 / \mu_1$
$\rho_2 = \lambda_2 / \mu_2$

```
[9]
C = 100
mu = 0.8
lambda_1 = 60
lambda_2 = 40
RHO = (lambda_1 + lambda_2) / mu

# Incrementation of lambda between 0 and C, with incremental step of 0.4
lambda__ = np.arange(0, lambda_1+lambda_2, 0.4)
```

Рис. 2.6: Основные вероятностные характеристики

Значения стационарного распределения вероятностей в состоянии n и общей суммы:

```
sum_pn = 0
for n in np.arange(C+1):
    pn = p_n(C=C, n=n, rho=RHO)
    sum_pn += pn
    print("Распределения вероятностей нахождения системы в каждом {}-ый состоянии = {}".format(n, pn))
print("Сумма вероятностей в пространстве состояний системы X =", sum_pn)
```

Рис. 2.7: Проверка значения стационарного распределения вероятностей в состоянии n и общей суммы

```
Распределения вероятностей нахождения системы в каждом 88-ый состоянии = 0.007767820202099928
Распределения вероятностей нахождения системы в каждом 89-ый состоянии = 0.01090985983441001
Распределения вероятностей нахождения системы в каждом 90-ый состоянии = 0.015152583103347236
Распределения вероятностей нахождения системы в каждом 91-ый состоянии = 0.020813987779323126
Распределения вероятностей нахождения системы в каждом 92-ый состоянии = 0.02827987470016729
Распределения вероятностей нахождения системы в каждом 93-ый состоянии = 0.038010584274418405
Распределения вероятностей нахождения системы в каждом 94-ый состоянии = 0.050545989726620215
Распределения вероятностей нахождения системы в каждом 95-ый состоянии = 0.06650788121923712
Распределения вероятностей нахождения системы в каждом 96-ый состоянии = 0.08659880367088167
Распределения вероятностей нахождения системы в каждом 97-ый состоянии = 0.11159639648309494
Распределения вероятностей нахождения системы в каждом 98-ый состоянии = 0.14234234245292723
Распределения вероятностей нахождения системы в каждом 99-ый состоянии = 0.1797251798648071
Распределения вероятностей нахождения системы в каждом 100-ый состоянии = 0.2246564748310089
Сумма вероятностей в пространстве состояний системы X = 1.0
```

Рис. 2.8: Значения стационарного распределения вероятностей в состоянии n и общей суммы

Вероятность блокировки по времени E:

```
[11]
# just apply the function p_C defined above
E = p_C(C=C, rho=RHO)
print("Вероятность блокировки по времени E =", E)

Вероятность блокировки по времени E = 0.2246564748310089
```

Рис. 2.9: Вероятность блокировки по времени

Вероятность блокировки по вызовам B_i:

```
[12]
B_1 = (lambda_1 / (lambda_1 + lambda_2)) * E
B_2 = (lambda_2 / (lambda_1 + lambda_2)) * E
print("B1 = {}, B2 = {}".format(B_1, B_2))

B1 = 0.13479388489860533, B2 = 0.08986258993240356
```

Рис. 2.10: Вероятность блокировки по вызовам:

Среднее число обслуживаемых в системе запросов $\bar{N}$:

```
[14]
N = mean_request(C=C, rho=RHO)
print("Среднее число N обслуживаемых в системе запросов =", N)

Среднее число N обслуживаемых в системе запросов = 96.9179406461239
```

Рис. 2.11: Среднее число обслуживаемых запросов

- Построение график зависимости вероятности блокировки от интенсивности поступления запросов на обслуживание

```
[17]
# array_E of p_C when lmabda change
rho = lambda__ / mu
array_E = []
for ro in rho:
    array_E.append(p_C(C=C, rho=ro))
plt.plot(lambda__, array_E, color='blue', label='E=C')
plt.xlabel('Интенсивность поступления запросов')
plt.ylabel('Вероятность блокировки')
plt.title("Вероятность блокировки по времени и вероятность блокировки по нагрузке в зависимости от интенсивности запросов")
plt.legend()
plt.show()
```

Рис. 2.12: Построение график вероятностей блокировки по времени и нагрузке

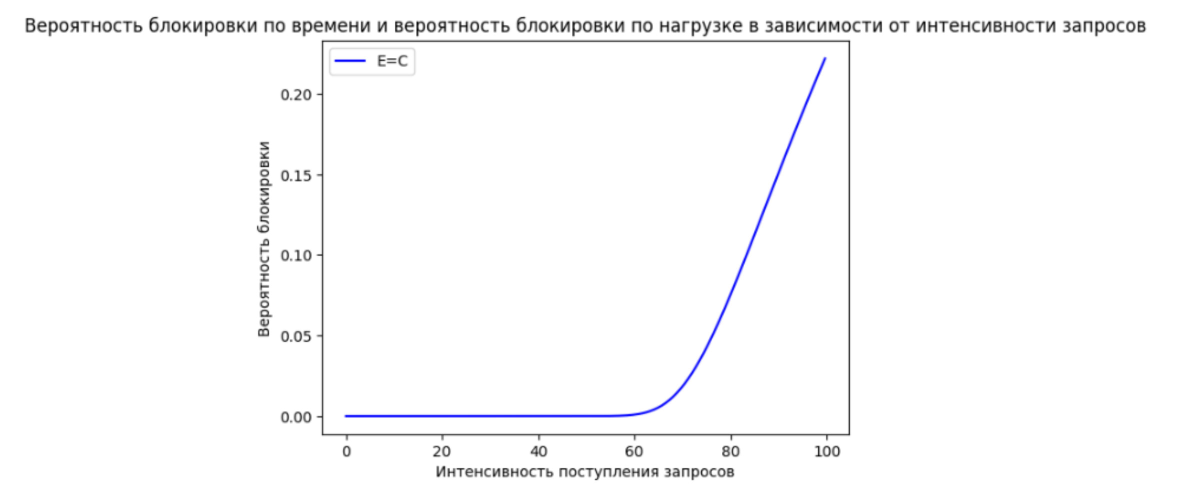

Рис. 2.13: График блокировки по времени и вероятность блокировки по нагрузке в зависимости от интенсивности запросов

```
# array_E of p_C when lmabda change
B_1_array = []
for lam, E in zip(lambda__, array_E):
    B_1_array.append(E* lam/(lam+lambda_2))
B_2_array = []
for lam, E in zip(lambda__, array_E):
    B_2_array.append(E* lambda_2/(lam+lambda_2))
#B12_array = []
#for B1, B2 in zip(B_1_array, B_2_array):
    #B12_array.append(B1+B2)
plt.plot(lambda__, B_1_array, color= 'blue', label="B_1")
plt.plot(lambda__, B_2_array, '',color= 'green', label="B_2")
#plt.plot(lambda__, B12_array, '*',color='blue', label="B1+B2")
plt.title("Вероятность блокировки по вызовам в зависимости от интенсивности запросов")
plt.legend()
plt.xlabel('Интенсивность поступления запросов')
plt.ylabel('Вероятность блокировки')
plt.show()
```

Рис. 2.14: Построение график вероятностей блокировки по вызовам

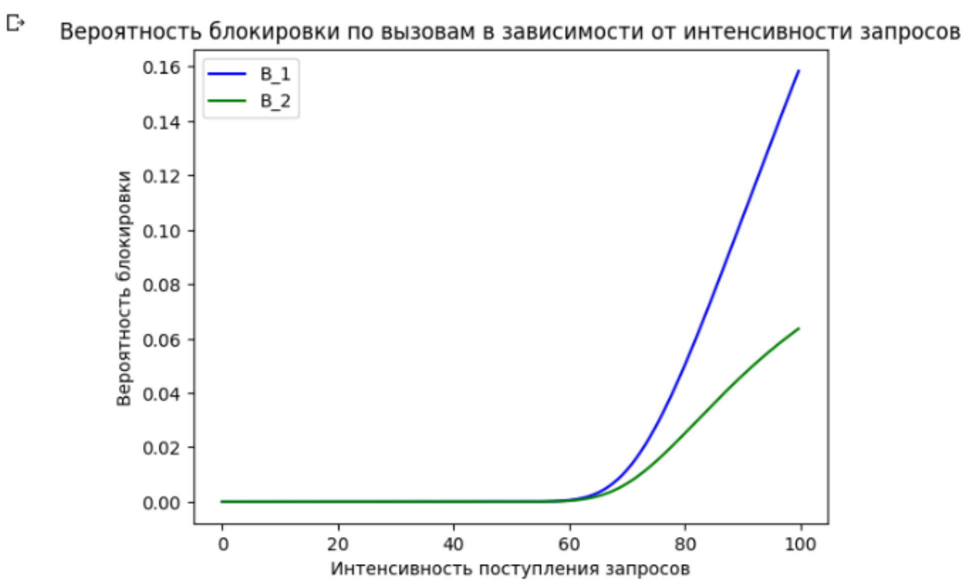

Рис. 2.15: График блокировки по вызовам в зависимости от интенсивности запросов

- Построение график среднего числа обслуживаемых запросов от интенсивности поступления запросов:

```
[23]
mean_request_array = []
for lam in lambda__:
    rho = lam/mu
    mean_request_array.append(mean_request(C=C, rho=rho))
plt.plot(lambda__, mean_request_array, color ='green', label = 'N')
plt.title("Среднее число обслуживаемых в зависимости от интенсивности запросов")
plt.xlabel('Интенсивность поступления запросов')
plt.ylabel('Среднее число запросов')
plt.legend()
plt.show()
```

Рис. 2.16: Построение график среднего числа обслуживаемых в зависимости от интенсивности запросов

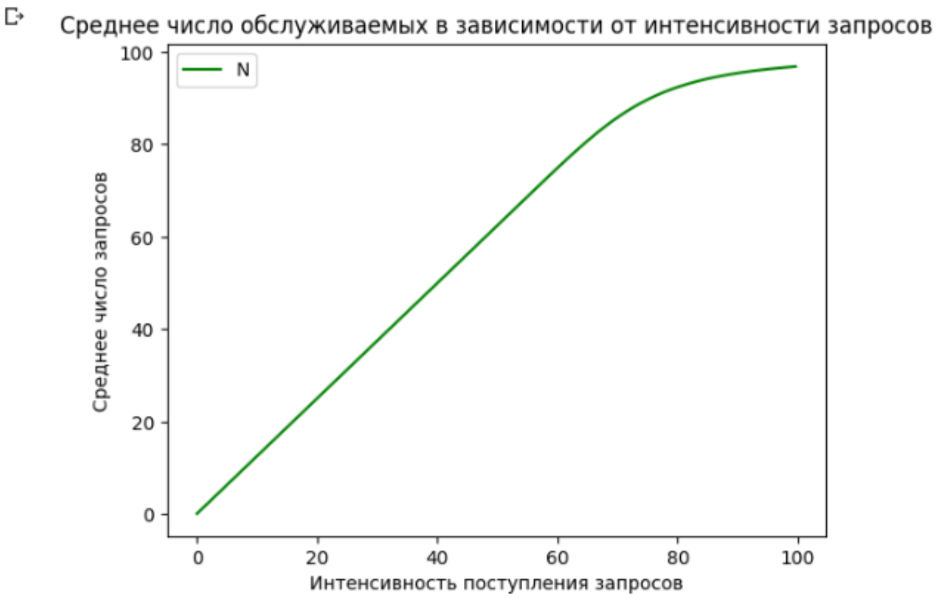

Рис. 2.17: График среднего числа обслуживаемых запросов