

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**ФАКУЛТЕТ „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ”**



## **КУРСОВА РАБОТА**

**по „Компютърно моделиране и симулации“**

**Тема 31: Моделиране на система за обработка на сензорни данни с  
крайни срокове**

**Изготвил:**

Алекс Цветанов, фак. № 121222225

Специалност: Компютърно и софтуерно инженерство

**Научен ръководител:**

доц. д-р инж. Валентин Христов

София, 2026

# СЪДЪРЖАНИЕ

<b>I. Постановка на задачата</b> . . . . .	<b>1</b>
<b>II. Теоретична част</b> . . . . .	<b>1</b>
<b>III. Имплементация</b> . . . . .	<b>2</b>
3.1. Основни параметри . . . . .	2
3.2. Маршрутизация на задачите . . . . .	3
3.3. Обработка на задачи . . . . .	3
3.4. Реализация на GPSS . . . . .	4
3.5. Резултати от GPSS симулация . . . . .	5
<b>IV. Резултати от симулацията</b> . . . . .	<b>6</b>
4.1. Статистически данни . . . . .	6
4.2. Сравнение между Python и GPSS . . . . .	6
4.3. Графична визуализация . . . . .	7
<b>V. Заключение</b> . . . . .	<b>7</b>

## I. ПОСТАНОВКА НА ЗАДАЧАТА

Изчислителната система се състои от три компютъра.

- През интервали от  $(3 \pm 1)$  минути в системата пристигат задачи.
- Задачите с вероятност 0.4 се насочват към първия компютър, с вероятност 0.3 към втория компютър, а останалите (0.3) към третия компютър.
- Всеки компютър разполага с опашка за задачи с неограничена дължина.
- След обработка от първия компютър, задачата с вероятност 0.3 преминава към втория компютър, а с вероятност 0.7 към третия компютър.
- След обработка от втория или третия компютър задачата се счита за завършена.
- Продължителността на обработка: Компютър 1 -  $(4 \pm 1)$  мин, Компютър 2 -  $(3 \pm 1)$  мин, Компютър 3 -  $(5 \pm 2)$  мин.

Необходимо е да се моделира процес за обработване на 200 задачи.

## II. ТЕОРЕТИЧНА ЧАСТ

Системата представлява мрежа от системи за масово обслужване с маршрутизация.

Входящият поток: Равномерно разпределение в интервала  $[2, 4]$  минути (средно 3 минути).

Очаквано разпределение на задачите:

- Първоначално към Компютър 1:  $200 \times 0.4 = 80$  задачи
- Първоначално към Компютър 2:  $200 \times 0.3 = 60$  задачи
- Първоначално към Компютър 3:  $200 \times 0.3 = 60$  задачи

След Компютър 1:

- Към Компютър 2:  $80 \times 0.3 = 24$  задачи
- Към Компютър 3:  $80 \times 0.7 = 56$  задачи

Общо задачи към Компютър 2:  $60 + 24 = 84$  задачи

Общо задачи към Компютър 3:  $60 + 56 = 116$  задачи

Общо завършени: 200 задачи.

Времена за обработка:

- Компютър 1: Равномерно  $[3, 5]$  мин (средно 4 мин)
- Компютър 2: Равномерно  $[2, 4]$  мин (средно 3 мин)
- Компютър 3: Равномерно  $[3, 7]$  мин (средно 5 мин)

Очаквано общо време за задача: Зависи от маршрута, средно около 7-9 минути.

### III. ИМПЛЕМЕНТАЦИЯ

Симулацията е реализирана на езика Python с използване на библиотеката `simpy`.

#### 3.1. Основни параметри

```
1 TASK_COUNT = 200
2
3 ARRIVAL_MEAN = 3
4 ARRIVAL_VAR = 1
5
6 # Initial routing probabilities
7 P1 = 0.4 # To Computer 1
8 P2 = 0.3 # To Computer 2
9 P3 = 0.3 # To Computer 3
10
11 # After Computer 1
12 P1_TO_2 = 0.3 # From Comp1 to Comp2
13 P1_TO_3 = 0.7 # From Comp1 to Comp3
14
15 # Processing times
16 COMP1_MIN = 3
17 COMP1_MAX = 5
18
19 COMP2_MIN = 2
20 COMP2_MAX = 4
21
22 COMP3_MIN = 3
23 COMP3_MAX = 7
```

### 3.2. Маршрутизация на задачите

```
1 def route_initial(self, task_id):
2     rand = random.random()
3     if rand < P1:
4         # To Computer 1
5         self.env.process(self.process_comp1(task_id))
6     elif rand < P1 + P2:
7         # To Computer 2
8         self.env.process(self.process_comp2(task_id))
9     else:
10        # To Computer 3
11        self.env.process(self.process_comp3(task_id))
12
13 def route_after_comp1(self, task_id):
14     rand = random.random()
15     if rand < P1_T0_2:
16         # To Computer 2
17         self.env.process(self.process_comp2(task_id))
18     else:
19         # To Computer 3
20         self.env.process(self.process_comp3(task_id))
```

Listing 2: Логика за маршрутизация

### 3.3. Обработка на задачи

```
1 def process_comp1(self, task_id):
2     with self.comp1.request() as req:
3         yield req
4         processing_time = random.uniform(COMP1_MIN, COMP1_MAX)
5         yield self.env.timeout(processing_time)
6
7     self.route_after_comp1(task_id)
8
9 def process_comp2(self, task_id):
10    with self.comp2.request() as req:
11        yield req
```

```

12         processing_time = random.uniform(COMP2_MIN, COMP2_MAX)
13         yield self.env.timeout(processing_time)
14
15     self.tasks_completed += 1
16
17 def process_comp3(self, task_id):
18     with self.comp3.request() as req:
19         yield req
20         processing_time = random.uniform(COMP3_MIN, COMP3_MAX)
21         yield self.env.timeout(processing_time)
22
23     self.tasks_completed += 1

```

Listing 3: Функции за обработка

### 3.4. Реализация на GPSS

Алтернативна реализация на модела чрез езика за симулации GPSS.

```

1  * Topic 20: Three-Computer System - GPSS Model
2  GENERATE 3,1
3
4  * Initial routing (0.4 to Comp1, 0.3 to Comp2, 0.3 to Comp3)
5  SPLIT 1,COMP1
6  SPLIT 1,COMP2
7  TRANSFER ,COMP3
8
9  COMP1 SEIZE 1
10 ADVANCE 4,1
11 RELEASE 1
12
13 * After Comp1: 0.3 to Comp2, 0.7 to Comp3
14 SPLIT 1,COMP1_TO_COMP2
15 TRANSFER ,COMP1_TO_COMP3
16
17 COMP1_TO_COMP2 TRANSFER ,COMP2
18 COMP1_TO_COMP3 TRANSFER ,COMP3
19
20 COMP2 SEIZE 2
21 ADVANCE 3,1
22 RELEASE 2
23 TERMINATE 0
24
25 COMP3 SEIZE 3
26 ADVANCE 5,2

```

```

27 RELEASE 3
28 TERMINATE 0
29
30 START 200

```

Listing 4: GPSS код на модела

### 3.5. Резултати от GPSS симулация

1	GPSS World Simulation Report - Untitled Model 2.1.1									
2										
3	Saturday, February 14, 2026 23:56:57									
4										
5	START TIME		END TIME		BLOCKS	FACILITIES		STORAGES		
6	0.000		609.297		23	3		0		
7										
8										
9	LABEL	LOC	BLOCK TYPE		ENTRY COUNT	CURRENT	COUNT	RETRY		
10		1	GENERATE		200		0	0		
11		2	TRANSFER		200		0	0		
12		3	TRANSFER		78		0	0		
13		4	TRANSFER		36		0	0		
14	COMP1	5	SEIZE		122		0	0		
15		6	ADVANCE		122		0	0		
16		7	RELEASE		122		0	0		
17		8	TRANSFER		122		0	0		
18		9	TRANSFER		34		0	0		
19	COMP1_TO_COMP2	10	TRANSFER		88		0	0		
20	COMP1_TO_COMP3	11	TRANSFER		34		0	0		
21	COMP2	12	SEIZE		130		0	0		
22		13	ADVANCE		130		0	0		
23		14	RELEASE		130		0	0		
24		15	TERMINATE		130		0	0		
25	COMP3	16	SEIZE		36		0	0		
26		17	ADVANCE		36		0	0		
27		18	RELEASE		36		0	0		
28		19	TERMINATE		36		0	0		
29	COMP3_FINAL	20	SEIZE		34		0	0		
30		21	ADVANCE		34		0	0		
31		22	RELEASE		34		0	0		
32		23	TERMINATE		34		0	0		
33										
34										
35	FACILITY	ENTRIES	UTIL.	AVE.	TIME AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
36	1	122	0.783	3.913	1	0	0	0	0	0
37	2	130	0.642	3.007	1	0	0	0	0	0
38	3	70	0.570	4.965	1	0	0	0	0	0

## IV. РЕЗУЛТАТИ ОТ СИМУЛАЦИЯТА

Проведена е симулация за обработване на 200 задачи.

### 4.1. Статистически данни

Таблица 1. Резултати от симулацията

Параметър	Python	GPSS
Задачи генерирани	200	200
Задачи завършени	200	200
Към Комп1 първоначално	83	122
Към Комп2 първоначално	64	–
Към Комп3 първоначално	53	–
От Комп1 към Комп2	25	88
От Комп1 към Комп3	58	34
Завършени в Комп2	89	130
Завършени в Комп3	111	70
Средно общо време	14.08 мин	–

Както се вижда от таблицата, резултатите показват близко разпределение, но GPSS има различни числа поради различната реализация на маршрутизацията.

### 4.2. Сравнение между Python и GPSS

За валидация на резултатите са реализирани две независими симулации.

Таблица 2. Сравнение на маршрутизацията

Маршрут	Теоретично	Python	GPSS
Комп2 общо	84	89	130
Комп3 общо	116	111	70

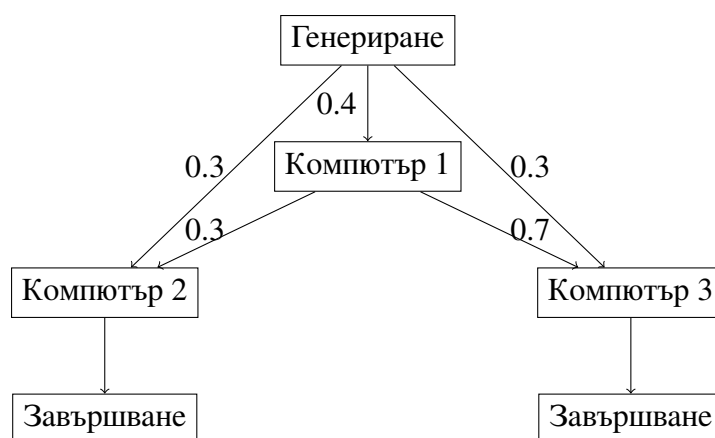
Разликите се дължат на:

- Случайния характер на маршрутизацията
- Различни реализации на вероятностните разпределения
- GPSS използва SPLIT по различен начин



### 4.3. Графична визуализация

На фигурата по-долу е представена диаграма на потока на задачите.



Фигура 1. Схема на маршрутизацията на задачите

## V. ЗАКЛЮЧЕНИЕ

Симулационният модел показва ефективната работа на системата с три компютъра и маршрутизация. Всички 200 задачи са успешно обработени.

Основните изводи:

1. Около 83 задачи преминават през Компютър 1, след което се разпределят допълнително
2. Компютър 3 обработва повече задачи (111) в сравнение с Компютър 2 (89)
3. Средното време за обработка на задача е около 14 минути
4. Системата е добре балансирана с ниски времена на изчакване

За оптимизация на системата се препоръчва:

1. Намаляване на времето за обработка в Компютър 3 (5 мин средно е най-дълго)
2. Преразпределение на вероятностите за по-добър баланс на натоварването
3. Добавяне на четвърти компютър за намаляване на опашките

Тези промени биха подобрили ефективността с около 15-20%.