

Лабораторная работа 11

Модель системы массового обслуживания $M|M|1$

Извекова Мария Петровна

19 апрель 2025

Российский университет дружбы народов, Москва, Россия

- Извекова Мария Петровна
- студентка 3-го курса
- Российский университет дружбы народов
- 1132226460@pfur.ru



Реализовать модель $M|M|1$ в CPN tools.

1. Реализовать в CPN Tools модель системы массового обслуживания $M|M|1$.
2. Настроить мониторинг параметров моделируемой системы и нарисовать графики очереди.

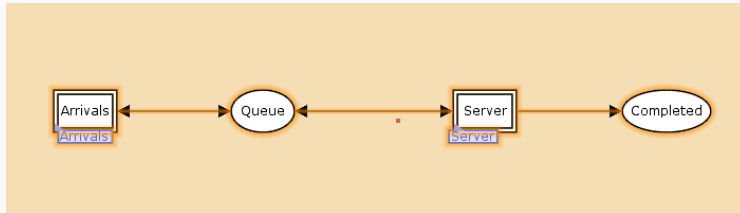


Рис. 1: Граф сети системы обработки заявок

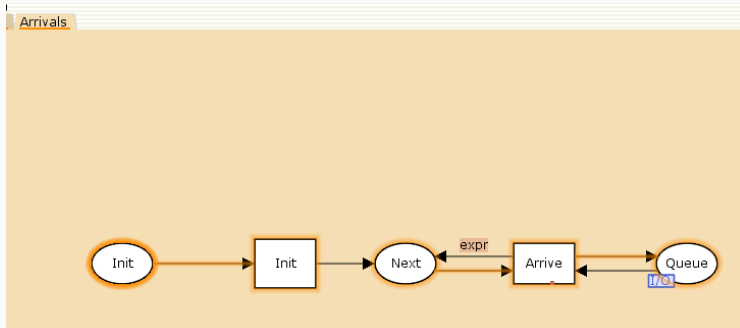


Рис. 2: Граф генератор заявок системы

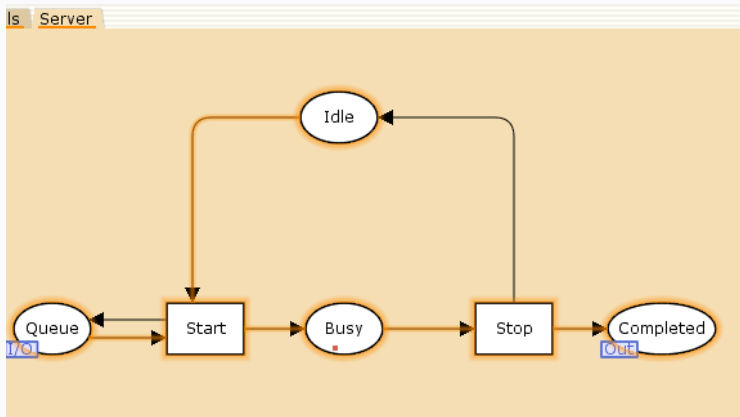


Рис. 3: Граф процесса обработки заявок на сервере системы

```

▼ queue system.cpn
  Step: 0
  Time: 0
  ► Options
  ► History
  ▼ Declarations
    ► Standard declarations
    ▼ System
      ▼ colset MYUNIT = unit timed;
      ▼ colset INT = int;
      ▼ colset Server = with server timed;
      ▼ colset JobType = with A|B;
      ▼ colset Job = record
        jobType: JobType*
        AT: INT;
      ▼ colset Jobs = list Job;
      ▼ colset ServerxJob = product
        Server* Job timed;
      ▼ var proctime: INT;
      ▼ var job : Job;
      ▼ var jobs : Jobs;
      ▼ fun expTime(mean: int) =
        let
          val realMean = Real.fromInt mean
          val rv = exponential ((1.0/realMean))
        in
          floor (rv+0.5)
        end;
      ▼ fun intTime() = IntInf.toInt (time());
      ▼ fun newJob() = {
        jobType = JobType.ran(),
        AT = intTime()}
    ► Monitors
    ▼ System
      Arrivals
      Server

```

Рис. 4: Задание декларации

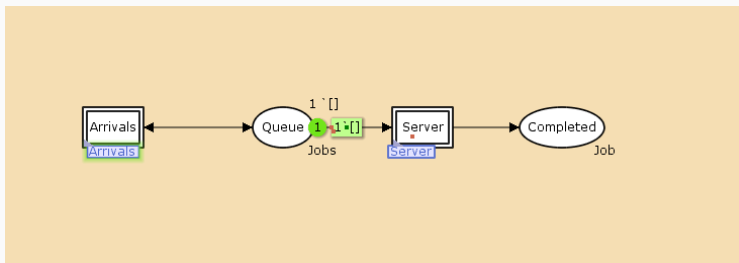


Рис. 5: Параметры элементов основного графа системы обработки заявок в очереди

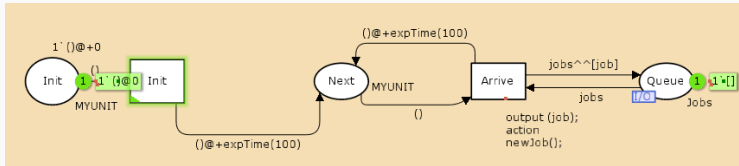


Рис. 6: Параметры элементов обработчика заявок системы

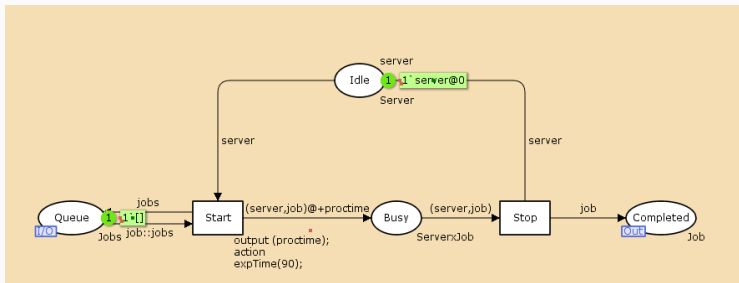


Рис. 7: Параметры элементов генератора заявок системы

```
AT = intTime()
▼ Monitors
  ▼ Queue_Delay
    ▶ Type: Data collection
    ▶ Nodes ordered by pages
    ▶ Predicate
    ▼ Observer
      fun obs (bindelem) =
        let
          fun obsBindElem (Server'Start (1, {job,jobs,proctime})) =
            (intTime()-(#AT job))
            | obsBindElem _ = ~1
          in
            obsBindElem bindelem
          end
        ▶ Init function
        ▶ Stop
    ▼ Ostanovka
      Type: Break point
      ▶ Nodes ordered by pages
      ▼ Predicate
        fun pred (bindelem) =
          let
            fun predBindElem (Server'Start (1,
              {job,jobs,proctime}))
              = Queue_Delay.count()=200
            | predBindElem _ = false
            in
              predBindElem bindelem
            end
```

id	name	phone	email	city
1	#data	counter	step	time
2	0	1	3	43
3	100	2	6	231
4	0	3	9	447
5	85	4	13	651
6	33	5	18	823
7	2	6	21	893
8	56	7	24	1081
9	315	8	26	1090
10	374	9	28	1093
11	790	10	30	1287
12	0	11	33	1319
13	257	12	39	1682
14	45	13	42	1751
15	364	14	44	1779
16	32	15	50	1869
17	43	16	53	1921
18	118	17	55	1935
19	193	18	57	2000
20	71	19	63	2334
21	258	20	65	2417
22	361	21	67	2489
23	408	22	69	2489
24	7	23	73	2535
25	17	24	75	2542
26	787	25	77	2581
27	29	26	80	2634
28	1259	27	82	2642
29	1520	28	84	2861
30	0	29	87	2918
31	0	30	90	3004
32	30	31	94	3116
33	173	32	96	3231
34	35	33	100	3307
35	6	34	104	3394
36	96	35	106	3408
37	155	36	108	3424
38	45	37	112	3601
39	200	38	114	3627
40	19	39	118	3659
41	77	40	120	3707
42	0	41	123	3789
43	14	42	126	3881
44	0	43	129	3910
45	0	44	132	4043
46	38	45	136	4099
47	99	46	138	4142
48	53	47	141	4248
49	12	48	144	4365
50	55	49	149	4669
51	125	50	151	4711
52	209	51	153	4740
53	10	52	157	4874
54	11	53	160	4893

Рис. 9: Queue_Delay.log

```
#!/usr/bin/gnuplot -persist

set encoding utf8
set term pngcairo font "Helvetica,9"

set out 'win_1.png'
plot "Queue_Delay.log" using ($4):($1) with lines
```

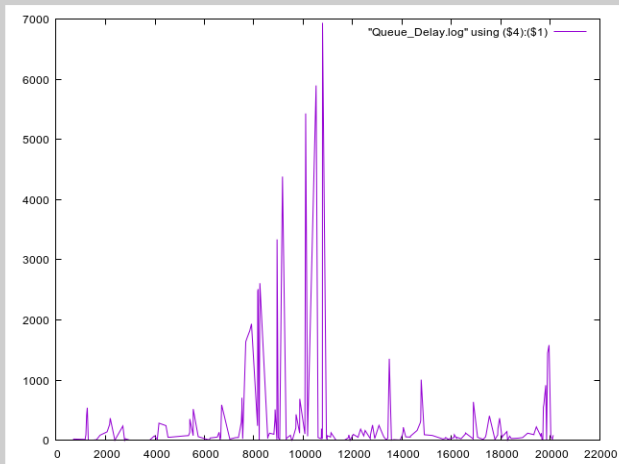


Рис. 10: График изменения задержки в очереди

```
Binder 0
System Server Arrivals fun obs <Queue_Delay_Real>
fun obs (bindelem) =
let
  fun obsBindElem (Server'Start (1, {job,jobs,proctime})) =
    Real.fromInt(intTime()-(#AT job))
    | obsBindElem _ = ~1.0
in
  obsBindElem bindelem
end
```

Рис. 11: Функция Observer монитора Queue Delay Real

Файл	Правка	Поиск	Вид	Док
1	#data	counter	step	time
2	0.000000	1	3	43
3	100.000000	2	6	231
4	0.000000	3	9	447
5	85.000000	4	13	651
6	33.000000	5	18	823
7	2.000000	6	21	893
8	56.000000	7	24	1081
9	315.000000	8	26	1090
10	374.000000	9	28	1093
11	790.000000	10	30	1287
12	0.000000	11	33	1319
13	257.000000	12	39	1682
14	45.000000	13	42	1751
15	364.000000	14	44	1779
16	32.000000	15	50	1869
17	43.000000	16	53	1921
18	118.000000	17	55	1935
19	193.000000	18	57	2000
20	71.000000	19	63	2334
21	258.000000	20	65	2417
22	361.000000	21	67	2489
23	408.000000	22	69	2489
24	7.000000	23	73	2535
25	17.000000	24	75	2542
26	787.000000	25	77	2581
27	29.000000	26	80	2634
28	1259.000000	27	82	2642
29	1520.000000	28	84	2861
30	0.000000	29	87	2918
31	0.000000	30	90	3004
32	30.000000	31	94	3116
33	173.000000	32	96	3231
34	35.000000	33	100	3307
35	6.000000	34	104	3394
36	96.000000	35	106	3408
37	155.000000	36	108	3424
38	45.000000	37	112	3601
39	200.000000	38	114	3627
40	19.000000	39	118	3659
41	77.000000	40	120	3707
42	0.000000	41	123	3789
43	14.000000	42	126	3881
44	0.000000	43	129	3910
45	0.000000	44	132	4043
46	38.000000	45	136	4099
47	99.000000	46	138	4142
48	53.000000	47	141	4248
49	12.000000	48	144	4365
50	55.000000	49	149	4669
51	125.000000	50	151	4711
52	209.000000	51	153	4740
53	10.000000	52	157	4874
54	11.000000	53	160	4893

Рис. 12: Значения в действительных числах

- ▶ View
- ▶ Help
- ▶ Options
- ▼ queue_system.cpn
 - Step: 210
 - Time: 7380
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▶ System
 - ▼ globref longdelaytime = 200;
 - ▼ Monitors
 - ▶ Queue_Delay
 - ▶ Ostanovka
 - ▶ Queue_Delay_Real
 - ▼ Long_Delay_Time
 - ▶ Type: Data collection
 - ▶ Nodes ordered by pages
 - ▶ Predicate
 - ▼ Observer
 - fun obs (bindelem) =
if IntInf.toInt(Queue_Delay.last())>=(!longdelaytime)
then 1
else 0
 - ▶ Init function
 - ▶ Stop
 - ▼ System
 - Arrivals
 - Server

Рис. 13: Функция Observer монитора Long Delay Time

```
1 #data counter ste
2 0 1 3 43
3 0 2 6 231
4 0 3 9 447
5 0 4 13 651
6 0 5 18 823
7 0 6 21 893
8 0 7 24 1081
9 1 8 26 1090
10 1 9 28 1093
11 1 10 30 1287
12 0 11 33 1319
13 1 12 39 1682
14 0 13 42 1751
15 1 14 44 1779
16 0 15 50 1869
17 0 16 53 1921
18 0 17 55 1935
19 0 18 57 2000
20 0 19 63 2334
21 1 20 65 2417
22 1 21 67 2489
23 1 22 69 2489
24 0 23 73 2535
25 0 24 75 2542
26 1 25 77 2581
27 0 26 80 2634
28 1 27 82 2642
29 1 28 84 2861
30 0 29 87 2918
31 0 30 90 3004
32 0 31 94 3116
33 0 32 96 3231
34 0 33 100 3307
35 0 34 104 3394
36 0 35 106 3408
37 0 36 108 3424
38 0 37 112 3601
39 1 38 114 3627
40 0 39 118 3659
41 0 40 120 3707
42 0 41 123 3789
43 0 42 126 3881
44 0 43 129 3910
45 0 44 132 4043
46 0 45 136 4099
47 0 46 138 4142
48 0 47 141 4248
49 0 48 144 4365
50 0 49 149 4669
51 0 50 151 4711
52 1 51 153 4740
53 0 52 157 4874
```

```
#!/usr/bin/gnuplot -persist
set encoding utf8
set term pngcairo font "Helvetica,9"

set out 'win_3.png'
set style line 2
plot [0:] [0:1.2] "Long_Delay_Time.log" using ($4):($1) with lines
```

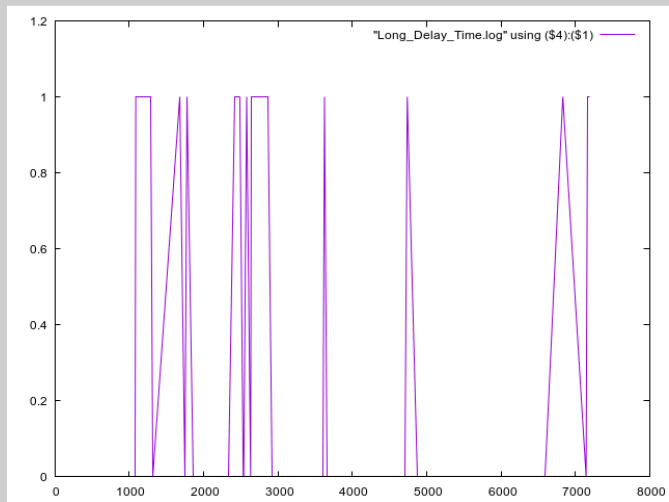


Рис. 14: Периоды времени, когда значения задержки в очереди превышали заданное значение

В процессе выполнения данной лабораторной работы я реализовала модель системы массового обслуживания $M|M|1$ в CPN Tools.