

Презентация по лабораторной работе №11

Имитационное моделирование

Екатерина Канева, НФИбд-02-22

18 апреля 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Канева Екатерина Павловна
- студент группы НФИбд-02-22
- Российский университет дружбы народов
- 1132222004@rudn.ru
- <https://nevseros.github.io/ru/>

Вводная часть

Построить модель $M|M|1$ в CPN Tools.

1. Построить модель $M|M|1$.
2. Выполнить мониторинг параметров модели, построить графики.

Выполнение работы

Сначала я построила лист System:

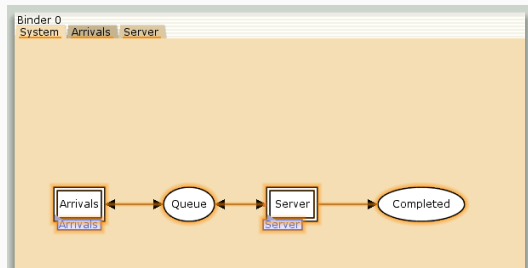


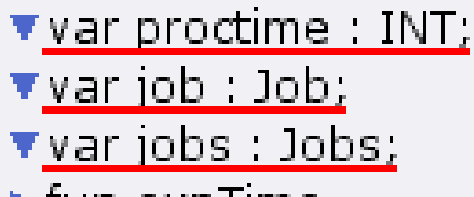
Рис. 1: Лист System.

Задала множества цветов:

```
▼ System  
  ▼ colset UNIT = 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;
```

Рис. 2: Множества цветов.

Задала переменные:

A screenshot of a code editor showing three lines of code. Each line is preceded by a blue downward-pointing arrow. The text of each line is underlined in red. The lines are: 'var proctime : INT;', 'var job : Job;', and 'var jobs : Jobs;'.

```
▼ var proctime : INT;  
▼ var job : Job;  
▼ var jobs : Jobs;
```

Рис. 3: Переменные.

Задала функции модели:

```
▼ 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()}
```

Рис. 4: Функции.

Построила граф на листе Arrivals:

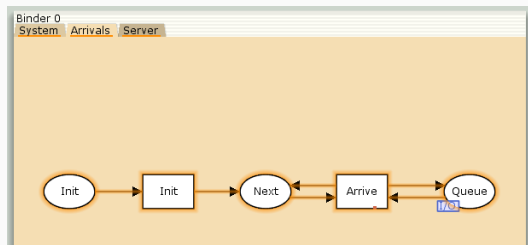


Рис. 5: Лист Arrivals.

Построила граф на листе Server:

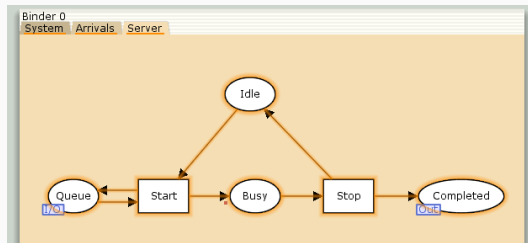


Рис. 6: Лист Server.

Потом на каждом из листов я задала параметры. Для листа System:

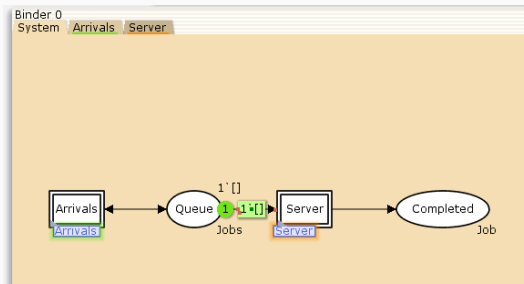


Рис. 7: Лист System после задания параметров на графе.

Для листа Arrivals:

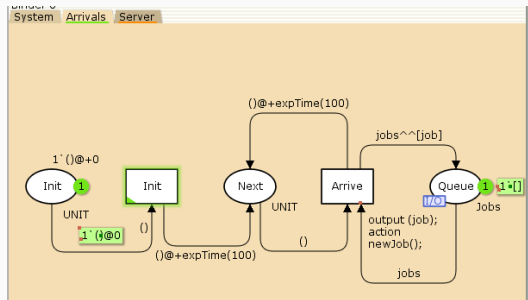


Рис. 8: Лист Arrivals после задания параметров на графе.

Для листа Server:

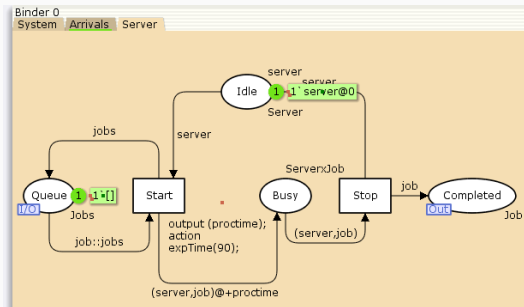


Рис. 9: Лист Server после задания параметров на графе.

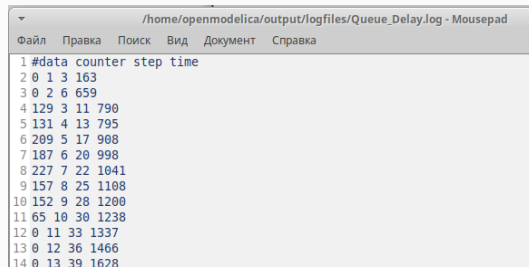
С помощью палитры Monitoring установила точку останова на переход Start, новый монитор назвала Ostanovka. Функцию Predicate отредактировала, ниже изменённая часть:

```
fun predBindElem (Server'Start (1, {job,jobs,proctime}))  
    = Queue_Delay.count( )=200
```

Далее на Start установила Data Coll, назвала новый монитор Queue Delay, отредактировала функцию Observer:

```
fun obsBindElem (Server'Start (1, {job, jobs, proctime}))  
  = (intTime() - (#AT job))
```

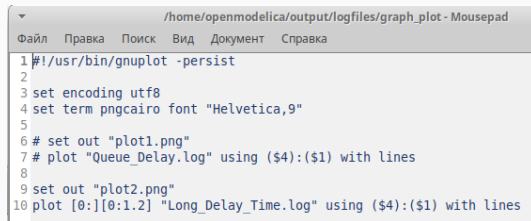
Запустила моделирование, получила на выводе файл Queue_Delay.log:



```
1 #data counter step time
2 0 1 3 163
3 0 2 6 659
4 129 3 11 790
5 131 4 13 795
6 209 5 17 908
7 187 6 20 998
8 227 7 22 1041
9 157 8 25 1108
10 152 9 28 1200
11 65 10 30 1238
12 0 11 33 1337
13 0 12 36 1466
14 0 13 39 1628
```

Рис. 10: Содержимое Queue_Delay.log.

Написала скрипт, подходящий для построения обоих графиков, нужных в лабораторной:

A screenshot of a text editor window titled "/home/openmodelica/output/logfiles/graph_plot - Mousepad". The window has a menu bar with "Файл", "Правка", "Поиск", "Вид", "Документ", and "Справка". The text area contains a gnuplot script with line numbers 1 through 10. The script sets encoding to utf8, sets the terminal to pngcairo with Helvetica font, and defines two plots: plot1.png and plot2.png. Plot 1 uses "Queue_Delay.log" and plot 2 uses "Long_Delay_Time.log", both using columns (\$4) and (\$1) with lines.

```
1#!/usr/bin/gnuplot -persist
2
3set encoding utf8
4set term pngcairo font "Helvetica,9"
5
6# set out "plot1.png"
7# plot "Queue_Delay.log" using ($4):($1) with lines
8
9set out "plot2.png"
10plot [0:][0:1.2] "Long_Delay_Time.log" using ($4):($1) with lines
```

Рис. 11: Скрипт для построения графиков.

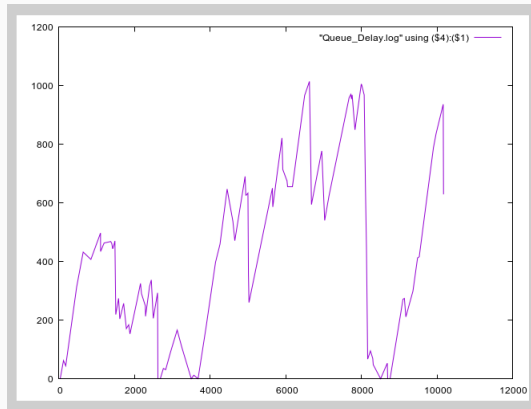


Рис. 12: График значений задержки в очереди.

Далее на Start снова установила Data Coll, назвала новый монитор Queue Delay Real, отредактировала функцию Observer:

```
fun obsBindElem (Server'Start (1, {job, jobs, proctime}))  
    = Real.fromInt(intTime() - (#AT job))
```

Далее на Start снова установила Data Coll, назвала новый монитор Long Delay Time, отредактировала функцию Observer:

```
fun obs (bindelem) =  
  if IntInf.tiInt(Queue_Delay.last())>=(!longdelaytime)  
    then 1  
    else 0
```

Задала глобальную переменную, используемую в прошлой функции Observer:

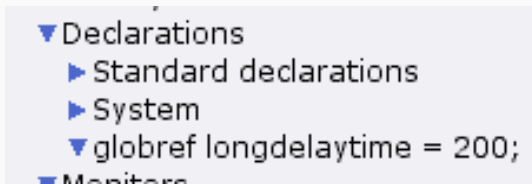


Рис. 13: Глобальная переменная longdelaytime.

График превышения периодов задержки

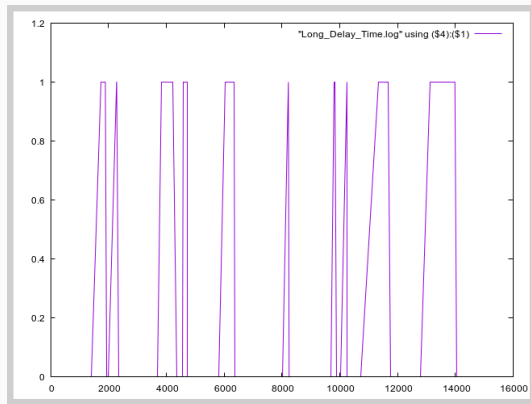


Рис. 14: График периодов превышения задержки.

Заключение

Построили модель $M|M|1$ в CPN Tools.