Лабораторная работа №11

Имитационное моделирование

Екатерина Канева, НФИбд-02-22

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Построить модель M|M|1 в CPN Tools.

2 Задание

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

3 Выполнение лабораторной работы

Модель состояла из 3 листов. Сначала я построила лист System (рис. 3.1):

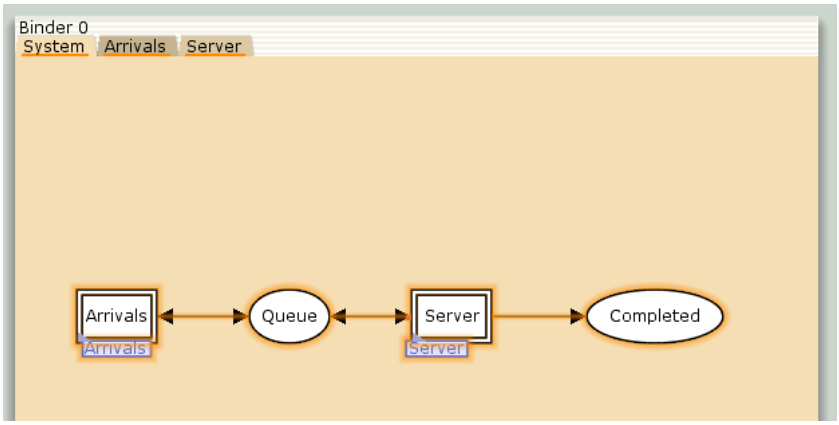

Рис. 3.1: Лист System.

Далее я задала декларации системы: множества цветов (рис. 3.2), переменные (рис. 3.3), функции (рис. 3.4) модели.

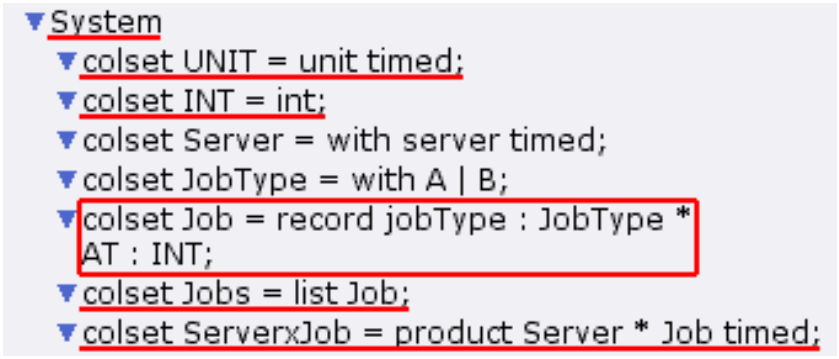

Рис. 3.2: Множества цветов.

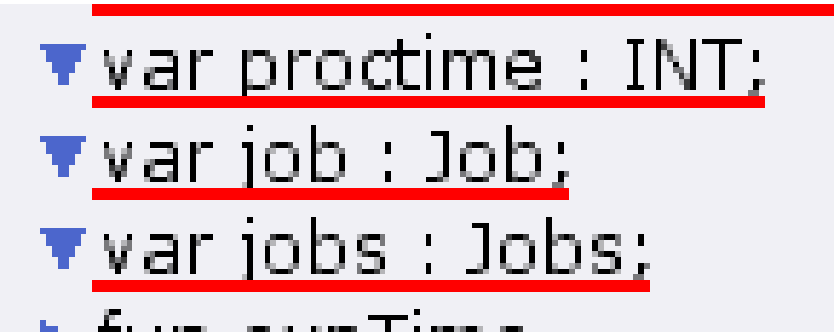

Рис. 3.3: Переменные.

```
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. 3.4: Фнукции.

Далее я построила листы Arrivals (рис. 3.5) и Server (рис. 3.6):

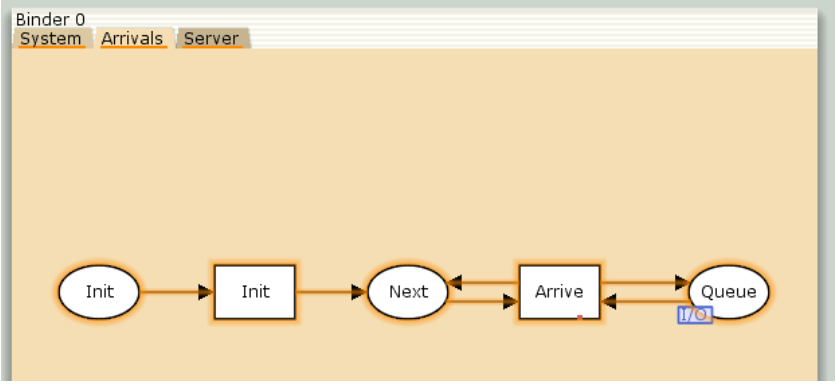

Рис. 3.5: Лист Arrivals.

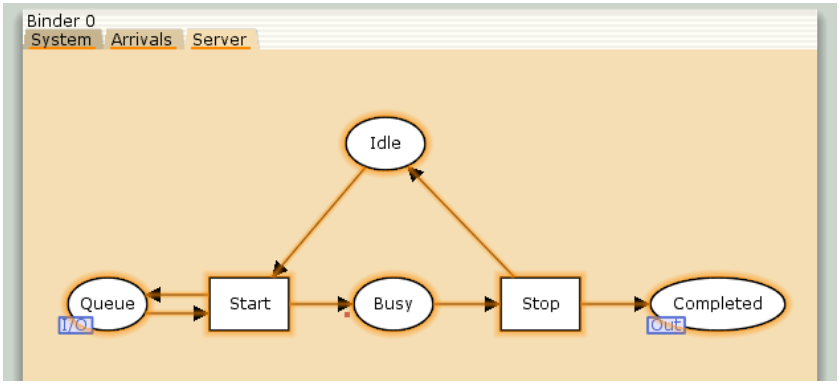

Рис. 3.6: Лист Server.

Потом я задала параметры модели на графах сети. Там было много параметров, поэтому вся информация будет на картинках (рис. 3.7, 3.8 и 3.9):

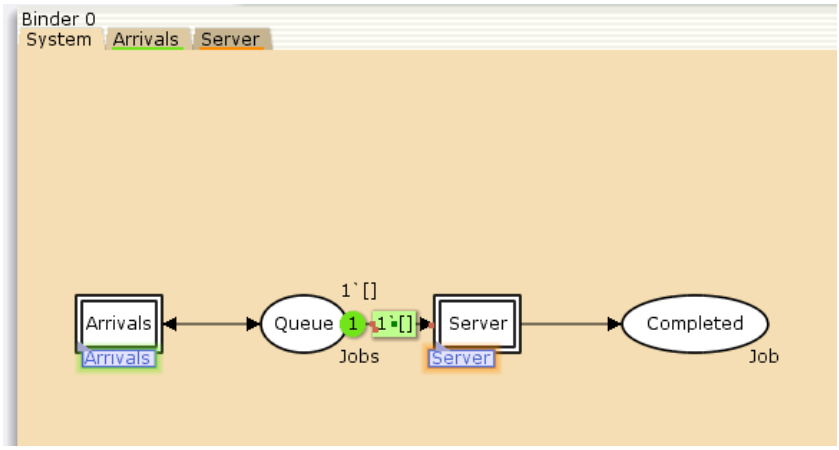

Рис. 3.7: Лист System после задания параметров на графе.

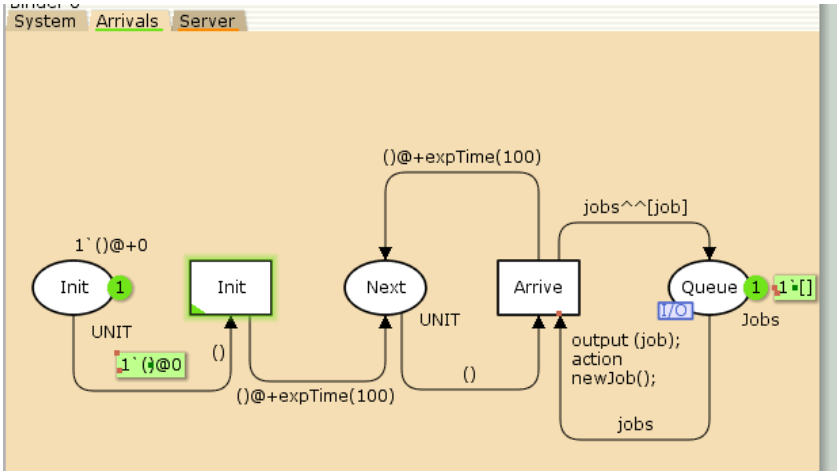

Рис. 3.8: Лист Arrivals после задания параметров на графе.

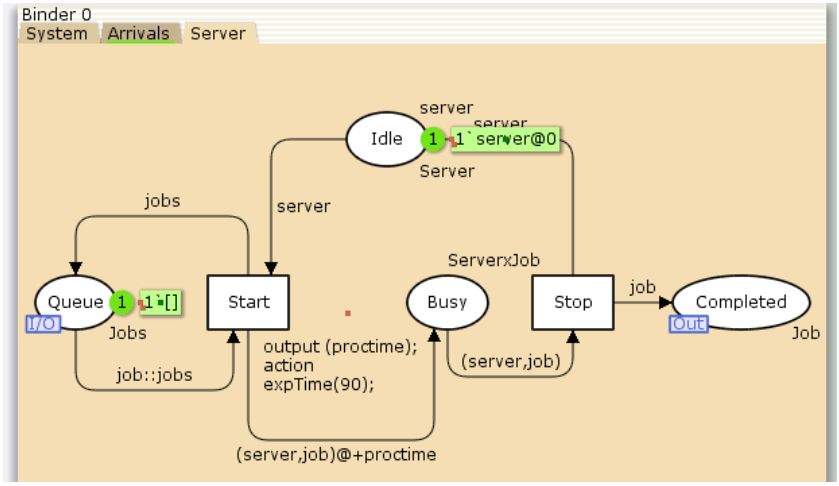

Рис. 3.9: Лист Server после задания параметров на графе.

Далее я приступила к мониторингу сети. Для этого с помощью палитры Monitoring установила точку останова на переход Start, новый монитор назвала Ostanovka, функцию Predicate изменила на следующую:

```
fun pred (bindelem) =
  let
    fun predBindElem (Server'Start (1, {job,jobs,proctime}))
      = Queue_Delay.count()=200
    | predBindElem _ = false
```

```
  in
    predBindElem bindelem
end
```

Потом выбрала Data Coll, установила снова на Start, назвала Queue Delay и изменила функцию Observer на следующую:

```
fun obs (bindelem) =
  let
    fun obsBindElem (Server'Start (1, {job, jobs, proctime}))
      = (intTime() - (#AT job))
      | obsBindElem _ = ~1
  in
    obsBindElem bindelem
  end
```

Далее я запустила моделирование, выполнила более 300 шагов, получила на выводе файл `Queue_Delay.log` (рис. 3.10):

Рис. 3.10: Содержимое Queue_Delay.log.

Далее я построила график значений задержки в очереди, для этого я написала следующий код — он также подходит и для графика, который надо будет построить позже (рис. 3.11):

/home/openmodelica/output/logfiles/graph_plot - Mousepad

Файл Правка Поиск Вид Документ Справка

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

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

# set out "plot1.png"
# plot "Queue_Delay.log" using ($4):($1) with lines

set out "plot2.png"
plot [0:][0:1.2] "Long_Delay_Time.log" using ($4):($1) with lines
```

Рис. 3.11: Скрипт для построения графиков.

Получился следующий график (рис. 3.12):

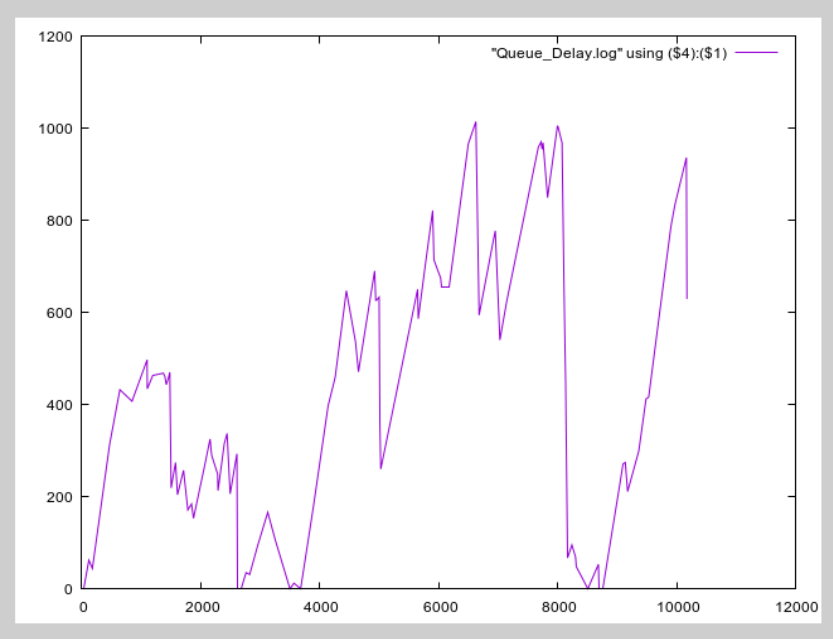

Рис. 3.12: График значений задержки в очереди.

Далее я с помощью палитры Monitoring снова установила точку останова на переход Start, новый монитор назвала Queue Delay Real, функцию Observer изменила на следующую:

```
fun obs (bindelem) =
  let
    fun obsBindElem (Server'Start (1, {job, jobs, proctime}))
```

```
      = Real.fromInt(intTime() - (#AT job))
      | obsBindElem _ = ~1.0
  in
    obsBindElem bindelem
  end
```

После этого я получила файл `Queue_Delay_Real.log`, похожий на `Queue_Delay.log`, только здесь значения имеют действительный тип. После этого я с помощью палитры Monitoring я снова установила Data Coll на Start. Теперь новый монитор я назвала Long Delay Time и сделала следующйю функцию Observer:

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

Далее было необходимо определить глобальную переменную `longdelaytime`, которая бы была границей для большой задержки (рис. 3.13):

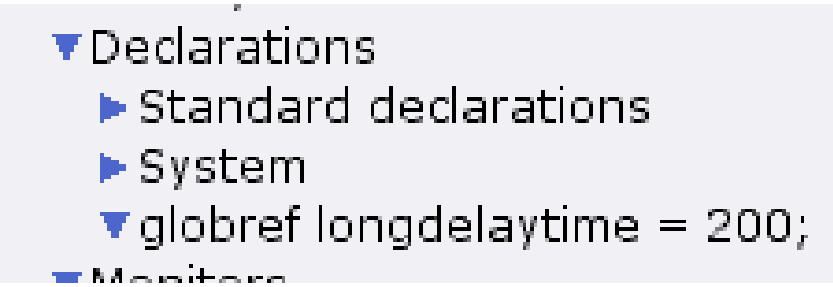

Рис. 3.13: Глобальная переменная longdelaytime.

После этого я немного изменила скрипт для построения графика в GNU Plot (рис. 3.11) и построила график (рис. 3.14), демонстрирующий, в какие периоды времени значения задержки в очереди превышали заданное значение 200:

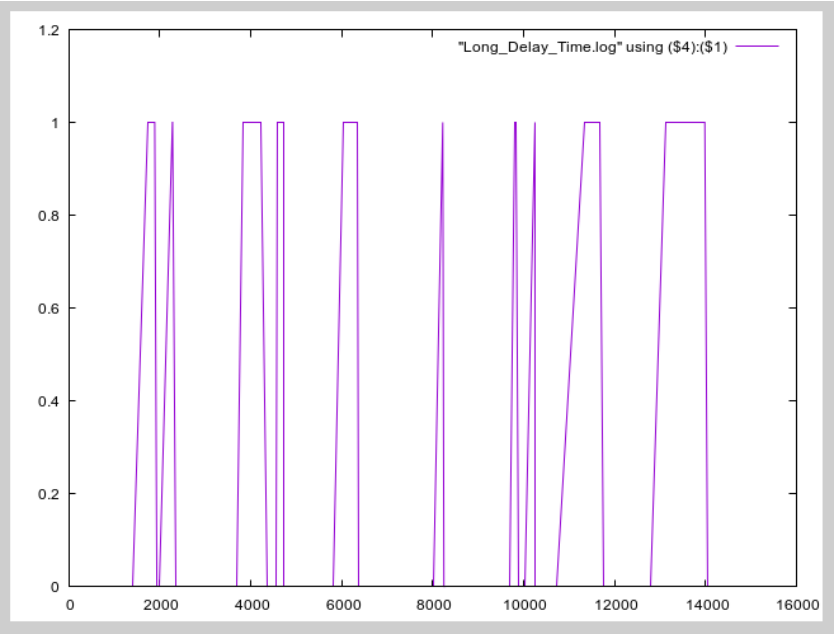

Рис. 3.14: График периодов превышения задержки.

Когда значение графика равно 1, была задержка больше допустимой, когда 0 — превышения не было. Видим, что если ставить переменную 200, то задержка редко превышала допустимую.

4 Выводы

Построили модель M|M|1 в CPN Tools.

Список литературы