РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11

дисциплина: Моделирование информационных процессов

Студент: Доре Стевенсон Элгар

Группа: НКН-бд-01-19

МОСКВА

2023 г.

Постановка задачи

Построение модели СМО M|M|1 в CPNTools.

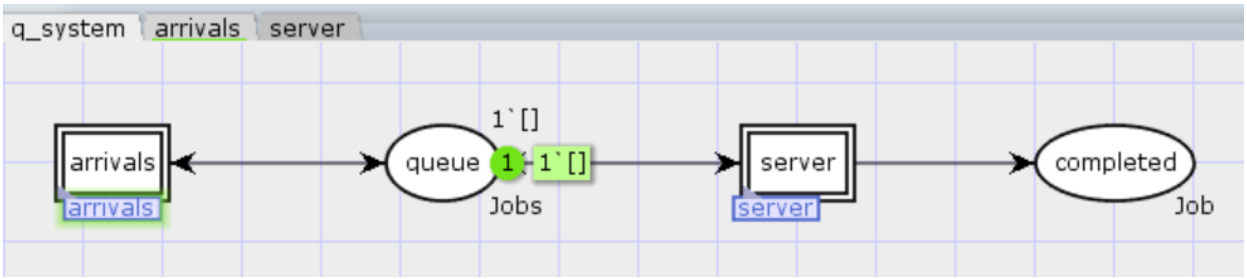
В систему поступает поток заявок двух типов, распределённый по пуассоновскому закону. Заявки поступают в очередь сервера на обработку. Дисциплина очереди - FIFO. Если сервер находится в режиме ожидания (нет заявок на сервере), то заявка поступает на обработку сервером.

Выполнение работы

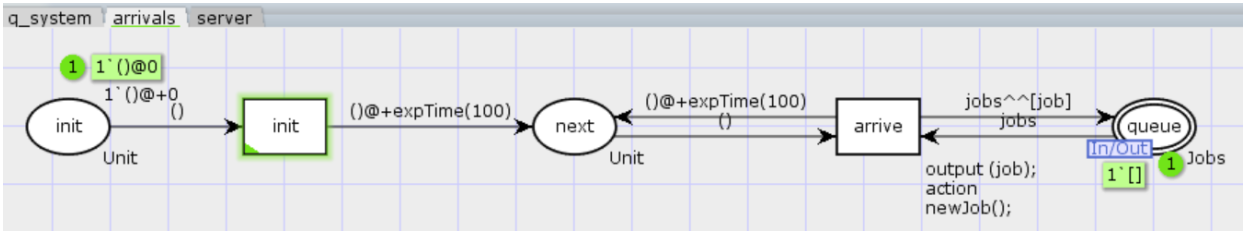
1 Построение модели

Будем использовать три отдельных листа: на первом листе опишем граф системы (q_system), на втором — генератор заявок (arrivals), на третьем — сервер обработки заявок (server).

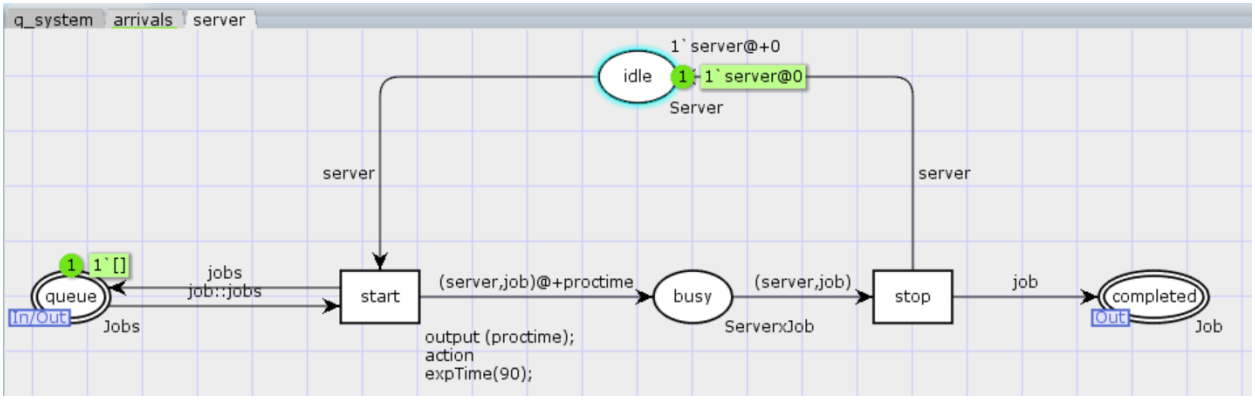
1.1 Граф системы обработки заявок



1.2 Граф генератора заявок системы



1.3 Граф процесса обработки заявок на сервере системы



1.4 Декларации системы

```
Declarations
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;
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()};
```

2 Мониторинг параметров моделируемой системы

2.1 Монитор Ostanovka

```
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
```

2.2 Монитор Queue Delay

```
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
```

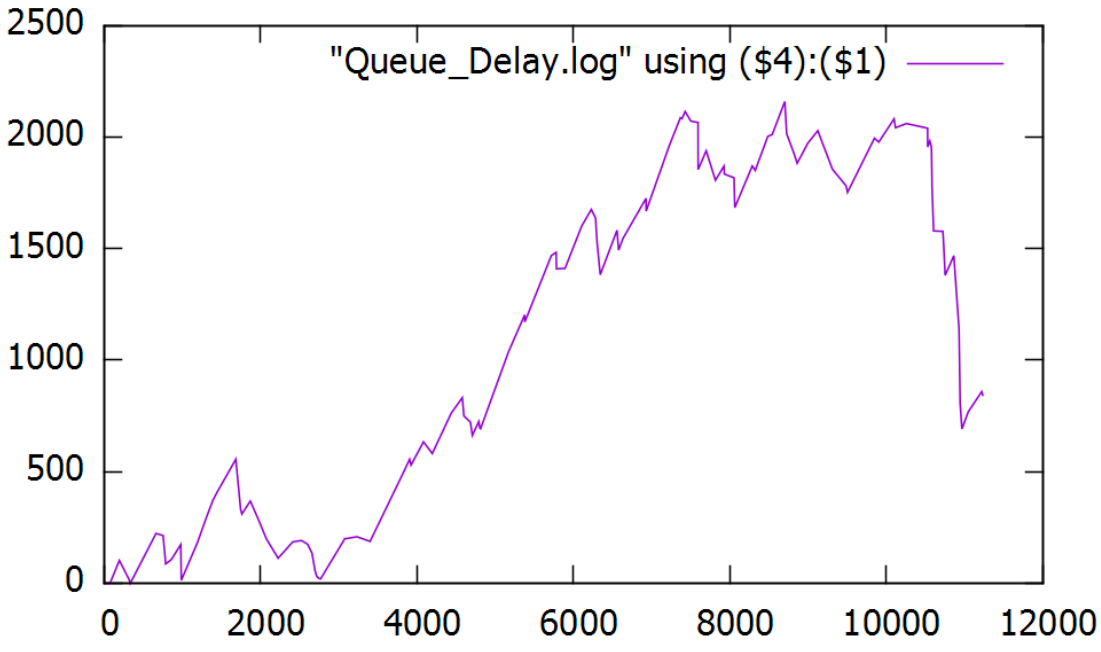
2.3 Монитор Queue Delay Real

```
▼Queue Delay Real
  ▶Type: Data collection
  ▶Nodes ordered by pages
  ▶Predicate
  ▼Observer
      fun obs (bindelem) =
      let
        fun obsBindElem (server'start (1, {job,jobs,proctime})) =
      Real.fromInt(intTime()-(#AT job))
          | obsBindElem _ = ~1.0
      in
        obsBindElem bindelem
      end
  ▶Init function
  ▶Stop
```

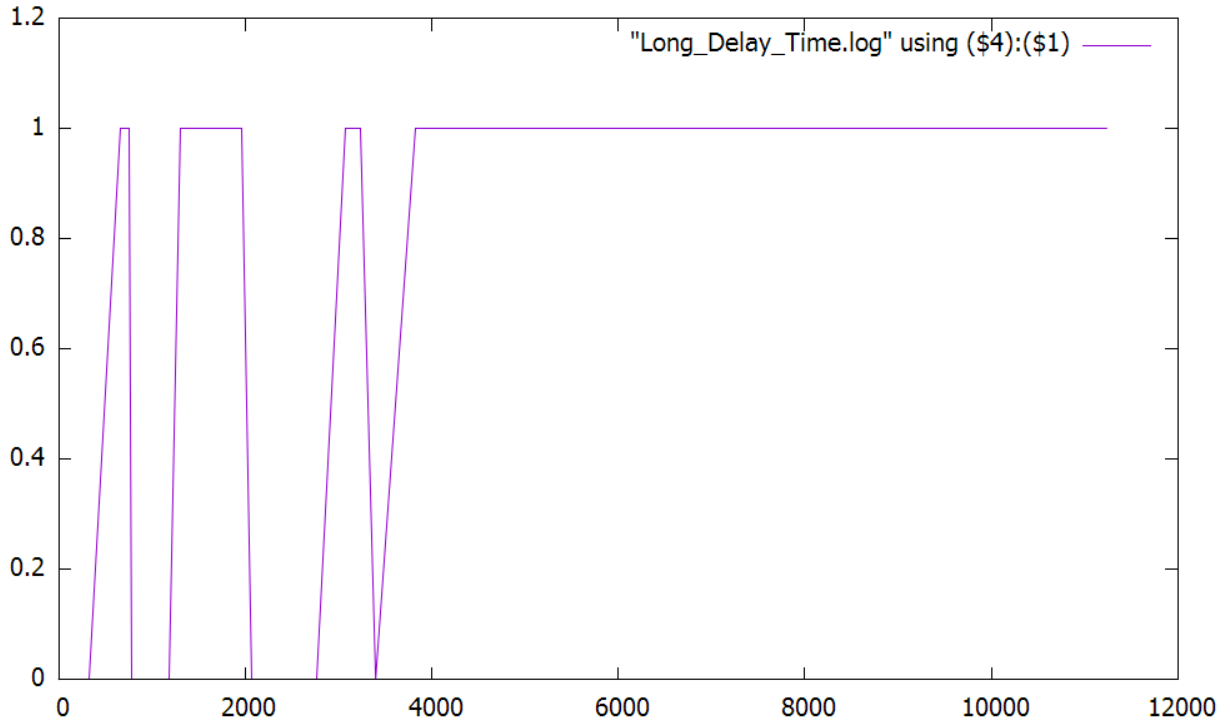
2.4 Декларации и монитор Long Delay Time

```
▼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
```

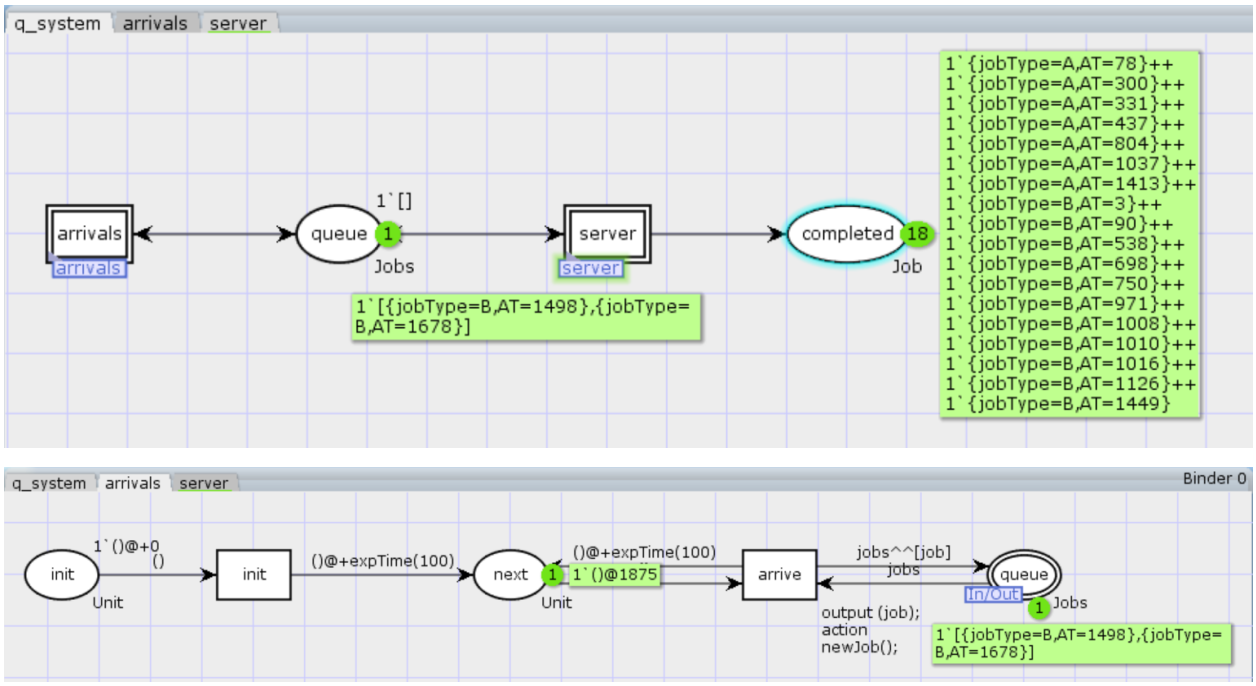
2.5 График изменения задержки очереди

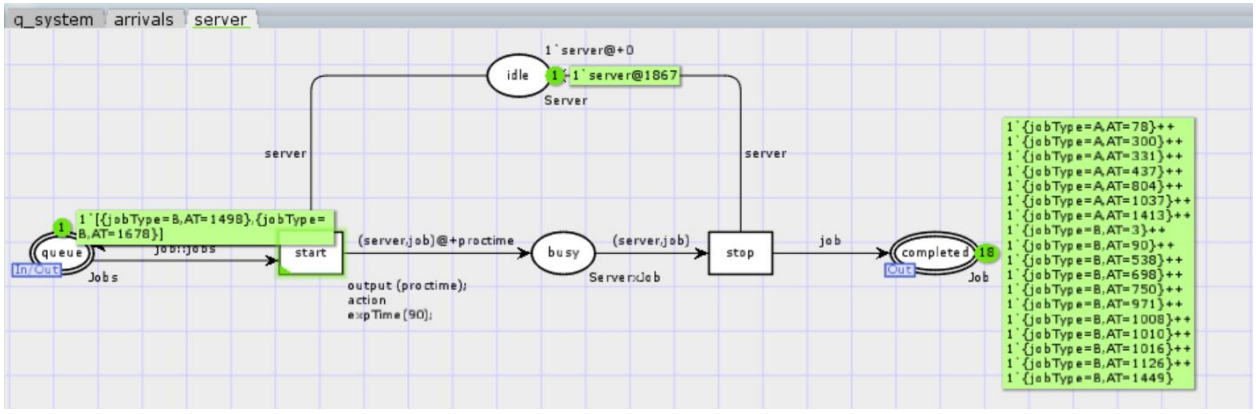


2.6 Периоды времени, когда значения задержки в очереди превышали заданные значения



2.7 Запуск системы обработки заявок в очереди (56 шагов)





Заключение

В ходе лабораторной работы была построена модель СМО M|M|1 в CPNTools. Также для данной модели были созданы различные мониторы для отслеживания параметров очереди. При помощи GNU Plot были построены график изменения задержки очереди и график, отражающий периоды времени, когда значение очереди превышает заданное значенние.