Лабораторная работа 11

Мажитов Магомед Асхабович

Содержание

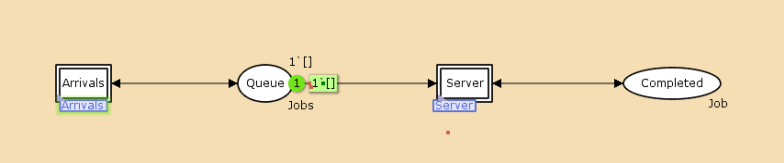
1 Цель работы 1
2 Выполнение лабораторной работы 1
3 Выводы 4

1 Цель работы

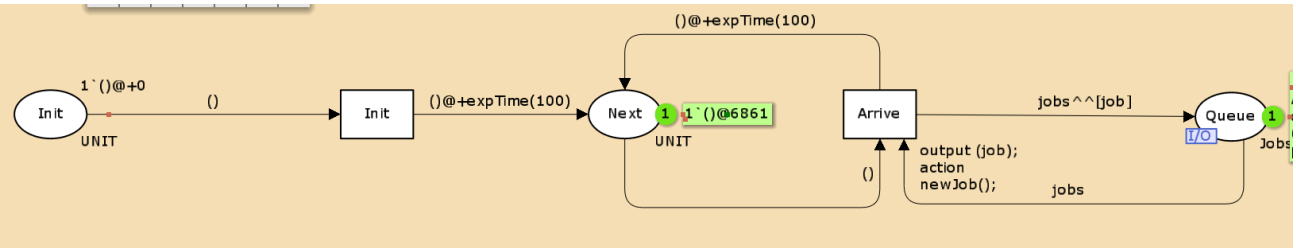
В систему поступает поток заявок двух типов, распределённый по пуассоновскому закону. Заявки поступают в очередь сервера на обработку. Дисциплина очереди - FIFO. Если сервер находится в режиме ожидания (нет заявок на сервере), то заявка поступает на обработку сервером.

2 Выполнение лабораторной работы

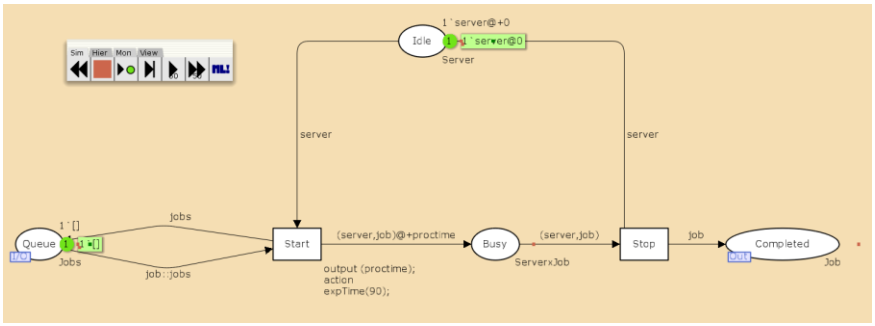
1. Рисуем граф сети.



Граф сети модели



Граф Arrivals



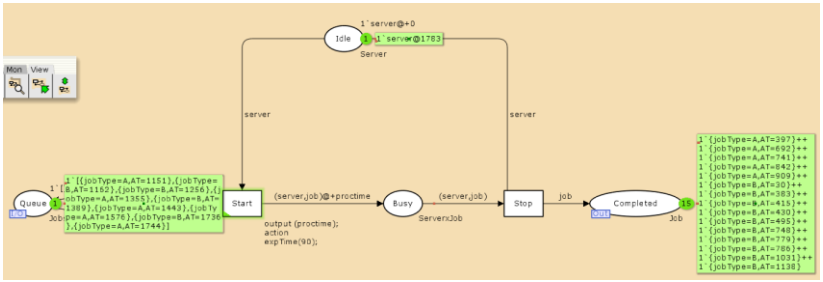
Граф Sevrer

2. Зададим декларации модель.

```
Declarations
  Standard declarations
  globref longdelaytime = 200;
  colset UNIT = unit timed;
  colset INT = int;
  colset Server
  colset JobType
  colset Job = record jobType : JobType *
    AT:INT;
  colset Jobs = list Job;
  colset ServerxJob = product Server * Job timed;
  var proctime
  var 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
```

Декларации модели

3. Если прокрутить моделирование, то сможешь увидеть как пакеты поступают в систему и обрабатываются.



Моделирование

4. Добавим мониторы. Изменим предикат, задав число шагов, через которое будем останавливать мониторинг.

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

Функция Predicate монитора Ostanovka

5. Добавим Data call.

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

Функция Observer монитора Queue Delay

6. Запустив, мы получим log файл, при помощи которого мы можем построить график изменения задержки в очереди.

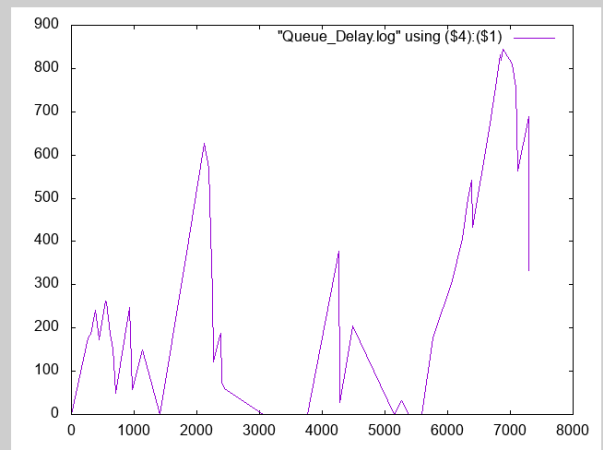


График изменения задержки в очереди

7. Посчитаем задержку в действительных значениях. С помощью палитры Monitoring выбираем Data Call и устанавливаем на переходе Start. Появившийся в меню монитор называем Queue Delay Real.

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

Функция Observer монитора Queue Delay Real

8. Запустив, мы получим log файл, при помощи которого мы можем построить график изменения задержки в очереди.

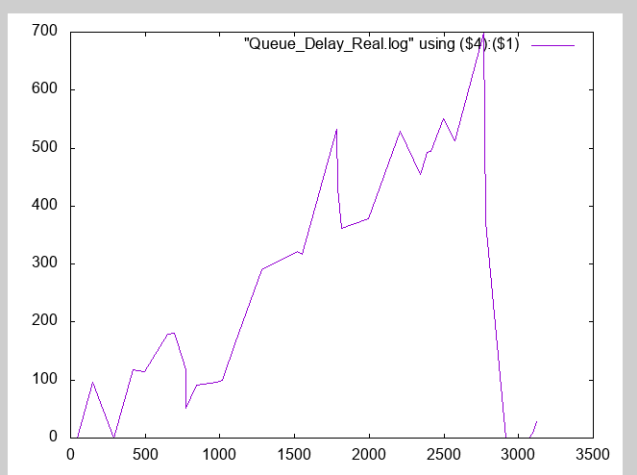


График изменения задержки в очереди

9. Посчитаем, сколько раз задержка превысила заданное значение. С помощью палит- ры Monitoring выбираем Data Call и устанавливаем на переходе Start. Монитор называем Long Delay Time.

10. Запустив, мы получим log файл, при помощи которого мы можем построить график изменения задержки в очереди.

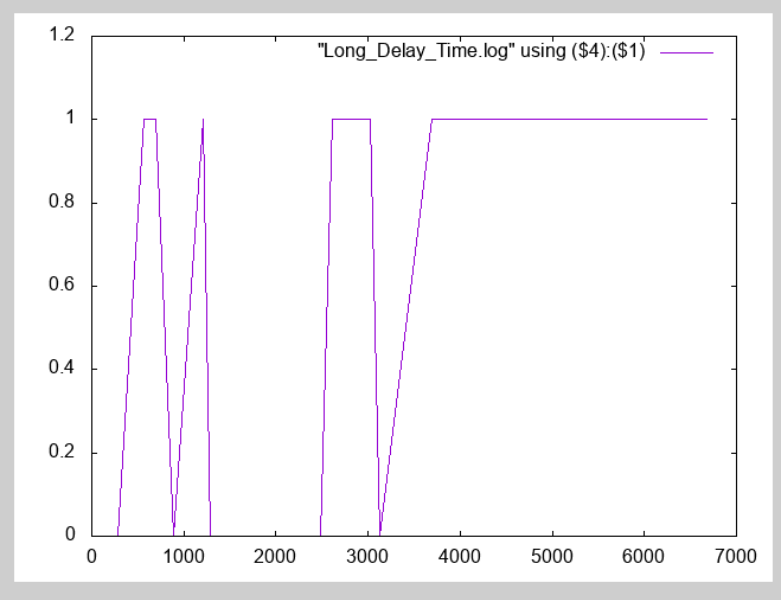


График изменения задержки в очереди

3 Выводы

Во время выполнения лабораторной работы, я провел моделирование M|M|1.