

Государственное образовательное учреждение  
высшего профессионального образования

“ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ”

**Кафедра «Информационные и вычислительные системы»**

Лабораторная работа № 5

По курсу “Высокопроизводительные вычислительные системы”

МУЛЬТИКОМПЬЮТЕРНАЯ СИСТЕМА

Вариант № 1

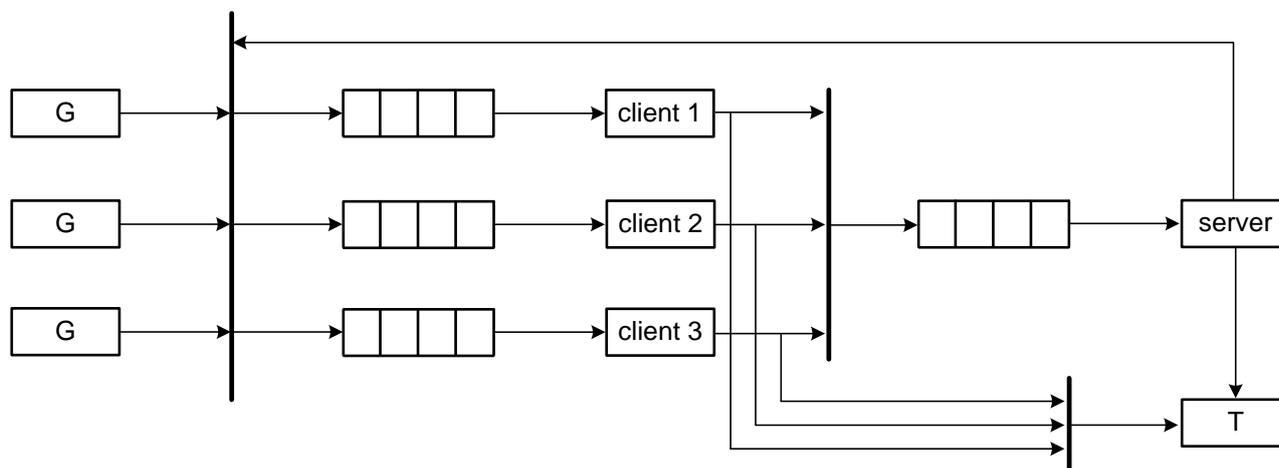
Выполнил студент группы ПВТ - 711  
Бабиев Д. П.

Проверил: Мануйлов Ю. С.

Вариант № 1:

n	I (1/s)	Qs	Qc	U1	U2	Client	Server	P1	P2	P3	P4	P5	P6
3	1	1	1	1	2	Э	Р	0.2	0.2	0.2	0.2	0.1	0.1

### 1 Структурная схема имитационной модели



### 2 Исходных код имитационных программ на языке GPSS

```
funExpo FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38/.8,1.6/
.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/
.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7./.9998,8.
```

```
; random client choice
funChoiceClient FUNCTION RN2,D3
0.33,1/0.66,2/1.0,3
```

```
; random task choice
funChoiceTask FUNCTION RN3,D6
0.2,1/0.4,2/0.6,3/0.8,4/0.9,5/1.0,6
```

```
tabTime TABLE M1,0,0.5,50
```

```
v_server VARIABLE P0
```

```
; P2 - processing on server (1=yes,0=no)
; P3 - processing on client (1=yes,0=no)
; P4 - processing on other client (1=yes,0=no)
; P5 - processing on server (1=yes,0=no)
; P6 - processing on client (1=yes,0=no)
; P7 - number of client (1-3)
; P8 - number of task (1-6)
; P9 - number of other client (1-3)
```

```
GENERATE 1,FN$funExpo
ASSIGN 7,1
TRANSFER ,THE_START
```

```

GENERATE 1, FN$funExpo
ASSIGN 7,2
TRANSFER ,THE_START

GENERATE 1, FN$funExpo
ASSIGN 7,3
TRANSFER ,THE_START

THE_START ASSIGN 2,0
          ASSIGN 3,0
          ASSIGN 4,0
          ASSIGN 5,0
          ASSIGN 6,0

          ASSIGN 8, FN$funChoiceTask
          TEST NE P8,1,C ; client
          TEST NE P8,2,C_S ; client-server
          TEST NE P8,3,C_S_C ; client-server-client
          TEST NE P8,4,C_S_oC ; client-server-otherclient
          TEST NE P8,5,C_S_oC_S ; client-server-otherclient-server
          TEST NE P8,6,C_S_oC_S_C ; client-server-otherclient-server-client

C TRANSFER ,PROC_CLT

C_S ASSIGN 2,1
    TRANSFER ,PROC_CLT

C_S_C ASSIGN 2,1
      ASSIGN 3,1
      TRANSFER ,PROC_CLT

C_S_oC ASSIGN 2,1
        ASSIGN 4,1
        TRANSFER ,PROC_CLT

C_S_oC_S ASSIGN 2,1
          ASSIGN 4,1
          ASSIGN 5,1
          TRANSFER ,PROC_CLT

C_S_oC_S_C ASSIGN 2,1
            ASSIGN 3,1
            ASSIGN 4,1
            ASSIGN 5,1
            ASSIGN 6,1
            TRANSFER ,PROC_CLT

PROC_CLT QUEUE P7
          SEIZE P7
          DEPART P7
          ADVANCE 0,3, FN$funExpo
          RELEASE P7
          TEST E P2,0, PROC_SRV
          TRANSFER ,THE_END

PROC_oCLT ASSIGN 9, FN$funChoiceClient
          TEST NE P9, P7, PROC_oCLT
          QUEUE P9
          SEIZE P9
          DEPART P9
          ADVANCE 0,3, FN$funExpo

```

```

RELEASE      P9
ASSIGN       4,0
TEST  E     P5,0,PROC_SRV
TRANSFER    ,THE_END

PROC_SRV    QUEUE      v_server
           SEIZE      v_server
           DEPART     v_server
           ADVANCE    0.2,0.2
           RELEASE    v_server
           TEST  E    P4,0,PROC_oCLT
           ASSIGN     2,0
           TEST  E    P3,0,PROC_CLT
           TRANSFER   ,THE_END

THE_END     TABULATE   tabTime
           TERMINATE  1

START 100000
    
```

### 3 Результаты работы

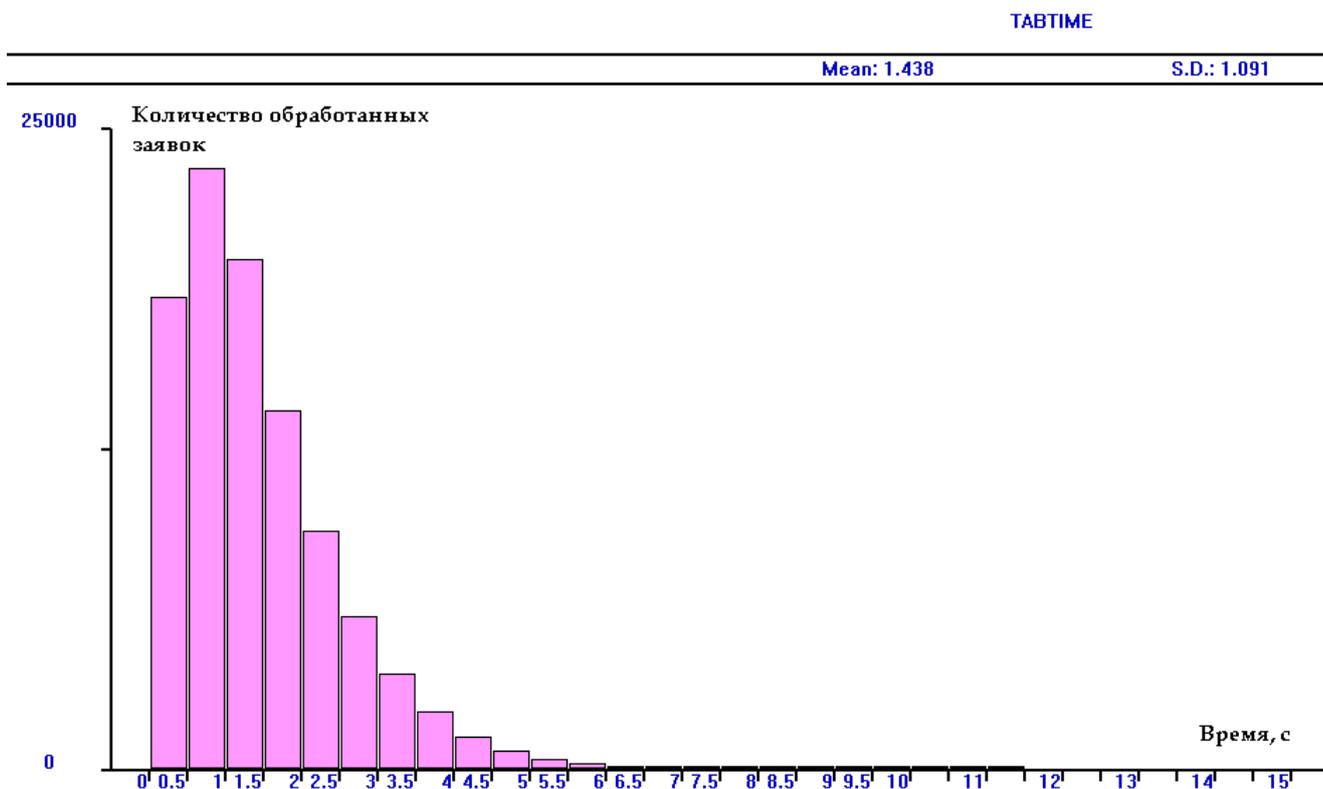


Рис. 3.1 Время обработки запросов

Экспериментально полученные средние значения величин времени обработки заявки на клиентском компьютере  $t_c = 0.3$  и на сервере  $t_s = 0.2$ .

$$U: 1 < 1.438 < 2$$

1. Расчет производительности клиентской ЭВМ

$$V_c = Q_c / t_c = 1 / 0.3 = 3.33(3) \text{ млн.оп/с}$$

2. Расчет производительности сервера

$$V_s = Q_s / t_s = 1 / 0.2 = 5 \text{ млн.оп/с}$$

## Вывод

В процессе выполнения лабораторной работы построена модель мультимедийной системы. Экспериментальным путем подобраны средние значения времени обработки заявки на клиенте (0.3 с) и на сервере (0.2 с), так чтобы среднее время обработки запроса попадало в заданный интервал ( $1 \div 2$ ).

Сервер оказывает большое влияние на быстродействие всей системы, так как большинство видов запросов обращаются к нему. Следовательно, увеличение производительности сервера приведет к увеличению производительности всей системы.

Если выйдет из строя сервер, то большинство запросов не будут выполнены (за исключением автономной работы клиента). Если выйдет из строя клиент, то запросы, связанные с ним не будут выполнены, однако система будет продолжать работу.

Система может быть масштабируема:

- за счет увеличения количества клиентов. Этот вариант увеличит производительность, если скорость обработки запросов сервером велика.
- за счет ускорения работы сервера. Подходит, когда сервер не справляется с обработкой запросов от клиентов.
- за счет увеличения производительности клиента (если скорость обработки запросов сервером велика).