

## Имитационное моделирование ИПС, построенных на основании архитектуры клиент/сервер

Ахмед Али (Абдель Карим) Альмхерат  
*Харьковский национальный университет им. В. Н. Каразина, Украина*

Simulating models for retrieval system with homogeneous and heterogeneous query flows are built at the article. Model experiments were showed that models qualitative behavior corresponded with qualitative behavior of real system.

### 1. Введение

Информационно-поисковые системы (ИПС) представляют важный класс программного обеспечения, предназначенного для поддержки процессов принятия решений. Особенностью этого класса информационных систем является относительная статичность базы данных (БД) при необходимости обработки интенсивного потока запросов различного типа. Критичными характеристиками распределенной ИПС с архитектурой клиент/сервер являются вероятность потери запроса и среднее время ожидания ответа на запрос. В этой ситуации при проектировании системы необходимо оценить возможные значения для этих характеристик, что позволяет дать обоснованный ответ на вопрос о целесообразности реализации архитектуры клиент/сервер для проектируемой ИПС.

Вероятностные свойства потоков заявок на обслуживание являются предметом изучения теории массового обслуживания [1]. Применение результатов этой теории к проектированию компьютерных сетей достаточно подробно освещено в [2]. Однако специфика потоков заявок на обслуживание, соответствующих потокам запросов в ИПС, построенных на основе архитектуры клиент/сервер, не нашла отражения в моделях, рассмотренных и изученных в литературе.

Целью настоящей работы является разработка имитационных моделей обслуживания однородных и неоднородных потоков запросов в ИПС, построенной на основании архитектуры клиент/сервер.

### 2. Изложение результатов

Для исследования потоков запросов к ИПС в работе построены имитационные модели, описание которых на языке системы GPSS World (ограниченная версия системы и документация к ней доступна на сайте <http://www.minutemansoftware.com>) приведено ниже.

Для модели однородного потока заявок используется поток Пуассона с интенсивностью  $\lambda$  :

$$P(\xi_t = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad k = 0, 1, \dots \quad (1)$$

Для моделирования неоднородного потока заявок, содержащего заявки  $n$  различных типов с интенсивностями  $\lambda_1, \dots, \lambda_n$  соответственно используется поток со следующей функцией распределения вероятностей:

$$\begin{cases} P(\xi_t = ( )) = e^{-\lambda t} \\ P(\xi_t = (i_1, i_2, \dots, i_k)) = e^{-\lambda t} \left( \prod_{j=1}^k \frac{\lambda_{i_j}}{\lambda} \right) \frac{(\lambda t)^k}{k!}, \quad k = 1, 2, \dots \end{cases} \quad (2)$$

В формулах (2)  $\lambda$  определяется как сумма  $\lambda_i$ :  $\lambda = \lambda_1 + \dots + \lambda_n$ .

В обеих моделях предполагается, что время обработки распределено равномерно на интервале  $[\tau - \Delta\tau, \tau + \Delta\tau]$  так, чтобы отношение  $\delta = \frac{\sigma_\tau}{\tau}$  было равно 0,01. Таким образом, величина  $\Delta\tau$  определяется по формуле  $\Delta\tau = \delta \cdot \tau\sqrt{3}$ .

В тексте описания модели с однородным потоком используются следующие объекты.

Константы:

Delta –  $\delta$ , при моделировании принимается значение 1%;

MaxTime – общее время наблюдения, при моделировании принимается значение 10000 с (около суток).

Варьируемые факторы:

CRate – интенсивность потока запросов к ИПС, при моделировании варьируется в диапазоне от 0,01 до 10 1/с;

BLen – длина буфера, при моделировании варьируется обычно от 0 до 100;

HTMean – среднее время обработки запроса, при моделировании варьируется в диапазоне от 0,01 до 10 с.

Исследуемые характеристики:

RFreq – вероятность потери запроса;

WTime – среднее время ожидания выполнения запроса.

```
*****
; Model constants:
Delta      EQU      0.01    ; handling time relative error
MaxTime    EQU      10000
*****
; Model factors and their default values:
CRate      EQU      1       ; calling rate
BLen       EQU      50      ; buffer length
HTMean     EQU      1       ; handling time mean
*****
; Initialization section
*****
GENERATE   , , 0, 1, 10
```

```

; Initialization of studying parameters:
    SAVEVALUE RFreq,0    ; rejecting frequency
    SAVEVALUE WTime,0    ; waiting time
; Working variables:
    ; set half of handling time range
    SAVEVALUE HTRange,(Delta#X$HTMean#Sqr(3))
    SAVEVALUE QGen,0     ; quantity of generated
                                ; queries
    SAVEVALUE QHandled,0 ; quantity of handled
                                ; queries
    SAVEVALUE QLost,0    ; quantity of lost queries
    TERMINATE
*****
;                               Model section
*****
    GENERATE (Exponential(1,0,1/CRate)) ; queries flow
                                ; generator
    SAVEVALUE QGen+,1
    TEST LE (Q$Accepted),BLen,Rejected ; test buffer
                                ; overflow
    QUEUE Accepted ; start handling
    SEIZE Server
    ADVANCE HTMean,X$HTRange
    RELEASE Server
    DEPART Accepted ; end handling
    ; begin of WTime calculating
    SAVEVALUE QHandled+,1
    SAVEVALUE OldWTime,X$WTime
    SAVEVALUE WTime+,((M1-X$OldWTime)/X$QHandled)
    ; end of WTime calculating
    TRANSFER ,Destroy
    ; begin of RFreq calculating
Rejected SAVEVALUE QLost+,1
    SAVEVALUE RFreq,(X$QLost/X$QGen)
    ; end of RFreq calculating
Destroy TERMINATE
*****
;                               Timer section
*****
    GENERATE MaxTime, , ,1 ; Halting time = MaxTime s
    TERMINATE 1
*****

```

Как видно из исходного текста модель состоит из трех сегментов: сегмента инициализации, сегмента основной модели и сегмента таймера. Такая структура делает модель устойчивой по отношению к изменениям условий экспериментов.

С целью качественной проверки поведения модели был поставлен ряд экспериментов.

1. Выяснение зависимости вероятности потери запроса от длины буфера при разных интенсивностях потоков запросов ( $CR$ ) и фиксированном (на уровне

1,0 с) среднем времени обработки. Результат эксперимента представлен на рис. 1.

2. Выяснение зависимости вероятности потери запроса от длины буфера при разных временах обработки запросов ( $HTime$ ) при фиксированной (на уровне 1,0 1/с) интенсивности потока заявок. Результат эксперимента представлен на рис. 2.

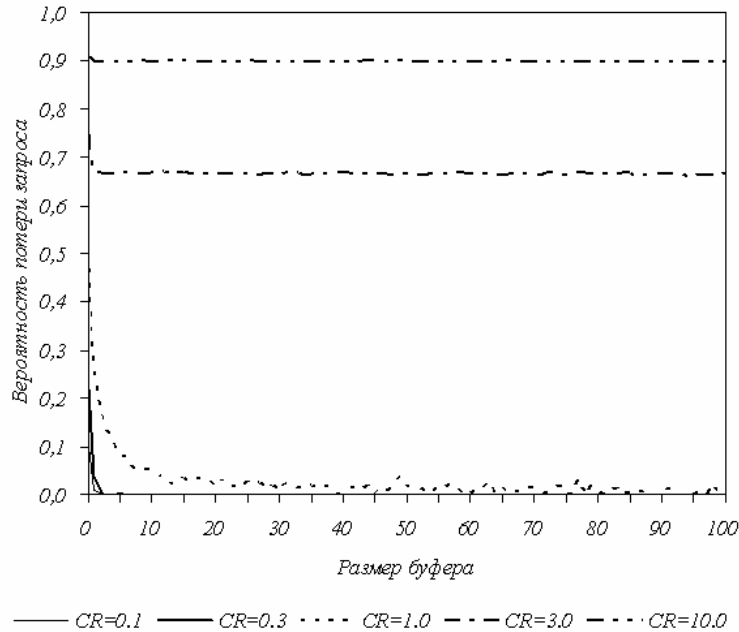


Рис. 1. Зависимость вероятности потери запроса от длины буфера при разных интенсивностях потоков запросов

Из графиков на рис. 1, рис. 2 видно, что с ростом интенсивности потока запросов при фиксированном времени обработки запроса, либо времени обработки запроса при фиксированной интенсивности потока запросов, происходит увеличение вероятности потери запроса. Кроме того, видно, что увеличение буфера уменьшает вероятность потери запроса. Иными словами, результаты экспериментов на имитационной модели согласуются с ожидаемым качественным поведением ИПС.

В тексте описания модели с неоднородным потоком параметры модели читаются из внешнего файла, содержащего интенсивности потоков заявок каждого типа и среднее время их обработки ИПС. Текст самой модели приведен ниже.

```

*****
; Model constants:
Delta      EQU      0.01 ; handling time relative error
MaxTime    EQU      10000
*****
; Model factors and their default values:
CRate      EQU      1 ; calling rate
BLen       EQU      50 ; buffer length

```

HTMean EQU 1 ; handling time mean

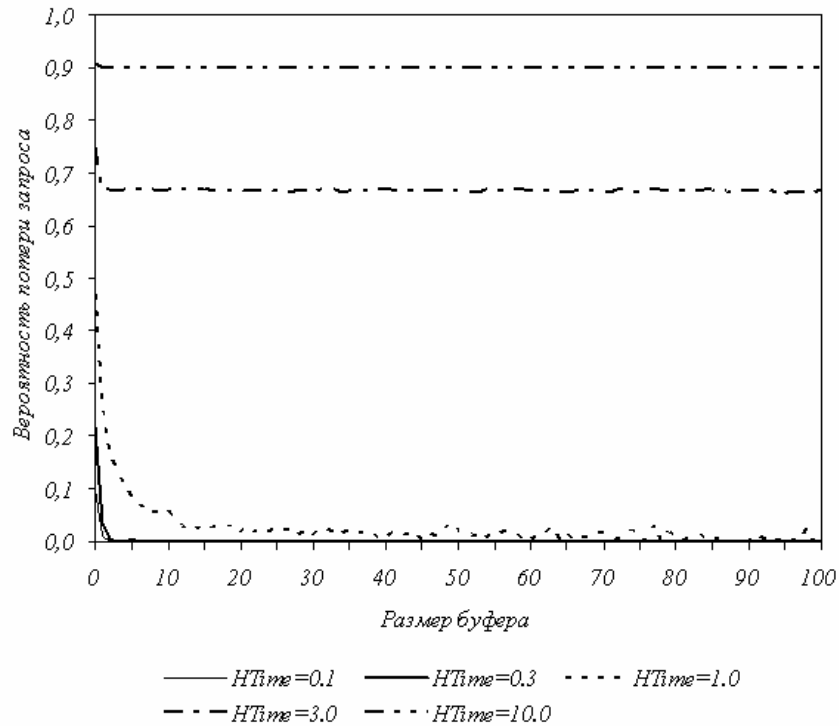


Рис. 2. Зависимость вероятности потери запроса от длины буфера при разных временах обработки запросов

```

*****
* Input section:
*   Quantity of query types (TypeQTY) and
*   correspondent calling rates (matrix Queries) are
*   got at the section
*****
* TypeQTY: quantity of query types
* Queries: queries characteristics TypeQTY x 2 - matrix
*   Queries[i,1] - i-th query calling rate
*   Queries[i,2] - i-th query handling time
* SumRate:
*   storage for the sum of query calling rates
*****
Queries MATRIX          ,100,2
*****
Input  GENERATE      ,,0,1,10      ; Section priority = 10
       OPEN          ("params.txt"),1,FO_Error
       SAVEVALUE     TypeQTY,0
RdLoop READ          RBuff,1,EOF
       SAVEVALUE     TypeQTY+,1

```

```

MSAVEVALUE Queries,X$TypeQTY,1,(Word(P$RBuff,1))
MSAVEVALUE Queries,X$TypeQTY,2,(Word(P$RBuff,2))
SAVEVALUE SumRate+,(Word(P$RBuff,1))
TRANSFER      ,RdLoop
EOF           CLOSE      ,1,FC_Error
             TEST E      X$TypeQTY,0,FR_Error
             SAVEVALUE ErrMsg,"No errors"
             SAVEVALUE Counter,0
             SAVEVALUE RFreq,0 ; rejecting frequency
             SAVEVALUE WTime,0 ; waiting time
             SAVEVALUE HTRange,(Delta#X$HTMean#Sqr(3))
             SAVEVALUE QGen,0 ; quantity of generated
                               ; queries
             SAVEVALUE QHandled,0; quantity of handled
                               ; queries
             SAVEVALUE QLost,0 ; quantity of lost queries
TERMINATE
FO_Error    SAVEVALUE ErrMsg,"File opening error"
TERMINATE
FR_Error    SAVEVALUE ErrMsg,"Input file is not correct"
TERMINATE
FC_Error    SAVEVALUE ErrMsg,"File closing error"
TERMINATE
*****
* Model section:
*****
GENERATE      (Delay(1))
SAVEVALUE    QGen+,1
TEST LE      (Q$Accepted),BLen,Rejected ; test buffer
QUEUE        Accepted ; start handling
ASSIGN       Type,(GetType(2))
SEIZE        Server
ADVANCE      (ServiceTime(3))
RELEASE      Server
DEPART       Accepted ; end handling
; begin of WTime calculating
SAVEVALUE    QHandled+,1
SAVEVALUE    OldWTime,X$WTime
SAVEVALUE    WTime+,(M1-X$OldWTime)/X$QHandled)
; end of WTime calculating
TRANSFER     ,Destroy
; begin of RFreq calculating
Rejected    SAVEVALUE QLost+,1
           SAVEVALUE RFreq,(X$QLost/X$QGen)
; end of RFreq calculating
Destroy    TERMINATE

```

```

*****
; Timer section
*****
          GENERATE      MaxTime, , ,1 ; Halting time = MaxTime s
          TERMINATE     1
*****
* GetType function:
*****
PROCEDURE GetType(RND) BEGIN
  TEMPORARY pSum,type,tval;
  tval=Uniform(RND,0,X$SumRate);
  type=1;
  pSum=Queries[type,1];
  WHILE(tval > pSum) DO BEGIN
    type=type+1;
    pSum=pSum+Queries[type,1];
  END;
  RETURN type;
END;
*****
* Delay function:
*****
PROCEDURE Delay(RND) BEGIN
  TEMPORARY scale;
  scale=X$SumRate;
  IF(scale>0) THEN RETURN Exponential(RND,0,scale);
  ELSE RETURN 1;
END;
*****
* Service function:
*****
PROCEDURE ServiceTime(RND) BEGIN
  TEMPORARY tmp;
  tmp=Queries[P$Type,2];
  RETURN Uniform(RND,0.9#tmp,1.1#tmp);
END;
*****

```

### 3. Выводы

Таким образом, в статье построены имитационные модели для информационно-поисковых систем, построенных на основании архитектуры клиент/сервер для однородных и неоднородных потоков запросов. Поставлены численные эксперименты и приведены их результаты, которые показывают, что качественное поведение построенных моделей соответствует ожидаемому.

В дальнейшем планируется проведение на этих моделях исследований с целью:

1. Выяснения характера зависимости вероятности потери запроса и среднего времени ожидания реакции системы от основных факторов – интенсивности потока заявок, среднего времени обработки и размера буфера для системы с однородными запросами.

2. Выяснения характера зависимости вероятности потери запроса и среднего времени ожидания реакции системы от основных факторов – суммарной интенсивности потока заявок, среднего средних времен обработки и размера буфера для системы с неоднородными запросами.
3. Разработки методики определения основных характеристик информационно-поисковых систем при их проектировании.

#### ЛИТЕРАТУРА

1. Гнеденко В. В., Коваленко И. Н. Введение в теорию массового обслуживания. – М.: Наука, 1987. – 336 с.
2. Вишневский В. М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003 – 506 с.