

УДК 519.2

Л.В. Кулагина

**МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ
ДЛЯ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Нижегородский государственный технический университет им. Р.Е. Алексеева

Рассматриваются методы построения информационных систем на распределенной базе данных. Предлагается математическое описание процессов создания и передачи данных на распределенные серверы.

Ключевые слова: базы данных, информационные системы, клиент серверные технологии, математическая модель, теория массового обслуживания.

Большинство систем, с которыми человек имеет дело, являются сложными. Попытка их математического описания с помощью детерминистических моделей приводит к большой погрешности истинного положения вещей. При решении задач анализа и проектирования таких систем приходится считаться с тем, что случайность является определяющей для процессов, протекающих в системах. При этом пренебрежение случайностью приводит к искажению, к ошибкам в выводах и практических рекомендациях.

Первые задачи теории систем массового обслуживания были рассмотрены в начале XX века. Эти задачи были актуальны в связи со стремлением упорядочить работу телефонной сети и разработать методы, позволяющие заранее повысить качество обслуживания потребителей в зависимости от числа используемых устройств. Оказалось, что ситуации, возникающие на телефонных станциях, являются типичными не только для телефонной связи. Работа аэродромов, морских и речных портов, магазинов, терминальных классов, электронных вычислительных комплексов, радиолокационных станций и т.д. может быть описана в рамках теории систем массового обслуживания. Работа корпоративной информационной системы (КИС) тоже может быть описана с помощью теории массового обслуживания. Например: обмен данными между серверами, поддержание в актуальном состоянии структуры распределенных баз данных, формирование отчетных форм при помощи приложения, в котором идет обращение к серверу баз данных. На рис. 1 представлена схема формирования отчетных форм. Данные процессы являются системой массового обслуживания и относятся к классу – системы с ожиданием (с очередью), причем обслуживание упорядоченное, т.е. запросы обрабатываются в порядке поступления. Данные системы являются системами неограниченным ожиданием. Любой запрос рано или поздно будет обслужен.

В рассматриваемых задачах каждый запрос к базе данных требует его обработки в течение некоторого случайного промежутка времени, зависящего от содержания запроса. Таким образом, работу сервера можно рассматривать как операцию массового обслуживания, состоящую из элементарных операций – обработки отдельных запросов. Одной из характерных особенностей системы массового обслуживания является наличие некоторого потока событий (запросов).

Под потоком событий понимается последовательность однородных событий, появляющихся одно за другим в случайные моменты времени. В общем случае это дает последовательность случайных точек, которые делят числовую ось времени на случайные интервалы. Приведем примеры потоков: последовательность вызовов на телефонной станции, поток заявок на ремонт какого-то оборудования, поток отказов (сбоев) ЭВМ и т.д. В данной работе мы имеем дело с потоком запросов пользователей к базе данных и поддержании ее в актуальном состоянии.

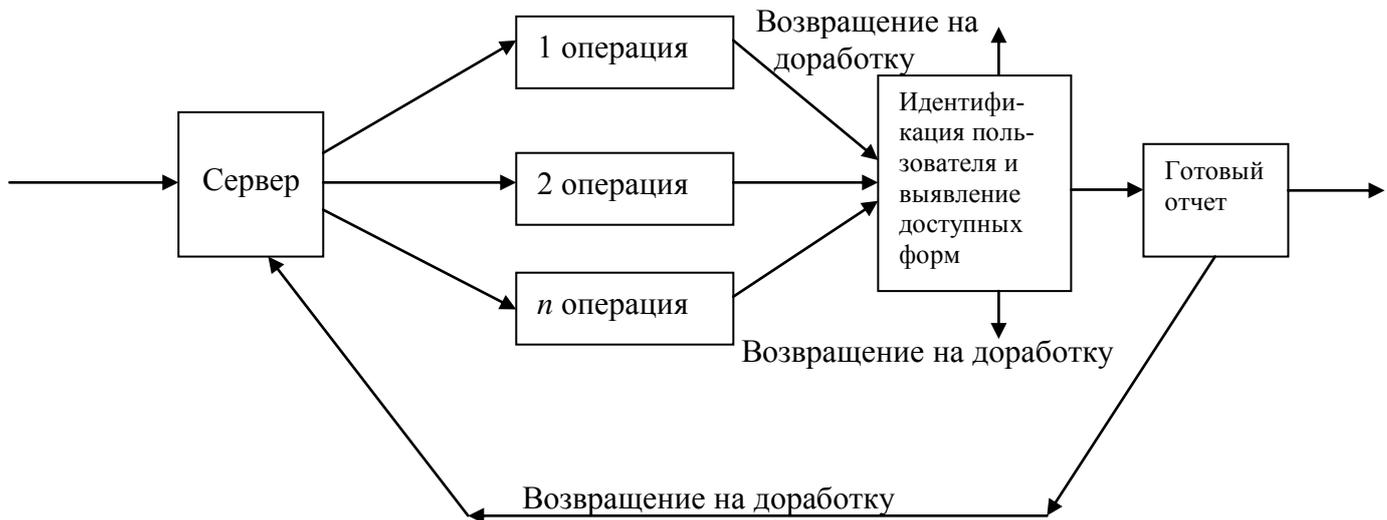


Рис. 1. Структурная схема потока запросов на построение отчетных форм

Главной характеристикой любого потока является его интенсивность λ , которая равна среднему числу событий, происходящих за единицу времени. Рассмотрим частный случай, когда интенсивность является постоянной величиной. Пусть T случайная величина, равная времени между соседними обращениями к базе данных, тогда ее функция распределения вероятности равна:

$$F(t) = 1 - e^{-\lambda t}, \quad (1)$$

а, следовательно, функция плотности вероятности T имеет вид:

$$f(t) = \lambda e^{-\lambda t}, \quad t > 0. \quad (2)$$

Кроме того, известно, что числовые характеристики случайной величины T определяются равенствами:

$$M(T) = 1/\lambda, \quad (3)$$

$$D(T) = 1/\lambda^2. \quad (4)$$

Если на предприятии с числом пользователей, равным 1000, обращения к системе происходят с постоянной интенсивностью, примерно 10 обращений в минуту, то данный поток запросов является потоком Пальма, т.е. обладает свойствами ординарности и стационарности с ограниченным последствием. Вероятность обращений к базе данных за промежуток времени длительностью t , независимо от начала и конца этого промежутка, определяется по формуле Пуассона:

$$P_t(k) = (\lambda t)^k e^{-\lambda t} / k!. \quad (5)$$

Зафиксируем t и найдем наиболее вероятное значение $k=k_0$, т.е. k , при котором вероятность $P_t(k)$ наибольшая. Для этого необходимо решить неравенства:

$$P_t(k_0-1) \leq P_t(k_0) \quad (6) \quad \text{и} \quad P_t(k_0) \geq P_t(k_0+1). \quad (7)$$

Используя формулу Пуассона, получаем, что $\lambda t - 1 \leq k_0 \leq \lambda t$.

Рассмотрим теперь математическую модель механизма создания и передачи на распределенные серверы печатных форм отчетности, который рассмотрен в работе [1-3]. При формировании отчетных форм возникает поток запросов пользователей к базе данных. Данный поток не обладает постоянной интенсивностью, а следовательно, является нестационарным пуассоновским потоком с мгновенной плотностью $\lambda(t)$:

$$\lambda(t) = \lim_{\Delta t \rightarrow \infty} \frac{M(t + \Delta t) - M(t)}{\Delta t}, \quad (8)$$

тогда функция плотности распределения будет иметь вид:

$$f(\tau) = \lambda(t_0 + \tau) e^{-\int_{t_0}^{t_0+\tau} \lambda(t) dt}, \quad (9)$$

где τ – временной интервал, на который попадает n событий.

Исходя из изложенного, можно математически описать работу информационной системы на уровне приложений и базы данных.

Одной из наиболее сложных задач при разработке ИС, построенных на распределенной базе данных, является задача организации взаимодействия входящих в ее состав функциональных элементов (ФЭ), представляющих собой функционально завершенные программные единицы приложений – процедуры, функции, объекты и серверы, которые хранят процедуры баз данных, команды Transact_SQL. Управление ФЭ предлагается проводить на двух уровнях: приложения и базы данных. Управление ФЭ на уровне приложения подробно разобраны в работах [4-7]. Рассмотрим подробнее управление ФЭ в ИС на уровне БД. Для описания структуры ИС любого уровня сложности достаточно нескольких типов элементов: набора ФЭ и анализатора состояний. Разработка анализатора состояний, использующего непосредственно проблемные данные, для каждой ИС является задачей уникальной и не менее сложной, чем создание структуры взаимодействия ФЭ. Для упрощения данной задачи определим уровень абстрактного описания состояния ИС. Рассмотрим ИС как некоторый объект, который состоит из набора функциональных элементов F_1, F_2, \dots, F_n и может находиться в конечном множестве состояний в абстрактном пространстве состояний. Каждое из состояний ИС зависит от значений данных хранящихся в БД и характеризуется признаками состояния P_1, P_2, \dots, P_m в абстрактном многомерном пространстве состояний. Для регистрации признаков состояния в БД [4-6] заводится служебная таблица «Sign_of_condition». Ее структура представлена в табл. 1.

Таблица

Структура признака состояния базы данных

№ п/п	Название поля	Тип поля
1	Kod	Int IDENTITY
2	Name	Char
3	Sign	Bit in not null
5	Old_Sign	Bit in not null
4	Kod_DB	Tinyint
6	Condition	char
7	Condition_TRUE	char
8	Condition_FALSE	char
9	Condition_Invert_Sing	char

Признак состояния в таблице представляет собой запись. Рассмотрим один из признаков состояния более подробно. Признак состояния представляет собой логическое поле Sign, принимающее два значения: либо установлен, либо сброшен. В логическом поле Sign хранится текущее положение признака. В логическом поле Old_Sign хранится предыдущее значение признака. Изменение предыдущего значения происходит при отработке триггера данной таблицы на UPDATE. Пространство признаков представляет собой гиперпространство. Поскольку признак принимает два значения, то работу ИС можно описать в виде переходов между вершинами m -мерного гиперкуба. Значение конкретного признака зависит от условия данного признака, хранящегося в поле «Condition» и написанного на Transact-SQL. Во время проверки условия функциональные элементы могут работать тремя способами, отличающимися между собой условием запуска ФЭ. С каждым способом запуска будут связаны свои ФЭ. Пусть с первым способом запуска, назовем его «пролог», связаны определенные ФЭ, которые полностью и последовательно выполняются при каждой установке значения «уста-

новлен». Данные ФЭ перечисляются в поле «Condition_TRUE», согласно требованиям Transact-SQL. С установкой значения «сброшен» связан второй способ запуска ФЭ, называемый «Эпилогом». ФЭ выполняются аналогичным образом и перечисляются в поле «Condition_FALSE». Кроме этого, есть еще один способ запуска ФЭ, называемый «Очередью». При каждой смене значения ПС выполняются элементы «Очереди». Введение служебной таблицы «Sign_of_condition» с признаками абстрактного пространства состояний позволяет более легкими средствами построить типовой анализатор состояния ИС - Контроллер Состояний (КС), входящий в состав подсистемы управления ИС, предназначенный для управления запуском ФЭ в зависимости от изменений в данных ИС. Для построения КС можно использовать службу «SQL Server Agent». Работа «SQL Server Agent» строится с использованием компонента следующего типа: «Jobs» (задание). Компоненты этого типа описывают задания, которые выполняются автоматически в соответствии с установленным расписанием или вызываются вручную при необходимости. Кроме этого, задания могут вызываться в моменты простоя процессора. Для облегчения построения КС в базе данных заводятся две процедуры. Первая процедура «EXE_Sign» предназначена для работы с конкретной записью в таблице «Sign_of_condition». Другая процедура «EXE_ALL_Sign» представляет собой курсор по всем строчкам таблицы «Sign_of_condition», предназначенной для конкретной базы данных или для всех баз распределенной ИС. Данные в эту таблицу заносятся на центральном сервере и, если необходимо, реплицируются на другие распределенные серверы механизмами, описанными в [7, 8]. Периодически при запуске задания происходит проверка условий и изменения признаков состояний, которые осуществляются последовательным перебором. После отработки последнего зарегистрированного ПС в «Sign_of_condition» управление передается первому, а затем циклически просматриваются все условия признаков состояния и запускаются необходимые функциональные элементы.

Переход из одного состояния в другое возможен в результате выполнения какого-либо ФЭ или изменений данных с клиентских мест, или репликаций данных. Будем также считать, что каждый ФЭ выполняется только в том случае, когда база данных переходит в некоторое множество заранее определенных состояний, т.е. действия ФЭ являются реакцией ИС на переход в эти состояния.

Для описания структуры приложения любого уровня сложности было рассмотрено несколько типов элементов: набора функциональных элементов (ФЭ) и анализатора состояний. Классом особых объектов являлся элемент признак состояния (ПС), который принимает одно из двух значений: «установлен» или «сброшен». Работа данной системы может быть описана по показательному закону с параметром μ , т.е.

$$f(t) = \mu e^{-\mu t}, \quad t > 0. \quad (10)$$

Из этого следует, что поток обслуживания – простейший.

Обозначим данную систему через S , а ее состояния через S_0 – установлен, а S_1 – сброшен. Из состояния S_0 в S_1 систему, очевидно, переводит поток заявок с интенсивностью λ ; из S_0 в S_1 — «поток обслуживания» с интенсивностью μ .

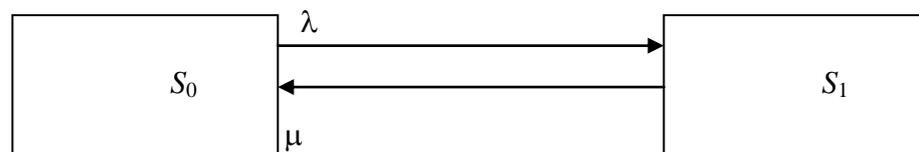


Рис. 2. Структурная схема системы S

Пусть вероятности состояний S_0 и S_1 соответственно равны $p_0(t)$ и $p_1(t)$. Очевидно, для любого момента t справедливо равенство

$$p_0(t) + p_1(t) = 1. \quad (11)$$

Составим дифференциальные уравнения Колмогорова для вероятностей состояний согласно правилу, указанному ранее:

$$\begin{cases} \frac{dp_0}{dt} = -\lambda p_0 + \mu p_1, \\ \frac{dp_1}{dt} = -\mu p_1 + \lambda p_0. \end{cases} \quad (12)$$

Из двух уравнений (12) одно является лишним, так как p_0 и p_1 связаны соотношением (11). Потому отбросим второе уравнение, а в первое подставим вместо p_1 выражение $(1-p_0)$ и придем к равенству:

$$\frac{dp_0}{dt} = -\lambda p_0 + \mu(1-p_0)$$

или

$$\frac{dp_0}{dt} = -(\mu + \lambda)p_0 + \mu. \quad (13)$$

Поскольку в начальный момент канал свободен, уравнение следует решать при начальных условиях:

$$p_0(0) = 1, p_1(0) = 0. \quad (14)$$

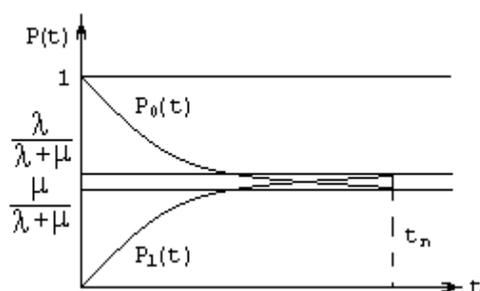


Рис. 3. Графики функций $p_0(t)$ и $p_1(t)$

Решение задачи Коши (11), (12) имеет вид:

$$p_0(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t};$$

$$p_1(t) = \frac{\lambda}{\lambda + \mu} (1 - e^{-(\lambda + \mu)t}).$$

Количественная оценка характеристик сервера, основанная на установившихся (предельных) вероятностях состояний, справедлива только лишь для моментов времени $t > t_n$, где t_n – время переходного процесса, начиная с которого вероятности состояний будут отличаться от своих предельных значений на достаточно малую величину.

Это время t_n можно легко оценить с помощью показателя экспоненты, равного $(\lambda + \mu)t$.

Библиографический список

1. Кулагина, Л.В. Разработка инструментария для создания много серверной системы ввода и обработки данных / Л.В. Кулагина, Н.В. Кулагин // Труды СВМО. 2006. № 2. С. 233–235.
2. Кулагина, Л.В. Разработка много серверной системы передачи и обработки данных / Л.В. Кулагина, Н.В. Кулагин // Труды СВМО. 2005. Т.7. № 1. С 421–422.
3. Кулагина, Л.В. Некоторые вопросы построения и программной реализации корпоративных информационных систем / Л.В. Кулагина, Н.В. Кулагин // Вестник Саратовского государственного технического университета. 2009. №1(37). С. 95–98.
4. Кулагин, Н.В. Математическая модель системы обработки информации и управления систем / Н.В. Кулагин, Т.Ф. Мамедова // Средневожское математическое общество. Препринт № 37. – Саранск.-2001.
5. Кулагина, Л.В. Механизм управления информационной системой на много серверной платформе систем / Л.В. Кулагина, Н.В. Кулагин // Научно-технические ведомости СПбГПУ. 2009. № 1(72). С. 134–137.

6. **Кулагина, Л.В.** Механизм управления много серверной информационной системой / Л.В. Кулагина, Н.В. Кулагин // Будущее технической науки: тез. докл. VIII Международной молодежной научно-технич. конф. / НГТУ. Н. Новгород. 2009. С. 68–70.
7. **Кулагина, Л.В.** Управление взаимодействием функциональных элементов в системах, построенных на распределенных базах данных систем / Л.В. Кулагина, Н.В. Кулагин // Труды СВМО. 2009. Т. 11. №1. С. 252–256.
8. **Ивченко, Г.И.** Теория массового обслуживания / Г.И. Ивченко, В.А. Каштанов, И.Н. Коваленко. – М.: Высш. шк. 1982. – 237 с.

*Дата поступления
в редакцию 01.02.2011*

L.V. Kulagina

**MATHEMATICAL MODEL OF A DISTRIBUTED DATABASE
FOR ENTERPRISE SYSTEMS**

In this paper the methods of information systems in a distributed database are considered. The mathematical notation of the data creation and transmission to distributed servers processes is suggested.

Key words: databases, information system, client an server technology, mathematical model, queneing theory.