УДК 004.75
doi:10.31799/1684-8853-2022-2-32-41

Научные статьи
Articles

Анализ средней задержки для одной модели сети массового обслуживания с резервированием ресурсов

М. Н. Шелест[a]***,*** *ассистент, orcid.org/0000-0002-2073-6053, mshshelest@mail.ru*
[a]*Санкт-Петербургский государственный университет аэрокосмического приборостроения, Б. Морская ул., 67, Санкт-Петербург, 190000, РФ*

Введение: *в настоящее время актуальной задачей является разработка новых методов анализа сложных информационных систем, к которым предъявляются повышенные требования по сохранению целостности данных. Важной характеристикой качества работы таких систем считается среднее время выполнения транзакции. К сожалению, на сегодня практически отсутствуют математические модели и оценки для быстродействия подобных систем.* ***Цель:*** *разработка и анализ модели распределенной информационной системы на базе сетей массового обслуживания.* ***Результаты:*** *описан тип информационных систем с повышенными требованиями к сохранению целостности данных. Приведены допущения для подобных систем. Предложен удобный способ их представления в виде графа зависимости маршрутов транзакций. Каждый маршрут в такой системе представлен в виде одной тандемной системы массового обслуживания, для которой приведено вычисление характеристик ее функционирования. Данный способ представления позволил упростить анализ сложных систем, в результате чего были получены замкнутые выражения для оценки временных характеристик рассматриваемого типа систем. Также приведены два механизма декомпозиции предложенного графа с последующим вычислением нижней границы для среднего времени выполнения транзакции. Методами имитационного моделирования проанализирована точность обоих подходов.* ***Практическая значимость:*** *предложенные модели позволяют оценить предельные скоростные характеристики информационной системы еще на этапе ее проектирования.*

Ключевые слова *— информационная система, транзакция, целостность, сеть массового обслуживания, граничные значения.*

Для цитирования: Шелест М. Н. Анализ средней задержки для одной модели сети массового обслуживания с резервированием ресурсов. *Информационно-управляющие системы*, 2022, № 2, с. 32–41. doi:10.31799/1684-8853-2022-2-32-41
For citation: Shelest M. N. Average delay estimation for one queueing network model with resource reservation. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 2, pp. 32–41 (In Russian). doi:10.31799/1684-8853-2022-2-32-41

Введение

В настоящее время актуальной задачей является анализ характеристик быстродействия больших информационных систем (ИС), состоящих из баз данных (БД), систем управления БД, каналов связи, пользовательских и прикладных интерфейсов и пр. [1, 2]. Информационный обмен между абонентами и ИС осуществляется посредством транзакций. В данном случае под транзакцией понимается последовательность операций над данными ИС. Помимо быстродействия, особое внимание в таких системах уделяется также сохранению целостности данных [3, 4]. Целостность может нарушаться, например, вследствие неправильно установленных политик доступа различных абонентов к одним и тем же записям БД для чтения и изменения. Например, во время процесса изменения данных в определенных записях одним пользователем к тем же данным для чтения может обращаться другой пользователь. Пока изменение записей первым пользователем не завершено, данные не могут считаться окончательными. Следовательно, данные, прочитанные вторым пользователем, будут неактуальными. Для избегания подобных ситуаций зачастую применяются механизмы резервирования (блокировки) записей или целых таблиц (ресурсов БД) [5, 6]. Основной смысл данных механизмов заключается в обеспечении порядка, при котором с одними и теми же данными одновременно может работать только одна транзакция [7].

Выделяют несколько уровней резервирования ресурсов.

1. *Мгновенное резервирование*: резервируется только тот ресурс, над которым выполняется операция в данный момент. При таком механизме транзакции блокируют относительно небольшое количество ресурсов, что способствует быстроте их выполнения. Минусом данного резервирования является возможное нарушение целостности данных.

2. *Ретроспективное резервирование*: резервируются те ресурсы, над которыми уже были выполнены операции в рамках транзакции, и тот ресурс, над которым выполняется операция в данный момент. Резервирование с ресурсов будет снято только тогда, когда выполнение транзакции полностью завершится. В данном случае целостность данных не нарушается, но такой тип резервирования может привести к возникновению взаимных блокировок. Взаимная блокировка — это ситуация, когда две или более транзак-

ции препятствуют выполнению друг друга [8]. Данный конфликт не разрешим без операции отката транзакций [9, 10], что негативно влияет на скорость работы ИС.

3. *Проспективное резервирование*: перед началом транзакции резервируются все ресурсы, над которыми планируется проведение операций. Снятие блокировки произойдет только тогда, когда выполнение данной транзакции полностью завершится. Благодаря подобному механизму резервирования в системе не возникают взаимные блокировки, так как невозможно начать выполнение ни одной транзакции, пока заблокирован хотя бы один необходимый ресурс. Такой механизм резервирования ресурсов также способствует полному сохранению целостности данных в ИС.

В настоящей работе рассматриваются ИС, к которым предъявляются повышенные требования к сохранению целостности данных без возможного возникновения взаимных блокировок, что делает необходимым применение третьего механизма резервирования.

Формальная модель сети

Представление системы в виде сети массового обслуживания

Введем модель рассматриваемой системы на основе сетей массового обслуживания (СеМО). При таком представлении ресурсам ИС соответствуют обслуживающие устройства (ОУ) сети, транзакциям — заявки, а процессу выполнения транзакции — прохождение заявки по маршруту внутри СеМО. В дальнейшем будут использованы следующие допущения:

— маршруты представляют собой простые цепи (внутри каждого маршрута все ОУ различны);

— множество разрешенных маршрутов в сети (типов транзакций) заранее определено и конечно;

— на множестве маршрутов задано вероятностное распределение.

Для каждого из типов транзакций на входе системы находится отдельный буфер. Контроллер СеМО, отвечающий за сохранение целостности данных, разрешает очередной заявке отправиться из буфера по маршруту, только если все ОУ на маршруте свободны. Одновременно производится блокировка данных ОУ для прочих заявок до тех пор, пока данная не покинет систему. Благодаря такому механизму резервирования на каждом маршруте может находиться не более одной заявки. Алгоритм, в соответствии с которым принимается решение о следующей взятой на обслуживание заявке, называется дисциплиной управления буфером [11, 12]. Такое решение, например, может основываться на принципе первый пришел — первый вышел или на принципе приоритезации наименее сложных заявок [13, 14].

Рассмотрим представление ИС в виде СеМО на примере системы, обрабатывающей четыре типа транзакций. Каждая из транзакций состоит из последовательности команд, представление которых на языке SQL приведено в Приложении.

1. Выполнение операции через банкомат:

— добавление в таблицу «Операции через банкомат» (*Tab*1) записи о проведении пользователем операции по снятию наличных через банкомат;

— добавление в таблицу «Операции клиентов» (*Tab*4) записи о проведенной пользователем операции с указанием даты.

2. Запрос на предоставление выписки по операциям:

— добавление в таблицу «Запросы клиентов» (*Tab*2) записи о запросе клиента выписки по счету на определенный день;

— выборка из таблицы «Операции клиентов» (*Tab*4) записей о совершенных пользователем операциях за указанный день.

3. Выполнение платежа по кредиту:

— добавление в таблицу «Платежи по кредиту» (*Tab*3) записи о внесении клиентом платежа по кредиту;

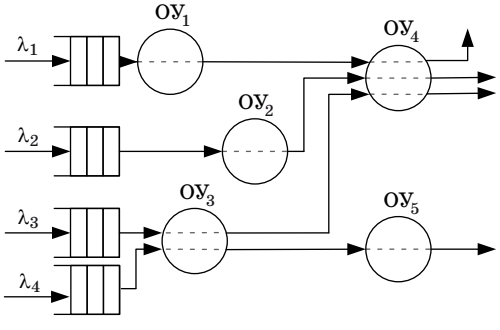
— добавление в таблицу «Операции клиентов» (*Tab*4) записи о проведенной пользователем операции с указанием даты.

4. Обновление кредитной истории:

— выборка из таблицы «Платежи по кредиту» (*Tab*3) суммы платежей по кредиту, внесенных клиентом за указанный месяц;

— обновление в таблице «Кредитная история» (*Tab*5) суммы оставшегося платежа по кредиту для клиента за указанный месяц.

Для данного примера ресурсами, над которыми выполняются операции транзакций, являются таблицы *Tab*1, *Tab*2, *Tab*3, *Tab*4 и *Tab*5. В СеМО им соответствуют $ОУ_1$, $ОУ_2$, $ОУ_3$, $ОУ_4$ и $ОУ_5$. На рис. 1, представляющем составленную таким об-



■ ***Рис. 1.*** Пример СеМО
■ ***Fig. 1.*** Queuing network example

разом результирующую сеть, λ_j обозначает интенсивность поступления в систему потока транзакций j-го типа ($\lambda_j = p_j L$, где p_j — вероятность появления транзакции j-го типа, L — общая интенсивность входного потока транзакций).

Опишем работу механизма резервирования для приведенного примера. Если в данный момент в сети исполняется заявка третьего типа, т. е. заявка, последовательно обслуживающаяся в $ОУ_3$ и $ОУ_4$, то эти обслуживающие устройства на ее маршруте будут заблокированы для остальных заявок. Тогда, поскольку $ОУ_3$ является частью 4-го маршрута, а $ОУ_4$ является частью 1-го и 2-го маршрутов, данные маршруты будут на это время заблокированы, и соответствующие им заявки будут пребывать в буферах.

Из логики приведенного механизма резервирования следует, что на каждом маршруте может находиться не более одной заявки. Следовательно, каждый маршрут можно представить в виде тандемной системы массового обслуживания (СМО). Тогда в случае, если входной поток заявок является пуассоновским, такая тандемная СМО сводится к системе массового обслуживания типа $M/G/1$ [15]. Полученная таким образом тандемная СМО будет обладать следующими характеристиками обслуживания:

— интенсивность обслуживания

$$\mu_j^{\Sigma} = \frac{1}{\sum\limits_{n \in R_j} \mu_n^{-1}}, \qquad (1)$$

где R_j — маршрут ($j = 1..S$), S — количество возможных маршрутов в сети; μ_n — интенсивность обслуживания n-го обслуживающего устройства исходной СеМО;

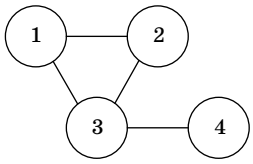
— плотность вероятности времени обслуживания

$$f_j(t) = \bigotimes_{n \in R_j} g_n(t), \qquad (2)$$

где символ $\otimes$ обозначает последовательную свертку плотностей вероятности времени обслуживания $g_n(t)$ для каждого обслуживающего устройства, входящего в состав маршрута R_j.

Представление системы в виде графа зависимости маршрутов

Для упрощения проведения анализа можно предложить альтернативное представление системы. Введем граф зависимости маршрутов (ГЗМ) $G(\mathbf{R}, \mathbf{E})$, вершины которого соответствуют множеству возможных маршрутов $\mathbf{R} = \{R_1, R_2, ..., R_s\}$. Ребра в данном графе будут ставиться согласно следующему правилу: $e_{a,b} \subset \mathbf{E}$, если $(R_a \cap R_b) \notin \varnothing$, т. е. ребра отражают наличие общих ОУ на паре маршрутов. На рис. 2 изображен граф зависимости маршрутов для СеМО, приведенной на рис. 1.



■ ***Рис. 2.*** Пример графа зависимости маршрутов
■ ***Fig. 2.*** Route dependency graph example

Тогда каждый узел в данном графе будет соответствовать тандемной СМО, параметры которой были описаны в предыдущем подразделе. Такое представление системы в виде ГЗМ содержит в себе всю необходимую информацию для проведения дальнейшего анализа ее характеристик быстродействия и производительности.

Заметим, что средняя задержка в СеМО с резервированием ресурсов напрямую зависит от общей интенсивности входного потока заявок L, набора маршрутов, вида ГЗМ (набора узлов и ребер $\mathbf{R}$ и $\mathbf{E}$) и от дисциплины управления буфером A. Далее будем предполагать, что A принадлежит множеству всех физически реализуемых дисциплин управления буфером $\mathcal{A}$. Следовательно, можно представить среднюю задержку в такой сети как некую функцию от перечисленных параметров:

$$\overline{T} = F(L, \mathbf{R}, \mathbf{E}, A). \qquad (3)$$

Уменьшение среднего времени пребывания заявки в сети при заданной интенсивности входного потока L возможно за счет оптимальной дисциплины управления буфером. Следовательно, можно записать следующее выражение для минимально возможно средней задержки в сети:

$$\overline{T}_{\text{opt}} = \inf_{A \in \mathcal{A}} F(L, \mathbf{R}, \mathbf{E}, A). \qquad (4)$$

Функцию, представленную в выражении (3), не всегда можно записать в замкнутом виде. Однако можно провести модификацию исходного ГЗМ путем удаления или добавления ребер. Тем самым могут быть найдены замкнутые выражения для вычисления граничных значений для средней задержки заявки в сети.

Исходя из всего вышеописанного, можно сформулировать следующую лемму.

Лемма. Для $\forall \mathbf{E}'$ и $\forall \mathbf{E}''$ таких, что $\mathbf{E}' \subseteq \mathbf{E} \subseteq \mathbf{E}''$, справедливо следующее неравенство:

$$\inf_{A \in \mathcal{A}} F(L, \mathbf{R}, \mathbf{E}', A) \le \overline{T}_{\text{opt}} \le \inf_{A \in \mathcal{A}} F(L, \mathbf{R}, \mathbf{E}'', A).$$

Здесь $\mathbf{E}'$ — неполное множество ребер исходного ГЗМ (крайний случай: $\mathbf{E}' = \varnothing$ — нуль-граф); $\mathbf{E}''$ — исходное множество ребер с некоторым количеством добавленных ребер (крайний случай: $\mathbf{E}'' = \mathbf{R} \times \mathbf{R}$ — полносвязный граф).

Доказательство: Доказательство основывается на том факте, что последовательность обслуживаемых заявок, совместимая с ограничениями ГЗМ $G(\mathbf{R}, \mathbf{E}'')$, будет также применима и для исходной сети с ГЗМ $G(\mathbf{R}, \mathbf{E})$, и, соответственно, для сети с ГЗМ $G(\mathbf{R}, \mathbf{E}')$.

Анализ нижней границы

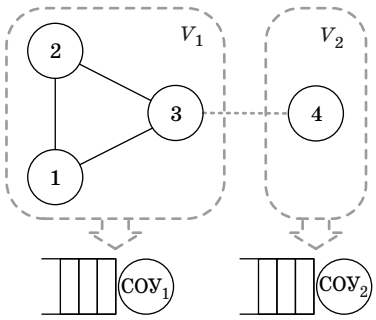
Основываясь на лемме, выведем общую формулу нижнего граничного значения для средней задержки в СеМО с резервированием ресурсов.

Согласно лемме, если удалить из исходного ГЗМ некоторые ребра и тем самым распараллелить работу некоторых тандемных СМО, то значение средней задержки в полученной сети будет не больше значения средней задержки в исходной СеМО. Таким образом, для получения замкнутого выражения для средней задержки в полученной сети удалим ребра так, чтобы исходный ГЗМ оказался разбит на отдельные полносвязные подграфы (рис. 3).

В каждом из полученных полносвязных подграфов на обслуживании одновременно может находиться только одна заявка (в силу взаимного пересечения всех маршрутов подграфа). Таким образом, можно также упростить каждый отдельный полносвязный подграф до составного ОУ (СОУ), характеристики обслуживания в котором вычисляются по следующим формулам:

— вероятность попадания заявки в СОУ

$$P_k = \sum_{i \in V_k} p_i, \qquad (5)$$



■ ***Рис. 3***. Пример разбиения исходного ГЗМ на отдельные полносвязные подграфы

■ ***Fig. 3***. An example of splitting initial the RDG into separate complete subgraphs

где V_k — множество маршрутов, входящих в k-й полносвязный подграф ($k = 1..K$, K — количество полносвязных подграфов); p_i — вероятность поступления заявки на i-й маршрут;

— интенсивность обслуживания в СОУ

$$M_k = \frac{P_k}{\sum\limits_{i \in V_k} p_i (\mu_i^{\Sigma})^{-1}}, \qquad (6)$$

где μ_i^{Σ} — интенсивность обслуживания заявки на i-м маршруте, вычисляется по формуле (1);

— интенсивность поступления заявок в СОУ

$$\Lambda_k = \sum_{i \in V_k} \lambda_i, \qquad (7)$$

где λ_i — интенсивность поступления заявок на i-й маршрут;

— плотность вероятности времени обслуживания в СОУ

$$\varphi_k(t) = \frac{1}{P_k} \sum_{i \in V_k} p_i f_i(t), \qquad (8)$$

где $f_i(t)$ — плотность вероятности времен обслуживания в i-м маршруте, вычисляется по формуле (2).

Поскольку очевидно, что $\mathbf{R} = \bigcup_{k=1}^{K} V_k$ и $\mathbf{E}' = \bigcup_{k=1}^{K} E_k'$ (здесь E_k' — ребра, входящие в состав k-го полносвязного подграфа), то можно представить выражение (4) в следующем виде:

$$\inf_{A \in \mathcal{A}} F(L, \mathbf{R}, \mathbf{E}', A) = \inf_{A \in \mathcal{A}} \sum_{k=1}^{K} \left[P_k F(\Lambda_k, V_k, E_k', A) \right].$$

Очевидно, что

$$\inf_{A \in \mathcal{A}} \sum_{k=1}^{K} P_k F(\Lambda_k, V_k, E_k', A) \geq$$
$$\geq \sum_{k=1}^{K} P_k \inf_{A \in \mathcal{A}} F(\Lambda_k, V_k, E_k', A).$$

Каждый полносвязный подграф, полученный в результате разбиения исходного графа, является системой типа $M/G/1$. Для систем подобного типа известно, что дисциплиной управления буфером, обеспечивающей минимальное среднее время нахождения заявки в сети, является дисциплина SJF (Shortest Job First — короткие заявки вперед) [11, 16]. Тогда справедливо следующее равенство:

$$\sum_{k=1}^{K} P_k \inf_{A \in \mathcal{A}} F(\Lambda_k, V_k, E_k', A) =$$
$$= \sum_{k=1}^{K} P_k F(\Lambda_k, V_k, E_k', SJF).$$

Таким образом, для нахождения нижней границы среднего времени пребывания заявки в сети можно записать часть леммы в следующем виде:

$$\inf_{A\in\mathcal{A}} F(L, \mathbf{R}, \mathbf{E}, A) \geq \sum_{k=1}^{K} P_k F(\Lambda_k, V_k, E'_k, SJF). \quad (9)$$

Каждый элемент суммы в правой части неравенства (9) может быть рассчитан по формуле Фиппса [17]:

$$F(\Lambda_k, V_k, E'_k, SJF) = \\ = \frac{\Lambda_k}{M_k}\int_0^\infty t^2\varphi_k(t)\mathrm{d}t \int_0^\infty \frac{\varphi_k(t)\mathrm{d}t}{\left(1-\Lambda_k\int_0^t x\varphi_k(x)\mathrm{d}x\right)^2} + \frac{1}{M_k}.$$

Таким образом, результирующее выражение для вычисления нижней границы среднего времени пребывания заявки в СеМО выглядит следующим образом:

$$\overline{T} \geq \sum_{k=1}^{K}\left[\frac{P_k\Lambda_k}{M_k}\int_0^\infty t^2\varphi_k(t)\mathrm{d}t \int_0^\infty \frac{\varphi_k(t)\mathrm{d}t}{\left(1-\Lambda_k\int_0^t x\varphi_k(x)\mathrm{d}x\right)^2} + \frac{P_k}{M_k}\right].$$

Способы уточнения нижней границы

Для упрощения процесса расчета нижнего граничного значения предлагается разбивать ГЗМ на отдельные полносвязные подграфы. Стоит отметить, что таких способов разбиения существует много, и от того, какой именно получился результат, будет зависеть точность полученного нижнего граничного значения для средней задержки в СеМО. Так, принимая во внимание лемму, частным случаем нахождения тривиальной нижней границы для средней задержки в сети является полное удаление ребер из ГЗМ ($\mathbf{E}' = \varnothing$).

В данной работе предлагается два способа разбиения (декомпозиции) ГЗМ на отдельные полносвязные подграфы с целью повысить точность предложенной нижней границы.

Первый способ разбиения ГЗМ будет называться VBS (Volume-Based Strategy). Данный способ основан на размере отдельных полносвязных подграфов.

Алгоритм 1 (VBS).

1. $k = 1$.
2. **while** $R \notin \varnothing$ **do**.
3. $k = K + 1$.
4. Поиск максимального по размеру полносвязного подграфа V_k.
5. $R = R\backslash V_k$.
6. **end**.

При таком разбиении ГЗМ на каждой итерации выбирается подграф с максимальным количеством вершин из всех найденных подграфов. Данная процедура выполняется до тех пор, пока весь граф не будет разбит на отдельные полносвязные подграфы.

Второй способ разбиения графа будет называться LBS (Load-Based Strategy). Данный способ основан на загруженности отдельных полносвязных подграфов.

Алгоритм 2 (LBS).

1. $k = 1$.
2. **while** $R \notin \varnothing$ **do**.
3. $k = K + 1$.
4. Поиск полносвязного подграфа V_k с максимальной загрузкой.
5. $R = R\backslash V_k$.
6. **end**.

В предложенном алгоритме разбиения ГЗМ на каждой итерации выбирается полносвязный подграф, в котором средняя задержка, рассчитанная по формуле Поллачека — Хинчина, будет максимальна [18].

Стоит отметить, что в случае, когда в сети имеются перегруженные узлы, описанные способы для получения граничных значений будут показывать результат по крайней мере не хуже, чем когда в сети нагрузка распределена равномерно. Данный эффект связан с тем, что при разбиении ГЗМ на отдельные полносвязные подграфы наиболее нагруженные узлы могут быть изолированы от остальной части сети, тем самым освобождая прохождение заявок по менее загруженным узлам. При этом разбиение ГЗМ с помощью алгоритма LBS даст наилучший эффект, потому что в данном случае разбиение основывается именно на загруженности узлов.

Численные примеры

Для СеМО, представленной на рис. 1, проведем сравнение нижних границ, полученных с помощью описанных способов разбиения ГЗМ.

Поскольку при использовании LBS-алгоритма одними из параметров, которые могут влиять на разбиение ГЗМ, являются вероятности выбора маршрутов в сети, то проведем вычисления при нескольких наборах данных вероятностей. В табл. 1 представлены вероятности выбора маршрутов, используемые для проведения серии тестов.

Первым этапом для поиска полносвязных подграфов будет являться нахождение всех клик

■ ***Таблица 1***. Параметры тестовых сценариев

■ ***Table 1***. Test case options

№ теста	Вероятности различных транзакций			
	p_1	p_2	p_3	p_4
1	0,25	0,25	0,25	0,25
2	0,06	0,06	0,06	0,82
3	$(1-q)/3$	$(1-q)/3$	$(1-q)/3$	q

(полносвязных подграфов, не являющихся частью более крупных полносвязных подграфов) в ГЗМ [19]. На рис. 4 приведены все клики, входящие в состав графа, изображенного на рис. 2.

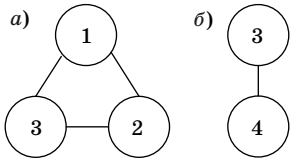
С другой стороны, так как алгоритм VBS не принимает во внимание вероятности выбора маршрутов, то результаты разбиения ГЗМ при использовании данного алгоритма для всех предложенных тестовых сценариев будут одинаковы. Клика № 1 будет являться первым найденным полносвязным подграфом, так как в ее состав входит наибольшее количество вершин среди представленных клик. Второй подграф будет состоять только из одной оставшейся вершины. Результат подобного разбиения был приведен ранее как пример на рис. 3.

Вероятности выбора маршрутов влияют на загруженность узлов графа, поэтому для того, чтобы определить, каким образом будет проведено разбиение ГЗМ при использовании алгоритма LBS, для каждого тестового сценария необходимо провести расчет загрузки для каждой найденной клики. В табл. 2 приведены расчеты среднего времени пребывания заявки в подсетях, образованных найденными кликами.

Параметры для расчета: $\mu_n = 1$; $L = 0{,}5$; $\lambda_s = 0{,}5p_s$; $g_n(t) = e^{-t}$.

В соответствии с выражениями (1) и (2) вычисляем: $\mu_s^{\Sigma} = 0{,}5$; $f_s(t) = te^{-t}$.

По результатам расчетов, приведенных в таблице, можно определить, что в первом тестовом сценарии среднее время пребывания заявок в подсети, образованной кликой № 1, больше, чем в подсети, образованной кликой № 2. Соответственно, клика № 1 выбирается в качестве первого полносвязного подграфа, а вторым



■ ***Рис. 4***. Клика № 1 (*а*) и клика № 2 (*б*), входящие в состав исходного ГЗМ

■ ***Fig. 4***. Clique No. 1 (*а*) and clique No. 2 (*б*) which are part of the initial route dependency graph

■ ***Таблица 2***. Расчеты для алгоритма LBS

■ ***Table 2***. Calculations for the LBS algorithm

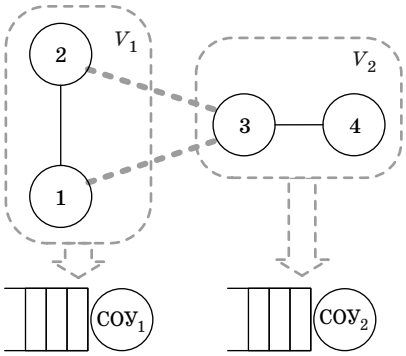
№ теста	Клика № 1	Клика № 2
1	Используя формулы (5)–(8), вычислим:	
	$P_1 = 0{,}75$; $M_1 = 0{,}5$; $\Lambda_1 = 0{,}375$; $\varphi_1(t) = te^{-t} \Rightarrow \nu_1 = 1/\sqrt{2}$	$P_2 = 0{,}5$; $M_2 = 0{,}5$; $\Lambda_2 = 0{,}25$; $\varphi_2(t) = te^{-t} \Rightarrow \nu_2 = 1/\sqrt{2}$
	Вычислим среднее время пребывания заявки в подсети, образованной кликой, используя формулу Поллачека — Хинчина:	
	$\overline{T}_1 = (\Lambda_1(1+\nu^2_1)/(2M_1(M_1\Lambda_1)) + 1/M_1 = 6{,}5$	$\overline{T}_2 = (\Lambda_2(1+\nu^2_2)/(2M_2(M_2\Lambda_2)) + 1/M_2 = 3{,}5$
2	Используя формулы (5)–(8), вычислим:	
	$P_1 = 0{,}18$; $M_1 = 0{,}5$; $\Lambda_1 = 0{,}09$; $\varphi_1(t) = te^{-t} \Rightarrow \nu_1 = 1/\sqrt{2}$	$P_2 = 0{,}88$; $M_2 = 0{,}5$; $\Lambda_2 = 0{,}44$; $\varphi_2(t) = te^{-t} \Rightarrow \nu_2 = 1/\sqrt{2}$
	Вычислим среднее время пребывания заявки в подсети, образованной кликой, используя формулу Поллачека — Хинчина:	
	$\overline{T}_1 = (\Lambda_1(1+\nu^2_1)/(2M_1(M_1\Lambda_1)) + 1/M_1 \approx 2{,}3293$	$\overline{T}_2 = (\Lambda_2(1+\nu^2_2)/(2M_2(M_2\Lambda_2)) + 1/M_2 = 13$

подграфом будет оставшаяся последняя вершина. Таким образом, разбиение ГЗМ на отдельные полносвязные подграфы в первом тестовом сценарии при алгоритме LBS не будет отличаться от разбиения по алгоритму VBS (см. рис. 3).

Расчеты по второму тестовому сценарию показали, что среднее время пребывания заявок в подсети, образованной кликой № 2, больше, чем в подсети, образованной кликой № 1. Поэтому в качестве первого полносвязного подграфа выбирается клика № 2, после чего остается последний полносвязный подграф, состоящий из двух вершин. На рис. 5 представлены результаты разбиения ГЗМ для второго тестового сценария при использовании алгоритма LBS.

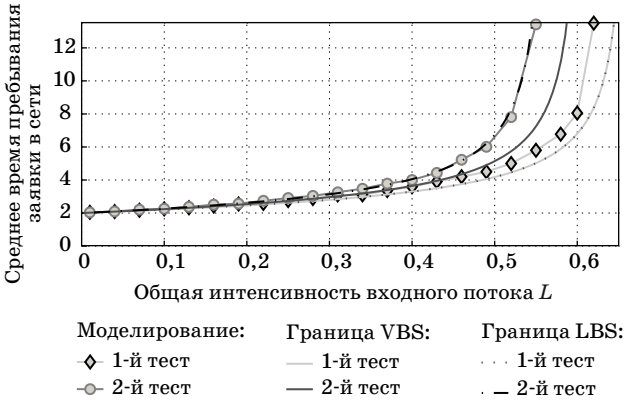
Проведено сравнение моделирования работы сети с нижними границами среднего времени пребывания заявки в сети, построенными при использовании VBS- и LBS-алгоритмов для двух тестовых сценариев (рис. 6).

На графике видно, что для первого тестового сценария граничные значения, полученные при помощи алгоритмов VBS и LBS, сливаются. Для второго тестового сценария разбиение ГЗМ при использовании двух описанных алгоритмов оказалось различным, что повлекло за собой разни-



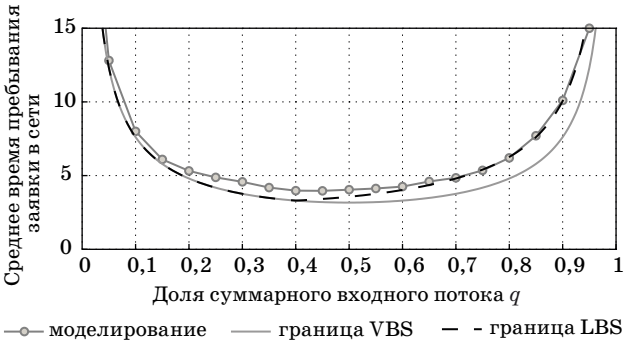
■ ***Рис. 5.*** Разбиение ГЗМ для второго тестового сценария при использовании LBS-алгоритма

■ ***Fig. 5.*** RDG partitioning for the second test case using the LBS algorithm



■ ***Рис. 6.*** Зависимость среднего времени пребывания заявки в системе от интенсивности общего входного потока

■ ***Fig. 6.*** Plot of the application average delay time in the system on the total input flow intensity



■ ***Рис. 7.*** Зависимость среднего времени пребывания заявки в системе от доли заявок, поступающих на 4-й маршрут

■ ***Fig. 7.*** Plot of application average delay time in the system on the share of applications arriving to the 4th route

цу в полученных границах для среднего времени пребывания заявки в СеМО. Наиболее выгодной для второго тестового сценария является граница, построенная с помощью LBS-алгоритма для разбиения ГЗМ (см. рис. 6), так как она полностью совпадает с результатом моделирования работы СеМО. В то же время вторая граница показала себя еще хуже, чем та же оценка для первого тестового сценария. Тем самым подтверждается вывод, что данные методы оценки работают лучше, когда в сети присутствуют перегруженные узлы.

Рассмотрим, как ведут себя полученные границы для среднего времени пребывания заявки в сети при фиксированной интенсивности входного потока и различных вероятностях выбора 4-го маршрута. Результаты моделирования работы СеМО при интенсивности входного потока $L = 0{,}5$ показаны на рис. 7.

Полученный график демонстрирует, что достаточно низкий перекос вероятностей в сторону одного из маршрутов (при $q \le 0{,}4$) не приводит к значительной выгоде использования стратегии LBS для разбиения ГЗМ на отдельные полносвязные подграфы. И напротив, на участке, где перекос вероятности значителен (при $q > 0{,}4$), стратегия VBS заметно проигрывает стратегии LBS в оценке среднего времени пребывания заявки в СеМО с резервированием ресурсов. На графике также видно, что при крайних значениях долей суммарного входного потока ($q < 0{,}05$ и $q > 0{,}95$) граничные значения практически полностью совпадают с результатами моделирования. Такой эффект возникает из-за того, что в этих случаях в системе образуются сильно перегруженные узлы. Тем самым задержка во всей сети практически полностью определяется задержкой только в этих перегруженных узлах, для которых возможно вывести замкнутое выражение для расчета средней задержки.

Заключение

На основе предложенной формальной модели СеМО с резервированием ресурсов предложен способ нахождения граничных значений для средней задержки в сети. В работе приведены несколько способов, благодаря которым может быть повышена точность полученных граничных значений. Данные способы основаны на применении разных подходов к разбиению исходного ГЗМ на отдельные полносвязные подграфы. Для проведения анализа полученных границ были рассмотрены численные примеры, которые позволяют сделать вывод о том, что наибольшей точности оценки среднего времени пребывания заявки в СеМО можно достичь применением стратегии разбиения LBS.

Полученные результаты могут быть использованы для анализа и оценки производительности таких систем управления БД, как MySQL, MS SQL и др., в которых используются механизмы блокировки транзакций, схожие с рассматриваемыми в данной работе [20, 21].

Приложение

Представление команд на языке SQL

1. Выполнение операции через банкомат:

— добавление в таблицу «Операции через банкомат» (*Tab*1) записи о проведении пользователем операции по снятию наличных через банкомат:

INSERT INTO Tab1 (Id_ATM_operation, Code_ATM, Type_operation, User, Amount) VALUES (7039, 0034156, 'Cash withdrawal', 0374, 3600);

— добавление в таблицу «Операции клиентов» (*Tab*4) записи о проведенной пользователем операции с указанием даты:

INSERT INTO Tab4 (User, Type_operation, Date, Code_operation) VALUES (0374, 'ATM operation', #18.11.2018#, 7039).

2. Запрос на предоставление выписки по операциям:

— добавление в таблицу «Запросы клиентов» (*Tab*2) записи о запросе клиента выписки по счету на определенный день:

INSERT INTO Tab2 (Id_query, Type_query, Code_operator, User, Condition) VALUES (3156, 'Statement of account for a day', 0047, 1501, #20.11.2018#);

— выборка из таблицы «Операции клиентов» (*Tab*4) записей о совершенных пользователем операциях за указанный день:

SELECT Date, Type_operation FROM Tab4 WHERE User = 1501 AND Date = #20.11.2018#.

3. Выполнение платежа по кредиту:

— добавление в таблицу «Платежи по кредиту» (*Tab*3) записи о внесении клиентом платежа по кредиту:

INSERT INTO Tab3 (Id_loan_payment, Code_credit, User, Amount_paid, Payment_date) VALUES (7153, 31568109, 0356, 4700, #10.11.2018#);

— добавление в таблицу «Операции клиентов» (*Tab*4) записи о проведенной пользователем операции с указанием даты:

INSERT INTO Tab4 (User, Type_operation, Date, Code_operation) VALUES (0356, 'Making a loan payment', #10.11.2018#, 7153).

4. Обновление кредитной истории:

— выборка из таблицы «Платежи по кредиту» (*Tab*3) суммы платежей по кредиту, внесенных клиентом за указанный месяц:

SELECT Sum(Amount_payment) as Sum1 FROM Tab3 WHERE User = 1092 AND Month(Payment_date) = 11;

— обновление в таблице «Кредитная история» (*Tab*5) суммы оставшегося платежа по кредиту для клиента за указанный месяц:

UPDATE Tab5 SET Month_payment = Month_payment − Sum1 WHERE User = 1092 AND Month = 11.

Литература

1. **Дубенко Ю. В., Дышкант Е. Е.** Нечеткая система определения оптимальных методов для прогнозирования параметров сложных технических систем. *Изв. вузов. Поволжский регион. Технические науки*, 2018, т. 47, № 3, с. 58–69. doi:10.21685/2072-3059-2018-3-6
2. **Муравьева-Витковская Л. А., Полторанин Р. В.** Оценивание временных характеристик функционирования корпоративных компьютерных сетей. *Изв. вузов. Приборостроение*, 2018, т. 61, № 3, с. 197–201. doi:10.17586/0021-3454-2018-61-3-197-201
3. **Lakshmanan V., Tigani J.** *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*. O'Reilly Media, 2019. 475 p.
4. **Campbell L., Majors C.** *Database Reliability Engineering: Designing and Operating Resilient Database Systems*. O'Reilly Media, 2017. 294 p.
5. **Калимолдаев М. Н., Бияшев Р. Г., Рог О. А.** Анализ методов атрибутного разграничения доступа. *Прикладная дискретная математика*, 2019, № 44, с. 43–57. doi:10.17223/20710410/44/4
6. **Богачев Д. Г.** Имитационная модель среды информационного обмена, включающая математическую модель резервирования ресурсов пакетной сети передачи данных с множественным доступом. *Вестник евразийской науки*, 2018, т. 10, № 1, с. 52–59. https://esj.today/ (дата обращения: 05.08.2021).
7. **Попов С. Г., Фридман В. С.** Обзор методов динамического распределения данных в распределенных системах управления базами данных. *Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление*, 2018, т. 11, № 4, с. 82–107. doi:10.18721/JCSTCS.11407. https://infocom.spbstu.ru/ (дата обращения: 05.08.2021).
8. **Tanenbaum A. S., Bos H.** *Modern Operating Systems*. Pearson, 2015. 1101 p.
9. **Challawala S., Mehta C., Patel K., Lakhatariya J.** *MySQL 8 for Big Data: Effective Data Processing with MySQL 8, Hadoop, NoSQL APIs, and Other Big Data Tools*. Packt Publishing, 2017. 226 p.
10. **Фомин Д. С., Бальзамов А. В.** Проблематика обработки транзакций при использовании микросервисной архитектуры. *Изв. вузов. Поволжский регион. Технические науки*, 2021, т. 58, № 2, с. 15–23. doi:10.21685/2072-3059-2021-2-2

11. **Telek M., Lakatos L., Szeidl L.** *Introduction to Queueing Systems with Telecommunication Applications*. Springer International Publishing. 2019. 559 p.
12. **Petrov A.** *Database Internals: A Deep Dive into How Distributed Data Systems Work*. O'Reilly Media, 2019. 376 p.
13. **Weiss G.** *Scheduling and Control of Queueing Networks*. Cambridge University Press, 2021. 200 p.
14. **Егорова Т. А., Муравьева-Витковская Л. А., Ли Шицзя.** Анализ процессов приоритетного управления потоками данных в программных системах. *Изв. вузов. Приборостроение*, 2019, т. 62, № 3, с. 208–2011. doi:10.17586/0021-3454-2019-62-3-208-211
15. **Куприянов Д. О.** Математическое моделирование потока заявок к облачному вычислительному кластеру. *T-Comm*, 2020, т. 14, № 10, с. 39–44. doi:10.36724/2072-8735-2020-14-10-39-44
16. **Горчакова Е., Зацаринная Ю. Н., Ушенина И.** Анализ критериев диспетчеризации и методов их оптимизации в операционных системах. *Вестник Казанского технологического университета*, 2015, т. 18, № 10, с. 155–157.
17. **Phipps T. E.** Machine repair as priority waiting-line problem. *Operations Res*, 1956, vol. 4, iss. 1, pp. 76–85.
18. **Shortle J. F., Harris C. M., Thompson J. M., Gross D.** *Fundamentals of Queueing Theory*. Wiley, 2018. 576 p.
19. **Pal M., Ghorai G., Samanta S.** *Modern Trends in Fuzzy Graph Theory*. Springer Singapore, 2020. 311 p.
20. **Delaney K.** *SQL Server Concurrency: Locking, Blocking and Row Versioning*. Simple Talk Publishing, 2012. 181 p.
21. **Филипенков А. В., Кузьмин Е. Л.** Сравнение существующих систем управления базами данных в целях выбора наилучшей при реализации требований по сокращению затрат и импортозамещению. *Газовая промышленность*, 2018, № 4(767), с. 24–29.

UDC 004.75
doi:10.31799/1684-8853-2022-2-32-41

Average delay estimation for one queueing network model with resource reservation

M. N. Shelest[a], Assistant Professor, orcid.org/0000-0002-2073-6053, mshshelest@mail.ru
[a]Saint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaia St., 190000, Saint-Petersburg, Russian Federation

Introduction: An urgent task of today is to develop new analysis methods for complex information systems that now demand higher standards for maintaining data integrity. One of the important quality characteristics of such systems is average transaction time. However, currently there are almost no mathematical models and speed estimation tools for such systems. **Purpose:** To develop and analyze a distributed information system model based on queueing networks. **Results:** A type of information systems that demand higher standards for maintaining data integrity has been described, the corresponding assumptions for such systems are given. A convenient way to represent such systems as transaction paths dependency graphs has been proposed, with each path being represented as one tandem queueing system. The calculation of their functional characteristics have been provided. This method of representation has made it possible to simplify the analysis of complex systems, which resulted in obtaining closed-form expressions for temporal estimation of the system type in question. In addition, two mechanisms of decomposition of the proposed graph are considered with the subsequent calculation of the lower bound for average transaction time. The accuracy of both approaches is analyzed with simulation modeling methods. **Practical relevance:** The proposed models allow estimating speed limits of an information system during the design phase.

Keywords — information system, transaction, integrity, queueing network, boundary values.

For citation: Shelest M. N. Average delay estimation for one queueing network model with resource reservation. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 2, pp. 32–41 (In Russian). doi:10.31799/1684-8853-2022-2-32-41

References

1. Dubenko Yu. V., Dyshkant E. E. Fuzzy system for determining optimal methods to forecast parameters of complex technical systems. *University Proceedings. Volga Region. Engineering Sciences*, 2018, vol. 47, no. 3, pp. 58–69 (In Russian). doi:10.21685/2072-3059-2018-3-6
2. Muraveva-Vitkovskaya L. A., Poltoranin R. V. Estimation of time characteristics of corporate computer networks functioning. *Journal of Instrument Engineering*, 2018, vol. 61, no. 3, pp. 197–201 (In Russian). doi:10.17586/0021-3454-2018-61-3-197-201
3. Lakshmanan V., Tigani J. *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*. O'Reilly Media, 2019. 475 p.
4. Campbell L., Majors C. *Database Reliability Engineering: Designing and Operating Resilient Database Systems*. O'Reilly Media, 2017. 294 p.
5. Kalimoldayev M., Biyashev R., Rog O. Analysis of the methods for attribute-based access control. *Applied Discrete Mathematics*, 2019, no. 44, pp. 43–57 (In Russian). doi:10.17223/20710410/44/4
6. Bogachev D. G. Simulation model of the information exchange environment, including a mathematical model for reservation of resources of a packet data network with multiple access. *Vestnik yevraziyskoy nauki*, 2018, vol. 10, no. 1, pp. 52–59 (In Russian). Available at: https://esj.today/ (accessed 5 August 2021).
7. Popov S. G., Fridman V. S. Review of methods for dynamic distribution of data in distributed database management systems. *St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*, 2018, vol. 11, no. 4, pp. 82–107 (In Russian). doi:10.18721/JCSTCS.11407. Available at: https://infocom.spbstu.ru/(accessed 5 August 2021).

8. Tanenbaum A. S., Bos H. *Modern Operating Systems*. Pearson, 2015. 1101 p.
9. Challawala S., Mehta C., Patel K., Lakhatariya J. *MySQL 8 for Big Data: Effective Data Processing with MySQL 8, Hadoop, NoSQL APIs, and Other Big Data Tools*. Packt Publishing, 2017. 226 p.
10. Fomin D. S., Bal'zamov A. V. The problem of transaction processing using microservice architecture. *University Proceedings. Volga Region. Engineering Sciences*, 2021, vol. 58, no. 2, pp. 15–23 (In Russian). doi:10.21685/2072-3059-2021-2-2
11. Telek M., Lakatos L., Szeidl L. *Introduction to Queueing Systems with Telecommunication Applications*. Springer International Publishing, 2019. 559 p.
12. Petrov A. *Database Internals: A Deep Dive into How Distributed Data Systems Work*. O'Reilly Media, 2019. 376 p.
13. Weiss G. *Scheduling and Control of Queueing Networks*. Cambridge University Press, 2021. 200 p.
14. Yegorova T. A., Muraveva-Vitkovskaya L. A., Li Shijia. Process analysis of data flow priority management in software systems. *Journal of Instrument Engineering*, 2019, vol. 62, no. 3, pp. 208–2011 (In Russian). doi:10.17586/0021-3454-2019-62-3-208-211
15. Kupriyanov D. O. Mathematical modeling of requests flow to cloud compute cluster. *T-Comm*, 2020, vol. 14, no. 10, pp. 39–44 (In Russian). doi:10.36724/2072-8735-2020-14-10-39-44
16. Gorchakova E., Zatsarinnaya Yu. N., Ushenina I. Analysis of dispatching criteria and their optimization methods in operating systems. *Vestnik Kazanskogo tekhnologicheskogo universiteta*, 2015, vol. 18, no. 10, pp. 155–157 (In Russian).
17. Phipps T. E. Machine repair as priority waiting-line problem. *Operations Res*, 1956, vol. 4, iss. 1, pp. 76–85.
18. Shortle J. F., Harris C. M., Thompson J. M., Gross D. *Fundamentals of Queueing Theory*. Wiley, 2018. 576 p.
19. Pal M., Ghorai G., Samanta S. *Modern Trends in Fuzzy Graph Theory*. Springer Singapore, 2020. 311 p.
20. Delaney K. *SQL Server Concurrency: Locking, Blocking and Row Versioning*. Simple Talk Publishing, 2012. 181 p.
21. Filipenkov A. V., Kuzmin E. L. Comparison of existing database management systems for selection of the best one by implementation of cost reduction and import substitution requirements. *GAZ Industry of Russia*, 2018, no. 4(767), pp. 24–29 (In Russian).

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.