

## Математическая модель функционирования распределённой информационной системы на базе трехуровневой клиент-серверной архитектуры

*А. Н. Скоба, Айеш Ахмед Нафеа Айеш (Ирак)*

*Южно-Российский государственный политехнический университет (НПИ)  
им.М. И. Платова, Новочеркасск*

**Аннотация:** В данной статье, с использованием аппарата замкнутых экспоненциальных сетей массового обслуживания (СеМО), разработана математическая модель для решения задачи получения интегральных показателей распределённой информационной системы на базе локальной вычислительной сети (ЛВС) с использованием трехуровневой клиент-серверной архитектуры. Приведены базовые соотношения для конструирования матриц переходных вероятностей и интенсивностей обслуживания в узлах сети.

**Ключевые слова:** распределённая информационная система, тонкий-клиент, сервер-приложений, сервер баз данных, концептуальная модель, матрица переходных вероятностей, интенсивности обслуживания, стационарная вероятность, уравнение глобального баланса, среднее время реакции системы на запросы “тонких-клиентов”.

В работе представлена математическая модель функционирования распределенной информационной системы (РИС) на базе ЛВС с использованием трехуровневой клиент-серверной архитектуры. Основным ее отличием от двухуровневой архитектуры “клиент-сервер” является физическое разделение программ, отвечающих за хранение данных (сервер баз данных – сервер БД) от программ эти данные обрабатывающих (сервер приложений – СП) [1]. Такое разделение компонент позволяет оптимизировать нагрузку как на сетевое, так и на вычислительное оборудование РИС [2]. С позиции программной реализации, данную архитектуру реализуют: сервера БД, например, *MySQL*-сервер; сервера приложений - *web*-сервера; роль клиента выполняет любой браузер.

Таким образом, согласно [1], работа РИС на базе ЛВС трехуровневой архитектуры “клиент-сервер” построена следующим образом: СУБД и БД в виде набора файлов размещаются на жестком диске специально выделенных компьютеров (сервера БД), на СП – размещается программное обеспечение делового анализа (бизнес-приложения); существует множество клиентских

компьютеров, на каждом из которых установлен так называемый “тонкий клиент” – клиентское приложение, реализующее интерфейс пользователя. На каждом из клиентских компьютеров пользователь имеет возможность запустить приложение – “тонкий клиент”, которое через пользовательский интерфейс обращается к программному обеспечению делового анализа – соответствующему бизнес-приложению, размещенному на СП. Бизнес-приложение анализирует требования пользователя и формирует запросы на языке *SQL* к БД, расположенных на серверах БД. Информация по *SQL* запросам копируется на СП, а затем, с помощью программного обеспечения делового анализа происходит ее окончательная обработка и возврат в клиентское приложение пользователя, которое используя пользовательский интерфейс, отображает результат выполнения запроса на экране компьютера.

Математическая постановка задачи. По сравнению с [3, 4] постановка задачи модифицируется следующим образом. Имеется ЛВС, включающая: множество клиентских компьютеров (клиентских приложений) –  $A = \{A_1, \dots, A_s, \dots, A_n\}$ ; множество серверов приложений (СП) –  $SP = \{SP_1, \dots, SP_s, \dots, SP_n\}$ ; множество серверов баз данных –  $DB = \{DB_1, \dots, DB_s, \dots, DB_n\}$ ; множество интенсивностей формирования запросов клиентскими приложениями к программному обеспечению делового анализа –  $A = \{\lambda_1, \dots, \lambda_s, \dots, \lambda_n\}$ ; множество запросов клиентских приложений на инициализацию соответствующих бизнес-приложений –  $Q = \{Q_1, \dots, Q_b, \dots, Q_q\}$ ; множество бизнес-приложений –  $r = \{r_1, \dots, r_b, \dots, r_q\}$ ; множество отношений (баз данных) –  $R = \{R_1, \dots, R_j, \dots, R_d\}$ ; множество объемов отношений –  $V = \{V_1, \dots, V_j, \dots, V_d\}$ ; скорости считывания данных в СП –  $VSP = \{VSP_1, \dots, VSP_s, \dots, VSP_n\}$ ; скорости записи данных в СП –  $DSP = \{DSP_1, \dots, DSP_s, \dots, DSP_n\}$ ; скорости считывания данных в серверах БД –  $VDB = \{VDB_1, \dots, VDB_s, \dots, VDB_n\}$ ; скорости записи данных в серверах БД –  $ddb = \{$

$DDB_1, \dots, DDB_s, \dots, DDB_n$ ; производительность процессов СП –  
 $PSP = \{PSP_1, \dots, PSP_s, \dots, PSP_n\}$ ; производительность процессоров серверов баз  
данных –  $PDB = \{PDB_1, \dots, PDB_s, \dots, PDB_n\}$ ; скорость передачи файлов по каналу  
связи –  $\theta$ ; постоянная задержка при передаче данных по каналу связи –

$\theta_0$ ; постоянная задержка при обработке данных в СП –  $\alpha_1$ ; постоянная  
задержка при обработке данных в сервере БД –  $\alpha_2$ ; матрица вероятностей  
формирования запросов клиентскими приложениями –

$F = \|A_{A_s Q_l}\| = \|f_{sl}\|, (s = \overline{1, n}, l = \overline{1, q})$ , где элемент  $f_{sl}$  представляет собой вероятность  
того, что  $s$ -й тонкий клиент сформировал  $l$ -й запрос, причём  $\sum_{l=1}^q f_{sl} = 1, (s = \overline{1, d})$ ;

матрица  $\Omega = \|\omega_{Q_l r_j}\| = \|\omega_{lj}\|, (l = \overline{1, q}, j = \overline{1, s})$ , где

$$\omega_{lj} = \begin{cases} 1, & \text{если } l\text{-й запрос инициализирует } j\text{-е бизнес-приложение,} \\ 0, & \text{в противном случае;} \end{cases}$$

матрица распределения бизнес-приложений по СП –

$Y = \|y_{r_j SP_i}\| = \|y_{ji}\|, (j = \overline{1, s}, i = \overline{1, n})$ , где

$$y_{ji} = \begin{cases} 1, & \text{если } r_j \in SP_i, \\ 0 - & \text{в противном случае, причём } \sum_{i=1}^n y_{ji} = 1, (j = \overline{1, n}); \end{cases}$$

матрица объемов считываемой информации  $B = \|b_{r_j R_j}\| = \|b_{ij}\|, (i = \overline{1, s}, j = \overline{1, d})$ , где  $b_{ij}$  –  
объем считываемой информации по  $SQL$  запросу, сформированного бизнес-  
приложением  $r_j$  к отношению  $R_j$ ;

$$\delta(B) = \|\delta(b_{r_j R_j})\| = \|\delta_{ij}\|, (i = \overline{1, s}, j = \overline{1, d}), \text{ где}$$

$$\delta_{ij} = \begin{cases} 1, & \text{если бизнес-приложение } r_j \text{ требует для своего выполнения} \\ & \text{отношения } R_j, \\ 0, & \text{в противном случае;} \end{cases}$$

матрица объемов информации, полученной после процессорной обработки бизнес-приложениями  $\bar{B} = \|\bar{b}_{r_i R_j}\| = \|\bar{b}_{ij}\|, (i = \overline{1, s}, j = \overline{1, d})$ , где  $\bar{b}_{ij}$  – объем информации получаемой после процессорной обработки бизнес приложения  $r_j$ , которое обращалось к отношению  $R_j$ ; матрица распределения отношений по серверам БД  $X = \|x_{R_j DB_k}\| = \|x_{jk}\|, (j = \overline{1, d}, k = \overline{1, n})$ , где

$$x_{jk} = \begin{cases} 1, \text{ если } R_j \in DB_k, \\ 0 - \text{ в противном случае, причём } \sum_{k=1}^n x_{jk} = 1, (j = \overline{1, d}). \end{cases}$$

*Допущения модели.*

- 1) Число СП совпадает с числом серверов БД, т.е.  $|SP| = |DB|$ ;
- 2) рассматривается случай, когда для инициализации бизнес-приложения  $r_j (j = \overline{1, s})$  требуется отношения  $R_k (k = \overline{1, d})$  такие, что  $r_j \in SP_i \Leftrightarrow R_k \in DB_i, (i = \overline{1, n})$ .

Концептуальная модель. По сравнению с [5] концептуальная модель функционирования РИС модифицируется: добавляются приборы –  $DB_1, \dots, DB_s, \dots, DB_n$ , моделирующие работу серверов БД; буферные памяти серверов БД –  $BDB_1, \dots, BDB_s, \dots, BDB_n$ ; приборы –  $SP_1, \dots, SP_s, \dots, SP_n$ , – моделирующие работу СП; буферные памяти СП –  $BSP_1, \dots, BSP_s, \dots, BSP_n$ .  
Концептуальная модель РИС представлена на рис. 1



$\{i_{2n+2,1}, \dots, i_{2n+s+1,s}, \dots, i_{2n+2,n}, \dots, i_{2n+s+1,1}, \dots, i_{2n+s+1,s}, \dots, i_{2n+s+1,n}, \dots, i_{3n+1,1}, \dots, i_{3n+1,s}, \dots, i_{3n+1,n}\}$ ,  $i = \overline{1, s}$ , где

$$i_{1s} = \begin{cases} 1, \text{ если } s\text{-й пользователь (тонкий клиент, } s = \overline{1, n}) \text{ находится в} \\ \text{активном состоянии (формирует запрос на инициализацию} \\ \text{бизнес - приложения),} \\ 0 - \text{ в противном случае;} \end{cases}$$

$\{i_{sr}, s = \overline{2, n+1}, r = \overline{1, n}\}$  – описывает очереди к каналу и состояние канала, где  $i_{sr}$  – количество запросов  $r$ -го пользователя (тонкого клиента) в  $s$ -ой буферной памяти канала и на обслуживании в канале;

$\{i_{sr}, s = \overline{n+2, 2n+1}, r = \overline{1, n}\}$  – описывает очереди запросов к серверам приложений и состояния серверов приложений, где  $i_{sr}$  – количество запросов  $r$ -го пользователя в буферной памяти  $s$ -го сервера приложений и на обслуживании в  $s$ -ом сервере приложений;

$\{i_{sr}, s = \overline{2n+2, 3n+1}, r = \overline{1, n}\}$  – описывает очереди к серверам баз данных и состояние серверов баз данных, где  $i_{sr}$  – количество сообщений  $r$ -го пользователя в буферной памяти  $s$ -го сервера баз данных и на обслуживании в  $s$ -ом сервере баз данных.

При этом имеют место следующие ограничения:

$$1) \sum_{k=1}^{3n+1} \sum_{r=1}^n i_{kr} = n \cdot r \cdot d;$$

$$2) \sum_{k=1}^{3n+1} i_{kr} = r \cdot d.$$

Представляющие интерес характеристически определяются стационарными вероятностями состояний сети. Пусть  $P(\bar{i})$  – стационарная вероятность того, что сеть находится в состоянии  $\bar{i}$ , где  $\bar{i} = (i_{11}, \dots, i_{1r}, \dots, i_{1n}, \dots, i_{3n+1,1}, \dots, i_{3n+1,r}, \dots, i_{3n+1,n})$ . В работах [6 - 8] было показано, что процесс изменения состояний сети описывается однородным регулярным

марковским процессом, и уравнение глобального баланса для стационарного режима функционирования такой сети будет иметь вид:

$$\sum_{k=1}^{3n+1} \sum_{r=1}^n P(\bar{i}) \mu_{kr} = \sum_{l=1}^{3n+1} \sum_{k=1}^{3n+1} \sum_{r=1}^n P(\bar{i} + \bar{1}_{lr} - \bar{1}_{kr}) \mu_{lr} P_{lk}(r),$$

где  $\mu_{sr} (r = \overline{1, n}, s = \overline{1, 3n+1})$  – интенсивность обслуживания в  $s$ -м центре сообщения  $r$ -го пользователя;  $P_{lk}(r), (l = \overline{1, 3n+1}, k = \overline{1, 3n+1}, r = \overline{1, n})$  – вероятность того, что сообщение  $r$ -го пользователя после обслуживания в  $l$ -м центре попадёт в  $k$ -й центр;  $\bar{1}_{sr} (s = \overline{1, 3n+1}, r = \overline{1, n})$  – вектор, в  $s$ -ой координате которого на  $r$ -ом месте стоит 1, а все остальные значения равны нулю.

Подробные расчет стационарных вероятностей состояний сети  $P(\bar{i})$  приведен в работах [3 - 5].

Конструирование элементов матриц переходных вероятностей  $P = \|P_{ik}(r)\|, (r = \overline{1, n}; i, k = \overline{1, 3n+1})$  по сравнению с [4] модифицируется следующим образом:

$$P_{ik}(r) = \begin{cases} 1 - \sum_{l=1}^q \sum_{j=1}^s f_{rl} \omega_{lj} y_{jr}, i = 1, k = r + 1; \\ \sum_{l=1}^q \sum_{j=1}^s f_{rl} \omega_{lj} y_{jr}, i = 1, k = n + r + 1; \\ 1, \text{ если } \sum_{l=1}^q \sum_{j=1}^s f_{rl} \omega_{lj} y_{jz} \neq 0, \text{ при } z = \overline{1, n}, i = z + 1, \\ i \neq r + 1, k = 1; z = \overline{1, n}, i = 2n + 1 + z, k = n + 1 + z; \\ \frac{\sum_{l=1}^q \sum_{j=1}^s f_{rl} \omega_{lj} y_{jz}}{\sum_{l=1}^q \sum_{j=1}^s \sum_{\substack{k=1 \\ k \neq r}}^d f_{rl} \omega_{lj} y_{jk}}, i = r + 1, z = \overline{1, n}, k = n + 1 + z, k \neq n + r + 1; \\ \sum_{l=1}^q \sum_{j=1}^s \sum_{\substack{k=1 \\ k \neq r}}^d f_{rl} \omega_{lj} \delta_{jk} x_{kz}, z = \overline{1, n}, i = n + 1 + z, k = 2n + 1 + z; \\ 1 - \sum_{l=1}^q \sum_{j=1}^s \sum_{\substack{k=1 \\ k \neq r}}^d f_{rl} \omega_{lj} \delta_{jk} x_{kz}, z = \overline{1, n}, i = n + 1 + z, k = z + 1; \\ 0, \text{ во всех остальных случаях.} \end{cases}$$

При расчете интенсивностей обслуживания  $\mu_{sr}(s = \overline{1, 3n + 1}, r = \overline{1, n})$  по сравнению с [5, 9] были выделены две группы потока заявок: первая группа включает запросы, формируемые  $r$ -м пользовательским приложением на инициализацию бизнес-приложений, размещенных на  $r$ -ом сервере приложений, которые в свою очередь, формируют  $SQL$ -запросы к базам данных, размещенных на  $r$ -ом сервере баз данных. Для запросов данной группы расчет интенсивностей обслуживания заявок основывается на вычислении следующих величин:

$$\mu_{sr} = \left( \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} b_{rjk} x_{kr}}{PSP_r} + \alpha_1 \right)^{-1}, r = \overline{1, n}, z = \overline{1, n}, s = n + 1 + z; z = r;$$

$$\mu_{sr} = \left( \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} b_{jk} x_{kr}}{VDB_r} + \alpha_2 \right)^{-1}, r = \overline{1, n}, z = \overline{1, n}, s = 2n + 1 + z; z = r;$$

вторая группа включает запросы, формируемые  $r$ -м пользовательским приложением на инициализацию бизнес-приложений, размещенных на  $r$ -м сервере приложений, которые, в свою очередь формируют  $SQL$ -запросы к базам данных, размещенных на других (отличных от  $r$ ) серверах баз данных. Для них имеют место выражения вида:

$$\mu_{sr} = \left( \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} b_{jk} \sum_{z=1}^n x_{kz}}{DBS_r} + \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} b_{rjk} \sum_{z=1}^n x_{kz}}{PSP_r} + \alpha_1 \right)^{-1},$$

$$r = \overline{1, n}, z = \overline{1, n}, s = n + 1 + z; z \neq r;$$

$$\mu_{sr} = \left( \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} \delta_{jk} V_k \sum_{z=1}^n x_{kz}}{VDB_z} + \alpha_2 \right)^{-1},$$

$$r = \overline{1, n}, z = \overline{1, n}, s = 2n + 1 + z, z \neq r.$$

При расчете интенсивностей обслуживания заявок в канале передачи данных будем полагать, что самым объемом запроса на инициализацию соответствующего бизнес-приложения можно пренебречь, т.е.  $\mu_{sr} = \theta_0^{-1}, r = \overline{1, n}, z = \overline{1, n}, s = 1 + z, z = r$ , а при передаче ответных сообщений на терминале  $r$ -го тонкого клиента, будем считать, что

$$\mu_{sr} = \left( \theta_0 + \frac{\sum_{l=1}^q f_{rl} \sum_{j=1}^s \sum_{k=1}^d \omega_{lj} y_{jk} \bar{b}_{jk} \sum_{z=1}^n x_{kz}}{\theta} \right)^{-1}, r = \overline{1, n}, z = \overline{1, n}, s = 1 + z, z \neq r.$$

В работах [3 - 5] показано, что расчет среднего времени реакции системы на запросы пользователя  $\bar{T}$  сводится по существу к расчету нормализующей константы  $G$ , для вычисления которой может быть использован рекуррентный метод Бузена [6 - 10].

### Литература

1. Карпов А.Е. Архитектура распределенных систем программного обеспечения, М., МАКС Пресс, 2007. -130с.
2. Богусловский Л.Б., Ляхов А.И. Оценка производительности распределенных информационно-вычислительных систем архитектуры “КЛИЕНТ-СЕРВЕР” / Автоматика, телемеханика. – 1995. – С.160-175.
3. Скоба А.Н., Состина Е.В. Математическая модель оптимального размещения распределенной базы данных по узлам ЛВС на базе файл-серверной архитектуры. // Инженерный вестник Дона. 2015. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2015/2881.
4. Скоба А.Н., Состина Е.В. Математическая модель оптимального размещения распределенной базы данных по узлам ЛВС на базе двухуровневой клиент-серверной архитектуры. // Инженерный вестник Дона. 2015. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2015/2882.
5. Скоба А.Н., Панфилов А.Н. Модель оптимального размещения информационных ресурсов по узлам распределенной информационной системы предприятия на базе двухуровневой архитектуры “клиент-сервер” с учетом влияния блокировок // Изв. вузов. Электромеханика. 2017. Т. 60, № 2. С. 77-84.

6. Жожикашвили В.А., Вишневский В.М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. - М.: Радио и связь, 1988. - 192с.
7. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. - М.: Техносфера, 2003. - 512 с.
8. Chakka R., Harrison P.G. A Markov modulated multi-server queue with negative customers –the MM CPP/GE/c/LG-queue // Acta Informatika/-2001.-v.37. pp. 785-799.
9. Черноморов Г.А. Теория принятия решений: Учебное пособие / Юж.-Рос.гос. техн.ун-т.-3-е изд.перераб.и доп.-Новочеркасск: Ред.журн.-«Иzv. Вузov. Электрoомеханика»,2005. -448с.
10. Buzen J.P. Computational Algorithms for Closed Queueing Networks with Exponential Servers. Commun. ACM. 1983. Vol.16, №9.pp.527-531.

### References

1. Karpov A.E. Arhitektura raspredel'nykh sistem programmnoho obespechenija. [The architecture of distributed software systems]. MAKS Press, 2007. 130 p.
  2. Boguslavsky L. B., Lyakhov A. I. The performance evaluation of distributed information systems of the "Client-server» architecture. Automation and telemechanics.1995. pp. 160-175.
  3. Skoba A.N., Sostina E.V. Inženernyj vestnik Dona (Rus), 2015. №2.URL:ivdon.ru/ru/ magazine/archive/n2y2015/2881.
  4. Skoba A.N., Sostina E.V. Inženernyj vestnik Dona (Rus), 2015. №2. URL:ivdon.ru/ru/ magazine/archive/n2y2015/2882.
  5. Skoba A.N., Panfilov A.N. Izv. vuzov. Jelektromehanika. 2017. V.60, №2. pp.77-84.
-



6. Zhozhikashvili V.A., Vishnevskiy V.M. Seti massovogo obsluzhivaniya. Teoriya i primeneniye k setyam EVM [Queueing networks. Theory and its network application]. M. Radio i svyaz', 1988. 192 p.
7. Vishnevskiy V.M. Teoreticheskie osnovy proektirovaniya komp'yuternykh setey [Theoretical foundations of computer network design]. M. Tekhnosfera, 2003. 512 p.
8. Chakka R., Harrison P.G. A Markov modulated multi-server queue with negative customers. Jhe MM CPP/GE/c/LG-queue. Acta Informatika.2001. v.37.pp.785-799.
9. Chernomorov G.A. Teoriya prinyatiya resheniy [Decision-making theory]: Uchebnoe posobie. Yuzh. Ros.gos.tekhn. un-t. 3-e izd. pererab. i dop.Novocherkassk: Red. zhurn."Izv.vuzov.Elektromekhanika", 2005. 448 p.
10. Buzen J.P. Computational Algorithms for Closed Queueing Networks with Exponential Servers. Commun. ACM. 1983. Vol.16, №9. pp.527-531.