

Программная реализация моделей оптимального распределения информационных ресурсов

В. К. Михайлов, Айеш Ахмед Нафеа Айеш (Ирак), А. Н. Скоба

Южно-Российский государственный политехнический университет (НПИ) им.

М. И. Платова, Новочеркасск

Аннотация: В данной статье описаны этапы программной реализации задачи вычисления среднего времени реакции распределённой системы обработки информации (СОИ) на запросы пользователей и алгоритма оптимального распределения информационных отношений по узлам распределённой СОИ, реализованной на базе архитектуры «файл-сервер» и двухуровневой архитектуры «клиент-сервер» с учетом и без учета влияния блокировок. Описана процедура конструирования исходных данных для моделирования: элементов матрицы начального распределения информационных ресурсов (отношений) по узлам распределённой СОИ, элементов матрицы информационной взаимосвязи запросов пользователей распределённой СОИ и самих информационных ресурсов, элементов матрицы вероятностей формирования пользователями информационных запросов. Разработана и программно реализована процедура, позволяющая существенно упростить вычисление нормализующей константы, а соответственно, и среднего времени реакции системы на запросы пользователей. Приведены результаты численных экспериментов.

Ключевые слова: распределённая система обработки информации, архитектура «файл-сервер», двухуровневая архитектура «клиент-сервер», нелинейная комбинаторная оптимизация, генератор равновероятного выбора, метод Жордана-Гаусса, нормализующая константа, среднее время реакции системы, блокировка отношения.

Ранее, в работах [1-3], были представлены разработанные авторами математические модели функционирования распределённых систем обработки информации (СОИ) на базе архитектуры «файл-сервер» и двухуровневой архитектуры «клиент-сервер» без учета и с учетом влияния блокировок для решения задач оптимального распределения информационных ресурсов.

В качестве исходных данных для моделирования были выбраны: множество узлов сети – $U=\{U_1, \dots, U_s, \dots, U_n\}$; множество пользователей – $A=\{A_1, \dots, A_s, \dots, A_n\}$; множество отношений – $R=\{R_1, \dots, R_j, \dots, R_d\}$; множество интенсивностей формирования запросов – $\lambda=\{\lambda_1, \dots, \lambda_s, \dots, \lambda_n\}$; множество запросов на чтение – $Q=\{Q_1, \dots, Q_b, \dots, Q_d\}$; множество объёмов отношений – $V=\{V_1, \dots, V_j, \dots, V_d\}$; скорость считывания в узлах – $VV=\{VV_1, \dots, VV_s, \dots, VV_n\}$;

скорость записи в узлах – $VD = \{VD_1, \dots, VD_s, \dots, VD_n\}$; скорость передачи данных по каналу связи – θ ; постоянная задержка при передаче данных по каналу связи – θ_0 ; постоянная задержка при обработке в узле – α_0 ; производительность процессора U_z -го узла – $PU_z, (z = \overline{1, n})$; матрица вероятностей формирования запросов пользователями – $F = \|f_{s,l}\| = \|f_{sl}\|, (s = \overline{1, n}, l = \overline{1, q})$, где элемент f_{sl} представляет собой вероятность того, что s -й пользователь сформировал l -й запрос; матрица объёмов считываемой информации $B = \|b_{Q,R_j}\| = \|b_{lj}\|, (l = \overline{1, q}, j = \overline{1, d})$, где b_{lj} – объём считываемой информации по l -му запросу на чтение из j -го отношения; $\delta(B) = \|\delta(b_{Q,R_j})\| = \|\delta_{lj}\|, (l = \overline{1, q}, j = \overline{1, d})$, где

$$\delta_{lj} = \begin{cases} 1, \text{ если } l\text{-й запрос обращается к } j\text{-му отношению,} \\ 0, \text{ такого отношения нет,} \end{cases}$$

причём $\sum_{j=1}^d \delta_{lj} = 1, l = \overline{1, q}$; матрица распределения отношений по узлам

распределённой СОИ- $X = \|x_{R_j, U_k}\| = \|x_{jk}\|, (j = \overline{1, d}, k = \overline{1, n})$, где

$$x_{jk} = \begin{cases} 1, \text{ если } R_j \in U_k, \\ 0 - \text{ в противном случае,} \end{cases}$$

причём $\sum_{k=1}^n x_{jk} = 1, j = \overline{1, d}$.

В общем виде задача оптимального распределения информационных ресурсов по критерию минимума среднего времени реакции системы на запросы пользователей была сведена к задаче [4]:

$$\left. \begin{aligned} \bar{T} = \bar{T}(U, R, X, \Lambda, V, Q, VV, VD, \theta, PU, F, B) \rightarrow \min \\ \text{при ограничении} \\ \sum_{k=1}^n x_{jk} = 1, j = \overline{1, d}. \end{aligned} \right\} \quad (1)$$

Как показал проведенный анализ, задача (1) является задачей нелинейной комбинаторной оптимизации с булевыми переменными. Ввиду того, что функция \bar{T} имеет сложный вид, а так же, ввиду отсутствия в настоящее время алгоритмов решения такого класса задач (кроме как метод полного перебора) [5], для решения задачи (1) был применен разработанный авторами эвристический алгоритм, подробное описание которого представлено в работе [4].

Таким образом, программная реализация задачи (1) включала в себя следующие этапы: процесс конструирования исходных данных; конструирование элементов матриц переходных вероятностей $\|P_{ik}(r)\|$, где $(i, k = \overline{1, 2n+1}; r = \overline{1, n})$; процесс вычисления интенсивностей обслуживания в узлах сети μ_{sr} , $(s = \overline{1, 2n+1}, r = \overline{1, n})$; решение систем линейных алгебраических уравнений вида $e_{sr} = \sum_{j=1}^{2n+1} e_{jr} P_{js}(r)$, $s = \overline{1, 2n+1}, r = \overline{1, n}$; вычисление величины \bar{T} ; реализацию эвристического алгоритма.

В процессе формирования исходных данных, было необходимо сформировать матрицу начального распределения информационных отношений по узлам распределённой СОИ – $X = \|x_{jk}\|$, где $(j = \overline{1, d}; k = \overline{1, n})$; матрицу $\delta(B) = \|\delta(b_{Q_l R_j})\| = \|\delta_{lj}\|$, где $(j = \overline{1, d}; l = \overline{1, q})$, описывающую информационную связь между запросами и отношениями; матрицу $F = \|f_{A_s Q_l}\| = \|f_{sl}\|$, где $(s = \overline{1, n}; l = \overline{1, q})$ – вероятностей формирования запросов пользователями.

Конструирование элементов данных матриц было проведено с использованием программно реализованного «генератора равновероятного выбора», работающего по следующему алгоритму. Задаются числа: n (число узлов распределённой СОИ), q (число информационных запросов), d (количество информационных отношений, подлежащих распределению по узлам распределённой СОИ). Далее, интервал $(0,1)$ разбивается на n

подинтервалов, равной длины, при этом, каждому подинтервалу будет соответствовать конкретный узел распределенной СОИ, запрос или информационный фрагмент. С помощью стандартных процедур, генерируются случайные числа, равномерно распределенные на интервале $(0,1)$. Очередное случайное число представляется в виде: $\xi_1 = \{A\xi_0\}$, где $\{C\}$ – операция определения дробной части от произведения чисел $A \in [11111110; 44444445]$ и $\xi_0 = 0,19091994$. Определяется номер подинтервала, в который попадает полученное число ξ_1 , которые и определяют соответствующие элементы матриц $X, \delta(B), F$. После этого, генерируется следующее случайное число $\xi_2 = \{A\xi_1\}$ и т.д. В общем случае $\xi_j = \{A\xi_{j-1}\}$.

Расчет элементов матриц переходных вероятностей $\|P_{ik}(s)\|, (i, k = \overline{1, 2n+1}, s = \overline{1, n})$ и интенсивностей обслуживания в узлах сети $\mu_{sr}, (s = \overline{1, 2n+1}, r = \overline{1, n})$ был произведён по формулам, приведенным в работах [1-3]. Система линейных алгебраических уравнений вида $e_{sr} = \sum_{j=1}^{2n+1} e_{jr} P_{js}(r), s = \overline{1, 2n+1}, r = \overline{1, n}$ была решена методом Жордана-Гаусса, что позволило определить точное значение всех переменных.

Как было показано в работах [6-10], расчёт величины \bar{T} сводится по существу к расчёту нормализующей константы $G(N_1, \dots, N_n)$, для вычисления которой был использован рекуррентный метод Бузена [11], который сводит расчёт нормализующей константы к итерационной процедуре вида:

$$G_m(\bar{n}) = G_{m-1}(\bar{n}) + \sum_{r=1}^n x_{mr} G_m(\bar{n} - \bar{1}_r), \quad (2)$$

где $\bar{n} = (n_1, \dots, n_r), n_i = \overline{0, 1}; x_{mr} = \frac{e_{mr}}{\mu_{mr}}, (m = \overline{1, 2n+1}; r = \overline{1, n})$. Предполагается также, что $G_m(\bar{n}) = 1$, если все $n_i = 0, (i = \overline{1, n})$ и $G_m(\bar{n} - \bar{1}_r) = 0$, если хотя бы одна из координат вектора $\bar{n} - \bar{1}_r < 0$. Значение нормализующей константы

$G(N_1, \dots, N_n)$ получается по формуле (2) при $m = 2n + 1$ и $\bar{n} = (N_1, \dots, N_n)$, где $N_i = 1, (i = \overline{1, n})$. Используемая процедура расчёта величины $G(N_1, \dots, N_n)$ имеет трудоёмкость порядка $\sim O(mn2^{2n})$. Авторами была разработана и программно реализована оригинальная процедура вычисления величины $G_m(\bar{n})$, связанная с переходом к представлению векторов \bar{n} и $\bar{n} - \overline{1_r}$ в десятичной системе счисления, благодаря чему отпала необходимость, при выполнении операции поиска значений $G_{m-1}(\bar{n})$ и $G_m(\bar{n} - \overline{1_r})$ в массиве, сравнивать эти вектора до достижения соответствия между их разрядами. Предложенная процедура свелась к стандартному обращению к элементу массива по его индексу. В связи с этим произошло снижение трудоёмкости до $\sim O(m2^n)$, и как следствие, снижение общего времени работы алгоритма нахождения нормализующей константы. Так, например, для $n=20$ и $d=30$, $q=40$ нормализующая константа $G = 2.79542 * 10^{44}$ была найдена за $t \approx 180$ секунд, а при поиске значений в массиве по их векторам – процедура вычисления могла занимать несколько десятков часов.

Программная реализация задачи (1) была проведена на языке C#, платформе .NET Framework 4.6.1, и компьютере на базе процессора фирмы Intel, с тактовой частотой 1,8 ГГц, объемом оперативной памяти 8 ГБ, при численных значениях исходных данных, приведённых в работе [4].

С использованием разработанного программного продукта был проведён ряд численных экспериментов, результаты которых могут быть использованы разработчиками при выборе рациональной организации вычислительного процесса на базе распределённой СОИ.

На рисунках 1-2 приведены графики зависимости \bar{T} от размерности решаемых задач оптимального распределения информационных отношений по узлам распределённой СОИ – $\bar{n}_i = (n \times q \times d), i = \overline{1, 6}$, где $\bar{n}_1 =$

$(5 \times 10 \times 10)$, $\bar{n}_2 = (7 \times 30 \times 20)$, $\bar{n}_3 = (9 \times 50 \times 30)$, $\bar{n}_4 = (11 \times 70 \times 40)$,
 $\bar{n}_5 = (13 \times 90 \times 50)$, $\bar{n}_6 = (15 \times 100 \times 60)$.

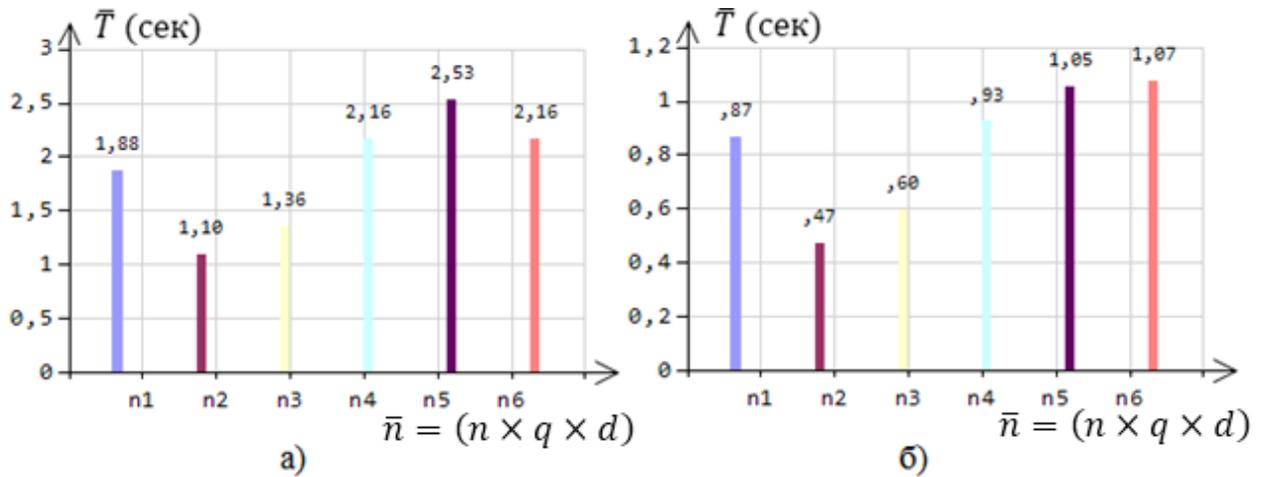


Рис.1. Зависимость \bar{T} от размерности решаемых задач для архитектуры «файл-сервер»: а – с учетом блокировок; б – без учета блокировок

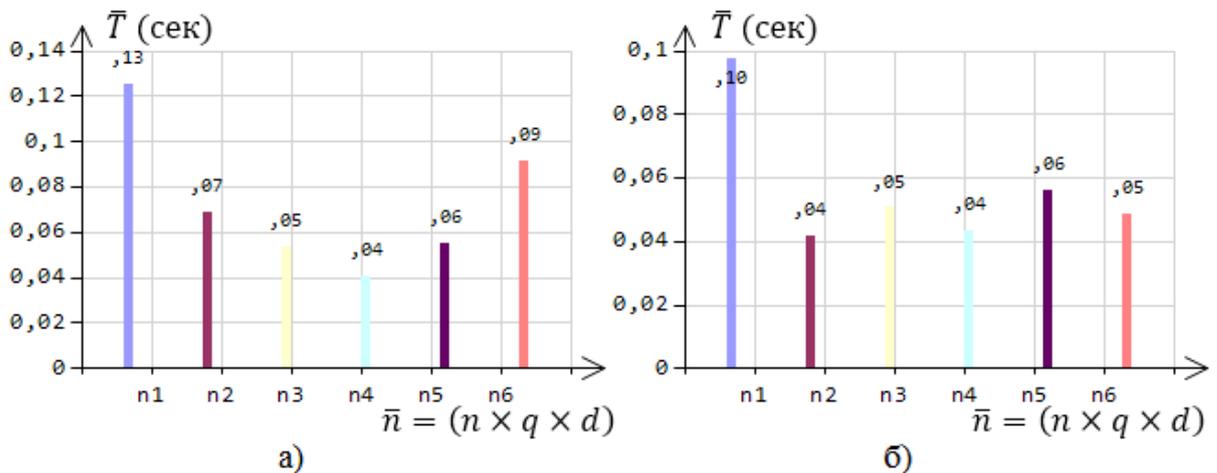


Рис. 2. Зависимость \bar{T} от размерности решаемых задач для двухуровневой архитектуры «клиент-сервер»: а – с учетом блокировок; б – без учета блокировок

Вид графиков показывает, что учёт блокировок для распределенной СОИ, реализованных на базе двухуровневой архитектуры «клиент-сервер», при заданных параметрах системы, не оказывает существенного влияния на реактивность системы (рис. 2), однако, при использовании архитектуры «файл-сервер», влияние блокировок на величину \bar{T} становится более

значительным (рис. 1) и её необходимо учитывать при проектировании реальных распределенных СОИ.

На рисунках 3-4 представлены графики зависимости величины \bar{T} от скорости считывания в узлах VV (КБ/сек) при различных размерностях задач \bar{n}_i и при фиксированной скорости передачи данных по каналу $\theta=5000$ (КБ/сек).

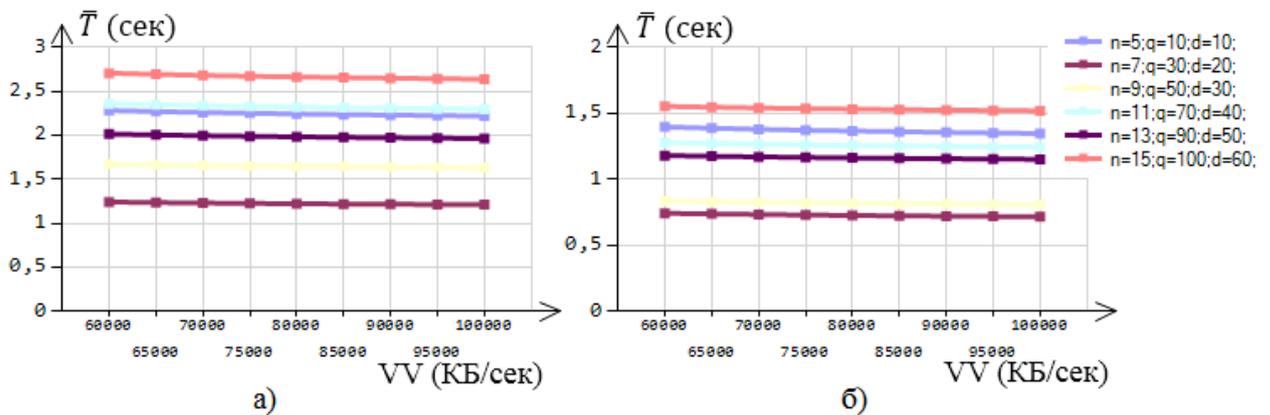


Рис.3. Зависимость \bar{T} от скорости считывания в узлах для архитектуры «файл-сервер»: а – с учетом блокировок; б – без учета блокировок

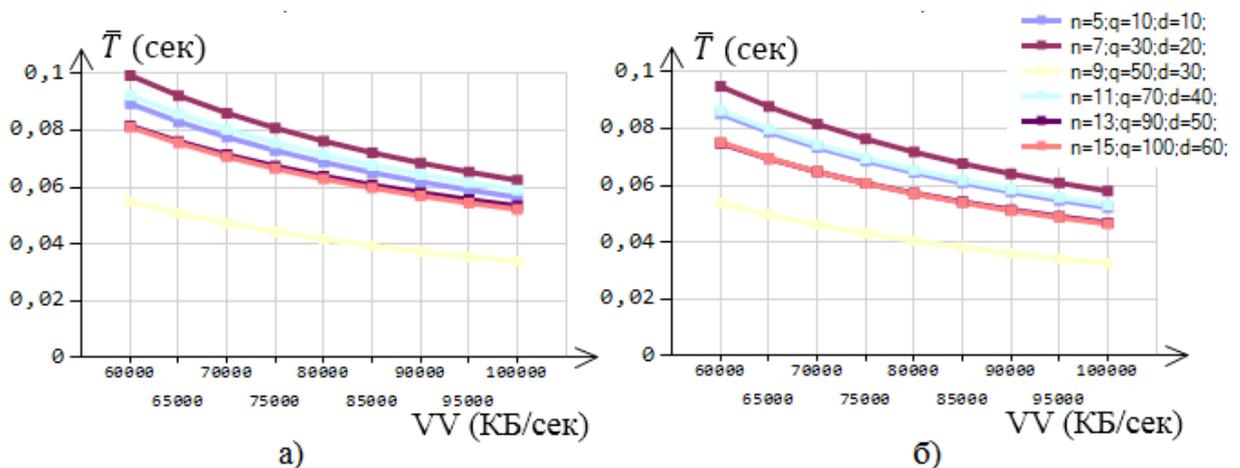


Рис.4. Зависимость \bar{T} от скорости считывания в узлах для двухуровневой архитектуры «клиент-сервер»: а – с учетом блокировок; б – без учета блокировок

Анализ графиков данных зависимостей показывает, что использование двухуровневой архитектуры «клиент-сервер» приводит к значительно более быстрому уменьшению величины \bar{T} при увеличении скорости считывания в

узлах $VV \in [6 * 10^4; 1 * 10^5]$ (КБ/сек) (рис. 4), как в моделях с учетом влияния блокировок, так и без, по сравнению с архитектурой «файл-сервер» (рис. 3), которая демонстрирует на графиках практически статическую зависимость \bar{T} от скорости считывания VV в том же диапазоне её изменения.

На рисунках 5-6 представлены графики зависимости величины \bar{T} от скорости передачи данных по каналу θ при различных размерностях задач \bar{n}_i .

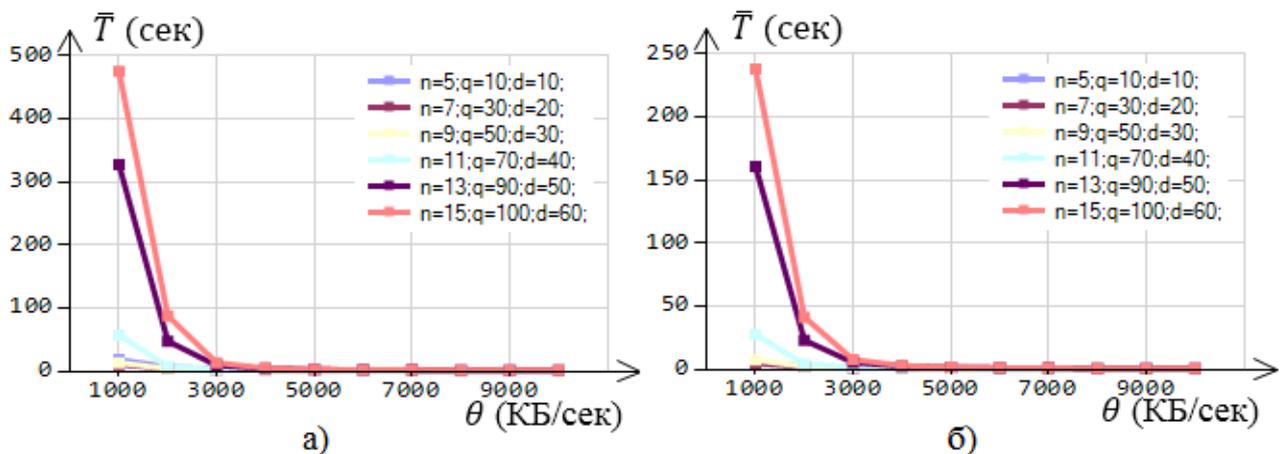


Рис.5. Зависимость \bar{T} от скорости передачи данных по каналу θ для архитектуры «файл-сервер»: а – с учетом блокировок; б – без учета блокировок

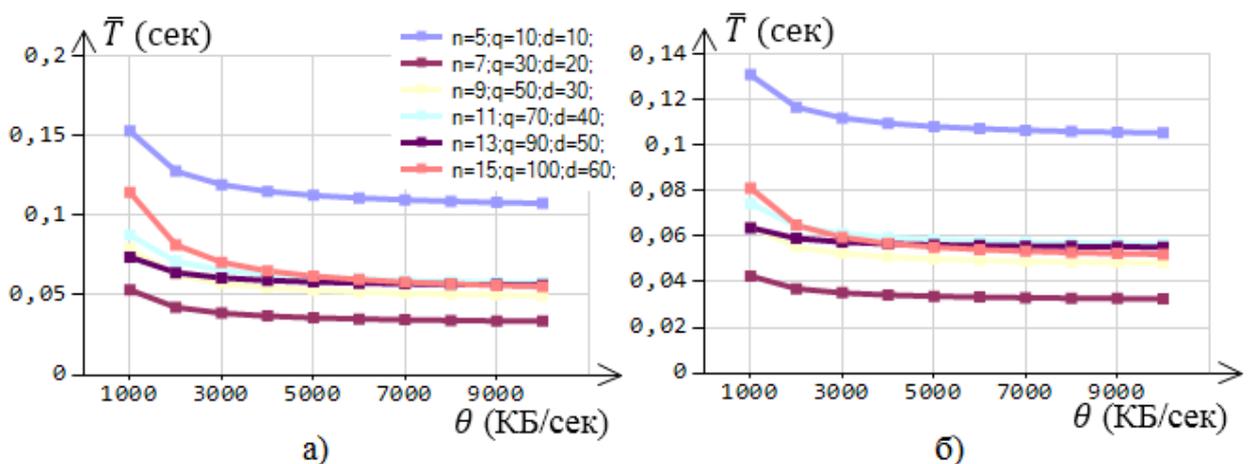


Рис.6. Зависимость \bar{T} от скорости передачи данных по каналу θ для двухуровневой архитектуры «клиент-сервер»: а – с учетом блокировок; б – без учета блокировок

Анализ графиков указанных зависимостей показал, что наиболее сильное изменение (уменьшение) величины \bar{T} для распределенных СОИ, реализованных на базе архитектуры «файл-сервер» происходит при $\theta \in [1000; 3000]$ КБ/сек, как с учетом, так и без учета влияния блокировок (рис. 5), а для распределенных СОИ, реализованных на базе двухуровневой архитектуры «клиент-сервер» - при $\theta \in [1000; 2000]$ КБ/сек. При этом, дальнейшее увеличение величины θ (исследовался диапазон изменения $\theta \in [1000; 10000]$ КБ/сек) не приводит к значительным изменениям величины \bar{T} .

На рисунках 7-8 представлены графики зависимости величины \bar{T} от производительности узлов PU распределенной СОИ для различных размерностей задач.

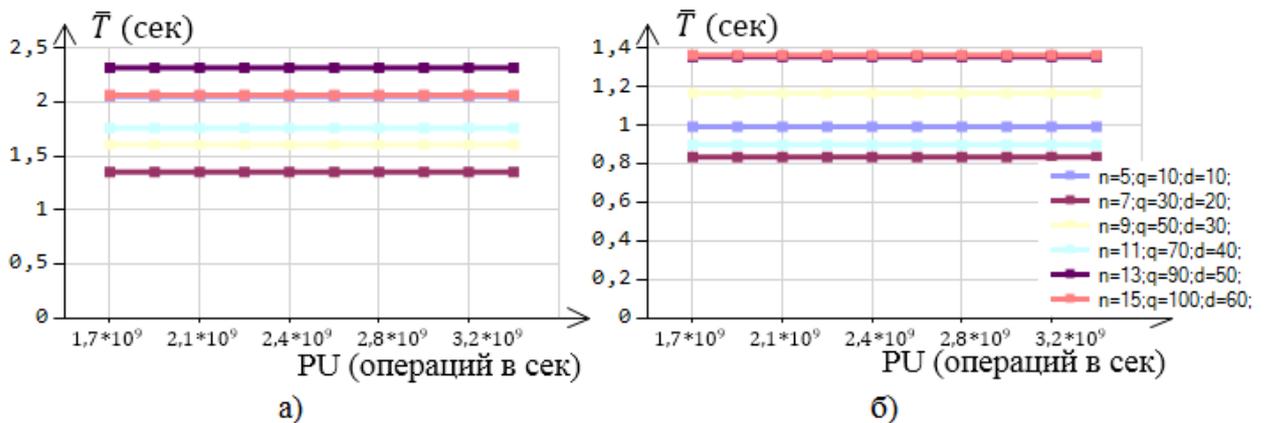


Рис.7. Зависимость \bar{T} от производительности узлов PU для архитектуры «файл-сервер»: а – с учетом блокировок; б – без учета блокировок

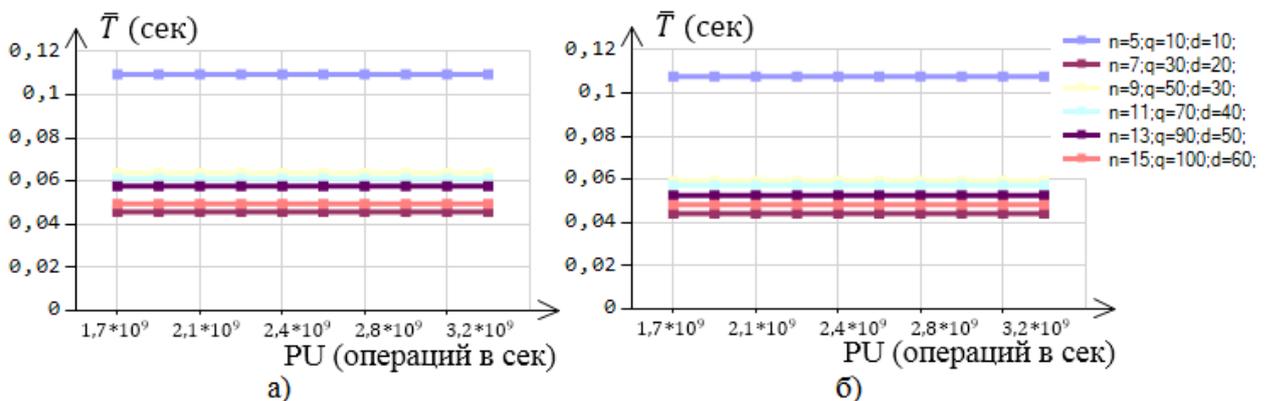


Рис.8. Зависимость \bar{T} от производительности узлов PU для двухуровневой архитектуры «клиент-сервер»: а – с учетом блокировок; б – без учета блокировок

Анализ графиков данных зависимостей показал, что изменение величины $PU \in [1,7 * 10^9; 3,5 * 10^9]$ операций в секунду практически не оказывает влияния на динамику изменения величины \bar{T} для различных размерностей задач, как для архитектуры «файл-сервер», так и для двухуровневой архитектуры «клиент-сервер». Данный факт можно объяснить достаточно «значительными» значениями величины PU , по сравнению с остальными значениями параметров распределенной СОИ: скорости передачи данных по каналу θ ; скорости считывания в узлах VV ; скорости записи в узлах VD .

Литература

1. Скоба А.Н., Состина Е.В. Математическая модель оптимального размещения распределённой базы данных по узлам ЛВС на базе файл-серверной архитектуры. // Инженерный вестник Дона. 2015. №2. URL: ivdon.ru/ru/magazine/archive/n2y2015/2881.
2. Скоба А.Н., Логанчук М.Л. Математическая модель функционирования распределённой информационной системы на базе архитектуры «файл-сервер» с учётом влияния блокировок // Инженерный вестник Дона. 2015. №3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3276.
3. 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.
4. Скоба А.Н., Айеш Ахмед Нафеа Айеш (Ирак), Михайлов В.К. Эвристический алгоритм решения задачи оптимального размещения

информационных ресурсов // Инженерный вестник Дона. 2018. №1. URL:
ivdon.ru/ru/magazine/archive/n1y2018/4744

5. Клейнрок Л. Вычислительные системы с очередями : Пер. с англ.-
М.Мир,1979.-600с.

6. Вишневский В.М. Теоретические основы проектирования
компьютерных сетей.- М.: Техносфера, 2003.- 512 с.

7. Жожикашвили В.А., Вишневский В.М. Сети массового обслуживания.
Теория и применение к сетям ЭВМ.-М. : Радио и связь, 1988.-192с.

8. Герасимов А.И. Теория и практическое применение стохастических
сетей.-М.: Радио и связь.,1994.-175с.

9. AntunesC.H. etal. A Multiple Objective Routing Algorithm for Integrated
Communication Network // Proc. ITC-16.-1999.V.3b.-PP.1291-1300.

10.Chakka R., Harrison P.G. A Markov modulated multi-server queue with
negative customers – The MM CPP/GE/c/LG-queue // ActaInformatika/-2001.-
v.37.-pp.785-799.

11.Круглый З.Л. Алгоритмы расчёта моделей структур вычислительных
систем с различными классами заданий // Управляющие системы и машины.
1989. №4. С.22-24.

References

1. SkobaA.N., SostinaE.V. Inženernyj vestnik Dona (Rus), 2015, № 2. URL:
ivdon.ru/ru/magazine/archive/n2y2015/2881.

2. Skoba A.N., Loganchuk M.L. Inženernyj vestnik Dona (Rus), 2015. №3.
URL: ivdon.ru/ru/magazine/archive/n3y2015/3276.

3. Skoba A.N., Sostina E.V. Inženernyj vestnik Dona (Rus), 2015. № 2. URL:
ivdon.ru/ru/ magazine/archive/n2y2015/2882.

4. Skoba A.N., Ayyesh Akhmed Nafea Ayyesh (Iraq), Mikhaylov V.K.
Inženernyjvestnik Dona (Rus), 2018. №1. URL:
ivdon.ru/ru/magazine/archive/n1y2018/4744.

5. Kleynrok L. Vychislitel'nye sistemy s ocheredyami [Queueing Systems]: Per. s angl.M.Mir, 1979. 600p.
6. Vishnevskiy V.M. Teoreticheskie osnovy proektirovaniya komp'yuternykh setey [Theoretical foundations of computer network design]. M.:Tekhnosfera, 2003.512p.
7. Zhozhikashvili V.A., Vishnevskiy V.M. Seti massovogo obsluzhivaniya. Teoriya i primeneniye k setyam EVM [Queueing networks. Theory and its network application].M.: Radio isvyaz', 1988.192p.
8. Gerasimov A.I. Teoriya i prakticheskoe primeneniye stokhasticheskikh setey [Theory and practical application of stochastic networks].M.: Radio isvyaz',1994.175p.
9. Antunes C. H. et al. A Multiple Objective Routine Algorithm for Integrated Communication Network. Proc ITC-16. 1999. V. 3b. pp. 1291-1300.
10. Chakka R., Harrison P. G. A Markov modulated multi-server queue with negative customers - The MM CPP/GE/c/L G-queue. Acta Informatica, 2001, V.37,pp.785-799.
11. Kruglyj Z.L. Upravljajushhie sistemy i mashiny. 1980. №4. pp.22-24.