

Интервальные вычислительные модули для расчета параметров производительности серверных компьютерных систем

Аноприенко А.Я., Иваница С.В., АльРабаба Хамза
Донецкий национальный технический университет,
anoprien@cs.dgtu.donetsk.ua

Abstract

Anopriyenko A., Ivanitsa S., AlRababa Hamzah "Interval computing modules for calculating performance parameters of server computer systems" This paper gives methods for preparation of interval results using interval arithmetic. Showing the transition from traditional view to interval from network parameters. Examine the work of computational modules configured using a scripting language JavaScript. The analysis depending of interval results on the range of input parameters.

Keywords: interval computations, server computer systems, computing modules, JavaScript

Введение

Инфраструктура поддержки Web-служб включает в себе множество аппаратных ресурсов, которые разделяются между программно реализуемыми процессами различных типов, в том числе Web-сервисами и серверами приложений, блоками управления протоколами, а также операционными системами [1]. При этом для расчета ряда основных результатов, касающихся производительности серверных систем, входные параметры усредняются (например, среднее время обслуживания одного запроса, средняя входная интенсивность запросов, средняя пропускная способность системы и др.). Такой подход позволяет говорить и о средних оценках производительности серверных компьютерных систем, что в свою очередь приводит к крайней объективности («размытости») полученных результатов.

При переходе к интервальным методам вычислений (представление входных параметров в виде интервалов, применение интервальной арифметики, получение результирующего параметра в виде интервального значения) становится возможным выразить строгий диапазон полученных результатов при использовании интервальных операций и функций, что позволяет провести более качественную оценку эффективности работы серверных компьютерных систем [2].

В работе проведен анализ расчетов базовых результатов, связанных с производительностью серверных компьютерных систем [1]. При этом для предложенных в [1] расчетов, используются только усредненные значения переменных. Рассмотренные в [3 – 6] методы интервального анализа позволили сформировать новый подход для расчета параметров, связанных с произ-

водительностью серверных компьютерных систем. Основу этого подхода составляют большинство свойств интервалов (актуальных, прежде всего, при формировании интервалов исходных данных) и реализация арифметических операций над интервальным типом данных [7].

Цель работы: разработка и реализация интервальных расчетов с использованием языка сценариев JavaScript; получение интервальных результатов, связанных с производительностью серверных компьютерных систем; выбор критериев для перехода от усредненных параметров, описывающих производительность, к вещественным интервалам; получение достоверных результатов и их анализ.

Особенности интервальной арифметики при разработке расчетных модулей

В контексте интервального анализа интервал $[a, b]$ представляет числовые промежутки в качестве основного объекта данных и не содержит никакой дополнительной информации о самой величине. Поэтому интервал $[a, b]$ трактуется как множество возможных значений неизвестной истинной величины в пределах значений $a \div b$ числовой оси.

Актуальность применения интервальной арифметики заключается, прежде всего, в ее применении при реализации компьютерных вычислений [2]. В частности, внешне приближенные значения интервальной арифметики позволяют выразить строгий диапазон полученных результатов при использовании интервальных операций и функций. Это дает качественные различия в научных расчетах по отношению к традиционным вычислениям (в которых используются данные без учета ошибок округ-

ления), так как полученный результат в виде интервала (набора значений) гарантированно содержит истинный (правильный, точный) результат внутри его крайних значений (границ интервала). В данном разделе описаны все необходимые свойства и методы работы с интервалами, применение которых является необходимым «стартовым» набором для успешного применения интервальных вычислений при расчетах параметров серверных компьютерных систем.

Так, для пары интервальных значений $\mathbf{x} = [x_1, x_2]$ и $\mathbf{y} = [y_1, y_2]$ в интервальном анализе определены четыре базовых арифметических операции «ор» (operation):

$$\mathbf{x} \text{ ор } \mathbf{y} = \{x \text{ ор } y \mid x \in \mathbf{x} \wedge y \in \mathbf{y}\} \quad (1)$$

для ор $\in \{+, -, \times, \div\}$.

Таким образом, результатом каждой из четырех основных операций «ор» является интервал, значения которого соответствуют точному диапазону каждого набора значений из областей интервалов-операндов при выполнении над ними определенной в «ор» операции [4]. Хотя запись (1) характеризует эти операции согласно математической нотации, но при этом целесообразно представить каждую операцию в отдельности, например

$$\mathbf{x} + \mathbf{y} = [x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]; \quad (2)$$

$$\mathbf{x} - \mathbf{y} = [x_1, x_2] - [y_1, y_2] = [x_1 - y_2, x_2 - y_1]; \quad (3)$$

$$\mathbf{x} \times \mathbf{y} = [x_1, x_2] \times [y_1, y_2] = [\min(x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2), \max(x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2)]; \quad (4)$$

$$\mathbf{x} \div \mathbf{y} = \mathbf{x} \times 1/\mathbf{y} = [x_1, x_2] \times [1/y_1, 1/y_2], 0 \notin \mathbf{y}. \quad (5)$$

Разработка интервальных вычислительных модулей

В данной работе рассматривается группа модулей [7], которые позволяют выполнить расчеты отдельных серверных параметров на основе базовых законов и закономерностей [1]:

- 1) закон для времени отклика (Response Time Law);
- 2) закон Литтла (Little's Law);
- 3) закон для потребности в обслуживании (Service Demand Law);
- 4) закон для коэффициента использования (Utilization Law);
- 5) закон для формирования пропускной способности (Forced Flow Law).

На рисунке 1 приведена графическая нотация рассматриваемых расчетных модулей. При этом для каждого закона справедливы отношения (6) – (10) соответственно:

$$R = M / X_0 - Z \Rightarrow [r_1, r_2] = [m_1, m_2] / [x_{01}, x_{02}] - [z_1, z_2]; \quad (6)$$

$$N = X \cdot R \Rightarrow [n_1, n_2] = [x_1, x_2] \times [r_1, r_2]; \quad (7)$$

$$D_i = U_i / X_0 \Rightarrow [d_{i1}, d_{i2}] = [u_{i1}, u_{i2}] / [x_{01}, x_{02}]; \quad (8)$$

$$U_i = S_i \cdot X_i \Rightarrow [u_{i1}, u_{i2}] = [s_{i1}, s_{i2}] \times [x_{i1}, x_{i2}]; \quad (9)$$

$$X_i = V_i \cdot X_0 \Rightarrow [x_{i1}, x_{i2}] = [v_{i1}, v_{i2}] \times [x_{01}, x_{02}], \quad (10)$$

где R – время отклика для Web-запроса; M – количество источников запросов; X_0 – количество выполняемых запросов в единицу времени (пропускная способность); Z – время обдумывания пользователем; N – количество запросов; X – выходная интенсивность; D_i – потребность в обслуживании; U_i – коэффициент использования i -го ресурса; S_i – время обслуживания Web-запроса на i -ом ресурсе за одно посещение ресурса; X_i – пропускная способность на выходе из i -ой очереди; V_i – количество посещений i -ой очереди запросом.

На начальном этапе использования интервального анализа в расчетных модулях для серверных компьютерных систем, необходима адаптация базовых законов и закономерностей. Такая адаптация заключается, прежде всего, в том, чтобы в расчетах использовать не усредненные точечные значения исходных параметров, а интервальные, учитывающие отклонения (колебания) значений этих параметров, вызванные целым набором различных факторов. Это может быть, например, нестабильность физической среды передачи данных, неравномерность распределения сетевых ресурсов, мгновенно изменяющаяся нагрузка на аппаратные и программные компоненты серверных систем и т. п.

Таким образом, возможны два пути перехода от традиционного представления входных параметров к интервальному.

1. Входной параметр a может быть определен как интервальный $\mathbf{a} = [a_1, a_2]$, предельные значения которого выражаются как

$$\mathbf{a} = [a_1, a_2] = [a - \Delta_a, a + \Delta_a], \quad (11)$$

где Δ_a – предельное отклонение от усредненного значения параметра a (значение, которое использовалось бы при традиционных расчетах).

Учитывая (8) очевидно, что $a = m(\mathbf{a})$ – радиус интервала \mathbf{a} . При этом Δ_a фактически является абсолютной погрешностью a , которая может выбираться в зависимости от факторов, влияющих на достоверность получения численного значения входного параметра (это могут быть инструментальные, методические или субъективные погрешности [8]). Очевидно, что ширина интервала $w(\mathbf{a}) = 2\Delta_a$.

2. Входной параметр a может быть определен как интервальный $a = [a_{\min}, a_{\max}]$, предельные значения которого a_{\min} и a_{\max} могут быть получены как минимальное и максимальное значение из выборки (или группы выборок) величины a . При этом из статистических соображений рекомендуется, чтобы число случаев, включенных в выборочную совокупность (объем выборки) было не менее 30–35 [9].

Использование вычислительных модулей позволяет автоматизировать расчеты для получения интервальных результатов. При этом разрабатываемые модули должны соответствовать следующим требованиям:

- 1) вся арифметика в модулях – интервальная, т.е. все арифметические операции в модулях производятся над интервальными типами данных;
- 2) расчетные модули реализуются с помощью языка сценариев JavaScript;
- 3) в модулях осуществляется контроль корректности введенных значений:
 - контроль соответствия введенных интервальных значений для параметра по отношению к его единице измерения: например, при введении интервала

для временного параметра $t = [-2, 2]$ сек., вычисления будут производиться с модифицированным значением $t = [0, 2]$ сек. (отбрасываются отрицательные значения времени);

- выбор формата числа в зависимости от смыслового значения параметра: например, при введении значения 50,2 для количества узлов K , вычисления будут производиться с целочисленным значением $K = 50$ (количество всегда задается натуральным числом);
- обмен значений границ входного параметра-интервала при ошибочном вводе, когда левая (нижняя) граница оказалась большей, чем правая (верхняя): например, при ошибочном введении интервала для временного параметра $t = [3.6, 2.8]$ сек., вычисления будут производиться с исправленным значением $t = [2.8, 3.6]$ сек. (правильное формирование границ интервала);
- 4) результат вычислений отображается двумя значениями – границами вещественного интервала.

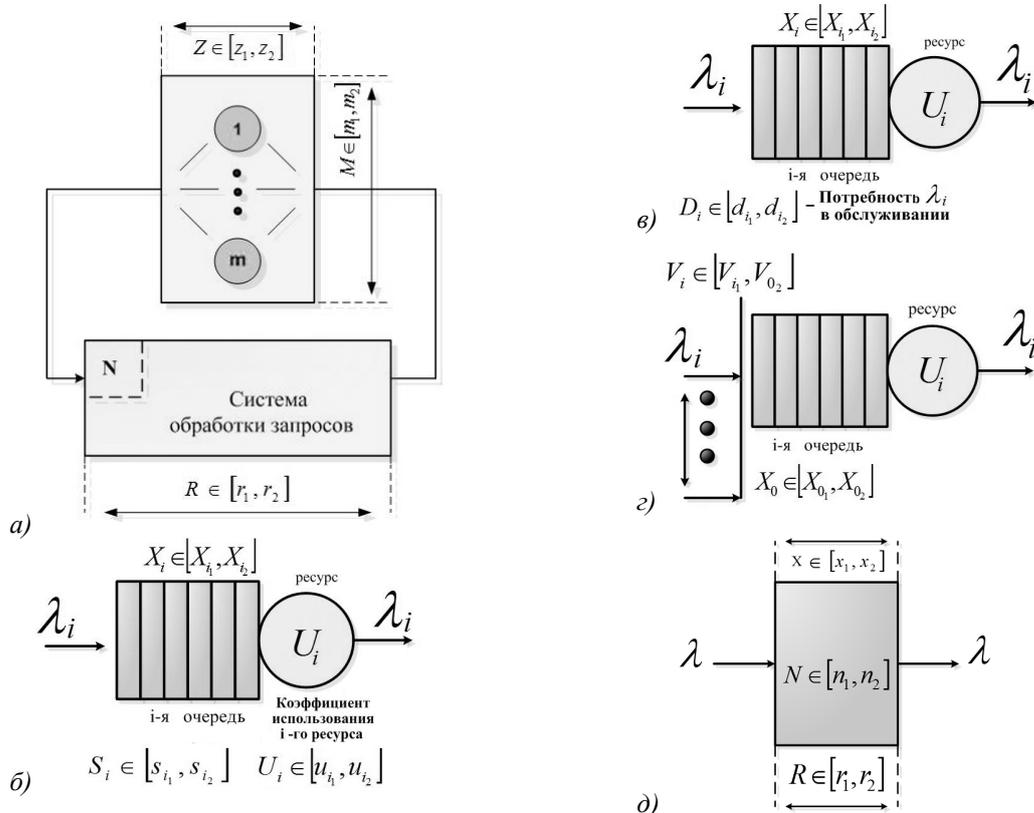


Рисунок 1. – Графическая нотация расчетных модулей для базовых законов (λ_i – входная интенсивность запросов к i -ой очереди): а) закон для времени отклика; б) закон для коэффициента использования; в) закон для потребности в обслуживании; г) закон для формирования пропускной способности; д) закон Литтла

На рисунку 2 приведена схема расчетного модуля для расчета времени отклика (по соответствующему закону), выполненного на языке сценариев JavaScript. На рисунке показано соответствие программного кода и проверок на корректность ввода и вывода данных.

На рисунке 3 показан интерфейс вычислительных модулей, состоящий из набора веб-страниц, на которых пользователь может производить необходимые вычисления. Каждая страница позволяет выполнять расчет параметров серверных компьютерных систем согласно выбранному базовому закону производительности, и включает в себя графическую нотацию закона (рис. 1) и таблицу интервальных входных параметров и результата.

Получение зависимостей интервальных результатов от исходных параметров серверных компьютерных систем

Программная реализация интервальных расчетных модулей позволяет проводить ряд исследований, относящихся к получению зависимостей диапазонов (интервалов) полученных результатов от диапазонов значений исходных данных. Так, на рисунке 4 показаны графики, демонстрирующие зависимость значений времени отклика (с точки зрения пользователя, времени на получение ответа на свой запрос к Web-службе) от различных вариаций входных данных.

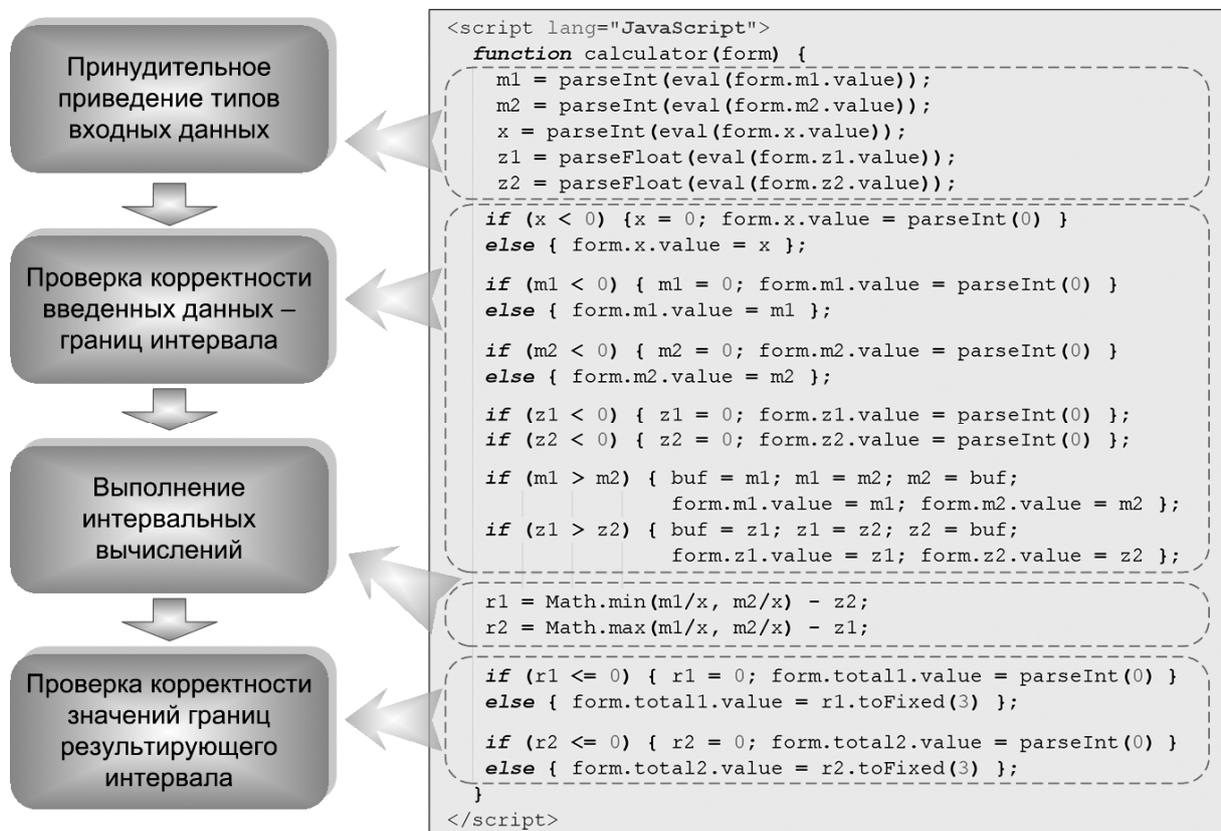


Рисунок 2. – Выполненный с помощью языка JavaScript расчетный модуль для определения времени отклика по закону «Response Time Law»

При этом очевидны прямая пропорциональность искомой величины по отношению к числу клиентов (пользователей) и обратная пропорциональность – по отношению к пропускной способности системы.

На рисунке 5 приведены графики, показывающие рабочие диапазоны количества обслуженных запросов от заданных диапазонов значений, определяющих время обдумывания пользователя и пропускную способность устройства или системы.

Интервальные расчетные модули, реализованные в совокупности средств JavaScript + HTML, могут быть размещены на сайте и представлять собой ресурс для онлайн-вычислений параметров производительности серверных компьютерных систем. С помощью данного ресурса пользователям предоставляется возможность получать интервальные значения отдельных серверных параметров на основе базовых законов.

Выводы

Таким образом, в работе были рассмотрены основные концепции использования средств и методов интервального анализа при реализации комплексных расчетов параметров серверных компьютерных систем. При расчете параметров сетевой инфраструктуры использование средств интервального анализа позволяет оперировать точными значениями входных параметров. Полученный при этом диапазон результирующих значений позволяет провести

более качественную оценку эффективности работы серверных компьютерных систем.

В работе также проведен анализ результатов, связанных с производительностью серверных систем при работе с расчетными модулями, использующими интервальные арифметические операции. Результаты показали эффективность использования такого подхода, который может быть использован в дальнейшем в качестве основы для вычислительного комплекса, направленного на анализ и оценку производительности Web-служб.

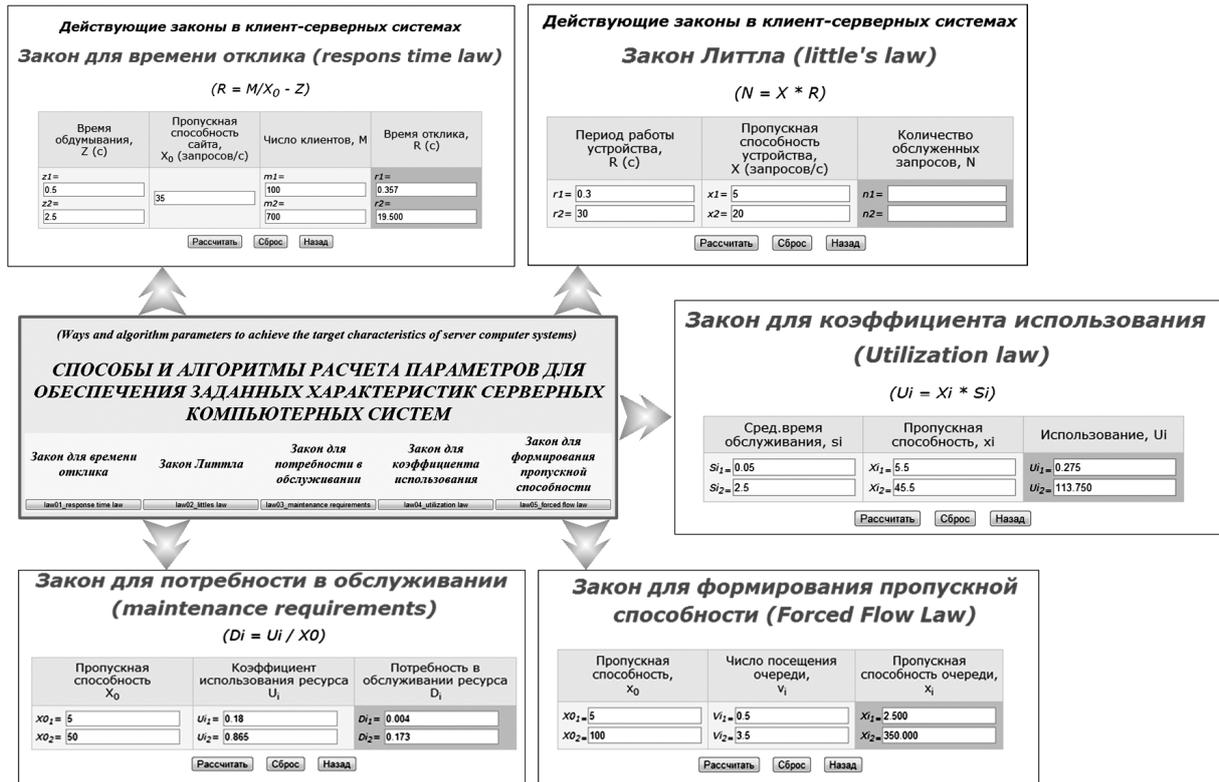


Рисунок 3. – Интерфейс вычислительных интервальных модулей

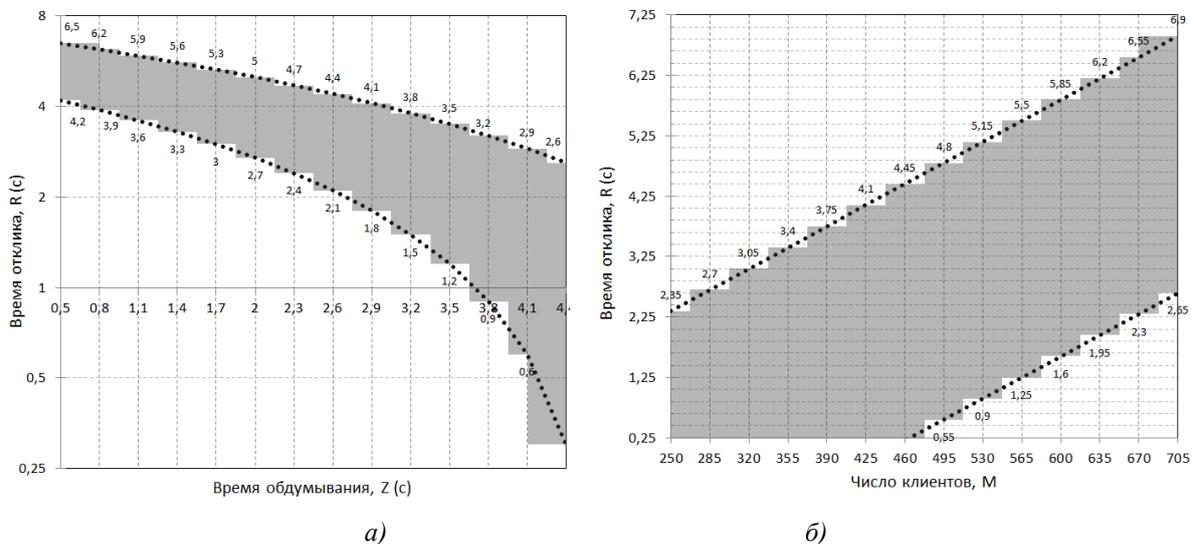


Рисунок 4. – Диапазоны полученных результатов при расчете времени отклика: зависимости от значений пропускной способности (а) и числа клиентов (б)

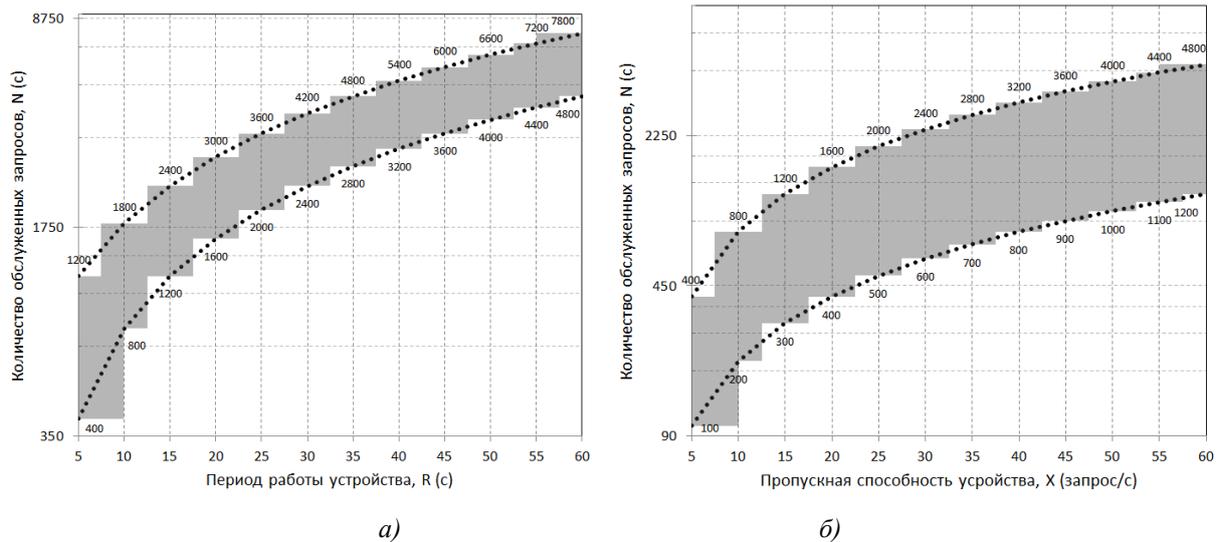


Рисунок 5. – Диапазоны полученных результатов при расчете количества обслуженных запросов по закону Литтла: зависимости от периода работы устройства (а) и диапазона значений, определяющих пропускную способность устройства (б)

Литература

1. Менаске Д. Производительность Web-служб. Анализ, оценка и планирование: Пер. с англ. / Дэниел А. Менаске, Виргилио А.Ф. Алмейда. – СПб: ООО «ДиаСофт-ЮП», 2003. – 480 с.
2. Аноприенко А.Я. Интервальные вычисления и перспективы их развития в контексте кодологической эволюции / А.Я. Аноприенко, С.В. Иваница // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2010). Выпуск 8 (168): Донецк: ДонНТУ, 2010. – С. 150 – 160.
3. Калмыков С.А. Методы интервального анализа / С.А. Калмыков, Ю.И. Шокин, З.Х. Юлдашев. – Новосибирск: Наука, 1986. – 224 с.
4. Kearfott R.B. Interval computations: introduction, uses and resources. Department of Mathematics University of Southwestern Louisiana – USL Box 4 – 1010, Lafayette, LA 7054 – 1010. – USA. Электронный ресурс. – Режим доступа: <http://www.nsc.ru/interval/Introduction/BakerSurvey.pdf>.
5. Добронез Б.С. Интервальная математика: Уч. пос. Краснояр. гос. ун-т. – Красноярск. 2004. – 216 с.
6. Алефельд Г. Введение и интервальные вычисления. / Г. Алефельд, Ю. Херцбергер. – М.: Мир, 1987. – 360 с.
7. Аноприенко А.Я. Интервальный анализ и его применение при расчетах параметров серверных компьютерных систем / А.Я. Аноприенко, С.В. Иваница, Хамза Аль Рабаба // Научный журнал «Радиоэлектроника, информатика, управління». – Запорожье, 2012. – С. 70 – 74.
8. Назаров Н.Г. Метрология. Основные понятия и математические модели. – М.: Всп. шк., 2002. – 348 с.
9. Якушев А.И., Воронов Л.Н., Федотов Н.М. Взаимозаменяемость, стандартизация и технические измерения: учебник для вузов – 6-е изд., перераб. и доп. – М.: Машиностроение, 1987. – 352 с.