

ИСПОЛЬЗОВАНИЕ ЧИСЛЕННЫХ МЕТОДОВ ОПТИМИЗАЦИИ ДЛЯ ОЦЕНКИ РИСКА ПОТЕРИ ЗАЯВОК В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

**В.Я. Пророк, доктор технических наук, доцент.
Санкт-Петербургский университет ГПС МЧС России.
А.А. Корытко.
Военно-космическая академия им. А.Ф. Можайского**

Рассматривается задача выбора оптимального значения производительности базового вычислительного элемента распределенной вычислительной системы автоматизированной информационно-управляющей системы Единой государственной системы предупреждения и ликвидации чрезвычайных ситуаций в условиях среды целенаправленного возмущения. Поиск оптимального значения осуществляется на основе использования метода штрафных функций, для нахождения значений которых используется алгоритм нечеткой классификации штрафов.

Ключевые слова: метод штрафных функций, распределенная вычислительная система, алгоритм нечеткой классификации штрафов

USAGE OF THE NUMERICAL METHOD OPTIMISATION FOR RATING OF RISKS OF LOSSES OF REQUESTS IN DISTRIBUTED COMPUTATION SYSTEMS

V.Y. Prorok. Saint-Petersburg university of State fire service of EMERCOM of Russia.
A.A. Korytko. Military space academy named after A.F. Mozhayskogo

Consideres the problem of choosing the optimal value of the performance of basic computing element of the distributed system of the automated information control system of uniform state system of prevention and liquidation of emergency situations in environment deliberate disturbance. The search of the optimal value is performed by using the method of penalty functions for finding the values which uses an algorithm fuzzy classification fines.

Keywords: method of penalty functions, a distributed computation system, algorithm fuzzy classification penalties

Современные действия МЧС России характеризуются решительностью целей, высокой динамичностью и маневренностью сил, широким применением новых технических средств. В этих условиях резко увеличивается количество информации, которую должны перерабатывать органы управления для обеспечения обоснованности принимаемых решений. Целями развития автоматизированной информационно-управляющей системы (АИУС) являются повышение оперативности, надежности, обоснованности и качества принятия управленческих решений по предупреждению и ликвидации чрезвычайных ситуаций в России на основе развития информатизации и интеграции информационных ресурсов, широкой и всесторонней автоматизации процессов управления силами и средствами Единой государственной системы предупреждения и ликвидации чрезвычайных ситуаций (РСЧС).

Согласно закону Мура производительность процессоров удваивается каждые 18 месяцев. Рост уровня производительности современных вычислительных систем (ВС) осуществляется за счет совершенствования элементной базы, архитектуры построения

процессоров, интегрирования в кристалл процессора все большего количества вычислительных ядер и устройств. Увеличение числа вычислительных устройств, функционирующих в едином информационном пространстве, сопровождается увеличением потерь их суммарной производительности на организацию асинхронного выполнения вычислительных процессов, повышенным тепловыделением, а также быстрорастущей сложностью построения и дальнейшего программного и технического сопровождения системы. С целью минимизации перечисленных эффектов, возникающих при создании распределенных многопроцессорных ВС, в зависимости от спектра решаемых задач и требований, предъявляемых к оперативности их решения, осуществляется выбор типа многопроцессорной ВС (SMP, NUMA и др.), а также оптимального значения производительности одной вычислительной единицы системы для заданной архитектуры процессора и топологии вычислительной сети.

В данной статье предлагается методика расчета оптимального значения производительности одного вычислительного ядра многопроцессорной ВС АИУС, обслуживающей множественный поток заявок.

В терминах теории систем массового обслуживания основные показатели эффективности функционирования ВС делятся на две группы - счетные и временные [1]. К счетным показателям относятся функции распределения интенсивности поступления заявок на вход ВС, вероятность отказа в приеме на обслуживание заявки, вероятность нулевого ожидания заявкой обслуживания в ВС, среднее число заявок в системе, среднее число свободных каналов в системе, среднее число занятых каналов. К наиболее существенным временным показателям относятся функция распределения времени пребывания заявки в системе, а также функция распределения времени ожидания заявок в ВС. Перечисленные показатели определяются интенсивностью потока заявок рабочей нагрузки, интенсивностью обслуживания заявок одним обслуживающим каналом, а также их количеством.

Современные специализированные распределенные ВС создаются на базе многоядерных процессоров специального и универсального назначения, таким образом, целесообразно рассматривать в качестве базового вычислительного элемента вычислительное ядро (ВЯ) процессора. Значение производительности одного ВЯ будем считать оптимальным, если его использование соответствует минимальному значению функционала вида:

$$C(\mu | \lambda, \beta_0, \beta_i, i \in \overline{1, M}) = \beta_0 \cdot K_{\text{пр}}(\mu) + \sum_{i=1}^M \beta_i(l_1, l_2, \dots, l_m) \cdot P_i(\omega_i > \omega_i^* | \mu), \quad (1)$$

где ω_i, ω_i^* – время нахождения заявки i -го типа и ее максимально допустимое время обслуживания соответственно в вычислительном комплексе, в котором интенсивность обслуживания заявок одним ВЯ равна μ ; β_0, β_i – штрафы за простой вычислительного комплекса и за превышение допустимой длительности обслуживания заявки i -го типа в условиях L ; $K_{\text{пр}}$ – коэффициент простоя вычислительного комплекса с интенсивностью обслуживания заявок одним ВЯ - μ ; $P(\omega_i > \omega_i^* | \mu)$ – вероятность превышения допустимого времени ожидания обслуживания заявкой i -го типа в вычислительном комплексе с интенсивностью обслуживания заявок одним ВЯ - μ ; M – количество типов потоков, циркулирующих в вычислительной системе; $l_i \in L$, здесь L – множество текущих задач, для выполнения которых необходима своевременная обработка заявки i -го типа.

Расчет оптимального значения производительности одного вычислительного ядра на основе алгоритма нечеткой классификации штрафов

Для определения требуемого значения производительности одного ВЯ распределенной ВС, обеспечивающего минимизацию функционала (1), воспользуемся методом штрафных функций, основанным на применении алгоритма нечеткой классификации штрафов. Вероятности $P(\omega_i > \omega_i^* | \mu)$ выражения (1) вычисляются либо аналитически (методами теории систем массового обслуживания), либо с помощью процедуры имитационного моделирования. Согласно работе [2] в случаях, когда получить распределение времени ожидания заявок в явном виде аналитическими методами невозможно, его можно с удовлетворительной точностью аппроксимировать выражением вида:

$$P(\omega \leq \tau) = 1 - \rho \cdot \exp(-\rho \tau / \omega),$$

где ρ - коэффициент загрузки ВС; ω - среднее время ожидания заявки.

Штраф, назначаемый за превышение допустимого времени нахождения заявки в ВС, является динамическим, то есть принимает различные значения в зависимости от сложившейся ситуации – текущего приоритета решаемых задач и потребности для их решения в результатах обработки заявки. Формализовать функции штрафов на основе статистической информации, как правило, не удастся, так как статистическая выборка ограничена значениями, которые были достигнуты в процессе функционирования ВС, и не содержит информации о качестве функционирования ВС во всевозможных условиях. Для нахождения значений штрафов используются методы нечеткого кластерного анализа. Применительно к рассматриваемой задаче используется алгоритм нечеткой классификации штрафов, суть которого заключается в разбиении объектов обучающей выборки (ОВ) по классам путем кластеризации на основе алгоритма *c-means*, использующего нечеткую логику.

В качестве исходных данных алгоритм нечеткой классификации штрафов использует:
– ОВ с известными параметрами первоначальной классификации штрафов:

$$\{Z_{lnj} : l = \overline{1, M}; n = \overline{1, v}\},$$

где M – число объектов выборки для заявок j -го типа; v - размерность вектора показателя $X_j = \{x_1, x_2, \dots, x_v\}$, характеризующего степень выраженности совокупности задач, решаемых ВС;

– функции принадлежности штрафов:

$$\{\mu_{li} : l = \overline{1, M}; i = \overline{1, K}\},$$

где K – количество центров классов установленных штрафов за превышение допустимого времени нахождения заявки в системе;

– универсальные множества $E_j = \{e_{j1}, e_{j2}, e_{j3}\}$, элементы которых характеризуют возможные решения классификатора о величине штрафа за задержку обработки заявки типа j , элементы множества e_{j1}, e_{j2}, e_{j3} соответствуют низкому, среднему и высокому штрафу соответственно.

Значения указанных параметров определяются на основе мнения экспертов, исходя из их опыта и интуиции. Возможные значения термина, характеризующего величины векторов

X_j , принадлежат интервалу $\overline{1,10}$. Дополнительно к векторам X_j вводится вектор весовых коэффициентов $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_v\}$, где значение λ_i – нормированный коэффициент важности задачи i -го типа, рассчитываемый методом последовательных вариаций. Благодаря введению данного вектора весовых коэффициентов появляется возможность увеличения степени влияния на результаты работы алгоритма нечеткой кластеризации штрафов задач, потребность выполнения которых возникает сравнительно редко, но вместе с тем требующих наибольшей оперативности. Далее для каждого объекта ОВ осуществляется свертка векторов X_j в скалярную форму с помощью выражения: $x_j = X_j \cdot \Lambda^T$. На основе сформированной нечеткой базы знаний нечеткий классификатор штрафов реализует отображение вида $x_j \rightarrow y \in E_j$ суть которого заключается в следующем.

Мера принадлежности [3] штрафа к одному из элементов множества E_j для заданного уровня четкости – α согласно работе [4] вычисляется на основе итерационной процедуры, включающей два основных этапа.

Первый этап заключается в расчете центров классов классификации объектов множества ОВ $V_{i\alpha}$:

$$V_{i\alpha}^s = \frac{\sum_{x \in X_j(\alpha)} (\mu_{i\alpha}^s(x))^2 \cdot x}{\sum_{x \in X_j(\alpha)} (\mu_{i\alpha}^s(x))^2}, \text{ если } \sum_{x \in X_j(\alpha)} (\mu_{i\alpha}^s(x))^2 \neq 0,$$

где s – номер итерации алгоритма.

Второй этап заключается в расчете значений функции принадлежности потенциального штрафа в условиях обстановки x_i для заданного уровня нечеткости:

$$\mu_{i\alpha}^s(x) = \frac{1/\|x - V_{i\alpha}\|^2}{\sum_{j=1}^K (1/\|x - V_{j\alpha}\|^2)}, \text{ если } \|x - V_{i\alpha}\|^2 > 0;$$

$$\mu_{ij}^s(x) = \begin{cases} 1, & j = \alpha \\ 0, & j \neq \alpha \end{cases}, \text{ где } j = \overline{1, K}, \text{ если } \|x - V_{i\alpha}\|^2 = 0.$$

Оба этапа выполняются итерационно до тех пор, пока разница полученных значений центров классов для заданной ОВ не будет удовлетворять условию:

$$|V_{i\alpha}^s - V_{i\alpha}^{s-1}|^2 \leq \varepsilon,$$

где ε – минимально допустимое различие значений центров классов, полученных на двух последовательных итерациях.

Результатами работы описанного алгоритма являются вектор, содержащий центры кластеров для каждого класса, а также соответствующие им значения функции принадлежности объектов ОВ к классам классификатора. Поскольку значения принадлежности штрафа к одному из классов характеризуются нормированной величиной, четкое значение штрафа в ситуации, характеризуемой j объектом ОВ, получим, воспользовавшись выражением:

$$\beta_j = \sum_{i=1}^K e_i \cdot \mu_i(x), \forall x \in (V_L; V_R), \quad (2)$$

где e_i – элемент множества E_j .

Выражение (2) осуществляет приведение к четкости величины $\mu_i(x)$, в соответствии с дискретным вариантом центроидного метода подобный прием является классическим и используется во многих алгоритмах нечеткого логического вывода [5].

Аргументом функции, характеризующей значения величины штрафа за простой вычислительных устройств, является число вычислительных ядер, используемых в ВС. При этом данная функция должна достаточно быстро возрастать при приближении значения ее аргумента к величине, задаваемой числом вычислительных ядер на одном типовом процессоре, и продолжать увеличиваться с дальнейшим ростом значений аргумента. Указанному свойству удовлетворяет логарифмическая зависимость, следовательно, штраф за простой ВС можно представить математическим выражением следующего вида:

$$\beta_0(n) = \log_k(n),$$

где k - число ядер в базовом процессоре; n - общее число ядер в вычислительной системе.

Классическим способом определения оптимального значения производительности в постановке данной задачи (1) является решение дифференциального уравнения:

$$\beta_0 \cdot \frac{K_{np}(\mu_1)}{\Delta\mu_1} + \sum_{i=1}^M \beta_i(l_1, l_2, \dots, l_m) \cdot \frac{P(\omega_i > \omega_i^* | \mu_i)}{\Delta\mu_1} = 0. \quad (3)$$

Но в случае многопроцессорных ВС вид функций $P(\omega_i > \omega_i^* | \mu_i)$, $K_{np}(\mu_1)$ не позволяет решить дифференциальное уравнение (3) аналитическими методами. В связи с чем для нахождения производительности вычислительного ядра, обеспечивающей минимум функционала (1), применяются численные методы оптимизации нулевого порядка. При фиксированных значениях штрафов, входящих в функционал (1), рассматриваемая задача сводится к задаче одномерной минимизации.

Существует два основных класса функций - унимодальные функции, то есть функции, имеющие единственный экстремум, а также многоэкстремальные функции. Так как универсальных методов, пригодных для минимизации произвольных функций одной переменной не существует, необходимо определить, к какому из перечисленных классов принадлежит функционал (1). Для решения данной задачи используются методы выпуклого анализа [6].

В случае показательного распределения времени обслуживания заявок ВС слагаемые функционала (1), представляющие вклад заявок i -го потока в его итоговое значение, являются строго выпуклыми функциями, компонент, характеризующий простой ВС, является выпуклой функцией. В соответствии с теоремой 3.2 работы [6], гласящей, что если f_1, \dots, f_m – выпуклые функции на выпуклом множестве X , $\alpha_1, \dots, \alpha_m$ – неотрицательные числа, тогда функция $f(x) = \sum_{i=1}^m \alpha_i f_i(x)$ выпукла на X , функционал (1) - выпуклый. Вид функционала (1) и его компонент для 32-х ядерной ВС с дисциплиной обслуживания FIFO, созданной на базе двухъядерных процессоров, обслуживающей заявки двух типов, представлен на рис. 1. Штраф за простой ВС - β_0 равен 5, за превышение допустимого времени обслуживания заявки первого типа - β_1 равен 6,95, заявки второго типа - β_2 равен 1,97. Заявки первого типа поступают на обслуживание группами по 400 штук в условиях отсутствия в очереди необработанных заявок второго типа, при этом их суммарное количество ограничено 4 500 единицами. Интенсивность поступления заявок второго типа принимается равной 0,97 заявок в секунду. Среднее время обслуживания заявки первого

типа в 4 раза продолжительнее, чем для заявки второго типа. Предельно допустимое время нахождения в системе заявки первого типа полагается равным 12 000 с, заявки второго типа 2 000 с. Поступление заявки второго типа в процессе обработки заявки первого типа не прерывает выполнения последней до тех пор, пока не будет обработана вся группа.

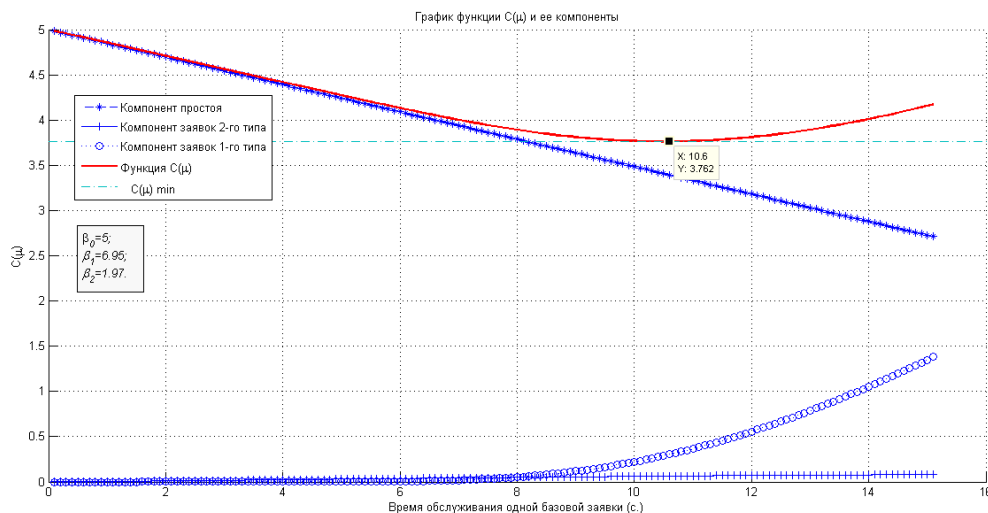


Рис. 1. Значения оптимальной интенсивности обслуживания заявок одним ядром для 32-х ядерной вычислительной системы

Расчет субоптимального значения производительности вычислительного ядра осуществляется путем полного перебора всевозможных комбинаций штрафов за простой, полученных квантованием интервала $(V_L; V_R)_i$ по Δx_i и последующего нахождения значения интенсивности обслуживания заявки, при котором функционал (1) достигает минимального значения. Поиск соответствующего значения аргумента осуществляется с помощью какого-либо алгоритма минимизации значений функции, например метода Фибоначчи. Оценка требуемого значения производительности вычислительного ядра, близкого к оптимальному значению, осуществляется методами математической статистики и теории вероятностей на основе частот проявления уникальных значений производительности.

Описание процедуры расчета оптимального значения производительности одного вычислительного ядра

В качестве примера произведем расчет требуемого значения производительности ВС, используемой для построения графика функционала (1), изображенного на рис. 1. Считается, что критическое множество задач, решаемых рассматриваемой ВС, состоит из пяти элементов – задач x_1, x_2, x_3, x_4, x_5 соответственно. Обучающие выборки первоначальной классификации штрафов представлены в табл. 1 для потока заявок первого типа и табл. 2 для потока заявок второго типа.

Таблица 1. Вектор X_1 и соответствующая ему классификация штрафов

№	y_1	y_2	y_3	y_4	y_5	μ_1	μ_2	μ_3
1	2	4	5	1	1	0,7	0,3	0
2	2	4	5	1	2	0,65	0,35	0
3	3	6	7	1	3	0,5	0,4	0,1
4	4	8	8	2	4	0,2	0,6	0,2

№	y_1	y_2	y_3	y_4	y_5	μ_1	μ_2	μ_3
5	5	10	9	2	5	0	0,7	0,3
6	6	10	10	1	6	0	0,55	0,45
7	7	10	10	3	8	0	0,4	0,6
8	8	10	10	3	9	0	0,2	0,8
9	10	10	10	4	8	0	0,15	0,85
10	10	10	10	4	9	0	0,1	0,9

Таблица 2. Вектор X_2 и соответствующая ему классификация штрафов

№	x_1	x_2	x_3	x_4	x_5	μ_1	μ_2	μ_3
1	2	1	2	2	1	0,9	0,1	0
2	3	2	3	2	2	0,8	0,2	0
3	3	2	4	2	1	0,7	0,3	0
4	4	3	5	3	3	0,4	0,5	0,1
5	5	5	7	4	5	0,2	0,7	0,1
6	6	5	8	5	6	0	0,65	0,35
7	7	6	8	6	8	0	0,5	0,5
8	8	6	9	7	9	0	0,3	0,7
9	9	8	10	8	8	0	0,15	0,85
10	10	10	10	9	9	0	0,05	0,95

Универсальное множество E_1 состоит из элементов $e_{11} = 3,84$, $e_{12} = 7,68$, $e_{13} = 11,52$, множество E_2 - из элементов $e_{21} = 1$, $e_{22} = 2$, $e_{23} = 3$. Начальное значение векторов весовых коэффициентов Λ_1 и Λ_2 одинаково и представляется в виде $\Lambda = \{0, 2; 0, 2; 0, 2; 0, 2; 0, 2; 0, 2\}$, в результате применения метода последовательных вариаций данные вектора преобразуются к виду $\Lambda_1 = \{0, 23; 0, 29; 0, 13; 0, 25; 0, 1\}$, $\Lambda_2 = \{0, 26; 0, 25; 0, 1; 0, 22; 0, 17\}$. График функций принадлежности объектов ОБ, а также функции штрафа, полученные в результате применения описанного выше алгоритма для исходных данных, используемых при построении графика функции, изображенного на рис. 1, приведены на рис. 2. Значения соответствующих центров классов представлены в табл. 3.

Таблица 3. Значения центров классов элементов множеств X_1 и X_2

	K_1	K_2	K_3
X_1	2,98	5,99	7,97
X_2	2,31	5,67	8,56

Интервал возможного значения штрафа за превышение допустимого времени обработки заявки i -го типа в зависимости от условий функционирования ВС рассчитывается на множестве возможных значений аргумента x_i , принадлежащего интервалу $(K_1, K_3)_i$. Для полученных таким образом значений штрафов рассчитывается интервал локализации минимума функционала (1) с помощью метода Фибоначчи. При этом число итераций метода Фибоначчи полагается равным 12, что при интервале возможных значений длительности обработки заявки базового типа от 0,1 с до 8 с допускает максимальную длину локализации

минимума (1), равную 0,017 с. Здесь предполагается, что для заданной архитектуры базового процессора ВС отношение времени обслуживания заявки первого типа к заявке второго типа является константой. В проводимом вычислительном эксперименте отношение времени выполнения заявки первого типа к заявке второго типа считается равным 4.

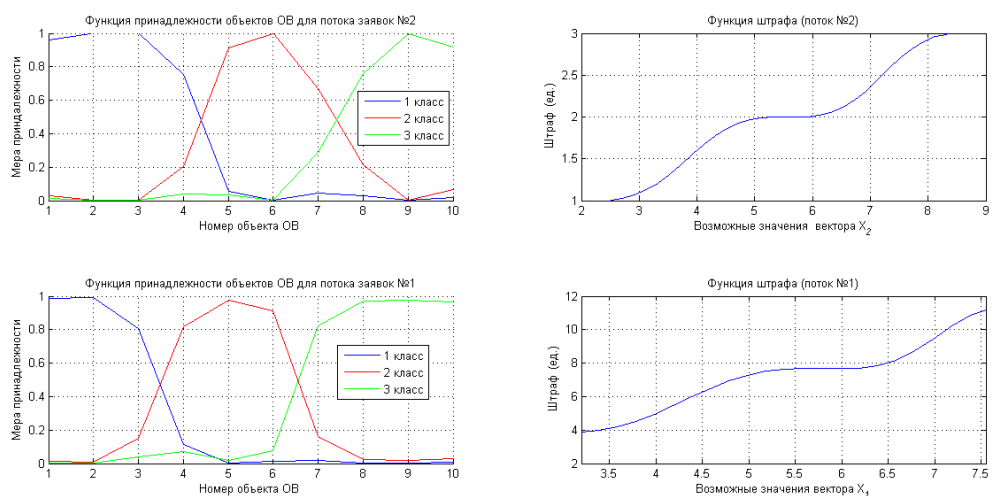


Рис. 2. Результаты работы алгоритма нечеткой классификации штрафов

Значения производительности одного ВЯ, полученные в результате применения описанной методики, а также частота их проявления представлены на рис. 3.

Полученные частоты требуемых значений производительности одного ВЯ позволяют получить распределение соответствующей дискретной случайной величины. Таким образом, для ВС, функционирующих в условиях целенаправленного возмущения, оказывающего влияние на эффективность процесса функционирования системы, становится возможным использование принципа гарантированного результата [7]. Данный принцип гарантирует, что в любых возможных условиях среды функционирования ВС вероятность превышения времени нахождения заявки не превысит предельно допустимого значения.

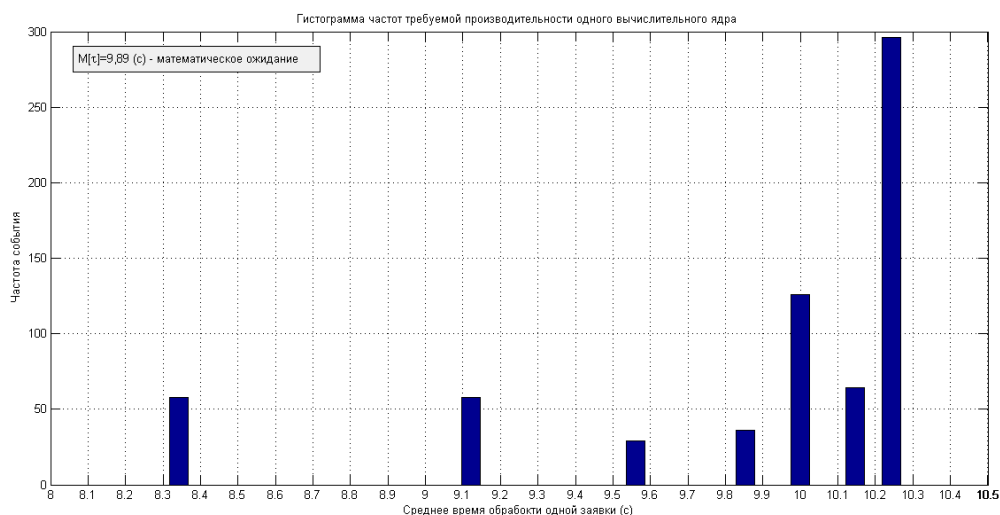


Рис. 3. Результаты применения описанной методики

Использование алгоритма нечеткой классификации штрафов позволяет рассчитывать оптимальные значения требуемой производительности вычислительных единиц распределенных ВС для режимов функционирования системы, в которых отсутствует

статистическая информация о ее поведении. Использование описанной методики на этапе составления тактико-технических заданий позволяет обоснованно формулировать требования к производительности базовых вычислительных устройств, используемых для построения ВС. Описанная методика позволяет рассчитывать так называемую относительную производительность, не учитывающую потери производительности на организацию асинхронного выполнения программ в системе. Решение данной задачи осуществляется индивидуально для конкретной ВС, а ее результаты используются в качестве соответствующих поправок для результатов, полученных с помощью применения описанной методики.

Литература

1. Рыжиков Ю.И. Теория очередей и управление запасами. СПб.: Питер, 2001. 384 с.
2. Основы теории вычислительных систем / С.А. Майоров [и др.]. М.: Высш. шк., 1978. 408 с.
3. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. 432 с.
4. Wu K.L., Yang M.S. A clustering validity index for fuzzy clustering. *Pattern Recognition Letters*. Vol. 26. Issue 9. 2005. 1 July P. 1275–1291.
5. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия «Телеком», 2002. 382 с.
6. Сухарев А.Г., Тихомиров В.В., Федоров В.В. Курс методов оптимизации. М.: Физматлит, 2008. 368 с.
7. Калинин В.Н. Теоретические основы системных исследований: краткий авторский курс лекций для адъюнктов академии. СПб.: Воен. косм. акад. им. А.Ф. Можайского, 2011. 278 с.