

Повышение ресурсной эффективности распределенных архитектур систем обработки данных на основе априорных данных о поздних сроках завершения работ

А. Б. Клименко ¹ ✉

¹ Научно-исследовательский институт многопроцессорных вычислительных систем
им. академика А.В. Каляева Южного федерального университета,
ул. Чехова, д. 2, г. Таганрог 347928, Российская Федерация

✉ e-mail: anna_klimenko@mail.ru

Резюме

Цель исследования. Целью данного исследования является разработка метода повышения эффективности распределенных архитектур систем обработки данных, функционирующих в «туманном» и «краевом» слоях сети. В условиях высокой динамики как сетевой инфраструктуры, так и нагрузки, задача формирования архитектуры систем обработки данных решается регулярно (миграция виртуальных машин, горизонтальное масштабирование). При этом практически не рассматривается вопрос расхода вычислительного ресурса вычислительных узлов, в то время как зачастую используемые устройства обладают сравнительно небольшими мощностями, а высокая их загруженность ведет к сокращению срока эксплуатации. Поэтому создание методов формирования эффективной в аспекте сохранения вычислительного ресурса архитектуры системы вычислительных устройств является актуальной задачей.

Методы. Основными научными методами, применяемыми в рамках данного исследования, являются анализ предметной области, методы исследования операций, методы оптимизации и компьютерное моделирование, подтверждающие целесообразность основных аспектов разрабатываемого метода. Для реализации повышения эффективности размещения вычислительных задач на узлах фрагмента сети в данной работе сформулирована многокритериальная задача оптимизации, где каждому элементу векторной целевой функции соответствует индивидуальное значение вероятности безотказной работы вычислительного устройства. Для получения оценочных значений стоимостной функции используются априорные оценки поздних сроков завершения решения вычислительных задач узлами, поскольку выделяемый для решения ресурс зависит от выделенного времени, а время решения задачи, соответственно, от выделяемого вычислительного ресурса. Значение стоимостной функции вычисляется на основании приближенных априорных оценок, что приводит к позитивному эффекту по части расхода вычислительных ресурсов устройств.

Результаты. Результатом исследования является разработанный метод повышения эффективности распределенных архитектур систем обработки данных, функционирующих в «туманном» и «краевом» слоях сети.

Заключение. Предложенный в данной работе метод позволяет выбрать такое размещение нагрузки, чтобы уменьшить загруженность устройств и таким образом снизить расход вычислительного ресурса.

Ключевые слова: туманные вычисления; распределенные системы; архитектура; обработка данных.

Конфликт интересов: Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Клименко А. Б. Повышение ресурсной эффективности распределенных архитектур систем обработки данных на основе априорных данных о поздних сроках завершения работ // Известия Юго-Западного государственного университета. 2023; 27(2): 124-139. <https://doi.org/10.21869/2223-1560-2023-27-2-124-139>.

Поступила в редакцию 12.05.2023'

Подписана в печать 04.07.2023

Опубликована 02.08.2023

Distributed Data Proceeding Systems Architectures Resource Efficiency Improvement on the Basis of Apriory Data about the Jobs Late Completion Times

Anna B. Klimenko ¹ ✉

¹ Scientific Research Institute of Multiprocessor Computer Systems of Southern Federal University
2 Chekhov str., Taganrog 347928, Russian Federation

✉ e-mail: anna_klimenko@mail.ru

Abstract

Purpose of research. The purpose of this study is to develop a method for improving the efficiency of distributed architectures of data processing systems operating in the fog and edge layers of the network. In conditions of high dynamics of both the network infrastructure and the load, the task of forming the architecture of data processing systems is solved regularly (migration of virtual machines, horizontal scaling etc.) At the same time, the issue of the consumption of the residual resource of computing nodes is practically not considered, while the often used devices have relatively low capacity, and their high workload leads to a reduction in the service life. Therefore, the creation of methods for forming an architecture of a computing device system that is effective in terms of saving a computing resource is an urgent task.

Methods. The main scientific methods used in this study are domain analysis, operations research methods, optimization methods and computer modeling, confirming the feasibility of the main aspects of the developed method.

To improve the efficiency of placing computational tasks on the nodes of a network fragment, this paper formulated a multicriteria optimization problem, where each element of the vector objective function corresponds to an individual value of the probability of failure-free operation of a computing device. To obtain estimated values of the cost function, a priori estimates of the late completion of the solution of computational problems by nodes are used, since the resource allocated for solving depends on the allocated time, and the time for solving the problem, respectively, on the allocated computing resource. The value of the cost function is calculated on the basis of approximate a priori estimates, which leads to a positive effect in terms of the consumption of computing resources of devices.

Results. The result of the study is a developed method for improving the efficiency of distributed architectures of data processing systems operating in the fog and edge layers of the network.

Conclusion. The method proposed in this work allows to choose such a load distribution in order to reduce the workload of devices and thus reduce the consumption of computing resources of the devices.

Keywords: fog computing; distributed systems; architecture; data processing.

Conflict of interest. The author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Klimenko A. B. Distributed Data Proceeding Systems Architectures Resource Efficiency Improvement on the Basis of Apriory Data about the Jobs Late Completion Times. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2023; 27(2): 124-139 (In Russ.). <https://doi.org/10.21869/2223-1560-2023-27-2-124-139>.

Received 12.05.2023

Accepted 04.07.2023

Published 02.08.2023

Введение

В настоящее время развитие сетевых и информационных технологий привело к ситуации, когда архитектуры систем обработки данных могут иметь вариативный, изменяющийся во времени характер как за счет динамики внешних слоев сети (туманного и краевого), так и за счет необходимости реконфигурации, которая, в свою очередь, возникает по причине необходимости обеспечения QoS в условиях изменяющейся нагрузки и может быть реализована за счет горизонтального масштабирования. Периодическая необходимость реконфигурации таких систем актуализирует такие вопросы, как:

- выбор и разработка методов реконфигурации с акцентом на минимизацию времени проведения процедуры, что связано с необходимостью обработки данных в реальном времени [1,2];

- выбор и разработку методов формирования сообществ вычислительных устройств с оптимизацией по выбранной целевой функции [3-5].

Выбор целевой функции, одной или нескольких, определяет качественные и количественные характеристики устройств, которые будут участвовать в обработке данных. В этой области существует достаточно широкий круг работ, где оптимизация осуществляется в рамках следующих критериев:

- время выполнения комплекса задач [6-8];

- равномерное распределение нагрузки [9,10];

- энергозатраты при выполнении комплекса задач [11-13];

- объемы пересылаемых по сети данных [14-16].

Однако практически не уделено внимание вопросу сохранения вычислительного ресурса устройств, в то время как их работа в условиях повышенной нагрузки приводит к сокращению срока службы и, следовательно, к повышению эксплуатационных издержек.

Целью данного исследования является разработка метода повышения эффективности архитектуры системы обработки данных в «туманном» или «краевом» слое по критерию ресурсосбережения устройств, участвующих в вычислениях и передаче данных.

Проблема формирования архитектуры системы обработки данных в контексте данного исследования отличается от классической задачи составления расписания работ для множества вычислительных устройств и скорее близка к проблеме, описанной и решенной в [17] с тем отличием, что в [17] множество вычислительных устройств объединено полносвязной коммуникационной средой без транзитных участков, тогда как такая модель коммуникационной среды не отражает особенностей распределенных вычислений в туманных и краевых средах. Кроме того, отличие составляет отсутствие явных ограничений на ресурсы, которые вычис-

лительный узел может выделить для решения той или иной вычислительной задачи: здесь ресурс является также варьируемым параметром, значение которого определяет качество формируемой архитектуры.

Появление транзитных узлов в структуре коммуникационной среды привносит необходимость рассмотрения новых аспектов задачи, таких, как – время передачи данных транзитным узлом и, собственно, дополнительной нагрузки на узел, которая при этом возникает.

Поэтому формирование архитектуры системы обработки данных в туманном или краевом слое сети некорректно рассматривать без учета нагрузки на транзитные узлы и без определения подмножеств устройств-участников вычислительного процесса.

Материалы и методы

Ценностная функция эффективности ресурсосберегающей архитектуры системы обработки данных

Формирование архитектуры системы обработки данных актуализирует вопрос о том, какая архитектура будет считаться предпочтительной. Например, возможен вариант такого размещения вычислительных задач, когда будет задействовано большое количество узлов (по задаче на узел) с необходимостью задействования узлов для передачи данных между вычислительными задачами. Тогда ожидается, что такой способ распределения нагрузки приведет к

меньшей нагрузке на каждый вычислительный узел и, следовательно, такую архитектуру можно считать предпочтительной. Другой вариант размещения – использование меньшего количества вычислительных узлов с возрастанием нагрузки на каждый из них, но с уменьшением их количества, задействованного в вычислениях и передаче данных. Здесь имеет место ситуация, когда некоторые узлы могут быть загружены в большей степени, но при этом некоторые вообще не будут принимать участия в вычислениях и/или передаче данных и, соответственно, решение комплекса задач не окажет влияния на модельные значения остаточного вычислительного ресурса и вероятности безотказной работы.

Помимо этого, необходимо ответить на вопрос, что при формировании архитектуры системы обработки данных будет являться целевой функцией. Возможными вариантами могли бы быть выражения для оценивания ВБР устройств при последовательном и параллельном соединении (см. (1) и (2))

$$P_0(\tau) = 1 - \prod (1 - P_j(\tau)), P_j(\tau) = e^{-\lambda_j t^{\frac{kD_j}{10}}}. \quad (1)$$

$$P_0(\tau) = \prod (P_j(\tau)), P_j(\tau) = e^{-\lambda_j t^{\frac{kD_j}{10}}}. \quad (2)$$

Однако учитывая временный характер объединения вычислительных узлов в систему обработки данных, вполне очевидно, что граф сети будет произвольным, и разбить его на параллельно-последовательные участки однозначно вряд ли удастся.

Поэтому предлагается следующее:

1. Пусть каждый узел, участвующий в вычислениях или передаче данных, стремится к минимизации индивидуальной нагрузки.

2. Будем полагать, что каждый узел-участник важен, и ухудшение ситуации на одном узле не может быть возмещено улучшением ситуации на другом узле.

3. Также будем полагать, что могут быть выделены приоритетные узлы, которые по желанию владельца должны сохранять как можно больший средний остаточный ресурс в процессе вычислений.

Таким образом, ЦФ функция для формирования эффективной архитектуры будет векторной, т.е. многокритериальной.

Приведем формальную постановку задачи.

Пусть имеется граф фрагмента сети $G = \langle V, U \rangle$, где V – множество вычислительных узлов, U – множество ребер, определяющих каналы связи. $V = \{v_j\} = \{\langle j, p_j \rangle\}$, где j – идентификатор узла; p_j – производительность узла.

Комплекс задач, предназначенных для решения, описывается направленным ациклическим графом $G' = \langle T, C \rangle$, вершины которого T ассоциированы с задачами, а ребра C – с информационными связями между задачами. $T = \{t_i\} = \{\langle i, w_i, d_i \rangle\}$, где i – идентификатор задачи;

w_i – вычислительная сложность подзадачи (в командах);

d_i – объем данных, который должен быть передан другой задаче.

Решением задачи формирования архитектуры системы обработки данных будет сформированная матрица назначения задач на вычислительные узлы:

$$A = \begin{vmatrix} 1 & & \dots \\ & \dots & \\ \dots & & 1 \end{vmatrix} \quad (3)$$

Учитывая индивидуальную заинтересованность каждого узла в снижении нагрузки, получим ряд ЦФ, соответствующих выбранному множеству узлов.

$$\begin{aligned} P_1(\tau) &= e^{-\lambda_0 \tau \cdot 2^{kD_1/10}}; \\ P_2(\tau) &= e^{-\lambda_0 \tau \cdot 2^{kD_2/10}}; \\ &\dots \\ P_m(\tau) &= e^{-\lambda_0 \tau \cdot 2^{kD_m/10}}, \end{aligned} \quad (4)$$

где m – число узлов, принимающих участие в обработке или передаче данных;

τ – момент времени, для которого рассчитываются (4).

Соответственно, тогда решения будут представлены фронтом Парето, состоящим из множества матриц (3).

Сформированная таким образом задача многокритериальной оптимизации, в соответствии с утверждением о том, что ситуация компенсации ухудшения на одном узле улучшением на другом нецелесообразна, сводится к однокритериальной задаче путем мультипликативной свертки (5), что также целесообразно, учитывая значения множителей.

$$P_0(\tau) = \prod_{i=1}^m (P_i(\tau))^{\xi_i}, \quad (5)$$

где ξ_i – коэффициент, выражающий предпочтения пользователя по ряду вычислительных узлов.

Отметим, что сведение к однокритериальной задаче посредством мультипликативной свертки отражает принятое утверждение о том, что каждый узел заботится об индивидуальной загруженности, и не рассматривается ситуация, когда ухудшение показателя одного узла может быть компенсировано улучшением показателей других.

Метод формирования ресурсосберегающей архитектуры на основе априорных оценок поздних сроков завершения задач

Из постановки задачи и сформированной ЦФ видно, что количество множителей (индивидуальных ЦФ) зависит от сформированной системы вычислительных узлов, а значение индивидуальной ЦФ зависит от загруженности узлов. Такая постановка отличает сформированную задачу от аналогичных, сводящихся к перебору возможных распределений задач по заранее заданному множеству вычислительных устройств. Управляемыми параметрами здесь являются:

- множество выбранных для распределения нагрузки узлов;
- матрица прикрепления задач к узлам (может быть распределено более одной задачи на узел);
- и, наконец, множество узлов, участвующих в вычислениях и передаче данных, которое добавляется в результате формирования маршрутов передачи данных между задачами.

Стоит отметить, что введение транзитных узлов в качестве варьируемого параметра приводит к тому, что, имея ограничение на время выполнения комплекса задач, время, доступное для решения задач узлами также будет варьироваться в зависимости от количества транзитных узлов и, следовательно, будет варьироваться загруженность устройств.

Таким образом, сформированная задача относится к задачам структурно-параметрической оптимизации и может быть решена разбиением на ряд подзадач, а именно:

- формирования сетевой инфраструктуры;
- размещения задач по узлам;
- формирования маршрутов передачи данных;
- оцениванием сформированной архитектуры посредством стоимостной функции (5).

Следует отметить, что стоимостная функция имеет в качестве базового компонента загруженность устройства, а загруженность, в свою очередь, зависит как от трудоемкости выполняемых задач, так и от времени, за которое эти задачи выполняются. Время выполнения каждой индивидуальной задачи не определено: задача может быть выполнена за время, равное позднему сроку решения задачи, либо быстрее. Однако сокращение времени решения задачи приводит к экспоненциальному росту интенсивности отказов и, следовательно, значительно ухудшает значения ВБР. Поэтому будем считать, что с точ-

ки зрения индивидуальных ЦФ более выгодно растянуть решение задачи на время, определяемое поздним сроком завершения задачи, однако это не всегда возможно в силу наличия транзитных узлов на передачу данных между задачами.

Единственным известным поздним сроком завершения решения задачи является ограничение на время выполнения комплекса задач, следовательно, для прочих задач поздние сроки должны быть каким-то образом вычислены из имеющихся исходных данных. При этом могут быть неизвестны характеристики в момент времени, когда формируется архитектура каналов связи и их состояние, в то время как именно характеристики каналов передачи данных будут определять время, необходимое для передачи данных транзитными узлами и, соответственно, время, отводимое для решения задач обработки данных.

Поэтому целесообразность размещения двух задач с информационным

обменом на узлах будем определять количеством транзитных узлов, допуская при этом доступность канала передачи данных, а для определения времени, необходимого при расчете загруженности узлов, будем пользоваться априорными оценками на базе предположения, что на каждую задачу в графе задач, учитываемая задачи передачи данных, отводятся равные доли времени, исходя из расположения задачи на графе задач.

Сформированная задача относится к классу np (являясь структурно-параметрической), поэтому получение решения целесообразно производить при помощи одного из известных метавристических – имитации отжига, генетических алгоритмов и др. В данной работе был использован алгоритм имитации отжига [18-20].

Таким образом, укрупненная схема метода формирования архитектуры системы обработки данных будет иметь вид, как представлено на рис. 1.

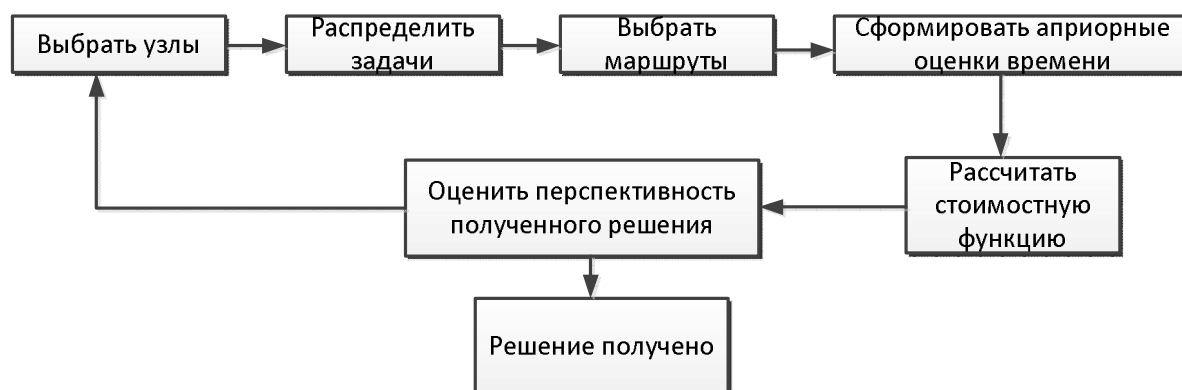


Рис. 1. Метод повышения эффективности распределенной архитектуры системы обработки данных

Fig.1. Data processing system distributed architecture efficiency improvement method

Формирование априорных оценок времени решения задач на основе данных о графе задач и сетевом графе

Оценочные значения времени, за которое должны быть решены вычислительные задачи, могут быть получены на основе анализа графов комплекса задач и графа сети, но при условии закрепления задач за узлами. При этом, как уже было сказано ранее, для вычисления ЦФ требуется информация о времени, отводимом для решения задачи

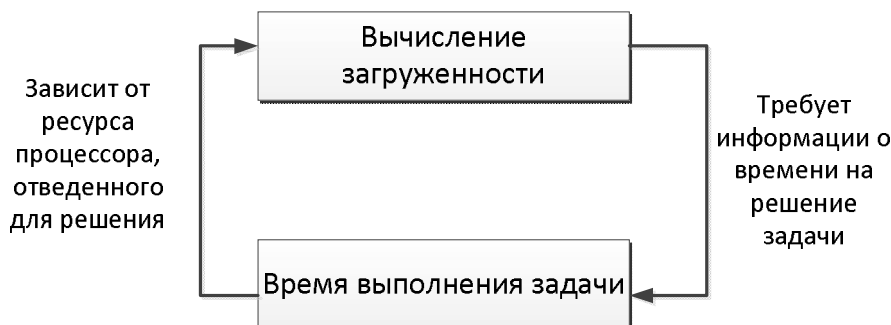


Рис. 2. Взаимная зависимость параметров времени решения задач и загрузки узлов вычислительных узлов

Fig. 2. The mutual dependency of task processing time and node workload parameters

Проведем оценивание времени решения задач следующим образом.

1. Рассмотрим граф комплекса задач, выделяя критический путь. Каждую задачу на критическом пути взвесим значением T/N , где T – ограничение на время выполнения комплекса задач, N – количество задач на критическом пути.

2. Составляем все возможные маршруты, которые начинаются и(или) заканчиваются на критическом пути. Для невзвешенных вершин производим разметку следующим образом:

$$T_i = (T - \sum T_c) / K, \quad (6)$$

на узле, в то время как время решения задачи узлом зависит от загруженности, т.е. от доли вычислительного ресурса, отводимого для решения этой задачи (рис.2). Поэтому использование априорных оценок, являясь довольно грубым приближением, позволяет по крайней мере оценить перспективность получаемых решений относительно эффективности архитектуры системы обработки данных.

где T_c – значения временных оценок для вершин на критическом пути; K – количество вершин в маршруте кроме взвешенных.

Таким образом, получены приближенные оценки времени, за которые должны быть решены вычислительные задачи.

Рассмотрим граф сети на произвольной итерации формирования архитектуры, т.е. когда задачи закреплены за узлами.

1. Случайным образом формируем маршруты передачи данных между двумя информационно связанными задачами.

2. Если вершины графа сети, на которых закреплены инцидентные вершины графа зада, также инцидентны, то модификации временных оценок не производится.

3. В противном случае:

- формируется задача передачи данных, трудоемкость которой рассчитывается исходя из трудоемкости приема и отправки данных между задачами i и $i+1$;

- задача передачи данных закрепляется за узлами на маршруте между теми узлами, за которыми закреплены задачи, инцидентные на графе задач;

- для маршрута на графе сети за вычетом одной из конечных вершин оценочное время модифицируется следующим образом:

$$T_j = \frac{T_i + T_{i+1}}{n - 1}, \quad (7)$$

где T_i – оценка времени на решение i -й задачи; T_{i+1} – оценка времени на решение инцидентной ей задачи на графе задач; n – количество узлов на графе сети в маршруте от вершины, за которой закреплена i -я задача, до вершины, за которой закреплена $i+1$ задача.

Таким образом, получаем априорные оценки времени на решение комплекса задач при условии закрепления за вычислительными узлами задач на одной из итераций формирования архитектуры. Эти временные оценки используются при расчете загруженности вычислительных узлов и, следовательно, для вычисления стоимостной функции.

Очевидно, что такие оценки имеют весьма приближенный характер, одна-

ко, в общем, они отражают тенденцию сокращения времени, отпущенного для решения вычислительных задач, при появлении маршрутов транзита данных. И, помимо этого, априорные оценки отражают тенденцию сокращения допустимого времени на передачу данных, при наличии ограничений на время выполнения комплекса задач.

Результаты и их обсуждение

В качестве примеров рассмотрим следующие структуры графов комплексов задач.

Для графа А структура фрагмента сети показана на рис. 5, для графа В – на рис. 6, соответственно.

Применяя последовательно шаги, представленные на схеме рис. 1, были получены следующие результаты (1000 итерация алгоритма имитации отжига).

На графиках рис. 8,10 видно, что применение мультипликативной стоимостной функции и использование априорных временных оценок приводят к значительному улучшению показателя загруженности как параметра индивидуальных ЦФ и, следовательно, к улучшению индивидуальных значений критериев эффективности архитектуры системы обработки данных.

Загруженность сетевой инфраструктуры также сокращается за счет размещения информационно связанных задач на одном вычислительном узле, что исключает передачу в сеть сравнительно большой доли данных.

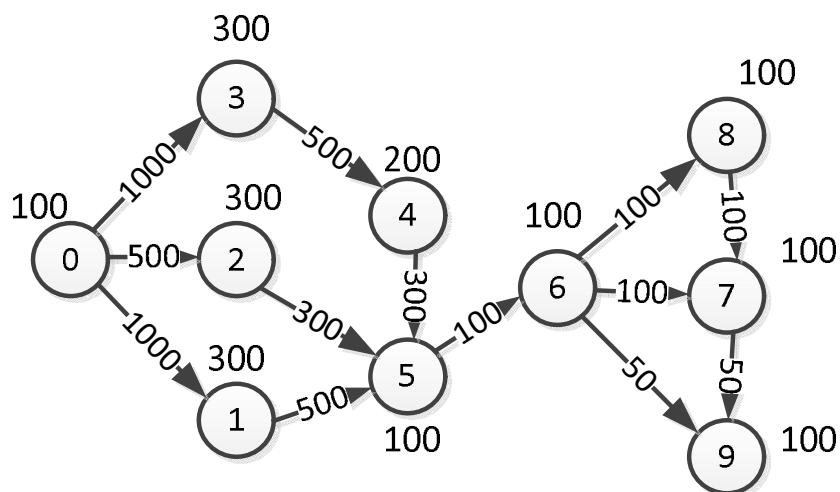


Рис. 3. Граф комплекса задач А

Fig. 3. Task complex A graph

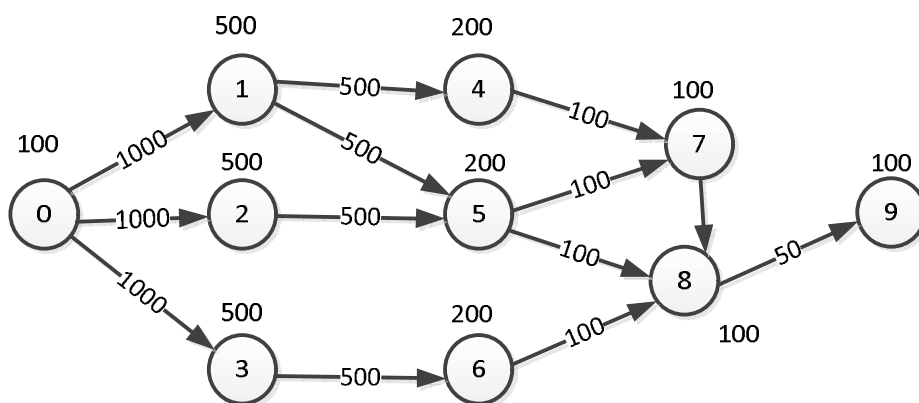


Рис. 4. Граф комплекса задач В

Fig. 4. Task complex B graph

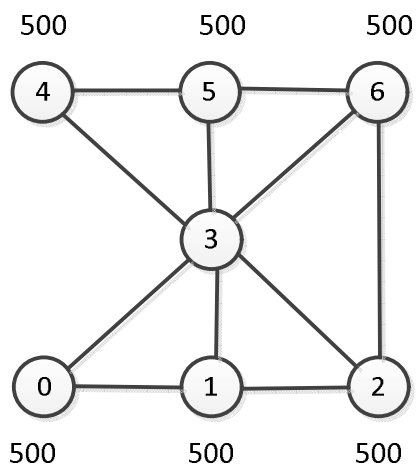


Рис. 5. Граф сети А

Fig. 5. Network A graph

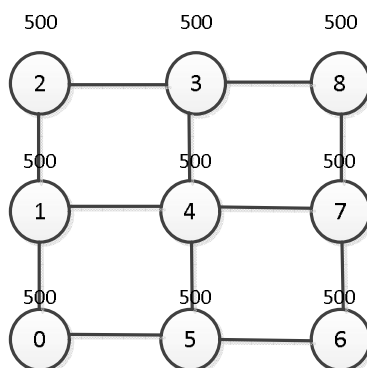


Рис. 6. Граф сети В

Fig. 6. Network B graph

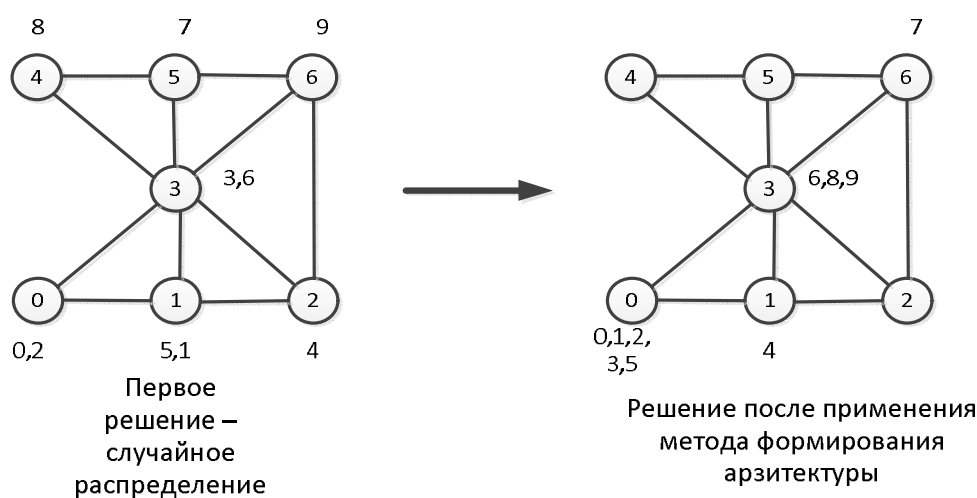


Рис. 7. Формирование эффективной архитектуры для графов А

Fig. 7. The efficient architecture forming for A graphs

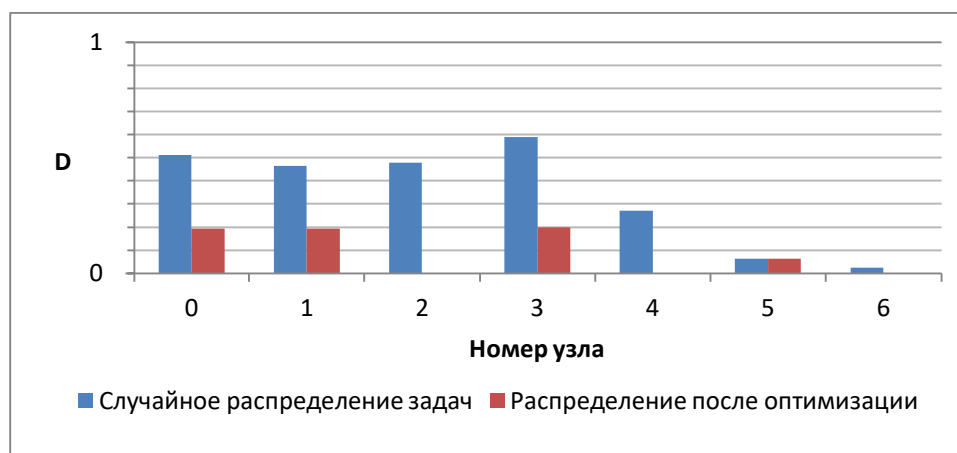


Рис. 8. Сравнение индивидуальной загрузки узлов при случайном распределении задач и после оптимизации по мультипликативной стоимостной функции и априорным оценкам времени решения задач для графов А

Fig. 8. The comparison of the individual nodes workload with the random tasks distribution and after the optimization by means of multiplicative convergence and the apriory tasks processing times for A graphs

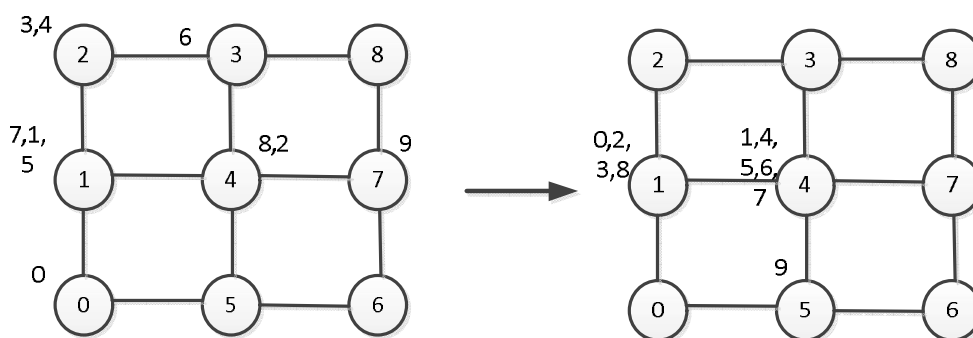


Рис. 9. Формирование эффективной архитектуры для графов В

Fig. 9. The efficient architecture forming for B graphs

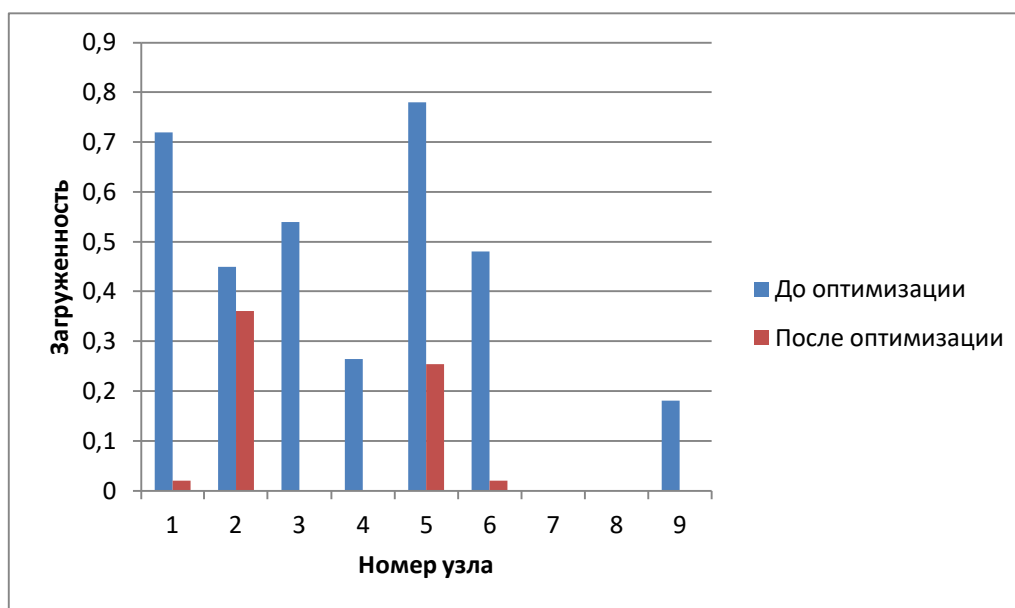


Рис. 10. Сравнение индивидуальной загрузки узлов при случайном распределении задач и после оптимизации по мультипликативной стоимостной функции и априорным оценкам времени решения задач для графов А

Fig.10. The comparison of the individual nodes workload with the random tasks distribution and after the optimization by means of multiplicative convergence and the apriory tasks processing times for A graphs

Например, для графа комплекса задач В вполне очевидно, что размещение задач 0 и 1 на соседних узлах приводит к необходимости передавать данные в сеть, в то время как размещение этих задач на одном узле, наоборот, предполагает отсутствие использования, что в свою очередь устраняет необходимость выделения дополнительного времени на передачу данных через сеть и приводит

к возможности распределить вычислительную нагрузку во времени.

Выводы

Учитывая возрастающую интенсивность использования туманных и краевых слоев сети для обработки больших данных, а также динамичность – как нагрузочную, так и инфраструктурную, формирование архитектур систем обра-

ботки данных становится актуальной технической проблемой.

Предложенная в данной статье формальная постановка задачи формирования ресурсно-эффективной архитектуры системы обработки данных отличается от предлагаемых ранее учетом влияния передачи данных по транзитным участкам сети, а также возможностью выделения узлов с необходимостью снижения нагрузки за счет использования весовых коэффициентов мультипликативной свертки.

Для решения поставленной структурно-параметрической задачи используются априорные оценки времени завершения работ, рассчитываемые на основе графов задач и графа рассматриваемого фрагмента сети. Проведение экс-

периментальных исследований показало, что даже использование таких неуточненных оценок, реализующих весьма грубые оценки, в отсутствие информации о выделенном ресурсе и времени решения задач узлами, приводит к существенному повышению эффективности в аспекте ресурсопотребления системы вычислительных узлов, формируемой для решения вычислительных задач обработки данных (до 50% снижения индивидуальной вычислительной нагрузки).

Следовательно, предложенный метод эффективен, а дальнейшее его исследование и уточнение с точки зрения получения более точных временных оценок выполнения задач узлами целесообразно.

Список литературы

1. Safronenkova I., Melnik Y. (2022). An Estimation of the Workload Relocation Techniques Application in Distributed CAD Systems Area. 10.1007/978-3-030-87178-9_56.
2. Hinkelmann Heiko, Zipf P., Glesner Manfred. (2009). A scalable reconfiguration mechanism for fast dynamic reconfiguration. 145 - 152. 10.1109/FPT.2008.4762377.
3. Huang Yong, Sun Silang, Chu Jing,. (2023). Energy- and time-optimal reconfiguration of spacecraft clusters with collision avoidance // Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 10.1177/09544100231179829.
4. Шевченко С. В. Оптимизация распределенных вычислений в системах параллельной обработки данных // ИТНОУ: информационные технологии в науке, образовании и управлении. 2018. №3 (7). URL: <https://cyberleninka.ru/article/n/optimizatsiya-raspredelennyh-vychisleniy-v-sistemah-parallelnoy-obrabotki-dannyh>.
5. Ворожцов А. С., Тутова Н. В., Тутов А. В. Оптимизация размещения облачных серверов в центрах обработки данных // T-Comm. 2015. №6. URL: <https://cyberleninka.ru/article/n/optimizatsiya-razmescheniya-oblachnyh-serverov-v-tsentrakh-obrabotki-dannyh>.
6. Jiang Junqiang, Cai Hailin, Xie Lunxin, Sun Zhifang, Pan Li, Yang Zhihe, Lu Ruiqi. (). A Time and Reliability Optimization Algorithm for Workflow Scheduling in Heterogeneous Distributed Computing System // Journal of Circuits, Systems and Computers. 2022. 31. 10.1142/S0218126622502528.

7. Kang Hongyue, Chang Xiaolin, Mišić Jelena, Misic Vojislav, Junchao Fan, Liu Yating. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach // IEEE Internet of Things Journal. 2023. PP. 1-1. 10.1109/JIOT.2023.3240173.
8. Kang Hongyue, Chang Xiaolin, Mišić Jelena, Misic Vojislav, Junchao Fan, Liu Yating. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach // IEEE Internet of Things Journal. 2023. PP. 1-1. 10.1109/JIOT.2023.3240173.
9. Geetmala Mrs, Badal Dr, Mishra Dr. A Clustered Approach for Load Balancing in Distributed Systems // Journal of University of Shanghai for Science and Technology. 2021. 23. 448-463. 10.51201/JUSST/21/05241.
10. Kumar Ravi, Rajagopalan S., Joseph P. Light Weight Native Edge Load Balancers for Edge Load Balancing // Green Intelligent Systems and Applications. 2023. 3. 48-55. 10.53623/gisa.v3i1.256.
11. Zhang Xiaohui, Nazari Hamed. An Efficient Resource Management Mechanism Based on Developed Political Optimizer in Fog Computing // Cybernetics and Systems. 2022. 1-15. 10.1080/01969722.2022.2145648.
12. Mahmoudi Zahra, Darbanian Elham, Nickray Mohsen. OPTIMAL ENERGY CONSUMPTION AND COST PERFORMANCE SOLUTION WITH DELAY CONSTRAINTS ON FOG COMPUTING // Jordanian Journal of Computers and Information Technology. 2023. 9. 1. 10.5455/jjcit.71-1667637331.
13. Duolikun Dilawaer, Nakamura Shigenari, Enokido Tomoya, Takizawa Makoto. (2022). Energy-Consumption Evaluation of the Tree-Based Fog Computing (TBFC) Model. 10.1007/978-3-031-20029-8_7.
14. Morkevicius Nerijus & Liutkevicius Agnius & Venčkauskas Algimantas. Multi-Objective Path Optimization in Fog Architectures Using the Particle Swarm Optimization Approach // Sensors. 2023. 23. 3110. 10.3390/s23063110.
15. The Network Load Balancer in Decentralized Systems // Advances in Cyber-Physical Systems. 2023. 8. 25-34. 10.23939/acps2023.01.025.
16. Xia Bin, Kong Fanyu, Zhou Jun, Tang Xiaosong, Gong Hong. A Delay-Tolerant Data Transmission Scheme for Internet of Vehicles Based on Software Defined Cloud-Fog Networks // IEEE Access. 2020. PP. 1-1. 10.1109/ACCESS.2020.2983440.
17. Барский А.Б. Параллельные информационные технологии. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007.
18. Савин А. Н., Тимофеева Н. Е. Применение алгоритма оптимизации методом имитации отжига на системах параллельных и распределённых вычислений // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2012. №1. URL: <https://cyberleninka.ru/article/n/primenenie-algoritma-optimizatsii-metodom-imitatsii-otzhiga-na-sistemah-parallelnyh-i-raspredelennyh-vychisleniy>.

19. Ingber A, Ingber Lester (2002). Simulated annealing: Practice versus theory.
20. Kirkpatrick S., Gelatt Jr. C. D., and Vecchi M. P. Optimization by Simulated Annealing // *Science*. 220. 1983. P. 671-680.

References

1. Safronenkova, I. & Melnik, Y.. (2022). An Estimation of the Workload Relocation Techniques Application in Distributed CAD Systems Area. 10.1007/978-3-030-87178-9_56.
2. Hinkelmann, Heiko & Zipf, P. & Glesner, Manfred. (2009). A scalable reconfiguration mechanism for fast dynamic reconfiguration. 145 - 152. 10.1109/FPT.2008.4762377.
3. Huang, Yong & Sun, Silang & Chu, Jing. Energy- and time-optimal reconfiguration of spacecraft clusters with collision avoidance. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 2023. 10.1177/09544100231179829.
4. Shevchenko Sergej Vasil'evich Optimizatsiya raspredelennykh vychislenii v sistemakh parallel'noi obrabotki dannykh [Optimization of distributed computing in parallel data processing systems]. *ITNOU: informacionnye tehnologii v nauke, obrazovanii i upravlenii = ITNOU: Information Technologies in Science, Education and Management*, 2018, no.3 (7). Available at: <https://cyberleninka.ru/article/n/optimizatsiya-raspredelennykh-vychisleniy-v-sistemakh-parallel'noi-obrabotki-dannykh>.
5. Vorozhtsov A. S., Tutova N. V., Tutov A. V. Optimizatsiya razmeshcheniya oblachnykh serverov v tsentrakh obrabotki dannykh [Optimization of cloud server placement in data centers]. *T-Comm*. 2015, no. 6. Available at: <https://cyberleninka.ru/article/n/optimizatsiya-razmeshcheniya-oblachnykh-serverov-v-tsentrakh-obrabotki-dannykh>.
6. Jiang, Junqiang & Cai, Hailin & Xie, Lunxin & Sun, Zhifang & Pan, Li & Yang, Zhihe & Lu, Ruiqi. A Time and Reliability Optimization Algorithm for Workflow Scheduling in Heterogeneous Distributed Computing System. *Journal of Circuits, Systems and Computers*. 2022. 31. 10.1142/S0218126622502528.
7. Kang, Hongyue & Chang, Xiaolin & Mišić, Jelena & Misic, Vojislav & Junchao, Fan & Liu, Yating. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach. *IEEE Internet of Things Journal*. 2023. PP. 1-1. 10.1109/JIOT.2023.3240173.
8. Kang, Hongyue & Chang, Xiaolin & Mišić, Jelena & Misic, Vojislav & Junchao, Fan & Liu, Yating. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach. *IEEE Internet of Things Journal*. 2023. PP. 1-1. 10.1109/JIOT.2023.3240173.
9. Geetmala, Mrs & Badal, Dr & Mishra, Dr. A Clustered Approach for Load Balancing in Distributed Systems. *Journal of University of Shanghai for Science and Technology*. 2021. 23. 448-463. 10.51201/JUSST/21/05241.
10. Kumar, Ravi & Rajagopalan, S. & P., Joseph. Light Weight Native Edge Load Balancers for Edge Load Balancing. *Green Intelligent Systems and Applications*. 2023. 3. 48-55. 10.53623/gisa.v3i1.256.

11. Zhang, Xiaohui & Nazari, Hamed. An Efficient Resource Management Mechanism Based on Developed Political Optimizer in Fog Computing. *Cybernetics and Systems*. 2022. 1-15. 10.1080/01969722.2022.2145648.
12. Mahmoudi, Zahra & Darbanian, Elham & Nickray, Mohsen. Optimal energy consumption and cost performance solution with delay constraints on fog computing. *Jordanian Journal of Computers and Information Technology*. 2023. 9. 1. 10.5455/jjcit.71-1667637331.
13. Duolikun, Dilawaer & Nakamura, Shigenari & Enokido, Tomoya & Takizawa, Makoto. (2022). Energy-Consumption Evaluation of the Tree-Based Fog Computing (TBFC) Model. 10.1007/978-3-031-20029-8_7.
14. Morkevicius, Nerijus & Liutkevicius, Agnius & Venčkauskas, Algimantas. Multi-Objective Path Optimization in Fog Architectures Using the Particle Swarm Optimization Approach. *Sensors*. 2023. 23. 3110. 10.3390/s23063110.
15. The Network Load Balancer in Decentralized Systems. *Advances in Cyber-Physical Systems*. 2023. 8. 25-34. 10.23939/acps2023.01.025.
16. Xia, Bin & Kong, Fanyu & Zhou, Jun & Tang, Xiaosong & Gong, Hong. A Delay-Tolerant Data Transmission Scheme for Internet of Vehicles Based on Software Defined Cloud-Fog Networks. *IEEE Access*. 2020. PP. 1-1. 10.1109/ACCESS.2020.2983440.
17. Barskij A.B. *Parallel'nye informacionnye tehnologii* [Parallel information technologies]. Moscow, 2007.
18. Savin A. N., Timofeeva N. E. Primenenie algoritma optimizacii metodom imitacii otzhiga na sistemah parallel'nyh i raspredel'nyh vychislenij [Application of the optimization algorithm by simulated annealing on systems of parallel and distributed computing]. *Izv. Sarat. un-ta. Nov. ser. Ser. Matematika. Mehanika. Informatika = Izv. Sarat. un-ta. Nov. ser. Ser. Mathematics. Mechanics. Computer Science*. 2012. №1. Available at: [https:// cyberleninka.ru/article/n/primenenie-algoritma-optimizatsii-metodom-imitatsii-otzhiga-na-sistemah-parallelnyh-i-raspredel'nyh-vychisleniy](https://cyberleninka.ru/article/n/primenenie-algoritma-optimizatsii-metodom-imitatsii-otzhiga-na-sistemah-parallelnyh-i-raspredel'nyh-vychisleniy).
19. Ingber, A & Ingber, Lester. (2002). Simulated annealing: Practice versus theory.
20. Kirkpatrick S., Gelatt Jr. C. D., and Vecchi M. P. Optimization by Simulated Annealing. *Science*. 1983. 220. P. 671-680.

Информация об авторе / Information about the Author

Клименко Анна Борисовна, кандидат технических наук, старший научный сотрудник, Научно-исследовательский институт многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета, г. Таганрог, Российская Федерация, e-mail: anna_klimenko@mail.ru, ORCID: <https://orcid.org/0000-0001-6527-8108>

Anna B. Klimenko, Cand. of Sci. (Engineering), Senior Research Fellow, Scientific Research Institute of Multiprocessor Computer Systems of Southern Federal University, Taganrog, Russian Federation, e-mail: anna_klimenko@mail.ru, ORCID: <https://orcid.org/0000-0001-6527-8108>