

РАЗДЕЛ 2

МОДЕЛИ, СИСТЕМЫ, СЕТИ В ТЕХНИКЕ

УДК 004.02

DOI 10.21685/2227-8486-2020-2-8

МЕТОД ПРОГРАММНОЙ ОПТИМИЗАЦИИ ПРОЦЕССА ДВУНАПРАВЛЕННОЙ ПЕРЕДАЧИ ДАННЫХ МЕЖДУ КОМПОНЕНТАМИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

*А. С. Бычков, В. А. Калашников, В. В. Борискин,
Н. С. Карамышева, О. В. Юрова*

METHOD FOR SOFTWARE OPTIMIZATION OF BIDIRECTIONAL DATA TRANSFER BETWEEN INFORMATION SYSTEM COMPONENTS

*A. S. Bychkov, V. A. Kalashnikov, V. V. Boriskin,
N. S. Karamysheva, O. V. Yurova*

Аннотация. *Предмет и цель работы.* Дано описание метода программной оптимизации процесса двунаправленной передачи данных между компонентами информационной системы без использования вертикального и горизонтального масштабирования. Метод был применен к информационной системе, состоящей из центрального сервера и множества клиентских подсистем. Характерной особенностью информационной системы является постоянное возрастание нагрузки на центральный сервер, вызванное увеличением числа участников обмена информацией и объема передаваемых данных. *Методы.* Метод предлагает распределить нагрузку на центральный сервер путем автоматического распределения расчетного времени запуска процедуры обмена данных между сколь угодно количеством участников в зависимости от заданной плотности обращения по времени, тем самым устранив проблему пиковых нагрузок на центральный сервер при массовом обращении клиентских подсистем. *Результаты и выводы.* Применение данного метода на целевой информационной системе позволило не прибегать к срочным дорогостоящим модернизациям архитектуры центральной части, вызванным ростом числа новых участников обмена информацией и расширением массива информации. Также данный метод позволяет развернуть целевую информационную систему в новом контуре на относительно малых вычислительных мощностях и отсрочить процедуру масштабирования.

Ключевые слова: программное распределение нагрузки, оптимизация передачи данных, информационные системы, конечный автомат.

Abstract. *Subject and goals.* This article describes a method for software optimization of bidirectional data transfer between components of an information system without using vertical and horizontal scaling. The method described in the article was applied to an information system consisting of a central server and a set of client subsystems.

© Бычков А. С., Калашников В. А., Борискин В. В., Карамышева Н. С., Юрова О. В., 2020

A constant increase in the load on the central server (caused by an increase in the number of participants in the exchange of information and an increase in the volume of transmitted data) is a characteristic feature of the described information system. The method allows to distribute the load on the central server by automatic distributing the start time of the data exchange procedure between as many participants as it is needed depending on the time specified request density; thereby this eliminates the problem of peak loads on the central server that occurs as a result of mass requests from client subsystems. The application of this method on the target information system made it possible to avoid urgent expensive upgrades of the architecture of the central part whose would have been necessary due to the growth of the number of new participants in the information exchange and the expansion of the information array. This method also allows to deploy the target information system in the new environment on relatively small computing capacities and delay the scaling procedure.

Keywords: software load distributing, data transfer optimization, information systems, finite state machine.

Введение

Наиболее характерная черта текущего состояния мировой экономики – колоссальные успехи и достижения в области вычислительной техники и информационных технологий, а также развитие наукоемких производств. Высокие темпы развития науки и технологий обработки информации, а главное масштабы их внедрения во многие сферы жизни превратили научно-техническую революцию в естественный процесс, она стала перманентной [1]. В условиях динамичного развития рынка, усложнения его инфраструктуры информация становится таким же стратегическим ресурсом, как материалы и энергия.

Современные технологии, позволяющие создавать, хранить, перерабатывать информацию, обеспечивать эффективные способы ее представления, стали важным фактором обеспечения конкурентоспособности и средством повышения эффективности управления всеми сферами общественной жизнедеятельности. Уровень информатизации является сегодня одним из главных факторов успешного развития любого предприятия.

На фоне повсеместной информатизации общества и производства стали появляться компании, занимающиеся разработкой комплексных информационных систем. Однако каждое подобное предприятие через некоторое время сталкивается с проблемой роста нагрузок при передаче данных между компонентами информационной системы по мере роста ее функциональности.

Рассматривается один из путей оптимизации передачи данных между компонентами информационной системы и снижения нагрузок без использования дорогостоящего процесса масштабирования.

Описание целевой системы. Постановка задачи

Целевая информационная система (ИС) состоит из центрального сервера обработки данных, единой базы данных (БД) [2, 3], а также из множества клиентских подсистем [4]. ИС также взаимодействует со сторонними сервисами. Общая схема ИС представлена на рис. 1.

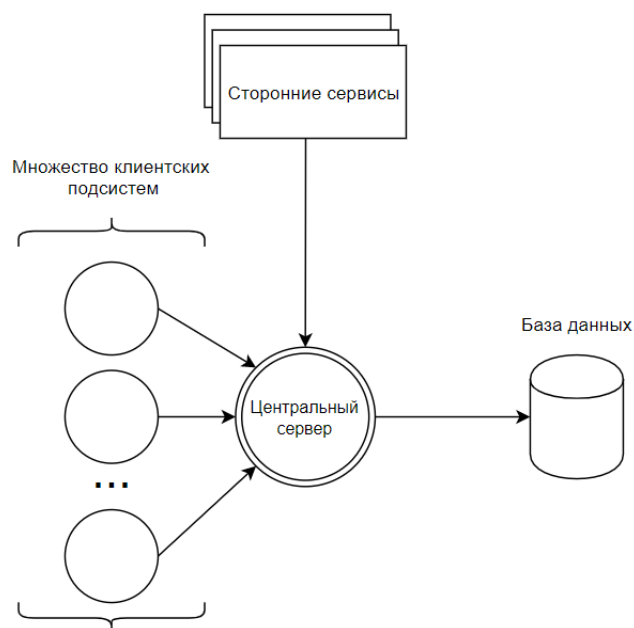


Рис. 1. Общая схема ИС

Клиентская подсистема состоит из внутреннего сервера организации, локальной БД и множества вспомогательных ПО, которые устанавливаются в зависимости от потребностей обслуживаемой организации [5]. Схема клиентской подсистемы представлена на рис. 2.

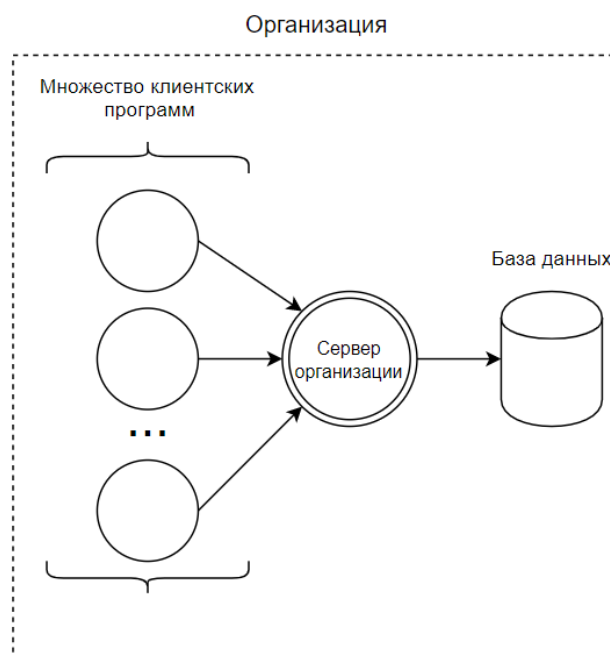


Рис. 2. Общая схема подсистемы

В ходе деятельности клиентских подсистем создается огромный массив информации [6], который необходимо передать на центральный сервер. Центральный сервер также производит обработку данных и должен передать клиентской системе свои, хранящиеся в БД данные и пакеты информации, полученные от связанных подсистем. Процесс обмена данными между клиентской подсистемой и центральным сервером далее будет называться синхронизацией [7].

Пакет синхронизации представляет собой XML-документ с цифровой подписью [8–11], где данные разнесены по секциям. Клиентская подсистема инициирует процедуру синхронизации, выполняя запрос разрешения процедуры, сервер передает код разрешения или запрета. Если клиент получил запрет, то он ждет отведенное время t , после чего снова шлет запрос на разрешение процедуры. Если клиент получил код разрешения, то он отправляет на сервер XML-пакет с данными, сервер же помещает этот пакет в очередь на обработку.

Математическая модель работы модуля синхронизации продемонстрирована на рис. 3, где

$E_{request}^i$ – запрос разрешения на процедуру синхронизации данных с сервером i -го клиента;

X_1 – сигнал разрешения на процедуру синхронизации;

$\overline{X_1}$ – запрещающий сигнал;

t – время, через которое повторно выполняется запрос [8];

E_{ready}^i – отправка пакета данных на сервер для обработки i -го клиента;

E_{ready}^{i+1} – отправка пакета данных на сервер сторонним клиентом;

E_{queue} – состояние нахождения в очереди на обработку;

X_2 – сигнал перехода на процедуру обработки;

$\overline{X_2}$ – запрещающий сигнал, обработчик занят;

$E_{process}^i$ – обработка данных от i -го клиента;

$E_{response}^i$ – формирование ответного пакета данных для i -го клиента.

С ростом функциональных возможностей ИС и числа обслуживаемых организаций также возрастает нагрузка на центральный сервер до критического. В таких случаях применяют вертикальное и горизонтальное масштабирование. Вертикальное масштабирование позволяет увеличивать производительность системы путем замены вычислительных компонентов на более мощные. Горизонтальное масштабирование системы увеличивает производительность за счет установки новых узлов с сервером балансировки нагрузки, распределяющим задачи между узлами. Вертикальное и горизонтальное масштабирование позволило бы решить задачу, но лишь на время, так как такие способы на каждой итерации масштабирования требуют экспоненциально увеличивающихся ресурсов.

Требуется реорганизовать работу модуля синхронизации данных для распределения нагрузки на сервер по времени так, чтобы информация не теряла свою актуальность и попадала на центральный сервер своевременно.

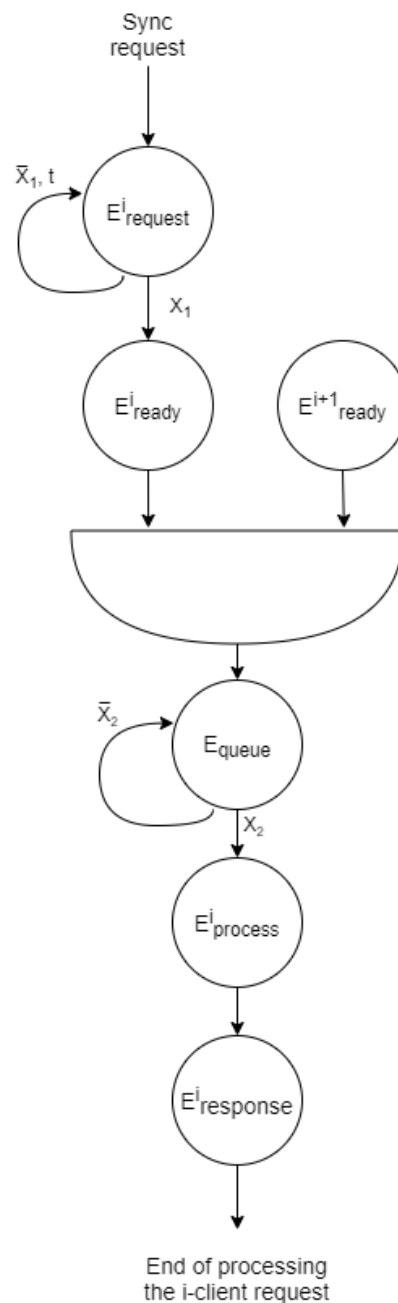


Рис. 3. Математическая модель алгоритма синхронизации

Решение задачи

Для решения поставленной задачи в первую очередь сама синхронизация была разделена на логические типы [12–14], где участвуют только некоторые секции из имеющихся. Данный шаг позволил настроить периодичность передачи данных и оптимизировать время сеанса синхронизации для каждого из логических типов синхронизации. Однако все еще существует необходимость полной синхронизации, где должны участвовать все секции [15].

Как правило, полная синхронизация производится при первичном заполнении клиентской БД при вводе новой сущности или в организациях с малым оборотом данных, где нецелесообразно разделять синхронизацию на отдельные секции.

Чтобы избавиться от единовременного обращения клиентских подсистем, было решено создать модуль автоматического распределения времени синхронизации [16, 17], создав тем самым расширяемое расписание обращений.

Прежде всего были определены понятия разрешенного и запрещенного времени обращения. Разрешенный временной промежуток устанавливается для явного указания оптимального времени синхронизации. Иначе указывается только запрещенный промежуток для всех клиентов, в котором не должен проводиться сеанс синхронизации [18]. Разделение промежутков на запрещенные и разрешенные гарантирует выделение удобного времени для технического обслуживания ИС, например, создание бэкапа БД в ночное время.

На WEB-интерфейсе ИС инженером задается временной промежуток выражением, которое представляет собой последовательность временных интервалов. Выражение должно иметь вид HH:mm-HH:mm. Перечисление интервалов задается знаком точки с запятой (;). Для указания периода, когда синхронизация запрещена, перед интервалом необходимо поставить восклицательный знак (!). Количество разрешенных промежутков указывает на количество необходимых сеансов синхронизации, время каждого сеанса должно укладываться в соответствующий временной интервал.

При указании запрещающих выражений сеанс синхронизации будет распределен на все 24 часа за исключением указанных интервалов [19].

Процедура автоматического распределения начинается с расчета базового времени для каждой клиентской подсистемы, оно рассчитывается по формуле

$$T_b = ID_o / k, \quad (1)$$

где T_b – базовое время; ID_o – числовой идентификатор организации в центральном сервере; k – коэффициент распределения базового времени.

Коэффициент распределения базового времени k свой для каждого типа синхронизации и задается из допустимой плотности сеансов синхронизации по времени. Чем меньше коэффициент, тем больше сдвиг по времени получает каждая организация на проведение синхронизации.

Расчет итогового времени для разрешенных временных промежутков рассчитывается по формуле

$$T_{si} = \left[\frac{P_{si} + T_b}{P_{ei}} \right], \quad (2)$$

где T_{si} – итоговое время запуска сеанса синхронизации для i -го разрешенного интервала, указанное в минутах от начала дня; P_{si} – время начала разрешенного периода для i -го сеанса; P_{ei} – время конца разрешенного периода для i -го сеанса.

В случае указания запрещенных промежутков происходит проверка вхождения T_b в эти самые промежутки. В случае, если T_b попадает в запрещенный промежуток, то происходит перерасчет по формуле

$$T_s = \left\lceil \frac{P_{ei} + T_b}{24 \cdot 60} \right\rceil, \quad (3)$$

где T_s – новое рассчитанное время для запуска синхронизации; P_{ei} – время конца i -го запрещенного для синхронизации периода.

В случае, если T_s попадает в следующий запрещенный период, то происходит его повторный перерасчет по формуле (3).

На расчет T_s выделено ограниченное количество попыток в случае, если было задано некорректное выражение. По истечению количества попыток происходит сброс указанных выражений и установка значений по умолчанию. Далее происходит регенерация T_s проблемного типа синхронизации уже по новому выражению [20].

Результат

В ходе ввода в эксплуатацию сервиса автораспределения были собраны данные о нагрузках на сервер для оценки эффективности метода.

Для сравнения далее приведены графики количества запросов на сеанс синхронизации, распределенных по времени, за дни, в которых наблюдалась низкая активность.

На рис. 4 приведены графики до и после ввода автоматического распределения.

График, изображенный сплошной линией – график до ввода оптимизации. Весь массив запросов пришелся на временной промежуток от начала и до конца рабочего дня в обслуживаемых организациях. Всего было обработано 33 445 пакетов данных. Из рисунка видно, что пик пришелся на 7 часов утра.

График, изображенный пунктирной линией – график после ввода автоматического распределения. Было обработано 33 835 пакетов данных. Наблюдается перераспределение нагрузки на ночное время, а также сниженные показатели пиков в дневные часы.

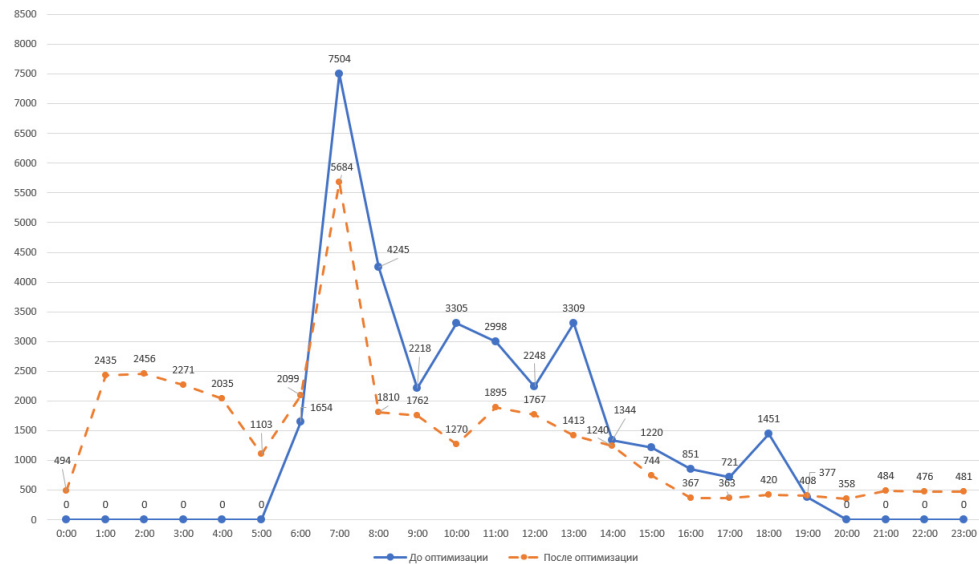


Рис. 4. Графики количества сеансов синхронизации по часам

Далее на рис. 5 приведены графики количества запросов на обработку пакетов в интервале 10 минут. Данные брались за тот же день, что и для графиков на рис. 4.

На рис. 5 изображен сплошной линией график, по которому видно, что основной массив запросов за час укладывается в первые 2 минуты. Это вызвано установкой у всех клиентских подсистем начинать передачу данных в строго определенное время.

После ввода автоматического распределения наблюдается равномерное распределение количества запросов в первые 4 минуты, не превышая критического значения (график представлен пунктиром).

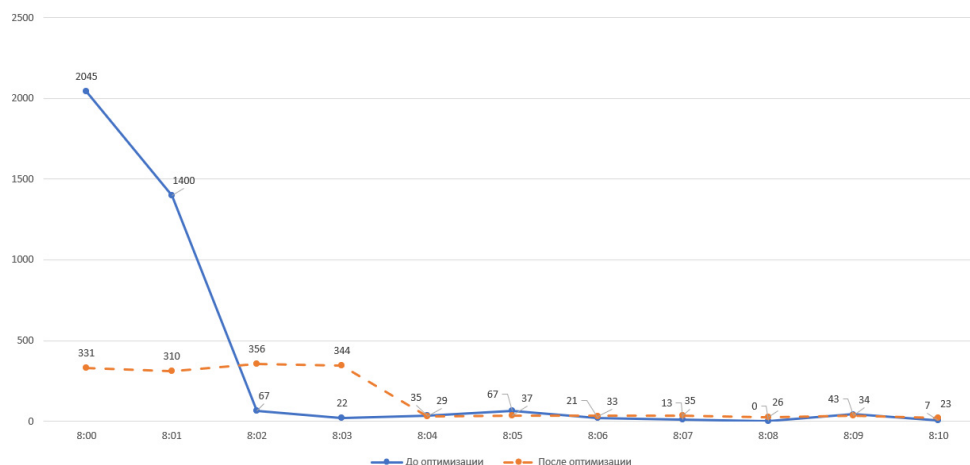


Рис. 5. Графики количества сеансов синхронизации на интервале 10 минут

Заключение

В ходе тестов после ввода в эксплуатацию модуля автоматического распределения времени синхронизации наблюдалось значительное снижение нагрузки на центральный сервер в пиковые часы.

Рассмотренный метод решения задачи оптимизации нагрузки на сервер оказался достаточно эффективным и экономически менее затратным, чем горизонтальное и вертикальное масштабирование ИС. Достоинством метода можно считать гибкость, которая обеспечивает возможность быстрого распределения времени сеансов синхронизации для новых подключенных клиентских систем.

Библиографический список

1. Авдошин, С. М. Информатизация бизнеса. Управление рисками / С. М. Авдошин, Е. Ю. Песоцкая. – Москва : ДМК Пресс, 2011. – 176 с.
2. Грофф, Д. SQL. Полное руководство / Д. Грофф, П. Вайнберг, Э. Оппель. – Санкт-Петербург : Вильямс. Издательский дом, 2015. – № 3. – С. 87.
3. Wilton, P. Beginning SQL / P. Wilton, J. Colby. – Indianapolis : John Wiley & Sons Limited, 2005. – P. 7.
4. Трутнев, Д. Р. Архитектуры информационных систем. Основы проектирования / Д. Р. Трутнев. – Санкт-Петербург : НИУ ИТМО, 2012. – 66 с.

5. Бабинский, А. З. Определение базовых сервисов, разработка методики наполнения и методов реализации образовательных порталов / А. З. Бабинский, А. А. Букатов, В. А. Шапиро. – 2003. – С. 7. – URL: credit.edu.ru
6. Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения / В. А. Мачалин, А. Н. Токарев, Д. А. Трокоз, Д. В. Пашченко, М. П. Синев // Радиотехника. – 2010. – № 8. – С. 51–54.
7. Моделирование автоматизированной информационной системы государственных услуг / Д. В. Пашченко, Д. А. Трокоз, М. П. Синев, А. И. Мартышкин, В. В. Кормишина, Д. Д. Малинин // XXI век: итоги прошлого и проблемы настоящего плюс. – 2018. – № 2. – С. 6–9.
8. Вашкевич, Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки / Н. П. Вашкевич. – Москва, 2004. – С. 5.
9. Вашкевич, Н. П. Достоинство формального языка, основанного на концепции недетерминизма, при структурной реализации параллельных систем логического управления процессами и ресурсами / Н. П. Вашкевич, Р. А. Бикташев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 1. – С. 4–8.
10. Вашкевич, Н. П. Формализация алгоритмов управления многопоточным доступом к разделяемым ресурсам на основе использования событийных недетерминированных автоматов / Н. П. Вашкевич, Р. А. Бикташев, М. П. Синев // Интеллект. Инновации. Инвестиции. – 2014. – № 1. – С. 128–133.
11. Новосельцев, В. Б. Обработка рекурсивных данных конечными автоматами / В. Б. Новосельцев, В. В. Соколова // Известия Томского политехнического университета. – 2006. – № 7.
12. Темпоральный анализ киберфизических систем с использованием теории автоматов / Д. А. Трокоз, Н. В. Исхаков, М. П. Синев, М. А. Митрохин, Н. О. Сивишкина // XXI век: Итоги прошлого и проблемы настоящего плюс. – 2019. – № 3 (47).
13. Кормен, Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – Санкт-Петербург : Вильямс. Издательский дом, 2013. – № 3. – С. 140–173.
14. Стивенс, Р. Алгоритмы. Теория и практическое применение / Р. Стивенс. – John Wiley & Sons, 2013. – С. 72.
15. Макконнелл, С. Совершенный код: Практическое руководство по разработке программного обеспечения / С. Макконнелл. – Санкт-Петербург : Русская редакция/БХВ, 2017. – С. 126–129.
16. Real: методология и CASE-средство для разработки систем реального времени и информационных систем / А. Н. Терехов, К. Ю. Романовский, Д. В. Кознов, П. С. Долгов, А. Н. Иванов. – URL http://www.math.spbu.ru/user/dkoznov/papers/Real_General.pdf (дата обращения: 21.03.2020).
17. Scheduling queues in the ethernet switch, considering the waiting time of frames / E. Kizilov, N. Konnov, K. Nikishin, D. Pashchenko, D. Trokoz // 2016 International Conference on Electronic, Information and Computer Engineering. – Hong Kong : ICEICE, 2016.
18. Кнут, Д. Искусство программирования / Д. Кнут – Санкт-Петербург : Вильямс. Издательский дом, 2000. – Т. 1. Основные алгоритмы. – С. 27.
19. Синев, М. П. Реализация стратегий объективного контроля в радиолокационных комплексах / Д. В. Пашченко, М. П. Синев // Проведение научных исследований в области обработки, хранения, передачи и защиты информации : сб. науч. тр. – Ульяновск : УлГТУ, 2009. – № 4. – С. 155–158.
20. Pashchenko, D. V. Directly executable formal models of middleware for MANET and Cloud Networking and Computing / D. V. Pashchenko [et al.] // Journal of Physics: Conference Series. – IOP Publishing. – 2016. – Т. 710. – № 1. – С. 2024.

References

1. Avdoshin C. M., Pesotskaya E. Yu. *Informatizatsiya biznesa. Upravlenie riskami* [Computerization of business. Risk management]. Moscow: DMK Press, 2011, 176 p. [In Russian]
2. Groff D., Vaynberg P., Oppel' E. *SQL. Polnoe rukovodstvo* [SQL. Complete guide]. Saint-Petersburg: Vil'yams. Izdatel'skiy dom, 2015, no. 3, p. 87. [In Russian]
3. Wilton P., Colby J. *Beginning SQL*. Indianapolis: John Wiley & Sons Limited, 2005, p. 7.
4. Trutnev D. R. *Arkhitektury informatsionnykh sistem. Osnovy proektirovaniya* [Architecture of information systems. Design basics]. Saint-Petersburg: NIU ITMO, 2012, 66 p. [In Russian]
5. Babinskiy A. Z., Bukatov A. A., Shapiro V. A. *Opredelenie bazovykh servisov, razrabotka metodiki napolneniya i metodov realizatsii obrazovatel'nykh portalov* [Definition of basic services, development of methods of filling and implementation of educational portals]. 2003, p. 7. Available at: credit.edu.ru [In Russian]
6. Machalin V. A., Tokarev A. N., Trokoz D. A., Pashchenko D. V., Sinev M. P. *Radio-tekhnika* [Radiotechnics]. 2010, no. 8, pp. 51–54. [In Russian]
7. Pashchenko D. V., Trokoz D. A., Sinev M. P., Martyshkin A. I., Kormishina V. V., Malinin D. D. *XXI vek: itogi proshlogo i problemy nastoyashchego plyus* [XXI century: results of the past and problems of the present plus]. 2018, no. 2, pp. 6–9. [In Russian]
8. Vashkevich N. P. *Nedeterminirovannyye avtomaty v proektirovanii sistem parallel'noy obrabotki* [Nondeterministic automata in the design of parallel processing systems]. Moscow, 2004, p. 5. [In Russian]
9. Vashkevich N. P., Biktashev R. A. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki* [News of higher educational institutions. Volga region. Technical science]. 2011, no. 1, pp. 4–8. [In Russian]
10. Vashkevich N. P., Biktashev R. A., Sinev M. P. *Intellekt. Innovatsii. Investitsii* [Intelligence. Innovations. Investment]. 2014, no. 1, pp. 128–133. [In Russian]
11. Novosel'tsev V. B., Sokolova V. V. *Izvestiya Tomskogo politekhnicheskogo universiteta* [News of Tomsk Polytechnic University]. 2006, no. 7. [In Russian]
12. Trokoz D. A., Iskhakov N. V., Sinev M. P., Mitrokhin M. A., Sivishkina N. O. *XXI vek: Itogi proshlogo i problemy nastoyashchego plyus* [XXI century: Results of the past and problems of the present plus]. 2019, no. 3 (47). [In Russian]
13. Kormen T., Leyzerson Ch., Rivest R., Shtayn K. *Algoritmy. Postroenie i analiz* [Algorithms. The construction and analysis]. Saint-Petersburg: Vil'yams. Izdatel'skiy dom, 2013, no. 3, pp. 140–173. [In Russian]
14. Stivens R. *Algoritmy. Teoriya i prakticheskoe primeneniye* [Algorithms. Theory and practical application]. John Wiley & Sons, 2013, p. 72. [In Russian]
15. Makconnell S. *Sovershennyy kod: Prakticheskoe rukovodstvo po razrabotke programmnogo obespecheniya* [Perfect code: a Practical guide to software development]. Saint-Petersburg: Russkaya redaktsiya/BKhV, 2017, pp. 126–129. [In Russian]
16. Terekhov A. N., Romanovskiy K. Yu., Koznov D. V., Dolgov P. S., Ivanov A. N. *Real: metodologiya i CASE-sredstvo dlya razrabotki sistem real'nogo vremeni i informatsionnykh sistem* [Real: Case-a tool for developing real-time systems and information systems]. Available at http://www.math.spbu.ru/user/dkoznov/papers/Real_General.pdf (accessed Mar. 21, 2020). [In Russian]
17. Kizilov E., Konnov N., Nikishin K., Pashchenko D., Trokoz D. *2016 International Conference on Electronic, Information and Computer Engineering*. Hong Kong: ICEICE, 2016.
18. Knut D. *Iskusstvo programmirovaniya* [The art of programming]. Saint-Petersburg: Vil'yams. Izdatel'skiy dom, 2000, vol. 1, p. 27. [In Russian]

19. Sinev M. P., Pashchenko D. V. *Provedenie nauchnykh issledovaniy v oblasti obrabotki, khraneniya, peredachi i zashchity informatsii: sb. nauch. tr.* [Research in the field of processing, storage, transmission and information protection : collection of scientific works]. Ulyanovsk: UIGTU, 2009, no. 4, pp. 155–158. [In Russian]
20. Pashchenko D. V. et al. *Journal of Physics: Conference Series*. IOP Publishing. 2016, vol. 710, no. 1, p. 2024.

Бычков Андрей Станиславович
кандидат технических наук, доцент,
кафедра вычислительной техники,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
e-mail: a_s_bychkov@mail.ru

Bychkov Andrey Stanislavovich
candidate of technical sciences,
associate professor,
sub-department of computer engineering,
Penza State University
(40 Krasnaya street, Penza, Russia)

Калашников Виталий Александрович
студент,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: mymailfordev@gmail.com

Kalashnikov Vitaliy Aleksandorovich
student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Борискин Вячеслав Владимирович
студент,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: s.boriskin287@gmail.com

Boriskin Vyacheslav Vladimirovich
student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Карамышева Надежда Сергеевна
кандидат технических наук, доцент,
кафедра вычислительной техники,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: karamyshevans@yandex.ru

Karamysheva Nadezhda Sergeevna
candidate of technical sciences,
associate professor,
sub-department of computer engineering,
Penza State University
(40 Krasnaya street, Penza, Russia)

Юрова Ольга Викторовна
кандидат технических наук, доцент,
кафедра вычислительной техники,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: olyura55@rambler.ru

Yurova Olga Viktorovna
candidate of technical sciences,
associate professor,
sub-department of computer engineering,
Penza State University
(40 Krasnaya street, Penza, Russia)

Образец цитирования:

Метод программной оптимизации процесса двунаправленной передачи данных между компонентами информационной системы / А. С. Бычков, В. А. Калашников, В. В. Борискин, Н. С. Карамышева, О. В. Юрова // Модели, системы, сети в экономике, технике, природе и обществе. – 2020. – № 2 (34). – С. 110–120. – DOI 10.21685/2227-8486-2020-2-8.