

*Intellectual Technologies
on Transport
No 2*



*Интеллектуальные технологии
на транспорте
№ 2*

*Санкт-Петербург
St. Petersburg
2020*

Интеллектуальные технологии на транспорте
№ 2, 2020

ISSN 2413-2527

Сетевой электронный научный журнал, свободно распространяемый через Интернет.
Публикуются статьи на русском и английском языках с результатами исследований
и практических достижений в области интеллектуальных технологий
и сопутствующих им научных исследований.

Журнал основан в 2015 году.

Учредитель и издатель

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I» (ФГБОУ ВО ПГУПС)

Главный редактор

Хомоненко А. Д., д.т.н., проф., С.-Петербург, РФ

Сопредседатели редакционного совета

Панычев А. Ю., ректор ПГУПС, С.-Петербург, РФ

Чаркин Е. И., директор по ИТ-технологиям ОАО «РЖД», Москва, РФ

Редакционный совет

Ададулов С. Е., проф., Москва, РФ
Дудин А. Н., д.т.н., проф., БГУ, Минск, Беларусь
Корниенко А. А., проф., ПГУПС, С.-Петербург, РФ
Ковалец П., проф., Техн. ун-т, Варшава, Польша
Меркурьев Ю. А., проф., РТУ, Рига, Латвия
Нестеров В. М., проф., СПбГУ, С.-Петербург, РФ

Пустарнаков В. Ф., ген. дир. «Газинформсервис»,
С.-Петербург, РФ
Титова Т. С., проф., проректор ПГУПС,
С.-Петербург, РФ
Федоров А. Р., ген. дир. «ДигДез», С.-Петербург, РФ
Юсупов Р. М., проф., чл.-корр. РАН, С.-Петербург, РФ

Редакционная коллегия

Бубнов В. П., проф., С.-Петербург, РФ – зам. гл. ред.
Александрова Е. Б., проф., С.-Петербург, РФ
Атилла Элчи, проф., ун-т Аксарай, Турция
Басыров А. Г., проф., С.-Петербург, РФ
Безродный Б. Ф., проф., Москва, РФ
Благовещенская Е. А., проф., С.-Петербург, РФ
Булавский П. Е., д.т.н., доц., С.-Петербург, РФ
Василенко М. Н., проф., С.-Петербург, РФ
Глухов А. П., д.т.н., Москва, РФ
Гуда А. Н., проф., Ростов-на-Дону, РФ
Железняк В. К., проф., Новополоцк, Беларусь
Заборовский В. С., проф., С.-Петербург, РФ
Зегжда П. Д., проф., С.-Петербург, РФ
Канаев А. К., проф., С.-Петербург, РФ
Котенко А. Г., д.т.н., доц., С.-Петербург, РФ
Куренков П. В., проф., Москва, РФ
Лецкий Э. К., проф., Москва, РФ

Макаренко С. И., д.т.н., доц., С.-Петербург, РФ
Мирзоев Т. А., асс. проф., Джорджия, США
Наседкин О. А., к.т.н., доц., С.-Петербург, РФ
Никитин А. Б., проф., С.-Петербург, РФ
Новиков Е. А., д.т.н., доц., С.-Петербург, РФ
Охтилев М. Ю., проф., С.-Петербург, РФ
Привалов А. А., проф., С.-Петербург, РФ
Соколов Б. В., проф., С.-Петербург, РФ
Таранцев А. А., проф., С.-Петербург, РФ
Утепбергенов И. Т., проф., Алматы, Казахстан
Филипченко С. А., к.т.н., доц., Москва, РФ
Фозилов Ш. Х., проф., Ташкент, Узбекистан
Фу-Ниан Ху, проф., Цзянсу, Китай
Хабаров В. И., проф., Новосибирск, РФ
Ходаковский В. А., проф., С.-Петербург, РФ
Чехонин К. А., проф., Хабаровск, РФ
Ялышев Ю. И., проф., Екатеринбург, РФ

Адрес редакции:

190031, Санкт-Петербург, Московский пр., 9, ауд. 1–210
e-mail: itt-pgups@yandex.ru

Журнал зарегистрирован Федеральной службой по надзору в сфере связи и массовых коммуникаций,
свидетельство Эл № ФС77-61707 от 07 мая 2015 г.

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ).

Периодичность выхода – 4 номера в год. Выпуски журнала доступны на сайте <http://itt-pgups.ru>.

Копии архивов с выпусками журнала проходят государственную регистрацию как электронное издание
сетевого распространения в НТЦ "Информрегистр".

Информация предназначена для детей старше 12 лет.

© Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I», 2020

Intellectual Technologies on Transport

Issue 2, 2020

ISSN 2413-2527

Network electronic scientific journal, open access. It publishes articles in Russian and English with the results of research and practical achievements in the field of intelligent technologies and associated research.

Founded in 2015.

Founder and Publisher

Federal State Educational Institution of Higher Education
«Emperor Alexander I Petersburg State Transport University»

Editor-in-Chief

Khomonenko A. D., Dr. Sc., Prof., St. Petersburg, Russia

Co-chairs of the Editorial Council

Panychev A. Yu., rector of PSTU, St. Petersburg, Russia
Charkin E. I., CIO of JSC "RZD", Moscow, Russia

Editorial Council Members

Adadurov S. Ye., Prof., Moscow, Russia	Pustarnakov V. F., CEO at «Gazinformservice» Ltd., St. Petersburg, Russia
Dudin A. N., Prof., BSU, Minsk, Belarus	Titova T. S., Prof., PSTU, St. Petersburg, Russia
Kornienko A. A., Prof., PSTU, St. Petersburg, Russia	Fedorov A. R., CEO at «Digital Design» Ltd., St. Petersburg, Russia
Kovalets P., Prof., Tech. University, Warsaw, Poland	Yusupov R. M., Prof., Corr. Member of RAS, St. Petersburg, Russia
Merkuryev Y. A., Prof., RTU, Academician of the Latvian Academy of Sciences, Riga, Latvia	
Nesterov V. M., Prof., SPbSU, St. Petersburg, Russia	

Editorial Board Members

Bubnov V. P., Prof., St. Petersburg, Russia – Deputy Editor-in-Chief	Makarenko S. I., Dr. Sc., As. Prof., St. Petersburg, Russia
Aleksandrova E. B., Prof., St. Petersburg, Russia	Mirzoev T. A., As. Prof., Georgia, USA
Atilla Elci, Prof., Aksaray University, Turkey	Nasedkin O. A., As. Prof., St. Petersburg, Russia
Basyrov A. G., Prof., St. Petersburg, Russia	Nikitin A. B., Prof., St. Petersburg, Russia
Bezrodny B. F., Prof., Moscow, Russia	Novikov Y. A., Dr. Sc., As. Prof., St. Petersburg, Russia
Blagoveshchenskaya E. A., Prof., St. Petersburg, Russia	Ohtilev M. Y., Prof., St. Petersburg, Russia
Bulavsky P. E., Dr. Sc., As. Prof., St. Petersburg, Russia	Privalov A. A., Prof., St. Petersburg, Russia
Vasilenko M. N., Prof., St. Petersburg, Russia	Sokolov B. V., Prof., St. Petersburg, Russia
Glukhov A. P., Dr. Sc., St. Petersburg, Russia	Tarantsev A. A., Prof., St. Petersburg, Russia
Guda A. N., Prof., Rostov-on-Don, Russia	Utepbergenov I. T., Prof., Almaty, Kazakhstan
Zheleznyak V. K., Prof., Novopolotsk, Belarus	Filipchenko S. A., As. Prof., Moscow, Russia
Zaborovsky V. S., Prof., St. Petersburg, Russia	Fozilov Sh. Kh., Prof., Tashkent, Uzbekistan
Zegzhda P. D., Prof., St. Petersburg, Russia	Fu-Nian Hu, Prof., Jiangsu, China
Kanaev A. K., Prof., St. Petersburg, Russia	Khabarov V. I., Prof., Novosibirsk, Russia
Kotenko A. G., Dr. Sc., As. Prof., St. Petersburg, Russia	Khodakovskiy V. A., Prof., St. Petersburg, Russia
Kurenkov P. V., Prof., Moscow, Russia	Chekhnin K. A., Prof., Khabarovsk, Russia
Letzky E. C., Prof., Moscow, Russia	Jalyshev Y. I., Prof., Ekaterinburg, Russia

Editorial address:

190031, St. Petersburg, Moskovsky ave., 9, aud. 1–210
e-mail: itt-pgups@yandex.ru

The journal is registered by the Federal Service for Supervision of Communications and Mass Media,
EL no. FS77-61707 testimony from May 7, 2015.

The journal is registered in the Russian Science Citation Index (RSCI).

Frequency of release - 4 issues per year. Issues of the magazine are available at <http://itt-pgups.ru>.

Copies of the archives with the issues of the journal are state-registered as an electronic publication of network distribution
in the Scientific and Technical Center "Informregister".

The content is for children over the age of 12.

Содержание

Куренков П. В.

Моделирование топологии взаимодействия элементов и потоков различных типов в системе
внешнеторговых перевозок 5

Баданина О. В., Гиндин С. И., Хомоненко А. Д.

Оценка оперативности передачи больших данных на примере базы данных PostgreSQL,
платформы Hadoop и системы Sqoop 18

Кикоть А. В.

Сравнение моделей и методов интеллектуальной обработки геопространственных данных
на основе семантической технологии 28

Красновидов А. В., Логинов П. А.

Об определении названия функции по абстрактному синтаксическому дереву с помощью
нейронной сети..... 36

Хабаров Р. С., Лохвицкий В. А., Дудкин А. С.

Аппроксимация времени пребывания для системы массового обслуживания Fork-Join
на основе инвариантов отношения 46

Мальцев Н. С., Молодкин И. А., Гильванов Р. Г.

Сравнительный анализ скорости вычислений на центральном и графическом процессоре 51

Султонов Ш. Х., Крицкий Н. А., Султонова З. Р.

Структура управляющей программы и способ для обнаружения обледенения на поверхности
стрелочных переводов 59

Сергеев Р. А., Бубнов В. П.

Применение нейронных сетей для определения физического состояния человека 65

Кудряшов А. Н.

Анализ целевых возможностей наземного комплекса управления космическими аппаратами
на основе сетей Петри и экспертного оценивания 71

Брызгалов А. А.

Исследование современных анализаторов текстовой информации 78

Contents

<i>Kurenkov P. V.</i> Modeling the Topology of the Interaction of Elements and Flows of Various Types in the System of Foreign Trade Transportation	5
<i>Badanina O. V., Gindin S. I., Khomonenko A. D.</i> Estimation of Efficiency of Transfer of Big Data on the Example of PostgreSQL Database, Hadoop Platform and Sqoop System	18
<i>Kikot A. V.</i> Comparison of Models and Methods for Geospatial Data Intelligent Processing Based on Semantic Technology	28
<i>Krasnovidov A. V., Loginov P. A.</i> About Determining a Function Name by Abstract Syntactic Tree Using Neural Network	36
<i>Khabarov R. S., Lokhvitsky V. A., Dudkin A. S.</i> Sojourn Time Approximation for Fork-Join Queue Based on Relationship Invariants	46
<i>Maltsev N. S., Molodkin I. A., Gilvanov R. G.</i> Comparative Analysis of Computing Speed on Central and Graphic Processor	51
<i>Sultonov Sh. Kh., Kritsky N. A., Sultonova Z. R.</i> Structure of the Control Program and Method for Detecting Icing on the Surface of Switches	59
<i>Sergeev R. A., Bubnov V. P.</i> The Use of Neural Networks to Determine the Physical Condition of a Person	65
<i>Kudryashov A. N.</i> Analysis of the Target Capabilities of the Ground-Based Spacecraft Control System Based on Petri Nets and Expert Evaluation	71
<i>Bryzgalov A. A.</i> Research of Modern Text Information Analyzers	78

Моделирование топологии взаимодействия элементов и потоков различных типов в системе внешнеторговых перевозок

Д.э.н., к.т.н. П. В. Куренков
 Российский университет транспорта (МИИТ)
 Москва, Россия
 petrkurenkov@mail.ru

Аннотация. Рассматриваются связи потоков различных типов (транспортных, грузовых, информационных, финансовых, энергетических, правовых и других) между собой и элементами в системе доставки внешнеторговых грузов. Приводится модель топологии взаимодействия потоков с определением интегрального показателя качества того или иного маршрута или варианта доставки. Приведен пример расчета.

Ключевые слова: элемент, поток, система, взаимодействие, доставка, вагон, груз, мультиплекс, комплекс.

ВВЕДЕНИЕ

В работе [1] отмечается, что оптимизация управления потоками создаст более благоприятные условия для ритмичной работы и согласования перевозок с другими видами транспорта, а правильно рассчитанный **подвод грузов к портам** и пограничным переходам даст возможность экономического использования подвижного состава.

Решение этой проблемы на новом уровне стало возможным в связи с повсеместным внедрением цифровых информационных технологий [2–9].

Впервые **транспортные, грузовые, информационные, финансовые и энергетические потоки** в едином комплексе рассмотрены в работе Белого О. В. [10].

Функцию стыкования **транспортных, грузовых, информационных и финансовых потоков с энергетическими** выполняют энергодиспетчеры дирекций железных дорог, энергоучастков, электростанций, тяговых подстанций и т. д.

На основании изложенного предлагается иерархическая схема пунктов взаимодействия **потоков** различных типов, включающая диспетчерские, информационные, ситуационные и другие центры управления грузовыми перевозками в системе смешанных сообщений (рис. 1).

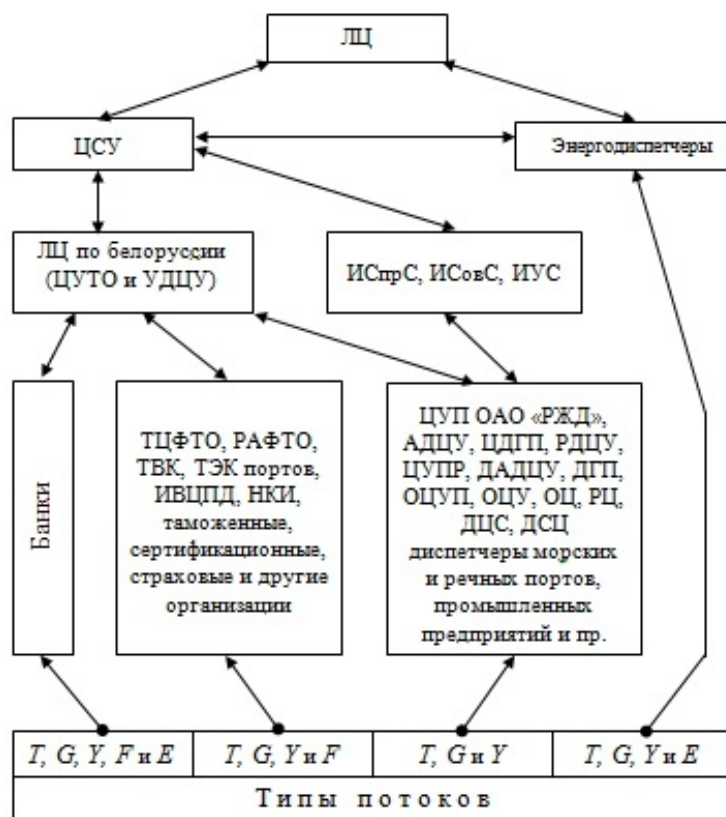


Рис. 1. Иерархическая схема пунктов взаимодействия потоков различных типов

ПРЕДСТАВЛЕНИЕ СТРУКТУРЫ СДВТГ В ВИДЕ

СОВОКУПНОСТИ МУЛЬТИПЛИЦИАЛЬНЫХ КОМПЛЕКСОВ

Всю структуру системы доставки внешнеторговых грузов (СДВТГ) можно представить как совокупность **элементов** $X = \{ \text{станция (порт) погрузки, грузоотправитель, грузовладелец, железнодорожные перегоны, станции расформирования или формирования составов поездов, станция (порт) выгрузки, грузополучатель, экспедитор и т. д.} \}$ и совокупность **потоков** (транспортных — T , грузовых — G , информационных — Y , финансовых — F , энергетических — E и правовых связей — P), взаимодействующих между собой. В свою очередь каждый **элемент, поток и правовую связь** можно категорировать — отнести к той или иной категории.

Каждому **элементу** или подсистеме СДВТГ должен быть поставлен в соответствие динамический массив, состоящий из постоянных и переменных параметров текущего, прогнозного и архивного состояний. Параметрами являются не только нормативно-справочная информация Правил перевозок грузов, Тарифных руководств и различных ГОСТов, но и функциональные свойства **элементов**, определяемые паспортами клиентов, техническо-распорядительными актами работы станций, планом формирования поездов, сводами обычаев морских торговых портов, уставами предприятий, лицензиями, сертификатами и т. д.

Каждый параметр, характеризующий **элементы** (производительность перегрузочного оборудования, складская емкость, длина перегона, наличие лицензий или права оказания различных видов услуг в области транспортного бизнеса, порядок обработки на станциях прибывающих и отправляемых составов поездов, порядок подачи/уборки вагонов в/из порта, право на оплату тарифа по Прейскуранту 10-01 и по ставкам Тарифной политики, банк, в котором находится расчетный счет субъекта РТУ, базисные условия внешнеторговых контрактов и т. д.), имеет свое функциональное назначение и оказывает влияние как на состояние и функционирование отдельно взятого **элемента** или подсистемы СДВТГ, так и на порядок взаимодействия **элементов с потоками** различных типов и категорий, **потоков** различных типов между собой, **элементов и потоков с правовыми связями** — то есть на состояние и функционирование СДВТГ в целом.

Другими словами, одно из важнейших (качественных) свойств СДВТГ заключено главным образом в силе **связности** ее **элементов, транспортных, грузовых, информационных, финансовых и энергетических потоков** при определенной технологии и **правовой базе** их взаимодействия или, иначе, это есть то **топологическое пространство состояния**, в котором заключено качественное свойство СДВТГ. Оно представляет собой пространство, которое можно назвать пространством **структуры элементов, структур всех типов потоков и структуры правовых связей** различных категорий, так как именно структура взаимосвязи **элементов, потоков и правовой базы** или же отдельных подсистем СДВТГ определяющим образом влияет на ее функционирование. С изменением (исчезновением или ослаблением) связности между отдельными **элементами** (закрытие линий, стихийные бедствия, войны, межгосударственные

конфликты, арест расчетного счета, расторжение договоров купли-продажи, договоров на экспедирование, фрахтование, декларирование, лишение лицензий, отключение электричества и т. д.) происходит либо исчезновение самой СДВТГ, либо изменение ее функциональных свойств (качества).

Информацию о качественном состоянии СДВТГ в статике может дать ее описание в терминах **отношений инцидентности**.

В общем случае, насыщенный линейный план в k -мерном пространстве представляет собой совокупность из $M+1$ точек, размещенных в этом пространстве таким образом, что они не принадлежат одновременно никакому подпространству размерности меньше, чем M . Такие точки задают вершины геометрической фигуры, которая называется **k -мерным симплексом** [11–15]. Фигуры, составленные из симплексов (так называемые полиэдры), рассматривал еще Анри Пуанкаре при построении теории **гомологии** — одного из разделов **топологии** [16, 17].

Понятие «симплекс» формулируется следующим образом: **симплекс** — это выпуклая оболочка линейно независимых точек в евклидовом пространстве или гомеоморфный образ; указанные точки называются вершинами симплекса, а уменьшенное на единицу их число — **размерностью симплекса**. Всякое подмножество вершин симплекса также определяет симплекс — грань исходного симплекса. Правильное примыкание симплексов в полиэдре означает, что симплексы могут пересекаться только по их общей грани.

Однако, согласно компьютерной терминологии, существует и отличное понятие. В [18, с. 438] сказано, что «между вершинами симплекса данные могут перемещаться только в одном направлении и невозможно движение **потока** данных в противоположном направлении». При этом имеется в виду только **информационный поток**. В СДВТГ не только **информационный**, но и все остальные типы **потоков** перемещаются в обоих направлениях (обмен информацией, возврат неправильно начисленных денежных сумм, возвращение подвижного состава к своему владельцу, возврат груза на станцию отправления вследствие коммерческой неисправности и др.), поэтому данный термин для моделирования соответствующих процессов неприемлем. Термин «**дуплекс**» (*duplex*), характеризующий возможность одновременной передачи данных (движения **информационного потока**) в обоих направлениях [18, с. 157], также является не совсем подходящим, поскольку между одними и теми же **элементами** СДВТГ и субъектами РТУ (ж.-д. станциями, портами, экспедиторами и т. д.) перемещаются **потоки** различных типов. Наиболее подходящим для формализации взаимодействий между **элементами, потоками и правовыми связями** в СДВТГ представляется термин «**мультиплекс**» (*multiplex*), характеризующий перемещение всех типов **потоков** в любых направлениях, их взаимодействие с **элементами, правовыми связями** и между собой [19–21].

M -мерный мультиплекс определяется своими $M+1$ вершинами $t_1, t_2, \dots, t_n, \dots, t_i, g_1, g_2, \dots, g_b, \dots, g_s; y_1, y_2, \dots, y_j, \dots, y_m; f_1, f_2, \dots, f_c, \dots, f_h; p_1, p_2, \dots, p_d, \dots, p_v$ и т. д., которые являются точками общего положения. Это означает, что все они не лежат ни в одной из $(M-1)$ -мерной гиперплоскости.

Для моделирования взаимодействия элементов x_i , потоков t_a, g_b, y_j, f_c, e_w и правовых связей p_d ($x \in X, t \in T, g \in G, y \in Y, f \in F, e \in E, p \in P$) на произведениях X и T, X и G, X и Y, X и F, X и E, X и P , а также всех остальных введены отношения $\lambda_i^x \in X \times T, \lambda_g^x \in X \times G, \lambda_y^x \in X \times Y, \lambda_f^x \in X \times F, \lambda_e^x \in X \times E, \lambda_p^x \in X \times P$ и т. д., которые существуют между множествами X и T, X и G, X и Y, X и F, X и E, X и P и т. д. тогда и только тогда, когда входящие в их состав элементы, потоки и правовые связи взаимодействуют между собой. Причем, если взаимодействие имеет место, то $\lambda \geq 1$ (значения λ принимаются в зависимости от принятой категории элемента, потока и правовой связи, в противном случае — равны 0). Данные отношения можно представить матрицами инцидентности $\Lambda_i^x=(\lambda_{ia}), \Lambda_g^x=(\lambda_{ib}), \Lambda_y^x=(\lambda_{ij}), \Lambda_f^x=(\lambda_{ic}), \Lambda_e^x=(\lambda_{iw}), \Lambda_p^x=(\lambda_{id})$ и т. д., в которых

$$\begin{aligned} \lambda_{ia} &= 1, 2 \dots N_t, \text{ если } (x_i, t_a) \in \lambda_i^x; \lambda_{ia} = 0, \text{ если } (x_i, t_a) \notin \lambda_i^x; \\ \lambda_{ib} &= 1, 2 \dots N_g, \text{ если } (x_i, g_b) \in \lambda_g^x; \lambda_{ib} = 0, \text{ если } (x_i, g_b) \notin \lambda_g^x; \\ \lambda_{ij} &= 1, 2 \dots N_y, \text{ если } (x_i, y_j) \in \lambda_y^x; \lambda_{ij} = 0, \text{ если } (x_i, y_j) \notin \lambda_y^x; \\ \lambda_{ic} &= 1, 2 \dots N_f, \text{ если } (x_i, f_c) \in \lambda_f^x; \lambda_{ic} = 0, \text{ если } (x_i, f_c) \notin \lambda_f^x; \\ \lambda_{iw} &= 1, 2 \dots N_e, \text{ если } (x_i, e_w) \in \lambda_e^x; \lambda_{iw} = 0, \text{ если } (x_i, e_w) \notin \lambda_e^x; \\ \lambda_{id} &= 1, 2 \dots N_p, \text{ если } (x_i, p_d) \in \lambda_p^x; \lambda_{id} = 0, \text{ если } (x_i, p_d) \notin \lambda_p^x \text{ и т. д.,} \end{aligned} \quad (1)$$

где $N_x, N_t, N_g, N_y, N_f, N_e$ и N_p — количество категорий, на которые делятся элементы, потоки и правовые связи.

С геометрической точки зрения отношения $\lambda_i^x, \lambda_g^x, \lambda_y^x, \lambda_f^x, \lambda_e^x, \lambda_p^x$ и т. д. согласно [11, 12, 14, 22, 23] определяют симплицальные комплексы, обозначенные через $K_x(Y; \lambda)$. Назовем их мультиплициальными комплексами и обозначим $M_x^t(T; \lambda_i^x), M_x^g(G; \lambda_g^x), M_x^y(Y; \lambda_y^x), M_x^f(F; \lambda_f^x), M_x^e(E; \lambda_e^x), M_x^p(P; \lambda_p^x)$ и т. д.

Под каждым мультиплексом подразумевается взаимодействие элементов, потоков различных типов и правовых связей между собой. Например: «грузовая отправка ↔ судно», «транзитный вагон с переработкой ↔ сортировочная станция», «таможенные платежи ↔ банк», «штурманская расписка ↔ ТЭК», «документ, подтверждающий экспорт товара ↔ возврат НДС», «договор на оказание информационных услуг ↔ дислокация вагона или груза», «энергетический поток ↔ железнодорожный регион», «комплект перевозочных документов ↔ товарная контора», «заявка на грузовую перевозку ↔ ТЦФТО» и т. д. Совокупность таких мультиплексов составляет структуру всей СДВТГ.

В таблице 1 приведены обозначения отношений и матриц инцидентности, а также мультиплициальных комплексов для соответствующих произведений структур элементов, потоков и правовых связей.

Таблица 1

Обозначения мультиплициальных комплексов, отношений и матриц инцидентности

Прямые произведения	Отношения инцидентности	Матрицы инцидентный	Мультиплициальные комплексы
X и T	$\lambda_{ix} \in X \times T$	$\Lambda_i^x = (\lambda_{ia})$	$M_x^t(T; \lambda_i^x)$
X и G	$\lambda_{gx} \in X \times G$	$\Lambda_g^x = (\lambda_{ib})$	$M_x^g(G; \lambda_g^x)$
X и Y	$\lambda_{yx} \in X \times Y$	$\Lambda_y^x = (\lambda_{ij})$	$M_x^y(Y; \lambda_y^x)$
X и F	$\lambda_{fx} \in X \times F$	$\Lambda_f^x = (\lambda_{ic})$	$M_x^f(F; \lambda_f^x)$
T и G	$\lambda_{gt} \in T \times G$	$\Lambda_g^t = (\lambda_{ab})$	$M^g(G; \lambda_g^t)$
T и Y	$\lambda_{yt} \in T \times Y$	$\Lambda_y^t = (\lambda_{aj})$	$M^y(Y; \lambda_y^t)$
T и F	$\lambda_{ft} \in T \times F$	$\Lambda_f^t = (\lambda_{ac})$	$M^f(F; \lambda_f^t)$

Мультиплициальные комплексы, представляющие в совокупности СДВТГ, определяются следующим образом:

1. $M_x^t(T; \lambda_i^x), M_x^g(G; \lambda_g^x), M_x^y(Y; \lambda_y^x), M_x^f(F; \lambda_f^x), M_x^e(E; \lambda_e^x), M_x^p(P; \lambda_p^x)$ и др. в свою очередь являются соответственно совокупностями мультиплексов $\sigma_s^{ix}, s=0, 1, \dots, N_t^x; \sigma_s^{gx}, s=0, 1, \dots, N_g^x; \sigma_s^{yx}, s=0, 1, \dots, N_y^x; \sigma_s^{fx}, s=0, 1, \dots, N_f^x; \sigma_s^{ex}, s=0, 1, \dots, N_e^x; \sigma_s^{px}, s=0, 1, \dots, N_p^x$ и т. д.

2. Каждые из мультиплексов $\sigma_s^{ix} \in M_x^t, \sigma_s^{gx} \in M_x^g, \sigma_s^{yx} \in M_x^y, \sigma_s^{fx} \in M_x^f, \sigma_s^{ex} \in M_x^e, \sigma_s^{px} \in M_x^p$ и т. д. определяются некоторыми подмножествами из $(s+1)$ различных t_a, g_b, y_j, f_c, e_w и p_d , для каждого из которых имеется, по крайней мере, одно $x_M \in X, t_M \in T, g_M \in G, y_M \in Y, f_M \in F, e_M \in E$ и $p_M \in P$ такие, что $(x_M, t_a) \in \lambda_i^x, (x_M, g_b) \in \lambda_g^x, (x_M, y_j) \in \lambda_y^x, (x_M, f_c) \in \lambda_f^x, (x_M, e_w) \in \lambda_e^x, (x_M, p_d) \in \lambda_p^x$ и т. д. для каждого из $(s+1)$ значений t_a, g_b, y_j, f_c, e_w и p_d .

3. Мультиплекс σ_0^a отождествляется с $t_a, a=1, \dots, l$, где l — число потоков множества T);

σ_0^b — с $g_b, b=1, \dots, z$ (z — число потоков множества G);

σ_0^j — с $y_j, j=1, \dots, m$ (m — число потоков множества Y);

σ_0^c — с $f_c, c=1, \dots, h$ (h — число потоков множества F);

σ_0^w — с $e_w, w=1, \dots, u$ (u — число потоков множества E);

σ_0^d — с $h_d, d=1, \dots, p$ (p — число правовых связей множества P).

4. Каждое подмножество мультиплексов $\sigma_s^{ix}, \sigma_s^{gx}, \sigma_s^{yx}, \sigma_s^{fx}, \sigma_s^{ex}, \sigma_s^{px}$ и т. д., определяемое их $q^{ix}+1, q^{gx}+1, q^{yx}+1, q^{fx}+1, q^{ex}+1, q^{px}+1$ и т. д. вершинами ($q^{ix} < s^{ix}, q^{gx} < s^{gx}, q^{yx} < s^{yx}, q^{fx} < s^{fx}, q^{ex} < s^{ex}, q^{px} < s^{px}$ и т. д.), называются $q^{ix-}, q^{gx-}, q^{yx-}, q^{fx-}, q^{ex-}, q^{px-}$ -гранями соответствующих мультиплексов и образуют в свою очередь подмножества $\sigma_q^{ix} \in M_x^t, \sigma_q^{gx} \in M_x^g, \sigma_q^{yx} \in M_x^y, \sigma_q^{fx} \in M_x^f, \sigma_q^{ex} \in M_x^e, \sigma_q^{px} \in M_x^p$ ($\sigma_q^{ix} < \sigma_s^{ix}, \sigma_q^{gx} < \sigma_s^{gx}, \sigma_q^{yx} < \sigma_s^{yx}, \sigma_q^{fx} < \sigma_s^{fx}, \sigma_q^{ex} < \sigma_s^{ex}, \sigma_q^{px} < \sigma_s^{px}$).

Числа $N_i^x, N_g^x, N_y^x, N_f^x, N_e^x$ и N_p^x определяют размерность мультиплициальных комплексов $M_i^x(N_i^x = \dim M_i^x), M_g^x(N_g^x = \dim M_g^x), M_y^x(N_y^x = \dim M_y^x), M_f^x(N_f^x = \dim M_f^x), M_e^x(N_e^x = \dim M_e^x)$ и $M_p^x(N_p^x = \dim M_p^x)$, означающих наибольшую размерность для соответствующих мультиплексов.

Множества T, G, Y, F, E и P являются множествами вершин мультиплициальных комплексов $M_x^t(T; \lambda_i^x), M_x^g(G; \lambda_g^x), M_x^y(Y; \lambda_y^x), M_x^f(F; \lambda_f^x), M_x^e(E; \lambda_e^x), M_x^p(P; \lambda_p^x)$ и т. д.

Другими словами, мультиплексы x_i (элементы СДВТГ) представляют собой выпуклые многогранники

с вершинами в евклидовых пространствах $E_t^x, E_g^x, E_y^x, E_f^x, E_e^x, E_p^x$ и т. д., а мультиплициальные комплексы $M_x^t(T; \lambda_t^x), M_x^g(G; \lambda_g^x), M_x^y(Y; \lambda_y^x), M_x^f(F; \lambda_f^x), M_x^e(E; \lambda_e^x), M_x^p(P; \lambda_p^x)$ и т. д. — совокупности таких многогранников в тех же пространствах, вершинами которых являются **потоки** различных типов и **правовые связи** (t_a, g_b, y_j, f_c, e_w и p_d). При такой постановке можно оценить *прямое влияние* структур **элементов, потоков** и **правовых связей** друг на друга.

Отношение λ^{-1} даст сопряженные мультиплициальные комплексы $M_t^x(X; \lambda_{tx}^{-1}), M_g^x(X; \lambda_{gx}^{-1}), M_y^x(X; \lambda_{yx}^{-1}), M_f^x(X; \lambda_{fx}^{-1}),$

$M_e^x(X; \lambda_{ex}^{-1}), M_p^x(X; \lambda_{px}^{-1})$ и т. д. Теперь матрицами инцидентности для $\lambda_{tx}^{-1}, \lambda_{gx}^{-1}, \lambda_{yx}^{-1}, \lambda_{fx}^{-1}, \lambda_{ex}^{-1}, \lambda_{px}^{-1}$ и т. д. являются матрицы $\Lambda_{tx}^T, \Lambda_{gx}^T, \Lambda_{yx}^T, \Lambda_{fx}^T, \Lambda_{ex}^T, \Lambda_{px}^T$ и т. д., полученные транспонированием $\Lambda_{tx}, \Lambda_{gx}, \Lambda_{yx}, \Lambda_{fx}, \Lambda_{ex}, \Lambda_{px}$ и т. д. В этом случае изучается *обратное влияние* структур **элементов, потоков** и **правовых связей** друг на друга.

В таблице 2 приведены обозначения отношений и транспонированных матриц инцидентностей, а также сопряженных мультиплициальных комплексов для соответствующих произведений **элементов, потоков** различных типов и **правовых связей**.

Таблица 2

Обозначения сопряженных мультиплициальных комплексов, отношений и матриц инцидентности

Обратные произведения	Отношения инцидентности	Матрицы инцидентностей	Сопряженные мультиплициальные комплексы
T и X	$\lambda_{tx}^{-1} \in T \times X$	$\Lambda_{tx}^T = (\lambda_{ai})$	$M_t^x(X; \lambda_{tx}^{-1})$
G и X	$\lambda_{gx}^{-1} \in G \times X$	$\Lambda_{gx}^T = (\lambda_{bi})$	$M_g^x(X; \lambda_{gx}^{-1})$
Y и X	$\lambda_{yx}^{-1} \in Y \times X$	$\Lambda_{yx}^T = (\lambda_{ji})$	$M_y^x(X; \lambda_{yx}^{-1})$
F и X	$\lambda_{fx}^{-1} \in F \times X$	$\Lambda_{fx}^T = (\lambda_{ci})$	$M_f^x(X; \lambda_{fx}^{-1})$
G и T	$\lambda_{gt}^{-1} \in G \times T$	$\Lambda_{gt}^T = (\lambda_{ba})$	$M_g^t(T; \lambda_{gt}^{-1})$
Y и T	$\lambda_{yt}^{-1} \in Y \times T$	$\Lambda_{yt}^T = (\lambda_{ja})$	$M_y^t(T; \lambda_{yt}^{-1})$
F и T	$\lambda_{ft}^{-1} \in F \times T$	$\Lambda_{ft}^T = (\lambda_{ca})$	$M_f^t(T; \lambda_{ft}^{-1})$

Таким образом, структура СДВТГ рассматривается как состоящая из 21 взаимодействующих комплексов (21 прямых и 21 обратных): комплексов **элементов**, комплексов **потоков** различных типов — **транспортных, грузовых, информационных, финансовых, энергетических** и комплексов **правовых связей**.

ОПРЕДЕЛЕНИЕ ВЗАИМОЗАВИСИМОСТИ РАЗЛИЧНЫХ ТИПОВ ЭЛЕМЕНТОВ, ПОТОКОВ, ПРАВОВЫХ СВЯЗЕЙ И СТЕПЕНИ СЛОЖНОСТИ СТРУКТУРЫ СДВТГ

Козволюционное взаимодействие, рассмотренное под углом зрения его организации, есть цепная связь конъюгирующих комплексов, для которой характерны асимметрия и наличие связи (по А. А. Богданову) [23]. Разрыв связи в козволюционном взаимодействии влечет за собой или дезорганизацию, или появление отдельных независимых комплексов. Эти идеи, выдвинутые Богдановым в начале XX века, относятся к исследованию формирующих организационных механизмов или механизмов формирования организационного взаимодействия (в нашем случае, если следовать В. Н. Образцову, В. В. Зонкову, А. В. Комарову, С. В. Милославской, В. А. Персианову и др., — теории комплексной эксплуатации разных видов транспорта, совмещенной технологии грузовых перевозок в смешанном сообщении, системному анализу структуры и технологии функционирования транспортных узлов), и не касаются механизмов изменения организационных структур или организационных форм управления системой доставки внешнеторговых грузов (СДВТГ). Рассмотрим некоторые особенности применения мультиплициальных комплексов для оценки структуры СДВТГ.

Многомерную связанную структуру мультиплициальных комплексов $M_x^t(T; \lambda_{tx})$ и $M_t^x(X; \lambda_{tx}^{-1}), M_x^g(G; \lambda_{gx})$ и $M_g^x(X; \lambda_{gx}^{-1}), M_x^y(Y; \lambda_{yx})$ и $M_y^x(X; \lambda_{yx}^{-1})$ и т. д. можно изучать различными способами с привлечением алгебраических методов. Для этого введено понятие **q-связности** [22]. Этот подход нашел прикладное применение в работах ученых

России (**Б. Б. Жардемова** [23]) и США (**Дж. Касти** [24] и **Р. Эткина** [25]).

Понятие **q-связности** относится к изучению таких связей в структуре комплексов $M_x^t(T; \lambda_{tx})$ и $M_t^x(X; \lambda_{tx}^{-1}), M_x^g(G; \lambda_{gx})$ и $M_g^x(X; \lambda_{gx}^{-1}), M_x^y(Y; \lambda_{yx})$ и $M_y^x(X; \lambda_{yx}^{-1}), M_x^f(F; \lambda_{fx})$ и $M_f^x(X; \lambda_{fx}^{-1}), M_x^e(E; \lambda_{ex})$ и $M_e^x(X; \lambda_{ex}^{-1}), M_x^p(P; \lambda_{px})$ и $M_p^x(X; \lambda_{px}^{-1})$ и т. д., при которых каждый мультиплекс имеет общую вершину с соседними мультиплексами при $q_i=0, 1, \dots, \dim M_i - 1; q_g=0, 1, \dots, \dim M_g - 1; q_y=0, 1, \dots, \dim M_y - 1; q_f=0, 1, \dots, \dim M_f - 1; q_e=0, 1, \dots, \dim M_e - 1$ и $q_p=0, 1, \dots, \dim M_p - 1$. Геометрически эти связи содержат информацию относительно того, каким образом взаимодействуют между собой различные типы **элементов, потоков** и **правовых связей** в СДВТГ. Другими словами, для представления мультиплексов, больших, чем $\dim M_t = N_t, \dim M_g = N_g, \dim M_y = N_y, \dim M_f = N_f, \dim M_e = N_e$ и $\dim M_p = N_p$ рассматриваемые комплексы $M_x^t(T; \lambda_{tx})$ и $M_t^x(X; \lambda_{tx}^{-1}), M_x^g(G; \lambda_{gx})$ и $M_g^x(X; \lambda_{gx}^{-1}), M_x^y(Y; \lambda_{yx})$ и $M_y^x(X; \lambda_{yx}^{-1}), M_x^f(F; \lambda_{fx})$ и $M_f^x(X; \lambda_{fx}^{-1}), M_x^e(E; \lambda_{ex})$ и $M_e^x(X; \lambda_{ex}^{-1}), M_x^p(P; \lambda_{px})$ и $M_p^x(X; \lambda_{px}^{-1})$ и все остальные распадаются на несвязанные мультиплексы на каждом уровне q_t, q_g, q_y, q_f, q_e и q_p .

На мультиплексах комплексов $M_t^x, M_g^x, M_y^x, M_f^x, M_e^x, M_p^x$ и т. д. можно ввести **отношения эквивалентностей** соответственно $\gamma_q^{tx}, \gamma_q^{gx}, \gamma_q^{yx}, \gamma_q^{fx}, \gamma_q^{ex}, \gamma_q^{px}$ и т. д., определяемые как $(\sigma_p^{tx}, \sigma_r^{tx}) \in \gamma_q^{tx}, (\sigma_p^{gx}, \sigma_r^{gx}) \in \gamma_q^{gx}, (\sigma_p^{yx}, \sigma_r^{yx}) \in \gamma_q^{yx}, (\sigma_p^{fx}, \sigma_r^{fx}) \in \gamma_q^{fx}, (\sigma_p^{ex}, \sigma_r^{ex}) \in \gamma_q^{ex}, (\sigma_p^{px}, \sigma_r^{px}) \in \gamma_q^{px}$ и т. д. тогда и только тогда, когда мультиплексы $\sigma_p^{tx}, \sigma_p^{gx}, \sigma_p^{yx}, \sigma_p^{fx}, \sigma_p^{ex}, \sigma_p^{px}$ и т. д. **q-связаны** соответственно с $\sigma_r^{tx}, \sigma_r^{gx}, \sigma_r^{yx}, \sigma_r^{fx}, \sigma_r^{ex}, \sigma_r^{px}$ и т. д. Множества всех классов отношений эквивалентностей $\gamma_q^{tx}, \gamma_q^{gx}, \gamma_q^{yx}, \gamma_q^{fx}, \gamma_q^{ex}, \gamma_q^{px}$ и т. д. образуют соответственно фактор-множества $M_t^x/\gamma_q^{tx}, M_g^x/\gamma_q^{gx}, M_y^x/\gamma_q^{yx}, M_f^x/\gamma_q^{fx}, M_e^x/\gamma_q^{ex}, M_p^x/\gamma_q^{px}$ и т. д., которые определяют разбиение пар мультиплициальных комплексов $M_x^t(T; \lambda_{tx})$ и $M_t^x(X; \lambda_{tx}^{-1}),$ а также всех остальных на попарно непересекающиеся подсистемы (разбиение структуры СДВТГ на отдельные

подсистемы элементов, потоков различных типов и правовых связей, а также пар потоков различных типов и т. д.), называемых **классами отношений эквивалентности** $Q_q^{ix}, Q_q^{gx}, Q_q^{yx}, Q_q^{fx}, Q_q^{ex}, Q_q^{px}$ и т. д.

Мультиплексы, связанные общей вершиной, у которых $q_x^i=0, q_x^g=0, q_x^y=0, q_x^f=0, q_x^e=0, q_x^p=0$ образуют соответственно множества мультиплексов классов $Q_0^i, Q_0^g, Q_0^y, Q_0^f,$

Q_0^e, Q_0^p . Мультиплексы, у которых $q_x^i=1, q_x^g=1, q_x^y=1, q_x^f=1, q_x^e=1, q_x^p=1$ — классов $Q_1^{ix}, Q_1^{gx}, Q_1^{yx}, Q_1^{fx}, Q_1^{ex}, Q_1^{px}$ и т. д.

В таблице 3 приведены обозначения граней мультиплексов, отношений эквивалентности и их классов, а также фактор-множеств для соответствующих мультиплициальных комплексов.

Таблица 3

Грани мультиплексов, классы отношений эквивалентностей, фактор-множества мультиплициальных комплексов

Мультиплициальные комплексы		Грани соответствующих мультиплексов	Отношения эквивалентностей	Фактор-множества	Классы отношений эквивалентностей
Прямые	Сопряженные				
$M_x^i(T; \lambda_i^x)$	$M_i^x(X; \lambda_{ix}^{-1})$	σ_a^x и σ_i^x	γ_q^{ix}	M_i^x/γ_q^{ix}	Q_q^{ix}
$M_x^g(G; \lambda_g^x)$	$M_g^x(X; \lambda_{gx}^{-1})$	σ_g^x и σ_g^x	γ_q^{gx}	M_g^x/γ_q^{gx}	Q_q^{gx}
$M_x^y(Y; \lambda_y^x)$	$M_y^x(X; \lambda_{yx}^{-1})$	σ^x и σ^y	γ_q^{yx}	M_y^x/γ_q^{yx}	Q_q^{yx}
$M_x^f(F; \lambda_f^x)$	$M_f^x(X; \lambda_{fx}^{-1})$	σ_e^x и σ_f^x	γ_q^{fx}	M_f^x/γ_q^{fx}	Q_q^{fx}
$M_x^e(G; \lambda_e^x)$	$M_g^i(T; \lambda_{gi}^{-1})$	σ_b^i и σ_a^g	γ_q^{gt}	M_g^i/γ_q^{gt}	Q_q^{gt}
$M_x^y(Y; \lambda_y^i)$	$M_y^i(T; \lambda_{yi}^{-1})$	σ^i и σ_a^y	γ_q^{yt}	M_y^i/γ_q^{yt}	Q_q^{yt}
$M_x^f(F; \lambda_f^i)$	$M_f^i(T; \lambda_{fi}^{-1})$	σ_e^i и σ_d^f	γ_q^{ft}	M_f^i/γ_q^{ft}	Q_q^{ft}

Необходимо отметить, что если два мультиплекса q -связаны, то они также $(q-1)$ -, $(q-2)$ -, ..., 0 -связаны в СДВТГ. Следовательно, сопряженные комплексы $M_x^i(T; \lambda_{ix})$ и $M_i^x(X; \lambda_{ix}^{-1})$, а также все остальные можно рассматривать как состоящие из подмножеств мультиплексов одной размерности, имеющих общие грани. Иначе говоря, для каждого значения размерностей $q_i^x = 0, 1, \dots, \dim M_i^x$; $q_g^x = 0, 1, \dots, \dim M_g^x$; $q_y^x = 0, 1, \dots, \dim M_y^x$; $q_f^x = 0, 1, \dots, \dim M_f^x$; $q_e^x = 0, 1, \dots, \dim M_e^x$ и $q_p^x = 0, 1, \dots, \dim M_p$ можно определить число различных классов эквивалентности $Q^x, Q_q^i, Q_q^g, Q_q^y, Q_q^f, Q_q^e$ и Q_q^p . Эта операция называется **q-анализом** структуры СДВТГ, а вектора

$$\begin{aligned}
 Q^{ix} &= \{Q^{ix}_{\dim M}, Q^{ix}_{\dim M-1}, \dots, Q^{ix}_1, Q^{ix}_0\}, \\
 Q^{gx} &= \{Q^{gx}_{\dim M}, Q^{gx}_{\dim M-1}, \dots, Q^{gx}_1, Q^{gx}_0\}, \\
 Q^{yx} &= \{Q^{yx}_{\dim M}, Q^{yx}_{\dim M-1}, \dots, Q^{yx}_1, Q^{yx}_0\}, \\
 Q^{fx} &= \{Q^{fx}_{\dim M}, Q^{fx}_{\dim M-1}, \dots, Q^{fx}_1, Q^{fx}_0\}, \\
 Q^{ex} &= \{Q^{ex}_{\dim M}, Q^{ex}_{\dim M-1}, \dots, Q^{ex}_1, Q^{ex}_0\}, \\
 Q^{px} &= \{Q^{px}_{\dim M}, Q^{px}_{\dim M-1}, \dots, Q^{px}_1, Q^{px}_0\} \text{ и т. д. —}
 \end{aligned}
 \tag{2}$$

структурными векторами, соответственно, элементов, потоков и правовых связей различных типов.

Если множества X и T содержат соответственно n элементов и l потоков, то матрице A_{ix} соответствует матрица размера $n \times l$. Аналогично матрице A_{gx} соответствует матрица размера $n \times d$, матрице A_{yx} — матрица размера $n \times t$, матрице A_{fx} — матрица размера $n \times h$, матрице A_{ex} — матрица размера $n \times u$, матрице A_{px} — матрица размера $n \times v$ и т. д., состоящие из нулей и чисел, больших или равных единице.

В произведениях $A_{ix}A_{ix}^T, A_{gx}A_{gx}^T, A_{yx}A_{yx}^T, A_{fx}A_{fx}^T, A_{ex}A_{ex}^T, A_{px}A_{px}^T$ и т. д. числа, стоящие на местах $(i,a), (i,b), (i,j), (i,c), (i,w), (i,d)$ и т. д., являются скалярными произведениями соответственно строк i и столбцов a, b, j, c, w, d матриц $A_{ix}, A_{gx}, A_{yx}, A_{fx}, A_{ex}, A_{px}$ и т. д. Поэтому они равняются значениям $(q_x^i+1), (q_x^g+1), (q_x^y+1), (q_x^f+1), (q_x^e+1)$ и (q_x^p+1) , в которых $q_x^i, q_x^g, q_x^y, q_x^f, q_x^e$ и q_x^p являются размерностями общих граней мультиплексов σ_p^{ix} и $\sigma_i^{ix}, \sigma_p^{gx}$ и $\sigma_i^{gx}, \sigma_p^{yx}$ и $\sigma_i^{yx}, \sigma_p^{fx}$ и $\sigma_i^{fx}, \sigma_p^{ex}$ и $\sigma_i^{ex}, \sigma_p^{px}$ и σ_i^{px} и т. д.

Итак, алгоритм оценки связности элементов, потоков и правовых связей в структуре СДВТГ состоит

в следующем. Чтобы найти значения $q_x^i, q_x^g, q_x^y, q_x^f, q_x^e, q_x^p$ и т. д. для общих граней всех пар x_i -мультиплексов в мультиплициальных комплексах $M_x^i(T; \lambda_{ix}), M_x^g(G; \lambda_{gx}), M_x^y(Y; \lambda_{yx}), M_x^f(F; \lambda_{fx}), M_x^e(E; \lambda_{ex})$ и $M_x^p(P; \lambda_{px})$ необходимо:

– составить матрицы $A_{ix}A_{ix}^T, A_{gx}A_{gx}^T, A_{yx}A_{yx}^T, A_{fx}A_{fx}^T, A_{ex}A_{ex}^T, A_{px}A_{px}^T$ и т. д. соответственно размеров $n \times l, n \times d, n \times t, n \times h, n \times u, n \times v$ и т. д., обозначив их $\Omega_x^i, \Omega_x^g, \Omega_x^y, \Omega_x^f, \Omega_x^e, \Omega_x^p$ и т. д.;

– оценить $A_{ix}A_{ix}^T - \Omega_x^i, A_{gx}A_{gx}^T - \Omega_x^g, A_{yx}A_{yx}^T - \Omega_x^y, A_{fx}A_{fx}^T - \Omega_x^f, A_{ex}A_{ex}^T - \Omega_x^e, A_{px}A_{px}^T - \Omega_x^p$ и т. д.,

где $\Omega_x^i = [\omega_{ia}], \Omega_x^g = [\omega_{gb}], \Omega_x^y = [\omega_{yj}], \Omega_x^f = [\omega_{fc}], \Omega_x^e = [\omega_{ew}], \Omega_x^p = [\omega_{pd}]$ и т. д.,

а $\omega_{ia} \geq 1$ для $i, a = 1, 2, \dots, n$;

$\omega_{ib} \geq 1$ для $i, b = 1, 2, \dots, n$;

$\omega_{ij} \geq 1$ для $i, j = 1, 2, \dots, n$;

$\omega_{ic} \geq 1$ для $i, c = 1, 2, \dots, n$;

$\omega_{iw} \geq 1$ для $i, w = 1, 2, \dots, n$;

$\omega_{id} \geq 1$ для $i, d = 1, 2, \dots, n$ и т. д.

Матрицы $\Omega_x^i, \Omega_x^g, \Omega_x^y, \Omega_x^f, \Omega_x^e, \Omega_x^p$ и др. называются **частотными матрицами отношений**, симметричными относительно главной диагонали. Так что $(\forall i, a)(\omega_{ia} = \omega_{ai}), (\forall i, b)(\omega_{ib} = \omega_{bi}), (\forall i, j)(\omega_{ij} = \omega_{ji}), (\forall i, c)(\omega_{ic} = \omega_{ci}), (\forall i, w)(\omega_{iw} = \omega_{wi}), (\forall i, d)(\omega_{id} = \omega_{di})$ и т. д.

Анализ для $M_x^i(X; \lambda_{ix}^{-1}), M_x^g(X; \lambda_{gx}^{-1}), M_x^y(X; \lambda_{yx}^{-1}), M_x^f(X; \lambda_{fx}^{-1}), M_x^e(X; \lambda_{ex}^{-1}), M_x^p(X; \lambda_{px}^{-1})$ и т. д. проводится путем составления матриц $A_{ix}A_{ix}^T - \Omega_x^i, A_{gx}A_{gx}^T - \Omega_x^g, A_{yx}A_{yx}^T - \Omega_x^y, A_{fx}A_{fx}^T - \Omega_x^f, A_{ex}A_{ex}^T - \Omega_x^e, A_{px}A_{px}^T - \Omega_x^p$ и т. д., в которых $\Omega_x^i, \Omega_x^g, \Omega_x^y, \Omega_x^f, \Omega_x^e, \Omega_x^p$ и т. д. являются матрицами, соответственно, размеров $l \times l, d \times d, t \times t, h \times h, u \times u, v \times v$ и т. д., состоящими из чисел, больших или равных единице.

Для оценки качественного (топологического) свойства структуры СДВТГ введен показатель «мера сложности» структуры — $\Psi_{\text{СДВТГ}}(M)$. Меры сложности взаимодействия структур элементов $\Psi_x(M)$, потоков различных типов $\Psi_i(M), \Psi_g(M), \Psi_y(M), \Psi_f(M), \Psi_e(M)$ и правовых связей $\Psi_p(M)$ определяются через координаты соответствующих векторов $Q^x, Q^i, Q^g, Q^y, Q^f, Q^e$ и Q^p по формулам, приведенным в таблице 4.

Определение меры сложности взаимодействия структур элементов, потоков и правовых связей в СДВТГ

Мера сложности структур элементов, потоков и правовых связей при их влиянии друг на друга	Мера сложности взаимодействия структур
$\Psi_x^i(M_x^i) = 2\{\sum(i+1)*Q_i^{ix}\} / [(N_x^i+1)(N_x^i+2)]$	$\Psi_{ix}(M) = \sqrt{\Psi_x^i(M)^2 + \Psi_i^x(M)^2}$
$\Psi_i^x(M_i^x) = 2\{\sum(a+1)*Q_a^{ix}\} / [(N_i^x+1)(N_i^x+2)]$	
$\Psi_x^g(M_x^g) = 2\{\sum(i+1)*Q_i^{gx}\} / [(N_x^g+1)(N_x^g+2)]$	$\Psi_{gx}(M) = \sqrt{\Psi_x^g(M)^2 + \Psi_g^x(M)^2}$
$\Psi_g^x(M_g^x) = 2\{\sum(b+1)*Q_b^{gx}\} / [(N_g^x+1)(N_g^x+2)]$	
$\Psi_x^y(M_x^y) = 2\{\sum(i+1)*Q_i^{yx}\} / [(N_x^y+1)(N_x^y+2)]$	$\Psi_{yx}(M) = \sqrt{\Psi_x^y(M)^2 + \Psi_y^x(M)^2}$
$\Psi_y^x(M_y^x) = 2\{\sum(j+1)*Q_j^{yx}\} / [(N_y^x+1)(N_y^x+2)]$	
$\Psi_f^c(M_f^c) = 2\{\sum(c+1)*Q_c^{fx}\} / [(N_f^c+1)(N_f^c+2)]$	$\Psi_{fx}(M) = \sqrt{\Psi_f^c(M)^2 + \Psi_c^f(M)^2}$
$\Psi_c^f(M_c^f) = 2\{\sum(i+1)*Q_i^{cf}\} / [(N_c^f+1)(N_c^f+2)]$	
$\Psi_g^e(M_g^e) = 2\{\sum(b+1)*Q_b^{eg}\} / [(N_g^e+1)(N_g^e+2)]$	$\Psi_{ge}(M) = \sqrt{\Psi_g^e(M)^2 + \Psi_e^g(M)^2}$
$\Psi_e^g(M_e^g) = 2\{\sum(a+1)*Q_a^{eg}\} / [(N_e^g+1)(N_e^g+2)]$	
$\Psi_y^j(M_y^j) = 2\{\sum(j+1)*Q_j^{yx}\} / [(N_y^j+1)(N_y^j+2)]$	$\Psi_{yj}(M) = \sqrt{\Psi_y^j(M)^2 + \Psi_j^y(M)^2}$
$\Psi_j^y(M_j^y) = 2\{\sum(a+1)*Q_a^{jy}\} / [(N_j^y+1)(N_j^y+2)]$	
$\Psi_f^d(M_f^d) = 2\{\sum(c+1)*Q_c^{fd}\} / [(N_f^d+1)(N_f^d+2)]$	$\Psi_{fd}(M) = \sqrt{\Psi_f^d(M)^2 + \Psi_d^f(M)^2}$
$\Psi_d^f(M_d^f) = 2\{\sum(a+1)*Q_a^{fd}\} / [(N_d^f+1)(N_d^f+2)]$	

Здесь $N_x, N_i, N_g, N_y, N_f, N_e, N_p$ и т. д. — размерности мультиплициальных комплексов $M_x, M_i, M_g, M_y, M_f, M_e, M_p$ и т. д., равные наибольшим значениям $q^{ix}, q^{gx}, q^{yx}, q^{fx}, q^{ex}$ и q^{px} ($N_i = \dim M_i, N_g = \dim M_g, N_y = \dim M_y, N_f = \dim M_f, N_e = \dim M_e$ и $N_p = \dim M_p$);

$Q_i^x, Q_a^i, Q_b^g, Q_j^y, Q_c^f, Q_w^e$ и Q_d^p — соответственно i -я, a -я, b -я, j -я, c -я, w -я и d -я компоненты, векторов $Q^x, Q^i, Q^g, Q^y, Q^f, Q^e$ и Q^p , получаемые в результате q -анализа.

Таким образом, получаем разные векторы для структуры элементов Q^x , для структур потоков Q^i, Q^g, Q^y, Q^f, Q^e и для структуры правовых связей Q^p . Вычисляя меры сложности расположения в системе элементов $\Psi_x(M)$, потоков $\Psi_i(M), \Psi_g(M), \Psi_y(M), \Psi_f(M), \Psi_e(M)$ и правовых связей $\Psi_p(M)$ с учетом их взаимодействия, могут быть получены количественные характеристики качества структуры СДВТГ.

Длина вектора $\Psi_{СДВТГ}(M)$ от начала координат, который является полным вектором структуры СДВТГ, равна

$$\Psi_{СДВТГ} = \sqrt{\Psi_{ix}^2 + \Psi_{gx}^2 + \Psi_{yx}^2 + \Psi_{fx}^2 + \dots + \Psi_{fp}^2} \quad (4)$$

Несмотря на то, что q -анализ является довольно эффективным подходом при изучении степени связности СДВТГ, тем не менее он не дает необходимой информации о том, насколько важен (интегрирован) каждый мультиплекс в общей структуре СДВТГ. Для учета индивидуальных свойств мультиплексов необходимо знать численное значение степени интегрированности каждого отдельного мультиплекса в структуре всей СДВТГ. Относительная важность того или иного элемента, потока и правовой связи в общей структуре СДВТГ характеризуется их эксцентриситетами.

В связи с этим вводятся следующие обозначения:

\hat{g} — верхнее значение q для x_i или t_a, g_b, y_j, f_c, e_w и p_d , т. е. $q = \dim M$;

\check{g} — нижнее значение q для x_i или t_a, g_b, y_j, f_c, e_w и p_d .

То есть q равно наибольшему значению q , при котором $x_i, t_a, g_b, y_j, f_c, e_w$ и p_d имеет связность с каким-либо другим мультиплексом, соответственно, из комплексов $M_x^i(T; \lambda_{xi})$ и $M_i^x(X; \lambda_{ix}^{-1}), M_x^g(G; \lambda_{gx})$ и $M_g^x(X; \lambda_{gx}^{-1}), M_x^f(F; \lambda_{fx})$ и $M_f^x(X; \lambda_{fx}^{-1}), M_x^e(E; \lambda_{ex})$ и $M_e^x(X; \lambda_{ex}^{-1}), M_x^p(P; \lambda_{px})$ и $M_p^x(X; \lambda_{px}^{-1})$, а также всех остальных.

Тогда эксцентриситеты отдельно взятых мультиплексов определяются следующим образом:

$$\begin{aligned} Ecc(x_i, t_a) &= (\hat{g} - \check{g}) / (\check{g} + 1); \\ Ecc(x_i, g_b) &= (\hat{g} - \check{g}) / (\check{g} + 1); \\ Ecc(x_i, y_j) &= (\hat{g} - \check{g}) / (\check{g} + 1); \\ Ecc(x_i, f_c) &= (\hat{g} - \check{g}) / (\check{g} + 1); \\ Ecc(x_i, e_w) &= (\hat{g} - \check{g}) / (\check{g} + 1); \\ Ecc(x_i, p_d) &= (\hat{g} - \check{g}) / (\check{g} + 1) \text{ и т. д.} \end{aligned} \quad (5)$$

Разность $(\hat{g} - \check{g})$ является мерой необычности любого мультиплекса σ . При этом равенство $(\hat{g} - \check{g}) = 2$ считается более значимым, если $q = 1$, а не когда $q = 10$. Поэтому в качестве значимости эксцентриситетов будем использовать вышеприведенные отношения (2), а не абсолютную разность $(\hat{g} - \check{g})$. Значения эксцентриситетов различных мультиплексов позволяют оценить, насколько каждый элемент, поток или правовая связь интегрированы в общую структуру СДВТГ, то есть их значимость в ее функционировании.

В таблице 5 приведены обозначения структурных векторов, частотных матриц отношений, матричного анализа мультиплициальных комплексов и эксцентриситетов мультиплексов в этих комплексах.

Таблица 5

Структурные вектора, частотные матрицы отношений и эксцентриситеты мультиплексов в мультиплициальных комплексах

Структурные вектора	Частотные матрицы отношений	Матричный анализ комплексов	Эксцентриситеты мультиплексов
1	2	3	4
Q^{ix}	$\Omega_x^i (\forall i, a)(\omega_{ia} = \omega_{ai})$	$\Lambda_{ix} \Lambda_{ix}^T - \Omega_x^i$	$Ecc(x_i, t_a)$
Q^{gx}	$\Omega_x^g (\forall i, b)(\omega_{ib} = \omega_{bi})$	$\Lambda_{gx} \Lambda_{gx}^T - \Omega_x^g$	$Ecc(x_i, g_b)$

Таблица 5 (продолжение)

1	2	3	4
Q^x	$\Omega_x^y (\forall i, j)(\omega_{ij} = \omega_{ji})$	$A_{yx}A_{yx}^T - \Omega_x^y$	$Ecc(x_i, y_j)$
Q^c	$\Omega_c^f (\forall i, c)(\omega_{ic} = \omega_{ci})$	$A_{fc}A_{fc}^T - \Omega_c^f$	$Ecc(x_i, f_c)$
Q^{gt}	$\Omega^{gs} (\forall a, b)(\omega_{ab} = \omega_{ba})$	$A_{tg}A_{tg}^T - \Omega^{gs}$	$Ecc(t_a, g_b)$
Q^{jt}	$\Omega_j^y (\forall a, j)(\omega_{aj} = \omega_{ja})$	$A_{jy}A_{jy}^T - \Omega_j^y$	$Ecc(t_a, y_j)$
Q^f	$\Omega_f^c (\forall a, c)(\omega_{ac} = \omega_{ca})$	$A_{cf}A_{cf}^T - \Omega_f^c$	$Ecc(t_a, f_c)$

ВЫБОР СХЕМЫ ПОСТАВКИ ВНЕШНЕТОРГОВОГО ГРУЗА

В качестве примера можно рассмотреть различные варианты маршрутов экспортно-импортных перевозок через морские порты и через сухопутные пограничные переходы. При этом могут иметь место варианты схем взаимного расположения звеньев в цепи транспортировки через морские (речные) порты России, СНГ и Балтии.

Алгоритм состоит в следующем. Составляются матрицы инцидентий (взаимодействий) A между элементами, потоками и правовыми связями, описывающие структуру СДВТГ, транспонированные матрицы инцидентий A^T — для изучения обратного влияния взаимодействующих элементов, потоков и правовых связей друг на друга, матрицы $AA^T - \Omega$ для анализа структур элементов, потоков и правовых связей, производится q -анализ комбинациями номеров строк и столбцов матриц Ω и q -анализ матриц $AA^T - \Omega$.

Целые числа на диагоналях пространственных матриц Ω являются размерностями мультиплексов. Наибольшие по величине числа показывают размерности комплексов M .

Используя результаты анализа, полученные в ходе исследования различных вариантов доставки грузов через различные порты СНГ, России и Балтии, выбирается рациональный, оценив на сложность структуру СДВТГ для каждого из них.

Количество элементов, потоков и правовых связей множеств X, T, G, Y, F, E и P может быть различным для каждого из исследуемых вариантов (маршрутов или схем) доставки.

Для оценки качества структуры СДВТГ, а в частном случае — для выбора схемы поставки внешнеторговых грузов — количество принимаемых во внимание факторов полностью зависит от требований заказчика на данную услугу и может быть любым, вплоть до простейшего варианта, когда во внимание принимается только общая стоимость доставки, включающая железнодорожный тариф, морской фрахт и различные сборы в пунктах отправления, назначения и перевалки с одного вида транспорта на другой.

Мера сложности структуры СДВТГ может определяться не только для схем поставок, определения степени взаимодействия различных типов элементов и потоков, но и для субъектов, оказывающих информационные, таможенные, экспедиторские, сюрвейерские, складские, стивидорные и другие услуги.

Ранжирование элементов, транспортных, грузовых, информационных, финансовых, энергетических и правовых потоков по категориям может выполняться как исходя из существующих разработок ученых и практиков, так и в зависимости от мнения субъекта, по заказу которого данные расчеты производятся. При отсутствии необходимости ранжирование может вообще не производиться. Аналогичным образом может осуществляться дифференцирование перевозчиков, экспедиторов, декларантов и других субъектов, получивших лицензии на различные виды

деятельности (оказание различных услуг) в области транспортно-бизнеса — собственным уставом, выданным установленным порядком в ОАО «РЖД» или в Минтрансе РФ сертификатам или же в зависимости от мнения самих грузовладельцев.

Предлагаемый алгоритм позволяет, во-первых, произвести оценку и анализ структуры СДВТГ одним интегральным показателем качества, во-вторых, определить степень интегрированности симплексов, дуплексов, мультиплексов, элементов, потоков, подсистем и отдельно взятых структур в общую структуру СДВТГ и, в-третьих, сгруппировать отдельные категории симплексов, дуплексов, мультиплексов, элементов, потоков, подсистем, отдельно взятых структур, маршрутов или схем поставок в СДВТГ в зависимости от сложности или отдельных критериев и по ранжиру качества выбрать наиболее подходящие для пользователя транспортными услугами.

МОДЕЛИРОВАНИЕ ЭКСПОРТА УГЛЯ ЧЕРЕЗ МОРСКИЕ ПОРТЫ

Экспортная работа с углем проводится в настоящее время в следующих портах:

- Российской Федерации:
 - Автово, Высоцк, Выборг, Мурманск Октябрьской ж. д.;
 - Туапсе, Ейск, Таганрог Северо-Кавказской ж. д.;
 - Восточный (ст. Находка-Восточная) Дальневосточной ж. д.;
- Украины:
 - порт Южный (ст. Береговая) Одесской ж. д.;
- стран Балтии:
 - Таллинн, Рига, Вентспилс.

Зарождение **транспортных и грузовых потоков** с экспортным углем происходит в угольных бассейнах Западно-Сибирской, Восточно-Сибирской, Дальневосточной, Красноярской ж. д. Часть экспортного угля поступает из северо-восточных районов КНР ввиду явной выгоды его перевозки по территории России и перевалки в морском порту Восточный.

Управление доставкой экспортного угля (как и вообще внешнеторговых грузов независимо от их номенклатуры) может быть эффективным только при наличии полной информации о состоянии элементов СДВТГ.

В беспрепятственном продвижении **транспортных потоков** сдерживающим фактором являются отцепки от угольных маршрутов из-за перелома весовых норм по станции Хабаровск-2, а также по технической неисправности. Для уменьшения их негативных последствий и, в идеальном случае, их ликвидации необходим учет и статистика конкретных причин задержек (несвоевременная смена бригад, перелом веса, технические неисправности, нарушение плана формирования и т. д.) по видам маршрутов, по срокам задержки, по станциям и дорогам.

Перечень объектов может быть расширен или сужен в зависимости от степени подробности анализа. Более того, каждое из множеств X, T, G, Y, F, E и P может быть описано более детально как объединение других множеств. В частности, из элементов можно выделить подмножества элементов однородного типа:

- угледобывающие предприятия, объединяемые в угольные бассейны;
- станции, на которых формируется **грузовой поток**;
- транзитные ж.-д. станции, учет которых необходим для сопряжения с информационной базой и учета продвижения груза и его задержек (**информационными потоками**);
- морские порты — своеобразные перекрестья, в которых фокусируются пучки грузов, отправляемых со станций угледобывающих бассейнов и подвергаемых перевалке на морские суда;
- грузоотправители, грузополучатели и грузовладельцы.

Одной из наиболее важных проблем, связанных с экспортом угля, является анализ экономических аспектов его прохождения через морские порты. Доставка угля железнодорожным транспортом, его перевалка на морские суда и дальнейшая транспортировка морем может рассматриваться как комплекс параллельно происходящих и взаимодействующих между собой процессов, для моделирования которых может быть применена теория взаимодействующих процессов (ТВП) [20, 21, 26].

Напомним, что ТВП разрабатывалась первоначально для описания широкого круга машинных применений по управлению процессами и дискретному моделированию событий, начиная с торговых автоматов и заканчивая сложными вычислительными системами с разделяемыми ресурсами. Поэтому является желательной разработка варианта такой теории, учитывающей реалии функционирования СДВТГ с целью ее **логистизации**. Попытаемся сделать некоторый обзор наших размышлений в этом направлении.

В качестве основных действий, с помощью которых можно комбинировать отдельные процессы P и Q , предлагается использовать следующие операции:

- детерминированный, недетерминированный и генеральный выборы (“|”, “П”, “П”);
- рекурсия и операция неподвижной точки (“ $x \rightarrow P$ ”, “ $\mu P(x \rightarrow P)$ ”);
- последовательная и параллельная композиция (“ $P ; Q$ ”, “ $P \parallel Q$ ”);
- чередование, сокрытие и подчинение (“ $P \parallel Q$ ”, “ $P \setminus Q$ ”, “ $P \parallel Q$ ”).

С помощью этих действий, например, процесс **ПЕРЕВАЛКА**, служащий узловым процессом во взаимодействии транспортных единиц **ВАГОН** и **СУДНО**, может быть описан через процессы, соответствующим следующим объектам управления: **ПРИЧАЛ**, **ПЕРЕГР-ФРОНТ**, **ПУТЬ**, **СКЛАД**, **ПЕРЕГР-МЕХАНИЗМ**, **ОТПРАВКА** [9, 10].

По нашему мнению, следует ввести еще ряд действий, обоснованием для чего являются следующие факторы:

- для описания взаимодействия грузовых потоков в пункте перевалки имеет смысл использовать понятие абстрактного **транспортера** и операции **сцепления** процессов:

$ПОГРУЗКА \gg ВАГОН \gg ПЕРЕВАЛКА \gg СУДНО$;

- для описания задержек и отцепок, связанных с функционированием процесса **ВАГОН** (а также неполадок в функционировании процесса **ПЕРЕГР-МЕХАНИЗМ**) имеет смысл использовать понятия **прерывания** (“ \wedge ”) и **катастрофы** (“ Δ ”):

$ВАГОН \wedge ОТЦЕПКА$,

$ВАГОН \Delta ТЕХНИЧЕСКАЯ НЕИСПРАВНОСТЬ ВАГОНА$.

Естественной структурой, накапливающей материальные объекты и сцепляющей транзакции взаимодействующих процессов, моделирующих грузовые потоки, является **СКЛАД**. Однако данная структура замыкает на себя и весомый информационный поток, связывая между собой такие объекты, как: объединения-экспортеры, марки угля, станции отправления, фирмы-отправители и др. Отдельные элементы вышеуказанного информационного потока находятся между собой в некотором отношении, которое может быть задано матрицей инцидентности, что позволяет использовать топологический анализ их взаимосвязи [20, 21].

В статье [21] приведена терминология, относящаяся к симплициальным комплексам, и рассмотрена задача восстановления комплекса по совокупности размерностей его симплексов с использованием понятия p -матрицы. В настоящей работе мы сосредоточим свое дальнейшее внимание на анализе связности, который может быть проведен с помощью исследования цепей q -связи и групп гомологии. Остановимся кратко на соответствующих понятиях.

Данные два симплекса S_i и S_j комплекса K соединены **цепью q -связи**, если существует последовательность симплексов $S_i, S_k, S_l, \dots, S_j$ в K такая, что два последовательных симплекса обладают общей гранью, причем размерности этих общих граней не меньше q . Понятие q -связности симплексов является отношением эквивалентности, и, таким образом, задача изучения глобальной структуры связности комплекса K сводится к рассмотрению классов q -эквивалентности. Если комплекс K имеет размерность n , то для каждого значения $q = 0, 1, \dots, n$ можно определить число различных классов q -эквивалентности Q_q и составить **структурный вектор комплекса** $Q = (Q_n, Q_{n-1}, \dots, Q_1, Q_0)$.

Образно говоря, q -связность можно представить так. Предположим, что мы обзреваем комплекс K через очки, которые дают возможность видеть его только в размерности q и выше. Тогда мы увидим комплекс K разбитым на Q_q несвязных частей.

Помимо структурного вектора комплекса, важной характеристикой является эксцентриситет, определяющий степень интегрированности каждого отдельного симплекса в структуре всего комплекса. Эксцентриситет симплекса S вычисляется по формуле

$$ecc(S) = \frac{d(S) - q(S)}{q(S) + 1}, \quad (6)$$

где $d(s)$ — размерность симплекса S , а $q(S)$ — наибольшее значение q , при котором симплекс S становится связанным с каким-либо другим симплексом из K .

Проведем анализ связности применительно к подсистеме, включающей объединения-экспортеры угля S_1, \dots, S_{18} и станции отправления X_1, \dots, X_{24} , связывая их экспортом угля

через морской порт Восточный (расшифровку обозначений см. в конце указанной статьи). Соответствующая матрица инцидентности A имеет вид:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	
S_1	1																								
S_2		1	1	(l)	(l)																				
S_3						1	1	1																	
S_4			1	1																					
S_5									1	1	1														
S_6							1	*	1	1	1	1	1	1	1	1	1	1	1	1					
S_7														1								1			
S_8	1			1	1	*		1	1	1							1		1			1	1		
S_9	1																								
S_{10}										1															
S_{11}	1																								
S_{12}											1														
S_{13}								1																	
S_{14}								1																	1
S_{15}								1																	
S_{16}	1																								
S_{17}	1							1																	
S_{18}								1																	

В результате проведенного q -анализа в применении к комплексу $K = K_S(X; A)$ получены следующие разбиения K_q на компоненты связности:
 $q = 12 \div 10, Q_q = 1, K_q = (\{S_6\})$;
 $q = 9 \div 5, Q_q = 2, K_q = (\{S_6\}, \{S_8\})$;
 $q = 4, Q_q = 1, K_q = (\{S_6, S_8\})$;

$q = 3, Q_q = 2, K_q = (\{S_2\}, \{S_6, S_8\})$;
 $q = 2, Q_q = 3, K_q = (\{S_2\}, \{S_3\}, \{S_5, S_6, S_8\})$;
 $q = 1, Q_q = 5, K_q = (\{S_2, S_4\}, \{S_3, S_5, S_6, S_8\}, \{S_7\}, \{S_{14}\}, \{S_{17}\})$;
 $q = 0, Q_q = 1, K_q = (\{все S_i\})$.
 Структурный вектор комплекса K равен $Q = (1^3, 2^5, 1, 2, 3, 5, 1)$. Значения $d(S)$, $q(S)$ и эксцентриситета каждого симплекса сведены в таблицу 6.

Таблица 6

Значения $d(S)$, $q(S)$ и эксцентриситета каждого симплекса

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}
$d(S)$	0	3	2	1	2	12	1	9	0	0	0	0	0	1	0	0	1	0
$q(S)$	0	1	0	1	2	4	0	4	0	0	0	0	0	0	0	0	0	0
$ecc(S)$	0	1	2	0	0	8/5	1	1	0	0	0	0	0	1	0	0	1	0

Проведенный анализ позволяет сделать вывод, что S_3 и S_6 (Гленкор и Карботранс) являются наиболее выделяющимися среди всех прочих в своей деятельности по экспорту угля. Аналогичные рассуждения и анализ можно провести в применении к обратному отношению A' и комплексу $K' = K_X(S; A')$, что дает результаты:
 $q = 5, Q_q = 1, K_q = (\{S_8\})$;
 $q = 4, Q_q = 2, K_q = (\{S_8\}, \{S_9\})$;

$q = 3, Q_q = 3, K_q = (\{S_1\}, \{S_8\}, \{S_9\})$;
 $q = 2, Q_q = 4, K_q = (\{S_1\}, \{S_2\}, \{S_8\}, \{S_9, S_{10}, S_{11}\})$;
 $q = 1, Q_q = 8, K_q = (\{S_1\}, \{S_2, S_5\}, \{S_3, S_4\}, \{S_6\}, \{S_7\}, \{S_8\}, \{S_9, S_{10}, S_{11}, S_{17}, S_{19}\}, \{S_{15}\})$;
 $q = 0, Q_q = 2, K_q = (\{все S_i, кроме S_2-S_5\}, \{S_2, S_3, S_4, S_5\})$,
 и, следовательно, структурный вектор комплекса K' равен $Q' = (1, 2, 3, 4, 8, 2)$. Для значений $d(X)$, $q(X)$ и $ecc(X)$ получены следующие результаты (см. таблицу 7).

Таблица 7

Значения $d(X)$, $q(X)$ и $ecc(X)$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20-24}
$d(X)$	3	2	1	1	1	1	1	5	4	2	2	0	0	0	1	0	1	0	1	0
$q(X)$	0	1	1	1	1	0	0	0	2	2	2	0	0	0	0	0	1	0	1	0
$ecc(X)$	3	1/2	0	0	0	1	1	5	2/3	0	0	0	0	0	1	0	0	0	0	0

Отсюда следует, что наиболее выделяющимися среди станций отправления, на которых ведется работа по экспорту угля, являются Нерюнгри и Партизанск.

Рассмотрим отношение Λ с гомологической точки зрения. Для соответствующего комплекса K гомология тривиальна, и соответствующие числа Бетти $b_0 = 1$, $b_i = 0$ (для всех $i > 0$). Это указывает на «заполненность», «сбалансированность» сложившейся экономической подсистемы. Однако, при условии, что две единицы в матрице Λ заменить нулями, как указано, одномерные симплексы $S_2 = X_2 + X_3$, $S_4 = X_3 + X_4$ и грань симплекса S_8 : $S_8^{24} = X_2 + X_4$ образуют нетривиальную гомологию $S_2 + S_4 - S_8^{24}$ (1-цикл, не являющийся границей цепи). Аналогично, при добавлении двух единиц на места, указанные в матрице Λ звездочками, двумерный симплекс $S_3 = X_6 + X_7 + X_8$ и грани симплексов S_6 и S_8 : $S_6^{789} = X_7 + X_8 + X_9$, $S_8^{679} = X_6 + X_7 + X_9$ образуют нетривиальную гомологию $S_3 + S_6^{789} + S_8^{679}$ (2-цикл, не являющийся границей цепи).

ЗАКЛЮЧЕНИЕ

Проведенный анализ позволяет сделать вывод, что соответствующие объединения, а именно (*Иркутскуглесбыт, Камчатоблававтопранс, Магаданская ТЭЦ*), или же (*Гленкор, Карбопранс, Магаданская ТЭЦ*), при соответствующих (гипотетических) изменениях в своей экономической деятельности могут войти в отношения сложной конкуренции, либо что параллельная деятельность этих объединений создаст (при тех же изменениях) экономическую нишу, которая может оказаться выгодной для других экспортеров.

ЛИТЕРАТУРА

1. Воронин В. С. Информационное обеспечение перевозок // Железнодорожный транспорт. 2001. № 6. С. 65–66.
2. Ткачев И. Г. Организационно-экономический механизм развития транспортных систем на базе цифровых технологий: дис. ... канд. экон. наук: 08.00.05. — СПб.: ПГУПС, 2019. — 148 с.
3. Бубнова Г. В. Цифровая логистика и безопасность цепей поставок / Г. В. Бубнова, П. В. Куренков, А. Г. Некрасов // Логистика. 2017. № 7(128). С. 46–50.
4. Куренков П. В. Цифровизация логистики мультимодальных перевозок / П. В. Куренков, А. А. Сафронова, Д. Г. Кахриманова // Эксплуатация морского транспорта. 2018. № 1 (86). С. 3–8.
5. Транспортные коридоры и оси в цифровой транспортной системе / В. П. Куприяновский, Г. В. Бубнова, А. А. Зенкин [и др.] // Транспорт: наука, техника, управление. Научный информационный сборник / ВИНТИ РАН. — 2017. № 7. С. 11–20.
6. Экономика инноваций цифровой железной дороги. Опыт Великобритании / В. П. Куприяновский, Г. В. Бубнова, П. В. Куренков [и др.] // International Journal of Open Information Technologies. 2017. Т. 5, № 3. С. 79–99.
7. Строительство и инженерия на основе стандартов BIM как основа трансформаций инфраструктур в цифровой экономике / В. П. Куприяновский, С. А. Синягов, П. В. Куренков [и др.] // International Journal of Open Information Technologies. 2017. Т. 5, № 5. С. 46–79.
8. Прорывные инновационные технологии для инфраструктур. Евразийская цифровая железная дорога как основа логистического коридора нового шелкового пути /

И. А. Соколов, В. П. Куприяновский, О. Н. Дунаев [и др.] // International Journal of Open Information Technologies. 2017. Т. 5, № 9. С. 102–118.

9. Куприяновский В. П. Грузопассажирские транспортные коридоры в евро-азиатском цифровом пространстве / В. П. Куприяновский, П. В. Куренков, О. Н. Мадяр // Транспорт: наука, техника, управление. Научный информационный сборник / ВИНТИ РАН. — 2017. № 11. С. 8–17.
10. Белый О. В. Стратегия развития транспортной отрасли в условиях системного кризиса страны // Бюллетень транспортной информации. 1999. № 8 (50). С. 2–6.
11. Введение в топологию: Учебное пособие / Ю. Г. Борисович, Н. М. Близняков, Я. А. Израилевич, Т. Н. Фоменко. — 2-е изд., доп. — М.: Наука. Физматлит, 1995. — 416 с.
12. Багимов А. В. Моделирование взаимодействия субъектов транспортного рынка в системе внешнеторговых перевозок / А. В. Багимов, П. В. Куренков // Вестник государственного морского университета имени адмирала Ф. Ф. Ушакова. 2014. № 3 (8). С. 5–12.
13. Понтрягин Л. С. Основы комбинаторной топологии. — 2-е изд. — М.: Наука, 1976. — 136 с.
14. Спенсьер Э. Алгебраическая топология. — М.: Мир, 1971. — 680 с.
15. Фоменко А. Т. Наглядная геометрия и топология: Математические образы в реальном мире. — 2-е изд. — М.: Изд-во Моск. ун-та, Изд-во "ЧеРо", 1998. — 416 с.
16. Свитцер Р. М. Алгебраическая топология — гомотопии и гомологии / Пер. с англ. Ю. П. Соловьева. — М.: Наука. Главная редакция физико-математической литературы, 1985. — 608 с.
17. Теория гомологий: Введение в алгебраическую топологию = Homology theory: An introduction to algebraic topology / П. Дж. Хилтон, С. Уайли; пер. с англ. А. В. Архангельского; под ред. П. С. Александрова. — М.: Мир. Редакция литературы по математическим наукам, 1966. — 452 с.
18. Толковый словарь по вычислительным системам / Под ред. В. Иллинуорта [и др.]; пер. с англ. А. К. Белоцкого и [др.]; под ред. Е. К. Масловского. — М.: Машиностроение, 1989. — 567 с.
19. Куренков П. В. Логистический подход к выбору схемы поставки товаров / П. В. Куренков, А. А. Калущин // Железнодорожный транспорт. 2000. № 11. С. 40–43.
20. Куренков П. В. Внешнеторговые перевозки в смешанном сообщении. Экономика. Логистика. Управление: Монография / П. В. Куренков, А. Ф. Котляренко. — Самара: СамГАПС, 2003. — 634 с.
21. Куренков П. В. Моделирование экспорта угля через морские порты / П. В. Куренков, И. С. Фролов // Транспорт: наука, техника, управление: сборник обзорной информации / ВИНТИ РАН. — 2001. № 7. С. 34–37.
22. Александров П. С. Комбинаторная топология. — М.; Л.: Гостехиздат, 1947. — 660 с.
23. Жардемев Б. Б. Формирование и развитие структур железнодорожных станций и узлов: Методы исследования и оценки / Б. Б. Жардемев; МПС РФ. МИИТ. — М.: МИИТ, 1999. — 150 с.
24. Касти Дж. Большие системы: Связность, сложность и катастрофы / Пер. с англ. под ред. Ю. П. Гупало, А. А. Пионтковского. — М.: Мир, 1982. — 216 с. — (Техника и экология).

25. Эткин Р. Х. Городская структура // Математическое моделирование / Под ред. Дж. Эндрюса, Р. Мак-Лоуна; пер. с англ. Ю. П. Гупало. — М.: Мир, 1979. — С. 235–248.

26. Хоар Ч. Взаимодействующие последовательные процессы = Communicating Sequential Processes /

Пер. с англ. А. А. Бульонковой; под ред. А. П. Ершова. — М.: Мир. Редакция литературы по математическим наукам, 1989. — 264 с.

Modeling the Topology of the Interaction of Elements and Flows of Various Types in the System of Foreign Trade Transportation

Grand PhD in Econ. Sc., PhD in Eng. Sc. P. V. Kurenkov
Russian University of Transport (MIIT)
Moscow, Russia
petrkurenkov@mail.ru

Abstract. The relationships of flows of various types (transport, freight, information, financial, energy, legal and others) are considered between themselves and the elements in the delivery system of foreign trade goods. A model of the topology of the interaction of flows with the definition of an integral indicator of the quality of a particular route or delivery option is presented. An example of calculation is given.

Keywords: element, flow, system, interaction, delivery, wagon, cargo, multiplex, complex.

REFERENCES

1. Voronin V. S. Information Support of Transportation [Informatsionnoe obespechenie perevozk], *Railway Transport [Zheleznodorozhnyy transport]*, 2001, No. 6, Pp.65–66.
2. Tkachev I. G. Organizational and Economic Mechanism for The Development of Transport Systems Based on Digital Technologies [Organizatsionno-ekonomicheskii mekhanizm razvitiya transportnykh sistem na baze tsifrovyykh tekhnologiy]: diss. on competition of a scientific degree Ph.D. (Econ.), Saint Petersburg, 2019, 148 p.
3. Bubnova G. V., Kurenkov P. V., Nekrasov A. G. Digital Logistics and Security of Supply Chains [Tsifrovaya logistika i bezopasnost' tsepey postavok], *Logistics [Logistika]*, 2017, No. 7 (128), Pp. 46–50.
4. Kurenkov P. V., Safronova A. A., Kakhrimanova D. G. Digitalization of Logistic of Multimodal Transportation [Tsifrovizatsiya logistiki mul'timodal'nykh perevozk], *Operation of Maritime Transport [Ekspluatatsiya morskogo transporta]*, 2018, No. 1 (86), Pp. 3–8.
5. Kupriyanovsky V. P., Bubnova G. V., Zenkin A. A., et al. Transport Corridors and Axes in a Digital Intermodal System [Transportnye koridory i osi v tsifrovoy transportnoy sisteme], *Transport: Science, Technology, Management. Scientific Information Collection [Transport: nauka, tekhnika, upravlenie. Sbornik naychnoy informatsii]*, 2017, No. 7, Pp. 11–20.
6. Kupriyanovsky V. P., Bubnova G. V., Kurenkov P. V., et al. Economics of Innovations for Digital Railways. Experience in the UK [Ekonomika innovatsiy tsifrovoy zheleznoy dorogi. Opyt Velikobritanii], *International Journal of Open Information Technologies*, 2017, Vol. 5, No. 3, Pp. 79–99.
7. Kupriyanovsky V. P., Sinyagov S. A., Kurenkov P. V., et al. Building and Engineering Based on BIM Standards as the Basis for Transforming Infrastructures in the Digital Economy [Stroitel'stvo i inzheneriya na osnove standartov BIM kak osnova transformatsiy infrastruktur v tsifrovoy ekonomike], *International Journal of Open Information Technologies*, 2017, Vol. 5, No. 5, Pp. 46–79.
8. Sokolov I. A., Kypriyanovsky V. P., Dunaev O. N., et al. On Breakthrough Innovative Technologies for Infrastructures. The Eurasian Digital Railway as a Basis of the Logistic Corridor of the New Silk Road [Proryvnye innovatsionnye tekhnologii dlya infrastruktur. Evraziyskaya tsifrovaya zheleznaaya doroga kak osnova logisticheskogo koridora novogo shelkovogo puti], *International Journal of Open Information Technologies*, 2017, Vol. 5, No. 9, Pp. 102–118.
9. Kupriyanovsky V. P., Kurenkov P. V., Madyar O. N. Cargo Passenger Transport Corridors in the Euro-Asian Digital Space [Gruzopassazhirskie transportnye koridory v evroaziatskom tsifrovom prostranstve], *Transport: Science, Technology, Management. Scientific Information Collection [Transport: nauka, tekhnika, upravlenie. Sbornik naychnoy informatsii]*, 2017, No. 11, Pp. 8–17.
10. Belyi O. V. The Development Strategy of the Transport Sector in a Systemic Crisis of the Country [Strategiya razvitiya transportnoy otrasli v usloviyakh sistemnogo krizisa strany], *Bulletin of Transport Information [Byulleten' transportnoy informatsii]*, 1999, No. 8 (50), Pp. 2–6.
11. Borisovich Yu. G., Bliznyakov N. M., Izrailevich Ya. A., Fomenko T. N. Introduction to topology: Study guide [Vvedenie v topologiyu: Uchebnoe posobie], Moscow, Nauka Publishers, 1995, 416 p.
12. Bagimov A. V., Kurenkov P. V. Modelling of Interaction of Subjects of the Transport Market in System of the Foreign Trade Transportations [Modelirovanie vzaimodeystviya sub'ektov transportnogo rynka v sisteme vneshnetorgovykh perevozk], *Bulletin of the Admiral F. F. Ushakov State Maritime University [Vestnik gosudarstvennogo morskogo universiteta imeni admirala F. F. Ushakova]*, 2014, No. 3 (8), Pp. 5–12.
13. Pontryagin L. S. Fundamentals of combinatorial topology [Osnovy kombinatornoy topologii], Moscow, Nauka Publishers, 1976, 136 p.
14. Spenier E. Algebraic topology [Algebraicheskaya topologiya], Moscow, Mir Publishers, 1971, 680 p.
15. Fomenko A. T. Visual geometry and topology: Mathematical images in the real world [Naglyadnaya geometriya i topologiya: Matematicheskie obrazy v real'nom mire], Moscow, Moscow State University, 1998, 416 p.
16. Switzer R. M. Algebraic topology — homotopy and homology [Algebraicheskaya topologiya — gomotopii i gomologii], Moscow, Nauka Publishers, 1985, 608 p.
17. Hilton P. J., Wylie S. Homology theory: An introduction to algebraic topology [Teoriya gomologiy: Vvedenie v algebraicheskuyu topologiyu], Moscow, Mir Publishers, 1966, 452 p.

18. Dictionary of computing [Tolkovyy slovar' po vychislitel'nym sistemam], Moscow, Mashinostroenie Publishers, 1989, 567 p.

19. Kurenkov P. V., Kalushin A. A. Logistic Approach to the Choice of Goods Delivery Scheme [Logisticheskiy podkhod k vyboru skhemy postavki tovarov], *Railway transport [Zheleznodorozhnyy transport]*, 2000, No. 11, Pp. 40–43.

20. Kurenkov P. V., Kotlyarenko A. F. Foreign trade transportations in mixed traffic. Economy. Logistics. Management: Monograph [Vneshnetorgovye perevozki v smeshannom soobshchenii. Ekonomika. Logistika. Upravlenie: Monografiya]. Samara, Samara State Transport Academy, 2003, 634 p.

21. Kurenkov P. V., Frolov I. S. Modeling Coal Exports Through Seaports [Modelirovanie eksporta uglya cherez morskoye prymorskiye porty], *Transport: Science, Technology, Management: A Collection of Overview Information [Transport: nauka, tekhnika, upravlenie: sbornik obzornoy informatsii]*, 2001, No. 7, Pp. 34–37.

22. Aleksandrov P. S. Combinatorial topology [Kombinatornaya topologiya], Moscow, Leningrad, Gostekhizdat, 1947, 660 p.

23. Zhardemov B. B. Formation and development of structures of railway stations and junctions: Research and evaluation methods [Formirovanie i razvitie struktur zheleznodorozhnykh stantsiy i uzlov: Metody issledovaniya i otsenki], Moscow, Russian University of Transport (MIIT), 1999, 150 p.

24. Casti J. Connectivity, complexity, and catastrophe in large-scale systems [Bol'shie sistemy: Svyaznost', slozhnost' i katastrofy], Moscow, Mir Publishers, 1982, 216 p.

25. Jetkin R. H. City Structure [Gorodskaya struktura], *In: Mathematical modeling [Matematicheskoe modelirovanie]*. Moscow, Mir Publishers, 1989, Pp. 235–248.

26. Hoare C. A. R. Communicating Sequential Processes [Vzaimodeystvuyushchie posledovatel'nye protsessy], Moscow, Mir Publishers, 1989, 264 p.

Оценка оперативности передачи больших данных на примере базы данных PostgreSQL, платформы Hadoop и системы Sqoop

магистр О. В. Баданина, к.т.н. С. И. Гиндин
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, Россия
olgagaykova@gmail.com, sgindin@gmail.com

д.т.н. А. Д. Хомоненко
Петербургский государственный университет
путей сообщения императора Александра I,
Военно-космическая академия им. А. Ф. Можайского
Санкт-Петербург, Россия
khomon@mail.ru

Аннотация. Рассматривается оценка характеристик оперативности передачи больших данных. Для этого используются демонстрационная база данных PostgreSQL как источник данных, платформа Hadoop как инструмент хранения данных, а также инструмент передачи данных Sqoop. Результаты экспериментов демонстрируют изменение параметров, характеризующих оперативность передачи данных при импорте, в зависимости от их объема и типа. Полученные результаты экспериментов могут быть использованы для прогнозирования характеристик оперативности обработки больших данных. Такой прогноз можно провести на основе использования полученных характеристик оперативности в качестве обучающей выборки для нейронной сети. Кроме того, прогноз оперативности обработки больших данных целесообразно проводить с помощью моделей многоканальных систем массового обслуживания с разогревом и охлаждением, а также сетей массового обслуживания. Расчет показателей оперативности перемещения больших данных в различных режимах функционирования информационной инфраструктуры целесообразно выполнять для оптимального аварийного восстановления хранилищ данных и обеспечения требуемой киберустойчивости цифровых платформ в целом.

Ключевые слова: большие данные, оперативность передачи данных, СУБД PostgreSQL, платформа Hadoop, инструмент передачи данных Sqoop, программная платформа MapReduce.

ВВЕДЕНИЕ

В современном мире каждый день увеличивается количество разработок в сфере информационных технологий, которые интегрируются в различные направления деятельности общества. В результате возрастают требования для решения новых задач, которые предполагают выработку огромных объемов данных.

Согласно прогнозу международной исследовательской консалтинговой компании IDC (International Data Corporation) [1], общий объем всех данных в мире изменится от 33 зеттабайт в 2018 году до 175 зеттабайт к 2025 году. Такой рост показателей связывают с увеличением количества умных гаджетов, роботов и устройств, окружающих нас каждый день, которые будут производить значительное количество данных.

С увеличением объема генерируемых человечеством данных увеличивается потребность в хранилищах для большого объема информации, а также в методах и средствах их обработки и анализа. Большая часть глобальных данных является неструктурированной или частично

структурированной. Они могут быть представлены в различных форматах, например, это могут быть веб-журналы, видеозаписи, текстовые документы, машинный код, геопространственные данные и многое другое. Источники хранения этих данных также многообразны и могут располагаться как в организации, так и за ее пределами. В результате компании имеют огромные наборы данных, которые необходимо анализировать, получать результаты и делать выводы на их основе, но не имеют инструментов для проведения всех необходимых работ. Проблематика анализа больших данных представлена в работе [2].

Традиционные методы анализа информации не позволяют обеспечивать эффективную работу с очень большим объемом неструктурированных данных. Это связано с тем, что часто большая часть генерируемой информации не представляет значительной важности для исследования и практического использования. Поэтому необходимо сначала извлечь ценную информацию, поскольку традиционные системы изначально разрабатывались для хранения релевантных данных и имеют ограничения масштабируемости. Традиционные решения обеспечивают работу со структурированными данными и позволяют структурировать пакеты исходных данных небольшого объема, затем выполняются обработка, хранение и анализ уже структурированных данных.

Указанные факторы, связанные с необходимостью качественной обработки огромных объемов неструктурированных данных, неэффективности традиционных методов, увеличением масштабности задач на предприятиях, привели к появлению и развитию технологий больших данных. В настоящее время с понятием «Big Data» тесно связывается характеристика технологий для их сбора, хранения, обработки, анализа, визуализации и использования.

Большие данные характеризуются следующими признаками:

- Volume — объем; означает, что анализируются большие массивы информации;
- Velocity — скорость; скорость накопления данных постоянно увеличивается;
- Variety — многообразие; данные могут быть представлены в разных форматах и взяты из разных источников.

КОМПАНИИ В ОБЛАСТИ ТЕХНОЛОГИЙ БОЛЬШИХ ДАННЫХ

В настоящее время все большее число компаний используют технологии больших данных, поскольку с их

помощью анализ, обработка данных и прогнозирование становятся гораздо эффективнее, что позволяет корпорациям увеличивать свои доходы, оптимизировать работу и добиваться лучших результатов.

Существенный вклад в изучение технологий для больших данных вносят крупные мировые компании, они также занимаются обработкой и анализом данных и создают программно-аппаратные комплексы. Их можно разделить на несколько групп:

1. Поставщики инфраструктуры. Решают задачи хранения и предобработки данных, например: IBM, Microsoft, Oracle, SAP и другие.

2. Датамайнеры — разработчики алгоритмов, которые помогают заказчикам извлекать ценные сведения. Среди них Yandex Data Factory, «Алгомост», Glowbyte Consulting, CleverData и др.

3. Системные интеграторы — компании, которые внедряют системы анализа больших данных на стороне клиента: «Форс», «Крок» и др.

4. Потребители — компании, которые покупают программно-аппаратные комплексы и заказывают алгоритмы у консультантов. Это «Сбербанк», «Газпром», МТС, «Мегафон» и другие компании из отраслей финансов, телекоммуникаций, ритейла.

5. Разработчики готовых сервисов. Предлагают готовые решения на основе доступа к большим данным. Они открывают возможности Big Data для широкого круга пользователей.

ХАРАКТЕРИСТИКА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ И ТЕХНОЛОГИЙ БОЛЬШИХ ДАННЫХ

Технологии, которые используются для сбора, обработки больших данных делятся на следующие группы: технологическое оборудование, сервисные услуги, программное обеспечение.

Для проведения исследования выбран инструмент обработки, относящийся к последней из перечисленных групп, а именно Hadoop — свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределенных программ, работающих на кластерах из сотен и тысяч узлов. Платформа Hadoop является основополагающей технологией, предназначенной для хранения и обработки больших данных. Описание принципов работы и преимуществ Hadoop представлено в работе [3].

В качестве среды подготовки данных для Hadoop рассматривается база данных PostgreSQL [4]. При работе с большими данными также рассматриваются NoSQL базы данных, в статье [5] проведено сравнение работы СУБД PostgreSQL и MongoDB. Основные различия заключаются в том, что PostgreSQL является свободным программным обеспечением, а то время как СУБД MongoDB тщательно разработана и имеет хорошую поддержку. При этом СУБД MongoDB опережает PostgreSQL для операций обновления и поиска неиндексированных данных. Но в то же время PostgreSQL имеет преимущество на операциях вставки данных. Применение СУБД MongoDB является оправданным решением для хранения редко изменяющихся и часто читаемых данных. Более подробный обзор технологий NoSQL приведен в работе [6].

Для исследования оперативности импорта данных в Hadoop источником передаваемых данных выбрана СУБД PostgreSQL, так как она является наиболее подходящей для проводимого исследования.

Для передачи данных на платформу Hadoop выбран Sqoop [7] — инструмент, предназначенный для передачи данных между Hadoop и реляционными базами данных. Sqoop действует как промежуточный уровень между Hadoop и системами реляционных баз данных. Этот инструмент позволяет импортировать данные из реляционных СУБД в Hadoop, а затем экспортировать обработанные данные обратно в СУБД. Sqoop достаточно прост в использовании, он автоматизирует большую часть рассматриваемого процесса, обеспечивает отказоустойчивость, а также предоставляет интерфейс командной строки.

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК ОПЕРАТИВНОСТИ ПЕРЕДАЧИ ДАННЫХ

Исследование характеристик оперативности передачи данных производилось на примере демонстрационной базы данных [8], предоставляемой компанией Postgres Professional. Предметной областью выбранной базы данных являются авиаперевозки по России. Эта база данных представлена в трех версиях: данные по полетам за один месяц (small-database, обозначена как S-база), данные за три месяца (medium-database, обозначена как M-база), данные за год (big-database, обозначена как B-база). Состав базы и размерность можно проследить по таблице 1.

Таблица 1

Сводные данные о размерах БД и таблиц

	Размеры БД и таблиц			Назначение таблиц и БД
	S-база	M-база	B-база	
DataBase	300 MB	700 MB	2,5 GB	БД в целом
Aircrafts	16 KB	16 KB	16 KB	Самолеты
Airports	48 KB	48 KB	48 KB	Аэропорты
Boarding_passes	31 MB	102 MB	427 MB	Посадочные талоны
Booking	13 MB	30 MB	105 MB	Бронирования
Flights	3 MB	6 MB	19 MB	Рейсы
Seats	88 KB	88 KB	88 KB	Места
Ticket_flights	64 MB	145 MB	516 MB	Перелеты

В рамках исследования проведен единовременный импорт всех таблиц базы данных, каждой из трех версий, а также выполнен отдельный импорт каждой из таблиц базы данных. Несмотря на то, что таблицы Aircrafts, Airports, Seats одинаковы во всех вариантах, они, как и остальные таблицы, размер которых увеличивается в зависимости от размера базы, импортированы три раза для сравнения показателей [8].

Импорт данных исследовался с использованием парадигмы выполнения распределенных вычислений Map-Reduce [9]. Для импорта таблиц использовалась стандартная команда Sqoop [10]:

```
sqoop import --connect jdbc:postgresql://<имя хоста>/<имя базы данных>
--username<пользователь, под которым идет подключение к базе данных>
--password<пароль от пользователя>
--table<имя таблицы> -m 1
```

Для упрощения анализа по результатам импорта таблиц составлены сводные таблицы результатов. Первым параметром для сравнения является время передачи данных из таблиц (см. табл. 2). При этом можно увидеть, что показатели для таблиц S-базы и M-базы примерно одинаковы. Причиной этого является сравнительно небольшой объем данных, а также незначительная разница в объемах данных. В то же время для В-базы время передачи в 1,5 раза

больше, что является следствием того, что объем передаваемых данных вырос в 3-4 раза.

Таблица 2

Время передачи данных

	S-база	M-база	В-база
Все таблицы	227 s	247 s	301 s
Aircrafts	44 s	50 s	32 s
Airports	34 s	31 s	66 s
Boarding_passes	34 s	34 s	52 s
Booking	36 s	34 s	69 s
Flights	32 s	33 s	40 s
Seats	31 s	30 s	38 s
Ticket_flights	36 s	37 s	57 s

Следующий показатель — Aggregate resource allocation (Совокупное распределение ресурсов), где MB-sec — совокупный объем памяти (в мегабайтах), выделенное приложением, умноженное на число секунд, в течение которого приложение выполнялось. Очередной показатель — vcores-seconds — агрегированное число vcores (виртуальных ядер), выделенных приложением, умноженное на число секунд, в течение которых приложение работало (см. табл. 3). При сравнении определяется та же закономерность, что и для времени передачи данных. Распределения ресурсов для таблиц S- и M-базы примерно равны, для В-базы показатели возрастают в 1,5-2 раза.

Таблица 3

Сводная таблица совокупного распределения ресурсов

	S-база	M-база	В-база
Aircrafts	87 533 MB-sec / 59 vcore-sec	95 607 MB-sec / 64 vcore-sec	67 176 MB-sec / 46 vcore-sec
Airports	71 092 MB-sec / 49 vcore-sec	64 187 MB-sec / 43 vcore-sec	124 551 MB-sec / 84 vcore-sec
Boarding_passes	72 255 MB-sec / 50 vcore-sec	74 437 MB-sec / 51 vcore-sec	118 027 MB-sec / 85 vcore-sec
Booking	75 056 MB-sec / 52 vcore-sec	71 981 MB-sec / 49 vcore-sec	140 808 MB-sec / 99 vcore-sec
Flights	68 304 MB-sec / 46 vcore-sec	69 303 MB-sec / 48 vcore-sec	83 870 MB-sec / 58 vcore-sec
Seats	65 145 MB-sec / 44 vcore-sec	63 916 MB-sec / 43 vcore-sec	81 566 MB-sec / 56 vcore-sec
Ticket_flights	79 008 MB-sec / 55 vcore-sec	81 686 MB-sec / 57 vcore-sec	130 015 MB-sec / 95 vcore-sec

В ходе экспериментов рассматривались группы показателей. Первая из них — File system counters (Метрики операций файловой системы). Из этой группы были рассмотрены следующие два показателя. File: Number of bytes written — общее количество байтов, записанных в локальную файловую систему; для всех операций импорта он примерно одинаков, идеального ожидаемого значения

параметра нет, так как оно зависит от логики маппера (функция map). HDFS: Number of bytes written — общее количество байтов, записанных в определенную файловую систему. Так как параметр зависит от объема данных, то очевидно, что с возрастанием количества обрабатываемых данных увеличивается значение показателя. Результаты эксперимента отображены в таблице 4.

Таблица 4

Число записанных байтов

	БД	aircrafts	airports	Boarding_passes	Bookings	flights	seats	ticket_flights
File system counters								
File: Number of bytes written	S-база	208 088	208 292	208 294	208 255	208 379	208 254	208 293
	M-база	208 088	208 292	208 294	208 255	208 379	208 254	208 293
	В-база	207 330	207 660	207 798	207 881	208 129	208 120	208 293
HDFS: Number of bytes written	S-база	233	7 605	15 146 303	10 042 404	3 471 389	21 254	37 943 723
	M-база	233	7 605	50 032 241	22 678 839	7 409 102	21 254	86 219 771
	В-база	233	7 605	213 846 934	80 679 926	25 577 549	21 254	311 249 367

Следующие рассмотренные параметры принадлежат к группе Job Counters (Счетчики задачи). К ним относятся:

1. Total time spent by all maps in occupied slots (ms) — время выполнения map задач в миллисекундах. По полученным результатам для таблиц aircrafts, airports, seats (табл. 5) можно сказать, что время выполнения изменяется для каждой операции импорта, разница в интервале 5-7 тысяч миллисекунд. Такую погрешность можно объяснить выполнением операционной системой фоновых задач. Для оставшихся таблиц (boarding_passes, booking, flights, ticket_flights) прослеживается незначительное увеличение результатов при сравнении таблиц S-базы и M-базы, в то время как показатель для таблиц В-базы увеличивается в 2-2,5 раза.

2. Total time spend by all map task (ms) — общее время выполнения всех задач в миллисекундах, включая спекулятивные задачи. Согласно таблице 5 значение этого показателя имеет результаты, аналогичные предыдущему рассматриваемому параметру.

3. Total vcore-milliseconds taken by all map task. Этот счетчик измеряет ресурсы процессора, используемые всеми map-задачами. А именно: это агрегированное количество vcores, которое было назначено каждой map-задаче, умноженное на количество секунд, которое проработал mapper (картопостроитель).

4. Total megabyte-milliseconds taken by all map tasks — это счетчик, который измеряет ресурсы памяти, используемые всеми map-задачами. А именно: это совокупный объем памяти в мегабайтах, который был выделен каждой map-задаче, умноженное на количество секунд, которое проработал mapper.

Оба перечисленных параметра показывают схожие результаты (см. табл. 5). Для таблиц, размер которых изменяется, величина показателя увеличивается в 2,5-3 раза, в то время как для таблиц, размер которых не изменялся, значение колеблется, но разница является незначительной.

Таблица 5

Счетчики задач

	БД	aircrafts	airports	Boarding_passes	Bookings	flights	seats	ticket_flights
Job Counters								
Total time spent by all maps in occupied slots (ms)	S-база	62 784	64 544	69 312	70 320	61 568	54 688	90 248
	M-база	53 120	55 000	86 688	76 216	65 176	55 624	108 032
	В-база	56 056	87 120	212 864	195 896	84 680	62 544	246 088
Total time spent by all map tasks (ms)	S-база	7 848	8 068	8 664	8 790	7 696	6 836	11 281
	M-база	6 640	6 875	10 836	9 527	8 147	6 953	13 504
	В-база	7 257	10 890	26 608	24 487	10 585	11 568	30 761
Total vcore-milliseconds taken by all map tasks	S-база	7 848	8 068	8 664	8 790	7 696	6 836	11 281
	M-база	6 640	6 875	10 836	9 527	8 147	6 953	13 504
	В-база	7 257	10 890	26 608	24 487	10 585	11 568	30 761
Total megabyte-milliseconds taken by all map tasks	S-база	8 036 352	8 261 632	8 871 936	9 000 960	7 880 704	7 000 064	11 551 744
	M-база	6 799 360	7 040 000	11 096 064	9 755 648	8 342 528	7 119 872	13 828 096
	В-база	7 431 168	9 151 360	27 246 592	25 074 688	10 839 040	11 845 632	31 499 264

Следующая группа — Map-Reduce Framework (метрики, управляемые Map Reduce Framework) — является наиболее обширной. Рассмотрим более подробно входящие в нее показатели.

Параметр Map input records обозначает количество входных записей, потребляемых всеми картами в задании. Он увеличивается для каждой успешной записи, прочитанной из RecordReader, и передается методу картографа. Записи, которые не удалось прочитать задачам карты, не включены в эти счетчики.

Параметр Map output records — это количество выходных записей карт, созданных всеми картами в задании. Он увеличивается для каждой успешной записи, записанной картографами. Записи, которые не удалось записать задачам карты, не включены в эти счетчики.

Значение этих двух параметров (при успешном импорте таблицы) равняется количеству строк в таблице. Сравнив исходные и полученные значения (табл. 6), можно сказать, что все данные таблиц были импортированы успешно.

Параметр GC time elapsed характеризует общее время, в миллисекундах, потраченное на сборку мусора. Выполняется в Java путем поиска неиспользуемых объектов (на которые больше нет ссылок) и освобождения памяти.

CPU time spent (ms) характеризует накопленное процессорное время для задачи в миллисекундах. Как и для других параметров, связанных со временем, для таблиц S-базы и M-базы разница в значениях незначительна и наблюдается увеличение времени в 1,1-1,3 раза, в то время как для В-базы этот показатель увеличивается в 2-2,5 раза.

Таблица 6

Число записей Map-Reduce Framework

	БД	aircrafts	airports	Boarding_passes	Bookings	flights	seats	ticket_flights
MAP-REDUCE Framework								
Map input records	С-база	9	104	579 686	262 788	33 121	1 339	1 045 726
	М-база	9	104	1 894 295	593 433	65 664	1 339	2 360 335
	В-база	9	104	7 925 812	2 111 110	214 867	1 339	8 391 852
Map output records	С-база	9	104	579 686	262 788	33 121	1 339	1 045 726
	М-база	9	104	1 894 295	593 433	65 664	1 339	2 360 335
	В-база	9	104	7 925 812	2 111 110	214 867	1 339	8 391 852
GC time elapsed (ms)	С-база	67	65	95	129	97	70	117
	М-база	78	85	130	109	115	68	146
	В-база	68	80	274	186	126	84	314
CPU time spent (ms)	С-база	1 468	951	2 577	4 623	3 482	1 076	5 217
	М-база	1 233	1 123	6 249	4 186	4 107	1 045	8 171
	В-база	1 451	1 468	16 731	10 839	5 968	1 405	18 433

Параметр Bytes Written группы File output counters обозначает число байтов, записанное каждой задачей. Из приведенных в таблице 7 результатов видно, что число записанных байтов напрямую зависит от размера таблиц. Таким образом, мы видим, что для таблиц, имеющих одинаковый

размер, число байтов не изменяется (таблицы aircrafts, airports, seats), в то время как для других таблиц с увеличением размерности увеличивается число записанных байтов в тех же пропорциях, в которых увеличивается размер таблиц.

Таблица 7

Число записанных байтов

	БД	aircrafts	airports	Boarding_passes	Bookings	flights	seats	ticket_flights
File output counters								
Bytes Written	С-база	233	7 605	15146 303	10 042 404	3 471 369	21 254	37 943 723
	М-база	233	7 605	50 032 241	22 678 839	7 409 102	21 254	86 219 771
	В-база	233	7 605	213 846 934	80 679 926	25 577 549	21 254	311 249 367

Таблица 8

Скорость передачи данных

	БД	aircrafts	airports	Boarding_passes	Bookings	flights	seats	ticket_flights
Import Job Base								
Transferred	С-база	233 bytes in 50,941 sec (4,5739 bytes/sec)	7,4268 KB in 40,7748 sec (186,5122 bytes/sec)	14,4446 MB in 40,005 sec (369,7368 KB/sec)	9,5772 MB in 44,4965 sec (220,4002 KB/sec)	3,3106MB in 38,7671 sec (87,4455 KB/sec)	20,7559 KB in 37,18 sec (571,6515 bytes/sec)	36,186 MB in 42,4297 sec (873,3128 KB/sec)
	М-база	233 bytes in 56,5832 sec (4,1178 bytes/sec)	7,4268 KB in 36,655 sec (207,4753 bytes/sec)	47,7145 MB in 40,6875 sec (1,1727 MB/sec)	21,6282 MB in 40,8707 sec (541,8872 KB/sec)	7,0659 MB in 39,5649 sec (182,8757 KB/sec)	20,7559 KB in 35,3561 sec (601,1412 bytes/sec)	82,2256 MB in 43,0105 sec (1,9118 MB/sec)
	В-база	233 bytes in 41,9235 sec (5,5577 bytes/sec)	7,4268 KB in 76,6164 sec (99,2608 bytes/sec)	203,9403 MB in 59,5601 sec (3,4241 MB/sec)	76,9424 MB in 80,4052 sec (979,8994 KB/sec)	24,3927 MB in 47,7911 sec (9522,6507 KB/sec)	20,7559 KB in 44,8828 sec (473,5443 bytes/sec)	296,8305 MB in 66,1472 sec (4,4874 MB/sec)

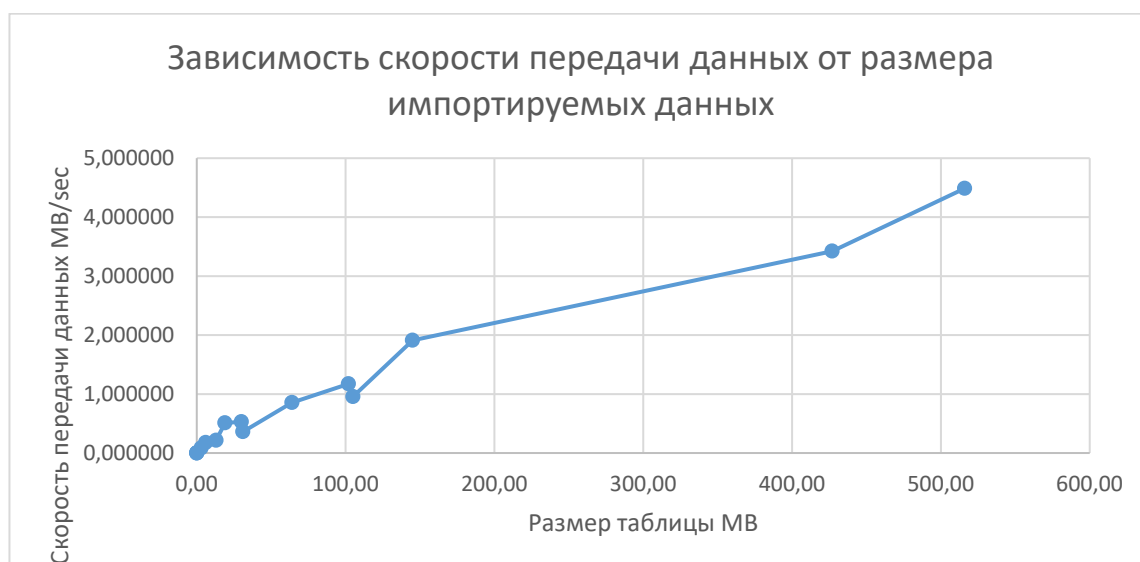


Рис. 1. Зависимость скорости передачи данных от размера импортируемых данных

В таблице 8 приведены значения скорости передачи данных для разных таблиц демонстрационной базы данных. По данным таблицы построен сводный график (рис. 1). По оси X располагается значение размера таблицы в MB, по оси Y располагается скорость передачи данных, измеряемая в MB в секунду. На основе анализа полученного графика сделаны следующие выводы:

- при увеличении размера импортируемых данных, скорость передачи данных возрастает;
- график зависимости схож с графиком зависимости степенной функции $y = x^{1/2}$;
- значительный рост скорости наблюдается при импорте данных размером от 105 до 516 MB, скорость возрастает от 2 до 4,5 MB/sec. Для таблиц меньшего размера скорость передачи колеблется в промежутке между 0,000005 и 2 MB/sec;
- предполагается, что при передаче данных большего размера, скорость передачи будет расти, но с меньшей интенсивностью; для определения точных данных, необходимо большее количество экспериментов.

Выводы

По результатам исследования сделаны следующие общие выводы:

- для сравнительно малых размеров данных (таких как в S-базе и M-базе) время передачи меняется незначительно, в то время как для B-базы время передачи в 1,5 раза больше, что является следствием того, что объем передаваемых данных вырос в 3-4 раза по сравнению с M-базой;
- распределение ресурсов для таблиц S-базы и M-базы примерно равны, для B-базы показатели возрастают в 1,5-2 раза;
- значение параметров, относящихся к группе Job Counters, для таблиц S- и M-базы примерно равны, для B-базы показатели возрастают в 2-3 раза;
- из группы Map-Reduce Framework рассмотрен параметр CPU-time spent. Значение параметра для таблиц S- и M-базы примерно равны, для B-базы показатели возрастают в 2-2,5 раза;

– при увеличении размера импортируемых данных, скорость передачи данных возрастает. График зависимости схож с графиком зависимости степенной функции $y = x^{1/2}$. Значительный рост скорости наблюдается при импорте данных размером от 105 до 516 MB, для данных меньшего размера скорость передачи данных изменяется незначительно.

Просуммировав результаты, можно сказать, что для данных небольших размеров (до 100 MB) значение параметров оперативности импорта примерно одинаковы, но при возрастании размеров импортированных данных показатели возрастают. При увеличении размера данных в 3-4 раза, значения параметров вырастают в 2-3 раза. Такая зависимость прослеживается для всех выбранных параметров оперативности передачи данных.

Для более точных выводов и прогнозирования результатов необходимо провести исследования на наборах данных большего размера.

О ПРОГНОЗНОМ РАСЧЕТЕ ОПЕРАТИВНОСТИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Полученные результаты экспериментов могут быть использованы для прогнозного расчета характеристик оперативности обработки больших данных. В частности, такой прогноз можно провести на основе использования полученных в настоящей статье характеристик оперативности в качестве обучающей выборки для нейронной сети по аналогии с [11].

В качестве еще одного широко распространенного способа получения прогнозных оценок оперативности обработки больших данных можно отметить численные модели и методы теории массового обслуживания [12, 13]. Среди них можно выделить численные методы, основанные на применении распределений фазового типа для исследования многоканальных моделей систем массового обслуживания (СМО) с разогревом и охлаждением [14, 15].

На начальном этапе для оценивания оперативности обработки больших данных производится формирование исходных данных: определяются характеристики центра обработки больших данных по числу параллельных каналов,

производительности каналов, интенсивности поступления запросов на обработку данных. Применительно к архитектуре Map-Reduce, реализующей концепцию распределенных параллельных вычислений (рис. 2), можно воспользоваться моделями многоканальных СМО или моделями сетей массового обслуживания (СМО).

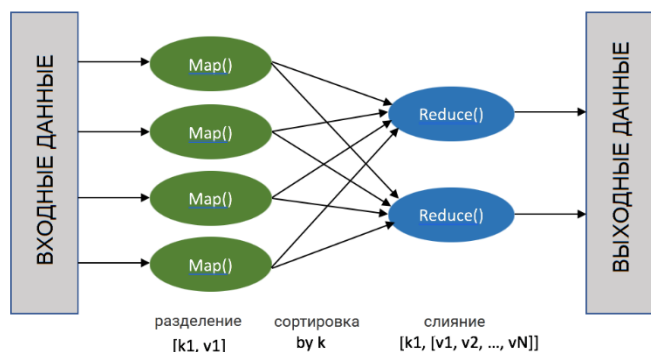


Рис. 2. Архитектура Map-Reduce

Для расчета характеристик оперативности с помощью модели многоканальной СМО осуществляется составление и решение системы уравнений баланса на уровне средних интенсивностей потоков заявок. Затем производится построение n-канальной модели СМО GI/G/PH/PH/n с «разогревом» и/или «охлаждением». В модели СМО такого типа подразумевается рекуррентный входящий поток и произвольное распределение длительности обслуживания заявок, а также произвольное распределения длительности «разогрева» и/или «охлаждения», аппроксимируемые с помощью одного из распределений фазового типа (Эрланга, гиперэкспоненциального, Кокса). Расчет характеристик таких СМО подразумевает составление векторно-матричных уравнений баланса переходов между состояниями вида

$$\begin{aligned} \gamma_0 D_0 &= \gamma_0 C_0 + \gamma_1 B_1, \\ \gamma_j D_j &= \gamma_{j-1} A_{j-1} + \gamma_j C_j + \gamma_{j+1} B_{j+1}, \\ j &= 1, 2, \dots \end{aligned}$$

Здесь A, B, C, D — матрицы интенсивностей переходов между микросостояниями модели многоканальной СМО в стационарном режиме, j — номер яруса диаграммы переходов, соответствующий числу заявок в системе в стационарном режиме, γ — стационарное распределение вероятностей микросостояний системы.

Решение указанных уравнений основано на реализации расчетной схемы Гаусса — Зейделя и использовании итерационного алгоритма Такахаша — Таками. Характеристики модели многоканальной СМО рассчитываются и уточняются в соответствии с алгоритмом [12, 13].

Еще ряд подходов к прогнозному расчету характеристик оперативности обработки больших данных приводится в работах [16–18]. Достоинством предлагаемых авторами подходов является учет особенностей архитектуры Map-Reduce, часто используемой при обработке больших данных. Предложены методы расчета длительности обработки исходной заявки в многоканальной системе массового обслуживания с учетом разделения заявки на независимые подзадачи и их параллельной обработки с последующим объединением результатов. Длительность

указанного процесса представляется как распределение максимума случайных длительностей выполнения подзадач. В частности, в [16] получено точное решение по определению максимума общего времени обслуживания независимыми каналами с экспоненциальным распределением длительности и различной интенсивностью, а также аппроксимации для произвольного распределения.

В работе [17] результирующее распределение получено в матрично-экспоненциальной форме и позволяет вычислить ряд начальных моментов высших порядков. Этот метод характеризуется высокой вычислительной сложностью. Разработанный в [18] метод позволяет рассчитать начальные моменты распределения максимума случайных величин и может использоваться для определения времени обслуживания заявки в СМО с учетом процессов Split-Join (распараллеливания-объединения). Метод обладает сравнительно невысокой вычислительной сложностью при достаточной для практического применения точности получаемого решения.

Расчет показателей оперативности перемещения больших данных в различных режимах функционирования информационной инфраструктуры целесообразно выполнять для оптимального аварийного восстановления хранилищ данных и обеспечения требуемой киберустойчивости цифровых платформ в целом. Примеры постановок и решения подобного типа задач подробно рассматриваются в работах [19, 20].

ЗАКЛЮЧЕНИЕ

Полученные в процессе исследования результаты можно использовать на этапе проектирования информационных систем и баз данных, реализующих хранение и обработку больших данных. В частности, с их помощью можно получить прогнозные оценки оперативности процессов передачи хранимой и обрабатываемой информации. Здесь могут быть использованы, например, решения с помощью нейронных сетей, а также с помощью методов и моделей исследования систем сетей массового обслуживания. Расчет оперативности перемещения больших данных в различных режимах функционирования информационной инфраструктуры целесообразно выполнять для оптимального аварийного восстановления хранилищ данных и обеспечения требуемой киберустойчивости цифровых платформ.

ЛИТЕРАТУРА

1. Reinsel D. The Digitization of the World — From Edge to Core / D. Reinsel, J. Gantz, J. Rydning // IDC Whitepaper. No. US44413318. November 2018. 28 p. URL: <http://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf> (дата обращения 10.06.2020).
2. Pehcevski J. Big Data Analytics: Methods and Applications. — Oakville, Canada: Arcler Press, 2019. — 428 p.
3. Data Processing in High-Performance Computing Systems / A. Adamov, A. I. Buranbaeva, S. I. Gindin, et al. // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: Сб. материалов VI Международной научно-практической конференции (Республика Беларусь, Минск, 20–21 мая 2020 г.): в 3-х ч. — Ч. 1. — Минск: Бестпринт, 2020. — С. 33–52.
4. PostgreSQL — объектно-реляционная система управления базами данных — 31.07.2019 // WEB Creator.

URL: <http://web-creator.ru/articles/postgresql> (дата обращения 10.06.2020).

5. Зимовец А. И., Хомоненко А. Д. Обоснование выбора модели хранения данных для системы мониторинга космического пространства // Автоматика на транспорте. 2019. Т. 5, № 2. С. 221–232.

DOI: 10.20295/2412-9186-2019-5-2-221-232.

6. Гайкова О. В. Модели баз данных NoSQL / О. В. Гайкова, В. В. Рогальчук, А. Д. Хомоненко // Модели и методы исследования информационных систем: Монография / А. Д. Хомоненко, В. П. Бубнов, А. В. Забродин [и др.]; под ред. А. Д. Хомоненко. — СПб.: Лань, 2019. — С.25–30. — 204 с. — (Учебники для вузов. Специальная литература).

7. Fundamentals of Apache Sqoop // Hadoop Sqoop Tutorial / DeZyre — Online Training Courses, Certification From Industry Experts. URL: <http://www.dezyre.com/hadoop-tutorial/hadoop-sqoop-tutorial> (дата обращения 10.06.2020).

8. Performance Monitoring, Testing and Optimizing Hadoop-MapReduce Job using Hadoop Counters — 21.10.2015 // Hadoop Mania. URL: <http://hadoopmania.blogspot.com/2015/10/performance-monitoring-testing-and.html> (дата обращения 10.06.2020).

9. Dean J. MapReduce: Simplified Data Processing on Large Clusters / J. Dean, S. Ghemawat // Communication of the ACM. 2008. Vol. 51, Is. 1. Pp. 107–113.

DOI: 10.1145/1327452.1327492.

10. Демонстрационная база данных // Компания Postgres Professional. URL: <http://postgrespro.ru/education/demodb> (дата обращения 10.06.2020).

11. Integration of Big Data Processing Tools and Neural Networks for Image Classification / N. E. Kosykh, A. D. Khomonenko, A. P. Bochkov, A. V. Kikot // Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019). CEUR Workshop Proceedings. 2020. Vol. 2556. Pp. 52–58.

12. Рыжиков Ю. И. Алгоритмический подход к задачам массового обслуживания: Монография. — СПб: ВКА им. А. Ф. Можайского, 2013. — 496 с.

13. Хомоненко А. Д. Численные методы анализа систем и сетей массового обслуживания. — М.: Министерство обороны СССР, 1991. — 197 с.

14. Khalil M. M. Load Balancing Cloud Computing with Web-Interface Using Multi-Channel Queuing Systems with

Warming Up and Cooling / M. M. Khalil, A. D. Khomonenko, S. I. Gindin // Intelligent Distributed Computing XIII: Proceedings of International Symposium on Intelligent and Distributed Computing (IDC 2019), (Saint Petersburg, Russia, October 7–9, 2019) / I. Kotenko (eds), et al. // Studies in Computational Intelligence. 2020. Vol. 868. Pp. 385–393.

DOI: 10.1007/978-3-030-32258-8_45.

15. Khomonenko A. D. Stochastic Models for Cloud Computing Performance Evaluation / A. D. Khomonenko, S. I. Gindin // Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia (CEE-SECR'14), (Moscow, Russia, October 23–25, 2014).

DOI: 10.1145/2687233.2687256.

16. Harrison P. G. Queueing Models with Maxima of Service Times / P. G. Harrison, S. Zertal // Computer Performance Evaluation. Modelling Techniques and Tools: Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS 2003), (Urbana, IL, USA, September 2–5, 2003) / P. Kemper, W. H. Sanders (eds.) // Lecture Notes in Computer Science, Vol. 2794. Pp. 152–168. DOI: 10.1007/978-3-540-45232-4_10.

17. Fiorini P. M. Exact Analysis of Some Split-Merge Queues / P. M. Fiorini, L. Lipsky // ACM SIGMETRICS Performance Evaluation Review. 2015. Vol. 43, No. 2. Pp. 51–53. DOI: 10.1145/2825236.2825257.

18. Рыжиков Ю. И. Метод расчета длительности обработки задач в системе массового обслуживания с учетом процессов Split-Join / Ю. И. Рыжиков, В. А. Лохвицкий, Р. С. Хабаров // Известия высших учебных заведений. Приборостроение. 2019. Т. 62, № 5. С. 419–423.

DOI: 10.17586/0021-3454-2019-62-5-419-423.

19. Воробьев Е. Г. Математические модели управления системой обеспечения доступности информации и оценки качества ее функционирования // Научные технологии в космических исследованиях Земли. 2019. Т. 11, № 2. С. 51–62. DOI: 10.24411/2409-5419-2018-10259.

20. Petrenko S. A. Method of Ensuring Cyber Resilience of Digital Platforms Based on Catastrophe Theory / S. A. Petrenko, D. E. Vorobieva // Proceedings of XXII International Conference on Soft Computing and Measurements (SCM 2019), (Saint Petersburg, Russia, May 23–25, 2019). Pp. 97–101.

DOI: 10.1109/SCM.2019.8903658.

Estimation of Efficiency of Transfer of Big Data on the Example of PostgreSQL Database, Hadoop Platform and Sqoop System

Master of Science O. V. Badanina, PhD S. I. Gindin
Emperor Alexander I Petersburg
State Transport University
Saint Petersburg, Russia
olgagaykova@gmail.com, sgindin@gmail.com

Grand PhD A. D. Khomonenko
A. F. Mozhaisky Military Space Academy,
Emperor Alexander I Petersburg
State Transport University
Saint Petersburg, Russia
khomon@mail.ru

Abstract. The evaluation of the characteristics of the efficiency of large data transmission is considered. To do this, use the PostgreSQL demo database as a data source; Hadoop platform, as a data storage tool, as well as a Sqoop data transfer tool. The experimental results demonstrate a change in the parameters characterizing the efficiency of data transfer during import, depending on their volume and type. The obtained experimental results can be used for the predictive calculation of the performance characteristics of big data processing. Such a forecast can be made on the basis of using the obtained characteristics of efficiency as a training sample for a neural network. In addition, it is advisable to forecast the speed of processing big data using models of multichannel queuing systems with heating and cooling, as well as queuing networks. It is advisable to calculate the efficiency indicators for moving big data in various modes of functioning of the information infrastructure for optimal disaster recovery of data warehouses and ensuring the required cyber stability of digital platforms in general.

Keywords: big data, data transfer efficiency, PostgreSQL DBMS, Hadoop platform, Sqoop data transfer tool, Map-Reduce software platform.

REFERENCES

1. Reinsel D., Gantz J., Rydning J. The Digitization of the World — From Edge to Core, *IDC Whitepaper*, No. US44413318, November 2018, 28 p. Available at: <http://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-data-age-whitepaper.pdf> (accessed 10 June 2020).
2. Pehcevski J. *Big Data Analytics: Methods and Applications*. Oakville, Canada, Arcler Press, 2019, 428 p.
3. Adamov A., Buranbaeva A. I., Gindin S. I., et al. Data Processing in High-Performance Computing Systems, *BIG DATA and Advanced Analytics: Proceedings of the VI International Scientific and Practical Conference [BIG DATA i analiz vysokogo urovnya: Sbornik materialov VI Mezhdunarodnoy nauchno-prakticheskoy konferentsii]*, Belarus, Minsk, May 20–21, 2020, Chapter 1, Minsk, Best Print Publishers, 2020, Pp. 33–52.
4. PostgreSQL — object-relational database management system [PostgreSQL — ob"ektno-relyatsionnaya sistema upravleniya bazami dannykh], *WEB Creator*. Published at 31 July 2019. Available at: <http://web-creator.ru/articles/postgresql> (accessed 10 June 2020).
5. Zimovets A. I., Khomonenko A. D. The Rationale for Choosing a Data Storage Model for a Space Monitoring System, *Automation on Transport*, 2019, Vol. 5, No. 2, Pp. 221–232. DOI: 10.20295/2412-9186-2019-5-2-221-232.
6. Gaikova O. V., Rogalchuk V. V., Khomonenko A. D. NoSQL Database Models [Modeli baz dannykh NoSQL]. In: *Khomonenko A. D., Bubnov V. P., Zabrodin A. V., et al. Models and methods of research of information systems: Monograph [Modeli i metody issledovaniya informatsionnykh sistem: Monografiya]*, St. Petersburg, LAN Publishing House, 2019, Pp. 25–30.
7. Fundamentals of Apache Sqoop, *DeZyre — Online Training Courses, Certification From Industry Experts*. Available at: <http://www.dezyre.com/hadoop-tutorial/hadoop-sqoop-tutorial> (accessed 10 June 2020).
8. Performance Monitoring, Testing and Optimizing Hadoop-MapReduce Job using Hadoop Counters, *Hadoop Mania*. Published at October 21, 2015. Available at: <http://hadoop-mania.blogspot.com/2015/10/performance-monitoring-testing-and.html> (accessed 10 June 2020).
9. Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters, *Communication of the ACM*, 2008, Vol. 51, Is. 1, Pp. 107–113. DOI: 10.1145/1327452.1327492.
10. Demonstration database [Demonstratsionnaya baza-dannykh], *Postgres Professional [Kompaniya Postgres Professional]*. Available at: <http://postgrespro.ru/education/demodb> (accessed 10 June 2020).
11. Kosykh N. E., Khomonenko A. D., Bochkov A. P., Kikot A. V. Integration of Big Data Processing Tools and Neural Networks for Image Classification, *Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019), CEUR Workshop Proceedings*, 2020, Vol. 2556, Pp. 52–58.
12. Ryzhikov Yu. I. Algorithmic approach to queuing tasks: Monograph [Algoritmicheskiy podkhod k zadacham massovogo obsluzhivaniya: Monografiya], Saint Petersburg, A. F. Mozhaisky Military Space Academy, 2013, 496 p.
13. Khomonenko A. D. Numerical methods for analysis of queuing systems and networks [Chislennyye metody analiza sistem i setey massovogo obsluzhivaniya], Moscow, Ministry of Defense of the USSR, 1991, 197 p.

14. Khalil M. M., Khomonenko A. D., Gindin S. I. Load Balancing Cloud Computing with Web-Interface Using Multi-Channel Queuing Systems with Warming Up and Cooling. In: *Kotenko I. (eds), et al. Intelligent Distributed Computing XIII: Proceedings of International Symposium on Intelligent and Distributed Computing (IDC 2019), Saint Petersburg, Russia, October 7–9, 2019. Studies in Computational Intelligence*, 2020, Vol. 868, Pp. 385–393.

DOI: 10.1007/978-3-030-32258-8_45.

15. Khomonenko A. D., Gindin S. I. Stochastic Models for Cloud Computing Performance Evaluation, *Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia (CEE-SECR'14), Moscow, Russia, October 23–25, 2014*. DOI: 10.1145/2687233.2687256.

16. Harrison P. G., Zertal S. Queuing Models with Maxima of Service Times. In: *Kemper P., Sanders W. H. (eds.) Computer Performance Evaluation. Modelling Techniques and Tools: Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS 2003), Urbana, IL, USA, September 2–5, 2003. Lecture Notes in Computer Science*, Vol. 2794, Pp. 152–168.

DOI: 10.1007/978-3-540-45232-4_10.

17. Fiorini P. M., Lipsky L. Exact Analysis of Some Split-Merge Queues, *ACM SIGMETRICS Performance Evaluation Review*, 2015, Vol. 43, No. 2, Pp. 51–53.

DOI:10.1145/2825236.2825257.

18. Ryzhikov Yu. I., Lokhvitsky V. A., Khabarov R. S. Method of Calculating Task Treatment Duration in Queueing System with Consideration of Split-Join Processes [Metod rascheta dlitel'nosti obrabotki zadach v sisteme massovogo obsluzhivaniya s uchetom protsessov Split-Join], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 2019, Vol. 62, No. 5, Pp. 419–423.

DOI: 10.17586/0021-3454-2019-62-5-419-423.

19. Vorobiev E. G. Mathematical Models of Ensuring Information Availability Management System and Quality Assessment of Its Functioning [Matematicheskie modeli upravleniya sistemoy obespecheniya dostupnosti informatsii i otsenki kachestva ee funktsionirovaniya], *High Technologies in Earth Space Research [Naukoemkie tekhnologii v kosmicheskikh issledovaniyakh Zemli]*, 2019, Vol. 11, No. 2, Pp. 51–62.

DOI: 10.24411/2409-5419-2018-10259.

20. Petrenko S. A., Vorobieva D. E. Method of Ensuring Cyber Resilience of Digital Platforms Based on Catastrophe Theory, *Proceedings of XXII International Conference on Soft Computing and Measurements (SCM 2019), Saint Petersburg, Russia, May 23–25, 2019*, Pp. 97–101.

DOI:10.1109/SCM.2019.8903658.

Сравнение моделей и методов интеллектуальной обработки геопространственных данных на основе семантической технологии

аспирант А. В. Кикоть

Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
a.v.kikot@yandex.ru

Аннотация. Выполнено исследование современных методов интеллектуальной обработки геопространственных данных, сделан вывод о преобладании семантической технологии. Рассмотрены и оценены модель с предположением инъекции, модель с предположением сюръекции и модель на основе достижения инъекции множества GeoSPARQL-запросов на множество SQL-запросов. Сделан вывод о необходимости применения комплексного подхода на основе виртуализации данных.

Ключевые слова: SPARQL, GeoSPARQL, Linked Data, СУБД, геопространственные данные, инфраструктура геопространственных данных.

ВВЕДЕНИЕ

Современная наука накопила большое количество различных методов интеллектуальной обработки данных. Наиболее активным направлением сейчас является применение методов машинного обучения, лежащих в парадигме нейросетевого моделирования, например [1]. Также активно изучаются методы автоматического выявления зависимостей, позволяющих получать новые факты на основе больших массивов информации. Основу математического аппарата данных методов составляет теория вероятностей, статистика, теория массового обслуживания, методы математического программирования и другие.

Вместе с тем существует альтернативный подход, основывающийся на теории множеств, математической логике, теории графов и топологии — семантическая технология и технология связанных данных (Linked Data).

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА ГЕОПРОСТРАНСТВЕННЫХ ДАННЫХ

Поисковый анализ этого направления по индексируемым отечественным и зарубежным научным исследованиям с глубиной 5 лет определил 88 целевых ключевых слов общей частотностью применения — 257 185. Во время поиска выявленные ключевые слова после исключения дублирования дополнительно фильтровались с применением Закона рассеяния информации Брэдфорда в соответствии с [2]. То есть, совокупность ключевых слов, ранжированных по убыванию частотности, разбивалась на три группы таким образом, чтобы соблюдалось условие

$$\sum D_{I_i} = \sum D_{II_i} = \sum D_{III_i}, \quad (1)$$

где D_i — частотность ключевого слова, а D_{I_i} , D_{II_i} , D_{III_i} — частотность ключевых слов, попавших в I, II и III группы соответственно. Результат кластеризации приведен на рисунке 1.

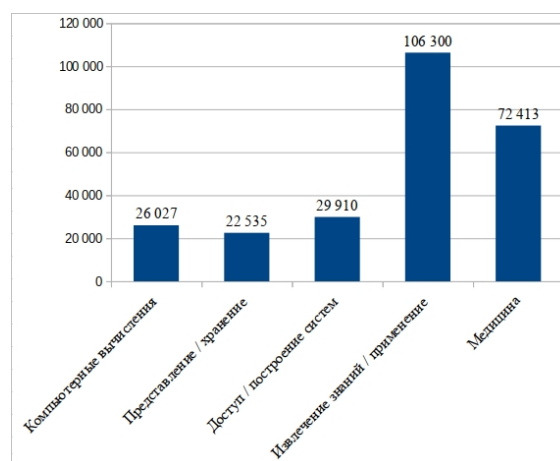


Рис. 1. Перечень групп отобранных ключевых слов

Группы, вносящие наибольший вес, показывают высокую заинтересованность бизнеса, государства и особенно медицины в области интеллектуальной обработки данных. С точки зрения обработки геопространственных данных (ГПД) они не представляют интерес. Для оставшихся групп распределение частотности ключевых слов приведено на рисунке 2.

Данная диаграмма может быть интерпретирована следующим образом: в настоящее время наибольший интерес представляет семантическая обработка текстовых данных с хранением в базах данных, обеспечением соответствующего языка запросов и конкурентного доступа к ним.

Исследования с одновременным применением ключевых слов *geospatial*, *model*, *query*, *semantic* и *database* в настоящее время обладают средней частотностью. Результаты научных исследований за рубежом по тематике интеллектуальной обработки ГПД в основном могут быть отнесены к одной из следующих четырех групп:

- применение интеллектуальной обработки ГПД для тематических задач, например [3–7];
- разработка онтологий представления ГПД, например [8];

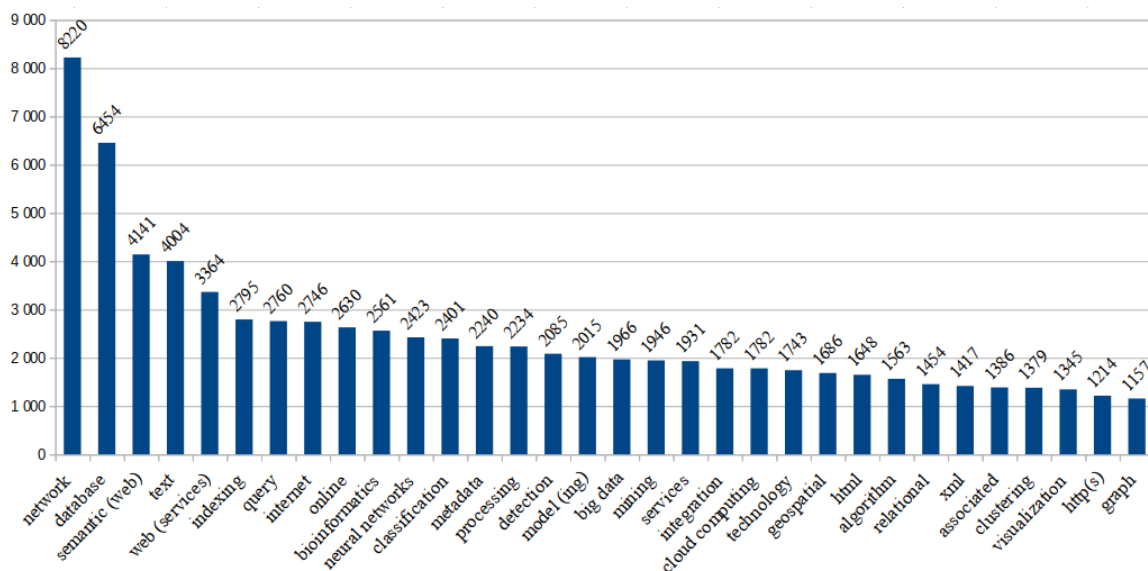


Рис. 2. Частоты употребления ключевых слов в области интеллектуальной обработки данных

- разработка архитектуры информационных систем для обработки ГПД, например [9–11];
- разработка механизмов хранения ГПД и их тестирования, например [12–15].

В обобщенном виде с множеством допущений приведенные работы формируют набор сценариев интеллектуальной обработки ГПД, приведенный на рисунке 3.

В России также выполняются исследования в обсуждаемой предметной области с аналогичным распределением работ по вышеописанным группам. В частности, архитектурные вопросы рассматриваются в [16–18], а разработка онтологий — в работах [19, 20]. Ряд работ посвящен конкретным изделиям с уже реализованной интеллектуальной обработкой ГПД (например система «ГеоМета» [21, 22]).

ОБЩАЯ МОДЕЛЬ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ГПД

Ограничиваясь неполной индукцией для приведенных выше научных исследований, можно сделать ключевой вывод: современные и перспективные методы интеллектуальной обработки данных строятся на парадигме

OBDA (Ontology Based Data Access), заключающейся в построении дополнительного концептуального уровня абстракции над данными (таблица 1), позволяющего манипулировать ими при выборке.

Таблица 1

Уровни абстракции над данными

Уровень абстракции	Вариант метода реализации	Пример реализации
M2 – Метамодел	Онтологическая модель	Поддержка SPARQL-запросов с отображением на реляционную схему
M1 – Модель	Реляционная модель	Поддержка реляционной схемы в СУБД и языка запросов SQL к ней
M0 – Данные	База данных	Разработка СУБД

Формально OBDA-система задается тройкой следующего вида:

$$O = \langle T, S, M \rangle, \tag{2}$$



Рис. 3. Основные подходы к обработке GeoSPARQL-запросов

где T — интенсиональный уровень онтологии, в терминах дескрипционной логики — терминологический компонент (TBox);

S — реляционная база данных, в рамках которой реализуется хранение и обработка целевых данных;

M — набор правил преобразования, таких, что для каждого правила выполняется $F(x) \leftarrow \Psi(x)$, где $F(x)$ — запрос из S , возвращающий кортежи значений для x , а $\Psi(x)$ — запрос из T , сформированный на основе x .

Таким образом, любая реализация OBDA-системы заключается в последовательном переходе с заданного уровня абстракции к уровню данных. То есть интеллектуальный запрос сначала семантически, а затем лексически преобразуется к запросу, принимаемому на вход системой управления базами данных. Пример такого преобразования для SPARQL-запроса с привязкой по обозначениям к формуле (2) приведен на рисунке 4.

Вывод 1. Интеллектуальность обработки геопространственных данных должна достигаться применением языка запросов SPARQL и, в частности, его пространственного расширения GeoSPARQL.

Продолжая рассматривать обсуждаемую систему в качестве «черного ящика», необходимо определиться с механизмом хранения данных. Анализ современных решений по манипуляции данными [23] и способов организации баз знаний (в частности, проекты SAP OrientDB, Neo4j, Allegro Graph, DataStax, ArangoDB, TigerGraph, Cambridge Semantics, Stardog и ряд других) позволяет сформировать перечень применяемых механизмов хранения, приведенных на рисунке 5.

Одновременно со сказанным следует принимать во внимание доступность механизмов централизованного накопления и обработки данных, содержащих пространственную компоненту. Безусловно, одним из наиболее развитых и доступных для детального анализа решением



Рис. 4. Пример преобразования SPARQL-запроса

является объектная надстройка PostGIS для реляционной системы управления базами данных PostgreSQL. PostGIS, как расширение, внедренное в огромное количество публичных и коммерческих проектов, указывает на ключевую роль объектно-реляционной схемы данных для задачи хранения и обработки ГПД.

Вывод 2. В качестве механизма хранения должна применяться объектно-реляционная схема данных, а разрабатываемая система должна поддерживать векторно-топологический метод представления ГПД.

Объединение выводов 1 и 2 позволяет перейти к моделированию с применением UML. В частности, обобщенное описание системы интеллектуальной обработки ГПД в терминах прецедентов использования примет вид, приведенный на рисунке 6.

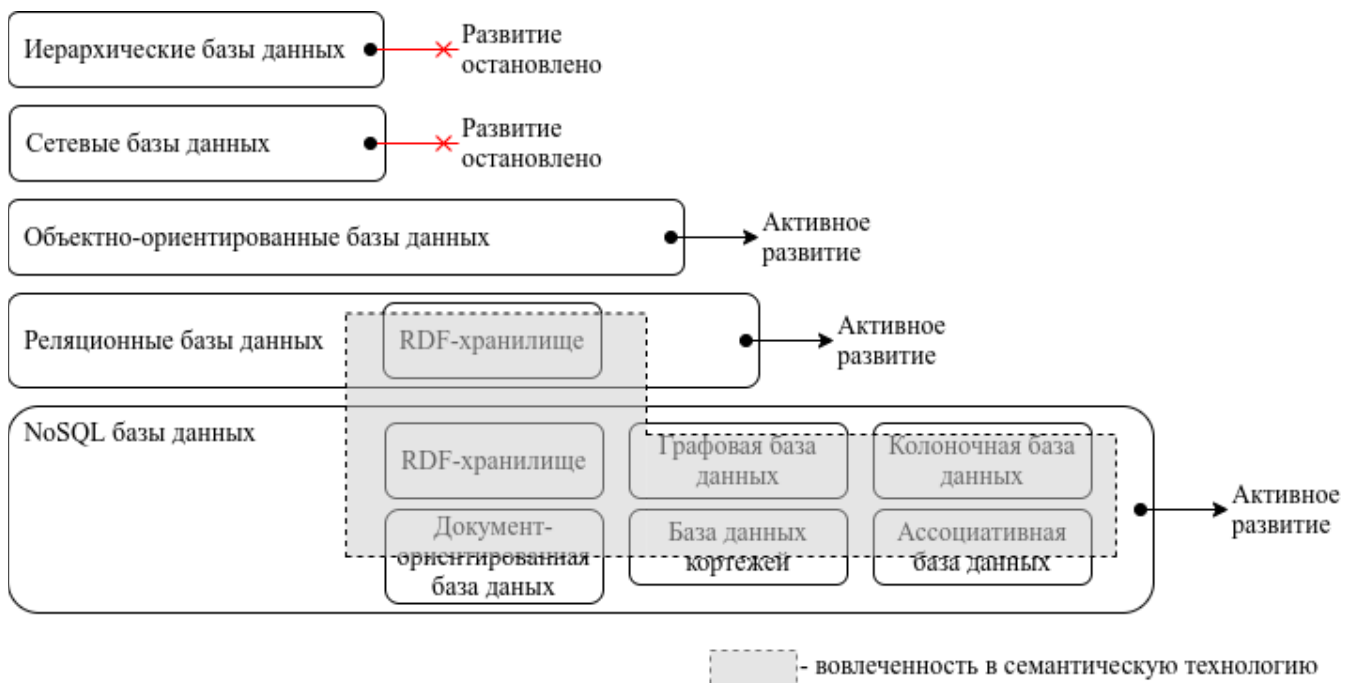


Рис. 5. Применяемость СУБД для организации систем хранения знаний



Рис. 6. Диаграмма прецедентов использования для системы интеллектуальной обработки ГПД

ВАРИАНТЫ РЕАЛИЗАЦИИ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ГПД

Прецедент «Взаимодействие» на рисунке 6, скрывающий в себе всю сложность системы, в соответствии с рисунком 3 может быть декомпозирован несколькими базовыми вариантами, приведенными на рисунке 7.

Очевидно, что «Взаимодействие» должно завершаться успешно проходящим все необходимые проверки SQL-запросом. Наиболее простой «Вариант 1» возмо-

жен только для заранее определенного перечня исходных SPARQL-запросов, так как прямое отображение между двумя этими языками без значительных затрат невозможно [24]. Поэтому данный вариант нежелателен.

«Вариант 2» в общем виде подразумевает трансформацию SPARQL-запроса во множество SQL-запросов, каждый из которых генерируется и выполняется по мере необходимости. Такое решение позволит поддерживать большее количество типов SPARQL-запросов, но сложно формализуемо и проверяемо для вложенных запросов [25].

В «Варианте 3» применяется внешняя для целевой база данных с предварительно приспособленной для запросов RDF-триплетов структурой. Такая архитектура позволит поддерживать все множество SPARQL-запросов, но со значительным дублированием данных. Отдельная вычислительно емкая задача будет заключаться в поддержании механизма ACID (атомарность, согласованность, изолированность и стойкость).

Рассматривая совокупность возможных конструкций GeoSPARQL и SQL в качестве множеств, приведенные три варианта могут быть представлены следующим образом:

Обозначим SP — множество всех языковых конструкций SPARQL, а SQ — множество всех языковых конструкций SQL. В таком случае, искомым является функция отображения вида

$$f: SP \rightarrow SQ. \tag{3}$$

В соответствии с (3) приведенные на рисунке 7 варианты могут рассматриваться с применением одного из трех подходов.

Подход 1 — поиск инъекции («Вариант 1»): предполагаем $SP \subset SQ = true$, тогда искомое отображение f для SPARQL-запроса q должно удовлетворять условию вида:

$$\forall q_1, q_2 \in SP, q_1 \neq q_2, \exists q'_1, q'_2 \in SQ : f(q_1) = q'_1, f(q_2) = q'_2, q'_1 \neq q'_2 \tag{4}$$

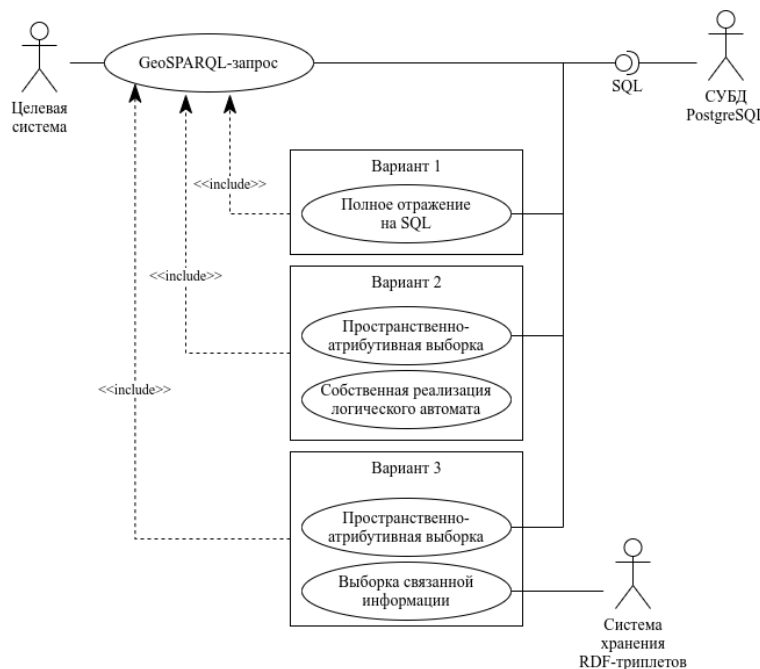


Рис. 7. Варианты декомпозиции прецедента «Взаимодействие»

Подход 2 — поиск сюръекции («Вариант 1»): предполагаем $SQ \subset SP = true$, тогда искомое отображение f для SPARQL-запроса q должно удовлетворять условию

$$\forall q \in SP, \exists q' \in SQ : f(q') = q. \quad (5)$$

Подход 3 — достижение инъекции («Вариант 2» и «Вариант 3»): предполагаем $SP \subset SQ = false$, $SQ \subset SP = false$. Тогда вводится дополнительное множество MDL , такое, что отображение $y: SP \rightarrow MDL$ и отображение $z: MDL \rightarrow SQ$ дополняют отображение f до инъективного.

Принимая во внимание порядок обработки SPARQL-запроса [26], может быть сделан вывод о невозможности предположений $SP \subset SQ = true$ и $SQ \subset SP = true$, что приводит к невозможности «Варианта 1» целиком. «Вариант 2» и «Вариант 3» в силу вышеприведенных рассуждений сложны в реализации.

Вывод 3. Анализ возможных технических реализаций рассмотренных вариантов показывает, что наиболее производительным и расширяемым будет решение, построенное с применением комплексирования.

В качестве примера подхода с комплексированием может быть отмечена технология Ontop [27]. Наиболее важной интерфейсной особенностью данной технологии является организация полноценной GeoSPARQL-точки доступа, скрывающей от пользователя все необходимые преобразования его запроса, и в то же время за счет виртуализации данных поддерживающей на уровне хранения широкий перечень реляционных систем управления базами данных.

ВЫВОДЫ

Выполненный поиск подхода к организации системы интеллектуальной обработки ГПД позволяет сделать следующие выводы:

1. Применение семантической технологии для интеллектуальной обработки ГПД является перспективным и вычислительно достижимым.
2. Интеллектуальность может быть достигнута применением GeoSPARQL в качестве языка манипулирования данными и описания конструкций логического вывода.
3. Архитектура информационной системы с поддержкой GeoSPARQL-запросов носит комбинированный характер с применением слоя виртуализации данных, а ее выбор требует отдельного мотивированного вывода.

ЛИТЕРАТУРА

1. Integration of Big Data Processing Tools and Neural Networks for Image Classification / N. E. Kosykh, A. D. Khomonenko, A. P. Bochkov, A. V. Kikot // Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019). CEUR Workshop Proceedings. 2020. Vol. 2556. Pp. 52–58.
2. Налимов В. В. Наукометрия. Изучение науки как информационного процесса / В. В. Налимов, З. М. Мульченко. — М.: Наука. Гл. ред. физ.-мат. лит., 1969. — 192 с. — (Физико-математическая библиотека инженера).
3. A Geospatial Decision Support System to Assist Olive Growing at the Landscape Scale / P. Manna, A. Bonfante, A. Basile, et al. // Computers and Electronics in Agriculture.

2020. Vol. 168. Article 105143. 11 p. Published online at December 26, 2019. DOI: 10.1016/j.compag.2019.105143.

4. Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance / C. Ray, R. Dréo, C. Iphar, et al. // Data in Brief. 2019. Vol. 25. Article 104141. 17 p. Published online at June 11, 2019. DOI: 10.1016/j.dib.2019.104141.

5. Daras G. A Spatial Data Pre-Processing Tool to Improve the Quality of the Analysis and to Reduce Preparation Duration / G. Daras, B. Agard, B. Penz // Computers & Industrial Engineering. 2018. Vol. 119. Pp. 219–232. DOI: 10.1016/j.cie.2018.03.025.

6. Deeken H. Grounding Semantic Maps in Spatial Databases / H. Deeken, T. Wiemann, J. Hertzberg // Robotics and Autonomous Systems. 2018. Vol. 105. Pp. 146–165. DOI: 10.1016/j.robot.2018.03.011.

7. Rule-Based Topology System for Spatial Databases to Validate Complex Geographic Datasets / J. Martinez-Llario, E. Coll, M. Núñez-Andrés, C. Femenia-Ribera // Computers & Geosciences. 2017. Vol. 103. Pp. 122–132. DOI: 10.1016/j.cageo.2017.03.013.

8. Ontology-Based Data Access for Maritime Security / S. Brüggemann, K. Bereta, G. Xiao, M. Koubarakis // The Semantic Web. Latest Advances and New Domains: Proceedings of 13th International European Semantic Web Conference (ESWC 2016) (Heraklion, Greece, May 29–June 2, 2016) / H. Sack, E. Blomqvist, et al. (eds) // Lecture Notes in Computer Science. Vol. 9678. Pp. 741–757. DOI: 10.1007/978-3-319-34129-3_45.

9. Firemap: A Dynamic Data-Driven Predictive Wildfire Modeling and Visualization Environment / D. Crawl, J. Block, K. Lin, I. Altintas // Procedia Computer Science. 2017. Vol. 108. Pp. 2230–2239. DOI: 10.1016/j.procs.2017.05.174.

10. A New Methodology for the Retrieval and Evaluation of Geographic Coordinates within Databases of Scientific Plant Collections / U. R. Magdalena, L. A. Estevão da Silva, R. O. Lima, et al. // Applied Geography. 2018. Vol. 96. Pp. 11–15. DOI: 10.1016/j.apgeog.2018.05.002.

11. de Jong K. A Physical Data Model for Spatio-Temporal Objects / K. de Jong, D. Karssenbergh // Environmental Modelling and Software. 2019. Vol. 122. Article 104553. 15 p. DOI: 10.1016/j.envsoft.2019.104553.

12. GeoTriples: Transforming Geospatial Data into RDF Graphs Using R2RML and RML Mappings / K. Kyzirakos, D. Savva, I. Vlachopoulos, et al. // Journal of Web Semantics. 2018. Vol. 52–53. Pp. 16–32. DOI: 10.1016/j.websem.2018.08.003.

13. A complete Translation from SPARQL into Efficient SQL / B. Elliott, E. Cheng, C. Thomas-Ogbuji, Z. Meral Ozsoyoglu // Proceedings of the 13th International Database Engineering & Applications Symposium (IDEAS'09), (Cetraro, Calabria, Italy, September 16–18, 2009). — New York, USA, 2009. — Pp. 31–42. DOI: 10.1145/1620432.1620437.

14. STypeS: Nonrecursive Datalog Rewriter for Linear TGDs and Conjunctive Queries / S. Kikot, R. Kontchakov, S. Rapisarda, M. Zakharyashev // On the Move to Meaningful Internet Systems: Proceedings of the Confederated International Conference (OTM 2018), (Valetta, Malta, October 22–26, 2018). Part 2 / H. Panetto, C. Debruyne, et al. (eds) // Lecture Notes in Computer Science. Vol. 11230. Pp. 441–460.

DOI: 10.1007/978-3-030-02671-4_27.

15. Garbis G. Geographica: A Benchmark for Geospatial RDF Stores (Long Version) / G. Garbis, K. Kyzirakos, M. Koubarakis // The Semantic Web: Proceedings of 12th International Semantic Web Conference (ISWC 2013), (Sydney, Australia, October 21–25, 2013). Part 2 / H. Alani, L. Kagal, et al. (eds) // Lecture Notes in Computer Science. Vol. 8219. Pp. 343–359. DOI: 10.1007/978-3-642-41338-4_22.

16. Комосов Ю. А. Необходимость, сущность и пути реализации новой модели представления пространственных данных // Геодезия и картография. 2009. № 11. С. 26–32.

17. Лядова Л. Н. Подходы к описанию вертикальных и горизонтальных трансформаций метамodelей / Л. Н. Лядова, А. П. Серый, А. О. Сухов // Математика программных систем: межвузовский сборник научных статей. Выпуск 9 / под ред. А. И. Микова и Л. Н. Лядовой. — Пермь, Изд-во ПГНИУ, 2012. — С. 33–49.

18. Блискавицкий А. А. Семантика геопространственных объектов, функциональная грамматика и интеллектуальные ГИС // Известия высших учебных заведений. Геология и разведка. 2014. № 2. С. 62–69.

19. Беляков С. Л. Обновление информационной основы ГИС с использованием онтологий / С. Л. Беляков, Я. А. Коломийцев // Известия Южного федерального университета. Технические науки. 2012. № 4 (129). С. 251–255.

20. Динь Ле Дат. Разработка и реализация формальных онтологий пространственных данных и сервисов: дис. ... канд. физ.-мат. наук: 05.13.11. — Москва, 2008. — 196 с.

21. Кошкарев А. В. Инфраструктура распределенной среды хранения, поиск и преобразования пространствен-

ных данных / А. В. Кошкарев, В. М. Ряховский, В. А. Серебряков // Открытое образование. 2010. № 5. С. 61–73.

22. Среда интеграции пространственных данных «ГеоМета» / О. М. Атаева, К. А. Кузнецов, В. А. Серебряков, В. И. Филиппов // Интернет и современное общество: Сборник научных статей. Материалы XIV Всероссийской объединенной конференции (IMS-2011) (Санкт-Петербург, 12–14 октября 2011 г.). — СПб.: ООО «МультиПроджектСистемСервис», 2011. — С. 11–16.

23. Модели и методы исследования информационных систем: Монография / А. Д. Хомоненко, В. П. Бубнов, А. В. Забродин [и др.]; под ред. А. Д. Хомоненко. — СПб.: Лань, 2019. — 204 с. — (Учебники для вузов. Специальная литература).

24. A Direct Mapping of Relational Data to RDF. W3C Recommendation 27 September 2012. // W3C. URL: <http://www.w3.org/TR/rdb-direct-mapping> (дата обращения 01.06.2020).

25. R2RML: RDB to RDF Mapping Language. W3C Recommendation 27 September 2012. // W3C. URL: <http://www.w3.org/TR/r2rml> (дата обращения 01.06.2020).

26. RDF Database Systems: Triples Storage and SPARQL Query Processing / O. Curé, G. Blin (eds). — Morgan Kaufmann, 2015. — 256 p. DOI: 10.1016/C2013-0-14009-3.

27. Bereta K. Ontop-spatial: Ontop of Geospatial Databases / K. Bereta, G. Xiao, M. Koubarakis // Journal of Web Semantics: Science, Services and Agents on the World Wide Web. 2019. Vol. 58. Article 100514. DOI: 10.1016/j.websem.2019.100514.

Comparison of Models and Methods for Geospatial Data Intelligent Processing Based on Semantic Technology

PhD student A. V. Kikot
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia
a.v.kikot@yandex.ru

Abstract. The article investigates modern methods of geospatial data intelligent processing and makes a conclusion about the semantic technology predominance. A model with an injection assumption, a model with a surjection assumption, and a model based on achieving the injection of a GeoSPARQL queries set on a SQL queries set are considered and evaluated. It is concluded that it is necessary to apply a comprehensive approach based on data virtualization.

Keywords: SPARQL, GeoSPARQL, Linked Data, DBMS, geospatial data, geospatial data infrastructure.

REFERENCES

1. Kosykh N. E., Khomonenko A. D., Bochkov A. P., Kikot A. V. Integration of Big Data Processing Tools and Neural Networks for Image Classification, *Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019), CEUR Workshop Proceedings*, 2020, Vol. 2556, Pp. 52–58.
2. Nalimov V. V., Mulchenko Z. M. *Scientometrics. Study of science as an information process* [Naukometriya. Izuchenie nauki kak informatsionnogo protsesssa], Moscow, Nauka Publishers, 1969, 192 p.
3. Manna P., Bonfante A., Basile A., et al. A Geospatial Decision Support System to Assist Olive Growing at the Landscape Scale, *Computers and Electronics in Agriculture*, 2020, Vol. 168, article 105143, 11 p. Published online at December 26, 2019. DOI: 10.1016/j.compag.2019.105143.
4. Ray C., Dréo R., Iphar C., et al. Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance, *Data in Brief*, 2019, Vol. 25, article 104141, 17 p. Published online at June 11, 2019. DOI: 10.1016/j.dib.2019.104141.
5. Daras G., Agarad B., Penz B. A Spatial Data Pre-Processing Tool to Improve the Quality of the Analysis and to Reduce Preparation Duration, *Computers & Industrial Engineering*, 2018, Vol. 119, Pp. 219–232. DOI: 10.1016/j.cie.2018.03.025.
6. Deeken H., Wiemann T., Hetzberg J. Grounding Semantic Maps in Spatial Databases, *Robotics and Autonomous Systems*, 2018, Vol. 105, Pp. 146–165. DOI: 10.1016/j.robot.2018.03.011.
7. Martinez-Llario J., Coll E., Núñez-Andrés M., Femenia-Ribera C. Rule-Based Topology System for Spatial Databases to Validate Complex Geographic Datasets, *Computers & Geosciences*, 2017, Vol. 103, Pp. 122–132. DOI: 10.1016/j.cageo.2017.03.013.
8. Brüggemann S., Bereta K., Xiao G., Koubarakis M. Ontology-Based Data Access for Maritime Security. In: *Sack H., Blomqvist E. (eds) The Semantic Web. Latest Advances and New Domains: Proceedings of 13th International European Semantic Web Conference (ESWC 2016), Heraklion, Greece, May 29–June 2, 2016. Lecture Notes in Computer Science*, Vol. 9678, Pp. 741–757. DOI: 10.1007/978-3-319-34129-3_45.
9. Crawl D., Block J., Lin K., Altintas I. Firemap: A Dynamic Data-Driven Predictive Wildfire Modeling and Visualization Environment, *Procedia Computer Science*, 2017, Vol. 108, Pp. 2230–2239. DOI: 10.1016/j.procs.2017.05.174.
10. Magdalena U. R., Estevão da Silva L. A., Lima R. O., et al. A New Methodology for the Retrieval and Evaluation of Geographic Coordinates within Databases of Scientific Plant Collections, *Applied Geography*, 2018, Vol. 96, Pp. 11–15. DOI: 10.1016/j.apgeog.2018.05.002.
11. de Jong K., Karssenbergh D. A Physical Data Model for Spatio-Temporal Objects, *Environmental Modelling and Software*, 2019, Vol. 122, article 104553, 15 p. DOI: 10.1016/j.envsoft.2019.104553.
12. Kyzirakos K., Savva D., Vlachopoulos I., et al. GeoTriples: Transforming Geospatial Data into RDF Graphs Using R2RML and RML Mappings, *Journal of Web Semantics*, 2018, Vol. 52–53, Pp. 16–32. DOI: 10.1016/j.websem.2018.08.003.
13. Elliott B., Cheng E., Thomas-Ogbuji C., Meral Ozsoyoglu Z. A complete Translation from SPARQL into Efficient SQL. In: *Proceedings of the 13th International Database Engineering & Applications Symposium (IDEAS'09), Cetraro, Calabria, Italy, September 16–18, 2009*, New York, USA, 2009, Pp. 31–42. DOI: 10.1145/1620432.1620437.
14. Kikot S., Kontchakov R., Rapisarda S., Zakharyashev M. STyPeS: Nonrecursive Datalog Rewriter for Linear TGDs and Conjunctive Queries. In: *Panetto H., Debruyne C., et al. (eds) On the Move to Meaningful Internet Systems: Proceedings of the Confederated International Conference (OTM 2018), Valletta, Malta, October 22–26, 2018, Part 2. Lecture Notes in Computer Science*, Vol. 11230. Pp. 441–460. DOI: 10.1007/978-3-030-02671-4_27.
15. Garbis G., Kyzirakos K., Koubarakis M. Geographica: A Benchmark for Geospatial RDF Stores (Long Version). In: *Alani H., Kagal L., et al. (eds) The Semantic Web: Proceedings of 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21–25, 2013, Part 2. Lecture Notes in Computer Science*, Vol. 8219, Pp. 343–359. DOI: 10.1007/978-3-642-41338-4_22.
16. Komosov Yu. A. Necessity, Essence and Ways to Implement a New Model of Spatial Data Representation [Nebokhodimost', sushchnost' i puti realizatsii novoy modeli pred-

stavleniya prostranstvennykh dannyykh], *Geodesy and Cartography* [Geodeziya i kartografiya], 2009, No. 11, Pp. 26–32.

17. Lyadova L. N., Seryy A. P., Sukhov A. O. Approaches to the Description of Vertical and Horizontal Transformations and Metamodels [Podkhody k opisaniyu vertical'nykh i gorizontal'nykh transformatsiy metamodeley], *Mathematics of Software Systems: Intercollegiate Collection of Scientific Articles. Issue 9* [Matematika programmnykh sistem: mezhvuzovskiy sbornik nauchnykh statey. Vypusk 9], Perm, Perm State University, 2012, Pp. 33–49.

18. Bliskavitsky A. A. The Semantics of The Geospatial Objects, Functional Grammar and Intellectual GIS [Semantika geoprostranstvennykh ob'ektov, funktsional'naya grammatika i intellektual'nye GIS], *Proceedings of Higher Educational Establishments. Geology and Exploration* [Izvestiya vysshikh uchebnykh zavedeniy. Geologiya i razvedka], 2014, No. 2, Pp. 62–69.

19. Beliacov S. L., Kolomiytsev Ya. A. Updating of the Information Base of the GIS Using Ontologies [Obnovlenie informatsionnoy osnovy GIS s ispolzovaniem ontologiy], *Izvestiya Southern Federal University. Engineering Science* [Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskies nauki], 2012, No. 4 (129), Pp. 251–255.

20. Dinh Le Dat Metadata Model Ontology for Geospatial Data and Services [Razrabotka i realizatsiya formal'nykh ontologiy prostranstvennykh dannyykh i servisov]: diss. on competition of a scientific degree Ph.D. (Physico-math.), Moscow, 2008, 196 p.

21. Koshkarev A. V., Ryakhovskiy V. M., Serebryakov V. A. Infrastructure of Distributed Environment of Spatial Data Storage, Search and Processing [Infrastruktura raspredelennoy sredy khraneniya, poisk i preobrazovaniya prostranstvennykh

dannyykh], *Open Education* [Otkrytoe obrazovanie], 2010, No. 5, Pp. 61–73.

22. Ataeva O. M., Kuznetsov K. A., Filippov V. I., Serebriakov V. A. Spatial Data Integration Environment «GeoMeta» [Sreda integratsii prostranstvennykh dannyykh «GeoMeta»], «*Internet and Modern Society*»: *Proceedings of the XIV All-Russian Scientific Conference* [Internet i sovremennoe obshchestvo: Sbornik nauchnykh statey. Materialy XIV Vserossiyskoy ob'edinennoy konferentsii] (IMS-2011), Saint Petersburg, October 12–14, 2011, Saint Petersburg, MultiProjectSystemService LLC, 2011, Pp. 11–16.

23. Khomonenko A. D., Bubnov V. P., Zabrodin A.V., et al. Models and methods of research of information systems: Monograph [Modeli i metody issledovaniya informatsionnykh system: Monografiya], St. Petersburg, LAN Publishing House, 2019, 204 p.

24. A Direct Mapping of Relational Data to RDF. W3C Recommendation 27 September 2012.

Available at: <http://www.w3.org/TR/rdb-direct-mapping> (accessed 01 Jun 2020).

25. R2RML: RDB to RDF Mapping Language. W3C Recommendation 27 September 2012.

Available at: <http://www.w3.org/TR/r2ml> (accessed 01 June 2020).

26. Curé O., Blin G. (eds) RDF Database Systems: Triples Storage and SPARQL Query Processing, Morgan Kaufmann, 2015, 256 p. DOI: 10.1016/C2013-0-14009-3.

27. Bereta K., Xiao G., Koubarakis M. Ontop-spatial: Ontop of Geospatial Databases, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2019, Vol. 58, article 100514.

DOI: 10.1016/j.websem.2019.100514.

Об определении названия функции по абстрактному синтаксическому дереву с помощью нейронной сети

к.т.н. А. В. Красновидов, бакалавр П. А. Логинов

Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
alexkrasnovidow@mail.ru

Аннотация. Языки программирования высокого уровня всегда грамматически структурированы, поэтому алгоритмы их синтаксического анализа хорошо разработаны. В своем большинстве они относятся к классу линейных алгоритмов, что позволяет с их помощью обнаруживать большинство синтаксических ошибок за один просмотр исходного текста программы. При этом почти все ошибки гарантированно обнаруживаются. Однако существует целый ряд задач синтаксического анализа, для решения которых не существует линейных алгоритмов. К числу таких задач принадлежит обнаружение копии кода, классификация кода, определение дефектных модулей и т. п. Подобные задачи могут быть решены с помощью адаптивных алгоритмов. Показано, что наиболее подходящим средством реализации подобных алгоритмов является использование аппарата нейронных сетей. В качестве примера решена задача разработки алгоритма определения названия функции языка программирования Python по абстрактному синтаксическому дереву этой функции. Построена и обучена нейронная сеть, основанная на рекуррентных слоях.

Ключевые слова: языки программирования, синтаксическое дерево, адаптивный алгоритм, нейронная сеть, словарь токенов, модель CNN, модель LTSN.

ВВЕДЕНИЕ

При выполнении синтаксического анализа исходных текстов программ возникает целый ряд различных задач. Их можно условно разбить на две большие группы:

1. Анализ исходного текста программы с целью выявления синтаксических и, частично, семантических ошибок во время трансляции.

2. Анализ исходного текста программы, не связанный непосредственно с его трансляцией.

Задачи, относящиеся к первой группе, опираются на представлении языков программирования как некоторых формальных систем. Синтаксис языка формируется таким образом, чтобы можно было легко проверить, является ли некоторая последовательность символов правильным предложением языка. Благодаря хорошей структурированности подавляющего большинства современных языков высокого уровня для решения такой задачи возможно и целесообразно использовать линейные алгоритмы.

Ко второй группе можно отнести следующие задачи:

• Обнаружение копии кода. В больших проектах используется при рецензировании кода для обнаружения одинаковых методов. Один из них может быть удален, тогда оставшийся использовать повторно.

• Классификация кода. Может классифицировать код по задаче, которую он решает. Классификация позволит определить тип изменения и не выполнять просмотр незначительных изменений программного кода.

• Определение программных модулей, подверженных дефектам и требующих тщательного тестирования [1];

• Генерация словесного описания кода. Может быть использована для автоматического документирования программного кода. [2];

• Локализация ошибок.

Эти задачи решаются не во время компиляции или интерпретации программы, а в процессе выполнения специализированной обработки текстов программ. Особенностью таких задач является то, что они не требуют абсолютной точности решения. Для решения подобных задач не существует линейных алгоритмов, однако они могут быть решены с помощью адаптивных алгоритмов. Такие алгоритмы могут быть реализованы с помощью аппарата нейронных сетей. В настоящей статье рассматривается задача определения названия функции по ее абстрактному синтаксическому дереву. Обычно название функции в той или иной степени отражает ее назначение. Например, функция *sin* выполняет вычисление тригонометрической функции. Название функции может состоять из нескольких слов. Например, функция *button1_Click* языка C# обрабатывает событие *нажатие кнопки № 1*. Однако разработчики программного обеспечения часто присваивают функциям имена, не отражающие назначение функции. Присваивать функциям правильные названия особенно важно при выпуске программной документации [3]. Кроме того, решение этой задачи поможет реализовать плагин для подсказки правильного названия функции, например при вводе или редактировании исходного текста программы.

АБСТРАКТНОЕ СИНТАКСИЧЕСКОЕ ДЕРЕВО ПРОГРАММЫ

В процессе обработки исходного текста программы выполняются ее лексический и синтаксический анализы. На этапе лексического анализа входная строка из последовательности символов переводится в последовательность лексем (токенов) следующим образом: строка исходного текста *position = position + rate * 60* после лексического анализа преобразуется в строку

$\langle id \rangle \langle = \rangle \langle id \rangle \langle + \rangle \langle id \rangle \langle * \rangle \langle num \rangle \langle ; \rangle$ (*)

Суть лексического анализа заключается в выделении из последовательности символов входной последовательности лексем (токенов), неделимых с точки зрения последующего анализа. Токены делятся на классы [4].

Синтаксический анализ — это процесс сопоставления последовательности входных лексем с правилами языка. В простейшем случае синтаксический анализ позволяет определить принадлежность строки исходного текста данному языку. Однако на практике одного такого ответа (принадлежит или нет) недостаточно, и, как правило, параллельно с синтаксическим разбором происходят какие-либо действия, направленные на дальнейшее преобразование входной строки. Для выполнения дальнейшей обработки на этом этапе строится некоторое внутреннее представление программы, удобное для дальнейшего анализа и обработки. Часто в качестве такого представления используется абстрактное синтаксическое дерево (АСД) [5]. Пример такого дерева для последовательности (*) показан на рисунке 1.

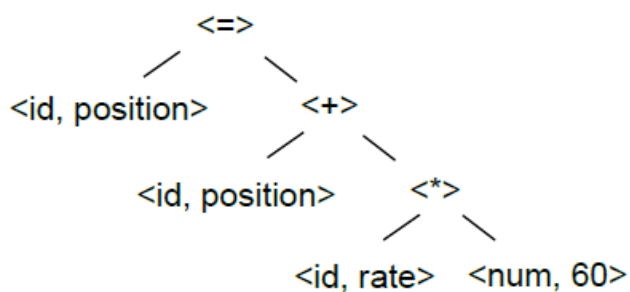


Рис. 1. Абстрактное синтаксическое дерево

Как можно видеть, АСД представляет собой структурное представление исходной программы, очищенное от элементов конкретного синтаксиса (в рассматриваемом примере в АСД не попал «разделитель», т. к. он не имеет отношения непосредственно к семантике данного фрагмента программы, а лишь к конкретному синтаксису языка). В качестве узлов в АСД выступают операторы, к которым присоединяются их аргументы, которые, в свою очередь, также могут быть составными узлами [4]. Узлы АСД получаются из лексем, выделенных на этапе лексического анализа. На рисунке 1 АСД построено полностью из лексем, кроме того, показано, что узлы, как и лексемы, содержат дополнительные атрибуты – имена идентификаторов, значения чисел и т. д.

Далее показано, что АСД могут быть использованы в качестве исходных данных при обучении нейронных сетей.

ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ ТОКЕНА

Нейронная сеть принимает на вход данные в цифровом формате. Это значит, что токены (символы) перед подачей их на вход нейронной сети должны быть преобразованы в цифровой формат. Одним из часто используемых способов такого преобразования заключается в создании словаря, в котором каждый токен будет сопоставлен с числовым значением. После этого можно перевести входную последовательность токенов в цифровой формат. Однако число, которым закодирован токен, фактически является его весом. Поэтому, если токены будут пронумерованы

последовательно или в случайном порядке, то некоторые токены будут в несколько раз значимее других без каких-либо на то оснований. Такой способ приемлем, если экспериментатору известны веса токенов, и он задает их вручную. В других случаях следует воспользоваться методами подбора правильных весов.

Существует множество методов подготовки данных для обучения. Популярными в работе с текстами являются OneHotEncoding (OHE), TF-IDF (TF — termfrequency, IDF — inversedocumentfrequency) и векторное представление слов (Embedding).

TF-IDF

TF-IDF — это простой и удобный способ оценить важность токена для какой-либо последовательности относительно всех остальных последовательностей. Принцип работы метода заключается в следующем: если токен встречается в какой-либо последовательности часто, при этом встречаясь редко во всех остальных последовательностях, то такой токен имеет большую значимость для этой последовательности. Эта метрика хороша тем, что токены, не важные для вообще всех последовательностей, получают очень низкий вес TF-IDF (потому что часто встречаются во всех последовательностях), а важные — высокий. Значение TF-IDF рассчитывается для каждого токена. TF — это частота токена в последовательности, для расчета берется количество вхождений токена в последовательность и делится на длину (количество токенов) последовательности. IDF — обратная частота токена в последовательностях, рассчитывается следующим образом: количество всех последовательностей в коллекции делится на количество последовательностей, в которых употребляется токен. Таким образом, наибольший вес IDF будет иметь токен, который встречается только в одной последовательности. Итоговый вес токена является произведением величин TF и IDF. Данный метод подходит для решения задач классификации, поэтому он использован в рамках настоящей работы.

РЕШЕНИЕ ЗАДАЧИ ОПРЕДЕЛЕНИЯ ИМЕНИ ФУНКЦИИ ПО АБСТРАКТНОМУ СИНТАКСИЧЕСКОМУ ДЕРЕВУ ТЕЛА ФУНКЦИИ

Задачу определения названия функции по абстрактному синтаксическому дереву (АСД) тела этой функции логично решать при помощи классификации. Задача классификации заключается в определении по входным данным, к какому классу эти данные принадлежат. Задачи классификации с помощью нейронных сетей решаются следующим образом. Берется набор данных и размечается. Каждому примеру сопоставляют метку. В поставленной задаче примером является АСД тела функции, а метка — название этой функции. В итоге получается определенное количество уникальных меток. Каждая уникальная метка является уникальным классом, который будет определять нейронная сеть. Далее размеченный набор данных подается на вход нейронной сети. Нейронная сеть имеет свои внутренние параметры, отвечающие за определение класса (предсказание). Таким образом, нейронная сеть анализирует данные и пытается предсказать, к какому классу они относятся. Каждый раз, когда нейронная сеть ошибается, она корректирует свои внутренние параметры, чтобы улучшить качество предсказания. В задаче определения названия функции по абстрактному синтаксическому де-

реву тела этой функции данными являются АСД тела функций, а метками — названия функций. В результате подачи АСД тела функции на вход нейронной сети, она выдает вероятностное распределение между классами (массив значений вероятностей). Выбирается класс с наибольшей вероятностью.

Перед обучением нейронной сети необходимо собрать данные для обучения, обработать их и построить модель.

ДАННЫЕ ДЛЯ ОБУЧЕНИЯ

В качестве примера в рамках настоящей статьи рассматривается задача определения названия функции языка

Python. Данные для обучения были взяты из набора данных «PythonASTs», включающего в себя более 150 тысяч функций, написанных на языке Python. Каждая программа преобразована в последовательность узлов АСД программы. К набору данных прилагается лексический анализатор, который может преобразовать исходный текст программы в последовательность узлов АСД. На рисунке 2 изображено абстрактное синтаксическое дерево следующей функции:

```
defsum(a, b):
    return a+b
```

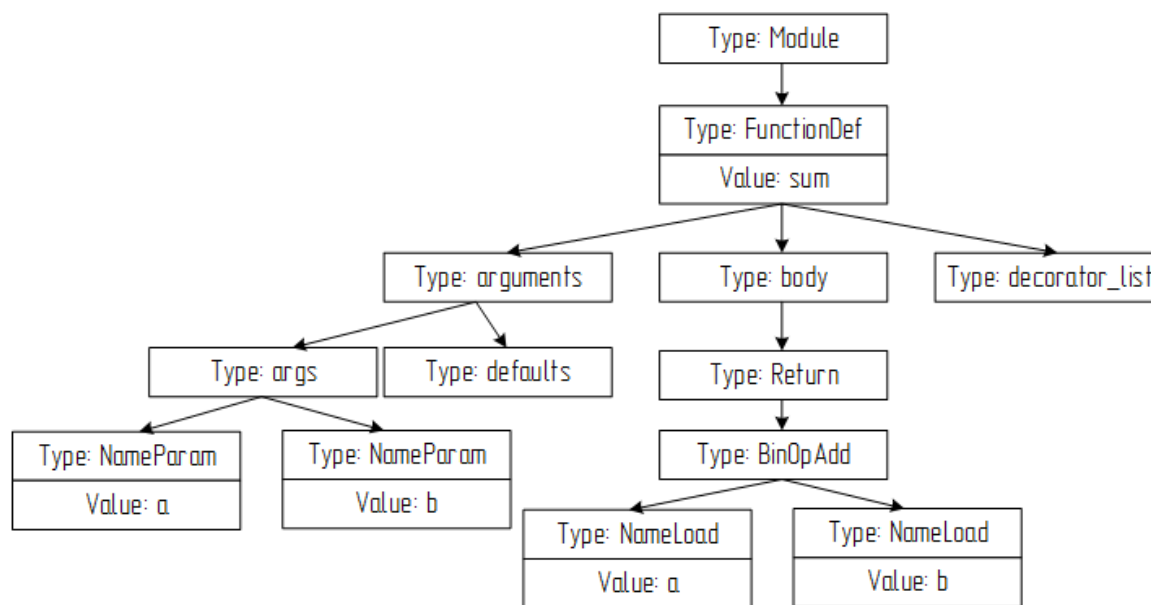


Рис. 2. Абстрактное синтаксическое дерево функции на языке Python

Работа лексического анализатора заключается в следующем. Он имплементирует пакет *ast* стандартной библиотеки Python для преобразования исходного кода программы в абстрактное синтаксическое дерево. Далее он обходит АСД с помощью алгоритма прямого обхода дерева. Каждое посещение узла дерева добавляется в последовательность. Каждый узел несет информацию: тип узла, дочерние узлы (если они есть), значение узла (если оно есть). Таким образом, получается последовательность

структур данных, где каждый элемент — это узел дерева. Дерево содержит узлы двух типов: терминальные и нетерминальные. Терминальные узлы имеют поле *value* (значение) и не имеют дочерних узлов [6]. Структура начала последовательности приведена на рисунке 3.

В итоге прямого обхода получается последовательность просмотренных узлов дерева в порядке их просмотра.

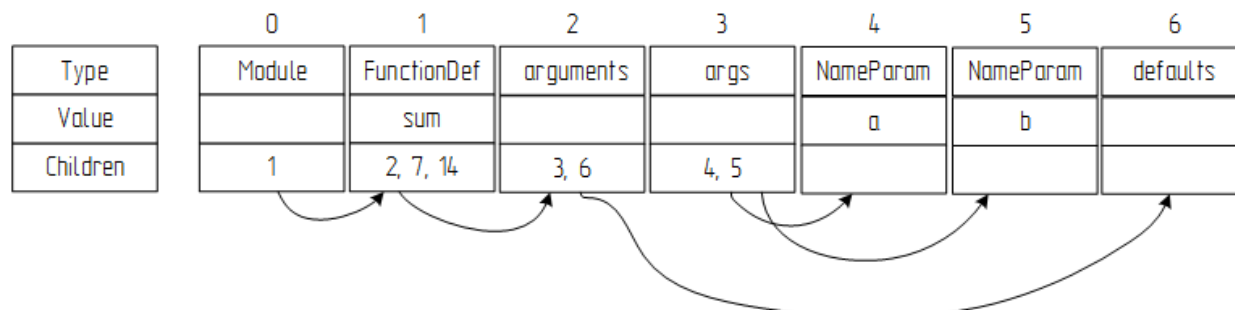


Рис. 3. Структура данных набора «PythonASTs»

ФОРМИРОВАНИЕ СЛОВАРЯ ФУНКЦИЙ

Модель нейронной сети в процессе работы решает задачу классификации: по АСД функции, модель относит эту функцию к одному из классов. Классами в данном

случае являются названия функций. Чтобы нейронная сеть смогла справиться с задачей классификации, необходимо производить обучение с большим количеством примеров для каждого класса. Если количество примеров какой-

нибудь функции недостаточно, то нейронная сеть не сможет запомнить зависимости, характерные для данной функции, и примеры этой функции будут классифицированы некорректно.

Первый шаг в формировании такого словаря заключается в подсчете для каждого названия функции количества примеров названия этой функции, содержащихся в данных (в метках, а не узлах). После этого получается большой словарь, который содержит все уникальные функции из данных.

Второй шаг заключается в выборе порога вхождения функции в словарь по количеству примеров этой функции. Все функции, которые ниже этого порога, отсекаются и не будут участвовать в обучении и тестах. Этот порог определяет качество и актуальность будущей модели. Если порог будет выбран очень большой, то получится мало различных функций, и модель будет классифицировать между маленьким количеством классов. Такая модель будет бесполезна. Если же выбрать маленький порог, то получится большое количество классов. У многих классов будет маленькое количество обучающих примеров, что негативно скажется на качестве модели. Выбор порога определяется размером набора данных, чем больше примеров функций с одинаковым названием, тем больше можно выбрать порог. С этим значением можно экспериментировать в небольшом диапазоне, подбирая оптимальное значение. Для набора данных «PythonASTs» выбран порог со значением 300. То есть функции, которые имеют менее 300 примеров, будут исключены из выборки. В результате отбора останется примерно 111 уникальных функций (111 классов).

Третий шаг заключается в исключении функций, которые сложно классифицировать. Существует ряд функций, структура которых во многих программах отличается. Например, если посмотреть на коды функций «setter», то все они очень похожи, поэтому их легко классифицировать. Но если сравнивать коды функций «main» или «gun», то для большинства программ они уникальны. Отсюда следует необходимость исключения подобных функций из выборки. Также исключаются некоторые служебные функции и функции с названиями «f», «fo», «func» и т. д.

На четвертом шаге производится объединение в один класс некоторых классов. Некоторые названия функций в выборке одинаковы по смыслу, но имеют различные имена. Например, функции с именами «representation» и «string» выполняют одинаковую задачу, поэтому могут быть объединены в один класс «to_string». Также на этом шаге названия приводятся к нижнему регистру. Такие функции, как «setup» и «setUp» становятся одним классом.

После выполнения всех шагов получается словарь названий функций, которые будут участвовать в обучении. По этому словарю производится фильтрация всех функций. В выборке остаются только те функции, названия которых присутствуют в словаре.

ФОРМИРОВАНИЕ СЛОВАРЯ ЗНАЧЕНИЙ

Многие узлы АСД имеют собственное значение. Значение может многое сказать о функции. Например, если в функцию передается аргумент с именем «file_name», а в самой функции используются методы «open», «getline», то векторное представление АСД функции с таким набором

токенов можно классифицировать намного точнее, чем если бы эти данные были неизвестны. Значениями являются константы, имена идентификаторов, строки (комментарии). Они дают дополнительную важную информацию, но так как существует огромное количество уникальных значений, целесообразно иметь возможность использования только части значений. Для этого необходимо произвести отбор тех значений, которые будут использоваться в качестве токенов в векторном представлении функций. Для отбора значений формируется словарь уникальных значений со счетчиком для каждого значения, чтобы определить его частоту (по аналогии со словарем названий функций). Словарь необходимо отсортировать в порядке убывания. Наиболее часто встречающиеся значения оказываются в начале словаря. Однако значительное количество часто встречающихся значений являются нейтральными и не оказывают значительного влияния на принятие решения. Таким образом, для завершения формирования словаря необходимо его укоротить.

Эта задача решается экспериментальным методом. Необходимо обучить модель с разным размером словаря, причем менять размер при экспериментах можно с большим разбросом [7].

ФИЛЬТРАЦИЯ ДАННЫХ

На предыдущих этапах был сформирован набор данных, который состоит из метки (названия функции) и векторного представления АСД этой функции. Каждый узел АСД хранит информацию о типе узла и значении узла (если есть).

После формирования набора данных необходимо преобразовать узлы функций путем выбора для каждого узла одного из двух возможных вариантов: использовать в качестве токена тип узла или значение узла.

Выбор производится при помощи сформированного словаря значений. Процесс выбора следующий: последовательно просматриваются все узлы всех функций и в каждом узле проверяется, есть ли у узла значение. Если его нет, то в качестве токена выбирается тип узла, если оно есть, то проверяется, входит ли значение в словарь значений: если да, то выбирается значение узла, если нет — то тип узла.

Последним шагом подготовки данных является удаление первого элемента векторов токенов, который несет информацию о названии функции. После этого имеется набор данных, имеющий следующую структуру: метка (название функции/класс) и сопоставленный с ней вектор токенов.

АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ

Архитектура нейронной сети определяет, из каких слоев состоит модель, и то, как эти слои соединены между собой. В настоящей работе используется последовательная архитектура модели. Это значит, что слои расположены друг за другом.

В данном случае модель нейронной сети решает задачу классификации. На ее вход поступает последовательность токенов, на выходе модели получается вероятностное распределение классов, где каждый класс является названием функции. Таким образом, нейронная сеть определяет, с какой вероятностью последовательность токенов относится к каждому из классов.

В качестве данных для обучения используются последовательности токенов, где важен порядок токенов, и токены имеют зависимости. Использование представления последовательностей токенов в виде разреженного вектора в данной задаче неэффективно, так как токены имеют зависимости. Поэтому для решения этой задачи больше подходят рекуррентные нейронные сети или сверточные нейронные сети, потому что они учитывают связи между токенами.

Первым слоем выбран Embedding, потому что он работает с последовательностями, нормализует данные и подходит и для рекуррентных и для сверточных сетей.

Последний слой — полносвязный слой, у которого количество нейронов равно количеству определяемых классов (названий функций). На выходе этого слоя выдается вектор распределения вероятностей каждого класса. Общая архитектура модели нейронной сети для решения рассматриваемой задачи представлена на рисунке 4.

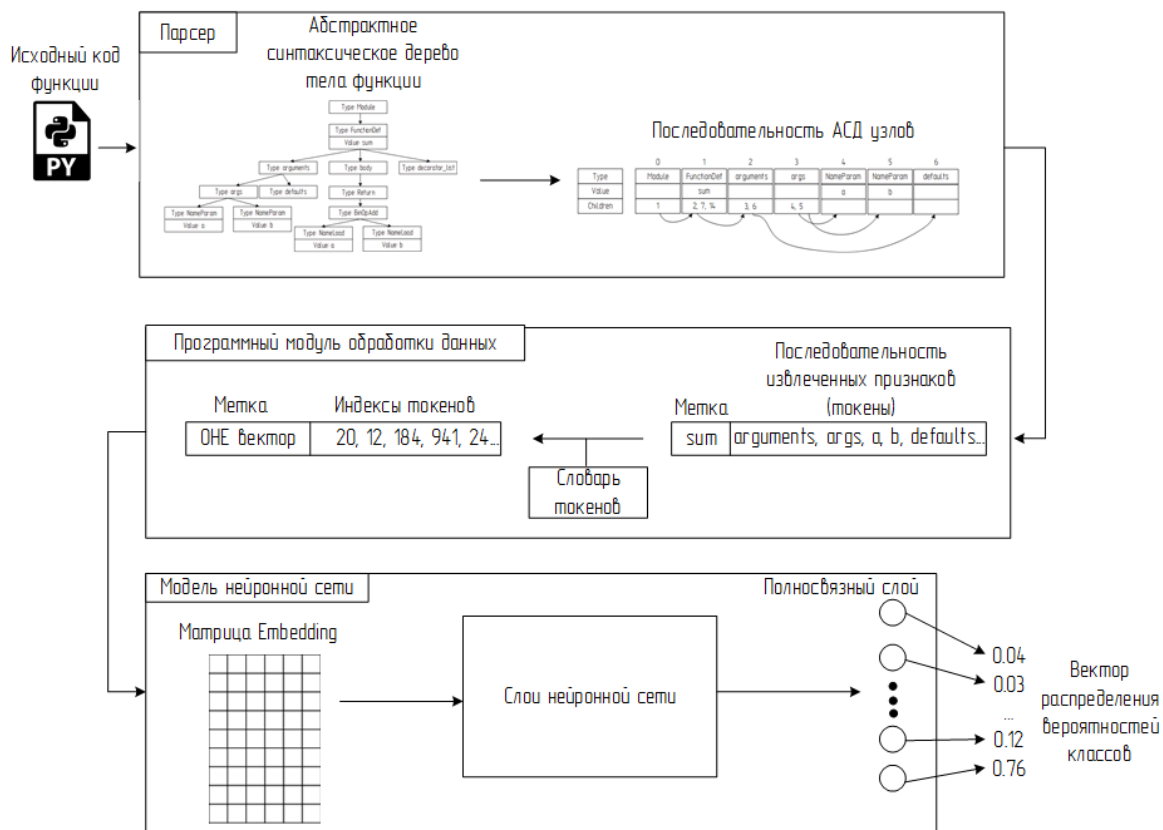


Рис. 4. Общая архитектура модели нейронной сети

ПОДГОТОВКА НАБОРОВ ДАННЫХ

Исходный набор данных «PythonASTs» был обработан несколько раз с различными параметрами. В результате получено несколько наборов данных для обучения, которые приведены в таблице 1.

Таблица 1

Сформированные наборы данных для обучения

Размер словаря токенов	Порог вхождения функции в выборку	Число функций для обучения	Число классов
15 000	300	130 300	114
15 000	190	146 540	203
9 000	300	128 898	111
9 000	190	155 470	203
2 000	300	128 898	111
2 000	190	155 470	203

Так как модель нейронной сети предполагает, что первый слой – это слой Embedding, то последовательности токенов должны быть преобразованы в вектора чисел одинаковой длины. Поэтому необходимо выбрать оптимальную длину последовательностей. Сделать это можно

экспериментально, пробуя разные длины. Начать можно со значений, близких к средней длине последовательности.

После выбора длины последовательности необходимо привести вектора к одной длине. Последовательность обрывается, если она длиннее выбранного числа, и дополняется нулями, если короче, чем выбранное число. Причем нули добавляются в начале последовательности, так как при использовании рекуррентных сетей там теряется меньше информации [8, 9].

Далее подготавливаются метки, с которыми сверяется модель при обучении. Каждая метка сопоставлена с одной последовательностью. Метка несет информацию о названии функции.

Распределение вероятностей каждой метки происходит на последнем полносвязном слое модели. Количество нейронов этого слоя равно количеству классов. Выходом этого слоя является вектор со значением вероятности для каждой метки.

Все этапы подготовки данных для обучения были произведены с помощью языка программирования Python в среде Google Colab. В подготовке данных были использованы стандартные библиотеки Python 3 и возможности

Python 2. Следует отметить, что на этапах обработки и подготовки данных к обучению всегда использовалась пара наборов. Один набор является обучающим, второй — тестовым. Такое деление наборов необходимо для проверки того, как нейронная сеть справляется с примерами, которые она не наблюдала при обучении, так как нейронная сеть имеет свойство переобучаться и выдавать результат лучше на тех данных, на которых она учится.

ЭКСПЕРИМЕНТАЛЬНЫЙ ПОДБОР АРХИТЕКТУРЫ МОДЕЛИ

Так как архитектура модели определяется большим количеством параметров, то имеется возможность построить множество разных моделей, решающих одну и ту же задачу, а затем выбрать ту, которая наилучшим способом справляется с решением поставленной задачи.

Для разрабатываемой модели ранее были определены первый и последний слои. Первый слой является слоем Embedding, а последний слой — полносвязный.

Построение моделей выполнялось при помощи языка Python. Для этого были использованы две популярные

библиотеки для работы с нейронными сетями: библиотека TensorFlow и библиотека Keras (является надстройкой над TensorFlow и еще несколькими библиотеками) [10].

Точность работы модели определяется по тому, правильно ли она определяет метки примеров тестовой выборки, то есть тех примеров, которых нейронная сеть не видела при обучении. Точность модели показывает, с какой вероятностью модель определяет правильную метку для примера данных. Именно такая точность является качественным показателем модели: чем больше эта точность, тем модель лучше.

Первая модель была построена на основе сверточных слоев. Эта модель имеет аббревиатуру CNN (Convolution Neural Network). Помимо входного, выходного и сверточных слоев в модели присутствуют слои пулинга и слой прореживания. В библиотеке Keras сверточный слой называется Conv1D, полносвязный — Dense, а прореживание — Dropout. Структурная схема модели представлена на рисунке 5.

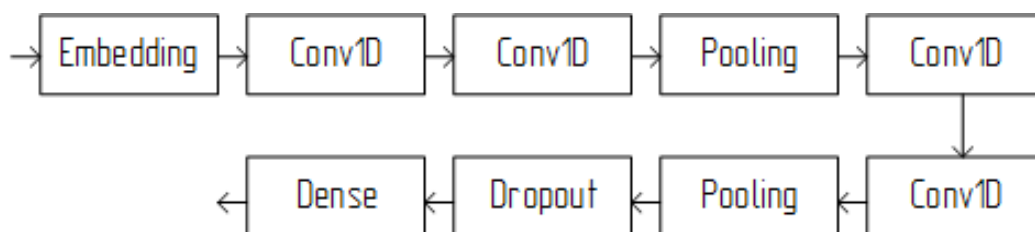


Рис. 5. Структурная схема модели CNN

На этой модели были опробованы различные наборы подготовленных данных. Обучение заканчивается, когда модель долгое время не улучшается или происходит большое переобучение. Результаты обучения приведены в таблице 2.

Таблица 2

Результаты обучения модели CNN

№	Набор данных (словарь-порог-количество классов)	Длина последовательности	Число эпох	Точность, %
1	2000-190-203	300	45	70,33
2	2000-300-111	300	25	73,53
3	9000-190-203	300	30	70,62
4	9000-300-111	300	30	75,14
5	15000-300-203	300	30	69,51
6	15000-300-114	300	40	75,00
7	9000-300-111	50	30	74,13
8	9000-300-111	150	30	75,07
9	9000-300-111	500	30	75,1

Выполненные эксперименты с этой моделью показали, что ее переобучение происходит быстро, что демонстрирует рисунок 6.

Из анализа таблицы 2 видно, что модель выдает лучшую точность на наборе данных 9000-300-111. Поэтому дальнейшие эксперименты производились именно с этим набором. В таблице 3 приведены параметры лучшего результата работы модели.

Таблица 3

Лучший результат модели CNN

Размер словаря токенов	9 000
Число классов	111
Длина последовательности	300
Число эпох на обучение	30
Время обучения	14 минут 45 секунд
Размер модели	9,85 МБ
Точность классификации	75,14 %

Вторая модель построена на основе LSTM (Long Short Term Memory)-слоев. Модели LSTM являются модификацией рекуррентных слоев [11]. Нейрон LSTM имеет более сложную структуру, которая лучше справляется с затуханием информации при обработке длинных последовательностей. Модель обучается на лучшем наборе по результатам работы первой модели, что дает возможность сравнить точность двух моделей.

Структура модели приведена на рисунке 7. Результаты обучения приведены в таблице 4. Результаты сравнения работы моделей приведены в таблице 5.

Таблица 4

Результаты обучения модели LSTM

Набор данных (словарь-порог-количество классов)	Длина последовательности	Число эпох	Точность, %
9 000-300-111	300	89	77,8

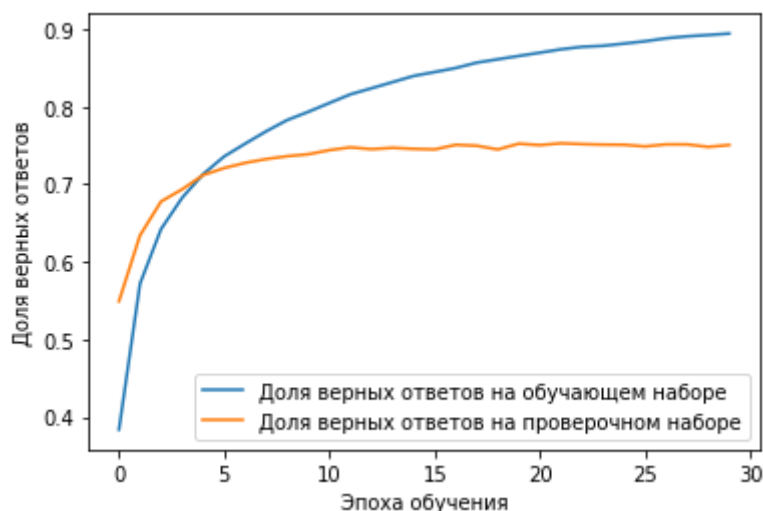


Рис. 6. График обучения модели CNN

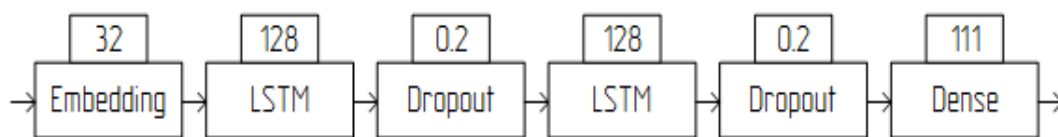


Рис. 7. Структурная схема модели LSTM

Таблица 5

Сравнение обученных моделей

	CNN	LSTM
Точность	75,14 %	77,8 %
Время обучения	14 минут 45 секунд	3 часа 12 минут 50 секунд
Размер модели	9,85 МБ	6,00 МБ
Реальные примеры	Хорошо	Очень хорошо

Приведенные в таблице результаты сравнения обученных моделей позволяют сделать вывод о предпочтительности использования модели LSTM.

АПРОБАЦИЯ РЕЗУЛЬТАТОВ

Апробация производилась на модели, основой которой являются рекуррентные слои (LSTM). Данными для обучения модели является набор данных «PythonASTs», который состоит из готовых абстрактных синтаксических деревьев программ. Деревья преобразованы в последовательности узлов этих деревьев. По последовательности

узлов человеку сложно определить, что делает функция. Поэтому для проверки качества набора данных и обучения модели, производится проверка работы модели на примерах функций, взятых не из набора «PythonASTs». Примеры функций, на которых выполнялась проверка, взяты из некоторых открытых проектов, размещенных на GitHub — крупнейшем веб-сервисе для поддержки IT-проектов и их совместной разработки. Для проверки взято несколько примеров функций, которые умеет распознавать модель. С помощью предоставленного парсера, составителями набора «PythonASTs», исходные коды примеров функций преобразованы к такому же виду, как и в наборе, только теперь имеется исходный код функции.

Пример 1. Функция называется *to_string*. Такое название функции имеется в словаре классов модели. Результат определения имени функции *to_string* показан на рисунке 8.

В левой части рисунка 8 приведен код функции *to_string*. В ней производится аккумуляция ключей и значений объекта *param* в строку, и эта строка передается на выходе из функции. Модель с очень высокой точностью определяет правильное название функции.

```
def f(self):
    out = ""
    for item in self.params:
        out += "%s = %s\n" % (item, self.params[item])
        out += "\n"
    return out
```

```
99.67% ██████████ to_string
0.19% ██████████ info
0.12% ██████████ show
0% ██████████ __getstate__
0% ██████████ SpecificCodeCPP
```

Рис. 8. Результат определения имени функции *to_string*

Пример 2. Функция называется *test_module*. Такого названия функции нет в словаре модели, поэтому ожидается увидеть похожее название. Результат определения имени функции *test_module* показан на рисунке 9.

Ключевым словом языка Python для проверки значений является *assert*. В функции (рис. 9) при помощи этого слова тестируются некоторые значения. Модель подбирает правильное название функции, а также выдает несколько семантически связанных названий.

```
def f(self):
    assert modlevel[0] == 42
    assert not hasattr(test_modlevel, 'answer')
```

Percentage	Function Name
35.95%	test_simple
34.3%	test
24.57%	tearDown
5.11%	check
0.02%	setup

Рис. 9. Результат определения имени функции *test_module*

```
def f(request):
    if request.user.is_authenticated():
        return redirect('index')
    elif request.POST:
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(username=username,
                            password=password)
        if user is not None:
            if user.is_active:
                login(request, user)
                return redirect('index')
        else:
            return render(request,
                          'login/login.html',{'error': '...'})
```

Percentage	Function Name
98.78%	login
1.19%	register
0.03%	index
0%	reset
0%	create

Рис. 10. Результат определения имени функции *login_user*

ЗАКЛЮЧЕНИЕ

В статье решена задача определения названия функции по абстрактному синтаксическому дереву ее тела. В качестве языка программирования рассмотрен набирающий все большую популярность язык Python. Для решения этой задачи построены несколько вариантов глубоких нейронных сетей, две лучшие выбраны для экспериментирования. Точнее всего справилась модель нейронной сети, основанная на рекуррентных слоях. Эта модель правильно определяет название функции, которая не участвовала в обучении, с вероятностью 98,8 %. Однако она обучается дольше, чем модель, основанная на сверточных слоях. Реализованная модель нейронной сети может выдавать лучшие результаты, если использовать больший набор данных и лучше обработать данные для обучения. В статье был использован набор данных, имеющий длину 300. При работе с ним модель показала наилучший результат, что согласуется с результатом, полученным в [12]. Одно из главных преимуществ модели в том, что она быстро обучается. И она не привязана к конкретному языку про-

Пример 3. Функция называется *login_user*. На рисунке 10 приведена функция авторизации пользователя на сайте. Оригинального названия в модели нет, но она определяет назначение функции верно.

Таким образом, в результате апробации было выяснено, что модель работает правильно на реальных примерах. Отсюда следует, что данные для обучения были корректными, и модель обучена правильно.

граммирования: можно определять названия функций других языков, если обучить модель на абстрактных синтаксических деревьях этих языков. После улучшения модели путем коррекции параметров и использования большего набора данных ее можно использовать в областях суммаризации программного кода, машинного обучения, коррекции кода.

ЛИТЕРАТУРА

1. A Suitable AST Node Granularity and Multi-Kernel Transfer Convolutional Neural Network for Cross-Project Defect Prediction / J. Deng, L. Lu, S. Qui, Y. Ou. // IEEE Access. 2020. Vol. 8. Pp. 66647–66661. DOI: 10.1109/Access.2020.2985780.
2. code2seq: Generating Sequences from Structured Representations of Code / U. Alon, O. Levy, S. Brody, E. Yahav // Proceeding of Seventh International Conference on Learning Representations (ICLR'2019), (New Orleans, May 6–9, 2019) / ArXiv. 2019. Vol. abs/1808.01400. 22 p.

3. code2vec: Learning Distributed Representations of Code / U. Alon, M. Zilberstein, O. Levy, E. Yahav // Proceedings of the ACM on Programming Languages. Article 40 (January 2019). 29 p. DOI: 10.1145/3290353.

4. Опалева Э. А. Языки программирования и методы трансляции: Учебное пособие / Э. А. Опалева, В. П. Самойленко. — СПб.: БХВ-Петербург, 2005. — 480 с.

5. A Novel Neural Source Code Representation Based on Abstract Syntax Tree / J. Zhang, X. Wang, H. Zhang, et al. // Proceedings of IEEE/ACM 41st International Conference on Software Engineering (ICSE'2019), (Montreal, Canada, May 25–31, 2019). — Pp. 783–794. DOI: 10.1109/ICSE.2019.00086.

6. Transfer Convolutional Neural Network for Cross-Project Defect Prediction / S. Qiu, X. Hu, J. Deng, et al. // Applied Science. 2019. Vol. 9, No. 13. Article 2660. 17 p. DOI: 10.3390/App9132660.

7. Николенко С. Глубокое обучение. Погружение в мир нейронных сетей / С. Николенко, А. Кадурин, Е. Ар-

хангельская. — СПб.: Питер, 2018. — 480 с. — (Библиотека программиста).

8. Moolayil J. Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python. — Apress, 2019. — 197 p.

9. Методы и модели исследования сложных систем и обработки больших данных: Монография / И. Ю. Парамонов, В. А. Смагин, Н. Е. Косых, А. Д. Хомоненко; под ред. В. А. Смагина, А. Д. Хомоненко. — СПб.: Лань, 2020. — 236 с. — (Учебники для вузов. Специальная литература).

10. Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow / I. Vasilev, D. Slater, G. Spacagna, et al. — Second Edition. — Packt Publishing, 2019. — 386 p.

11. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. — Springer International Publishing, 2018. — 520 p.

12. Алексеев А. С. Распознавание зашумленных текстовых символов с помощью обучаемой нейронной сети / А. С. Алексеев, А. В. Красновидов // Интеллектуальные технологии на транспорте. 2018. № 2 (14). С. 28–33.

About Determining a Function Name by Abstract Syntactic Tree Using Neural Network

PhD A. V. Krasnovidov, Bachelor P. A. Loginov
Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
alexkrasnovidow@mail.ru

Abstract. High-level programming languages always have grammatical structure, do the algorithms for their parsing are well developed. Mainly they belong to the class of linear algorithms, which allows them to detect most syntax errors in a single scan of the source code of the program. In most cases, almost all errors are guaranteed to be detected. However, there are a number of syntactic analysis task for which linear algorithms do not exist. These are kinds of tasks embrace detecting a copy of the program code, classifying the code, identifying defective modules, etc. However, like tasks can be solved using adaptive algorithms. It is shown the most suitable means of implementing such algorithms is the use of means of neural networks. The task of developing an algorithm to determine the name of a function of the Python programming language according the abstract syntax tree of this function as an example considered. A neural network based on recurrent layers was built and trained.

Keywords: programming languages, syntax tree, adaptive algorithm, neural network, token dictionary, CNN model, LTSN model.

REFERENCES

1. Deng J., Lu L., Qui S., Ou Y. A Suitable AST Node Granularity and Multi-Kernel Transfer Convolutional Neural Network for Cross-Project Defect Prediction, *IEEE Access*, 2020, Vol. 8, Pp. 66647–66661. DOI: 10.1109/Access.2020.2985780.
2. Alon U., Levy O., Brody S., Yahav E. code2seq: Generating Sequences from Structured Representations of Code, *Proceeding of Seventh International Conference on Learning Representations (ICLR'2019), New Orleans, May 6–9, 2019. ArXiv*, 2019, Vol. abs/1808.01400, 22 p.
3. Alon U., Zilberstein M., Levy O., Yahav E. code2vec: Learning Distributed Representations of Code, *Proceedings of the ACM on Programming Languages*, Article 40 (January 2019), 29 p. DOI: 10.1145/3290353.
4. Opaleva E. A., Samoylenko V. P. Programming languages and translation methods: Study guide [Yazyki programmirovaniya i metody translyatsii: Uchebnoe posobie], Saint Petersburg, BHV-Petersburg, 2005, 480 p.
5. Zhang J., Wang X., Zhang H., et al. A Novel Neural Source Code Representation Based on Abstract Syntax Tree, *Proceedings of IEEE/ACM 41st International Conference on Software Engineering (ICSE'2019), Montreal, Canada, May 25–31, 2019*, Pp. 783–794. DOI: 10.1109/ICSE.2019.00086.
6. Qui S., Hu X., Deng J., et al. Transfer Convolutional Neural Network for Cross-Project Defect Prediction, *Applied Science*, 2019, Vol. 9, No. 13, article 2660, 17 p. DOI: 10.3390/App9132660.
7. Nikolenko S., Kadurin A., Arkhangel'skaya E. Deep learning. Dive into the world of neural networks [Glubokoe obuchenie. Pogruzhenie v mir neyronnykh setey], Saint Petersburg, Piter Publishing House, 2018, 480 p.
8. Moolayil J. Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python. Apress, 2019, 197 p.
9. Paramonov I. Yu., Smagin V. A., Kosykh N. Ye., Khomonenko A. D. Methods and models for the study of complex systems and processing of big data: Monograph [Metody i modeli issledovaniya slozhnykh sistem i obrabotki bol'shikh dannykh: Monografiya], Saint Petersburg, LAN Publishing House, 2020, 236 p.
10. Vasilev I., Slater D., Spacagna G., et al. Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow. Second Edition. Packt Publishing, 2019, 386 p.
11. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. Springer International Publishing, 2018, 520 p.
12. Alekseev A. S., Krasnovidov A. V. Identification of Images Using Neural Network Training [Raspoznavanie zashumlennykh tekstovykh simvolov s pomoshch'yu obuchaemoy neyronnoy seti], *Intellectual Technologies on Transport [Intellektual'nye tekhnologii na transporte]*, 2018, No. 2 (14), Pp. 28–33.

Аппроксимация времени пребывания для системы массового обслуживания Fork-Join на основе инвариантов отношения

Р. С. Хабаров, к.т.н. В. А. Лохвицкий, к.т.н. А. С. Дудкин
 Военно-космическая академия им. А. Ф. Можайского
 Санкт-Петербург, Россия
 xabarov1985@gmail.com, lokhv_va@mail.ru, andry-ll@mail.ru

Аннотация. Предлагается метод аппроксимации среднего времени пребывания заявок в системе массового обслуживания Fork-Join с экспоненциальным распределением времени обслуживания на основе инвариантов отношения. Идея заключается в применении интуитивной пропорции между временными характеристиками подобных систем массового обслуживания. Показано, что по сравнению с известными методами, предложенная аппроксимация обладает большей точностью, особенно при увеличении количества каналов в системе.

Ключевые слова: системы массового обслуживания, процессы расщепления и слияния заявок, процесс fork-join, инварианты отношения.

ВВЕДЕНИЕ

Для оценивания показателей оперативности обработки запросов в системах, использующих технологии распределенных и параллельных вычислений, применяются системы массового обслуживания типа Split-Merge и Fork-Join.

Общая идея функционирования системы Split-Merge и Fork-Join заключается в следующем: поступающие в систему заявки «расщепляются» на n подзадач, каждая из которых отправляется в канал с номерами 1, 2, ..., n соответственно. Обработанные подзадачи попадают в буфер синхронизации, где дожидаются окончания обслуживания родственных им подзадач. В момент окончания обслуживания последней родственной подзадачи происходит синхронизация, т. е. объединение, после чего заявка покидает систему. Считается, что синхронизация происходит мгновенно.

Отличие систем Split-Merge и Fork-Join показано на рисунках 1 и 2 на примере трех каналов обслуживания. В случае Fork-Join освободившийся канал может быть занят подзадачей следующей заявки (рис. 1).

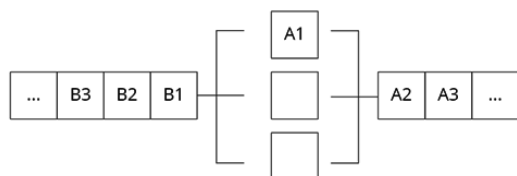


Рис. 1. Схема обслуживания заявок в СМО Fork-Join

При организации обслуживания Split-Merge (рис. 2) в момент поступления заявки происходит блокировка, и освободившиеся каналы простаивают, дожидаясь обслуживания последней из подзадач текущей заявки.

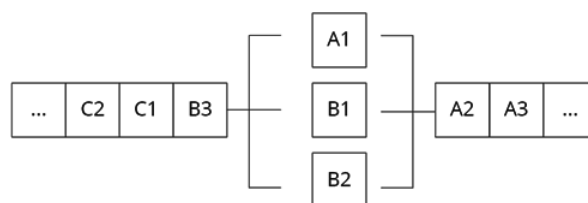


Рис. 2. Схема обслуживания заявок в СМО Split-Merge

Процессам Fork-Join и Split-Merge посвящено достаточно большое количество работ [1–14]. Для системы Split-Merge в [6] получено точное решение по определению максимума времени обслуживания независимых каналов с экспоненциальным временем обслуживания и различными интенсивностями, а также аппроксимации для случая общего распределения. В [8] упомянутое распределение получено для гомогенных и гетерогенных серверов, причем представление его в матрично-экспоненциальной форме позволило найти как первый, так и моменты высших порядков. Надо отметить, что указанный способ характеризуется высокой вычислительной сложностью, являющейся следствием входящих в него трудоемких операций обращения и кронекерова произведения матриц. Применение кронекеровой алгебры связано со значительным дополнительным расходом памяти, а также множеством избыточных операций с нулевыми операндами. В [1] найдено точное решение для произвольного распределения обслуживания на основе численного интегрирования по Чебышеву — Лаггеру. Решение обладает сравнительно небольшой трудоемкостью и высокой точностью.

Для систем Fork-Join точное выражение для среднего времени пребывания в системе с произвольным распределением обслуживания было получено только для системы с двумя каналами [5, 7]. Для случая $n > 2$ и экспоненциального обслуживания с помощью различных методов были получены аппроксимации среднего времени пребывания [9, 12, 13]. Приведем здесь кратко полученные результаты.

В [7] приведена точная формула для среднего времени пребывания в системе, $n = 2$ и экспоненциального распределения времени обслуживания

$$v_1 = (H_2 - \rho/8)v_1^M, \quad (1)$$

где $H_n = \sum_{i=1}^n 1/i$ и для $n = 2$ $H_2 = 1,5$,

$$\rho = \lambda/\mu,$$

$v_1^m = \frac{1}{\mu - \lambda}$ — среднее время пребывания в СМО M/M/1.

В [9] предложена аппроксимация для среднего времени пребывания в СМО Fork-Join с n каналами обслуживания

$$v_1 \approx \left[\frac{H_n}{H_2} + \frac{4}{11} \left(1 - \frac{H_n}{H_2} \right) \rho \right] \frac{12-\rho}{8} \frac{1}{\mu-\lambda}, n \geq 2 \quad (2)$$

В [12] Varki и Merchant предложили формулу

$$v_1 \approx \frac{1}{\mu} \left[H_n + \frac{\rho}{2(1-\rho)} \left(\sum_{i=1}^n \frac{1}{i-\rho} + (1-2\rho) \sum_{i=1}^n \frac{1}{i(i-\rho)} \right) \right], n \geq 2. \quad (3)$$

В [13] предложен метод аппроксимации времени отклика на основе комбинации методов интерполяции высокой и слабой входных нагрузок:

$$v_1 \approx [H_n + (V_n - H_n)\rho] \frac{1}{\mu-\lambda}, \quad (4)$$

$$\text{где } V_n = \sum_{i=1}^n \binom{n}{i} (-1)^{i-1} \sum_{m=1}^i \binom{i}{m} \frac{(m-1)!}{i^{m+1}}.$$

В данной работе предложен метод на основе аппроксимации инвариантов отношения, позволяющий получить оценку среднего времени пребывания для СМО Fork-Join с экспоненциальным распределением времени обслуживания с большей точностью.

ИДЕЯ МЕТОДА

В [15] для нахождения временных характеристик многоканальных СМО с приоритетами был использован метод на основе идеи инвариантов отношения между искомыми характеристиками:

$$\vec{M}_k / \vec{G}_k / n \approx \vec{M}_k / \vec{G}_k / 1 \cdot \frac{M/G/n}{M/G/1}. \quad (5)$$

Все обозначения приведены в нотации Кендалла. Методы расчета систем, указанных в правой части, считаются известными. В частности, для расчета систем применяются итерационные методы Такахаси — Таками или матрично-геометрической прогрессии [16–19]. Для беспriorитетных систем $M/G/1$ и $M/G/n$ с неоднородным потоком заявок использовалась суммарная интенсивность потока заявок и средневзвешенные моменты распределения длительности обслуживания.

Предложим подобный подход для нахождения среднего времени обслуживания в СМО Fork-Join с экспоненциальным временем обслуживания и количеством каналов. Перепишем пропорцию (5) в следующем виде:

$$FJ_n \approx FJ_2 \cdot \frac{M/M/n}{M/M/2}. \quad (6)$$

Здесь FJ_2 и FJ_n — СМО Fork-Join с 2-мя и n каналами соответственно. Для нахождения среднего времени пребывания в FJ_2 воспользуемся формулой (1). Расчет оставшихся СМО из правой части (6) будем осуществлять известными методами [16].

Приведем здесь схему реализации метода:

а) Рассчитать среднее время ожидания V_{FJ}^2 в СМО Fork-Join с 2-мя каналами обслуживания согласно формуле (1).

б) Для системы $M/M/n$ рассчитать среднее время пребывания V^n согласно известным формулам:

$$V^n = \frac{p_n}{\mu(1-\rho)^2} + \frac{1}{\mu}, \quad (7)$$

где

$$\rho = \frac{\lambda}{\mu}, p_n = p_0 \frac{\rho^n}{n!},$$

$$p_0 = \left[1 + \sum_{i=1}^{n-1} \frac{\rho^i}{i!} + \frac{\rho^n}{n!} \frac{1}{1-\frac{\rho}{n}} \right]^{-1}.$$

в) Для системы $M/M/2$ рассчитать среднее время пребывания заявок V^2 согласно (7), приняв $n = 2$.

г) Получить среднее время пребывания в СМО Fork-Join для n каналов обслуживания на основе соотношения

$$V_{FJ}^n = V_{FJ}^2 \cdot \frac{V^n}{V^2}.$$

Приведем результаты расчетов среднего времени пребывания в СМО Fork-Join в сравнении с методами, предложенными в [9, 12, 13] (формулы (2), (3), (4)) и данными имитационного моделирования.

На рисунках 3 и 4 приведены результаты расчетов среднего времени пребывания в зависимости от коэффициента загрузки для СМО Fork-Join с $n = 3$ и $n = 10$ соответственно. Для результатов расчетов согласно формулам (2), (3) и (4) использованы обозначения «NT», «VM», «Varma» соответственно. Результаты имитационного моделирования обозначены как «ИМ», результаты расчетов на основе инвариантов отношения — как «Инв».

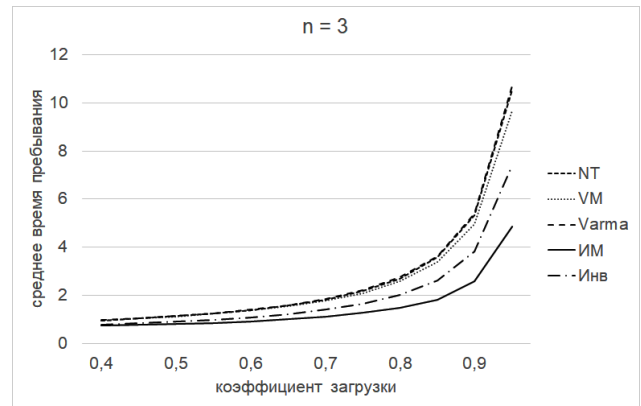


Рис. 3. Среднее время пребывания в СМО Fork-Join с $n = 3$ в зависимости от коэффициента загрузки

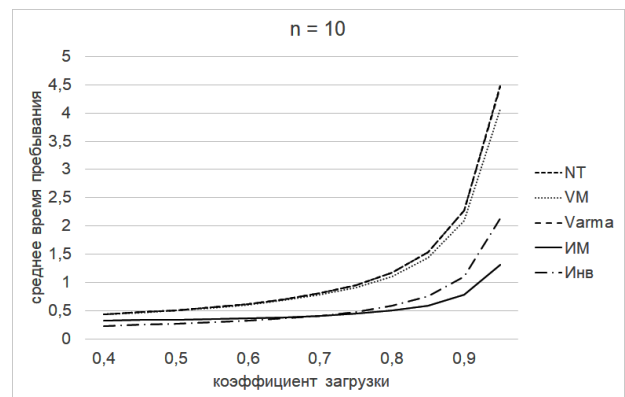


Рис. 4. Среднее время пребывания в СМО Fork-Join с $n = 10$ в зависимости от коэффициента загрузки

На рисунках 5 и 6 приведены результаты расчетов в зависимости от количества каналов обслуживания для коэффициентов загрузки 0,7 и 0,9 соответственно.

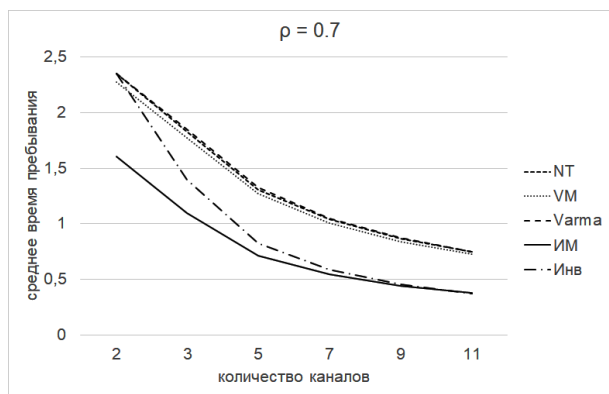


Рис. 5. Среднее время пребывания в СМО Fork-Join при $\rho = 0,7$ в зависимости от количества каналов обслуживания

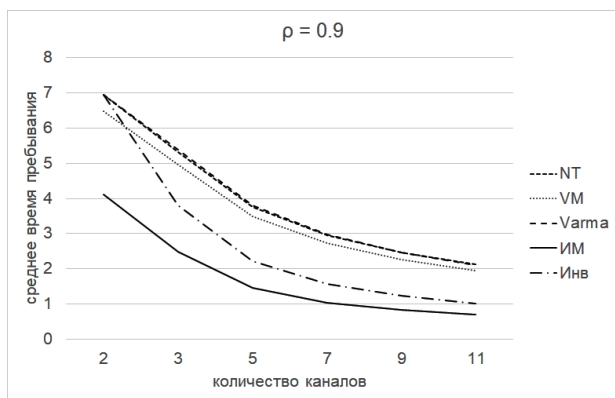


Рис. 6. Среднее время пребывания в СМО Fork-Join при $\rho = 0,9$ в зависимости от количества каналов обслуживания

На рисунках 7 и 8 приведены результаты расчетов относительной погрешности оценки среднего времени пребывания при $\rho = 0,9$ в зависимости от количества каналов обслуживания и при $n = 3$ в зависимости от коэффициента загрузки. За условный эталон приняты данные имитационного моделирования.

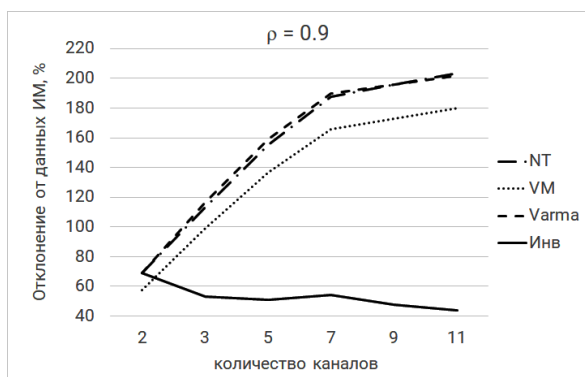


Рис. 7. Относительная погрешность расчета среднего времени пребывания в СМО Fork-Join с $\rho = 0,9$ в зависимости от количества каналов обслуживания

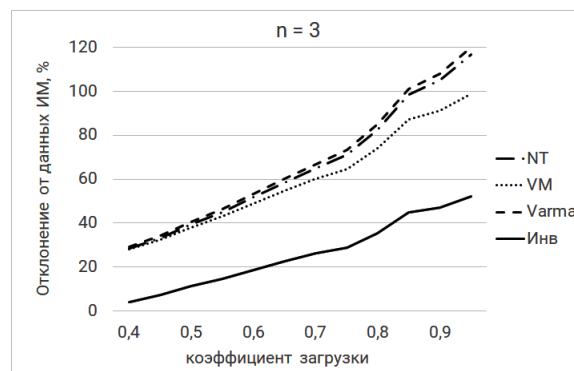


Рис. 8. Относительная погрешность расчета среднего времени пребывания в СМО Fork-Join при $n = 3$ в зависимости от коэффициента загрузки

Как видно из рисунков, точность аппроксимации методом инвариантов отношения выше рассчитанных по формулам (2–4), особенно это заметно при росте количества каналов обслуживания и коэффициента загрузки.

ЗАКЛЮЧЕНИЕ

Метод инвариантов отношения в сравнении с существующими методами показал большую точность аппроксимации для среднего времени пребывания для СМО с процессами Fork-Join. Было показано, что с ростом количества каналов точность оценки для предложенного метода увеличивается, тогда как для существующих методов заметно снижается. В дальнейших исследованиях планируется обобщить полученные результаты для случая неэкспоненциального распределения времени обслуживания.

ЛИТЕРАТУРА

1. Рыжиков Ю. И. Метод расчета длительности обработки задач в системе массового обслуживания с учетом процессов Split-Join / Ю. И. Рыжиков, В. А. Лохвицкий, Р. С. Хабаров // Известия высших учебных заведений. Приборостроение. 2019. Т. 62, № 5. С. 419–423. DOI: 10.17586/0021-3454-2019-62-5-419-423.
2. Хабаров Р. С. Модель оценивания оперативности многопоточной обработки задач в распределенной вычислительной среде с учетом процессов Split-Join / Р. С. Хабаров, В. А. Лохвицкий // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». 2019. Вып. 1. С. 26–34.
3. Alomari F. B. Efficient Response Time Approximation for Multiclass Fork and Join Queues in Open and Closed Queueing Networks / F. B. Alomari, D. A. Menascé // IEEE Transaction on Parallel and Distributed Systems. 2014. Vol. 25, Is. 6. Pp. 1437–1446. DOI: 10.1109/TPDS.2013.70.
4. Baccelli F. The Fork-Join Queue and Related Systems with Synchronization Constraints: Stochastic Ordering and Computable Bounds / F. Baccelli, A. M. Makowski, A. Shwartz // Advanced in Applied Probability. 1989. Vol. 21. Pp. 629–660. DOI: 10.2307/1427640.
5. Baccelli F. Two Parallel Queues Created by Arrivals with Two Demands: The M/G/2 Symmetrical Case // INRIA Report. No. 426 (July, 1986).
6. Fiorini P. M. Exact Analysis of Some Split-Merge Queues / P. M. Fiorini, L. Lipsky // ACM SIGMETRICS Performance Evaluation Review. 2015. Vol. 43, No. 2. Pp. 51–53. DOI: 10.1145/2825236.2825257.

7. Flatto L. Two Parallel Queues Created by Arrivals with Two Demands I / L. Flatto, S. Hahn // *SIAM Journal on Applied Mathematics*. 1984. Vol. 44, No. 5. Pp. 1041–1053.

8. Harrison P. G. Queueing Models with Maxima of Service Times / P. G. Harrison, S. Zertal // *Computer Performance Evaluation. Modelling Techniques and Tools: Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS 2003)*, (Urbana, IL, USA, September 2–5, 2003) / P. Kemper, W. H. Sanders (eds) // *Lecture Notes in Computer Science*, Vol. 2794. Pp. 152–168. DOI: 10.1007/978-3-540-45232-4_10.

9. Nelson R. Approximate Analysis of Fork/Join Synchronization in Parallel Queues / R. Nelson, A. N. Tantawi // *IEEE Transactions on Computers*. 1988. Vol. 37, No. 6. Pp. 739–743.

10. Olvera-Cravioto M. Parallel Queues with Synchronization / M. Olvera-Cravioto, O. Ruiz-Lacedelli // *ArXiv*. 2015. Vol. abs/1501.00186. 37 p.

11. Qiu Z. Beyond the Mean in Fork-Join Queues: Efficient Approximation for Response-Time Tails / Z. Qiu, J. G. Perez, P. G. Harrison // *Performance Evaluations*. 2015. Vol. 91. Pp. 99–116. DOI: 10.1016/j.peva.2015.06.007.

12. Varki E. The M/M/1 Fork-Join Queue with Variable SubTasks. [Electronic resource] / E. Varki, A. Merchant, H. Chen. 15 p.
URL: <http://www.cs.inh.edu/~varki/publication/2002-nov-open.pdf> (дата обращения 15.06.2020).

13. Varma S. Interpolation Approximations for Symmetric Fork-Join Queues / S. Varma, A. M. Makowski // *Performance Evaluation*. 1994. Vol. 20, Is. 1–3. Pp. 245–265. DOI: 10.1016/0166-5316(94)90016-7.

14. Wright P. E. Two Parallel Processors with Coupled Inputs // *Advances in Applied Probability*. 1992. Vol. 24, Is. 4. Pp. 986–1007. DOI: 10.2307/1427722.

15. Рыжиков Ю. И. Расчет многоканальных систем обслуживания с абсолютным и относительным приоритетами на основе инвариантов отношения / Ю. И. Рыжиков, А. Д. Хомоненко // *Интеллектуальные технологии на транспорте*. 2015. № 3 (3). С. 11–16.

16. Вишневский В. М. Теоретические основы проектирования компьютерных сетей: Монография. — М.: ЗАО «РИЦ «Техносфера», 2003. — 512 с.

17. Рыжиков Ю. И. Алгоритм расчета многоканальной системы с эрланговским обслуживанием // *Автоматика и телемеханика*. 1980. № 5. С. 30–37.

18. Рыжиков Ю. И. Итеративный метод расчета многоканальных систем с произвольным распределением времени обслуживания / Ю. И. Рыжиков, А. Д. Хомоненко // *Проблемы управления и теория информации*. 1980. № 3. С. 32–38.

19. Takahashi Y. A Numerical Method for the Steady-State Probabilities of a G1/G/C Queuing System in a General Class / Y. Takahashi, Y. Takami // *Journal of the Operations Research Society of Japan*. 1976. Vol. 19, Is. 2. Pp. 147–157. DOI: 10.15807/jorsj.19.147.

Sojourn Time Approximation for Fork-Join Queue Based on Relationship Invariants

R. S. Khabarov, PhD V. A. Lokhvitsky, PhD A. S. Dudkin

A. F. Mozhaisky Military Space Academy

Saint Petersburg, Russia

xabarov1985@gmail.com, lokhv_ca@mail.ru, andry-ll@mail.ru

Abstract. A method for mean sojourn time approximation for task in the Fork-Join queuing system with an exponential distribution of service times based on relation invariants is proposed. The idea is to use an intuitive proportion between the time characteristics of such queuing systems. It is shown that, compared with the known methods, the proposed approximation is more accurate, especially with an increase in the number of channels in the system.

Keywords: queuing systems, splitting and merging applications, fork-join process, relationship invariants.

REFERENCES

1. Ryzhikov Y. I., Lokhvitsky V. A., Khabarov R. S. Method of Calculating Task Treatment Duration in Queuing System with Consideration of Split-Join Processes [Metod rascheta dlitel'nosti obrabotki zadach v sisteme massovogo obsluzhivaniya s uchetom processov Split-Join], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroeniye]*, 2019 Vol. 62. No. 5, Pp. 419–423. DOI: 10.17586/0021-3454-2019-62-5-419-423.
2. Khabarov R. S., Lokhvitskiy V.A. Efficiency Evaluation Model of Parallel Processing in Distributed Environment Using Split-Join Queue [Model' otsenivaniya operativnosti mnogopotochnoy obrabotki zadach v raspredelennoy vychislitel'noy srede s uchetom protsessov Split-Join], *Vestnik of Russian New University. Complex Systems: Models, Analysis, Management [Vestnik Rossiyskogo novogo universiteta. Slozhnye sistemy: modeli, analiz i upravlenie]*, 2019, Is. 1, Pp. 26–34.
3. Alomari F. B., Menascé D. A. Efficient Response Time Approximation for Multiclass Fork and Join Queues in Open and Closed Queuing Networks, *IEEE Transaction on Parallel and Distributed Systems*, 2014, Vol. 25, Is. 6, Pp. 1437–1446. DOI: 10.1109/TPDS.2013.70.
4. Baccelli F., Makowski A. M., Schwartz A. The Fork-Join Queue and Related Systems with Synchronization Constraints: Stochastic Ordering and Computable Bounds, *Advanced in Applied Probability*, 1989, Vol. 21, Pp. 629–660. DOI: 10.2307/1427640.
5. Baccelli F. Two Parallel Queues Created by Arrivals with Two Demands: The M/G/2 Symmetrical Case, *INRIA Report*, No. 426 (July, 1986).
6. Fiorini P. M., Lipsky L. Exact Analysis of Some Split-Merge Queues, *ACM SIGMETRICS Performance Evaluation Review*, 2015, Vol. 43, No. 2. Pp. 51–53. DOI: 10.1145/2825236.2825257.
7. Flatto L., Hahn S. Two Parallel Queues Created by Arrivals with Two Demands I, *SIAM Journal on Applied Mathematics*, 1984, Vol. 44, No. 5, Pp. 1041–1053.
8. Harrison P. G., Zertal S. Queuing Models with Maxima of Service Times, In: *Kemper P., Sanders W. H. (eds) Computer Performance Evaluation, Modelling Techniques and Tools: Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS 2003), Urbana, IL, USA, September 2–5, 2003. Lecture Notes in Computer Science*, Vol. 2794. Pp. 152–168. DOI: 10.1007/978-3-540-45232-4_10.
9. Nelson R., Tantawi A. N. Approximate Analysis of Fork/Join Synchronization in Parallel Queues, *IEEE Transactions on Computers*, 1988, Vol. 37, No. 6, Pp. 739–743.
10. Olvera-Cravioto M., Ruiz-Lacedelli O. Parallel Queues with Synchronization, *ArXiv*, 2015, Vol. abs/1501.00186, 37 p.
11. Qiu Z., Perez J. G., Harrison P. G. Beyond the Mean in Fork-Join Queues: Efficient Approximation for Response-Time Tails, *Performance Evaluations*, 2015, Vol. 91, Pp. 99–116. DOI: 10.1016/j.peva.2015.06.007.
12. Varki E., Merchant A., Chen H. The M/M/1 Fork-Join Queue with Variable SubTasks, 15 p. Available at: <http://www.cs.inh.edu/~varki/publication/2002-nov-open.pdf> (accessed 15 June 2020).
13. Varma S., Makowski A. M. Interpolation Approximations for Symmetric Fork-Join Queues, *Performance Evaluation*, 1994, Vol. 20, Is. 1–3. Pp. 245–265. DOI: 10.1016/0166-5316(94)90016-7.
14. Wright P. E. Two Parallel Processors with Coupled Inputs, *Advances in Applied Probability*, 1992, Vol. 24, Is. 4, Pp. 986–1007. DOI: 10.2307/1427722.
15. Ryzhikov Y. I., Khomonenko A. D. Calculation of Multi-Channel Queuing Systems with Absolute and Relative Priorities on the Basis of Invariants Relationship [Raschet mnogokanal'nykh sistem obsluzhivaniya s absolyutnym i otnositel'nykh prioritetami na osnove invariantov otnosheniya], *Intellectual Technologies on Transport [Intellektual'nye tekhnologii na transporte]*, 2015, No. 3 (3), Pp. 11–16.
16. Vishnevsky V. M. Theoretical foundations of computer network design: Monograph [Teoreticheskie osnovy proektirovaniya komp'yuternykh setey: Monografiya], Moscow, TECHNOSPHERA Publishing Company, 2003, 512 p.
17. Ryzhikov Y. I. Algorithm for Calculating a Multichannel System with Erlang Service [Algoritm rascheta mnogokanal'noy sistemy s erlangovskim obsluzhivaniem], *Automation and Remote Control [Avtomatika i telemekhanika]*, 1980, No. 5, Pp. 30–37.
18. Ryzhikov Y. I., Khomonenko A. D. An Iterative Method for Calculating Multichannel Systems with an Arbitrary Distribution of Service Time [Iterativnyy metod rascheta mnogokanal'nykh sistem s proizvol'nym raspredeleniem vremeni obsluzhivaniya], *Problems of Control and Information Theory [Problemy upravleniya i teoriya informatsii]*, 1980, No. 3, Pp. 32–38.
19. Takahashi Y., Takami Y. A Numerical Method for the Steady-State Probabilities of a G1/G/C Queuing System in a General Class, *Journal of the Operations Research Society of Japan*, 1976, Vol. 19, Is. 2, Pp. 147–157. DOI: 10.15807/jorsj.19.147.

Сравнительный анализ скорости вычислений на центральном и графическом процессоре

бакалавр Н. С. Мальцев, И. А. Молодкин, к.воен.н. Р. Г. Гильванов

Петербургский государственный университет путей сообщения Императора Александра I

Санкт-Петербург, Россия

22nikich98@mail.ru, molodkin@pgups.ru, gilvanov1950@mail.ru

Аннотация. Описывается развитие технологии параллельных вычислений на графическом процессоре (GPU), рассматривается организация вычислений на центральном и графическом процессоре, особенности внутренней архитектуры. Раскрываются фундаментальные принципы работы программ с применением параллельных вычислений на основе передовой технологии CUDA от компании NVIDIA. Проводится исследование скорости выполнения ряда операций (матричное умножение, поиск максимального значения, трансформация вектора, вычисление суммы двух векторов) при разных входных данных с помощью разработанного программного комплекса. Производится анализ полученных результатов. Приводятся рекомендации по использованию технологии CUDA — условия, при соблюдении которых вычисления целесообразно выполнять на GPU (высокая степень распараллеливания; большой объем вычислений; возможность загрузить данные до проведения вычислений).

Ключевые слова: сравнительный анализ, скорость вычислений, CUDA, центральный процессор, графический процессор, параллельные вычисления.

ВВЕДЕНИЕ

В настоящее время развитие информационных технологий вызывает стремительный рост требований к вычислительной мощности компьютеров. Архитекторы центральных процессоров придерживаются стратегии увеличения производительности за счет увеличения числа ядер на кристалле и вводе новых инструкций для параллельной обработки данных.

Однако существует и другой подход для ускорения вычислений — применение графических процессоров (GPU), которые превосходят центральные процессоры (CPU) за счет графических ускорителей (рис. 1). Идея использования графических ускорителей для проведения не предназначенных им вычислений интересовала программистов еще с конца 90-х годов. Но только с появлением шейдеров в 2003 году произошел главный толчок к развитию абсо-

лютно новой технологии GPGPU (General Purpose Graphics Processing Units). Важную роль в развитии данной технологии сыграло специальное расширение для языка C под названием «BrookGPU». До появления данного расширения, программисты могли работать с GPU лишь через API Direct3D или OpenGL. Оно позволило программистам работать с привычной им средой, а уже сам компилятор, используя специальные библиотеки, реализовывал взаимодействие с GPU на низком уровне.

Такой подход привлек внимание лидеров данной индустрии — AMD и NVIDIA. Они начали разрабатывать собственные программные платформы для возможности реализовывать неграфические вычисления на своих видеокартах. Зная все особенности своей продукции, разработчики GPU смогли максимально эффективно оптимизировать программный комплекс для интересующих программистов аппаратных решений. В результате компания NVIDIA предложила нам платформу CUDA (Compute Unified Device Architecture), а компания AMD в свою очередь разработала свою технологию под названием CTM (Close To Metal) или, как называют ее разработчики, AMD Stream Computing [1]. Рассмотрим некоторые возможности технологии CUDA и на практике оценим вычислительные возможности видеокарты GTX 1070.

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ГРАФИЧЕСКОГО ПРОЦЕССОРА

Прежде всего стоит рассмотреть некоторые особенности при выполнении расчетов на графическом процессоре. Основное преимущество GPU заключается в том, что оно изначально проектируется для обработки множества потоков, CPU же выполняет поток последовательных инструкций на каждом своем ядре. Любой графический процессор представляет собой мультипроцессор, состоящий из определенного количества вычислительных кластеров, с множеством арифметико-логических устройств (АЛУ) (англ. *Arithmetic and Logic Unit*, ALU) в каждом. Например, графический

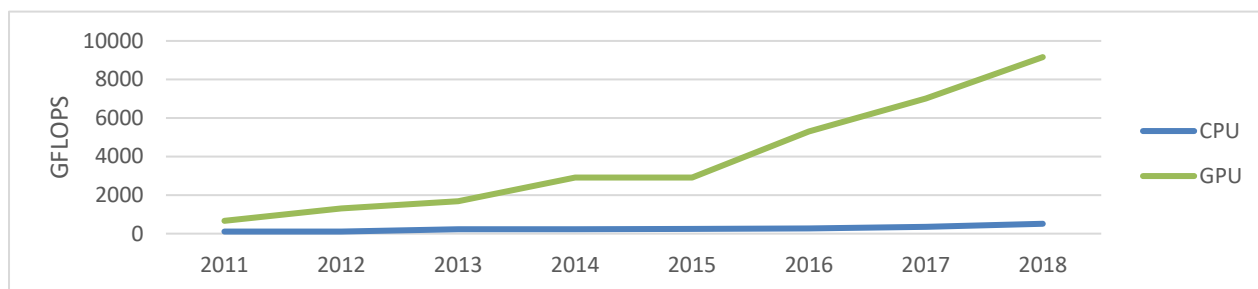


Рис. 1. График производительности GPU в сравнении с CPU

чип GP104 видеокарты GTX 1070 состоит из 1 920 потоковых вычислительных ядер и 120 блоков TMU (Texture Mapping Unit) при неизменных 64 блоках ROP (Raster Operator Unit). CPU используют SIMD (Single Instruction Multiple Data) блоки SSE для векторных вычислений где одна инструкция выполняется над многочисленными данными, что требует трансформации данных в 4 вектора. GPU же имеет параллельную вычислительную систему с архитектурой SIMT (Single Instruction Multiple Thread) в которой одна команда параллельно выполняется множеством относительно независимых потоков. Множество всех этих потоков, запущенных в рамках одной задачи, носит название grid (сетка). Это избавляет программистов от необходимости преобразовывать данные в векторы, а также позволяет использовать произвольные ветвления в потоках [2].

Другой важной особенностью является организация памяти в CUDA и распределение уровней доступа потоков к различным ее частям. Глобальная память — это наиболее простой вид памяти для понимания. Термин «глобальная» связан с тем фактом, что данный вид памяти доступен как со стороны других устройств, так и со стороны самого графического процессора как на запись, так и на чтение. В принципе, к ней может получить доступ любая составляющая часть устройства. Глобальная память (global memory) дает потокам наивысший уровень общего доступа за счет своей физической реализации в виде интегральных микросхем, расположенных непосредственно на самой плате графического адаптера. Расположение вне процессора делает данную память наиболее медленной по сравнению с другими типами памяти, предоставляемыми для вычислений на видеокарте.

Разделяемая память SM (shared memory) расположена в каждом SM-блоке, а ее размер обычно составляет 16 Кбайт. Данная память доступна исключительно тем потокам, которые выполняются на ядрах текущего SM-блока [3]. Так как для параллельного исполнения на одном SM может быть отведено более одного блока, то весь доступный в SM объем разделяемой памяти распределяется между этими блоками поровну. Физически разделяемая память расположена достаточно близко к ядрам SM, следовательно, она обладает высокой скоростью доступа.

Регистровая память — основной вид памяти, который может служить операндами элементарных машинных команд, а также обладает самым высоким быстродействием. Все регистры одного SM поровну разделяются между всеми потоками, запущенными на этом SM.

ТЕХНОЛОГИЯ CUDA

CUDA — вычислительная архитектура, разработанная компанией NVIDIA для реализации параллельных программ. Она позволяет программисту задействовать все возможные ресурсы графического процессора и направить их на реализацию высокопроизводительных приложений. Для программистов было реализовано специальное расширение для языка C, которое предоставляло возможность включать специальные функции в текст программы, тем самым позволяя разработчику организовывать доступ к набору инструкций GPU и управлять его памятью [4].

Запуск программы на графическом процессоре происходит следующим образом:

1. Центральный процессор выделяет необходимое количество памяти на устройстве.
2. Центральный процессор копирует данные из ОЗУ в память графического процессора.
3. Центральный процессор запускает функцию (задание) на графическом процессоре.
4. Графический процессор выполняет эту функцию (задание).
5. Центральный процессор копирует результаты из памяти графического процессора в ОЗУ.

Если время, потраченное на выполнение функции, окажется меньше времени, потраченного на выделение памяти и запуск этих функций, тогда использование графического процессора в данном случае бессмысленно.

Архитектура GPU отличается от архитектуры CPU. Графический процессор спроектирован таким образом, чтобы выполнять огромное количество потоков (элементарных параллельных процессов) [5]. Подробная архитектура центрального и графического процессора представлена на рисунке 2.

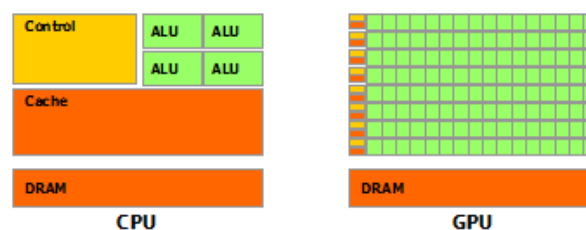


Рис. 2. Архитектура CPU и GPU

На рисунке видно, что в GPU есть много простых арифметико-логических устройств, которые объединены в несколько групп и обладают общей памятью. Это позволяет повысить эффективность в вычислительных функциях, однако усложняет программирование. Архитектура CPU также имеет сверхоперативную память (cache) и управляющее устройство (control), но гораздо меньше арифметико-логических устройств. Учитывая характеристики GPU и грамотно реализуя доступ к памяти, можно добиться наибольшей эффективности в вычислительных процессах.

Графический процессор предназначен для работы с большими объемами данных и расчетов. Он представляет из себя массив потоковых процессоров (Streaming Processor Array), который состоит из так называемых кластеров текстурных процессоров (Texture Processor Clusters, TPC). Каждый кластер текстурных процессоров состоит из мультипроцессоров (Streaming Multi-processor), в которых находятся несколько потоковых процессоров (Streaming Processors) или ядер [6]. GPU фактически является устройством, которое реализует вычислительную модель входящих и исходящих потоков данных. Они представляют из себя одинаковые элементы и имеют возможность обрабатываться независимо друг от друга. Каждый графический адаптер обладает количественной характеристикой скорости выполнения определенных операций. В NVIDIA эта характеристика обозначается как Compute Capability Version.

В CUDA потоки для расчетов группируются в иерархию — grid / block / thread. Их структура представлена на рисунке 3.

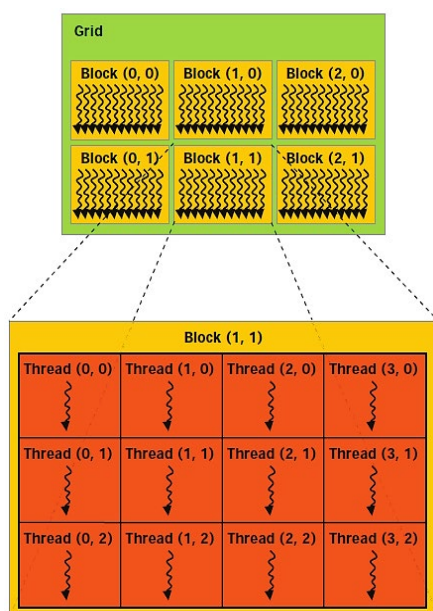


Рис. 3. Структура потоков в CUDA

Как видно из рисунка, верхний уровень (grid) отвечает ядру и объединяет все потоки (thread), которые выполняет данное ядро. Grid — одномерный или двумерный массив блоков (block), которые являются полностью независимыми наборами скоординированных между потоков (thread). Потоки из разных блоков не могут взаимодействовать. В CUDA есть такое понятие, как warp — группа из потоков (количество потоков зависит от архитектуры графического процессора). Потоки одного warp имеют возможность выполняться одновременно, а потоки разных warp могут быть на разных стадиях выполнения программ. Данные обрабатываются методом SIMT (Single Instruction — Multiple Threads), а управление работой warp выполняется на аппаратном уровне [7–8].

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА

Для проведения сравнительного анализа на языке C++ был разработан программный комплекс, выполняющий различные операции над матрицами разных размеров. Выбор матричных операций обоснован частым применением при проведении научных и других видов расчетов.

Работа программного комплекса делится на ряд определенных этапов.

Первый этап заключается в псевдослучайной генерации значений элементов матриц. Генерация осуществляется вызовом метода *generate* Стандартной Библиотеки Шаблонов (STL).

На втором этапе происходит определение доступных графических процессоров ЭВМ и задается точка отсчета для замеров времени работы алгоритмов программы. Замер времени производится с помощью библиотеки Chrono, входящей в состав STL.

На третьем этапе поочередно выполняется весь набор следующих операций:

1. Матричное умножение.
2. Поиск максимального значения.
3. Трансформация вектора.
4. Вычисление суммы двух векторов.
5. Сортировка.

6. Подсчет числа элементов, удовлетворяющих условию.

Все названные операции в программном комплексе выполняются как на графическом, так и на центральном процессоре.

Четвертый этап вычисляет время работы всех алгоритмов, фиксирует итоговое время работы для каждого типа процессора и выводит полученные значения на экран.

Вышеперечисленные операции реализованы на основе нескольких библиотек.

Матричное умножение на центральном процессоре происходит с применением библиотеки Armadillo.

Armadillo — это высококачественная библиотека линейной алгебры C++, нацеленная на достижение хорошего баланса между скоростью и простотой использования. Она полезна для разработки алгоритмов непосредственно на языке программирования C++ и быстрого преобразования исследовательского кода в производственные среды. Синтаксис (API) намеренно похож на Matlab [9].

Библиотека предоставляет эффективные классы для векторов, матриц и кубов, а также более 200 связанных функций (например, смежные и несмежные представления подматриц). Различные разложения матриц обеспечиваются за счет интеграции с LAPACK или одной из его высокопроизводительных замен (например, OpenBLAS, Intel MKL, AMD ACML, платформа Apple Accelerate и т. д.).

Сложный анализатор выражений (с помощью шаблонного метапрограммирования C++) автоматически объединяет несколько операций во время компиляции для увеличения скорости и эффективности. Библиотека может использоваться для машинного обучения, распознавания образов, компьютерного зрения, обработки сигналов, статистики и эконометрики.

Для реализации многопоточности при вычислении на центральном процессоре библиотека Armadillo обращается к OpenMP.

Интерфейс OpenMP задуман как стандарт для программирования на масштабируемых SMP-системах (SSMP, ccNUMA, etc.) в модели общей памяти (shared memory model) [10].

В стандарт OpenMP входят спецификации набора директив компилятора, процедур и переменных среды. Разработкой стандарта занимается организация OpenMP ARB (ARchitecture Board), в которую вошли представители крупнейших компаний-разработчиков SMP-архитектур и программного обеспечения. OpenMP можно рассматривать как высокоуровневую настройку над Pthreads (или аналогичными библиотеками нитей). За счет идеи «инкрементального распараллеливания» OpenMP идеально подходит для разработчиков, желающих быстро распараллелить свои вычислительные программы с большими параллельными циклами.

Разработчик не создает новую параллельную программу, а просто последовательно добавляет в текст последовательной программы OpenMP-директивы. При этом OpenMP — достаточно гибкий механизм, предоставляющий разработчику большие возможности контроля над поведением параллельного приложения. Многопоточное вычисление на графическом процессоре осуществляется благодаря библиотеке Thrust.

Thrust — это библиотека параллельного программирования C++, которая напоминает стандартную библиотеку. Высокоуровневый интерфейс Thrust значительно повышает производительность труда программистов, обеспечивая переносимость производительности между графическими процессорами и многоядерными процессорами [11]. Взаимодействие с установленными технологиями, такими как CUDA, TBB и OpenMP, облегчает интеграцию с существующим программным обеспечением.

Все классы и функции этой библиотеки — шаблонные. Для работы с этой библиотекой требуется подключить заголовочные файлы. В Thrust нет стандартных библиотечных файлов (.lib, .a, .dll, .so).

Ряд понятий и подходов Thrust заимствовала из STL. Thrust предоставляет набор различных параллельных примитивов, таких как различные преобразования, сортировка, операции *reduce* и *scan*. Применяя Thrust, многие действия могут быть записаны просто и понятно с использованием минимального объема кода. Все последние версии CUDA включают в себя Thrust, так что для работы с Thrust никаких дополнительных установок не понадобится.

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Моделирование проводилось на ПЭВМ со следующими характеристиками:

- видеокарта GTX 1070 (графический чип GP104);
- центральный процессор Intel Core i5-6600K;
- ОЗУ G.skill 16 Гб;
- операционная система Windows 10.

Рассмотрим полученные в ходе анализа результаты.

1. Матричное умножение.

Проведены замеры времени выполнения умножения для квадратных матриц различного размера от 85x85 до 25 000 x 25 000 элементов. Сравнение результатов времени выполнения матричного умножения для некоторых случаев представлены на рисунке 4.

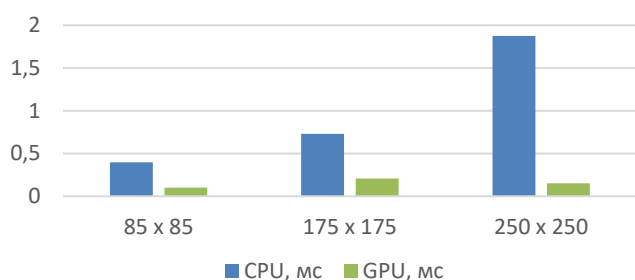


Рис. 4. Затраты времени на матричное умножение

Поскольку матричное умножение обладает высокой степенью распараллеливания, графический чип GP104 за счет большого числа ядер CUDA будет быстрее выполнять расчеты по сравнению с центральным процессором (рис. 4). Высокая степень распараллеливания достигается за счет возможности вычисления любого элемента результирующей матрицы без необходимости вычисления всех предыдущих.

В таблице 1 приведены замеры времени умножения для всех рассмотренных матриц.

Таблица 1

Матричное умножение

Размер матрицы, элементов	CPU, мс	GPU, мс	CPU/GPU
85 x 85	0,397	0,102	4
175 x 175	0,730	0,208	4
250 x 250	1,875	0,151	12
500 x 500	14,000	0,345	41
1 000 x 1 000	78,000	1,031	76
2 000 x 2 000	915,000	3,810	240
4 000 x 4 000	4 373,000	14,817	295
8 000 x 8 000	34 641,000	59,705	580
16 000 x 16 000	277 249,000	234,000	1185
25 000 x 25 000	1 215 248,000	579,000	2099

На основе таблицы 1 построим гистограмму отношения скорости вычисления на ЦП и ГП (рис. 5).

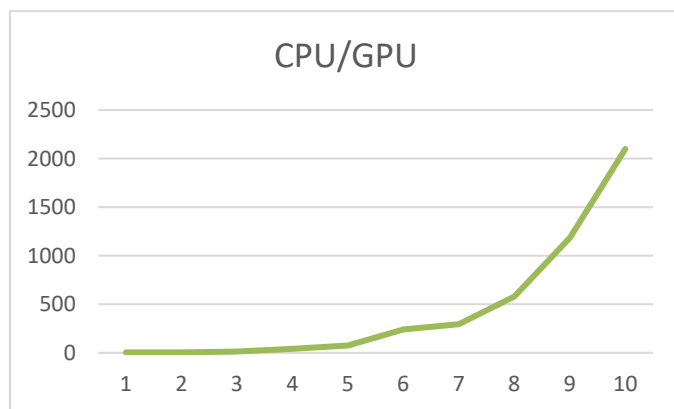


Рис. 5. Гистограмма отношения скорости вычисления на ЦП и ГП

На основе гистограммы можно сделать вывод о том, что с ростом размерности матрицы разница между требуемым временем на поведение операции на ЦП и ГП будет увеличиваться, причем нелинейно.

2. Поиск максимального значения в векторе.

Операция поиска максимального значения в векторе (массиве) очень распространена во всех областях при работе с данными.

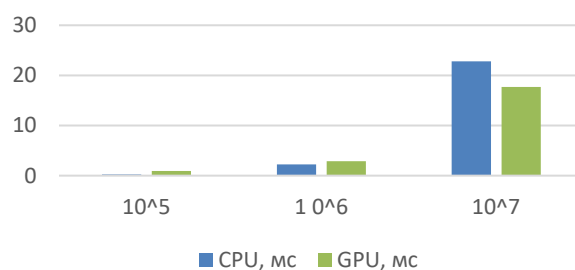


Рис. 6. Затраты времени на поиск максимального значения в векторе

Как следует из рисунка 6, при малых размерах вектора центральный процессор производит поиск быстрее. Это связано с тем, что при проведении вычислений на графическом процессоре, необходимо провести ряд «дорогих» по времени подготовительных и заключительных операций. К этим операциям относятся операции выделения

памяти в ГП, переноса данных из ОЗУ в память ГП, переноса результата из памяти ГП обратно в ОЗУ, а также операция освобождения памяти ГП.

Когда в векторе содержится всего несколько десятков значений, мы можем видеть, что требуется около 0,74 мс для проведения накладных операций (см. таблицу 2).

Все эти накладные операции сводят на нет смысл применения графического процессора при поиске максимального или минимального значения в относительно небольшом векторе или в случае поиска в наборе векторов небольшой размерности.

Таблица 2

Поиск максимального значения в векторе

Размер вектора, элементов	CPU, мс	GPU, мс
10	< 0,001	0,74
100	< 0,001	0,741
1 000	0,002	0,595
10 000	0,022	0,599
100 000	0,228	0,925
1 000 000	2,242	2,866
10 000 000	22,817	17,693
100 000 000	227,98	150,725

Однако, в случае поиска максимального или минимального значения, нужно учитывать еще и низкую приспособленность задачи к распараллеливанию.

3. Трансформация вектора.

Другой важной операцией над матрицами является трансформация. Суть этой операции заключается в применении заданной операции ко всем элементам матрицы.

В качестве заданной операции над вектором была взята следующая функция:

$$x = x \cdot x + x/2,$$

где x — текущий элемент матрицы.

В таблице 3 представлены замеры времени на проведение трансформации векторов различного размера.

Таблица 3

Трансформация векторов

Размер вектора, элементов	Подготовка GPU, мс	Вычисление на GPU, мс	Перенос результата, мс	Всего на GPU, мс	CPU, мс
10	0,493	0,048	0,041	0,582	< 0,001
100	0,493	0,069	0,068	0,632	< 0,001
1 000	0,372	0,063	0,069	0,505	0,001
10 000	0,493	0,073	0,077	0,643	0,009
100 000	1,050	0,072	0,186	1,31	0,095
1 000 000	2,156	0,13	1,238	3,526	1,001
10 000 000	17,678	0,862	18,97	37,511	9,662
100 000 000	169,608	8,194	141,852	319,654	95,821

В этом эксперименте помимо общего времени вычисления было измерено время, требуемое на выполнение накладных операций. Можно отметить, что за счет параллелизма сами вычисления на ГП выполняются очень быстро, однако, с учетом работы с памятью, итоговый результат получается гораздо хуже, чем у ЦП (рис. 7). Как и ожидалось, время на подготовительные операции в ГП намного превышает время на передачу результата.

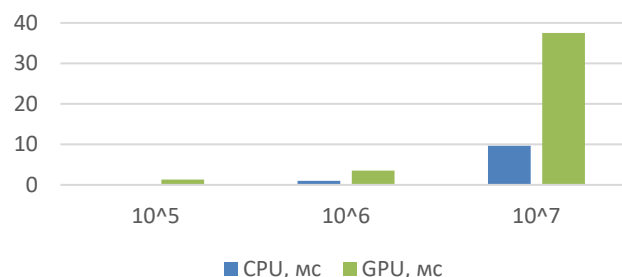


Рис. 7. Затраты времени на трансформацию вектора

4. Вычисление суммы двух векторов.

При вычислении суммы векторов наблюдается ситуация, аналогичная возникающей при поиске максимального или минимального значения в векторе (рис. 8).

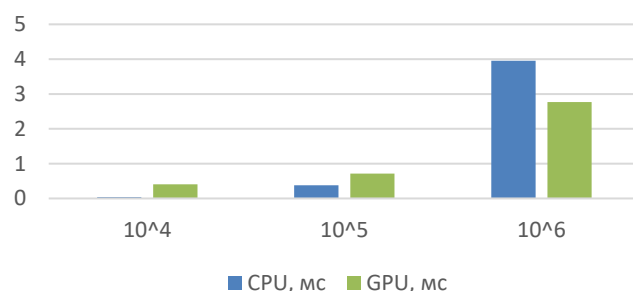


Рис. 8. Затраты времени на суммирование векторов

Результаты замеров времени суммирования различных векторов размещены в таблице 4. По результатам проведенных тестов можно утверждать, что при работе с векторами размером менее 1 000 000 применение ГП приведет к замедлению работы.

Таблица 4

Суммирование векторов

Размер вектора, элементов	CPU, мс	GPU, мс
10	< 0,001	0,664
100	< 0,001	0,451
1 000	0,003	0,398
10 000	0,037	0,408
100 000	0,377	0,716
1 000 000	3,953	2,770
10 000 000	38,335	19,316
100 000 000	387,087	152,244

Современные процессоры позволяют добиться высокого быстродействия за счет параллелизма на основе ядер и потоков. В процессорах Intel поддержка инструкций SIMD (single instruction, multiple data), которые обеспечивали параллелизм на уровне данных, впервые заявлена в серии Pentium III.

5. Сортировка вектора.

Еще одной часто встречаемой задачей при работе с данными является сортировка. Она может осуществляться с применением широкого спектра алгоритмов. В библиотеке STL есть метод *sort*, который сортирует вектор с применением распараллеливания на процессоре. Основанная на STL библиотека Thrust также содержит метод *sort*, который в свою очередь может производить сортировку уже с применением мощностей ГП.

Сортировка слиянием, применяемая в обеих библиотеках, позволяет раскрыть весь потенциал как центрального, так и графического процессора за счет разбиения на блоки.

Уже при сортировке вектора, содержащего 100 000 элементов, центральному процессору требуется более чем в два раза больше времени. При увеличении количества элементов разница существенно возрастает (рис. 9) даже несмотря на необходимость не только загружать весь вектор в память ГП, но и выгружать обратно уже отсортированный (см. таблицу 5).

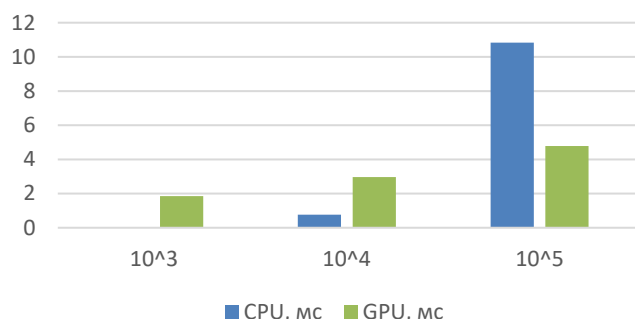


Рис. 9. Затраты времени на сортировку вектора

Таблица 5

Сортировка вектора

Размер вектора, элементов	CPU, мс	GPU, мс
10	< 0,001	1,095
100	0,003	1,818
1 000	0,035	1,853
10 000	0,769	2,970
100 000	10,829	4,787
1 000 000	88,604	6,391
10 000 000	879,600	76,652
100 000 000	5639,000	518,995

Как и в рассмотренных ранее тестах, при сортировке существует определенный порог, при котором ГП начинает существенно выигрывать по времени у ЦП.

6. Подсчет числа элементов, удовлетворяющих условию.

Последняя рассматриваемая операция — подсчет числа элементов, удовлетворяющих заданному условию. В этом случае требуется загрузить в память ГП весь вектор, а выгрузить лишь одно число, что позволит сократить время на работу с памятью. Однако, как показало исследование, этот факт не смог дать графическому процессору преимущество в скорости (рис. 10).

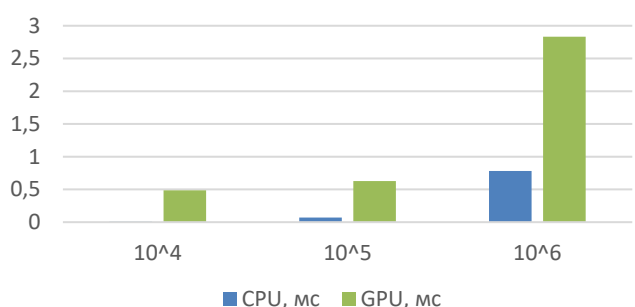


Рис. 10. Затраты времени на подсчет числа элементов, удовлетворяющих условию

Для примера рассмотрим результаты поиска в векторе, состоящем 100 000 000 элементов (табл. 6). Замеры времени на накладные операции показали, что из 154 мс, затраченных на поиск, 149 мс занял процесс передачи вектора из ОЗУ в память ГП. Таким образом, сам поиск занял всего лишь 3% от общего времени.

Таблица 6

Подсчет числа элементов, удовлетворяющих условию

Размер вектора, элементов	CPU, мс	GPU, мс
10	< 0,001	0,445
100	< 0,001	0,555
1 000	< 0,001	0,457
10 000	0,006	0,486
100 000	0,070	0,627
1 000 000	0,782	2,832
10 000 000	5,93	17,022
100 000 000	61,874	154,38

Этот случай показывает, что даже несмотря на то, что задача может быть пригодна для распараллеливания, результат зависит и от других факторов, которые могут сыграть ключевую роль при выборе процессора для проведения расчетов.

ЗАКЛЮЧЕНИЕ

Применение графических процессоров при обработке больших массивов данных, безусловно, может ускорить получение результата. На сегодняшний день продажи CUDA-процессоров достигли миллионов, а разработчики программного обеспечения, ученые и исследователи широко используют CUDA в различных областях, включая обработку видео и изображений, вычислительную биологию и химию, моделирование динамики жидкостей, восстановление изображений, полученных путем компьютерной томографии, сейсмический анализ, трассировку лучей и многое другое. Технология Nvidia CUDA хорошо подходит для решения широкого круга задач и существенно облегчает разработку ПО за счет развитого набора расширений для языков C и C++, позволяющих выражать как параллелизм данных, так и параллелизм задач на уровне мелких и крупных структурных единиц [12].

Конечно, графические процессоры не заменят центральные из-за особенностей своей архитектуры и работы. Наоборот, производители видеокарт стараются сделать видеочипы как можно более похожими по функционалу на ЦП — добавляют поддержку расчетов с одинарной и двойной точностью и т. п. В свою очередь, центральные процессоры развивают возможности параллелизма за счет увеличения числа ядер и поддержки новых инструкций, что является основным направлением развития этого типа процессоров.

Рассмотренные в ходе исследования варианты применения графического процессора показали, что далеко не каждая задача может быть эффективно решена на нем. Даже в случае хорошего распараллеливания накладные расходы времени, обусловленные особенностями архитектуры ГП, могут оказать существенное влияние на итоговое время работы.

На основе полученных результатов можно сформулировать следующие критерии для задач, которые могут быть решены на ГП:

- высокая степень распараллеливания;
- наличие большого количества вычислений;
- возможность загрузить данные до проведения вычислений.

Однако даже если задача удовлетворяет этим критериям, это не гарантирует ускорение ее решения на ГП, а лишь заставляет задуматься о применении технологии CUDA. Однозначно сделать выбор между ЦП и ГП можно только после тестирования. К счастью, благодаря хорошо продуманному синтаксису, для переноса вычислений на ГП не понадобится много времени.

ЛИТЕРАТУРА

1. Cheng J. Professional CUDA C Programming / J. Cheng, M. Grossman, T. McKercher. — John Wiley & Sons, 2014. — 523 p.
2. Sanders J. CUDA by Example: An Introduction to General-Purpose GPU Programming / J. Sanders, E. Kandrot. — Addison-Wesley Professional, 2011. — 309 p.
3. NVIDIA CUDA™ NVIDIA CUDA C Programming Guide. Version 4.1. — Santa Clara, CA, USA, 2011. — 168 p.
4. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие / А. В. Боресков, А. А. Харламов, Н. Д. Марковский [и др.]. — М.: Издательство МГУ, 2012. — 336 с. — (Суперкомпьютерное образование).
5. Боресков А. В. Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. — М.: ДМК Пресс, 2010. — 232 с.
6. Storti D. W. CUDA for Engineers: An Introduction to High-Performance Parallel Computing / D. W. Storti, M. Yurtoglu. — Addison-Wesley Professional, 2016. — 347 p.
7. Farber R. CUDA Application Design and Development. — Elsevier, Morgan Kaufmann, 2011. — 332 p.
8. Soyata T. GPU Parallel Program Development Using CUDA — Boca Raton, FL, USA: CRC Press, 2018. — 475 p.
9. Sanderson C. Armadillo: a template-based C++ library for linear algebra / C. Sanderson, R. R. Curtin // Journal of Open Source Software. 2016. Vol. 1. 26 p. DOI: 10.21105/joss.00026.
10. Van Der Pas R. Using OpenMP — The Next Step: Affinity, Accelerators, Tasking, and SIMD (Scientific and Engineering Computation) / R. Van Der Pas, E. Stotzer, C. Terboven. — Cambridge: The MIT Press, 2017. — 392 p.
11. Kirk D. B. Programming Massively Parallel Processors: A Hands-on Approach / D. B. Kirk, W-m. W. Hwu. — Third Edition. — Elsevier, Morgan Kaufmann, 2016. — 572 p.
12. Параллельные вычисления CUDA // Nvidia. URL: <http://www.nvidia.com.ua/object/cuda-parallel-computing-ru.html> (дата обращения 10.06.2020).

Comparative Analysis of Computing Speed on Central and Graphic Processor

Bachelor N. S. Maltsev, I. A. Molodkin, PhD in Military Science R. G. Gilvanov
Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
22nikich98@mail.ru, molodkin@pgups.ru, gilvanov1950@mail.ru

Abstract. The development of parallel computing technology on a graphics processor (GPU) is described, the organization of computing on a central and graphic processor, and internal architecture features are considered. The fundamental principles of programs using parallel computing based on advanced CUDA technology from NVIDIA are revealed. A study of the speed of a number of operations (matrix multiplication, maximum value search, vector transformation, calculation of the sum of 2 vectors) with different input data using the developed software package is made. The analysis of the results is carried out. Recommendations are given on the use of CUDA technology - the conditions under which it is advisable to perform calculations on a GPU (high degree of parallelization; a large amount of calculations; the ability to download data before calculations).

Keywords: benchmarking, computing speed, CUDA, central processing unit, graphics processor, parallel computing.

REFERENCES

1. Cheng J., Grossman M., McKercher T. Professional CUDA C Programming. John Wiley & Sons, 2014, 523 p.
2. Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, 2011, 309 p.
3. NVIDIA CUDA™ NVIDIA CUDA C Programming Guide. Version 4.1. Santa Clara, CA, USA, 2011, 168 p.
4. Boreskov A. V., Kharlomov A. A., Markovsky N. D., et al. Parallel computing on GPU. Architecture and software model of CUDA: Study Guide [Parallel'nye vychisleniya na GPU. Arkhitektura i programmaya model' CUDA: Uchebnoe posobie]. Moscow, Lomonosov Moscow State University, 2012, 336 p.
5. Boreskov A. V., Kharlamov A. A. Technology Basics of CUDA [Osnovy raboty s tekhnologiyey CUDA]. Moscow, DMK Press Publishers, 2010, 232 p.
6. Storti D. W., Yurtoglu M. CUDA for Engineers: An Introduction to High-Performance Parallel Computing. Addison-Wesley Professional, 2016, 347 p.
7. Farber R. CUDA Application Design and Development. Elsevier, Morgan Kaufmann, 2011, 332 p.
8. Soyata T. GPU Parallel Program Development Using CUDA. Boca Raton, FL, USA, CRC Press, 2018, 475 p.
9. Sanderson C., Curtin R. R. Armadillo: a template-based C++ library for linear algebra, *Journal of Open Source Software*, 2016, Vol. 1, 26 p. DOI: 10.21105/joss.00026.
10. Van Der Pas R., Stotzer E., Terboven C. Using OpenMP — The Next Step: Affinity, Accelerators, Tasking, and SIMD (Scientific and Engineering Computation). Cambridge, The MIT Press, 2017, 392 p.
11. Kirk D. B., Hwu W-m. W. Programming Massively Parallel Processors: A Hands-on Approach, Third Edition. Elsevier, Morgan Kaufmann, 2016, 572 p.
12. Parallel computing of CUDA [Parallel'nye vychisleniya CUDA], *NVidia*. Available at: <http://www.nvidia.com.ua/object/cuda-parallel-computing-ru.html> (accessed 10 June 2020).

Структура управляющей программы и способ для обнаружения обледенения на поверхности стрелочных переводов

аспирант Ш. Х. Султонов, бакалавр Н. А. Крицкий
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, Россия
Sultonovsh@yandex.ru, Thenikitakri@gmail.com

З. Р. Султонова
Навоийский государственный горный институт
Навои, Узбекистан
Sultonovazr@yandex.ru

Аннотация. Обоснована актуальность разработки системы обнаружения и контроля обледенения поверхности стрелочных переводов («стрелок») на железнодорожных путях. Разработана структура управляющей программы для системы обнаружения снега и наледи на контролируемой поверхности. Предложен способ обнаружения снега и наледи на поверхности стрелочных переводов. Техническим результатом использования названной системы является повышение чувствительности, точности и достоверности информации, уменьшение энергопотребления, увеличение надежности и ресурса стрелочных переводов («стрелок») на железнодорожных путях.

Ключевые слова: датчик обледенения, упреждающее управление, устройство обогрева, контрольная поверхность, алгоритм, структурная схема, прогнозирование, синтез, обледенение, нотация UML, подсистема.

ВВЕДЕНИЕ

Повышение скорости движения поездов в условиях решения задачи сокращения эксплуатационных расходов выдвигает новые требования к оборудованию железных дорог, в том числе и к электрообогреву. Наиболее важными являются повышение срока службы и сокращение затрат на их обслуживание.

Высокие требования, в свою очередь, вызывают необходимость поиска новых технических решений, в частности таких, как выбор энергоэффективных, надежных, экологичных систем, отличающихся повышенной точностью измерения физических величин и, как следствие, получения более точной информации для принятия решений в системе управления.

На железных дорогах для предотвращения отказов срабатывания стрелочных переводов и других механических устройств в зимнее время вследствие напессовывания снега и обледенения широко применяют устройства для их локального обогрева.

Многообразие технических средств борьбы со снегом и льдом можно классифицировать по их разновидностям (видам и подвидам — типам): механические, пневматические, тепловые и комбинированные. Среди тепловых (теплообменных) устройств самое широкое применение нашли электрические.

АКТУАЛЬНОСТЬ РАЗРАБОТКИ СИСТЕМЫ ОБНАРУЖЕНИЯ И КОНТРОЛЯ ОБЛЕДЕНЕНИЯ ПОВЕРХНОСТИ СТРЕЛОЧНЫХ ПЕРЕВОДОВ

В настоящее время для очистки стрелочного перевода от снега и льда путем электрообогрева применяется

устройство СЭИТ-04М, содержащее электронагреватели, установленные на рамных рельсах, датчик температуры окружающего (наружного) воздуха, коммутатор, датчик осадков, элемент сравнения, в котором датчик температуры наружного воздуха соединен со вторым входом элемента сравнения, при этом устройством содержит только один датчик температуры рельса, установленный на рамных рельсах и соединенный с первым входом элемента сравнения [1–4].

При всех положительных качествах устройства-аналога причинами, препятствующими достижению технического результата, являются следующие его недостатки:

1. Не пояснено, к какому типу (по принципу действия — «сравнения входных сигналов») относится электронное устройство обработки данных (ЭУОД), равно как не расшифрованы понятия «специальный алгоритм» и «специальное программное обеспечение». При этом есть основания предполагать выполнение ЭУОД в виде (по типу) автоматического регулятора, который уступает по своим возможностям современному типу ЭУОД — логических устройств (логических устройств, реализующих логические операции/функции над входными сигналами согласно правилам алгебры логики и выдающих сигналы, соответствующие значениям реализуемых функций).

2. Не оговорены возможность/невозможность оперативного изменения указанных алгоритма и программного обеспечения, вмешательства управляющего персонала (в том числе диспетчеров) различных уровней в алгоритм или в виде ручного двух- и трехуровневого дублирования с приоритетным ранжированием. Это отрицательно сказывается на технико-логистических возможностях прототипа при разветвленной реализации устройства, увеличивает вероятность аварийных ситуаций, сбоях организационного характера.

3. Недостаточно использованы современные возможности цифровых технологий, автоматизированных рабочих мест оператора.

СТРУКТУРА УПРАВЛЯЮЩЕЙ ПРОГРАММЫ

Задачей, на решение которой направлено заявляемое изобретение, является повышение технико-эксплуатационных и экономических характеристик и расширение технико-эксплуатационных возможностей устройства электрообогрева стрелочных переводов (стрелок) или иных локальных участков железнодорожной инфраструктуры за счет повышения надежности, снижения энергозатрат и

обеспечения возможности местного и многоуровневого дистанционного управления, и контроля [5].

Одним из требований системы устройства обнаружения осадков в виде льда или снега на контролируемой поверхности является регулярное обновление измерения о наличии или отсутствии льда, а также снега на рабочей поверхности датчика обледенения. В ответ на запрос ЭУОД чувствительные элементы датчика передают ему свои данные.

ЭУОД объединяет данные (оцифрованные сигналы) из установленных источников. Собранные данные архивируются, и с помощью данных из этого архива и установленного решающего правила (выработка признака наличия осадков) данной системы создается заключение о наличии или отсутствии льда, а также снега на рабочей пластине поверхности датчика по установленному критерию, основанному на использовании явления временной, по крайней мере, частичной, стабилизации температуры рабочей поверхности [2].

Значения сигналов, накопленные в ходе цикла, аккумулируются в ОЗУ электронного устройства обработки данных и передаются через последовательный канал связи на

внешней флэш-памяти, которая сохраняет эти данные вместе с состоянием признака, вырабатываемого оптическим датчиком, и текущим временем в виде файла. Полученные данные можно распечатать, направив их на специальный принтер, или же отобразить в различных форматах.

Из описания видно, что одна часть общей системы занимается сбором данных, другая обобщает данные, полученные из установленных чувствительных элементов (источников), третья выполняет архивирование данных, и, наконец, четвертая вырабатывает признак наличия или отсутствия осадков. На рисунке 1 изображена одна из возможных архитектур системы, которую можно построить на основе предложенного описания. Она представляет собой многоуровневую архитектуру, в которой отражены все этапы обработки данных в системе, то есть сбор, обобщение и архивирование данных и выработка признака осадков. Такая многоуровневая архитектура вполне годится для данной системы, так как каждый этап основывается на обработке данных, выполненной на предыдущем этапе.

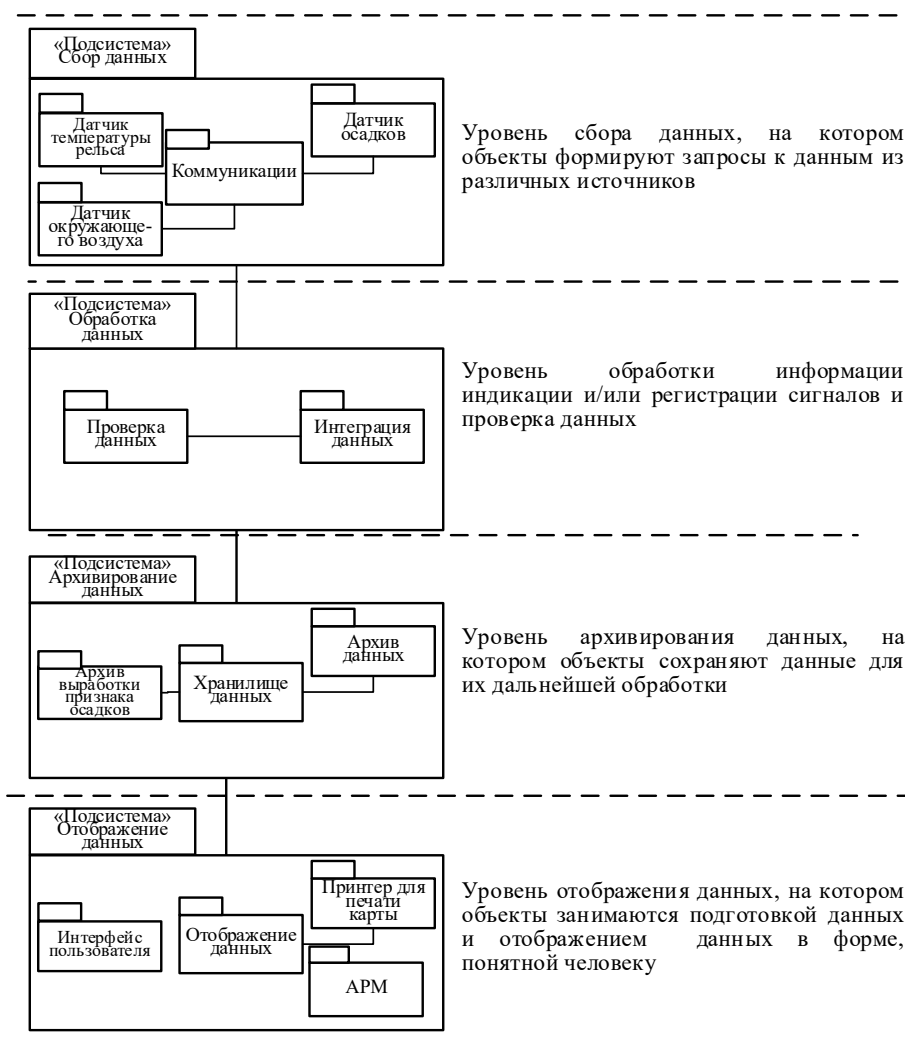


Рис. 1. Многоуровневая архитектура системы обнаружения и контроля обледенения поверхности

На рисунке 1 показаны все уровни системы, а также компоненты подсистем. Эти компоненты очень абстрактны

и построены на информации, содержащейся в описании системы. Названия уровней заключены в прямоугольники,

что в нотации UML обозначает подсистемы (набор объектов и других подсистем). Это обозначение используется для того, чтобы показать, что каждый уровень включает в себя множество других компонентов.

ОКРУЖЕНИЕ СИСТЕМЫ КОНТРОЛЯ НАЛЕДИ И МОДЕЛИ ЕЕ ИСПОЛЬЗОВАНИЯ

Процесс проектирования состоит в выявлении взаимоотношений между проектируемым программным обеспечением и его окружением. Выявление этих взаимоотношений помогает решить, как обеспечить необходимую функциональность системы и как структурировать систему, чтобы она могла эффективно взаимодействовать со своим окружением.

Модель окружения системы с помощью пакетов языка UML можно представить в развернутом виде как совокупность подсистем. Такое представление показывает, что рабочее окружение системы противообледенения находится внутри подсистемы, занимающейся сбором данных. Там же показаны другие подсистемы, которые образуют систему контроля состояния обледенения или снега на контролируемой поверхности [6–9].

Система противообледенения взаимодействует с внешними объектами во время запуска и завершения работы, при составлении отчетов (при выработке анализа (признака) наличия осадков) на основе собранных данных, а также при тестировании и калибровке метеорологических приборов (рис. 2).

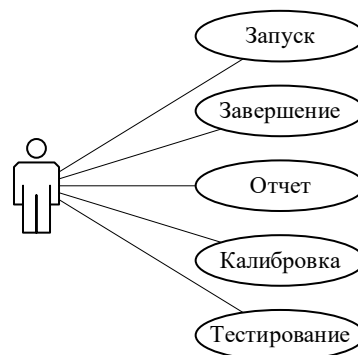


Рис. 2. Варианты использования системы противообледенения

Таблица 1

Описание варианта использования выработки признака наличия осадков

Система	Коммуникация
Вариант использования	Выработка признака наличия осадков
Участники	Система сбора данных, система противообледенения, сравнительный элемент
Данные	По окончании цикла измерения элемент сравнения отправляет сводку с данными, снятыми системой сбора данных с чувствительных элементов в определенный временной период. В сообщении содержатся начальная температура рабочей поверхности пластины данного чувствительного элемента, время с момента включения нагревателя и время до значения, превышающего температуру фазового превращения воды «твердое-жидкое» (0 °C)
Входные сигналы	Датчик подключают к аппаратной части с устройствами управления, измерения, обработки информации индикации и/или регистрации сигналов и передачи данных, входящей, наряду с датчиком, в состав устройства обнаружения обледенения или снега на контролируемой поверхности.
Ответ	Итоговые данные отправляются в систему сбора противообледенительных данных
Комментарии	Запрашивается отчет по установленному критерию, основанному на использовании явления временной стабилизации; эта частота запросов может отличаться для разных станций, а также может измениться в будущем. Делается заключение о наличии или отсутствии льда или снега на рабочей пластине датчика по установленному критерию, основанному на использовании явления временной стабилизации.

Описание специального алгоритма информационно-управляющей системы обнаружения обледенения или снега на контролируемой поверхности предлагается вести в виде блок-схем. Блок-схемы позволяют представить алгоритмы в обозримом виде, что дает возможность анализировать их работу, искать логические ошибки в процедуре их реализации [10–12]. Типовая блок-схема (1 — логический элемент «ИЛИ»; & — логический элемент «И»; Tr — RS-триггер) приведена на рисунке 3.

Стрелки в блок-схеме указывают последовательность логических операций в зависимости от значения параметров и выходных сигналов логических элементов схемы в текущий момент времени.

После создания типовой блок-схемы необходимо определить структурное построение информационно-управляющей системы, способной реализовывать полученные алгоритмы управления. В общем виде структурную схему информационно-управляющей системы обнаружения снега и наледи на контролируемой поверхности можно представить в соответствии с рисунком 4.

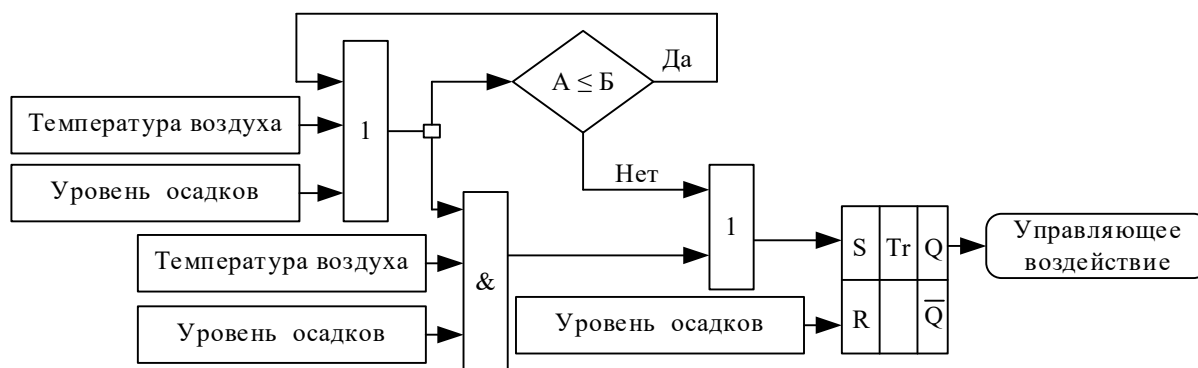


Рис. 3. Типовая блок-схема алгоритма формирования управляющего воздействия

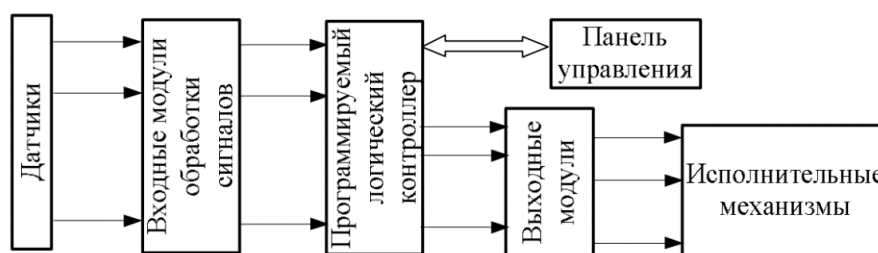


Рис. 4. Типовая структурная схема информационно-управляющей системы обнаружения снега и наледи на поверхности

СПОСОБ ОПРЕДЕЛЕНИЯ ОБЛЕДЕНЕНИЯ ИЛИ СНЕГА

С целью определения эффективности разработанного алгоритма формирования управляющих воздействий системы предлагается способ определения обледенения или снега на контролируемой поверхности с возможностью прогноза.

На контролируемой поверхности устанавливают датчики, имеющие два чувствительных элемента, снабженных теплопроводными пластинами с внешними рабочими поверхностями для воздействия окружающей среды, встроенными датчиками температуры пластин и нагревателями на тыльной их поверхности [13, 14].

Отслеживание изменения во времени температуры рабочей поверхности обоих чувствительных элементов определяется по формуле (1) посредством датчика контроля состояния.

$$T_1 = f_1 \cdot (t), T_2 = f_2 \cdot (t), \quad (1)$$

где T_1, T_2 — температуры рабочей поверхности; f_1, f_2 — функции эмпирической зависимости; t — время с момента включения нагревателя до достижения значения фазового превращения воды.

Система определяет значения разности, обеспечивая, совместно с асинхронностью, включение нагревателей, аннулирование вкладов полезных сигналов, обусловленных плавлением льда или снега, и обнуление величины вкладов вредных сигналов, обусловленных воздействием воздушных потоков согласно формуле (2).

$$\Delta T(t) = T_1(t) - T_2(t). \quad (2)$$

При наличии на пластине льда и/или снега вклады полезных сигналов в моменты времени, соответствующие фазовому переходу, аннулируются ($\Delta T(t) = 0$) (рис. 5).

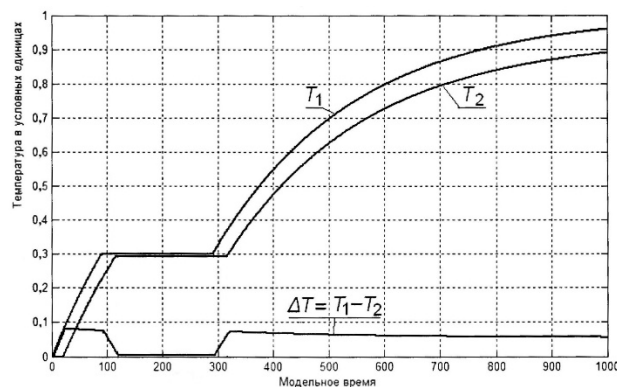


Рис. 5. Временная стабилизация температуры пластин T_1, T_2 на уровне фазового перехода

Обнуление разности $T_1(t) - T_2(t)$ свидетельствует о наличии осадков на датчике и в локальной зоне контролируемой поверхности.

ЗАКЛЮЧЕНИЕ

Предлагаемый способ обнаружения осадков позволяет построить алгоритм информационно-управляющей системы предупреждения обледенения или снега на поверхности обогреваемого рельса на основе формирования требований до значения, превышающего температуру фазового превращения воды «твердое-жидкое» — 0°C .

ЛИТЕРАТУРА

1. Патент № 2582627 Российская Федерация, МПК E01B 7/24 (2006.01). Устройство электрообогрева стрелочных переводов типа СЭИТ-04: № 2015106701/11: заявл. 26.02.2015: опубл. 27.04.2016 / Бараусов В. А., Кочубей В. Ф. — 18 с.
2. Патент № 2685631 Российская Федерация, МПК E01B 7/00 (2006.01), E01B 19/00 (2006.01), E01H

8/08 (2006.01), В64D 15/20 (2006.01). Способ и устройство обнаружения обледенения или снега на контролируемой поверхности: № 2018125617: заявл. 11.07.2018; опубл. 22.04.2019 / Селянин С. Г.; заявитель Барausов В. А., Григорьев П. В. — 20 с.

3. Wołoszyn M. Analysis of Resistive and Inductive Heating of Railway Turnouts / M. Wołoszyn, K. Jakubiuk, M. Flis // *Przegląd Elektrotechniczny*. 2016. No. 4. С. 54–57.

DOI: 10.15199/48.2016.04.14.

4. Brodowski D. Railway Turnouts Heating — The New Method. Contactless Heating as a Way to Melt Snow in the Turnout Faster. Technical information. — Warsaw, Instytut Kolejnictwa, 2012.

5. Barausov V. A. Control Software for Surface Ice and Snow Detecting Device / V. A. Barausov, V. P. Bubnov, Sh. Kh. Sul-tonov // *Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019)*. CEUR Workshop Proceedings. 2020. Vol. 2556. Pp. 75–79.

6. Султонов Ш. Х. Программа для автоматизации вычислений величин нагрузок, действующих на кузов вагона при прочностных расчетах / Ш. Х. Султонов, В. П. Бубнов // *Известия СПбГЭТУ «ЛЭТИ»*. 2019. № 9. С. 37–43.

7. Bubnov V. P. On the Application of Information Technology for the Development of Software for Automation the Modeling Process of Wagons at Wagon Enterprises JSC «O'zbekiston Temir Yo'llari» / V. P. Bubnov, Sh. Kh. Sul-tonov // *Вестник ТашИИТ*. 2019. № 2. С. 84–92.

8. Бубнов В. П. Алгоритм автоматизации проектно-конструкторских работ и его применение в вагоностроении / В. П. Бубнов, Ш. Х. Султонов // *Проблемы математической*

и естественнонаучной подготовки в инженерном образовании: Сборник трудов IV Международной научно-методической конференции (Санкт-Петербург, 03 ноября 2016 г.). — Т. 2. — СПб.: ПГУПС, 2017. — С. 16–21.

9. Модели и методы исследования информационных систем: Монография / А. Д. Хомоненко, В. П. Бубнов, А. В. Забродин [и др.]; под ред. А. Д. Хомоненко. — СПб.: Лань, 2019. — 204 с. — (Учебники для вузов. Специальная литература).

10. Методы интеграции инструментальных систем в процессе разработки безопасных приложений / С. Е. Ададуров, А. В. Красновидов, И. В. Коротеев, А. Д. Хомоненко // *Проблемы информационной безопасности. Компьютерные системы*. 2017. № 4. С. 80–86.

11. Бей И. Взаимодействие MATLAB с ANSI C, Visual C++, Visual Basic и Java / И. Бей, Д. Ксу. — М.: Вильямс, 2005. — 207 с.

12. Смоленцев Н. К. Создание Windows-приложений с использованием математических процедур MATLAB. — М.: ДМК-Пресс, 2008. — 456 с.

13. Свидетельство о государственной регистрации программы для ЭВМ Российская Федерация. Программа для устройства обнаружения обледенения или снега на контролируемой поверхности: № 2020612292/69 / В. А. Барausов, П. В. Григорьев, Ш. Х. Султонов.

14. Свидетельство о государственной регистрации программы для ЭВМ № 2019661192 Российская Федерация. Программа для оценки силового воздействия подвижного состава на железнодорожный путь: № 2019660103: заявл. 06.08.2019; опубл. 21.08.2019 / Р. В. Рахимов. — 1 с.

Structure of the Control Program and Method for Detecting Icing on the Surface of Switches

PhD student Sh. Kh. Sultonov, Bachelor N. A. Kritsky
Emperor Alexander I Peterburg State Transport University
Saint Petersburg, Russia
Sultonovsh@yandex.ru, Thenikitakri@gmail.com

Z. R. Sultonova
Navoi State Mining Institute
Navoiy, Uzbekistan
Sultonovazr@yandex.ru

Abstract. The article considers the process of developing the structure of the control program for detecting snow and ice on the controlled surface, choosing the optimal method for detecting snow and ice, and also considers the essence of the development of a system for detecting and controlling icing on the surface of switches ("arrows") on railway tracks, the technical result of which is to increase the sensitivity, accuracy and reliability of information, as well as to reduce energy consumption, increase reliability and resource.

Keywords: icing sensor, ice thickness, electronic data processing device (EDI), comparison element, electric heating systems, UML notation, subsystem, use case.

REFERENCES

1. Barausov V. A., Kochubej V. F. Electrical Heating Device of Track Switches Type SEIT-04 [Ustroystvo elektroobogreva strelochnykh perezodov tipa SEIT-04], patent RU No. 2582627, published at April 27, 2016, 18 p.
2. Selyanin S. G. Method and Device of Detecting Icing or Snow on a Controlled Surface [Sposob i ustroystvo obnaruzheniya obledeneniya ili snega na kontroliruemoy poverkhnosti], patent RU No. 2685631, published at April 22, 2019, 20 p.
3. Wołoszyn M., Jakubiuk K., Flis M. Analysis of Resistive and Inductive Heating of Railway Turnouts, *Przegląd Elektrotechniczny*, 2016, No. 4, Pp. 54–57. DOI: 10.15199/48.2016.04.14.
4. Brodowski D. Railway Turnouts Heating — The New Method. Contactless Heating as a Way to Melt Snow in the Turnout Faster. Technical information. Warsaw, Railway Research Institute, 2012.
5. Barausov V. A., Bubnov V. P., Sultonov Sh. Kh. Control Software for Surface Ice and Snow Detecting Device, *Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019)*, *CEUR Workshop Proceedings*, 2020, Vol. 2556, Pp. 75–79.
6. Sultonov Sh. Kh., Bubnov V. P. Program for Automation of Calculations of Load Values, Acting on a Wagon Body at Electronic Calculations with Further Three-Dimensional Modeling [Programma dlya avtomatizatsii vychisleniy velichin nagruzok, deystvuyushchikh na kuzov vagona, pri prochnostnykh raschetakh], *Proceedings of Saint Petersburg Electrotechnical University [Izvestiya SPbGETU «LETI»]*, 2019, № 9, Pp. 37–43.
7. Bubnov V. P., Sultonov Sh. Kh. On the Application of Information Technology for the Development of Software for Automation the Modeling Process of Wagons at Wagon Enterprises JSC «O'zbekiston Temir Yo'llari», *Journal of Tashkent Institute Railway Engineering [Vestnik TashIIT]*, 2019, No. 2, Pp. 84–92.
8. Bubnov V. P., Sultonov Sh. Kh. Algorithm of Automation of Design and Development Work and its Application in Railroad Cars Building [Algoritm avtomatizatsii proektno-konstruktorskikh rabot i ego primenenie v vagonostroenii]. In: *Problems of Mathematical and Natural Science Training in Engineering Education: Proceedings of the IV International Scientific and Methodological Conference [Problemy matematicheskoy i estestvennonauchnoy podgotovki v inzhenernom obrazovanii: Sbornik trudov IV Mezhdunarodnoy nauchno-metodicheskoy konferentsii]*, Saint Petersburg, November 03, 2016, Volume 2, St. Petersburg, PSTU, 2017, Pp. 16–21.
9. Khomonenko A. D., Bubnov V. P., Zabrodin A. V., et al. Models and methods of research of information systems: Monograph [Modeli i metody issledovaniya informatsionnykh sistem: Monografiya], St. Petersburg, LAN Publishing House, 2019, 204 p.
10. Adadurov S. E., Krasnovidov A. V., Koroteev I. V., Khomonenko A. D. Methods of Integration of Instrumental Systems in Development Process of Safe Applications [Metody integratsii instrumental'nykh sistem v protsesse razrabotki bezopasnykh prilozheniy], *Information Security Problems. Computer Systems [Problemy informatsionnoy bezopasnosti. Komp'yuternye sistemy]*, 2017, No. 4, Pp. 80–86.
11. Bay I., J. Xu. Interaction of MATLAB with ANSI C, Visual C ++, Visual Basic and Java [Vzaimodeistvie Matlab s ANSI C, Visual C ++, Visual Basic i Java], Moscow, Williams Publishing House, 2005, 207 p.
12. Smolentsev N. K. Creating Windows applications using MATLAB mathematical procedures [Sozдание Windows-prilozheniy s ispol'zovaniem matematicheskikh protsedur MATLAB], Moscow, DMK-Press Publ. House, 2008, 456 p.
13. Barausov V. A., Grigoryev P. V., Sultonov Sh. Kh. Software for Detecting Icing or Snow on a Controlled Surface [Programma dlya ustroystva obnaruzheniya obledeneniya ili snega na kontroliruemoy poverkhnosti], Certificate of State registration of a computer program.
14. Rakhimov R. V. Program for Assessing the Power Impact of Rolling Stock on Railway Tracks [Programma dlya otsenki silovogo vozdeystviya podvizhnogo sostava na zheleznodorozhnyy put'], Certificate of State registration of a computer program RU No. 2019661192, published at August 21, 2019.

Применение нейронных сетей для определения физического состояния человека

магистр Р. А. Сергеев, д.т.н. В. П. Бубнов

Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
roma.svetl@gmail.com, bubnov1950@yandex.ru

Аннотация. Рассматривается разработанное мобильное приложение, которое представляет собою систему мониторинга физического состояния работников железной дороги. Работа этой системы заключается в том, что пользователь (работник на РЖД) надевает фитнес-браслет Mi band 2 и подключает его к android-смартфону по Bluetooth, а специально разработанное приложение автоматически замеряет пульс и отправляет его на сервер для обработки данных с целью получения результатов состояния работника.

Ключевые слова: база данных, нейронная сеть, android-приложение, физическое состояние человека, фитнес-браслет, нейросетевое прогнозирование, разработка андроид-приложения.

ВВЕДЕНИЕ

В связи с повышенными нагрузками на физическое состояние работников РЖД, становится актуальной задача мониторинга и контроля физического состояния работника.

Целью данной работы является разработка мобильного приложения, контролирующего физическое состояние работников железной дороги, для операционной системы Android. Данное приложение снижает риск возникновения аварийных ситуаций и повышает безопасность пассажиров и сотрудников РЖД.

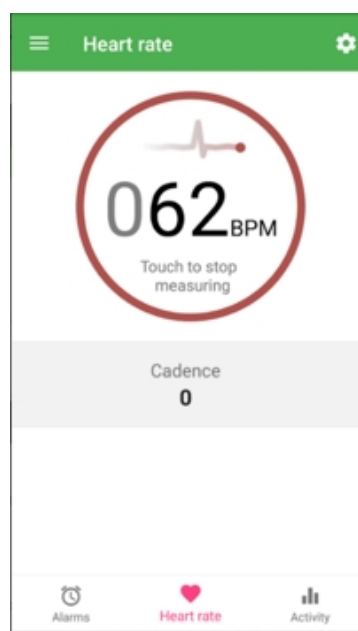
АНАЛИЗ РЫНКА

Проанализировав рынок на наличие аналогичных приложений мониторинга и контроля физического состояния работника, мы выявили несколько приложений, три из которых близки по смыслу к системе, описываемой в этой работе.

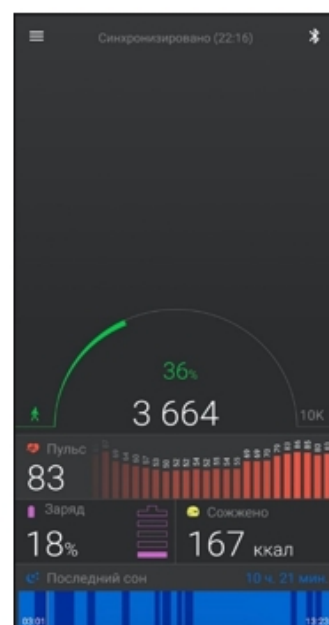
1. Mi HR with Smart Alarm — be fit Band (рис. 1, а). Отличное приложение, в котором имеется множество опций для различных тренировок, также позволяет на протяжении всей тренировки отслеживать пульс, а позже просматривать его в виде удобного графика. К сожалению, в нем нет необходимого функционала, а именно «анализа физического состояния и определения его статуса».

2. Master for Mi Band (рис. 1, б). Замечательное приложение для работы с умным браслетом Mi Band и умными весами Mi Scale имеет такие функции, как: шагомер, измеритель пульса, умный будильник. К сожалению, в нем нет необходимого функционала.

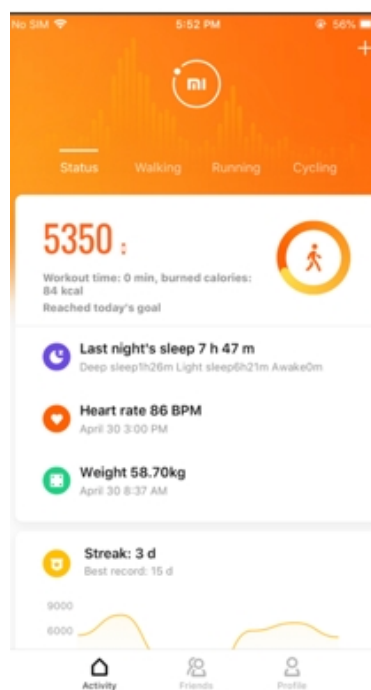
3. Mi Fit (рис. 1, в). Это официальное приложение от создателей данного фитнес-браслета. Mi Fit отслеживает активность, анализирует качество сна и оценивает прогресс ваших тренировок, но, как и в предыдущих двух приложениях, в нем отсутствует необходимый функционал.



а)



б)



в)

Рис. 1. Android-приложения мониторинга и контроля физического состояния работника

Все эти приложения позволяют измерить пульс человека и просмотреть его историю, но ни в одном приложении нет функционала для предсказания физического состояния человека по его физиологическим параметрам и пульсу в конкретный момент времени, а также отсутствует веб-панель для отслеживания целой группы людей одним оператором. В связи с этим было решено разработать данное Android-приложение.

ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

В данном разделе представлены UML-диаграммы взаимодействия разрабатываемого Android-приложения с сервером.

МОДЕЛИРОВАНИЕ ОПЕРАЦИЙ

Согласно правилам взаимодействия программ в сети, приложения должны обмениваться четко структурированными и согласованными пакетами данных [1]. В разработанной системе используется формат JSON для обмена данными между приложением и сервером.

При выполнении запросов от приложения к серверу неприемлемо блокирование потока работы Android-приложения, поэтому все запросы к серверу приложение выполняет в отдельном потоке [2].

На рисунке 2 представлена диаграмма процесса регистрации пользователя в системе.

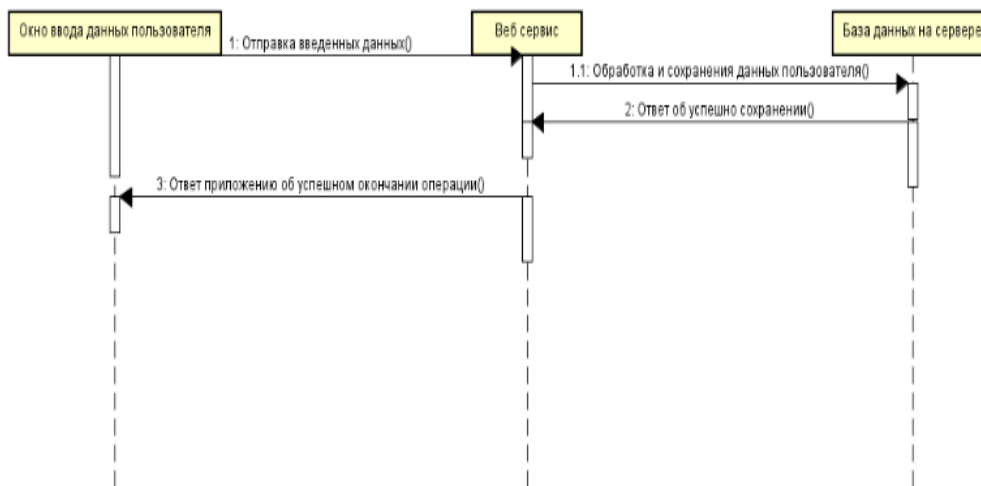


Рис. 2. Диаграмма последовательности процесса регистрации пользователя в системе

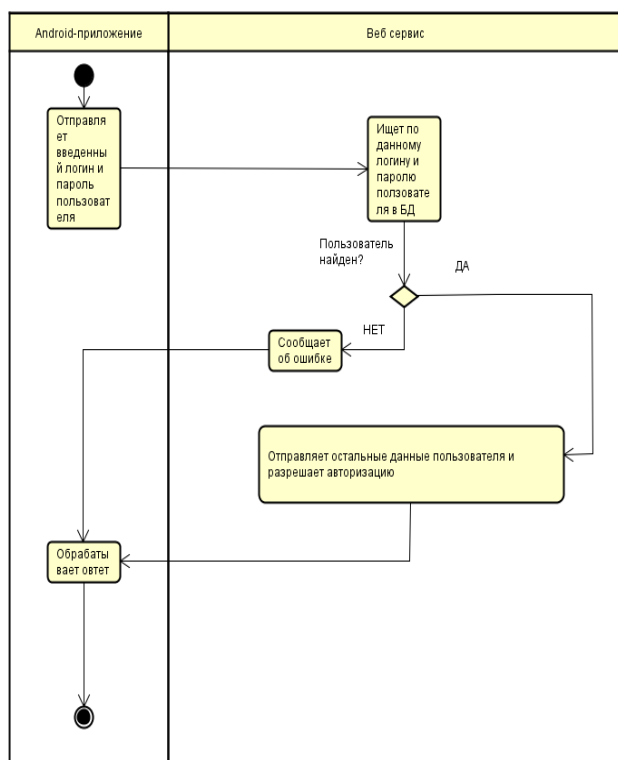


Рис. 3. Диаграмма процесса авторизации пользователя в системе

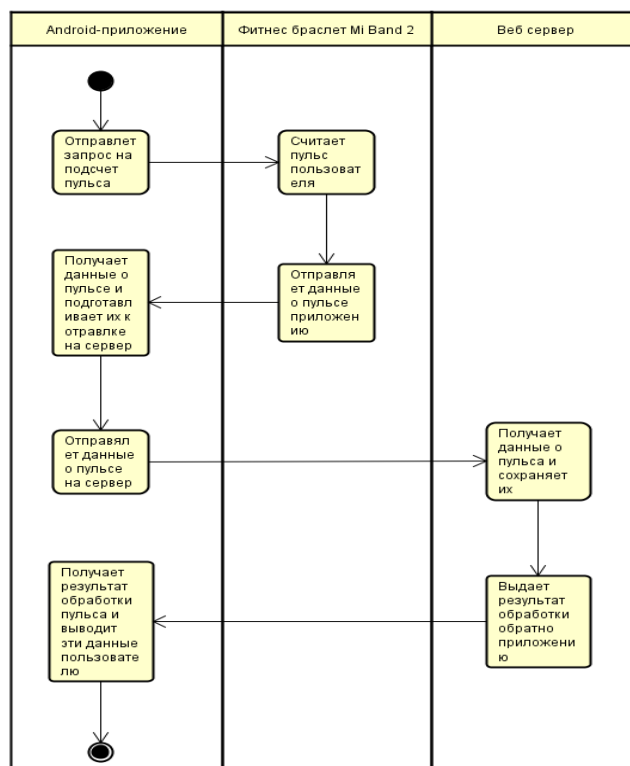


Рис. 4. Диаграмма процесса подсчета пульса и отправки его значения на сервер для последующей обработки

Чтобы избежать потерь персональных данных пользователя, все пароли хешируются, приложение ведет взаимодействие с базой данных через специальное серверное API. Главным преимуществом данного API является способность принимать большие потоки данных, моментально их обрабатывать и выдерживать большие нагрузки.

На рисунке 3 представлена диаграмма процесса авторизации пользователя в системе.

При авторизации очень важно точно идентифицировать нужного пользователя по логину и паролю, чтобы избежать потери его персональных данных. Для этого используется специально разработанное серверное API.

На рисунке 4 представлена диаграмма процесса подсчета пульса пользователя и отправка его значения на сервер для последующей обработки.

На рисунке 5 представлена диаграмма процесса подключения фитнес-браслета к приложению.

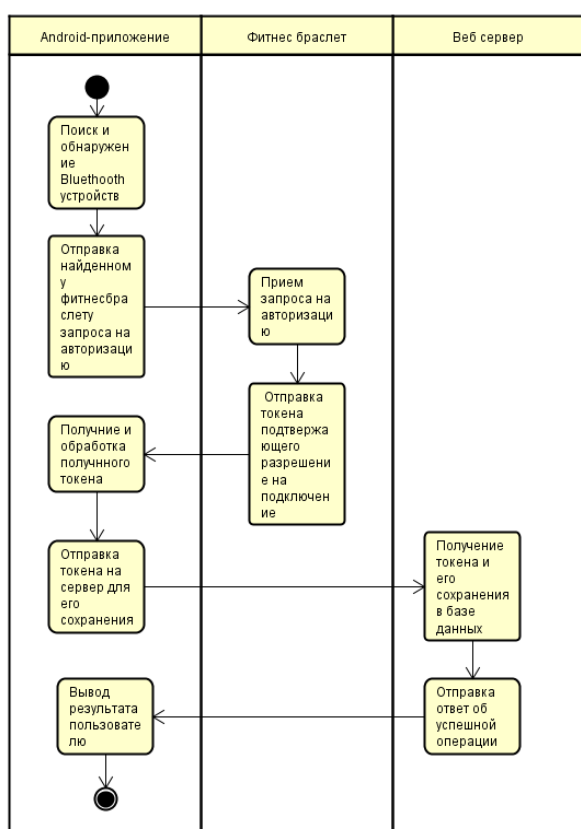


Рис. 5. Диаграмма процесса подключения фитнес-браслета к приложению

Для связи приложения с фитнес-браслетом используется беспроводная сеть Bluetooth.

Главной особенностью фитнес-браслета является то, что на браслете запущены все необходимые сервисы для получения информации о пульсе в виде сокетов. Благодаря этому разработанное приложение может получить токен для доступа к этим сокетам и получать данные о пульсе с фитнес-браслета. При таком способе взаимодействия браслет выступает в роли сервера, а приложение — в роли клиента.

Преимущества данного способа связи заключаются в экономии заряда батареи на обоих устройствах [3].

На рисунке 6 представлена диаграмма получения истории пульса пользователя.

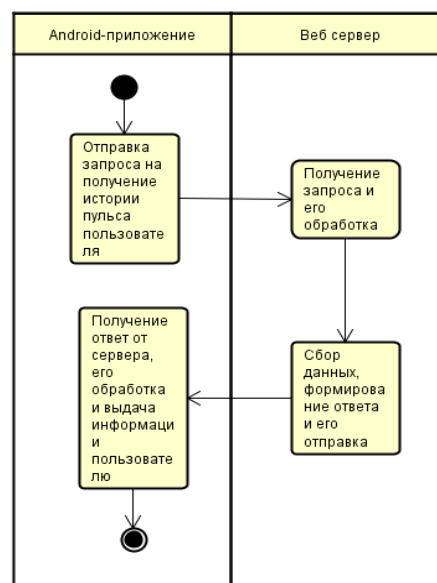


Рис. 6. Диаграмма получения истории пульса пользователя

На рисунке 7 показана диаграмма процесса заполнения подробных личных данных о пользователе.

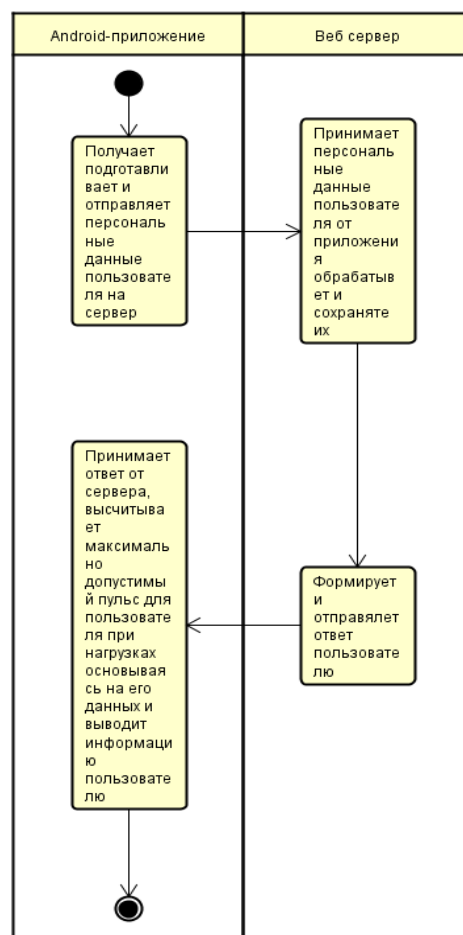


Рис. 7. Диаграмма процесса заполнения подробных личных данных о пользователе.

ПРОЕКТИРОВАНИЕ МАКЕТА ПРИЛОЖЕНИЯ

Для любого приложения главным является удобство и простота его использования, поэтому при создании макета приложения использовался стиль Material Design.

Material Design — это общий разработанный концепт дизайна всех приложений для Android версий 5+. Его главной чертой является четкость и выделение всех действий через анимацию [4]. Таким образом, пользователь всегда наглядно видит, что делает его приложение и не впадает в состояние «неведения».

Все важные элементы интерфейса выделяются с помощью теней и анимаций, реагирующих на действия пользователя [4].

При выборе этого дизайна следует придерживаться правилу «трех цветов», что означает, что в оформлении всего приложения будет использоваться всего 3 основных цвета и их оттенки. Обязательным условием Material Design является использование одного общего шрифта на все приложение [4].

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ БАЗЫ ДАННЫХ

При проектировании базы данных для разрабатываемой системы была выбрана реляционная модель базы данных. В базе данных с такой моделью используются связи между таблицами типа «один ко многим».

Например, с одной записью из таблицы с пользователями соотносится сразу несколько записей из таблицы с данными об истории пульса для каждого пользователя.

Для того, чтобы корректно сохранить данные, написанные в кодировке CP1251 («Кириллица»), использовалась кодировка базы данных UTF8_bin («Многоязычная бинарная»). Такой метод кодировки позволяет корректно сохранять любые данные в любой кодировке, но минусом этого метода является чрезмерное использование дискового пространства при хранении данных.

ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРИЛОЖЕНИЯ

Для проектирования структуры данного мобильного приложения использовалась схема разделения данных приложений «Model-View-Controller» (MVC).

Плюсом этой схемы проектирования является удобство разработки и четкое структурирование кода, что положительно сказывается на скорости разработки приложения и простоте дальнейшей поддержки кода и его доработки.

Все элементы, отвечающие за визуальное отображение интерфейса приложения, создаются в специальных файлах формата *.xml* [2].

Все файлы, отвечающие за логику работы приложения, описываются в специальных файлах формата *.java*. Таким образом код не «перемешивается» и остается легко читаемым [2].

Также, чтобы избежать повторения одного и того же кода несколько раз в разных классах, было решено вынести повторяющийся код в отдельный класс и использовать его в остальных классах через наследование.

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ВЕБ-ПАНЕЛИ

Всего было создано две таблицы:

1) Users:

```
id – bigint(auto increment unique)
name – string
surname – string
```

```
age – byte
weight – byte
height – byte
sex – boolean(0 – жен. 1 – муж.)
email – string(unique)
password – string
auth_token – varchar(40)
created_at – date
updated_at – date
deleted_at – date;
```

2) Pulse_history:

```
id – bigint(auto increment unique)
user_id – bigint
pulse – integer
status – integer
created_at – date
updated_at – date
deleted_at – date.
```

Схема работы сервера:

1. Пользователь регистрирует учетную запись в приложении, и его данные записываются в таблицу «users».

2. Приложение автоматически замеряет пульс и отправляет его на сервер для сохранения в таблице «pulse_history».

3. Веб-панель, с помощью которой система наблюдает в режиме реального времени, проверяет историю пульса пользователя и, в случае выявления аномалии, оповещает оператора и пользователя о возможном приступе с рекомендациями о том, как этот приступ можно предотвратить.

ПРОЕКТИРОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ

В данной системе используется библиотека TensorFlow.js. Данная библиотека была основана на DeepLearning.js. Для запуска и обучения нейронной сети на сервере используется программная платформа NodeJS [5].

Установка TensorFlow:

```
yarn add @tensorflow/tfjs
```

Подключение библиотеки в скрипт:

```
var tf = require('@tensorflow/tfjs');
```

Callback-функция для обучения нейронной сети:

```
async function doTraining(model) {
  const history = await model.fit(xs, ys, {
    epochs: 1000,
    callbacks: {
      onEpochEnd: async (epoch, logs) => {
        console.log(epoch, logs.loss);
      }
    }
  });
}
```

Объявление модели нейронной сети:

```
const model = tf.sequential();
model.add(tf.layers.dense({units: 1, inputShape: [5]}));
model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});
```

Подготовка данных для обучения нейронной сети:

```
let arr = [
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 80) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 120) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 81) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 121) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 82) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 122) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 83) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 123) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 84) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 124) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 80) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 120) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 81) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 121) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 82) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 122) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 83) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 123) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 84) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 124) * 100],
  [(1 / 186) * 100, (1 / 91) * 100, (1 / 24) * 100, 1 / 1, (1 / 150) * 100],
  [(1 / 171) * 100, (1 / 88) * 100, (1 / 32) * 100, 1 / 2, (1 / 150) * 100],
];
const xs = tf.tensor2d(arr);
const ys = tf.tensor2d(
  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 2, 2],
  [arr.length, 1]
);
```

Запуск обучения и прогнозирования:

```
doTraining(model).then(() => {
  console.log(model.predict(tf.tensor2d([(1 / 186) * 100,
  (1 / 91) * 100, (1 / 24) * 100, 1 / 2, (1 / 85) * 100], [1,5])).print());
});
```

В данном примере была описана модель однослойного перцептрона с пятью нейронами. В качестве функции потерь использовалась «среднеквадратическая ошибка». В качестве функции оптимизатора использовался «Стохастический градиентный спуск». Для получения корректных и постоянных данных понадобилось 1000 эпох. Также была проведена нормализация данных с помощью деления единицы на входные значения и последующего умножения на 100.

Обозначения входных параметров в данном примере:

- 1) 186, 171 — рост пользователя в сантиметрах;
- 2) 91, 88 — вес пользователя в килограммах;
- 3) 24, 32 — возраст пользователя в годах;
- 4) 1 означает, что пользователь мужчина;

- 5) 2 означает, что пользователь женщина;
- 6) 80, 81, 82, 83, 84, 120, 121, 122, 123, 124 — пульс пользователя в определенный момент времени.

Обозначение результирующих параметров для обучения нейронной сети:

- 1) 0 — самочувствие пользователя «отлично»;
- 2) 1 — самочувствие пользователя «внимание»;
- 3) 2 — самочувствие пользователя «плохо».

На выходе нейронной сети рассчитывается вероятность статуса самочувствия пользователя в диапазоне [0, 1].

По результирующему коэффициенту можно отделить статус пользователя так:

- 1) до 0,35 => «отлично»
- 2) 0,35–0,7 => «внимание»
- 3) от 0,7 => «плохо»

Диапазон 0,35–0,7 немного больше других диапазонов, потому что для данной системы важнее найти «большого» пользователя, даже если пара лишних пользователей попадут в «зону риска».

ЗАКЛЮЧЕНИЕ

В рамках настоящей работы разработана система мониторинга физического состояния работников железной дороги. Дальнейшее развитие системы предполагает разработку прогнозирования возможных сердечно-сосудистых заболеваний, основываясь на физиологических данных работника.

Возможно масштабирование системы путем разработки новых сервисов для более качественного анализа физического состояния человека.

ЛИТЕРАТУРА

1. Машнин Т. С. Web-сервисы Java. — СПб.: БХВ-Петербург, 2012. — 560 с. — (Профессиональное программирование: PRO).
2. Android. Разработка приложений = Android: Application Development / Р. Роджерс, Д. Ломбардо, З. Медникс, Б. Мейк. — М.: ЭКОМ Паблишерз, 2010. — 400 с.
3. Асмаков С. Интерфейс Bluetooth: разберемся с нюансами // Компьютер Пресс. 2013. № 3 (279). С. 34–36.
4. Васильева Е. В. Flat-Design и система интернационального стиля: графические принципы и визуальная форма / Е. В. Васильева, Ж. С. Гарифуллина (Аристов) // Знак: проблемное поле медиаобразования. 2018. № 3 (29). С. 43–49.
5. Джулли А. Библиотека Keras — инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow = Deep learning with Keras. Implement neural networks with Keras on Theano and TensorFlow / А. Джулли, С. Пал.; пер. с англ. А. А. Слинкин. — М.: ДМК-Пресс, 2018. — 294 с.

The Use of Neural Networks to Determine the Physical Condition of a Person

Master of Science R. A. Sergeev, Grand PhD V. P. Bubnov
Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
roma.svetl@gmail.com, bubnov1950@yandex.ru

Abstract. The developed mobile application is a system for monitoring the physical condition of railway workers. The idea of this system is that a user (an employee on Russian Railways) puts on a Mi band 2 fitness bracelet and connects it to an android smartphone via Bluetooth, and a specially developed application automatically measures the pulse and sends it to the server for data processing in order to obtain status results employee.

Keywords: database, neural network, android application, physical condition of a person, fitness bracelet, neural network prediction, android application development.

REFERENCES

1. Mashnin T. S. Java Web services [Web-servisy Java], St. Petersburg, BHV-Petersburg, 2012, 560 p.
2. Rogers R., Lombardo J., Mednieks Z., Meike B. Android: Application Development [Android. Razrabotka prilozheniy], Moscow, ECOM Publishers, 2010, 400 p.

3. Asmakov S. Bluetooth Interface: Let's Look at the Nuances [Interfeys Bluetooth: razberemysya s nyuansami], *Computer Press [Komp'yuter Press]*, 2013, No. 3 (279), Pp. 34–36.

4. Vasylieva E. V., Garifullina (Aristova) Zh. S. Flat-Design and the System of International Style: Graphic Principles and Visual Form [Flat-Design i sistema internatsional'nogo stilya: graficheskie printsipy i vizual'naya forma], *SIGN: Problematic Field of Media Education [Znak: problemnoe pole mediaobrazovaniya.]*, 2018, No. 3 (29), Pp. 43–49.

5. Gulli A., Pal S. Deep learning with Keras. Implement neural networks with Keras on Theano and TensorFlow [Biblioteka Keras — instrument glubokogo obucheniya. Realizatsiya neyronnykh setey s pomoshch'yu bibliotek Theano i TensorFlow], Moscow, DMK-Press Publishers, 2018, 294 p.

Анализ целевых возможностей наземного комплекса управления космическими аппаратами на основе сетей Петри и экспертного оценивания

к. воен. н. А. Н. Кудряшов

Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия,
kudrspb@mail.ru

Аннотация. Разработан алгоритм, включающий следующие действия. Проводится анализ структур наземного комплекса и построение графа его технологической структуры. Далее моделируется выполнение выбранных технологических операций. Построенная имитационная модель позволяет учитывать влияние возможных воздействий на элементы наземного комплекса и оценивать полноту выполнения операций. Для более детального анализа достигнутой степени выполнения операций предлагается использовать экспертные оценки, сформулированные вербально. Выполняется переход от лингвистической шкалы к количественной шкале с использованием нечетких чисел. Полученные данные о целевых возможностях комплекса в заданной ситуации сводятся в единую таблицу. Такие таблицы могут послужить основой для принятия обоснованных решений по применению наземных средств управления космическими аппаратами, а освоение данного алгоритма может быть полезным при подготовке соответствующих специалистов.

Ключевые слова: космический аппарат, наземный комплекс управления, технологическая операция, имитационная модель, показатели возможностей, экспертная оценка, нечеткое число.

ВВЕДЕНИЕ

Необходимость анализа целевых возможностей (ЦВ) наземного комплекса управления космическими аппаратами (НКУ КА) не вызывает сомнений. В реальной практике управления КА результаты такого анализа могут послужить основой для принятия обоснованных решений по перепланированию применения средств НКУ, изменению программ сеансов управления КА в конкретной сложившейся ситуации.

Кроме того, освоение разработанного алгоритма может быть полезным при обучении операторов НКУ. Целесообразность использования алгоритма при проведении занятий (тренировок) заключается в повышении эффективности обучения за счет приобретения навыков выполнения операций управления КА как в штатном режиме, так и при введении любых мыслимых и немыслимых нештатных ситуаций.

А с экономической стороны это выгодно потому, что в процессе обучения не требуется использования разнородной реальной техники и многочисленного персонала, достаточно одного персонального компьютера. К тому же все отказы, нештатные ситуации, введенные преподавателем (инструктором) или возникшие из-за ошибок, совершенных самим обучаемым, будут условными, виртуаль-

ными, не наносящими вреда реальной технике (КА и средствам НКУ).

В статье для наглядности сделан акцент на рассмотрение функционирования НКУ КА определенного класса, а именно КА наблюдения (мониторинга, дистанционного зондирования Земли). Тем не менее, представляемый алгоритм без принципиальных доработок может быть использован для анализа целевых возможностей НКУ КА других типов.

ОСНОВНАЯ ЧАСТЬ

Алгоритм (от имени арабского математика Аль-Хорезми) — конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающих порядок (последовательность) действий для решения некоторой задачи [1]. В результате проведенных исследований определилась необходимая последовательность (алгоритм) следующих шести действий по оцениванию целевых возможностей НКУ КА наблюдения.

АНАЛИЗ СТРУКТУР НКУ

НКУ КА представляет собой многоуровневую иерархическую организационно-техническую систему управления, отличающуюся сложностью структурного построения, включающего структуры: функциональную, технологическую, техническую, топологическую, организационную, структуру программно-математического и информационного обеспечения (ПМИО) [2].

Функциональная структура — совокупность конкретных целей, функций и задач НКУ, а также концептуальных отношений между ними, определяющих взаимосвязи функций и задач, необходимых для достижения целей, стоящих перед НКУ.

Технологическая структура НКУ — совокупность выполняемых в процессе управления технологических операций, направленных на решение задач, стоящих перед системой, и логико-временных связей между ними.

Техническая структура НКУ — комплекс технических средств и каналов связи между ними, необходимых для реализации соответствующих технологических операций.

Топологическая структура — совокупность территориально (пространственно) распределенных узлов (районов) дислокации или расположения отдельных элементов и подсистем НКУ и взаимосвязей между ними.

Организационная структура — совокупность смен и расчетов, выполняющих технологические операции на технических средствах НКУ, и находящихся в субординационных или координационных связях между собой.

Структура ПМИО НКУ — совокупность наборов информационных данных, знаний о заданной предметной области, моделей, методов, алгоритмов и программ для ЭВМ, необходимых для автоматизированного решения задач, определяемых технологической структурой.

Указанные структуры тесно взаимосвязаны. Анализ выполнения расчетами центров управления полетами (ЦУП) и отдельных командно-измерительных комплексов (ОКИК) технологических операций подготовки и проведения сеансов управления КА наблюдения с применением технических средств НКУ показал следующее.

Объединяющими (интегрирующими) являются технологическая и топологическая структуры. И действительно, функциональная структура детализируется в технологической, а элементы технической и организационной структур пространственно (топологически) совмещены для выполнения технологических операций. Элементы же структуры ПМИО вообще строго привязаны к элементам структуры технических, в первую очередь вычислительных, средств НКУ. Следовательно, представленная в виде графа, вершины которого интерпретируются как технологические операции, выполняемые соответствующими расчетами с использованием аппаратно-программных средств, технологическая структура НКУ в полной мере отражает процесс его функционирования.

Процесс функционирования НКУ, то есть процесс управления КА, включает в себя две составляющие:

– выдача управляющих воздействий на КА (выдача на КА и исполнение на его борту командно-программной информации (КПИ));

– прием, обработка, анализ информации о состоянии КА (информация обратного канала (оперативного контроля) (ИОК), телеметрическая информация (ТМИ), измерения текущих навигационных параметров (ИТНП)). Соответственно, разрабатываемая имитационная модель также будет состоять из двух частей.

Перейдем к рассмотрению первой части — моделированию процесса выдачи и реализации КПИ, а именно: разовых команд (РК), исполняемых непосредственно в зоне радиовидимости наземных средств, и рабочих программ (РП), представляющих из себя исходные числовые массивы для работы бортовых цифровых вычислительных машин (БЦВМ) вне зон радиовидимости.

Исходя из проведенного анализа работы расчетов, функционирования средств НКУ и бортовой аппаратуры, для формирования имитационной модели представим моделируемый процесс в графическом виде (рис. 1).

На рисунке цифрами обозначены следующие операции:

1. Ввод КПИ в комплекс средств автоматизации (КСА) ЦУП.
2. Передача КПИ на командно-измерительную систему (КИС)
3. Выдача РК на КА.

4. Передача РК из бортового приемника (БПРМ) в систему трансляции команд (СТК).

5. Выдача команды из СТК на исполнительные органы (ИО).

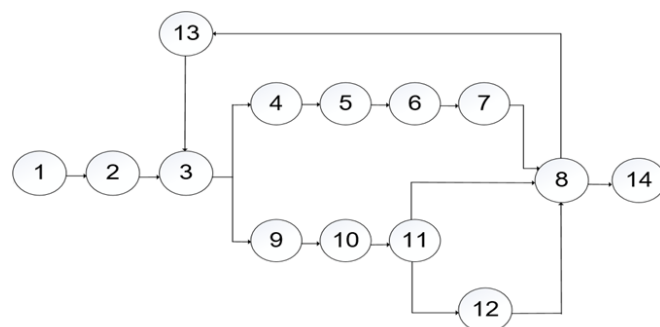


Рис. 1. Технологическая структура процесса выдачи на КА и исполнения на его борту КПИ

6. Передача от ИО в СТК сигнала об исполнении команды.

7. Выдача из СТК в передатчик (БПРД) квитанции на РК.

8. Выдача квитанций на РК, РП, ИОК из БПРД на КИС.

9. Выдача КИС рабочих программ на КА.

10. Передача массивов рабочих программ через БПРМ в бортовую цифровую вычислительную машину (БЦВМ).

11. Выдача из БЦВМ квитанций о получении РП.

12. Формирование и выдача из БЦВМ массива ИОК.

13. Повтор, отмена или изменение РК (РП).

14. Окончание сеанса.

Данная технологическая структура отображает процесс передачи сформированной КПИ на КА и исполнения этих команд на его борту в статическом виде. Для рассмотрения этого процесса в динамике и предлагается построение имитационной модели.

МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКИХ ОПЕРАЦИЙ НА ЭЛЕМЕНТАХ СТРУКТУРЫ НКУ С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ

Одним из видов имитационных моделей являются так называемые сети Петри (СП), причинно-следственные модели параллельных действий [3–5]. Именно такую модель предлагается использовать для моделирования функционирования НКУ. При построении этой модели ее переходы и позиции интерпретировались следующим образом. Переходы — это операции технологического цикла управления (ТЦУ), а наличие маркеров во входных позициях — условия для их выполнения [6].

Множество позиций P предлагается разбить на три подмножества:

- P_1 — логические позиции, отображающие технологическую последовательность выполнения операций;
- P_2 — «возмущаемые», или «управляющие» позиции, отсутствие маркеров в которых говорит о невозможности выполнения соответствующих операций из-за отказа технического средства, ошибки персонала, других воздействий или эти операции осознанно исключены из ТЦУ.
- P_3 — контрольные позиции, количество маркеров в которых отражает степень выполнения ТЦУ (позиции отображения ситуации).

Введение позиций нескольких видов, особенно применение термина «управляющая позиция» предполагает использование формализмов так называемых E-сетей [7, 8]. Однако, по мнению автора, использование сетей Петри в классическом виде для поставленных задач представляется более удобным.

Для этого сопоставим каждой технологической операции (см. рис. 1) переход СП, а связи, отражающие логическую последовательность операций, отобразим, связав соответствующие логические позиции. Результат этих действий показан на рисунке 2.

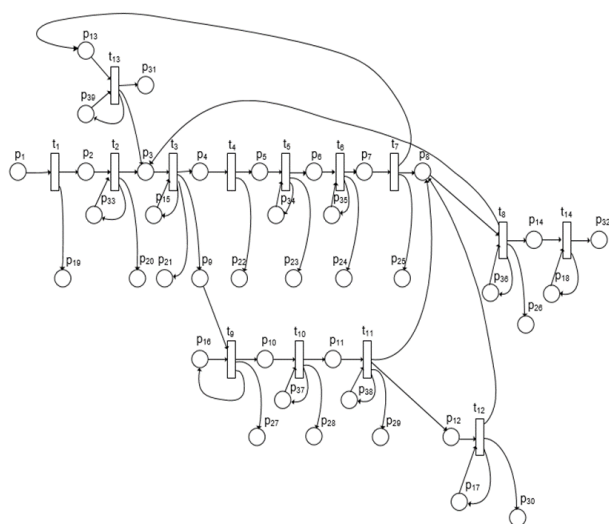


Рис. 2. Графическое представление 1-го фрагмента СП

Итак, интерпретация 14 переходов СП — это 14 технологических операций, представленных выше. Интерпретация наличия или отсутствия маркеров в позициях приведена ниже в таблице 1.

Таблица 1

Интерпретация маркировки СП в имитационной модели

Номер позиции	Интерпретация наличия (отсутствия) маркера
1	КПИ сформирована и введена в КСА ЦУП
2-14	Создание условия для выполнения следующей технологической операции
15	Возможна и разрешена выдача очередной разовой команды (РК)
16	Возможна и разрешена выдача рабочей программы (РП)
17	Возможна и разрешена выдача команды на съем информации оперативного контроля (ИОК)
18	Возможна и разрешена выдача команды на окончание сеанса
19	Отображение программы сеанса
20 и далее до 32	Накопление информации о выполнении технологических операций (контрольные позиции)
33	Выход из строя штатного канала ЦУП-ОКИК
34	Не проходит сигнал СТК-ИО
35	Неисправность конкретного ИО КА
36	Отсутствие квитанций на РК (РП)
37	Не проходит загрузка БЦВМ массивами РП
38	Отсутствие квитанций на обработку РП

При введении в «возмущаемые» позиции (№№ 15–18, 33–38) подмаркировки, отражающей конкретную ситуацию, появляется возможность по контрольной подмаркировке (позиции №№ 19–32) фиксировать ход процесса функционирования НКУ при выдаче и реализации КПИ, то есть определять текущую степень выполнения ТЦУ.

Аналогично построен второй фрагмент СП, моделирующий процессы сбора, обработки и анализа информации о состоянии КА (ТМИ, ТВИ, ИТНП) [6].

ОПРЕДЕЛЕНИЕ СТЕПЕНИ ВЫПОЛНЕНИЯ ТЦУ

«Выходом» построенной имитационной модели функционирования НКУ являются вектора контрольной подмаркировки

$$\mu_1 = (m(p_{20}), m(p_{21}), \dots, m(p_{32}))$$

для первого фрагмента модели, отображающего выдачу и реализацию КПИ, и

$$\mu_2 = (m(p_{31}), m(p_{32}), \dots, m(p_{43}))$$

для второго фрагмента, моделирующего сбор и обработку ТМИ.

Каждый элемент этих векторов — количество маркеров в соответствующей позиции после проведения имитационного эксперимента, состоящего в запуске сети Петри с введенными возмущающими воздействиями. Сопоставление количества маркеров в контрольной подмаркировке количеству потенциально выполнимых технологических операций проведем по таблице следующего вида.

Таблица 2

Результаты экспериментов на сети Петри

Результат выполнения операции	Требуемое по ТЦУ количество	Фактическое количество
РП передана на ОКИК	2	1
РП выдана на КА	2	1
ТМ записана на ОКИК	5	5
ТМ обработана в ТМЦ	5	4
ТМ выдана в ЦУП	5	4
ИОК записана на ОКИК	2	2
Проведены ИТНП	5	4
...

При необходимости операции ТЦУ могут быть представлены более детально. Операция ИТНП добавлена в таблицу из тех соображений, что, если КИС способна выдавать РК и получать на них квитанции, значит, она способна измерять, по крайней мере, радиальную скорость движения КА.

Целью функционирования НКУ является строгое выполнение заданного ТЦУ КА. А ТЦУ КА наблюдения сформирован для полного обеспечения решения КА его целевой задачи — сбора информации об объектах наблюдения. Следовательно, основным показателем целевых возможностей НКУ является степень выполнения им операций этого технологического цикла, соотношенная с полнотой обеспечения решения целевой задачи КА. Данные имитационных экспериментов, приведенные в таблице 2, дают представление о потенциально достижимой степени выполнения ТЦУ. В том случае, когда цифры в столбцах таблицы совпадают, оценивание возможностей НКУ мож-

но считать законченным — ТЦУ КА будет выполнен полностью. Как показал анализ, при той избыточности, которая заложена в совокупности развернутых для каждого типа КА наземных комплексов управления, требуемые объемы технологических операций могут быть обеспечены с многократным резервированием. Необходимость более детального анализа целевых возможностей НКУ возникает при существенном сокращении числа привлекаемых ОКИК, увеличении числа обслуживаемых КА, выходе из строя ключевых технических средств, каналов связи, «обвальных» отказах программного обеспечения, вынужденном переходе к сокращенным, а то и однопунктным технологиям управления и т. д.

Для такого более детального анализа целевых возможностей НКУ предлагается привлечение явных и неявных экспертных знаний.

ПРЕДСТАВЛЕНИЕ ЭКСПЕРТНЫХ МНЕНИЙ
О СТЕПЕНИ ВЫПОЛНЕНИЯ ТЦУ ТЕРМАМИ
ЛИНГВИСТИЧЕСКОЙ ПЕРЕМЕННОЙ

Выполнение технологической операции каждого типа преследует конкретную цель, достижение которой обеспечивает целевое применение КА наблюдения. Так, для операций выдачи КПИ основная цель – закладка РП, содержащей задание для бортового специального комплекса по объектам наблюдения. Для операций съема и обработки ТМИ это оценка способности бортовых систем КА выполнять и обеспечивать выполнение задач по его предназначению. Для операций ИТНП и последующей их обработки – определение параметров орбиты КА и его места на ней с той степенью точности, которая позволит использовать координатно-временной метод управления целевой аппаратурой. Для оценки степени достижения указанных целей по каждому типу операции предлагается привлечь экспертов, причем специализирующихся в соответствующих предметных областях.

При использовании экспертных знаний для оценивания состояния сложных объектов существует довольно жесткое ограничение: «Эксперт может при сравнении двух объектов, сказать, какой из двух лучше (метод парных сравнений), дать им оценки типа «хороший», «приемлемый», «плохой», упорядочить несколько объектов по привлекательности, но обычно не может ответить, во сколько раз или на сколько процентов один объект лучше другого». Однако широко распространен подход, когда ответы экспертов стараются рассматривать как числа, производят произвольную обработку их мнений, приписывая этим мнениям численные значения – баллы, коэффициенты, которые потом обрабатывают как результаты обычных физико-технических измерений. Выводы, полученные в результате обработки такой «оцифровки», могут не иметь отношения к реальности [9].

Возможность избежать подобных ошибок появляется в случае применения процедуры «арифметизации» вербальной экспертной информации. Это технология извлечения экспертных знаний, позволяющая общаться с экспертом на привычном для него профессиональном языке, побуждает его мыслить системно, обладает возможностью формализовать полученную информацию.

Процедура начинается с извлечения и представления экспертных знаний значениями (термами) лингвистиче-

ской переменной. Эти значения получают в ходе опроса экспертов. Они представляют собой ряд вербальных суждений от «невыполнимо» до «полное выполнение» для каждой технологической операции (цикла операций) управления КА.

ПРИМЕНЕНИЕ ПРОЦЕДУРЫ «АРИФМЕТИЗАЦИИ» ВЕРБАЛЬНОЙ
ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Переход от лингвистической к количественной шкале (процентам) осуществляется с использованием специальной системы координат (рис. 3), на оси абсцисс которой задаются лингвистические значения переменной γ (верхняя шкала от «Невыполнимо» до «Полное выполнение») и количественные значения показателя целевых возможностей НКУ x (шкала от 0 до 100). По оси ординат расположена функция принадлежности $\mu(x) = [0,1]$ как мера степени уверенности («правильности») перевода из одной шкалы в другую. На рисунке приведены наименования термов (ранжированный ряд вербальных значений лингвистической переменной γ). Им поставлены в соответствие числовые значения мод этих термов, а также коэффициенты нечеткости слева α и справа β для всех термов (по названию мод).

Так, например, значению x_i на оси ординат соответствуют два значения функции принадлежности: α и b . Поскольку $b > a$, то явно предпочтительнее отнести значение x_i к моде «Ниже среднего». При попадании x_i в зону, где значения $\mu(x)$ около 0,5, в указанной ситуации нельзя отдать предпочтение какой-либо из мод данного интервала, поэтому значение лингвистической переменной примет вид: между «Ниже среднего» и «Среднее».

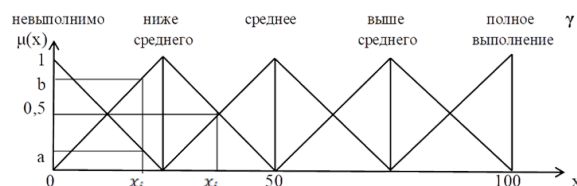


Рис. 3. Соотношение лингвистической γ и количественной x переменных

Для использования эксперта как интеллектуальной информационно-измерительной системы, по положениям теории планирования эксперимента [10], ему должны быть предоставлены значения признаков факторного пространства, описывающих рассматриваемое состояние НКУ. В качестве таковых используем данные из приведенной выше таблицы 2.

После определения экспертом по этой таблице значений лингвистической переменной γ и выполнения преобразований перевода этих значений в количественную шкалу (рис. 1), значения частных показателей целевых возможностей НКУ по выполнению циклов технологических операций могут быть представлены в виде нечетких унимодальных (L, R)-чисел следующего вида:

$$A_i = (a_i, \alpha_i, \beta_i),$$

где: a_i — значение (мода) нечеткого числа, α_i и β_i — левый и правый коэффициенты нечеткости (величины максимальных отклонений значения нечеткого числа), $i = \{1, 2, 3\}$ — индекс, определяющий тип цикла опера-

ций (КПИ, ИТНП, ТМИ соответственно). Функции принадлежности количественной переменной x_i (процент выполнения i -го цикла операций) имеют вид:

$$\mu(x_i) = L(x_i) = 1 - \frac{(a_i - x_i)}{a_i}, \text{ если } a_i > x_i,$$

$$\mu(x_i) = R(x_i) = 1 - \frac{(x_i - a_i)}{\beta_i}, \text{ если } a_i < x_i.$$

Далее казалось бы целесообразным построение интегрального показателя целевых возможностей НКУ. Тем более что такое построение возможно — с помощью мультипликативной или аддитивной, а то и полиномиальной [9, 10] свертки частных показателей с использованием специальных правил арифметических операций над нечеткими числами (L, R)-типа. Однако в таком интегральном показателе теряется физический смысл, наглядность отображения полноты выполнения ТЦУ. Более приемлемым представляется применение лексикографического упорядочения показателей: КПИ → ИТНП → ТМИ (исходя из значимости целей выполнения данных циклов опе-

раций (см. выше)), и сведение всех данных в общую таблицу, так называемую «опросную матрицу».

ЗАПОЛНЕНИЕ ОПРОСНОЙ МАТРИЦЫ ОЦЕНКИ ЦЕЛЕВЫХ ВОЗМОЖНОСТЕЙ НКУ ДЛЯ ЗАДАННОЙ СИТУАЦИИ

Согласно технологиям теории планирования эксперимента [3, 10] именно такие матрицы и должны заполняться экспертами. В таблице 3 приведен фрагмент опросной матрицы с ответами экспертов о значениях (модах) частного показателя в лингвистическом и оцифрованном видах. В нем указаны интервалы количественной шкалы, соответствующие значениям лингвистической переменной γ . Их границы определяются значениями предельных отклонений от мод значений i -го частного показателя (коэффициенты нечеткости α_i и β_i). В таблице 3 также включены степени уверенности в правильности оцифровки $\mu(x_i)$ и характеристики степени выполнения ТЦУ по типам операций (расшифровки значений γ).

Таблица 3

Оценка целевых возможностей НКУ для заданной ситуации (опросная матрица)

Цикл (i)	Достижение цели цикла (мода переменной γ)	Интервал (терм) шкалы x (%)	Мода терма шкалы x (%)	Значение показателя x_i (%)	Степень уверенности эксперта $\mu(x_i)$	Характеристика степени выполнения цикла операций обеспечения целевого применения КА (расшифровка значений переменной γ)
КПИ (1)	«среднее»	25–75	50	50	1,00	Задание по объектам наблюдения может быть выполнено, однако его коррекция в текущих полетных сутках невозможна.
ИТНП (2)	«выше среднего»	50–100	75	80	0,75	ИТНП достаточно для применения координатно-временного метода управления целевой аппаратурой КА.
ТМИ (3)	«выше среднего»	50–100	75	80	0,75	ТМИ достаточно для оценки способности бортовых систем обеспечить решение целевых задач КА.

Значений γ всего пять, соответственно и пять интервалов количественной шкалы x . Типов циклов три. Следовательно, все многообразие ситуаций, все множество признаков факторного пространства состояний НКУ с точки зрения выполнения его целевых задач может быть сведено к трем (из возможных пятнадцати) строкам опросной матрицы. Представляется целесообразным заблаговременное формулирование вербальных характеристик значений γ для каждого интервала и каждого типа циклов технологических операций. Создание таких предварительных таблиц-заготовок опросных матриц позволит существенно повысить оперативность оценивания целевых возможностей НКУ в конкретной ситуации.

ЗАКЛЮЧЕНИЕ

Предложенный алгоритм без принципиальных доработок может быть использован для анализа целевых возможностей НКУ КА различных типов. Использование получаемых в ходе экспертного оценивания таблиц данных о целевых возможностях НКУ КА в условиях влияния возмущающих воздействий, в конкретной ситуации, несомненно, будет способствовать повышению адекватности и обоснованности принимаемых управленческих решений.

ЛИТЕРАТУРА

1. Кобельков Г. М. Алгоритм вычислительный // Большая российская энциклопедия — электронная версия. URL: <http://bigenc.ru/mathematics/text/1810319> (дата обращения 03.07.2020).
2. Системный анализ и организация автоматизированного управления космическими аппаратами: Курс лекций / А. Н. Павлов, А. Н. Кудряшов, Ю. С. Мануйлов, С. А. Осипенко. — СПб.: ВКА им. А. Ф. Можайского, 2013. — 237 с.
3. Военная системотехника и системный анализ: Учебник / А. Н. Павлов, Б. В. Соколов, Б. В. Москвин, Д. Н. Верзилин; под общ. ред. А. Н. Павлова. — СПб.: ВКА им. А. Ф. Можайского, 2010. — 251 с.
4. Mesoscopic supply chain simulation / T. Hennies, T. Reggelin, J. Tolujew, P.-A. Piccut // Journal of Computational Science. 2014. Vol. 5, Is. 3. Pp. 463–470. DOI: 10.1016/j.jocs.2013.08.004.
5. Agent-Based Models of Geographical Systems / A. J. Heppenstall, et al. (eds). — Dordrecht: Springer, 2012. — 768 p.
6. Кудряшов А. Н. Имитационная модель проведения средствами отдельного командно-измерительного ком-

плекса сеансов управления космическими аппаратами / А. Н. Кудряшов, И. А. Кобылин // Сб. тезисов III Всероссийской научно-технической конференции «Теоретические и прикладные проблемы развития и совершенствования автоматизированных систем управления военного назначения» (Санкт-Петербург, 22 ноября 2017 г.). — СПб.: ВКА им. А. Ф. Можайского, 2017. С. 33–34.

7. Решмин Б. И. Имитационное моделирование и системы управления: Учебно-практическое пособие. — Вологда: Инфра-Инженерия, 2016. — 74 с.

8. Акопов А. С. Имитационное моделирование: Учебник и практикум для академического бакалавриата. — М.: Издательство Юрайт, 2017. — 389 с. — (Бакалавр. Академический курс).

9. Спесивцев А. В. Мягкие измерения и мягкие вычисления при моделировании состояния сложных объектов на базе экспертных знаний // Управление в условиях неопределенности: Монография / Под общ. ред. проф. С. В. Прокочниной. — СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2014. — С.217–263.

10. Павлов А. Н. Методика построения псевдоуниверсальных сверток лингвистических показателей на основе теории планирования эксперимента // Сборник докладов XII Международной конференции по мягким вычислениям и измерениям (SCM'2008), (Санкт-Петербург, 23–25 июня 2008 г.). — СПб: Изд-во СПбГЭТУ «ЛЭТИ», 2008. — Т. 1. — С. 169–172.

Analysis of the Target Capabilities of the Ground-Based Spacecraft Control System Based on Petri Nets and Expert Evaluation

PhD in Military Science A. N. Kudryashov
A. F. Mozhaisky Military Space Academy
Saint Petersburg, Russia
kudrspb@mail.ru

Abstract. An algorithm has been developed that includes the following actions. The analysis of the structures of the ground complex and the construction of a graph of its technological structure are carried out. Next, the execution of selected technological operations is simulated. The constructed simulation model allows us to take into account the influence of possible impacts on the elements of the ground complex and evaluate the completeness of operations. Expert assessments formulated verbally are proposed for a more detailed analysis of the degree of operation. Next, the transition from linguistic to quantitative scale using fuzzy numbers. The obtained data on the target capabilities of the complex in a given situation are summarized in a single table. Such tables can serve as the basis for making informed decisions on the use of ground-based spacecraft controls, and mastering this algorithm can be useful in preparing the appropriate specialists.

Keywords: spacecraft, ground control complex, technological operation, simulation model, capability indicators, expert judgment, fuzzy number

REFERENCES

1. Kobelkov G. M. Computational Algorithm [Algoritm vychislitel'nyy], *Great Russian Encyclopedia — Electronic Version [Bol'shaya rossiyskaya entsiklopediya — elektronnyaya versiya]*. Available at: <http://bigenc.ru/mathematics/text/1810319> (accessed 03 July 2020).
2. Pavlov A. N., Kudryashov A. N., Manuylov Yu. S., Osipenko S. A. System analysis and organization of automated control of spacecraft: A course of lectures [Sistemnyy analiz i organizatsiya avtomatizirovannogo upravleniya kosmicheskimi apparatami: Kurs lektsiy], Saint Petersburg, A. F. Mozhaisky Military Space Academy, 2013, 237 p.
3. Pavlov A. N., Sokolov B. V., Moskvina B. V., Verzilin D. N. Military system engineering and system analysis: Textbook [Voennaya sistemotekhnika i sistemnyy analiz: Uchebnik], Saint Petersburg, A. F. Mozhaisky Military Space Academy, 2010, 251 p.
4. Hennies T., Reggelin T., Tolujew J., Piccut P.-A. Mesoscopic supply chain simulation, *Journal of Computational Science*, 2014, Vol. 5, Is. 3, Pp. 463–470. DOI: 10.1016/j.jocs.2013.08.004.
5. Heppenstall A. J., et al. (eds) Agent-Based Models of Geographical Systems. Dordrecht, Springer, 2012, 768 p.
6. Kudryashov A. N., Kobylin I. A. Simulation Model of Conducting Space Vehicle Control Sessions by Means of a Separate Command and Measurement Complex [Imitatsionnaya model' provedeniya sredstvami otdel'nogo komandno-izmeritel'nogo kompleksa seansov upravleniya kosmicheskimi apparatami], *Collection of theses III All-Russian Scientific and Technical Conference «Theoretical and applied problems of development and improvement of automated control systems for military purposes» [Sbornik tezisov III Vserossiyskoy nauchno-tekhnicheskoy konferentsii «Teoreticheskie i prikladnye problemy razvitiya i sovershenstvovaniya avtomatizirovannykh sistem upravleniya voennogo naznacheniya»]*, Saint Petersburg, November 22, 2017, Saint Petersburg, A. F. Mozhaisky Military Space Academy, 2017, Pp. 33–34.
7. Reshmin B. I. Simulation modeling and control systems: Educational and practical guide [Imitatsionnoe modelirovanie i sistemy upravleniya: Uchebno-prakticheskoe posobie], Vologda, Infraengineering, 2016, 74 p.
8. Akopov A. S. Simulation modeling [Imitatsionnoe modelirovanie], Moscow, URAIT Publishing House, 2017, 389 p.
9. Spesivtsev A. V. Soft Measurements and Soft Calculations for Modeling the State of Complex Objects Based on Expert Knowledge [Myagkie izmereniya i myagkie vychisleniya pri modelirovanii sostoyaniya slozhnykh ob'ektov na baze ekspertnykh znaniy], *In: Management in conditions of uncertainty: Monography [Upravlenie v usloviyakh neopredelennosti: Monografiya]*, Saint Petersburg, St. Petersburg Electrotechnical University «LETI», 2014, P. 217–263.
10. Pavlov A. N. Method of Constructing Pseudo-Universal Convolutions of Linguistic Indicators Based on the Theory of Experiment Planning [Metodika postroeniya psevdouniversal'nykh svertok lingvisticheskikh pokazateley na osnove teorii planirovaniya eksperimenta], *Proceedings of XII International Conference on Soft Computing and Measurement [Sbornik dokladov XII Mezhdunarodnoy konferentsii po myagkim vychisleniyam i izmereniyam] (SCM'2008)*, Saint Petersburg, June 23–25, 2008, Saint Petersburg, St. Petersburg Electrotechnical University «LETI», 2008, Vol. 1, Pp. 169–172.

Исследование современных анализаторов текстовой информации

магистр А. А. Брызгалов

Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
threnhawk@gmail.com

Аннотация. Представлен аналитический обзор двух известных программных систем, решающих задачи обработки естественного языка: Amazon Comprehend — сервис обработки естественного языка и IBM-Watson — суперкомпьютер фирмы IBM, оснащенный системой искусственного интеллекта. Проведен их анализ, выявлены особенности и недостатки рассмотренных сервисов. Используя информацию по истории ПГУПС, проведено тестирование сервиса Amazon Comprehend, выявлены недокументированные ограничения при обработке русского текста.

Ключевые слова: естественный язык, искусственный интеллект, NLP, машинное обучение, Amazon Comprehend, IBM-Watson.

ВВЕДЕНИЕ

Искусственный интеллект (ИИ) — это наделение компьютеров человеческими способностями, из которых владение языком едва ли не важнейшая.

Обработка естественного языка (далее — NLP, англ. *Natural Language Processing*) — область, находящаяся на пересечении информатики, искусственного интеллекта и математической лингвистики [1].

NLP изучает проблемы компьютерного анализа и синтеза естественных языков в целях создания более удобной формы взаимодействия компьютера и человека.

Задолго до того, как обработка естественного языка стала модной темой в области искусственного интеллекта, люди придумывали правила и машины для манипулирования языком, начав практику комбинирования букв. Среди философов давно шли споры о природе человека и процессе познания мира, нейрофизиологи и психологи разработали ряд теорий относительно работы человеческого мозга и мышления, математики задавались вопросами оптимальных расчетов, наконец зародился фундамент математической теории вычислений — теории алгоритмов — были созданы первые компьютеры.

Истоки обработки естественного языка можно отследить в начале компьютерной эры, когда Алан Тьюринг в 1950 году описал умную машину, способную легко взаимодействовать с человеком посредством текста на экране.

В 1954 году Джорджтаунским университетом совместно с IBM был проведен эксперимент машинного перевода более чем шестидесяти русских фраз на английский язык. Эксперимент нельзя назвать полнофункциональной системой, так как он был задуман в первую очередь для того, чтобы, показывая возможности машинного перевода, привлечь правительственный и общественный интерес к этой научной теме и увеличить финансирование [2].

К середине 1960-х годов ряд исследователей, воодушевленных ростом мощности и доступности компьютеров общего назначения, вдохновленных мечтой об искусственном интеллекте человеческого уровня, разрабатывали системы, направленные на подлинное понимание языка и диалог с машиной.

Важным этапом в развитии понимания естественного языка является созданная в 1966 году американским специалистом по информатике Джозефом Вейценбаумом программа ELIZA. Как пояснял Вейценбаум, его программа «делала возможной беседу определенного рода между человеком и компьютером на естественном языке». Программа ELIZA — первый в мире чат-бот и виртуальный собеседник, превратившая строгое человеко-машинное взаимодействие в иллюзию человеческого общения.

Еще одной невероятно успешной демонстрацией понимания естественного языка и искусственного интеллекта явилась программа Терри Винограда SHRDLU, разработанная в 1968–1970 годах. Программа могла понимать интерактивный диалог на английском языке, чтобы отвечать на вопросы и выполнять команды, имеющие отношения к виртуальному миру.

Но какими бы на вид интеллектуальными ни казались программы ELIZA и SHRDLU — это были очень простые программы с большим количеством ограничений. Однако они выполнили свою миссию в истории развития искусственного интеллекта и вдохновили разработчиков на дальнейшие исследования в области понимания естественного языка, что в конечном итоге оказалось гораздо сложнее, чем считалось ранее [3].

В период 1980–1990-х годов темпы роста направления искусственного интеллекта увеличились. Это было связано с внедрением машинного обучения алгоритмов обработки языка, с постоянным увеличением вычислительной мощности и постепенным ослаблением доминирования хомских теорий лингвистики (Теория Хомского о языке). Вычислительная грамматика, инструменты и практические ресурсы, связанные с математической лингвистикой, стали доступны вместе с синтаксическими анализаторами. Получили направление исследования по основным и футуристическим темам, таким как устранение неоднозначности слов и статистически окрашенный естественный язык, работа над лексиконом. К этому стремлению развития NLP присоединились и другие важные темы, такие как статистическая обработка языка, извлечение информации и автоматическое обобщение [4].

В настоящее время мы находимся на пике интереса к обработке естественного языка. Благодаря прорывам в машинном обучении в последнее десятилетие мы наблюдаем

серьезное улучшение в деле распознавания речи и машинного перевода. Генераторы языка уже достаточно хороши для того, чтобы писать связные новостные статьи, а виртуальные помощники становятся частью нашей повседневной жизни. Качество современных систем машинного перевода уже вплотную приближается к человеческому [5].

В глубоком обучении языку все большую роль, в свою очередь, играет обучение «без учителя», с использованием громадных массивов дешевых размеченных текстов. Обилие данных позволяет обучать достаточно сложные модели языка с миллиардами настроечных параметров и недостижимым прежде качеством решения практических задач. Понятно, что создание подобных моделей вручную даже большими коллективами лингвистов практически нереально.

Обработка естественного языка — собирательный термин, относящийся к автоматической вычислительной обработке человеческих языков. Обработка включает в себя как алгоритмы, которые принимают созданный человеком текст в качестве входных данных, так и алгоритмы, которые производят естественно выглядящий текст в качестве выходных данных.

NLP, также известная как компьютерная лингвистика, представляет собой комбинацию искусственного интеллекта и лингвистики, которая позволяет нам разговаривать с машинами так, как если бы они были людьми.

ЗАДАЧИ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

Проблема обработки естественного языка была приоритетной в развитии искусственного интеллекта в течение десятилетий. В 1980–90-е годы были предприняты значительные усилия по созданию чат-бота для общения с человеком и выдачи заранее написанного ответа на заданный вопрос. Такой тип системы обычно называют конечным автоматом, или детерминированным конечным автоматом.

Основным недостатком такой системы была реализация на основе правил и условного оператора if-else, которая может быть сложной структурой для декодирования и обновления [6].

К основным направлениям обработки естественного языка относят:

- проверку орфографии;
- автозаполнение предложений;
- обобщение документов;
- ответ на вопрос;
- распознавание именованных сущностей;
- машинный перевод.

По данным исследовательской компании «Gather», клиенты в области обработки естественного языка сталкиваются со следующими проблемами [7]:

- 80% времени исследователей данных тратится на утомительные задачи, которые можно было бы автоматизировать;
- 63% компаний рассматривают наличие технических навыков как проблему для внедрения;
- 70% предприятий повысят производительность труда сотрудников с помощью ИИ к 2021 году;
- 60% компаний рассматривают нормативные ограничения как препятствие для внедрения ИИ.

Недавно появилось несколько очень интересных приложений, в которых обработка естественного языка может внести значительный вклад в изучение NLP и в распространение искусственного интеллекта на предприятиях. И одно из них — это крупнейший сервис на рынках платформ электронной коммерции и публично-областных вычислений Amazon Web Services (AWS) Amazon Comprehend.

AMAZON COMPREHEND

Amazon Comprehend — это сервис обработки естественного языка, запущенный в 2017 году, в котором для поиска закономерностей и взаимосвязей в тексте применяются технологии машинного обучения. Amazon Comprehend использует машинное обучение для извлечения аналитических сведений и взаимосвязей из неструктурированных данных. Сервис определяет язык текста, извлекает ключевые фразы, распознает людей, места, бренды или события, определяет степень позитивности или негативности текста, анализирует текст с помощью токенизации и частей речи и в результате автоматически группирует набор текстовых файлов по темам [8]. На рисунке 1 представлен пример работы Comprehend.

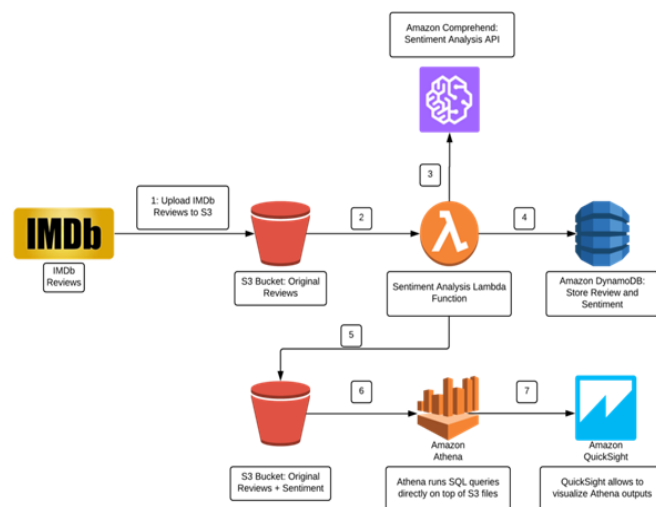


Рис. 1. Пример потока отзывов imdb, использующих Amazon Comprehend

Программное обеспечение Amazon Comprehend состоит из нескольких приложений:

1. Извлечение ключевых фраз (рис. 2). API извлечения ключевых фраз возвращает ключевые фразы или тезисы разговора и оценку уверенности в том, что это ключевая фраза.

2. Анализ настроений (рис. 3). API анализа настроений возвращает общее настроение текста (положительное, отрицательное, нейтральное или смешанное).

3. Синтаксический анализ. API Amazon Comprehend Syntax позволяет клиентам анализировать текст с помощью токенизации и частей речи (PoS), а также определять границы слов и метки, такие, как существительные и прилагательные в тексте.

Key phrases
the XVIII century
the end
the Department
water communications
the state road construction Expedition
the survey, design, construction, reconstruction and operation
Russian Railways
1801-1809

Рис. 2. Ключевые фразы

Sentiment

Neutral 0.99 confidence	Positive 0.00 confidence
Negative 0.00 confidence	Mixed 0.00 confidence

Рис. 3. Настроение текста

4. Распознавание сущностей (рис. 4). API распознавания сущностей возвращает именованные сущности («Люди», «Места», «Местоположения» и т. д.), которые автоматически классифицируются на основе предоставленного текста.

Analyzed text

From the XVIII century to the end, the Department of water communications and the state road construction Expedition were engaged in survey, design, construction, reconstruction and operation of Russian Railways. In 1801-1809, both these departments were headed by Nikolai Petrovich Rumyantsev (1754-1826), a prominent statesman, diplomat, member of the State Council, and Senator. In 1809, submitted to the Emperor Alexander I "proposals for reliable measures for the establishment of convenient communications on land throughout Russia". This document not only proposed a new, unified, and improved management structure for all Russian communication routes, but also suggested the need to establish a special higher technical educational institution in the state to be capable of leading the construction of roads and bridges, river and sea ports, and civil and transport structures. On the basis of the in accordance with the most highly approved Manifesto on November 20, 1809, an independent Agency was organized—the Manag water and land communications. The same Manifesto established the Corps of railway engineers and the Institute of the Corps of railway engineers (now Saint Petersburg state University of railway engineering). The building was created

▼ Results

Entity	Category	Confidence
XVIII century	Date	0.99+
Department of water communications	Organization	0.93
Russian Railways	Organization	0.94
both	Quantity	0.86
Nikolai Petrovich Rumyantsev	Person	0.99+
1809	Date	0.99+
Emperor Alexander I	Person	0.84

Рис. 4. Сущности

5. Пользовательские объекты. Позволяют настроить Amazon Comprehend для определения терминов, характерных для конкретного домена. Используя AutoML, Comprehend изучит небольшой частный набор примеров

(например, список стратегий и текст, в котором они используются), а затем обучит пользовательскую модель распознавать эти термины в любом другом блоке.

6. Обнаружение языка. API обнаружения языка автоматически обнаруживает текст, написанный на более чем 100 языках, и возвращает доминирующий язык с показателем уверенности, подтверждающим, что этот язык является доминирующим.

7. Пользовательская классификация. API пользовательской классификации позволяет легко создавать пользовательские модели классификации текста с помощью специфичных для компании меток.

8. Тематическое моделирование. Определяет соответствующие термины или темы из коллекции документов, хранящихся в Amazon. Этот API определит наиболее распространенные темы в коллекции и организует их в группы, а затем сопоставит принадлежность документов темам.

9. Поддержка нескольких языков. Amazon Comprehend может выполнять анализ текстов на английском, французском, немецком, итальянском, португальском и испанском языках. Это позволяет создавать приложения, которые могут обнаруживать текст на нескольких языках, конвертировать его в английский, французский, немецкий, итальянский, португальский и испанский языки с помощью Amazon Translate, а затем использовать Amazon Comprehend для выполнения анализа текста.

Используя информацию по истории ПГУПС, проведено тестирование этого сервиса. Документация сервиса утверждает, что программа поддерживает анализ русского текста, но, когда на вход был дан текст на русском языке, сервис не смог распознать его, и, следовательно, анализ ключевых фраз и настроений, распознавание сущностей и т. д. не могли быть использованы. Таким образом, для того, чтобы входные данные были прочитаны сервисом, возникла необходимость перевести текст на английский язык.

Основываясь на очень скудной информации [9], полученной в ходе прохождения уроков на сайте Amazon, попытаемся описать, как работают некоторые функции.

Информация о настроении описывает общее настроение текста, а также оценки настроений каждого ярлыка: положительный, отрицательный, нейтральный и смешанный. Все эти оценки настроений возвращаются из модели глубокого обучения, созданной на программной платформе MXNet, и изображаются в виде числа от 0 до 1, где 1 — полная достоверность метки настроения.

Допустим, мы имеем текстовое сообщение (описание фильма, рецензия, комментарий):

This has to be one of the most boring unoriginal films I've ever seen.

Или же

Knives Out is the kind of «whodunnit» film that will never fail to entertain.

В первом примере система должна выдать отрицательный результат, так как комментарий негативный, а во втором, соответственно, положительный. Подобного рода задачи в машинном обучении называются классификацией, а метод — «обучение с учителем». То есть сначала алгоритм

(1) «тренируется» на обучающей выборке, сохраняя необходимые коэффициенты и другие данные модели, затем, при поступлении новых данных, с определенной вероятностью классифицирует их.

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad (1)$$

где β_0 и β_1 — коэффициенты, полученные на основе обучения на тестовых данных. Формула (1) в конечном счете возвращает значение от 0 до 1, то есть чем ближе к 0, тем больше вероятность, что текст несет негативную информацию.

Чтобы достаточно точно анализировать настроение, Comprehend производит семантический анализ. Например, мы с легкостью можем заменить слово «черный» словом «темный», так как их смысл очень схож. Такие слова можно назвать семантически схожими. В эту группу слов входят синонимы, гипонимы, гиперонимы и т. д. Amazon заменяет каждое слово в списке номером его семантической группы. В итоге мы получим нечто вроде «мешка слов», но с более глубоким смыслом.

Суть модели заключается в следующем: на вход дается большой объем текста, на выходе мы получаем взвешенный вектор фиксированной длины для каждого слова (длина вектора задается вручную). Фиксированная длина используется в dataset. Dataset для машинного обучения — это обработанная и структурированная информация в табличном виде. Скорее всего, за меру близости слов Amazon берет косинусное расстояние. Но так как математические модели закрыты, мы можем только догадываться, на каких алгоритмах основана данная функция.

КЛЮЧЕВЫЕ СЛОВА

Оценку информативности фраз можно вычислить по таким характеристикам, как частота слов и расположение слов в документе. Частота слов содержит количество вхождений слова или фразы в заданный документ. Чем больше вхождений исследуемого слова в документ, тем выше его информативность. Но необходимо отфильтровывать часто встречающиеся слова (стоп-слова), которые не содержат какой-либо информации о документе. Расположение слов в документе также учитывается; наиболее информативные фразы встречаются, как правило, в начале документа, в аннотации и в заголовке. Самой распространенной мерой для расчета информативности терминов в документе является TF-IDF (2). Вес термина пропорционален количеству употреблений данного термина в документе и обратно пропорционален частоте употребления в других документах коллекции. Особенностью этой меры является то, что при изменении количества документов в коллекции необходимо пересчитывать частоты всех терминов.

$$TF - IDF = TF * IDF. \quad (2)$$

TF (Term Frequency) — частота термина в анализируемом документе, отношение числа вхождения термина к общему количеству терминов в документе (3).

$$TF = \frac{n_i}{\sigma_k n_k}, \quad (3)$$

IDF (Inverse Document Frequency) — инвертированная частота документа — обратная частота, с которой термин встречается в других документах коллекции (4).

$$IDF = \log \frac{N}{df}, \quad \dots\dots(4)$$

где N — общее количество документов в коллекции (корпусе); df — количество документов, содержащих термин. Выбор основания логарифма не имеет значения, так как не влияет на соотношение весов терминов.

Не менее важной частью задачи является оценка эффективности найденного решения. Эффективность оценивается релевантностью автоматически найденных 8 ключевых фраз в документе по отношению к заведомо известным ключевым фразам. Точность (Precision) — отношение числа релевантных ключевых фраз, найденных автоматически, к общему числу найденных ключевых фраз в документе (5).

$$P = \frac{|Trel \cap Tretr|}{|Tretr|}, \quad (5)$$

где P — точность; Trel — множество релевантных терминов; Tretr — множество найденных терминов.

Полнота (Recall) — это отношение числа релевантных ключевых фраз, найденных автоматически, к общему числу релевантных ключевых фраз в документе (6).

$$R = \frac{|Trel \cap Trelf|}{|Trelf|}, \quad (6)$$

где R — полнота; Trelf — множество релевантных терминов, содержащихся в документе; Trel — множество найденных релевантных терминов.

F-мера (F-score, F-measure) — объединение точности и полноты в одной усредненной величине — определяется как взвешенное гармоническое среднее точности и полноты (7).

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}}, \quad \alpha \in [0,1], \quad (7)$$

где F — мера; P — точность; R — полнота.

Amazon Comprehend — полностью управляемый сервис, который использует машинное обучение для извлечения аналитических сведений и неструктурированных взаимосвязей, однако опыт в сфере машинного обучения не требуется. Amazon использует [10, с. 184–186] объектно-ориентированное хранилище, предоставляя услуги через вызов веб-сервисов.

Мощная функциональность, предлагаемая Amazon Comprehend, заключается в том, что можно создать пользовательскую модель для анализа вместо использования других встроенных моделей. Заметим, что в целях повышения наглядности для визуальной аналитики этому сервису неплохо предоставлять диаграмму.

С помощью Amazon Comprehend можно получить ответы на интересующие вопросы из текстов, организовывать и группировать документы по темам, облегчающим поиск информации. Это дает возможность сосредоточить поиск на основных темах и контексте статей, а не просто ключевых словах, что является, без сомнения, преимуществом этого сервиса.

IBM WATSON

Несмотря на растущее осознание стратегической ценности искусственного интеллекта в бизнесе, большинство организаций все еще борются с фундаментальной информационной архитектурой [11, с. 85–86]. Рутинная работа по поиску, сбору, организации изолированных данных, подготовке этих данных для анализа часто замедляет разработку искусственного интеллекта. Согласно прогнозу исследовательской компании Gather на 2020 год, предприятиям необходимо быстро и легко пользоваться моделями искусственного интеллекта. Для достижения этой цели важную роль будет играть автоматизация.

IBM Watson — суперкомпьютер фирмы IBM, оснащенный системой искусственного интеллекта, созданный группой исследователей под руководством Дэвида Феруччи. Назван в честь первого президента IBM Томаса Уотсона [12].

Основная задача IBM Watson — понимать вопросы, сформулированные на естественном языке, и находить на них ответы с помощью искусственного интеллекта.

Auto AI — это мощный набор возможностей в компьютере Watson для автоматизации многих, часто сложных и трудоемких, задач, связанных с проектированием, оптимизацией и управлением моделями ИИ на предприятии. В результате специалисты по обработке данных могут быть освобождены, чтобы посвятить больше времени разработке, тестированию и развертыванию моделей машинного обучения [13].

Чтобы ускорить разработку и внедрение моделей машинного обучения, сервис автоматизировал следующие функции [14]:

1. Подготовка данных. Применяются различные алгоритмы или оценки для анализа, очистки и подготовки необработанных данных для машинного обучения. Сервис автоматически обнаруживает и классифицирует функции на основе типа данных, таких как категориальные или числовые.

2. Выбор модели. Используется новый подход, который позволяет тестировать и ранжировать алгоритмы-кандидаты по небольшим подмножествам данных, постепенно увеличивая размер подмножества для наиболее перспективных алгоритмов для достижения наилучшего соответствия.

3. Проектирование объектов. Сервис автоматизирует проектирование объектов, которое пытается преобразовать необработанные данные в комбинацию функций, которая наилучшим образом представляет решение задачи для достижения наиболее точного прогноза.

4. Оптимизация гиперпараметров. Уточняет наиболее эффективные модели конвейеров. Для этого находится кортеж гиперпараметров, который дает оптимальную модель, оптимизирующую заданную функцию потерь на заданных независимых данных.

Одной из интересных функций Watson является работа с гиперпараметрами.

Оптимизация гиперпараметров (ОГ) — это механизм автоматического исследования пространства поиска потенциальных гиперпараметров, построения серии моделей и сравнения моделей с использованием интересующих мет-

рик. Чтобы использовать ОГ, необходимо указать диапазоны значений для изучения каждого гиперпараметра [15, 16].

В настоящее время Watson поддерживает два метода ОГ.

Метод Random реализует простой алгоритм, который будет случайным образом присваивать значения гиперпараметров из диапазонов, заданных для эксперимента.

RBFOpt использует метод, называемый радиальной базисной функцией ядра, для исследования пространства поиска. Эффективное определение параметров нейронных сетей является сложной задачей из-за чрезвычайно большого конфигурационного пространства и вычислительных затрат на оценку предлагаемой конфигурации (оценка одной конфигурации может занять от нескольких часов до нескольких дней). Для решения этой сложной задачи RBFOpt использует основанный на модели алгоритм глобальной оптимизации, который не требует производных. Аналогично байесовской оптимизации, которая приспособливает Гауссову модель к неизвестной целевой функции, этот метод подходит к модели радиальной базисной функции [17].

Оптимизация гиперпараметров требует установки верхнего предела количества моделей, которые она будет строить. Из количества настраиваемых N гиперпараметров, метод использует первые $N+1$ модели, построенные в качестве базовых моделей, после чего он начинает фактическую оптимизацию.

Производительность модели измеряется с использованием числовых и графических подходов. В бинарной классификации информативные меры обобщения могут быть получены из таблицы для расчета чувствительности (sensitivity, SE), специфичности (specificity, SP) и точности (accuracy, Acc) среди других показателей.

Чувствительность SE, или истинно положительный показатель (true positive rate, TPR), используется для количественной оценки, насколько эффективно классификаторы правильно распознают фактические положительные случаи. Специфичность SP, или истинно отрицательный показатель (true negative rate, TNR), обеспечивает способность классификатора правильно распознать фактические негативные случаи. Следовательно, SE и SP могут быть определены как (8) и (9) соответственно:

$$SE = \frac{TP}{TP+FN}, \quad (8)$$

$$SP = \frac{TN}{TN+FP}. \quad (9)$$

где TP — истинно положительный показатель (true positive), FP — ложно положительный показатель (false positive), TN — истинно отрицательный показатель (true negative) и FN — ложно отрицательный показатель (false negative)

Положительные и отрицательные прогностические значения (PPV и NPV соответственно) являются долями прогнозируемых положительных и отрицательных показателей.

Следовательно,

$$PPV = \frac{TP}{TP+FP}, \quad (10)$$

$$NPV = \frac{TN}{FN+TN}. \quad (11)$$

Точность классификации, определенная в (12), представляет собой процент от общего количества предметов, классифицированных правильно и часто используемых для оценки качества прогностических моделей,

$$Acc = \frac{TP+TN}{N}, \quad (12)$$

где $N = TP + TN + FP + FN$.

Однако эта мера эффективности может вводить в заблуждение, особенно в больших наборах несбалансированных данных с множеством классов категорий, так как в целом точность зависит от частоты класса. Чтобы преодолеть это ограничение, сбалансированная точность (balanced accuracy, bAcc) рассматривается и используется в Watson вместе с другими методами оценки производительности модели. Формально bAcc может быть определено следующим образом:

$$bAcc = \frac{1}{2} \left(\frac{TP}{P} + \frac{TN}{N} \right). \quad (13)$$

Кривая рабочей характеристики приемника (receiver operating characteristic, ROC) [18] представляет собой стандартный метод, используемый в качестве графического показателя производительности, чтобы суммировать прогнозные показатели бинарной классификации. Кривая ROC строит график TPR по оси y против ложных срабатываний FPR по оси x , производимых классификационной моделью, где каждая точка на кривой ROC соответствует классификатору.

Количественную интерпретацию ROC дает показатель AUC [18] (англ. *area under ROC curve* — площадь под ROC-кривой) — площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций. Чем выше показатель AUC, тем качественнее классификатор, при этом значение 0,5 демонстрирует непригодность выбранного метода классификации (соответствует случайному гаданию). Значение менее 0,5 говорит о том, что классификатор действует с точностью до наоборот: если положительные назвать отрицательными и наоборот, классификатор будет работать лучше. Чтобы правильно измерить AUC, важно определить его доверительный интервал (confidence interval, CI). В Watson CI вычисляются с использованием пакета pROC.

Метрика F1, также известная как F-оценка, или F-мера, принимает во внимание точность классификации как показано в (14).

$$F_1 = 2 * \left(\frac{PPV * TPR}{PPV + TPR} \right). \quad (14)$$

Для реализации Auto AI в любой разработке AI требуются следующие инструменты или сервисы:

1. Watson Studio — интегрированная среда для проектирования, разработки, развертывания и тестирования моделей с помощью функции автоматического AI.

2. Watson Machine Learning. Этот сервис требуется для развертывания модели ML.

3. Watson Open Scale. Сервис необходим для управления моделью ML в производстве.

4. Хранилище объектов. Необходимо для хранения набора данных.

IBM Watson — это инструмент, который ускоряет развертывание искусственного интеллекта и машинного обучения. Но для того, чтобы пользоваться всеми преимуществами данного сервиса, нужно подключать много других платных сервисов, что является недостатком этого сервиса.

Также в качестве недостатка Amazon и IBM можно выделить закрытость базовых моделей разработчика. Фактически пользователь имеет дело с «черным ящиком», который не дает полного понимания того, на каких алгоритмах работает их ИИ, тем самым тем самым хороня идеи, улучшающие работу данных сервисов, которые могли бы появиться у специалистов из других фирм.

ЗАКЛЮЧЕНИЕ

Искусственный интеллект напрямую соотносится с Data Science — наукой о данных, направленной на извлечение бизнес-ценности из массива информации [19].

ИИ оперирует огромными массивами, анализирует поступающие данные и разрабатывает на их основе адаптивные решения.

Вот несколько проблем, которые можно решить с помощью алгоритмов машинного обучения:

- оперативное реагирование;
- разработка стратегий развития бизнеса;
- решение проблем, связанных с человеческим фактором;
- борьба с мошенничеством;
- увеличение прибыли и пр.

Но, к сожалению, большинство внедряющих технологии ИИ компаний пока так и не научились извлекать из них реальную выгоду. Проблема в том, что чаще всего компании пока ограничиваются в сфере ИИ маленькими проектами, которые не дают серьезной отдачи.

Использование искусственного интеллекта постепенно становится необходимостью во всех отраслях бизнеса. Настоящий эффект от внедрения ИИ может быть достигнут только при масштабировании пилота, который показал результативность, на всю компанию либо на достаточно крупные ее подразделения. Эксперты отмечают, что чаще успеха добиваются те, для кого ИИ — инструмент увеличения доходов, а не средство снижения издержек.

Если компания хочет добиться успеха в мире бизнеса, она должна отлично разбираться в новейших технологиях. ИИ — 1718 одна из наиболее удивительных и значимых современных разработок, преимуществами которой стоит обязательно воспользоваться.

ЛИТЕРАТУРА

1. Обработка естественного языка // Википедия. Дата обновления: 27.05.2020.

URL: http://ru.wikipedia.org/wiki/Обработка_естественного_языка. (дата обращения 22.06.2020).

2. Bird S. Natural Language Processing with Python. — O'Reilly Media, 2009. — 502 p.

3. Russel S. Artificial Intelligence: A modern Approach / S. Russel, P. Norvig. — Third Edition. — Pearson, 2009. — 1152 p.

4. Будников Е. А. Обзор некоторых статистических моделей естественных языков // Машинное обучение и анализ данных. 2011. Т. 1, № 2. С. 245–250.
5. Степанов П. А. Автоматизация обработки текстов естественного языка // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2013. Т. 11, Вып. 2. С. 106–112.
6. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — Cambridge: The MIT Press, 2016. — 800 p.
7. Gartner Predicts 70 Percent of Organizations Will Integrate AI to Assist Employees' Productivity by 2021. — 24.01.2019. URL: <http://www.gartner.com/en/newsroom/press-releases/2019-01-24-gartner-predicts-70-percent-of-organizations-will-int> (дата обращения 22.06.2020).
8. Amazon Comprehend — обработка естественного языка (NLP) и машинное обучение (ML) // Amazon Web Services. URL: <http://aws.amazon.com/ru/comprehend> (дата обращения 22.06.2020).
9. Escalona T. Detect Sentiment from Customer Reviews Using Amazon Comprehend — 26.01.2018 // AWS Machine Learn Blog / Amazon Web Services. URL: <http://aws.amazon.com/ru/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-amazon-comprehend> (дата обращения 28.06.2020).
10. Методы и модели исследования сложных систем и обработки больших данных: Монография / И. Ю. Парамонов, В. А. Смагин, Н. Е. Косых, А. Д. Хомоненко; под ред. В. А. Смагина, А. Д. Хомоненко. — СПб.: Лань, 2020. — 236 с. — (Учебники для вузов. Специальная литература).
11. Модели и методы исследования информационных систем: Монография / А. Д. Хомоненко, В. П. Бубнов, А. В. Забродин [и др.]; под ред. А. Д. Хомоненко. — СПб.: Лань, 2019. — 204 с. — (Учебники для вузов. Специальная литература).
12. IBM Watson // IBM — Российская Федерация. URL: <http://www.ibm.com/ru-ru/watson> (дата обращения 22.06.2020).
13. Serving COVID-19 Epidemic Models with Watson Machine Learning — 07.04.2020 / L. Cmielowski, A. Quemy, R. Bigaj, W. Sobala // IBM Watson / Medium. URL: <http://medium.com/ibm-watson/serving-covid-19-epidemic-models-with-watson-machine-learning-378b6fe9407b> (дата обращения 23.06.2020).
14. IBM Watson Machine Learning vs Oracle Machine Learning // TrustRadius. URL: <http://www.trustradius.com/compare-products/ibm-watson-machine-learning-vs-oracle-machine-learning> (дата обращения 22.06.2020).
15. Токмакова А. А. Получение устойчивых оценок гиперпараметров линейных регрессионных моделей // Машинное обучение и анализ данных. 2011. Т. 1, № 2. С. 140–155.
16. Раднаев Ч. Б. Оптимизация гиперпараметров сверточной нейронной сети // Молодежь и современные технологии: Сб. трудов XV Международной научно-практической конференции студентов, аспирантов и молодых ученых (Томск, 04–07 декабря, 2017 г.). — Томск: Национальный исследовательский Томский политехнический университет, 2018. — С. 81–82.
17. An Effective Algorithm for Hyperparameter Optimization of Neural Networks. / A. Fokoue, G. I. Diaz, G. Nannicini, H. Samulowitz // IBM Journal of Research and Development. 2017. Vol. 61, No. 4–5, 9 p. DOI:10.1147/JRD.2017.2709578.
18. Богданов Л. Ю. Оценка эффективности бинарных классификаторов на основе логической регрессии методом ROC-анализа // Вестник Саратовского государственного технического университета. 2010. № 4 (50), Вып. 2. С. 93–97.
19. Юрченко А. В. К концепции информационно-аналитической системы поддержки научных исследований, основанных на интенсивном использовании цифровых данных // Вычислительные технологии. 2017. Т. 22, № 4. С. 105–120.

Research of Modern Text Information Analyzers

Master of Science A. A. Bryzgalov

Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
threnhawk@gmail.com

Abstract. An analytical overview of two well-known software systems that solve natural language processing tasks is presented: Amazon Comprehend, a natural language processing service, and IBM-Watson, an IBM supercomputer equipped with an artificial intelligence system. Their analysis is carried out, features and shortcomings of the considered services are revealed. Using information on the history of PGUPS, Amazon Comprehend service was tested, undocumented restrictions were revealed when processing Russian text.

Keywords: natural language, artificial intelligence, NLP, machine learning, Amazon Comprehend, IBM-Watson.

REFERENCES

1. Natural Language Processing [Obrabotka estestvennogo yazyka], *Wikipedia [Vikipediya]*. Update May 27, 2020. Available at: http://ru.wikipedia.org/wiki/Обработка_естественного_языка (accessed 22 June 2020).
2. Bird S. *Natural Language Processing with Python*. — O'Reilly Media, 2009. — 502 p.
3. Russel S., Norvig P. *Artificial Intelligence: A modern Approach*. Third Edition. Pearson, 2009, 1152 p.
4. Budnikov E. A. Overview of Some Statistical Models of Natural Languages [Obzor nekotorykh statisticheskikh modeley estestvennykh yazykov], *Journal of Machine Learning and Data Analysis [Mashinnoe obuchenie i analiz dannykh]*, 2011, Vol. 1, No. 2, Pp. 245–250.
5. Stepanov P. A. Processing Automation of Natural Language Texts [Avtomatizatsiya obrabotki tekstov estestvennogo yazyka], *Vestnik of Novosibirsk State University. Series: Information Technologies [Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnye tekhnologii]*, 2013, Vol. 11, Is. 2, Pp. 106–112.
6. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge, The MIT Press, 2016, 800 p.
7. Gartner Predicts 70 Percent of Organizations Will Integrate AI to Assist Employees' Productivity by 2021. Published at January 24, 2019. Available at: <http://www.gartner.com/en/newsroom/press-releases/2019-01-24-gartner-predicts-70-percent-of-organizations-will-int#> (accessed 22 June 2020).
8. Amazon Comprehend — Natural Language Processing (NLP) and Machine Learning (ML) [Amazon Comprehend — obrabotka estestvennogo yazyka (NLP) i mashinnoe obuchenie (ML)], Amazon Web Services. Available at: <http://aws.amazon.com/ru/comprehend> (accessed 22 June 2020).
9. Escalona T. Detect Sentiment from Customer Reviews Using Amazon Comprehend, AWS Machine Learn Blog. Amazon Web Services. Published at January 26, 2018. Available at: <http://aws.amazon.com/ru/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-ama-zon-comprehend> (accessed 28 June 2020).
10. Paramonov I. Yu., Smagin V. A., Kosykh N. E., Khomonenko A. D. *Methods and models of research of complex systems and big data processing; Monograph [Metody i modeli issledovaniya slozhnykh sistem i obrabotki bol'shikh dannykh: Monografiya]*, St. Petersburg, LAN Publishing House, 2020, 236 p.
11. Khomonenko A. D., Bubnov V. P., Zabrodin A. V., et al. *Models and methods of research of information systems: Monograph [Modeli i metody issledovaniya informatsionnykh sistem: Monografiya]*, St. Petersburg, LAN Publishing House, 2019, 204 p.
12. IBM Watson, IBM — Russian Federation [IBM — Rossiyskaya Federatsiya]. Available at: <http://www.ibm.com/ru-ru/watson> (accessed 22 June 2020).
13. Cmielowski L., Quemy A., Bigaj R., Sobala W. *Serving COVID-19 Epidemic Models with Watson Machine Learning, IBM Watson*. Medium. Published at April 07, 2020. Available at: <http://medium.com/ibm-watson/serving-covid-19-epide-mic-models-with-watson-machine-learning-378b6fe9407b> (accessed 23 June 2020).
14. IBM Watson Machine Learning vs Oracle Machine Learning, TrustRadius. Available at: <http://www.trustradius.com/compare-products/ibm-watson-machine-learning-vs-oracle-machine-learning> (accessed 22 June 2020).
15. Tokmakova A. A. Obtaining Stable Estimates of Hyperparameters of Linear Regression Models [Poluchenie ustoychivyykh otsenok giperparametrov lineynykh regressionnykh modeley], *Journal of Machine Learning and Data Analysis [Mashinnoe obuchenie i analiz dannykh]*, 2011, Vol. 1, No. 2, Pp. 140–155.
16. Radnaev Ch. B. Optimization of Convolutional Neural Network Hyperparameters [Optimizatsiya giperparametrov svertochnoy neyronnoy seti], *Youth and modern technologies: Proceedings of the XV International Scientific and Practical Conference of Students, Postgraduates and Young Scientists [Molodezh' i sovremennye tekhnologii: Sbornik trudov XV Mezhdunarodnoy nauchno-prakticheskoy konferentsii studentov, aspirantov i molodykh uchenykh]*, Tomsk, December 4–7, 2017, Tomsk, National Research Tomsk Polytechnic University, 2018, Pp. 81–82.
17. Fokoue A., Diaz G. I., Nannicini G., Samulowitz H. An Effective Algorithm for Hyperparameter Optimization of Neural Networks. *IBM Journal of Research and Development*, 2017, Vol. 61, No. 4–5, 9 p. DOI:10.1147/JRD.2017.2709578.
18. Bogdanov L. Y. The Evaluation of Performance of Binary Classifiers Based on Logistic Regression Using Roc Analysis [Otsenka effektivnosti binarnykh klassifikatorov na osnove logicheskoy regressii metodom ROC-analiza], *Vestnik Saratov*

State Technical University [Vestnik Saratovskogo gosudarstvennogo tekhnicheskogo universiteta], 2010, No. 4 (50). Is. 2, Pp. 93–97.

19. Yurchenko A. V. On the Concept of Information-Analytical System for Supporting Data Intensive Science [K kont-

septsii informatsionno-analiticheskoy sistemy podderzhki nauchnykh issledovaniy, osnovannykh na intensivnom ispol'zovanii tsifrovyykh dannykh], Computational Technologies [Vychislitel'nye tekhnologii], 2017, Vol. 22, No. 4, Pp. 105–120.