

ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

2019
Т. 8, № 3

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий,
в которых должны быть опубликованы результаты диссертаций
на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет
(национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Искусственный интеллект и машинное обучение
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

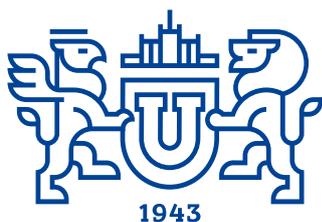
Редакционная коллегия

Л.Б. Соколинский, д.ф.-м.н., проф., *гл. редактор*
В.П. Танана, д.ф.-м.н., проф., *зам. гл. редактора*
М.Л. Цымблер, к.ф.-м.н., доц., *отв. секретарь*
Г.И. Радченко, к.ф.-м.н., доц.
Я.А. Краева, *техн. секретарь*

Редакционный совет

С.М. Абдуллаев, д.г.н., профессор
А. Андреяк, PhD, профессор (Германия)
В.И. Бердышев, д.ф.-м.н., акад. РАН, *председатель*
В.В. Воеводин, д.ф.-м.н., чл.-кор. РАН

Дж. Донгарра, PhD, профессор (США)
С.В. Зыкин, д.т.н., профессор
Д. Маллманн, PhD, профессор (Германия)
А.В. Панюков, д.ф.-м.н., профессор
Р. Продан, PhD, профессор (Австрия)
А.Н. Томилин, д.ф.-м.н., профессор
В.Е. Третьяков, д.ф.-м.н., чл.-кор. РАН
В.И. Ухоботов, д.ф.-м.н., профессор
В.Н. Ушаков, д.ф.-м.н., чл.-кор. РАН
М.Ю. Хачай, д.ф.-м.н., профессор
А. Черных, PhD, профессор (Мексика)
П. Шумяцкий, PhD, профессор (Бразилия)



BULLETIN

OF THE SOUTH URAL STATE UNIVERSITY 2019
vol. 8, no. 3

SERIES

“COMPUTATIONAL
MATHEMATICS AND SOFTWARE
ENGINEERING”

ISSN 2305-9052

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya “Vychislitel'naya Matematika i Informatika”

South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- Artificial intelligence and machine learning
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

Editorial Board

L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russia)
V.P. Tanana, South Ural State University (Chelyabinsk, Russia)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russia)
G.I. Radchenko, South Ural State University (Chelyabinsk, Russia)
Ya.A. Kraeva, South Ural State University (Chelyabinsk, Russia)

Editorial Council

S.M. Abdullaev, South Ural State University (Chelyabinsk, Russia)
A. Andrzejak, Heidelberg University (Germany)
V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
J. Dongarra, University of Tennessee (USA)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
D. Mallmann, Julich Supercomputing Centre (Germany)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russia)
R. Prodan, University of Innsbruck (Innsbruck, Austria)
P. Shumyatsky, University of Brasilia (Brazil)
A. Tchernykh, CICESE Research Center (Mexico)
A.N. Tomilin, Institute for System Programming of the RAS (Moscow, Russia)
V.E. Tretyakov, Ural Federal University (Yekaterinburg, Russia)
V.I. Ukhobotov, Chelyabinsk State University (Chelyabinsk, Russia)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
V.V. Voevodin, Lomonosov Moscow State University (Moscow, Russia)
S.V. Zykin, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Omsk, Russia)

Содержание

| | |
|--|----|
| СХОДИМОСТЬ МЕТОДА КОНЕЧНЫХ ЭЛЕМЕНТОВ ДЛЯ КРАЕВОЙ ЗАДАЧИ С ВЫРОЖДЕНИЕМ НА ВСЕЙ ГРАНИЦЕ ОБЛАСТИ Е.И. Рукавишникова | 5 |
| ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ СИЛОВЫХ НАГРУЗОК НА ОПОРЫ НАДВОДНОЙ КОНСТРУКЦИИ НА ОСНОВЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ВОЛНОВЫХ ПРОЦЕССОВ С.В. Проценко, А.М. Атаян, А.Е. Чистяков, А.В. Никитина, В.Н. Литвинов, А.А. Филина | 27 |
| ОБЗОР ПОДХОДОВ И ПРАКТИЧЕСКИХ ОБЛАСТЕЙ ПРИМЕНЕНИЯ РАСПОЗНАВАНИЯ ВИДОВ ФИЗИЧЕСКОЙ АКТИВНОСТИ ЧЕЛОВЕКА Е.С. Тарантова, К.В. Макаров, А.А. Орлов | 43 |
| ОБЗОР МОДЕЛЕЙ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ Н.А. Ежова, Л.Б. Соколинский | 58 |
| ТЕНДЕНЦИИ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНЫХ УЗЛОВ СОВРЕМЕННЫХ СУПЕРКОМПЬЮТЕРОВ Е.О. Тютляева, И.О. Одинцов, А.А. Московский, Г.В. Мармузов | 92 |

Contents

| | |
|--|----|
| CONVERGENCE OF THE FINITE ELEMENT METHOD FOR BOUNDARY VALUE PROBLEM WITH DEGENERATION ON WHOLE BOUNDARY OF THE DOMAIN E.I. Rukavishnikova | 5 |
| EXPERIMENTAL RESEARCH OF POWER LOADS ON THE SUPPORTS OF THE SURFACE STRUCTURE BASED ON THE MATHEMATICAL MODEL OF WAVE PROCESSES S.V. Protsenko, A.M. Atayan, A.E. Chistyakov, A.V. Nikitina, V.N. Litvinov, A.A. Filina | 27 |
| SURVEY ON APPROACHES AND PRACTICAL AREAS OF HUMAN ACTIVITY RECOGNITION APPLICATION E.S. Tarantova, K.V. Makarov, A.A. Orlov | 43 |
| SURVEY OF PARALLEL COMPUTATION MODELS N.A. Ezhova, L.B. Sokolinsky | 58 |
| DEVELOPMENT TRENDS OF MODERN SUPERCOMPUTERS E.O. Tyutlyaeva, I.O. Odintsov, A.A. Moskovsky, G.V. Marmuzov | 92 |



This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

СХОДИМОСТЬ МЕТОДА КОНЕЧНЫХ ЭЛЕМЕНТОВ ДЛЯ КРАЕВОЙ ЗАДАЧИ С ВЫРОЖДЕНИЕМ НА ВСЕЙ ГРАНИЦЕ ОБЛАСТИ

© 2019 Е.И. Рукавишникова

Вычислительный центр ДВО РАН
(680000 Хабаровск, ул. Ким Ю Чена, д. 65)

E-mail: rukavishnikova-55@mail.ru

Поступила в редакцию: 04.09.2018

В статье рассматривается задача Дирихле с однородным граничным условием для эллиптического уравнения второго порядка с вырождением на всей дважды непрерывно дифференцируемой границе двумерной области Ω . Определяется обобщенное решение этой задачи, которое существует и единственно в весовом пространстве $\dot{W}_{2,\alpha}^1(\Omega)$. Для решения сформулированной задачи разработан метод конечных элементов, схема которого построена на основе определения обобщенного решения исходной дифференциальной задачи в пространстве $\dot{W}_{2,\alpha}^1(\Omega)$. С этой целью двумерная выпуклая область разбивается на треугольники со специальным сгущением к границе. Далее, введено пространство конечных элементов $V^h \subset \dot{W}_{2,\alpha}^1(\Omega)$, которое содержит непрерывные функции, линейные на каждом треугольном элементе сеточной области Ω^h и равные нулю на множестве $\bar{\Omega} \setminus \Omega^h$, показана однозначная разрешимость схемы метода конечных элементов. Для обобщенного решения u из подпространства $\dot{W}_{2,\alpha-1}^2(\Omega)$ пространства $\dot{W}_{2,\alpha}^1(\Omega)$, используя значения в узлах триангулированной области Ω^h , строится интерполиант $u_I \in V^h$, устанавливается факт его сходимости по норме $W_{2,\alpha}^1(\Omega)$. Главным результатом работы является доказательство сходимости приближенного решения предложенного метода к точному решению в весовом пространстве Соболева.

Ключевые слова: краевая задача с вырождением, весовое пространство Соболева, обобщенное решение, метод конечных элементов.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Рукавишникова Е.И. Сходимость метода конечных элементов для краевой задачи с вырождением на всей границе области // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 5–26. DOI: 10.14529/cmse190301.

Введение

Сингулярность решения краевых задач может быть вызвана особенностью исходных данных (коэффициентов, правых частей уравнения и граничных условий), геометрией границы области, внутренними свойствами решения. Краевая задача называется задачей с сильной сингулярностью, если ее решение $u \notin H^1(W_2^1)$ или интеграл Дирихле от функции u расходится. Для таких задач, когда решение имеет сингулярность на конечном множестве точек на границе области, были разработаны различные методы конечных элементов, позволяющие находить приближенное решение (см. [1–10]). В том случае, когда $u \in H^1$ и $u \notin H^2$ краевую задачу называют задачей со слабой сингулярностью решения. В работе [11] была сформулирована краевая задача с вырождением решения на всей границе области, исследовано его существование и единственность в весовом пространстве С.Л. Соболева. В [12–14] изучены коэрцитивные и дифференциальные свойства решения.

Целью данной работы является построение метода конечных элементов (МКЭ) для первой краевой задачи для эллиптического уравнения второго порядка с вырождением на всей дважды непрерывно дифференцируемой границе двумерной области Ω и

исследование сходимости приближенного решения МКЭ к точному решению в норме весового пространства $W_{2,\alpha}^1(\Omega)$.

В разделе 1 приведена постановка исходной дифференциальной задачи, сформулированы вспомогательные сведения и результаты, из которых следует существование и единственность ее обобщенного решения. В разделе 2 описано построение схемы МКЭ, первым шагом в осуществлении которого является триангуляция двумерной выпуклой области со специальным сгущением треугольных элементов к границе. Далее, введено пространство конечных элементов и показана единственность приближенного решения МКЭ в нем. В разделе 3 устанавливается сходимость приближенного решения построенного метода к точному обобщенному решению задачи по норме весового пространства $W_{2,\alpha}^1$. В заключении сформулированы основные результаты исследования аппроксимации решения задачи Дирихле для эллиптического уравнения второго порядка с вырождением на границе, указаны области их применения и определено направление дальнейшей работы.

1. Постановка задачи

Пусть R^2 — двумерное евклидово пространство с произвольной точкой $x = (x_1, x_2)$ в нем и $dx = dx_1 dx_2$; $\Omega \subset R^2$ — ограниченная область с достаточно гладкой границей $\partial\Omega$, $\bar{\Omega}$ — замыкание области, т.е. $\bar{\Omega} = \Omega \cup \partial\Omega$.

Определим весовую функцию $\rho(x)$ как расстояние любой точки $x \in \Omega$ до границы $\partial\Omega$.

Далее положим, что $s \in \{1; 2\}$, γ — действительное число, удовлетворяющее условию $1/2 - s < \gamma < 1/2$. Через $W_{2,\gamma}^s(\Omega)$ обозначим весовое пространство Соболева функций f , заданных на Ω , с конечной нормой

$$\|f\|_{W_{2,\gamma}^s(\Omega)} = \|f\|_{L_2(\Omega)} + |f|_{W_{2,\gamma}^s(\Omega)}, \quad (1)$$

эквивалентной [15] норме

$$\|f\|_{W_{2,\gamma}^{s*}(\Omega)} = \|f\|_{L_{2,\gamma}(\Omega)} + |f|_{W_{2,\gamma}^s(\Omega)}, \quad (2)$$

где

$$\|f\|_{L_2(\Omega)} = \left(\int_{\Omega} |f|^2 dx \right)^{1/2}, \quad \|f\|_{L_{2,\gamma}(\Omega)} = \left(\int_{\Omega} \rho^{-2\gamma} |f|^2 dx \right)^{1/2},$$

$$|f|_{W_{2,\gamma}^s(\Omega)} = \sum_{\substack{k_1, k_2=0 \\ |k|=s}}^s \left\| \rho^{-\gamma} \frac{\partial^{|k|} f}{\partial x_1^{k_1} \partial x_2^{k_2}} \right\|_{L_2(\Omega)}, \quad k = (k_1, k_2), \quad k_1 \geq 0, \quad k_2 \geq 0 \text{ — целые числа,}$$

$|k| = k_1 + k_2$. При $\gamma = 0$ пространства $W_{2,0}^s(\Omega)$ и $W_2^s(\Omega)$ совпадают.

Введем пространство функций f

$$\dot{W}_{2,\gamma}^s(\Omega) = \{f : f \in W_{2,\gamma}^s(\Omega), \quad f|_{\partial\Omega} = 0\}$$

и пространство $L_{2,-1-\gamma}$ функций $F(x)$, $x \in \Omega$, с конечной нормой

$$\|F\|_{L_{2,-1-\gamma}(\Omega)} = \left(\int_{\Omega} |\rho^{1+\gamma} F|^2 dx \right)^{1/2}.$$

Отметим [15], что верно вложение пространств

$$W_{2,\gamma}^s(\Omega) \subset W_{2,\gamma_l}^{s-l}(\Omega) \quad (3)$$

при $\gamma_l < 1/2$, $\gamma_l \leq \gamma + l$, $0 \leq l \leq s$, $\partial\Omega \in C^{(1)}$, в частности, $W_{2,\gamma_0}^0(\Omega) = L_{2,\gamma_0}(\Omega)$, если $s = l$. Также имеют место неравенства [11]

$$\|f\|_{L_2(\Omega)} \leq C_1 \|f\|_{W_{2,\gamma}^s(\Omega)}, \quad \|\rho^{-s-\gamma} f\|_{L_2(\Omega)} \leq C_2 \|f\|_{W_{2,\gamma}^s(\Omega)}, \quad (4)$$

справедливые для функций f из $\dot{W}_{2,\gamma}^s(\Omega)$, если $\partial\Omega \in C^{(s_0+1)}$, s_0 — наименьшее натуральное число такое, что $s + \gamma - 1/2 \leq s_0 < s + \gamma + 1/2$, $s \leq 2s_0$, с константами C_1, C_2 , не зависящими от f .

В области Ω рассмотрим дифференциальное уравнение

$$-\sum_{k,l=1}^2 \frac{\partial}{\partial x_k} \left(a_{kl}(x) \frac{\partial u}{\partial x_l} \right) + a(x)u = F(x), \quad x \in \Omega, \quad (5)$$

с краевым условием

$$u|_{\partial\Omega} = 0, \quad \partial\Omega \in C^{(2)}. \quad (6)$$

Правая часть уравнения (5) удовлетворяет включению

$$F \in L_{2,-1-\alpha}(\Omega), \quad (7)$$

т.е. $\rho^{1+\alpha} F \in L_2(\Omega)$, $-1/2 < \alpha < 1/2$.

Коэффициенты уравнения $a_{kl}(x) = a_{lk}(x)$ ($k, l = 1, 2$) — дифференцируемые на Ω функции, удовлетворяют неравенствам

$$|a_{kl}(x)| \leq C_3 \rho^{-2\alpha}(x), \quad (8)$$

$$\left| \frac{\partial a_{kl}(x)}{\partial x_1} \right|, \left| \frac{\partial a_{kl}(x)}{\partial x_2} \right| \leq C_4 \rho^{-2\alpha-1}(x), \quad (9)$$

функция $a(x)$ — положительная и подчинена неравенству

$$a(x) \leq C_5 \rho^{-2\alpha-2}(x). \quad (10)$$

Предполагается выполненным условие ультраэллиптичности

$$\sum_{k,l=1}^2 a_{kl}(x) \zeta_k \zeta_l \geq C_6 \rho^{-2\alpha}(x) \sum_{k=1}^2 \zeta_k^2, \quad x \in \Omega, \quad C_6 > 0, \quad (11)$$

где ζ_1, ζ_2 — любые действительные параметры, C_3-C_6 — константы, не зависящие от x .

Введем соответственно билинейную и линейную формы

$$E(u, v) = \int_{\Omega} \left[\sum_{k,l=1}^2 a_{kl} \frac{\partial u}{\partial x_k} \frac{\partial v}{\partial x_l} + auv \right] dx, \quad (F, v) = \int_{\Omega} Fv dx.$$

Наряду с задачей (5), (6) сформулируем следующую вариационную задачу: найти функцию $u \in \dot{W}_{2,\alpha}^1(\Omega)$, для которой равенство

$$E(u, v) = (F, v) \quad (12)$$

справедливо для всех $v \in \dot{W}_{2,\alpha}^1(\Omega)$.

Функцию $u \in \dot{W}_{2,\alpha}^1(\Omega)$, обладающую свойством (12), назовем обобщенным решением уравнения (5) в пространстве $\dot{W}_{2,\alpha}^1(\Omega)$.

Заметим, что в силу соотношений (4), (7), (8), (10), (11) при $s = 1$, $\gamma = \alpha$ (см. [11]) билинейная форма $E(u, v)$ является непрерывной на пространстве $\dot{W}_{2,\alpha}^1(\Omega)$ и $\dot{W}_{2,\alpha}^1(\Omega)$ -эллиптической, а линейная форма (F, v) непрерывна на этом же пространстве. На основании теоремы Лакса—Мильграма [16], существует единственное решение вариационной задачи (12).

Кроме этого, из работы [13] следует, что если выполнено условие (7), то обобщенное решение $u \in \dot{W}_{2,\alpha}^1(\Omega)$ задачи (5), (6) принадлежит пространству $\dot{W}_{2,\alpha-1}^2(\Omega)$. Ввиду соотношения (3) при $s = 2$, $\gamma = \alpha - 1$, $l = 1$, $\gamma_l = \alpha$, имеет место вложение пространств $\dot{W}_{2,\alpha-1}^2(\Omega) \subset \dot{W}_{2,\alpha}^1(\Omega)$.

Для решения вариационной задачи (12) применим метод конечных элементов.

2. Построение схемы метода конечных элементов

Полагая, что область определения решения Ω выпуклая, произведем ее триангуляцию T_h (рис. 1). Для этого через точки, находящиеся от границы $\partial\Omega$ на расстояниях, равных $b\left(\frac{j}{n}\right)^r$, $j = 0, \dots, n$, $r > 1$ ($b < \delta_\Omega/2$, δ_Ω — диаметр вписанной в Ω окружности, r — параметр сжатия, характеризующий степень сгущения точек), проводятся кривые Γ_j , $j = 0, \dots, n$, разделяющие область Ω на слои Q_j , $j = 1, \dots, n$. Линия Γ_n при этом делит Ω на две подобласти: внешнюю Ω_1 (приграничную полосу) и внутреннюю Ω_2 . На каждой кривой Γ_j , $j = 0, \dots, n$ ($\Gamma_0 = \partial\Omega$, $\Gamma_n = \partial\Omega_1$) фиксируются M_j равноотстоящих узлов. Число M_j , $j = 1, \dots, n$, определяется функцией $\psi(j) = \left[l_j / \left(b \left(\left(\frac{j}{n} \right)^r - \left(\frac{j-1}{n} \right)^r \right) \right) \right] + 1$, где l_j — длина кривой Γ_j , $M_0 = 2M_1$. Соединением сначала последовательно всех точек на кривых Γ_j ломаными линиями, а затем каждого из узлов на Γ_{j-1} с ближайшей из узловых точек, принадлежащих кривой Γ_j , $j = 1, \dots, n$, подобласть Ω_1 разбивается на элементы треугольного типа со сгущением к границе $\partial\Omega$. Объединение всех треугольников, вершины которых являются узловыми точками только кривых Γ_{j-1} и Γ_j , $j = 1, \dots, n$, обозначим через Q_j^h .

Во внутренней подобласти Ω_2 проводится квазиравномерная триангуляция, в результате которой имеем конечное число регулярных треугольников с наибольшей стороной порядка $1/n$. При этом точки разбиения (вершины треугольников) на границе $\partial\Omega_1$ должны входить в число вершин треугольников на Ω_2 . (На рис. 1 подобласть Ω_1 разделена на слои Q_1, \dots, Q_4 и $\Gamma_4 = \partial\Omega_1$).

Проведенная триангуляция T_h области Ω удовлетворяет следующим свойствам:

1. $\bar{\Omega} = \Omega^h \cup \Omega'$, где $\Omega^h = \bigcup_{m=1}^N K_m$, $\{K\} = \{K_1, \dots, K_N\}$ — множество замкнутых треугольников, называемых конечными элементами, Ω' — объединение сегментов, отсекаемых треугольниками K_m , хотя бы одна вершина которых принадлежит границе $\partial\Omega$.
2. Общими для треугольников K могут быть только стороны или вершины.
3. Наибольшая сторона h в треугольниках K_m имеет порядок $1/n$, где n — число слоев в приграничной полосе Ω_1 .
4. $\sup_{K \in T_h} \frac{h_{max}(K)}{h_{min}(K)} \leq C_7$, где постоянная C_7 не зависит от h .

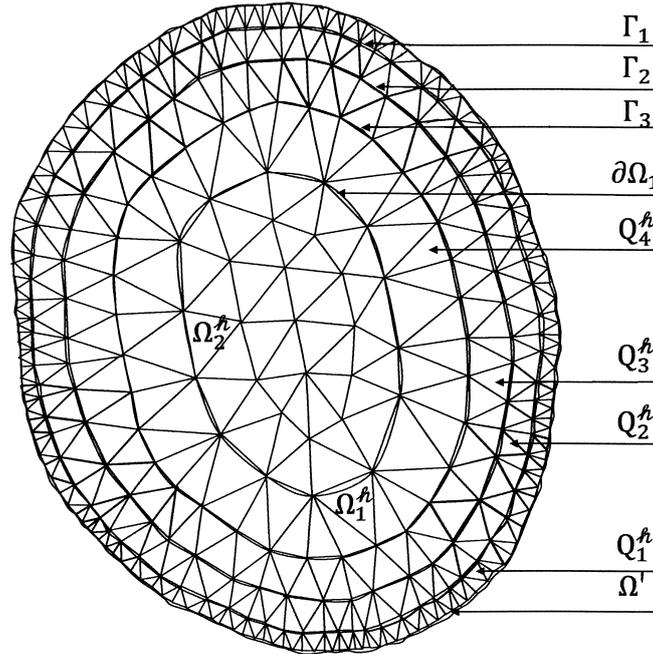


Рис. 1. Построение сеточной области Ω^h

Отметим также, что вершины P_1, \dots, P_{N_h} треугольников K являются узлами триангуляции.

Обозначим через $V^h \subset \dot{W}_{2,\alpha}^1(\Omega)$ пространство непрерывных функций, линейных на каждом K_m из триангуляции T_h и равных нулю на $\bar{\Omega} \setminus \Omega^h$. Для функции u из пространства $\dot{W}_{2,\alpha-1}^2(\Omega)$ определим интерполянт $u_I(x) = \sum_{i=1}^{N_h} u(P_i)\varphi_i(x)$, где $\varphi_i(x)$ — базисная функция, которая в точке P_i равна единице, в остальных узлах — нулю и линейна на каждом треугольнике из множества $\{K\}$.

Построенному конечномерному пространству $V^h \subset \dot{W}_{2,\alpha}^1(\Omega)$ сопоставим следующую дискретную задачу: найти функцию $u_h \in V^h$, удовлетворяющую равенству

$$E(u_h, v_h) = (F, v_h) \quad (13)$$

для любой функции v_h из V^h .

Здесь $E(u_h, v_h)$ и (F, v_h) — билинейная и линейная формы из задачи (12). Согласно теореме Лакса—Мильграма [16], задача (13) имеет единственное решение u_h , которое будем называть приближенным решением по методу конечных элементов.

3. Сходимость метода конечных элементов в весовом пространстве

Рассмотрим семейство (V^h) подпространств пространства $\dot{W}_{2,\alpha}^1(\Omega)$, где параметр h (h — наибольшая из длин сторон треугольников K в триангуляциях T_h) в пределе равен нулю. Каждому пространству конечных элементов V^h поставим в соответствие приближенное решение u_h , удовлетворяющее условию (13).

Установим сходимость метода по норме $W_{2,\alpha}^1(\Omega)$:

$$\lim_{h \rightarrow 0} \|u - u_h\|_{W_{2,\alpha}^1(\Omega)} = 0, \quad (14)$$

где u — решение задачи (12), u_h — решение задачи (13).

Лемма 1. Пусть u — решение задачи (12), u_h — решение задачи (13). Тогда существует такая положительная постоянная C_8 , не зависящая от подпространства V^h , что

$$\|u - u_h\|_{W_{2,\alpha}^1(\Omega)} \leq C_8 \inf_{v_h \in V^h} \|u - v_h\|_{W_{2,\alpha}^1(\Omega)}.$$

В силу непрерывности на пространстве $\dot{W}_{2,\alpha}^1(\Omega)$ и $\dot{W}_{2,\alpha}^1(\Omega)$ -эллиптичности билинейной формы $E(u, v)$ последнее неравенство доказывается по аналогии с [17].

Так как $\inf_{v_h \in V^h} \|u - v_h\|_{W_{2,\alpha}^1(\Omega)} \leq \|u - u_I\|_{W_{2,\alpha}^1(\Omega)}$, то для того, чтобы установить соотношение (14), покажем, что интерполянт u_I сходится к точному решению u задачи (12).

Введем следующие обозначения:

$$Q_1^h = \bigcup_{m=1}^{N_1} K_m, \text{ где } K_m \text{ — треугольники, вершины которых принадлежат слою } Q_1 \text{ из } \Omega_1$$

(Q_1 — первый слой деления подобласти Ω_1 , $\Omega_1 = \bigcup_{j=1}^n Q_j$, n — число слоев), а N_1 — число этих треугольников;

$\Omega' = \bigcup_{m=1}^{N_2} D_m$, где N_2 — число сегментов D_m , ограниченных хордами, соединяющими рядом лежащие узлы кривой Γ_0 , и дугами, которыми эти хорды стягиваются;

$$\Omega_1^{1,h} = \bigcup_{m=N_1+1}^{N_3} K_m, \text{ где } K_m \text{ — треугольники из } \Omega_1^h \setminus Q_1^h, \Omega_1^h = (\Omega \setminus \Omega') \setminus \Omega_2^h, N_3 \text{ — число}$$

треугольников во всей подобласти Ω_1^h , а $\Omega_2^h = \bigcup_{m=N_3+1}^N K_m$, где $K_m \in T_h \cap \Omega_2$, N — общее число треугольников в сеточной области Ω^h .

Ввиду того, что $u_I|_{\Omega'} = 0$, имеем

$$\begin{aligned} \|u - u_I\|_{W_{2,\alpha}^1(\Omega)}^2 &= (\|u - u_I\|_{W_{2,\alpha}^1(\Omega_2^h)}^2 + \|u - u_I\|_{W_{2,\alpha}^1(\Omega_1^{1,h})}^2 + \\ &\quad + \|u - u_I\|_{W_{2,\alpha}^1(Q_1^h)}^2 + \|u\|_{W_{2,\alpha}^1(\Omega')}^2)^{1/2}. \end{aligned} \quad (15)$$

Теорема 1. Пусть $u \in \dot{W}_{2,\alpha-1}^2(\Omega)$, $u_I \in V^h$ — ее интерполянт, построенный по проведенной триангуляции T_h области Ω . Тогда

$$\lim_{h \rightarrow 0} \|u - u_I\|_{W_{2,\alpha}^1(\Omega)} = 0. \quad (16)$$

Для доказательства теоремы 1 нам понадобятся следующие вспомогательные утверждения.

Лемма 2. Если выполняются условия теоремы 1, то

$$\lim_{h \rightarrow 0} \|u - u_I\|_{W_{2,\alpha}^1(\Omega_2^h)} = 0. \quad (17)$$

Доказательство. Используя неравенства

$$\begin{aligned} |u - u_I|_{W_{2,\alpha}^1(\Omega_2^h)} &\leq \left(\sup_{x \in \Omega_2^h} \rho^{-\alpha}(x) \right) |u - u_I|_{W_2^1(\Omega_2^h)}, \\ |u|_{W_2^2(\Omega_2^h)} &\leq \frac{|u|_{W_{2,\alpha-1}^2(\Omega_2^h)}}{\inf_{x \in \Omega_2^h} \rho^{1-\alpha}(x)}, \end{aligned}$$

оценки [17]

$$|u - u_I|_{W_2^1(\Omega_2^h)} \leq C_9 h |u|_{W_2^2(\Omega_2^h)}, \quad |u - u_I|_{L_2(\Omega_2^h)} \leq C_{10} h^2 |u|_{W_2^2(\Omega_2^h)}$$

и полагая, что $\sup_{x \in \Omega_2^h} \rho^{-\alpha}(x) \leq C_{11}$, $\inf_{x \in \Omega_2^h} \rho^{1-\alpha}(x) \leq C_{12}$, где C_{11}, C_{12} — постоянные величины, получаем

$$|u - u_I|_{W_{2,\alpha}^1(\Omega_2^h)} \leq C_{13} h |u|_{W_{2,\alpha-1}^2(\Omega_2^h)}, \quad (18)$$

$$\|u - u_I\|_{L_2(\Omega_2^h)} \leq C_{14} h^2 |u|_{W_{2,\alpha-1}^2(\Omega_2^h)}. \quad (19)$$

(Здесь и далее постоянные множители, стоящие перед нормами и полунормами функции u в пространствах Соболева, не зависят от нее и h — наибольшей из длин сторон треугольников в сеточной области).

Из неравенств (18), (19) следует оценка

$$\|u - u_I\|_{W_{2,\alpha}^1(\Omega_2^h)} \leq C_{15} h |u|_{W_{2,\alpha-1}^2(\Omega_2^h)}.$$

Перейдя к пределу при $h \rightarrow 0$, в результате будем иметь равенство (17). Лемма доказана. \square

Лемма 3. Если выполняются условия теоремы 1, то

$$\lim_{h \rightarrow 0} \|u - u_I\|_{W_{2,\alpha}^1(\Omega_1^{1,h})} = 0. \quad (20)$$

Доказательство. Так как для всех треугольников K из $\Omega_1^{1,h}$ верны неравенства

$$|u - u_I|_{W_{2,\alpha}^1(K)} \leq \left(\sup_{x \in K} \rho^{-\alpha}(x) \right) |u - u_I|_{W_2^1(K)}, \quad (21)$$

$$|u|_{W_2^2(K)} \leq \frac{|u|_{W_{2,\alpha-1}^2(K)}}{\inf_{x \in K} \rho^{1-\alpha}(x)} \quad (22)$$

и справедливы оценки [17] при $p = q = 2$, $m = 0, 1$ для невесовых норм пространств Соболева и Лебега

$$|u - u_I|_{W_2^1(K)} \leq C_{16} h_K |u|_{W_2^2(K)}, \quad (23)$$

$$\|u - u_I\|_{L_2(K)} \leq C_{17} h_K^2 |u|_{W_2^2(K)} \quad (24)$$

(h_K — наибольшая длина стороны треугольника K), то из соотношений (21)–(24) следует

$$|u - u_I|_{W_{2,\alpha}^1(K)} \leq C_{16} h_K \frac{\sup_{x \in K} \rho^{-\alpha}(x)}{\inf_{x \in K} \rho^{1-\alpha}(x)} |u|_{W_{2,\alpha-1}^2(K)},$$

$$\|u - u_I\|_{L_2(K)} \leq C_{17} h_K^2 \frac{|u|_{W_{2,\alpha-1}^2(K)}}{\inf_{x \in K} \rho^{1-\alpha}(x)}.$$

Учитывая, что для предложенной триангуляции области $\Omega_1^{1,h}$

$$\inf_{x \in K} \rho^{1-\alpha}(x) \geq C_{18} \left(\frac{j-1}{n} \right)^{r(1-\alpha)}, \quad h_k \leq C_{19} \left(\frac{j}{n} \right)^{r-1} \frac{1}{n}, \quad r > 1,$$

$$\sup_{x \in K} \rho^{-\alpha}(x) \leq C_{20} \left(\frac{j}{n}\right)^{-r\alpha}, \quad -\frac{1}{2} < \alpha < 0,$$

$$\sup_{x \in K} \rho^{-\alpha}(x) \leq C_{21} \left(\frac{j-1}{n}\right)^{-r\alpha}, \quad 0 \leq \alpha < \frac{1}{2},$$

где $j = 2, \dots, n$ (n — число слоев разбиения подобласти Ω_1), получаем оценки

$$|u - u_I|_{W_{2,\alpha}^1(K)}^2 \leq C_{22} \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(K)}^2, \quad \alpha \in \left(-\frac{1}{2}; 0\right),$$

$$|u - u_I|_{W_{2,\alpha}^1(K)}^2 \leq C_{23} \left(\frac{j}{j-1}\right)^{2r} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(K)}^2, \quad \alpha \in \left[0; \frac{1}{2}\right),$$

$$\|u - u_I\|_{L_2(K)}^2 \leq C_{24} \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \left(\frac{j}{n}\right)^{2r(1+\alpha)} \frac{1}{j^4} \|u\|_{W_{2,\alpha-1}^2(K)}^2, \quad \alpha \in \left(-\frac{1}{2}; \frac{1}{2}\right). \quad (25)$$

В силу того, что $\|u\|_{W_{2,\alpha-1}^2(\Omega)} \leq C_{25}$, для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что $\|u\|_{W_{2,\alpha-1}^2(\Gamma_0^\delta)}^2 < \frac{\varepsilon^2}{4C_{26}^2}$, где $\Gamma_0^\delta = \{x \in \bar{\Omega} : \rho(x) \leq \delta\}$ есть δ -окрестность границы Γ_0 в области $\bar{\Omega}$ (полоса ширины δ).

Обозначим через n_ε число слоев Q_j , $j = 1, \dots, n_\varepsilon$, содержащихся в Γ_0^δ . (Ясно, что $n_\varepsilon \rightarrow \infty$ при $\varepsilon \rightarrow 0$). Таким образом,

$$\|u\|_{W_{2,\alpha-1}^2\left(\bigcup_{j=1}^{n_\varepsilon} Q_j\right)}^2 < \frac{\varepsilon^2}{4C_{26}^2}. \quad (26)$$

Задав ε , определим δ , а следовательно, и n_ε так, чтобы выполнялось неравенство (26). Покажем, что в этом случае имеет место оценка

$$\|u - u_I\|_{W_{2,\alpha}^1\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 < \frac{\varepsilon^2}{2}, \quad (27)$$

где $\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h \subset \bigcup_{j=1}^{n_\varepsilon} Q_j$.

Просуммировав неравенства (25) по всем треугольникам K из $\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h$ по слоям Q_j^h , будем иметь

$$\begin{aligned} \sum_{K \in \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} |u - u_I|_{W_{2,\alpha}^1(K)}^2 &\leq C_{22} \sum_{j=2}^{n_\varepsilon-1} \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{22} \left(2^{2r(1-\alpha)} \frac{1}{4} \|u\|_{W_{2,\alpha-1}^2(Q_2^h)}^2 + \dots + \left(1 + \frac{1}{n_\varepsilon - 2}\right)^{2r(1-\alpha)} \frac{1}{(n_\varepsilon - 1)^2} \|u\|_{W_{2,\alpha-1}^2(Q_{n_\varepsilon-1}^h)}^2\right) \leq \\ &\leq C_{22} 2^{2r(1-\alpha)-2} (\|u\|_{W_{2,\alpha-1}^2(Q_2^h)}^2 + \dots + \|u\|_{W_{2,\alpha-1}^2(Q_{n_\varepsilon-1}^h)}^2) \leq \\ &\leq C_{27} \|u\|_{W_{2,\alpha-1}^2\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 \text{ при } -\frac{1}{2} < \alpha < 0, \quad (28) \end{aligned}$$

$$\begin{aligned} \sum_{K \in \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} |u - u_I|_{W_{2,\alpha}^1(K)}^2 &\leq C_{23} \sum_{j=2}^{n_\varepsilon-1} \left(\frac{j}{j-1}\right)^{2r} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{23} 2^{2r-2} (\|u\|_{W_{2,\alpha-1}^2(Q_2^h)}^2 + \dots + \|u\|_{W_{2,\alpha-1}^2(Q_{n_\varepsilon-1}^h)}^2) \leq \\ &\leq C_{28} \|u\|_{W_{2,\alpha-1}^2\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 \text{ при } 0 \leq \alpha < \frac{1}{2}, \end{aligned} \quad (29)$$

$$\begin{aligned} \sum_{K \in \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} \|u - u_I\|_{L_2(K)}^2 &\leq C_{24} \sum_{j=2}^{n_\varepsilon-1} \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \left(\frac{j}{n}\right)^{2r(1+\alpha)} \frac{1}{j^4} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{24} 2^{2r(1-\alpha)-4} \left(\frac{n_\varepsilon-1}{n}\right)^{2r(1+\alpha)} (\|u\|_{W_{2,\alpha-1}^2(Q_2^h)}^2 + \dots + \|u\|_{W_{2,\alpha-1}^2(Q_{n_\varepsilon-1}^h)}^2) \leq \\ &\leq C_{29} \|u\|_{W_{2,\alpha-1}^2\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 \text{ при } -\frac{1}{2} < \alpha < \frac{1}{2}. \end{aligned} \quad (30)$$

Принимая во внимание, что

$$\|u - u_I\|_{W_{2,\alpha}^1\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 \leq 2 \left(\|u - u_I\|_{L_2\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 + |u - u_I|_{W_{2,\alpha}^1\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 \right),$$

и используя неравенства (26), (28)–(30), устанавливаем оценку (27)

$$\|u - u_I\|_{W_{2,\alpha}^1\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 < \frac{C_{26}\varepsilon^2}{2C_{26}} = \frac{\varepsilon^2}{2}, \quad \alpha \in \left(-\frac{1}{2}; \frac{1}{2}\right).$$

Аналогично просуммируем неравенства (25) по треугольникам K из $\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h$ по слоям Q_j^h , $j = n_\varepsilon, \dots, n$, в результате будем иметь

$$\begin{aligned} \sum_{K \in \Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} |u - u_I|_{W_{2,\alpha}^1(K)}^2 &\leq C_{22} \sum_{j=n_\varepsilon}^n \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{30} \left(1 + \frac{1}{n_\varepsilon - 1}\right)^{2r(1-\alpha)} \frac{1}{n_\varepsilon^2} \|u\|_{W_{2,\alpha-1}^2\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2, \text{ если } -\frac{1}{2} < \alpha < 0, \end{aligned} \quad (31)$$

$$\begin{aligned} \sum_{K \in \Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} |u - u_I|_{W_{2,\alpha}^2(K)}^2 &\leq C_{23} \sum_{j=n_\varepsilon}^n \left(\frac{j}{j-1}\right)^{2r} \frac{1}{j^2} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{31} \left(1 + \frac{1}{n_\varepsilon - 1}\right)^{2r} \frac{1}{n_\varepsilon^2} \|u\|_{W_{2,\alpha-1}^2\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2, \text{ если } 0 \leq \alpha < \frac{1}{2}, \end{aligned} \quad (32)$$

$$\begin{aligned} \sum_{K \in \Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h} \|u - u_I\|_{L_2(K)}^2 &\leq C_{24} \sum_{j=n_\varepsilon}^n \left(\frac{j}{j-1}\right)^{2r(1-\alpha)} \left(\frac{j}{n}\right)^{2r(1+\alpha)} \frac{1}{j^4} \|u\|_{W_{2,\alpha-1}^2(Q_j^h)}^2 \leq \\ &\leq C_{32} \left(1 + \frac{1}{n_\varepsilon - 1}\right)^{2r(1-\alpha)} \left(\frac{n_\varepsilon}{n}\right)^{2r(1+\alpha)} \frac{1}{n_\varepsilon^4} \|u\|_{W_{2,\alpha-1}^2\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2, \text{ если } -\frac{1}{2} < \alpha < \frac{1}{2}. \end{aligned} \quad (33)$$

Так как при $n \rightarrow \infty$ и $n_\varepsilon \rightarrow \infty$, а $h \rightarrow 0$, то из оценок (31)–(33) получаем

$$\lim_{h \rightarrow 0} \|u - u_I\|_{W_{2,\alpha}^1\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 = 0.$$

Значит, для любого $\varepsilon > 0$ существует такое $h_0(\varepsilon)$, что неравенство

$$\|u - u_I\|_{W_{2,\alpha}^1\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 < \frac{\varepsilon^2}{2} \quad (34)$$

справедливо для всех $h \leq h_0(\varepsilon)$, $-1/2 < \alpha < 1/2$. Ввиду того, что

$$\|u - u_I\|_{W_{2,\alpha}^1(\Omega_1^{1,h})}^2 = \|u - u_I\|_{W_{2,\alpha}^1\left(\Omega_1^{1,h} \setminus \bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2 + \|u - u_I\|_{W_{2,\alpha}^1\left(\bigcup_{j=2}^{n_\varepsilon-1} Q_j^h\right)}^2,$$

из (27), (34) следует предельное равенство (20). Лемма доказана. \square

Лемма 4. Пусть выполняются условия теоремы 1. Тогда

$$\lim_{h \rightarrow 0} \|u - u_I\|_{W_{2,\alpha}^1(Q_1^h)} = 0. \quad (35)$$

Доказательство. Чтобы установить справедливость равенства (35), воспользуемся неравенством

$$\|u - u_I\|_{W_{2,\alpha}^1(Q_1^h)}^2 \leq C_{33} \|u - u_I\|_{W_{2,\alpha}^{1,*}(Q_1^h)}^2 \leq 2C_{33} (\|u - u_I\|_{L_{2,\alpha}(Q_1^h)}^2 + |u - u_I|_{W_{2,\alpha}^1(Q_1^h)}^2). \quad (36)$$

Сначала оценим второе слагаемое, стоящее в правой части (36). Имеем

$$|u - u_I|_{W_{2,\alpha}^1(Q_1^h)}^2 \leq 2(|u|_{W_{2,\alpha}^1(Q_1^h)}^2 + |u_I|_{W_{2,\alpha}^1(Q_1^h)}^2). \quad (37)$$

Умножим полунорму $|u|_{W_{2,\alpha}^1(K)}$, где K — произвольный треугольник из Q_1^h , на $h^{(1)} = \max_{x \in K \subset Q_1^h} \rho(x)$. Выполним преобразование переноса начала координат O и поворота осей OX_1 и OX_2 исходной системы координат таким образом, чтобы ось $O'X'_1$ прошла через два узла треугольника K , принадлежащие кривой Γ_0 , а ось $O'X'_2$ — через тот узел K на Γ_0 , который лежит левее (в том случае, когда основанием треугольника K является звено ломаной Γ_0), или так, чтобы начало координат O' совпало с единственным узлом треугольника K на кривой Γ_0 , а ось $O'X'_1$ прошла через то звено ломаной Γ_0 , левой узловой точкой которого этот узел является (в том случае, когда основанием K является одно из звеньев ломаной Γ_1), получим новую, локальную систему координат $O'X'_1X'_2$ (рис. 2). (Заметим, что в результате проведенных преобразований координат рассматриваемые нами полунормы и нормы остаются неизменными).

Оценим $h^{(1)}|u_I|_{W_{2,\alpha}^1(K)}$ в системе координат $O'X'_1X'_2$

$$h^{(1)}|u_I|_{W_{2,\alpha}^1(K)} = h^{(1)} \left| \sum_{i=1}^3 u(P_i)\varphi_i \right|_{W_{2,\alpha}^1(K)} \leq \leq C_{34}|u(\bar{x}'_1, \bar{x}'_2)| \left(\int_K \rho^{-2\alpha}(x') dx'_1 dx'_2 \right)^{1/2} \leq C_{35}|u(\bar{x}'_1, \bar{x}'_2)|\rho^{1-\alpha}(\bar{x}'_1, \bar{x}'_2), \quad (38)$$

где (\bar{x}'_1, \bar{x}'_2) — координаты узла в K , в котором функции $u(x')$ и $\rho(x')$ принимают наибольшие значения среди узлов треугольника K . В точке $\bar{x}' = (\bar{x}'_1, \bar{x}'_2) \in K$ из Q_1^h имеем

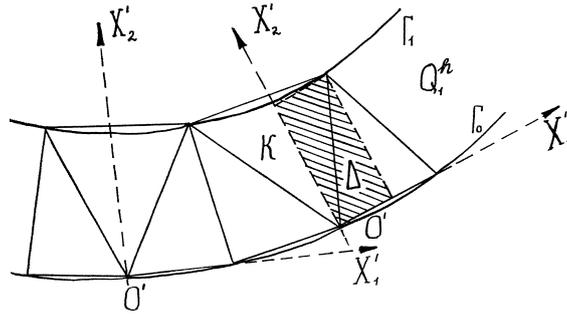


Рис. 2. Локальная система координат для треугольника K

$$|u(\bar{x}'_1, \bar{x}'_2)\rho^{1-\alpha}(\bar{x}'_1, \bar{x}'_2)| = \left| u(\bar{x}'_1, 0)\rho^{1-\alpha}(\bar{x}'_1, 0) + \int_0^{\bar{x}'_2} \frac{\partial(u(\bar{x}'_1, x'_2)\rho^{1-\alpha}(\bar{x}'_1, x'_2))}{\partial x'_2} dx'_2 \right|. \quad (39)$$

Получим оценки для слагаемых, стоящих в правой части равенства (39):

$$\begin{aligned} \left| \int_0^{\bar{x}'_2} \frac{\partial(u(\bar{x}'_1, x'_2)\rho^{1-\alpha}(\bar{x}'_1, x'_2))}{\partial x'_2} dx'_2 \right| &= \left| \int_0^{\bar{x}'_2} \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) dx'_2 + \right. \\ &\quad \left. + \int_0^{\bar{x}'_2} (1-\alpha)u(\bar{x}'_1, x'_2)\rho^{-\alpha}(\bar{x}'_1, x'_2) \frac{\partial \rho(\bar{x}'_1, x'_2)}{\partial x'_2} dx'_2 \right| \leq \\ &\leq \int_0^{\bar{x}'_2} \left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| dx'_2 + C_{36} \int_0^{\bar{x}'_2} |u(\bar{x}'_1, x'_2)\rho^{-\alpha}(\bar{x}'_1, x'_2)| dx'_2, \quad (40) \end{aligned}$$

$$\begin{aligned} |u(\bar{x}'_1, 0)\rho^{1-\alpha}(\bar{x}'_1, 0)| &= \left| u(0, 0)\rho^{1-\alpha}(0, 0) + \int_0^{\bar{x}'_1} \frac{\partial(u(x'_1, 0)\rho^{1-\alpha}(x'_1, 0))}{\partial x'_1} dx'_1 \right| \leq \\ &\leq \int_0^{\bar{x}'_1} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| dx'_1 + \int_0^{\bar{x}'_1} \left| (1-\alpha)u(x'_1, 0)\rho^{-\alpha}(x'_1, 0) \frac{\partial \rho(x'_1, 0)}{\partial x'_1} \right| dx'_1 \leq \\ &\leq \int_0^{\bar{x}'_1} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| dx'_1 + C_{37} \int_0^{\bar{x}'_1} |u(x'_1, 0)\rho^{-\alpha}(x'_1, 0)| dx'_1. \quad (41) \end{aligned}$$

Оценим модули подынтегральных функций интегралов (40), (41), получим

$$\left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| \leq \left| \frac{\partial u(x'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(x'_1, x'_2) \right| + \int_{x'_1}^{\bar{x}'_2} \left| \frac{\partial}{\partial t} \left(\frac{\partial u(t, x'_2)}{\partial x'_2} \rho^{1-\alpha}(t, x'_2) \right) \right| dt, \quad (42)$$

$$|u(\bar{x}'_1, x'_2) \rho^{-\alpha}(\bar{x}'_1, x'_2)| \leq |u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)| + \int_{x'_1}^{\bar{x}'_1} \left| \frac{\partial}{\partial t} (u(t, x'_2) \rho^{-\alpha}(t, x'_2)) \right| dt, \quad (43)$$

если $0 \leq x'_1 \leq \bar{x}'_1$,

$$\left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| \leq \left| \frac{\partial u(x'_1, x'_2)}{\partial x'_1} \rho^{1-\alpha}(x'_1, x'_2) \right| + \int_{x'_2}^{\bar{x}'_2} \left| \frac{\partial}{\partial t} \left(\frac{\partial u(x'_1, t)}{\partial x'_1} \rho^{1-\alpha}(x'_1, t) \right) \right| dt, \quad (44)$$

$$|u(x'_1, 0) \rho^{-\alpha}(x'_1, 0)| \leq |u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)| + \int_{x'_2}^{\bar{x}'_2} \left| \frac{\partial}{\partial t} (u(x'_1, t) \rho^{-\alpha}(x'_1, t)) \right| dt, \quad (45)$$

если $0 \leq x'_2 \leq \bar{x}'_2$.

Проинтегрируем неравенства (42), (43) по $x'_1 \in (0, \bar{x}'_1)$, а неравенства (44), (45) по $x'_2 \in (0, \bar{x}'_2)$, будем иметь

$$\int_0^{\bar{x}'_1} \left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| dx'_1 \leq \int_0^{\bar{x}'_1} \left| \frac{\partial u(x'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(x'_1, x'_2) \right| dx'_1 + \bar{x}'_1 \int_0^{\bar{x}'_1} \left| \frac{\partial}{\partial x'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(x'_1, x'_2) \right) \right| dx'_1, \quad (46)$$

$$\int_0^{\bar{x}'_1} |u(\bar{x}'_1, x'_2) \rho^{-\alpha}(\bar{x}'_1, x'_2)| dx'_1 \leq \int_0^{\bar{x}'_1} |u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)| dx'_1 + \bar{x}'_1 \int_0^{\bar{x}'_1} \left| \frac{\partial}{\partial x'_1} (u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)) \right| dx'_1, \quad (47)$$

$$\int_0^{\bar{x}'_2} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| dx'_2 \leq \int_0^{\bar{x}'_2} \left| \frac{\partial u(x'_1, x'_2)}{\partial x'_1} \rho^{1-\alpha}(x'_1, x'_2) \right| dx'_2 + \bar{x}'_2 \int_0^{\bar{x}'_2} \left| \frac{\partial}{\partial x'_2} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \rho^{1-\alpha}(x'_1, x'_2) \right) \right| dx'_2, \quad (48)$$

$$\int_0^{\bar{x}'_2} |u(x'_1, 0) \rho^{-\alpha}(x'_1, 0)| dx'_2 \leq \int_0^{\bar{x}'_2} |u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)| dx'_2 + \bar{x}'_2 \int_0^{\bar{x}'_2} \left| \frac{\partial}{\partial x'_2} (u(x'_1, x'_2) \rho^{-\alpha}(x'_1, x'_2)) \right| dx'_2. \quad (49)$$

Вычислив интегралы в левых частях неравенств (46)–(49) и разделив обе части неравенств (46), (47) на \bar{x}'_1 , а неравенств (48), (49) на \bar{x}'_2 , используем неравенство Коши–Буняковского для интегралов, стоящих в правых частях полученных неравенств, и в результате имеем

$$\begin{aligned} \left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| &\leq \frac{1}{\bar{x}'_1} \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_1} \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_2 \partial x'_1} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 (1-\alpha)^2 \rho^{-2\alpha}(x'_1, x'_2) \left(\frac{\partial \rho(x'_1, x'_2)}{\partial x'_1} \right)^2 dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2}, \quad (50) \end{aligned}$$

$$\begin{aligned} |u(\bar{x}'_1, x'_2) \rho^{-\alpha}(\bar{x}'_1, x'_2)| &\leq \frac{1}{\bar{x}'_1} \left(\int_0^{\bar{x}'_1} u^2(x'_1, x'_2) \rho^{-2\alpha}(x'_1, x'_2) dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_1} (-\alpha)^2 u^2(x'_1, x'_2) \rho^{-2(\alpha+1)}(x'_1, x'_2) \left(\frac{\partial \rho(x'_1, x'_2)}{\partial x'_1} \right)^2 dx'_1 \right)^{1/2} \left(\int_0^{\bar{x}'_1} dx'_1 \right)^{1/2}, \quad (51) \end{aligned}$$

$$\begin{aligned} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| &\leq \frac{1}{\bar{x}'_2} \int_0^{\bar{x}'_2} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_2 \left(\int_0^{\bar{x}'_2} dx'_2 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_2} \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_1 \partial x'_2} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_2 \right)^{1/2} \left(\int_0^{\bar{x}'_2} dx'_2 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_2} (1-\alpha)^2 \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) \left(\frac{\partial \rho(x'_1, x'_2)}{\partial x'_2} \right)^2 dx'_2 \right)^{1/2} \left(\int_0^{\bar{x}'_2} dx'_2 \right)^{1/2}, \quad (52) \end{aligned}$$

$$\begin{aligned} |u(x'_1, 0) \rho^{-\alpha}(x'_1, 0)| &\leq \frac{1}{\bar{x}'_2} \left(\int_0^{\bar{x}'_2} u^2(x'_1, x'_2) \rho^{-2\alpha}(x'_1, x'_2) dx'_2 \right)^{1/2} \left(\int_0^{\bar{x}'_2} dx'_2 \right)^{1/2} + \\ &+ \left(\int_0^{\bar{x}'_2} (-\alpha)^2 u^2(x'_1, x'_2) \rho^{-2(\alpha+1)}(x'_1, x'_2) \left(\frac{\partial \rho(x'_1, x'_2)}{\partial x'_2} \right)^2 dx'_2 \right)^{1/2} \left(\int_0^{\bar{x}'_2} dx'_2 \right)^{1/2}. \quad (53) \end{aligned}$$

Учитывая ограниченность частных производных функции $\rho(x')$, запишем оценки (50)–(53) в следующем виде:

$$\begin{aligned} \left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| &\leq \left(\frac{1}{\bar{x}'_1} \right)^{1/2} \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_1 \right)^{1/2} + \\ &+ (\bar{x}'_1)^{1/2} \left(\int_0^{\bar{x}'_1} \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_2 \partial x'_1} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_1 \right)^{1/2} + \\ &+ C_{38} (\bar{x}'_1)^{1/2} \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) dx'_1 \right)^{1/2}, \quad (54) \end{aligned}$$

$$\begin{aligned} |u(\bar{x}'_1, x'_2) \rho^{-\alpha}(\bar{x}'_1, x'_2)| &\leq \left(\frac{1}{\bar{x}'_1} \right)^{1/2} \left(\int_0^{\bar{x}'_1} u^2(x'_1, x'_2) \rho^{-2\alpha}(x'_1, x'_2) dx'_1 \right)^{1/2} + \\ &+ (\bar{x}'_1)^{1/2} \left(\int_0^{\bar{x}'_1} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) dx'_1 \right)^{1/2} + \\ &+ C_{39} (\bar{x}'_1)^{1/2} \left(\int_0^{\bar{x}'_1} u^2(x'_1, x'_2) \rho^{-2(\alpha+1)}(x'_1, x'_2) dx'_1 \right)^{1/2}, \quad (55) \end{aligned}$$

$$\begin{aligned} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| &\leq \left(\frac{1}{\bar{x}'_2} \right)^{1/2} \int_0^{\bar{x}'_2} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_2 \right)^{1/2} + \\ &+ (\bar{x}'_2)^{1/2} \int_0^{\bar{x}'_2} \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_2 \partial x'_1} \right)^2 \rho^{2(1-\alpha)}(x'_1, x'_2) dx'_2 \right)^{1/2} + \\ &+ C_{40} (\bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) dx'_2 \right)^{1/2}, \quad (56) \end{aligned}$$

$$\begin{aligned} |u(x'_1, 0) \rho^{-\alpha}(x'_1, 0)| &\leq \left(\frac{1}{\bar{x}'_2} \right)^{1/2} \left(\int_0^{\bar{x}'_2} u^2(x'_1, x'_2) \rho^{-2\alpha}(x'_1, x'_2) dx'_2 \right)^{1/2} + \\ &+ (\bar{x}'_2)^{1/2} \int_0^{\bar{x}'_2} \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 \rho^{-2\alpha}(x'_1, x'_2) dx'_2 \right)^{1/2} + \\ &+ C_{41} (\bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} u^2(x'_1, x'_2) \rho^{-2(\alpha+1)}(x'_1, x'_2) dx'_2 \right)^{1/2}. \quad (57) \end{aligned}$$

Проинтегрируем неравенства (54), (55) по $x'_2 \in (0, \bar{x}'_2)$, а неравенства (56), (57) по $x'_1 \in (0, \bar{x}'_1)$ и, применив неравенство Коши—Буняковского для интегралов в правых частях полученных неравенств, с учетом того, что $C_{42} \bar{x}'_2 < \bar{x}'_1 \leq \bar{x}'_2 \leq h^{(1)}$, будем иметь следующие оценки:

$$\begin{aligned}
 \int_0^{\bar{x}'_2} \left| \frac{\partial u(\bar{x}'_1, x'_2)}{\partial x'_2} \rho^{1-\alpha}(\bar{x}'_1, x'_2) \right| dx'_2 &\leq \left(\frac{\bar{x}'_2}{\bar{x}'_1} \right)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{2(1-\alpha)}(x'_1, x'_2) \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\
 &+ (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{2(1-\alpha)}(x'_1, x'_2) \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_2 \partial x'_1} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\
 &+ C_{38} (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{-2\alpha}(x'_1, x'_2) \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 dx'_1 dx'_2 \right)^{1/2} \leq \\
 &\leq C_{43} \left(\max_{x' \in \Delta} \rho^2(x') \right)^{1/2} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + \\
 &+ h^{(1)} \left(\int_{\Delta} \rho^{2(1-\alpha)}(x') \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x')}{\partial x'_{s_1} \partial x'_{s_2}} \right)^2 dx' \right)^{1/2} + \\
 &+ C_{38} h^{(1)} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2}, \quad (58)
 \end{aligned}$$

$$\begin{aligned}
 \int_0^{\bar{x}'_2} |u(\bar{x}'_1, x'_2) \rho^{-\alpha}(\bar{x}'_1, x'_2)| dx'_2 &\leq \left(\frac{\bar{x}'_2}{\bar{x}'_1} \right)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{-2\alpha}(x'_1, x'_2) u^2(x'_1, x'_2) dx'_1 dx'_2 \right)^{1/2} + \\
 &+ (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{-2\alpha}(x'_1, x'_2) \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\
 &+ C_{39} (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_2} \int_0^{\bar{x}'_1} \rho^{-2(\alpha+1)}(x'_1, x'_2) u^2(x'_1, x'_2) dx'_1 dx'_2 \right)^{1/2} \leq \\
 &\leq C_{44} \left(\max_{x' \in \Delta} \rho^2(x') \right)^{1/2} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2} + \\
 &+ h^{(1)} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + C_{39} h^{(1)} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2}, \quad (59)
 \end{aligned}$$

$$\begin{aligned}
 \int_0^{\bar{x}'_1} \left| \frac{\partial u(x'_1, 0)}{\partial x'_1} \rho^{1-\alpha}(x'_1, 0) \right| dx'_1 &\leq \left(\frac{\bar{x}'_1}{\bar{x}'_2} \right)^{1/2} \left(\int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{2(1-\alpha)}(x'_1, x'_2) \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\
 &+ (\bar{x}'_1 \bar{x}'_2)^{1/2} \int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{2(1-\alpha)}(x'_1, x'_2) \left(\frac{\partial^2 u(x'_1, x'_2)}{\partial x'_1 \partial x'_2} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\
 &+ C_{40} (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{-2\alpha}(x'_1, x'_2) \frac{\partial u(x'_1, x'_2)}{\partial x'_1} \right)^2 dx'_1 dx'_2 \right)^{1/2} \leq
 \end{aligned}$$

$$\begin{aligned} &\leq \left(\max_{x' \in \Delta} \rho^2(x') \right)^{1/2} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + \\ &\quad + h^{(1)} \left(\int_{\Delta} \rho^{2(1-\alpha)}(x') \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x')}{\partial x'_1{}^{s_1} \partial x'_2{}^{s_2}} \right)^2 dx' \right)^{1/2} + \\ &\quad + C_{40} h^{(1)} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2}, \quad (60) \end{aligned}$$

$$\begin{aligned} \int_0^{\bar{x}'_1} |u(x'_1, 0) \rho^{-\alpha}(x'_1, 0)| dx'_1 &\leq \left(\frac{\bar{x}'_1}{\bar{x}'_2} \right)^{1/2} \left(\int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{-2\alpha}(x'_1, x'_2) u^2(x'_1, x'_2) dx'_1 dx'_2 \right)^{1/2} + \\ &\quad + (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{-2\alpha}(x'_1, x'_2) \left(\frac{\partial u(x'_1, x'_2)}{\partial x'_2} \right)^2 dx'_1 dx'_2 \right)^{1/2} + \\ &\quad + C_{41} (\bar{x}'_1 \bar{x}'_2)^{1/2} \left(\int_0^{\bar{x}'_1} \int_0^{\bar{x}'_2} \rho^{-2(1+\alpha)}(x'_1, x'_2) u^2(x'_1, x'_2) dx'_1 dx'_2 \right)^{1/2} \leq \\ &\leq \left(\max_{x' \in \Delta} \rho^2(x') \right)^{1/2} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2} + \\ &\quad + h^{(1)} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + C_{41} h^{(1)} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2}, \quad (61) \end{aligned}$$

где Δ — прямоугольник из Q_1^h (рис. 2).

Принимая во внимание, что $\left(\max_{x' \in \Delta} \rho^2(x') \right)^{1/2} = h^{(1)}$, из оценок (38)–(41) и (58)–(61) устанавливаем

$$\begin{aligned} h^{(1)} |u_I|_{W_{2,\alpha}^1(K)} &\leq h^{(1)} \left(C_{45} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + \right. \\ &\left. + C_{46} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2} + C_{47} \left(\int_{\Delta} \rho^{-2(\alpha-1)}(x') \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x')}{\partial x'_1{}^{s_1} \partial x'_2{}^{s_2}} \right)^2 dx' \right)^{1/2} \right). \end{aligned}$$

Перейдя к системе координат OX_1X_2 , возведем обе части полученного неравенства в квадрат, будем иметь

$$\begin{aligned} |u_I|_{W_{2,\alpha}^1(K)}^2 &\leq 3 \int_{\Delta} \left(C_{45}^2 \rho^{-2\alpha}(x) \sum_{s=1}^2 \left(\frac{\partial u(x)}{\partial x_s} \right)^2 + \right. \\ &\quad \left. + C_{46}^2 \rho^{-2(1+\alpha)}(x) u^2(x) + C_{47}^2 \rho^{-2(\alpha-1)}(x) \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x)}{\partial x_1^{s_1} \partial x_2^{s_2}} \right)^2 \right) dx. \end{aligned}$$

Просуммировав неравенства по всем K и Δ из Q_1^h , запишем

$$|u_I|_{W_{2,\alpha}^1(Q_1^h)}^2 \leq C_{48} \|u\|_{W_{2,\alpha-1}^2(Q_1^h)}^2 + C_{49} \|u\|_{L_{2,1+\alpha}(Q_1^h)}^2 + C_{50} \|u\|_{W_{2,\alpha}^1(Q_1^h)}^2. \quad (62)$$

Тогда из неравенств (37), (62) следует оценка

$$|u - u_I|_{W_{2,\alpha}^1(Q_1^h)}^2 \leq 2(C_{48}\|u\|_{W_{2,\alpha-1}^2(Q_1^h)}^2 + C_{49}\|u\|_{L_{2,1+\alpha}(Q_1^h)}^2 + C_{51}\|u\|_{W_{2,\alpha}^1(Q_1^h)}^2). \quad (63)$$

Так как $\|u\|_{W_{2,\alpha-1}^2(\Omega)}^2 \leq C_{25}$, $\|u\|_{L_{2,\alpha}(\Omega)}^2 \leq C_{52}$, $\|u\|_{L_{2,1+\alpha}(\Omega)}^2 \leq C_{53}$, $\|u\|_{W_{2,\alpha}^1(\Omega)}^2 \leq C_{54}$, то для произвольного $\varepsilon > 0$ найдутся такие положительные числа $\delta_1, \dots, \delta_4$, что $\|u\|_{L_{2,\alpha}(\Gamma_0^{\delta_1})}^2 < \frac{\varepsilon^2}{16C_{33}}$, $\|u\|_{W_{2,\alpha-1}^2(\Gamma_0^{\delta_2})}^2 < \frac{\varepsilon^2}{24C_{48}}$, $\|u\|_{L_{2,1+\alpha}(\Gamma_0^{\delta_3})}^2 < \frac{\varepsilon^2}{24C_{49}}$, $\|u\|_{W_{2,\alpha}^1(\Gamma_0^{\delta_4})}^2 < \frac{\varepsilon^2}{24C_{51}}$, где $\Gamma_0^{\delta_i} = \{x \in \bar{\Omega} : \rho(x) \leq \delta_i, i = 1, \dots, 4\}$ есть δ_i -окрестность границы Γ_0 в области $\bar{\Omega}$. Задав ε , определим $\delta = \min_i \delta_i$ и n_ε — число слоев Q_j , $j = 1, \dots, n_\varepsilon$, содержащихся в Γ_0^δ ($\Gamma_0^\delta = \bigcup_{j=1}^{n_\varepsilon} Q_j$) так, чтобы выполнялись неравенства

$$\begin{aligned} \|u\|_{L_{2,\alpha}\left(\bigcup_{j=1}^{n_\varepsilon} Q_j\right)}^2 &< \frac{\varepsilon^2}{16C_{33}}, \quad \|u\|_{W_{2,\alpha-1}^2\left(\bigcup_{j=1}^{n_\varepsilon} Q_j\right)}^2 < \frac{\varepsilon^2}{24C_{48}}, \quad \|u\|_{L_{2,1+\alpha}\left(\bigcup_{j=1}^{n_\varepsilon} Q_j\right)}^2 < \frac{\varepsilon^2}{24C_{49}}, \\ \|u\|_{W_{2,\alpha}^1\left(\bigcup_{j=1}^{n_\varepsilon} Q_j\right)}^2 &< \frac{\varepsilon^2}{24C_{51}}. \end{aligned} \quad (64)$$

Вследствие того, что $Q_1^h \subset \bigcup_{j=1}^{n_\varepsilon} Q_j$, имеем

$$\|u\|_{L_{2,\alpha}(Q_1^h)}^2 < \frac{\varepsilon^2}{16C_{33}}, \quad (65)$$

$$\|u\|_{W_{2,\alpha-1}^2(Q_1^h)}^2 < \frac{\varepsilon^2}{24C_{48}}, \quad \|u\|_{L_{2,1+\alpha}(Q_1^h)}^2 < \frac{\varepsilon^2}{24C_{49}}, \quad \|u\|_{W_{2,\alpha}^1(Q_1^h)}^2 < \frac{\varepsilon^2}{24C_{51}}. \quad (66)$$

Тогда из неравенств (63), (66) следует

$$|u - u_I|_{W_{2,\alpha}^1(Q_1^h)}^2 < \frac{\varepsilon^2}{4C_{33}}. \quad (67)$$

Теперь перейдем к оценке первого слагаемого правой части неравенства (36). Воспользуемся тем, что

$$\|u - u_I\|_{L_{2,\alpha}(Q_1^h)}^2 \leq 2(\|u\|_{L_{2,\alpha}(Q_1^h)}^2 + \|u_I\|_{L_{2,\alpha}(Q_1^h)}^2). \quad (68)$$

Оценим $\|u_I\|_{L_{2,\alpha}(Q_1^h)}^2$. Для любого K из Q_1^h в локальной системе координат $O'X'_1X'_2$ (рис. 2) имеем

$$\|u_I\|_{L_{2,\alpha}(K)} = \left\| \sum_{i=1}^3 u(P_i)\varphi_i \right\|_{L_{2,\alpha}(K)} \leq C_{55}|u(\bar{x}')| \left(\int_K \rho^{-2\alpha}(x') dx' \right)^{1/2} \leq C_{56}|u(\bar{x}')|\rho^{1-\alpha}(\bar{x}').$$

(Здесь $u(\bar{x}') = \max_i u(P_i)$, $\rho(\bar{x}') = \max_i \rho(P_i)$, P_i — узлы треугольника K , $i = 1, 2, 3$). Используя неравенства (39)–(41) и (58)–(61), получаем

$$\begin{aligned} \|u_I\|_{L_{2,\alpha}(K)} &\leq h^{(1)} \left(C_{57} \left(\int_{\Delta} \rho^{-2\alpha}(x') \sum_{s=1}^2 \left(\frac{\partial u(x')}{\partial x'_s} \right)^2 dx' \right)^{1/2} + \right. \\ &+ C_{58} \left(\int_{\Delta} \rho^{-2(1+\alpha)}(x') u^2(x') dx' \right)^{1/2} + C_{59} \left(\int_{\Delta} \rho^{-2(\alpha-1)}(x') \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x')}{\partial x'_1 s_1 \partial x'_2 s_2} \right)^2 dx' \right)^{1/2} \Big). \end{aligned}$$

Перейдем к системе координат OX_1X_2 и, возведя обе части неравенства в квадрат, запишем оценку

$$\begin{aligned} \|u_I\|_{L_{2,\alpha}(K)}^2 &\leq 3(h^{(1)})^2 \int_{\Delta} \left(C_{57}^2 \rho^{-2\alpha}(x) \sum_{s=1}^2 \left(\frac{\partial u(x)}{\partial x_s} \right)^2 + \right. \\ &\quad \left. + C_{58}^2 \rho^{-2(1+\alpha)}(x) u^2(x) + C_{59}^2 \rho^{-2(\alpha-1)}(x) \sum_{s_1+s_2=2} \left(\frac{\partial^2 u(x)}{\partial x_1^{s_1} \partial x_2^{s_2}} \right)^2 \right) dx. \end{aligned}$$

Просуммировав неравенства по всем K и Δ из Q_1^h , с учетом того, что $h^{(1)} \leq h$, будем иметь

$$\|u_I\|_{L_{2,\alpha}(Q_1^h)}^2 \leq h^2 (C_{60} \|u\|_{W_{2,\alpha}^1(Q_1^h)}^2 + C_{61} \|u\|_{W_{2,\alpha-1}^2(Q_1^h)}^2 + C_{62} \|u\|_{L_{2,1+\alpha}(Q_1^h)}^2).$$

Отсюда

$$\lim_{h \rightarrow 0} \|u_I\|_{L_{2,\alpha}(Q_1^h)}^2 = 0.$$

Следовательно, для любого $\varepsilon > 0$ существует такое $h_0(\varepsilon)$, что неравенство

$$\|u_I\|_{L_{2,\alpha}(Q_1^h)}^2 \leq \frac{\varepsilon^2}{16C_{33}} \tag{69}$$

верно для всех $h \leq h_0(\varepsilon)$. Из неравенств (65), (68), (69) получим

$$\|u - u_I\|_{L_{2,\alpha}(Q_1^h)}^2 < \frac{\varepsilon^2}{4C_{33}}. \tag{70}$$

Теперь, используя неравенство (36) и оценки (67), (70), установим предельное равенство (35). Лемма доказана. \square

Замечание 1. Так как справедливо неравенство (64), то приняв во внимание, что $\Omega' \subset \bigcup_{j=1}^{n_\varepsilon} Q_j$, будем иметь

$$\|u\|_{W_{2,\alpha}^1(\Omega')} < \varepsilon^2.$$

Доказательство теоремы 1. Перейдем к пределу при $h \rightarrow 0$ в равенстве (15). Учитывая соотношения (17), (20), (35) и последнюю оценку, получаем требуемое утверждение (16). Теорема 1 доказана. \square

Теперь установим основной результат работы.

Теорема 2. Пусть коэффициенты $a_{kl}(x) = a_{lk}(x)$, $(k, l = 1, 2)$ и $a(x) > 0$ удовлетворяют неравенствам (8)–(11), выполнено условие (7). Тогда приближенное решение u_h задачи (13) сходится при $h \rightarrow 0$ в пространстве $\mathring{W}_{2,\alpha}^1(\Omega)$ к решению u задачи (12).

Доказательство. Чтобы установить справедливость соотношения (14), совершим предельный переход при $h \rightarrow 0$ в неравенствах

$$\|u - u_h\|_{W_{2,\alpha}^1(\Omega)} \leq C_8 \inf_{v_h \in V^h} \|u - v_h\|_{W_{2,\alpha}^1(\Omega)} \leq C_8 \|u - u_I\|_{W_{2,\alpha}^1(\Omega)}$$

и применим к ним (16). Теорема 2 доказана. \square

Заключение

В настоящей работе построен метод конечных элементов для решения задачи Дирихле для эллиптического уравнения второго порядка с вырождением на всей дважды непрерывно дифференцируемой границе двумерной области Ω . Доказано, что решение МКЭ сходится в весовом пространстве $\dot{W}_{2,\alpha}^1(\Omega)$ к обобщенному решению из пространства $\dot{W}_{2,\alpha-1}^2(\Omega) \subset \dot{W}_{2,\alpha}^1(\Omega)$. Разработанные и исследованные схемы метода конечных элементов с сетками, сгущающимися к границе области, могут быть использованы для решения задач гидродинамики, электромагнетизма, диффузии, теории пластичности и др., приводящих к краевым задачам для эллиптических уравнений с вырождением на границе. В дальнейшем, благодаря сужению пространства $\dot{W}_{2,\alpha-1}^2(\Omega)$, которому принадлежит обобщенное решение поставленной задачи (см. [14]), планируется установить скорость сходимости приближенного решения к точному по норме весового пространства Соболева.

Литература

1. Рукавишников В.А., Рукавишникова Е.И. Метод конечных элементов для первой краевой задачи с согласованным вырождением исходных данных // Доклады Академии Наук. 1994. Т. 338, № 6. С. 731–733.
2. Assous F., Ciarlet P. Jr., Segré J. Numerical Solution of the Time-Dependent Maxwell Equations in Two-Dimensional Singular Domain: The Singular Complement Method // Journal of Computational Physics. 2000. Vol. 161. P. 218–249. DOI: 10.1006/jcph.2000.6499.
3. Costabel M., Dauge M., Schwab C. Exponential Convergence of hp-FEM for Maxwell's Equations with Weighted Regularization in Polygonal Domains // Mathematical Models and Methods in Applied Sciences. 2005. Vol. 15, No. 4. P. 575–622. DOI: 10.1142/S0218202505000480.
4. Arroyo D., Bespalov A., Heuer N. On the Finite Element Method for Elliptic Problems with Degenerate and Singular Coefficients // Mathematics of Computation. 2007. Vol. 76, No. 258. P. 509–537. DOI: 10.1090/S0025-5718-06-01910-7.
5. Li H., Nistor V. Analysis of a Modified Schrödinger Operator in 2D: Regularity, Index, and FEM // Journal of Computational and Applied Mathematics. 2009. Vol. 224, No. 1. P. 320–338. DOI: 10.1016/j.cam.2008.05.009.
6. Рукавишников В.А., Кузнецова Е.В. Схема метода конечных элементов для краевой задачи с несогласованным вырождением исходных данных // Сибирский журнал вычислительной математики. 2009. Т. 12, № 3. С. 313–324.
7. Rukavishnikov V.A., Rukavishnikova H.I. The Finite Element Method for a Boundary Value Problem with Strong Singularity // Journal of Computational and Applied Mathematics. 2010. Vol. 234, No. 9. P. 2870–2882. DOI: 10.1016/j.cam.2010.01.020.

8. Rukavishnikov V.A., Mosolapov A.O. New Numerical Method for Solving Time-Harmonic Maxwell Equations with Strong Singularity // Journal of Computational Physics. 2012. Vol. 231, No. 6. P. 2438–2448. DOI: 10.1016/j.jcp.2011.11.031.
9. Rukavishnikov V.A., Rukavishnikova H.I. On the Error Estimation of the Finite Element Method for the Boundary Value Problems with Singularity in the Lebesgue Weighted Space // Numerical Functional Analysis and Optimization. 2013. Vol. 34, No. 12. P. 1328–1347. DOI: 10.1080/01630563.2013.809582.
10. Rukavishnikov V.A., Rukavishnikova E.I. The Finite Element Method for Boundary Value Problems with Strong Singularity and Double Singularity // Lecture Notes in Computer Science. 2013. Vol. 8236. P. 110–121. DOI: 10.1007/978-3-642-41515-9_10.
11. Никольский С.М. Вариационная проблема для уравнения эллиптического типа с вырождением на границе // Труды Математического института имени В.А. Стеклова. 1979. Т. 150. С. 212–238.
12. Лизоркин П.И., Никольский С.М. Эллиптическое уравнение с вырождением. Вариационный метод // Доклады Академии Наук СССР. 1981. Т. 257, № 1. С. 42–45.
13. Лизоркин П.И., Никольский С.М. Эллиптические уравнения с вырождением. Дифференциальные свойства решений // Доклады Академии Наук СССР. 1981. Т. 257, № 2. С. 278–282.
14. Рукавишников В.А., Рукавишникова Е.И. Об изоморфном отображении весовых пространств эллиптическим оператором с вырождением на границе области // Дифференциальные уравнения. 2014. Т. 50, № 3. С. 349–355. DOI: 10.1134/S037406411403008X.
15. Никольский С.М. Приближение функций многих переменных и теоремы вложения. М.: Наука, 1977. 456 с.
16. Обэн Ж.-П. Приближенное решение эллиптических краевых задач, пер. с англ. М.: Мир, 1977. 383 с.
17. Сьярле Ф. Метод конечных элементов для эллиптических задач, пер. с англ. М.: Мир, 1980. 512 с.

Рукавишникова Елена Ивановна, к.ф.-м.н., доцент, с.н.с., лаборатория математического моделирования в физике и технике, Вычислительный центр ДВО РАН (Хабаровск, Российская Федерация)

CONVERGENCE OF THE FINITE ELEMENT METHOD FOR BOUNDARY VALUE PROBLEM WITH DEGENERATION ON THE WHOLE BOUNDARY OF DOMAIN

© 2019 E.I. Rukavishnikova

Computing center FEB RAS

(str. Kim Yu Chena 65, Khabarovsk, 680000 Russia)

E-mail: rukavishnikova-55@mail.ru

Received: 04.09.2018

In this paper we consider the Dirichlet problem with homogeneous boundary condition for a second-order elliptic equation with degeneration on the entire twice continuously differentiable boundary of two-dimensional domain Ω . We define a generalized solution of this problem, which exists and is unique in the weighted Sobolev space $\dot{W}_{2,\alpha}^1(\Omega)$. To solve the formulated problem a finite element method is developed, the scheme of which is constructed on the basis of the definition of a generalized solution of the original differential problem in the space $\dot{W}_{2,\alpha}^1(\Omega)$. For this purpose a two-dimensional convex domain is divided into triangles with special condensation to the boundary. Next we introduce a finite element space $V^h \subset \dot{W}_{2,\alpha}^1(\Omega)$ that contains continuous functions linear on each triangular element of grid region Ω^h and equal to zero on the set $\bar{\Omega} \setminus \Omega^h$, and show unique solvability of the scheme of the finite element method. For the generalized solution u from the subspace $\dot{W}_{2,\alpha-1}^2(\Omega)$ of the space $\dot{W}_{2,\alpha}^1(\Omega)$, using its values in the nodes of the triangulated domain, an interpolant $u_I \in V^h$ is constructed, and the fact of its convergence with respect to the norm $W_{2,\alpha}^1(\Omega)$ is established. The main result of the work for the proposed method for solving the first boundary value problem with degeneration is the proof of the convergence of the approximate solution to the exact solution in the weighted Sobolev space.

Keywords: boundary value problem with degeneration, Sobolev weighted space, generalized solution, finite element method.

FOR CITATION

Rukavishnikova E.I. Convergence of the Finite Element Method for Boundary Value Problem with Degeneration on Whole Boundary of the Domain. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. vol. 8, no. 3. pp. 5–26. (in Russian) DOI: 10.14529/cmse190301.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Rukavishnikov V.A., Rukavishnikova E.I. The Finite Element Method for the First Boundary Value Problem with Coordinated Degeneration of the Initial Data. *Russian Academy of Sciences. Doklady. Mathematics*. 1995. vol. 50, no. 2. pp. 335–339.
2. Assous F., Ciarlet P. Jr., Segré J. Numerical Solution of the Time-Dependent Maxwell Equations in Two-Dimensional Singular Domain: The Singular Complement Method. *Journal of Computational Physics*. 2000. vol. 161. pp. 218–249. DOI: 10.1006/jcph.2000.6499.
3. Costabel M., Dauge M., Schwab C. Exponential Convergence of hp-FEM for Maxwell's Equations with Weighted Regularization in Polygonal Domains. *Mathematical*

- Models and Methods in Applied Sciences*. 2005. vol. 15, no. 4. pp. 575–622. DOI: 10.1142/S0218202505000480.
4. Arroyo D., Beshpalov A., Heuer N. On the Finite Element Method for Elliptic Problems with Degenerate and Singular Coefficients. *Mathematics of Computation*. 2007. vol. 76, no. 258. pp. 509–537. DOI: 10.1090/S0025-5718-06-01910-7.
 5. Li H., Nistor V. Analysis of a Modified Schrödinger Operator in 2D: Regularity, Index, and FEM. *Journal of Computational and Applied Mathematics*. 2009. vol. 224, no. 1. pp. 320–338. DOI: 10.1016/j.cam.2008.05.009.
 6. Rukavishnikov V.A., Kuznetsova E.V. A Scheme of a Finite Element Method for Boundary Value Problems with Non-Coordinated Degeneration of Input Data. *Numerical Analysis and Applications*. 2009. vol. 2, no. 3. pp. 250–259. DOI: 10.1134/S1995423909030069
 7. Rukavishnikov V.A., Rukavishnikova H.I. The Finite Element Method for a Boundary Value Problem with Strong Singularity. *Journal of Computational and Applied Mathematics*. 2010. vol. 234, no. 9. pp. 2870–2882. DOI: 10.1016/j.cam.2010.01.020.
 8. Rukavishnikov V.A., Mosolapov A.O. New Numerical Method for Solving Time-Harmonic Maxwell Equations with Strong Singularity. *Journal of Computational Physics*. 2012. vol. 231, no. 6. pp. 2438–2448. DOI: 10.1016/j.jcp.2011.11.031.
 9. Rukavishnikov V.A., Rukavishnikova H.I. On the Error Estimation of the Finite Element Method for the Boundary Value Problems with Singularity in the Lebesgue Weighted Space. *Numerical Functional Analysis and Optimization*. 2013. vol. 34, no. 12. pp. 1328–1347. DOI: 10.1080/01630563.2013.809582.
 10. Rukavishnikov V.A., Rukavishnikova E.I. The Finite Element Method for Boundary Value Problems with Strong Singularity and Double Singularity. *Lecture Notes in Computer Science*. 2013. vol. 8236. pp. 110–121. DOI: 10.1007/978-3-642-41515-9_10.
 11. Nikol'skij S.M. A Variational Problem for an Equation of Elliptic Type with Degeneration on the Boundary. *Proceedings of the Steklov Institute of Mathematics*. 1981. vol. 150. pp. 227–254.
 12. Lizorkin P.I., Nikol'skij S.M. An Elliptic Equation with Degeneracy. A Variational Method. *Soviet Mathematics. Doklady*. 1981. vol. 23. pp. 237–240.
 13. Lizorkin P.I., Nikol'skij S.M. Elliptic Equations with Degeneracy. Differential Properties of Solutions. *Soviet Mathematics. Doklady*. 1981. vol. 23. pp. 268–271.
 14. Rukavishnikov V.A., Rukavishnikova E.I. On the Isomorphic Mapping of Weighted Spaces by an Elliptic Operator with Degeneration on the Domain Boundary. *Differential Equations*. 2014. vol. 50, no. 3. pp. 345–351. DOI: 10.1134/S0012266114030082
 15. Nikol'skii S.M. Approximation of Functions of Several Variables and Imbedding Theorems. Springer-Verlag, New York. 1975. 420 p.
 16. Aubin J.P. Approximation of Elliptic Boundary-Value Problems. Wiley-Interscience Inc., New York, London, Sydney, Toronto. 1972. 360 p.
 17. Ciarlet P.G. The Finite Element Method for Elliptic Problems. North Holland Publishing Company, Amsterdam, New York, Oxford, 1978. 529 p.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ СИЛОВЫХ НАГРУЗОК НА ОПОРЫ НАДВОДНОЙ КОНСТРУКЦИИ НА ОСНОВЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ВОЛНОВЫХ ПРОЦЕССОВ

© 2019 С.В. Проценко¹, А.М. Атаян¹, А.Е. Чистяков¹, А.В. Никитина²,
В.Н. Литвинов³, А.А. Филина⁴

¹Донской государственный технический университет
(344000 Ростов-на-Дону, пл. Гагарина, д. 1),

²Южный федеральный университет
(347922 Таганрог, ул. Чехова, д. 2),

³Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ
(347740 Зерноград, ул. Ленина, д. 21),

⁴ООО «НИЦ супер-ЭВМ и нейрокомпьютеров»
(347900 Таганрог, пер. Итальянский, д. 106)

E-mail: rab55555@rambler.ru, atayan24@mail.ru, cheese_05@mail.ru,
nikitina.vm@gmail.com, litvinovvn@rambler.ru, j.a.s.s.y@mail.ru

Поступила в редакцию: 16.05.2019

Статья посвящена исследованию воздействий волновых процессов на надводные сооружения с помощью математической модели волновых процессов, базирующейся на системе уравнений Навье—Стокса, включающей три уравнения движения в областях с динамически изменяемой геометрией расчетной области. Метод поправки к давлению использовался для дискретизации гидродинамической модели. Разностные схемы, описывающие математическую модель выхода волны на берег, построены на основе интегроинтерполяционного метода с применением схемы с весами. Для решения системы сеточных уравнений использовался адаптивный модифицированный попеременно-треугольный итерационный метод. Практическая значимость численных алгоритмов и реализующего их комплекса программ состоит в возможности их применения при исследовании гидрофизических процессов в прибрежных водных системах, для расчета поля скоростей и давления водного потока, а также оценки гидродинамического воздействия на берегозащитные сооружения и прибрежные конструкции при наличии поверхностных волн.

Ключевые слова: волновой процесс, математическая модель, зоны осушения и затопления, надводные конструкции, силовое воздействие, программный комплекс, алгоритм.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Проценко С.В., Атаян А.М., Чистяков А.Е., Никитина А.В., Литвинов В.Н., Филина А.А. Экспериментальное исследование силовых нагрузок на опоры надводной конструкции на основе математической модели волновых процессов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 27–42. DOI: 10.14529/cmse190302.

Введение

Процесс освоения мелководных береговых зон за счет конструктивных преобразований, влекущих за собой изменение прибрежных рельефов, приобретает все большее народно-хозяйственное значение. Изучение и прогноз гидродинамического воздействия на технические объекты, включая установленные на дне водного тела различного вида

берегозащитные сооружения и прибрежные конструкции, является важной актуальной задачей. В последние десятилетия появилось значительное число как экспериментальных, так и теоретических работ, посвященных моделированию и анализу динамически меняющихся процессов береговой зоны, но эффективность предлагаемых подходов пока далека от практически необходимой. Разрабатываемые модели зачастую имеют узкий диапазон применимости для прогноза и изучения влияния процессов гидродинамики на конкретные технические конструкции, строятся на основе практических исследований в сложных условиях реальной акватории, и, как следствие, используют весьма ограниченную базу данных натуральных измерений. Разрабатываемые модельные задачи водной экологии являются пространственно-неоднородными, их аналитические решения зачастую либо невозможно найти, либо можно получить только при некоторых предположениях и идеализации из-за использования нелинейных уравнений Навье—Стокса.

Статья посвящена разработке и исследованию нелинейных моделей волновых гидродинамических процессов, которые могут оказывать негативное воздействие на технические сооружения, расположенные на береговой линии водоема. Разработанные алгоритмы решения модельных задач гидрофизики и их численная реализация в виде проблемно-ориентированного программного комплекса имеют широкое практическое применение для исследования волновых процессов прибрежных систем, а также для расчета полей скоростей, давления водной среды и функции возвышения уровня при наличии поверхностных волн для оценки гидродинамического воздействия на берегозащитные сооружения и прибрежные конструкции, установленные на донной поверхности водного тела.

Статья организована следующим образом. В разделе 1 приводится постановка трёхмерной задачи волновой гидродинамики. Раздел 2 посвящен описанию упрощенной двумерной модели волновой гидродинамики. В разделе 3 приводятся результаты исследования разностных схем, построенных по методу частичной заполненности ячеек модельной области, используемых для решения разработанных модельных задач волновой гидродинамики. Раздел 4 посвящен описанию результатов численных экспериментов на основе волновой гидродинамической модели. В заключении приводится анализ результатов, полученных в работе, и указаны области применения разработанного программного комплекса.

1. Постановка трехмерной задачи волновой гидродинамики

Пространственно-неоднородная трехмерная математическая модель волновой гидродинамики мелководного водоема включает [1–3]:

– уравнения движения (Навье—Стокса):

$$\begin{aligned} u'_t + uu'_x + vu'_y + wu'_z &= -\frac{1}{\rho} P'_x + (\mu u'_x)'_x + (\mu u'_y)'_y + (v u'_z)'_z, \\ v'_t + uv'_x + vv'_y + wv'_z &= -\frac{1}{\rho} P'_y + (\mu v'_x)'_x + (\mu v'_y)'_y + (v v'_z)'_z, \\ w'_t + uw'_x + vw'_y + ww'_z &= -\frac{1}{\rho} P'_z + (\mu w'_x)'_x + (\mu w'_y)'_y + (v w'_z)'_z + g; \end{aligned} \quad (1)$$

– уравнение неразрывности:

$$\rho'_t + (\rho u)'_x + (\rho v)'_y + (\rho w)'_z = 0, \quad (2)$$

где $V = \{u, v, w\}$ — вектор скорости водного потока мелководного водоема; ρ — плотность водной среды; P — гидродинамическое давление; g — ускорение свободного падения; μ, ν — коэффициенты турбулентного обмена в горизонтальном и вертикальном направлениях; n — вектор нормали к поверхности, описывающей границу расчетной области.

К системе (1)–(2) добавим граничные условия:

- на входе (левая граница): $V = V_0, P'_n = 0,$
- донная граница: $\rho\mu(\mathbf{V}_\tau)'_n = -\boldsymbol{\tau}, \mathbf{V}_n = 0, P'_n = 0,$
- боковая граница: $(\mathbf{V}_\tau)'_n = 0, \mathbf{V}_n = 0, P'_n = 0,$
- верхняя граница: $\rho\mu(\mathbf{V}_\tau)'_n = -\boldsymbol{\tau}, w = -\omega - P'_t / \rho g, P'_n = 0,$ (3)
- на поверхности надводной конструкции: $\rho\mu(\mathbf{V}_\tau)'_n = -\boldsymbol{\tau}, w = 0, P'_n = 0,$

где ω — интенсивность испарения жидкости, $\mathbf{V}_n, \mathbf{V}_\tau$ — нормальная и тангенциальная составляющая вектора скорости, $\boldsymbol{\tau} = \{\tau_x, \tau_y, \tau_z\}$ — вектор тангенциального напряжения.

На рис. 1 представлена геометрия водного тела.

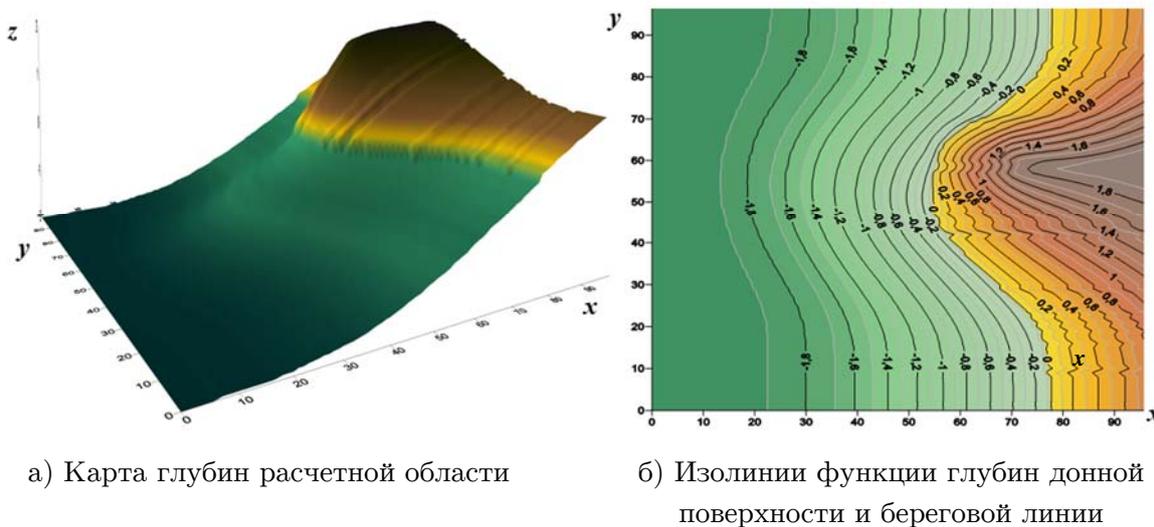


Рис. 1. Модельная область

Пусть $\boldsymbol{\tau} = \rho_a C_d_s |\mathbf{w}| \mathbf{w}$ — вектор тангенциального напряжения для свободной поверхности, $C_d_s = 0,0026$, \mathbf{w} — вектор скорости ветра относительно воды, ρ_a — плотность атмосферы, C_d_s — безразмерный коэффициент поверхностного сопротивления, который зависит от скорости ветра, рассматривается в диапазоне 0,0016–0,0032.

Для дна зададим вектор тангенциального напряжения с учетом движения воды следующим образом: $\boldsymbol{\tau} = \rho C_d_b |\mathbf{V}| \mathbf{V}$, $C_d_b = gk^2 / h^{1/3}$, где $k = 0,04$ — групповой коэффициент шероховатости в формуле Мэннинга, рассматривается в диапазоне 0,025–0,2; $h = H + \eta$ — глубина акватории, [м]; H — глубина до невозмущенной поверхности, [м]; η — высота свободной поверхности относительно геоида (уровень моря), [м].

Будем использовать аппроксимацию, позволяющую строить неоднородный по глубине коэффициент вертикального турбулентного обмена на основании измеренных пульсаций скоростей водного потока [4]:

$$v = C_s^2 \Delta^2 \frac{1}{2} \sqrt{\left(\frac{\partial \bar{U}}{\partial z}\right)^2 + \left(\frac{\partial \bar{V}}{\partial z}\right)^2}, \quad (4)$$

где C_s — безразмерная эмпирическая константа, определяемая на основе расчета процесса затухания однородной изотропной турбулентности; Δ — характерный масштаб сетки; \bar{U}, \bar{V} — усредненные по времени пульсации компонент скорости водного потока в горизонтальном направлении.

При решении задачи вида (1)–(3) использовался метод поправки к давлению. Случай переменной плотности для данного метода описан в работах [3–5]. Переход к системе сеточных уравнений для решения модельных задач гидродинамики проводился с учетом «заполненности» контрольных ячеек расчетной области, что позволило повысить точность решения задач для динамически меняющейся геометрии расчетной области за счет более точной аппроксимации ее границы. Под «заполненностью» контрольной ячейки расчетной области будем понимать коэффициент, представляющий собой отношение объема части ячейки, заполненной средой, к общему объему ячейки [3, 6]. Для решения дискретных аналогов [7–9] системы уравнений (1) использовался алгоритм модифицированного попеременно-треугольного метода вариационного типа.

2. Упрощенная двумерная модель волновой гидродинамики

Для построения двумерной математической модели движения водной среды использовалась трехмерная гидростатическая модель, включающая:

– уравнения Навье—Стокса:

$$\begin{aligned} u'_t + uu'_x + vu'_y + wu'_z &= -\frac{1}{\rho} P'_x + (\mu u'_x)'_x + (\mu u'_y)'_y + (\eta u'_z)'_z, \\ v'_t + uv'_x + vv'_y + wv'_z &= -\frac{1}{\rho} P'_y + (\mu v'_x)'_x + (\mu v'_y)'_y + (\eta v'_z)'_z, \end{aligned} \quad (5)$$

– уравнение неразрывности (для несжимаемой жидкости):

$$u'_x + v'_y + w'_z = 0, \quad (6)$$

– уравнение гидростатики:

$$P = \rho g (z + \xi). \quad (7)$$

Добавим условия на границе:

– условие непроницаемости и трения (на дне):

$$\rho_v \eta u'_n = \tau_{x,b}, \quad \rho_v \eta v'_n = \tau_{y,b}, \quad \mathbf{V}_n = 0,$$

– зададим подъем уровня и ветровые напряжения (на поверхности водоема):

$$\rho \eta u'_n = -\tau_{x,p}, \quad \rho \eta v'_n = -\tau_{y,p}, \quad w = -\xi'_t,$$

– условие скольжения без трения (на боковых границах):

$$u'_n = 0, \quad v'_n = 0, \quad \xi'_n = 0,$$

где ξ — функция возвышения уровня.

Для расчета давления в гидростатическом приближении при наличии объекта, находящегося на поверхности водоема, использовалось уравнение:

$$P = \rho g (z + \chi) + P_g = \rho g (z + \chi + P_g / \rho g).$$

Функция возвышения уровня была доопределена в области надводного объекта:

$$\xi = \chi + P_g / \rho g,$$

где χ — функция, используемая для описания геометрии дна надводного тела, P_g — давление на поверхности надводного объекта.

В гидростатическом случае уравнение неразрывности (6) имеет вид [10–13]:

$$\theta'_t + (Hu)'_x + (Hv)'_y = 0, \quad (8)$$

где $\theta = \min(\chi, \xi)$, $H = h + \theta$, h — глубина водоема.

Система уравнений мелкой воды при наличии объекта, находящегося на поверхности водоема, будет иметь вид:

$$\begin{aligned} Hu'_t + Huv'_x + Hvu'_y &= -gH\xi'_x + (H\mu u'_x)'_x + (H\mu v'_y)'_y + (\tau_{x,p} - \tau_{x,b}) / \rho, \\ Hv'_t + Huv'_x + Hvu'_y &= -gH\xi'_y + (H\mu v'_x)'_x + (H\mu v'_y)'_y + (\tau_{y,p} - \tau_{y,b}) / \rho. \end{aligned} \quad (9)$$

Предложенная двумерная модель волновых гидродинамических процессов может описывать динамические процессы водной среды как в случае наличия надводных объектов, расположенных на поверхности водоема, так и при их отсутствии. Из разработанной системы уравнений можно получить классическую модель движения мелкой воды и 2D-модель без учета динамического изменения геометрии поверхности водоема.

3. Исследование разностных схем, построенных по методу частичной заполненности ячеек модельной области

Для проверки эффективности работы метода, учитывающего частичную заполненность расчетных ячеек модельной области, рассмотрим задачу о стационарном течении жидкости между двумя соосными бесконечно длинными круговыми цилиндрами:

$$uu'_x + vv'_y = -\rho^{-1}P'_x + \mu\Delta u, \quad uv'_x + vv'_y = -\rho^{-1}P'_y + \mu\Delta v, \quad r_1 \leq r \leq r_2, \quad r = \sqrt{x^2 + y^2}. \quad (10)$$

Решение этой задачи имеет вид:

$$u(x, y) = -\frac{5y}{x^2 + y^2}, \quad v(x, y) = \frac{5x}{x^2 + y^2}, \quad P(x, y) = P(r_1) - \frac{12,5\rho}{x^2 + y^2} + \rho/2. \quad (11)$$

Опишем численное решение тестовой модельной задачи течения вязкой жидкости между двумя соосными полуцилиндрами ($x \geq 0$). Радиусы внутреннего и внешнего цилиндров возьмем равными: $r_1 = 5$ м, $r_2 = 10$ м. Впишем расчетную область в прямоугольник ($0 \leq x \leq 10, -10 \leq y \leq 10$) размерами 10×20 м². Зададим компоненты скорости движения жидкости: $u(0, y) = -5/y$ м/с, $v(0, y) = 0$ м/с в сечении цилиндра плоскостью $x = 0$. На грубой сетке (размерность 21×11 узлов, шаги по пространственным направлениям – 1 м, шаг по времени 0,1 с, длина расчетного интервала 10 с) наиболее видны погрешности численных решений. Зададим коэффициент турбулентного обмена $\mu = 1$ м²/с и плотность среды $\rho = 1000$ кг/м³.

На рис. 2 представлено численное решение задачи, где отражены потоки движения жидкости $|k_0 \mathbf{V}|$ с помощью цветовой палитры.

Согласно рис. 2а решение задачи течения жидкости между двумя соосными полуцилиндрами, полученное на сетках, учитывающих заполненность ячеек, является достаточно гладким. Численное решение, представленное на рис. 2б, полученное со ступенча-

той аппроксимацией границы раздела двух сред, имеет существенные ошибки в определении направления векторов скорости потока жидкости.

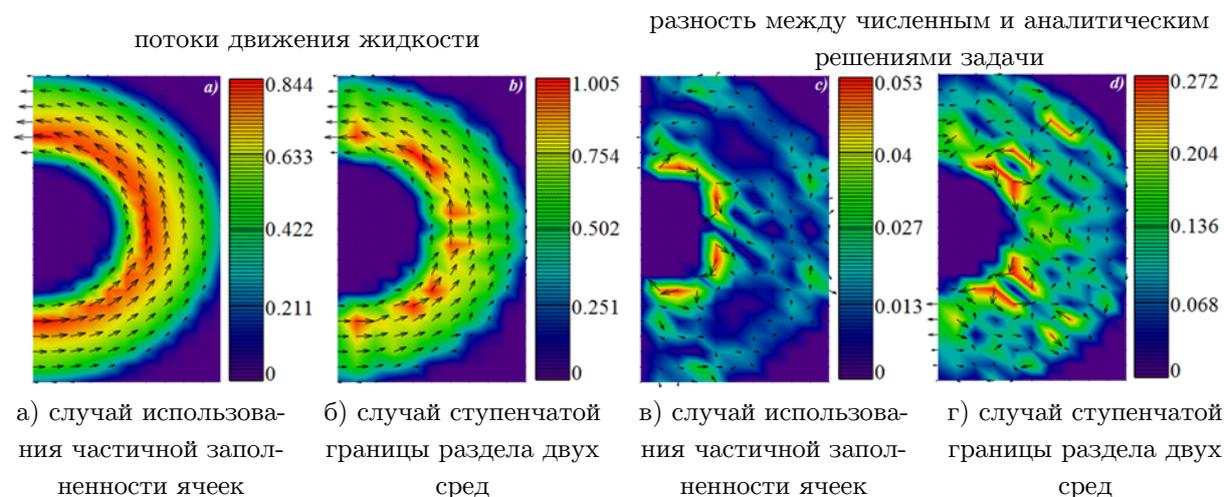


Рис. 2. Численное решение задачи

Рис. 2в, 2г отражают значения погрешности численного решения задачи на сетках, учитывающих заполненность ячеек (случай гладкой границы), и на сетках со ступенчатой аппроксимацией границы. Экспериментальное исследование точности разработанных схем проведено с помощью найденного решения на последовательности сгущающихся сеток: 11×21 , 21×41 , 41×81 , 81×161 расчетных узлов. В таблице представлены значения погрешностей численного решения тестовой задачи течения жидкости между двумя полуцилиндрами для гладкой и ступенчатой границ.

Таблица

Погрешность решения тестовой задачи волновой гидродинамики

| Размеры сетки | 11×21 | 21×41 | 41×81 | 81×161 |
|---|----------------|----------------|----------------|-----------------|
| Максимальное значение погрешности в случае гладкой границы, м/с | 0,053 | 0,052 | 0,058 | 0,056 |
| Среднее значение погрешности в случае гладкой границы, м/с | 0,023 | 0,012 | 0,006 | 0,003 |
| Максимальное значение погрешности в случае ступенчатой границы, м/с | 0,272 | 0,731 | 0,717 | 0,75 |
| Среднее значение погрешности в случае ступенчатой границы, м/с | 0,165 | 0,132 | 0,069 | 0,056 |

Анализ результатов расчетов погрешности численного решения тестовой задачи на последовательности сгущающихся сеток, представленных в таблице, позволяет сделать вывод об эффективности использования разностных схем, учитывающих частичную заполненность расчетных ячеек. Было установлено, что дробление сетки в 8 раз по каждому из пространственных направлений при решении тестовой модельной задачи не приводит к повышению точности, которой обладают решения, полученные на сетках, учитывающих частичную заполненность расчетных ячеек.

4. Описание программного комплекса

Для реализации модельных задач гидродинамики был разработан программный комплекс, реализованный на C++. Параллельные алгоритмы, реализованные в про-

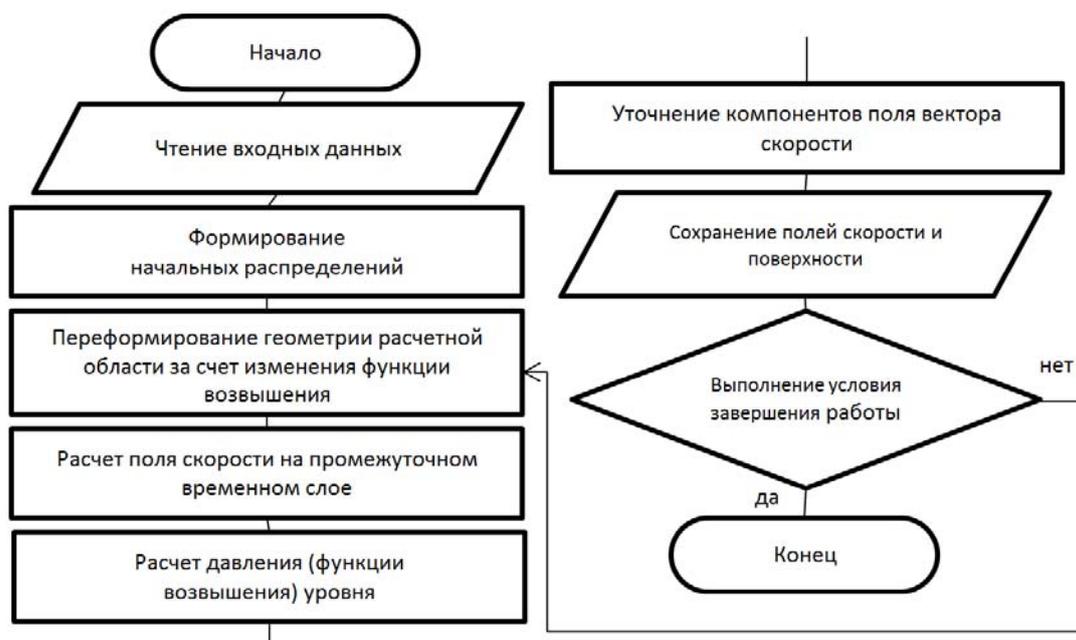


Рис. 3. Алгоритм работы программного комплекса

граммном комплексе для решения возникающих в процессе дискретизации модельных задач систем сеточных уравнений модифицированным адаптивным попеременно-треугольным методом вариационного типа, были разработаны с использованием технологии MPI. Схема алгоритма работы программы, численно реализующей разработанные 2D и 3D модели волновой гидродинамики, представлена на рис. 3.

Разработанный программный комплекс включает: управляющий блок (содержит цикл по временной переменной, и вызываются функции: расчет поля скорости без учета функции возвышения уровня, расчет функции возвышения уровня, расчет двумерного поля скорости, проверка наличия на поверхности водной среды конструкции и функции ввода-вывода данных); блок ввода начальных распределений для расчета скорости течений и функции возвышения уровня (задаются начальные распределения поля скорости и функции возвышения уровня и начальные значения степени заполненности расчетных ячеек); блок построения сеточных уравнений для 2D и 3D полей скоростей водного потока без учета давления (для 2D модели мелкой воды — без учета функции возвышения уровня); блок построения сеточных уравнений для расчета поля давления и функции возвышения уровня; блок проверки наличия на поверхности водной среды конструкции; блок расчета поля скорости с учетом давления; блок решения сеточных уравнений модифицированным адаптивным попеременно-треугольным методом вариационного типа; блок вывода значений поля скоростей и функции давления (поля функции возвышения уровня).

5. Результаты численных экспериментов на основе волновой гидродинамической модели

Разработанный комплекс программ позволяет задавать сложную конфигурацию надводного объекта, а также вид и характеристики источника колебаний. Рис. 4 отражает результаты прогноза изменения волновых гидродинамических процессов при обтекании водной средой надводного тела с учетом сложной геометрии объекта, находящегося в жидкости.

В качестве примера использования разработанного комплекса программ была решена модельная задача расчета гидродинамического воздействия волн на опоры сооружения. Надводная конструкция имела следующие размеры: ширину 5 м, длину 10 м, глубину погружения 20 см, была закреплена на донной поверхности водоема на 6 опорах. Рассматриваемый участок моделирования имеет размеры 50 на 50 м по горизонтали и 1 м по вертикали (в глубину). Источник возмущений был задан на некотором расстоянии от надводного объекта. Будем считать, что в начальный момент времени жидкость находится в состоянии покоя, чтобы отраженная от границ расчетной области волна не изменяла параметры силового гидродинамического воздействия на надводный объект, представляющий собой техническое сооружение. В модельной задаче на основе численного эксперимента рассчитывается последующее движение водной среды при наличии на поверхности надводного объекта, а также силовые гидродинамические нагрузки на опоры сооружения. При расчетах использовалась сетка 100x100 расчетных узлов, шаг по времени составил 0,01 секунды (рис. 4, 5). Источник колебаний имеет синусоидальную форму и был распределен по левой границе. При распространении плоской волны, которая встречает препятствие в виде надводного тела, представленного на рис. 4, 5, происходит отражение волновых колебаний от неподвижного объекта, меняющего профиль волны.

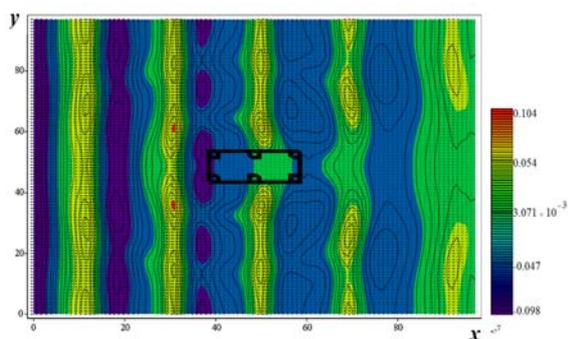


Рис. 4. Поле движения водной среды

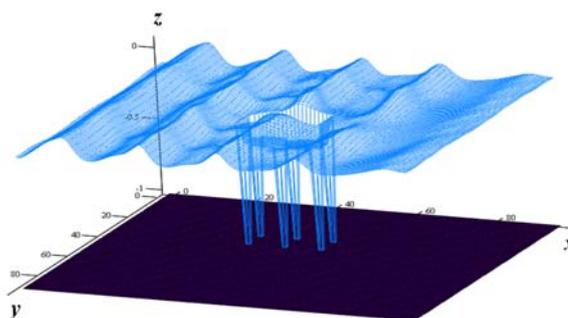


Рис. 5. Функция возвышения уровня

Проведенные численные эксперименты по моделированию распространения гидродинамических волновых процессов позволили проанализировать воздействия волн на сооружения, имеющие донные опоры. На рис. 6 представлено силовое гидродинамическое воздействие на опоры надводной конструкции, закрепленной на дне водоема (см. рис. 3, 4). Силовые нагрузки отмечены по оси ординат в тоннах, по оси абсцисс отражено время в секундах, отсчитанное от начала колебательного процесса.

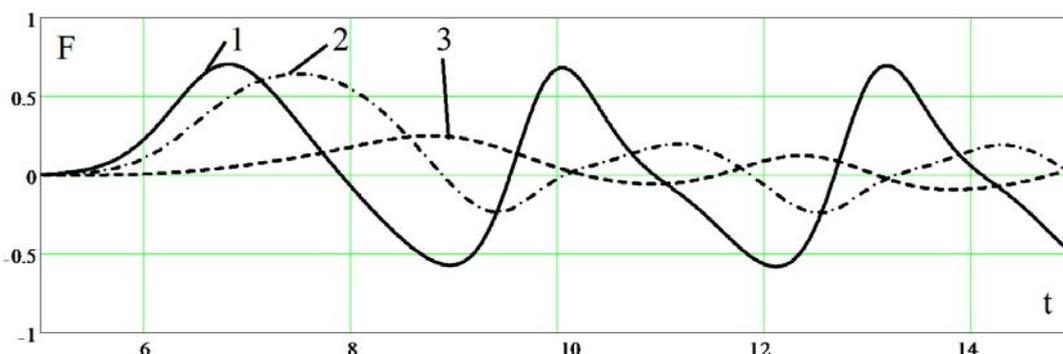


Рис. 6. График силовых нагрузок на опоры конструкции: 1 — со стороны набегающей волны на переднюю пару опор, 2 — на центральную пару, 3 — на дальнюю пару

Разработанные численные алгоритмы решения модельных задач и реализующий их комплекс программ может использоваться при исследовании гидрофизических процессов в прибрежных водных системах [14–20], а также для нахождения поля скоростей и давления водной среды, оценки возможного негативного воздействия на береговую линию при наличии поверхностных волн. Разработанный программный инструментарий позволяет задавать параметры источника колебаний, а также батиметрию водоема. Результаты численных экспериментов по моделированию распространения волновых гидродинамических процессов при выходе волны на берег с учетом геометрий дна объекта, находящегося в жидкости и дна водоема представлены на рис. 7 в различные моменты времени.

Разработка трехмерной модели волновых гидродинамических процессов на основании натуральных данных позволила описать движение водной среды мелководного водоема при учете выхода волны на берег. Создан современный программный комплекс, который адаптирован для моделирования гидродинамических волновых процессов, использу-

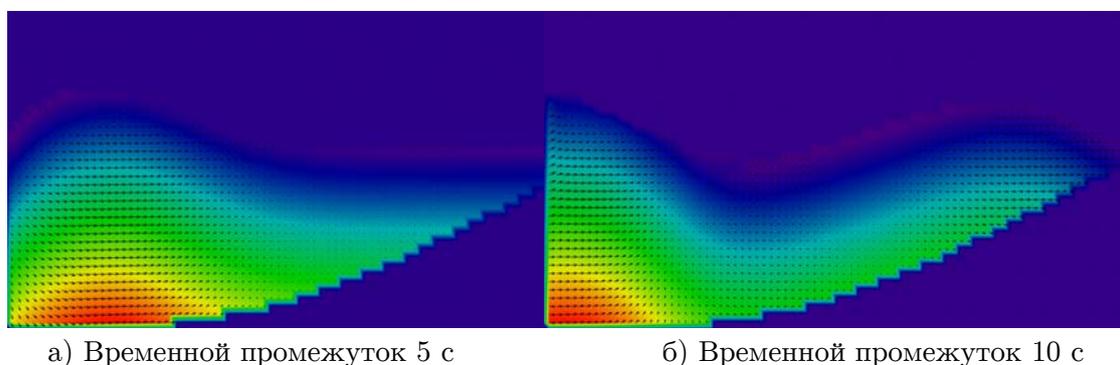
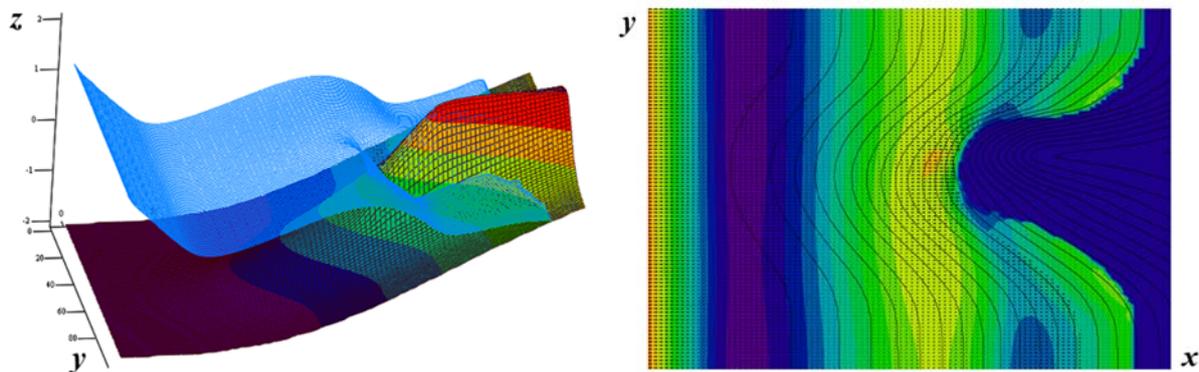


Рис. 7. Поле вектора скорости движения водной среды (срез плоскости XOZ)

ется в широком диапазоне параметров для расчета полей скоростей и давления водной среды, и оценки гидродинамического воздействия на берег при наличии поверхностных волн. На основе разработанного комплекса программ построен прогноз изменения гидродинамических волновых процессов прибрежной зоны водного тела. При практическом использовании проблемно-ориентированного комплекса программ решалась модельная задача расчета полей скорости водного потока и давления водной среды. Источник возмущений задается на заданном расстоянии от линии берега. Область моделирования имеет размеры 50 на 50 м и глубину 2 м, пиковая точка возвышается над уровнем моря на 2 м. Предположим, что жидкость находится в состоянии покоя в начальный момент времени. При решении поставленной модельной задачи использовалась сетка размерами $100 \times 100 \times 40$ расчетных узлов, шаг по времени составил 0,01 с. Рис. 7 демонстрирует поле вектора скорости движения водной среды при накате волны на береговую линию, при этом динамически меняется функция возвышения уровня, образуются зоны затопления и обмеления.

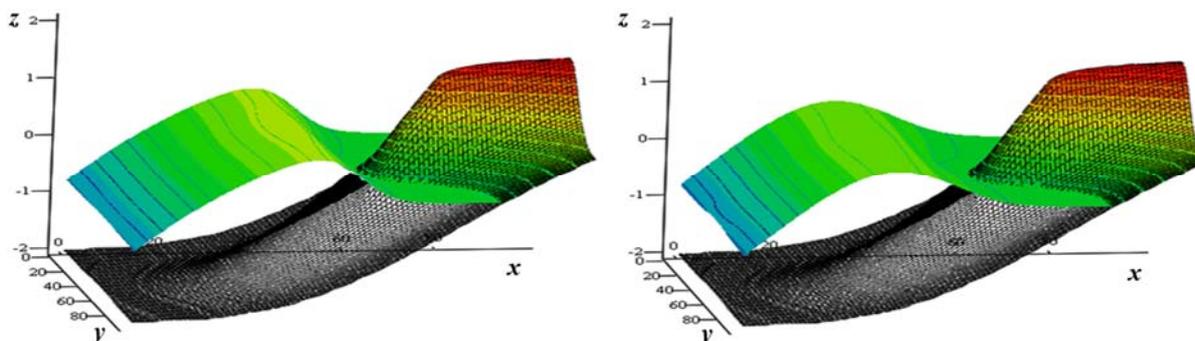
Из рис. 8 видно, что участок суши был затоплен набегающей волной. Учет затопления и осушения прибрежных участков осуществлялся за счет пересчета заполненностей расчетных ячеек. Предложенный подход позволяет решать задачи в областях со сложной и динамически перестраиваемой геометрией границы. Разработанный программный комплекс обладает отличительной чертой — при моделировании распространения поверхностных колебаний учитывается выход волны на береговую линию.



а) График функции возвышения уровня б) График функции рельефа дна

Рис. 8. Графики функций возвышения уровня и рельефа дна

На рис. 9 представлены результаты моделирования динамического нестационарного процесса, характеризующего выход волны на берег, полученные на основе двух различных моделей при одинаковых входных параметрах. Временной интервал равен 5 с.



а) На основе двумерной системы уравнений мелкой воды б) На основе математической модели, включающей три уравнения движения

Рис. 9. Сопоставление результатов моделирования выхода волны на берег

На рис. 10 представлены графики функций возвышения уровня, полученные на основе двумерной системы уравнений мелкой воды и математической модели, включающей три уравнения движения при сечении плоскостью $y = 25$.

Результаты численного эксперимента, представленные на рис. 10, достаточно сильно отличаются друг от друга. Профили волны, рассчитанной на основе двумерной модели мелкой воды, обгоняют профили волны, полученной на основе трехмерной модели. Максимальное расстояние по высоте между двумя профилями волны составило 0,329 м.

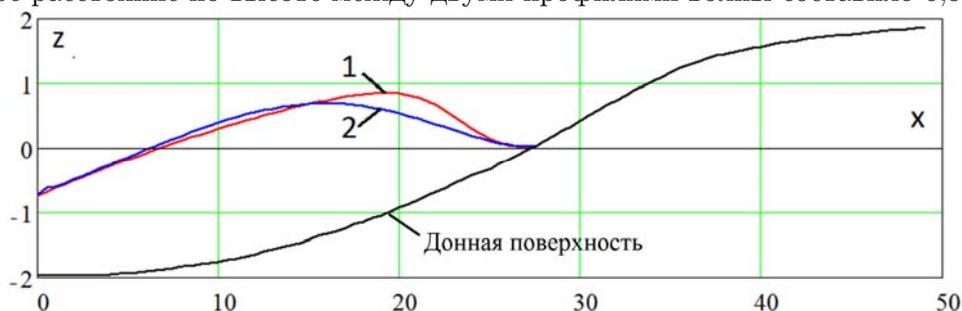


Рис. 10. Сопоставление результатов расчета функций возвышения уровня:
1 — на основе двумерной модели, 2 — на основе трехмерной модели

Была численно решена задача расчета гидродинамического воздействия волн на конструкцию с размерами: 10 на 10 м, погруженную на глубину 0,5 м, которая установлена на дне водоема, рассматриваемый для моделирования участок имеет размеры 50 на 50 м по горизонтали и 4 м по вертикали (по 2 м в глубину и в высоту). Зададим на некотором расстоянии от надводного объекта источник возмущений.

На рис. 11 представлены результаты численного эксперимента на основе трехмерной модели волновых процессов.

Разработанный программный комплекс использовался для численной реализации предложенной трехмерной модели волновой гидродинамики и построения прогноза движения водной среды при наличии технического объекта, расположенного ниже уровня невозмущенной поверхности водоема, были рассчитаны силовые гидродинамические нагрузки на опоры сооружения.

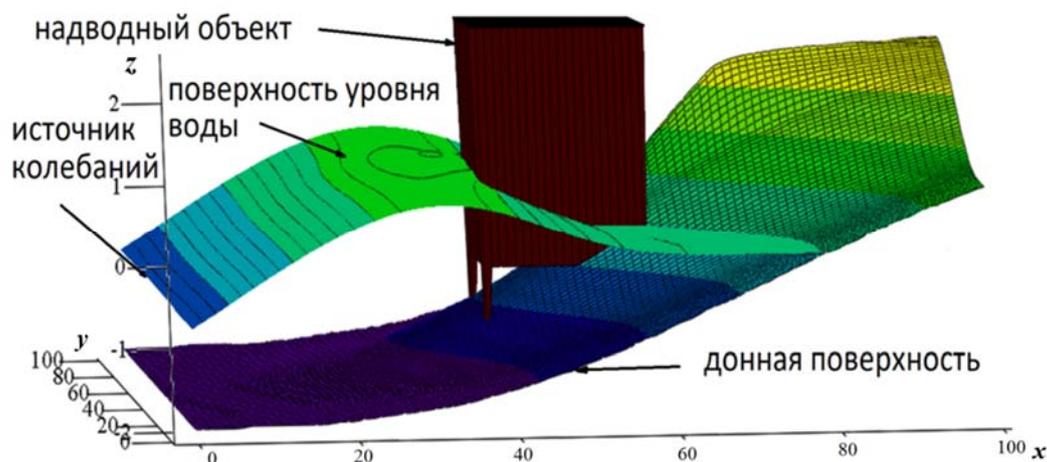


Рис. 11. Графики функций возвышения уровня и рельефа дна

Заключение

Разработка и исследование модели трехмерных волновых процессов с учетом выхода волны на берег позволила провести сравнение результатов расчета двумерной и трехмерной моделей изменения волновых процессов на мелководье. Протестировано использование метода частичной заполненности расчетных ячеек, доказано его преимущество, выражающееся в значительном уменьшении погрешности решения задач волновой гидродинамики в областях со сложной геометрией. Приведено описание разработанного программного комплекса, позволяющего менять характеристики и расположение источника колебаний, а также учитывать форму зон и степень интенсивности осушения и затопления прибрежных участков. Созданный программный комплекс можно широко использовать для практического исследования расчета силового воздействия волн на геометрию донной поверхности, а также объекты надводной и прибрежной инфраструктуры. Разработанный проблемно-ориентированный программный комплекс может использоваться для прогнозирования изменения процессов водной экологии в результате опасных явлений природного и техногенного характеров.

Работа выполнена при поддержке гранта РФФИ №19-07-00623.

Литература

1. Белоцерковский О.М., Гущин В.А., Щенников В.В. Метод расщепления в применении к решению задач динамики вязкой несжимаемой жидкости // Ж. вычисл. матем. и матем. физ. 1975. Т. 15, № 1. С. 197–207.
2. Сухинов А.И., Чистяков А.Е., Алексеенко Е.В. Численная реализация трехмерной модели гидродинамики для мелководных водоемов на супервычислительной системе // Математическое моделирование. 2011. Т. 23, № 3. С. 3–21.
3. Сухинов А.И., Чистяков А.Е. Параллельная реализация трехмерной модели гидродинамики мелководных водоемов на супервычислительной системе // Вычислительные методы и программирование: Новые вычислительные технологии. 2012. Т. 13, С. 290–297.
4. Белоцерковский О. М. Турбулентность: новые подходы. М.: Наука, 2003.
5. Гущин В.А., Миткин В.В., Рождественская Т.И., Чашечкин Ю.Д. Численное и экспериментальное исследование тонкой структуры течения стратифицированной жидкости вблизи кругового цилиндра // Прикладная механика и техническая физика. 2007. Т. 48, № 1 (281). С. 43–54.
6. Сухинов А.И., Чистяков А.Е., Тимофеева Е.Ф., Шишениа А.В. Математическая модель расчета прибрежных волновых процессов // Математическое моделирование. 2012. Т. 24, №8. С. 32–44.
7. Сухинов А.И., Чистяков А.Е., Фоменко Н.А. Методика построения разностных схем для задачи диффузии-конвекции-реакции, учитывающих степень заполненности контрольных ячеек // Известия ЮФУ. Технические науки. 2013. №4. С 87–96.
8. Самарский А.А. Теория разностных схем. М.: Наука, 1989.
9. Самарский А.А., Вабищевич П.Н. Численные методы решения задач конвекции-диффузии. М.: Эдиториал УРСС, 1999.
10. Васильев В.С., Сухинов А.И. Прецизионные двумерные модели мелких водоемов // Математическое моделирование. 2003. Т. 15, № 10. С. 17–34.
11. Сухинов А.И., Чистяков А.Е., Проценко Е.А. Математическое моделирование транспорта наносов в прибрежных водных системах на многопроцессорной вычислительной системе // Вычислительные методы и программирование. 2014. Т. 15. С. 610–620.
12. Sukhinov A.I., Chistyakov A.E., Protsenko E.A. Mathematical Modeling of Sediment Transport in the Coastal Zone of Shallow Reservoirs. // Mathematical Models and Computer Simulations, 6:4 (2014), P. 351–363. DOI: 10.1134/S2070048214040097.
13. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. М.: Наука, 1978.
14. Коновалов А.Н. К теории попеременно - треугольного итерационного метода // Сибирский математический журнал. 2002. Т. 43., № 3. С. 552–572.
15. Сухинов А.И., Чистяков А.Е. Адаптивный модифицированный попеременно-треугольный итерационный метод для решения сеточных уравнений с несамосопряженным оператором // Математическое моделирование. 2012. Т. 24, № 1. С. 3–20.
16. Sukhinov A.I., Sukhinov A.A. Reconstruction of 2001 Ecological Disaster in the Azov Sea on the Basis of Precise Hydrophysics Models. Parallel Computational Fluid Dynamics, Multidisciplinary Applications, Proceedings of Parallel CFD 2004 Conference, Las

- Palmas de Gran Canaria, Spain, Elsevier, Amsterdam-Berlin-London-New York-Tokyo. 2005. P. 231–238. DOI: 10.1016/B978-044452024-1/50030-0.
17. Alekseenko E., Roux B., Sukhinov A., Kotarba R., Fougere D. Coastal Hydrodynamics in a Windy Lagoon // *Nonlinear Processes in Geophysics*. 2013. Vol. 20, No. 2. P. 189–198. DOI: 10.1016/j.compfluid.2013.02.003.
 18. Alekseenko E., Roux B., Sukhinov A., Kotarba R., Fougere D. Nonlinear Hydrodynamics in a Mediterranean Lagoon // *Computational Mathematics and Mathematical Physics*. 2017. Vol. 57, No. 6. P. 978–994. DOI: 10.5194/npg-20-189-2013.
 19. Сухинов А.И., Чистяков А.Е., Семенякина А.А., Никитина А.В. Численное моделирование экологического состояния Азовского моря с применением схем повышенного порядка точности на многопроцессорной вычислительной системе // *Компьютерные исследования и моделирование*. 2016. Т. 8, № 1. С. 151–168.
 20. Гуцин В.А., Семенякина А.А., Сухинов А.И., Чистяков А.Е., Никитина А.В. Модель транспорта и трансформации биогенных элементов в прибрежной системе и ее численная реализация // *Журнал вычислительной математики и математической физики*. 2018. Т. 58, № 8. С. 120–137. DOI: 10.31857/S004446690002007-8.

Проценко Софья Владимировна, аспирант кафедры «Математика и информатика», Донской государственной технической университет (Ростов-на-Дону, Российская Федерация)

Атаян Ася Михайловна, кафедра «Программное обеспечение вычислительной техники и автоматизированных систем», Донской государственной технической университет (Ростов-на-Дону, Российская Федерация)

Чистяков Александр Евгеньевич, д.ф.-м.н., кафедра «Программное обеспечение вычислительной техники и автоматизированных систем», Донской государственной технической университет (Ростов-на-Дону, Российская Федерация)

Никитина Алла Валерьевна, д.т.н., доцент, кафедра «Интеллектуальные и многопроцессорные системы», Южный федеральный университет (Таганрог, Российская Федерация)

Литвинов Владимир Николаевич, к.т.н., доцент, кафедра «Теплоэнергетика и информационно-управляющие системы», Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ (Зерноград, Российская Федерация)

Филина Алёна Александровна, к.т.н., ООО «НИЦ супер-ЭВМ и нейрокомпьютеров» (Таганрог, Российская Федерация)

EXPERIMENTAL RESEARCH OF POWER LOADS ON THE SUPPORTS OF THE SURFACE STRUCTURE BASED ON THE MATHEMATICAL MODEL OF WAVE PROCESSES

© 2019 S.V. Protsenko¹, A.M. Atayan¹, A.E. Chistyakov¹, A.V. Nikitina²,
V.N. Litvinov³, A.A. Filina⁴

¹Don State Technical University (Gagarina sq. 1, Rostov-on-Don, 344000 Russia),

²Southern Federal University (Chekhova, 2, Taganrog, 347922 Russia),

³Azov-Black Sea engineering institute of FSBHEEPT "Don State Agrarian University"
(Lenina 21, Zernograd, 347740 Russia),

⁴Scientific Research Center of supercomputers and Neurocomputers
(per. Italyansky 106, Taganrog, 347900 Russia)

E-mail: rab55555@rambler.ru, atayan24@mail.ru, cheese_05@mail.ru,
nikitina.vm@gmail.com, litvinovvn@rambler.ru, j.a.s.s.y@mail.ru,

Received: 16.05.2019

The article is devoted to the research of the effects of wave processes on above-water constructions using a mathematical model of wave processes based on the system of Navier–Stokes equations, which includes three equations of motion in areas with dynamically changing geometry of the computational domain. The pressure correction method is used for the approximation of a hydrodynamic model. The difference schemes describing the mathematical model of the wave exit to the shore are constructed on the basis of the integro-interpolation method using the scheme with weights. An adaptive alternating-triangular iterative method is used to solve the system of grid equations. The practical significance of numerical algorithms and the complex of programs implementing them consists in the possibility of their application in the study of hydrophysical processes in coastal water systems, as well as for the construction of the velocity field and pressure of the aquatic environment, as well as the assessment of the hydrodynamic effect on coastal protection structures and riparian constructions in the presence of surface waves.

Keywords: wave process, mathematical model, areas of drainage and flooding, surface structures, force, software, algorithm.

FOR CITATION

Protsenko S.V., Atayan A.M., Chistyakov A.E., Nikitina A.V., Litvinov V.N., Filina A.A. Experimental Research of Power Loads on the Supports of the Surface Structure Based on the Mathematical Model of Wave Processes. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. vol. 8, no. 3. pp. 27–42. (in Russian) DOI: 10.14529/cmse190302.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Belocerkovskij O.M., Gushchin V.A., Shchennikov V.V. Splitting Method Applied to Solving Problems of Viscous Incompressible Fluid Dynamics. *Zhurnal vychislitel'noj ma-*

- tematiki i matematicheskoy fiziki* [Journal of Computational Mathematics and Mathematical Physics]. 1975. vol. 15, no. 1. pp. 197–207. (in Russian)
2. Suhinov A.I., Chistyakov A.E., Alekseenko E.V. Numerical Implementation of a Three-Dimensional Hydrodynamics Model for Shallow Water Bodies on a Supercomputing System. *Matematicheskoe modelirovanie* [Mathematical Modeling]. 2011. vol. 23, no. 3. pp. 3–21. (in Russian)
 3. Suhinov A.I., Chistyakov A.E. Parallel Implementation of a Three-Dimensional Model of Hydrodynamics of Shallow Water Bodies on a Supercomputing System. *Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tekhnologii* [Computational Methods and Programming: New Computational Technologies]. 2012. vol. 13. pp. 290–297. (in Russian)
 4. Belocerkovskij O. M. *Turbulentnost': novye podhody* [Turbulence: New Approaches]. Nauka, M., 2003
 5. Gushchin V.A., Mitkin V.V., Rozhdestvenskaya T.I., Chashechkin YU.D. Numerical and Experimental Study of the Fine Structure of Stratified Fluid Flow Near a Circular Cylinder. *Prikladnaya mekhanika i tekhnicheskaya fizika* [Applied Mechanics and Technical Physics]. 2007. vol. 48, no. 1 (281). pp. 43–54. (in Russian)
 6. Suhinov A.I., Chistyakov A.E., Timofeeva E.F., Shishenya A.V. Mathematical Model of Calculation of Coastal Wave Processes. *Matematicheskoe modelirovanie* [Mathematical Modeling]. 2012. vol. 24, no. 8. pp. 32–44. (in Russian)
 7. Suhinov A.I., Chistyakov A.E., Fomenko N.A. A Method of Constructing Difference Schemes for the Diffusion-Convection-Reaction Problem, Taking Into Account the Degree of Filling of Control Cells. *Izvestiya YUFU. Tekhnicheskie nauki* [News of Southern Federal University. Technical science]. 2013. vol. 4. pp. 87–96. (in Russian)
 8. Samarskij A.A. *Teoriya raznostnyh skhem* [Theory of Difference Schemes]. M. Nauka, 1989.
 9. Samarskij A.A., Vabishchevich P.N. *Chislennyye metody resheniya zadach konvekcii-diffuzii* [Numerical Methods for Solving Convection-Diffusion Problems]. M., Editorial URSS, 1999.
 10. Vasil'ev V.S., Suhinov A.I. Precision Two-Dimensional Models of Shallow Water Bodies. *Matematicheskoe modelirovanie* [Mathematical Modeling]. 2003. vol. 15, no. 10. pp. 17–34. (in Russian)
 11. Suhinov A.I., Chistyakov A.E., Procenko E.A. Mathematical Modeling of Sediment Transport in Coastal Water Systems on a Multiprocessor Computer System. *Vychislitel'nye metody i programmirovaniye* [Computational Methods and Programming]. 2014. vol. 15. pp. 610–620. (in Russian)
 12. Sukhinov A.I., Chistyakov A.E., Protsenko E.A. Mathematical Modeling of Sediment Transport in the Coastal Zone of Shallow Reservoirs. *Mathematical Models and Computer Simulations*. 2014. vol. 6, no. 4. pp. 351–363. DOI: 10.1134/S2070048214040097.
 13. Samarskij A.A., Nikolaev E.S. *Metody resheniya setochnykh uravnenij* [Methods for Solving Grid Equations]. M. Nauka, 1978.
 14. Konovalov A.N. On the Theory of the Alternating - Triangular Iterative Method. *Sibirskiy matematicheskij zhurnal* [Siberian Mathematical Journal]. 2002. vol. 43, no. 3. pp. 552–572. (in Russian)

15. Suhinov A.I., Chistyakov A.E. Adaptive Modified Alternating-Triangular Iterative Method for Solving Grid Equations With Non-Self-Adjoint Operator. *Matematicheskoe modelirovanie* [Mathematical Modeling]. 2012. vol. 24, no. 1. pp. 3–20. (in Russian)
16. Sukhinov A.I., Sukhinov A.A. Reconstruction of 2001 Ecological Disaster in the Azov Sea on the Basis of Precise Hydrophysics Models. Parallel Computational Fluid Dynamics, Multidisciplinary Applications, Proceedings of Parallel CFD 2004 Conference, Las Palmas de Gran Canaria, Spain, ELSEVIER, Amsterdam-Berlin-London-New York-Tokyo. 2005. pp. 231–238. DOI: 10.1016/B978-044452024-1/50030-0.
17. Alekseenko E., Roux B., Sukhinov A., Kotarba R., Fougere D. Coastal Hydrodynamics in a Windy Lagoon. *Nonlinear Processes in Geophysics*. 2013. vol. 20, no. 2. pp. 189–198. DOI: 10.1016/j.compfluid.2013.02.003.
18. Alekseenko E., Roux B., Sukhinov A., Kotarba R., Fougere D. Nonlinear Hydrodynamics in a Mediterranean Lagoon. *Computational Mathematics and Mathematical Physics*. 2017. vol. 57, no. 6. pp. 978–994. DOI: 10.5194/npg-20-189-2013.
19. Sukhinov A.I., Chistyakov A.E., Semenyakina A.A., Nikitina A.V. Numerical Modeling of an Ecological Condition of the Sea of Azov With Application of Schemes of the Raised Accuracy Order on the Multiprocessor Computing System. *Komp'yuternye issledovaniya i modelirovanie* [Computer Researches and Modeling]. 2016. vol. 8, no. 1. pp. 151–168. (in Russian)
20. Gushchin V.A., Sukhinov A.I., Nikitina A.V., Chistyakov A.E., Semenyakina A.A. A Model of Transport and Transformation of Biogenic Elements in the Coastal System and Its Numerical Implementation. *Zhurnal vychislitel'noj matematiki i matematicheskoy fiziki* [Computational Mathematics and Mathematical Physics]. 2018. vol. 58, no. 8. pp. 1316–1333. (in Russian)

ОБЗОР ПОДХОДОВ И ПРАКТИЧЕСКИХ ОБЛАСТЕЙ ПРИМЕНЕНИЯ РАСПОЗНАВАНИЯ ВИДОВ ФИЗИЧЕСКОЙ АКТИВНОСТИ ЧЕЛОВЕКА

© 2019 Е.С. Тарантова, К.В. Макаров, А.А. Орлов

Муромский институт (филиал)

ФГБОУ ВО «Владимирский государственный университет

имени А.Г. и Н.Г. Столетовых»

(602264 Муром, ул. Орловская, д. 23)

E-mail: elena.tarantova@yandex.ru, k-distance@rambler.ru, AlexeyAlexOrlov@gmail.com

Поступила в редакцию: 21.01.2019

Распознавание видов физической активности человека является одним из актуальных направлений исследования в области машинного обучения, так как результаты распознавания необходимы при решении многих практических задач. В статье приводится обзор подходов и практических областей применения методов распознавания видов физической активности человека. Рассматриваются датчики, используемые для распознавания видов физической активности человека, и представлены критерии их выбора. Представлены возможные пути решения проблемы выбора места размещения и ориентации носимых датчиков. В статье рассматриваются основные этапы распознавания видов физической активности человека. Представлены извлекаемые признаки и методы их отбора для повышения точности классификации видов физической активности человека и снижения вычислительных затрат за счет уменьшения числа признаков. Сформулированы достоинства и недостатки популярных методов классификации. Рассматриваются метрики, используемые для оценки качества обучения модели классификации. Наиболее применяемой метрикой качества является кривая ошибок. Также представлены практические задачи, в которых необходимы результаты распознавания видов физической активности человека. Основными областями применения метода распознавания являются медицина, производство, фитнес и безопасность людей. В заключении представлены возможные направления будущих исследований.

Ключевые слова: распознавание образов, машинное обучение, виды физической активности человека.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Тарантова Е.С., Макаров К.В., Орлов А.А. Обзор подходов и практических областей применения распознавания видов физической активности человека // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 43–57. DOI: 10.14529/cmse190303.

Введение

Распознавание видов физической активности человека является одной из актуальных задач в области машинного обучения из-за сложности и разнообразия видов физической активности, выполняемых человеком.

Целью распознавания является определение деятельности человека на основе данных датчиков для последующего анализа системой с учетом практической задачи. Сложный и изменчивый характер данных об активности создает многочисленные проблемы, которые влияют на производительность систем, используемых для решения практических задач. При проведении распознавания видов активности человека также возникают трудности из-за большого разнообразия датчиков для распознавания и вари-

антов их расположения, выбора информативных признаков для обучения и методов классификации.

В статье представлен обзор этапов распознавания видов физической активности с использованием датчиков и методов машинного обучения, а также обзор практических задач, требующих применения метода распознавания физической активности человека.

Статья организована следующим образом. В разделе 1 будут рассмотрены классификации видов активности человека, датчики, используемые для распознавания видов активности, расположение датчиков, их ориентация и этапы проведения распознавания видов физической активности человека.

Раздел 2 будет посвящен практическим задачам в различных областях жизни человека, которые требуют применения метода распознавания физической активности человека. В заключении подводятся итоги и описываются возможные направления будущих исследований.

1. Обзор этапов распознавания видов физической активности человека

1.1. Виды физической активности

В работах [34, 38] предлагают способ классификации активности путем ее группировки по продолжительности и сложности. Таким образом, в работе выделяют виды активности низкого уровня и высокого уровня.

К активности низкого уровня относятся такие виды активности, которые состоят из одного повторяющегося действия. Примерами видов активности низкого уровня являются ходьба, бег, приседания, ходьба по лестнице.

Активность высокого уровня — состоит из набора активностей низкого уровня. Продолжительность активности высокого уровня обычно больше, чем активности низкого уровня. Примерами видов активности высокого уровня являются готовка, уборка, поход в магазин.

В статье [10] распознавали активности высокого уровня и низкого уровня. В ходе исследования было выявлено, что виды активности низкого уровня распознаются с точностью около 93 %, а виды активности высокого уровня — 50 %.

Такая низкая точность связана с тем, что разными людьми вид активности высокого уровня может быть выполнен многими различными способами.

В статье [25] выделяют несколько групп активности в зависимости от практического применения. Классификация видов активности с учетом практического применения представлена в табл. 1.

1.2. Датчики, используемые для распознавания видов активности

Для решения задачи распознавания видов физической активности обычно применяют два типа датчиков: носимые датчики и датчики окружающей среды, например, как в работе [31]. Носимые датчики — это датчики, которые закреплены на человеке. Датчики окружающей среды — это датчики, которые расположены вокруг человека.

Носимые датчики делятся на два вида: датчики движения и физиологические датчики.

Датчики движения — это недорогие и небольшие датчики, которые измеряют ускорение и вращательные силы и, как правило, характеризуются низким потреблением энергии. К датчикам движения относятся акселерометры, гироскопы.

По полученным данным с этих датчиков можно с высокой точностью распознавать виды активности низкого уровня, как указано в работах [16, 28]. Однако при распознавании видов активности высокого уровня до сих пор возникают трудности.

Таблица 1

Виды активности с учетом практического применения

| Практическое применение | Виды активности |
|-------------------------|---|
| Передвижение | Ходьба, бег, сидит, стоит, лежит, ходьба по лестнице, передвижение на эскалаторе, передвижение на лифте |
| Транспорт | Езда на автобусе, езда на велосипеде, вождение автомобиля |
| Ежедневные занятия | Еда, питье, работа на компьютере, просмотр телевизора, чтение, чистка зубов, уборка |
| Фитнес | Гребля, поднятие тяжестей, спиннинг, скандинавская ходьба, отжимания |

Физиологические датчики — это устройства для восприятия различных проявлений жизнедеятельности человека и их преобразования в электрические или другие сигналы. Примерами физиологических датчиков служат датчики электрокардиограммы, сердечного ритма.

Физиологические датчики не предоставляют достаточной информации для распознавания видов активности, поскольку люди могут выполнять каждое действие в различных условиях. Однако физиологические датчики используют для дополнения показаний датчиков движения. Например, ходьбу и ходьбу с переносом тяжести практически невозможно различить, используя только датчики движения. Однако разница может наблюдаться в отдельных физиологических сигналах, как, например, в работе [31].

К датчикам окружающей среды можно отнести датчики температуры, влажности, освещенности, камеры, микрофоны и датчики предметов, окружающих человека.

Как и в случае физиологических датчиков, датчики температуры, влажности, и освещенности отдельно не предоставляют достаточной информации для распознавания.

Преимущество камер и датчиков окружающих предметов в том, что данные с этих датчиков позволяют распознать виды активности высокого уровня.

Однако у данных датчиков есть ряд недостатков:

1. Методы обработки информации с датчиков требуют больших вычислительных затрат.
2. Распознавать виды активности невозможно, если человек находится вне досягаемости датчиков.

1.3. Выбор датчиков для распознавания видов активности человека

В работе [31] предлагается ряд критериев для выбора датчиков, необходимых для распознавания видов активности человека:

1. Вопросы конфиденциальности (сколько и как конфиденциальная информация записывается и хранится датчиком).
2. Удобство и простота настройки датчика (в случае носимых датчиков пользователь должен иметь возможность надевать датчики без посторонней помощи и в короткие сроки).
3. Техническое обслуживание датчика (время автономной работы, насколько легко отремонтировать компоненты датчика).
4. Стоимость датчика (насколько дорого заменить датчик в случае выхода из строя или потери датчика).
5. Вычислительная сложность при обработке информации датчика.

Таким образом, исходя из данных требований, преимуществ и недостатков рассмотренных датчиков, для распознавания видов активности человека следует применять сочетание датчиков движения, физиологических датчиков, а также датчиков окружающих предметов.

В работах по распознаванию видов активности часто используют датчики смартфона, так как большинство смартфонов оснащены различными датчиками, а в частности акселерометром, гироскопом, удобны для человека из-за постоянного присутствия смартфона в его жизни, а также возможна обработка полученных данных непосредственно на смартфоне, как это описано в статье [40].

1.4. Размещение и ориентация датчиков

Выбор места размещения датчиков, таких как акселерометр и гироскоп, является одной из проблем в задаче распознавания видов активности человека, так как разные части тела человека обеспечивают различную чувствительность к видам активности.

В работах по распознаванию видов активности человека [14, 15, 18, 20, 37, 39] датчики размещали на следующих частях тела человека: руки, запястье, грудь, пояс, лодыжки, а также в карманах, расположенных в районе бедра человека.

В статье [20] проводят анализ зависимости точности от размещения датчиков. Лучшая точность обеспечивается при размещении датчиков в карманах (98,85 % и 98,03 %).

В результате проведенного анализа можно сделать вывод о необходимости выбора оптимального положения датчиков в зависимости от практической задачи и видов активности, которые должны быть распознаны.

Датчики, такие как акселерометр и гироскоп, чувствительны к изменениям ориентации в телефоне, которая влияет на результаты распознавания активности. При небольшом изменении ориентации, произойдет значительное изменение сигнала датчика. Это изменение также повлияет на скорость распознавания.

Фиксированная ориентация смартфона ограничивает свободу использования человеком телефона. Это делает систему менее привлекательной и менее практичной для пользователей.

Таким образом, система распознавания активности должна быть инвариантна к ориентации в противном случае, пользователи должны будут разместить датчики в определенной ориентации для успешного распознавания активности.

1.5. Выбор частоты дискретизации

Частота дискретизации играет значительную роль в процессе распознавания активности. Чем выше частота, тем выше точность классификации.

Во многих работах [7, 33, 36] частота дискретизации составляет от 20 Гц до 50 Гц. Низкую частоту дискретизации используют для экономии заряда батареи смартфона. Однако есть ряд работ [21, 32], где частота дискретизации составляет 100 Гц.

В статье [22] проведено исследование по оптимизации частоты дискретизации акселерометров в задаче распознавания видов активности человека. Было выявлено, что частота дискретизации, используемая в исследованиях, на 57 % выше, чем необходимо, что приводит к увеличению неоправданных вычислительных затрат.

В любом случае, частота дискретизации в диапазоне от 2 Гц до 100 Гц подходит для распознавания видов активности человека.

1.6. Предварительная обработка данных и сегментация

Предварительная обработка представляет собой процесс обработки неполных данных, устранения выбросов и нормализации данных с использованием различных методов предварительной обработки, например, фильтра высоких и низких частот.

Также на данном этапе происходит синхронизация данных с различных датчиков. Чтобы избежать взаимодействия с переходными процессами удаляются 10 секунд от начала и конца каждой активности.

Сегментация временного ряда — это процесс разбиения временного ряда на значимые сегменты. Размер окна, которое нужно захватить и обработать, обычно составляет от 1 до 10 секунд, как в работе [27]. Если длина окна становится слишком большой, вероятность того, что данное окно содержит более одного вида активности, увеличивается. Если длина окна слишком маленькая, то возникают проблемы связанные с повышением вычислительной сложности из-за частого запуска алгоритма распознавания. Кроме того, короткие временные окна могут не предоставлять достаточную информацию для полного описания выполняемой активности.

Таблица 2

Извлекаемые признаки

| Статистические | Частотные |
|-------------------------------|--------------------------------------|
| Среднее значение | Угловая скорость |
| Медиана | Быстрое преобразование Фурье |
| Стандартное отклонение | Дискретное косинусное преобразование |
| Пик абсолютных данных | |
| Корреляция между осями | |
| Градиент | |
| Межквартильный диапазон | |
| Среднее абсолютное отклонение | |
| Эксцесс | |
| Асимметрия | |
| Среднеквадратичное значение | |
| Минимальное значение | |
| Максимальное значение | |
| Диапазон значений | |

1.7. Методы извлечения и обработки признаков

Извлечение признаков — это процесс преобразования сегментированных данных временного ряда, специфических для предметной области, в набор признаков. Существует два метода извлечения признаков — статистический метод и преобразование данных в частотную область. В работах по распознаванию видов физической активности [3, 5, 8, 17, 23, 24, 29] извлекают признаки, которые представлены в табл. 2.

Некоторые признаки, вычисляемые на основе обработанного набора данных, могут быть избыточными или неинформативными, что может негативно повлиять на точность распознавания. Для того, чтобы выбрать информативные признаки и снизить вычислительные затраты за счет уменьшения числа признаков применяют методы отбора признаков, указанные в работах [8, 9, 31, 41].

В работе с помощью метода Relief-F вычисляется и нормализуется вектор весов признаков, а затем отбираются признаки, вес которых превышает значение заданного порога.

Метод Correlation-based Feature Selection (CFS) сочетает оценочную формулу с соответствующей корреляционной мерой и эвристической стратегией поиска.

Метод Fast Correlation Based Filter начинает работать с полным множеством признаков, использует меру симметричной неопределенности для определения зависимостей между признаками и позволяет выбрать подмножество путем поиска и последовательного исключения малоинформативных признаков.

Метод Sequential forward feature selection (SFFS) на каждой итерации добавляет к набору признак, обеспечивающий наилучшую для данной итерации эффективность распознавания.

Генетический алгоритм моделирует процесс естественного отбора: получает последующее поколение признаков методом скрещивания, мутации и селекции и останавливает свою работу, если последующее поколение не улучшило максимальный из результатов.

Метод взаимной информации (Mutual Information) определяет нелинейную корреляционную зависимость взамен вычисления корреляции Пирсона «признак-признак» и «признак-метка».

1.8. Методы классификации

В работах по распознаванию видов физической активности применяются следующие методы классификации: деревья решений, метод k-ближайших соседей, метод опорных векторов, методы глубокого обучения. Достоинства и недостатки данных методов, указанные в работе [1], представлены в табл. 3.

Исходя из данной таблицы и результатов исследований, приведенных в работе [2], для распознавания видов физической активности лучше использовать метод опорных векторов. Метод опорных векторов позволяет проводить обучение в условиях поступления данных в реальном времени, обеспечивает уверенную классификацию и работает по малой выборке обучающих данных.

Достоинства и недостатки методов классификации

| Метод | Достоинства | Недостатки |
|---------------------------|--|--|
| Деревья решений | Простая реализация, интерпретация и отсутствие подготовки данных для их дальнейшего использования. Работа с категориальными и интервальными переменными. Работа с большим объемом информации | Отсутствие оптимальности дерева решений в целом, необходимость регулировки его длины. Переизбыток данных и плохая читабельность |
| Метод k-ближайших соседей | Простота реализации, работа с большим количеством данных | Необходимость начального определения точного числа классов. Предполагает сравнение нового неизвестного объекта с объектами из уже категоризованной выборки, что требует выполнения линейного числа операций |
| Метод опорных векторов | Единственное верное решение, нахождение максимальной ширины полосы разделения, вследствие чего производится уверенная классификация. За счет использования ядра описывающего связь между элементами выборки, можно использовать гиперплоскости разной сложности | Большая чувствительность к шумам, стандартизации исходных данных, отсутствие общего подхода к автоматическому выбору ядра в случае линейно неразделимости классов. Низкая скорость обучения |
| Методы глубокого обучения | Можно легко обновлять новыми данными с помощью пакетного распространения. Архитектура может быть адаптирована ко многим типам проблем, а скрытые слои уменьшают потребность в проектировании признаков | Требуют очень большого объема данных, значительных вычислительных ресурсов и большого опыта для настройки |

1.9. Метрики качества обучения

Для оценки качества классификации используется следующий ряд метрик: доля правильных ответов метода (Accuracy), точность (Precision), полнота (Recall), F-мера (F_measure) и ROC-кривая (Receiver Operating Characteristic).

Матрица ошибок — это способ разбить объекты на четыре категории в зависимости от комбинации ответа алгоритма. Для расчета значений данных метрик используются следующие элементы матрицы ошибок:

- True Positives (TP) — количество положительных экземпляров, которые были классифицированы как положительные.
- True Negatives (TN) — количество отрицательных экземпляров, которые были классифицированы как отрицательные.
- False Positives (FP) — количество отрицательных экземпляров, которые были классифицированы как положительные.
- False Negatives (FN) — количество положительных экземпляров, которые были классифицированы как отрицательные.

Доля правильных ответов (accuracy) — одна из самых простейших оценок, так как дает некорректный результат в задачах с неравными классами.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Точность системы в пределах класса — это доля объектов действительно принадлежащих данному классу относительно всех объектов, которые система отнесла к этому классу. Точность, которую часто называют положительной прогностической величиной, представляет собой отношение правильно классифицированных положительных случаев к общему числу случаев, классифицированных как положительные.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Полнота системы — это доля найденных классификатором объектов, принадлежащих классу, относительно всех объектов этого класса в тестовой выборке.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

$$F_measure = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (4)$$

Кривая ошибок или ROC-кривая — графическая характеристика качества классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций.

AUC-ROC (или ROC AUC) — площадь (Area Under Curve) под кривой ошибок. Данная кривая представляет собой линию от (0,0) до (1,1) в координатах True Positive Rate (TPR) и False Positive Rate (FPR).

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

В исследованиях по распознаванию видов физической активности человека для оценки работы метода классификации наиболее часто применяют кривую ошибок.

2. Обзор практических задач, требующих применения метода распознавания физической активности человека

2.1. Фитнес

Для того чтобы предотвратить заболевания все здоровые люди нуждаются в выполнении физической активности в течение 30 минут в течение пяти дней в неделю. Для удобства отслеживания выполняемых видов физической активности создаются приложения-трекеры.

Одной из значимых областей применения распознавания видов активности человека является мониторинг уровня его физической активности. В качестве примера можно рассмотреть систему Fitbit, описанную в работе [13], которая представляет собой небольшое устройство, содержащее 3D-датчик движения. SmartTrack автоматически распознает и записывает активность, а также фиксирует статистические данные: продолжительность активности, сожженные калории и частоту сердечного ритма. Данная система позволяет распознать следующие виды активности: ходьба, бег, танцы, езда на велосипеде, теннис, баскетбол, футбол, плавание. По умолчанию SmartTrack распознает активность, только когда человек выполнял ее в течение 15 минут.

Помимо приложений, контролирующих физическую активность человека в целом, существуют приложения мониторинга конкретных видов спорта.

В статье [4] анализируют прыжки с трамплина из данных датчика ускорения расположенного на теле спортсмена. С датчиками, прикрепленными к ногам, рукам и груди спортсмена, авторы могли распознавать характерные особенности движения и извлекать биомеханические описательные параметры.

2.2. Медицина

В области медицины приложения для распознавания физической активности также играют важную роль, так как позволяют дистанционно контролировать корректность выполнения рекомендаций или плана, назначенного врачом, тем самым сокращая время пребывания пациента в больнице.

После перенесенной болезни пациенту необходимо пройти реабилитацию. Физическая телереабилитация подразумевает дистанционный мониторинг выполнения пациентом физических упражнений с целью восстановления здоровья, физического состояния и его трудоспособности.

Продукт VirtualRehab Body [42] это комплекс лечебных модулей для реабилитации верхних и нижних конечностей тела. Он используется по всему миру в стационарных, амбулаторных и домашних условиях.

Комплекс включает в себя три модуля: оценки, фитнес-игры и упражнения. Для обнаружения движений используется единственная камера. Отсутствие необходимости прикреплять датчики к пациенту обеспечивает быструю настройку и простоту использования.

В работах [11, 19] разработан прототип контекстной системы напоминания для выполнения ежедневной деятельности. Эта система помогает пожилым людям с деменцией самостоятельно выполнять различную ежедневную активность, вместо того, чтобы полагаться на опекунов.

2.3. Производство и услуги

Факт нахождения человека на рабочем месте не подразумевает того, что работник эффективно выполняет свои обязанности. Распознавание видов активности в сфере производства и услуг позволит контролировать эффективности работы сотрудников, выявлять неloyальных сотрудников, некачественное выполнение работы и мошеннические схемы. На основе полученных данных руководители организаций и предприятий смогут своевременно принимать меры по оптимизации работы персонала и повысить эффективность производственных процессов.

В статье [35] распознают активность работников на интеллектуальном производстве с использованием сигналов инерциальных датчиков. На основе полученных данных распознаются 6 видов активности в задачах сборки.

В статье [12] демонстрируются возможности системы распознавания видов активности для мониторинга персонала больницы.

2.4. Безопасность

Еще одной важной областью, в которой применяется распознавание видов физической активности, является выявление потенциально опасных ситуаций в жизни человека.

В статье [26] распознают виды активности для автоматической генерации отчетов после патрулирования, чтобы провести анализ происходящего во время патрулирования в течение нескольких часов.

В работе [30] представлен алгоритм обнаружения падения человека. Этот алгоритм извлекает признаки только в том случае, если человек находится в активном состоянии. Обнаружив падение человека, система может оповестить родственников или персонал больницы об опасной ситуации.

В работе [6] представлена система распознавания агрессивного поведения человека с использованием одной или нескольких камер. Данная система распознает акты вандализма и боевые действия. Подобные системы применяются для распознавания агрессивного поведения в метро, аэропортах, банках, тюрьме.

Заключение

Таким образом, в задачах распознавания видов физической активности выделяют два вида активностей, которые подлежат распознаванию: активности низкого уровня и высокого уровня.

Для получения данных, необходимых для распознавания видов физической активности человека, следует применять сочетание датчиков движения, физиологических датчиков, а также датчиков окружающих предметов.

Система распознавания видов активности должна быть инвариантна к ориентации и положению датчиков, в противном случае, пользователи должны будут разместить датчики в определенной позиции и ориентации для успешного распознавания видов активности.

Для распознавания видов активности целесообразно использовать частоту дискретизации в диапазоне от 2 Гц до 50 Гц. Полученные данные с датчиков необходимо предварительно обработать с целью обработки неполных данных, устранения выбросов и нор-

мализации данных. Размер окна, которое нужно захватить и обработать, обычно составляет от 1 до 10 секунд.

Для того, чтобы повысить точность классификации, из полученных данных извлекают признаки. Существует два метода извлечения признаков – статистический метод и преобразование данных в частотную область.

Для того, чтобы выбрать информативные признаки и снизить вычислительные затраты за счет уменьшения числа признаков применяют методы отбора признаков.

В качестве метода классификации при решении задачи распознавания видов физической активности человека с носимых датчиков предлагается использовать метод опорных векторов. Метод опорных векторов позволяет проводить обучение в условиях поступления данных в реальном времени, обеспечивает уверенную классификацию и работает по малой выборке обучающих данных. Для оценки работы метода классификации предлагается использовать кривую ошибок.

Таким образом, проведя обзор подходов и практических областей применения распознавания видов физической активности человека, были определены основные проблемы и направления будущих исследований:

1. Данные обучающей выборки в аналогичных исследованиях собираются в лабораторных условиях, из-за этого в реальных условиях жизни виды активности распознаются с низкой точностью.
2. Высокая вычислительная сложность распознавания из-за расчета лишних неинформативных признаков.
3. Низкая точность распознавания видов активности высокого уровня.
4. Использование большого количества датчиков, которые не способствуют распознаванию видов физической активности человека.

Литература/References

1. Баев Н.О. Использование метода опорных векторов в задачах классификации // Международный журнал информационных технологий и энергоэффективности. 2017. Т. 2, № 2(4). С. 17–21. [Baev N.O. Support Vector Machines in Image Classification. *Mezhdunarodnyy zhurnal informatsionnykh tekhnologiy i energoeffektivnosti* [International Journal of Information Technologies and Energy Efficiency]. 2017. no. 2(4). pp. 17–21. (in Russian).]
2. Тарантова Е.С., Макаров К.В. Выбор признаков и метода классификации видов физической активности в задаче построения телереабилитационной системы // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, Информатика. Медицинское приборостроение. 2018. Т. 8, №1(26). С. 54–62. [Tarantova E.S., Makarov K.V. Feature Selection and Choosing the Method for Physical Activity Classification in the Task of Building the Telerehabilitation System. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravleniye, Vychislitel'naya tekhnika, Informatika. Meditsinskoye priborostroyeniye* [News of South-West State University. Series: Management, Computer Engineering, Computer Science. Medical Instrument]. 2018. vol. 8, no. 1(26). pp. 54–62. (in Russian).]
3. Altun K., Barshan B. Human Activity Recognition Using Inertial/Magnetic Sensor Units // Springer-Verlag Berlin Heidelberg. 2010. P. 38–51. DOI: 10.1007/978-3-642-14715-9_5.

4. Bächlin M., Kusserow M, Tröster G., et al. Ski Jump Analysis of an Olympic Champion with Wearable Acceleration Sensors // International Symposium on Wearable Computers (ISWC), Oct. 10–13, 2010, Seoul, South Korea. DOI: 10.1109/iswc.2010.5665851.
5. Bayat A., Pomplun M., Tran D.A. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones // The 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops. Procedia Computer Science, Aug. 17–20, 2014, Niagara Falls, Canada. Vol. 34. P. 450–457. DOI: 10.1016/j.procs.2014.07.009.
6. Brémond F., Thonnat M., Zúñiga M. Video-understanding Framework for Automatic Behavior Recognition // Behavior Research Methods. 2006. Vol. 38. P. 416–426. DOI: 10.3758/bf03192795.
7. Bulling A., Blanke U., Schiele B. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors // ACM Computing Surveys (CSUR). 2014. Vol. 46. P. 1–33. DOI: 10.1145/2499621.
8. Capela N.A., Lemaire E.D., Baddour N. Feature Selection for Wearable Smartphone-Based Human Activity Recognition with Able Bodied, Elderly, and Stroke Patients // PLoS ONE. 2015. Vol. 10. DOI: 10.1371/journal.pone.0124414.
9. Chandrashekar G., Sahin F. A Survey on Feature Selection Methods // Computers and Electrical Engineering. 2014. Vol. 40. P. 16–28. DOI: 10.1016/j.compeleceng.2013.11.024.
10. Dernbach S., Das B., Krishnan N.C., et al. Simple and Complex Activity Recognition through Smart Phones // 2012 Eighth International Conference on Intelligent Environments, Jun. 26–29 2012, Guanajuato, Mexico. DOI: 10.1109/IE.2012.39.
11. Du K., Zhang D., Musa M.W., et al. Handling Activity Conflicts in Reminding System for Elders with Dementia // 2008 Second International Conference on Future Generation Communication and Networking, Dec. 13–15, 2008, Hainan Island, China. P. 416–421. DOI: 10.1109/FGCN.2008.117.
12. Favela J., Tentory M., Castro L.A., et al. Activity Recognition for Context-aware Hospital Applications: Issues and Opportunities for the Deployment of Pervasive Networks // Mobile Networks and Applications. 2007. P. 155–171. DOI: 10.1007/s11036-007-0013-5.
13. Fitbit Inc. Fitbit Official Site for Activity Trackers & More // Fitbit. URL: <https://www.fitbit.com/home> (дата обращения: 30.10.2018).
14. Frank K., Röckl M., Nadas M.J.V., et al. Comparison of Exact Static and Dynamic Bayesian Context Inference Methods for Activity Recognition // 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), March 29–April 2, 2010, Mannheim, Germany. P. 189–195. DOI: 10.1109/percomw.2010.5470671.
15. Gao L., Bourke A.K., Nelson J. Evaluation of Accelerometer Based Multi-sensor Versus Single-sensor Activity Recognition Systems // Medical Engineering and Physics. 2014. Vol. 36. P. 779–785. DOI: 10.1016/j.medengphy.2014.02.012.
16. Guo J., Zhou X., Sun Y., et al. Smartphone-Based Patients' Activity Recognition by Using a Self-Learning Scheme for Medical Monitoring // Journal of Medical Systems. 2016. Vol. 40. P. 140. DOI: 10.1007/s10916-016-0497-2.

17. He Y., Li Y. Physical Activity Recognition Utilizing the Built-in Kinematic Sensors of a Smartphone // *International Journal of Distributed Sensor Networks*. 2013. Vol. 4. DOI: 10.1155/2013/481580.
18. He Z-Y., Jin L-W. Activity Recognition from Acceleration Data Using AR Model Representation and SVM // *2008 International Conference on Machine Learning and Cybernetics*, Jul. 12–15, 2008, Kunming, China. DOI: 10.1109/icmlc.2008.4620779.
19. Hua S., Kim S.J., Kawanishi N., et al. A Context-aware Reminding System for Daily Activities of Dementia Patients // *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, Jun. 22–29, 2007, Toronto, Ont., Canada. DOI: 10.1109/icdcs.2007.8.
20. Janidarmian M., Fekr A.R., Radecka K., et al. A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition // *Sensors*. 2017. Vol. 17. DOI: 10.3390/s17030529.
21. Kao T.P., Lin C.W., Wang J.S. Development of a Portable Activity Detector for Daily Activity Recognition // *2009 IEEE International Symposium on Industrial Electronics (ISIE 2009)*, Jul. 5–8, 2009 Seoul, South Korea. P. 115–120. DOI: 10.1109/isie.2009.5222001.
22. Khan A., Hammerla N., Mellor S., et al. Optimising Sampling Rates for Accelerometer-Based Human Activity Recognition // *Pattern Recognition Letters*. 2016. P. 33–40. DOI: 10.1016/j.patrec.2016.01.001.
23. Khan A.M., Lee Y-K., Lee S.Y., et al. A Triaxial Accelerometer-based Physical-activity Recognition via Augmented-signal Features and a Hierarchical Recognizer // *IEEE Transactions on Information Technology in Biomedicine*. 2010. Vol. 14. P. 1166–1172. DOI: 10.1109/titb.2010.2051955.
24. Khan A.M., Siddiqi M.H., Lee S.W. Exploratory Data Analysis of Acceleration Signals to Select Light-weight and Accurate Features for Real-time Activity Recognition on Smartphones // *Sensors*. 2013. Vol. 13. P. 13099–13122. DOI: 10.3390/s131013099.
25. Lara O.D., Labrador M.A. A Survey on Human Activity Recognition Using Wearable Sensors // *IEEE Communications Surveys and Tutorials*. 2013. Vol. 15. P. 1192–1209. DOI: 10.1109/surv.2012.110112.00192.
26. Minnen D., Westeyn T., Ashbrook D., et al. Recognizing Soldier Activities in the Field // *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*. March 26–28, 2007, RWTH Aachen University, Germany. P. 236–241. DOI: 10.1007/978-3-540-70994-7_40.
27. Morales J., Akopian D. Physical Activity Recognition by Smartphones, a Survey // *Bio-cybernetics and Biomedical Engineering*. 2017. Vol. 37. P. 388–400. DOI: 10.1016/j.bbe.2017.04.004.
28. Murao K., Terada T. A Recognition Method for Combined Activities with Accelerometers // *UbiComp '14 Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, Sept. 13–17, 2014, Seattle, Washington. P. 787–796. DOI: 10.1145/2638728.2641304.
29. Ortiz J.L.R. Smartphone-based Human Activity Recognition. SPRINGER, 2015. 133 p.
30. Putra I.P.E.S., Brusey J., Gaura E., et al. An Event-triggered Machine Learning Approach for Accelerometer-based Fall Detection // *Sensors*. 2017. Vol. 18. P. 1–18. DOI: 10.3390/s18010020.

31. Reiss A. Personalized Mobile Physical Activity Monitoring for Everyday Life, 2014. 176 p.
32. Reiss A., Stricker D. Creating and Benchmarking a New Dataset for Physical Activity Monitoring // PETRA '12 Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, Jun. 6–8, 2012, Heraklion, Crete, Greece. DOI: 10.1145/2413097.2413148.
33. Siirtola P., Rönning J. Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data // International Journal of Interactive Multimedia and Artificial Intelligence. 2012. Vol. 1. P. 38. DOI: 10.9781/ijimai.2012.155.
34. Tam D., Huynh G. Human Activity Recognition with Wearable Sensors // TU Darmstadt. 2008. 147 p.
35. Tao W., Lai Z-H., Leu M.C., et al. Worker Activity Recognition in Smart Manufacturing Using IMU and sEMG Signals with Convolutional Neural Networks // 46th SME North American Manufacturing Research Conference, Jun. 18–22, 2018, Texas, USA. P. 1159–1166. DOI: 10.1016/j.promfg.2018.07.152.
36. Twomey N., Diethe T., Kull M., et al. The SPHERE Challenge Activity Recognition with Multimodal Sensor Data. 2016. 14 p.
37. Uddin M., Salem A., Nam I., et al. Wearable Sensing Framework for Human Activity Monitoring // WearSys '15 Proceedings of the 2015 Workshop on Wearable Systems and Applications. May 18, 2015, Florence, Italy. P. 21–26. DOI: 10.1145/2753509.2753513.
38. Walse K., Dharaskar R.V. A Survey on Human Activity Recognition Using Smartphone // International Journal of Advance Research in Computer Science and Management Studies. 2017. Vol. 5. P. 118–125.
39. Yang A.Y., Jafari R., Sastry S.S., et al. Distributed Recognition of Human Actions Using Wearable Motion Sensor Networks // Journal of Ambient Intelligence and Smart Environments. 2009. Vol. 1. P. 103–115.
40. Yu H., Cang S., Wang Y. A Review of Sensor Selection, Sensor Devices and Sensor Deployment for Wearable Sensor-based Human Activity Recognition Systems // 2016 10th Int. Conf. on Software, Knowledge, Information Management & Applications (SKIMA), Dec. 15–17, 2016, Chengdu, China. P. 250–257. DOI: 10.1109/skima.2016.7916228.
41. Zhang M., Sawchuk A.A. Feature Selection-Based Framework for Human Activity Recognition Using Wearable Multimodal Sensors // 6th International ICST Conference on Body Area Networks. 12 Jun. 2012. DOI: 10.4108/icst.bodynets.2011.247018.
42. VirtualRehab Body — Upper and Lower Extremity Therapy for Various Neuromotor Impairments. URL: https://evolvrehab.com/virtualrehab/virtualrehab_body/ (дата обращения: 14.01.2019).

Тарантова Елена Сергеевна, магистрант, кафедра физики и прикладной математики, Муромский институт (филиал) ФГБОУ ВО «Владимирский государственный университет имени А.Г. и Н.Г. Столетовых» (Муром, Российская Федерация)

Макаров Кирилл Владимирович, к.т.н., доцент, кафедра физики и прикладной математики, Муромский институт (филиал) ФГБОУ ВО «Владимирский государственный университет имени А.Г. и Н.Г. Столетовых» (Муром, Российская Федерация)

Орлов Алексей Александрович, д.т.н., доцент, кафедра физики и прикладной математики, Муромский институт (филиал) ФГБОУ ВО «Владимирский государственный университет имени А.Г. и Н.Г. Столетовых» (Муром, Российская Федерация)

SURVEY ON APPROACHES AND PRACTICAL AREAS OF HUMAN ACTIVITY RECOGNITION APPLICATION

© 2019 E.S. Tarantova, K.V. Makarov, A.A. Orlov

Murom Institute (branch) Federal State Budgetary

*Educational Institution of Higher Education "Vladimir State University named after
Alexander and Nikolay Stoletovs"*

(st. Orlovskaya 23, Murom, 602264 Russia)

E-mail: elena.tarantova@yandex.ru, k-distance@rambler.ru, AlexeyAlexOrlov@gmail.com

Received: 21.01.2019

Human activity recognition is one of the relevant fields of research in machine learning since recognition results are necessary for solving many practical problems. The article provides a survey on approaches and areas of practical application of methods for human activity recognition. The sensors used for human activity recognition are considered, and the criteria for their selection are presented. Possible solutions to the problem of choosing the location and orientation of wearable sensors are presented. The main stages of human activity recognition are discussed in the article. Extracted features and methods of their selection to increase the accurate classification of human activity recognition and reduce computational complexity by cutting down the number of features are presented. The advantages and disadvantages of popular classification methods are formulated. The metrics used to evaluate the quality of learning classification models are considered. The most commonly used quality metric is the error curve. Practical tasks in which the results of human activity recognition are needed are also presented. The main areas of human activity recognition application are medicine, manufacturing, fitness, and safety of people. In conclusion, possible future research directions are presented.

Keywords: pattern recognition, machine learning, types of human physical activity.

FOR CITATION

Tarantova E.S., Makarov K.V., Orlov A.A. Survey on Approaches and Practical Areas of Human Activity Recognition Application. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. vol. 8, no. 3. pp. 43–57. (in Russian) DOI: 10.14529/cmse190303.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

ОБЗОР МОДЕЛЕЙ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

© 2019 Н.А. Ежова, Л.Б. Соколинский

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: EzhovaNA@susu.ru, Leonid.Sokolinsky@susu.ru

Поступила в редакцию: 01.06.2019

Цель данного обзора — дать максимально полное представление о достижениях и современном состоянии дел в разработке аналитических моделей параллельных вычислений, позволяющих предсказать время вычислений, ускорение, эффективность и масштабируемость параллельных алгоритмов применительно к различным целевым многопроцессорным платформам. Важность моделей параллельных вычислений вытекает из того, что они до реализации параллельного алгоритма в виде программы позволяют понять, насколько эффективно данный алгоритм может использовать конкретную многопроцессорную платформу, и при необходимости внести изменения в дизайн алгоритма, либо рассмотреть вариант замены целевой аппаратной платформы. В обзоре показывается эволюция моделей параллельных вычислений, происходившая одновременно с эволюцией многопроцессорных систем, от одноуровневых моделей с общей памятью до многоуровневых иерархических моделей с распределенной памятью, ориентированных на кластерные вычислительные системы с многоядерными ускорителями. В заключении обзора приводятся рекомендации по выбору возможных направлений дальнейших исследований в области разработки математических моделей параллельных вычислений.

Ключевые слова: модель параллельных вычислений, обзор, параллельное программирование, многопроцессорные системы, оценка производительности, предсказание времени выполнения алгоритма.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Ежова Н.А., Соколинский Л.Б. Обзор моделей параллельных вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 58–91. DOI: 10.14529/cmse190304.

Введение

Модель вычислений представляет собой упрощенное, абстрактное описание компьютера. Проектировщики новых компьютеров, разработчики алгоритмов и программисты могут использовать модель вычислений для оценки своей работы, включая соответствие архитектуры компьютера различным приложениям, вычислительную сложность алгоритма и прогнозируемое быстродействие программы на различных вычислительных системах. Хорошая модель вычислений облегчает работу разработчиков компьютеров, алгоритмов и программ тем, что позволяет адекватно отобразить выбираемое решение на реальные компьютеры [1]. Подобную модель вычислений в литературе называют «соединяющей моделью (*bridging model*)» [2]. Универсальная соединяющая модель (см. рис. 1 а) может быть применима к любым алгоритмам и любым компьютерам. Такой универсальной соединяющей моделью вычислений для последовательных компьютеров и алгоритмов явились архитектура фон Неймана [3] и модель RAM (Random Access Machine) [4–8]. При появлении параллельных компьютеров были сделаны многочисленные попытки построить такую же универсальную соединяющую модель для параллельных алгоритмов [9], однако эти попытки потерпели неудачу. Это объясняется, главным образом, большим разнообразием многопроцессорных архитектур, появляющихся и развивающихся в настоящее время в погоне за повышением производительности.

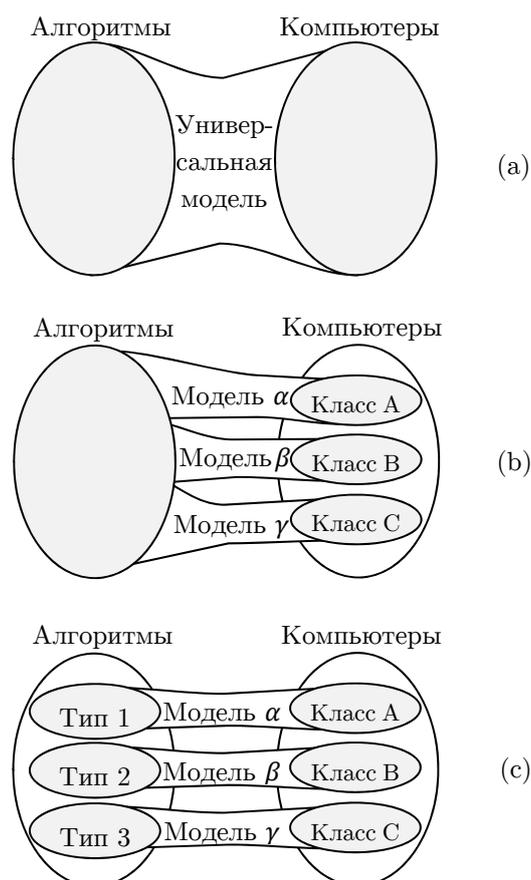


Рис. 1. Модели, соединяющие алгоритмы и компьютеры

В этих условиях создание простой, адекватной и универсальной модели параллельных вычислений является практически неразрешимой задачей. Для преодоления возникших трудностей был применен подход, схематично изображенный на рис. 1 б, в соответствии с которым многопроцессорные архитектуры были разделены на три класса: архитектуры с общей памятью, архитектуры с распределенной памятью и иерархические многопроцессорные архитектуры [1]. Для каждой из таких архитектур строились отдельные модели параллельных вычислений, которые, однако, оставались универсальными по отношению к множеству параллельных алгоритмов. Подобный подход позволил построить простые и универсальные модели с высоким уровнем абстракции, такие, как PRAM [10], BSP [2], LogP [11]. В целях адаптации таких моделей ко все более усложняющимся архитектурам многопроцессорных систем были предприняты многочисленные попытки их уточнения и расширения, что привело к появлению адекватных, но сложных для практического применения моделей параллельных вычислений (см., например, [12–17]). Исправить ситуацию в определенной мере позволяет подход, предполагающий разбиение всего множества алгоритмов по типам. Примерами различных типов алгоритмов могут быть итерационные численные алгоритмы, алгоритмы на графах, алгоритмы обработки больших данных и так далее. Для каждой пары (тип алгоритма, класс архитектуры) строится своя модель параллельных вычислений. Подобный подход позволяет достичь приемлемого компромисса между точностью оценок и простотой использования. В качестве примера можно привести модель параллельных вычислений BSF [18, 19], ориентированную на итерационные алгоритмы с высокой вычислительной сложностью и большие многопроцессорные системы кластерного типа.



Рис. 2. Классификация моделей параллельных вычислений

Цель этой статьи — показать современное состояние дел в области разработки моделей параллельных вычислений и дать обзор наиболее важных и интересных моделей, разработанных к настоящему моменту. В обзоре используется классификация моделей, изображенная на рис. 2. Все модели параллельных вычислений делятся на два класса: модели для многопроцессорных систем с общей памятью и модели для многопроцессорных систем с распределенной памятью. Каждый из этих классов, в свою очередь, делится на два подкласса: одноуровневые модели и многоуровневые модели.

Модели параллельных вычислений с общей памятью ориентированы на многопроцессорные архитектуры класса SMP и отчасти класса NUMA. Архитектура SMP (Symmetric MultiProcessor) — объединяет в себе симметричные мультипроцессоры, являющиеся наиболее распространенным подклассом многопроцессорных систем с общей памятью [20]. Отличительными особенностями SMP систем является то, что все процессоры, входящие в систему, являются одинаковыми, и каждый процессор имеет одно и то же время доступа к любой ячейке оперативной памяти. Наиболее популярными представителями класса SMP являются современные серверные системы, построенные на многоядерных процессорах [21]. Многопроцессорные системы с архитектурой NUMA (Non-Uniform Memory Access) [22] также имеют общую оперативную память, но в отличие от SMP время доступа процессора к различным ячейкам памяти может существенно различаться. В настоящее время многопроцессорные системы класса NUMA практически исчезли с рынка.

Модели параллельных вычислений с распределенной памятью ориентированы на масово-параллельные [23] и кластерные архитектуры [24]. В простейшем случае многопроцессорная система с распределенной памятью представляет собой совокупность однопроцессорных серверов, объединенных высокоскоростной соединительной сетью. Именно такую структуру имели первые кластеры класса «беовульф» [25]. Однако, современные кластерные вычислительные системы имеют существенно более сложную структуру, включающую в себя SMP-узлы, внутри которых наряду с центральными многоядерными процессорами могут быть установлены дополнительные многоядерные ускорители, например, графические процессорные устройства (ГПУ) [26]. В этом случае может быть применено *двухуровневое* моделирование: сначала на уровне SMP-узла используется вычислительная модель для общей памяти, а затем на уровне кластерной системы в целом используется вычислительная модель для распределенной памяти.

Модели параллельных вычислений с многоуровневой (иерархической) памятью [27] ориентированы на современные SMP-системы, в которых кроме основной памяти имеются несколько уровней сверхоперативной кэш-памяти, встраиваемой в процессоры [28, 29]. Сюда же можно отнести и новый вид быстрой энергонезависимой памяти Intel Optane на базе технологии 3D XPoint [30], занимающей промежуточное положение между оперативной памятью и накопителями класса SSD. В общем случае модель многопроцессорной архитектуры с многоуровневой (иерархической) памятью можно определить как однородную систему, включающую в себя несколько уровней памяти с различным временем доступа. При этом определенные уровни памяти могут быть приватными для отдельных процессоров или их групп (в случае многоядерных сокетов под процессором понимается отдельное процессорное ядро). Модели параллельных вычислений для систем с многоуровневой памятью являются наиболее адекватными для приложений с интенсивным обменом данными между различными уровнями иерархической памяти. В качестве целевых аппаратных платформ для таких моделей часто фигурируют многоядерные кластеры с ГПУ, используемыми как ускорители, для которых характерно гибридное программирование с одновременным использованием технологий параллельных вычислений CUDA, OpenMP и MPI [31].

Обзор организован следующим образом. В разделе 1 дается определение модели вычислений и обсуждаются требования к качественной модели параллельных вычислений. В разделе 2 дается обзор одноуровневых моделей для многопроцессорных систем с общей памятью. Раздел 3 посвящен обзору многоуровневых моделей для многопроцессорных систем с общей памятью. В разделе 4 рассматриваются одноуровневые модели для многопроцессорных систем с распределенной памятью. В разделе 5 представлены многоуровневые модели для многопроцессорных систем с распределенной памятью. В разделе 6 обсуждаются близкие работы, являющиеся современными и более ранними обзорами моделей параллельных вычислений. Заключение содержит выводы и рекомендации относительно направления дальнейших исследований в области разработки новых моделей параллельных вычислений.

1. Требования к модели параллельных вычислений

Модель параллельных вычислений в общем случае должна включать в себя следующие пять компонентов, некоторые из которых в определенных случаях могут быть тривиальны [32]: *архитектурный компонент*, описываемый как помеченный граф, узлы которого соответствуют модулям с различной функциональностью, а дуги — межмодульным соединениям для передачи данных; *спецификационный компонент*, определяющий, что есть (синтаксически) корректный алгоритм/программа; *компонент выполнения*, задающий последовательность состояний архитектурных модулей, обеспечивающих корректное выполнение алгоритма/программы для определенных входных данных; *распараллеливающий компонент*, определяющий способы распределения вычислений, синхронизации и обмена данными между процессорными модулями, работающими независимо (параллельно); *стоимостный компонент*, определяющий одну или несколько метрик, позволяющих предсказать параметры выполнения алгоритма/программы (время выполнения, объем необходимой оперативной памяти и др.) в каждом конкретном случае.

В ранней работе Скилликорна [33] к модели параллельных вычислений предъявляются следующие требования: архитектурная независимость, согласованность, дескриптивная простота. *Архитектурная независимость* модели означает, что модель применима к

широкому классу многопроцессорных архитектур. Программа, написанная согласно модели, не должна требовать переписывания при портировании на другую архитектуру, необходима только перекомпиляция. *Согласованность* предполагает, что время, предсказываемое стоимостными метриками модели для данной архитектуры, должно (асимптотически) соответствовать реальному времени выполнения программы. *Дескриптивная простота* предполагает, что модель абстрагируется от низкоуровневых механизмов параллельной обработки, межпроцессорных коммуникаций и синхронизации процессов, что обеспечивает легкость ее применения при практическом программировании.

В качестве наиболее важных свойств модели параллельных вычислений в современной литературе выделяют следующие [34, 35].

- *Юзабилити*, определяющая легкость описания алгоритма и анализа его стоимости средствами модели (модель должна быть легкой в использовании).
- *Переносимость*, характеризующая широту класса целевых платформ, для которых модель оказывается применимой.
- *Предиктивность*, выражающаяся в возможности с помощью стоимостных метрик модели предсказать *временные характеристики* выполнения алгоритмов/программ на целевой вычислительной системе.

Основными *временными характеристиками* являются следующие:

- 1) *реальное (астрономическое) время* выполнения программы;
- 2) *абстрактное время* выполнения программы, позволяющее предсказать, какая из двух программ будет выполняться быстрее;
- 3) *масштабируемость* алгоритма/программы, определяющая максимальное количество вычислительных модулей в многопроцессорной системе, после которого прирост ускорения становится нулевым или отрицательным.

2. Одноуровневые модели с общей памятью

Модель параллельных вычислений PRAM (Parallel Random Access Machine) [36] предполагает, что все процессоры работают синхронно под управлением одного тактового генератора и имеют произвольный доступ к общему адресному пространству оперативной памяти. Каждый процессор может выполнить арифметическую операцию, логическую операцию или операцию доступа к памяти за один машинный такт. Модель PRAM появилась в конце 1970-х годов как естественное расширение модели RAM (Random Access Machine) и была описана сразу в нескольких работах, среди которых можно выделить [10, 37, 38]. Модель PRAM получила широкое распространение и интенсивно использовалась при проектировании алгоритмов [39].

PRAM-компьютер (см. рис. 3) состоит из K процессоров, соединенных с общей памятью неограниченного размера. Каждый процессор имеет свой уникальный номер. Предполагается, что доступ к любой ячейке памяти происходит за один машинный такт. В общем случае каждый процессор может выполнять свою собственную программу, однако на практике большинство *PRAM-алгоритмов* реализуются по схеме SPMD (Single Program Multiple Data) [40, 41], предполагающей, что все процессоры выполняют одну и ту же программу. На каждом машинном такте конкретный процессор может находиться либо в состоянии простоя, либо в активном состоянии. При этом все процессоры, находящиеся в активном состоянии, выполняют одну и ту же команду, но над разными данными.

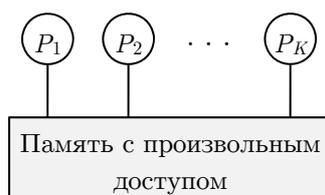


Рис. 3. PRAM-компьютер

Различают следующие три варианта модели PRAM в зависимости от политики разрешения конфликтов при параллельном доступе нескольких процессоров к одной и той же ячейке памяти в течении одного такта.

- EREW (Exclusive Read Exclusive Write) PRAM модель запрещает параллельный доступ к ячейке памяти и по чтению, и по записи.
- CREW (Concurrent Read Exclusive Write) PRAM модель разрешает параллельные чтения значения из ячейки, но запрещает параллельную запись в одну и ту же ячейку памяти.
- CRCW (Concurrent Read Concurrent Write) PRAM модель разрешает параллельный доступ к ячейке памяти как по чтению, так и по записи.

Указанные варианты модели PRAM характеризуются значительными различиями по времени выполнения параллельных алгоритмов. Рассмотрим следующую задачу: определить, являются ли все элементы двоичного массива $M[1 \dots n]$ нулями. Результатом вычислений в этом случае должно быть значение переменной A , являющееся результатом выполнения логической операции «ИЛИ» для всех элементов массива M . Параллельный CRCW PRAM алгоритм может решить эту задачу на n процессорах следующим образом. Первоначально переменной A присваивается значение 1. Затем все процессоры параллельно тестируют по одному элементу массива таким образом, чтобы индекс элемента совпадал с номером процессора. Если процессор обнаруживает в соответствующей ячейке значение 1, то он присваивает переменной A значение 0. Очевидно, что описанный CRCW PRAM алгоритм корректно решит поставленную задачу на n процессорах за $O(1)$ машинных тактов. В то же время, CREW PRAM алгоритм в общем случае потребует для решения этой задачи $\Omega(\log n)$ машинных тактов независимо от количества используемых процессоров [42]. С другой стороны, p -процессорная CREW PRAM модель может быть эмулирована с помощью p -процессорной EREW PRAM модели с замедлением не более, чем в $O(\log n)$ раз [43].

В теоретическом аспекте проектирование PRAM алгоритма состоит в разработке стратегии, позволяющей достигнуть максимального уровня параллелизма, при котором задача решается за минимальное количество параллельных шагов, используя разумное количество процессоров. В рамках модели PRAM задача считается хорошо распараллеливаемой, если она может быть решена за $O((\log n)^m)$ параллельных шагов при некотором фиксированном m с использованием полиномиального количества процессоров. Данный класс задач получил название NC и был формально введен в работах [44, 45] применительно к логическим схемам.

Для описания PRAM-алгоритмов удобно использовать *фреймворк* WT (*Work-Time*). В соответствии с WT-фреймворком PRAM-алгоритм представляется в виде последовательности шагов, каждый из которых представлен оператором одного из двух классов:

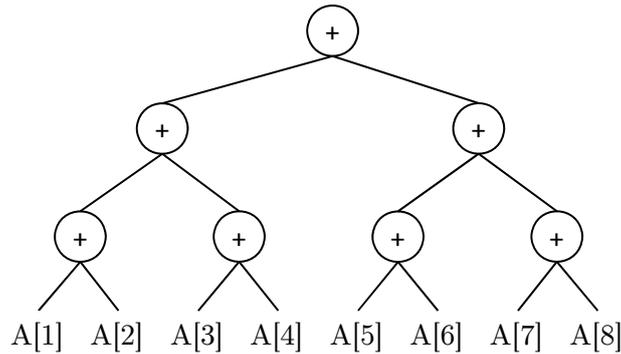


Рис. 4. PRAM-алгоритм, суммирующий элементы массива

последовательные операторы и параллельные операторы. Параллельные операторы специфицируются путем использования конструкций **pardo** и **for all**.

В качестве примера рассмотрим алгоритм PRAM SUM, суммирующий элементы числового массива $A[1..n]$, где $n = 2^l$ при некотором целом $l > 0$. Схема работы алгоритма PRAM SUM представлена на рис. 4. PRAM-алгоритм организуется как сбалансированное бинарное дерево, листьям которого соответствуют элементы массива A , а в качестве внутренних узлов фигурируют операции суммирования значений, представленных дочерними узлами. Алгоритм PRAM SUM выполняет вычисления снизу-вверх, осуществляя суммирование на каждом уровне за один параллельный шаг. При этом, в контексте рис. 4, на первом уровне задействуется четыре процессора, на втором — два, и на третьем — один. WT-реализация алгоритма PRAM SUM приведена в виде алгоритма 1. Из этой реализации видно, что решение состоит из k параллельных шагов. Учитывая, что $n = 2^l$, отсюда получаем $l = \log n$. Следовательно, вычислительная сложность алгоритма составляет $O(\log n)$ машинных тактов. При этом задействуется $n / 2$ процессоров. Таким образом, задача суммирования элементов массива принадлежит классу NC и является хорошо распараллеливаемой в рамках модели PRAM. Очевидно, что границей масштабируемости алгоритма PRAM SUM является $K \leq n / 2$.

Отметим, что в общем случае временная сложность алгоритма в модели PRAM зависит от деталей его реализации, и в этом плане модель PRAM является низкоуровневой. Основным недостатком модели PRAM является то, что она не применима к многопроцессорным системам с распределенной памятью, так как не учитывает расходы на межпроцессорные коммуникации.

Модель PRAM позволяет строить чрезвычайно эффективные параллельные алгоритмы. Однако это достигается ценой следующих серьезных ограничений: все процессоры работают синхронно и межпроцессорные коммуникации выполняются мгновенно. На практике во многих случаях модель PRAM оказывается неадекватной [46]. В соответствии с этим был предложен ряд расширений и улучшений модели PRAM, которые более соответствуют реальным параллельным архитектурам.

Параллельные операции чтения и записи в память, допускаемые моделью PRAM, на реальных компьютерах могут приводить к конфликтам доступа к ячейкам памяти. Для предотвращения массовых конфликтов доступа к памяти в работе [47] была предложена *модель модульно-параллельного компьютера (Module Parallel Computer)*, в соответствии с которой общее адресное пространство делится на m модулей. Каждый

Алгоритм 1 PRAM SUM – параллельное суммирование

Input: Числовой массив $A[1..n]$ размера $n = 2^l$ в оперативной памяти.

Output: Сумма элементов массива A .

```

begin
  d = n
  for j = 1 to l
    d = d/2
    for all  $1 \leq i \leq d$  par do
      A[i] = A[2i-1] + A[2i]
    end for
  end for
  return A[1]
end

```

модуль допускает только одно обращение в течении одного машинного такта. Указанная модель в определенной степени решает проблему конфликтов доступа к памяти, однако она игнорирует аспекты, связанные с пропускной способностью системной шины или соединительной сети. Другое, более реалистичное расширение модели PRAM, получившее название *QRQW PRAM (Queue-Read Queue-Write PRAM)*, было предложено в [48]. Для разрешения конфликтов доступа к памяти в модели QRQW PRAM используются очереди доступа по чтению и записи для каждой ячейки памяти. Время доступа к ячейке памяти прямо пропорционально длине соответствующей очереди. Указанный подход был обобщен в *модели QSM (Queuing Shared Memory)* [49], которая была разработана с целью показать, что модель параллельных вычислений с общей памятью может играть роль соединяющей модели для параллельных алгоритмов. Для этого была выполнена прямая и обратная эмуляция модели QSM средствами BSP и других релевантных моделей.

С развитием архитектур параллельных компьютеров становилось все более очевидным, что доступ к нелокальной памяти оказывает серьезное влияние на производительность. Этот факт попытались учесть авторы *модели BPRAM (Block Parallel Random Access Machine)* [50]. В этой модели в качестве дополнительного параметра вводится латентность и предусматривается блочная передача сообщений между процессорами. При передаче блока данных между двумя процессорами на передачу первого машинного слова отводится l тактов, а на передачу остальных машинных слов, входящих в блок, отводится b тактов на каждое слово, причем b много меньше l . Модель *PRAM(m)* [51] учитывает пропускную способность системной шины путем ограничения количества ячеек памяти, одновременно доступных для параллельных обращений в память. Эта модель базируется на модели CRCW PRAM с тем отличием, что на каждом такте могут быть выполнены не более m обращений к памяти.

Стандартная модель PRAM предполагает, что параллельные процессы автоматически синхронизируются на каждом такте без дополнительных временных затрат. Это допущение не является реалистичным, так как на практике, с одной стороны, не требуется синхронизировать параллельные процессы на каждом машинном такте, и, с другой стороны, синхронизация процессов требует накладных расходов. Для учета первого фактора *модели APRAM* [52] и *Asynchronous PRAM* [53] допускают асинхронное выполнение процессов между двумя последовательными точками синхронизации. В отличие от этого *модель XPRAM* [54] предполагает периодическую глобальную синхронизацию асинхронных процессов. Однако, указанные модели не учитывают накладные расходы, связанные с синхронизацией процессов.

Модель YPRAM [55] базируется на следующих трех характеристиках многопроцессорного компьютера, состоящего из P процессоров: *латентность*, *пропускная неэффективность* и *рекурсивная декомпозиция*. Под *латентностью* $\delta(P)$ понимается максимальное время, необходимое для передачи одного машинного слова между любыми двумя процессорами или между процессором и общей памятью. *Пропускная неэффективность* определяется следующим образом. Пусть массив M из N чисел равномерно распределен между P процессорами. Пусть $\tau(N)$ — максимальное время, требуемое для перестановки элементов массива M среди всех возможных перестановок. Тогда пропускная неэффективность $\beta(P)$ вычисляется по формуле

$$\beta(P) = \lim_{N \rightarrow \infty} \frac{\tau(N)P}{N}. \quad (1)$$

Рекурсивная декомпозиция подразумевает, что многопроцессорный компьютер делится на два субкомпьютера с равным количеством процессоров и равным объемом разделяемой памяти, доступ к которой осуществляется на основе механизма EREW. Каждый субкомпьютер является самостоятельным компьютером и может быть рекурсивно разделен на две части. Для субкомпьютера из S процессоров общее время, затрачиваемое на M чтений из памяти, оценивается как

$$\Theta \left(\lambda(S) + m + \frac{M}{S} \beta(S) \right), \quad (2)$$

где m — максимальное число чтений одного процессора.

Модель HPRAM (Hierarchical PRAM) [56] рассматривает многопроцессорную систему с общей памятью как динамически конфигурируемую иерархию синхронных PRAM-подсистем, работающих асинхронно по отношению друг к другу. Иерархия определяет политику синхронизации между независимыми PRAM-системами. Помимо инструкций, присущих классической модели PRAM, HPRAM включает в себя специальную команду *partition* (*разбиение*), добавляющую в модель управляемый асинхронный параллелизм. Команда *partition* логически разбивает PRAM-(под)систему из P процессоров на непересекающиеся подмножества и назначает каждому из них свой синхронный PRAM-алгоритм. Предусматриваются два варианта HPRAM-системы: *private HPRAM* и *shared HPRAM*. В *private HPRAM* команда *partition* разбивает память на непересекающиеся сегменты пропорционально размерам создаваемых PRAM-подсистем таким образом, что каждая PRAM-подсистема имеет свое собственное адресное пространство, недоступное для других PRAM-подсистем. В отличие от этого в *shared HPRAM* весь объем памяти остается в равной мере доступным для каждой создаваемой PRAM-подсистемы. *HPRAM-алгоритм* представляет собой совокупность PRAM-алгоритмов, организованных в иерархию. Каждая точка в иерархии ассоциируется с соответствующей командой *partition*. Благодаря введению иерархического разделения общей памяти, модели YPRAM и HPRAM в определенной мере позволяют абстрагироваться от низкоуровневых механизмов параллельной обработки, межпроцессорных коммуникаций и синхронизации процессов, и в этом плане частично удовлетворяют требованиям Скилликорна к моделям параллельных вычислений [33], однако это достигается существенным усложнением стоимостных метрик.

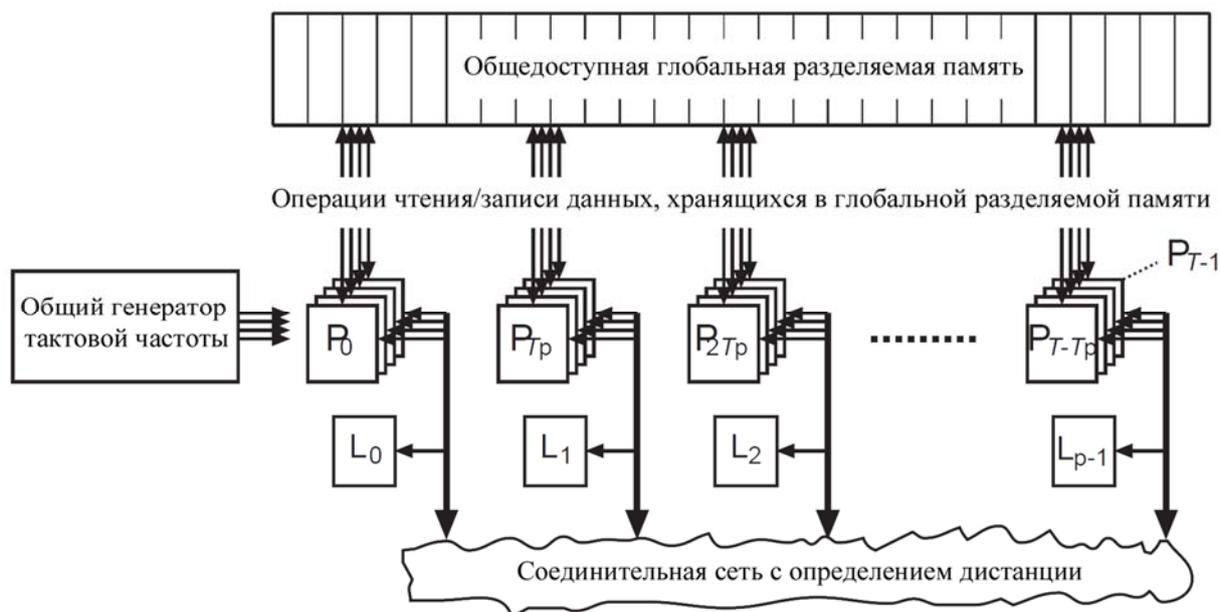


Рис. 5. PRAM-NUMA компьютер (P — процессор, L — локальная память)

В статье [57] предлагается *модель PRAM-NUMA*, являющаяся расширением модели PRAM на многопроцессорные системы с распределенной памятью, работающие в режиме эмуляция общей памяти [58, 59]. Подобные многопроцессорные архитектуры получили название NUMA (Non-Uniform Memory Access) [24]. В отличие от модели PRAM, модель PRAM-NUMA вводит новую метрику, задающую *дистанцию* (относительное расстояние) между процессорами. PRAM-NUMA компьютер включает в себя следующие компоненты (см. рис. 5): T процессоров, разбитых на p групп по T_p процессоров в группе; общедоступная глобальная разделяемая память; p модулей локальной памяти; соединительная сеть, способная определять дистанцию между процессорными группами.

Каждый процессор соединен с глобальной разделяемой памятью, и каждая процессорная группа соединена со своим собственным модулем локальной памяти. Соединительная сеть обеспечивает доступ процессоров одной группы к локальной памяти остальных групп, при этом латентность маршрутизации прямо пропорциональна дистанции между процессором и модулем памяти, к которому он обращается. Пропускная способность каналов, соединяющих группу процессоров с общей и локальной памятью, является одинаковой. Каждый процессор может находиться либо в PRAM режиме, либо в NUMA режиме. В одной группе процессоров два или более процессора могут находиться в NUMA режиме. В этом случае они выполняют один поток команд над различными данными. Следует отметить, что NUMA-режим требует использования специального языка программирования «E» [60], сложного в использовании. В работе [61] предложено некоторое расширение модели PRAM-NUMA путем введения понятия «толстых» потоков управления, представляющих собой совокупность нитей управления, количество которых может меняться в ходе выполнения программы. Все нити одного «толстого» потока работают по схеме SIMD. Модель PRAM-NUMA ориентирована на многоядерные процессоры с нестандартной архитектурой и не может применяться для большинства современных многопроцессорных систем, поставляемых на рынке высокопроизводительной вычислительной техники.

3. Многоуровневые модели с общей памятью

Многоуровневые модели с общей памятью ориентированы на однородные многопроцессорные системы, в которых память делится на уровни с разным временем доступа [1]. Указанный класс моделей предназначен для более точной и реалистичной оценки времени доступа к памяти, организованной в виде иерархии уровней с монотонно возрастающими объемом памяти и временем доступа на каждом следующем уровне. Такие модели наиболее подходят для задач, связанных с перемещением больших объемов данных между различными уровнями иерархической памяти. В большинстве случаев модели этого класса вводят дополнительное множество измеряемых системных параметров, которые используются затем в качестве параметров стоимостных функций для повышения точности получаемых оценок.

Одной из первых последовательных моделей с многоуровневой памятью была *модель НММ (Hierarchical Memory Model)* [62]. НММ-компьютер содержит неограниченное количество регистров R_1, R_2, R_3, \dots , каждый из которых может содержать целое число (или значение другого типа в зависимости от задачи). Операции выполняются так же как в модели RAM [8], за исключением того, что доступ к регистру R_i требует $\lceil \log i \rceil$ единиц времени, в то время как выполнение любой операции занимает одну единицу времени. Очевидно, что P-сложная задача, выполняемая на RAM-компьютере за время $T(n)$, может быть выполнена на НММ-компьютере за время $O(T(n) \log n)$.

Модель ВТ (Hierarchical Memory with Block Transfer) [63] расширяет модель НММ путем введения понятия блока, представляющего собой совокупность ячеек памяти с последовательными адресами. Время доступа к ячейке памяти с адресом x определяется как $f(x)$. Однако, время доступа к блоку ячеек $[x - l, x]$ составляет $f(x) + l$. В качестве функции f могут фигурировать $f(x) = \alpha$, $f(x) = x^\alpha$ и $f(x) = \log x$, где α — некоторая константа. Арифметические операции, как и в модели RAM, занимают одну единицу времени. В соответствии с этим сложение двух чисел, хранящихся в ячейках памяти с адресами x и y , и сохранение результата в ячейке памяти с адресом z занимает $f(x) + f(y) + 1 + f(z)$ единиц времени на ВТ-компьютере.

Еще одной ранней последовательной моделью вычислений для памяти с иерархическим доступом является *модель LPM (Logarithmic Pipelined Model)* [64]. Модель LPM предполагает, что доступ к m последовательным ячейкам памяти занимает $\log m$ единиц времени. Это предположение согласуется с технологией VLSI (Very Large Scale Integration) [65], предназначенной для создания проблемно-ориентированных сверхбольших микросхем, включающих в себя ЦПУ, ПЗУ и ОЗУ. Логические вентили, с помощью которых осуществляется доступ к ячейкам памяти в VLSI системах, организуются обычно в виде древовидной структуры. В соответствии с этим доступ к m последовательным ячейкам памяти не может быть осуществлен менее, чем за $\log m$ шагов.

Обобщением моделей НММ и ВТ является последовательная *модель вычислений УМН (Uniform Memory Hierarchy)* [66]. В рамках этой модели память представляется в виде иерархии модулей $\langle M_0, M_1, \dots \rangle$. Например, M_0 может представлять регистровую память, M_1 — кэш-память, M_2 — оперативную память и так далее. Каждый

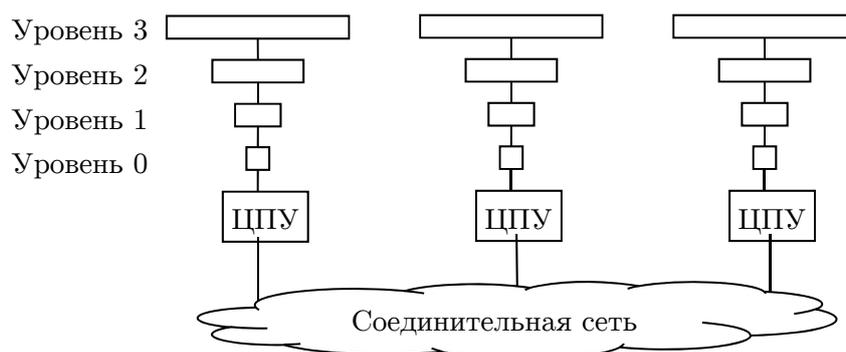


Рис. 6. Параллельная модель иерархической памяти

модуль памяти M_u характеризуется тремя параметрами: $\langle s_u, n_u, l_u \rangle$. Параметр s_u обозначает количество ячеек памяти в одном блоке; n_u — количество блоков в модуле; l_u — количество тактов, необходимых для копирования одного блока из M_u в M_{u+1} или обратно. Модель УМН обобщает модели НММ и ВТ в том смысле, что в модели УМН полагается: $n_u = \alpha s_u$, $s_{u+1} = \rho s_u$, $l_u = \rho f(u)$ для всех $u = 0, 1, \dots$. Здесь α и ρ — положительные целочисленные константы, $f(u)$ — функция, определяющая количество тактов, необходимых для копирования одной ячейки памяти из уровня M_u в уровень M_{u+1} . В качестве функции f могут фигурировать, например, такие функции: $f(u) = 1$, $f(u) = u$ или $f(u) = \rho^u$. В общем случае предполагается, что всегда имеет место неравенство $f(u) \leq f(u + 1)$.

Параллельные версии описанных многоуровневых моделей с общей памятью легко строятся путем репликации последовательной модели p раз. *Модели P-НММ и P-ВТ* [67] обобщают для параллельного случая модели НММ и ВТ соответственно. Параллельная модель с иерархической памятью строится как совокупность НММ или ВТ-компьютеров, объединенных соединительной сетью (см. рис. 6).

При этом предполагается, что обмен данными по соединительной сети происходит через память уровня 0. Пропускная способность соединительной сети полагается такой, что сортировка массива, распределенного по модулям памяти уровня 0, выполняется за время $O(\log P)$, где P — количество процессоров в параллельной машине. Отметим, что в рамках современных представлений указанная модель является адекватной, если уровни 1, 2, ... представляют внешнюю память.

Модель UPMH (Uniform Parallel Memory Hierarchy) [66] является параллельной версией модели УМН. UPMH — компьютер строится как совокупность УМН-машин, объединенных древовидной соединительной сетью. При этом для передачи сообщений используется память последнего уровня (самая медленная память).

4. Одноуровневые модели с распределенной памятью

Модели параллельных вычислений с распределенной памятью предполагают, что многопроцессорная вычислительная система строится как совокупность процессорных узлов, соединенных высокоскоростной коммуникационной сетью, обеспечивающей передачу сообщений от одного процессора другому. При этом каждый процессорный узел имеет

свою собственную приватную память, недоступную для других узлов. В этом разделе дается краткое описание моделей параллельных вычислений BSP, LogP, их модификаций, а также некоторых других одноуровневых моделей для многопроцессорных систем с распределенной памятью.

Модель BSP (Bulk-Synchronous Parallelism) была предложена Валиантом (Valiant) в работе [2]. Данная модель широко используется при разработке и анализе параллельных алгоритмов и программ. BSP-компьютер представляет собой систему из K процессоров P_1, \dots, P_K , имеющих приватную память M_1, \dots, M_K , и соединенных сетью, позволяющей передавать данные от одного процессора другому (см. рис. 7). Для соединительной сети вводятся следующие характеристики: g — время, необходимое для передачи по сети одного машинного слова; L — время, необходимое для выполнения глобальной синхронизации. Мы будем предполагать, что время измеряется в машинных тактах. Моделирование передачи сообщений в BSP-компьютере реализуется с помощью понятия h -сессии. h -сессия является абстракцией произвольной коммуникационной операции, в ходе которой каждый процессор передает не более h машинных слов и получает не более h машинных слов. Время на выполнение одной h -сессии в BSP-компьютере не может превышать $h \cdot g$.

BSP-программа состоит из n потоков команд, каждый из которых назначается отдельному процессору, и делится на *супершаги*, которые выполняются последовательно относительно друг друга. Каждый супершаг, в свою очередь, включает в себя следующие четыре последовательных шага: 1) вычисления на каждом процессоре с использованием только локальных данных; 2) глобальная барьерная синхронизация; 3) пересылка данных от любого процессора любым другим процессорам путем выполнения одной h -сессии; 4) глобальная барьерная синхронизация. Переданные данные становятся доступными для использования только после барьерной синхронизации. Пример BSP-программы из двух супершагов приведен на рис. 8. Жирными линиями обозначены локальные вычисления, тонкими линиями со стрелками — пересылка данных.

Стоимостная функция в модели BSP строится следующим образом [68, 69]. Пусть BSP-программа состоит из S супершагов. Предположим, что каждый процессор на i -том супершаге выполняет максимум w_i тактов в ходе локальных вычислений. Тогда общее время t_i , затрачиваемое системой на выполнение i -того супершага, вычисляется по формуле

$$t_i = w_i + h \cdot g + L. \quad (3)$$

Время T выполнения всей программы определяется по формуле

$$T = W + h \cdot g \cdot S + L \cdot S, \quad (4)$$

где $W = \sum_{i=1}^S w_i$. Значения W и S , как правило, зависят от количества процессоров K и от размера задачи.

При проектировании BSP-программы значения K , g и L рассматриваются как конфигурационные параметры многопроцессорной системы. Эффективный BSP-алгоритм должен минимизировать количество локальных вычислений, объем передаваемых данных и число глобальных синхронизаций при некоторых реалистичных значениях указанных параметров. Основными принципами разработки BSP-программ являются следующие.

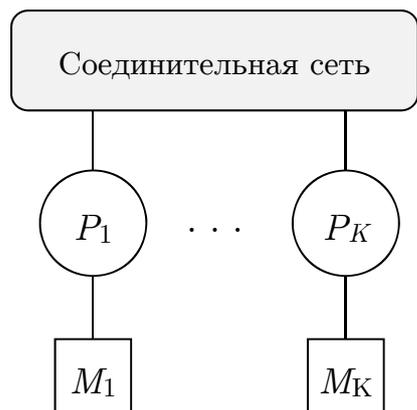


Рис. 7. BSP-компьютер

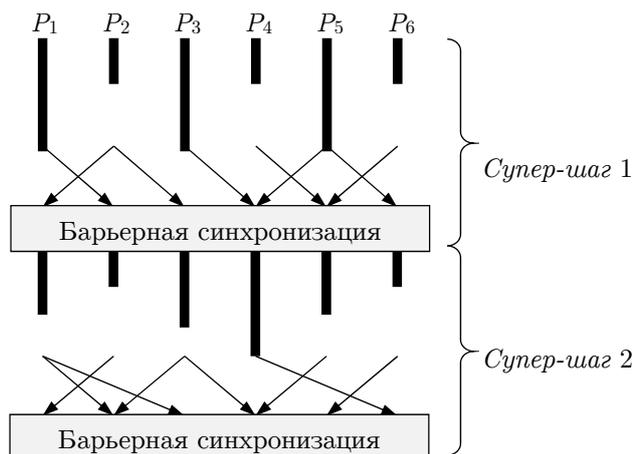


Рис. 8. Вычислительный процесс BSP

- Балансировка загрузки процессоров, позволяющая сократить максимальное время локальных вычислений.
- Локальность данных, позволяющая сократить затраты на коммуникации.
- Крупнозернистый параллелизм, позволяющая сократить затраты на коммуникации и глобальную синхронизацию.

Балансировка загрузки процессоров касается и равномерного распределения вычислительной нагрузки, и пропорционального распределения данных между процессорами. Принцип *локальности данных* говорит о том, что в локальной памяти процессора необходимо хранить данные, к которым этот процессор обращается чаще всего. *Крупнозернистый параллелизм* подразумевает разделение программы на крупные параллельные секции, каждая из которых выполняется на отдельном процессоре, содержит значительный объем вычислений и не предполагает обменов данными в ходе этих вычислений. Значения параметров g и L для соединительной сети конкретного компьютера могут быть получены путем бенчмаркинга (эталонного тестирования). Методика бенчмаркинга в контексте модели BSP описана в работе [70]. Там же можно найти полученные значения параметров для некоторых реальных многопроцессорных систем.

В качестве примера рассмотрим простейший параллельный алгоритм, вычисляющий произведение квадратной матрицы A размера $n \times n$ на вектор b размера n . Для решения задачи задействуем $K = n + 1$ процессоров: P_0, P_1, \dots, P_n . Строки матрицы A распределим по процессорам следующим образом: i -тая строка a_i матрицы A будет храниться в памяти процессора P_i ($i = 1, \dots, n$). В память каждого процессора также поместим полный вектор b . Каждый процессор P_i параллельно выполняет умножение своей строки a_i на вектор b и получает одно число. Это число пересылается процессору P_0 , который формирует результирующий вектор. Таким образом, мы имеем BSP-алгоритм, состоящий из одного супер-шага. Умножение строки a_i на вектор b состоит из n операций умножения и $n - 1$ операций сложения. Следовательно, временные затраты на вычисления составят $w = O(n) + O(n - 1) \approx O(n)$. Коммуникационные затраты составят gn в предположении, что $h = n$. Суммарные временные затраты, таким образом, составят $O(n) + gn + L$.

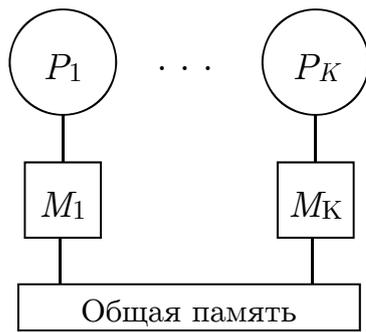


Рис. 9. BSPRAM-компьютер

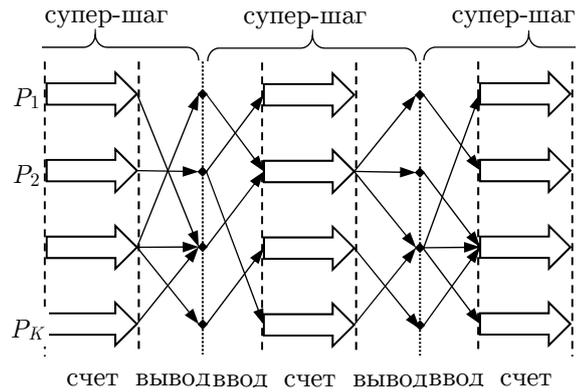


Рис. 10. Вычислительный процесс BSPRAM

В работе [71] предлагается более сложный BSP-алгоритм умножения плотной матрицы на вектор, вычислительная сложность которого при $K = n$ составляет $O(n) + g\sqrt{n} + L$. Основным недостатком модели BSP является то, что она предполагает передачу сообщений длиной в одно машинное слово и не учитывает тот факт, что передача h машинных слов в виде одного сообщения может быть более эффективной, чем передача h сообщений длиной в одно машинное слово.

Модель BSPRAM (Bulk Synchronous Parallel Random Access Machine), предложенная в работах [72, 73], является вариацией моделей BSP и PRAM. BSPRAM-компьютер состоит из K процессоров P_1, \dots, P_K , каждый из которых имеет свою локальную память M_1, \dots, M_K . В дополнение к этому имеется общая память, в равной мере доступная всем процессорам (см. рис. 9). Процессоры могут выполнять различные потоки команд. Вычислительный процесс в модели BSPRAM представляет собой последовательность супершагов (см. рис. 10). Супершаг включает в себя три фазы: «ввод», «счет» и «вывод». На фазе «ввод» процессоры читают данные из общей памяти; на фазе «вывод» процессоры пишут данные в общую память. Процессоры синхронизируются при переходе от текущего супершага к следующему; внутри супершага процессоры работают асинхронно. Указанный подход позволяет объединить преимущества моделей BSP и PRAM.

Модель параллельных вычислений LogP была предложена группой авторов в работе [11] как некоторое усовершенствование модели BSP. В качестве критики модели BSP указывались следующие факторы. На каждом супершаге количество данных, обрабатываемых одним процессором, должно быть примерно равным количеству слов, получаемых в ходе h -сессии, то есть h , что ограничивает зернистость параллелизма снизу. Далее, сообщения, передаваемые в конце выполнения супершага, не могут быть использованы реципиентом до начала следующего супершага. И последнее, модель BSP предполагает аппаратную поддержку механизма глобальной синхронизации, однако большинство многопроцессорных систем с распределенной памятью не имеют подобных аппаратных средств.

По аналогии с моделью BSP модель LogP предполагает, что компьютер состоит из процессорных модулей с приватной памятью, соединенных коммуникационной сетью. Основными стоимостными параметрами модели являются следующие.

L — (*latency*) верхняя граница латентности, представляющая собой время, необходимое для передачи сообщения длиной в одно машинное слово от одного процессорного модуля другому.

o — (*overhead*) накладные расходы, представляемые как промежуток времени, в течение которого процессорный модуль занят приемом или передачей сообщения; в это время процессорный модуль не может выполнять никакую другую работу.

g — (*gap*) задержка, представляющая собой минимальное время между двумя последовательными операциями чтения или передачи сообщений, выполняемыми процессорным модулем.

P — количество процессорных модулей.

При этом предполагается, что коммуникационная сеть имеет ограниченную пропускную способность, позволяющую каждому процессорному модулю получать или посылать одновременно не более $\lfloor L / g \rfloor$ сообщений. Если процессорный модуль пытается передать сообщение, перекрывающее этот лимит, то он переводится в состояние ожидания до тех пор, пока сообщение можно будет послать, не выходя за границу пропускной способности сети. Базовый вариант модели LogP предполагает, что все сообщения имеют небольшой размер (одно или небольшое количество машинных слов). Большие сообщения необходимо фрагментировать.

Время выполнения алгоритма в модели LogP определяется как максимум временных затрат среди всех процессорных модулей, участвующих в вычислениях. Время передачи одного короткого сообщения от одного процессорного модуля другому составляет $o + L + o$. Время доступа к элементу данных, располагающемуся в памяти другого процессорного модуля, будет равно $2L + 4o$. Для последовательной передачи n сообщений между процессорными модулями P_1 и P_2 может быть организован конвейер, как это показано на рис. 11. В этом случае передача n последовательных сообщений займет время, равное $(n - 1)g + o + L + o$ [74].

Сильной стороной модели LogP является ее простота. Однако, эта простота иногда может приводить к недостаточно точному предсказательному моделированию производительности алгоритмов и программ. Очевидным недостатком модели LogP является ограничение на размер сообщений. Попытка преодоления этого ограничения была сделана в ряде модификаций модели LogP, которые будут рассмотрены ниже.

В работе [75] было предложено расширение модели LogP путем добавления нового параметра G (*Gap per Byte*), задающего время, необходимое для передачи одного байта в составе длинного сообщения. Новая модель получила название **LogGP**. В модели LogGP время передачи сообщения длиной в m байт вычисляется по следующей формуле:

$$T_n = o + (m - 1)G + L + o. \quad (5)$$

Более радикальное расширение модели LogP предложено в статье [76], где описывается **параметризованная модель PlogP**, расширяющая модель LogP путем введения дополнительных параметров. Модель PlogP лучше учитывает особенности коммуникационного программного обеспечения такого, как MPI [77, 78]. Среди этих особенностей выделяются следующие: 1) накладные расходы o и задержка g в значительной мере зависят от размеров сообщения; 2) накладные расходы, связанные с посылкой и приемом сообщения, могут существенно различаться, так как обработка асинхронно приходящих сообщений требует принципиально иной реализации, чем синхронные вызовы операции посылки.

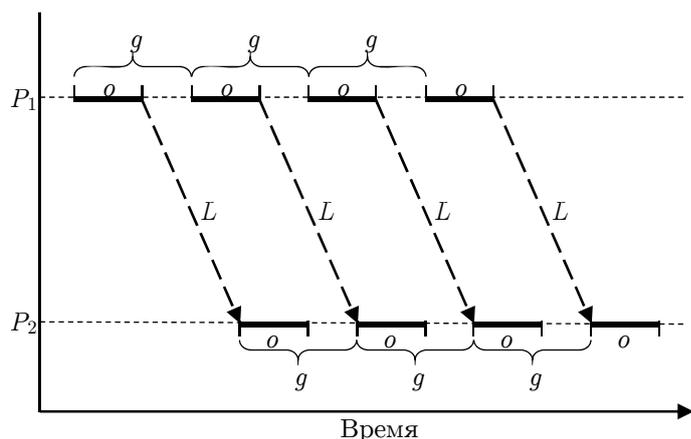


Рис. 11. Конвейерная передача сообщений в модели LogP

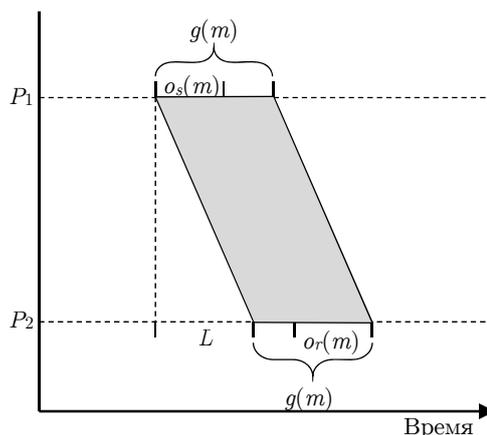


Рис. 12. Передача сообщения в модели PLogP

В соответствии с этим модель PLogP вместо параметров o и g вводит следующие параметры, зависящие от длины m сообщения:

- $o_s(m)$ — накладные расходы, связанные с посылкой сообщения длиной n .
- $o_r(m)$ — накладные расходы, связанные с приемом сообщения длиной n .
- $g(m)$ — задержка между двумя последовательными операциями чтения или передачи сообщений длиной m .

Семантика параметра L (латентность) также несколько меняется. В модели PLogP латентность L обозначает время, затрачиваемое на передачу первого бита сообщения. Параметр P имеет тот же смысл, что и в модели LogP. Моделирование посылки сообщения длиной m между процессорными модулями P_1 и P_2 в метрике модели PLogP проиллюстрировано на рис. 12. Таким образом, время, затрачиваемое на передачу одного сообщения длины m , составляет $T_1 = L + g(m)$.

Соотношения между параметрами моделей PLogP и LogP/LogGP приведены в табл. 1. При этом в качестве короткого сообщения берется один байт. Время передачи одного короткого сообщения в LogP/LogGP составляет $o + L + o$. В модели PLogP это время равно $L + g(1)$. Отсюда получается формула $o + L^{LogP/LogGP} + o = L^{PLogP} + g(1)$. Подставляя вместо o значение $(o_s(1) + o_r(1)) / 2$, получаем итоговое соотношение $L^{LogP/LogGP} = L^{PLogP} + g(1) - o_s(1) - o_r(1)$.

Таблица 1

Соотношения между LogP/LogGP и PLogP

| LogP/LogGP | PLogP |
|------------|------------------------------|
| L | $L + g(1) - o_s(1) - o_r(1)$ |
| o | $(o_s(1) + o_r(1)) / 2$ |
| g | $g(1)$ |
| G | $g(m) / m$ для $m \gg 1$ |
| P | P |

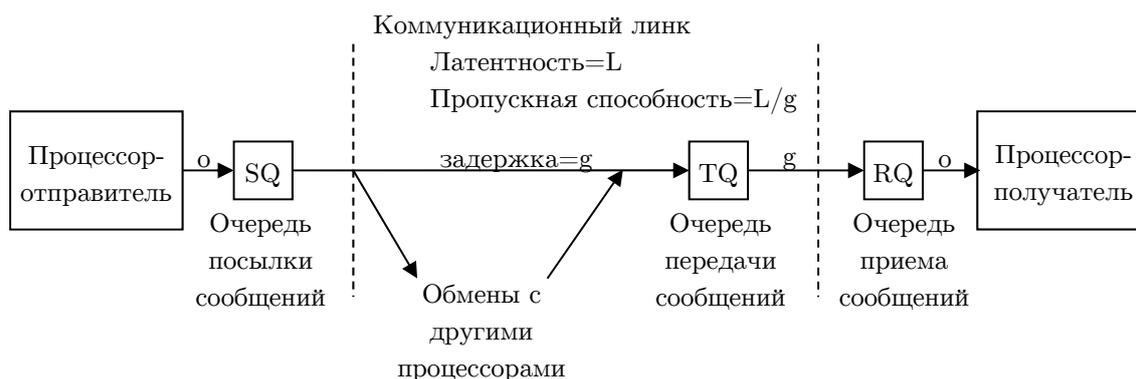


Рис. 13. Структура модели LogPQ

В работе [79] была предложена *модель параллельных вычислений LogPQ*, расширяющая модель LogP путем введения дополнительных параметров, связанных с обслуживанием очередей при передаче сообщений: SQ — размер очереди отправки сообщений, RQ — размер очереди приема сообщений и TQ — размер очереди передачи сообщений (см. рис. 13). Длина сообщений в модели LogPQ, также, как и в LogP, является постоянной и равна одному *коммуникационному слову* длиной w_c . Длинные сообщения представляются в виде последовательности сообщений длиной в одно *коммуникационное слово*. При этом длина w_p машинного слова может не совпадать с w_c . Соотношения между параметрами моделей LogPQ и LogP приведены в табл. 2, где $r = w_p / w_c$; L — латентность, представляющая собой время, необходимое для передачи сообщения длиной в одно коммуникационное слово; o — накладные расходы, представляемые как промежуток времени, в течение которого процессорный модуль занят приемом или передачей сообщения; g — временной интервал между двумя последовательными передачами сообщений от одного процессора другому.

Максимальная пропускная способность соединительной сети в модели LogPQ оценивается как $\lfloor L/g \rfloor + SQ + RQ$. Модель LogP, таким образом, может рассматриваться как частный случай модели LogP при $SQ = TQ = RQ = 1$. В работе [80] модель LogPQ была использована для оценки времени выполнения параллельного алгоритма перемножения матриц размера $n \times n$ на массивно-параллельном компьютере CM-5 [81]. Вычислительные эксперименты показали, что модель LogPQ более точно предсказывает реальное время выполнения программы по сравнению с LogP. Однако разница между двумя моделями оказалась существенной только для матриц небольшого размера ($n < 32$). Кроме этого, модель LogPQ оказалось сложно адаптировать к новым параллельным вычислительным архитектурам, и она не получила широкого распространения.

Таблица 2

Соответствие параметров LogPQ и LogP

| LogPQ | LogP |
|-------|--------------------|
| L | $L + o + (r - 1)g$ |
| o | $o + (r - 1)g$ |
| g | rg |
| P | P |

Модель параллельных вычислений LogGPS, представленная в работе [82], расширяет модель LogGP путем введения дополнительного параметра S , отражающего затраты на синхронизацию. Фактически модель LogGPS явилась адаптацией модели LogPQ к особенностям коммуникационных протоколов библиотеки MPICH [83], представляющей собой переносимую реализацию стандарта MPI. Еще одним расширением модели LogGP является *модель параллельных вычислений LoGPC* [84, 85], которая учитывает накладные расходы, связанные с конфликтами, возникающими при обмене сообщениями в мульти-процессоре с распределенной памятью, а также конвейерную передачу длинных сообщений с использованием интерфейса DMA [86, 87]. Модель LoGPC является более точной по сравнению с LogGP, однако это достигается ценой значительного усложнения анализа алгоритмов. Кроме этого, LoGPC существенно зависит от аппаратных особенностей соединительной сети.

5. Многоуровневые модели с распределенной памятью

Модель параллельных вычислений $\log_n P$ [88, 89] является расширением модели LogP. Модель $\log_n P$ ориентирована на кластерные архитектуры, в которых передача сообщений между узлами с иерархической памятью осуществляется с помощью программного обеспечения промежуточного слоя такого, как MPI. Кроме явных затрат на передачу сообщений, учитываемых моделью LogP, модель $\log_n P$ учитывает также неявные затраты, связанные с передачами данных между различными уровнями иерархической памяти, которые выполняются программным обеспечением промежуточного слоя при организации пересылок между процессорными узлами. Экспериментально было показано, что в некоторых приложениях при передаче сильно фрагментированных данных большого объема неявные затраты могут в несколько раз превосходить явные затраты на пересылку типа точка-точка. Стоимостная метрика модели $\log_n P$ базируется на следующих пяти параметрах.

- l — *эффективная латентность (effective latency)*, определяемая как разница между временем, затрачиваемым процессором на передачу фрагментированных и нефрагментированных данных. Эффективная латентность может быть вычислена с помощью платформенно-зависимой функции $f(s, d) = l$, где s — длина одного сообщения при передаче массива, d — длина пропусков между единичными сообщениями в передаваемом массиве.
- o — *эффективные накладные расходы (effective overhead)* определяются как время, затрачиваемое процессором на передачу единичного сообщения в случае нефрагментированных данных ($d = 1$). Эффективные накладные расходы могут быть вычислены с помощью платформенно-зависимой функции $f(s, 1) = o$.
- g — (*gap*) задержка, представляющая собой минимальное время между двумя последовательными операциями передачи сообщений. В модели $\log_n P$ предполагается, что $g = 0$.
- n — количество уровней иерархической памяти, используемых программным обеспечением промежуточного слоя при передаче сообщений. Фактически n определяет количество неявных обменов данными между различными уровнями иерархической памяти.
- P — количество процессорных модулей.

Время, затрачиваемое на передачу одного сообщения в модели $\log_n P$ определяется по формуле

$$T = \sum_{i=0}^{n-1} (o_i + l_i) = \sum_{i=0}^{n-1} (f_i(s, 1) + f_i(s, d)). \quad (6)$$

Частным случаем модели $\log_n P$ является модель $\log_3 P$ ($n = 3$), ориентированная на кластеры с SMP-узлами. В этом случае предыдущая формула преобразуется к виду

$$T = \sum_{i=0}^2 (o_i + l_i).$$

Заметим, что при $n = 1$ модель $\log_n P$ эквивалентна модели *Memory logP*, представленной в более ранней работе [90].

Модель параллельных вычислений HiHCoHP (Hierarchical Hyper-Clusters of Heterogeneous Processors) [91, 92] ориентирована на моделирование многоуровневых иерархических сетей из гетерогенных кластеров. На пути от отправителя P_s к адресату P_r сообщение пересекает несколько уровней в иерархии сетей, упорядоченных по возрастанию отношения латентности к пропускной способности. Коммуникационная стоимость вычисляется как функция, зависящая от стоимостей $\sigma_s^{(k)}$ и $\sigma_r^{(k)}$ обработки сообщения на уровне k , стоимостей $\pi_s^{(k)}$ и $\pi_r^{(k)}$ упаковки и распаковки каждого пакета сообщения, и следующих трех параметров: 1) латентности $\lambda^{(k)}$, которая включает в себя латентности уровней ниже k ; 2) пропускной способности $\beta^{(k)}$, включающей в себя пропускную способность сетей более низкого уровня; 3) ширины канала $\kappa^{(k)}$, вычисляемой как максимальное число пакетов, передаваемых за один такт по сети уровня k . Время передачи сообщения длиной m вычисляется по формуле $T(m) = \sigma_s^{(k)} + \pi_s^{(k)}p + \lambda^{(k)} + \frac{p-1}{\beta^{(k)}} + \sigma_r^{(k)} + \pi_r^{(k)}p$, где p — количество процессов. Модель HiHCoHP также включает параметр ρ_i , обозначающий процессорное время, затрачиваемое узлом P_i на выполнение единичного объема вычислений.

Модель параллельных вычислений D-BSP (Decomposable Bulk Synchronous Parallel) [17] является расширением модели BSP. Модель D-BSP представляет многопроцессорную вычислительную систему как совокупность кластеров, каждый из которых имеет свою внутреннюю соединительную сеть, характеризующуюся определенной пропускной способностью и задержкой при передаче сообщений. Общее количество n процессоров в системе предполагается равным степени числа два. Для фиксированного i , такого, что $0 \leq i \leq \log_2 n$, n процессоров делятся на 2^i непересекающихся i -кластеров $C_0^{(i)}, \dots, C_{2^i-1}^{(i)}$ по $n/2^i$ процессоров в каждом. Процессоры каждого i -кластера могут обмениваться сообщениями независимо от остальных. Структура D-BSP машины формируется как бинарное дерево высоты $\log_2 n$, узлами которого являются i -кластеры. Указанное бинарное дерево обладает следующими свойствами: кластер $C_j^{\log_2 n}$ содержит только один процессор P_j ($0 \leq j < n$); $C_j^{(i)} = C_{2j}^{(i+1)} \cup C_{2j+1}^{(i+1)}$ для $0 \leq i < \log_2 n$ и $0 \leq j < 2^i$. Выполне-

ние программы в модели D-BSP состоит из последовательности *мегашагов*. Каждый мегашаг представляет собой последовательность $1 + \log_2 n$ супершагов. В рамках i -того супершага процессоры выполняют некоторые вычисления и обмениваются данными исключительно в пределах их собственного i -кластера. В финале i -того супершага в каждом i -кластере выполняется барьерная синхронизация. Если на i -том супершаге каждый процессор выполняет не более w операций и передает или получает не более h сообщений, то время выполнения этого супершага оценивается по формуле $w + hg_i + l_i$, где g_i — задержка, а l_i — латентность для i -того уровня.

В работе [93] предлагается *модель параллельных вычислений HLogGP (Heterogeneous LogGP)*, являющаяся расширением модели LogGP и ориентированная на гетерогенные вычислительные кластеры. Модель HLogGP допускает наличие в кластере вычислительных узлов, имеющих различающиеся процессоры, модули памяти и сетевые адаптеры. Основная идея — заменить скалярные параметры на векторы и матрицы для учета специфики разнородных узлов кластерной вычислительной системы. Для моделирования гетерогенной вычислительной системы с M узлами вводится пять параметров: латентность L и байтовая задержка G представляют собой матрицы размера $M \times M$; величина накладных расходов o , задержка между сообщениями g и вычислительная мощность P являются векторами длины M . Если все эти параметры определены с достаточной точностью, модель HLogGP может достоверно предсказывать производительность параллельных алгоритмов, в которых коммуникационные затраты превалируют над вычислительными.

Модель параллельных вычислений LogfP, предложенная в [94], расширяет модель LogP на кластерные вычислительные системы, построенные с использованием соединительной сети InfiniBand [95]. Модель LogfP предполагает, что первые f коротких сообщений передаются с нулевыми накладными расходами. Способность InfiniBand мгновенно пересылать сразу несколько коротких сообщений (в количестве, не превышающем f , после чего начинает наблюдаться замедление) обнуляет параметр g . Более того, использование в сети InfiniBand технологии RDMA (Remote Direct Memory Access) при передаче сообщений делает неадекватным параметр o_R , определяющий накладные расходы на стороне получателя. Параметры модели LogfP измеряются с помощью эталонного теста RTT (Round-Trip Time) по схеме $1 : P - P : 1$. Модель LogfP позволяет повысить оценку производительности барьерной синхронизации на 40% по сравнению с другими известными моделями.

Модель параллельных вычислений PLP [96] ориентирована на иерархические гетерогенные вычислительные системы с однопроцессорными узлами и соединительной сетью Ethernet. Время передачи сообщения длиной m байт по схеме точка-точка вычисляется по формуле $T(m) = L + P_s + P_r$. Параметры P_s и P_r задают время, затрачиваемое процессорами отправителя и получателя на организацию передачи сообщения. Эти параметры в свою очередь зависят от физических параметров конкретных процессоров. Латентность L определяется следующим образом: $L = (n - 1)G + L_t$, где G — межкадровая задержка при отправке n последовательных Ethernet-кадров, на которые разбивается сообщение, и L_t — латентность при кадровой передаче.

Модель PLP применима как для обменов типа точка-точка, так и для коллективных коммуникаций.

Модель параллельных вычислений LogGOPS, предложенная в работе [97], представляет собой расширение модели LogGPS и предназначена для оценки масштабируемости параллельных алгоритмов, ориентированных на большие вычислительные системы, работающие под управлением MPI. Согласно этой модели, стоимость передачи сообщения длиной m байт составляет $2o + L + (m - 1)O + (m - 1)G$. Параметры o, L, G имеют тот же смысл, что и в модели LogGPS. Параметр O является специфичным для модели LogGOPS и обозначает накладные расходы в пересчете на байт. Модель LogGOPS применяется для линейных коммуникаций типа scatter (один посылает всем) и gather (один получает от всех), для широковещательных коммуникаций на основе бинарного дерева и для распределенных коммуникаций, включающих в себя схему «всем от всех» и барьерную синхронизацию.

Модель параллельных вычислений LogGPH [12] является расширением модели LogGP и ориентирована на многопроцессорные системы, построенные на основе иерархии соединительных сетей. Для каждого уровня в сетевой иерархии вводится свой вектор параметров. Два процесса, запущенные на одном процессоре, взаимодействуют на уровне 1 и передают друг другу сообщения через общую память. Процессы, запущенные на разных процессорах одного узла, взаимодействуют на уровне 2. Процессы, запущенные на разных процессорных узлах, взаимодействуют на уровне 3, обмениваясь сообщениями через соединительную сеть. Время передачи сообщения длины m на сетевом уровне i вычисляется по формуле $T_i(m) = 2o_i + (m - 1)G_i + L_i$.

В работе [98] предложена *модель параллельных вычислений Multi-BSP*, расширяющая модель BSP в двух направлениях. Первое, Multi-BSP является иерархической моделью с произвольным количеством уровней, отражающих реальные технические особенности иерархической памяти и различных уровней кэш-памяти современных многопроцессорных систем. Второе, в качестве дополнительного параметра Multi-BSP включает в себя объем памяти на каждом уровне. Для каждого уровня i в иерархии вводится вектор параметров (p_i, g_i, L_i, m_i) , где p_i — количество процессоров, g_i — задержка при передаче сообщения, L_i — затраты на синхронизацию, m_i — объем памяти/кэша. Для системы, включающей в себя d уровней, общее количество процессоров вычисляется по формуле $P_d = \prod_{i=1}^d p_i$, суммарный объем памяти вычисляется по формуле $M_d = m_d + \sum_{i=1}^{d-1} m_i \prod_{j=i+1}^d p_j$, суммарная задержка вычисляется по формуле $G_d = \sum_{i=1}^d g_i$.

Модель mlog_nP [99, 100] является расширением модели log_nP, ориентированным на вычислительные кластеры с многоядерными процессорами. Иерархию n -уровневой памяти, введенную в модели log_nP, авторы модели mlog_nP называют вертикальной. Дополнительно они вводят горизонтальную m -уровневую иерархию каналов передачи данных. По каналу нулевого уровня происходит обмен данными между ядрами одного процессора, канал первого уровня используется для обмена данными между ядрами разных процессоров одного процессорного узла, канал второго уровня — для обмена данными между ядрами разных процессорных узлов, и так далее. В соответствии с этим стоимость передачи сообщения на горизонтальном уровне i вычисляется по формуле

$$T_i = \sum_{j=0}^{n_i-1} (o_j^i + l_j^i) = \sum_{j=0}^{n_i-1} (f_j^i(s,1) + f_j^i(s,d)), \quad (7)$$

где семантика всех параметров с точностью до горизонтального уровня наследуется от модели $\log_n P$.

Модель параллельных вычислений SLOWER, предложенная в работе [101], ориентирована на экзамасштабные вычислительные системы. Главной целью модели SLOWER является оценка производительности P вычислительной системы, которая вычисляется по формуле

$$P = e(L, O, W) \times s(S) \times \mu(E) \times a(R), \quad (8)$$

где e обозначает эффективность ($0 < e < 1$), L — латентность, O — накладные расходы, W — задержка, вызванная ожиданием доступа к разделяемым ресурсам, s — степень параллелизма, S — мера недозагрузки, μ — скорость выполнения потока машинных команд, E — потребление энергии, a — доступность, R — надежность. Формула (8) включает в себя критические с точки зрения модели SLOWER параметры и показывает их взаимное влияние на общую производительность системы. По мнению авторов модели SLOWER эффективность e и доступность a не оказывают значительного влияния на общую производительность системы. Мера доступности a отражает эксплуатационные проблемы, препятствующие способности системы выполнять реальную вычислительную работу. Она зависит от параметра R , определяющего среднее время между отказами системы. Пиковая скорость выполнения потока машинных команд μ зависит от тактовой частоты, которая в определенном диапазоне прямо пропорциональна энергопотреблению E . Степень параллелизма s определяется количеством процессорных ядер, одновременно выделяемых для решения задачи. Мера недозагрузки S вычисляется как отношение времени простоя процессорного ядра ко времени его работы при решении задачи. Недозагруженность возникает либо когда процессорное ядро простаивает в ожидании работы или необходимых данных, либо в результате дисбаланса в распределении вычислительной работы между процессорными ядрами. Эффективность e вычислительной системы определяется как отношение скорости непрерывного выполнения машинных команд к пиковой скорости. Как следует из формулы (8), на эффективность влияет следующий ряд факторов. Латентность L представляет собой время, необходимое для инициализации доступа к удаленным данным или сервисам в незагруженной системе. Задержка W (ожидание отложенного доступа) возникает из-за конфликта доступа к разделяемым логическим или физическим ресурсам в загруженной системе. Накладные расходы O включают в себя процессорное время, которое тратится на управление параллельными ресурсами и диспетчеризацию процессов. Модель SLOWER может применяться к широкому классу вычислительных систем экзафлопсного уровня производительности, однако в настоящее время отсутствует математическая формализация этой модели.

Модель параллельных вычислений HLog_nGP (Heterogeneous Log_nGP), предложенная в работе [13], развивает идеи, заложенные в моделях *LogGP* и *log_nP*. Модель HLog_nGP ориентирована на кластерные вычислительные системы с графическими ускорителями [26]. Модель HLog_nGP предполагает, что передача данных в ГПУ-кластере может происходить на трех различных уровнях: через оперативную память, через шину PCI

и через соединительную сеть. В общем случае полагается, что количество таких *атомарных* коммуникационных уровней равно некоторой положительной константе n . В соответствии с этим стоимостная метрика модели $HLog_nGP$ строится на основе следующих шести параметров:

- n — количество атомарных уровней при передаче сообщения от одного процессорного узла другому;
- L — латентность, представляющая собой вектор латентностей (l_1, \dots, l_n) для различных атомарных коммуникационных уровней;
- o — накладные расходы, представляющие собой вектор накладных расходов (o_1, \dots, o_n) для каждого атомарного коммуникационного уровня;
- g — (*gap*) задержка, представляющая собой вектор (g_1, \dots, g_n) , определяющий для каждого атомарного коммуникационного уровня минимальное время между двумя последовательными операциями передачи сообщений;
- G — (*gap per byte*) задержка на байт, представляющая собой вектор (G_1, \dots, G_n) , определяющий для каждого атомарного коммуникационного уровня время, необходимое для передачи одного байта в составе длинного сообщения;
- P — вычислительная производительность, представляющая собой пару (P_C, P_G) , определяющую производительность ЦПУ и ГПУ соответственно.

Частный вид модели $HLog_3GP$ имеет три атомарных коммуникационных уровня и соответствует ГПУ- кластерам. В этом случае время, необходимое для передачи одного сообщения между процессорными узлами, оценивается по формуле

$$T_{comm} = \sum_{k=1}^3 L_k + 2 \max(o_k, g_k) + (s - 1)G_k, \quad (9)$$

где s обозначает длину сообщения в байтах.

Модель параллельных вычислений MBSP (Multi-memory BSP) [102] является расширением модели BSP и ориентирована на многопроцессорные системы с многоуровневой иерархической памятью и многоядерными процессорами. Стоимостная метрика модели MBSP базируется на семи параметрах (p, l, g, m, L, G, M) , где параметры (p, l, g) наследуются от модели BSP, а (m, L, G, M) являются новыми параметрами, специфицирующими иерархическую память. Параметр M задает объем приватной «быстрой» памяти, имеющейся у каждого процессора. Параметр m задает объем «медленной» памяти, доступной всем ядрам процессорного узла. Стоимость доступа к медленной памяти определяется парой (L, G) . Параметр G определяет время доступа к медленной памяти в пересчете на одно машинное слово. Параметр L задает латентность при обменах с медленной памятью. Время выполнения супершага вычисляется следующим образом: $\max(l, L) + w + gh + Gb$, где w — максимальное время вычислений, выполняемых одним процессором, h — максимальный размер сообщения, передаваемого или получаемого процессором, и b — максимальный объем обменов с медленной памятью, выполняемых одним процессором в ходе супершага.

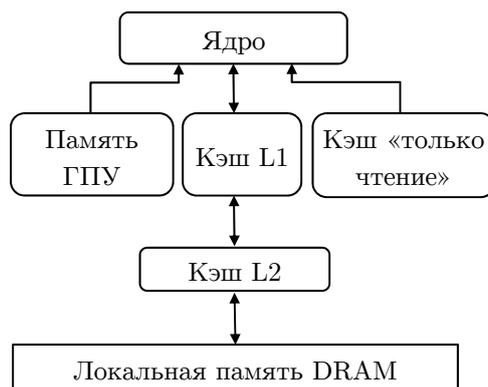


Рис. 14. Иерархия памяти в ГПУ с архитектурой Kepler

В работе [103] предложено *расширение модели BSP для ГПУ под управлением CUDA* (далее для краткости — CUDABSP). Модель CUDABSP ориентирована на предсказание времени выполнения CUDA-программы на кластерных вычислительных системах с графическими ускорителями архитектуры Kepler (см. рис. 14) на основе оценок затрат на коммуникации и вычисления, которые выполняются независимо. Время, расходуемое на вычисления, в свою очередь, оценивается на основе работы одного процессорного ядра и вычисляется по формуле

$$T_k = \frac{t \cdot (Comp + Comm_{DRAM} + Comm_{GPUM})}{R \cdot P \cdot \lambda}, \quad (10)$$

где t — количество нитей, выполняемых на ядре, $Comp$ — время, расходуемое нитью на вычисления, $Comm_{DRAM}$ — затраты на обмены с локальной памятью DRAM процессорного узла, $Comm_{GPUM}$ — затраты на обмены с памятью ГПУ, R — тактовая частота, P — количество ядер в ГПУ, λ — положительное число, больше либо равное единицы, моделирующее эффект аппаратной оптимизации. Затраты на глобальные коммуникации оцениваются также, как в модели BSP.

6. Другие обзоры

Одним из первых обзоров моделей параллельных вычислений была работа [104], опубликованная Маггом (Maggs), Матесоном (Matheson) и Тарьяном (Tarjan) в 1995 г. В данном обзоре рассматриваются модели как для общей, так и для распределенной памяти, включая модели PRAM, BSP, P-HMM, P-VT и некоторые другие.

Скилликорн (Skillicorn) и Талья (Talia) в обзоре [9] 1998 года рассматривают средства параллельного программирования в целом, включая сюда, помимо моделей, языки, фреймворки и технологии параллельного программирования. Они формулируют требования к инструментам параллельных вычислений (легкость программирования, легкость понимания, архитектурная независимость, гарантированная производительность, наличие стоимостных метрик и др.) и затем в контексте этих требований рассматривают широкий спектр известных инструментов, включая формализм Бирда—Мертенса, языки программирования Haskell, параллельный Prolog, Modula 3*, Occam, Linda, математические модели параллельных вычислений BSP, LogP, PRAM, библиотеки PVM, MPI и многие другие инструменты.

Грама (Grama), Кумар (Kumar), Ранка (Ranka) и Сингх (Singh) в 2001 году опубликовали обзор [35], в котором предложили свою иерархическую таксономию моделей параллельных вычислений в виде бинарного дерева. На верхнем уровне они поставили следующие два класса: «p-ported global memory» (общая память с параллельным многоканальным доступом) и «p-single ported memories» (распределенная память, состоящая из блоков с одноканальным доступом). Каждый из этих классов делится на два подкласса: «Bulk Access Locality» (блочный доступ к участку памяти) и «No Bulk Access Locality» (отсутствие блочного доступа к участку памяти). Блочный доступ к участку памяти предполагает наличие механизма передачи длинной последовательности байт, характеризующегося наличием латентности, предшествующей реальной пересылке данных. На третьем уровне иерархии вводится еще два субкласса: «No Data Volume Locality» (отсутствие разницы во времени доступа к различным областям памяти) и «Data Volume Locality» (наличие разницы во времени доступа к различным областям памяти). Далее авторы рассматривают и классифицируют известные модели параллельных вычислений. К настоящему времени этот обзор, однако, устарел, так как в нем отсутствуют модели, ориентированные на современные кластерные вычислительные системы с многоядерными ускорителями.

Обзор [1], опубликованный Жангом (Zhang), Ченом (Chen), Суном (Sun) и Миао (Miao) в 2007 г., является обновленной версией обзора [104]. Авторы разбивают все модели параллельных вычислений на три класса: модели с общей памятью, модели с распределенной памятью и модели с иерархической памятью. К моделям с общей памятью они относят PRAM, YPRAM и HPRAM. В класс моделей с распределенной памятью попадают BSP, BSPRAM, LogP, LogGP и некоторые другие. Под системами с иерархической памятью авторы понимают однопроцессорные и многопроцессорные системы с многоуровневой памятью, в которых доступ к каждому уровню различается по времени. В качестве таких уровней могут фигурировать регистровая память, кэш первого уровня, кэш второго уровня, память DRAM и так далее. К моделям с иерархической памятью авторы обзора отнесли такие разные модели, как P-HMM, UPMH, $\log_n P$, LogP-HMM и др. Данный подход к классификации моделей параллельных вычислений представляется не вполне корректным, так как измерения «общая — разделяемая память» и «одноуровневая — многоуровневая память» являются ортогональными.

Наиболее свежим является обзор [105] группы авторов (Rico-Gallego, Díaz-Martín, Manumachu, Lastovetsky), опубликованный в «ACM Computing Surveys» в 2019 г. Указанный обзор включает в себя 25 различных моделей параллельных вычислений, которые классифицируются по целевым аппаратным платформам (кластеры с однопроцессорными узлами, кластеры с гетерогенными узлами и гомогенной соединительной сетью, гетерогенные кластеры, системы на основе двухмерной соединительной сети, Мугинет-кластеры, многоуровневые гетерогенные кластеризованные грид-системы, многопроцессорные системы с сетью Infiniband, иерархические кластеры) и по следующим типам: оригинальность (основополагающие модели; производные модели), параметризация (модели, учитывающие параметры аппаратной платформы; модели, учитывающие параметры программного обеспечения промежуточного слоя), универсальность (платформенно-зависимые модели; платформенно-независимые модели). К недостаткам данного обзора следует отнести то, что авторы концентрируются на аспектах, связанных с затратами на коммуникации, отдавая предпочтение моделям, ориентированным на приложения, в которых затраты на коммуникации превалируют над затратами на вычисления.

Заключение

В данном обзоре сделана попытка рассмотреть и классифицировать все значимые модели параллельных вычислений, известные к настоящему моменту. Рассмотрены аспекты, связанные как с оценкой времени вычислений, так и с оценкой затрат на межпроцессорные коммуникации. Вместе с непрерывным усложнением архитектур современных многопроцессорных систем неизбежно растёт сложность параллельных вычислительных моделей, стремящихся как можно более точно предсказать время выполнения параллельного алгоритма на целевой аппаратной платформе. Однако сложные модели трудно использовать на практике. С другой стороны, простые классические модели типа BSP или LogP уже не могут обеспечить достаточную адекватность применительно к современным вычислительным кластерам и другим многопроцессорным платформам. Возможный выход из сложившейся ситуации заключается в том, что для создания современных простых и адекватных моделей параллельных вычислений необходимо ограничивать не только класс аппаратной платформы, но и класс алгоритмов, для которых предназначена модель.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00352 а, Правительства РФ в соответствии с Постановлением №211 от 16.03.2013 г. (соглашение № 02.А03.21.0011) и Министерства образования и науки РФ (государственное задание 2.7905.2017/8.9).

Литература/References

1. Zhang Y. et al. Models of Parallel Computation: a Survey and Classification // Frontiers of Computer Science in China. Higher Education Press, 2007. Vol. 1, No. 2. P. 156–165. DOI: 10.1007/s11704-007-0016-1.
2. Valiant L.G. A Bridging Model for Parallel Computation // Communications of the ACM. 1990. Vol. 33, No. 8. P. 103–111. DOI: 10.1145/79173.79181.
3. Campbell D.K.G. A Survey of Models of Parallel Computation. Technical Report No.YCS-97-278. 1997. 37 p.
4. Shepherdson J.C., Sturgis H.E. Computability of Recursive Functions // Journal of the ACM. ACM, 1963. Vol. 10, No. 2. P. 217–255. DOI: 10.1145/321160.321170.
5. Elgot C.C., Robinson A. Random-Access Stored-Program Machines, an Approach to Programming Languages // Journal of the ACM. ACM, 1964. Vol. 11, No. 4. P. 365–399. DOI: 10.1145/321239.321240.
6. Hartmanis J. Computational Complexity of Random Access Stored Program Machines // Mathematical Systems Theory. Springer-Verlag, 1971. Vol. 5, No. 3. P. 232–245. DOI: 10.1007/BF01694180.
7. Cook S.A., Reckhow R.A. Time Bounded Random Access Machines // Journal of Computer and System Sciences. Academic Press, 1973. Vol. 7, No. 4. P. 354–375. DOI: 10.1016/S0022-0000(73)80029-7.
8. Aho A. V., Hopcroft J.E., Ullman J.D. The Design and Analysis of Computer Algorithms. London, Amsterdam, Don Mills, Ontario, Sydney: Addison-Wesley, 1974. 470 p.
9. Skillicorn D.B., Talia D. Models and Languages for Parallel Computation // ACM Computing Surveys. 1998. Vol. 30, No. 2. P. 123–169. DOI: 10.1145/280277.280278.

10. Fortune S., Wyllie J. Parallelism in Random Access Machines // Proceedings of the Tenth Annual ACM Symposium on Theory of Computing — STOC'78. New York, New York, USA: ACM Press, 1978. P. 114–118. DOI: 10.1145/800133.804339.
11. Culler D. et al. LogP: Towards a Realistic Model of Parallel Computation // Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming — PPOPP'93. New York, New York, USA: ACM Press, 1993. P. 1–12. DOI: 10.1145/155332.155333.
12. Yuan L. et al. LogGPH: A Parallel Computational Model with Hierarchical Communication Awareness // Proceedings of the 2010 13th IEEE International Conference on Computational Science and Engineering — CSE'10. Washington, DC, US: IEEE Computer Society, 2010. P. 268–274. DOI: 10.1109/CSE.2010.40.
13. Lu F., Song J., Pang Y. HLognGP: A Parallel Computation Model for GPU clusters // Concurrency and Computation: Practice and Experience. 2015. Vol. 27, No. 17. P. 4880–4896. DOI: 10.1002/cpe.3475.
14. Qiao X., Chen S., Yang L.T. HPM: a Hierarchical Model for Parallel Computations // International Journal of High Performance Computing and Networking. 2004. Vol. 1, No. 1–3. P. 117–127. DOI: 10.1504/IJHPCN.2004.007571.
15. Rico-Gallego J.-A., Díaz-Martín J.-C. τ -Lop: Modeling Performance of Shared Memory MPI // Parallel Computing. North-Holland, 2015. Vol. 46. P. 14–31. DOI: 10.1016/J.PARCO.2015.02.006.
16. Rico-Gallego J.-A., Lastovetsky A.L., Diaz-Martin J.-C. Model-Based Estimation of the Communication Cost of Hybrid Data-Parallel Applications on Heterogeneous Clusters // IEEE Transactions on Parallel and Distributed Systems. 2017. Vol. 28, No. 11. P. 3215–3228. DOI: 10.1109/TPDS.2017.2715809.
17. Bilardi G. et al. On the Effectiveness of D-BSP as a Bridging Model of Parallel Computation // Proceedings of the International Conference on Computational Science — ICCS'01. Part II. Lecture Notes in Computer Science, Vol. 2074. Berlin, Heidelberg: Springer, 2001. P. 579–588. DOI: 10.1007/3-540-45718-6_63.
18. Ежова Н.А., Соколинский Л.Б. Модель параллельных вычислений для многопроцессорных систем с распределенной памятью // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2018. Том 7, № 2. С. 32–49. DOI: 10.14529/cmse180203. [Ezhova N.A., Sokolinsky L.B. Parallel Computational Model for multiprocessor Systems With Distributed Memory. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2018. vol. 7, no. 2. pp. 32–49. (in Russian) DOI: 10.14529/cmse180203.]
19. Ежова Н.А., Соколинский Л.Б. Исследование масштабируемости итерационных алгоритмов при суперкомпьютерном моделировании физических процессов // Вычислительные методы и программирование. 2018. Том 19, № 4. С. 416–430. DOI: 10.26089/NumMet.v19r437. [Ezhova N.A., Sokolinsky L.B. Scalability Evaluation of Iterative Algorithms for Supercomputer Simulation of Physical Processes. *Numerical methods and programming*. 2018. vol. 19, no. 4. pp. 416–430. (in Russian) DOI: 10.26089/NumMet.v19r437.]
20. Ceze L.H. Shared-Memory Multiprocessors // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 1810–1812. DOI: 10.1007/978-0-387-09766-4_142.
21. Nayfeh B.A., Olukotun K. A Single-chip Multiprocessor // Computer. 1997. Vol. 30, No. 9. P. 79–85. DOI: 10.1109/2.612253.
22. Bardine A. et al. NUMA Caches // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 1329–1338. DOI: 10.1007/978-0-387-09766-4_16.

23. Snir M. Distributed-Memory Multiprocessor // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 574–578.
24. Pfister G.F. In Search of Clusters. 2nd Edition. Upper Saddle River, NJ: Prentice Hall, 1998. 575 p.
25. Beowulf Cluster Computing with Linux / ed. Sterling T.L. Cambridge, London: MIT Press, 2002. 496 p.
26. Owens J.D. et al. GPU Computing // Proceedings of the IEEE. 2008. Vol. 96, No. 5. P. 879–899. DOI: 10.1109/JPROC.2008.917757.
27. Rochange C., Uhrig S., Sainrat P. Memory Hierarchy // Time-Predictable Architectures. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2014. P. 69–104. DOI: 10.1002/9781118790229.ch4.
28. Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach // Computer. Fifth Edit. Morgan Kaufmann, 2011. 856 p.
29. Bottomley J. Understanding Caching // Linux Journal. 2004. No. 117. P. 58–62.
30. Wu K. et al. Early Evaluation of Intel Optane Non-Volatile Memory with HPC I/O Workloads // arXiv:1708.02199v2 [cs.DC]. 2017. 6 p.
31. Yang C.-T., Huang C.-L., Lin C.-F. Hybrid CUDA, OpenMP, and MPI Parallel Programming on Multicore GPU Clusters // Computer Physics Communications. North-Holland, 2011. Vol. 182, No. 1. P. 266–269. DOI: 10.1016/J.CPC.2010.06.035.
32. Bilardi G., Pietracaprina A. Models of Computation, Theoretical // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 1150–1158. DOI: 10.1007/978-0-387-09766-4_218.
33. Skillicorn D.B. Parallelism and the Bird-Meertens Formalism. Kingston, Canada, 1992. 16 p.
34. Bilardi G., Pietracaprina A., Pucci G. A Quantitative Measure of Portability with Application to Bandwidth-Latency Models for Parallel Computing // Euro-Par'99 Parallel Processing. Euro-Par 1999. Lecture Notes in Computer Science, Vol 1685. Springer, Berlin, Heidelberg, 1999. P. 543–551. DOI: 10.1007/3-540-48311-X_76.
35. Grama A. et al. Architecture Independent Analysis of Parallel Programs // Proceedings of the International Conference on Computational Science — ICCS'01. Part II. Lecture Notes in Computer Science, Vol. 2074. Berlin, Heidelberg: Springer, 2001. P. 599–608. DOI: 10.1007/3-540-45718-6_65.
36. JaJa J.F. PRAM (Parallel Random Access Machines) // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 1608–1615. DOI: 10.1007/978-0-387-09766-4_23.
37. Goldschlager L.M. A Unified Approach to Models of Synchronous Parallel Machines // Proceedings of the Tenth Annual ACM Symposium on Theory of Computing — STOC'78. New York, New York, USA: ACM Press, 1978. P. 89–94. DOI: 10.1145/800133.804336.
38. Ladner R.E., Fischer M.J. Parallel Prefix Computation // Journal of the ACM. 1980. Vol. 27, No. 4. P. 831–838. DOI: 10.1145/322217.322232.
39. JaJa J.F. An Introduction to Parallel Algorithms. Redwood City, CA, USA: Addison Wesley Publishing Co., Reading, 1992. 576 p.
40. Darema F. et al. A Single-Program-Multiple-Data Computational Model for EPEX/FORTRAN // Parallel Computing. 1988. Vol. 7, No. 1. P. 11–24. DOI: 10.1016/0167-8191(88)90094-4.
41. Darema F. SPMD Computational Model // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 1933–1943. DOI: 10.1007/978-0-387-09766-4_26.
42. Cook S., Dwork C., Reischuk R. Upper and Lower Time Bounds for Parallel Random Access Machines without Simultaneous Writes // SIAM Journal on Computing. Society

- for Industrial and Applied Mathematics, 1986. Vol. 15, No. 1. P. 87–97. DOI: 10.1137/0215006.
43. Karp R.M., Ramachandran V. Parallel Algorithms for Shared-Memory Machines // Handbook of theoretical computer science. Volume A: Algorithms and Complexity / ed. Van Leeuwen J. Amsterdam, New York, Oxford, Tokyo: Elsevier, 1990. P. 871–941.
 44. Pippenger N. On Simultaneous Resource Bounds // 20th Annual Symposium on Foundations of Computer Science (SFCS 1979). San Juan, Puerto Rico: IEEE, 1979. P. 307–311. DOI: 10.1109/SFCS.1979.29.
 45. Pippenger N. Pebbling with an Auxiliary Pushdown // Journal of Computer and System Sciences. Academic Press, 1981. Vol. 23, No. 2. P. 151–165. DOI: 10.1016/0022-0000(81)90011-8.
 46. Snyder L. Type Architectures, Shared Memory, and the Corollary of Modest Potential // Annual Review of Computer Science. 1986. Vol. 1, No. 1. P. 289–317. DOI: 10.1146/annurev.cs.01.060186.001445.
 47. Mehlhorn K., Vishkin U. Randomized and Deterministic Simulations of PRAMs by Parallel Machines with Restricted Granularity of Parallel Memories // Acta Informatica. Springer-Verlag, 1984. Vol. 21, No. 4. P. 339–374. DOI: 10.1007/BF00264615.
 48. Gibbons P.B., Matias Y., Ramachandran V. The Queue-Read Queue-Write PRAM Model: Accounting for Contention in Parallel Algorithms // SIAM Journal on Computing. 1998. Vol. 28, No. 2. P. 733–769. DOI: 10.1137/S009753979427491.
 49. Gibbons P.B., Matias Y. Can a Shared-Memory Model Serve as a Bridging Model for Parallel Computation? // Theory of Computing Systems. 1999. Vol. 32, No. 3. P. 327–359. DOI: 10.1007/s002240000121.
 50. Aggarwal A., Chandra A.K., Snir M. On Communication Latency in PRAM Computations // Proceedings of the First Annual ACM Symposium on Parallel Algorithms and Architectures — SPAA'89. New York, New York, USA: ACM Press, 1989. P. 11–21. DOI: 10.1145/72935.72937.
 51. Mansour Y., Nisan N., Vishkin U. Trade-offs between Communication Throughput and Parallel Time // Journal of Complexity. Academic Press, 1999. Vol. 15, No. 1. P. 148–166. DOI: 10.1006/JCOM.1998.0498.
 52. Cole R., Zajicek O. The APRAM: Incorporating Asynchrony into the PRAM Model // Proceedings of the First Annual ACM Symposium on Parallel Algorithms and Architectures — SPAA'89. New York, New York, USA: ACM Press, 1989. P. 169–178. DOI: 10.1145/72935.72954.
 53. Gibbons P.B. A More Practical PRAM Model // Proceedings of the First Annual ACM Symposium on Parallel Algorithms and Architectures — SPAA'89. New York, New York, USA: ACM Press, 1989. P. 158–168. DOI: 10.1145/72935.72953.
 54. VALIANT L.G. General Purpose Parallel Architectures // Handbook of Theoretical Computer Science (Vol. A): Algorithms and Complexity. Elsevier, 1990. P. 943–971. DOI: 10.1016/B978-0-444-88071-0.50023-0.
 55. de la Torre P., Kruskal C.P. Towards a Single Model of Efficient Computation in Real Parallel Machines // Future Generation Computer Systems. North-Holland, 1992. Vol. 8, No. 4. P. 395–408. DOI: 10.1016/0167-739X(92)90071-I.
 56. Heywood T., Ranka S. A Practical Hierarchical Model of Parallel Computation I. The model // Journal of Parallel and Distributed Computing. Academic Press, 1992. Vol. 16, No. 3. P. 212–232. DOI: 10.1016/0743-7315(92)90034-K.

57. Forsell M. A PRAM-NUMA Model of Computation for Addressing Low-TLP Workloads // International Journal of Networking and Computing. [Hiroshima University], 2011. Vol. 1, No. 1. P. 21–35.
58. Ranade A.G. How to Emulate Shared Memory // Journal of Computer and System Sciences. Academic Press, 1991. Vol. 42, No. 3. P. 307–326. DOI: 10.1016/0022-0000(91)90005-P.
59. Forsell M. et al. Hardware and Software Support for NUMA Computing on Configurable Emulated Shared Memory Architectures // 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum. IEEE, 2013. P. 640–648. DOI: 10.1109/IPDPSW.2013.146.
60. Forsell M. E — A Language for Thread-Level Parallel Programming on Synchronous Shared Memory NOCs // WSEAS Transactions on Computers. 2004. Vol. 3, No. 3. P. 807–812.
61. Forsell M., Leppanen V. An Extended PRAM-NUMA Model of Computation for TCF Programming // International Journal of Networking and Computing. 2013. Vol. 3, No. 1. P. 98–115.
62. Aggarwal A. et al. A Model for Hierarchical Memory // Proceedings of the Nineteenth annual ACM Conference on Theory of Computing — STOC'87. New York, New York, USA: ACM Press, 1987. P. 305–314. DOI: 10.1145/28395.28428.
63. Aggarwal A., Chandra A.K., Snir M. Hierarchical Memory with Block Transfer // 28th Annual Symposium on Foundations of Computer Science (sfcs 1987). IEEE, 1987. P. 204–216. DOI: 10.1109/SFCS.1987.31.
64. Luccio F., Pagli L. A Model of Sequential Computation with Pipelined Access to Memory // Mathematical Systems Theory. Springer-Verlag, 1993. Vol. 26, No. 4. P. 343–356. DOI: 10.1007/BF01189854.
65. Mead C.A., Conway L.A. Introduction to VLSI systems. Boston, MA, USA: Addison-Wesley, 1980. 396 p.
66. Alpern B. et al. The Uniform Memory Hierarchy Model of Computation // Algorithmica. Springer-Verlag, 1994. Vol. 12, No. 2–3. P. 72–109. DOI: 10.1007/BF01185206.
67. Vitter J.S., Shriver E.A.M. Algorithms for parallel memory, II: Hierarchical multilevel memories // Algorithmica. Springer-Verlag, 1994. Vol. 12, No. 2–3. P. 148–169. DOI: 10.1007/BF01185208.
68. Tiskin A. BSP (Bulk Synchronous Parallelism) // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 192–199. DOI: 10.1007/978-0-387-09766-4_311.
69. Goudreau M. et al. Towards Efficiency and Portability: Programming with the BSP Model // Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures — SPAA'96. New York, NY, USA: ACM Press, 1996. P. 1–12. DOI: 10.1145/237502.237503.
70. Bisseling R.H. Parallel Scientific Computation: A Structured Approach using BSP and MPI. New York: Oxford University Press, 2004. 325 P.
71. McColl W.F. Scalable Computing // J. van Leeuwen (eds). Computer Science Today: Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Berlin, Heidelberg: Springer, 1995. P. 46–61. DOI: 10.1007/BFb0015236.
72. Tiskin A. The Bulk-synchronous Parallel Random Access Machine // Theoretical Computer Science. 1998. Vol. 196, No. 1–2. P. 109–130. DOI: 10.1016/S0304-3975(97)00197-7.
73. McColl W.F., Tiskin A. Memory-Efficient Matrix Multiplication in the BSP Model // Algorithmica. Springer-Verlag, 1999. Vol. 24, No. 3–4. P. 287–297. DOI: 10.1007/PL00008264.

74. Kielmann T., Gorchatch S. Bandwidth-Latency Models (BSP, LogP) // Encyclopedia of Parallel Computing. Boston, MA: Springer US, 2011. P. 107–112. DOI: 10.1007/978-0-387-09766-4_189.
75. Alexandrov A. et al. LogGP: Incorporating Long Messages into the LogP Model for Parallel Computation // Journal of Parallel and Distributed Computing. 1997. Vol. 44, No. 1. P. 71–79. DOI: 10.1006/jpdc.1997.1346.
76. Kielmann T., Bal H.E., Verstoep K. Fast Measurement of LogP Parameters for Message Passing Platforms // Parallel and Distributed Processing. IPDPS 2000. Lecture Notes in Computer Science, Vol. 1800. Berlin, Heidelberg: Springer, 2000. P. 1176–1183. DOI: 10.1007/3-540-45591-4_162.
77. Gropp W., Lusk E., Skjellum A. Using MPI: Portable Parallel Programming with the Message-Passing Interface. Second Ed. MIT Press, 1999.
78. Gropp W. MPI 3 and Beyond: Why MPI Is Successful and What Challenges It Faces // Recent Advances in the Message Passing Interface. EuroMPI 2012. Lecture Notes in Computer Science, Vol. 7490 / ed. Träff J.L., Benkner S., Dongarra J.J. Berlin, Heidelberg: Springer, 2012. P. 1–9. DOI: 10.1007/978-3-642-33518-1_1.
79. Touyama T., Horiguchi S. Parallel Computation Model LogPQ // High Performance Computing. ISHPC 1997. Lecture Notes in Computer Science, vol 1336 / ed. Polychronopoulos C., Joe K., Araki K. A.M. Berlin, Heidelberg: Springer, 1997. P. 327–334. DOI: 10.1007/BFb0024227.
80. Touyama T., Horiguchi S. Performance Evaluation of Practical Parallel Computation Model LogPQ // Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99). Washington, DC, USA: IEEE Computer Society, 1999. P. 216–221. DOI: 10.1109/ISPAN.1999.778942.
81. Palmer J., Steele G.L. Connection Machine model CM-5 System Overview // Frontiers'92, the Fourth Symposium on the Frontiers of Massive Parallel Computation, October 19-21, 1992, McLean, Virginia. IEEE Computer Society Press, 1992. P. 474–483. DOI: 10.1109/FMPC.1992.234877.
82. Ino F., Fujimoto N., Hagihara K. LogGPS: A Parallel Computational Model for Synchronization Analysis // ACM SIGPLAN Notices. 2001. Vol. 36, No. 7. P. 133–142. DOI: 10.1145/568014.379592.
83. Gropp W. et al. A High-performance, Portable Implementation of the MPI Message Passing Interface Standard // Parallel Computing. 1996. Vol. 22, No. 6. P. 789–828. DOI: 10.1016/0167-8191(96)00024-5.
84. Moritz C.A. et al. LoGPC: Modeling Network Contention in Message-Passing Programs // ACM SIGMETRICS Performance Evaluation Review. New York, New York, USA: ACM Press, 1998. Vol. 26, No. 1. P. 254–263. DOI: 10.1145/277851.277933.
85. Moritz C.A., Frank M.I. LoGPC: Modeling Network Contention in Message-Passing Programs // IEEE Transactions on Parallel and Distributed Systems. 2001. Vol. 12, No. 4. P. 404–415. DOI: 10.1109/71.920589.
86. Agarwal A. et al. The MIT Alewife Machine: A Large-Scale Distributed-Memory Multiprocessor // Scalable Shared Memory Multiprocessors. Proceedings of a workshop held May 26-27, 1990, in Seattle, Wash. / ed. Dubois M., Thakkar S. Boston, MA: Springer, 1992. P. 239–261. DOI: 10.1007/978-1-4615-3604-8_13.
87. Kubiawicz J., Agarwal A. Anatomy of a Message in the Alewife multiprocessor // ACM International Conference on Supercomputing 25th Anniversary Volume. New York, NY, USA: ACM Press, 2014. P. 193–204. DOI: 10.1145/2591635.2667168.

88. Cameron K.W., Ge R., Sun X.-H. lognP and log3P: Accurate Analytical Models of Point-to-point Communication in Distributed Systems // IEEE Transactions on Computers. 2007. Vol. 56, No. 3. P. 314–327. DOI: 10.1109/TC.2007.38.
89. Cameron K.W., Ge R. Predicting and Evaluating Distributed Communication Performance // Proceedings of the 2004 ACM/IEEE Conference on Supercomputing. IEEE, 2004. P. 15. DOI: 10.1109/SC.2004.40.
90. Cameron K.W., Sun X.-H. Quantifying Locality Effect in Data Access Delay: Memory logP // Proceedings of the 2003 IEEE International Parallel and Distributed Processing Symposium (IPDPS'03). IEEE Comput. Soc., 2003. P. 8. DOI: 10.1109/IPDPS.2003.1213137.
91. Cappello F. et al. HiHCoHP-Toward a Realistic Communication Model for Hierarchical Hyperclusters of Heterogeneous Processors // Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001. IEEE Comput. Soc., 2001. P. 6. DOI: 10.1109/IPDPS.2001.924978.
92. Cappello F. et al. An Algorithmic Model for Heterogeneous Hyper-Clusters: Rationale and Experience // International Journal of Foundations of Computer Science. World Scientific Publishing Company, 2005. Vol. 16, No. 02. P. 195–215. DOI: 10.1142/S0129054105002942.
93. Bosque J.L., Pastor L. A Parallel Computational Model for Heterogeneous Clusters // IEEE Transactions on Parallel and Distributed Systems. 2006. Vol. 17, No. 12. P. 1390–1400. DOI: 10.1109/TPDS.2006.165.
94. Hoefler T. et al. LogfP — a Model for Small Messages in InfiniBand // Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. Washington, DC, USA: IEEE Computer Society, 2006. P. 319–319. DOI: 10.1109/IPDPS.2006.1639624.
95. Jepsen T.C. InfiniBand // Distributed Storage Networks: Architecture, Protocols and Management. Chichester, West Sussex, England: John Wiley & Sons, 2013. P. 159–174. DOI: 10.1002/9780470871461.ch6.
96. Nasri W., Tarhouni O., Slimi N. PLP: Towards a Realistic and Accurate Model for Communication Performances on Hierarchical Cluster-based Systems // 2008 IEEE International Symposium on Parallel and Distributed Processing. IEEE, 2008. P. 1–8. DOI: 10.1109/IPDPS.2008.4536486.
97. Hoefler T., Schneider T., Lumsdaine A. LogGOPSim – Simulating Large-Scale Applications in the LogGOPS Model // Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing — HPDC'10. New York, New York, USA: ACM Press, 2010. P. 597–604. DOI: 10.1145/1851476.1851564.
98. Valiant L.G. A Bridging Model for Multi-core Computing // Journal of Computer and System Sciences. Elsevier Inc., 2011. Vol. 77, No. 1. P. 154–166. DOI: 10.1016/j.jcss.2010.06.012.
99. Tu B. et al. Performance Analysis and Optimization of MPI Collective Operations on Multicore Clusters // The Journal of Supercomputing. Springer US, 2012. Vol. 60, No. 1. P. 141–162. DOI: 10.1007/s11227-009-0296-3.
100. Tu B. et al. Accurate Analytical Models for Message Passing on Multi-core Clusters // 2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing. IEEE, 2009. P. 133–139. DOI: 10.1109/PDP.2009.18.
101. Sterling T. et al. SLOWER: A Performance Model for Exascale Computing // Supercomputing Frontiers and Innovations. 2014. Vol. 1, No. 2. P. 42–57. DOI: 10.14529/jsfi140203.

102. Gerbessiotis A. V. Extending the BSP Model for Multi-core and Out-of-core Computing: MBSP // *Parallel Computing*. Elsevier B.V., 2015. Vol. 41. P. 90–102. DOI: 10.1016/j.parco.2014.12.002.
103. Amaris M. et al. A Simple BSP-based Model to Predict Execution Time in GPU Applications // *2015 IEEE 22nd International Conference on High Performance Computing (HiPC)*. IEEE, 2015. P. 285–294. DOI: 10.1109/HiPC.2015.34.
104. Maggs B.M., Matheson L.R., Tarjan R.E. *Models of Parallel Computation: a Survey and Synthesis* // *Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences*. IEEE Comput. Soc. Press, 1995. P. 61–70. DOI: 10.1109/HICSS.1995.375476.
105. Rico-Gallego J.A. et al. A Survey of Communication Performance Models for High-Performance Computing // *ACM Computing Surveys*. ACM, 2019. Vol. 51, No. 6. P. 1–36. DOI: 10.1145/3284358.

Ежова Надежда Александровна, аспирант, кафедры системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Соколинский Леонид Борисович, д.ф.-м.н., профессор, проректор по информатизации, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse190304

SURVEY OF PARALLEL COMPUTATION MODELS

© 2019 N.A. Ezhova, L.B. Sokolinsky

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: EzhovaNA@susu.ru, Leonid.Sokolinsky@susu.ru

Received: 01.06.2019

This survey aims to present the state of the art in analytic parallel computation models, providing sufficiently detailed descriptions of particularly noteworthy efforts. Such models allow predicting the computation time, speedup, efficiency and scalability of parallel algorithms for various target multiprocessor platforms. Modeling the cost of computations and communications in multiprocessor systems is an important and challenging problem. It provides insights into the design of the parallel algorithms for optimization of their deployment in the increasingly complex high-performance computing. The survey shows the evolution of parallel computing models inspired by the evolution of multiprocessor systems, from single-level models with shared memory to multi-level hierarchical models with distributed memory, which correspond to multicore clusters. The review concludes with prospective directions for further research in the area of developing mathematical models for parallel computing.

Keywords: parallel computing model, survey, parallel programming, multiprocessor systems, performance evaluation, execution time prediction.

FOR CITATION

Ezhova N.A., Sokolinsky L.B. Survey of Parallel Computation Models. *Bulletin of the South Ural State University*. Series: Computational Mathematics and Software Engineering. 2019. vol. 8, no. 3. pp. 58–91. (in Russian) DOI: 10.14529/cmse190304.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

ТЕНДЕНЦИИ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНЫХ УЗЛОВ СОВРЕМЕННЫХ СУПЕРКОМПЬЮТЕРОВ*

© 2019 Е.О. Тютляева¹, И.О. Одинцов², А.А. Московский¹, Г.В. Мармузов¹

¹ЗАО «РСК Технологии»

(121170 Москва, пр. Кутузовский, д. 36, стр. 23),

²ООО «РСК Лабс»

(121205 Москва, ул. Большой Бульвар, д. 42, стр. 1)

E-mail: xgl@rsc-tech.ru, igor_odintsov@rsc-tech.ru, moskov@rsc-tech.ru,

gleb.marmuzov@rsc-tech.ru

Поступила в редакцию: 28.01.2019

В данной работе выполнен анализ вычислительных узлов современных суперкомпьютеров с двух точек зрения — аппаратно-компонентной и инфраструктурной. На основании проведённого анализа названы основные конструктивные элементы, которыми должен быть оснащен современный вычислительный узел. В статье приведены классификации архитектур современных универсальных и специализированных ядер с примерами; проведен обзор современных тенденций организации подсистемы памяти и внутриузлового интерконнекта; упомянуты способы использования энергонезависимых устройств хранения на узлах при организации современных высокопроизводительных систем хранения. Также разобраны основные требования к организации инфраструктуры узла современного суперкомпьютера, в частности, дана краткая классификация современных подходов к организации жидкостного охлаждения и мониторинга вычислительных узлов. Выявленные тенденции приводят к основным вариантам дизайна вычислительных узлов, состоящих из энергоэффективного универсального процессора и совокупности энергоэффективных специализированных ускорителей. В статье сделан акцент на современных технологиях, которые достигли стадии выхода в производство или, как минимум, создания рабочих прототипов. Обсуждаются современные суперкомпьютерные задачи и их отображение на архитектуру вычислительных узлов. В заключении приведено краткое обсуждение актуальных технологических проблем и основных направлений для сохранения прогресса в компьютерной отрасли.

Ключевые слова: высокопроизводительный вычислительный узел, разработка архитектуры вычислительного узла, анализ суперкомпьютерных архитектур, тенденции развития суперкомпьютеров.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Тютляева Е.О., Одинцов И.О., Московский А.А., Мармузов Г.В. Тенденции развития вычислительных узлов современных суперкомпьютеров // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 92–114. DOI: 10.14529/cmse190305.

Введение

При создании вычислительного узла для современного суперкомпьютера необходимо использование новых технологических подходов.

Простое масштабирование существующих технологий не будет эффективным решением. Задача, стоящая перед исследователями, гораздо глубже и обширнее, чем простое достижение заданной вычислительной мощности, она требует ряда инновационных решений. Можно сформулировать ряд основных задач, которые стоят перед разработчиками лидирующих суперкомпьютеров:

*Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2019»

Минимизация энергопотребления. При масштабировании текущих технологий может быть получен суперкомпьютер, обладающий энергопотреблением в сотни мегаватт. Необходимо разработать новые технологии, которые позволят уменьшить планируемое энергопотребление до экономически приемлемого уровня (20-40 MW).

Обеспечение модульности и высокой вычислительной плотности в рамках одного узла. Высокоскоростные интерфейсы в рамках одного узла.

Организация энергоэффективного межузлового интерконнекта с высокой пропускной способностью и минимальными задержками.

Организация высокопроизводительной системы хранения с достаточно высокой пропускной способностью для избежать простоев вычислительных ресурсов.

Подбор оптимальных типов памяти для построения наиболее эффективной структуры памяти узла.

Ряд вызовов связан и с программным обеспечением (ПО), где мы выделяем следующие важные аспекты:

Сложность создания массового параллельного программного обеспечения на современных языках программирования фактически поднимает вопрос о необходимости *новой парадигмы программирования*: гипер-параллельной.

Множество различных высокопроизводительных аппаратных архитектур поднимает задачу не просто переносимости программного обеспечения, а *максимально эффективной переносимости и автоматизированного отображения на архитектуру*.

Также, во многом от программной реализации зависят *надежность и отказоустойчивость* программного обеспечения.

При разработке современного узла важно отразить все ожидаемые архитектурные особенности, такие как количество процессоров, количество процессорных ядер, соотношение объема памяти и объема вычислений в узле, количество независимых вычислительных потоков [1].

Текст статьи организован следующим образом. Раздел 1 содержит классификации современных универсальных и специализированных ядер, а также описывает подходы к отображению задач на вычислительные ядра. В разделе 2 сформулированы основные конструктивные элементы, которыми должен быть оснащен современный вычислительный узел. Далее в разделе 3 разобраны основные требования к организации инфраструктуры узла современного суперкомпьютера. В разделе 4 предложены обобщенные варианты отображения задач на архитектуру вычислительных узлов. В заключении подведены итоги проведенного обзора и приведено обсуждение актуальных технологических вызовов.

1. Задачи и вычислительные ядра

Д. Рид и Д. Донгарра в статье [2] выделяют две основные экосистемы: обработка данных и вычислительная наука. Авторы подчеркивают, что эти в настоящее время эти экосистемы имеют различия как на аппаратном уровне, так и на уровне программного обеспечения. Тем не менее, практически одинаковый аппаратный уровень может быть использован как для обработки данных, так и для вычислений, а существенная разница наблюдается лишь на уровне инструментального и промежуточного программного обеспечения для ряда задач машинного обучения и глубокого обучения. Дополнительно заметим, что если ранее научные и инженерные вычисления относили к «классическим задачам НРС» и они составляли основной объем задач, то прогнозируется, что в будущем

они будут составлять около половины задач, а оставшийся объем займут задачи обработки больших данных, искусственного интеллекта, машинного обучения и глубокого обучения.

С точки зрения аппаратного уровня более корректной будет классификация отображения задач на вычислительные ядра, а также их количество с учетом масштабируемости задач на вычислительные нити.

В предлагаемом исследовании выделим два базовых класса архитектур ядер: универсальные и ускорители, а также множество их подклассов, которые очень часто определяются конкретным производителем.

Универсальные ядра (типичная роль — хост, базовые, «классические вычислительные», «толстые») можно классифицировать следующим образом:

- традиционный набор команд (CISC):
 - x86, IA-64, x86-64 (Intel, AMD);
- упрощенный набор команд (RISC):
 - POWER (IBM);
 - ARM (много производителей);
- сверхдлинное командное слово (VLIW):
 - Эльбрус (МЦСТ).

Типов и классов специализированных ядер достаточно много. С одной стороны, обратим внимание на то, что математические сопроцессоры и сопроцессоры ввода-вывода остались историей и их функциональность реализуется в универсальных ядрах. С другой стороны, создаются новые перспективные ускорители, например, квантовые. Немаловажен тот факт, что специализированные ядра кроме их эффективности на определенном классе задач, являются также и энергоэффективными.

Специализированные ядра (типичная роль — ускоритель, сопроцессор, «тонкие») могут быть классифицированы следующим образом:

- гомогенные (на базе упрощенных универсальных), например Intel Xeon Phi [3] (Intel);
- вычислительные (графические) ускорители (GPU, GPGPU), например Tesla [4] (NVIDIA);
- узкоспециализированные (оптическое быстрое преобразование Фурье);
- тензорные ускорители (матричное умножение и свертка), например TPU [5] (Google);
- нейроморфные (самообучающиеся), например Loihi [6] (Intel);
- ускорители алгоритмов работы машинного зрения, например Movidius (Intel);
- квантовые (криптография, искусственный интеллект, молекулярное моделирование), например Tangle Lake [7] (Intel);
- перепрограммируемые (FPGA), например Intel Stratix 10 SX. FPGA [8].

Подчеркнем тот факт, что многие задачи считаются как на универсальных ядрах, так и на специализированных (в том числе и упрощенных универсальных). Конечно, важен такой критерий как эффективность, в котором интересную роль играет точность вычислений:

- двойная точность — необходима в тех сферах, где появление ошибок является недопустимым (большинство научных задач, инженеринговых задач, ...);
- одинарная точность — допустима для задач симуляции, игровой физики;
- половинная точность — используется для глубокого обучения.

Таким образом, есть несколько основных аспектов, общих для всех перечисленных научных и инженерных направлений. Прежде всего, это широкий спектр временных и пространственных масштабов; и сложные, нелинейные пересечения множества

биологических и физических процессов. Это все требует качественного вычислительного моделирования, над которыми должны работать объединенные исследовательские группы из нескольких научных областей. Также важно учитывать существование огромных объемов разнообразных научных данных и беспрецедентные возможности для выявления междисциплинарных корреляций и статистического анализа. Во всех областях, от биологии до бизнеса, большие данные создают новые исследовательские возможности и предъявляют новые требования. Существуют оценки, например, в отчете DOE [9], что приблизительные вычислительные требования ряда задач к 2025 году возрастут в 100–1000 раз.

2. Компоненты узла суперкомпьютера

В данной статье мы сфокусируемся на основном конструктивном элементе — современном вычислительном узле.

Каждый узел должен быть оснащен следующими конструктивными элементами:

Процессоры с универсальными ядрами. Данные процессоры должны взять на себя основную нагрузку при решении вычислительных задач. Эти ядра также должны обеспечивать достаточную производительность для фрагментов кода, не применимых для расчетов на специальных ускорителях.

Ускорители со специализированными ядрами. Ускорители, в первую очередь, должны быть ориентированы на эффективное решение задач анализа данных; на вычислительных задачах должны использоваться для проведения специализированных вычислений. Кроме того, применение специализированных ускорителей должно способствовать достижению необходимой производительности (для решения перечисленных выше задач, стоящих на переднем крае науки) при этом уложившись в разумные границы энергопотребления.

Материнская плата и другие платы.

ОЗУ. В связи с возросшими объемами данных и актуальностью задач по обработке, классификации и анализу больших данных, к памяти на узле выдвигаются достаточно серьезные требования, касающиеся объема и пропускной способности. Возможно использование иерархических решений с высокопроизводительной оперативной памятью, дополненной Software-defined memory на базе NVMe (спецификация на протоколы доступа к твердотельным накопителям (SSD), подключенным по шине PCI Express).

Энергонезависимая память. Для организации уровня локального хранения на узле — может использоваться как кэш к системе хранения или элемент гиперконвергентной инфраструктуры.

Внутриузловой (внутренний) интерконнект. Один из ключевых элементов, обеспечивающих эффективность совместной работы всех вычислительных компонентов узла.

Высокопроизводительная фабрика, для обеспечения высокой пропускной способности при построении суперкомпьютера.

Контроллеры, обеспечивающие эффективное управление элементами инженерной инфраструктуры.

Простейшая классификация узлов может выглядеть так:

- гомогенный узел — узел, состоящий из однотипных ядер (как правило, универсальных);
- гибридный узел — узел, состоящий из универсальных ядер и ядер-ускорителей;

– гиперконвергентный узел — узел, полностью интегрирующий уровень хранения с уровнем обработки.

2.1. Универсальные ядра

При выборе процессоров с крупными ядрами следует рассматривать зарекомендовавших себя производителей, выпускающих процессоры общего назначения с крупными ядрами.

В настоящее время можно ориентироваться на данные рейтинга ТОП-500 и решения, разрабатываемые при создании прототипов суперкомпьютеров класса экзаскейл. Рассмотрим наиболее показательные архитектуры:

По данным статистики [10], 46,6% систем в рейтинге TOP-500 за ноябрь 2018-го года построены на базе *архитектур с традиционным набором команд (CISC)* — Intel Xeon E5 (Broadwell). Второе место занимают Xeon Gold (19,8%) и третье — Intel Xeon E5 (Haswell) (14,2%). Также архитектуру x86 планируется использовать для pre-exascale суперкомпьютера Frontera. Согласно заявлению директора TACC Dan Stanzone [11] выбор был остановлен на будущих процессорах Intel Xeon SP Platinum (Cascade Lake).

Процессоры с упрощенным набором команд (RISC) используются в суперкомпьютерах Summit и Sierra, которые по данным рейтинга ТОП-500 на ноябрь 2018-го года занимают первое и второе места в мире соответственно. Они базируются на процессорах IBM POWER9. IBM POWER9 поддерживает самые передовые технологии внутриузловое интерконнекта, включая NVIDIA NVLink, OpenCAPI и PCIe Gen4. Каждый узел суперкомпьютера Summit оснащен 2-мя процессорами IBM POWER9 и шестью ускорителями NVIDIA Tesla V100. Кроме этого, суперкомпьютер Sunway TaihuLight, разработанный в Китае, по данным рейтинга ТОП-500 на ноябрь 2018-го года занимает третье место в мире и базируется на процессорах Sunway SW26010. Это разработанные в Китае процессоры на базе 64-битной архитектуры RISC с технологией 28 нм.

Если говорить о универсальных ядрах с архитектурой ARM, то здесь следует упомянуть европейский проект ExaNode [12] в рамках которого проектируется многоядерный процессор общего назначения на базе ARMv8 CPU, плюс набор ускорителей FPGA. Кроме этого, согласно открытым источникам, японский проект построения эксафлопсного суперкомпьютера Post-K [13] также базируется на ARM-v8. Процессор Post-K — это вариант архитектуры ARMv8-A, но с 512-битным SVE векторным расширением с добавленным набором математических инструкций. Наконец, самый большой суперкомпьютер на базе архитектуры ARM — это Astra [14]. Производительность суперкомпьютера Astra достигает 2,322 PFLOPS с двойной точностью. Каждый узел базируется на двух процессорах ARM Cavium ThunderX2, без дополнительных ускорителей.

На данный момент выбор представленных на рынке архитектур и моделей процессоров с универсальными ядрами очень широк. Поэтому следует тщательно подходить к определению ключевых критериев и необходимых характеристик целевого вычислителя. К базовым характеристикам процессоров относят производительность и энергопотребление. Но следует также учитывать показатели надежности, совместимость с различными видами памяти, поддержку внутриузловых высокопроизводительных каналов передачи данных, совместимость со специализированными сопроцессорами, наличие прикладного программного стека (математических библиотек, оптимизированных для архитектуры прикладных библиотек, таких как BLAS, ScaLAPACK, FFTW, OpenFOAM

и т.п.), количество сокетов на плате и т.п. Все перечисленные аспекты оказывают существенное влияние на итоговую производительность целевого суперкомпьютера и удобство эксплуатации.

2.2. Специализированные ядра

Специализированные ядра, такие как ускорители и сопроцессоры чаще всего имеют узкую специфику, но за счет этого позволяют получить рекордное соотношение производительность/энергоэффективность при решении целевых задач.

Математические ускорители, как правило, интегрированы в основной процессор, остальные добавляются в вычислительный узел при помощи высокоскоростных каналов данных.

Если анализировать мировые тенденции, то лидирующее место среди ускорителей занимают GPGPU. Так по данным статистики [10], 12,8% систем в рейтинге TOP-500 за ноябрь 2018-го года используют ускорители NVIDIA Pascal — это первое место в статистике ускорителей и со-процессоров. Второе место (9,2%) и третье (2,6%) занимают системы, использующие соответственно NVIDIA Volta и NVIDIA Kepler. На четвертом месте появляются системы на базе Intel Xeon Phi (6 систем, 1,2%). Pre-Exascale суперкомпьютер Summit, занимающий лидирующую позицию в рейтинге TOP-500 за ноябрь 2018 также использует GPGPU ускорители. Каждый вычислительный узел оснащен шестью NVIDIA Tesla V100.

Другим направлением при ускорении вычислений является использование реконфигурируемой логики. В отличие от графических ускорителей, реконфигурируемые ускорители FPGA — это многоцелевые вычислительные устройства. Отличительными свойствами FPGA являются высокая пропускная способность ввода-вывода, и гибкий настраиваемый мелкозернистый параллелизм.

В настоящее время доступен широкий спектр решений на базе FPGA, например Xilinx 7-Series FPGAs [15] или программируемая вентиляционная матрица Intel Stratix 10 SX FPGA, выпущенная в 2018-м году, позволяет получить производительность до 10 TFlops с одинарной точностью [8].

Наконец, необходимо постоянно отслеживать прогресс в области перспективных направлений разработки ускорителей. В частности, интерес вызывают оптические процессоры и криптоакселераторы. Например, в настоящее время созданы оптические процессоры, которые позволяют выполнять быстрое преобразование Фурье практически мгновенно [16].

Использование ускорителей может позволить добиться необходимой производительности целевого суперкомпьютера и уложиться в разумный энергетический бюджет. Выбор ускорителей прежде всего зависит от специфики задач, которые будут решаться на целевом суперкомпьютере. В настоящее время при создании топовых суперкомпьютеров наблюдается тенденция к разработке многоцелевых машин, которые используются для широкого спектра задач, включая задачи вычислительного моделирования в различных областях науки и задачи обработки больших объемов данных. Таким образом, при выборе ускорителей необходимо учитывать потенциальную возможность применения ускорителей выбранного класса для решения большинства целевых задач и наличие стека программного обеспечения, который позволит эффективно использовать полученный гетерогенный суперкомпьютер.

2.3. Память

2.3.1. Оперативная память (ОЗУ)

К оперативной памяти предъявляются очень высокие требования. Практически все консорциумы заявляют о необходимости использования высокоскоростных интерфейсов и, дополнительно к ним, энергонезависимой памяти.

Например, для решения задач экзафлопсного масштаба также потребуются значительные объемы оперативной памяти на узле. В статье разработчиков аппаратно-программной платформы «Эльбрус» [17] формулируются следующие требования: 5 ПБ оперативной памяти с пропускной способностью 4 ТБ/с. Согласно оценке Coral-2, размер оперативной памяти должен быть не менее 8 ПБ; только проведение стандартных тестов потребует 5 ПБ) [18].

Согласно докладу Al Gara (Intel Fellow, Data Center Group), для того, чтобы суперкомпьютер класса экзафлопс смог успешно решать как задачи вычислительного моделирования, так и задачи искусственного интеллекта, оперативная память системы должна удовлетворять следующим требованиям: Объем оперативной памяти равен 6–12 ПБ, пропускная способность оперативной памяти — 100–200 ПБ/с, а энергонезависимой памяти I/O — 10–100 ТБ/с [19].

В настоящее время можно выделить следующие подходы к организации оперативной памяти на узле:

DDR-SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory — синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных). Синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных. Самая распространенная технология, поддерживается большинством процессоров. В настоящий момент лидирующий стандарт: DDR4. В следующем году ожидается выпуск стандарта DDR5 [20].

3D Stacked Memory — это технология трехмерного размещения памяти, которая позволяет интегрировать ОЗУ и логические блоки микропроцессора, тем самым существенно увеличивая пропускную способность [21]. Следует отметить, что плотность организации 3D Stacked Memory требует специального подхода к охлаждению — требуется либо специальная организация воздушных потоков, либо жидкостное охлаждение. Можно выделить следующие виды: *HBM (High Bandwidth Memory)* — память с высокой пропускной способностью). По конструкции HBM — это непланарная память с трехмерной конструкцией в виде куба или прямоугольный параллелепипед. В HBM несколько микросхем памяти сложены друг над другом, чтобы сформировать кубическую структуру. Благодаря этому снижается площадь, занимаемая чипами памяти, что делает возможным размещение ее в непосредственной близости к графическому процессору [22]. На текущий момент лидирующим является поколение HBM2. *HMC (Hybrid Memory Cube)*. Предоставляет пропускную способность до 480 ГБ/с на устройство, но обладает лимитированным объемом — до 8 ГБ, согласно стандартам, определенным консорциумом [23].

SSD DIMM — SSD накопители с интерфейсом DIMM, также есть варианты 3dX Point накопителей с интерфейсом DIMM. Такое решение может использоваться в качестве дополнительного уровня ОЗУ. Достоинствами данного подхода являются большой объем. Узким местом все еще остается доступное количество циклов перезаписи.

SDM (Software Defined Memory) — программно определяемая память. При наличии соответствующих программно-аппаратных решений может быть организован дополнительный уровень ОЗУ на базе энергонезависимой памяти. Достоинствами данного подхода является большой объем, а к недостаткам можно отнести небольшую пропускную способность по сравнению с DDR. Примером может являться технология IMDT [24] на базе 3DXPoint NVMe накопителей.

Графическая память. Память предназначенная для использования в графических картах. GDDR (Graphics Double Data Rate) — память предназначенная для использования в графических картах (видеокартах). Подвид энергозависимой динамической памяти с произвольным доступом (DRAM) и удвоенной скоростью передачи данных (DDR). В настоящее время доступны стандарты GDDR5 [25] и GDDR6 [26].

Наиболее популярным в настоящее время остается стандарт DDR-SDRAM, т.к. именно этот стандарт поддерживается большинством процессорных архитектур. Технология 3D Stacked Memory позволяет значительно увеличить пропускную способность, но обладает лимитированным объемом. Большой интерес представляют многоуровневые иерархии памяти позволяющие объединять технологии 3D Stacked Memory и DDR-SDRAM или DDR-SDRAM и SDM для получения больших объемов ОЗУ с высокой пропускной способностью.

2.3.2. Энергонезависимая память

Хотя анализ организации систем хранения выходит за рамки данной статьи, необходимо учесть, что на самом узле также должны присутствовать устройства для энергонезависимого хранения информации.

В настоящее время популярным решением становится интегрирование мощностей хранения в вычислительные узлы и объединение высокопроизводительным интерконнектом. В зависимости от выбранной конфигурации это могут быть твердотельные накопители (в том числе более дорогостоящие NVMe). Отдельное хранилище в большинстве случаев не может обеспечить пропускную способность, необходимую для задач обработки данных. В настоящее время можно выделить два основных пути для обеспечения надлежащих характеристик ввода-вывода:

- создание промежуточного буфера для работы с данными;
- полная интеграция уровня хранения с уровнем обработки (гиперконвергентность).

Создание промежуточного буфера (Burst Buffer [27]) для работы с данными подразумевает наличие твердотельных накопителей (в том числе возможно использование устройств SSD с интерфейсами DIMM и PCI-e для увеличения пропускной способности) для увеличения производительности ввода-вывода. Эти накопители могут быть установлены как непосредственно на вычислительных узлах, так и на специальных выделенных узлах, которые объединены высокопроизводительным интерконнектом наравне с вычислительными узлами.

Также используется специальное ПО, которое позволяет в фоновом режиме подкачивать на Burst Buffer данные для обработки, и перемещать в постоянное хранилище полученные результаты. Таким образом, приложение пользователя большую часть времени работает с локальными накопителями для быстрого ввода-вывода информации, а трансфер данных между Burst Buffer и основным хранилищем выполняется в фоновом режиме специальным ПО.

Например, на суперкомпьютере NERSC Cori используется BurstBuffer [28], организованный с использованием технологии DataWarp [29] от Cray.

Полная интеграция уровня хранения с уровнем обработки (гиперконвергентность) — это решение, которое используется в больших, гипер-масштабируемых ЦОД, таких как Google, Facebook или Amazon Web Services. Многие современные HPC платформы видят будущее именно в создании гиперконвергентных решений [30].

Группа компаний PCK также успешно разработала и продемонстрировала на европейской суперкомпьютерной выставке ISC'18 Гиперконвергентный вычислительный узел «PCK Торнадо» с прямым жидкостным охлаждением с использованием твердотельных дисков Intel SSD DC P4511 (NVMe, M.2) и Intel Optane SSD DC P4800X M.2 Series с Intel Memory Drive Technology (IMDT) [31].

2.4. Внутриузловой интерконнект

Процессоры, ускорители и память должны быть архитектурно интегрированы для эффективного взаимодействия.

В настоящее время внутриузловой интерконнект становится главным узким местом серверных узлов. Для создания современного высокопроизводительного узла необходимо рассмотреть лидирующие и перспективные варианты организации внутриузлового интерконнекта, различные топологии и технологии кластеризации (die-stacking).

Возможные варианты объединения можно классифицировать на два подхода:

- объединение высокоскоростными интерфейсами;
- интеграция на одном чипе:
 - интеграция универсальных и специализированных ядер на одном чипе;
 - интеграция вычислительных ядер и памяти на одном чипе (Memory-driven computing).

Объединение основного процессора и ускорителей-сопроцессоров высокоскоростными интерфейсами является классическим подходом, который позволяет разработчику серверного решения интегрировать топовые решения от ведущих мировых производителей.

Доминирующим решением в этой области является стандарт PCIe. В настоящее время актуальны спецификации версий 3.0 и 4.0.

PCIe 3.0. Пропускная способность у которого достигает порядка 1 ГБ/с на одиночную линию; интегрированная пропускная способность на 16 линий может достигать 32 ГБ/с в двух направлениях. Стандарт PCIe 3.0 поддерживается большинством производителей процессоров, сопроцессоров, а также устройств ввода-вывода.

PCIe 4.0. Обновленный стандарт PCIe 4.0 поддерживает двукратное увеличение скорости передачи данных (до 2 ГБ/с на одиночную линию и до 64 ГБ/с на 16 линий). Поддерживается некоторыми современными моделями процессоров, список устройств, поддерживающих PCIe 4.0, расширяются с каждым годом.

Кроме PCIe существуют и альтернативные решения — в первую очередь это новый открытый стандарт OpenCAPI, и Gen-Z а также частные решения NVIDIA NVLink, IBM Infinity Fabric, Intel UltraPath и т.п.

Рассмотрим подробнее некоторые (наиболее часто используемые) альтернативные решения. Среди открытых стандартов следует отметить ниже перечисленные технологии.

Gen-Z [32] — открытая технология, которая позволяет получить пропускную способность в 32 ГБ/с на канал и вплоть до 400 ГБ/с на несколько каналов. Открытый стандарт опубликован в 2018-м году.

OpenCAPI [33] — также открытый стандарт, разработанный широким консорциумом. Процессор IBM POWER9 поддерживает OpenCAPI. OpenCAPI 3.0 поддерживает пропускную способность до 25 Гбит/с на один канал, и вплоть до восьми каналов.

Более широкий спектр вариантов представляют частные разработки компаний. К недостаткам можно отнести ограниченную совместимость с устройствами других производителей. Ниже перечислен ряд проприетарных технологий внутриузлового интерконнекта.

NVIDIA NVLink [34]. Изначально разработанный для интеграции NVIDIA GPU и предоставления общей памяти, сейчас может быть применен для более широкого спектра решений, в том числе для организации высокоскоростных каналов «CPU – GPU». В частности, в топовом суперкомпьютере по данным рейтинга июнь 2018 используется соединение NVlink между процессорами IBM POWER9 и ускорителями NVIDIA. Пропускная способность (двунаправленная) одного линка NVLink 2.0 — 50 ГБ/с, агрегированная пропускная способность шести линков (двунаправленная) – 300 ГБ/с.

Intel Ultra Path Interconnect (Intel® UPI) [35] — проприетарный канал для объединения двух процессоров Intel Xeon Scalable.

Infinity Fabric [36] — интерконнект от AMD, может быть использован для объединения процессоров AMD (семейства Zen) и графических ускорителей (например, Vega). В мультисокетной конфигурации с процессорами EPYC и оперативной памятью DDR4-2666 каждый линк может достигать производительности в 42,667 ГБ/с, общая двунаправленная пропускная способность – 170,667 ГБ/с.

Отечественные микропроцессоры «Эльбрус-8С» поддерживают по 3 дуплексных канала с двунаправленной производительностью 16 ГБ/с [37].

Cavium Coherent Processor Interconnect CCPI2 — соединяет 2 процессора Cavium с производительностью 600 Гбит/с.

Кроме того, есть ряд решений, которые являются расширениями PCIe — например CCIX [38] и IBM CAPI [39].

Внутриузловой интерконнект оказывает существенное влияние на производительность вычислительного узла. При прочих равных (пиковая производительность, энергопотребление, наличие ПО для выбранной архитектуры) — предпочтение следует отдавать тем технологиям, которые поддерживают более производительные каналы передачи данных. С точки зрения каналов передачи данных наиболее перспективным в настоящее время является процессор IBM POWER9, который поддерживает NVLink и OpenCAPI.

Также следует следить за развитием перспективных технологий, которые могут позволить увеличить пропускную способность на несколько порядков. Особенный интерес в данном контексте представляет передача информации при помощи полупроводниковых лазеров. Потенциально данная технология может быть использована как для быстрой передачи данных внутри узла, так и для сверх-быстрого межузлового интерконнекта. В качестве примера успешной реализации приведем новый суперкомпьютер «Жорес», разработанный учеными Сколковского института науки и технологий, в котором

для передачи информации между узлами используются оптоволоконные каналы и полупроводниковые лазеры, основанные на полупроводниковых гетероструктурах [40].

3. Инфраструктура узла современного суперкомпьютера

3.1. Охлаждение

С целью увеличения вычислительной плотности, для охлаждения современных суперкомпьютеров следует использовать жидкостное охлаждение, а наиболее перспективным направлением являются охлаждающие пластины (*coldplates*) — плотно прилегающие пластины с хладагентом. В настоящее время существует огромное количество компаний, которые занимаются жидкостным охлаждением. В списке ниже рассмотрены наиболее распространенные подходы [41] к организации жидкостного охлаждения. *Охлаждающие пластины (coldplates)* — использование охлаждаемых жидкостью пластин, которые полностью покрывают всю элементосодержащую поверхность вычислительного узла. Примеры компаний, использующих данную технологию: Aquila [42], Dell. В России лидером по разработке вычислительных кластеров, охлаждаемых колдплейтами является группа компаний РСК [43].

Индивидуальные теплообменники (бобышки) — специальные элементы, позволяющие подводить охлаждающую жидкость напрямую к индивидуальным компонентам вычислительного узла. Примеры компаний: Asetek [44], Ebullient [45].

Жидкостное охлаждение на уровне шкафа. Примеры компаний: CoolIT, Inspur, HPE's Apollo.

Погружные (иммерсионные) системы — разработки, подразумевающие полное погружение вычислительных узлов в специальную диэлектрическую жидкость. Примеры компаний: ExaScaler Inc [46], 3M'S HPC [47]. Представителями в России являются IMMERS [48], а также погружные системы охлаждения реконфигурируемых вычислительных систем [49].

Другие разработки. Например, компания Liquid MIPS [50] представляет интересное направление — геотермальный кулинг.

В настоящее время можно говорить, что жидкостное охлаждение является однозначно лидирующим методом для охлаждения суперкомпьютеров больших масштабов. По сравнению с воздушным охлаждением методы жидкостного охлаждения обеспечивают более высокую надежность и низкое энергопотребление.

3.2. Датчики и сенсоры

Датчики и сенсоры, отвечающие за мониторинг состояния узла становятся критически важным элементом инфраструктуры. Наличие датчиков не только позволяет отслеживать критичные сбои в режиме реального времени, но и поможет прогнозировать потенциальные отказы оборудования, а также использовать данные о температуре и энергопотреблении для «умной» энергоэффективной балансировки вычислительной нагрузки.

Установленные датчики и сенсоры должны обеспечивать [51]:

- высокую точность и надежность измерений;
- измерение состояния отдельных компонентов вычислительного узла — таких как процессоры, оперативная память, ускорители, а также интерконнект и система охлаждения;
- корректное измерение энергопотребления;

– высокую частоту снятия данных.

Наиболее распространенным способом отслеживания состояния вычислительного узла являются встроенные датчики, информацию с которых можно получить через Intelligent Platform Management Interface (IPMI), который опрашивает board management controller (BMC). Другим способом является использование средств мониторинга и программных моделей, предоставляемых производителями, таких как Intel RAPL или IBM Amester. Существует и ряд других решений, таких как использование внешних устройств для измерения напряжения.

Независимо от выбранного способа организации аппаратного уровня мониторинга, необходимо предусмотреть наличие удобного пользовательского интерфейса для мониторинга присутствующих датчиков. Крайне желательно наличие унифицированного API, который позволит в едином стиле получать информацию с системных датчиков и сенсоров [52].

При наличии соответствующей аппаратной и программной инфраструктуры, позволяющей отслеживать основные характеристики вычислительного кластера на уровне базовых вычислительных элементов возможно «умное» управление кластером, в том числе отслеживание потенциальных сбоях по изменению характеристик температуры и энергопотребления, использование адаптивных алгоритмов балансировки нагрузки [53]. Данная инфраструктура должна существенно повысить время функциональной работы вычислительной системы.

4. Отображение вычислительных задач на архитектуру

Некоторая часть суперкомпьютеров создается для специфических задач, но большая часть должна быть готова к задачам из разных прикладных областей. Определим несколько основных классов задач и предложим обобщенные варианты архитектур узлов для них.

Мощные вычислительные задачи. Для этого класса задач мы рекомендуем гомогенные узлы с универсальными вычислителями, а также при наличии ограничений по энергопотреблению — узлы на базе универсальных процессоров RISC или CISC и дополнительных ускорителей GPGPU.

Задачи обработки больших данных. Данный класс задач для эффективной работы требует гиперконвергентные узлы или высокопроизводительную систему хранения данных, построенную с использованием специальных технологий организации промежуточных буферов между системой хранения и вычислительными узлами.

Задачи машинного обучения и глубокого обучения. Для этого класса задач мы рекомендуем гиперконвергентные узлы на базе универсальных процессоров RISC или CISC и дополнительных ускорителей GPGPU. Особое внимание следует уделить характеристикам внутриузловому интерконнекта, доступного для интеграции выбранных моделей процессоров и ускорителей.

Задачи из некоторых специфических областей. Для этого класса задач мы рекомендуем узлы на базе универсальных процессоров RISC или CISC и дополнительных ускорителей FPGA, реализующих алгоритмы для данных задач.

Тем не менее, основным критерием для быстрого решения конкретных задач на суперкомпьютерах является не только наличие соответствующей аппаратной части, но и эффективного прикладного программного обеспечения, функционирующего на данной

аппаратуре. Таким образом, важнейшими для узла характеристиками, на основе которых должен производиться детальный выбор узла являются:

- наличие необходимого прикладного программного обеспечения для данных архитектур;
- теоретическая производительность узла;
- максимальное энергопотребление узла;
- стандартизация и модульность компонент узла (как с точки зрения замены вышедших из строя, так и обновленных узлов).

Заключение

В данной статье рассмотрены основные конструктивные элементы современных вычислительных узлов. Даны классификации и наиболее популярные примеры универсальных и специализированных ядер, проведён анализ тенденций на основе лидирующий суперкомпьютеров рейтинга ТОП-500.

Основные итоги проведённого обзора заключаются в следующем:

Выбор архитектуры универсальных ядер и ускорителей во многом зависит от задач, которые планируется решать на проектируемом суперкомпьютере. В работе определены основные классы задач и предложены наиболее перспективные варианты архитектур для них.

При организации ОЗУ наиболее популярным в настоящее время остается стандарт DDR-SDRAM. Альтернативными решениями являются иерархические решения с использованием дополнительного уровня ОЗУ на базе энергонезависимой памяти, а также использование 3D Stacked Memory, которая позволяет увеличить пропускную способность, но обладает лимитированным объемом.

На узле также должны присутствовать энергонезависимые устройства хранения, которые могут быть использованы для создания промежуточного буфера для работы с данными или для полной интеграции уровня хранения с уровнем обработки (гиперконвергентность) в зависимости от выбранного типа организации системы хранения данных.

В связи с возрастающими требованиями к вычислительной плотности, для охлаждения современных суперкомпьютеров следует использовать жидкостное охлаждение. В статье рассмотрены популярные подходы к организации жидкостного охлаждения — погружное, с использованием индивидуальных теплообменников и охлаждающих пластин (coldplates) и приведены примеры успешных решений на базе перечисленных подходов.

Наконец, современный суперкомпьютер должен быть оснащён системой мониторинга. Соответственно, на уровне вычислительного узла должны присутствовать датчики и сенсоры, обеспечивающие высокую точность и надежность измерений; высокую частоту снятия данные, а также корректное измерение состояния отдельных компонентов вычислительного узла.

В заключение обсудим некоторые технологические проблемы. Прежде всего, это исчерпание потенциала закона Мура (точнее бизнес-прогноза Мура), вызванное тем, что проектная норма технологических процессов подходит к физически допустимому пределу. Согласно докладу [54] для сохранения прогресса в компьютерной отрасли есть три основных направления:

- изобретение новых устройств;

- изобретение новых архитектур;
- новые парадигмы вычислений.

Некоторые актуальные разработки, относящиеся к первым двум направлениям, упомянуты в статье. Прежде всего, это узкоспециализированные ускорители (аналоговые, квантовые, тензорные и нейроморфные). Существенный прогресс в настоящее время наблюдается в использовании различных наборов инструкций (ISA) — кроме традиционного CISC доступны процессоры и соответствующий стек ПО для различных RISC и VLIW архитектур. Также следует отметить прогресс в области организации иерархии памяти, особенно 3D Stacked Memory.

В статье мы старались сделать акцент на тех технологиях, которые достигли стадии выхода в производство или, как минимум, создания рабочих прототипов. За пределами обзора оказались направления работ, которые требуют преодоления существенных технологических барьеров. Тем не менее, в контексте потенциально перспективных технологий нельзя не отметить идеи использования сверхпроводников для создания цифровых и квантовых компьютеров, идеи использования других материалов и структур для создания вычислителей (например, углеродных нанотрубок или графена). Если существующие на текущий момент технологические барьеры в любом из этих направлений будут преодолены, это может положить начало новой эпохе энергоэффективных вычислений.

Очень важно, чтобы основные усилия были направлены на разработку соответствующего программного стека, который будет поддерживать новые программные парадигмы и аппаратные технологии. Это — гибридные вычисления, параллелизм на миллионы и более потоков, использование нестандартных архитектур и т.п. Необходимо решить ряд противоречий, например между необходимостью глубокой аппаратно-зависимой оптимизации ПО и необходимостью портирования этого ПО на широкий диапазон разноархитектурных вычислителей. Еще одно противоречие заключается в необходимости разработки новой сверх-масштабируемой парадигмы параллельных вычислений, которая позволит эффективно использовать вычислительные суперкомпьютеры будущего, и необходимости портирования на эти же суперкомпьютеры существующих основных вычислительных пакетов.

Таким образом, в ближайшем будущем прогресс в области повышения производительности машинных вычислений будет во многом зависеть от эффективности интеграции новых технологий (ускорителей, внутриузлового интерконнекта, межузлового интерконнекта, новых наборов инструкций, иерархий памяти) на уровне стека программного обеспечения. Прогресс может быть неравномерным, зависеть от предметной области, алгоритмической специфики и соответствующих архитектурных требований и степени адаптации (или разработки с нуля) специализированных программных пакетов к современным суперкомпьютерам.

Литература

1. Проект Российской Академии Наук: «Создание вычислительной системы для моделирования суперкомпьютера с производительностью эксафлопсного уровня». Институт прикладной математики им. М.В. Келдыша Российской академии наук. URL: <http://www.keldysh.ru/projects/exaflops.pdf> (дата обращения: 23.01.2019).
2. Reed D.A., Dongarra J. Exascale Computing and Big Data // Communications of the ACM.

2015. Vol. 58, No. 7. P. 56-68. DOI: 10.1145/2699414.
3. Chrysos G. Intel® Xeon Phi coprocessor (codename Knights Corner) // Proceedings of the 2012 IEEE Hot Chips 24 Symposium, HCS, August 27–29, 2012, Cupertino, CA. P. 1–31. DOI: 10.1109/HOTCHIPS.2012.7476487.
 4. Lindholm E., Nickolls J., Oberman S., Montrym J. NVIDIA Tesla: A Unified Graphics and Computing Architecture // IEEE Micro. 2008. Vol. 28, No. 2. P. 39–55. DOI: 10.1109/MM.2008.31.
 5. Jouppi N., Young C., Patil N., Patterson D. Motivation for and Evaluation of the First Tensor Processing Unit // IEEE Micro. 2018. Vol. 38, No. 3. P. 10–19. DOI: 10.1109/MM.2018.032271057.
 6. Davies M. et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning // IEEE Micro. 2018. Vol. 38, No. 1. P. 82–99. DOI: 10.1109/MM.2018.112130359.
 7. Hsu J. CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy. IEEE Spectrum Tech Talks. 2018. URL: <https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy> (дата обращения: 23.11.2018).
 8. Intel® Stratix® 10 SoC FPGAs. URL: <https://www.intel.com/content/www/us/en/products/programmable/soc/stratix-10.html> (дата обращения: 23.11.2018).
 9. Exascale Requirements Review. An Office of Science review sponsored jointly by Advanced Scientific Computing Research and High Energy Physics. June 10–12, 2015, Bethesda, Maryland. URL: <http://hepcce.org/files/2016/11/DOE-ExascaleReport-HEP-Final.pdf> (дата обращения: 13.11.2018).
 10. Top500 List Statistics. Release November 2018. URL: <https://www.top500.org/statistics/list/> (дата обращения: 16.11.2018).
 11. Hemsoth N. Cascade Lake at Heart of 2019 TACC Supercomputer. Онлайн ресурс технологических новостей Next Platform поддерживаемый Stackhouse Publishing Inc в партнерстве с The Register. 2018. URL: <https://www.nextplatform.com/2018/08/29/cascade-lake-heart-of-2019-tacc-supercomputer/> (дата обращения: 13.11.2018).
 12. Bartsch V. D6.3 Initial Project Press Release // ExaNoDe Consortium Public deliverable. Пресс-релиз по проекту ExaNode. 2016. URL: <http://exanode.eu/wp-content/uploads/2017/04/D6.3.pdf> (дата обращения: 16.11.2018).
 13. ARMv8 — A Scalable Vector Extension for Post-K. FUJITSU LIMITED. 2016. URL: <http://www.fujitsu.com/global/Images/armv8-a-scalable-vector-extension-for-post-k.pdf> (дата обращения: 22.01.2019).
 14. Astra. Top500 The List. URL: <https://www.top500.org/system/179565> (дата обращения: 16.11.2018).
 15. Xilinx. High Performance Computing and Data Storage. URL: <https://www.xilinx.com/applications/high-performance-computing.html> (дата обращения: 23.11.2018).
 16. Timmel A.N., Daly J.T. Multiplication with Fourier Optics Simulating 16-bit Modular Multiplication. URL: <https://arxiv.org/pdf/1801.01121.pdf> (дата обращения: 23.11.2018).
 17. Ким А.К., Перекаатов В.И., Фельдман В.М. На пути к российской экзасистеме: планы разработчиков аппаратно-программной платформы «Эльбрус» по созданию

- суперкомпьютера экзафлопсной производительности // Вопросы радиоэлектроники. Вычислительные системы на базе многоядерных микропроцессоров. 2018. № 2. С. 6–13.
18. CORAL Collaboration: Briefing on CORAL-2 RFP and Draft Technical Requirements // Vendor Webinar Meeting. December 6, 2017 URL: <https://procurement.ornl.gov/rfp/CORAL2/Brief-of-Draft-SOW-20171206-SA.PDF> (дата обращения: 23.11.2018).
 19. Farber R. HPC and AI — Two Communities Same Future. HPCwire: Global News and Information on High Performance Computing. 2018. URL: <https://www.hpcwire.com/2018/01/25/hpc-ai-two-communities-future/> (дата обращения: 23.11.2018).
 20. JEDEC DDR5 & NVDIMM-P Standards Under Development. Global Standards for the Microelectronics Industry. 2017. URL: <https://www.jedec.org/news/pressreleases/jedec-ddr5-nvdimm-p-standards-under-development> (дата обращения: 23.11.2018).
 21. Hadidi R. et al. Demystifying the Characteristics of 3D-Stacked Memories: A Case Study for Hybrid Memory Cube // Proceedings of the IEEE International Symposium on Workload Characterization, IISWC 2017, October 1–3, 2017, Seattle, WA, USA, P. 66-75. DOI: 10.1109/IISWC.2017.8167757.
 22. High Bandwidth Memory (HBM) DRAM. JESD235A. Global Standards for the Microelectronics Industry. 2015. URL: <https://www.jedec.org/standards-documents/docs/jesd235a> (дата обращения: 23.11.2018).
 23. Hybrid Memory Cube (HMC). Hybrid Memory Cube Consortium Page. URL: <http://hybridmemorycube.org/> (дата обращения: 16.11.2018).
 24. Intel® Memory Drive Technology Application Note. URL: <https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/intel-mem-drive-tech-appnote.pdf> (дата обращения: 23.11.2018).
 25. Graphics Double Data Rate (GDDR5) SGRAM standard. JESD212C. Global Standards for the Microelectronics Industry. 2016. URL: <https://www.jedec.org/standards-documents/docs/jesd212c> (дата обращения: 23.11.2018).
 26. Graphics Double Data Rate 6 (GDDR6) SGRAM standard. JESD250A. Global Standards for the Microelectronics Industry. 2017. URL: <https://www.jedec.org/standards-documents/docs/jesd250a> (дата обращения: 23.11.2018).
 27. Ferreira da Silva R., Callaghan S., Deelman E. On the use of burst buffers for accelerating data-intensive scientific workflows // Proceedings of the 12th Workshop on Workflows in Support of Large-Scale Science, WORKS '17. ACM, 2017. P. 2:1–2:9. DOI: 10.1145/3150994.3151000.
 28. Bhimji W., Bard D., Romanus M., Paul, D., Ovsyannikov A., Friesen B., et al. Accelerating Science with the NERSC Burst Buffer Early User Program. Lawrence Berkeley National Laboratory. 2016. URL: <https://escholarship.org/uc/item/9wv6k14t> (дата обращения: 23.11.2018).
 29. Cray® DataWarp™ Applications I/O Accelerator. URL: <https://www.cray.com/products/storage/datawarp> (дата обращения: 23.11.2018).
 30. Morgan T.P. For many hyperconverged is the next platform. Онлайн ресурс технологических новостей Next Platform поддерживаемый Stackhouse Publishing Inc в

- партнерстве с The Register. 2018. URL: <https://www.nextplatform.com/2018/01/29/hyperconverged-next-platform-many-jobs/> (дата обращения: 23.11.2018).
31. Бахур В. РСК представила гиперконвергентное HPC-решение на новейших компонентах. Интернет-издание о высоких технологиях cnews. 2018. URL: http://www.cnews.ru/news/line/2018-06-27_rsk_predstavila_giperkonvergentnoe_hpcreshenie (дата обращения: 23.11.2018).
 32. The GEN-Z Consortium. URL: <https://genzconsortium.org/> (дата обращения: 23.11.2018).
 33. The OpenCAPI Consortium. URL: <https://opencapi.org/> (дата обращения: 23.11.2018).
 34. NVLink Fabric. URL: <https://www.nvidia.com/ru-ru/data-center/nvlink/> (дата обращения: 23.11.2018).
 35. Краткое описание продукции: платформа масштабируемых процессоров Intel® Xeon®. URL: <https://www.intel.ru/content/www/ru/ru/processors/xeon/scalable/xeon-scalable-platform-brief.html> (дата обращения: 23.11.2018).
 36. Infinity Fabric (IF) — AMD. URL: https://en.wikichip.org/wiki/amd/infinity_fabric (дата обращения: 23.11.2018).
 37. Российские микропроцессоры серии «Эльбрус» и МЦСТ R и системные платы на их основе. Каталог продукции 2017. URL: http://mcst.ru/files/59db45/cf0cd8/50a21b/000000/katalog_produktsii_mtsst_hq.pdf (дата обращения: 23.11.2018).
 38. CCIX. URL: <https://www.ccixconsortium.com/> (дата обращения: 23.11.2018).
 39. Coherent Accelerator Processor Interface (CAPI). URL: <https://developer.ibm.com/linuxonpower/capi/> (дата обращения: 23.11.2018).
 40. Шустиков В. Ученые Сколтеха создали суперкомпьютер «Жорес». Фонд «Сколково», пресс-релизы. 2019. URL: <https://sk.ru/news/b/pressreleases/archive/2019/01/18/uchenye-skolteha-sozdali-superkompyuter-zhores.aspx> (дата обращения: 25.01.2019).
 41. Is Liquid Cooling Ready to Go Mainstream? HPCwire: Global News and Information on High Performance Computing. URL: <https://www.hpcwire.com/2017/02/13/liquid-cooling-ready-go-mainstream/> (дата обращения: 16.11.2018).
 42. Aquila. URL: <https://www.aquilagroup.com/cooling/> (дата обращения: 26.11.2018).
 43. Группа компаний РСК. URL: <http://www.rscgroup.ru/ru> (дата обращения: 25.01.2019).
 44. Asetek. URL: <https://www.asetek.com/> (дата обращения: 26.11.2018).
 45. Ebullient. URL: <http://ebullientcooling.com/> (дата обращения: 26.11.2018).
 46. ExaScaler Inc. Overview. URL: <http://www.exascalr.co.jp/en/company> (дата обращения: 26.11.2018).
 47. 3M Server Solutions for Data Centers. URL: https://www.3m.com/3M/en_US/data-center-us/solutions/data-center-servers/ (дата обращения: 26.11.2018).
 48. Абрамов С.М., Амелькин С.А., Романенко А.Ю., Симонов А.С., Чичковский А.А. Опыт реализации высокопроизводительных вычислительных систем с погружной жидкостной системой охлаждения // Труды 3-й Всероссийской научно-технической конференции

- «Суперкомпьютерные технологии» (СКТ-2014) (Дивноморское, Геленджик, 29 сентября – 4 октября 2014 г.). С. 9–15.
49. Левин И.И., Дордопуло А.И., Доронченко Ю.И., Раскладкин М.К., Федоров А.М. Погружная система охлаждения реконфигурируемых вычислительных систем на основе ПЛИС // Программные системы: теория и приложения. 2016. №4 (31). URL: <https://cyberleninka.ru/article/n/pogruzhnaya-sistema-ohlazhdeniya-rekonfiguriruemyh-vychislitelnyh-sistem-na-osnove-plis> (дата обращения: 28.11.2018).
50. Liquid MIPS. URL: <http://www.liquidmips.com/cms/en-us/howitworks.aspx> (дата обращения: 26.11.2018).
51. Libri A., Bartolini A., Benini L. Dwarf in a Giant: Enabling Scalable, High-Resolution HPC Energy Monitoring for Real-Time Profiling and Analytics. URL: <https://arxiv.org/pdf/1806.02698.pdf> (дата обращения: 19.11.2018).
52. Grant R.E., Levenhagen M., Olivier S.L., DeBonis D., Pedretti K.T., Laros III J.H. Standardizing Power Monitoring and Control at Exascale // Computer. 2016. Vol. 49, No. 10. P. 38–46. DOI: 10.1109/MC.2016.308.
53. Georgiou Y., Glesser D., Trystram D. Adaptive Resource and Job Management for Limited Power Consumption // IEEE International Parallel and Distributed Processing Symposium Workshop, 2015, Hyderabad. P. 863–870. DOI: 10.1109/IPDPSW.2015.118.
54. Shalf J.M., Leland R. Computing beyond Moore's Law // Computer. 2015. Vol. 48, No. 12. P. 14–23. DOI: 10.1109/mc.2015.37.

Тютляева Екатерина Олеговна, инженер НИР ЗАО «РСК Технологии», Переславль-Залесский, Россия.

Одинцов Игорь Олегович, руководитель отдела НИР, ООО «РСК Лабс», Санкт-Петербург, Россия.

Московский Александр Александрович, к.х.н., генеральный директор ЗАО «РСК Технологии», Москва, Россия.

Мармузов Глеб Владимирович, к.х.н., директор по техническому маркетингу, ЗАО «РСК Технологии», Москва, Россия.

DEVELOPMENT TRENDS OF MODERN SUPERCOMPUTERS

© 2019 E.O. Tyutlyayeva¹, I.O. Odintsov², A.A. Moskovsky¹, G.V. Marmuzov¹

¹ZAO RSC Technologies

(121170, Moscow, Kutuzovskiy av., 36, building 23),

²OOO RSC Labs

(121205, Moscow, Bolshoi Bulvar str., 42, building 1)

E-mail: xgl@rsc-tech.ru, igor_odintsov@rsc-tech.ru, moskov@rsc-tech.ru,

gleb.marmuzov@rsc-tech.ru

Received: 28.01.2019

This work includes analysis of the computation nodes of modern supercomputers from two perspectives; first the hardware components focus and secondly discuss the infrastructure. Identified trends leads to basic options of computation node design. The paper classifies the modern architectures of universal processing cores and specialized hardware accelerators cores; studies the recent trends in memory hierarchy design and intra-node interconnect; the paper includes ways of using the non-volatile memory in modern memory hierarchy. Furthermore, the paper analyses the recent trends in HPC infrastructure, in particular in modern liquid cooling approaches and monitoring. The basic variants of HPC computing nodes design are based on energy efficient universal processor and set of energy-efficient specialized hardware accelerators cores, according to the observed trends. The paper focuses on recent technologies that are currently at various stages of production or at the functional prototype stage. The study also discusses state-of-the-art computational challenges and algorithms-to-architecture mapping issues. Lastly, the paper discusses the current technological problems and main areas to maintain the progress in HPC area.

Keywords: high-performance compute node, architecture development of a high-performance compute node, analysis of supercomputer architectures, development trends of supercomputers.

FOR CITATION

Tyutlyayeva E.O., Odintsov I.O., Moskovsky A.A., Marmuzov G.V. Development Trends of Modern Supercomputers. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. vol. 8, no. 3. pp. 92–114. (in Russian) DOI: 10.14529/cmse190305.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. The project of the Russian Academy of Science: «Development of the compute system for simulation of the exascale supercomputer». Available at: <http://www.keldysh.ru/projects/exaflops.pdf> (accessed: 23.01.2019).
2. Reed D.A., Dongarra J. Exascale Computing and Big Data. *Communications of the ACM*. 2015. vol. 58, no. 7. pp. 56-68. DOI: 10.1145/2699414.
3. Chrysos G. Intel® Xeon Phi coprocessor (codename Knights Corner). Proceedings of the 2012 IEEE Hot Chips 24 Symposium, HCS, August 27–29, 2012, Cupertino, CA. pp. 1–31. DOI: 10.1109/HOTCHIPS.2012.7476487.

4. Lindholm E., Nickolls J., Oberman S., Montrym J. NVIDIA Tesla: A Unified Graphics and Computing Architecture. *IEEE Micro*. 2008. vol. 28, no. 2. pp. 39–55. DOI: 10.1109/MM.2008.31.
5. Jouppi N., Young C., Patil N., Patterson D. Motivation for and Evaluation of the First Tensor Processing Unit. *IEEE Micro*. 2018. vol. 38, no. 3. pp. 10–19. DOI: 10.1109/MM.2018.032271057.
6. Davies M. et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*. 2018. vol. 38, no. 1. pp. 82–99. DOI: 10.1109/MM.2018.112130359.
7. Hsu J. CES 2018: Intel’s 49-Qubit Chip Shoots for Quantum Supremacy. IEEE Spectrum Tech Talks. 2018. Available at: <https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy> (accessed: 23.11.2018).
8. Intel® Stratix® 10 SoC FPGAs. Available at: <https://www.intel.com/content/www/us/en/products/programmable/soc/stratix-10.html> (accessed: 23.11.2018).
9. Exascale Requirements Review. An Office of Science review sponsored jointly by Advanced Scientific Computing Research and High Energy Physics. June 10–12. 2015 BETHESDA, MARYLAND. Available at: <http://hepcce.org/files/2016/11/DOE-ExascaleReport-HEP-Final.pdf> (accessed: 13.11.2018).
10. Top500 List Statistics. Release November 2018. Available at: <https://www.top500.org/statistics/list/> (accessed: 16.11.2018).
11. Hemsoth N. Cascade Lake at Heart of 2019 TACC Supercomputer. Technology publication resource The Next Platform is published by Stackhouse Publishing Inc in partnership with The Register. Available at: <https://www.nextplatform.com/2018/08/29/cascade-lake-heart-of-2019-tacc-supercomputer/> (accessed: 13.11.2018).
12. Bartsch V. D6.3 Initial Project Press Release. ExaNoDe Consortium Public deliverable. 2016. Available at: <http://exanode.eu/wp-content/uploads/2017/04/D6.3.pdf> (accessed: 16.11.2018).
13. ARMv8 — A Scalable Vector Extension for Post-K. FUJITSU LIMITED. 2016 Available at: <http://www.fujitsu.com/global/Images/armv8-a-scalable-vector-extension-for-post-k.pdf> (accessed: 22.01.2019).
14. Astra. Top500 The List. Available at: <https://www.top500.org/system/179565> (accessed: 16.11.2018).
15. Xilinx. High Performance Computing and Data Storage. Available at: <https://www.xilinx.com/applications/high-performance-computing.html> (accessed: 23.11.2018).
16. Timmel A.N., Daly J.T. Multiplication with Fourier Optics Simulating 16-bit Modular Multiplication. Available at: <https://arxiv.org/pdf/1801.01121.pdf> (accessed: 23.11.2018).
17. Kim A.K., Perekatov V.I., Feldman V.M. On the way to russian exasistemes: plans of the Elbrus hardware-software platform developers on creation of an exaflops performance supercomputer. *Voprosy radioelektroniki*, 2018, no. 2, pp. 6–13. (in Russian)
18. CORAL Collaboration: Briefing on CORAL-2 RFP and Draft Technical Requirements. Vendor Webinar Meeting. 2017. Available at: <https://procurement.ornl.gov/rfp/CORAL2/Brief-of-Draft-SOW-20171206-SA.PDF> (accessed: 23.11.2018).

19. Farber R. HPC and AI — Two Communities Same Future. HPCwire: Global News and Information on High Performance Computing. 2018. Available at: <https://www.hpcwire.com/2018/01/25/hpc-ai-two-communities-future/> (accessed: 23.11.2018).
20. JEDEC DDR5 & NVDIMM-P Standards Under Development. Global Standards for the Microelectronics Industry. 2017. Available at: <https://www.jedec.org/news/pressreleases/jedec-ddr5-nvdimm-p-standards-under-development> (accessed: 23.11.2018).
21. Hadidi R. et al. Demystifying the Characteristics of 3D-Stacked Memories: A Case Study for Hybrid Memory Cube. Proceedings of the IEEE International Symposium on Workload Characterization, IISWC 2017, October 1–3, 2017, Seattle, WA, USA. pp. 66-75. DOI: 10.1109/IISWC.2017.8167757.
22. High Bandwidth Memory (HBM) DRAM. JESD235A. Global Standards for the Microelectronics Industry. 2015. Available at: <https://www.jedec.org/standards-documents/docs/jesd235a> (accessed: 23.11.2018).
23. Hybrid Memory Cube (HMC). Hybrid Memory Cube Consortium Page. Available at: <http://hybridmemorycube.org/> (accessed: 16.11.2018).
24. Intel® Memory Drive Technology Application Note. Available at: <https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/intel-mem-drive-tech-appnote.pdf> (accessed: 23.11.2018).
25. Graphics Double Data Rate (GDDR5) SGRAM standard. JESD212C. Global Standards for the Microelectronics Industry. 2016. Available at: <https://www.jedec.org/standards-documents/docs/jesd212c> (accessed: 23.11.2018).
26. Graphics Double Data Rate 6 (GDDR6) SGRAM standard. JESD250A. Global Standards for the Microelectronics Industry. 2017. Available at: <https://www.jedec.org/standards-documents/docs/jesd250a> (accessed: 23.11.2018).
27. Ferreira da Silva R., Callaghan S., Deelman E. On the use of burst buffers for accelerating data-intensive scientific workflows. Proceedings of the 12th Workshop on Workflows in Support of Large-Scale Science, WORKS '17. ACM, 2017. pp. 2:1–2:9. DOI: 10.1145/3150994.3151000.
28. Bhimji W., Bard D., Romanus M., Paul, D., Ovsyannikov A., Friesen B., et al. Accelerating Science with the NERSC Burst Buffer Early User Program. Lawrence Berkeley National Laboratory. 2016. Available at: <https://escholarship.org/uc/item/9wv6k14t> (accessed: 23.11.2018).
29. Cray® DataWarp™ Applications I/O Accelerator. Available at: <https://www.cray.com/products/storage/datawarp> (accessed: 23.11.2018).
30. Morgan T.P. For many hyperconverged is the next platform. Technology publication resource The Next Platform is published by Stackhouse Publishing Inc in partnership with The Register. 2018. Available at: <https://www.nextplatform.com/2018/01/29/hyperconverged-next-platform-many-jobs/> (accessed: 23.11.2018).
31. Bahur V. The RSC Technologies Company introduces hyperconverged HPC-solution based on state-of-the-are components. 2018. Available at: http://www.cnews.ru/news/line/2018-06-27_rsk_predstavila_giperkonvergentnoe_hpcreshenie (accessed: 23.11.2018). (in Russian)

32. The GEN-Z Consortium. Available at: <https://genzconsortium.org/> (accessed: 23.11.2018).
33. The OpenCAPI Consortium. Available at: <https://opencapi.org/> (accessed: 23.11.2018).
34. NVLink Fabric. Available at: <https://www.nvidia.com/ru-ru/data-center/nvlink/> (accessed: 23.11.2018).
35. Product Brief: Intel® Xeon® Scalable Platform. Available at: <https://www.intel.sg/content/www/xa/en/processors/xeon/scalable/xeon-scalable-platform-brief.html> (accessed: 23.11.2018).
36. Infinity Fabric (IF) — AMD. Available at: https://en.wikichip.org/wiki/amd/infinity_fabric (accessed: 23.11.2018).
37. Russian microprocessors «Elbrus» and «MCST R» series and boards based thereon. Production catalogue 2017. Available at: http://mcst.ru/files/59db45/cf0cd8/50a21b/000000/katalog_produktsii_mtsst_hq.pdf (accessed: 23.11.2018). (in Russian)
38. CCIX. Available at: <https://www.ccixconsortium.com/> (accessed: 23.11.2018).
39. Coherent Accelerator Processor Interface (CAPI). Available at: <https://developer.ibm.com/linuxonpower/capi/> (accessed: 23.11.2018).
40. Shustikov V. Skoltex researches developed supercomputer «Zhores». The Skolkovo Foundation Press-Release. 2019. Available at: <https://sk.ru/news/b/pressreleases/archive/2019/01/18/uchenye-skolteha-sozdali-superkompyuter-zhores.aspx> (accessed: 25.01.2019). (in Russian).
41. Is Liquid Cooling Ready to Go Mainstream? HPCwire: Global News and Information on High Performance Computing. Available at: <https://www.hpcwire.com/2017/02/13/liquid-cooling-ready-go-mainstream/> (accessed: 16.11.2018).
42. Aquila. Available at: <https://www.aquilagroup.com/cooling/> (accessed: 26.11.2018).
43. RSC Group. Available at: <http://www.rscgroup.ru/en> (accessed: 25.01.2019).
44. Asetek. Available at: <https://www.asetek.com/> (accessed: 26.11.2018).
45. Ebullient. Available at: <http://ebullientcooling.com/> (accessed: 26.11.2018).
46. ExaScaler Inc. Overview. Available at: <http://www.exascalr.co.jp/en/company> (accessed: 26.11.2018).
47. 3M Server Solutions for Data Centers. Available at: https://www.3m.com/3M/en_US/data-center-us/solutions/data-center-servers/ (accessed: 26.11.2018).
48. Abramov S.M., Amelkin S.A., Romanenko A.Y., Simonov A.S., Chichkovsky A.A. The experience of implementing the high-performance computing systems with immersion cooling. Proceedings of the All-Russian Scientific and Technical Conference «Supercomputer Technologies» (Divnomorskoe, Russia, September, 29 – October, 4, 2014). pp. 9–15. (in Russian).
49. Levin I., Dordopulo A., Doronchenko Y., Raskladkin M., Fedorov A. Immersion cooling system for FPGA-based reconfigurable computer systems. *Program systems: theory and applications*. 2016. vol. 7:4(31), pp. 65–81. Available at: http://psta.pstiras.ru/read/psta2016_4_65-81.pdf (accessed: 24.01.2019). (in Russian).

50. Liquid MIPS. Available at: <http://www.liquidmips.com/cms/en-us/howitworks.aspx> (accessed: 26.11.2018).
51. Libri A., Bartolini A., Benini L. Dwarf in a Giant: Enabling Scalable, High-Resolution HPC Energy Monitoring for Real-Time Profiling and Analytics. Available at: <https://arxiv.org/pdf/1806.02698.pdf> (accessed: 19.11.2018).
52. Grant R.E., Levenhagen M., Olivier S.L., DeBonis D., Pedretti K.T., Laros III J.H. Standardizing Power Monitoring and Control at Exascale. *Computer*, 2016. vol. 49, no. 10. pp. 38–46. DOI: 10.1109/MC.2016.308.
53. Georgiou Y., Glesser D., Trystram D. Adaptive Resource and Job Management for Limited Power Consumption. IEEE International Parallel and Distributed Processing Symposium Workshop, 2015, Hyderabad. pp. 863–870. DOI: 10.1109/IPDPSW.2015.118.
54. Shalf J.M., Leland R. Computing beyond Moore’s Law. *Computer*. 2015. vol. 48, no. 12. pp. 14–23. DOI: 10.1109/mc.2015.37.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки), 05.13.17 – Теоретические основы информатики (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции, издателя: 454080, г. Челябинск, проспект Ленина, 76, Издательский центр ЮУрГУ, каб. 32.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии <http://vestnikvmi.susu.ru>. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.**
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:
Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, кафедра СП,
ответственному секретарю Цымблеру М.Л.
3. Адрес электронной почты редакции: vestnikvmi@susu.ru
4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**