

ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

2017
Т. 6, № 4

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
по следующим группам специальностей:
05.13.00 – информатика, вычислительная техника и управление;
25.00.00 – науки о Земле

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
- Математическое и программное обеспечение высокопроизводительных вычислительных систем
- 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

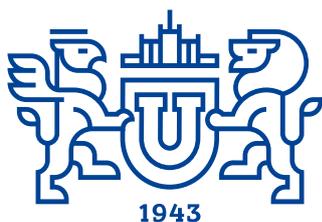
Редакционная коллегия

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.
А.С. Порозов, *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 2017
vol. 6, no. 4

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
- High performance computing
- 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)
A.S. Porozov, 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

Вычислительная математика

МЕТОД ОБРАБОТКИ ИНФОРМАЦИИ О НЕМЕТАЛЛИЧЕСКИХ ВКЛЮЧЕНИЯХ, ПОЛУЧАЕМОЙ ПРИ ОБСЛЕДОВАНИИ МИКРОШЛИФА ГОТОВОЙ СТАЛИ А.Д. Дрозин	5
---	---

МЕТОД ОБРАБОТКИ ИНФОРМАЦИИ, ПОЛУЧАЕМОЙ ПРИ ОБСЛЕДОВАНИИ МИКРОШЛИФА ГОТОВОЙ СТАЛИ В СЛУЧАЕ НЕРЕПРЕЗЕНТАТИВНЫХ ДАННЫХ А.Д. Дрозин, Н.М. Япарова	20
---	----

Информатика, вычислительная техника и управление

МЕТОДЫ И СРЕДСТВА ОРГАНИЗАЦИИ ГЛОБАЛЬНОЙ ОЧЕРЕДИ ЗАДАНИЙ В ТЕРРИТОРИАЛЬНО РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ А.В. Баранов, А.И. Тихомиров	28
--	----

ОБОБЩЕННАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ МОДУЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ПРОВЕРКИ ДОПУСТИМОСТИ КОНФИГУРАЦИЙ ТАКИХ СИСТЕМ А.Б. Глонина	43
--	----

Суперкомпьютерное моделирование

РАЗНОМАСШТАБНЫЕ ЗАДАЧИ ТЕПЛОМАССОБМЕНА В АТОМНОЙ ЭНЕРГЕТИКЕ В.Ю. Волков, Л.А. Голибродо, А.А. Крутиков, О.В. Кудрявцев, Ю.Н. Надинский, А.Т. Нечаев, А.П. Скибин	60
---	----

ГИБРИДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ КЛАСТЕРЫ ДЛЯ ИЗУЧЕНИЯ СТРУКТУРЫ, ФУНКЦИИ И РЕГУЛЯЦИИ БЕЛКОВ Д.А. Суцлатов, Н.Н. Попова, К.Е. Копылов, М.В. Шегай, Вл.В. Воеводин, В.К. Швядас	74
--	----

РАЗРАБОТКА КОМПЬЮТЕРНЫХ МОДЕЛЕЙ БАЛЛИСТИЧЕСКИХ ТКАНЕЙ С ПОВЕРХНОСТНОЙ ОБРАБОТКОЙ Н.Ю. Долганина, А.В. Игнатова	91
--	----

Contents

Computational Mathematics

THE METHOD OF PROCESSING INFORMATION ABOUT NONMETALLIC INCLUSIONS OBTAINED FROM THE INVESTIGATION OF THE SURFACE OF FINISHED STEEL SAMPLES A.D. Drozin	5
THE METHOD OF PROCESSING INFORMATION OBTAINED FROM THE INVESTIGATION OF THE STEEL SAMPLE SURFACE IN THE CASE OF UNREPRESENTATIVE DATA A.D. Drozin, N.M. Yaparova	20

Informatics, computers and control

METHODS AND TOOLS FOR ORGANIZING THE GLOBAL JOB QUEUE IN THE GEOGRAPHICALLY DISTRIBUTED COMPUTING SYSTEM A.V. Baranov, A.I. Tikhomirov	28
GENERAL MODEL OF REAL-TIME MODULAR COMPUTER SYSTEMS OPERATION FOR CHECKING ACCEPTABILITY OF SUCH SYSTEMS CONFIGURATIONS A.B. Glonina	43

Supercomputer modeling

MULTI-SCALE PROBLEMS OF HEAT AND MASS TRANSFER IN NUCLEAR INDUSTRY V.Yu. Volkov, L.A. Golibrodo, A.A. Krutikov, O.V. Kudryavtsev, Yu.N. Nadinsky, A.T. Nechaev, A.P. Skibin	60
HYBRID COMPUTING CLUSTERS TO STUDY PROTEIN STRUCTURE, FUNCTION AND REGULATION D.A. Suplatov, N.N. Popova, K.E. Kopylov, M.V. Shegay, Vl.V. Voevodin, V.K. Svedas	74
THE DEVELOPMENT OF COMPUTER MODELS OF BALLISTIC FABRIC WITH SURFACE TREATMENT N.Yu. Dolganina, A.V. Ignatova	91



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.

МЕТОД ОБРАБОТКИ ИНФОРМАЦИИ О НЕМЕТАЛЛИЧЕСКИХ ВКЛЮЧЕНИЯХ, ПОЛУЧАЕМОЙ ПРИ ОБСЛЕДОВАНИИ МИКРОШЛИФА ГОТОВОЙ СТАЛИ

© 2017 А.Д. Дрозин

Южно-Уральский государственный университет

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

E-mail: drozinad@susu.ru

Поступила в редакцию: 23.10.2017

Неметаллические включения, возникающие в процессе производства стали — это микрочастицы продуктов химических реакций и посторонних примесей, имеющие размеры 0,1–100 мкм. Они являются концентраторами напряжений в готовых изделиях и ухудшают механические свойства стали. Для их исследования из образца исследуемого металла готовят микрошлиф и наблюдают в микроскоп срезы включений плоскостью микрошлифа. Однако на плоскости шлифа исследователь видит не сами включения, а лишь их срезы. Между тем включения разных размеров могут давать срезы одинакового размера и, наоборот, включения одного размера могут давать срезы различных размеров. В данной работе рассматривается метод определения количеств и размеров неметаллических включений на основе этой информации, то есть воссоздание объемной картины распределения включений по размерам на основе плоской картины распределения их срезов плоскостью микрошлифа. Метод основан на теории вероятностей и математической статистике. Показано, что число срезов включений, попадающих в каждый размерный интервал, подчиняется нормальному закону. Разработана методика определения функции распределения включений по размерам для включений сферической формы и разработан соответствующий алгоритм.

Ключевые слова: обработка информации, кристаллография, неметаллические включения, стереология.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Дрозин А.Д. Метод обработки информации о неметаллических включениях, получаемой при обследовании микрошлифа готовой стали // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 5–19. DOI: 10.14529/cmse170401.

Введение

В процессе выплавки сталь загрязняется продуктами протекающих в ней химических реакций и внешними примесями. Эти загрязнения, представляющие собой микрочастицы размером от 0,1 до 100 мкм, являются неметаллическими включениями в стали. Они во многом определяют качество готовой стали [1–4].

Для определения загрязненности стали неметаллическими включениями обычно поступают следующим образом [5]. Из исследуемого затвердевшего металла изготавливают образцы, обычно в форме прямоугольного параллелепипеда. Каждый образец разрезают, полируют срез и рассматривают через микроскоп (рис. 1). Исследователь видит в микроскоп срезы неметаллических включений плоскостью разреза.

Далее картину на разрезе сравнивают с эталонными картами и устанавливают балл загрязненности стали неметаллическими включениями.

Ясно, что такой стандартный метод использует очень мало информации, предоставляемой исследователю на поверхности шлифа. Также и обработка информации (сравнение картины на шлифе с эталонными картами) дает мало информации о загрязненности стали.

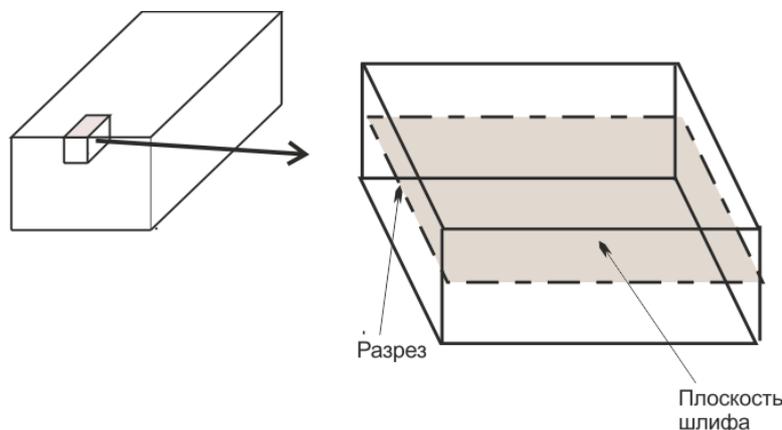


Рис. 1. Приготовление образца

В настоящей работе предпринята попытка провести обработку информации, позволяющую получить количественные данные: сколько включений каждого размера присутствуют в стали. Основная трудность заключается в том, что на плоскости шлифа исследователь видит не сами включения, а лишь их срезы (рис. 2).

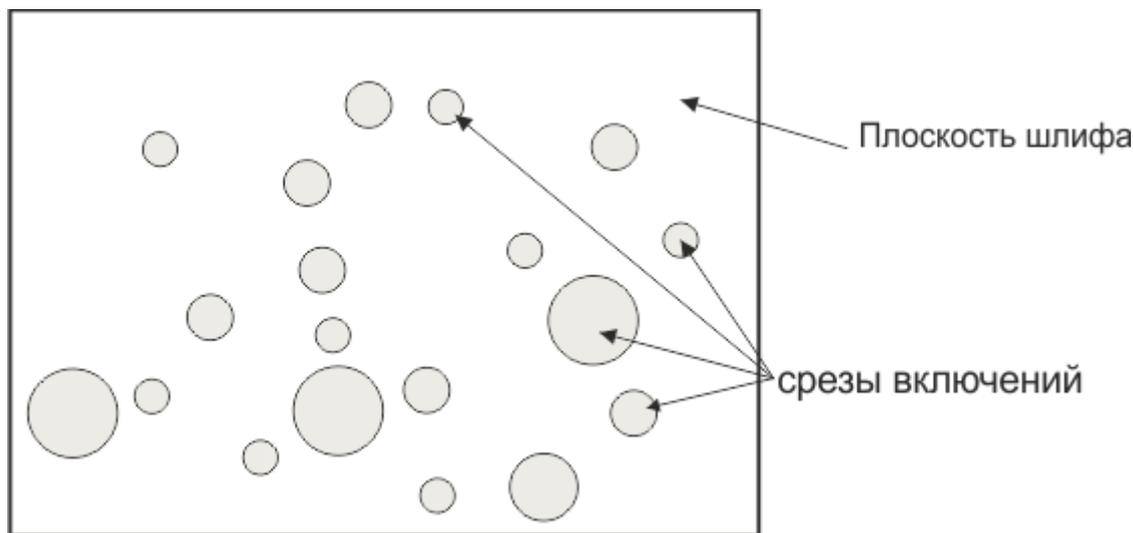


Рис. 2. Плоскость шлифа

Как показано на рис. 3, включения разных размеров могут дать срезы одинакового размера и, наоборот, включения одного размера могут давать разные срезы.

Разработан ряд методов определения распределения частиц по размерам (не только включений) из распределения по размерам их сечений. Подробные обзоры этих методов представлены в [6–12].

Нами был предложен новый метод определения распределения частиц по размерам более пригодный для металлографических исследований, основные идеи которого изложены в [13].

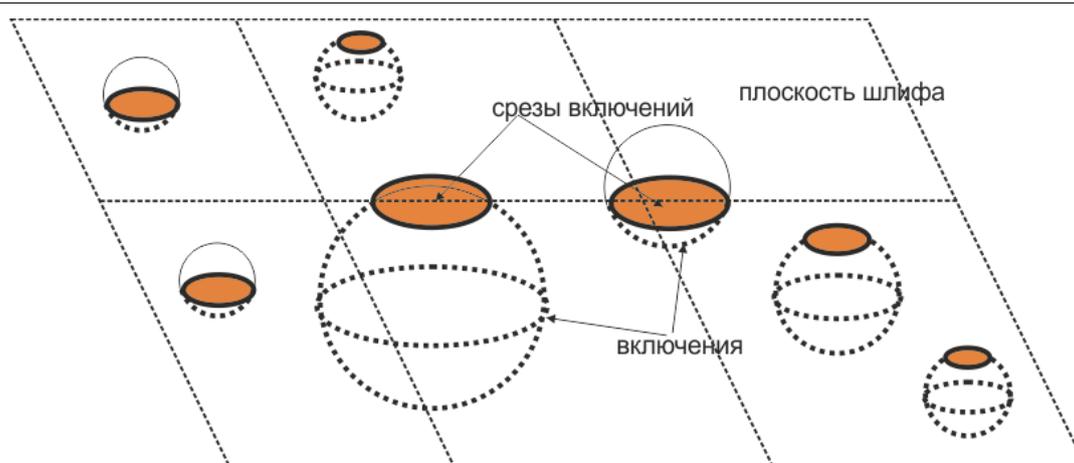


Рис. 3. Образование срезов включений

Статья организована следующим образом. В разделе 1 математически точно сформулирована цель исследования. В разделе 2 проведено математическое исследование зависимости числа срезов включений данного размера, как случайной функции распределения по размерам включений, дающих эти срезы. Показано, что число срезов включений, попадающих в данный размерный интервал, подчиняется нормальному закону. В разделе 3 проведены основные упрощения, позволяющие решить поставленную задачу. Получены необходимые соотношения. В разделе 4 приведен алгоритм вычислений. В разделе 5 приведены исходная таблица распределения срезов включений и таблица распределения включений, рассчитываемая по приведенному алгоритму. В заключении обсуждаются полученные результаты.

1. Постановка задачи

В объеме металла, представляющего собой прямоугольный параллелепипед высотой H и площадью основания S находятся неметаллические включения, представляющие собой шары разных радиусов. Пусть $N(R)$ такая (неизвестная) функция, что $N(R) V \Delta R$ — число находящихся в данном объеме $V=SH$ шаров с радиусами из интервала

$$\left[R - \frac{\Delta R}{2}; R + \frac{\Delta R}{2} \right] \text{ при бесконечно малом } \Delta R.$$

Для определения этой функции металл разрезают на некотором расстоянии a от основания и определяют распределение по размерам срезов попавших в сечение шаров следующим образом.

Возможные размеры радиусов срезов включений делят на M интервалов: $(0, r_1], (r_1, r_2], (r_2, r_3], \dots, (r_{M-1}, r_M]$ и подсчитывают числа срезов, попавших в каждый размерный интервал. Для повышения точности, шлиф несколько раз перетачивают и снова определяют количества срезов включений, попавших в каждый размерный интервал. Площади шлифов, полученных из одного образца, могут быть разными. Пусть было рассмотрено L шлифов с площадями S_1, \dots, S_L соответственно и получены количества $y_{l,i}$ ($l=1, \dots, L; i=1, \dots, M$) срезов, попадающих в i -ый размерный интервал $(r_{i-1}, r_i]$ при обследовании шлифа с номером l с площадью S_l .

Требуется определить функцию $N(R)$.

2. Теория метода

Определим количество срезов включений, попадающих в i -ый размерный интервал срезов $(r_{i-1}, r_i]$. Для функции $N(R)$, проведем построение, аналогичное построению определенного интеграла. Пусть R^* — радиус наибольшего из включений. Введем ось OR радиусов включений и разобьем отрезок $[0, R^*]$ произвольным образом на j^* частей точками $\rho_1, \dots, \rho_{j^*}$, но так, чтобы точки r_{i-1}, r_i вошли в число этих точек, как показано на рис. 4. Внутри каждого интервала (ρ_{j-1}, ρ_j) выберем «представитель» — размер $R = R_j$. К интервалу $(\rho_{j-1}, \rho_j]$ относятся $n_j = N(R_j)SH\Delta\rho_j$ шаров радиуса примерно R_j . Здесь $\Delta\rho_j = \rho_j - \rho_{j-1}$.

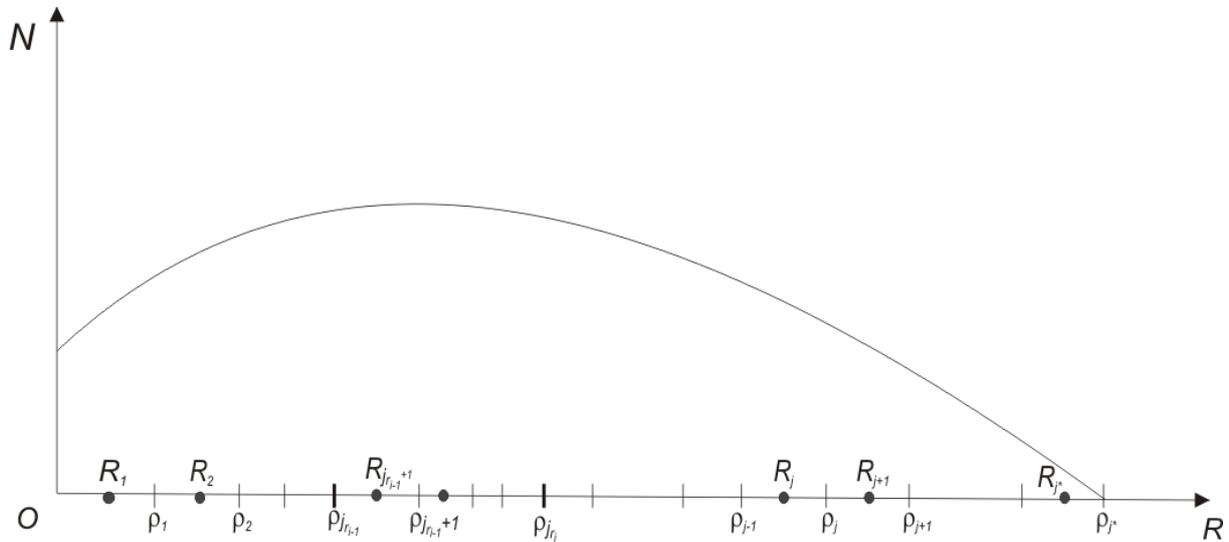


Рис. 4. Функция распределения включений по размерам

Рассмотрим находящийся в рассматриваемом объеме шар радиуса $R = R_j$ ($R > r_i$). Найдем вероятность того, что его срез попадет в рассматриваемый диапазон и радиус его сечения будет находиться в интервале $(r_{i-1}, r_i]$. Это произойдет в случае, если центр шара будет находиться на расстоянии h ($h_2 \leq h \leq h_1$) от плоскости шлифа (в обе стороны от нее), где $h_1 = \sqrt{R^2 - r_{i-1}^2}$, $h_2 = \sqrt{R^2 - r_i^2}$, как это показано на рис. 5.

Считая распределение центров шаров в пространстве равномерным, получим, что эта вероятность равна

$$P_{(r_{i-1}, r_i]}(R) = P(r_{i-1} < r \leq r_i) = 2 \frac{h_1 - h_2}{H} = \frac{2}{H} \left(\sqrt{R^2 - r_{i-1}^2} - \sqrt{R^2 - r_i^2} \right) \quad (1)$$

В случае, если $r_{i-1} < R \leq r_i$, радиус среза не может превзойти r_i и

$$P_{(r_{i-1}, r_i]}(R) = P(r_{i-1} < r \leq r_i) = 2 \frac{h_1}{H} = \frac{2}{H} \sqrt{R^2 - r_{i-1}^2} \quad (2)$$

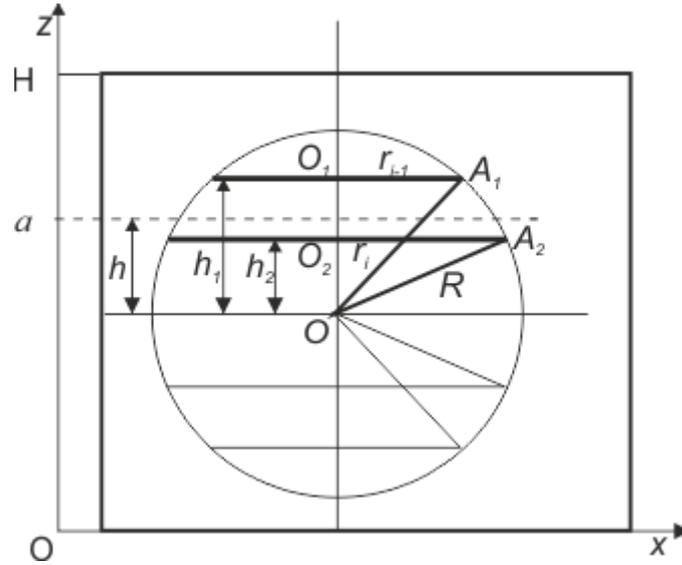


Рис. 5. К определению вероятности среза данного размера

В случае, если $R \leq r_{i-1}$, включение не может образовать среза, соответствующего данному диапазону и $p_{(r_{i-1}, r_i]}(R) = 0$.

Мы рассмотрели вероятность среза выбранного размера от одного включения, а всего у нас $n_j = N(R_j)SH\Delta\rho_j$ одинаковых (по нашему подходу) включений. Мы имеем классическую схему Бернулли с вероятностью «успеха» $p_j = p_{(r_{i-1}, r_i]}(R_j)$ и с вероятностью «неудачи» $q_j = 1 - p_j$ с числом «испытаний» $n_j = N(R_j)SH\Delta\rho_j$. Заметим, что $p_j < \frac{2R_j}{H}$, где R_j — величина порядка микрометров, а H — порядка сантиметров. Следовательно, p_j — это очень малая величина (менее 0,001), а $q_j = 1 - p_j$ — величина, близкая единице.

Математическое ожидание числа срезов выбранного размера равно

$$n_j p_j = \begin{cases} N(R_j)SH\Delta\rho_j \cdot \frac{2}{H} \sqrt{R_j^2 - r_{i-1}^2} = 2S\sqrt{R_j^2 - r_{i-1}^2} N(R_j)\Delta\rho_j, & (r_{i-1} < R_j \leq r_i), \\ 2S(\sqrt{R_j^2 - r_{i-1}^2} - \sqrt{R_j^2 - r_i^2}) N(R_j)\Delta\rho_j, & (R_j > r_i). \end{cases} \quad (3)$$

Дисперсия числа срезов выбранного размера равна

$$n_j p_j q_j = \begin{cases} N(R_j)SH\Delta\rho_j \cdot \frac{2}{H} \sqrt{R_j^2 - r_{i-1}^2} \cdot \left(1 - \frac{2}{H} \sqrt{R_j^2 - r_{i-1}^2}\right) = \\ = 2S\sqrt{R_j^2 - r_{i-1}^2} \cdot \left(1 - \frac{2}{H} \sqrt{R_j^2 - r_{i-1}^2}\right) N(R_j)\Delta\rho_j & (r_{i-1} < R_j \leq r_i), \\ 2S(\sqrt{R_j^2 - r_{i-1}^2} - \sqrt{R_j^2 - r_i^2}) \cdot \left(1 - \frac{2}{H}(\sqrt{R_j^2 - r_{i-1}^2} - \sqrt{R_j^2 - r_i^2})\right) N(R_j)\Delta\rho_j & (R_j > r_i). \end{cases} \quad (4)$$

Выражения (4) показывают, что в сомножителе $q_j = 1 - p_j$ следует пренебречь величиной p_j , так как, во-первых, $p_j \ll 1$ и, во-вторых, в противном случае, мы получили бы,

что распределение срезов включений зависит от высоты образца, чего не может быть. Картина срезов включений (и, в частности, дисперсия числа срезов включений) в данном месте не может зависеть от того, какой высоты образец мы при этом рассматриваем.

Таким образом, мы получили, что и математическое ожидание и дисперсия числа срезов $Y_{(r_{i-1}, r_i].j}$ включений из j -го интервала (радиусов включений) по R , дающих срезы из i -го диапазона (срезов включений) по r , равны одной и той же величине $\eta_{(r_{i-1}, r_i]}(R_j)\Delta\rho_j$, где

$$\eta_{(r_{i-1}, r_i]}(R_j) = \begin{cases} 2S\sqrt{R_j^2 - r_{i-1}^2}N(R_j), & \text{если } r_{i-1} < R_j \leq r_i, \\ 2S\left(\sqrt{R_j^2 - r_{i-1}^2} - \sqrt{R_j^2 - r_i^2}\right)N(R_j), & \text{если } R_j > r_i. \end{cases} \quad (5)$$

Свой вклад в рассматриваемый диапазон срезов дают включения из всех интервалов по R , начиная с $j_{r_{i-1}} + 1$. Полное число срезов в рассматриваемом диапазоне равно

$$Y_{(r_{i-1}, r_i]} = \sum_{j=j_{r_{i-1}}+1}^{j^*} Y_{(r_{i-1}, r_i].j}. \quad (6)$$

Математическое ожидание (и дисперсия) этой величины равны

$$\mu_i = M\left(Y_{(r_{i-1}, r_i]}\right) = \sum_{j=j_{r_{i-1}}+1}^{j^*} M\left(Y_{(r_{i-1}, r_i].j}\right) = \sum_{j=j_{r_{i-1}}+1}^{j^*} \eta_{(r_{i-1}, r_i]}(R_j)\Delta\rho_j$$

или, переходя к пределу при условии $\max_j \Delta\rho_j \rightarrow 0$,

$$\mu_i = \lim_{\max_j \Delta\rho_j \rightarrow 0} \sum_{j=j_{r_{i-1}}+1}^{j^*} \eta_{(r_{i-1}, r_i]}(R_j)\Delta\rho_j = \int_{r_{i-1}}^{R^*} \eta_{(r_{i-1}, r_i]}(\rho)d\rho.$$

Поменяв обозначение переменной интегрирования ρ на R и разбив, в соответствии с (5), отрезок интегрирования на две части, получим математическое ожидание и дисперсию полного числа срезов включений, попадающих в размерный диапазон $(r_{i-1}, r_i]$:

$$\mu_i = 2S \left(\int_{r_{i-1}}^{r_i} N(R)\sqrt{R^2 - r_{i-1}^2}dR + \int_{r_i}^{\infty} N(R)\left(\sqrt{R^2 - r_{i-1}^2} - \sqrt{R^2 - r_i^2}\right)dR \right). \quad (7)$$

Покажем, что распределение случайной величины $Y_{(r_{i-1}, r_i]}$ можно считать нормальным. Действительно, случайная величина $Y_{(r_{i-1}, r_i]}$ является суммой n слагаемых $Y_{(r_{i-1}, r_i]} = Y_1 + Y_2 + \dots + Y_n$, где n — общее число включений с радиусом большим r_{i-1} в рассматриваемом образце (в нашем случае $n = SH \int_{r_{i-1}}^{\infty} N(R)dR$). Здесь Y_k — случайная величина, равная 1, если срез от k -го включения попал в рассматриваемый размерный диапазон и равный нулю, если не попал, или вообще k -ое включение не попало в плоскость среза: $P(Y_k = 1) = p_k$, $P(Y_k = 0) = q_k = 1 - p_k$. Все величины Y_k можно считать независимыми. Так как все Y_k равномерно ограничены и дисперсия суммы их $Y_{(r_{i-1}, r_i]} = \sum_{k=1}^n Y_k$ имеет

дисперсию $S_n^2 = \sum_{k=1}^n p_k q_k$, стремящуюся к бесконечности при $n \rightarrow \infty$, то условия Линдеберга [14] выполняются и распределение центрированной суммы $\frac{Y_{(r_{i-1}, r_i]} - MY_{(r_{i-1}, r_i]}}{S_n}$ стремится к стандартному нормальному распределению $N(0,1)$ с математическим ожиданием, равным нулю, и средним квадратическим отклонением, равным 1.

В нашем случае общее число включений в образце — это очень большая величина. Поэтому считаем, что величина $\frac{Y_{(r_{i-1}, r_i]} - MY_{(r_{i-1}, r_i]}}{S_n} = \frac{Y_{(r_{i-1}, r_i]} - \mu_i}{\sqrt{\mu_i}}$ также распределена нормально с параметрами $(0,1)$. Следовательно, величина $Y_{(r_{i-1}, r_i]}$ распределена нормально с математическим ожиданием μ_i и средним квадратическим отклонением $\sqrt{\mu_i}$, где μ_i определяется выражением (7).

3. Сущность метода

3.1. Основные упрощения

Отметим на оси радиусов срезов точки r_1, \dots, r_M , являющиеся границами интервалов, в которых будем производить подсчет. Для каждого интервала $(r_{i-1}, r_i]$ мы можем, используя формулу (7), зная функцию $N(R)$, вычислить свое μ_i .

Разобьем ось радиусов включений на те же самые интервалы, что и ось срезов. Сделаем следующее упрощение. Будем считать, что функция $N(R)$ в каждом таком интервале постоянна: $N(R) = N_j$ для $r_{j-1} < R \leq r_j$, $j = 1, 2, \dots, M$.

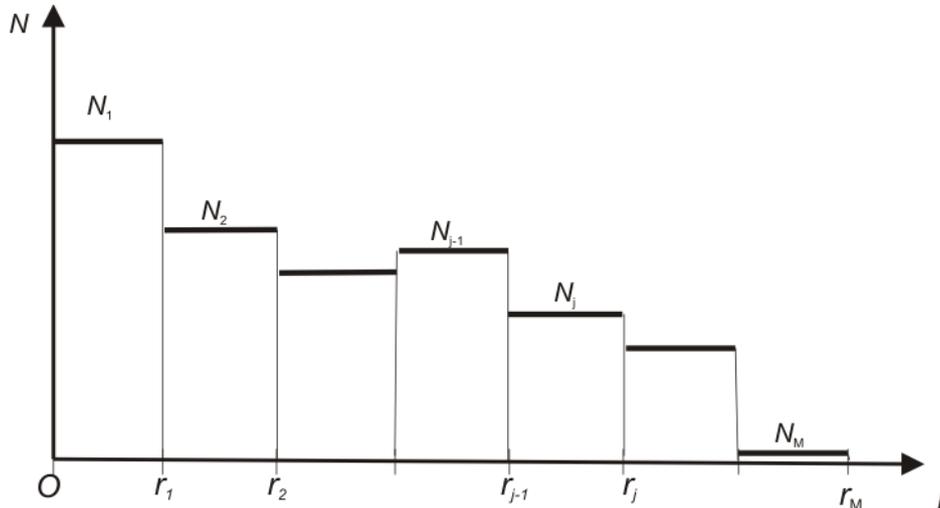


Рис. 6. Упрощенная функция распределения включений по размерам

Теперь формулу (7) можно записать в виде

$$\mu_i = 2S \left(N_i \int_{r_{i-1}}^{r_i} \sqrt{R^2 - r_{i-1}^2} dR + \sum_{j=i+1}^M N_j \int_{r_{j-1}}^{r_j} \left(\sqrt{R^2 - r_{i-1}^2} - \sqrt{R^2 - r_i^2} \right) dR \right). \quad (8)$$

Вычисляя интегралы, получим для $i = 1$:

$$\mu_1 = S \left(N_1 r_1^2 + N_2 \left(r_2^2 - r_1^2 - r_1^2 \ln \frac{r_2 - \sqrt{r_2^2 - r_1^2}}{r_1} - r_2 \sqrt{r_2^2 - r_1^2} \right) + \sum_{j=3}^M N_j \left(r_j^2 - r_{j-1}^2 - r_1^2 \ln \frac{r_j - \sqrt{r_j^2 - r_1^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_1^2}} - r_j \sqrt{r_j^2 - r_1^2} + r_{j-1} \sqrt{r_{j-1}^2 - r_1^2} \right) \right), \quad (9)$$

и для $i > 1$:

$$\begin{aligned} \mu_i = S & \left(N_i \left(r_{i-1}^2 \ln \frac{r_i - \sqrt{r_i^2 - r_{i-1}^2}}{r_{i-1}} + r_i \sqrt{r_i^2 - r_{i-1}^2} \right) + \right. \\ & + N_{i+1} \left(r_{i-1}^2 \ln \frac{r_{i+1} - \sqrt{r_{i+1}^2 - r_{i-1}^2}}{r_i - \sqrt{r_i^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_{i+1} - \sqrt{r_{i+1}^2 - r_i^2}}{r_i} + \right. \\ & \left. \left. + r_{i+1} \left(\sqrt{r_{i+1}^2 - r_{i-1}^2} - \sqrt{r_{i+1}^2 - r_i^2} \right) - r_i \sqrt{r_i^2 - r_{i-1}^2} \right) + \right. \\ & \left. + \sum_{j=i+2}^M N_j \left(r_{i-1}^2 \ln \frac{r_j - \sqrt{r_j^2 - r_{i-1}^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_j - \sqrt{r_j^2 - r_i^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_i^2}} + \right. \right. \\ & \left. \left. + r_j \left(\sqrt{r_j^2 - r_{i-1}^2} - \sqrt{r_j^2 - r_i^2} \right) - r_{j-1} \left(\sqrt{r_{j-1}^2 - r_{i-1}^2} - \sqrt{r_{j-1}^2 - r_i^2} \right) \right) \right). \end{aligned} \quad (10)$$

3.2. Вывод основных соотношений

Введем величины φ_{ij} , являющиеся коэффициентами при N_j в выражениях (9), (10):

$$\varphi_{1j} = \begin{cases} r_1^2, & \text{если } j=1, \\ r_2^2 - r_1^2 - r_1^2 \ln \frac{r_2 - \sqrt{r_2^2 - r_1^2}}{r_1} - r_2 \sqrt{r_2^2 - r_1^2} & \text{если } j=2, \\ r_j^2 - r_{j-1}^2 - r_1^2 \ln \frac{r_j - \sqrt{r_j^2 - r_1^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_1^2}} - r_j \sqrt{r_j^2 - r_1^2} + r_{j-1} \sqrt{r_{j-1}^2 - r_1^2} & \text{если } j > 2, \end{cases} \quad (11)$$

$$\varphi_{ij} = \begin{cases} 0, \text{ если } j < i, \\ r_{i-1}^2 \ln \frac{r_i - \sqrt{r_i^2 - r_{i-1}^2}}{r_{i-1}} + r_i \sqrt{r_i^2 - r_{i-1}^2}, \text{ если } i > 1, j = i, \\ r_{i-1}^2 \ln \frac{r_j - \sqrt{r_j^2 - r_{i-1}^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_j - \sqrt{r_j^2 - r_i^2}}{r_{j-1}} + \\ + r_j \left(\sqrt{r_j^2 - r_{i-1}^2} - \sqrt{r_j^2 - r_i^2} \right) - r_{j-1} \sqrt{r_{j-1}^2 - r_{i-1}^2}, \text{ если } i > 1, j = i + 1, \\ r_{i-1}^2 \ln \frac{r_j - \sqrt{r_j^2 - r_{i-1}^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_j - \sqrt{r_j^2 - r_i^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_i^2}} + \\ + r_j \left(\sqrt{r_j^2 - r_{i-1}^2} - \sqrt{r_j^2 - r_i^2} \right) - r_{j-1} \left(\sqrt{r_{j-1}^2 - r_{i-1}^2} - \sqrt{r_{j-1}^2 - r_i^2} \right), \text{ если } i > 1, j > i + 1 \end{cases} \quad (12)$$

$(i, j = 1, \dots, M)$.

Обозначим символом $\mu_{l,i}$ математическое ожидание числа $Y_{l,i}$ срезов из i -го диапазона на шлифе l ($l = 1, \dots, L$) с площадью S_l :

$$M(Y_{l,i}) = \mu_{l,i}. \quad (13)$$

В соответствии с выше изложенным, среднее квадратическое отклонение $\sigma(Y_{l,i})$ величины $Y_{l,i}$ равно $\sigma(Y_{l,i}) = \sqrt{\mu_{l,i}}$ и, следовательно, нормированная величина

$$Z_{l,i} = \frac{Y_{l,i} - \mu_{l,i}}{\sqrt{\mu_{l,i}}} \quad (14)$$

распределена нормально с математическим ожиданием равным 0 и средним квадратическим отклонением, равным 1: $Z_{l,i} \in N(0,1)$.

Введем также величины

$$\lambda_i = \sum_{j=1}^M \varphi_{ij} N_j \quad (i = 1, \dots, M) \quad . \quad (15)$$

В соответствии с (9–12) и (15)

$$\mu_{l,i} = S_l \lambda_i \quad . \quad (16)$$

Величина λ_i имеет смысл математического ожидания числа срезов из i -го диапазона на единичной площади.

Пусть N_1^e, \dots, N_M^e — оценки неизвестных величин N_1, \dots, N_M соответственно. Подберем их так, чтобы минимизировать уклонение

$$F(N_1^e, \dots, N_M^e) = \sum_{l=1}^L \sum_{i=1}^M \frac{(y_{l,i} - \mu_{l,i}^e)^2}{\mu_{l,i}^e} \quad (17)$$

при условии $N_j^e \geq 0$ ($j = 1, \dots, M$). Здесь $y_{l,i}$ — реализация случайной величины $Y_{l,i}$, то есть число срезов включений из i -го размерного диапазона, обнаруженных при обследовании l -го шлифа; по аналогии с (15), (16),

$$\mu_{l,i}^e = S_l \lambda_i^e, \quad (18)$$

где

$$\lambda_i^e = \sum_{j=1}^M \varphi_{ij} N_j^e, \quad j = 1, \dots, M. \quad (19)$$

Преобразуя выражение (17) следующим образом

$$F = \sum_{l=1}^L \left(\sum_{i=1}^M \frac{(y_{l,i} - S_l \lambda_i^e)^2}{S_l \lambda_i^e} \right), \quad (20)$$

получим

$$\begin{aligned} F &= \sum_{i=1}^M \left(\sum_{l=1}^L \frac{y_{l,i}^2 - 2y_{l,i} S_l \lambda_i^e + S_l^2 (\lambda_i^e)^2}{S_l \lambda_i^e} \right) = \sum_{i=1}^M \left(\sum_{l=1}^L \left(\frac{y_{l,i}^2}{S_l \lambda_i^e} - 2y_{l,i} + S_l \lambda_i^e \right) \right) = \\ &= \sum_{i=1}^M \left(\frac{1}{\lambda_i^e} \sum_{l=1}^L \frac{y_{l,i}^2}{S_l} - 2 \sum_{l=1}^L y_{l,i} + \lambda_i^e \sum_{l=1}^L S_l \right) = \sum_{i=1}^M \left(\frac{1}{\lambda_i^e} \sum_{l=1}^L \frac{y_{l,i}^2}{S_l} - 2 \sum_{l=1}^L y_{l,i} + \lambda_i^e \sum_{l=1}^L S_l \right). \end{aligned}$$

Введем величины

$$\sum_{l=1}^L S_l = S, \quad \sum_{l=1}^L \sum_{i=1}^M y_{l,i} = \bar{y}_\Sigma, \quad \sqrt{\frac{1}{S} \sum_{l=1}^L \frac{y_{l,i}^2}{S_l}} = \tilde{y}_i. \quad (21)$$

Тогда

$$F = \sum_{i=1}^M \left(\frac{S}{\lambda_i^e} \left(\frac{1}{S} \sum_{l=1}^L \frac{y_{l,i}^2}{S_l} \right) + \lambda_i^e S \right) - 2 \sum_{i=1}^M \sum_{l=1}^L y_{l,i} = \sum_{i=1}^M \left(\frac{S}{\lambda_i^e} \tilde{y}_i^2 + \lambda_i^e S \right) - 2 \bar{y}_\Sigma$$

или

$$F = S \sum_{i=1}^M \left(\frac{\tilde{y}_i^2}{\lambda_i^e} + \lambda_i^e \right) - 2 \bar{y}_\Sigma. \quad (22)$$

Частная производная F по λ_j^e равна

$$\frac{\partial F}{\partial \lambda_j^e} = S \left(1 - \left(\frac{\tilde{y}_j}{\lambda_j^e} \right)^2 \right) \quad (23)$$

Так как

$$\frac{\partial^2 F}{\partial \lambda_i^e \partial \lambda_j^e} = \begin{cases} 2 \frac{\tilde{y}_j^2}{(\lambda_j^e)^3} (> 0), & \text{если } i = j, \\ 0, & \text{если } i \neq j, \end{cases},$$

то матрица вторых производных функции $F(\lambda_1^e, \dots, \lambda_m^e)$ положительно определена при положительных λ^e (а только такие значения допускаются из физических соображений).

Таким образом, система уравнений

$$\frac{\partial F}{\partial \lambda_j^e} = 0, \quad j = 1, \dots, M, \quad (24)$$

в случае, если ее решения неотрицательны, определяет минимум функции F . Из (23), (24) следует, что

$$\lambda_j^e = \tilde{y}_j, \quad j = 1, \dots, M. \quad (25)$$

Найдя $\lambda_1^e, \dots, \lambda_M^e$, можно найти соответствующие значения N_1^e, \dots, N_M^e . Для этого введем матрицы

$$\bar{\Phi} = \begin{pmatrix} \varphi_{11} & \dots & \varphi_{1M} \\ \dots & \dots & \dots \\ \varphi_{M1} & \dots & \varphi_{MM} \end{pmatrix}, \quad \bar{\Lambda}^e = \begin{pmatrix} \lambda_1^e \\ \dots \\ \lambda_M^e \end{pmatrix}, \quad \bar{N}^e = \begin{pmatrix} N_1^e \\ \dots \\ N_M^e \end{pmatrix}. \quad (26)$$

Соотношения (19) в матричном виде могут быть записаны как

$$\bar{\Lambda}^e = \bar{\Phi} \cdot \bar{N}^e, \quad (27)$$

откуда

$$\bar{N}^e = \bar{\Phi}^{-1} \bar{\Lambda}^e. \quad (28)$$

Найдем значение функции F в точке минимума:

$$F_{\min} = S \sum_{i=1}^M \left(\frac{\tilde{y}_i^2}{\lambda_i^e} + \lambda_i^e \right) - 2\bar{y}_\Sigma = S \sum_{i=1}^M \left(\frac{\tilde{y}_i^2}{\tilde{y}_i} + \tilde{y}_i \right) - 2\bar{y}_\Sigma \quad \text{или} \\ F_{\min} = 2 \left(S \sum_{i=1}^M \tilde{y}_i - \bar{y}_\Sigma \right). \quad (29)$$

4. Алгоритм расчета

Таким образом, для определения наиболее вероятных количеств включений каждого размерного диапазона необходимо

Шаг 1. Взять образец исследуемого металла, сделать из него несколько шлифов и на каждом подсчитать количество срезов, попадающих в каждый размерный диапазон.

Шаг 2. Рассчитать величины $S, \bar{y}_\Sigma, \tilde{y}_i, \lambda_j^e$, по формулам (21), (25).

Шаг 3. Рассчитать оценки плотностей распределения частиц \bar{N}^e по формуле (28).

Шаг 4. В случае, если какие-то \bar{N}^e меньше нуля, переточить шлиф, подсчитать срезы включений и произвести новые расчеты с добавленными данными.

Шаг 5. Рассчитать количество частиц в каждом размерном интервале:

$$N_i = N_i^e \Delta r_i,$$

где Δr_i — длина i -го размерного интервала ($\Delta r_i = r_i - r_{i-1}$).

Шаг 6. СТОП.

5. Пример расчета

Ниже приведен пример расчета для исходных данных, указанных в табл. 1.

Таблица 1

Исходные данные для расчета

№ шлифа	Обследованная площадь, мм ²	Верхние границы диапазонов размеров включений, мкм							
		1	2	3	5	7	10	15	20
1	100	83	58	19	20	8	3	6	3
2	100	71	86	16	27	8	3	4	2
3	100	75	69	27	16	13	6	4	3

Расчеты по приведенному алгоритму, дают результаты, приведенные в табл. 2.

Таблица 2

Наиболее вероятные значения истинных количеств включений

Колич., мм ³	Верхние границы диапазонов размеров включений, мкм							
	1	2	3	5	7	10	15	20
	989,80	559,21	98,91	60,44	25,58	5,24	4,11	3,16

Заключение

Вредное влияние неметаллических включений является постоянной проблемой сталеплавленного производства. Для совершенствования методов снижения этого вреда необходимо уметь оценивать количество, размеры, форму и состав неметаллических включений, возникающих в готовой стали при различных режимах и технологиях плавки.

В данном исследовании решена задача оценки истинного количества и размеров неметаллических включений на основе данных, получаемых при исследовании поверхности шлифа образца исследуемого металла.

Пока метод разработан для включений сферической формы, однако он может быть обобщен на частицы других форм. Следующим этапом работы в этом направлении планируется обобщение метода для частиц, имеющих форму эллипсоида и прямоугольного параллелепипеда.

Получение исходных данных и расчеты могут быть легко реализованы на современных микроскопах, которые включают в себя компьютер, умеют распознавать образы и производить необходимые вычисления.

Литература

1. Lipiński T., Wach A. Size of Non-Metallic Inclusions High-Grade Medium Carbon Steel // Archives of Foundry Engineering. 2012. Vol. 14, No. 4. P. 55–60.
2. Lambrighs K., Verpoest L., et al. Influence of Non-Metallic Inclusions on the Fatigue Properties of Heavily Cold Drawn Steel Wires // Procedia Engineering. 2010. Vol. 2. Iss 1. P. 173–181. DOI: 10.1016/j.proeng.2010.03.019
3. Zeng D., Tian G., et al. Fatigue Strength Prediction of Drilling Materials Based on the Maximum Non-Metallic Inclusion Size // Journal of Materials Engineering and Performance. 2015. Vol. 24, Iss. 12. P. 4664–4672. DOI: 10.1007/s11665-015-1753-1

4. Рошин В.Е., Рошин А.В. Электрометаллургия и металлургия стали. Челябинск: Издательский центр ЮУрГУ. 2013. 571 с.
5. ГОСТ ИСО 4967:1998. Сталь. Определение содержания неметаллических включений. Металлографический метод с использованием эталонных шкал. Издание официальное. М.: Стандартинформ, 2010. 32 с.
6. Салтыков С.А. Стереометрическая металлография. М.: Металлургия, 1970. 376 с.
7. Чернявский К.С. Стереология в металловедении. М.: Металлургия, 1977. 280 с.
8. Chraponski J., Cwajna J., et al. Usefulness Evaluation of the Stereological Methods Applied for Grain Size Estimation // Acta Stereologica. 1999. Vol. 18, No. 1. P. 81–88.
9. Davtian A., Hahn U., et al. Estimating Number Density NV — a Comparison of an Improved Saltykov Estimator and the Disector Method // Image Analysis & Stereology. 2000. Vol. 19, No. 3. P. 209–214. DOI: 10.5566/ias.v19
10. Anderssen R., Jakeman A. Computational Methods in Stereology // Proceedings of the Fourth International Congress for Stereology. Gaithersburg, Maryland, USA. September 4–9, 1975. National Bureau of Standards. U.S. government printing office. Washington: 1976. P. 13–18.
11. Jakeman A., Anderssen R. On Optimal Forms for Stereological Data // Proceedings of the Fourth International Congress for Stereology, Gaithersburg, Maryland, USA, September 4–9, 1975. National Bureau of Standards. U.S. government printing office. Washington: 1976. P. 69–74.
12. Nicholson W. Estimation of Linear Functionals by Maximum Likelihood // Proceedings of the Fourth International Congress for Stereology, Gaithersburg, Maryland, USA, September 19–24, 1975. National Bureau of Standards. U.S. government printing office. Washington: 1976. P. 19–24.
13. Drozin A.D. Calculating of the True Sizes and the Numbers of Spherical Inclusions in Metal // Metallography, Microstructure, and Analysis. 2017. Vol. 6, Iss. 3. P. 240–246. DOI: 10.1007/s13632-017-0354-9
14. Феллер. В. Введение в теорию вероятностей и ее приложения. Т. 2. М.: Мир, 1963. 754 с.

Дрозин Александр Дмитриевич, д.т.н., профессор, Центр элитного образования, директор, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

**THE METHOD OF PROCESSING INFORMATION
ABOUT NONMETALLIC INCLUSIONS OBTAINED
FROM THE INVESTIGATION OF THE SURFACE
OF FINISHED STEEL SAMPLES**

© 2017 A.D. Drozin

South Ural State University

(76, Lenin Avenue, Chelyabinsk 454080),

E-mail: drozinad@susu.ru

Received: 23.10.2017

Nonmetallic inclusions formed in the process of steel production are microparticles of chemical reactions products and foreign impurities having dimensions of 0.1–100 microns. They concentrate stresses in finished products and impair the mechanical properties of steel. To study them, sections are made from the sample of the tested metal and microscopic sections of inclusions are observed in the microscope. However, on the section plane a researcher does not see the inclusions themselves, but only their sections, while inclusions of different sizes can give the sections of the same size and, conversely, the inclusions of the same size can produce the sections of various sizes. In this paper, we consider a method for determining the amounts and sizes of nonmetallic inclusions on the basis of this information, that is, reconstructing the volume picture of the inclusion size distribution based on epy plane distribution of their sections by a section plane. The method is based on probability theory and mathematical statistics. It is shown that the number of inclusion sections into each size interval obeys the normal law. A technique for determining the distribution function of inclusions with respect to sizes for inclusions of a spherical shape and an appropriate algorithm has been developed.

Keywords: information processing, crystallography, nonmetallic inclusions, stereology.

FOR CITATION

Drozin A.D. The Method of Processing Information about Nonmetallic Inclusions Obtained from the Investigation of the Surface of Finished Steel Samples Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. vol. 6, no. 4. pp. 05–19. (in Russian) DOI: 10.14529/cmse170401.

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. Lipiński T., Wach A. Size of Non-Metallic Inclusions High-Grade Medium Carbon Steel. Archives of Foundry Engineering. 2012. vol. 14, no. 4. pp. 55–60.
2. Lambrighs K., Verpoest L., et al. Influence of Non-Metallic Inclusions on the Fatigue Properties of Heavily Cold Drawn Steel Wires. Procedia Engineering. 2010. vol. 2, iss. 1. pp. 173–181. DOI: 10.1016/j.proeng.2010.03.019
3. Zeng D., Tian G., et al. Fatigue Strength Prediction of Drilling Materials Based on the Maximum Non-Metallic Inclusion Size. Journal of Materials Engineering and Performance. 2015. vol 24, iss. 12. pp. 4664–4672. DOI: 10.1007/s11665-015-1753-1

4. Roshchin, V.E., Roshchin, A.V. Elektrometallurgiya i metallurgiya stali [Electrometallurgy and Metallurgy of Steel]. Chelyabinsk. Publishing Center of SUSU. 2013. 571 p.
5. GOST ISO 4967:1998. Stal'. Opredelenie sodержaniya nemetallicheskih vklyuchenii. Metallograficheskii metod s ispol'zovaniem etalonnnykh shkal. Izdanie ofitsial'noe [ISO 4967:1998. Steel. Determination of the Content of Nonmetallic Inclusions. Metallographic Method Using Reference Scales. Official Edition]. Moscow. Standartinform. 2010. 32 p.
6. Saltykov S.A. Stereometricheskaya metallografiya [Stereometric Metallography]. Moscow. Metallurgy. 1970. 376 p.
7. Chernyavskii K.S. Stereologiya v metallovedenii [Stereology in Physical Metallurgy]. Moscow. Metallurgy. 1977. 280 p.
8. Chraponski J., Cwajna J., et al. Usefulness Evaluation of the Stereological Methods Applied for Grain Size Estimation. Acta Stereologica. 1999. vol. 18, no. 1. pp. 81–88.
9. Davtian A., Hahn U., et al. Estimating Number Density NV — a Comparison of an Improved Saltykov Estimator and the Disector Method. Image Analysis & Stereology. 2000. vol. 19, no. 3. pp. 209-214. DOI: 10.5566/ias.v19
10. Anderssen R., Jakeman A. Computational Methods in Stereology. Proceedings of the Fourth International Congress for Stereology, Gaithersburg, Maryland, USA, September 4–9, 1975. National Bureau of Standards. U.S. government printing office. Washington. 1976. pp. 13–18.
11. Jakeman A., Anderssen R., On Optimal Forms for Stereological Data. Proceedings of the Fourth International Congress for Stereology, Gaithersburg, Maryland, USA, September 4–9, 1975. National Bureau of Standards. U.S. government printing office. Washington. 1976. pp. 69–74.
12. Nicholson W. Estimation of Linear Functionals by Maximum Likelihood. Proceedings of the Fourth International Congress for Stereology, Gaithersburg, Maryland, USA, September 19–24, 1975. National Bureau of Standards. U.S. government printing office. Washington. 1976. pp. 19–24.
13. Drozin A.D. Calculating of the True Sizes and the Numbers of Spherical Inclusions in Metal. Metallography, Microstructure, and Analysis. 2017. vol. 6, iss. 3. pp. 240–246. DOI: 10.1007/s13632-017-0354-9.
14. Feller W. An Introduction to Probability Theory and its Applications, vol. 2. John Wiley & Sons Inc. New York, London, Sydney. 1968. 754 p.

МЕТОД ОБРАБОТКИ ИНФОРМАЦИИ, ПОЛУЧАЕМОЙ ПРИ ОБСЛЕДОВАНИИ МИКРОШЛИФА ГОТОВОЙ СТАЛИ В СЛУЧАЕ НЕРЕПРЕЗЕНТАТИВНЫХ ДАННЫХ

© 2017 А.Д. Дрозин, Н.М. Япарова

Южно-Уральский государственный университет

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

E-mail: drozinad@susu.ru, iaparovam@susu.ru

Поступила в редакцию: 23.10.2017

Для исследования неметаллических включений — вредных примесей, образовавшихся в процессе выплавки и кристаллизации стали, из образца исследуемого металла готовят полированный микрошлиф и рассматривают в микроскоп срезы включений плоскостью микрошлифа. При этом обычно делят возможные размеры срезов на интервалы и подсчитывают, сколько срезов включений попало в каждый интервал. В принципе, зависимость числа срезов от размера должна быть монотонно убывающей функцией. Однако исследователь может столкнуться со случаем, когда эта зависимость не выполняется. Например, в некоторые диапазоны размеров вообще не попало никаких срезов (хотя в более старших диапазонах срезы есть). Если к таким данным применить известные методики определения числа и размеров включений (давших эти срезы), то в некоторых диапазонах получатся отрицательные значения. Такое может произойти, когда включений мало и реализовался случай среза, далекий от наиболее вероятного. В работе предлагается методика, позволяющая, несмотря на это, пусть с некоторой погрешностью, рассчитать истинные количества и размеры неметаллических включений в объеме исследуемого металла. Применен численный метод условной оптимизации функции отклонений истинных количеств срезов включений от теоретического распределения.

Ключевые слова: обработка информации, кристаллография, неметаллические включения, стереология, условная оптимизация.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Дрозин А.Д., Япарова Н.М. Метод обработки информации, получаемой при обследовании микрошлифа готовой стали в случае нерепрезентативных данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 20–27. DOI: 10.14529/cmse170402.

Введение

В процессе выплавки сталь загрязняется продуктами протекающих в ней химических реакций и внешними примесями. Эти загрязнения, представляющие собой микрочастицы размером от 0,1 до 100 мкм, являются неметаллическими включениями в стали. Они, во многом, определяют качество готовой стали [1–4].

В работе [5] был предложен метод определения распределения включений по размерам на основе распределения их срезов плоскостью шлифа. Метод заключается в следующем. Из образца металла готовят полированный микрошлиф и, с помощью микроскопа, определяют распределение по размерам срезов попавших в сечение включений следующим образом. Возможные размеры радиусов срезов включений делят на M интервалов: $(0, r_1], (r_1, r_2], (r_2, r_3], \dots, (r_{M-1}, r_M]$ и подсчитывают числа срезов, попавших в каждый размерный интервал. Для повышения точности, шлиф несколько раз перетачивают и снова определяют количества срезов, попавших в каждый размерный интервал. Пусть, таким образом, было просмотрено L шлифов с площадями S_1, \dots, S_L соответственно и получены

количества $y_{l,i}$ ($l = 1, \dots, L; i = 1, \dots, M$) срезов, попадающих в i -ый размерный интервал $(r_{i-1}, r_i]$ при обследовании шлифа с номером l с площадью S_l .

Считая, что плотность истинного (неизвестного) распределения включений по размерам постоянна на каждом таком интервале, в [5] получили оценку математического ожидания $\mu_{l,i}^e$ числа срезов включений, попадающих в i -ый размерный интервал на шлифе с номером l , в виде

$$\mu_{l,i}^e = S_l \lambda_i^e, \quad (1)$$

где λ_i^e имеет смысл оценки математического ожидания числа срезов из i -го размерного диапазона на единичной площади и определяется по формуле

$$\lambda_i^e = \sum_{j=1}^M \varphi_{ij} N_j^e, \quad j = 1, \dots, M. \quad (2)$$

Здесь N_j^e — оценка истинной функции распределения включений по размерам на j -ом размерном интервале (неизвестная величина), φ_{ij} — коэффициент определяющий долю включений из j -го размерного интервала включений, образующих срезы, попадающих в i -ый размерный интервал срезов. Для сферических включений они находятся по формулам [5]:

$$\varphi_{1j} = \begin{cases} r_1^2, & \text{если } j = 1, \\ r_2^2 - r_1^2 - r_1^2 \ln \frac{r_2 - \sqrt{r_2^2 - r_1^2}}{r_1} - r_2 \sqrt{r_2^2 - r_1^2}, & \text{если } j = 2, \\ r_j^2 - r_{j-1}^2 - r_1^2 \ln \frac{r_j - \sqrt{r_j^2 - r_1^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_1^2}} - r_j \sqrt{r_j^2 - r_1^2} + r_{j-1} \sqrt{r_{j-1}^2 - r_1^2}, & \text{если } j > 2, \end{cases} \quad (3)$$

$$\varphi_{ij} = \begin{cases} 0, & \text{если } j < i, \\ r_{i-1}^2 \ln \frac{r_i - \sqrt{r_i^2 - r_{i-1}^2}}{r_{i-1}} + r_i \sqrt{r_i^2 - r_{i-1}^2}, & \text{если } i > 1, \quad j = i, \\ r_{i-1}^2 \ln \frac{r_j - \sqrt{r_j^2 - r_{i-1}^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_j - \sqrt{r_j^2 - r_i^2}}{r_{j-1}} + \\ + r_j \left(\sqrt{r_j^2 - r_{i-1}^2} - \sqrt{r_j^2 - r_i^2} \right) - r_{j-1} \sqrt{r_{j-1}^2 - r_{i-1}^2}, & \text{если } i > 1, \quad j = i + 1, \\ r_{i-1}^2 \ln \frac{r_j - \sqrt{r_j^2 - r_{i-1}^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_{i-1}^2}} - r_i^2 \ln \frac{r_j - \sqrt{r_j^2 - r_i^2}}{r_{j-1} - \sqrt{r_{j-1}^2 - r_i^2}} + \\ + r_j \left(\sqrt{r_j^2 - r_{i-1}^2} - \sqrt{r_j^2 - r_i^2} \right) - r_{j-1} \left(\sqrt{r_{j-1}^2 - r_{i-1}^2} - \sqrt{r_{j-1}^2 - r_i^2} \right), & \text{если } i > 1, \quad j > i + 1 \end{cases} \quad (4)$$

($i, j = 1, \dots, M$).

Рассматривается функция (5) отклонений истинных количеств срезов включений от определяемых по теоретическому распределению

$$F = \sum_{l=1}^L \left(\sum_{i=1}^M \frac{(y_{l,i} - S_l \lambda_i^e)^2}{S_l \lambda_i^e} \right) \quad (5)$$

и осуществляется поиск значений N_j^e (входящих в λ_i^e), обеспечивающих минимум функции F .

Для этого функцию F преобразовали к виду [5]

$$F = S \sum_{i=1}^M \left(\frac{\tilde{y}_i^2}{\lambda_i^e} + \lambda_i^e \right) - 2\bar{y}_\Sigma, \quad (6)$$

где

$$S = \sum_{l=1}^L S_l, \quad \bar{y}_\Sigma = \sum_{l=1}^L \sum_{i=1}^M y_{l,i}, \quad \tilde{y}_i = \sqrt{\frac{1}{S} \sum_{l=1}^L \frac{y_{l,i}^2}{S_l}}. \quad (7)$$

Далее производные функции (6) по λ_i^e приравнивали нулю и находили решение полученной системы уравнений. Оно оказалось равным

$$\lambda_j^e = \tilde{y}_j, j = 1, \dots, M. \quad (8)$$

Найдя $\lambda_1^e, \dots, \lambda_M^e$, находили соответствующие значения N_1^e, \dots, N_M^e . Для этого вводили матрицы

$$\bar{\Phi} = \begin{pmatrix} \varphi_{11} & \dots & \varphi_{1M} \\ \dots & \dots & \dots \\ \varphi_{M1} & \dots & \varphi_{MM} \end{pmatrix}, \quad \bar{\Lambda}^e = \begin{pmatrix} \lambda_1^e \\ \dots \\ \lambda_M^e \end{pmatrix}, \quad \bar{N}^e = \begin{pmatrix} N_1^e \\ \dots \\ N_M^e \end{pmatrix}, \quad (9)$$

решали систему уравнений

$$\bar{\Lambda}^e = \bar{\Phi} \cdot \bar{N}^e, \quad (10)$$

и получали решение

$$\bar{N}^e = \bar{\Phi}^{-1} \bar{\Lambda}^e \quad (11)$$

Статья организована следующим образом. В разделе 1 выявлены причины возникновения такой ситуации, когда обычные методы дают явно неверные результаты. В разделе 2 приведен алгоритм вычислений, основанный на численных методах условной оптимизации. В разделе 3 приведены исходная таблица распределения срезов включений и таблица распределения включений, рассчитываемая по приведенному алгоритму. В заключении обсуждаются полученные результаты.

1. Случай недопустимого решения

Может случиться, что полученные по изложенной выше методике значения некоторых оценок N_j^e окажутся отрицательными. Что это значит?

Методика, которой мы воспользовались в [5], основана, в частности, на принципе максимального правдоподобия, который предполагает, что в эксперименте реализовалась наиболее вероятная ситуация.

При исследовании шлифов мы можем столкнуться и с мало вероятными случаями распределения срезов включений. Например, может оказаться, что в какой-то размерный диапазон срезов вообще не попало ни одно включение, хотя это и мало вероятно, если есть включения больших размеров.

В этом случае наиболее правильно было бы повторить наблюдения, сделав и исследовав новые шлифы из данного образца.

Для случая, когда, в силу каких-либо причин, сделать это затруднительно или невозможно, для полного использования информации, полученной из имеющихся шлифов, нами разработан следующий алгоритм.

2. Алгоритм расчета

Пусть по алгоритму, изложенному в [5], найдены минимальное значение F_{\min} остаточной дисперсии при значениях (N_1^e, \dots, N_M^e) , некоторые из которых являются отрицательными.

Шаг 1. Определяем исходную допустимую точку: вектор $\bar{N}^k (N_1^k, \dots, N_M^k)$, где

$$N_j^k = \begin{cases} N_j^e, & \text{если } N_j^e \geq 0, \\ 0, & \text{если } N_j^e < 0. \end{cases}$$

и $k = 0$. Задаемся требуемой точностью δ .

Шаг 2. Поиск следующей точки.

2.1. Находим орт градиента \bar{g}^k функции $F(\bar{N}^k) = F(N_1^k, \dots, N_M^k)$.

$$2.1.1. \lambda_i^k = \sum_{j=1}^M \varphi_{ij} N_j^k, \quad i = 1, \dots, M.$$

$$2.1.2. \frac{\partial F}{\partial N_j^k} = S \sum_{i=1}^M \left(1 - \left(\frac{\tilde{y}_j}{\lambda_j^k} \right)^2 \right) \varphi_{ij} \quad j = 1, \dots, M.$$

$$2.1.3. \overline{\text{grad}} F(\bar{N}^k) = \left(\frac{\partial F}{\partial N_1^k}, \dots, \frac{\partial F}{\partial N_M^k} \right).$$

2.1.4. Обнуляем координаты градиента, которые могут вывести из области допустимых значений: если $N_j^k = 0$ и координата

$$\left(\overline{\text{grad}} F(\bar{N}^k) \right)_j = \frac{\partial F}{\partial N_j^k} > 0, \text{ принимаем } \left(\overline{\text{grad}} F(\bar{N}^k) \right)_j = 0.$$

2.1.5. Находим модуль градиента функции $F(\bar{N}^k)$:

$$|\overline{\text{grad}F}| = \sqrt{\sum_{j=1}^M (\overline{\text{grad}F}(\bar{N}^k))_j^2}.$$

2.1.6. Находим орт градиента:

$$\bar{g}^k = \frac{1}{|\overline{\text{grad}F}(\bar{N}^k)|} \overline{\text{grad}F}(\bar{N}^k).$$

2.2. Находим шаг h .

2.2.1. Задаемся исходным шагом $h_0 = \max_j N_j^k$.

2.2.2. Для тех j , для которых $N_j^k > 0$ и $(\overline{\text{grad}F}(\bar{N}^k))_j > 0$, находим ми-

нимальное значение p величин $p_j = \frac{N_j^k}{(\overline{\text{grad}F}(\bar{N}^k))_j}$.

2.2.3. Находим $h = \min(h_0, p)$.

2.3. Находим новую точку $N^{k+1} (N_1^{k+1}, \dots, N_M^{k+1})$.

$$\bar{N}^{k+1} = \bar{N}^k - h\bar{g}^k$$

2.4. Находим $F^{k+1} = F(\bar{N}^{k+1})$.

2.4.1. $\lambda_i^{k+1} = \sum_{j=1}^M \varphi_{ij} N_j^{k+1}$.

2.4.2. $F^{k+1} = S \sum_{i=1}^M \left(\frac{\tilde{y}_i^2}{\lambda_i^{k+1}} + \lambda_i^{k+1} \right) - 2\bar{y}_\Sigma$.

2.5. Если $F^k \geq F^{k+1}$, то устанавливаем $h = \frac{h}{2}$ и переходим к п. 2.3.

Шаг 3. Если $\max_j (|N_j^{k+1} - N_j^k|) > \delta$, переопределяем $\bar{N}^k = \bar{N}^{k+1}$, $F^k = F^{k+1}$ и переходим к шагу 2.

Шаг 4. Находим количество включений в единице объема в каждом размерном интервале:

$$N_i = N_i^k \Delta r_i,$$

где Δr_i — длина i -го размерного интервала ($\Delta r_i = r_i - r_{i-1}$).

Шаг 5. СТОП.

3. Пример расчета

Ниже приведен пример расчета исходных данных, указанных в табл. 1.

Таблица 1

Исходные данные для расчета

№ шлифа	Обследованная площадь, мм ²	Верхние границы диапазонов размеров включений, мкм							
		1	2	3	5	7	10	15	20
1	100	67	66	23	10	11	9	6	4
2	100	82	75	22	13	1	5	5	3
3	100	76	49	12	20	9	7	4	0

Расчеты по приведенному алгоритму, дают результаты, приведенные в табл. 2.

Таблица 2

Наиболее вероятные значения истинных количеств включений

Колич., мм ³	Верхние границы диапазонов размеров включений, мкм							
	1	2	3	5	7	10	15	20
	1025,30	493,66	103,28	39,76	15,57	11,40	4,37	3,37

Заключение

Теоретически, зависимость числа срезов включений от их размера должна быть монотонно убывающей функцией. Это связано с тем, что каждое включение может дать срезы всех размеров: от размера своего диаметра до нуля. В срезы большого размера свой вклад вносят только большие включения, а в срезы меньших размеров — они и еще более мелкие частицы с диаметром не меньше данного. Чем меньшим размерам соответствует данный диапазон, тем большее число включений вносят в него свой вклад. Поэтому функция распределения срезов включений по размерам должна быть монотонно убывающей. Если включений много, то наблюдаемая на шлифе картина распределения срезов включений обязана подчиняться этому правилу. Однако если включений мало, то может оказаться, что ни одно включение не было срезано так, чтобы срез имел диаметр, соответствующий данному размерному диапазону.

Если к таким данным применить известные методики определения числа и размеров включений (давших эти срезы), то в некоторых диапазонах получатся отрицательные количества включений.

Конечно, лучше всего в этом случае переточить образец, сделать новые шлифы, увеличить площадь наблюдения и заново провести все расчеты. Но это не всегда возможно.

Нами предложен новый метод расчета распределения включений по размерам на основе информации о количествах и размерах срезов включений плоскостью шлифа, пригодный даже для таких случаев.

Хотя метод и был разработан для включений сферической формы, он может быть обобщен на частицы в виде эллипсоидов и прямоугольных параллелепипедов.

Получение исходных данных и расчеты могут быть легко реализованы на современных микроскопах, обладающих мощными визуальными и вычислительными средствами.

Литература

1. Lipiński T., Wach A. Size of Non-Metallic Inclusions High-Grade Medium Carbon Steel // Archives of Foundry Engineering. 2012. Vol. 14, No. 4. P. 55–60.

2. Lambrighs K., Verpoest L., et al. Influence of Non-Metallic Inclusions on the Fatigue properties of Heavily Cold Drawn Steel Wires // *Procedia Engineering*. 2010. Vol. 2. Iss. 1. P. 173–181. DOI: 10.1016/j.proeng.2010.03.019
3. Zeng D., Tian G., et al. Fatigue Strength Prediction of Drilling Materials Based on the Maximum Non-Metallic Inclusion Size // *Journal of Materials Engineering and Performance*. 2015. Vol. 24, Iss. 12. P. 4664–4672. DOI: 10.1007/s11665-015-1753-1
4. Рощин В.Е., Рощин А.В. *Электрометаллургия и металлургия стали*. Челябинск: Издательский центр ЮУрГУ, 2013. 571 с.
5. Дрозин А.Д. Метод обработки информации о неметаллических включениях, получаемой при обследовании микрошлифа готовой стали // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2017. Т. 6, № 4. (См. настоящий выпуск).
6. *Численные методы условной оптимизации*. Под ред. Ф. Галла, У. Мюрреля. М.: Мир, 297 с.

Дрозин Александр Дмитриевич, д.т.н., профессор, Центр элитного образования, директор, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Наталья Михайловна Япарова, к.ф.-м.н., доцент, зав. кафедрой, кафедра вычислительной математики и высокопроизводительных вычислений, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse170402

**THE METHOD OF PROCESSING INFORMATION
OBTAINED FROM THE INVESTIGATION
OF THE STEEL SAMPLE SURFACE
IN THE CASE OF UNREPRESENTATIVE DATA**

© 2017 A.D. Drozin, N.M. Yaparova

South Ural State University

(76, Lenin Avenue, Chelyabinsk 454080),

E-mail: drozinad@susu.ru, iaparovnm@susu.ru

Received: 23.10.2017

To study nonmetallic inclusions — harmful impurities formed during melting and crystallization of steel, a polished section plain is prepared from the sample of the metal being studied. The sections of the inclusions by the section plain are examined by a microscope. The possible sizes of the inclusion sections are divided into intervals and the number of the inclusion sections falls into each such interval is calculated. The dependence of the number of the inclusion sections on the size must be a monotonically decreasing function. However, the researcher may encounter a case where this is not performed. For example, in some size ranges, there are no sections at all (although there are sections in the higher ranges). If such data are applied known techniques for determining the numbers and sizes of inclusions (which formed these sections), then in some ranges negative amounts of inclusions will be obtained. This can happen when the numbers of the inclusions are small and the unlikely case of a cut is realized. In this paper, a technique is proposed that allows, despite this, with some error, to estimate the true quantities and the sizes of the nonmetallic inclusions in the metal volume. A numerical method of conditional optimization is applied.

Keywords: information processing, crystallography, nonmetallic inclusions, stereology, optimization.

FOR CITATION

Drozin A.D., Yaparova N.M. The Method of Processing Information Obtained From the Investigation of the Steel Sample Surface in the Case of Unrepresentative Data. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. vol. 6, no. 4. pp. 20–27. (in Russian) DOI: 10.14529/cmse170402.

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. Lipiński T., Wach A. Size of Non-Metallic Inclusions High-Grade Medium Carbon Steel. Archives of Foundry Engineering. 2012. vol. 14, no. 4. pp. 55–60.
2. Lambrighs K., Verpoest L., et al. Influence of Non-Metallic Inclusions on the Fatigue Properties of Heavily Cold Drawn Steel Wires. Procedia Engineering. 2010. vol. 2, iss. 1. pp. 173–181. DOI: 10.1016/j.proeng.2010.03.019
3. Zeng D., Tian G., et al. Fatigue Strength Prediction of Drilling Materials Based on the Maximum Non-Metallic Inclusion Size. Journal of Materials Engineering and Performance. 2015. vol 24, iss. 12. pp. 4664–4672. DOI: 10.1007/s11665-015-1753-1
4. Roshchin, V.E., Roshchin, A.V. *Elektrometallurgiya i metallurgiya stali* [Electrometallurgy and Metallurgy of Steel]. Chelyabinsk. Publishing Center of SUSU. 2013. 571 p.
5. Drozin A.D. The Method of Processing Information about Nonmetallic Inclusions Obtained from the Investigation of the Surface of Finished Steel Samples. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika*. [Bulletin of South Ural State University. Series: Computational Mathematics and Software Engineering]. 2017. vol. 6, no. 4. (in Russian) (Cf. current issue).
6. Numerical Methods for Constrained Optimization. P. E. Gill and W. Murray. Academic Press, London. 1974. 303 p.

МЕТОДЫ И СРЕДСТВА ОРГАНИЗАЦИИ ГЛОБАЛЬНОЙ ОЧЕРЕДИ ЗАДАНИЙ В ТЕРРИТОРИАЛЬНО РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ*

© 2017 А.В Баранов, А.И Тихомиров

*Межведомственный суперкомпьютерный центр Российской академии наук – филиал
Федерального государственного учреждения «Федеральный научный центр
Научно-исследовательский институт системных исследований*

Российской академии наук»

(119334 Москва, Ленинский пр-т., д. 32а)

E-mail: tema4277@rambler.ru

Поступила в редакцию: 01.09.2017

В статье рассмотрена модель территориально распределенной вычислительной системы (ТРС), состоящей из объединенных каналами связи высокопроизводительных вычислительных установок. Вычислительные установки из состава ТРС представляют собой высокопроизводительные кластеры, различающиеся по архитектуре и производительности. Объединяющие их каналы связи имеют разные надежность и пропускную способность. Особенностью рассматриваемой модели является децентрализованная схема управления заданиями. Подобная схема подразумевает, что любая вычислительная установка в любой момент времени может выйти из состава ТРС по причине своей неисправности или неисправности канала связи. Устранение неисправности означает динамическое подключение вычислительной установки к ТРС. В этих условиях в ТРС организуется глобальная очередь заданий с абсолютными приоритетами, из которой задания распределяются по свободным ресурсам вычислительных установок. Абсолютные приоритеты предполагают вытеснение с выполнения низкоприоритетного задания поступившим в очередь высокоприоритетным заданием. Для формирования и хранения глобальной очереди заданий в условиях динамически изменяющегося состава ТРС необходима надежная распределенная информационная система (РИС). В качестве основы для ее построения авторами рассмотрен ряд известных распределенных СУБД. В статье сформулированы требования к РИС, проведен сравнительный анализ и сделан выбор решения, удовлетворяющего требованиям, рассмотрен разработанный авторами макет ТРС с децентрализованной схемой диспетчеризации заданий.

Ключевые слова: территориально распределенная система, распределенная информационная система, абсолютные приоритеты вычислительных заданий.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Баранов А.В., Тихомиров А.И. Методы и средства организации глобальной очереди заданий в территориально распределенной вычислительной системе // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 28–42. DOI: 10.14529/cmse170403.

Введение

С целью повышения производительности и надежности расчетов отдельные высокопроизводительные *вычислительные установки* (ВУ) нередко объединяются в *территориально распределенные вычислительные системы* (ТРС). Наибольшее распространение получила многоуровневая форма организации ТРС, когда в качестве ВУ используются высокопроизводительные *вычислительные кластеры* [1]. Для управления

*Статья рекомендована к публикации программным комитетом Международной конференции «Суперкомпьютерные дни в России – 2017».

отдельной ВУ применяется *локальная система управления ресурсами* (ЛСУР), действия всех ВУ координирует *глобальная система управления ресурсами* (ГСУР). После включения ВУ в состав ТРС вычислительные ресурсы этой ВУ не отчуждаются от их владельца и продолжают использоваться для выполнения локальных заданий, которые образуют *локальный поток заданий* [2]. После включения ВУ в состав ТРС совместно с заданиями локального на вычислительные ресурсы ВУ начинают поступать задания с других ВУ — задания из *глобального потока заданий*.

Очередность выполнения заданий устанавливается с использованием приоритетов (абсолютных или относительных). Для организации в территориально распределенной системе обработки вычислительных заданий в соответствии с приоритетами, требуется ведение *единой глобальной очереди заданий*, в которой все поступившие задания ранжируются, а затем распределяются строго в соответствии с приоритетом. Для ведения глобальной очереди заданий требуется наличие единой для всей ТРС *информационной системы*, исследование методов и средств построения которой и является целью настоящей работы.

Статья организована следующим образом. Раздел 1 посвящен исследуемой авторами модели ТРС, ее отличительным свойствам и характеристикам. В разделе 2 рассматриваются различные варианты организации глобальной системы управления ТРС, делается выбор в пользу децентрализованной схемы диспетчеризации с единой информационной системой. Организации такой информационной системы посвящен раздел 3. В разделе 4 обосновывается выбор распределенной СУБД в качестве основы информационной системы ТРС. Практическая реализация выбранного решения в виде макета ТРС рассмотрена в разделе 5. В заключении подводятся итоги и намечаются перспективы развития дальнейших исследований.

1. Модель территориально распределенной системы

В исследуемой авторами модели ТРС каждая ВУ представляется высокопроизводительным вычислительным кластером. Логически каждая ВУ может быть разделена на четыре взаимосвязанные подсистемы: *серверную, исполнительную, управляющую и подсистему доступа*. Каждая подсистема физически представляется набором из нескольких рабочих станций (вычислительных модулей), на которых установлены программные компоненты, соответствующие назначению подсистемы.

Серверная подсистема представлена серверами баз данных и используется для хранения данных, необходимых для функционирования ВУ: локальной и глобальной очередей заданий, исходных данных заданий.

Исполнительная подсистема представляет собой решающее поле кластера — набор вычислительных модулей, объединенных высокоскоростной низколатентной сетью.

Управляющая подсистема применяется для управления исполнительной подсистемой с использованием ЛСУР и диспетчеров заданий.

Подсистема доступа обеспечивает контроль поступающего в ВУ сетевого трафика и служит точкой входа в ВУ для удаленных пользователей, а также подсистем других ВУ из состава ТРС.

В исследуемой модели (рис. 1) допускается, что различные ВУ могут содержать разное число вычислительных модулей разной производительности и архитектуры, т.е. для ТРС характерна гетерогенность как для системы в целом, так и внутри отдельной ВУ.

Управление на уровне локальных ресурсов осуществляет ЛСУР, основными функциями которой являются ведение локальной очереди заданий, запуск и контроль процесса выполнения заданий на вычислительных ресурсах ВУ. Примерами ЛСУР ВУ могут служить известные системы пакетной обработки, такие как PBS, SLURM, Moab или отечественная Система управления прохождением параллельных заданий (СУППЗ) [3].

Единицей обработки информации в ТРС является *вычислительное задание*, под которым понимается набор, включающий входные данные, программу их обработки и паспорт задания. *Паспорт задания* – специальный объект, описывающий *ресурсный запрос задания*: количество процессоров (ядер), объем оперативной памяти и дискового пространства, *приоритет задания* и др. Различные задания от разных пользователей образуют поток заданий. В каждой ВУ всегда присутствует *локальный поток заданий*, задания которого допускают обработку только на локальных вычислительных ресурсах ВУ.

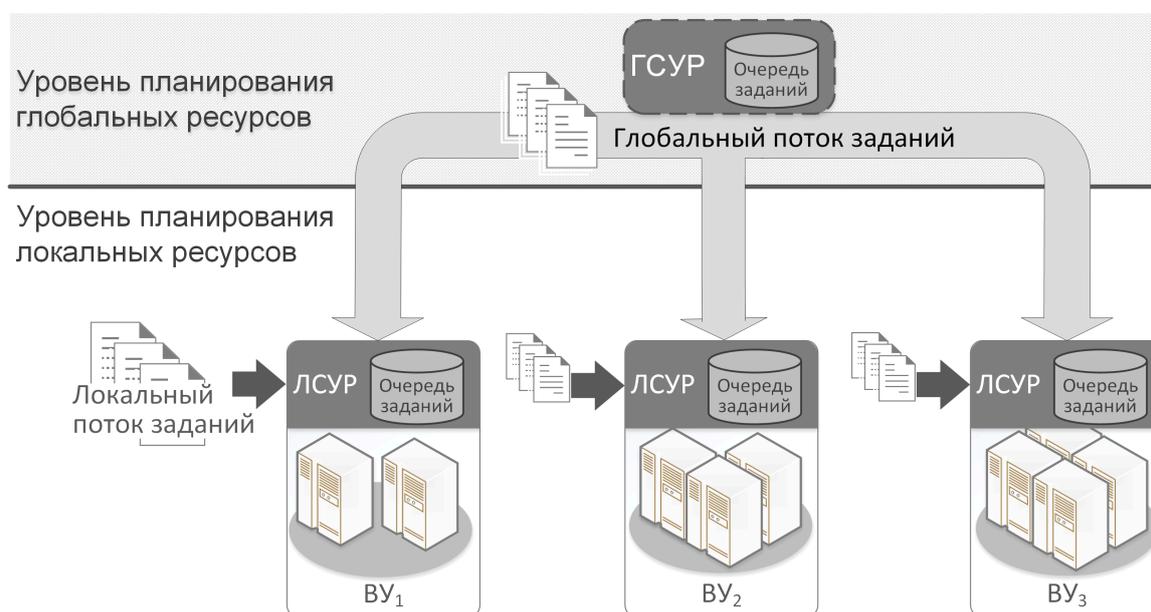


Рис. 1. Модель территориально распределенной вычислительной системы

Управление на уровне планирования глобальных ресурсов ТРС осуществляет ГСУР, основной функцией которой является планирование заданий глобального потока. В отличие от локального, задания глобального потока допускают обработку на вычислительных ресурсах любой ВУ ТРС и распределяются в ВУ, время обработки в которой будет минимальным. Важно отметить, что ГСУР не планирует задания на локальные ресурсы ВУ, а только выбирает т.н. *целевую ВУ* для размещения задания. Размещенное задание глобального потока поступает под управление ЛСУР целевой ВУ и планируется наряду с заданиями локального потока.

Таким образом, на вычислительные ресурсы ВУ поступает два потока задания: локальный и глобальный. Заданиям обоих потоков доступны все вычислительные ресурсы [2]. Очередность предоставления вычислительных ресурсов устанавливается в соответствии с приоритетами заданий. В исследуемой модели ТРС предусматриваются абсолютные приоритеты, отличие которых от относительных состоит в использовании механизма вытеснения высокоприоритетными заданиями менее приоритетных [4].

Существенной характеристикой исследуемой модели ТРС является ее динамически изменяющийся состав. Модель допускает в любой момент времени как включение в состав ТРС новых ВУ, так и исключение из состава ТРС отдельных ВУ. Выключение из состава ТРС (включение в состав ТРС) некоторой ВУ может быть связано как с неисправностью (возвращением в строй) самой ВУ, так и с неисправностью (возвращением в строй) коммуникационных каналов.

2. Варианты организации глобальной системы управления ТРС

Для организации ГСУР могут быть использованы *централизованная, иерархическая, децентрализованная* архитектуры [5]. При централизованной архитектуре в качестве управляющего центра назначается некоторая ВУ, на которую возлагается функция распределения вычислительных заданий глобального потока по доступным ВУ ТРС. Существенным недостатком централизованной архитектуры, ограничивающим возможность ее применения в ТРС, является наличие в системе единой точки отказа – центра управления. Централизованная диспетчеризация применяется в системах [6,7].

При иерархической архитектуре в системе выделяется несколько логических уровней, а также несколько центров управления, каждый из которых ассоциируется с одним из уровней. Между центрами управления разных уровней устанавливается иерархическая зависимость, то есть каждый центр управляет центрами нижележащего уровня. Логически иерархическая структура представляется в виде дерева, промежуточными вершинами и корнем которого являются центры управления. При этом у каждой вершины помимо связи с вышележащей вершиной может существовать связь с центрами управления одного логического уровня. В случае неисправности одного из центров управления исходное дерево разделяется на одно или более поддеревьев, каждое из которых начинает функционировать автономно. Наличие связей между центрами управления одного уровня позволяет путем перестройки восстановить исходное дерево без восстановления неисправного центра. Механизм перестройки дерева основан на назначении центрам новых зависимостей (могут изменяться уровни). Необходимость перестройки дерева в случае неисправности или недоступности какого-либо центра управления является главным недостатком иерархических систем.

С точки зрения авторов, наибольший интерес для дальнейших исследований представляет децентрализованная архитектура ГСУР, обеспечивающая надежность и масштабируемость ТРС. При децентрализованной архитектуре ГСУР в ТРС управление осуществляется *коллективом равноправных диспетчеров*, которые располагаются локально на каждой ВУ ТРС. Неисправность одного из диспетчеров никак не сказывается на функционировании остальных. При этом решения о распределении каждого задания глобальной очереди принимаются согласованно. Децентрализованный подход применяется, например, в системе PAUA [8].

Согласованность решения о распределении задания достигается коллективом диспетчеров либо в результате их *непосредственного взаимодействия*, либо *взаимодействия с использованием единой информационной системы* [9]. При непосредственном взаимодействии диспетчеры напрямую обмениваются вычислительными заданиями друг с другом, в этом случае распределение заданий основывается на *жадном алгоритме* [10]. Жадный алгоритм предполагает, что все задания глобального потока

заданий распределяется во все ВУ ТРС в момент их поступления. При таком подходе в ТРС отсутствует информация о загруженности ВУ, что приводит к невозможности организации приоритетной обработки заданий. Кроме этого, возникает опасность длительного пребывания заданий в локальной очереди ВУ после их распределения по причине большого числа заданий с высоким приоритетом, поступивших из локального потока. Этот недостаток может быть устранен путем реализации в ГСУР механизма перераспределения заданий, что в свою очередь является отдельной сложной задачей [11].

Второй способ взаимодействия диспетчеров — взаимодействие с использованием *единой информационной системы*. Способ предполагает распределение заданий *ленивым алгоритмом планирования*. В этом случае все задания глобального потока поступают в информационную систему, в которой формируется и хранится *глобальная очередь заданий*. Задания распределяются в ВУ только тогда, когда для их выполнения доступны необходимые вычислительные ресурсы. Контроль доступности вычислительных ресурсов ВУ обеспечивает соответствующий диспетчер. Наличие в системе глобальной очереди позволяет распределять задания с учетом их приоритетов, т.е. сначала диспетчеры обрабатывают задания с максимальным приоритетом, а затем — менее приоритетные.

3. Организация информационной системы ТРС

Рассмотренные авторами в предыдущих работах [4] ленивые алгоритмы планирования позволили определить следующий сценарий работы диспетчеров с глобальной очередью заданий.

1. Добавление заданий в глобальную очередь. Все новые задания глобального потока диспетчер помещает в информационную систему. При этом исходные данные задания остаются на ВУ, поместившей задание в глобальную очередь, и передаются на целевую ВУ только после назначения туда задания.
2. Изменение параметров задания в глобальной очереди. Пользователь, поместивший задание в информационную систему, может изменить его параметры, например, повысить приоритет. Задание может быть изменено и диспетчером, например, в случае если диспетчер изменяет статус выполнения задания.
3. Поиск вычислительных заданий. Диспетчер ВУ, вычислительные ресурсы которой свободны и готовы принять новое задание, осуществляет соответствующий запрос в информационную систему для получения нового задания. В поисковом запросе указываются критерии поиска: минимальный приоритет задания, требуемое число вычислительных модулей и др. Для обработки таких запросов требуется возможность анализа всех параметров всех заданий глобальной очереди.
4. Извлечение распределенного задания из глобальной очереди. Задание, удовлетворяющее критериям поискового запроса, извлекается диспетчером из глобальной очереди, но при этом информация о задании остается в информационной системе для отслеживания состояния его выполнения.

Приведенный сценарий позволяет сформулировать следующие требования к информационной системе для организации глобальной очереди заданий.

1. Доступность и масштабируемость. Информационная система должна всегда быть доступна для всех диспетчеров, а также обеспечивать масштабируемость ТРС.
2. Соответствие децентрализованной архитектуре ГСУР.

3. Надежность хранения глобальной очереди. В информационной очереди глобальная очередь должна сохраняться независимо от динамически изменяющегося состава ТРС. Неисправность отдельных компонентов ТРС не должна приводить к потере заданий, хранящихся в глобальной очереди.
4. Организация поиска данных по критериям (по приоритетам). Для извлечения заданий из глобальной очереди требуется выполнение сложных поисковых запросов, предусматривающих анализ отдельных полей заданий (например, приоритетов заданий) или поиск с использованием фильтров.
5. Возможность изменения параметров и извлечения заданий из очереди. Требуется обеспечить возможность изменения параметров заданий в глобальной очереди таким образом, чтобы извлеченные определенным диспетчером задания не фигурировали в результатах поиска других диспетчеров.

Рассматривая варианты организации информационной системы ТРС, следует учитывать, что, как и в случае с ГСУР, возможно применение одного из трех видов архитектуры: централизованной, иерархической и децентрализованной. Преимущества и недостатки каждого из трех подходов были рассмотрены выше при выборе архитектуры ГСУР. Здесь отметим особенности и примеры каждой архитектур применительно к организации информационной системы ТРС.

Централизованная архитектура информационной системы предполагает выделение отдельного компонента, отвечающего за хранение глобальной очереди заданий и обработку поступающих от диспетчеров заданий запросов. Основой централизованной информационной системы может выступать любая из известных СУБД: PostgreSQL, Oracle и др. Достоинствами такого решения является простота организации и контроля целостности заданий, хранящихся в глобальной очереди заданий. Недостаток, присущий всем централизованным архитектурам, заключается в наличии единой точки отказа.

В случае иерархической организации информационная система распределяется между несколькими серверами, каждый из которых ассоциируется с определенным логическим уровнем. Высший уровень содержит агрегированные сведения о нижележащих уровнях. При невозможности выполнить запрос, поступивший на один из уровней, происходит обращение к более высоким уровням системы. Примером иерархической организации является система MDS (Metacomputing Directory Service), основанная на распределенной службе каталогов LDAP (Lightweight Directory Access Protocol) [12]. MDS применялась в составе набора инструментов Globus Toolkit [13] вплоть до версии 4.0.

Децентрализованная архитектура информационной системы применима только при децентрализованной схеме диспетчеризации вычислительных заданий. В этом случае в ТРС отсутствует единая информационная система, вместо этого у каждого диспетчера имеется собственное хранилище. Децентрализованное решение обеспечивает масштабируемость и надежность ТРС, однако не позволяет распределять задания в соответствии с единым для всей ТРС критерием очередности распределения заданий, определяемой их приоритетами. Для выполнения последнего требования необходимо, чтобы хранилище данных каждого диспетчера было согласовано с хранилищами остальных диспетчеров. Добиться этого можно за счет использования распределенного подхода, при котором информационная система представляется коллективом равноправных, взаимодействующих подсистем хранения. В этом случае достигается сочетание преимуществ централизованного и децентрализованного подходов.

Применение распределенной информационной системы для хранения данных известно на примере пиринговой сети BitTorrent и используемой в ней DHT (Distributed Hash Table) [14]. Идея состоит в том, что каждый диспетчер хранит часть общих данных системы, данные при этом физически распределены по всем диспетчерам ТРС, однако логически они являются одним целым. Для хранения данных в состав каждого диспетчера входит локальный сервер данных. Серверы всех диспетчеров управляются распределенной СУБД.

Применяемые в современных распределенных СУБД механизмы сегментирования и дублирования позволяют при отказе одной из ВУ восстановить часть ее данных, тем самым не допустив потери данных. Для этого все данные, хранящиеся в распределенной базе данных, разделяются на один или несколько сегментов. Разные сегменты помещаются на разные серверы, при этом для каждого сегмента (т.н. первичного) может быть создан один или несколько дублирующих сегментов. Дублирующие сегменты хранятся отдельно от первичных, на других серверах. Изменения данных сначала фиксируются в первичном сегменте, а затем во всех дублирующих сегментах. В случае неисправности одного из серверов, хранящего дублирующий сегмент, система продолжает корректно работать, при восстановлении неисправного сервера он заново получает актуальную версию дублирующего сегмента. Неисправность сервера, хранящего первичный сегмент, приведет к восстановлению одного из дублирующих сегментов в качестве первичного. Выбор нового первичного сегмента осуществляется путем голосования. После восстановления ранее отказавшего сервера с первичным сегментом и копирования на него актуальной версии данных он продолжает функционировать в качестве дублирующего сегмента.

4. Выбор распределенной СУБД в качестве основы информационной системы ТРС

Большинство существующих распределенных СУБД относятся к типу noSQL-решений, для которых характерна, прежде всего, нереляционная модель хранения данных. Вместо реляционной (табличной) модели представления и хранения данных в noSQL-решениях применяются документо-ориентированные, ассоциативные, столбцово-ориентированные или векторные модели. Решения noSQL не поддерживают транзакции, ограничиваясь лишь поддержкой атомарности операций записи и чтений. Из-за нереляционной модели хранения и представления данных большинство существующих решений noSQL не предполагают использование языка запросов SQL. Вместо SQL используется собственный язык запросов, который в большинстве случаев лишен гибкости, свойственной SQL.

Для организации распределенной информационной системы авторами были рассмотрены следующие известные решения: ClickHouse [15], Elasticsearch [16], Redis [17].

Распределенная СУБД *Clickhouse* была разработана для решения аналитических задач, например, расчетов статистических показателей для большого объема данных и др. Особенности области применения являются большой объем данных, высокая скорость их поступления, отсутствие изменений ранее записанных данных. В виду большого объема данных и выполнения в подавляющем большинстве только агрегационных операций, как правило, к хранению аналитических данных не предъявляются требования надежности и целостности.

Основой распределенной СУБД *Elasticsearch* являются серверы полнотекстового поиска Lucene. Все данные в *Elasticsearch* хранятся в индексе в полуструктурированном

виде – *документе*. Использование документной модели хранения и представления данных позволяют сделать решение универсальным. Использование встроенного языка запросов QueryDSL совместно с инструментами полнотекстового поиска, предоставляемыми серверами Lucene, позволяют выполнять сложные запросы к хранящимся документам.

Распределенная СУБД *Redis* использует наиболее простую модель данных — ассоциативную. Все добавляемые в *Redis* данные ассоциируются с идентификатором, который в дальнейшем используется для выполнения запросов к хранящимся данным. Распределенная СУБД *Redis* часто используется для организации промежуточного хранения данных (кэширования данных) с целью снижения нагрузки на основное хранилище данных.

Результаты проведенного авторами сравнительного анализа существующих распределенных СУБД отражены в приведенной ниже табл.

Таблица

Сравнительный анализ распределенных СУБД

Критерий сравнения	Clickhouse	Elasticsearch	Redis
Доступность информационной системы всем ВУ	+	+	+
Соответствие децентрализованной архитектуре ГСУР	+	+	+
Надежность сохранения глобальной очереди заданий	— асинхронное дублирование данных	+	+
Возможность изменения и извлечения заданий из очереди	— данные не изменяются, а удаление возможно только за месяц	+	+
Организация поиска данных по критериям (по приоритетам)	+	+	—
	диалект SQL	QueryDSL + инструменты полнотекстового поиска	обращение только по идентификатору

Табл. показывает, что наиболее полно всем предъявленным требованиям удовлетворяет распределенная СУБД *Elasticsearch* и используемая в ней документо-ориентированная модель хранения и представления данных. Каждый поступающий паспорт задания представляется отдельным документом, поля которого могут быть проанализированы при выполнении поискового запроса.

5. Макет ТРС с распределенной информационной системой

Для экспериментальной проверки применимости выбранных решений авторами был разработан модельный макет ТРС с распределенной информационной системой, основанной на распределенной СУБД Elasticsearch. Распространенной практикой создания подобных модельных макетов является использование в качестве моделируемых серверов (рабочих станций, вычислительных модулей) виртуальных машин. При этом каждая подсистема ВУ — управляющая, исполнительная, серверная и подсистема доступа, моделируется одной или несколькими виртуальными машинами. Большое число одновременно выполняющихся виртуальных машин ведет к интенсивному расходу ресурсов физического сервера, используемого для макета. Авторы при построении макета применили альтернативный подход — контейнерную виртуализацию с использованием платформы Docker [18], требующей значительно меньшего объема аппаратных ресурсов по сравнению с гипервизорной виртуализацией.

Использование контейнерной виртуализации предполагает, что каждая рабочая станция любой подсистемы ВУ запускается в безопасной изолированной среде — контейнере. Для создания контейнера подготавливается образ — шаблон инфраструктуры, настроенный к запуску определенного программного обеспечения (СУБД, ЛСУР и др.).

Помимо снижения накладных расходов использование контейнерной виртуализации при подготовке макета ТРС позволяет добиться высокой скорости запуска как отдельного контейнера, так и всей многоконтейнерной системы, а также осуществлять контроль состояния ее выполнения. Контроль состояния выполнения предполагает, что при отказе одного из контейнеров ядро Docker может перезапустить неисправный контейнер.

Подготовленный модельный макет ТРС состоит из трех вычислительных установок, на каждой из которых располагается один сервер Elasticsearch. База данных разделена на 6 сегментов, каждый сегмент имеет два дубликата: первичный и вторичный (рис. 2).

С помощью подготовленного макета авторы провели исследования надежности и производительности распределенной информационной системы, основанной на Elasticsearch.

Исследование надежности осуществлялось следующим образом. В информационную систему был направлен поток заданий, состоящий из $M = 5000$ заданий. Размер каждого задания не превышал 15 Кбайт. Интервалы времени между поступлением заданий в очередь имели пуассоновское распределение с интенсивностью $\lambda = 20$. На фоне обработки поступающего входного потока заявок из состава ТРС исключались отдельные ВУ вместе с серверами баз данных, используемыми для хранения глобальной очереди. Через установленный интервал времени, достаточный для восстановления дублирующих копий сегментов неисправного сервера на оставшихся серверах, ранее исключенная ВУ вновь добавлялась в состав ТРС. При этом на основе анализа служебных данных, хранящихся в базе данных, авторами контролировалось автоматическое размещение части сегментов с других серверов ВУ на серверах вновь добавленной ВУ. Операция добавления и исключения ВУ, имитирующая систематические отказы, повторялась постоянно на фоне поступающего потока заданий. Выбор исключаемой ВУ и моментов времени исключения ВУ из состава ТРС осуществлялся случайным образом. После окончания поступления заданий в информационную систему на одном из серверов ВУ выполнялся агрегирующий запрос, определяющий количество документов, хранящихся (добавленных в результате эксперимента) в базе данных. Результат выполнения запроса сравнивался с исходным

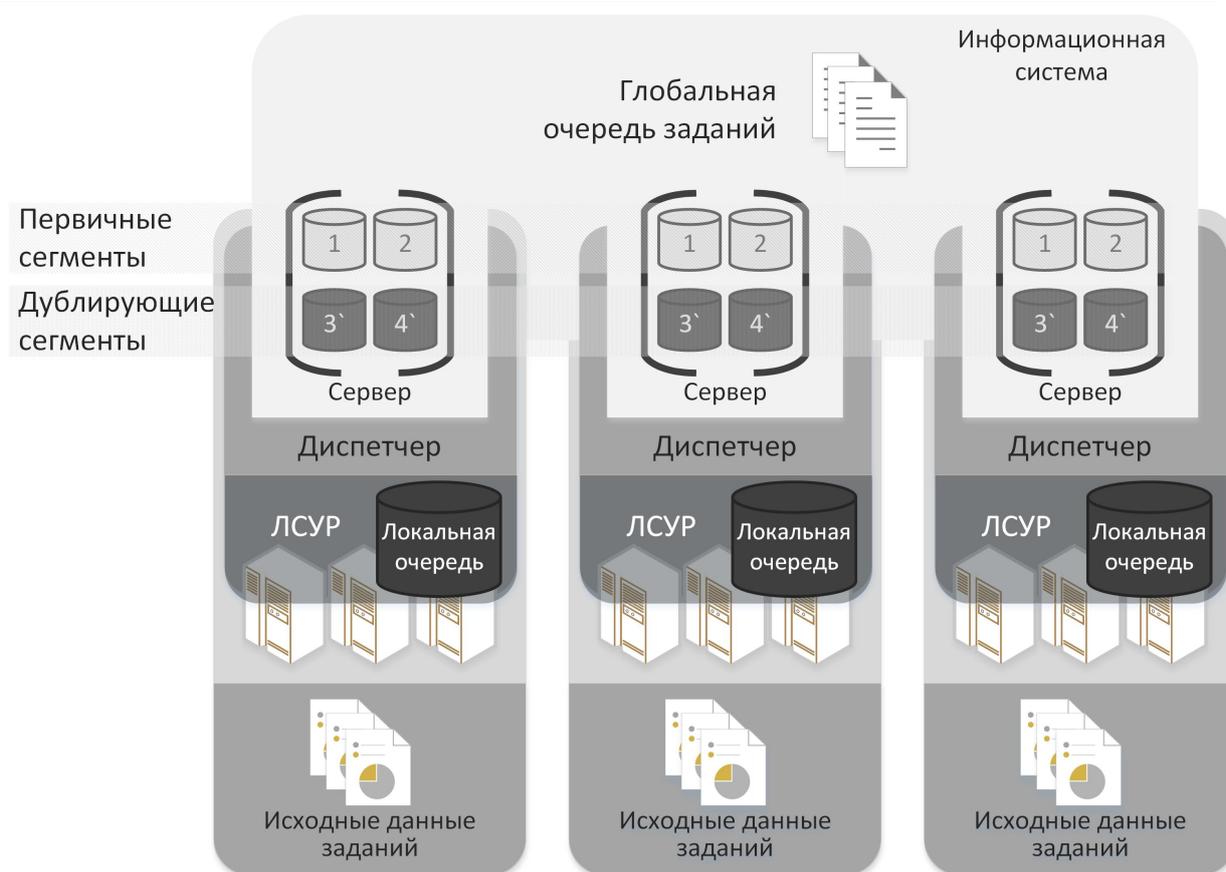


Рис. 2. Макет ТРС с использованием распределенной информационной системы на основе Elasticsearch

числом заданий (M). В результате проведения эксперимента авторами было установлено, что в конфигурации из трех серверов выход из строя одного из них не приводит к потере информации о заданиях глобальной очереди. Вывод из строя двух серверов приводит к потере информации о 20%-30% заданий. Дальнейшие эксперименты показали, что доля потерянных заданий может быть снижена за счет увеличения числа дубликатов каждого из сегментов с некоторым снижением производительности распределенной СУБД. Кроме этого, было обнаружено, что на надежности хранения данных отрицательно сказывается включение по умолчанию механизма кэширования поступающих в Elasticsearch данных. Все поступающие данные сохраняются в оперативной памяти сервера и не записываются на диск, пока объем поступивших данных не достигнет определенного значения, заданного в конфигурации сервера. Отказ сервера приводит к потере всех данных, не записанных на диск. Запрещение кэширования данных увеличивает надежность их хранения, но также приводит к некоторым потерям производительности.

Для оценки того, насколько существенными для информационной системы ТРС будут потери производительности при увеличении числа дубликатов и запрете кэширования, авторами был проанализирован опыт эксплуатации Распределенной инфраструктуры суперкомпьютерных приложений (РИСП) [1], объединяющей в ТРС вычислительные ресурсы филиалов МСЦ РАН нескольких городов России. Было установлено, что приемлемая скорость работы с очередью заданий составляет 10-15 документов (паспортов заданий) в секунду. Эксперимент по оценке производительности проводился в несколько этапов. На каждом из этапов эксперимента в информационную систему поступал поток

документов $M = 1000$, при этом интервалы времени между поступлением документов задавались согласно экспоненциальному распределению с интенсивностью (плотностью) $\lambda = [10, 50, 100]$ соответственно. В результате было установлено, что разработанная авторами распределенная информационная система способна обеспечить скорость 500-600 документов в секунду, что заведомо превышает необходимый уровень производительности.

Результаты проведенных экспериментов по оценке надежности и производительности разработанной информационной системы показали возможность ее использования в ТРС.

Заключение

При организации глобальной очереди заданий в территориально распределенной вычислительной системе неизбежно возникает противоречивый выбор между централизованной и децентрализованной схемами управления. С одной стороны, непрерывные изменения состава ТРС требуют децентрализованной схемы, как единственной способной обеспечить целостность распределенной системы. С другой стороны, глобальная очередь заданий является единой для всей ТРС, а абсолютные приоритеты заданий требуют оперативного распределения по вычислительным ресурсам системы. Для организации глобальной очереди необходима информационная система, являющаяся логическим центром ТРС.

Исследование, проведенное авторами в настоящей работе, показало, что информационная система, организованная на основе распределенной базы данных, обладает преимуществами централизованной и децентрализованной архитектур. В статье авторы рассмотрели распространенные модели распределенных СУБД и пришли к выводу, что для сценариев работы с глобальной очередью заданий в ТРС наиболее подходящей является документо-ориентированная модель хранения и представления данных, реализованная, в частности, в распределенной СУБД Elasticsearch. С целью экспериментальной проверки полученного вывода авторами был разработан макет ТРС, в рамках которого на базе СУБД Elasticsearch была реализована информационная система для формирования и хранения глобальной очереди заданий. Проведенные эксперименты показали как общую работоспособность системы, так и приемлемый уровень ее надежности и производительности.

Полученные авторами результаты относятся к начальному этапу построения ТРС суперкомпьютерных центров коллективного пользования. Для успешного завершения проекта авторы считают необходимым решение следующих научно-технических задач:

- исследование и разработка методов и алгоритмов оптимального планирования заданий в ТРС;
- разработка подсистемы информационной безопасности глобальной очереди заданий;
- разработка интерфейсов подключения к глобальной очереди заданий для различных локальных систем управления ресурсами.

В решении указанных задач авторы видят перспективы развития представленной работы.

Работа выполнена в рамках государственного задания с использованием вычислительных ресурсов Межведомственного суперкомпьютерного центра РАН (МСЦ РАН).

Литература

1. Савин Г.И., Шабанов Б.М., Корнеев В.В., Телегин П.Н., Семенов Д.В., Киселев А.В., Кузнецов А.В., Вдовикин О.И., Аладышев О.С., Овсянников А.П. Создание распределенной инфраструктуры для суперкомпьютерных приложений // Программные продукты и системы. 2008. № 2. С. 2–7.
2. Корнеев В.В., Семенов Д.В., Телегин П.Н., Шабанов Б.М. Отказоустойчивое децентрализованное управление ресурсами грид // Известия вузов. Электроника. 2015. Т. 20, № 1. С. 83–90.
3. Баранов А.В., Киселёв А.В., Старичков В.В., Ионин Р.П., Ляховец Д.С. Сравнение систем пакетной обработки с точки зрения организации промышленного счета. Научный сервис в сети Интернет: поиск новых решений // Труды Международной суперкомпьютерной конференции (Новороссийск, 17-22 сентября 2012 г.). М.: Изд-во МГУ, 2012. С. 506–508.
4. Баранов А.В., Тихомиров А.И. Планирование заданий в территориально распределенной системе с абсолютными приоритетами // Вычислительные технологии. 2017. Т. 22, № S1. С. 4–12.
5. Березовский П.С., Коваленко В.Н. Состав и функции системы диспетчеризации заданий в грид с некластеризованными ресурсами // Препринты ИПМ им. М. В. Келдыша. 2007. № 67. С. 29.
6. WMS Architecture overview. URL: <http://egee-jra1-wm.mi.infn.it/egee-jra1-wm/wms.shtml> (дата обращения: 27.03.2017).
7. Internal Architecture 5.14. URL: http://www.gridway.org/doku.php?id=documentation:release_5.14:iashtml (дата обращения: 20.03.2017).
8. Cirne W., Brasileiro F., Costa L., Paranhos D., Santos-Neto E., Andrade N. Scheduling in Bag-of-Task Grids: PAUA Case // 16th Symposium on Computer Architecture and High Performance Computing. Oct. 2004. pp. 124–131. DOI: 10.1109/CAHPCC.2004.37.
9. Коваленко В.Н., Орлов А.В. Управление заданиями в распределенной среде и протокол резервирования ресурсов // Препринты ИПМ им. М. В. Келдыша. 2002. №1. С. 1–25.
10. Buncic P., Saiz P., Peters A.J. The AliEn System, Status and Perspectives // 2003 Conference for Computing in High-Energy and Nuclear Physics, La Jolla, CA, USA, 24-28 Mar 2003. P. MOAT004. URL: <http://www.slac.stanford.edu/econf/C0303241/proc/papers/MOAT004.PDF> (дата обращения: 20.03.2017).
11. Топорков В.В., Емельянов Д.М., Потехин П.А. Формирование и планирование пакетов заданий в распределенных вычислительных системах // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2015. Т. 4, № 2. С. 44–57. DOI: 10.14529/cmse150204.
12. Валиев М.К., Китаев Е.Л., Слепенков М.И. Служба директорий LDAP как инструментальное средство для создания распределенных информационных систем // Препринты ИПМ им. М.В. Келдыша. 2000. № 23. С. 1–22.
13. Kesselman C., Fitzgerald S., Foster I., Tuecke S., Smith W. A Directory Service for Configuring High-Performance Distributed Computations // 6th IEEE Symposium on High Performance Distributed Computing. 1997. pp. 365–375. DOI: 10.1109/HPDC.1997.626445.

14. Loewenstern A. Norberg A. DHT Protocol. 2008. URL: http://bittorrent.org/beps/bep_0005.html (дата обращения: 11.03.2017).
15. ClickHouse Reference Manual. 2015. URL: https://clickhouse.yandex/reference_en.html (дата обращения: 16.02.2017).
16. Elastic Stack and Product Documentation. 2016. URL: <https://www.elastic.co/guide/index.html> (дата обращения: 22.01.2017).
17. Programming with Redis. 2016. URL: <https://redis.io/documentation> (дата обращения: 12.02.2017).
18. A. Prasad. Announcing Docker Compose. 2015. URL: <https://blog.docker.com/2015/02/announcing-docker-compose/>(дата обращения: 26.02.2017).

Баранов Антон Викторович, к.т.н., доц., в.н.с., Межведомственный суперкомпьютерный центр Российской академии наук – филиал Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (Москва, Российская Федерация)

Тихомиров Артем Игоревич, стажер-исследователь, Межведомственный суперкомпьютерный центр Российской академии наук – филиал Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (Москва, Российская Федерация)

DOI: 10.14529/cmse170403

METHODS AND TOOLS FOR ORGANIZING THE GLOBAL JOB QUEUE IN THE GEOGRAPHICALLY DISTRIBUTED COMPUTING SYSTEM

© 2017 A.V. Baranov, A.I. Tikhomirov

Joint Supercomputer Center of the Russian Academy of Sciences - Branch of Federal State Institution «Scientific Research Institute for System Analysis of the Russian Academy of Sciences», (pr. Leninsky 32a, Moscow, 119334 Russia)

E-mail: tema4277@rambler.ru

Received: 01.09.2017

The geographically distributed computing infrastructure (DCI) considered in the paper includes high performance computing systems united by communication channels. Computing systems from the DCI are high-performance clusters differing in architecture and performance. Communication channels uniting clusters have different reliability and bandwidth. The considered model of DCI has a decentralized jobs management and dispatching scheme. This scheme implies that at any time malfunction of any computing cluster or a failure in the communication channel can cause cluster's leaving the DCI. Cluster's or channel's troubleshooting means dynamically connecting the cluster to the DCI. The global job queue is organized in this computing infrastructure. Computing jobs have absolute priorities, and high priority job can interrupt low priority running jobs. Jobs from the global queue allocate on idle resources of computing systems. Forming and storing global job queue in conditions of dynamically changing DCI composition needs the reliable information system. The authors reviewed some distributed DBMSs as the basis of this information system. The article outlines the requirements for a distributed information system. The authors conducted a comparative analysis and selected a solution that

satisfies the requirements, and designed prototype of the geographically distributed computing infrastructure with the decentralized scheme of jobs dispatching.

Keywords: grid, information system, absolute priorities.

FOR CITATION

Baranov A.V., Tikhomirov A.I. Methods and Tools for Organizing the Global Job Queue in the Geographically Distributed Computing System. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 4. pp. 28–42. DOI: 10.14529/cmse170403.

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. Savin G.I., Shabanov B.M., Korneev V.V., Telegin P.N., Semenov D.V., Kiselev A.V., Kuznecov A.V., Vdovikin O.I., Aladyshev O.S., Ovsjannikov A.P. Creation of Distributed Infrastructure for Supercomputer Applications. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 2, pp. 2–7. (in Russian).
2. Korneev V.V., Semenov D.V., Telegin P.N., Shabanov B.M. Resilient Decentralized GRID Resources Control. *Izvestija vysshih uchebnyh zavedenij. Jelektronika* [Proceedings of Universities. Electronics]. 2015, vol. 20, no. 1, pp. 83–90. (in Russian).
3. Baranov A.V., Kiselev A.V., Starichkov V.V., Ionin R.P., Lyakhovets D.S. Comparison of Workload Management Systems from the Point of View of Organizing an Industrial Computing. *Nauchnyj servis v seti Internet: poisk novyh reshenij: Trudy mezhdunarodnoy superkomp'yuternoy konferentsii (Novorossiysk, 17–22 Sentyabrya 2012)* [Scientific Services and Internet: Search for New Solutions: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 17-22, 2012)]. Moscow, Publishing of Lomonosov Moscow State University, 2012, pp. 506–508. (In Russian).
4. Baranov A.V., Tikhomirov A.I. Scheduling of Jobs in a Territorially Distributed Computing System with Absolute Priorities. *Vychislitel'nye tehnologii* [Computational Technologies]. 2017, vol. 22, no. S1, pp. 4–12. (in Russian).
5. Berezovskij P.S., Kovalenko V.N. Structure and Functionality of the Job Management System for Grid with Non-Clustered Resources. *Preprinty IPM im. M. V. Keldysha* [KIAM Preprints]. 2007, no. 67, pp. 1–29. (in Russian).
6. WMS Architecture overview. Available at: <http://egee-jra1-wm.mi.infn.it/egee-jra1-wm/wms.shtml> (accessed: 27.03.2017).
7. Internal Architecture 5.14. Available at: http://www.gridway.org/doku.php?id=documentation:release_5.14:iashtml (accessed: 20.03.2017).
8. Cirne W., Brasileiro F., Costa L., Paranhos D., Santos-Neto E., Andrade N. Scheduling in Bag-of-Task Grids: PAUA Case. 16th Symposium on Computer Architecture and High Performance Computing. Oct. 2004, pp. 124–131. DOI: 10.1109/CAHPC.2004.37.
9. Kovalenko V.N., Orlov A.V. Metascheduling in GRID and Resource Reservation Protocol. *Preprinty IPM im. M. V. Keldysha* [KIAM Preprints]. 2002, no. 1, pp. 1–25. (in Russian).

10. Buncic P., Saiz P., Peters A.J. The AliEn System, Status and Perspectives. 2003 Conference for Computing in High-Energy and Nuclear Physics, La Jolla, CA, USA, 24–28 Mar 2003. Available at: <http://www.slac.stanford.edu/econf/C0303241/proc/papers/MOAT004.PDF> (accessed: 20.03.2017).
11. Toporkov V.V., Emel'janov D.M., Potehin P.A. Job Batch Generation and Scheduling in Distributed Computing Environments. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya: Vychislitel'naja matematika i informatika* [Bulletin of South Ural State University. Series: Computational Mathematics and Software Engineering]. 2015, vol. 4, no. 2. pp. 44–57. DOI: 10.14529/cmse150204 (in Russian).
12. Valiev M.K., Kitaev E.L., Slepnev M.I. LDAP Directory Service as a Tool for Implementation of Distributed Information Systems. *Preprinty IPM im. M. V. Keldysha* [KIAM Preprints]. 2000. no. 23. pp. 1–22. (in Russian).
13. Kesselman C., Fitzgerald S., Foster I., Tuecke S., Smith W. A Directory Service for Configuring High-Performance Distributed Computations. 6th IEEE Symposium on High Performance Distributed Computing. 1997. pp. 365–375. DOI: 10.1109/HPDC.1997.626445.
14. Loewenstern A. Norberg A. DHT Protocol. 2008. Available at: http://bittorrent.org/beps/bep_0005.html (accessed: 11.03.2017).
15. ClickHouse Reference Manual. 2015. Available at: https://clickhouse.yandex/reference_en.html (accessed: 16.02.2017).
16. Elastic Stack and Product Documentation. 2016. Available at: <https://www.elastic.co/guide/index.html> (accessed: 22.01.2017).
17. Programming with Redis. 2016. Available at: <https://redis.io/documentation> (accessed: 12.02.2017).
18. Prasad A. Announcing Docker Compose. 2015. Available at: <https://blog.docker.com/2015/02/announcing-docker-compose/> (accessed: 26.02.2017).

ОБОБЩЕННАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ МОДУЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ПРОВЕРКИ ДОПУСТИМОСТИ КОНФИГУРАЦИЙ ТАКИХ СИСТЕМ*

© 2017 А.Б. Глонина

*Московский государственный университет имени М.В. Ломоносова
(119991 Москва, ул. Ленинские горы, д. 1)*

E-mail: alevtina@lwk.cs.msu.su

Поступила в редакцию: 06.09.2017

В работе рассматривается задача проверки допустимости конфигураций модульных вычислительных систем реального времени (МВС РВ). Конфигурация считается допустимой, если все работы успевают выполняться в рамках своих директивных сроков. В статье приведены формальные определения конфигурации МВС РВ и временной диаграммы (ВД) ее функционирования. ВД, соответствующая заданной конфигурации, необходима для проверки критерия допустимости для данной конфигурации. Автором предложена обобщенная модель функционирования таких систем, основанная на математическом аппарате временных автоматов с остановкой таймеров. Представлен метод построения модели конкретной МВС РВ по описанию ее конфигурации. Модель позволяет получить временную диаграмму функционирования системы по вычислению соответствующей сети автоматов. Математический аппарат сетей временных автоматов позволил доказать, что модель детерминирована, а также удовлетворяет ряду требований корректности, выделенных из спецификаций на МВС РВ. Предложенные методы реализованы программно. Проведено экспериментальное сравнение предложенного подхода с альтернативным подходом (верификацией моделей систем), использующим такой же математический аппарат. Эксперименты подтвердили эффективность разработанной модели. Для апробации на данных, приближенных к реальным, программная реализация модели была также интегрирована с существующим средством планирования вычислений в МВС РВ, используемым в промышленности. Результаты апробации подтвердили применимость предложенного подхода на практике.

Ключевые слова: модульные вычислительные системы, проверка корректности расписаний, временные автоматы, системы реального времени

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Глонина А.Б. Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 43–59. DOI: 10.14529/cmse170404.

Введение

В настоящее время наиболее перспективным типом архитектуры встроенных вычислительных систем является модульная архитектура. Современные модульные вычислительные системы реального времени (МВС РВ) состоят из большого количества параллельно работающих вычислительных модулей. Так автомобили премиум-класса имеют вычислительные системы архитектуры AUTOSAR, состоящие из 10–20 вычислительных модулей, содержащие в общей сложности около 100 вычислительных ядер, на которых выполняются около 800 приложений [1]. Другим примером МВС РВ

*Статья рекомендована к публикации программным комитетом Международной конференции «Суперкомпьютерные дни в России – 2017».

являются системы интегрированной модульной авионики (ИМА), которые могут состоять из десятков модулей. На каждом модуле выполняются сотни задач, обрабатывающих большие объемы информации. Например, на борту Boeing 787 генерируется около 0,5 ТБ данных за полет, и ожидается, что на борту Airbus A350 за полет будет генерироваться около 1 ТБ данных [2]. В данной работе МВС РВ будут рассматриваться на примере систем ИМА, однако предложенные методы применимы и к другим МВС РВ.

МВС РВ состоят из стандартизированных модулей, каждый из которых содержит набор многоядерных процессоров. Один модуль может иметь процессоры нескольких типов, различающихся производительностью. Взаимодействие между модулями осуществляется посредством коммутируемых сетей с виртуальными каналами.

Рабочая нагрузка МВС РВ представляет собой набор разделов. Разделом называется группа логически связанных задач, взаимодействующих посредством общей памяти и, как правило, соответствующих одному приложению. Все задачи являются периодическими: за период должен выполняться экземпляр задачи, называемый работой. Между задачами с одинаковым периодом могут существовать зависимости по данным: очередная работа задачи-получателя не может начать выполнение до тех пор, пока не получит данные от всех соответствующих работ задач-отправителей.

Каждый раздел привязан к вычислительному ядру в составе процессора. При этом несколько разделов могут быть привязаны к одному и тому же ядру. На интервале планирования для ядра задается статическое расписание временных интервалов, называемых окнами, в течение каждого из которых выполняются работы определенного раздела. Каждый раздел имеет собственный планировщик, управляющий выполнением работ внутри окон и работающий согласно одному из алгоритмов планирования (как правило, динамических). Допускается использование разных алгоритмов планирования для разных разделов. Одним из наиболее часто применяемых алгоритмов планирования является планирование с вытеснением на основе статических приоритетов. Каждая работа должна завершить выполнение не позднее определенного для нее директивного срока. Если директивный срок наступает до завершения выполнения работы, то она считается опоздавшей и не может далее выполняться на вычислительном ядре.

Конфигурация МВС РВ определяет набор ее модулей, характеристики рабочей нагрузки, привязку разделов к ядрам и расписания окон для каждого ядра. Конфигурация является *допустимой*, если для нее все работы полностью выполняются в рамках своих директивных сроков.

В процессе проектирования МВС РВ анализируется большое число потенциальных конфигураций и возникает задача проверки допустимости конкретной конфигурации.

Существующие аналитические методы решения этой задачи (например, [3]) не учитывают таких особенностей организации МВС РВ, как использование в разных разделах различных алгоритмов планирования. Метод построения и последующей формальной верификации модели МВС РВ позволяет преодолеть эту проблему, однако он не применим для систем большой размерности, так как вычислительная сложность этого метода экспоненциально зависит от количества задач. Так в [4] для конфигурации с 25 задачами процесс верификации занимает уже несколько минут, в то время как реальные системы содержат сотни задач, а количество проверяемых конфигураций может достигать нескольких тысяч.

Другим подходом к проверке допустимости конфигураций является построение временной диаграммы (ВД) функционирования МВС, то есть моделирование функционирования системы, в процессе которого определяются моменты постановки каждой работы на выполнение и снятия ее с вычислительного ядра. Существующие средства построения ВД функционирования вычислительных систем обладают рядом недостатков: некоторые из них не поддерживают всех особенностей МВС РВ [5, 6], некоторые не допускают автоматизации построения моделей [7], некоторые рассчитаны на решение конкретных задач, не предполагают расширения и не имеют доказательства корректности используемых моделей [8, 9]. В данной работе предлагается подход, позволяющий преодолеть перечисленные недостатки.

Автором разработана обобщенная модель функционирования МВС РВ, на основе которой по описанию конфигурации конкретной МВС РВ автоматически строится модель этой системы. Это позволяет автоматически проверять допустимость большого числа конфигураций. В качестве математического аппарата для моделирования предлагается использовать сети временных автоматов с остановкой таймеров [10]. Модели, описанные с помощью этого аппарата, позволяют получить требуемые ВД, а также для них может быть строго доказано выполнение ряда требований корректности.

В работе доказано, что из выполнения выделенного набора требований корректности к обобщенной модели следует эквивалентность всех ВД, построенных моделью, соответствующей конкретной конфигурации. Это позволяет использовать для получения ВД любой сценарий выполнения модели, что делает возможной быструю проверку допустимости конфигураций МВС, содержащих большое количество параллельно работающих компонентов.

Статья имеет следующую структуру. В разделе 1 формализовано понятие конфигурации МВС РВ и критерия ее допустимости. В разделе 2 описаны сети временных автоматов с остановкой таймеров. В разделе 3 приведена обобщенная формальная модель функционирования МВС РВ. В разделе 4 представлен алгоритм построения моделей конкретных МВС на основе обобщенной модели. Раздел 5 посвящен обоснованию корректности и детерминированности моделей. В разделе 6 описана программная реализация и апробация предложенных методов. В заключении сформулированы итоги работы.

1. Формальное описание МВС РВ

Конфигурация МВС РВ имеет вид $\langle HW, WL, Bind, Sched \rangle$, где:

- $HW = \{HW_i\}_{i=1}^N$ — множество вычислительных ядер; $Type : HW \rightarrow \overline{1, N_t}$ — тип процессора, которому принадлежит ядро ($N_t \in \mathbb{N}$ — количество типов процессоров); считается, что в каждом процессоре все ядра одинаковы;
- $WL = \langle Part, G \rangle$ — рабочая нагрузка, где:
 - $Part = \{Part_i = \langle T_i, A_i \rangle\}_{i=1}^M$ — множество разделов, где:
 - * $M \in \mathbb{N}$ — количество разделов;
 - * $K_i \in \mathbb{N}$ — количество задач в i -ом разделе;
 - * $T_i = \{T_{ij}\}_{j=1}^{K_i}$ — множество задач i -го раздела, каждая из которых характеризуется уникальным в рамках раздела приоритетом (pr_{ij}), временем выполнения в худшем случае (WCET) на разных типах процессоров ($\overline{C_{ij}} = (C_{ij}^1, \dots, C_{ij}^{N_t})$), периодом (P_{ij}) и директивным сроком ($D_{ij} \leq P_{ij}$);

- * A_i — тип алгоритма планирования;
- $G = \langle \cup_{i=1}^M T_i, \{Msg_j\}_{j=1}^H \rangle$ — ациклический направленный граф зависимостей по данным, вершины которого соответствуют задачам, а дуги — сообщениям; каждое сообщение Msg_j характеризуется максимальными длительностями передачи через память модуля и через сеть (DM_j и DN_j соответственно);
- $Bind : Part \rightarrow HW$ — привязка разделов к ядрам;
- $Sched = \{ \{ \langle Start_{ij}, Stop_{ij} \rangle \}_{j=1}^{N_i^w} \}_{i=1}^M$ — расписание окон разделов, заданное на интервале планирования L , где L равно наименьшему общему кратному всех P_{ij} ; $N_i^w \in \mathbb{N}$ — количество окон для i -го раздела; $Start_{ij}, Stop_{ij} \in \overline{0, L}$ — времена начала и завершения j -го окна i -го раздела.

В данной работе предполагается, что время дискретно: все времена и длительности, входящие в состав конфигурации, задаются в квантах. Квант равен наибольшему общему делителю тактов процессоров. Пусть $CONF$ — множество всевозможных конфигураций МВС РВ.

Далее введем понятие временной диаграммы (ВД) функционирования МВС РВ.

Для каждой задачи T_{ij} на интервале планирования определен набор работ $W_{ij} = \{w_{ijk}\}_{k=1}^{L/P_{ij}}$. $D_{ijk} = P_{ij} * (k - 1) + D_{ij}$ — директивный срок k -й работы.

Пусть $e = \langle Type, Src, t \rangle$ — событие, где $Type \in \{EX, PR, FIN\}$ — тип события: постановка работы на выполнение (EX); вытеснение работы с ядра (PR); завершение работы, связанное с полным выполнением работы, либо с наступлением директивного срока (FIN). $Src \in \cup_i \cup_j W_{ij}$ — работа-источник события. $t \in \{n\}_{n=1}^L$ — время.

$E = \{EX, PR, FIN\} \times \cup_i \cup_j W_{ij} \times \{n\}_{n=1}^L$ — множество всевозможных событий.

$A = A_1, \dots, A_M$ — совокупность алгоритмов планирования, используемых в разделах.

На практике [9, 11] при проектировании МВС РВ предполагают, что время выполнения каждой работы на соответствующем ядре фиксировано и равно наихудшему, время передачи каждого сообщения также фиксировано и равно наихудшему, а алгоритм планирования детерминирован. В таких предположениях совокупность алгоритмов планирования A однозначно определяет отображение $Q : CONF \rightarrow 2^E$.

ВД для конфигурации $conf \in CONF$ — подмножество множества E , однозначно соответствующее $conf$, и являющееся результатом интерпретации A на $conf$.

Пусть $conf \in CONF$ — некоторая конфигурация системы, а $Q(conf)$ — ВД, соответствующая $conf$. Пусть R_{ijk} — количество интервалов выполнения работы w_{ijk} . Тогда упорядоченный набор событий для w_{ijk} имеет вид:

- \emptyset , если $R_{ijk} = 0$ (работа не была поставлена на выполнение);
- $\{ \langle w_{ijk}, EX, t_0 \rangle, \langle w_{ijk}, FIN, t_1 \rangle \} \subseteq Q(conf)$, если $R_{ijk} = 1$ (работа была поставлена на выполнение в момент t_0 , ни разу не вытеснялась и завершилась в момент t_1);
- $\{ \langle w_{ijk}, EX, t_0 \rangle, \langle w_{ijk}, PR, t_1 \rangle, \dots, \langle w_{ijk}, EX, t_{2R_{ijk}-2} \rangle, \langle w_{ijk}, FIN, t_{2R_{ijk}-1} \rangle \} \subseteq Q(conf)$, если $R_{ijk} > 1$, (работа была поставлена на выполнение в момент t_0 , несколько раз вытеснялась и завершилась в момент $t_{2R_{ijk}-1}$).

Таким образом, критерий допустимости конфигурации, заключающийся в том, что все работы должны быть полностью выполнены в рамках своих директивных сроков, имеет следующий формальный вид:

$$CR(conf) : \forall w_{ijk}, i \in \overline{1, M}, j \in \overline{1, K_i}, k \in \overline{1, L/P_{ij}}: R_{ijk} \geq 1 \text{ и } \sum_{r=1}^{R_{ijk}} (t_{2r-1} - t_{2r-2}) = C_{ij}^{Type(Bind(Part_i))}$$

В работе решается задача построения отображения $Q : CONF \rightarrow 2^E$, позволяющего для данной конфигурации МВС РВ получить ВД ее функционирования, необходимую для проверки критерия CR допустимости конфигурации. Для построения искомого отображения Q будем строить модель функционирования МВС РВ.

2. Сети временных автоматов с остановкой таймеров

К математическому аппарату для построения модели функционирования МВС РВ были сформулированы следующие требования: 1) возможность получения ВД для заданной конфигурации; 2) возможность формализации и проверки выполнения требований корректности к модели; 3) наличие программных средств для моделирования и проверки выполнения требований.

Автором был рассмотрен ряд математических аппаратов [10, 12–18] и из них были выбраны сети временных автоматов с остановкой таймеров [10], так как они в наибольшей степени удовлетворяют перечисленным требованиям.

Автомат с остановкой таймеров — это конечный автомат с целочисленными переменными и таймерами. Графически такой автомат может быть представлен размеченным ориентированным графом, вершины которого называются локациями, а дуги — переходами. Состоянием автомата называется совокупность его текущей локации, значений переменных и таймеров.

Таймер — это специальная переменная, принимающая вещественные значения. Начальное значение каждого таймера равно нулю. Таймер считается активным, если его условие активности (булево выражение над множеством переменных) в текущей локации автомата истинно. В противном случае таймер считается остановленным.

Для автомата указывается начальная локация. Каждой локации сопоставлен инвариант — булево выражение над таймерами и переменными. Автомат может оставаться в данной локации до тех пор, пока инвариант этой локации имеет истинное значение.

Каждый переход характеризуется условием перехода, действиями и, возможно, синхронизацией. Переход активен, если автомат находится в локации, предшествующей этому переходу, значения переменных и таймеров удовлетворяют условию перехода, и инвариант локации-назначения перехода имеет истинное значение. Переход может (но не обязан) быть совершен, если он активен и может быть выполнена синхронизация. При совершении перехода меняется текущая локация автомата, происходит синхронизация и выполняются действия (изменение переменных и обнуление таймеров).

Состояние автомата может измениться не только в результате выполнения перехода, но и в результате синхронного увеличения значений всех таймеров. Значения всех таймеров, кроме остановленных, могут быть увеличены на некоторую вещественную величину d , если $\forall d' : 0 < d' \leq d$ значения таймеров, увеличенные на d' , удовлетворяют инварианту текущей локации. В некоторых локациях, называемых срочными, продвижение времени невозможно.

Сеть автоматов представляет собой набор автоматов, функционирующих синхронно и взаимодействующих посредством общих переменных и каналов.

Канал — средство синхронизации автоматов. Синхронизация характеризуется именем канала и действием. Существует два парных действия: отправка и прием сигнала по каналу. Также существует два типа каналов — «точка-точка» и широкоэвещательные. Если в некоторый момент времени в двух автоматах сети активны переходы с парными действиями

синхронизации по одному и тому же каналу «точка-точка», то эти переходы выполняются одновременно. $N + 1$ автомат могут синхронизироваться посредством широкополосного канала, если в одном автомате активен переход с отправкой сигнала по этому каналу, а в остальных N — с приемом сигнала.

Формально автомат — это кортеж $A = \langle L, U, l_0, C, V, \bar{v}_0, AA, AS, E, I, P \rangle$, где:

- $L, U \subseteq L, l^0 \in L$ — множества локаций и срочных локаций, начальная локация;
- C — множество таймеров;
- V, \bar{v}_0 — множество переменных и их начальные значения;
- AA, AS — множество действий над таймерами и переменными и множество синхронизаций;
- E — множество переходов, $E \subseteq L \times B(C, V) \times AS \times AA \times L$. $B(C, V)$ — множество всевозможных условий над C и V ;
- $I : L \rightarrow B(C, V)$ — назначение инвариантов;
- $P : (L \times C) \rightarrow B(\emptyset, V)$ — назначение условий активности таймеров.

Пусть $A = A_1, \dots, A_n$ — сеть автоматов, где $A_i = \langle L_i, U_i, l_i^0, C_i, V_i, \bar{v}_i^0, AU_i, AS_i, E_i, I_i, P_i \rangle$. Состоянием сети называется кортеж $\langle \bar{l}, \bar{c}, \bar{v} \rangle \in (L_1 \times \dots \times L_n) \times \mathbb{R}_{\geq 0}^{|C|} \times \mathbb{Z}^{|V|}$, где $V = \bigcup_{i=1}^n V_i, C = \bigcup_{i=1}^n C_i$. Конечная или бесконечная последовательность состояний, получаемая при некотором сценарии функционирования сети автоматов, называется *вычислением* этой сети.

3. Обобщенная формальная модель функционирования МВС РВ

Введем ряд определений, необходимых для описания предложенной модели.

Параметризованный автомат с остановкой таймеров, или *конкретный тип автоматов* — кортеж $\langle L, U, l_0, C, V, \bar{p}, AU, AS, E, I, P \rangle$, где \bar{p} — целочисленные параметры с не заданными значениями. *Интерфейс автомата* — набор переменных и синхронизаций, посредством которых автомат взаимодействует с другими автоматами. *Базовый тип автоматов* определяется интерфейсом, то есть парой $\langle V_b, AS_b \rangle$, где V_b — множество переменных, AS_b — множество синхронизаций. Конкретный тип автоматов *реализует* базовый тип, если $V_b \subseteq V, AS_b \subseteq AS$.

Набор базовых типов автоматов с определенными между ними логическими связями назовем *обобщенной сетью автоматов*. Под логическими связями понимается спецификация того, какие базовые типы автоматов могут взаимодействовать и с помощью каких средств (переменных и синхронизаций). *Конкретная сеть автоматов* — набор конкретных типов автоматов с определенными между ними логическими связями. Конкретная сеть автоматов *реализует* обобщенную сеть автоматов, если каждый базовый тип автоматов обобщенной сети реализуется одним или несколькими конкретными типами автоматов конкретной сети, и логические связи между типами автоматов в базовой сети соответствуют логическим связям в конкретной сети.

Модельное время — значение специального таймера, который никогда не обнуляется, и условие активности которого всегда истинно.

Событие синхронизации — тройка $\langle CH, A, t \rangle$, где CH — канал, A — набор автоматов, участвующих в синхронизации, t — модельное время. *Временная диаграмма (ВД) сети автоматов* — набор событий синхронизации для некоторого вычисления сети.

Автором предложено представление обобщенной модели функционирования МВС РВ в виде введенной выше обобщенной сети автоматов с остановкой таймеров.

В разработанной обобщенной сети имеются следующие средства взаимодействия автоматов (см. обозначения, введенные в разделе 1):

- переменные is_ready_{ij} , is_failed_{ij} , $i \in \overline{1, M}, j \in \overline{1, K_i}$, $is_data_ready_h$, $h \in \overline{1, H}$, соответствующие готовности опозданию работы задачи T_{ij} и доставке сообщения по h -му виртуальному каналу;
- переменные $prio_{ij}$, $deadline_{ij}$, $i \in \overline{1, M}, j \in \overline{1, K_i}$, значения которых равны приоритету и директивному сроку задачи T_{ij} и не изменяются в процессе функционирования сети;
- каналы $wakeup_i$, $sleep_i$, $i \in \overline{1, M}$, соответствующие началу и завершению очередного окна i -го раздела;
- каналы $ready_i$, $finished_i$, $exec_{ij}$, $preempt_{ij}$, $i \in \overline{1, M}, j \in \overline{1, K_i}$, соответствующие готовности, завершению (вследствие окончания выполнения, либо наступления директивного срока), постановке на выполнение и вытеснению некоторой работы;
- широковещательные каналы $send_{ij}$, $receive_{ij}$, $i \in \overline{1, M}, j \in \overline{1, K_i}$, соответствующие отправке сообщения работой-отправителем и получению сообщения работой-получателем.

Обобщенная сеть автоматов состоит из следующих базовых типов автоматов.

1. Базовый тип автоматов **T**, моделирующий функциональную задачу и определенный следующим интерфейсом:

- получение сигналов по каналам **exec** и **preempt** и отправка сигналов по каналам **ready** и **finished**;
- отправка сигналов по широковещательному каналу **send** и получение сигналов по широковещательному каналу **receive**;
- изменение переменных is_ready , is_failed и $is_data_ready_h$, где h — номер виртуального канала, по которому работы данной задачи получают сообщения.

2. Базовый тип автоматов **TS**, моделирующий планировщик работ раздела и определенный следующим интерфейсом:

- получение сигналов по каналам **wakeup**, **sleep**, **ready** и **finished**;
- отправка сигналов по каналам $exec_j$, $preempt_j$, где j -й канал соответствует j -й задаче раздела;
- чтение значений переменных is_ready_j , $prio_j$, $deadline_j$.

3. Базовый тип автоматов **CS**, моделирующий планировщик ядра. Планировщик ядра переключает окна разделов согласно статическому расписанию, являющемуся элементом конфигурации МВС РВ. **CS** определен следующим интерфейсом:

- отправка сигналов по каналам $wakeup_i$ и $sleep_i$, где j -й канал соответствует j -му разделу;

4. Базовый тип автоматов **L**, моделирующий виртуальный канал и определенный следующим интерфейсом:

- получение сигналов по широковещательному каналу **send** и отправка сигналов по широковещательному каналу **receive**;
- изменение переменной is_data_ready .

Структура предложенной обобщенной сети автоматов, отражающая логические связи между ее компонентами, показана на рис. 1.

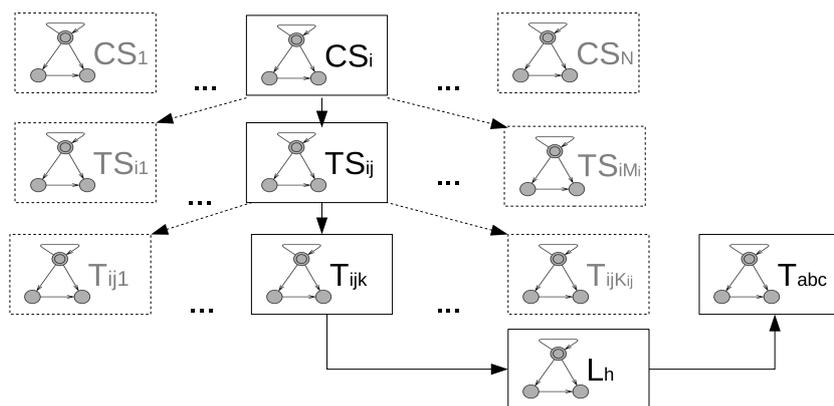


Рис. 1. Структура обобщенной сети автоматов, моделирующей функционирование МВС РВ

4. Проверка допустимости конфигураций конкретных МВС РВ

Конкретная сеть автоматов, реализующая предложенную обобщенную сеть автоматов является параметризованной моделью функционирования МВС РВ, параметрами которой являются элементы конфигурации системы. Эта модель содержит следующие конкретные типы автоматов: модели задачи, планировщика ядра, виртуального канала и нескольких планировщиков работ, работающих согласно различным алгоритмам планирования (с фиксированными и динамическими приоритетами, с вытеснением и без вытеснения).

Для данной конкретной сети автоматов и конфигурации МВС РВ $conf \in CONF$ экземпляр сети автоматов, моделирующий функционирование системы, может быть построен согласно следующему алгоритму:

- цикл по ядрам ($i \in [1, N]$):
 - цикл по разделам ядра ($j \in [1, M] : Bind(Part_j) = HW_i$):
 - * создание каналов $ready_j, finished_j, wakeup_j, sleep_j$;
 - * цикл по задачам ($k \in [1, K_j]$):
 - создание каналов $exec_{jk}, preempt_{jk}, send_{jk}, receive_{jk}$;
 - создание переменных $is_ready_{jk}, prio_{jk}, deadline_{jk}$ и их инициализация соответствующими значениями из $conf$;
 - создание переменных $is_data_ready_h$, соответствующих виртуальным каналам, для которых задача T_{jk} является получателем, и инициализация их значением 0;
 - порождение автомата конкретного типа, реализующего базовый тип T (моделирующего задачу), инициализация его интерфейса каналами $exec_{jk}, preempt_{jk}, send_{jk}, receive_{jk}, ready_j, finished_j$ и переменными $is_ready_{jk}, prio_{jk}, deadline_{jk}, is_data_ready_h$ и инициализация его параметров соответствующими значениями из $conf$;
 - * порождение автомата конкретного типа, реализующего базовый тип TS (моделирующего планировщик работ) и соответствующего алгоритму планирования A_j j -го раздела, инициализация его интерфейса каналами $exec_{jk}, preempt_{jk}, ready_j, finished_j, wakeup_j, sleep_j$ и переменными $is_ready_{jk}, prio_{jk}, deadline_{jk}$ ($k \in [1, K_j]$);

- порождение автомата конкретного типа, реализующего базовый тип CS (моделирующего планировщик ядра), инициализация его интерфейса каналами $wakeup_j$, $sleep_j$ и инициализация его параметров соответствующими значениями из $Sched$ ($j \in [1, M] : Bind(Part_j) = HW_i$);
- цикл по виртуальным каналам ($h \in [1, H]$):
 - порождение автомата конкретного типа, реализующего базовый тип L (моделирующего виртуальный канал), инициализация его интерфейса соответствующими каналами $send_{j_1k_1}$, $receive_{j_1k_1}$ и переменной $is_data_ready_h$ и инициализация его параметров соответствующими значениями из $conf$.

Таким образом, МВС РВ с фиксированной конфигурацией взаимно однозначно (в рамках предложенной модели) соответствует моделирующая ее сеть автоматов.

По ВД сети автоматов, построенной для некоторой конфигурации, может быть получена ВД МВС РВ, необходимая для проверки критерия допустимости данной конфигурации. Каждому событию синхронизации $\langle CH(AS_{sync}), A_{sync}, t_{sync} \rangle$ из ВД сети автоматов, такому что $CH(AS_{sync}) \in \{exec_{jk}, preempt_{jk}, finished_j\}$ однозначно ставится в соответствие событие $\langle Type, Src, t \rangle$ ВД МВС РВ по следующим правилам:

- тип события (EX, PR, FIN) соответствует каналу ($exec_{jk}, preempt_{jk}, finished_j$);
- работа-источник события определяется автоматом-моделью задачи, участвующим в синхронизации, и временем синхронизации: $Src = w_{ijk}$, где $i, j : A_{sync}^l = T_{ij}$, $k = \lfloor t_{sync}/P_{ij} \rfloor + 1$ (для EX и PR) и $k = \lceil t_{sync}/P_{ij} \rceil$ (для FIN);
- время события равно времени синхронизации: $t = t_{sync}$.

5. Корректность и детерминированность модели

В данном разделе приведен подход к доказательству корректности и детерминированности всех моделей, построенных на основе предложенной обобщенной модели.

Спецификации МВС РВ содержат ряд требований корректности к функционированию системы в целом и ее компонентов. Эти требования представляют собой ограничения на последовательности событий, происходящих в системе, и на длительности интервалов между определенными типами событий. Для того, чтобы гарантировать корректность проверки критерия допустимости конфигураций, используемая для этого модель должна удовлетворять требованиям корректности к целевой системе, применимым на выбранном для моделирования уровне абстракции.

Назовем модель МВС РВ *детерминированной*, если ВД, получаемая при ее выполнении, определяется однозначно. Детерминированность моделей крайне важна при проектировании МВС РВ большой размерности, так как из нее следует то, что для проверки допустимости некоторой конфигурации достаточно получить и проанализировать одно любое вычисление сети автоматов, моделирующей МВС РВ с данной конфигурацией. В этом состоит отличие предложенного метода от верификации модели системы [4], в процессе которой анализируются все возможные вычисления сети автоматов и которая имеет неприемлемую для исследования систем большой размерности вычислительную сложность.

Выбранный математический аппарат моделирования позволяет проверять выполнение требований к компонентам модели автоматически с помощью верификатора. Для этого используется подход «автоматов-наблюдателей» [19].

«Автомат-наблюдатель» — это автомат, работающий синхронно с исходным автоматом, и соответствующий, как правило, одному требованию из спецификации МВС. У автомата-наблюдателя существует «плохая» локация, соответствующая некорректным последовательностям синхронизаций в исходном автомате. Во всех локациях «наблюдателя» активны переходы с действиями синхронизации, парными ко всем возможным действиям синхронизации исходного автомата. Достижимость «плохой» локации означает, что в исходном автомате может произойти некорректная последовательность синхронизаций, соответствующая некорректной последовательности событий в МВС РВ.

В данной работе требуется проверить корректность построенных конкретных типов автоматов, то есть параметризованных автоматов, так как это необходимо для доказательства корректности всех моделей, синтезируемых на основе обобщенной модели и содержащих экземпляры автоматов этих конкретных типов. Таким образом, нужно убедиться в том, что для каждого конкретного типа автоматов некорректные последовательности синхронизаций невозможны ни при каких значениях его параметров. Поэтому в «автомате-наблюдателе» значения параметров выбираются недетерминированно. Кроме того, если исходный автомат использует переменные, изменяемые другими автоматами, то «наблюдатель» меняет значения этих переменных недетерминированно.

Для проверки корректности компонента параметризованной модели необходимо выделить из спецификаций МВС РВ применимые к нему на выбранном уровне абстракции требования, для каждого из них построить «автомат-наблюдатель» и проверить недостижимость его «плохой» локации. Такая проверка была выполнена автором для всех разработанных конкретных типов автоматов и требований, выделенных из стандарта ARINC653 [20] на системы ИМА. Проверка проводилась с использованием верификатора UPPAAL [13].

Рассмотрим следующий пример требования:

Для любого раздела системы верно, что в каждый момент времени в нем выполняется не более одной работы.

Это требование относится к базовому типу автоматов TS, моделирующему планировщик работ, и должно быть выполнено для всех конкретных типов автоматов, реализующих TS.

В терминах синхронизаций автоматов выполнением работ задачи T_{jk} являются интервалы между синхронизациями по каналам `execjk` и либо `preemptjk`, либо `finishedj`. Сформулированное требование означает, что за любой синхронизацией по каналу `execjk` должна следовать синхронизация по либо по каналу `preemptjk`, либо по каналу `finishedj`. «Автомат-наблюдатель», соответствующий данному требованию, приведен на рис. 2.

Проверка требований к параметризованной модели МВС РВ в целом не может быть выполнена автоматически, так как количество автоматов, входящих в модель, также является ее параметром и в общем случае неизвестно. Поэтому доказательство выполнения таких требований корректности было выполнено вручную посредством строгих логических рассуждений и основано на доказанном ранее выполнении требований корректности к отдельным компонентам модели.

Рассмотрим пример требования корректности к модели в целом и его доказательство:

Для любого ядра системы верно, что в каждый момент времени на нем выполняется не более одной работы.

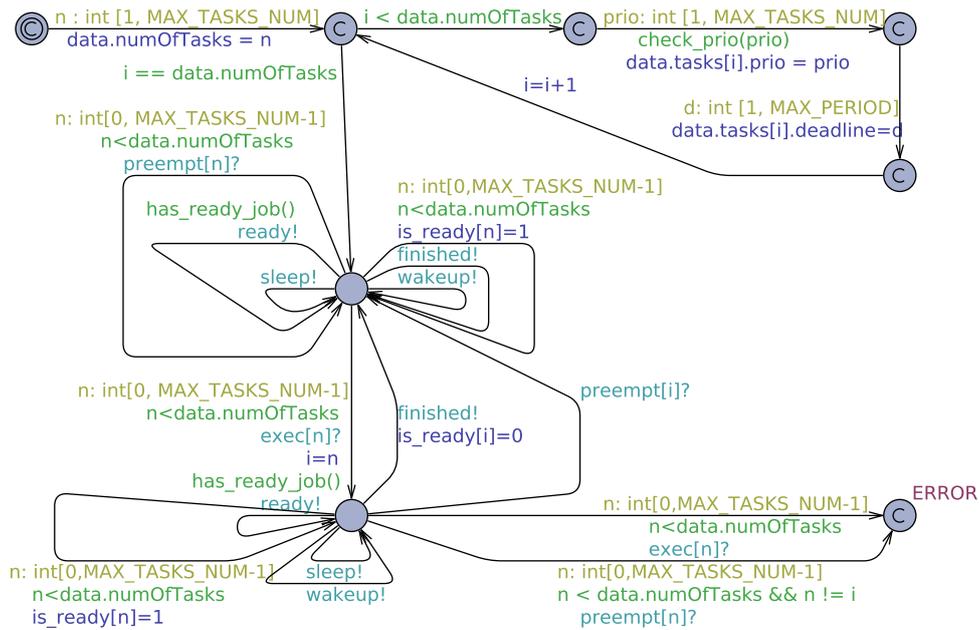


Рис. 2. Автомат-наблюдатель для требования к модели планировщика работ

Выполнение следующих требований к компонентам модели было автоматически доказано с использованием соответствующих «автоматов-наблюдателей»:

1. Для любого раздела системы верно, что в каждый момент времени в нем выполняется не более одной работы.
2. Для любого ядра системы верно, что в каждый момент времени на нем выполняется не более одного раздела.

Из выполнения этих требований к компонентам модели и следует выполнение сформулированного требования к модели в целом.

Доказательство детерминированности модели основывается на ранее доказанном выполнении требований корректности к модели. Приведем подход к этому доказательству.

Предположим, что модель не является детерминированной, то есть для экземпляра сети автоматов, соответствующей некоторой конфигурации, могут быть получены две разные ВД. Упорядочим события этих двух ВД по времени. Так как события в системе могут происходить одновременно, то к каждой временной точке ВД привязано некоторое множество событий. Пусть t_i — точка с наименьшим модельным временем, для которой множества событий двух ВД различаются. Это означает, что как минимум одно событие синхронизации присутствует в множестве событий для этой точки одной трассы и отсутствует в соответствующем множестве другой. Предположим, что это синхронизация по каналу finished_j . Так как для всех предыдущих временных точек множества событий совпадают, то возможно два варианта:

1. Согласно трассе, содержащей данное событие, некоторая работа выполнялась на ядре в общей сложности WCET единиц времени. Из этого следует, что, согласно второй трассе, в которой данного события нет, эта работа выполнялась на ядре в общей сложности более, чем WCET единиц времени.
2. Согласно трассе, содержащей данное событие, некоторая работа была снята с ядра по причине наступления ее директивного срока. Из этого следует, что согласно второй трассе, в которой данного события нет, эта работа осталась на ядре после наступления ее директивного срока.

Оба варианта противоречат доказанным ранее требованиям корректности. Для синхронизаций по другим типам каналов проводятся аналогичные рассуждения и делается вывод о том, что сформулированное предположение неверно.

Таким образом, доказано, что между конфигурацией МВС и ее моделью существует взаимно однозначное соответствие, модель детерминирована и корректна, между ВД модели и ВД МВС также существует взаимно однозначное соответствие. Следовательно, проверка критерия допустимости конфигураций с помощью предложенной модели корректна.

6. Программная реализация и апробация

Для апробации предложенного подхода автором была создана программная реализация разработанных моделей и методов.

Конкретные типы автоматов, моделирующие компоненты системы, были разработаны и верифицированы с использованием средства UPPAAL [13]. Проверифицированные автоматы были помещены в библиотеку автоматных моделей компонентов МВС РВ. При необходимости библиотека может быть пополнена пользовательскими моделями при условии выполнения для них требований корректности.

В открытых источниках не было найдено средства, позволяющего эффективно моделировать функционирование сетей временных автоматов с остановкой таймеров. Поэтому на языке C++ была реализована собственная библиотека моделирования таких автоматов. Также был реализован транслятор моделей, описанных на языке UPPAAL, в представление на языке C++. Разработанная библиотека автоматных моделей была преобразована с помощью транслятора в библиотеку программных моделей, формирующих параметризованную программную модель функционирования МВС (см. рис. 3).

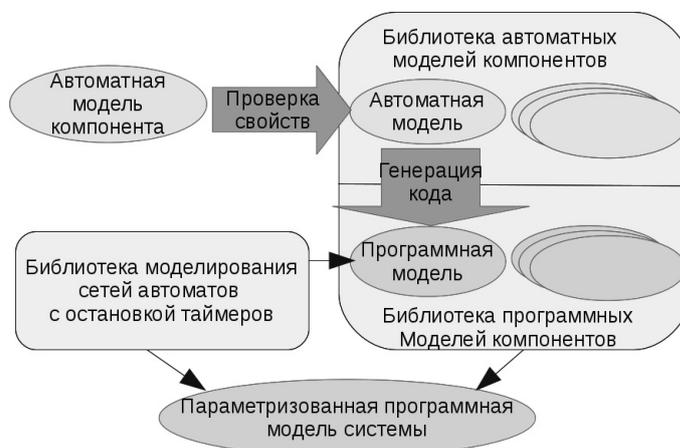


Рис. 3. Формирование параметризованной модели функционирования МВС

Для проверки предположения о большей эффективности предложенного подхода по сравнению с верификацией моделей МВС РВ была проведена серия экспериментов. В каждом эксперименте для фиксированной конфигурации МВС РВ на основе разработанных моделей компонентов строилась модель системы. В результате прогона полученной модели генерировалась ВД и с ее использованием проверялся критерий допустимости. Также проводилась верификация модели: проверялось, что ни для одного вычисления сети автоматов ни одна из переменных *is_failed* не принимает значение 1. Результаты сравнения времени проверки допустимости конфигурации с использованием ВД с

временем верификации соответствующей модели приведены в таблице и подтверждают эффективность предложенного подхода.

Таблица

Время проверки допустимости конфигураций с различным количеством работ, с

Количество работ	10	11	12	13	14	15	16	17	18
Верификация	0,57	1,16	2,22	5,05	10,43	23,51	48,13	112,28	215,91
Использование ВД	0,027	0,027	0,028	0,030	0,031	0,032	0,033	0,035	0,036

Разработанная модель была интегрирована с САПР ИМА, решающей одну из задач планирования вычислений [9]. На каждой итерации работы алгоритма планирования формируется конфигурация, для которой необходимо проверить критерий допустимости. Для конфигурации генерируется XML-файл с ее описанием и передается на вход параметризованной модели. По этому описанию строится и запускается экземпляр модели. В результате его выполнения генерируется ВД и в виде XML-файла передается в САПР ИМА, где производится проверка критерия допустимости. Эксперименты показали, что на данных, приближенных к реальным, генерация и прогон модели занимают несколько секунд (11 секунд для конфигурации с 12500 работами на интервале планирования), что свидетельствует о применимости предложенного подхода на практике.

Заключение

В работе формализована задача проверки допустимости конфигураций МВС РВ. Введен критерий допустимости, проверка которого предполагает анализ ВД функционирования системы. Предложена обобщенная модель функционирования МВС РВ, основанная на математическом аппарате сетей временных автоматов с остановкой таймеров. Представлен алгоритм построения модели МВС РВ по описанию ее конфигурации на основе обобщенной модели. Модель МВС РВ конкретной конфигурации представляет собой сеть автоматов, по вычислению которой может быть получена необходимая ВД МВС РВ. Предложенные методы реализованы программно. Благодаря выбранному математическому аппарату, для класса моделей, синтезируемых на основе обобщенной модели, доказана детерминированность и выполнение ряда требований корректности. Из детерминированности модели следует то, что для проверки допустимости конфигурации может быть использовано любое вычисление соответствующей сети автоматов. Поэтому предложенный подход значительно эффективнее подхода, заключающегося в верификации модели МВС РВ и использующего аналогичный математический аппарат, что было подтверждено экспериментально. Для апробации на данных, приближенных к реальным, разработанное программное средство интегрировано с САПР ИМА, используемой в промышленности. Результаты апробации показали применимость предложенного подхода на практике.

Возможными направлениями дальнейших исследований являются расширение библиотеки моделей компонентов МВС РВ, интеграция с другими средствами проектирования МВС РВ, расширение модели для ее использования при построении расписания окон разделов, а также применение подхода к другим классам распределенных систем.

Литература

1. Senthilkumar K., Ramadoss R. Designing Multicore ECU Architecture in Vehicle Networks Using AUTOSAR // Proceedings of the Third International Conference on Advanced Computing (ICoAC) (Chennai, India, December 14–16, 2011), 2011. P. 270–275. DOI: 10.1109/ICoAC.2011.6165187
2. From the Ground Up: How the Internet of Things Will Give Rise to Connected Aviation. Gogo LLC, 2016. 140 p.
3. Marinescu S., Tamas-Selicean D., Acretoaie V., et al. Timing Analysis of Mixed-Criticality Hard Real-Time Applications Implemented on Distributed Partitioned Architectures // Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (Krakow, Poland, September 17–21, 2012), 2012. P. 1–4. DOI: 10.1109/ETFA.2012.6489720
4. Macariu G., Cretu V. Timed Automata Model For Component-Based Real-Time Systems // Proceedings of the 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS) (Oxford, UK, March 22–26, 2010), 2010. P. 121–130. DOI: 10.1109/ECBS.2010.20
5. Craveiro J.P., Silveira R.O., Rufino J. HsSim: an Extensible Interoperable Object-Oriented n-Level Hierarchical Scheduling Simulator // Proceedings of the 3rd International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS) (Pisa, Italy, July 10, 2012), 2012. P. 9–14.
6. Singhoff F., Plantec A., Dissaux P., et al. Investigating the Usability of Real-Time Scheduling Theory with the Cheddar Project // Real-Time Systems. 2009. Vol. 43, No. 3. P. 259–295. DOI: 10.1007/s11241-009-9072-y
7. Khoroshilov A., Albitskiy D., Koverninskiy I., et al. AADL-Based Toolset for IMA System Design and Integration. // SAE Int. J. Aerosp. 2012. Vol. 5, No. 2. P. 294–299. DOI: 10.4271/2012-01-2146
8. Dissaux P., Marc O., Fotsing C., et al. The SMART project: Multi-Agent Scheduling Simulation Of Real-Time Architectures // Embedded Real Time Software and Systems. 2014.
9. Balashov V.V., Balakhanov V.A., Kostenko V.A. Scheduling of Computational Tasks in Switched Network-Based IMA systems // Proceedings of International Conference on Engineering and Applied Sciences Optimization (OPTI) (Athens, Greece, June 4–6, 2014), 2014. P. 1001–1014.
10. Cassez F., Larsen K. The Impressive Power of Stopwatches // CONCUR 2000 — Concurrency Theory (PA, USA, August 22–25, 2000). LNCS. Springer, 2000. Vol. 1877. P. 138–152. DOI: 10.1007/3-540-44618-4_12
11. Tretyakov A. Automation of Scheduling for Periodic Real-Time Systems // Proceedings of the Institute for System Programming. 2012. Vol. 22. P. 375–400. DOI: 10.15514/ISPRAS-2012-22-20
12. Smelyansky R.L. Model of Distributed Computing System Operation With Time // Programming and Computer Software, 2013. Vol. 39, No. 5. P. 233–241. DOI: 10.1134/S0361768813050046

13. Bengtsson J., Yi W. Timed Automata: Semantics, Algorithms and Tools // Lectures on Concurrency and Petri Nets. LNCS. Springer, 2004. Vol. 3098. P. 87–124. DOI: 10.1007/978-3-540-27755-2_3
14. Zuberek W.M. Timed Petri Nets Definitions, Properties and Applications // Microelectronics Reliability, 1991. Vol. 31, No. 4. P. 627–644. DOI: 10.1016/0026-2714(91)90007-T
15. Boudjadar A.J., Kim J.H., Larsen K.G., et al. Model Checking Process Algebra of Communicating Resources for Real-time Systems // Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS) (Madrid, Spain, July 8–11, 2014), 2014. P. 51–60. DOI: 10.1109/ECRTS.2014.24
16. Lime D., Roux O.H., Seidner C., et al. Romeo: A Parametric Model-Checker for Petri Nets with Stopwatches // Tools and Algorithms for the Construction and Analysis of Systems (TACAS) (York, UK, March 22–29, 2009). LNCS. Springer, 2009. Vol. 5505. P. 54–57. DOI: 10.1007/978-3-642-00768-2_6
17. Henzinger T. The Theory of Hybrid Automata // Verification of Digital and Hybrid Systems. NATO ASI Series. Springer, 2000. Vol. 170. P. 265–292. DOI: 10.1007/978-3-642-59615-5_1
18. Бусленко Н.П. Моделирование сложных систем. М.: Наука, 1978. 400 с.
19. Andre E. Observer Patterns for Real-Time Systems // Proceedings of the 18th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS) (Singapore, July 17-19, 2013), 2013. P. 125–134. DOI: 10.1109/ICECCS.2013.26
20. Avionics Application Software Standard Interface. ARINC Specification 653 // Aeronautical Radio. Annapolis, 1997.

Алевтина Борисовна Глоина, программист, лаборатория вычислительных комплексов, кафедра автоматизации систем вычислительных комплексов, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

DOI: 10.14529/cmse170404

GENERAL MODEL OF REAL-TIME MODULAR COMPUTER SYSTEMS OPERATION FOR CHECKING ACCEPTABILITY OF SUCH SYSTEMS CONFIGURATIONS

© 2017 A.B. Glonina

Lomonosov Moscow State University (GSP-1, Leninskie Gory 1, Moscow, 119991 Russia)

E-mail: alevtina@vk.cs.msu.su

Received: 06.09.2017

In this paper, we consider schedulability analysis problem for real-time modular computer systems (RT MCS). A system configuration is called schedulable if all the jobs complete within their deadlines. The paper contains formal definitions of a system configuration and a system operation trace which is necessary for checking the schedulability criterion. The author proposes a stopwatch automata-based general model of RT MCS operation. A model instance for a given RT MCS configuration is a network of stopwatch automata and it can be built automatically using the general model. A system operation trace can be obtained from the corresponding NSA trace. The stopwatch automata formalism allowed us to prove that the model is deterministic and satisfies a set of correctness requirements derived from RT MCS specifications. We implemented our approach in software and

compared it experimentally with Model Checking approach, which uses the same formalism. The results of the experiments confirm that our approach is much more efficient. The software implementation is also integrated with the existing industrial RT MCS scheduling tool. These experiments showed the applicability of the proposed approach to real scale systems.

Keywords: modular computer systems, schedulability analysis, stopwatch automata, simulation, real-time systems

FOR CITATION

Glonina A.B. General Model of Real-Time Modular Computer Systems Operation for Checking Acceptability of Such Systems Configurations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 4. pp. 43–59. (in Russian) DOI: 10.14529/cmse170404.

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. Senthilkumar K., Ramadoss R. Designing Multicore ECU Architecture in Vehicle Networks Using AUTOSAR. Proceedings of the Third International Conference on Advanced Computing (ICoAC) (Chennai, India, December 14–16, 2011), 2011. pp. 270–275. DOI: 10.1109/ICoAC.2011.6165187
2. From the Ground Up: How The Internet of Things Will Give Rise to Connected Aviation. Gogo LLC, 2016. 140 p.
3. Marinescu S., Tamas-Selicean D., Acretoaie V., et al. Timing Analysis of Mixed-Criticality Hard Real-Time Applications Implemented on Distributed Partitioned Architectures. Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (Krakow, Poland, September 17–21, 2012), 2012. pp. 1–4. DOI: 10.1109/ETFA.2012.6489720
4. Macariu G., Cretu V. Timed Automata Model for Component-Based Real-Time Systems. Proceedings of the 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS) (Oxford, UK, March 22–26, 2010), 2010. pp. 121–130. DOI: 10.1109/ECBS.2010.20
5. Craveiro J.P., Silveira R.O., Rufino J. HsSim: an Extensible Interoperable Object-Oriented n-Level Hierarchical Scheduling Simulator. Proceedings of the 3rd International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS) (Pisa, Italy, July 10, 2012), 2012. pp. 9–14.
6. Singhoff F., Plantec A., Dissaux P., et al. Investigating the Usability of Real-Time Scheduling Theory with the Cheddar Project. *Real-Time Systems*. 2009. vol. 43, no. 3. pp. 259–295. DOI: 10.1007/s11241-009-9072-y
7. Khoroshilov A., Albitskiy D., Koverninskiy I., et al. AADL-Based Toolset for IMA System Design and Integration. *SAE Int. J. Aerosp.* 2012. vol. 5, no. 2. pp. 294–299. DOI: 10.4271/2012-01-2146
8. Dissaux P., Marc O., Fotsing C., et al. The SMART Project: Multi-Agent Scheduling Simulation of Real-Time Architectures. *Embedded Real Time Software and Systems*. 2014.

9. Balashov V.V., Balakhanov V.A., Kostenko V.A. Scheduling of Computational Tasks in Switched Network-Based IMA Systems. Proceedings of International Conference on Engineering and Applied Sciences Optimization (OPTI) (Athens, Greece, June 4–6, 2014), 2014. pp. 1001–1014.
10. Cassez F., Larsen K. The Impressive Power of Stopwatches. CONCUR 2000 — Concurrency Theory (PA, USA, August 22–25, 2000). LNCS. Springer, 2000. vol. 1877. pp. 138–152. DOI: 10.1007/3-540-44618-4_12
11. Tretyakov A. Automation of Scheduling for Periodic Real-Time Systems. Proceedings of the Institute for System Programming. 2012. vol 22. pp. 375–400. DOI: 10.15514/ISPRAS-2012-22-20
12. Smelyansky R.L. Model of Distributed Computing System Operation with Time. Programming and Computer Software, 2013. vol. 39, no. 5. pp. 233–241. DOI: 10.1134/S0361768813050046
13. Bengtsson J., Yi W. Timed automata: Semantics, Algorithms and Tools. Lectures on Concurrency and Petri Nets. LNCS. Springer, 2004. vol. 3098. pp. 87–124. DOI: 10.1007/978-3-540-27755-2_3
14. Zuberek W.M. Timed Petri Nets Definitions, Properties and Applications. Microelectronics Reliability, 1991. vol. 31, no. 4. pp. 627–644. DOI: 10.1016/0026-2714(91)90007-T
15. Boudjadar A.J., Kim J.H., Larsen K.G., et al. Model Checking Process Algebra of Communicating Resources for Real-time Systems. Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS) (Madrid, Spain, July 8–11, 2014), 2014. pp. 51–60. DOI: 10.1109/ECRTS.2014.24
16. Lime D., Roux O.H., Seidner C., et al. Romeo: A Parametric Model-Checker for Petri Nets with Stopwatches. Tools and Algorithms for the Construction and Analysis of Systems (TACAS) (York, UK, March 22–29, 2009). LNCS. Springer, 2009. vol. 5505, pp. 54–57. DOI: 10.1007/978-3-642-00768-2_6
17. Henzinger T. The Theory of Hybrid Automata. Verification of Digital and Hybrid Systems. NATO ASI Series. Springer, 2000. vol. 170. pp. 265–292. DOI: 10.1007/978-3-642-59615-5_1
18. Buslenko N.P. *Modelirovanie slozhnyh sistem* [Complex Systems Modeling]. Moscow, Nauka, 1978. 400 p.
19. Andre E. Observer Patterns for Real-Time Systems. Proceedings of 18th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS) (Singapore, July 17-19, 2013), 2013. pp. 125–134. DOI: 10.1109/ICECCS.2013.26
20. Avionics Application Software Standard Interface. ARINC Specification 653. Aeronautical Radio. Annapolis, 1997.

РАЗНОМАСШТАБНЫЕ ЗАДАЧИ ТЕПЛОМАССОБМЕНА В АТОМНОЙ ЭНЕРГЕТИКЕ*

© 2017 В.Ю. Волков, Л.А. Голибродо, А.А. Крутиков, О.В. Кудрявцев,
Ю.Н. Надинский, А.Т. Нечаев, А.П. Скибин

АО ОКБ «ГИДРОПРЕСС»

(142103 Подольск, ул. Орджоникидзе, д. 21)

E-mail: vasya-volkov@yandex.ru, luka.golibrodo@gmail.com, krutikov_aa@mail.ru,
ovkudr@yandex.ru, artm26@mail.ru, skibin_ap@grpress.podolsk.ru

Поступила в редакцию: 25.08.2017

Данная статья посвящена обзору результатов, полученных в АО ОКБ «ГИДРОПРЕСС» с учетом наработанного опыта в области численного моделирования теплообмена в различных объектах атомной энергетики. Польза от применения CFD-технологий при проектировании реакторных установок заключается в возможности на базе ограниченного количества экспериментальных данных детально исследовать процессы теплообмена в установке с целью подтверждения или модернизации конструкторских решений на передовом научно-техническом уровне. Представлен ряд задач, для решения которых были использованы современные численные методы вычислительной гидродинамики с применением высокопроизводительной вычислительной техники. Показаны новые возможности расчетного моделирования при использовании современных суперкомпьютерных вычислительных технологий, а также сопутствующие вычислительные сложности и проблемы анализа результатов. Приведены примеры использования рассматриваемой технологии для моделирования экспериментальных стендов и природных объектов при различных режимах работы. Показана автоматическая обработка результатов, позволяющая проводить анализ больших задач размерностью до 1 млрд. контрольных объемов по интегральным параметрам, характеризующим работу реакторной установки, таким как распределение расходов на входе и на выходе из активной зоны, распределение подогревов в тепловыделяющих сборках активной зоны, и т.д.

Ключевые слова: атомная энергетика, вычислительная гидродинамика, суперкомпьютерные технологии, теплообмен.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Волков В.Ю., Голибродо Л.А., Крутиков А.А., Кудрявцев О.В., Надинский Ю.Н., Нечаев А.Т., Скибин А.П. Разномасштабные задачи теплообмена в атомной энергетике // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 60–73. DOI: 10.14529/cmse170405.

Введение

За время развития атомной отрасли были использованы различные экспериментальные и расчетные методы изучения течения и теплообмена теплоносителя в установках: масштабные и полномасштабные экспериментальные стенды, данные с действующих блоков, расчеты с помощью системных и ячейковых кодов. Однако каждый из приведенных способов исследования имеет свои особенности, например при создании масштабных стендов практически невозможно добиться соответствия по всем критериям подобия, учесть все конструктивные детали реальной установки и смоделировать натурные теплотехнические условия (давление, температура). С другой стороны данные

* Статья рекомендована к публикации программным комитетом научной конференции «Суперкомпьютерные дни в России – 2017».

с действующих блоков жестко ограничены проектом по количеству и месторасположению измерительной аппаратуры.

Результаты в области применения численного моделирования для исследования теплообмена, достигнутые за последние годы в ОКБ «ГИДРОПРЕСС», связаны с применением и развитием высокопроизводительных суперкомпьютеров (кластерных систем). В связи с ростом компьютерных ресурсов за последние 15 лет одним из главных инструментов по расчетному изучению теплообмена для однофазного течения стала вычислительная гидродинамика (CFD). Польза от применения CFD-технологий при проектировании реакторных установок (РУ) заключается в возможности на базе ограниченного количества экспериментальных данных детально исследовать процессы теплообмена в установке с целью подтверждения или модернизации конструкторских решений на передовом научно-техническом уровне. Фактически, CFD-модель представляет собой виртуальный аналог моделируемого объекта исследования (экспериментальный стенд, реактор и т.д.). Важно также отметить, что применение CFD-технологий на этапе планирования эксперимента позволяет более эффективно проводить физическое моделирование. Однако стоит отметить, что временные затраты на получение численного решения, стоимость расчета и наличие нормативных ограничений на применение вычислительной гидродинамики в атомной отрасли, ограничивают использование CFD-технологии для расчетного обоснования РУ.

В настоящее время типичная размерность сеточной модели для большинства реакторных задач атомной энергетики составляет более 200 млн. расчетных ячеек, для решения которых требуется проведение расчетов на суперкомпьютерах с количеством ядер от нескольких сотен до нескольких тысяч (указаны основные ресурсы, используемые в ОКБ «ГИДРОПРЕСС»). Для моделирования процессов гидродинамики и теплообмена, в основном, используются $k-\varepsilon$ и $k-\omega$ модели турбулентности, иногда используется масштабно-адаптивная модель (SAS), а в ряде случаев моделирование проводится при помощи метода крупных вихрей (LES).

В данной статье приведены некоторые решенные в ОКБ «ГИДРОПРЕСС» ресурсоемкие задачи, требующие высокопроизводительной вычислительной техники, для решения которых применялись CFD-технологии.

В настоящее время в ОКБ «ГИДРОПРЕСС» CFD-технологии применяются для подтверждения имеющихся технических решений, и оптимизации элементов конструкции. Это позволяет значительно сократить сроки проведения конструкторских работ, а также выявить и устранить ошибки проектирования на ранних стадиях (на стадии разработки технического проекта РУ). Ниже представлены характерные задачи проектирования и задачи вычислительной гидродинамики и теплообмена.

В разделе 1 представлены особенности моделирования теплообмена в пучках твэл и ТВС. В разделе 2 на примере расчета парогенератора представлено применение CFD в поддержку проектирования оборудования РУ. Раздел 3 посвящен особенностям полномасштабного моделирования РУ ВВЭР с учетом всех особенностей проточного тракта. В разделе 4 показаны особенности пост-процессинга и обработки задач размерностью порядка миллиарда контрольных объемов для анализа интегральных результатов расчета. Раздел 5 посвящен анализу эффективности распараллеливания для практических задач. Выводы по статье изложены в заключении.

1. Математическое моделирование течения в пучках ТВЭЛ и ТВС

С целью дальнейшего повышения мощности РУ с ВВЭР (водо-водяной энергетический реактор) внедряются различные новые конструктивные решения, например, интенсификаторы теплообмена в тепловыделяющих сборках (ТВС). Они приводят к усложнению структуры потоков в активной зоне и, следовательно, повышают требования к расчетам локальных параметров потока.

Современные вычислительные технологии позволяют моделировать структуру потока в пучках стержней с перемешивающими устройствами с высоким уровнем детализации. Однако, применение CFD-кодов к этому классу течений все еще весьма ограничено из-за сложной структуры течения в дистанционирующих и перемешивающих решетки, анизотропии турбулентности в пучках ТВЭЛ и большой размерности задач.

В рамках международной работы по верификации CFD-моделей и разработке расчетных методик в 2011-2012 гг. OECD (организация экономического сотрудничества и развития) проводился бенчмарк OECD/NEA-MATiS-H BENCHMARK [1]. Целью данной задачи было подробное моделирование структуры турбулентного потока после перемешивающих устройств с применением CFD-кодов. Результаты моделирования сравнивались с экспериментальными данными для нескольких сечений после перемешивающей решетки, полученными методом LDV (лазерная доплеровская анемометрия).

Экспериментальный участок стенда MATiS-H состоит из пучка стержней 5x5, установленного в горизонтальном положении, внешний диаметр стержня составляет 25,4 мм. Сетка расчетной области, полученная соединением сеток для всех областей вместе, состоит из 13 млн. узлов и 22,5 млн. элементов. Для моделирования использовался программный комплекс CFX 14. Фрагменты сетки представлены на рис. 1. Результаты по моделированию пульсаций скорости (интенсивности турбулентности) представлены на рис. 2. Сравнение полученных численных результатов с экспериментальными данными показало, что результаты моделирования хорошо согласуются с экспериментом.

Конструкция ТВС РУ ВВЭР состоит из следующих частей: головки, пучка из 312 ТВЭЛОВ (тепловыделяющих элементов) и ТВЭГОВ (тепловыделяющих элементов с уран-гадолиниевым топливом), хвостовика. Каркас ТВС собран из 18 направляющих каналов, 13 дистанционирующих решеток, инструментального канала и нижней решетки. При этом высота ТВС более 4,5 метров, а шаг между ТВЭЛАМИ (ТВЭГАМИ) 12,75 мм. Таким образом, при моделировании процессов теплообмена в ТВС, при построении сетки необходимо разрешать миллиметровые зазоры на масштабе нескольких метров.

В ОКБ «ГИДРОПРЕСС» разработана модель ТВС, размерностью около 1 млрд. расчетных ячеек. При этом для получения стабильного численного решения используются двухпараметрические $k-\varepsilon$ и $k-\omega$ модели турбулентности. Применение подобных моделей приводит к тому, что в численном расчете температура стенки может быть несколько завышена относительно реального значения. Применение двухпараметрических моделей турбулентности позволяет проводить расчеты на сеточных моделях с соотношением сторон до 1:10. Применение моделей рейнольдсовых напряжений или моделей крупных вихрей позволяет изучить детальнее структуру потока с наличием вторичных течений в ТВС. Однако для применения данных моделей требуются изотропные сетки с соотношением сторон близким 1:1, а также более детальное измельчение сетки, что приведет к увеличению

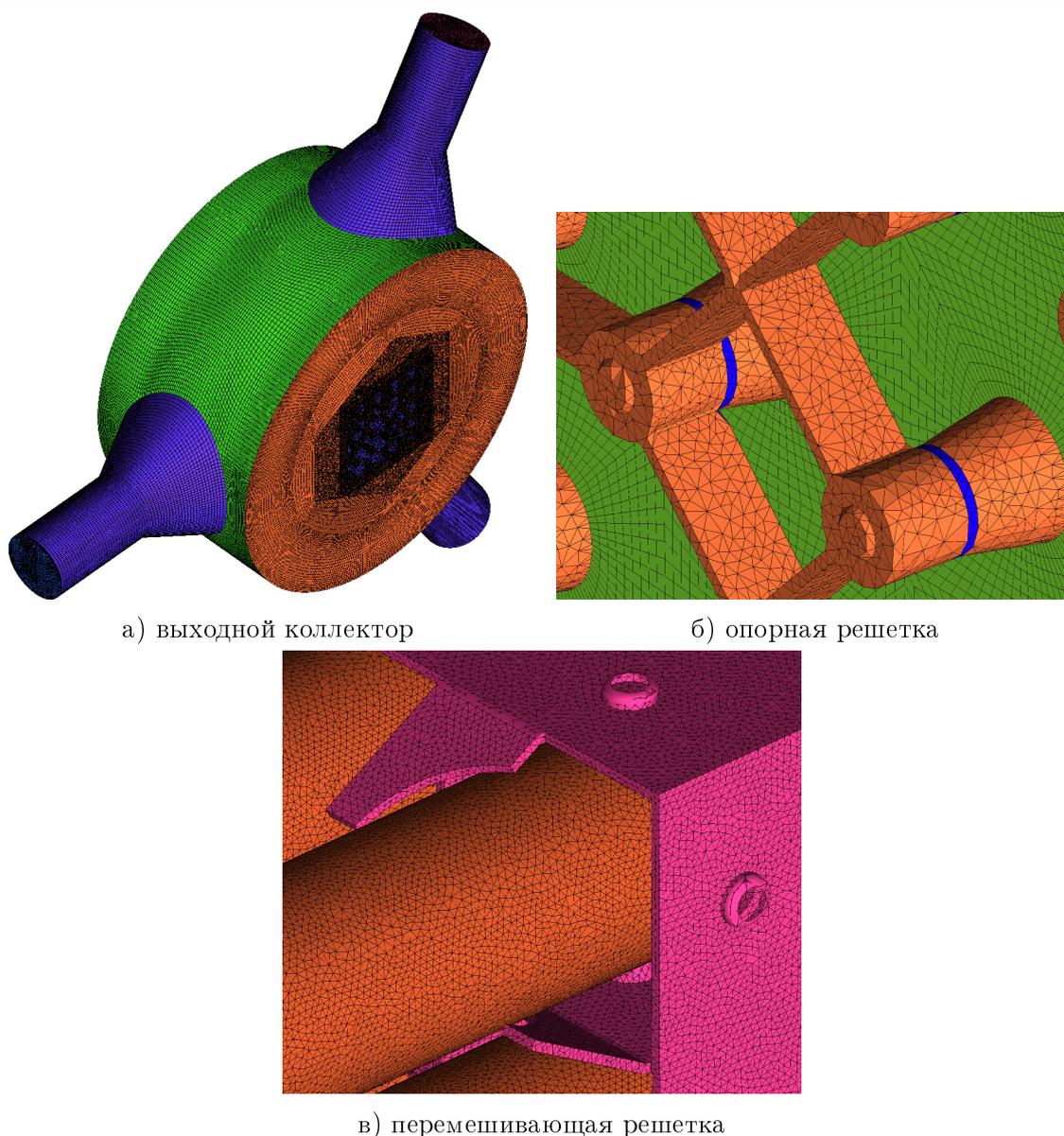


Рис. 1. Фрагменты расчетной сетки

количества расчетных ячеек на несколько порядков. Такой подход применяется для изучения процессов теплообмена в отдельных элементах ТВС.

В ОКБ «ГИДРОПРЕСС» подобные расчеты выполняются с применением суперкомпьютерных вычислительных систем, одна из которых насчитывает 40 вычислительных узлов. На каждом узле установлено 2 процессора Intel Xeon E5-2698 v4 2.2 Гц и 128 Мб оперативной памяти. Общее количество вычислительных ядер — 1600, общее количество оперативной памяти — 5 Тб. Характерное время счета 5000 итераций модели ТВС составляет около 5 суток. Занимаемый объем оперативной памяти — 3 Тб. Следует отметить, что расчетная сетка модели ТВС в области дистанционирующих решеток состоит из многогранных ячеек с выделением призматических слоев вблизи стенок расчетной области. В области твэльного пучка, вдали от дистанционирующих решеток, расчетная сетка состоит из призматических слоев, полученных методом экстрезии.

Пример определения гидравлического сопротивления дистанционирующих решеток по длине ТВС в безразмерных координатах представлен на рис. 3.

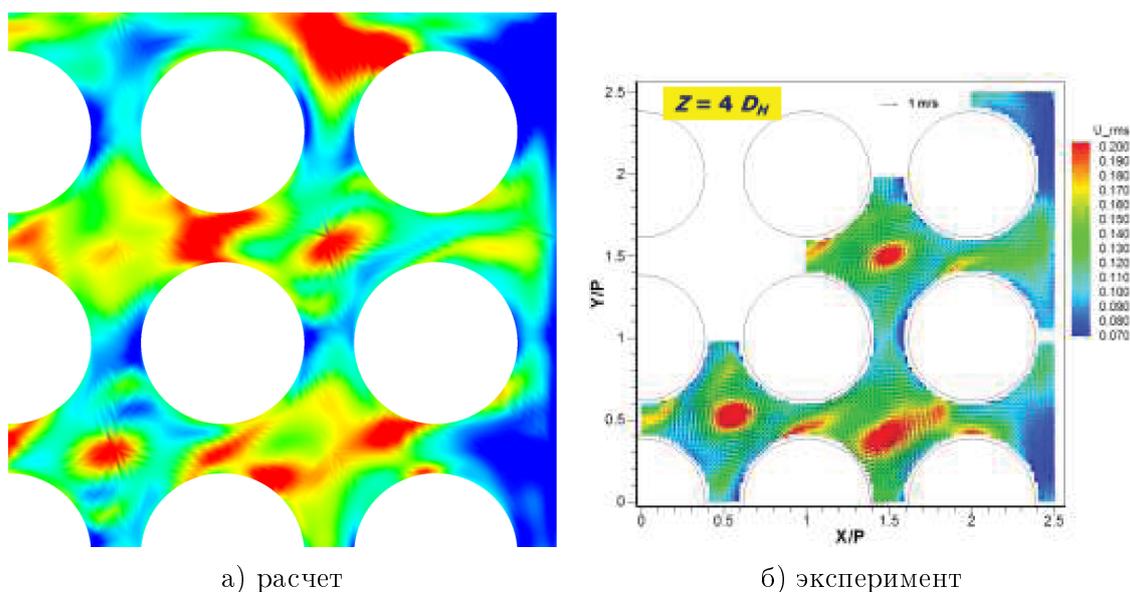


Рис. 2. Поля пульсаций продольной компоненты скорости в сечении за решеткой

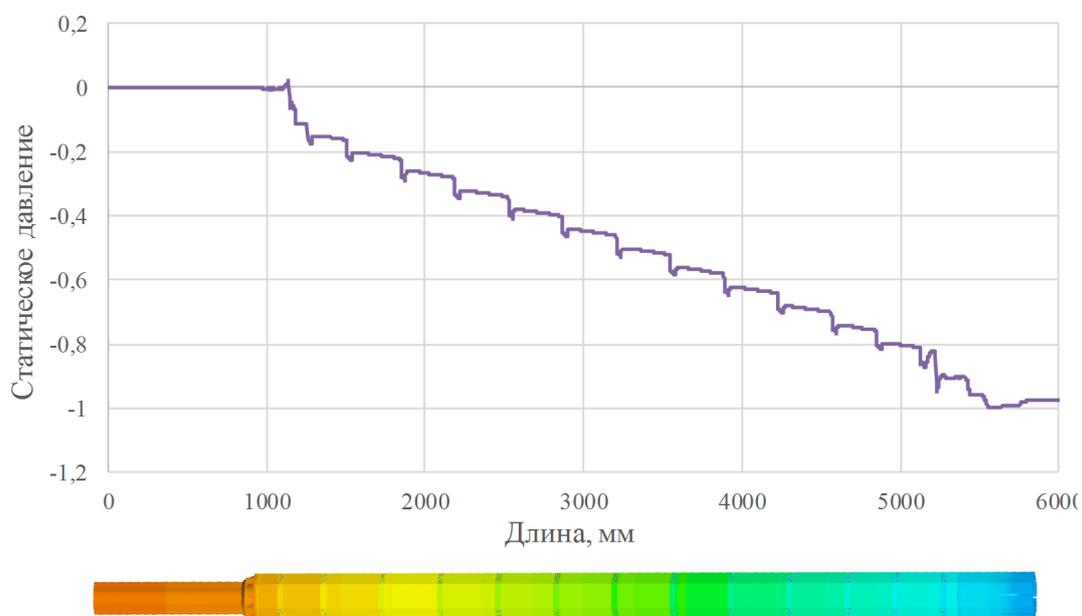
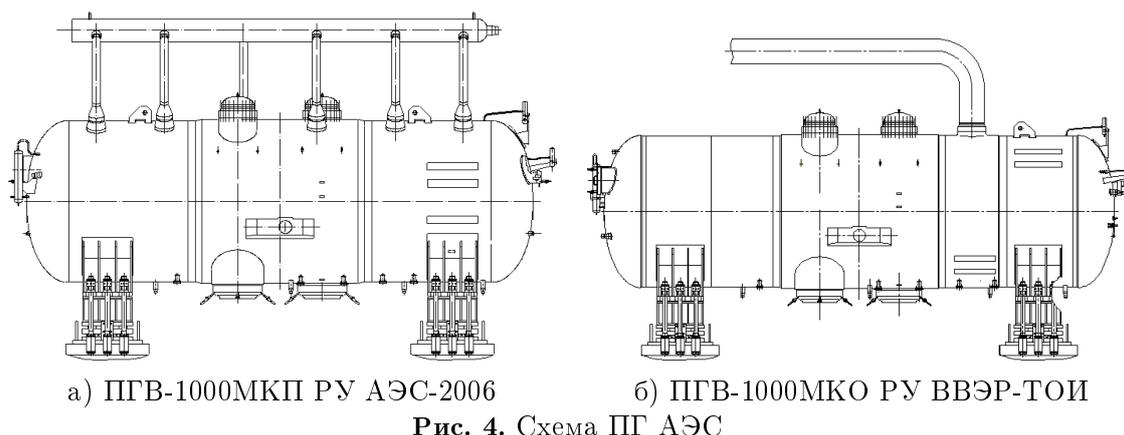


Рис. 3. Изменение статического давления по длине ТВС (значения приведены в нормированном виде)

2. Математическое моделирование процессов теплообмена в оборудовании РУ

Парогенератор ПГВ-1000МКО, разработанный в рамках проекта ВВЭР-ТОИ, имеет референтную конструкцию по отношению к проектам АЭС-2006 и серийной установке с реактором ВВЭР-1000 за исключением системы отвода пара из парогенератора. Отличие заключается в количестве патрубков для отвода пара, один в рассматриваемом ПГВ-1000МКО против десяти в референтных проектах. Данное конструктивное решение продиктовано особенностями компоновки РУ. Схема ПГ референтной конструкции и разработанного ПГВ-1000МКО представлены на рис. 4а и 4б соответственно [2–4].



С целью определения степени перфорации и параметров отверстий в пароприемных дырчатых листах (ППДЛ) ПГ, обеспечивающих снижение неравномерности поля скорости пара в паровом пространстве между погружными дырчатыми листами (ПДЛ) и ППДЛ до приемлемого уровня, был выполнен комплекс оптимизационных расчетов ППДЛ парогенератора ПГВ-1000МКО с применением CFD. В рамках работ по оптимизации ППДЛ была разработана компьютерная модель парового объема парогенератора ПГВ-1000МКО с использованием программных комплексов STAR-CCM+ и STAR-CD [5, 6], которая предусматривает возможность изменения степени перфорации, толщины листов ППДЛ и распределения паровой нагрузки на входе в паровой объем. Общая размерность модели превышает 200 млн. расчетных ячеек.

В результате расчетных исследований (рис. 5) была проведена оптимизация конструкции ППДЛ ПГВ-1000МКО с применением CFD методов и даны рекомендации по степени перфорации и параметрам отверстий в листах ППДЛ.

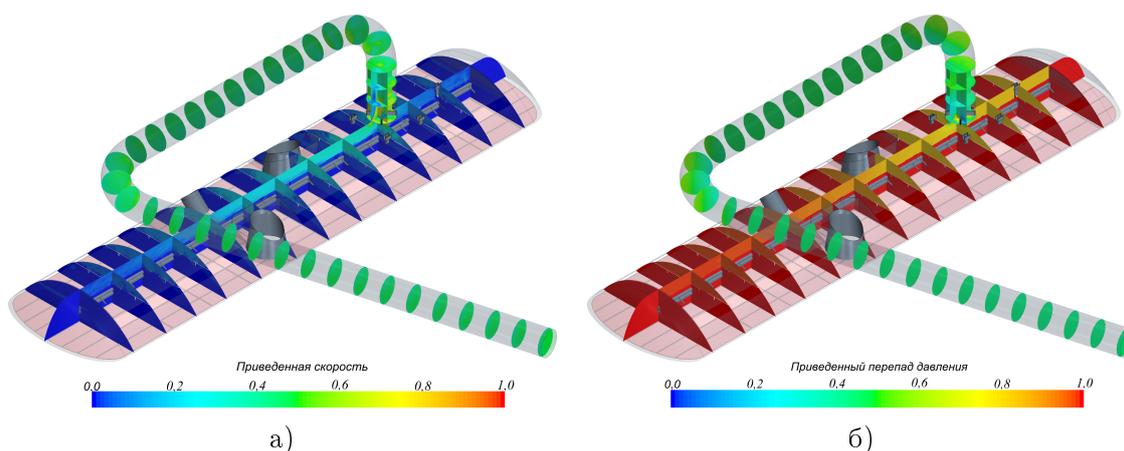


Рис. 5. Распределение модуля вектора скорости пара (а) и давления пара (б) в паровом объеме ПГ

Характерное время счета 5000 итераций модели парового объема ПГВ-1000МКО составляет около 1 суток (на вычислительной системе, насчитывающей 1600 ядер). Занимаемый объем оперативной памяти — 0,5 Тб. Следует отметить, что расчетная сетка модели парового объема ПГВ-1000МКО в области ППДЛ состоит из гексаэдрических ячеек с выделением призматических слоев вблизи стенок расчетной области. В области парового объема над и под ППДЛ, расчетная сетка состоит из многогранных ячеек.

Представленное расчетное исследование позволило усовершенствовать конструкцию пароприемного устройства до начала сепарационных испытаний парогенератора на блоке АЭС. Теперь нет необходимости изготовления большого количества перфорированных листов различной степени перфорации, их монтажа, подбора и т.д. По сути, такой подход приводит к значительному сокращению объема и времени пусконаладочных работ.

3. Математическое моделирование процессов теплообмена в РУ ВВЭР

Течение теплоносителя в реакторе сопровождается наличием трехмерных явлений, которые не могут быть в полной мере учтены при помощи одномерных инженерных подходов к численному моделированию. Особенности оборудования АЭС является сложная геометрия проточного тракта, при этом линейные размеры отличаются на несколько порядков. Общий вид реакторной установки проекта АЭС-2006 поколения 3+ с реактором ВВЭР-1200 представлен на рис. 6.

В качестве первоочередной задачи поддержки проектов РУ ВВЭР определено полномасштабное моделирование процессов смешения теплоносителя в верхней и нижней камерах реактора и процессов теплопередачи в ТВС, активной зоне и реакторе в целом. Результат проведенных расчетных исследований показывает необходимость учета всех элементов конструкции вследствие их влияния на процесс теплообмена [7, 8].

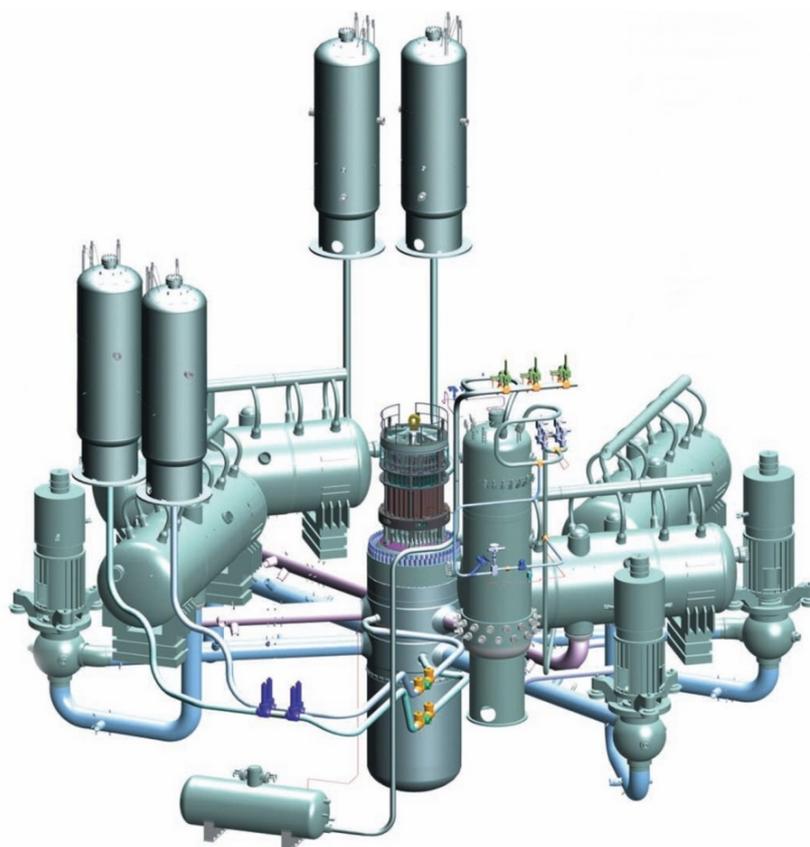


Рис. 6. Общий вид реакторной установки проекта АЭС-2006

Расчеты с помощью CFD позволяют получить ряд теплогидравлических параметров, характеризующих работу реактора, таких как распределение расходов

по тепловыделяющим сборкам на входе и на выходе из активной зоны, распределение подогревов в тепловыделяющих сборках активной зоны, коэффициенты гидравлического сопротивления элементов проточного тракта реактора и т.д. Проведено моделирование стационарных теплогидравлических процессов и сравнение с экспериментальными данными.

В результате работы создана CFD модель проточной части реактора ВВЭР-1200, позволяющая определять трехмерные распределения параметров теплоносителя при стационарных режимах работы реакторной установки. Общая сборка CFD модели и расчеты выполнены с использованием кода STAR-CCM+. Размерность сеточной модели составила порядка 700 миллионов ячеек.

Характерное время счета 5000 итераций модели реактора ВВЭР-1200 составляет около 4 суток (на вычислительной системе, насчитывающей 1600 ядер). Занимаемый объем оперативной памяти — 2 Тб. Следует отметить, что расчетная сетка модели реактора ВВЭР-1200 в области опор в нижней камере смешения и в области активной зоны состоит из гексаэдрических ячеек. В области входных и выходных патрубков, опускного участка и сборной камеры реактора, расчетная сетка состоит из многогранных ячеек.

На рис. 7 приведено распределение расходов через тепловыделяющие сборки на входе в активную зону, отнесенные к среднему расходу. Отклонение расчетных расходов от экспериментальных данных, не превышает 8%.

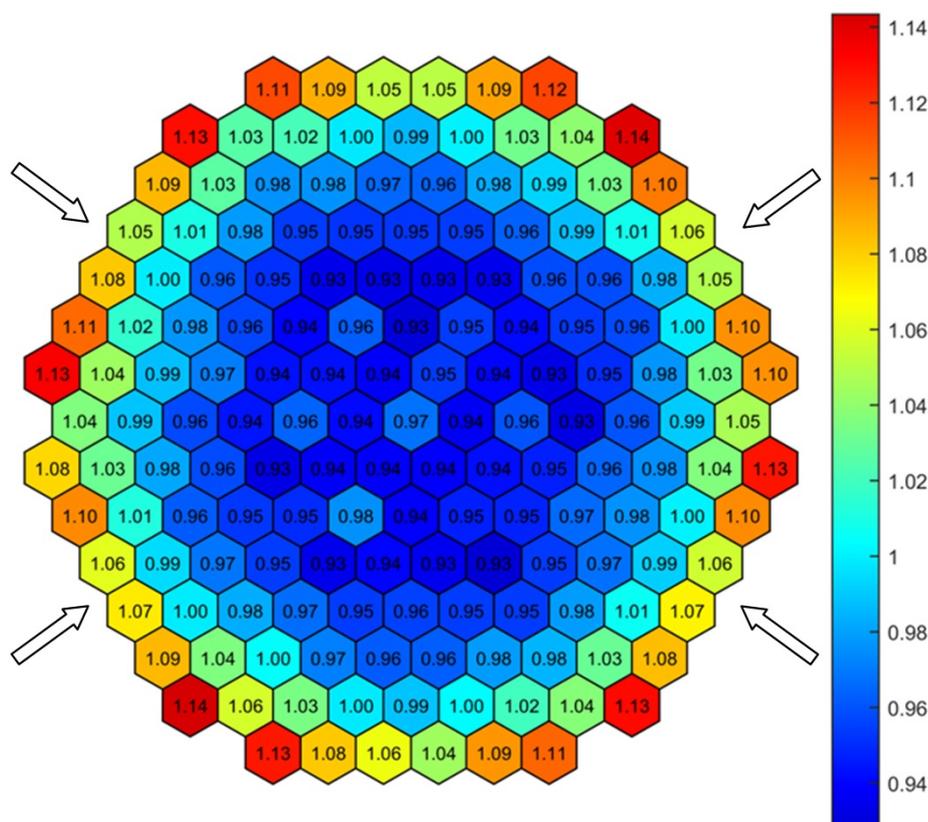


Рис. 7. Относительное осредненное по опорам ТВС распределение массового расхода на входе в активную зону

4. Обработка и анализ результатов CFD моделей большой размерности

Особенностью создаваемых CFD моделей является сложная геометрия и топология расчетной области, и как следствие большая размерность сеточной модели (порядка миллиарда контрольных объемов), что приводит к большому размеру файлов с расчетной моделью и результатами расчетов (сотни гигабайт). Таким образом, обратной стороной увеличения размерности и сложности CFD-моделей является возрастание сложности их обработки с целью последующего представления и анализа результатов.

Одним из способов решения обозначенной проблемы является уменьшение размерности расчетной модели в рамках создания информационных репрезентационных нодальных моделей, на которых проводится интегральное осреднение. Эти модели создаются на базе CFD-расчета и, следовательно, обладают такой же степенью достоверности по интегральным параметрам (распределение расходов, перепады давления, подогревы и др.). Вследствие уменьшения размерности нодальной модели по сравнению с исходной CFD-моделью, репрезентационная модель не отображает локальных распределений характеристик потока, что, однако несущественно при анализе процессов в рамках работы всей установки.

Для переноса данных с CFD-моделей на репрезентационные также требуются высокопроизводительные вычисления для осреднения результатов многомиллионных сеток, чтобы получить интегральные значения основных величин в объемах и по граням нодов репрезентационной модели. При этом стоит отметить, что процедура переинтерполяции данных с сетки на сетку стандартными средствами CFD кодов или не возможна или может приводить к значительным численным погрешностям нодальной модели.

При разработке репрезентационных моделей необходимо тщательно выбирать сеточную топологию, так как данная модель используется отдельно от CFD-модели и вместе с тем должна адекватно представлять проточный тракт и все необходимые теплогидравлические характеристики реального объекта. В тоже время для уменьшения количества операций интегрирования размерность репрезентационных моделей не следует завышать.

Основанная на результатах CFD расчета репрезентационная модель позволяет представлять теплообмен в рассматриваемой расчетной области, например в первом контуре РУ. Помимо всего подобные модели позволяют наглядно визуализировать расчетные данные, полученные при использовании различных теплогидравлических системных кодов, а также качественно и количественно сопоставлять друг с другом расчеты, полученные с помощью различных расчетных платформ.

Пример разработки подобной репрезентационной модели для РУ ВВЭР приведен на рис. 8. В данном случае размерность расчетной сетки для CFD-модели составила примерно 700 миллионов ячеек, а размерность для репрезентационной модели составила около 25 тысяч ячеек.

5. Оценка эффективности распараллеливания при моделировании процессов теплообмена

В заключение статьи приведен тест эффективности распараллеливания кода StarCCM+ 11.06 на задаче размерностью 1 млн. ячеек. Расчеты проводились на узлах вычислительных систем, построенных на процессорах Intel Xeon E5-2670 2.6 Гц (8 ядер) и на процессорах

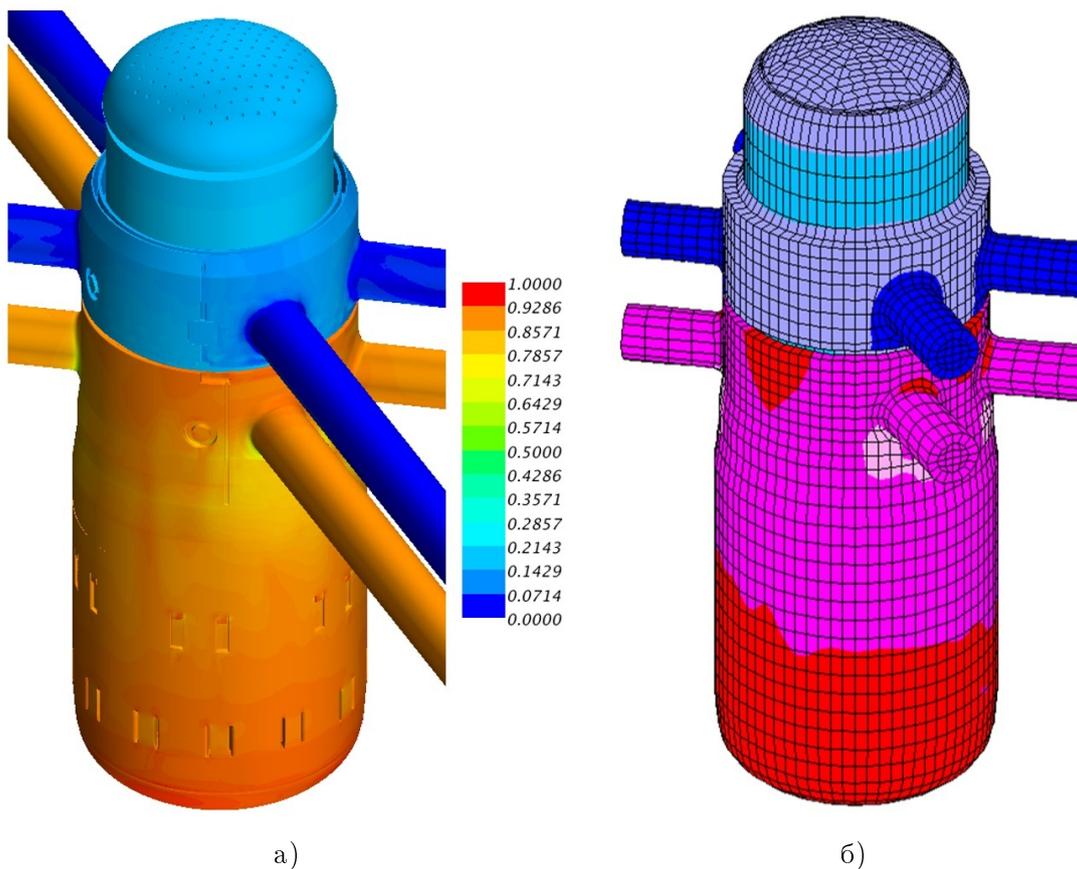


Рис. 8. CFD расчет (поле давления): а) - CFD модель; б) - репрезентационная модель

Intel Xeon E5-2698 v4 2.2 Гц (20 ядер). Каждый узел насчитывает 2 процессора. Количество контрольных объемов, приходящихся на одно вычислительное ядро превышает 25000 (производитель кода заявляет, что эффективность распараллеливания в версии StarCCM+ 11.06 сохраняется вплоть до нагрузки 10000 контрольных объемов на вычислительное ядро). На рис. 9 приведены зависимости коэффициента ускорения от количества ядер для двух систем, и сравнение их с линейным ростом производительности. По результатам тестирования можно сделать следующее заключение:

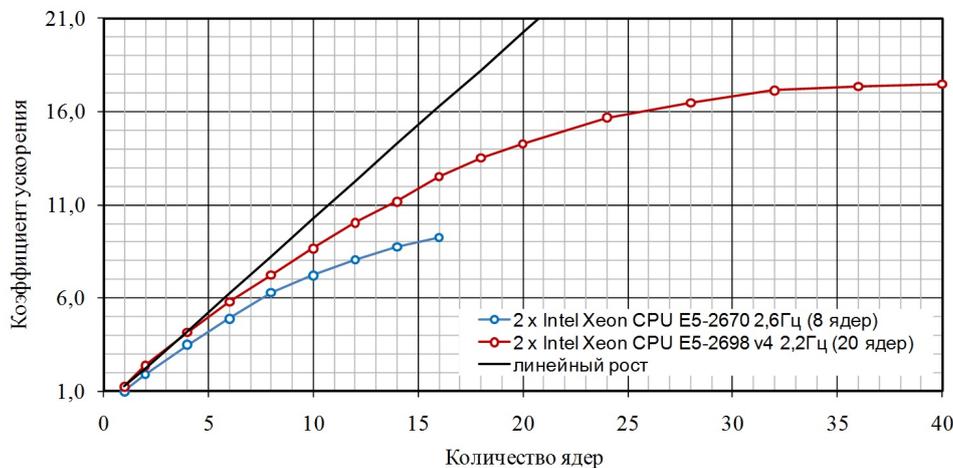


Рис. 9. Эффективность распараллеливания кода StarCCM+ 11.06

1. не для всех задач вычислительной гидродинамики возможно получить идеальную (линейную) эффективность распараллеливания;
2. вычислительное ядро процессора нового поколения Intel Xeon E5-2698 v4 2.2 Гц на 25% быстрее по сравнению с ядром процессора Intel Xeon E5-2670 2.6 Гц;
3. узел построенный на двух процессорах Intel Xeon E5-2698 v4 2.2Гц (40 ядер) примерно в два раза быстрее по сравнению с узлом, построенным на двух процессорах Intel Xeon E5-2670 2.6 Гц (16 ядер);
4. не смотря на большую эффективность и скорость счета, для узла, построенного на двух процессорах Intel Xeon E5-2698 v4 2.2Гц (40 ядер), рост производительности резко замедляется на количестве ядер 20–30 и практически отсутствует на количестве ядер 30–40.

Одновременно с ростом производительности новых вычислительных систем требуется и улучшение алгоритмов распараллеливания, что позволит более эффективно, быстро и дешево моделировать объекты атомной отрасли.

Заключение

При решении новых задач и разработке проектов, для которых еще не накоплены экспериментальные данные и промышленно-эксплуатационный опыт, применение суперкомпьютерных вычислений может оказаться разумной альтернативой экспериментальному изучению гидродинамических процессов и тепломассообмена в оборудовании РУ. Численное моделирование позволяет получить результат не по прошествии времени создания АЭС и отдельного оборудования, а еще на стадии технического проекта. При этом погрешности, которые имеют место при численном моделировании, сравнимы с погрешностью экспериментальных исследований и эмпирических корреляций.

Эффективность распараллеливания на реальных задачах существенно отличается от линейного роста производительности. Перед проведением серии расчетов необходимо проверять быстродействие применяемого кода на используемой суперкомпьютерной установке с целью определения количества используемых ядер на каждом вычислительном узле.

Применение суперкомпьютерных технологий для проведения разномасштабного моделирования течения теплоносителя в оборудовании реакторной установки (реактор, парогенераторы, системы первого и второго контуров) позволяет судить о перспективе использования CFD-кодов в атомной энергетике как повседневного расчетного инструмента, сокращающего длительность и стоимость работ по конструированию оборудования реакторной установки, а также повышающего качество их выполнения.

Литература

1. Голибродо Л.А., Вдовкина И.К., Галкин И.Ю. и др. Валидация трехмерных моделей гидродинамических процессов в теплообменном оборудовании // Главный механик – 2014. № 8. С. 23–25.
2. Скибин А.П., Волков В.Ю., Голибродо Л.А. и др. Оптимизация пароприемного устройства парогенератора реакторной установки. Высокопроизводительные вычисления на страже качества конструирования оборудования реакторной установки // Суперкомпьютерные технологии в науке, образовании и промышленности / под ред. акад. РАН В.А. Садовниченко, акад. РАН Г.И. Савина, чл.-корр. РАН Вл.В. Воеводина.

М. : Издательство Московского университета, 2015. С. 6–12.

3. Волков В.Ю., Голибродо Л.А., Крутиков А.А. и др. Оптимизация пароприемного дырчатого листа парогенератора ПГВ-1000МКО с применением CFD // Обеспечение безопасности АЭС с ВВЭР: Сборник трудов 9-й международной научно-технической конференции (Подольск, 19 –22 мая 2015 г.). Подольск, 2015.
4. Волков В.Ю., Голибродо Л.А., Крутиков А.А. и др. Оптимизация пароприемного дырчатого листа парогенератора ПГВ-1000 МКО с применением CFD // Бюллетень основных научно-технических работ АО ОКБ «ГИДРОПРЕСС» за 2014 год. Подольск: АО ОКБ «ГИДРОПРЕСС», 2016.
5. SIEMENS. USER GUIDE STAR-CCM+ Version 11.06, 2016. 10082 p.
6. Methodology STAR-CD Version 4.16 CD-adapco. CD-adapco Group, 2009.
7. Логвинов С.А., Безруков Ю.А., Драгунов Ю.Г. Экспериментальное обоснование теплогидравлической надежности реакторов ВВЭР. М.: ИКЦ «Академкнига», 2004. 255 с.
8. Волков В.Ю., Голибродо Л.А., Крутиков А.А. и др. Разработка CFD модели реактора проекта АЭС-2006 // Суперкомпьютерные дни в России: Труды международной научной конференции (Москва, 26 –27 сентября 2016 г.). Москва, 2016. С. 556–564.

Василий Юрьевич Волков, к.т.н, инженер-конструктор 1 категории АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Лука Антонович Голибродо, к.т.н, инженер-конструктор 1 категории АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Алексей Александрович Крутиков, к.т.н, инженер-конструктор 1 категории АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Олег Вячеславович Кудрявцев, инженер-конструктор 1 категории АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Юрий Николаевич Надинский, инженер-конструктор 1 категории АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Артем Тимурович Нечаев, инженер-конструктор АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

Александр Петрович Скибин, к.т.н, начальник группы АО ОКБ «ГИДРОПРЕСС» (Подольск, Российская Федерация)

MULTI-SCALE PROBLEMS OF HEAT AND MASS TRANSFER IN NUCLEAR INDUSTRY

© 2017 V. Yu. Volkov, L.A. Golibrodo, A.A. Krutikov, O.V. Kudryavtsev,
Yu.N. Nadinsky, A.T. Nechaev, A.P. Skibin

OKB «GIDROPRESS»

(142103 21 Ordzhonikidze st, Podolsk)

E-mail: vasya-volkov@yandex.ru, luka.golibrodo@gmail.com, krutikov_aa@mail.ru,
ovkudr@yandex.ru, artm26@mail.ru, skibin_ap@grpress.podolsk.ru

Received: 25.08.2017

This paper presents a review of the results obtained in OKB «GIDROPRESS» taking into account the experience gained in the field of numerical modeling of heat and mass transfer in various nuclear power facilities. The benefits of using CFD technology in the NPP design is the ability, based on a limited number of experimental data, to study in detail the heat and mass transfer processes in the facilities in order to confirm or improve design solutions at the advanced scientific and technical level. A number of problems are presented, for solution of which the modern numerical methods of computational hydrodynamics with high-performance computer technology were used. New possibilities of computational modeling with the use of modern supercomputer computing technologies, as well as accompanying computational complexity and difficulties of analysis of results are shown. Examples of use of the considered technology for modeling experimental rigs and full-scale objects under different operating conditions are given. The automatic post-processing of the results is shown. The procedure allows to analyze large problems with mesh size up to 1 billion control volumes by integral parameters, such as mass flow distribution at the core inlet and outlet, distribution of coolant heat-up over the core fuel assemblies, etc.

Keywords: nuclear energy, computational fluid dynamics, supercomputer technologies, heat transfer.

FOR CITATION

Volkov V.Yu., Golibrodo L.A., Krutikov A.A., Kudryavtsev O.V., Nadinsky Yu.N., Nechaev A.T., Skibin A.P. Multi-scale Problems of Heat and Mass Transfer in Nuclear Industry. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no. 4. pp. 60–73. (in Russian) DOI: 10.14529/cmse170405.

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. Golibrodo L., Vdovkina I., Galkin I., et al. Validation of Three-dimensional Models of Hydrodynamic Processes in Heat-exchange Equipment. *Glavniy mehanik* [Chief Mechanical Engineer]. 2014. No 8. pp. 23–25. (in Russian)
2. Skibin A., Volkov V., Golibrodo L. et al. Optimization of the Steam Receiving Device of the NPP Steam Generator. High Performance Computing for the Quality of Design of Reactor Equipment *Superkomputernie tehnologii v nauke, obrazovanii i promishlennosti* [Supercomputer Technologies in Science, Education and Industry / Ed. by Acad. RAS V. Sadovnichiy, Acad. RAS G. Savin, Acad. RAS V. Voevodin]. M.: Publishing house of Moscow University, 2015. pp. 6–12. (in Russian)
3. Volkov V., Golibrodo L., Krutikov A. et al. Optimization of the Steam Receiving

- Perforated Sheet of the PGV-1000MKO Steam Generator Using CFD. *Sbornik trudov 9 mezhdunarodnoi konferencii "Obespechenie bezopasnosti AES s VVER"* [Proceedings of the 9th International Scientific and Technical Conference «Safety of NPPs with VVER»]. Podolsk, 2015. (in Russian)
4. Volkov V., Golibrodo L., Krutikov A. et al. Optimization of the Steam Receiving Perforated Sheet of the Steam Generator PGV-1000 MKO Using CFD. *Bulluten osnovnih nauchno-thenicheskikh rabot AO OKB "GIDROPRESS"* [Bulletin of the main scientific and technical works of JSC OKB "GIDROPRESS"for the year]. 2014. Podolsk: JSC OKB "GIDROPRESS 2016. (in Russian)
 5. SIEMENS. USER GUIDE STAR-CCM+ Version 11.06, 2016. 10082 p.
 6. Methodology STAR-CD Version 4.16 CD-adapco. CD-adapco Group, 2009.
 7. Logvinov S., Bezrukov Yu., Dragunov Yu. *Experimentalnoe obosnovanie teplogidravlicheskoj nadegnosti reaktorov VVER* [Experimental substantiation of thermal and hydraulic reliability of VVER reactors]. M.: IKTs Akademkniga, 2004. 255 p.
 8. Volkov V., Golibrodo L., Krutikov A. et al. Development of the CFD model of the reactor of the NPP-2006 project // Supercomputernie dni v Rossii: Trudy meghdunarodnoj konferentsii (Moscv, 26 –27 sentyabra 2016) [Russian Supercomputing Days: Proceedings of the International Scintific Conference (Moscow, Russia, September, 26 –27, 2016). Moscow, 2016, pp. 556–564. (in Russian)

ГИБРИДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ КЛАСТЕРЫ ДЛЯ ИЗУЧЕНИЯ СТРУКТУРЫ, ФУНКЦИИ И РЕГУЛЯЦИИ БЕЛКОВ*

© 2017 Д.А. Суплатов, Н.Н. Попова, К.Е. Копылов, М.В. Шегай,
Вл.В. Воеводин, В.К. Швядас

*Московский государственный университет имени М.В. Ломоносова
(119991 Москва, Ленинские Горы, д. 1)*

*E-mail: d.a.suplatov@belozersky.msu.ru, popova@cs.msu.ru, kopylov@mail.chem.msu.ru,
max.shegai@gmail.com, voevodin@parallel.ru, vyta@belozersky.msu.ru*

Поступила в редакцию: 11.09.2017

Изучение структуры, функции и регуляции белков с использованием биоинформатики и молекулярного моделирования является комплексной задачей, требующей сочетания различных методов и способов их исполнения. На практике, речь идет о конвейере из последовательных этапов, исполняемых различными программами, предъявляющими свои требования к вычислительным ресурсам. Гибридные вычислительные кластеры — системы, обладающие существенной мощностью и разнообразием аппаратных возможностей — необходимы для того, чтобы оптимально исполнить каждую отдельную стадию единого комплексного решения. При этом GPU-ускорители открывают новые возможности для поиска эффективных решений ресурсоемких задач биоинформатики и молекулярного моделирования.

Ключевые слова: гибридные вычислительные кластеры, биоинформатика, молекулярное моделирование, последовательные этапы, кодизайн, GPU-ускорители.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Суплатов Д.А., Попова Н.Н., Копылов К.Е., Шегай М.В., Воеводин Вл.В., Швядас В.К. Гибридные вычислительные кластеры для изучения структуры, функции и регуляции белков // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 74–90. DOI: 10.14529/cmse170406.

Введение

Взаимодействие белков с небольшими молекулами (лигандами) является одним из ключевых процессов в биологии. На настоящий момент наиболее хорошо изучены механизмы действия лигандов, которые связываются в активных центрах ферментов, однако роль других карманов и полостей на поверхности белков изучены недостаточно. В последние годы появляется все больше свидетельств явления аллостерии — регуляции функции белков посредством связывания низкомолекулярных лигандов в специализированных регуляторных центрах на поверхности [1]. На рис. 1 показана структура одного из ферментов гликолиза — важнейшего метаболического пути клетки. Поверхность этого большого и сложно устроенного белка покрыта большим количеством полостей — потенциальных сайтов связывания. Среди них можно увидеть как активный центр, выполняющий главную функцию этого фермента, так и известный аллостерический (регуляторный) сайт. Тем не менее, роль остальных участков связывания, а их большинство,

* Статья рекомендована к публикации программным комитетом Международной конференции «Суперкомпьютерные дни в России — 2017».

остаётся неизвестной. Как определить те центры связывания, которые важны для функции и регуляции фермента, а также научиться управлять его активностью при помощи низкомолекулярных синтетических и природных соединений?

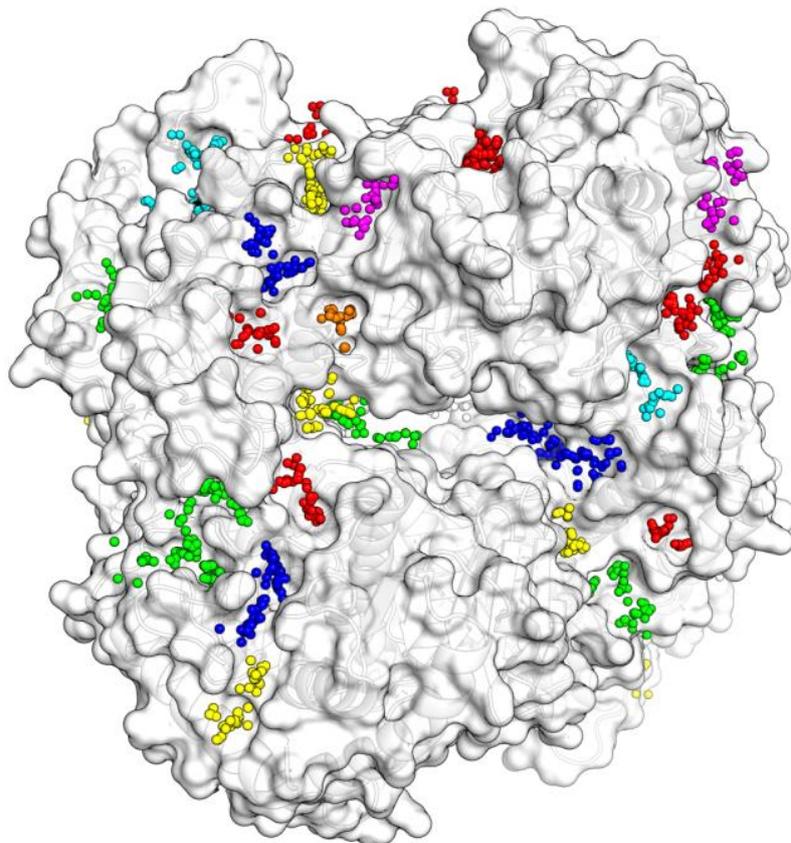


Рис. 1. Потенциальные сайты связывания (скопление шариков одного цвета) в структуре одного из ферментов гликолиза. Изображение подготовлено с помощью программы PyMol

Последние шестьдесят лет были периодом становления биоинформатики и молекулярного моделирования как ключевых подходов к изучению взаимосвязи между структурой и функцией в белках. Долгосрочные возможности вычислительных методов в биологии были осознаны в середине XX века после успеха пионерного исследования Фредерика Сэнгера по определению геномной последовательности белка инсулина. Основываясь на этом результате, Эмиль Цукеркандль и Лайнус Полинг в 1960-х годах высказали гипотезу о том, что аминокислотные последовательности можно «выравнивать» — т.е. сравнивать «побуквенно» и таким образом изучать эволюционные изменения в белках на молекулярном уровне — прообраз современного сравнительного анализа гомологичных (т.е. родственных) белков [2]. Работы Дэвида Филлипса по кристаллизации белков в конечном итоге привели к расшифровке первой трехмерной структуры лизоцима в 1965 году. В том же году увидел свет первый номер «Атласа последовательностей и структур белков» Маргарет Дэйхоф, а также ее первые программируемые компьютерные методы для изучения эволюции белков на основании выравнивания их аминокислотных последовательностей. Позже, в 1971 году, под руководством Вальтера Гамильтона был создан Protein Data Bank как хранилище трехмерных представлений био-

логических макромолекул и первоначально содержал всего 7 структур. Публикация первой полной геномной последовательности в 1977 продолжилась созданием банка геномной информации GenBank в 1982 году, а затем и базы Swiss-Prot, основанной в 1986 году для того, чтобы дополнить геномную информацию о белках экспериментально проверенной функциональной аннотацией. За несколько лет до этого, в 1981 году, IBM представила миру новую аппаратную платформу — персональный компьютер — который использовал микропроцессор 4.77МГц от Intel. Через два года, в 1983 году, компания Apple представила Lisa — первый в своем роде компьютер с графическим пользовательским интерфейсом, а в 1984 году в комплекте Macintosh появилась мышь. Это было началом новой эры в науке, когда накопление экспериментальной информации о белках соседствовало с развитием персональных компьютеров. Доступность — первичных данных, программ и алгоритмов, а также вычислительной инфраструктуры — и потребность в новых методах анализа поступающей биологической информации создали условия для развития биоинформатики и вычислительной биологии и предопределили их роль в современной науке. Оборудование для хранения данных становилось дешевле, вычислительная мощность ЭВМ продолжала увеличиваться. Вычисления, которые казались ресурсозатратными в 1980-х, стали выполняться за мгновения после того, как вначале 2000 г. появился Pentium 4 — первый настольный процессор с гигагерцовой тактовой частотой.

Тот факт, что компьютеры становились все быстрее, позволил перенести акцент с программирования на биологию. Другими словами, стратегия заключалась в том, чтобы уменьшить время, затрачиваемое на написание и отладку кода и вкладывать больше усилий в разработку новых методов, осмысление и систематизацию результатов, понимание биологических закономерностей и механизмов изучаемых процессов. Такие языки как C/C++ позволяют писать программы, которые выполняются быстрее и требуют меньше памяти, но труднее в изучении и требуют больших усилий для отладки и оптимизации кода. Одним из ключевых аспектов новой философии стал выбор такой стратегии и языка программирования, которые максимально снижали продолжительность решения конкретной задачи, что подразумевает не столько скорость работы программы, сколько суммарное время работы коллектива от постановки научной проблемы до получения значимых результатов [3]. Наибольшее распространение в биоинформатике получили такие языки как Java, Perl и Python — они проще в изучении, требуют, как правило, меньше строк кода и используют автоматическое управление памятью. Эти удобства достигаются за счет издержек при обработке данных, которые уменьшают производительность приложения. Ускоренная разработка программ за счет экономии времени на обучении и применении «медленных» языков программирования стала возможной во многом благодаря тому, что большинство отдельных задач в компьютерной биологии могут быть решены за относительно небольшое время. Так, программа Modeller, которая применяется для предсказания трехмерной организации белков по гомологии с более изученными родственными белками, занимает одно ядро процессора на две минуты для того, чтобы построить модель структуры большого гомотетрамерного белка глицеральдегид-3-фосфатдегидрогеназы, включающего более 1300 аминокислотных остатков. Выравнивание аминокислотных последовательностей нескольких сотен ДНК-зависимых РНК-полимераз, каждая из которых состоит из 1000 остатков (цепь β), можно построить с помощью программы Mafft за 3 секунды на одном ядре современного процессора.

Эксперимент по *in silico* докингу среднестатистического низкомолекулярного лиганда, включающего от нуля до десяти торсионных углов (степеней свободы), как правило, занимает от нескольких секунд до получаса с такими программами, как AutoDock и Leadfinder. Некоторые биоинформатические алгоритмы более требовательны к ресурсам, но могут быть значительно ускорены за счет эффективного использования современных многоядерных процессоров. В этом случае речь, как правило, идет о параллельном программировании в рамках общей памяти — OpenMP (например, выравнивание структур белков в программе МАТТ) или потоки в Java (например, поиск функционально важных остатков в больших суперсемействах ферментов с помощью программы Zebra). Только самые ресурсозатратные алгоритмы реализованы в виде параллельного кода с использованием MPI. Наиболее известными классами методов, имеющих оригинальные MPI реализации, являются филогенетический анализ, а также молекулярная динамика и связанные с ней подходы теоретической химии.

Анализ сложных биологических данных требует гораздо большего, чем запуск одной программы один раз — он подразумевает многократный запуск разных программ в определенной последовательности. При этом сложность представляет оптимизация не только основных приложений, но и скрепляющих их многочисленных подпрограмм, которые занимаются подготовкой ввода, анализом и систематизацией результатов. Например, сравнительный анализ родственных ферментов одного суперсемейства, объединяющего представителей с разными свойствами в рамках общей структурной организации, представляет огромный интерес как с точки зрения изучения структурно-функциональных взаимосвязей, так и для создания препаратов белков с улучшенными свойствами для практического применения, дизайна ингибиторов ферментов и лекарственных препаратов. Задача построения множественного выравнивания эволюционно удаленных белков подразумевает комбинированное использование как структурной, так и геномной информации. Можно сразу отметить, что подобные выравнивания сами по себе представляют коммерческую ценность [4, 5], что отражает сложность задачи с научной и технической точки зрения. На первом этапе происходит поиск эволюционно удаленных родственников по структурному сходству. Предполагается, что такие белки произошли от очень далекого общего предка, характеризуются широким функциональным разнообразием и могут сильно отличаться по аминокислотной последовательности. На втором этапе, каждая структура белка сопоставляется с базой данных известных аминокислотных последовательностей для выявления близких эволюционных родственников, которые обладают достаточным сходством по последовательностям и могут быть выравнены без использования структурной информации. Наконец, структуры удаленных гомологов выравниваются с использованием алгоритмов структурного сравнения, и соответствующие совмещения используются для выравнивания последовательностей гомологов без известной структуры. В те времена, когда объем информации в базах данных (в контексте рассматриваемого примера — количество известных структур и последовательностей представителей одного суперсемейства) был небольшим, подобные операции можно было выполнить относительно быстро в последовательном режиме, однако и польза от такого результата была ограничена. В последние годы бурное развитие методов геномики, протеомики, метаболомики и транскриптомики привело к увеличению объемов свободно доступных данных о структуре и функции ферментов. Первичные базы данных, собирающие информацию об аминокислотных последовательностях и струк-

турах белков, растут в геометрической прогрессии и на сегодняшний день насчитывают 85 млн. записей в UniProtKB и 130 тыс. в PDB, соответственно. Активно развиваются вторичные базы данных, обобщающие структурную и функциональную информацию, классифицирующие белки/ферменты с разным строением, свойствами и происхождением. Анализ этой информации открывает новые возможности для понимания структурно-функциональных взаимосвязей в белках, однако требует новых программных и аппаратных решений, которые позволили бы обрабатывать данные такого объема и получать биологически осмысленный результат в разумные сроки. Решить проблему могут суперкомпьютеры. Но поставить на поток ре-имплементацию всех необходимых программ из оригинальных кодов, в том числе написанных на Java, Perl, Python и др., на код с поддержкой MPI не представляется возможным. Более важно то, что не всегда это представляется лучшим решением.

Таким образом, изучение свойств и строения ферментов с использованием биоинформатики и молекулярного моделирования является комплексной задачей, требующей сочетания различных методов и способов их исполнения. Фактически, речь идет о конвейере из последовательных этапов, исполняемых различными программами, предъявляющими свои требования к вычислительным ресурсам. Одни приложения требуют больше оперативной памяти, другие выигрывают от быстрых процессоров. Есть программы, которые критически зависят от скорости чтения с диска. Некоторые алгоритмы могут быть значительно ускорены на графических процессорах и т.д. Подобная структура создает основу для кодизайна, т.е. создания оптимального единого решения за счет взаимосвязанной разработки программного обеспечения и выбора аппаратной конфигурации для каждой стадии и отдельно взятой программы. В этой статье мы обсуждаем поиск новых путей регуляции конкретного фермента с использованием биоинформатики и молекулярного моделирования. Мы описываем последовательные этапы предложенного нами решения, обсуждаем программные и аппаратные аспекты каждой отдельной стадии, приводим описание оригинальных программных продуктов. Мы делаем вывод, что для изучения структуры, функции и регуляции ферментов необходимы гибридные кластеры — вычислительные системы, обладающие как существенной мощностью, так и разнообразием аппаратных возможностей — которые позволяют оптимально исполнить каждую отдельную стадию единого комплексного решения. Мы также заключаем, что GPU-ускорители открывают новые возможности для применения методов биоинформатики и молекулярного моделирования при решении задач естественных наук.

Структура решения и описание этапов

Глицеральдегид-3-фосфатдегидрогеназа (ГАФД) — фермент гликолиза, катализирующий центральную реакцию этого метаболического пути — гликолитическую оксидоредукцию. Интерес к ГАФД обусловлен тем, что блокирование этой реакции препятствует не только накоплению макроэргических фосфатов, но и восстановленного никотинамидадениндинуклеотида, используемого для получения энергии в дыхательной цепи. Помимо гликолиза ГАФД также принимает участие в других важных для клетки процессах. Иными словами, ГАФД как ключевой фермент метаболизма в клетках явля-

ется многообещающей мишенью для создания лекарств от различных бактериальных инфекций.

Для изучения структуры и функции ГАФД с целью поиска новых путей регуляции этого фермента нами предложено решение, основанное на последовательном и взаимосвязанном исполнении методов биоинформатического анализа родственных белков, молекулярного моделирования структуры фермента, подходов теоретической химии и статистического анализа.

Поиск новых регуляторных сайтов связывания

Первым этапом решения является поиск новых сайтов связывания регуляторных лигандов в ГАФД с использованием оригинального подхода биоинформатического анализа. В основе биоинформатического анализа лежит понятие «выравнивание» — сопоставление аминокислотных последовательностей и элементов структуры родственных белков. Проблема выравнивания белков заключается в том, что длительная эволюция привела к значительным отличиям их пространственных структур и аминокислотных последовательностей. В этой связи задача биоинформатического анализа разделяется на три подзадачи: (1) поиск родственных белков в базах данных на основании сходства структур и последовательностей, (2) построение множественного выравнивания и (3) анализ выравнивания и аннотация новых сайтов в структурах белков.

В настоящий момент в банке данных трехмерных моделей белков PDB насчитывается более 130 тыс. структур (более 350 тыс. структур отдельных цепей) и подзадача поиска структурного сходства предполагает исполнение процедуры парного сравнения с каждой из них, что представляет значительную вычислительную сложность. Для управления процедурой поиска родственных белков по структурному сходству по базе данных PDB нами разработано оригинальное приложение. Программа осуществляет форматирование базы данных белковых структур и их предварительный анализ. Это позволяет заранее отбросить наименее похожие структуры и сфокусировать поиск на заведомо более близких родственниках. Следующей важной особенностью программы является сохранение однажды посчитанных парных выравниваний в базе данных, организованной на основе PostgreSQL. Фокусирование области поиска за счет предварительного анализа данных и депонирование однажды посчитанных результатов в базе данных позволяет сократить время расчета как при повторном запуске одной и той же подзадачи (например, для уточнения параметров поиска), так и для независимых подзадач, в которых используются структурно похожие белки. Однако, не для всех белков известна их структура — для недавно открытых, а также малоизученных белков, как правило, известна только последовательность аминокислот, но не их расположение в пространстве. В таких случаях необходимо использовать поиск родственных белков в базах данных на основании сходства аминокислотных последовательностей. Количество известных последовательностей белков в базах данных Swiss-Prot и TrEMBL составляет на сегодняшний день в сумме более 85 млн. и подзадача поиска сходства предполагает исполнение процедуры парного сравнения с каждой из них. Для исполнения этой вычислительно сложной подзадачи мы используем GPU реализацию популярной программы BLAST [6, 7].

Подзадача построения множественного выравнивания родственных белков, выявленных в результате решения предыдущей подзадачи поиска в базах данных, предпола-

гает использование двух типов алгоритмов — для выравнивания аминокислотных последовательностей и структур белков. Выравнивание последовательностей (фактически, сравнение строк текста) эволюционно близких белков не представляет существенной сложности и, как правило, может быть качественно исполнено на одном ядре современного процессора в пределах от нескольких секунд до минуты. Сопоставление структур, которое основано на пространственном совмещении координат атомов, напротив, представляет существенную вычислительную сложность. Для исполнения такого выравнивания нами была разработана оригинальная MPI ре-имплементация популярной программы МАТТ структурного выравнивания. Созданное приложение раМАТТ позволяет ускорить выравнивание структур белков в параллельном режиме с использованием MPI на кластере или суперкомпьютере. Программа полезна для выравнивания больших выборок, состоящих из сотен (и более) белков, которые характеризуются достаточным структурным сходством и наличием общего структурного ядра, а также предназначена для ускорения повторяющихся и часто исполняемых операций — пересчета выравнивания с уточненными параметрами, а также для работы публичных сервисов. Структурное выравнивание ГАФД из различных родственных организмов, в том числе человека и бактерий, приведено на рис. 2.

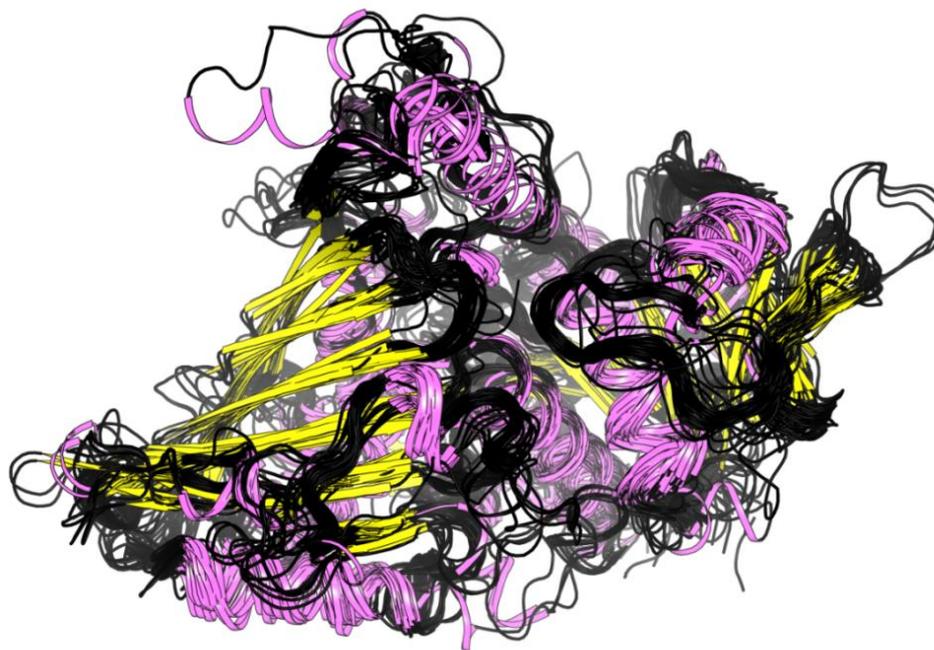


Рис. 2. Структурное выравнивание репрезентативной выборки ГАФД-подобных ферментов. Розовым и желтым цветом показаны участки с регулярной вторичной структурой — α -спирали и β -слои, соответственно. Черным показаны петли — участки без регулярной вторичной структуры. Изображение подготовлено с помощью программы PyMol

Последняя подзадача заключается в статистическом анализе полученного множественного выравнивания ГАФД из различных организмов, в том числе человека и бактерий, для выявления общих черт и особенностей организации активных и аллостерических центров. Суть подхода заключается в выявлении специфических позиций — таких участков структуры, которые отличают ферменты из разных организмов — и использо-

вании их в качестве критерия для ранжирования полостей на поверхности белка по степени функциональной значимости (рис. 3). Вычисления выполняются с использованием оригинальных программ Zebra и rocketZebra [8–10]. Наиболее ресурсоемкой стадией этой подзадачи является расчет случайной модели, в процессе которого колонка выравнивания перетасовывает несколько тысяч раз. Основная часть расчета случайной модели производится независимо для каждой колонки в выравнивании, что позволяет выполнять эти вычисления параллельно.

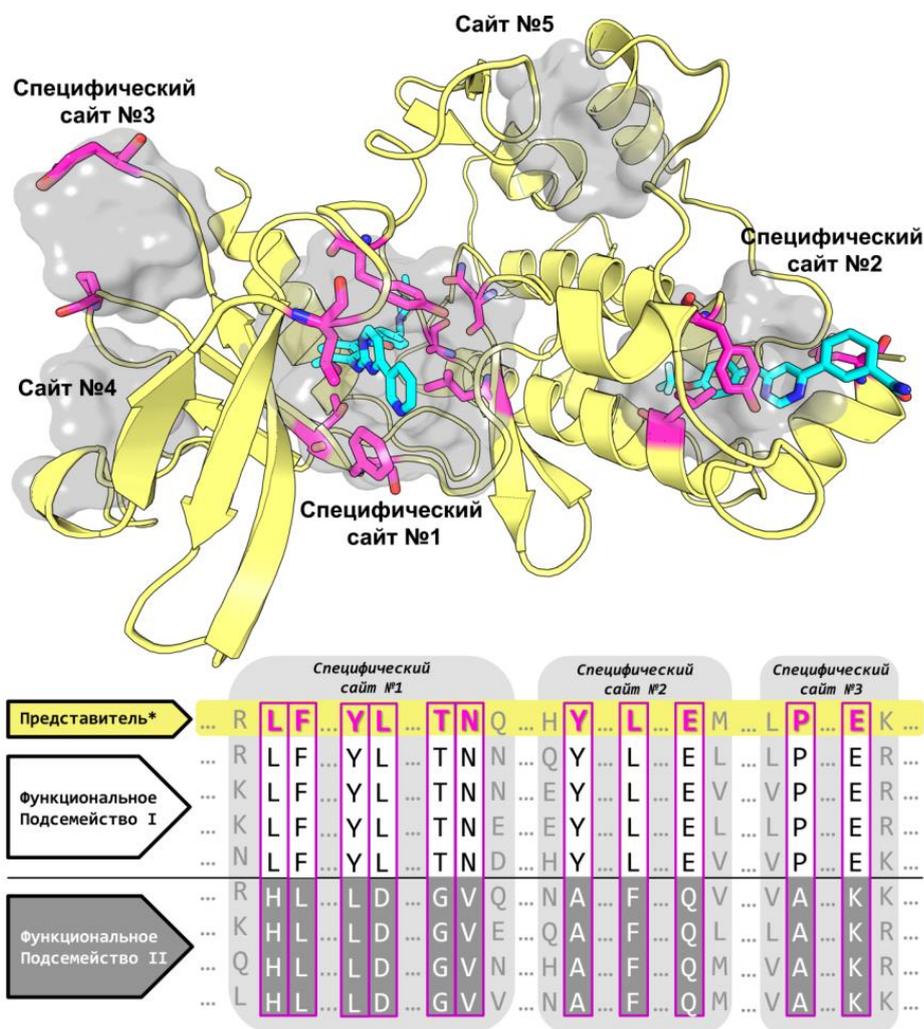


Рис. 3. Схематическое представление поиска новых сайтов связывания лигандов (функциональных и регуляторных) в родственных белках [10]. Специфические позиции подсемейств показаны в виде палочек в структуре (сверху) и пурпурным цветом в колонках множественного выравнивания (снизу). Изображение подготовлено с помощью программы PyMol

С точки зрения аппаратной реализации, предложенное решение задачи биоинформатического анализа ориентировано на исполнение в рамках одного вычислительного узла, содержащего многоядерный процессор максимальной мощности, достаточное количество оперативной памяти, мощный графический ускоритель, а также SSD-накопитель. SSD-диск используется для хранения всех необходимых баз данных, что существенно ускоряет решение. При этом обновление наиболее крупных баз данных

(около 100 ГБ) допускается производить не чаще двух-трех раз в год, что позволяет экономить ресурс SSD-устройства. Видеоускоритель используется для исполнения процедуры поиска по сходству последовательностей с помощью GPU-BLAST. Многоядерность основного вычислителя способствует ускорению остальных параллельных операций, например, поиск по структурному сходству в базе данных, который предполагает выполнение большого количества независимых парных сравнений. Особый случай представляет подзадача построения множественного структурного выравнивания, которая, в случае необходимости, может быть исполнена оригинальной программой `ragMATT` в режиме MPI на кластере или суперкомпьютере. Более подробно о задаче биоинформатического анализа можно узнать в недавних публикациях [11] и на сайте <https://biokinet.belozersky.msu.ru>. С оригинальной реализацией протокола построения множественных выравниваний функционально разнообразных семейств белков в рамках одного гибридного узла можно познакомиться по адресу <https://zeus.cmm.msu.ru/>.

В результате выполнения текущего этапа исследования был обнаружен новый, ранее неизвестный сайт связывания лигандов в структуре ГАФД.

Скрининг ингибиторов

Вторым этапом решения является скрининг нового регуляторного сайта связывания лигандов в структуре ГАФД, обнаруженного в результате выполнения первого этапа работы, большой библиотекой потенциальных ингибиторов. Был разработан оригинальный комплекс программ для исполнения высокопроизводительного скрининга ингибиторов на суперкомпьютере, который включает в себя следующие компоненты: универсальный планировщик для запуска в параллельном режиме большого количества независимых подзадач (где каждая подзадача подразумевает докинг одного ингибитора в заданный сайт); программа для форматирования большой библиотеки лигандов для скринирования на суперкомпьютере в параллельном режиме; программа для подготовки структуры белка к скринированию ингибиторов в разные сайты на его поверхности (т.е. для подготовки «сеток» разных сайтов в структуре одного белка); программа для запуска собственно виртуального скрининга в параллельном режиме на суперкомпьютере; набор программ для обработки выходных данных, содержащих результаты скрининга. Подробное описание комплекса программ и принципа их работы приведены в недавней публикации [12]. Использование суперкомпьютера позволило проводить анализ больших библиотек ингибиторов с более высокой точностью, которая достигается за счет использования более аккуратного и ресурсозатратного алгоритма оценки энергии связывания лиганда в структуре белка. Наличие суперкомпьютерной системы позволило также использовать для докинга не единственную структуру белка, а ансамбль конформеров фермента, полученный при помощи молекулярной динамики, для того, чтобы учесть подвижность аминокислотных остатков и небольшие флуктуации в структурах сайта связывания.

Таким образом, с программной точки зрения задача скрининга рассматривается как совокупность большого числа подзадач докинга индивидуальных лигандов, которые могут быть исполнены параллельно и независимо, поскольку не обмениваются информацией в процессе расчета. С аппаратной точки зрения, одна подзадача докинга индивидуального лиганда выполняется на одном ядре классического CPU в интервале от нескольких секунд до нескольких минут, при этом не предъявляет существенных требований к остальным ресурсам.

В результате выполнения текущего этапа исследования с использованием суперкомпьютера «Ломоносов-2» было выполнено в общей сложности около 30 млн. докингов, наиболее перспективные ингибиторы ГАФД из патогенных бактерий были отобраны для дальнейшего изучения.

Молекулярное моделирование комплексов фермент-ингибитор

На третьем этапе решения связывание в новом регуляторном сайте ГАФД наиболее перспективных ингибиторов, отобранных на предыдущем этапе, изучалось с использованием молекулярной динамики. Эта задача разделяется на три подзадачи — (1) подготовка моделей, (2) молекулярная динамика и (3) анализ результатов молекулярной динамики.

Подготовка моделей подразумевает построение полноразмерных молекулярных моделей белка и ингибитора, в том числе с учетом рКа ионизируемых групп и альтернативных степеней протонирования, а также параметризация этих моделей — т.е. приведение информации о них в соответствие с требованиями программы молекулярной динамики. Важной составляющей этой подзадачи является расчет зарядов на каждом атоме в структурах. Информация об атомах белка по умолчанию берется из стандартного силового поля. Однако для остальных молекул — например, кофакторов и ингибиторов — эти параметры необходимо вычислять. Процедура оценки зарядов атомов в низкомолекулярных соединениях включает, в том числе, методы, основанные на квантово-механических подходах теоретической химии, например, реализованных в популярном отечественном продукте Firefly (ранее известном как PC GAMESS). Особенностью этих методов, как правило, является интенсивное использование операций обращения к жесткому диску для записи и чтения промежуточных результатов. Суммарный объем перезаписываемой информации может достигать терабайта даже для небольших молекул (несколько десятков атомов), в этой связи использование SSD-накопителя ускорило бы процесс, однако быстро приведет к исчерпанию ресурса и, таким образом, экономически не оправдано. Представляется адекватным использовать для этой цели отдельные узлы с быстрыми классическими накопителями.

Полученные модели были использованы как стартовые конформации для метода молекулярной динамики. Использование суперкомпьютера позволило применить метод молекулярной динамики с более высокой точностью для изучения взаимодействия ингибиторов с аминокислотами нового регуляторного сайта связывания. Для этого температура была установлена на уровне 300К (27°C), поскольку использование более высоких значений чаще приводит к возникновению артефактов; использована более сложная 4-сайтовая молекулярная модель воды TIP4P-Ew, которая более правильно воспроизводит физико-химические характеристики растворителя; использован адекватный отступ между поверхностью белка и краем молекулярной ячейки для исключения артефактов в периодической системе; параметры для учета электростатических взаимодействий были установлены на максимальные рекомендованные значения. Для каждой модели фермента в комплексе с ингибитором были вычислены по пять независимых траекторий для того, чтобы собрать больше информации для статистической обработки. Метод молекулярной динамики является вычислительно сложным. С программной точки зрения некоторые процедуры молекулярной динамики уже частично реализованы на GPU и получили широкое распространение благодаря существенно более высокой скорости расче-

тов — например, популярный метод расчета PME электростатики в явно заданном растворителе и NVE/NVT/NPT ансамблях в пакете Amber версии 14 и выше [13–15]. С аппаратной точки зрения, стадии минимизации, нагрева и релаксации (10–20 коротких, последовательно запускаемых стадий) удобно исполнять на мощном локальном видеоускорителе, таком как GeForce GTX 980 Ti. Наиболее ресурсоемкая стадия свободной динамики в каждом случае исполнялась на четырех видеокартах Tesla K40 суперкомпьютера «Ломоносов-2».

Последняя подзадача подразумевает анализ результатов молекулярной динамики. Эта стадия зависит не столько от мощности компьютера и организации вычислительного процесса, сколько от выбранной стратегии анализа, а также подготовленности и опытности эксперта, и по этой причине мы не будем подробно описывать ее здесь.

По результатам молекулярного моделирования наиболее перспективные лиганды были рекомендованы для экспериментальной проверки ингибирующей активности по отношению к ГАФД из разных организмов.

Заключение

Экспериментальная проверка лигандов, отобранных по результатам биоинформатического анализа и компьютерного моделирования, показала наличие ингибирующей активности по отношению к бактериальной ГАФД и отсутствие влияния на родственный фермент в организме человека в изученных концентрациях. Последующая проверка наиболее эффективных ингибиторов бактериального ГАФД показала подавление роста культуры *Mycobacterium tuberculosis*. Таким образом, установлена противотуберкулезная активность предложенных соединений. Подходы, аналогичные использованным в этой работе, были ранее применены нами для изучения структуры и функции ферментов из других суперсемейств, поиска и характеристики новых центров связывания, а также получения препаратов ферментов с улучшенными свойствами [16–20].

Мы делаем вывод, что для изучения структуры, функции и регуляции ферментов необходимы вычислительные системы, обладающие как существенной мощностью, так и разнообразием аппаратных возможностей. Отдельные компоненты такого гибридного кластера должны быть ориентированы на конкретные методы компьютерной биологии, которые, являясь частью единого комплексного решения, используются последовательно и взаимосвязано при решении конкретных задач. Так, для исполнения методов биоинформатики в первую очередь необходимы отдельные независимые узлы с многоядерными процессорами максимальной мощности, GPU-ускорителями и SSD-накопителями для хранения баз данных. Для исполнения метода молекулярной динамики необходимы мощные GPU-ускорители. Для исполнения молекулярного скрининга необходимы быстрые CPU, при этом связь между ними не влияет на эффективность расчета, поскольку подзадачи докинга полностью независимы друг от друга.

Мы также хотим обратить внимание на то, что речь в статье идет именно о «вычислительных кластерах». Разумеется, когда речь заходит о больших организациях, таких как МГУ имени М.В. Ломоносова, масштабные вычислительные системы и суперкомпьютеры [21] необходимы для того, чтобы удовлетворить потребности всех пользователей в вычислительных ресурсах. Однако если отдельно рассматривать потребности небольших научных коллективов, то для их реализации будет достаточно и относительно

скромных ресурсов, как по размерам и сложности устройства, так и по стоимости. Это обусловлено, прежде всего, развитием GPU-технологий. Еще недавно для запуска одной молекулярной динамики требовалось использование сотен ядер классических CPU, установленных на десятках отдельных узлов и связанных дорогой сетью. Сегодня аналогичную скорость расчета можно получить даже на одном игровом видеоускорителе, таком как GeForce GTX 980 Ti. Стоимость таких устройств несопоставима меньше затрат на суперкомпьютер. Важно то, что использование мощных GPU-ускорителей позволяет не столько анализировать биологические системы с большей скоростью, сколько дает возможность увеличивать точность анализа за счет перенастройки параметров или усложнения соответствующих процедур. В последние годы были сделаны первые шаги в сторону создания GPU-реализаций некоторых биологически-ориентированных алгоритмов [22]. Пожалуй, на первом месте по популярности находится уже упомянутый метод молекулярной динамики, который частично реализован на GPU. Тем не менее, можно констатировать, что популярность классических CPU-реализаций алгоритмов в естественно-научной среде существенно выше спроса на GPU-версии. На практике получается, что издержки на обучение, установку и обновление GPU-приложений часто оказываются выше, чем положительный эффект от их использования, что связано со сложной организацией задач компьютерной биологии. В этом контексте фундаментальный и практический интерес представляет дальнейший поиск таких этапов в решении задач компьютерной биологии, реализация которых с использованием GPU была бы эффективной и открывала новые возможности для анализа возрастающих объемов биологических данных.

Работа выполнена при поддержке грантов РФФИ №17-07-00751 и РНФ №15-14-00069 с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова.

Литература

1. Суплатов Д.А., Швядас В.К. Изучение функциональных и аллостерических сайтов в суперсемействах белков // Acta Naturae. 2015. Т. 7, № 4. С. 39–52.
2. Zuckerkandl E., Pauling L. Evolutionary Divergence and Convergence in Proteins // Evolving Genes and Proteins. 1965. Vol. 97. P. 97–166.
3. Fourment M., Gillings M.R. A Comparison of Common Programming Languages Used in Bioinformatics // BMC bioinformatics. 2008. Vol. 9, No. 1. P. 82. DOI:10.1186/1471-2105-9-82.
4. Kourist R., Jochens H., Bartsch S., Kuipers R., Padhi S.K., Gall M., Dominique B., Henk-Jan J., Bornscheuer U.T. The α/β -hydrolase Fold 3DM Database (ABHDB) as a Tool for Protein Engineering // ChemBioChem. 2010. Vol. 11, No. 12. P. 1635–1643. DOI: 10.1002/cbic.201000213.
5. Программное обеспечение BioProduct. URL: <https://www.bio-product.nl/> (дата обращения: 28.09.2017).
6. Vouzis P.D., Sahinidis N.V. GPU-BLAST: Using Graphics Processors to Accelerate Protein Sequence Alignment // Bioinformatics. 2011. Vol. 27, No. 2. P. 182–188. DOI: 10.1093/bioinformatics/btq644.

7. Altschul S.F., Madden T.L., Schäffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J. Gapped BLAST and PSI-BLAST: a New Generation of Protein Database Search Programs // *Nucleic Acids Research*. 1997. Vol. 25, No. 17. P. 3389–3402. DOI: 10.1093/nar/25.17.3389.
8. Suplatov D., Shalaeva D., Kirilin E., Arzhanik V., Švedas V. Bioinformatic Analysis of Protein Families for Identification of Variable Amino Acid Residues Responsible for Functional Diversity // *Journal of Biomolecular Structure and Dynamics*. 2014. Vol. 32, No. 1. P. 75–87. DOI: 10.1080/07391102.2012.750249.
9. Suplatov D., Kirilin E., Takhaviev V., Švedas V. Zebra: Web-server for Bioinformatic Analysis of Diverse Protein Families // *Journal of Biomolecular Structure and Dynamics*. 2014. Vol. 32, No. 11. P. 1752–1758. DOI: 10.1080/07391102.2013.834514.
10. Suplatov D., Kirilin E., Arbatsky M., Takhaviev V., Švedas V. PocketZebra: a Web-server for Automated Selection and Classification of Subfamily-specific Binding Sites by Bioinformatic Analysis of Diverse Protein Families // *Nucleic Acids Research*. 2014. Vol. 42, No. W1. P. W344–W349. DOI: 10.1093/nar/gku448.
11. Suplatov D., Kirilin E., Švedas V. Bioinformatic Analysis of Protein Families to Select Function-related Variable Positions “Understanding Enzymes: Function, Design, Engineering and Analysis” (Allan Svendsen) Pan Stanford Publishing 2016. P. 351–385.
12. Suplatov D., Popova N., Zhumatiy S., Voevodin V., Švedas V. Parallel Workflow Manager for Non-parallel Bioinformatic Applications to Solve Large-scale Biological Problems on a Supercomputer // *Journal of Bioinformatics and Computational Biology*. 2016. Vol. 14, No. 2. P. 1641008. DOI: 10.1142/S0219720016410080.
13. Le Grand S., Götz A.W., Walker R.C. SPFP: Speed Without Compromise — A Mixed Precision Model for GPU Accelerated Molecular Dynamics Simulations // *Computer Physics Communications*. 2013. Vol. 184, No. 2. P. 374–380. DOI: 10.1016/j.cpc.2012.09.022.
14. Salomon-Ferrer R., Goetz A.W., Poole D., Le Grand S., Walker R.C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald // *Journal of Chemical Theory and Computation*. 2013. Vol. 9, No. 9. P. 3878–3888. DOI: 10.1021/ct400314y.
15. Goetz A.W., Williamson M.J., Xu D., Poole D., Le Grand S., Walker R.C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born // *Journal of Chemical Theory and Computation*. 2012. Vol. 8, No. 5. P. 1542–1555. DOI: 10.1021/ct200909j.
16. Bezsudnova E.Y., Stekhanova T.N., Suplatov D.A., Mardanov A.V., Ravin N.V., Popov V.O. Experimental and Computational Studies on the Unusual Substrate Specificity of Branched-chain Amino Acid Aminotransferase from *Thermoproteus uzoniensis* // *Archives of Biochemistry and Biophysics*. 2016. Vol. 607. P. 27–36. DOI: 10.1016/j.abb.2016.08.009.
17. Suplatov D.A., Voevodin V.V., Svedas V.K. Robust Enzyme Design: Bioinformatic Tools for Improved Protein Stability // *Biotechnology Journal*. 2015. Vol. 10, No. 3. P. 344–355. DOI: 10.1002/biot.201400150.
18. Shcherbakova T., Panin N., Suplatov D., Shapovalova I, Švedas V. The β D484N Mutant of Penicillin Acylase from *Escherichia coli* is More Resistant to Inactivation by Substrates and Can Effectively Perform Peptide Synthesis in Aqueous Medium // *Journal of Molec-*

- ular Catalysis B: Enzymatic. 2015. Vol. 112. P. 66–68. DOI: 10.1016/j.molcatb.2014.11.015.
19. Suplatov D., Panin N., Kirilin E., Shcherbakova T., Kudryavtsev P., Švedas V. Computational Design of a pH Stable Enzyme: Understanding Molecular Mechanism of Penicillin Acylase's Adaptation to Alkaline Conditions // PLoS ONE. 2014. Vol. 9, No. 6. P. e100643. DOI:10.1371/journal.pone.0100643.
20. Suplatov D., Besenmatter W., Švedas V., Svendsen A. Bioinformatic Analysis of Alpha/Beta-Hydrolase Fold Enzymes Reveals Subfamily-Specific Positions Responsible for Discrimination of Amidase and Lipase Activities // Protein Engineering, Design and Selection. 2012. Vol. 25, No. 11. P. 689–697. DOI:10.1093/protein/gzs068.
21. Воеводин Вл.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.А., Никитенко Д.А., Стефанов К.С., Воеводин Вад.В. Практика суперкомпьютера «Ломоносов» // Открытые системы. 2012. Т. 7. С. 36–39.
22. Nobile M.S., Cazzaniga P., Tangherloni A., Besozzi D. Graphics Processing Units in Bioinformatics, Computational Biology and Systems Biology // Briefings in Bioinformatics. 2016. P. 870–885. DOI: 10.1093/bib/bbw058.

Суплатов Дмитрий Андреевич, к.х.н., с.н.с. НИИ физико-химической биологии имени А.Н. Белозерского, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

Попова Нина Николаевна, к.ф.-м.н., доцент Факультета вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

Копылов Кирилл Евгеньевич, студент Химического факультета, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

Шегай Максим Викторович, аспирант Факультета вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

Воеводин Владимир Валентинович, чл.-корр. РАН, д.ф.-м.н., зав. кафедрой суперкомпьютеров и квантовой информатики Факультета вычислительной математики и кибернетики, зам. директора Научно-исследовательского вычислительного центра, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

Швядас Витас Каятоно, д.х.н., профессор Факультета биоинженерии и биоинформатики и НИИ физико-химической биологии имени А.Н. Белозерского, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

HYBRID COMPUTING CLUSTERS TO STUDY PROTEIN STRUCTURE, FUNCTION AND REGULATION

© 2017 D.A. Suplatov, N.N. Popova, K.E. Kopylov, M.V. Shegay,
Vl.V. Voevodin, V.K. Švedas

Lomonosov Moscow State University

(GSP-1, Leninskie Gory 1, Moscow, 119991 Russia)

*E-mail: d.a.suplatov@belozersky.msu.ru, popova@cs.msu.ru, kopylov@mail.chem.msu.ru,
max.shegai@gmail.com, voevodin@parallel.ru, vytaš@belozersky.msu.ru*

Received: 11.09.2017

Studying protein structure, function and regulation using bioinformatics and molecular modeling is a complex task that requires a combination of various methods and ways to implement them. The process can be seen as a pipeline of sequential steps executed by various programs which benefit from customized hardware. Hybrid computing clusters characterized by a significant performance and a variety of hardware capabilities are necessary to optimally execute each individual step of the complex solution. It can be specifically noted that GPU accelerators open new opportunities for efficient solution of resource-intensive tasks of bioinformatics and molecular modeling.

Keywords: Hybrid computing clusters, bioinformatics, molecular modeling, computational pipeline, sequential steps, co-design, GPU accelerators.

FOR CITATION

Suplatov D.A., Popova N.N., Kopylov K.E., Shegay M.V., Voevodin Vl.V., Švedas V.K. Hybrid Computing Clusters to Study Protein Structure, Function and Regulation. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. vol. 6, no. 4. pp. 74–90. (in Russian) DOI: 10.14529/cmse170406.

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. Suplatov D., Švedas V. Study of Functional and Allosteric sites in Protein Superfamilies. *Acta Naturae*. 2015. vol. 7, no. 4, pp. 34–45.
2. Zuckerkandl E., Pauling L., Evolutionary Divergence and Convergence in Proteins. *Evolving Genes and Proteins*. 1965. vol. 97. pp. 97–166.
3. Fourment M., Gillings M.R. A Comparison of Common Programming Languages Used in Bioinformatics. *BMC Bioinformatics*. 2008. vol. 9, no. 1. pp. 82. DOI: 10.1186/1471-2105-9-82.
4. Kourist R., Jochens H., Bartsch S., Kuipers R., Padhi S.K., Gall M., Dominique B., Henk-Jan J., Bornscheuer U.T. The α/β -hydrolase Fold 3DM Database (ABHDB) as a Tool for Protein Engineering. *ChemBioChem*. 2010. vol. 11, no. 12. pp. 1635–1643. DOI: 10.1002/cbic.201000213.

5. BioProduct software. Available at: <https://www.bio-product.nl/> (accessed: 28.09.2017).
6. Vouzis P.D., Sahinidis N.V. GPU-BLAST: Using Graphics Processors to Accelerate Protein Sequence Alignment. *Bioinformatics*. 2011. vol. 27, no. 2. pp. 182–188. DOI: 10.1093/bioinformatics/btq644.
7. Altschul S.F., Madden T.L., Schäffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J. Gapped BLAST and PSI-BLAST: a New Generation of Protein Database Search Programs. *Nucleic Acids Research*. 1997. vol. 25, no. 17. pp. 3389–3402. DOI: 10.1093/nar/25.17.3389.
8. Suplatov D., Shalaeva D., Kirilin E., Arzhanik V., Švedas V. Bioinformatic Analysis of Protein Families for Identification of Variable Amino Acid Residues Responsible for Functional Diversity. *Journal of Biomolecular Structure and Dynamics*. 2014. vol. 32, no. 1. pp. 75–87. DOI: 10.1080/07391102.2012.750249.
9. Suplatov D., Kirilin E., Takhaviev V., Švedas V. Zebra: Web-server for Bioinformatic Analysis of Diverse Protein Families. *Journal of Biomolecular Structure and Dynamics*. 2014. vol. 32, no. 11. pp. 1752–1758. DOI: 10.1080/07391102.2013.834514.
10. Suplatov D., Kirilin E., Arbatsky M., Takhaviev V., Švedas V. PocketZebra: a Web-server for Automated Selection and Classification of Subfamily-specific Binding Sites by Bioinformatic Analysis of Diverse Protein Families. *Nucleic Acids Research*. 2014. vol. 42, no. W1. pp. W344–W349. DOI: 10.1093/nar/gku448.
11. Suplatov D., Kirilin E., Švedas V. Bioinformatic Analysis of Protein Families to Select Function-related Variable Positions “Understanding Enzymes: Function, Design, Engineering and Analysis” (Allan Svendsen) Pan Stanford Publishing 2016. pp. 351–385.
12. Suplatov D., Popova N., Zhumatiy S., Voevodin V., Švedas V. Parallel Workflow Manager for Non-parallel Bioinformatic Applications to Solve Large-scale Biological Problems on a Supercomputer. *Journal of Bioinformatics and Computational Biology*. 2016. vol. 14, no. 2. pp. 1641008. DOI: 10.1142/S0219720016410080.
13. Le Grand S., Götz A.W., Walker R.C. SPFP: Speed Without Compromise — A Mixed Precision Model for GPU Accelerated Molecular Dynamics Simulations. *Computer Physics Communications*. 2013. vol. 184, no. 2. pp. 374–380. DOI: 10.1016/j.cpc.2012.09.022.
14. Salomon-Ferrer R., Goetz A.W., Poole D., Le Grand S., Walker R.C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *Journal of Chemical Theory and Computation*. 2013. vol. 9, no. 9. pp. 3878–3888. DOI: 10.1021/ct400314y.
15. Goetz A.W., Williamson M.J., Xu D., Poole D., Le Grand S., Walker R.C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *Journal of Chemical Theory and Computation*. 2012. vol. 8, no. 5. pp. 1542–1555. DOI: 10.1021/ct200909j.
16. Bezsudnova E.Y., Stekhanova T.N., Suplatov D.A., Mardanov A.V., Ravin N.V., Popov V.O. Experimental and Computational Studies on the Unusual Substrate Specificity of Branched-chain Amino Acid Aminotransferase from *Thermoproteus uzoniensis*. *Archives of Biochemistry and Biophysics*. 2016. vol. 607. pp. 27–36. DOI:10.1016/j.abb.2016.08.009.
17. Suplatov D.A., Voevodin V.V., Svedas V.K. Robust Enzyme Design: Bioinformatic Tools for Improved Protein Stability. *Biotechnology Journal*. 2015. vol. 10, no. 3. pp. 344–355. DOI:10.1002/biot.201400150.

18. Shcherbakova T., Panin N., Suplatov D., Shapovalova I, Švedas V. The β D484N Mutant of Penicillin Acylase from *Escherichia coli* is More Resistant to Inactivation by Substrates and Can Effectively Perform Peptide Synthesis in Aqueous Medium. *Journal of Molecular Catalysis B: Enzymatic*. 2015. vol. 112. pp. 66–68. DOI:10.1016/j.molcatb.2014.11.015.
19. Suplatov D., Panin N., Kirilin E., Shcherbakova T., Kudryavtsev P., Švedas V. Computational Design of a pH Stable Enzyme: Understanding Molecular Mechanism of Penicillin Acylase's Adaptation to Alkaline Conditions. *PLoS ONE*. 2014. vol. 9, no. 6. pp. e100643. DOI: 10.1371/journal.pone.0100643.
20. Suplatov D., Besenmatter W., Švedas V., Svendsen A. Bioinformatic Analysis of Alpha/beta-hydrolase Fold Enzymes Reveals Subfamily-specific Positions Responsible for Discrimination of Amidase and Lipase Activities. *Protein Engineering, Design and Selection*. 2012. vol. 25, no. 11. pp. 689–697. DOI: 10.1093/protein/gzs068.
21. Voevodin V.I., Zhumatiy S.A., Sobolev S.I., Antonov A.S., Bryzgalov P.A., Nikitenko D.A., Stefanov K.S., Voevodin Vad.V. Praktika supercomputera “Lomonosov” [Practice of the “Lomonosov” Supercomputer]. *Otkrytye sistemy [Open Systems]*. 2012. vol. 7, pp. 36–39 (in Russian)
22. Nobile M.S., Cazzaniga P., Tangherloni A., Besozzi D. Graphics Processing Units in Bioinformatics, Computational Biology and Systems Biology. *Briefings in Bioinformatics*. 2016. pp. 870–885. DOI: 10.1093/bib/bbw058.

РАЗРАБОТКА КОМПЬЮТЕРНЫХ МОДЕЛЕЙ БАЛЛИСТИЧЕСКИХ ТКАНЕЙ С ПОВЕРХНОСТНОЙ ОБРАБОТКОЙ

© 2017 Н.Ю. Долганина, А.В. Игнатова

Южно-Уральский государственный университет

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

E-mail: dolganinani@susu.ru, ignatovaav@susu.ru

Поступила в редакцию: 06.10.2017

Баллистические ткани на сегодняшний день широко применяются в качестве элементов защитных структур. Актуальными задачами при разработке бронеструктур являются минимизация их массы, уменьшение кинетической энергии пули, передаваемой объекту, расположенному за бронепанелью (снижение величины прогиба тыльной стороны панели). Значительная часть энергии пули рассеивается за счет работы сил трения при вытягивании нитей из ткани. Умение предсказывать работу баллистической ткани при вытягивании нитей позволит проектировать высокоэффективные бронеструктуры. Поэтому были разработаны малопараметрические численные модели вытягивания нити из арамидной ткани P110 полотняного переплетения, а также для этой ткани с разными типами поверхностной обработки (канифоль, силиконовая смазка) в пакете программ LS-DYNA. Поверхностная обработка ткани позволяет изменять коэффициент трения между нитями с минимальным увеличением веса, и в модели она учитывалась за счет изменения одного параметра — коэффициента сухого трения. Рассмотрено несколько способов распараллеливания задачи вытягивания нити из ткани, получены графики ускорения. Были получены расчетные зависимости нагрузки от перемещения при вытягивании нити из ткани с поверхностной обработкой и без нее. Расчетные результаты лежат в диапазоне разброса экспериментальных данных.

Ключевые слова: численные методы, метод конечных элементов, суперкомпьютерное моделирование, арамидная ткань, вытягивание нити из ткани, поверхностная обработка, LS-DYNA.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Долганина Н.Ю., Игнатова А.В. Разработка компьютерных моделей баллистических тканей с поверхностной обработкой // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 4. С. 91–100. DOI: 10.14529/cmse170407.

Введение

Основной задачей при проектировании защитных структур (бронезилетов) является минимизация их массы при сохранении заданного уровня защиты. Бронепанель должна поглотить максимум кинетической энергии пули при ударе, чтобы снизить запреградную энергию, передаваемую защищаемому объекту. В тканевом бронезилете большая часть кинетической энергии пули рассеивается за счет вытягивания нитей из тканей (фрикционного взаимодействия), а остальная часть энергии удара затрачивается на растяжение и разрыв нитей [1–3]. Поэтому повышение коэффициента трения в ткани позволит увеличить энергию рассеяния пули при ударе, снизить величину глубины прогиба тыльной стороны бронезилета и соответственно уменьшить вероятность получения серьезной запреградной (контузионной) травмы тела человека.

В настоящее время существует множество способов модернизации тканевых бронезилетов для повышения их эффективности: покрытие тканей слоем полимера [4], использование неньютоновских жидкостей [5–6], композиций с абразивными частицами [7–8] и др. Однако следует отметить, что пропитка латексом и аномально вязкими жидко-

стями многократно увеличивает поверхностную плотность тканей, покрытие композициями с абразивными частицами работает лишь для случая прокола/прореза. Сильная обработка (пропитка) тканей является нерациональной/неэффективной, т.к. значительно увеличивает поверхностную плотность ткани, затрудняет вытягивание нитей и приводит к их обрыву до начала вытягивания. Поэтому задача повышения коэффициента трения в ткани без разрыва нитей при вытягивании и минимальном увеличении поверхностной плотности является актуальной темой.

Для прогнозирования и изучения механизма вытягивания нитей из ткани используют как численные подходы, так и экспериментальные исследования [9–11]. В экспериментальных работах определяется максимальное усилие и энергия, затраченная при вытягивании нити, а также коэффициент трения и др. Однако экспериментальный подход не позволяет выявить влияние большого количества факторов при вытягивании нитей. Существует ряд численных подходов, которые учитывают механизмы вытягивания одиночной нити. Ткани моделируют сплошной средой [12], балочными [3], оболочечными [6, 11] и объемными [2, 9, 11] конечными элементами. Модели со сплошной средой не позволяют исследовать вытягивание нити из ткани. В моделях с балочными конечными элементами невозможно учесть контактное взаимодействие между нитями, их вытягивание. Для моделей тканей с объемными конечными элементами требуются большие вычислительные ресурсы. Отмеченные сложности моделирования (большие перемещения, анизотропия, односторонние фрикционные связи) не позволили получить удовлетворительное количественное согласие с экспериментом для всей диаграммы вытягивания [2, 6, 9–11]. Модели с оболочечными конечными элементами являются наиболее перспективными, т.к. позволяют учесть контактное взаимодействие нитей, позволяют учесть анизотропию нитей и требуют меньшие вычислительные ресурсы по сравнению с моделями из объемных конечных элементов.

В данной работе рассмотрена арамидная ткань P110 с поверхностной обработкой и без нее. Поверхностная обработка приводит к увеличению фрикционного взаимодействия нитей при минимальном утяжелении ткани. Разработаны численные модели тканей с разными видами поверхностных обработок. Данные модели позволяют прогнозировать диаграммы «нагрузка — перемещение» при вытягивании нити. Рассмотрено несколько способов распараллеливания задачи, рассчитан процесс вытягивания нити из ткани на суперкомпьютере в пакете программ LS-DYNA, проведено сравнение полученных результатов с экспериментальными данными, получены графики ускорения.

Статья организована следующим образом. В разделе 1 приведена постановка задачи. В разделе 2 описывается реализация задачи. В разделе 3 обсуждаются результаты исследований. В заключении суммируются основные результаты, полученные в данной работе.

1. Постановка задачи

В работе рассмотрена арамидная ткань полотняного переплетения типа P110 с поверхностной плотностью 110 г/кв.м. Для ткани P110 были рассмотрены несколько видов поверхностной обработки, которые позволяют изменить значения сил трения:

- 1) исходная ткань без поверхностной обработки;
- 2) канифоль сосновая В10 (производство ООО «Паяльные материалы», г. Рязань), утяжеление ткани 2,7 %;

- 3) профессиональная силиконовая смазка FELIX (производитель ООО «Тосол-Синтез-Инвест», г. Дзержинск), утяжеление ткани 19,4 %.

Проведено суперкомпьютерное моделирование вытягивания нити из ткани с поверхностной обработкой и без нее для прогнозирования зависимости нагрузки вытягивания от перемещения. Размеры образцов тканей в модели для исследования усилий вытягивания имели размеры 50×10 мм. Стороны ткани размером 50 мм были жестко закреплены. Скорость вытягивания нити в модели равна 0,1 м/с.

2. Реализация

Для решения задачи по вытягиванию нити из ткани в пакете программ LS-DYNA была разработана конечно-элементная модель ткани, состоящая из отдельных нитей. В расчетной модели нити имеют относительную свободу перемещения с возможностью вытягивания с учетом сухого трения. Нити в модели имеют прямоугольное поперечное сечение. На рис. 1 представлен повторяющийся элемент геометрии ткани, где отмечены номера ключевых точек. Координаты этих точек были введены в пакет программ ANSYS, после чего были построены соответствующие поверхности. Толщина нитей (оболочек) равна 50 мкм, ширина — 410 мкм. Далее набор поверхностей был размножен до получения необходимых размеров модели и построена сетка конечных элементов с одним оболочечным элементом по ширине нитей (рис. 2).

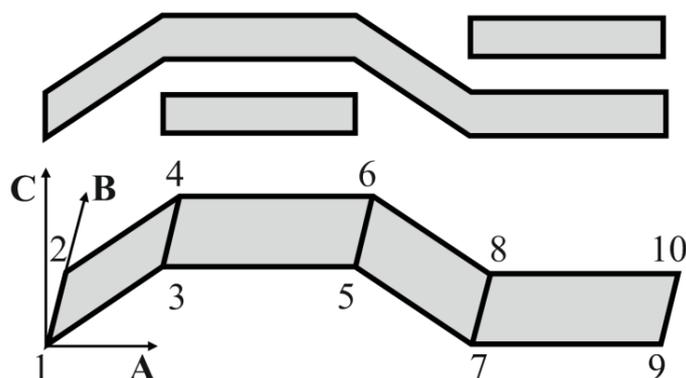


Рис. 1. Повторяющийся элемент геометрии нити

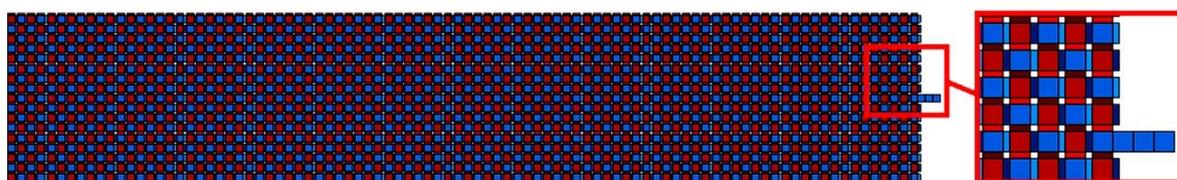


Рис. 2. Сетка конечных элементов

Для нитей был выбран ортотропный материал (*MAT_ENHANCED_COMPOSITE_DAMAGE) [13]. Характеристики материала приведены в таблице 1. Значения модулей упругости E_B , E_C , а также модулей сдвига G_{AB} , G_{BC} были приняты на порядок меньше модуля упругости E_A (определен экспериментально) согласно рекомендации [14]. Модели с оболочечными конечными элементами с одной точкой интегрирования (мембраны) не учитывают изгиб. Однако арамидные нити в ткани P110 состоят

из множества волокон диаметром 10–15 мкм с небольшой круткой и имеют слабое сопротивление изгибу. Поэтому для учета изгиба в модели было использовано три точки интегрирования по толщине. Также следует отметить, что на величину сопротивления изгибу влияет значение трансверсального модуля сдвига G_{CA} . Этот параметр был найден из предварительных расчетов сравнением расчетных и экспериментальных зависимостей нагрузки от перемещения при вытягивании нити из ткани без поверхностной обработки.

Таблица 1

Характеристики материала нитей

Параметр	Обозначение	Величина
Модули упругости, Па	E_A	$1,4 \cdot 10^{11}$
	E_B, E_C	$1,4 \cdot 10^{10}$
Плотность, кг/м ³	ρ	1 440
Коэффициент Пуассона	$\mu_{AB}, \mu_{BC}, \mu_{AC}$	0,001
Модули сдвига, Па	G_{AB}, G_{BC}	$1,4 \cdot 10^{10}$
	G_{CA}	$4 \cdot 10^7$

Контакт объектов задавали командой *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE. Для ткани без обработки статический коэффициент трения был определен экспериментально. Поверхностная обработка тканей в моделях учитывалась соответствующим изменением величины статического коэффициента сухого трения (таблица 2).

Таблица 2

Коэффициенты сухого трения

Тип пропитки	Значения коэффициентов сухого трения
нет	0,174
Канифоль	0,261
Силиконовая смазка	0,127

В расчете была использована явная схема интегрирования и опция mass scaling [13] позволяющая уменьшить время расчета задачи. Для ускорения расчета скорость вытягивания нити в модели была принята равной 0,1 м/с. При этом кинетическая энергия была пренебрежимо мала по сравнению с двумя основными формами диссипации энергии: внутренней энергии (internal energy) и энергии, затраченной на трение (sliding interface energy).

Экспериментальные кривые вытягивания нити из арамидной ткани были получены при квазистатических испытаниях на универсальной машине INSTRON 5882. Вытягиваемую нить закрепляли в тисочном захвате, поверхности которого покрыты тонким слоем эластомера, который не травмирует материал и обеспечивает эффективную передачу нагрузки на нить. На второй/другой тисочный захват были приклеены 2 полосы ленты из пластика на расстоянии 10 мм друг от друга, обеспечивающие защемление краев тканевого образца размерами 50×50 мм. Свободная зона, из которой происходило вытягивание нити, имела ширину 10 мм.

3. Результаты исследований

Расчеты были выполнены на суперкомпьютере «Торнадо ЮУрГУ» [15]. Были рассмотрены три различных способа декомпозиции модели для ткани без пропитки. В первом случае было использовано автоматическое распараллеливание модели (рис. 3 а), во втором случае модель была разбита на полосы, расположенные параллельно оси OY (рис. 3 б), в третьем случае — параллельно оси OX (рис. 3 в).

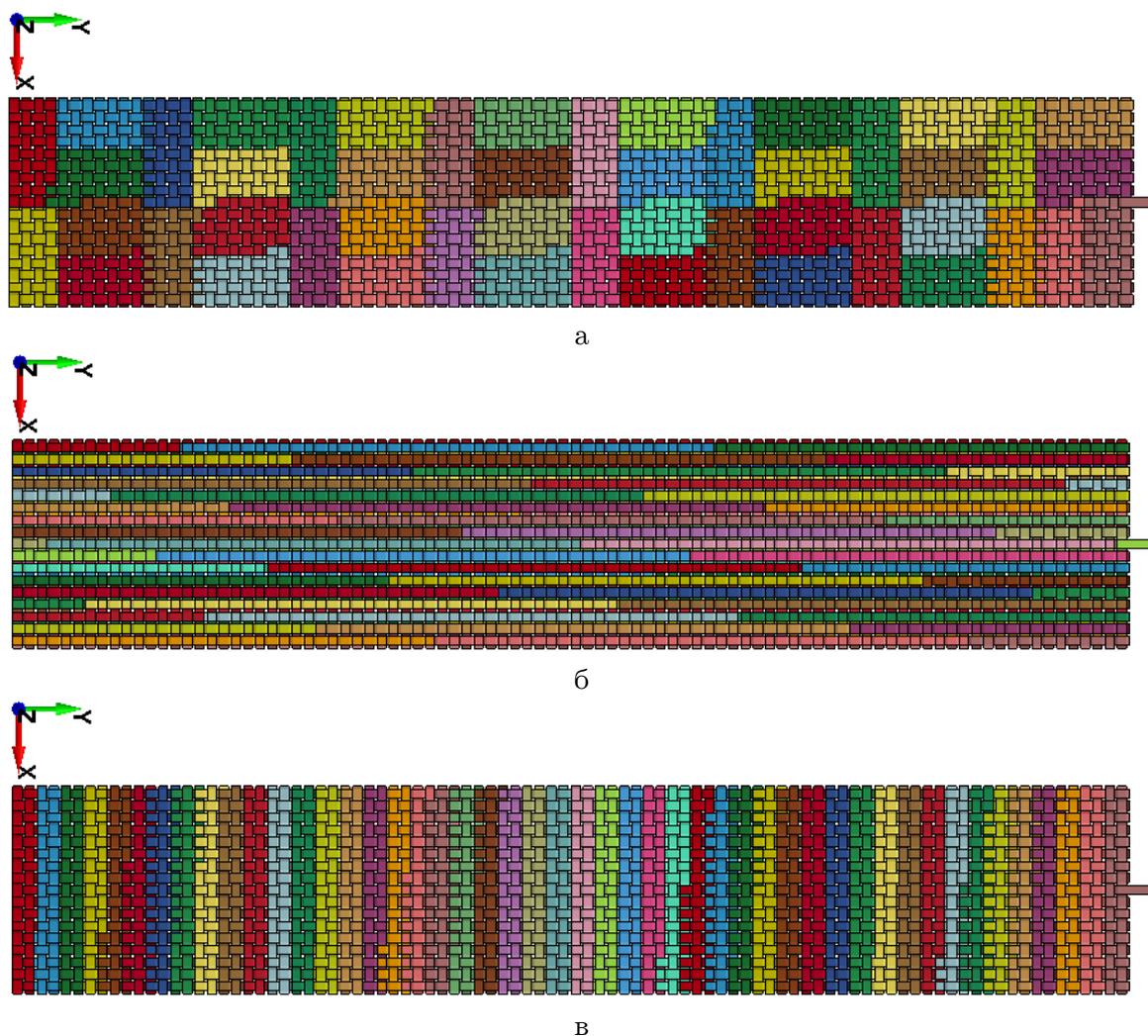


Рис. 3. Способы декомпозиции модели на 48 ядрах (а — автоматическое распараллеливание модели, б — полосы параллельно оси OY , в — полосы параллельно оси OX)

Графики ускорения для задачи по вытягиванию нити из ткани представлены на рис. 4. Данная задача была распараллелена до 48 ядер, это связано с имеющейся лицензией на пакет программ LS-DYNA. Время решения задачи на одном ядре составляет 20 680 с. Получено, что с увеличением числа ядер третий способ декомпозиции (распараллеливание на полосы, расположенные параллельно оси OX) дает более высокое ускорение.

На рис. 5 представлено сравнение расчетных и экспериментальных зависимостей «нагрузка — перемещение» при вытягивании нити из ткани с различными поверхностными обработками и без нее. Расчетные кривые лежат в пределах разброса эксперимен-

тальных данных. «Частоты» колебаний на расчетных и экспериментальных кривых совпадают и зависят лишь от геометрии переплетения ткани.

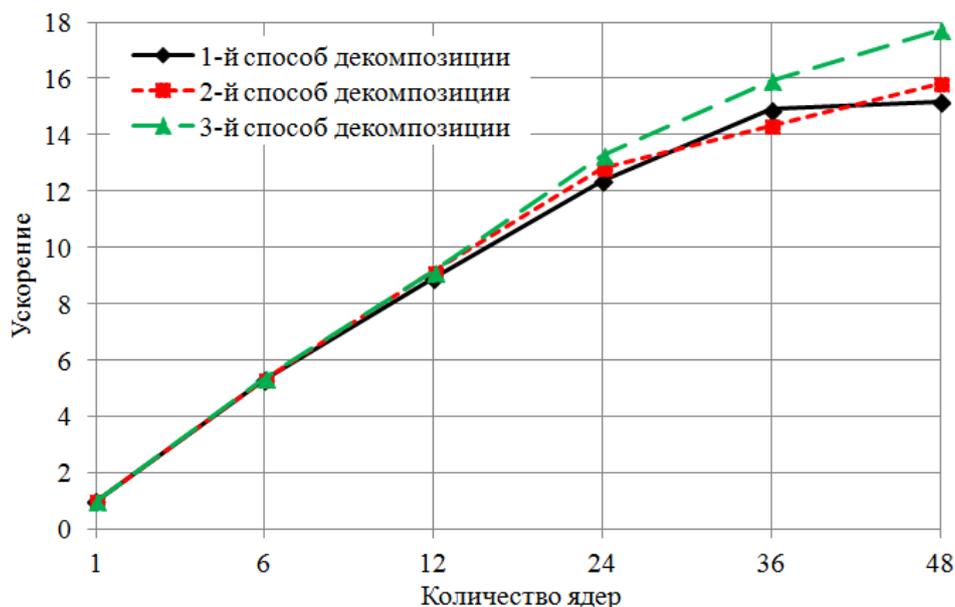


Рис. 4. Сравнение ускорений

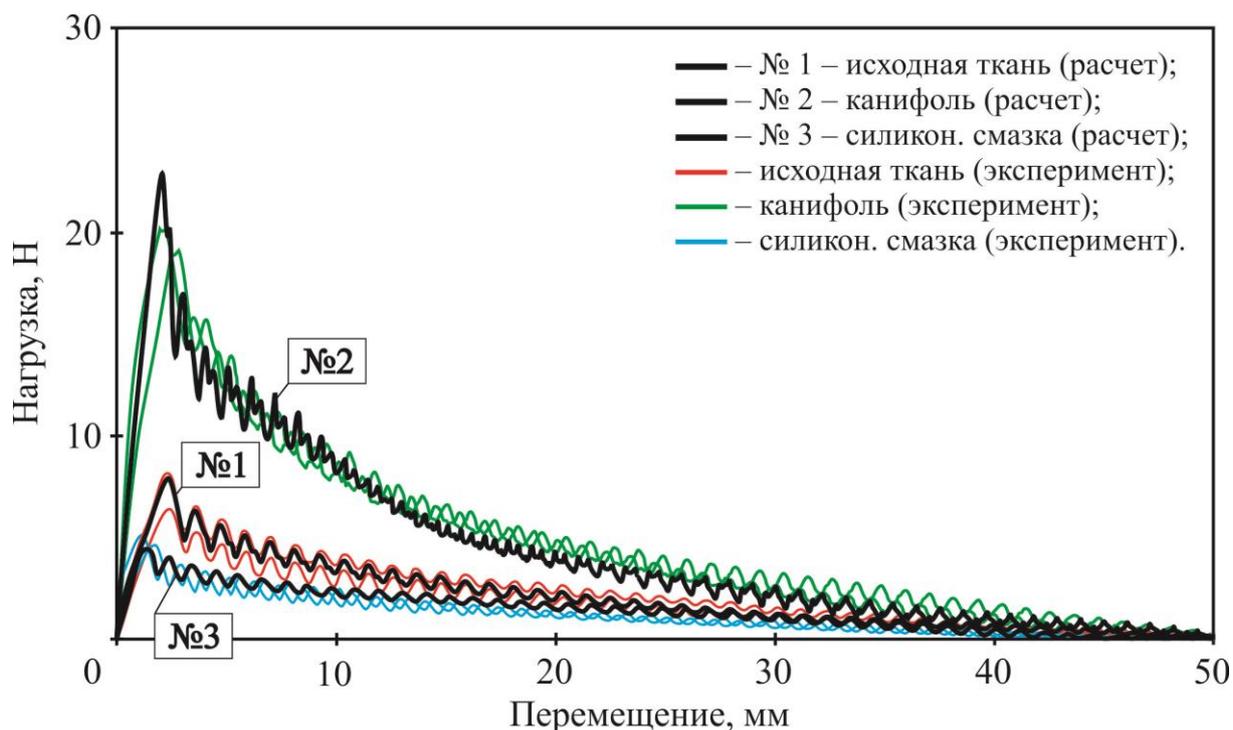


Рис. 5. Зависимости «нагрузка — перемещение» при вытягивании нити из ткани

Полученные кривые «нагрузка — перемещение» имеют два участка: 1) возрастания нагрузки до перемещения 2,5 мм и 2) снижения до нуля при полной вытяжке нити из ткани до 50 мм. На первом участке нить выпрямляется, при этом свободный конец нити еще неподвижен, на втором — нить начинает двигаться. Поверхностная обработка канифолью существенно повышает максимальное усилие вытягивания, увеличивая трение между нитями и повышая рассеяние энергии. Обработка ткани профессиональной сили-

коновой смазкой привела к уменьшению максимального усилия при вытяжке нити и энергии рассеяния на трение.

Заключение

Разработаны малопараметрические численные модели вытягивания нити из ткани с поверхностными обработками и без нее, позволяющие прогнозировать зависимости нагрузок от перемещений. Поверхностную обработку ткани в модели учитывали изменением одного параметра — коэффициента сухого трения. Расчетные и экспериментальные данные по изменению силы от перемещения хорошо согласуются между собой. Рассмотрены три способа декомпозиции модели и получены графики ускорения.

Разработанные в данной работе модели позволят прогнозировать поведение баллистических тканей с поверхностной обработкой при локальном ударе.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-08-01024 А.

Литература

1. Харченко Е.Ф., Ермоленко А.Ф. Композитные, текстильные и комбинированные бронематериалы. М.: ОАО ЦНИИСМ, 2013. 294 с.
2. Zhu D., Soranakom C., Mobasher B., Rajan S.D. Experimental Study and Modeling of Single Yarn Pull-Out Behavior of Kevlar[®] 49 Fabric // Composites: Part A. 2011. Vol. 42. P. 868–879. DOI: 10.1016/j.compositesa.2011.03.017.
3. Das S., Jagan S., Shaw A., Pal A. Determination of Inter-Yarn Friction and Its Effect on Ballistic Response of Para-Aramid Woven Fabric under Low Velocity Impact // Composite Structures. 2015. Vol. 120. P. 129–140. DOI: 10.1016/j.compstruct.2014.09.063.
4. Gawandi A., Thostenson E.T., Gillespie J.W.Jr. Tow Pullout Behavior of Polymer-Coated Kevlar Fabric // Journal of Materials Science. 2011. Vol. 46, No. 1. P. 77–89. DOI: 10.1007/s10853-010-4819-3.
5. Majumdar A., Butola B.S., Srivastava A. Development of Soft Composite Materials with Improved Impact Resistance Using Kevlar Fabric and Nano-Silica Based Shear Thickening Fluid // Materials and Design. 2014. Vol. 54. P. 295–300. DOI: 10.1016/j.matdes.2013.07.086.
6. Lee B.-W., Kim C.-G. Computational Analysis of Shear Thickening Fluid Impregnated Fabrics Subjected to Ballistic Impacts // Advanced Composite Materials. 2012. Vol. 21, No. 2. P. 177–192. DOI: 10.1080/09243046.2012.690298.
7. Mayo J.B.Jr., Wetzel E.D., Hosur M.V., Jeelani S. Stab and Puncture Characterization of Thermoplastic-Impregnated Aramid Fabrics // International Journal of Impact Engineering. 2009. Vol. 36. P. 1095–1105. DOI: 10.1016/j.ijimpeng.2009.03.006.
8. Соловьева Е.А., Курмашова И.А. Исследование механизма взаимодействия колющего оружия с защитными структурами на основе высокопрочных волокон // Вопросы оборонной техники. Серия 15. Композиционные неметаллические материалы в машиностроении. 2015. Вып. 4(179). С. 60–65.
9. Lopez-Galvez H., Rodriguez-Millan M., Feito N., Miguelez H. A Method for Inter-Yarn Friction Coefficient Calculation for Plain Weave of Aramid Fibers // Mechanics Research Communications. 2016. Vol. 74. P. 52–56. DOI: 10.1016/j.mechrescom.2016.04.004.

10. Tapie E., Guo Y.B., Shim V.P.W. Yarn Mobility in Woven Fabrics — a Computational and Experimental Study // International Journal of Solids and Structures. 2016. Vol. 80. P. 212–226. DOI: 10.1016/j.ijsolstr.2015.11.005.
11. Dong Z., Sun C.T. Testing and Modeling of Yarn Pull-Out in Plain Woven Kevlar Fabrics // Composites: Part A. 2009. Vol. 40. P. 1863–1869. DOI: 10.1016/j.compositesa.2009.04.019.
12. Valizadeh M., Lomov S., Ravandi S.A.H., Salimi M., Rad S.Z. Finite Element Simulation of a Yarn Pullout Test for Plain Woven Fabrics // Textile Research Journal. 2010. Vol. 80, No. 10. P. 892–903. DOI: 10.1177/0040517509346436.
13. LS-DYNA R7.0 Keyword User's Manual. LSTC. 2013. 2206 p.
14. Nilakantan G., Nutt S. Effects of Clamping Design on the Ballistic Impact Response of Soft Body Armor // Composite Structures. 2014. Vol. 108. P. 137–150. DOI: 10.1016/j.compstruct.2013.09.017.
15. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources // Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016). Arkhangelsk, Russia, March 29–31, 2016. CEUR Workshop Proceedings. 2016. Vol. 1576. P. 561–573.

Долганина Наталья Юрьевна, к.т.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Игнатова Анастасия Валерьевна, инженер, заочный факультет Политехнического института, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse170407

THE DEVELOPMENT OF COMPUTER MODELS OF BALLISTIC FABRIC WITH SURFACE TREATMENT

© 2017 N.Yu. Dolganina, A.V. Ignatova

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

E-mail: dolganinani@susu.ru, ignatovaav@susu.ru

Received: 06.10.2017

Ballistic fabrics are widely used in protective armor structures. Minimizing the mass of armored structures, reducing the kinetic energy of the bullet transmitted to the object located behind the armored panel (for reducing the amount of deflection of the rear side of the panel) are actual tasks for today. A significant part of the energy of the bullet is dissipated due to the work of frictional forces when pulling yarns from the fabric. Ability to predict the work of ballistic fabric when pulling yarn from the fabric will allow designing high-performance armored structures. Therefore, low-parametric numerical models of pulling a yarn from aramid fabric P110 of plain weave, as well as for this fabric with different types of surface treatment (rosin, silicone grease) were developed using the LS-DYNA software package. Surface treatment of the fabric allows you to change the coefficient of friction between threads with minimal weight gain. In the model, the surface treatment was accounted for by changing one parameter of the coefficient of dry friction. Several methods of parallelizing the problem of pulling a yarn from a fabric are considered. Numerical experiments were conducted to study the scalability of problem. The calculated curves of the load from displacement when drawing a thread from a fabric with and without surface treatment were obtained. The calculated results are in the range of the scatter of the experimental data.

Keywords: numerical methods, finite element method, modeling, aramid fabric, pull-out test, surface treatment, LS-DYNA.

FOR CITATION

Dolganina N.Yu., Ignatova A.V. The development of computer models of ballistic fabric with surface treatment. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. vol. 6, no. 4. pp. 91–100. (in Russian) DOI: 10.14529/cmse170407.

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. Kharchenko E.F., Ermolenko A.F. *Kompozitnye, tekstil'nye i kombinirovannye bronematerialy* [Composite, Textile and Combined Armor Materials]. Moscow, Publ. of TsNIISM, 2013. 294 p.
2. Zhu D., Soranakom C., Mobasher B., Rajan S.D. Experimental Study and Modeling of Single Yarn Pull-Out Behavior of Kevlar[®] 49 Fabric. *Composites: Part A*. 2011. vol. 42. pp. 868–879. DOI: 10.1016/j.compositesa.2011.03.017.
3. Das S., Jagan S., Shaw A., Pal A. Determination of Inter-Yarn Friction and Its Effect on Ballistic Response of Para-Aramid Woven Fabric under Low Velocity Impact. *Composite Structures*. 2015. vol. 120. pp. 129–140. DOI: 10.1016/j.compstruct.2014.09.063.
4. Gawandi A., Thostenson E.T., Gillespie J.W.Jr. Tow Pullout Behavior of Polymer-Coated Kevlar Fabric. *Journal of Materials Science*. 2011. vol. 46, no. 1. pp. 77–89. DOI: 10.1007/s10853-010-4819-3.
5. Majumdar A., Butola B.S., Srivastava A. Development of Soft Composite Materials with Improved Impact Resistance Using Kevlar Fabric and Nano-Silica Based Shear Thickening Fluid. *Materials and Design*. 2014. vol. 54. pp. 295–300. DOI: 10.1016/j.matdes.2013.07.086.
6. Lee B.-W., Kim C.-G. Computational Analysis of Shear Thickening Fluid Impregnated Fabrics Subjected to Ballistic Impacts. *Advanced Composite Materials*. 2012. vol. 21, No. 2. pp. 177–192. DOI: 10.1080/09243046.2012.690298.
7. Mayo J.B.Jr., Wetzel E.D., Hosur M.V., Jeelani S. Stab and Puncture Characterization of Thermoplastic-Impregnated Aramid Fabrics. *International Journal of Impact Engineering*. 2009. vol. 36. pp. 1095–1105. DOI: 10.1016/j.ijimpeng.2009.03.006.
8. Soloveva E.A., Kurmashova I.A. *Issledovanie mekhanizma vzaimodeistviia koliushchego oruzhiia s zashchitnymi strukturami na osnove vysokoprochnykh volokon* [Investigation of the Mechanism of Interaction of the Piercing Weapon with Protective Structures Based on High-Strength Fibers]. *Voprosy oboronnoi tekhniki. Seriya 15* [Defense Engineering Problems. Series 15]. 2015. no. 4(179). pp. 60–65. (in Russian)
9. Lopez-Galvez H., Rodriguez-Millan M., Feito N., Miguelez H. A Method for Inter-Yarn Friction Coefficient Calculation for Plain Wave of Aramid Fibers. *Mechanics Research Communications*. 2016. vol. 74. pp. 52–56. DOI: 10.1016/j.mechrescom.2016.04.004.

10. Tapie E., Guo Y.B., Shim V.P.W. Yarn Mobility in Woven Fabrics — a Computational and Experimental Study. *International Journal of Solids and Structures*. 2016. vol. 80. pp. 212–226. DOI: 10.1016/j.ijsolstr.2015.11.005.
11. Dong Z., Sun C.T. Testing and Modeling of Yarn Pull-Out in Plain Woven Kevlar Fabrics. *Composites: Part A*. 2009. vol. 40. pp. 1863–1869. DOI: 10.1016/j.compositesa.2009.04.019.
12. Valizadeh M., Lomov S., Ravandi S.A.H., Salimi M., Rad S.Z. Finite Element Simulation of a Yarn Pullout Test for Plain Woven Fabrics. *Textile Research Journal*. 2010. vol. 80, no. 10. pp. 892–903. DOI: 10.1177/0040517509346436.
13. LS-DYNA R7.0 Keyword User's Manual. LSTC. 2013. 2206 p.
14. Nilakantan G., Nutt S. Effects of Clamping Design on the Ballistic Impact Response of Soft Body Armor. *Composite Structures*. 2014. vol. 108. pp. 137–150. DOI: 10.1016/j.compstruct.2013.09.017.
15. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources. *Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016)*. Arkhangelsk, Russia, March 29–31, 2016. CEUR Workshop Proceedings. 2016. vol. 1576. pp. 561–573.

СВЕДЕНИЯ ОБ ИЗДАНИИ

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

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 следующим отраслям и группам специальностей: 05.13.00 – информатика, вычислительная техника и управление; 25.00.00 – науки о Земле (№ 421).

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

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 редакции 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.**