2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

doi 10.26089/NumMet.v24r107

УДК 517.968;
517.444;
519.642;
519.642.3

Онтологический анализ предметной области цифровой платформы Algo500

А. С. Антонов
Московский государственный университет имени М. В. Ломоносова,
Научно-исследовательский вычислительный центр, Москва, Российская Федерация
ORCID: 0000-0003-2820-7196, e-mail: asa@parallel.ru

Р. В. Майер
Московский государственный университет имени М. В. Ломоносова,
Научно-исследовательский вычислительный центр, Москва, Российская Федерация
ORCID: 0000-0002-3737-5530, e-mail: maier.rcc@gmail.com

Аннотация: Проект создания цифровой платформы Algo500 направлен на решение задачи совместного анализа свойств алгоритмов и особенностей архитектур суперкомпьютеров. В статье на основе методологии онтологического анализа рассматриваются и предлагаются понятия, модели и метамодели данных, обосновываются подходы к описанию некоторых понятий из мира высокопроизводительных вычислений (HPC), устанавливаются новые требования к моделям данных, которые должны обеспечить выполнение задач, поставленных при создании платформы Algo500.

Ключевые слова: онтологический анализ, Algo500, AlgoWiki, CompZoo, PerfData, задача, метод, алгоритм, реализация, суперкомпьютер, рейтинг, суперкомпьютерный эксперимент.

Благодарности: Результаты получены в Московском государственном университете имени М. В. Ломоносова при финансовой поддержке РНФ (договор № 20–11–20194). Работа выполнена с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М. В. Ломоносова.

Для цитирования: Антонов А.С., Майер Р.В. Онтологический анализ предметной области цифровой платформы Algo500 // Вычислительные методы и программирование. 2023. **24**, № 1. 89–114. doi 10.26089/NumMet.v23r107.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Ontological analysis of the subject area of the Algo500 digital platform

Alexander S. Antonov
Lomonosov Moscow State University, Research Computing Center, Moscow, Russia
ORCID: 0000-0003-2820-7196, e-mail: asa@parallel.ru

Rostislav V. Maier
Lomonosov Moscow State University, Research Computing Center, Moscow, Russia
ORCID: 0000-0002-3737-5530, e-mail: maier.rcc@gmail.com

Abstract: The Algo500 digital platform project is aimed at solving the problem of joint analysis of the properties of algorithms and features of supercomputer architectures. In this paper, based on the methodology of ontological analysis, concepts, models and metamodels of data are considered and proposed, approaches to the description of some concepts from the world of high-performance computing (HPC) are substantiated, new requirements for data models are established, which should ensure the fulfillment of the tasks set when creating the Algo500 platform.

Keywords: ontological analysis, Algo500, AlgoWiki, CompZoo, PerfData, problem, method, algorithm, implementation, supercomputer, rating, supercomputer experiment.

Acknowledgements: The results were obtained in Lomonosov Moscow State University with the financial support of the Russian Science Foundation (agreement No. 20–11–20194). The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

For citation: A. S. Antonov and R. V. Maier, "Ontological analysis of the subject area of the Algo500 digital platform," Numerical Methods and Programming. **24** (1), 89–114 (2023). doi 10.26089/NumMet.v24r107.

1. Введение. Проект создания цифровой платформы Algo500 направлен на решение задачи совместного анализа свойств алгоритмов и особенностей архитектур суперкомпьютеров [1]. Инструменты цифровой платформы должны позволять комплексно исследовать качество цепочки "Задача–Метод–Алгоритм–Реализация–Суперкомпьютер" на уровне любого ее звена [2]. В рамках проекта Algo500 предлагается практический подход, ориентированный не на отдельные частные оценки эффективности, а позволяющий решить проблему в целом, изучая свойства реализации любого алгоритма на любом суперкомпьютере.

Определение Algo500 как цифровой платформы подчеркивает масштабность и сложность задачи. Создаваемая платформа должна на основе единых подходов: 1) объединять данные о любых алгоритмах и архитектурах компьютеров; 2) позволять проводить анализ свойств любого алгоритма применительно к любой архитектуре суперкомпьютера на основании общего для всех алгоритмов подхода; 3) позволять Internet-сообществу коллективно дополнять и уточнять базу данных алгоритмов и их реализаций; 4) вносить динамические характеристики выполнения реализаций алгоритмов на различных вычислительных системах; 5) генерировать по запросу различные списки, ранжированные по параметрам, регистрируемым в платформе Algo500.

Структура цифровой платформы Algo500, основанная на подсистемах хранения данных, изначально состояла из трех взаимосвязанных компонент: AlgoWiki, CompZoo и PerfData [1]. В данной работе мы предлагаем ряд функций, связанных с построением рейтинговых списков, вынести в отдельную, четвертую компоненту RatingLists.

Первой и наиболее полно реализованной в настоящий момент является компонента AlgoWiki — открытая энциклопедия параллельных свойств алгоритмов, построенная на основе движка MediaWiki и предназначенная для создания, хранения, публикации описаний алгоритмов, их свойств и реализаций для различных классов архитектур суперкомпьютеров [3, 4]. Для описания алгоритмов предложена схема, задающая структуру описания и учитывающая наряду с такими классическими свойствами алгоритмов, как последовательная сложность, дополнительно параллельную структуру и параллельную сложность

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

алгоритмов, детерминизм и другие свойства [5]. Описания алгоритмов, содержащиеся в AlgoWiki, классифицированы по схеме "Задачи–Методы–Алгоритмы–Реализации" [2, 5–7], позволяющей быстро ориентироваться во всей совокупности алгоритмов за счет логического объединения алгоритмов в группы методов и задач.

Компонента CompZoo предназначена для хранения структурированных описаний архитектур суперкомпьютеров с указанием наиболее важных характеристик, оказывающих непосредственное влияние на производительность, энергоэффективность, время выполнения и другие показатели суперкомпьютеров. Одним из основных требований к CompZoo является необходимость хранения значительно более подробных описаний архитектур суперкомпьютеров, чем описания, присутствующие в таких наиболее известных суперкомпьютерных рейтингах, как Top500 или Graph500 [1, 8]. Более высокая детализация описаний архитектур суперкомпьютеров позволит проводить анализ параллельных свойств реализаций алгоритмов при их запуске на подмножестве вычислительных узлов, входящих в состав суперкомпьютера.

Основой компоненты PerfData является репозиторий данных, в котором наряду с программной реализацией алгоритмов и конфигурационными файлами для конкретной вычислительной системы хранятся сведения о выбранной для запуска программы совокупности вычислительных узлов суперкомпьютера, данные о входных аргументах, параметрах и результатах прогона программы [9, 10]. Наличие полной информации о параметрах запуска программной реализации алгоритма, включающей саму программу и данные, в принципе позволяет считать компонент PerfData своего рода доказательной базой платформы Algo500, доступной всем исследователям.

Развитие массового производства вычислительных систем в XX веке сопровождалось разработкой методологии сравнения суперкомпьютеров по производительности, в частности с целью определить направления дальнейшего развития. Идея сопоставления архитектур суперкомпьютеров по производительности, которую они достигают при решении систем линейных уравнений с плотной матрицей на основе теста Linpack (с наиболее известной реализацией HPL), воплотилась в рейтинге Top500 [11], являющимся в настоящее время наиболее упоминаемым при сравнении суперкомпьютеров. Развитие идеи сравнительного анализа суперкомпьютеров и алгоритмов привело к появлению рейтингов Graph500 [12] и HPCG [13], которые показывают производительность суперкомпьютеров при вычислениях принципиально иных задач, чем в тесте Linpack, а именно в обработке больших графов и разряженных матриц методом сопряженных градиентов соответственно. Рейтинг Green500 [14] характеризует суперкомпьютеры по их энергоэффективности при выполнении теста Linpack.

Несмотря на то что оценки производительности, например полученные в тесте Linpack, зачастую слабо коррелируют с показателями вычислительных процессов при расчетах реальных задач, методология сравнительного анализа суперкомпьютеров в настоящее время, как и прежде, основывается на проведении различных тестов [15, 16].

Одной из основных целей масштабируемой цифровой платформы Algo500 является предоставление пользователю рейтинговых списков, построенных по различным метрикам вычислительного процесса, которые получаются при прогонах[1] реализаций алгоритмов, представленных в AlgoWiki, на полных или частичных конфигурациях суперкомпьютеров, внесенных в CompZoo [15–17]. Формирование запросов к Algo500 для предоставления рейтинговых списков и иной связанной с ними аналитической информации в настоящее время отнесено к компоненте RatingLists.

Разработка программного обеспечения платформы Algo500 опирается на создание прототипов, позволяющих тестировать функциональность платформы как в целом, так и по ее отдельным компонентам AlgoWiki, CompZoo, PerfData и RatingLists [7]. Каждый прототип позволил отработать определенные подходы в создании описаний алгоритмов в виде Wiki-страниц, визуализации классификации алгоритмов, построении ядра компоненты CompZoo, конструктора запросов к CompZoo и т.д.

Однако возрастающая сложность прототипов программных компонент платформы Algo500 сделала актуальным проведение онтологического анализа предметной области данного проекта. В статье на основе методологии онтологического анализа [18, 19] рассматриваются и предлагаются понятия, модели и метамодели данных, обосновываются подходы к описанию некоторых понятий из мира высокопроизводительных вычислений (HPC), устанавливаются новые требования к моделям данных, которые должны обеспечить выполнение ранее сформулированных задач для платформы Algo500 [1, 4, 5, 8, 9, 15, 16].

[1]Термины "прогон" и "запуск" — профессионализмы в области высокопроизводительных вычислений, семантически связанные с выполнением программ на суперкомпьютерах.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

2. Роли пользователей и функциональность Algo500. С целью обеспечить устойчивость развития классификации алгоритмов и защиты информации от деструктивных действий Internet-пользователей в AlgoWiki были введены роли, ограничивающие возможность внесения изменений: 1) `guest` — незарегистрированный посетитель энциклопедии AlgoWiki, обладающий правом просмотра классификации алгоритмов и выбора wiki-статей; 2) `pma_tree` — зарегистрированный пользователь, имеющий право предлагать изменения в классификации алгоритмов; 3) `pma_root` — зарегистрированный пользователь, осуществляющий экспертизу и имеющий право утверждения или отклонения предложенных пользователями изменений (модератор) [7]. Ограничение функций пользователей за счет введения ролей имеет смысл при реализации концепции *изолированного представления* контента [7], которая предусматривает для пользователя с ролью `pma_tree` возможность внесения произвольного числа изменений в классификацию алгоритмов, видимых только этому пользователю до подтверждения модератором. Концепции изолированного представления контента и ролей пользователей распространяются на весь проект Algo500.

Перечисленные роли и связанные с ними права позволяют решить прагматические задачи платформы Algo500, но не имеют прямого отношения к заявленной цели проведения совместного анализа параллельных алгоритмов и архитектур суперкомпьютеров.

Отнесем к роли *“исследователь”* возможность использовать функционал платформы Algo500 для решения задач, требующих аналитической и синтетической мыслительной деятельности. Такие задачи в своей основе сводятся к определению понятий, сравнению, абстрагированию, систематизации и классификации, т.е. к тем операциям процесса познания, которые, в частности, изучаются логикой [20–22]. Разумеется, в задачи программного обеспечения платформы Algo500 не входит автоматизация процессов мыслительной деятельности: исследователь оперирует образами, представлениями, абстракциями, понятиями, относящимися к параллельным алгоритмам и суперкомпьютерам, которые непросто выразить даже на естественном языке, тем более создать их описание в рамках ограничений, присущих моделям данных в Algo500. Несмотря на это, проблема создания программного обеспечения и разработки функционала платформы Algo500, учитывающих логику исследователя и не противоречащих ей, является актуальной.

Из всех мыслительных операций наиболее просто программируется сравнение объектов, под которым подразумеваются как технические операции выборки, сортировки и фильтрации, так и более сложное сравнение, например описаний суперкомпьютеров. В логике установлено, что имеет смысл сравнение понятий, относящихся к одному роду (к общему универсуму). Поиск способа описания различных суперкомпьютеров, позволяющего проводить осмысленные сравнения, является одной из базовых задач при разработке программного обеспечения платформы Algo500.

Задача совместного анализа параллельных алгоритмов и суперкомпьютеров на платформе Algo500 с точки зрения реализации логических основ исследовательской деятельности, насколько мы знаем, рассматривается впервые. В дальнейшем мы полагаем, что любой пользователь платформы Algo500, в том числе `guest`, является исследователем, а обозначенный выше функционал, относящийся к роли “исследователь” — базовым по отношению к другим функциям Algo500.

Эффективное использование платформы Algo500 в научных изысканиях возможно при условии сохранения в течение длительного времени достаточно общих подходов к описаниям алгоритмов, архитектур суперкомпьютеров и к репозиторию программ, которые оставались бы инвариантными по отношению к быстроразвивающейся области высокопроизводительных вычислений (HPC). Для пользователей платформы Algo500 необходима такая совокупность понятий, которая, с одной стороны, должна позволять полноценно описывать предметную область, а с другой — оставаться консервативной, обеспечивая поколения исследователей инструментом с неизменной идеологией. Разработчику программного обеспечения платформы Algo500 необходимо использовать выделенную совокупность понятий как систему, а пользователю с ролью исследователя — понимать ее.

3. Модели данных в AlgoWiki.

3.1. Описание алгоритмов и их реализаций. Открытая энциклопедия параллельных свойств алгоритмов AlgoWiki содержит большое количество Wiki-страниц описаний алгоритмов и их программных реализаций, которые по содержанию соответствуют уровню научных статей. Создание новых и редактирование существующих описаний доступно зарегистрированным пользователям энциклопедии AlgoWiki. В работе [5] предложены структурные схемы (планы), следуя которым, исследователь, описывая новые

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

алгоритмы и реализации, может максимально полно раскрыть их параллельные свойства. Общая структура описаний облегчает исследователю сравнение родственных алгоритмов.

Описание алгоритма должно быть единственным и отделенным от описаний программных реализаций для суперкомпьютеров, принадлежащих различным классам архитектур. Исследователю предлагается описывать свойства алгоритмов в AlgoWiki по следующей структурной схеме:

1. Общее описание алгоритма.
2. Математическое описание алгоритма.
3. Вычислительное ядро алгоритма.
4. Макроструктура алгоритма.
5. Схема реализации последовательного алгоритма.
6. Последовательная сложность алгоритма.
7. Информационный граф алгоритма.
8. Ресурс параллелизма алгоритма.
9. Входные и выходные данные алгоритма.
10. Свойства алгоритма.

Описание программной реализации алгоритма должно явно показывать связь между свойствами программной реализации алгоритма и архитектурой компьютера, на которой она выполняется. При описании свойств программной реализации рекомендуется следовать следующему плану:

1. Ссылки на репозиторий программного обеспечения PerfData и другие источники.
2. Локальность данных и вычислений.
   - 2.1. Локальность реализации алгоритма.
     - 2.1.1. Структура обращений в память и качественная оценка локальности.
     - 2.1.2. Количественная оценка локальности.
3. Масштабируемость алгоритма и его реализации.
   - 3.1. Масштабируемость алгоритма.
   - 3.2. Масштабируемость реализации алгоритма.
4. Динамические характеристики и эффективность реализации алгоритма.
5. Результаты прогонов.

Отметим, что, с одной стороны, структурные схемы описаний алгоритмов и реализаций являются примером инвариантных понятий, общих для описаний любых алгоритмов и реализаций в энциклопедии AlgoWiki, с другой — они могут быть скорректированы средствами AlgoWiki вслед за развитием научной мысли.

3.2. Модель данных для классификации алгоритмов в AlgoWiki. Для систематизации описаний алгоритмов в AlgoWiki была создана классификация по схеме “Задачи–Методы–Алгоритмы–Реализации” [2, 6], которая соответствует принятому в научном сообществе разделению алгоритмов по задачам из различных разделов науки. Согласно современным подходам к проблеме классификации, отражающим опыт построения классификаций в различных науках, классификация может служить инструментом научного познания, как это было, например, в биологии и химии, но может остаться описательно-распознавательной [23, С. 32–42]. Изучение философских [23] и практических [24] аспектов построения классификаций позволяет констатировать, что предложенная в AlgoWiki классификация является искусственной и потенциально может быть скорректирована в будущем.

Рассмотрим модель данных, принятую в AlgoWiki для хранения классификации, и предоставляемые исследователю возможности по ее изменению. В основе схемы данных лежит ациклический ориентированный граф

$$G^a =< V^a, E^a >, \tag{1}$$

в котором V^a — множество вершин, E^a — множество связей между вершинами. Применение ациклического ориентированного графа позволяет реализовать различные принципы структурирования, лежащие в основе классификации. Построение классификации по принципу логического деления понятия [20], широко использующееся в классификации математических понятий, моделируется иерархической древовидной структурой, которая является частным случаем ациклического ориентированного графа. Использование фасетного или фасетно-перечислительного методов построения классификации [24] приводит к графовой структуре.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Классификация алгоритмов по схеме "Задачи–Методы–Алгоритмы–Реализации" относится к классу фасетно-перечислительных [24]. Множество классифицирующих признаков в этой схеме ограничено набором из задач, методов, алгоритмов и реализаций, т.е. для множества вершин V^a (1) справедливо

$$V^a = P \cup M \cup A \cup I, \tag{2}$$

где P, M, A, I — множества задач, методов, алгоритмов и их реализаций. Кроме классифицирующего признака каждый элемент из V^a обладает булевым флагом, определяющим существование или отсутствие Wiki-страницы у данного элемента, и текстовыми полями, содержащими название элемента на русском и английском языках.

Множество связей E^a (1) между элементами множеств P, M, A, I моделирует отношения подчиненности элементов в соответствии с логикой классификации. Множество связей E^a удобно представить следующим образом:

$$E^a = E_P \cup E_M \cup E_A \cup E_I, \tag{3}$$

где

$$E_P = \{(v_1, v_2) :\ v_1 = \emptyset,\ v_2 \in P \text{ или } v_1 \in P,\ v_2 \in P \cup M \cup A;\ v_1 \neq v_2\},$$
$$E_M = \{(v_2, v_3) :\ v_2 \in M,\ v_3 \in M \cup A;\ v_2 \neq v_3\},$$
$$E_A = \{(v_3, v_4) :\ v_3 \in A,\ v_4 \in A \cup I;\ v_3 \neq v_4\},$$
$$E_I = \{(v_4, v_5) :\ v_4 \in I;\ v_5 = \emptyset\}.$$

Отношения E_P, E_M, E_A, E_I содержат ограничения на операции с элементами классификации, например не допускается начинать классификацию с метода, добавлять реализацию алгоритма к проблеме или методу, добавлять дочерний элемент к реализации и т.д. Также на множество связей E^a накладывается ограничение, свойственное ациклическим графам: должны отсутствовать последовательности связей вида $(v_1, v_2), (v_2, v_3), \ldots, (v_n, v_1)$.

С графом G^a реализованы операции по изменению полей, входящих в вершины, и по изменению структуры графа: добавлению и удалению вершин, копированию и переносу подграфов [7]. Как отмечалось в разделе 2, все изменения, выполненные пользователем над графом G^a, проходят экспертизу и утверждение у модератора. Для того чтобы пользователь мог продолжать работать с произведенными изменениями, была реализована концепция изолированного представления [7]. Изолированное представление позволяет не только вносить изменения, но и разрабатывать иную классификацию в рамках ограничения (2), которое фактически определяет нерасширяемый словарь терминов классификации AlgoWiki.

4. Метамодель данных компоненты CompZoo.

4.1. Проблема создания описаний суперкомпьютеров. Регулярно публикуемый с 1993 г. международный рейтинг наиболее мощных суперкомпьютеров мира Top500 [11] представляет описания суперкомпьютеров в табличной форме с фиксированным набором характеристик: 1) позиция в рейтинге; 2) название суперкомпьютера (является ссылкой на более развернутое описание); 3) количество вычислительных ядер; 4) максимальная производительность на тесте Linpack; 5) пиковая производительность; 6) потребляемая мощность. Более развернутое описание суперкомпьютера, доступное по ссылке, в дополнение к перечисленным характеристикам содержит тип интерконнекта, год запуска в эксплуатацию, сведения о программном обеспечении и т.п. (список характеристик не является постоянным). Российский рейтинг Top50 [25], представляющий список самых производительных суперкомпьютеров России и стран СНГ, содержит более подробные описания, чем в Top500.

Как отмечалось в [1, 8], для выполнения совместного анализа параллельных алгоритмов и архитектур суперкомпьютеров требуется значительно более детальное описание суперкомпьютеров, чем в рейтингах Top500 и Top50. В НИВЦ МГУ была предпринята попытка определить расширенный набор характеристик суперкомпьютеров, которая показала неприменимость такого подхода в нашем случае: во-первых, широкий спектр целей исследований не позволил установить однозначный, фиксированный набор характеристик; во-вторых, развитие суперкомпьютерных технологий в будущем с высокой вероятностью потребует расширить список новыми характеристиками, которые будут актуальны для современных суперкомпьютеров и окажутся лишенными смысла для суперкомпьютеров прошлых поколений. В результате возникло понимание, что фундаментальными требованиями к CompZoo должны быть 1) предоставление исследователю возможности создавать *вариативные наборы характеристик суперкомпьютеров* с целью

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

накопления необходимого фактического материала; 2) обеспечение исследователя возможностью *сравнения суперкомпьютеров по характеристикам* при условии изменения списков характеристик.

Обозначим U_0 множество всех суперкомпьютеров, $U_\text{m} \subset U_0$ — множество моделей[2] суперкомпьютеров. Известно, что модели суперкомпьютеров — элементы множества U_m — уникальны по своим характеристикам:

$$U_\text{m} = \{x : P(\mathcal{F}(x))\}, \tag{4}$$

где x — переменная, обозначающая модель суперкомпьютера, $\mathcal{F} =< f_1, f_2, \ldots, f_N >$ — кортеж (упорядоченный набор) предметных функторов, определенных на U_m, P — предикат, истинность которого определяется соответствием значений предметных функторов из $\mathcal{F}$ характеристикам x. Каждому предметному функтору из $\mathcal{F}$ соответствует предметная функция, записанная на естественном языке: например, функтор f_2 может обозначать функцию “пиковая производительность”, f_3 — “год выпуска” и т.д. Будем считать, что f_1 обозначает одноместную функцию (свойство) “название”. Областью определения для всех функций является множество U_m, области значений каждой функции индивидуальны (см. раздел 4.4). Предикат P определяется как

$$P(x) = \bigwedge_{i=1}^{N} p_i(f_i(x), \alpha_i), \quad \alpha_i \in A, \quad N = |\mathcal{F}|, \tag{5}$$

где α_i — значения характеристик суперкомпьютера (константы), A — множество значений всех характеристик суперкомпьютера, включая неопределенное значение *undefined*, $p_i(f_i(x), \alpha_i)$ — двухместные предикаты со схемой $p_i(x, \alpha_i) = (f_i(x) = \alpha_i)$. Для приведенной выше в качестве примера функции f_2 и константы α_2 можно построить предикат p_2, которому соответствует фраза естественного языка “пиковая производительность x равна α_2”.

Формулы (4), (5), представляющие описание суперкомпьютеров по характеристикам, аналогичны символической записи понятия в современной логике [20, 21]. Несмотря на формальную аналогию, мы считаем некорректным использовать термин “понятие” из логики применительно к описаниям суперкомпьютеров в CompZoo, так как полнота набора характеристик $\mathcal{F}$, формируемого исследователем, никогда не достигнет содержания такого сложного понятия, как суперкомпьютер. Различие между моделями суперкомпьютеров — объективная реальность, которую нельзя подвергнуть сомнению по причине неполноты набора характеристик, представленного в описании CompZoo.

Пользуясь введенными обозначениями, требование к вариативности набора характеристик можно выразить следующим образом: *кортеж $\mathcal{F}$ должен изменяться в соответствии с целями исследования.*

Таким образом, для того чтобы создать описание суперкомпьютера в CompZoo, исследовать должен сначала определить набор характеристик, следуя определенным правилам, которые составляют *метамодель данных* компоненты CompZoo. Концептуально метамодель описывается в разделах 4.2–4.5.

4.2. Системный подход при описании суперкомпьютеров в компоненте CompZoo. Суперкомпьютер — сложная техническая система, для упрощения описания которой исследователь вынужден перейти к модельным представлениям о суперкомпьютерах и сохранять в Algo500 редуцированные описания суперкомпьютеров в рамках некоторого набора характеристик выбранной модели [26, 27].

Трудно представить существование такой модели суперкомпьютера, которая удовлетворяла бы любой цели исследователей и набор параметров которой корректно описывал бы любую архитектуру суперкомпьютеров, существовавших ранее, существующих в настоящее время и тем более будущих поколений. Если принять в качестве необходимой для реализации в CompZoo возможность создания описаний суперкомпьютеров, удовлетворяющих различным целям исследований и позволяющих хранить характеристики современных суперкомпьютеров, а также как прошлых, так и будущих поколений, то некоторые из наиболее общих моделей описания систем можно исключить из рассмотрения, не доводя до этапа программной реализации.

Не удовлетворяет сформулированному требованию *модель черного ящика*, широко применяемая, например в теории электрических цепей или в теории автоматического управления, поскольку она предполагает определенную статичность наборов входных, выходных параметров и законов, устанавливающих взаимосвязи между этими параметрами.

[2]Слово “модель” в статье используются в двух значениях: 1) как синоним для “тип устройства”; 2) как обозначение упрощенного представления о сложной системе.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Применение *модели структуры системы*, описывающей связи между подсистемами, является неоднозначным и неоправданно трудоемким. Для такой сложной технической системы, какой является суперкомпьютер, можно построить различные модели структуры системы: например, структурную модель инженерной подсистемы, структурную схему электропитания или структуру информационных связей между компонентами вычислительной системы и т.д. Из перечисленных нас главным образом интересует структура информационных связей, отражающая топологию вычислительной сети, которая, несмотря на существенное влияние на производительность суперкомпьютера, является не единственным фактором, заслуживающим внимания. Использование графовых моделей для представления полной структуры современных суперкомпьютеров с кластерной архитектурой практически невозможно вследствие большой аддитивной сложности кластеров: количество вычислительных узлов, объединенных коммуникационной сетью, может достигать нескольких тысяч. Попытка упростить модель структуры системы, представляя однотипные вычислительные узлы одной вершиной графа, разрушит саму модель.

В качестве основы для описания суперкомпьютеров в компоненте CompZoo примем *модель состава системы*, которая представляет собой список подсистем, образованный при выделении в системе иерархии подсистем необходимой степени вложенности [26]. Аналогом модели состава системы может выступать некоторое подмножество подсистем, представленных в спецификации суперкомпьютера. Такая аналогия носит необязательный характер для компоненты CompZoo, в которой декомпозиция прежде всего определяется целями исследования. Например, в составе суперкомпьютеров кластерной архитектуры нередко выделяют “группы вычислительных узлов” и “вычислительные узлы”. Эти подсистемы вполне могут присутствовать в качестве сборок в конструкторской документации. Однако для подсистем “CPU” (центральный процессор) и “CPU core” (вычислительное ядро) соответствия со спецификацией не наблюдается, поскольку для конструктора CPU является неделимым устройством, а в рамках исследования, проводимого на платформе Algo500, может оказаться полезным описание свойств ядра, кэш-памяти и контроллера памяти, входящих в состав CPU.

Анализ описаний современных суперкомпьютеров, представленных на официальных сайтах, показал, что модель состава системы (с определенной степенью детализации) используется исключительно широко и деление суперкомпьютера на подсистемы практически всегда имеется в описании суперкомпьютера, выполненном его разработчиками.

Принимая модель состава системы в качестве основы для создания описаний суперкомпьютеров, мы считаем важным обратить внимание на принципиальные недостатки такой декомпозиции суперкомпьютеров:

1. Свойства системы невозможно свести к свойствам ее элементов в отдельности. Установлено, что состав и связи между подсистемами проявляются в виде новых качеств, которые имеются только у системы в целом. Это свойство систем получило название *эмерджентность* [26].
2. Неполнота описания суперкомпьютера, присущая модели состава систем, может спровоцировать исследователя на приписывание несуществующих качеств суперкомпьютеру, в частности, по причине схожести описаний.

На рис. 1 a приводятся небольшие фрагменты описаний суперкомпьютеров Ломоносов–2 [28–30] и Fugaku [28, 31] кластерной архитектуры в виде модели состава системы. Из документации на суперкомпьютер Ломоносов–2 известно, что вычислительные узлы в составе групп Compute, Test и Pascal построены на основе центральных процессоров Intel Xeon E5-2697 v3. В вычислительных узлах других групп используются иные процессоры. Возможность организации ссылок на уже описанные подсистемы, в частности на процессор Intel Xeon E5-2697 v3 с кэш-памятью уровней L1–L3 и контроллером памяти MCU, является естественным требованием к схеме хранения данных модели состава системы.

На рис. 1 b показаны подграфы, представляющие части модели состава системы для суперкомпьютеров Ломоносов–2 и Fugaku. Соответствие между моделью состава систем (рис. 1 a) и некоторыми вершинами графа (рис. 1 b) показано синими пунктирными стрелками. Ациклический ориентированный граф $G^s =< V^s, E^s >$, где V^s — множество вершин, представляющее системы и подсистемы всех суперкомпьютеров, представленных в CompZoo, а E^s — множество связей, отражающих факт вхождения подсистемы в систему или в подсистему более высокого уровня, является предпочтительной схемой хранения модели состава системы, поскольку в отличие от иерархической структуры позволяет избежать дублирующих описаний подсистем, а при определенном подходе к описанию подсистем может быть сведен к иерархии.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

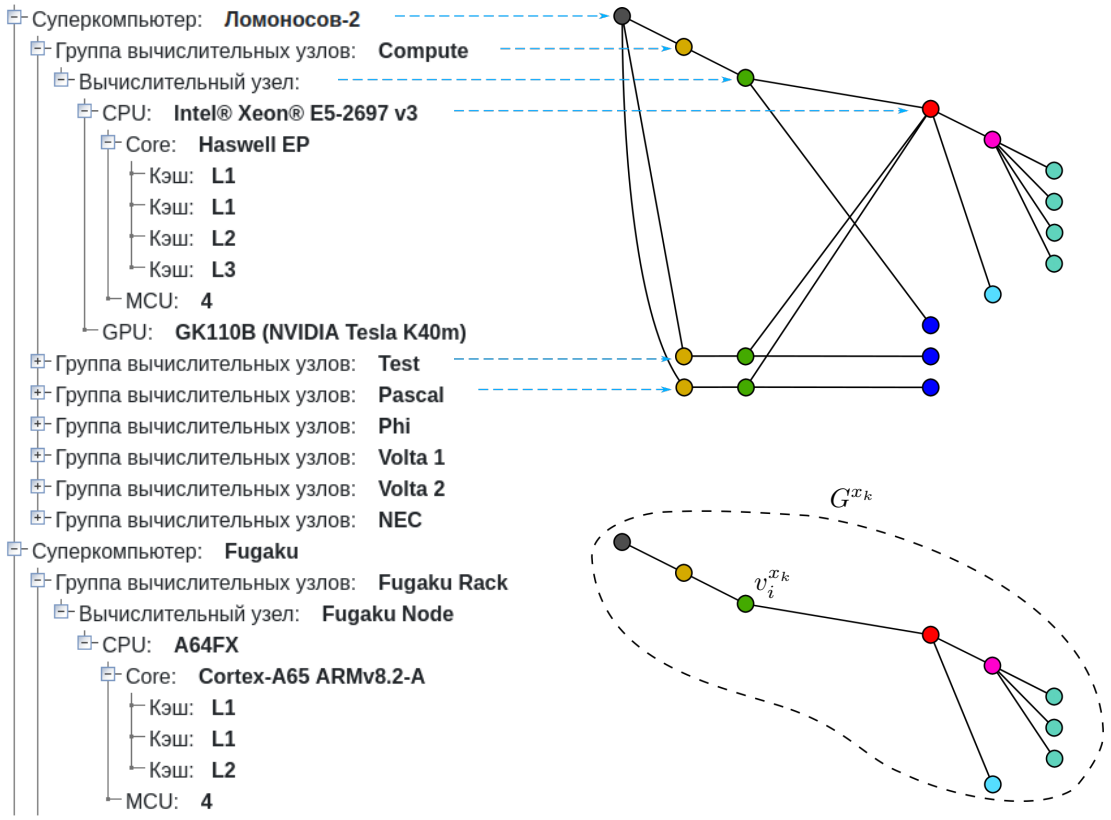

Рис. 1. Представление фрагментов моделей состава систем для суперкомпьютеров Ломоносов–2 и Fugaku: a) в виде списка подсистем в CompZoo; b) в виде графа

Fig. 1. Representation of fragments of system composition models for Lomonosov–2 and Fugaku supercomputers: a) as a list of subsystems in CompZoo; b) in the form of a graph

4.3. Логика описания вершин графа G^s. Рассмотрим, как связаны между собой формальное описание суперкомпьютера (4), (5) и графовое представление G^s модели состава системы. Верхним индексом будем обозначать принадлежность к описанию суперкомпьютера.

Пусть описание суперкомпьютера x_k ($k \in [1, |U_\text{m}|]$) представлено подграфом G^{x_k} графа G^s с множеством подсистем $V^{x_k} = \{v_i^{x_k}\} \subseteq V^s$ (рис. 1 b). Отношения подчиненности между подсистемами $E^{x_k} \subseteq E^s$ будет рассмотрено далее. Для каждого V^{x_k} по аналогии с (4), (5) запишем

$$V^{x_k} = \{v_i^{x_k} : P_i^{x_k}(\mathcal{F}_i^{x_k}(v_i^{x_k}))\}, \quad i = \overline{1, |V^{x_k}|}, \tag{6}$$

$$P_i^{x_k}(v_i^{x_k}) = \bigwedge_{j=1}^{N_i^{x_k}} p_{ij}^{x_k}(f_{ij}^{x_k}(v_i^{x_k}), \alpha_{ij}^{x_k}), \quad N_i^{x_k} = |\mathcal{F}_i^{x_k}|, \tag{7}$$

здесь $v_i^{x_k}$ — i-я подсистема суперкомпьютера x_k, $\mathcal{F}_i^{x_k} =< f_{ij}^{x_k} >$ — кортеж предметных функторов, $P_i^{x_k}$ — предикат, образованный конъюнкцией двухместных предикатов вида $p_{ij}^{x_k}(f_{ij}^{x_k}(v_i^{x_k}), \alpha_{ij}^{x_k})$, в которых каждому функтору $f_{ij}^{x_k}$ соответствует функция, записанная на естественном языке и определенная для данной подсистемы, $\alpha_{ij}^{x_k} \in A_i^{x_k}$ — значения характеристик подсистемы $v_i^{x_k}$, $A_i^{x_k}$ — множество значений всех характеристик подсистемы $v_i^{x_k}$, включая неопределенное значение *undefined*.

В (6), (7) содержится описание суперкомпьютера x_k, разделенное на описания подсистем $v_i^{x_k}$. Это дает возможность сокращенной записи характеристик, например типа “пиковая производительность вычислительного узла”, при этом всегда подразумевается, что вычислительный узел, как и любая другая подсистема, относится к определенному суперкомпьютеру x_k. Сформулированное в разделе 4.1 требование возможности изменения кортежа характеристик $\mathcal{F}$ суперкомпьютера сводится к требованию обеспечить в CompZoo возможность изменения более компактных кортежей $\mathcal{F}_i^{x_k}$ для подсистем.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Представленное в (6), (7) группирование характеристик $\mathcal{F}$ суперкомпьютера по кортежам $\mathcal{F}_i^{x_k}$ для подсистем носит синтаксический характер и мало способствует дальнейшему сравнению суперкомпьютеров по характеристикам подсистем, поскольку для суперкомпьютеров x_k и x_l кортежи $\mathcal{F}_i^{x_k}$ и $\mathcal{F}_i^{x_l}$ могут 1) относиться к физически разным подсистемам; 2) содержать различные характеристики.

Для сохранения семантики, присущей операции выделения подсистем в суперкомпьютере, предложим пользователю назначить каждой подсистеме $v_i^{x_k}$ тег. На рис. 1 a тегами подсистем являются “суперкомпьютер”, “группа вычислительных узлов”, “вычислительный узел” и другие. Отметим, что тег подсистемы и название подсистемы имеют различный смысл: тег обозначает подсистему и определяет ее тип (состав и порядок характеристик в кортеже $\mathcal{F}_i^{x_k}$), а название подсистемы, как было установлено ранее, является значением функтора $f_{i1}^{x_k}$, которому должна соответствовать функция “название” естественного языка. На рис. 1 a названия подсистем выделены полужирным шрифтом.

Введем формально для каждого суперкомпьютера x_k функцию T^{x_k}, определенную на множестве подсистем $V^{x_k} = \{v_i^{x_k}\}$, которая возвращает тег подсистемы. Зафиксируем определенное значение tag для T^{x_k}. Множество характеристик

$$\mathcal{C}_{\text{tag}} = \bigcup_{x_k,i} \mathcal{F}_i^{x_k} : \ \forall x_k \forall i \ T^{x_k}(v_i^{x_k}) = \text{tag}, \quad x_k \in U_{\text{m}}, \ i = \overline{1, |V^{x_k}|}, \tag{8}$$

назовем *категорией* с именем tag. К категории $\mathcal{C}_{\text{tag}}$ относятся все подсистемы из множества

$$V_{\text{tag}} = \bigcup_{x_k,i} v_i^{x_k} : \ \forall x_k \forall i \ \mathcal{F}_i^{x_k} \subset \mathcal{C}_{\text{tag}}, \quad x_k \in U_{\text{m}}, \ i = \overline{1, |V^{x_k}|}. \tag{9}$$

На рис. 1 b вершины графа, относящиеся к одной категории, окрашены в одинаковый цвет.

Рассмотрим, как используется группировка характеристик подсистем по категориям. На рис. 2 показана таблица, содержащая фрагменты описаний суперкомпьютеров Ломоносов-2 и Fugaku в части подсистем Core (ядро CPU). Среди столбцов, входящих в категорию Core (рис. 2), присутствуют характеристики, специфичные только для ядер процессоров Intel: “Технология Turbo Boost” и “Технология Hyper Threading”. Характеристики “Архитектура команд ARM” и “Ширина векторов SIMD” определены только для современных ядер ARM-процессоров.

Компонента CompZoo дает исследователю возможность определить наборы характеристик, например в следующем порядке:

1. Создать фиктивный базовый набор характеристик $\mathcal{F}^0 =< f_1^0, f_2^0, \ldots, f_6^0 >$ с тегом Core, включающий в себя “Обозначение ядра”, “Название фирмы-разработчика ядра”, “Архитектура ядра”, “Число ядер”, “Число потоков”, “Разрядность набора команд”.

Название суперкомпьютера	Core										
	Обозначение ядра	Название фирмы-разработчика ядра	Архитектура ядра	Число ядер	Число потоков	Разрядность набора команд	Технология Turbo Boost	Технология Hyper Threading	Расширения архитектуры	Архитектура команд ARM	Ширина векторов SIMD
1. Ломоносов-2	Cascade Lake	Intel	x86-64	18	36	64	v 2.0	используется	SSE4.2, AVX, AVX2, AVX-512	—	—
	Skylake	Intel	x86-64	18	36	64	v 2.0	используется	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, AVX, AVX2	—	—
2. Fugaku	Cortex-A65 ARMv8.2-A	Arm	RISC	48	48	32	—	—	Neon, Floating Point	A64, A32, T32	512 bit

Рис. 2. Фрагменты описаний суперкомпьютеров Ломоносов-2 и Fugaku в части подсистемы Core

Fig. 2. Fragments of descriptions of Lomonosov-2 and Fugaku supercomputers in terms of the Core subsystem

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

2. Создать набор характеристик $\mathcal{F}^1$ с тегом Core для описания ядер процессоров Intel. Для этого в качестве "родительского" выбирается набор характеристик из предыдущего пункта и расширяется специфическими для этого типа ядер характеристиками "Технология Turbo Boost", "Технология Hyper Threading", "Расширения архитектуры": $\mathcal{F}^1 = \mathcal{F}^0 \cup < f_7^1, f_8^1, f_9^1 >$.

3. Создать набор характеристик $\mathcal{F}^2$ с тегом Core для описания ядер процессоров ARM, основываясь на наборе характеристик, определенном как описано в п.1, и дополнить его характеристиками "Архитектура команд ARM", "Ширина векторов SIMD" и "Расширения архитектуры": $\mathcal{F}^2 = \mathcal{F}^0 \cup < f_{10}^2, f_{11}^2, f_{12}^2 >$.

Построение категории $\mathcal{C}_{\text{Core}}$ выполняется в компоненте CompZoo автоматически.

К категории $\mathcal{C}_{\text{Core}}$ относятся все ядра процессоров, для которых оказывается истинным предикат вида

$$P_{\text{Core}}(x_k) = p_1(f_1^0(x_k), \alpha_1^{x_k}) \wedge p_2(f_2^0(x_k), \alpha_2^{x_k}) \wedge \ldots \wedge p_6(f_6^0(x_k), \alpha_6^{x_k}) \wedge$$
$$\wedge \left[p_7(f_7^1(x_k), \alpha_7^{x_k}) \wedge p_8(f_8^1(x_k), \alpha_8^{x_k}) \wedge p_9(f_9^1(x_k), \alpha_9^{x_k}) \vee\right.$$
$$\left. \vee p_{10}(f_{10}^2(x_k), \alpha_{10}^{x_k}) \wedge p_{11}(f_{11}^2(x_k), \alpha_{11}^{x_k}) \wedge p_{12}(f_{12}^2(x_k), \alpha_{12}^{x_k})\right]. \quad (10)$$

Сопоставляя (8), (9) и (10), можно заметить аналогию с законом традиционной логики понятий: уменьшение содержания понятия ведет к увеличению его объема. Действительно, дизъюнкция предикатов внутри квадратных скобок в (10) (уменьшение содержания понятия за счет альтернатив) приводит к увеличению количества подсистем, относящихся к данной категории (9), что аналогично увеличению объема понятия.

Рассмотрим множество связей $E^{x_k} \subseteq E^s$ между подсистемами суперкомпьютера: $E^{x_k} = \{(v_i^{x_k}, v_j^{x_k}) : v_i^{x_k}, v_j^{x_k} \in V^{x_k},\ i \neq j,\ i,j = \overline{1, |V^{x_k}|}\}$. Из модели состава системы следует, что описания нескольких однотипных подсистем, например, CPU в составе вычислительного узла, заменяются одним описанием с указанием количества экземпляров, которое может быть указано в характеристиках любой из пары вершин $(v_i^{x_k}, v_j^{x_k})$.

4.4. Значения характеристик в CompZoo. Введение значения *undefined* в множество A значений функторов f_i из $\mathcal{F}$ (4) вызвано необходимостью расширения их области определения до U_{m}. Пусть каждой функции f_i соответствует домен значений D_i, тогда предикат P (5) задает некоторое подмножество декартова произведения доменов D_i или N-местное отношение. Графической интерпретацией N-местного отношения является таблица, подобная показанной на рис. 2, с ячейками, заполненными символом "—", для которых значения соответствующих функторов равны *undefined*.

Необходимость введения значения *undefined* возникает в двух случаях: 1) при реляционном представлении данных и в сопутствующих задачах сортировки и фильтрации данных; 2) если значение характеристики суперкомпьютера по каким-либо причинам неизвестно исследователю. Последнее наблюдается достаточно часто: производители устройств не следуют какому-либо унифицированному набору параметров, если устройство не стандартизировано, или по коммерческим причинам дают уникальные названия родственным технологиям и т.п.

Остальные значения, принадлежащие множеству A кроме *undefined*, в CompZoo представлены следующими типами: `TextAttr` предназначен для хранения значений текстовых характеристик, `NumAttr` — для значений числовых характеристик, `UrlAttr` — для значений текстовых характеристик в виде гиперссылок. Значения характеристик типа `NumAttr` могут быть именованными числами, например 10 Tflops, 1.34 MB и т.д. Использование такого ограниченного набора типов позволило организовать простыми средствами ввод значений характеристик и их отображение. Подробное обсуждение программной реализации типов переменных не входит в цели данной статьи.

Отметим, что у базового типа `TextAttr` существуют два расширения: есть возможность определять перечисления строк и множества. Первое расширение `TextAttr` задается через фигурные скобки, например `{да, нет}`, `{CISC, RISC, VLIW}` и др., второе — с помощью квадратных скобок, например множество значений "расширения архитектур" ядер в случае описания процессоров Intel задано как `[MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, AVX, AVX2]`, а для процессоров ARM как `[Neon, Floating Point]` (рис. 2). Для пользователя, осуществляющего ввод, значения, заданные в фигурных скобках, отображаются в виде выпадающего списка, а заданные в квадратных скобках — множеством виджетов типа checkbox.

4.5. Сравнение суперкомпьютеров по характеристикам. Порядок определения набора характеристик, описанный в разделе 4.3, позволяет выделить общие характеристики, присущие подсистемам

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

одной категории. В предикате P_{Core} (10), определяющем множество подсистем, относящихся к категории Core, общими являются характеристики $f_1^0, f_2^0, \ldots, f_6^0$. Множество V_{tag} (9) подсистем, относящихся к одной категории, является универсумом, на котором имеет смысл сравнивать подсистемы. В целом суперкомпьютеры в CompZoo, представленные совокупностями описаний подсистем в рамках модели состава систем, сравнивают по общим характеристикам категорий подсистем.

В вопросе о сравнении суперкомпьютеров по характеристикам снова обращаем внимание на аналогию между описаниями в CompZoo и понятиями: в логике установлено, что сравнимыми являются понятия только с одним и тем же родом [20, 21]. Аналогия между порядком определения набора характеристик подсистем в CompZoo и родо-видовым определением понятий, которое для исследователя является естественным мыслительным процессом, дает основание считать подход к созданию описаний подсистем, представленный в CompZoo, адекватным целям исследователя.

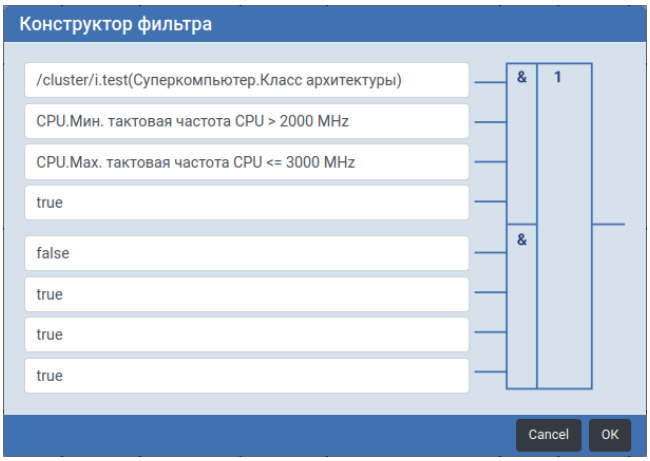

Рис. 3. Окно конструктора фильтра в CompZoo. В первой строке задано регулярное выражение для поиска архитектур типа cluster

Fig. 3. Filter constructor window in CompZoo. The first line contains a regular expression to search for architectures of type cluster

Выше под “сравнением” подразумевался мыслительный процесс, свойственный исследователю. В программе CompZoo вместо такого “сравнения” реализуются технические операции *сортировки* и *фильтрации*. Поскольку в CompZoo множество значений характеристик расширено за счет значения *undefined*, необходимо для него определить отношения равенства и неравенства.

В CompZoo операция фильтрации (рис. 3) имеет приоритет над сортировкой, т.е. сначала происходит выделение некоторого подмножества суперкомпьютеров, а затем сортировка по одной из характеристик. В настоящее время сортировка характеристик типа `TextAttr` и `UrlAttr` осуществляется по значениям, неравным *undefined*, и затем в конец отсортированного списка добавляются описания суперкомпьютеров, попавшие в фильтр, для которых значение характеристики равно *undefined*. Для характеристик типа `NumAttr` исследователю предлагается выбрать, считать *undefined* наименьшим или наибольшим значением. При определении критериев фильтрации достаточно иметь возможности для включения или исключения характеристик со значениями *undefined*.

4.6. Проблемы применения метамодели данных компоненты CompZoo. При создании описаний суперкомпьютеров в компоненте CompZoo исследователь использует наборы характеристик, которые сам определяет, следуя допущениям и ограничениям *метамодели данных* компоненты CompZoo (см. разделы 4.2–4.5).

Допущения метамодели данных CompZoo:

1. Описание суперкомпьютера может производиться в соответствии с моделью состава системы: суперкомпьютер представи́м в виде набора подсистем, каждая из которых в свою очередь может быть представлена своими подсистемами. Глубина декомпозиции определяется целями исследования.

2. Описания подсистем могут создаваться независимо и могут входить в состав описаний различных суперкомпьютеров (графовая модель хранения описаний).

3. Каждая подсистема может иметь уникальный набор характеристик, состав которого определяется исследователем.

4. Набор характеристик подсистемы может быть образован от ранее описанного набора характеристик за счет наследования части уже известных и определения новых уникальных характеристик. Определенные таким способом описания подсистем относятся к одной категории.

5. Описание суперкомпьютера неделимо: фильтрация или сортировка подсистем по характеристикам всегда приводит к переупорядочиванию табличного представления описаний суперкомпьютеров.

Ограничения в компоненте CompZoo, следующие из метамодели данных:

1. Слово “суперкомпьютер” является зарезервированным в качестве тега вершины графа, с кото-

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

рой начинается описание суперкомпьютера. Пользователь не может использовать слова из множества зарезервированных слов (задаются из технических потребностей, в статье не рассматриваются).

2. Первой характеристикой, которую определяет исследователь в описании подсистемы, должна быть характеристика, смысл которой состоит в передаче названия подсистемы. Это относится к подсистемам, которые могут входить в состав спецификации на суперкомпьютер как покупное изделие. Для иных подсистем, таких как кэш-память, название может отсутствовать.

3. Суперкомпьютеры могут сравниваться только по характеристикам подсистем, принадлежащим общей категории.

Проблемы, которые потенциально могут возникнуть при создании описаний суперкомпьютеров в принятой метамодели данных:

1. Проблема описания отношения подчиненности между подсистемами $E^{x_k} \subseteq E^s$ возникает при необходимости учета количества однотипных подсистем. Для описаний суперкомпьютеров распространены количественные характеристики вида “группа содержит 1487 вычислительных узлов”, “вычислительный узел включает в себя 4 CPU и 2 GPU” и т.п. Выбор прагматичного подхода к описанию заключает в себе дилемму, в какой из подсистем указывать количество вычислительных узлов: в характеристиках 1) группы или 2) вычислительного узла. Первый вариант аналогичен конструкторской документации, в которой принято указывать количество подсистем в спецификации подсистемы верхнего уровня. Однако подсистема “группа вычислительных узлов” присутствует не во всех описаниях суперкомпьютеров кластерной архитектуры, например не выделяется разработчиками Frontier [32], и может совершенно отсутствовать в ранних архитектурах суперкомпьютеров. В этом случае исследователю придется либо искусственно вводить в описание суперкомпьютера подсистему “группы вычислительных узлов”, либо смириться с неявно заданным ограничением при сравнении, которое будет выполняться только для суперкомпьютеров с группами вычислительных узлов, либо включать количественную характеристику в такую вышестоящую в иерархии подсистему, которая точно присутствует во всех описаниях, например “суперкомпьютер”. Второй вариант, связанный с добавлением данной количественной характеристики к описанию самого вычислительного узла, фактически приводит к отказу от графовой модели хранения описаний подсистем: невозможно многократно использовать описание типовой подсистемы, если в ее характеристиках указано количество, имеющее значение для определенного суперкомпьютера.

2. Проблема точности описаний суперкомпьютеров возникает в связи с различными названиями подсистем, которые им дают разработчики. Так, одинаковые по сути “разделы” суперкомпьютера Ломоносов-2 [30] и “primary compute system” у Frontera [33] целесообразно свести в категорию с общим названием, что может вызвать несогласие у некоторых исследователей. Другим примером, иллюстрирующим эту же проблему, является приведение к одинаковому виду характеристик однотипных подсистем: например, задержки памяти могут описываться в тактах или в наносекундах.

3. Проблема наименования значений информационных величин вызвана тем, что:

1) принятые в области HPC обозначения единиц измерения flops, teps и др. не стандартизированы и могут отличаться в различных источниках по написанию;
2) в последние годы в зарубежных источниках, следуя рекомендациям международных стандартов МЭК (ИЭК), все чаще используют с информационными величинами приставки “киби-”, “меби-” и т.д., обозначающие кратность целой степени числа 2, вместе с тем по законодательству Российской Федерации [34] применение аутентичного перевода стандарта МЭК [35] является необязательным, и в отечественной литературе в лучшем случае наблюдается соответствие обозначений информационных величин рекомендациям из ГОСТ 8.417–2002 [36].

Вероятно, для системы CompZoo потребуется дополнительное соглашение о смысловом значении обозначений конкретных величин.

5. Схема данных компоненты PerfData.

5.1. Суперкомпьютерный эксперимент как источник данных для рейтинговых списков. Построение различных рейтинговых списков для алгоритмов, представленных в AlgoWiki, подразумевает наличие обширного массива значений разнообразных метрик, полученных в результате прогонов различных реализаций алгоритмов на доступных исследователю суперкомпьютерах с разными объемами входных данных и прочими параметрами, контролируемыми исследователем. Многообразие используемых алгоритмов и их реализаций отличает рейтинги в Algo500 от рейтингов Top500, Graph500, HPCG, Green500 и других, в которых алгоритмы заданы жестко, а условия их выполнения могут варьировать-

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

ся менее гибко. Анализ практических аспектов работы с метриками (структурирование данных при их внесении в Algo500, соблюдение определенной последовательности действий при прогонах реализаций алгоритмов и обработке полученных первичных данных различными группами исследователей из разных стран, поиск и выбор результатов, которые действительно имеет смысл сравнивать и т.д.) показывает, что задача создания базы данных метрик является первичной по отношению к получению рейтинговых списков. Без решения проблемы структурирования данных о прогонах реализаций алгоритмов обсуждение проблемы выбора критериев для построения рейтингов не имеет смысла.

Для систематизации процессов, связанных с накоплением данных, являющихся основой для построения рейтинговых списков, предлагаем использовать структуру, называемую *суперкомпьютерный эксперимент*. Такое название структуры будем считать рабочим, так как в настоящее время в науке нет общепризнанного определения понятия "суперкомпьютерный эксперимент". Ситуация, в которой структура имеет название, а понятие еще нуждается в точном определении, выглядит парадоксально лишь на первый взгляд: построить модель сложного понятия вполне возможно в ущерб полноте описания (таким образом мы поступали с описанием суперкомпьютеров в разделе 4).

С одной стороны, в задачи данной работы не входит введение и обоснование новых понятий, с другой — появилась потребность в термине для обозначения структуры данных в проекте Algo500, поэтому приведем некоторые аргументы в пользу введения в область высокопроизводительных вычислений понятия суперкомпьютерный эксперимент:

1. Суперкомпьютерный эксперимент проводится над специфическими объектами — суперкомпьютерами и вычислительными процессами, которые происходят в суперкомпьютере, — что выделяет его среди других видов экспериментов.

2. Цели суперкомпьютерного эксперимента кардинально отличаются от целей вычислительного, симуляционного и других видов компьютерных экспериментов тем, что в специально созданных условиях (на определенной конфигурации суперкомпьютера и программного обеспечения) изучается динамика изменения различных метрик вычислительного процесса, при этом сам результат вычислительного процесса особой ценности не представляет. Изменение условий проведения суперкомпьютерного эксперимента, например количества вычислительных узлов, выделенных для задачи, при неизменных входных данных программы позволит получить новые метрики вычислительного процесса, но не изменит результат вычислений.

3. Числовые значения таких характеристик суперкомпьютера и вычисляемой им задачи, как производительность, энергоэффективность, средняя нагрузка CPU, количество ошибок при обращении к кэш-памяти и т.п., в настоящее время не могут быть получены никак иначе, как в результате проведения спланированного суперкомпьютерного эксперимента. Суперкомпьютер является технически сложной аппаратно-программной системой, которая обладает и детерминированными, и стохастическими признаками поведения, что делает затруднительным построение адекватной математической модели и оправданным экспериментальное изучение.

4. Термин "суперкомпьютерный эксперимент" в отличие от "тест" подчеркивает выбранный метод изучения сложной технической системы и наводит на мысль о возможности выделения в суперкомпьютерном эксперименте по аналогии с реальным экспериментом таких стадий проведения, как планирование, описание условий и методики выполнения измерений, обработки и анализа результатов и т.д.

Несмотря на приведенные доводы, корректное определение понятия суперкомпьютерного эксперимента требует глубокого изучения этой области знаний, в частности в связи с причислением его к классу измерительных экспериментов. Исследователю необходимо ясное понимание процесса измерения в суперкомпьютерном эксперименте для проведения дальнейшего сравнительного анализа результатов экспериментов.

Поскольку для платформы Algo500 мы пытаемся максимально использовать общепринятую терминологию, будем применять для описания измерительного суперкомпьютерного эксперимента основные метрологические понятия, относящиеся к процессу измерений. Определения для ряда понятий и термины взяты из аутентичного перевода Международного словаря по метрологии (VIM) [37], при составлении которого были аккумулированы знания из различных научных дисциплин; соответствующие определения можно найти в нормативных документах Государственной системы обеспечения единства измерений [38]:

1.1 Величина — свойство явления, тела или вещества, которое может быть выражено количественно в виде числа с указанием отличительного признака как основы для сравнения.

...

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Примечание 2. Указание основы для сравнения может касаться единицы измерения, методики измерения, стандартного образца или их комбинации.

2.1 Измерение — процесс экспериментального получения одного или более значений величины, которые могут быть обоснованно приписаны величине.

. . .

Примечание 2. Измерение подразумевает сравнение величин и включает счет объектов.

1.2 Род величины — общий аспект для взаимного сопоставления величин.

Примечание 1. Разделение величин по их родам является, в некоторой степени, произвольным.

Примечание 2. Однородные величины в рамках данной системы величин имеют одинаковую размерность величины. Однако величины одинаковой размерности не обязательно будут однородными.

1.9 Единица измерения — действительная скалярная величина, определенная и принятая по соглашению, с которой можно сравнить любую другую величину такого же рода и и выразить их отношение в виде числа.

. . .

Примечание 2. Единицы измерения величин одинаковой размерности могут иметь одинаковые наименования и обозначения, даже когда величины не являются однородными.

1.27 Шкала измерений — упорядоченный набор значений величин данного рода, используемый для ранжирования в соответствии с размером величин этого рода.

2.6 Методика измерений — детальное описание измерения в соответствии с одним или более принципами измерений и данным методом измерений, которое основано на модели измерений и включает вычисления, необходимые для получения результата измерения.

Воспользуемся приведенными выше метрологическими понятиями для описания *измерения* такой *величины,* как реальная максимальная производительность суперкомпьютеров $R_{\max}$. Специфика измерения производительности суперкомпьютеров $R_{\max}$ подробно изложена, например в [39, С. 162–178]. В качестве *единицы измерения* производительности принято использовать 1 flops (1 flops равен 1 операции с плавающей точкой, выполненной вычислительным устройством за 1 с). Измерение производительности суперкомпьютера $R_{\max}$ проводится в его полной конфигурации с использованием всех вычислительных ядер на определенном алгоритме и объеме данных, как это указано, например, в *методике измерений*, описанной в тесте Linpack. В большинстве случаев значение $R_{\max}$ получают методом косвенных измерений по известному числу операций с плавающей точкой N в алгоритме теста Linpack и измеренному интервалу времени t, в течение которого выполняется реализация теста:

$$R_{\max} = N/t, \quad [R_{\max}] = 1 \text{ flops}.$$

В значения производительностей, полученных по методике теста Linpack, специалистами вкладывается однозначный смысл: они показывают только то, насколько хорошо суперкомпьютеры могут решать системы уравнений с плотной матрицей указанным методом [39].

Покажем, пользуясь определениями метрологических понятий, как связаны измеренные значения производительности $R_{\max}$ и методика измерений и какие выводы из этого следуют.

Прежде всего отметим, что наличие единицы измерения для $R_{\max}$ часто трактуется и как существование единой *шкалы измерений* производительности, по которой сравнивают не только значения, полученные по одной и той же методике, но и значения $R_{\max}$, полученные в различных методиках измерений: для различных реализаций алгоритмов, для устройств, принадлежащих к разным классам архитектур, таким как CPU и GPU, и т.п. Пользуясь приведенными выше определениями, можно сказать, что при таком подходе основание для сравнения — *методику измерения* — заменили на *единицу измерения*, которая во многих случаях используется для теоретической (асимптотической) оценки производительности вычислительных систем и к тому же в настоящее время не имеет утвержденной собственной методики измерения, единой для всех типов вычислительных устройств.

Связь между методикой измерений и значениями величины $R_{\max}$, полученными экспериментально, станет логически обоснованной, если принять, что при выполнении экспериментов с *различными методиками измерения* производятся измерения значений *разнородных величин*, т.е. причисление измеряемой величины к тому или иному *роду величин* производится на основании *методики измерения*, по которой проводилось измерение. Тогда одинаковая размерность единиц измерения может не приниматься во

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

внимание, так как метрология допускает существование разнородных величин с единицами измерений одинаковой размерности.

Отказ от общей единицы измерений при сравнении производительностей в пользу сравнения на основании методик измерения неизбежно приводит к введению *шкал измерений*, общих для экспериментов с одной методикой измерений. Анализ свойств шкал измерений, таких как выполнение отношений эквивалентности и порядка, существование единицы измерения, нуля и т.п. [40], выходит за рамки статьи. Отметим, что отображение значений измеряемой величины на числовую шкалу является достаточным основанием для существования у шкалы измерений величины свойства порядка и, следовательно, второй подход к пониманию измерений в суперкомпьютерных экспериментах, основанный на методике измерений, не противоречит принятой методологии сравнения суперкомпьютеров посредством рейтинговых списков.

Таким образом, представлены две трактовки результатов измерения величины $R_{\max}$ в суперкомпьютерном эксперименте: 1) в качестве основания для сравнения значений величины принимается введенная единица измерения 1 flops; 2) основанием для сравнения значений величин является методика измерений. Полагаем, что измерения и некоторых других величин, характеризующих вычислительные процессы на суперкомпьютере, могут трактоваться аналогично. Выбор исследователем того или иного подхода означает и выбор логического основания для сравнения, и выбор технической реализации процедуры сравнения на платформе Algo500.

5.2. Типизация суперкомпьютерных экспериментов. Построение рейтинговых списков тесно связано с выбором основания для сравнения, в качестве которого в предыдущем разделе рассматривались единица измерения величины или методика измерения. При любом подходе методика измерения должна включаться в описание суперкомпьютерного эксперимента как часть условий проведения эксперимента. В условия суперкомпьютерного эксперимента также могут входить: 1) программная реализация алгоритма; 2) программа-генератор или иной источник данных; 3) указание типа вычислительного устройства или класса архитектуры суперкомпьютера, на которых проводится эксперимент, и другие факторы, оказывающие решающее влияние на выполнение эксперимента.

При выполнении эксперимента в одних и тех же условиях можно получить значения нескольких метрик вычислительного процесса, поэтому наряду с описанием условий эксперимента для компоненты PerfData важно описать и в каком формате представлен результат эксперимента. В задачи эксперимента, кроме получения значений величин для рейтинговых списков, может входить экспериментальное изучение зависимости какой-либо величины от набора изменяющихся параметров. Совокупность описаний условий проведения и результатов суперкомпьютерных экспериментов в платформе Algo500 определяет *тип суперкомпьютерного эксперимента.*

С учетом нового понятия *рейтинговый список в Algo500 — это упорядоченный список, построенный по результатам различных суперкомпьютерных экспериментов, относящихся к одному типу.*

5.3. Формализованное описание суперкомпьютерного эксперимента в Algo500. Решение проблемы описания суперкомпьютерных экспериментов начиналось с поиска подходящих прототипов в области технологий обработки и хранения многомерных данных (OLAP), систем обработки и управления лабораторной информацией (LIMS, ЛИУС, ЛИС), систем хранения и передачи результатов физических экспериментов и др. В статье [41] представлен онтологический анализ предметной области проекта хранилища обобщенных вычислительных экспериментов (ОВЭ), являющихся основой для математического моделирования реальных физических процессов. Применительно к проблеме описания суперкомпьютерного эксперимента из этой статьи полезными являются формализованное представление ОВЭ, структура и требования к хранилищу ОВЭ. Из списка технологий заслуживает внимания иерархический формат данных (Heirarchical Data Format, HDF), предложенный в Национальном центре суперкомпьютерных приложений (университет штата Иллинойс, США). В настоящее время широко распространен формат версии HDF5 [42]. Центральной структурой данных форматов HDF является ориентированный граф, в вершинах которого могут находиться описания групп данных, наборов данных в виде многомерных массивов, именованных типов данных. Для формата HDF5 разработаны библиотеки на Python, Java, JavaScript и других языках программирования.

В обоих проектах особый интерес вызывают структуры, предназначенные для описания условий и результатов экспериментов, однако они непосредственно не могут быть использованы в проекте Algo500, так как не позволяют описать суперкомпьютерный эксперимент. С технической точки зрения специфика описания суперкомпьютерного эксперимента заключается в том, что весь объем информации об экспе-

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

рименте должен быть разделен на части, отличающиеся по представлению данных, и распределен по различным хранилищам. Наиболее часто использующаяся информация для построения рейтингов, поиска экспериментов и т.п. должна храниться в SQL базе данных, часть описаний — в виде Wiki-страниц, программный код — в Git-репозитории [7].

На рис. 4 суперкомпьютерный эксперимент представлен в виде модели черного ящика, где $\boldsymbol{u}$ — вектор параметров, которые варьируются при прогонах реализации алгоритма, $\boldsymbol{y}$ — вектор результатов суперкомпьютерного эксперимента, $\boldsymbol{p}$ — контролируемые неизменные условия эксперимента, $\boldsymbol{q}$ — состояние суперкомпьютера перед прогоном программы, $\boldsymbol{r}$ — неконтролируемые и случайные факторы, сопровождающие выполнение реализаций алгоритмов на суперкомпьютере. Описать состояние суперкомпьютера $\boldsymbol{q}$, как одно из множества состояний конечного автомата, практически невозможно. Поскольку операционная система "стабилизирует" работу суперкомпьютера, обеспечивая высвобождение неиспользуемых ресурсов и возврат в состояние, близкое к исходному, после завершения какой-либо программы, влияние $\boldsymbol{q}$ на результат прогона реализации алгоритма отнесем к случайным факторам $\boldsymbol{r}$ и исключим $\boldsymbol{q}$ из рассмотрения. Если эксперимент проводился с целью изучения влияния случайных факторов $\boldsymbol{r}$ на прогоны реализаций алгоритмов, то полученные в результате прогонов статистические характеристики для $\boldsymbol{r}$ должны быть отнесены к $\boldsymbol{y}$. Если при проведении эксперимента имеют значение статистические характеристики случайных факторов $\boldsymbol{r}$, то эти характеристики явным образом должны описываться в условиях $\boldsymbol{p}$. Таким образом, важнейшими элементами в описании суперкомпьютерного эксперимента являются его условия $\boldsymbol{p}$, $\boldsymbol{u}$ и результат $\boldsymbol{y}$. Кроме них модель черного ящика неявно включает в себя целевое назначение *Goal* и название *Name* эксперимента. Кортеж $< Name, Goal, \boldsymbol{p}, \boldsymbol{u}, \boldsymbol{y} >$ в первом приближении содержит описание суперкомпьютерного эксперимента.

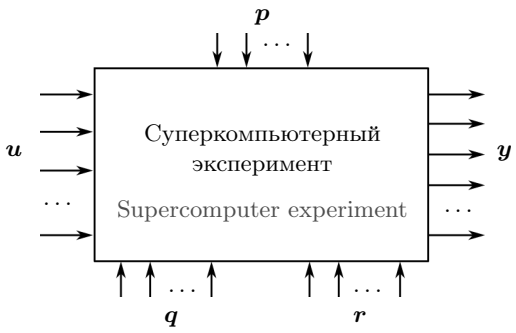

Рис. 4. Модель черного ящика для суперкомпьютерного эксперимента

Fig. 4. Black box model for supercomputer experiment

На рис. 5 упрощенно показана схема данных, предназначенная для хранения описаний суперкомпьютерных экспериментов в Algo500. В состав PerfData входит SQL сервер, в котором записаны данные об эксперименте, представляющие оперативный интерес, и Git сервер для хранения программ, составляющих часть условий суперкомпьютерного эксперимента. Компоненты AlgoWiki и CompZoo используются для хранения других специфических частей описания суперкомпьютерного эксперимента.

Для описания типа суперкомпьютерного эксперимента предназначена таблица `TypeOfExp` (рис. 5), поля которой имеют следующие назначения и свойства:

1. `Name` — наименование типа суперкомпьютерных экспериментов, позволяющее исследователю идентифицировать рейтинговый список. По значению поля `Name` формируется ассоциативная связь с Wiki-страницей, содержащей описание типа суперкомпьютерного эксперимента. На Wiki-странице содержится развернутое текстовое описание цели эксперимента, условий и ожидаемых результатов, методики выполнения эксперимента, методики измерений и т.п.

2. `Goal` — краткое описание назначения суперкомпьютерных экспериментов, относящихся к данному типу. Описание цели, содержащееся в этом поле, дополняет и поясняет название типа суперкомпьютерного эксперимента.

3. `ReferenceToImplementation` — идентификатор реализации алгоритма, к которой относятся эксперименты данного типа. Идентификатором служит маршрут (путь) от задачи верхнего уровня до реализации в графе G^a (1)–(3), представляющем классификацию алгоритмов в AlgoWiki. В графовой модели классификации алгоритмы и их реализации могут относиться к разным методам и задачам. Применение для идентификации реализаций алгоритмов маршрута в графе связывает тип эксперимента с теми методами и задачами, к которым относится реализация алгоритма. Связь между таблицей `TypeOfExp` и реализациями алгоритмов ограничивает исследователя в том, что не допускает создания типов экспериментов, в которых реализации алгоритмов выступали бы в качестве одного из изменяющихся параметров условий эксперимента.

4. `ConditionsFormat` — объект, содержащий информацию о существенных для данного типа экспериментов постоянных условиях $\boldsymbol{p}$ и изменяющихся условиях $\boldsymbol{u}$. Объект включает в себя наименования

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

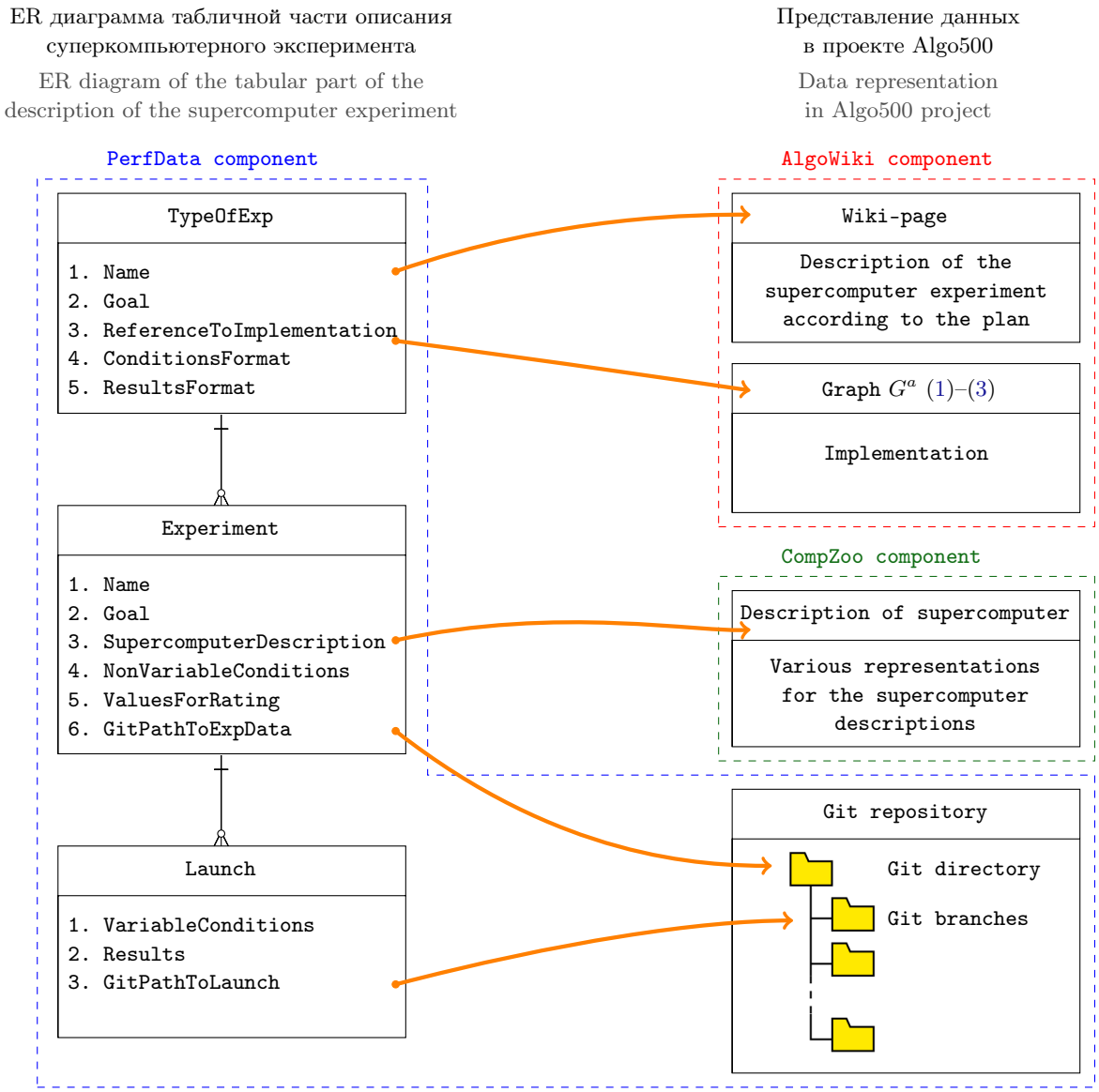

Рис. 5. Информационные связи между различными частями описания суперкомпьютерного эксперимента и их представление в проекте Algo500

Fig. 5. Information links between different parts of the description of the supercomputer experiment and their representation in the Algo500 project

величин, единицы их измерения и информацию о необходимом количестве запусков: например, пусть u_1 представляет количество ядер CPU, выделенных для прогона реализации алгоритма, а u_2 — размер входного массива данных. Параметрам u_1 и u_2 соответствуют домены значений U_1 и U_2, которые могут задаваться или для всего типа экспериментов, или для эксперимента, относящегося к данному типу. По декартовому произведению $U_1 \times U_2$ определяются все возможные пары (u_1, u_2).

5. `ResultsFormat` — объект, содержащий информацию о результатах, которые должны быть получены в эксперименте данного типа. До этого момента считалось, что рейтинговый список получается по результатам суперкомпьютерных экспериментов, однако суперкомпьютерный эксперимент может иметь целью изучение каких-либо зависимостей $(y_1, y_2, \ldots, y_k) = (F_1(\boldsymbol{u}), F_2(\boldsymbol{u}), \ldots, F_k(\boldsymbol{u}))$. В принципе, один и тот же суперкомпьютерный эксперимент может преследовать цель экспериментального изучения зависимостей и быть источником данных $(y_1^{\mathrm{R}}, y_2^{\mathrm{R}}, \ldots, y_l^{\mathrm{R}})$ для рейтинговых списков. Например, если изучается зависимость производительности y_1 от (u_1, u_2) на области входных условий $U_1 \times U_2$, то y_1^{R} может со-

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

держать значение максимальной производительности. Описание способов нахождения значений величин, далее представимых в виде рейтинга, должно содержаться на Wiki-странице, ассоциированной с типом суперкомпьютерного эксперимента.

Рассмотрим кратко назначение наиболее важных полей таблицы `Experiment` (рис. 5), предназначенной для хранения оперативной информации из описания суперкомпьютерного эксперимента. Поле `SupercomputerDescription` содержит ссылку на описание суперкомпьютера в CompZoo, `NonVariableConditions` — значения параметров, входящих в $\boldsymbol{p}$ соответственно с порядком, установленным в поле `ConditionsFormat`, `ValuesForRating` — значения величин, которые согласно информации из `ResultsFormat` могут использоваться для построения рейтингов. В поле `GitPathToExpData` содержится ссылка на Git-репозиторий с программным обеспечением, которое также является частью условий суперкомпьютерного эксперимента. Между таблицами `TypeOfExp` и `Experiment` существует связь типа 1:M.

Каждому эксперименту из таблицы `Experiment` соответствует множество прогонов реализаций алгоритма, описываемых в `Launch` (связь типа 1:M, рис. 5). Поле `Launch.VariableConditions` содержит значения параметров, входящих в $\boldsymbol{u}$ соответственно установленному в поле `TypeOfExp.ConditionsFormat` порядку, поле `Launch.Results` — значения $(y_1, y_2, \ldots, y_k)$, `Launch.GitPathToLaunch` — ссылку на ветку в Git-директории.

5.4. Программа как часть условий суперкомпьютерного эксперимента. К специфике суперкомпьютерного эксперимента относится то, что в состав его условий должны входить программные компоненты: программная реализация алгоритма, программы, генерирующие различные начальные данные, скрипты, предназначенные для компиляции программ и сборки исполняемых файлов, необходимые сторонние библиотеки программ и т.д. Объем программных компонент, в частности данных для графовых алгоритмов, может быть от десятка гигабайт до десятков петабайт для одного суперкомпьютерного эксперимента.

Основными целями всестороннего описания условий суперкомпьютерного эксперимента, включающих в себя программные компоненты, являются обеспечение возможности обоснования выводов теоретических исследований и создание удобного для практического использования репозитория программных реализаций алгоритмов, описание которых приводится в AlgoWiki.

Схема хранения в PerfData программных компонентов как части условий суперкомпьютерного эксперимента представлена на рис. 5. В Git-репозитории для каждого суперкомпьютерного эксперимента создается директория, которая содержит одну или несколько Git-веток для каждого запуска реализации алгоритма с параметрами из домена значений $\boldsymbol{u}$.

5.5. Проблемы, связанные с сохранением данных в PerfData. Разработка и тестирование нескольких прототипов компоненты PerfData позволили сформулировать следующие проблемы:

1. Проблема определения типа суперкомпьютерного эксперимента и описания условий вызвана тем, что ранее исследователям не предлагалось описывать прогоны реализаций алгоритмов в соответствии с такой схемой. Несмотря на то что мы наблюдаем по публикациям, посвященным высокопроизводительным вычислениям в различных областях науки, как авторы более или менее подробно описывают условия прогонов, примеров структурированного описания условий суперкомпьютерных экспериментов немного. Образцами по-прежнему остаются методики рейтингов Top500, Green500 и Graph500. Продумывание и четкое описание типа суперкомпьютерного эксперимента является сложной задачей для исследователя.

2. Проблема представления плана эксперимента через описание домена его входных данных состоит в том, что в настоящее время проведение эксперимента предполагается только по схеме перебора значений входных параметров из множества, образованного декартовым произведением доменов значений каждого параметра. С учетом организационных трудностей и финансовых затрат, неизбежно сопровождающих проведение масштабного суперкомпьютерного эксперимента, вполне можно предположить, что распространение получат многофакторные эксперименты, проводимые по дробным факторным планам, с целью проверки статистических гипотез [43, 44]. Проблема описания планов выполнения таких экспериментов остается актуальной.

3. Проблема описания частной конфигурации суперкомпьютера связана с тем, что 1) в большинстве случаев эксперимент выполняется не в полной конфигурации суперкомпьютера, а на части его вычислительных ресурсов; 2) для суперкомпьютерного эксперимента может иметь значение указание конкретных групп вычислительных узлов, а не только их числа. Предполагается, что полная информация о конфигурации суперкомпьютера содержится в файлах Git-репозитория, однако по ним сложно получить параметры конфигурации и организовать выборку суперкомпьютерных экспериментов.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

4. Проблема дублирования значительных объемов данных в Git вызвана тем, что предлагаемая в настоящее время структура хранения данных в Git-репозитории (рис. 5) не является подобной структуре *тип эксперимента — эксперимент — прогоны реализаций алгоритмов*. В Git-репозитории не представлена область хранения данных для типа эксперимента, поэтому входные данные больших объемов, которые фактически относятся ко всем экспериментам данного типа, будут дублироваться в Git-директориях, предназначенных для каждого эксперимента.

6. Построение рейтинговых списков в компоненте RatingLists. В этой статье функционал платформы Algo500, связанный с формированием рейтинговых списков, т.е. конструирование пользователем запросов к базам данных и представление в браузере результатов запросов, предлагается выделить в отдельную компоненту RatingLists. В отличие от остальных компонент платформы Algo500, в RatingLists пользователь не изменяет никаких данных об алгоритмах, суперкомпьютерах или суперкомпьютерных экспериментах. Конструирование запросов и анализ результатов может выполнять любой исследователь, в том числе незарегистрированный пользователь `guest`.

6.1. Информационные связи между компонентами Algo500 и конструктор запросов в RatingLists. На рис. 6 показаны информационные связи между AlgoWiki, PerfData, CompZoo. Для формирования рейтингов в RatingLists используются данные, созданные в перечисленных компонентах, поэтому на рис. 6 компонента RatingLists не показана. Графовая структура данных G^a (1)–(3) классификации алгоритмов представлена в ER-нотации. С каждой реализацией алгоритма может быть выполнено несколько экспериментов различных типов, что показывают связи `link 1` и `link 2` (рис. 6). К эксперименту относится серия прогонов реализаций алгоритмов (`link 4`), в которой содержатся результаты, представляющие зависимости измеряемых величин от изменяющихся входных параметров. Эксперимент может выполняться на определенном суперкомпьютере, при этом на суперкомпьютере могут проводиться различные эксперименты (`link 3`). Таблица описаний суперкомпьютеров `Description of supercomputer` является виртуальной — она формируется на основе графовой модели описаний суперкомпьютеров (раздел 4.3).

Экспериментальные данные, используемые в RatingLists для построения рейтинга и изучения зависимостей между величинами, содержатся в полях таблиц `Experiment` и `Launch` соответственно (раздел 5.3). Доступ к `Launch` происходит через таблицу `Experiment` (связь `link4`), поэтому подробно будем рассматривать только выборки из `Experiment`. Процедура сортировки рейтинга является стандартной и не нуждается в описании.

Формирование рейтинга сводится к фильтрации данных в таблице `Experiment`. Введем обозначения: T — множество типов экспериментов, содержащихся в таблице `TypeOfExp`; E — множество экспериментов из таблицы `Experiment`; e — элемент множества E; $RL(e, m)$ — функция, возвращающая для метрики m результат r, полученный в эксперименте e; P — предикат, выделяющий некоторое подмножество из E. Рейтинговый список R задается формулой:

$$R = \{r : \ r = RL(e, m), \ e \in E \wedge P(e)\}. \tag{11}$$

Предикат P определяется пользователем в конструкторе запросов. Перемещаясь по вершинам графа G^a (1)–(3), исследователь выбирает подграф $G^a_R \subseteq G^a$, в котором определяет подмножество реализаций $I_R \subseteq I$. Конструктор запросов позволяет исследователю выбрать подмножество типов экспериментов $T_I \subseteq T$, которые могут выполняться с реализациями I_R (`link1`, рис. 6). Как было установлено в разделе 5.1, исследователю может понадобиться рейтинговый список, построенный по результатам экспериментов, которые объединяет не принадлежность к определенному типу экспериментов, а измерение одинаковой метрики. В конструкторе запросов должна быть возможность выделить несколько типов экспериментов, результаты которых будут объединяться в один рейтинг $T_R \subseteq T_I$, при этом во всех $T_i \in T_R$ должна измеряться одна и та же метрика m. Выделенный по типам экспериментов T_R набор экспериментов E_{T_R} может быть дополнительно ограничен (`link3`, рис. 6) подмножеством суперкомпьютеров S_R, отфильтрованным из множества всех описаний суперкомпьютеров S по совокупности характеристик (раздел 4.4). Конструктор запросов должен обеспечивать возможность выбора не только I_R, T_R, S_R, но и логических операций в предикате $P = P(e, I_R, T_R, S_R)$.

Подобным образом по предикату P, определяемому пользователем, через связь `link4` (рис. 6) происходит выборка из таблицы `Launch` значений зависимостей величин от входных параметров эксперимента $(y_1, y_2, \ldots, y_k) = (F_1(\boldsymbol{u}), F_2(\boldsymbol{u}), \ldots, F_k(\boldsymbol{u}))$. Дальнейшее развитие AlgoWiki предусматривает внедрение в Wiki-страницы динамического контента в виде таблиц и графиков, построенных на основе эксперимен-

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

тальных данных $\boldsymbol{y}_i$ о зависимостях $y_i = F_i(\boldsymbol{u})$. Однако исследователю могут понадобиться значения $y_{i,j}$ измеряемой величины y_i, соответствующие определенным значениям $\boldsymbol{u}_j$ параметров эксперимента. Например, если в эксперименте проводилось изучение зависимости производительности от объема входного массива данных на сетке 1 ГБ, 10 ГБ, 100 ГБ, 1000 ГБ, то конструктор запросов должен обеспечивать возможность построения рейтинга по производительности для любого из перечисленных значений объема входных данных.

Наиболее часто используемые исследователями запросы могут быть предварительно определены в Algo500, для них должны использоваться более простые формы параметризации, чем конструктор запросов.

6.2. Роль запросов в RatingLists. В рамках платформы Algo500 исследование завершается анализом данных с помощью инструментов компоненты RatingLists. Получение исследователем информации для сравнительного анализа тесно связано с экспериментальной базой, представленной в компоненте PerfData, и с параметризацией всех объектов, использующихся в (11), поэтому планирование будущих запросов на ранних этапах исследования может существенно конкретизировать его цели.

Допустим, цель исследователя состоит в изучении производительности суперкомпьютеров, которая получается за счет использования различных методов решения определенной задачи, представленной в AlgoWiki. Будем считать, что описания алгоритмов и реализаций уже существуют в AlgoWiki. В таком случае исследователь должен пройти следующие этапы: 1) выделить множество реализаций алгоритмов I_1, относящихся к методам решения данной задачи ($I_1 \subset I$); 2) для каждой реализации алгоритмов из I_1 создать и описать тип эксперимента; 3) создать описания суперкомпьютеров, на которых планируется проводить эксперимент, если они отсутствуют в CompZoo; 4) выполнить серию экспериментов каждого типа и занести их в PerfData; 5) построить рейтинговый список и провести анализ результатов. Трудоемкость этого процесса может быть оправдана при его тщательном планировании: если в экспериментах наряду с производительностью будут измеряться и другие величины, характеризующие вычислительный процесс в данной задаче, то ценность выполненного исследования, очевидно, увеличится.

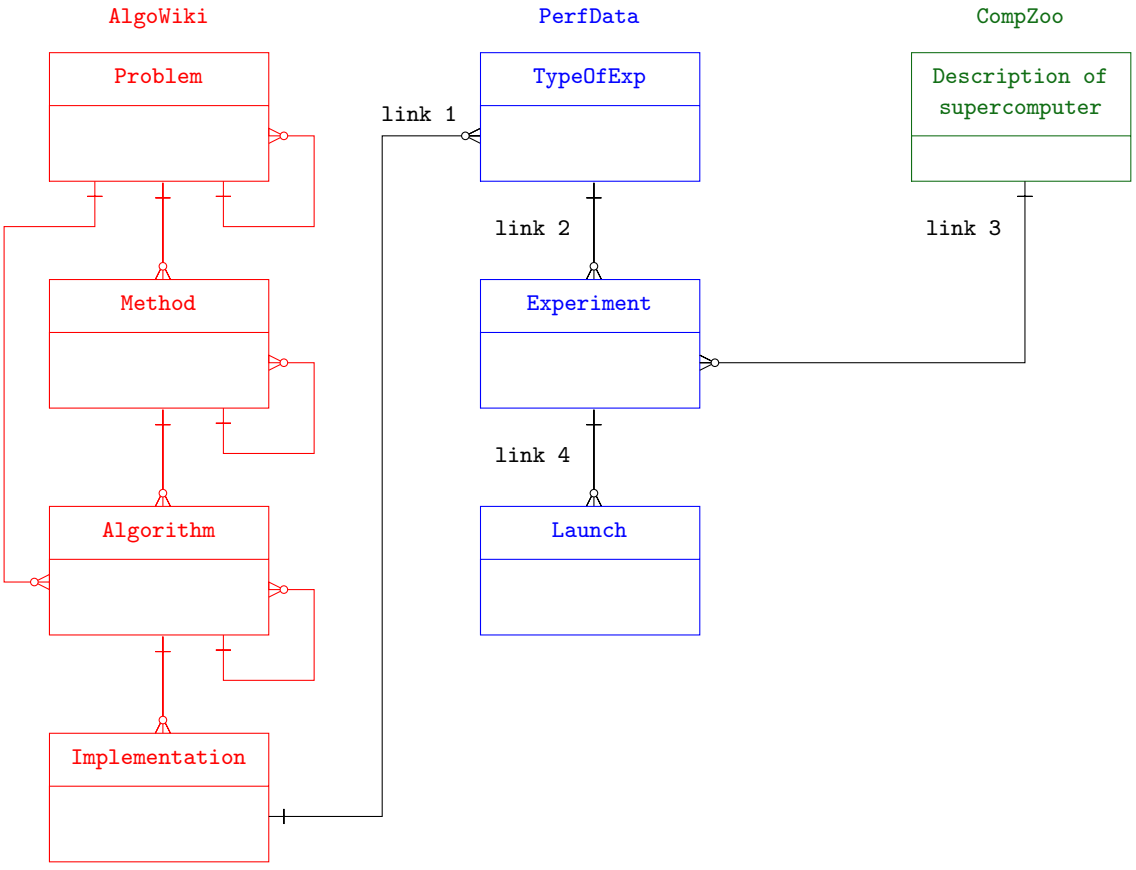

Рис. 6. Информационные связи между компонентами Algo500

Fig. 6. Information links between Algo500 components

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

Приведем в качестве примера запроса для рейтинга, который невозможно получить в рамках обсуждаемых в данной статье моделей данных, запрос о производительности алгоритмов, предназначенных для решения этой же задачи, имеющих последовательную сложность не выше $O(n^2)$ и параллельную сложность не выше $O(n)$. Такой рейтинг в настоящее время невозможно получить, потому что параллельная и последовательная сложности присутствуют только в описаниях алгоритмов на Wiki-страницах, и поэтому они не могут использоваться в качестве параметров в предикате P (11). Для того чтобы можно было построить рейтинг с ограничениями по параллельной и последовательной сложности алгоритмов, нужно ввести их в систему в качестве параметров, принадлежащих, например, вершинам графа G^a (1), моделирующего классификацию алгоритмов.

Приведенный пример рейтинга, не реализуемого в Algo500, показывает необходимость пересмотра модели данных AlgoWiki с целью инкапсуляции в вершины графа G^a формализованных теоретических характеристик из описаний задач, методов, алгоритмов и реализаций. Развитие проекта Algo500 должно опираться на глубокий анализ потребностей исследователя, формально выраженных через запросы в RatingLists, и предусматривать реализацию системы статических параметров или механизмов для их создания в процессе работы с Algo500.

7. Заключение. В статье проведен онтологический анализ предметной области цифровой платформы Algo500, учитывающий потребности пользователя при осуществлении исследовательской деятельности; в дополнение к ранее определенным понятиям введены новые понятия и проанализированы связи между ними: предложена метамодель для описания суперкомпьютеров, учитывающая развитие суперкомпьютерных технологий, дано описание измерительного процесса в суперкомпьютерном эксперименте с учетом метрологических понятий, на основе понятия о суперкомпьютерном эксперименте развита модель данных для рейтинговых списков, показана взаимосвязь между моделями, составляющими основу компонент AlgoWiki, CompZoo, PerfData платформы Algo500, при формировании рейтинговых списков и других запросов в компоненте RatingLists.

Для предлагаемых моделей и метамоделей сформулированы наиболее существенные ограничения в их использовании. Некоторые из ограничений могут быть сняты в результате принятия договоренностей, удовлетворяющих большинство пользователей платформы Algo500.

Проведенный анализ показал, что модели данных, лежащие в основе Algo500, требуют дальнейшего развития: в AlgoWiki предлагается ввести в вершины графа формализованные теоретические характеристики, в CompZoo — создать общие подходы к описанию вычислительных систем; в PerfData — разработать способы описания условий и результатов суперкомпьютерных экспериментов. Развитие моделей данных также следует из анализа запросов в RatingLists.

Авторы выражают благодарность студентам факультета ВМК МГУ имени М. В. Ломоносова Д. И. Личманову, А. Д. Матвееву, К. С. Мокрову, А. А. Щербакову за обсуждение ряда вопросов и помощь в работе.

Список литературы

1. *Antonov A.S., Nikitenko D.A., Voevodin Vl.V.* Algo500 — a new approach to the joint analysis of algorithms and computers // Lobachevskii J. Math. 2020. **41**, N 8. 1435–1443. doi 10.1134/S1995080220080041.
2. *Antonov A., Frolov A., Konshin I., Voevodin Vl.* Hierarchical domain representation in the AlgoWiki encyclopedia: from problems to implementations // Communications in Computer and Information Science. Vol. 910. Cham: Springer, 2018. 3–15. doi 10.1007/978-3-319-99673-8_1.
3. Открытая энциклопедия свойств алгоритмов. https://algowiki-project.org. Cited February 15, 2023.
4. *Воеводин Вл.В.* Открытая энциклопедия свойств алгоритмов AlgoWiki: от мобильных платформ до экзафлопсных суперкомпьютерных систем // Вычислительные методы и программирование. 2015. **16**, № 1. 99–111. doi 10.26089/NumMet.v16r111.
5. *Voevodin Vl.V., Antonov A.S., Dongarra J.* AlgoWiki: an open encyclopedia of parallel algorithmic features // Supercomputing Frontiers and Innovations. 2015. **2**, N 1, 4–18. doi 10.14529/jsfi150101.
6. *Popov A., Nikitenko D., Antonov A., Voevodin Vl.* Formal model of problems, methods, algorithms and implementations in the advancing AlgoWiki open encyclopedia // CEUR Workshop Proc. 2018. **2281**. 1–11. http://ceur-ws.org/Vol-2281/paper-01.pdf. Cited February 15, 2023.
7. *Antonov A.S., Maier R.V.* A new representation of algorithmic approaches in the AlgoWiki encyclopedia // Lobachevskii J. Math. 2021. **42**, N 7. 1483–1491. doi 10.1134/S1995080221070039.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

8. *Nikitenko D., Antonov A., Zheltkov A., Voevodin Vl.* Describing HPC system architecture for understanding its capabilities // Communications in Computer and Information Science. Vol. 1331. Cham: Springer, 2020. 425–435. doi 10.1007/978-3-030-64616-5_37.
9. *Желтков А.А.* Разработка методов построения рейтингов вычислительных систем, основанных на реализациях различных алгоритмов // Суперкомпьютерные дни в России: Труды международной конференции, 23–24 сентября 2019. М.: МАКС Пресс, 2019. 192–199.
10. *Antonov A.S., Maier R.V.* Development and implementation of the Algo500 scalable digital platform architecture // Lobachevskii J. Math. 2022. **43**, N 4. 837–847. doi 10.1134/S1995080222070058.
11. Home | TOP500. https://www.top500.org. Cited February 15, 2023.
12. Graph 500 | large-scale benchmarks. https://graph500.org. Cited February 15, 2023.
13. HPCG - June 2022 | Top500. https://www.top500.org/lists/hpcg/2022/06/. Cited February 15, 2023.
14. Green500 - June 2022 | Top500. https://www.top500.org/lists/green500/2022/06/. Cited February 15, 2023.
15. *Antonov A.S., Dongarra J., Voevodin Vl.* Algowiki project as an extension of the Top500 methodology // Supercomputing Frontiers and Innovations. 2018. **5** (1), 4–10. doi 10.14529/jsfi180101.
16. *Antonov A.* Wiki representation and analysis of knowledge about algorithms // Lecture Notes in Computer Science. Vol. 13708. Cham: Springer, 2022. 604–616. doi 10.1007/978-3-031-22941-1_44.
17. *Antonov A., Voevodin Vad., Voevodin Vl., Teplov A.* A study of the dynamic characteristics of software implementation as an essential part for a universal description of algorithm properties // Proc. 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. February 17–19, 2016, Heraklion, Greece. New York: IEEE Press, 2016. 359–363. doi 10.1109/PDP.2016.24.
18. *Боргест Н.М.* Научный базис онтологии проектирования // Онтология проектирования. 2013. № 1. 7–25.
19. *Guizzardi G.* Ontological foundations for structural conceptual models. Enschede: University of Twente, 2005. https://ris.utwente.nl/ws/portalfiles/portal/6042428/thesis_Guizzardi.pdf. Cited February 16, 2023.
20. *Войшвилло Е.К., Дегтярев М.Г.* Логика. М.: ВЛАДОС-ПРЕСС, 2001.
21. *Анисов А.М.* Современная логика. М.: Институт философии РАН, 2002.
22. *Бочаров В.А., Маркин В.И.* Основы логики. М.: ИНФРА-М, 1998.
23. *Субботин А.Л.* Классификация. М.: Институт философии РАН, 2001.
24. Системы управления терминологией, базами знаний и контентом. Концептуальные аспекты разработки и интернализации систем классификации: ГОСТ Р ИСО 22274-2016. М.: Стандартинформ, 2017.
25. Top50 | Суперкомпьютеры. http://top50.supercomputers.ru/list. Cited February 16, 2023.
26. *Перегудов Ф.И., Тарасенко Ф.П.* Введение в системный анализ. М.: Высшая школа, 1989.
27. *Тарасенко Ф.П.* Прикладной системный анализ (Наука и искусство решения проблем). Учебник. Томск: Изд-во Том. ун-та, 2004.
28. *Антонов А.С., Афанасьев И.В., Воеводин Вл.В.* Высокопроизводительные вычислительные платформы: текущий статус и тенденции развития // Вычислительные методы и программирование. 2021. **22**, № 2. 138–181. doi 10.26089/NumMet.v22r210.
29. *Voevodin Vl.V., Antonov A.S., Nikitenko D.A., et al.* Supercomputer Lomonosov-2: large scale, deep monitoring and fine analytics for the user community // Supercomputing Frontiers and Innovations. 2019. **6**, N 2. 4–11. doi 10.14529/jsfi190201.
30. PARALLEL.RU. Суперкомпьютерный комплекс МГУ, включая суперкомпьютер “ЛОМОНОСОВ-2”; | PARALLEL.RU - Информационно-аналитический центр по параллельным вычислениям. https://parallel.ru/cluster/lomonosov2.html. Cited February 16, 2023.
31. About Fugaku | RIKEN Center for Computational Science RIKEN Website. https://www.r-ccs.riken.jp/en/fugaku/about/. Cited February 16, 2023.
32. Frontier User Guide - OLCF User Documentation. https://docs.olcf.ornl.gov/systems/frontier_user_guide.html#system-overview. Cited February 16, 2023.
33. System Architecture - TACC Frontera User Guide. https://frontera-portal.tacc.utexas.edu/user-guide/system/. Cited February 16, 2023.
34. О стандартизации в Российской Федерации: федеральный закон от 29.06.2015 № 162—ФЗ // Собрание законодательства РФ. 2015. № 27. Ст. 3953.
35. Государственная система обеспечения единства измерений. Величины и единицы. Часть 13. Информатика и информационные технологии: ГОСТ Р МЭК 80000-13-2016. М.: Стандартинформ, 2017.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

36. Государственная система обеспечения единства измерений. Единицы величин: ГОСТ 8.417-2002. М.: Стандартинформ, 2019.
37. Международный словарь по метрологии: основные и общие понятия и соответствующие термины: пер. с англ. и фр. / Всерос. науч.-исслед. ин-т метрологии им. Д. И. Менделеева, Белорус. гос. ин-т метрологии. Изд. 2-е, испр. СПб.: НПО "Профессионал", 2010.
38. Государственная система обеспечения единства измерений. Метрология. Основные термины и определения: РМГ 29–2013. М.: Стандартинформ, 2014.
39. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
40. Государственная система обеспечения единства измерений. Шкалы измерений. Термины и определения: РМГ 83–2007. М.: Стандартинформ, 2008.
41. *Подвесовский А.Г., Коростелев Д.А., Лупачев Е.А., Беляков Н.В.* Построение хранилища обобщенных вычислительных экспериментов на основе онтологического подхода // Онтология проектирования. 2022. **12**, № 1. 41–56. doi 10.18287/2223-9537-2022-12-1-41-56.
42. HDF5 File Format Specification Version 2.0. http://davis.lbl.gov/Manuals/HDF5-1.8.7/H5.format.html. Cited February 17, 2023.
43. *Налимов В.В., Чернова Н.А.* Статистические методы планирования экстремальных экспериментов. М.: Наука, 1965.
44. *Налимов В.В.* Теория эксперимента. М.: Наука, 1971.

Поступила в редакцию
25 ноября 2022 г.

Принята к публикации
13 февраля 2023 г.

Информация об авторах

Александр Сергеевич Антонов — к.ф.-м.н., вед. научн. сотр.; Московский государственный университет имени М. В. Ломоносова, Научно-исследовательский вычислительный центр, Ленинские горы, 1, стр. 4, 119234, Москва, Российская Федерация.

Ростислав Валерьевич Майер — к.п.н., программист; Московский государственный университет имени М. В. Ломоносова, Научно-исследовательский вычислительный центр, Ленинские горы, 1, стр. 4, 119234, Москва, Российская Федерация.

References

1. A. S. Antonov, D. A. Nikitenko, and Vl. V. Voevodin, "Algo500 — A New Approach to the Joint Analysis of Algorithms and Computers," Lobachevskii J. Math. **41** (8), 1435–1443 (2020). doi 10.1134/S1995080220080041.
2. A. Antonov, A. Frolov, I. Konshin, and Vl. Voevodin, "Hierarchical Domain Representation in the AlgoWiki Encyclopedia: From Problems to Implementations," in *Communications in Computer and Information Science* (Springer, Cham, 2018), Vol. 910, pp. 3–15. doi 10.1007/978-3-319-99673-8_1.
3. Open Encyclopedia of Algorithmic Features. https://algowiki-project.org. Cited February 15, 2023.
4. Vl. V. Voevodin, "An Open AlgoWiki Encyclopedia of Algorithmic Features: From Mobile to Extreme Scale," Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie) **16** (1), 99–111 (2015). doi 10.26089/NumMet.v16r111.
5. Vl. V. Voevodin, A. S. Antonov, and J. Dongarra, "AlgoWiki: an Open Encyclopedia of Parallel Algorithmic Features," Supercomput. Front. Innov. **2** (1), 4–18 (2015). doi 10.14529/jsfi150101.
6. A. Popov, D. Nikitenko, A. Antonov, and Vl. Voevodin, "Formal Model of Problems, Methods, Algorithms and Implementations in the Advancing AlgoWiki Open Encyclopedia," CEUR Workshop Proc. **2281**, 1–11 (2018). https://ceur-ws.org/Vol-2281/paper-01.pdf. Cited February 15, 2023.
7. A. S. Antonov and R. V. Maier, "A New Representation of Algorithmic Approaches in the AlgoWiki Encyclopedia," Lobachevskii J. Math. **42** (7), 1483–1491 (2021). doi 10.1134/S1995080221070039.
8. D. Nikitenko, A. Antonov, A. Zheltkov, and Vl. Voevodin, "Describing HPC System Architecture for Understanding Its Capabilities," in *Communications in Computer and Information Science* (Springer, Cham, 2020), Vol. 1331, pp. 425–435. doi 10.1007/978-3-030-64616-5_37.

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

9. A. A. Zheltkov, "Development of Methods for Constructing Ratings of Computing Systems Based on Implementations of Various Algorithms," in *Proc. Int. Conf. on Russian Supercomputing Days, Moscow, Russia, September 23–24, 2019* (MAKS Press, Moscow, 2019), pp. 192–199.
10. A. S. Antonov and R. V. Maier, "Development and Implementation of the Algo500 Scalable Digital Platform Architecture," Lobachevskii J. Math. **43** (4), 837–847 (2022). doi 10.1134/S1995080222070058.
11. Home | TOP500. https://www.top500.org. Cited February 15, 2023.
12. Graph 500 | large-scale benchmarks. https://graph500.org. Cited February 15, 2023.
13. HPCG - June 2022 | Top500. https://www.top500.org/lists/hpcg/2022/06/. Cited February 15, 2022.
14. Green500 - June 2022 | Top500. https://www.top500.org/lists/green500/2022/06/. Cited February 15, 2023.
15. A. S. Antonov, J. Dongarra, and Vl. Voevodin, "Algowiki Project as an Extension of the Top500 Methodology," Supercomput. Front. Innov. **5** (1), 4–10 (2018). doi 10.14529/jsfi180101.
16. A. Antonov, "Wiki Representation and Analysis of Knowledge About Algorithms," in *Lecture Notes in Computer Science* (Springer, Cham, 2022), Vol. 13708, pp. 604–616. doi 10.1007/978-3-031-22941-1_44.
17. A. Antonov, Vad. Voevodin, Vl. Voevodin, and A. Teplov, "A Study of the Dynamic Characteristics of Software Implementation as an Essential Part for a Universal Description of Algorithm Properties," in *Proc. 24th Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing, Heraclion, Greece, February 17–19, 2016* (IEEE Press, New York, 2016), pp. 359–363. doi 10.1109/PDP.2016.24.
18. N. M. Borgest, "Scientific Basis for the Ontology of Designing," Ontology of Designing, No. 1, 7–25 (2013). https://www.ontology-of-designing.ru/article/2013_1(7)/Ontology_of_Designing_1_2013.pdf. Cited February 16, 2023.
19. G. Guizzardi, *Ontological Foundations for Structural Conceptual Models* (University of Twente, Enschede, 2005). https://ris.utwente.nl/ws/portalfiles/portal/6042428/thesis_Guizzardi.pdf. Cited February 16, 2023.
20. E. K. Voishvillo and M. G. Degtyarev, *Logic* (VLADOS-PRESS, Moscow, 2001) [in Russian].
21. A. M. Anisov, *Modern Logic* (Inst. of Philosophy, Moscow, 2002) [in Russian].
22. V. A. Bocharov and V. I. Markin, *Fundamentals of Logic* (INFRA-M, Moscow, 1998) [in Russian].
23. A. L. Subbotin, *Classification* (Inst. of Philosophy, Moscow, 2001) [in Russian].
24. GOST R ISO 22274-2016. Systems to manage terminology, knowledge and content. Concept-related aspects for developing and internationalizing classification systems. (Standartinform Publ., Moscow, 2017) [in Russian].
25. Top50 | Supercomputers. http://top50.supercomputers.ru/list. Cited February 16, 2023.
26. F. I. Peregudov and F. P. Tarasenko, "Introduction to Systems Analysis," (Vysshaya Shkola, Moscow, 1989) [in Russian].
27. F. P. Tarasenko, "Applied Systems Analysis (The Science and Art of Problem Solving): Textbook," (Tomsk State University Press, Tomsk, 2004) [in Russian].
28. A. S. Antonov, I. V. Afanasyev, and Vl. V. Voevodin, "High-Performance Computing Platforms: Current Status and Development Trends," Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie). **22** (2), 138–181 (2021). doi 10.26089/NumMet.v22r210.
29. Vl. V. Voevodin, A. S. Antonov, D. A. Nikitenko, et al., "Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community," Supercomput. Front. Innov. **6** (2), 4–11 (2019). doi 10.14529/jsfi190201.
30. PARALLEL.RU: Supercomputer Lomonosov-2. https://parallel.ru/cluster/lomonosov2.html. Cited February 16, 2023.
31. About Fugaku | RIKEN Center for Computational Science RIKEN Website. https://www.r-ccs.riken.jp/en/fugaku/about/. Cited February 16, 2023.
32. Frontier User Guide - OLCF User Documentation. https://docs.olcf.ornl.gov/systems/frontier_user_guide.html#system-overview. Cited February 16, 2023.
33. System Architecture - TACC Frontera User Guide. https://frontera-portal.tacc.utexas.edu/user-guide/system/. Cited February 16, 2023.
34. Russian Federation. The Federal Law of the Russian Federation N 162-FZ of July 29, 2015: "About Standardization in the Russian Federation" (Collection of Legislation of the Russian Federation, 2015, No. 27, Article 3953) [in Russian].
35. GOST 80000-13-2016. State system for ensuring the uniformity of measurements. Quantities and units. Part 13. Information science and technology. (Standartinform, Moscow, 2017) [in Russian].

2023, **24** (1), 89–114. doi 10.26089/NumMet.v24r107

36. GOST 8.417-2002. State system for ensuring the uniformity of measurements. Units of quantities. (Standartinform, Moscow, 2019) [in Russian].
37. *International vocabulary of metrology: basic and general concepts and associated terms (VIM)*. Joint Committee for Guides in Metrology (JCGM/WG 2). (NPO Professional, St. Petersburg, 2010) [in Russian].
38. RMG 29–2013. State system for ensuring the uniformity of measurements. Metrology. Basic terms and definitions. (Standartinform, Moscow, 2014) [in Russian].
39. V. V. Voevodin and Vl. V. Voevodin, *Parallel Computing* (BHV-Petersburg, St. Petersburg, 2002) [in Russian].
40. RMG 83–2007. State system for ensuring the uniformity of measurements. Scales of measurements. Terms and definitions. (Standartinform, Moscow, 2008) [in Russian].
41. A. G. Podvesovskii, D. A. Korostelyov, E. A. Lupachev, and N. V. Belyakov, "Building a Repository of Generalized Computational Experiments Based on the Ontological Approach," Ontology of Designing **12** (1), 41–56 (2022). doi 10.18287/2223-9537-2022-12-1-41-56.
42. HDF5 File Format Specification Version 2.0. http://davis.lbl.gov/Manuals/HDF5-1.8.7/H5.format.html. Cited February 17, 2023.
43. V. V. Nalimov and N. A. Chernova, *Statistical Methods for Planning Extreme Experiments* (Nauka, Moscow, 1965) [in Russian].
44. V. V. Nalimov, *Theory of Experiment* (Nauka, Moscow, 1971) [in Russian].

Received
November 25, 2022

Accepted for publication
February 13, 2023

Information about the authors

Alexander S. Antonov — Ph. D., Leading Scientist; Lomonosov Moscow State University, Research Computing Center, Leninskie Gory, 1, building 4, 119234, Moscow, Russia.

Rostislav V. Maier — Ph. D., programmer; Lomonosov Moscow State University, Research Computing Center, Leninskie Gory, 1, building 4, 119234, Moscow, Russia.