УДК 004
DOI: 10.26102/2310-6018/2024.47.4.015

Характеристическая функция акторной вычислительной системы

А.А. Зеленский, А.А. Грибков✉

Научно-производственный комплекс «Технологический центр», Москва, Зеленоград, Российская Федерация

Резюме. Статья посвящена исследованию задачи определения для акторной вычислительной системы комплексного показателя осуществимости, который может быть выражен в виде бинарной характеристической функции. Эта функция зависит от разрешимости и перечислимости множества промежуточных значений параметров решаемой вычислительной задачи, реализуемости вычислительной системы, т. е. ее способности выполнять весь комплекс необходимых вычислительных операций за заданный ограниченный интервал времени (цикл вычислений), а также от степени доверия к функциональной надежности и информационной безопасности вычислительной системы, выражаемых в виде интегрального показателя доверия. В статье излагается описание акторной модели вычислительной системы в рамках теории чисел. Предлагаемое описание опирается на представление вычислительной системы в виде композиции акторов – носителей функций, определения вычислимости этих функций, а также разрешимости и перечислимости числовых множеств значений параметров, задаваемых для вычислительной системы и возникающих в ней в процессе решения поставленных задач. Рассмотрены подходы к обеспечению разрешимости, реализуемости и доверия к вычислительной системе. Констатировано, что выбор память-ориентированной архитектуры вычислений, исходя из требования реализуемости, также является целесообразным с точки зрения обеспечения разрешимости, перечислимости и обеспечения доверия к вычислительной системе.

Ключевые слова: вычислительная система, акторная модель, память-ориентированная архитектура, осуществимость, реализуемость, вычислимость, разрешимость, перечислимость, доверие.

Благодарности: Исследование выполнено при поддержке Российского научного фонда по гранту № 24-19-00692, http://rscf.ru/project/24-19-00692/

Для цитирования: Зеленский А.А., Грибков А.А. Характеристическая функция акторной вычислительной системы. *Моделирование, оптимизация и информационные технологии.* 2024;12(4). URL: https://moitvivt.ru/ru/journal/pdf?id=1722 DOI: 10.26102/2310-6018/2024.47.4.015

Characteristic function of an actor computing system

A.A. Zelenskii, A.A. Gribkov✉

Scientific and Production Complex "Technological Center", Moscow, Zelenograd, the Russian Federation

Abstract. The paper is devoted to the study of the problem of determining a complex feasibility indicator for an actor computing system, which can be expressed as a binary characteristic function. This function depends on the solvability and enumerability of the set of intermediate values of the parameters of the computational problem to be solved, the feasibility of the computational system, i.e. its ability to perform the entire set of necessary computational operations for a given limited time interval (computation cycle), as well as on the degree of confidence in the functional reliability and information security of the computational system, expressed in the form of an integral confidence index. The paper presents a description of the actor model of a computing system in the framework of number theory. The proposed

description is based on the representation of a computing system in the form of a composition of actors – function carriers, definitions of computability of these functions, as well as solvability and enumerability of numerical sets of parameter values set for a computing system and arising in it in the process of solving the set tasks. Approaches to ensuring solvability, realisability and trust in the computational system are considered. It is stated that the choice of memory-oriented architecture of computations based on the requirement of realisability is also reasonable from the point of view of providing decidability, enumerability and ensuring trust to the computing system.

Keywords: computing system, actor model, memory-oriented architecture, feasibility, realisability, computability, solvability, enumerability, confidence.

Acknowledgements: The research was supported by the Russian Science Foundation under grant No. 24-19-00692, http://rscf.ru/project/24-19-00692/

For citation: Zelenskii A.A., Gribkov A.A. Characteristic function of an actor computing system. *Modeling, Optimization and Information Technology*. 2024;12(4). URL: https://moitvivt.ru/ru/journal/pdf?id=1722 DOI: 10.26102/2310-6018/2024.47.4.015 (In Russ.).

Введение

Ключевым фактором современного технологического развития являются вычислительные системы: высокопроизводительные, обеспечивающие обработку больших объемов данных, и быстродействующие, работающие со сравнительно небольшими объемами данных, но в режиме реального времени, то есть с жесткими ограничениями по длительности интервала времени, за который выполняется необходимый комплекс вычислительных операций.

Требование быстродействия, предъявляемое к вычислительным системам, направлено на обеспечение их реализуемости. Реализуемость вычислительной системы – это ее способность выполнить необходимое вычисление за интервал времени, равный или меньший заданного максимально допустимого при имеющихся вычислительных ресурсах. Интервал времени, за которое выполняется вычисление, называется циклом вычисления и зависит от совокупности интервалов времени, необходимых для выполнения всех вычислительных операций, из которых складывается указанное необходимое вычисление.

Требование реализуемости может быть сформулировано в двух формах: через требование к быстродействию, необходимому для выполнения потребного комплекса вычислений за заданное время, либо через требование к суммарному времени выполнения потребного комплекса вычислительных операций, которое должно быть меньше заданного максимально допустимого.

Наряду с реализуемостью, к вычислительным системам предъявляются еще две группы требований, определяющие их осуществимость, т. е. способность решать поставленные задачи.

Одна группа требований наиболее последовательно формулируется в рамках теории чисел и сводится к обеспечению разрешимости и перечислимости множества вычислительных параметров системы для всей совокупности решаемых посредством рассматриваемой вычислительной системы частных задач. Удовлетворение этого требования зависит от вычислимости задействованных в системе функций, соответствующих одной или нескольким выполняемым операциям и одновременно являющихся функциональной характеристикой акторов – элементов, в виде связанной совокупности которых представляется вычислительная система при использовании для ее описания акторной модели.

Другая группа требований связана с обеспечением функциональной надежности и информационной безопасности вычислительной системы. Эти требования отражает

доверие – интегральный показатель, объединяющий требования к функциональной надежности и информационной безопасности вычислительной системы на всех уровнях ее реализации (уровне электронной компонентной базы, аппаратном уровне, уровнях системного и прикладного программного обеспечения). Удовлетворение требованиям функциональной надежности и информационной безопасности обеспечивается при разработке и тестировании элементов, модулей и вычислительной системы в целом.

Логичным представляется рассмотрение требований в следующей последовательности. Вначале необходимо определить разрешимость композиции функций, определяющей вычислительную систему. Однако, даже если композиция разрешима, это еще не значит, что вычислительная система реализуема: требования по ее быстродействию (выражаемые через максимально допустимую длительность цикла вычислений) могут быть на практике невыполнимыми. Поэтому вторым этапом рассмотрения требований к вычислительной системе является оценка ее реализуемости. Наконец, заключительным этапом является определение уровня доверия к вычислительной системе. Если функциональная надежность и/или информационная безопасность системы недостаточно высоки для решения поставленных задач, то вычислительная система не может считаться осуществимой.

Требования осуществимости вычислительной системы

Вычислимость, разрешимость и перечислимость. Определение разрешимости и перечислимости множеств, определяющих вычислительную систему, авторы данной статьи предлагают проводить в рамках акторной модели [1, 2]. Основной областью приложения вычислительных систем, построенных на базе акторной модели, являются системы управления сложными объектами в режиме реального времени. Целью управления при этом является периодическое (с интервалом, равным длительности цикла вычисления) переопределение параметров объекта управления. При этом множество параметров объекта управления дискретно и имеет конечную мощность.

Основными структурными элементами акторной модели вычислительной системы являются акторы и их композиция. Актор (функция) – элемент, реализующий определенную функцию; способен получать и передавать сообщения (в том числе аргументы функций) от других акторов. Композиция акторов (функций) – совокупность акторов, объединенных в цикле вычисления в определенной конфигурации; конфигурация характеризуется последовательностью задействования функций (акторов), их распределением по потокам исполнения и группам, соответствующим этапам решения вычислительной задачи.

Элементы акторной модели (акторы и композиция) характеризуются вычислимостью, а формируемые ими в процессе функционирования множества – разрешимостью и перечислимостью [3, с. 8–15].

Множество входных и выходных значений параметров актора a_i:

$$X_i = \begin{Bmatrix} x_i^{(11)} & x_i^{(12)} & \dots \\ x_i^{(21)} & x_i^{(22)} & \dots \\ \dots & \dots & \dots \\ x_i^{(m1)} & x_i^{(m2)} & \dots \end{Bmatrix},\ Z_i = \begin{Bmatrix} z_i^{(11)} & z_i^{(12)} & \dots \\ z_i^{(21)} & z_i^{(22)} & \dots \\ \dots & \dots & \dots \\ z_i^{(k1)} & z_i^{(k2)} & \dots \end{Bmatrix}, \quad (1)$$

где m, k – число входных и выходных параметров i-го актора.

Множество значений входных, промежуточных и выходных параметров композиции Q:

$$X = \begin{Bmatrix} \chi_{11} & \chi_{12} & \cdots \\ \chi_{21} & \chi_{22} & \cdots \\ \cdots & \cdots & \cdots \\ \chi_{\omega 1} & \chi_{\omega 2} & \cdots \end{Bmatrix},\ Y = \begin{Bmatrix} \upsilon_{11} & \upsilon_{12} & \cdots \\ \upsilon_{21} & \upsilon_{22} & \cdots \\ \cdots & \cdots & \cdots \\ \upsilon_{\gamma 1} & \upsilon_{\gamma 2} & \cdots \end{Bmatrix},\ Z = \begin{Bmatrix} \zeta_{11} & \zeta_{12} & \cdots \\ \zeta_{21} & \zeta_{22} & \cdots \\ \cdots & \cdots & \cdots \\ \zeta_{\omega 1} & \zeta_{\omega 2} & \cdots \end{Bmatrix}, \quad (2)$$

где ω – число параметров на входе и выходе вычислительной системы (например, число параметров объектов управления), γ – число промежуточных параметров, задействованных в композиции.

Из указанных трех множеств неопределенным и подлежащим анализу является только множество Y (греч. «ипсилон»). Множество X (греч. «хи») определено как известные входные параметры (соответствующие исходному состоянию объекта управления), а множество Z (греч. «дзета») – как требуемые значения параметров объекта, которые должны быть обеспечены в процессе управления.

Композиция Q складывается как конъюнкция реализуемых в процессе вычислений алгоритмов:

$$Q = \cup_{j=1}^{n} q_j, \quad (3)$$

где q_j – j-й алгоритм из n формирующих Q.

В свою очередь q_j реализуется в виде композиции последовательно задействуемых акторов (функций), формирующих j-й алгоритм:

$$q_j = \circ_{i=1}^{m_j}\ a_i = a_1 \circ a_2 \ldots \circ a_i \ldots \circ a_{m_j}. \quad (4)$$

Вычислимость композиции $\varphi(Q)$ определяется вычислимостью $\varphi(q_j)$ образующих ее алгоритмов [1, с. 23]:

$$\varphi(Q) = \prod_{j=1}^{n} \varphi(q_j), \quad (5)$$

где $\varphi(q_j) = \{0,1\}$.

Это означает, что для выполнения всех задач вычислительной системы необходимо, чтобы все алгоритмы реализации этих задач обеспечивали получение решений.

Множество является разрешимым, если его характеристическая функция вычислима, и перечислимым, если оно разрешимо [1, с. 9, 13]. Для множества Y характеристическая функция представляет собой композицию функций, соответствующих акторам. Эта функция равна «1» на всех элементах множества и «0» – на элементах, не входящих во множество. Разрешимость $\Phi(Q)$ и перечислимость $T(Q)$ множества промежуточных значений параметров композиции определяются разрешимостью и перечислимостью множеств входных и выходных значений параметров акторов:

$$\forall(\Phi(X_i) = 1, \Phi(Z_i) = 1) | i = 1 \ldots n \rightarrow \Phi(Y) = 1,$$
$$\forall(T(X_i) = 1, T(Z_i) = 1) | i = 1 \ldots n \rightarrow T(Y) = 1. \quad (6)$$

Одним из характеризующих акторы параметров является область определения аргументов соответствующих им функций. В частности, в рамках теоретико-множественной концепции сложность объекта управления представляется в виде (дискретного, непрерывного или комбинированного) множества возможных значений. В процессе вычислений параметры объекта управления, определяющие его сложность, формируют дискретное конечное множество. Известно, что если множество конечно, то функция всегда вычислима, следовательно все акторы вычислимы, т. е. возможно вычисление соответствующих им функций некоторыми заданными алгоритмами.

Разрешимость и перечислимость множеств входных и выходных значений параметров акторов определяется вычислимостью функций, соответствующих акторам.

Если все функции (акторы) вычислимы, то вычислима и образованная из них композиция. Кроме того, если все акторы вычислимы, то множество значений композиции разрешимо.

Перечислимость множества значений композиции акторов не следует из вычислимости всех формирующих композицию акторов и разрешимости множеств входных и выходных значений их параметров. Для ее обеспечения необходимо дополнительное определение алгоритма перечисления возможных значений. В общем случае конечная мощность множества (дискретного, непрерывного или комбинированного) не означает его перечислимости, соответствующей исключительно дискретному представлению множества значений. В случае использования вычислительной системы для решения задач управления технологическим оборудованием дискретность и конечность множества значений параметров объекта управления означает перечислимость представляющего его множества значений как на входе и выходе, так и множества промежуточных значений.

Реализуемость вычислительной системы. Наибольшие сложности с обеспечением осуществимости вычислительной системы возникают в связи с реализуемостью вычислений, необходимых для управления сложными объектами в реальном времени. Преодоление указанных сложностей может основываться либо на повышении быстродействия используемой в вычислительной системе электронной компонентной базы, либо на совершенствовании архитектуры вычислений.

Повышение быстродействия электронной компонентной базы обеспечивается за счет использования в интегральных микросхемах все более «тонких» проектных норм. К настоящему времени промышленно уже освоены (на тайваньской компании TSMC[1]) проектные нормы 2 нм. В ближайшее время, вероятно, станут доступны проектные нормы 1 нм. Однако дальнейшее уменьшение проектных норм становиться все более сложным ввиду существующих объективных физических ограничений [4]. Кроме того, крайне высокая технологическая сложность оборудования для реализации проектных норм 7 нм и менее, выпускаемого всего одной компанией в мире (ASML, Нидерланды), дает возможность ограничения доступа к нему стран (Китая, России и др.), не относящихся к аффилированным США. Создание собственной промышленной базы для выпуска необходимого оборудования и микросхем на базе «сверхтонких» проектных норм, – задача чрезвычайно сложная, требующая формирования целого комплекса новых высокотехнологичных производств, что требует значительного времени и огромных инвестиций. В частности, суммарный объем инвестиций ведущих компаний – производителей микроэлектронного оборудования за последние 30 лет, позволившие им достичь текущего уровня технологий, составляет около 100 млрд. долл. [5]. Китай в настоявшее время реализует последовательную государственную промышленную политику по созданию собственной технологической базы в области микроэлектроники, однако, несмотря на значительные инвестиции, поставленные цели пока не достигнуты и в краткосрочной перспективе (5–10 лет) не будут достигнуты.

Альтернативный путь достижения высокого быстродействия вычислений – совершенствование их архитектуры таким образом, чтобы стало возможным конфигурирование цикла вычислений, т. е. чтобы формирующие вычисление операции могли (в определенных пределах) изменять порядок исполнения, распределение по потокам исполнения и группам, соответствующим этапам вычислений. В результате конфигурирования цикла вычислений суммарное время выполнения вычислений

[1] Annual Reports. Taiwan Semiconductor Manufacturing Company Limited, 2024. URL: https://investor.tsmc.com/english/annual-reports [Accessed 25th September 2024].

существенно сокращается, обеспечивая повышение быстродействия и достижение реализуемости вычислений.

Определение понятия реализуемости и описание последовательности ее количественной оценки было представлено авторами ранее, в предыдущих работах [6-8].

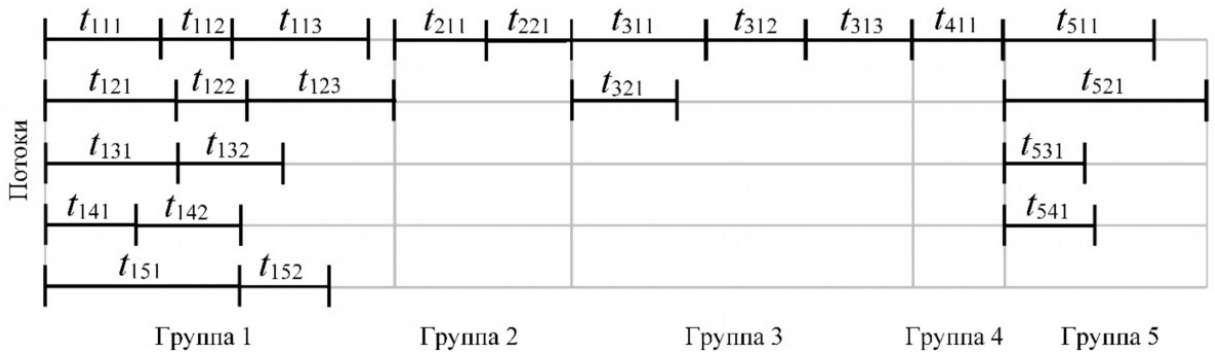

Рисунок 1 – Диаграмма цикла вычислений
Figure 1 – Calculation cycle diagram

Проведенный анализ показал, что длительность цикла вычислений может быть определена следующим образом [1] (Рисунок 1):

$$\tau = \sum_{u=1}^{u_{max}} max\left(\bigcup_{p=1}^{p_{u\,max}} \sum_{j=1}^{m_{up}} t_{uj}^{(p)}\right), \quad (7)$$

где $t_{uj}^{(p)}$ – длительность выполнения j-й операции p-го потока u-й группы; u_{max} – число групп операций в цикле вычисления; p_{umax} – количество параллельных потоков обработки данных в u-й группе; m_{up} – количество элементов в p-м параллельном потоке обработки данных в u-й группе.

Длительности выполнения вычислительных операций в формуле (7) предполагаются заданными, определяемыми имеющимся быстродействием вычислительной системы.

Исходя из обеспечиваемой длительности цикла вычислений (7) может быть сформулировано условие реализуемости вычислительной системы:

$$\tau \leq \tau_{max}, \quad (8)$$

где τ_{max} – максимально допустимая длительность цикла вычислений, определяемая целью вычислений (например, потребным быстродействием управления в реальном времени).

Максимальная сложность алгоритма, позволяющего определить оптимальную конфигурацию цикла вычислений, т. е. распределение операций по последовательности, потокам и группам, обеспечивающее наименьшее значение τ, соответствует нотации $O(n! \cdot 2^{n-1} \cdot n^2)$. По мере роста числа элементов (акторов вычислительной системы) сложность алгоритма быстро растет. Если при числе элементов, равном 10, в процессе оптимизации необходимо рассмотреть $1{,}8 \times 10^9$ вариантов, то при числе элементов, равном 20, требуется рассмотреть уже $1{,}2 \times 10^{24}$ вариантов.

Для снижения сложности алгоритма поиска можно использовать методы с выборкой последовательности элементов по устанавливаемым критериям, обеспечивающим репрезентативность при меньшем числе рассматриваемых вариантов. Пределом упрощения оптимизационного алгоритма за счет корректирования правил выборки является нотация $O(n^2)$, соответствующая сложности сортировки при простом

переборе. Дальнейшее упрощение задачи конфигурирования цикла вычисления связано с использованием лучших по времени и памяти алгоритмов сортировки (сортировки слиянием, пирамидальной сортировки, сортировки Хоара, сортировки с помощью двоичного дерева, различных гибридных методов сортировки, сочетающих вставку, слияние и поиск места вставки). Достижимым является снижение сложности используемого алгоритма конфигурирования до нотации $O(n \log n)$.

Использование указанных возможностей снижения сложности алгоритма конфигурирования цикла вычислений, очевидно, не позволит определить оптимальную конфигурацию. В случае высокой сложности решаемых задач (например, при управлении в реальном времени объектом высокой сложности) получаемая длительность цикла вычислений может превысить допустимую. В этом случае может оказаться целесообразным проведение комбинаторной оптимизации без каких-либо упрощений. Такая оптимизация, естественно, не может быть выполнена в режиме реального времени.

В некоторых случаях, напротив, сложность оптимизационной задачи оказывается сравнительно невысокой, т. е. достижение требуемой длительности цикла вычислений обеспечивается при не самой лучшей конфигурации. В таких случаях допустимо использование упрощенной комбинаторной оптимизации или адаптивного конфигурирования [9]. При использовании адаптивного конфигурирования в каждый момент времени задействуются максимально доступные ресурсы вычислительной системы, распределяемые по потокам исполнения, обеспечивая тем самым сокращение времени исполнения всех элементов группы и, в результате, всего цикла вычислений.

Условие реализуемости управления, соответствующее второй возможной форме требования реализуемости, определяет требуемое быстродействие в зависимости от длительности τ цикла вычислений и сложности Ω вычислительной задачи, решаемой за цикл вычислений (например, сложности объекта управления), имеет следующий вид [3]:

$$\Psi = \Omega / \tau. \quad (9)$$

В случае управления движением длительность цикла вычислений определяется допустимой погрешностью Δr воспроизводства траектории движения и скорости v этого движения:

$$\tau \leq \min (\Delta r / v). \quad (10)$$

Универсальной методологии определения сложности вычислительной задачи в настоящее время не существует. Применительно к вычислительной задаче, решаемой при управлении сложными объектами, авторами была предложена формула, соответствующая представлению сложности в рамках теоретико-множественной концепции сложности [10], в которой бесконечное множество состояний управляемого объекта сводится к счетному множеству дискретных состояний контролируемых параметров [3]:

$$\Omega = n \cdot m^{1/2} \cdot q \cdot g^{1/2} \cdot p \cdot s^{1/2}, \quad (11)$$

где n – число типов элементов, m – среднее число элементов одного типа, q – число типов связей, g – среднее число связей одного типа, p – среднее число контролируемых параметров, посредством которых описывается состояние отдельного элемента, s – среднее число отслеживаемых состояний контролируемых параметров.

Практическое использование условия реализуемости в виде требования к быстродействию затруднительно. Главным образом это связано с тем, что данная форма требования реализуемости не предполагает учета конфигурации цикла вычислений, напрямую влияющего на необходимое быстродействие. В результате, возможна лишь

качественная оценка реализуемости при сравнительном анализе аналогичных по конфигурации цикла вычислительных систем.

Доверие к вычислительной системе. Доверие к вычислительной системе – интегральный показатель, определяемый функциональной надежностью и информационной безопасностью системы. Необходимость использования показателя доверия обусловлена отличием любой вычислительной системы от ее идеального представления. При практической реализации вычислительной системы ее свойства могут отличаться от планируемых, при производстве может быть допущен брак, повреждения могут возникать в процессе эксплуатации, вычислительная система может быть повреждена или дискредитирована вследствие нарушений информационной безопасности и т. д.

Если оценивать уровень доверия на каждом из технологических уровней количественно в диапазоне от 0 до 1, где «0» – полное отсутствие доверия, а «1» – полное доверие, то интегральный показатель c уровня доверия к системе управления технологического оборудования будет рассчитываться по формуле [11]:

$$c = \prod_{i=1}^{4} c_{i1}\, c_{i2} = \prod_{i=1}^{4} \prod_{j=1}^{4} \left(\sum_{p=1}^{p_{imax}} \left(c_{ijp}^{E} w_{ijp}^{E}\right) + \left(1 - \sum_{p=1}^{p_{imax}} \left(c_{ijp}^{E} w_{ijp}^{E}\right)\right) \sum_{p=1}^{p_{imax}} \left(c_{ijp}^{T} w_{ijp}^{T}\right)\right), \quad (12)$$

где $c_{ij} = c_{ij}^{E} + \left(1 - c_{ij}^{E}\right) c_{ij}^{T}$ – показатели доверия к функциональной надежности (j = 1) и информационной безопасности (j = 2) на уровне электронной компонентной базы (i = 1), на уровне приборов (i = 2), на уровне системного программного обеспечения (i = 3) и на уровне прикладного программного обеспечения (i = 4); $c_{ij}^{E} = \sum_{p=1}^{p_{imax}} \left(c_{ijp}^{E} w_{ijp}^{E}\right)$ $c_{ij}^{T} = \sum_{p=1}^{p_{imax}} \left(c_{ijp}^{T} w_{ijp}^{T}\right)$ – показатели доверия к результатам разработки и результатам тестирования функциональной надежности или информационной безопасности на заданном i-м технологическом уровне; p_{imax} – число элементов вычислительной системы на i-м технологическом уровне; c_{ijp}^{E}, c_{ijp}^{T} – показатели доверия к результату разработки или тестирования p-го элемента на i-м технологическом уровне по j-му требованию (функциональной надежности или информационной безопасности); w_{ijp}^{E}, w_{ijp}^{T} – статистические веса $\left(\sum_{p=1}^{p_{imax}} w_{ijp}^{E} = 1;\ \sum_{p=1}^{p_{imax}} w_{ijp}^{T} = 1\right)$ p-го элемента на i-м технологическом уровне при оценке показателя доверия по j-му требованию к результату разработки или тестирования.

Обеспечение полного доверия к вычислительной системе, соответствующего c = 1, на практике не представляется возможным. Поэтому для каждой вычислительной системы в зависимости от решаемых задач и имеющихся возможностей (влияющих на определение уровня допустимого риска) устанавливается минимальный уровень доверия, который вычислительная система должна обеспечивать. Если уровень доверия оказывается ниже установленного минимального, то констатируется невозможность осуществления вычислительной системы.

Представление характеристической функции. Характеристическая функция вычислительной системы определяет возможность ее осуществления и зависит от трех вспомогательных функций: функции вычислимости композиции, функции реализуемости и функции доверия. Вспомогательные функции и характеристическая функция – бинарные, множество их значений {0,1}.

Функция вычислимости композиции функций (акторов) может быть определена через вычислимость всех задействованных в композиции функций, объединенных в цепочки алгоритмов:

$$\varphi(Q) = \prod_{j=1}^{n} \varphi\left(q_j\right). \quad (13)$$

Функция реализуемости вычислительной системы определяется тем, укладывается ли длительность цикла вычислений в установленные ограничения:

$$\theta(\tau) = \begin{cases} 1 \text{ при } \tau \le \tau_{max} \\ 0 \text{ при } \tau > \tau_{max} \end{cases}. \quad (14)$$

Функция доверия к вычислительной системе зависит от того, удовлетворяется ли требование по уровню доверия:

$$\varsigma(c) = \begin{cases} 1 \text{ при } c \ge c_{min} \\ 0 \text{ при } c < c_{min} \end{cases}. \quad (15)$$

Объединяя вспомогательные функции, получаем характеристическую функцию вычислительной системы:

$$\Xi(\varphi, \theta, \varsigma) = \varphi(Q) \cdot \theta(\tau) \cdot \varsigma(c) = \begin{cases} 1 \text{ при } \cap\,(\varphi(Q) = 1,\ \tau \le \tau_{max}, c \ge c_{min} \\ 0 \text{ при } \cup\,(\varphi(Q) = 0,\ \tau > \tau_{max}, c < c_{min} \end{cases}. \quad (16)$$

Обеспечение осуществимости вычислительной системы

Для чего нужна характеристическая функция? В решении каких проблем вычислительных систем нам может помочь знание значения характеристической функции?

Первым очевидным применением характеристической функции является выявление вычислительных систем, которые при имеющемся уровне реализации в акторах функций, быстродействии вычислений (зависящем от уровня используемой электронной компонентной базы, программного обеспечения и архитектуры вычислительной системы), качества электронных компонентов, программ и их безопасности, не могут быть осуществлены, а также причин невозможности осуществления.

Не менее важным, но существенно более сложным применением характеристической функции является определение направлений совершенствования вычислительных систем, позволяющих обеспечить их осуществимость.

Обеспечение в акторах вычислимости функций. Определение направлений совершенствования вычислительных систем начнем с рассмотрения обеспечения в акторах вычислимости функций. Для этого акторы должны удовлетворять ряду критериев, которые можно условно разделить на функциональные, структурные и наблюдательные.

К функциональным критериям относятся:

– применимость – актор должен быть способен обрабатывать все возможные типы сообщений, которые он может получить;

– корректность – значение функции, реализуемой актором, который должен быть корректным, т. е. иметь заданный формат представления (единственное значение, счетное число значений, интервал значений, вид зависимости нескольких аргументов и др.);

– определенность – актор должен иметь четко определенное поведение и набор инструкций.

К функциональным также относится критерий стабильности, формулируемый по-разному для детерминированных акторов, соответствующих арифметико-логическим элементам вычислительной системы, и недетерминированных акторов, соответствующих аналоговым элементам, в том числе на основе кубитов, модулям интеллектуальных подсистем, построенных на базе нейросетей, и др.

Применительно к детерминированным акторам определенность соответствует повторяемости – при одинаковых входных данных и состоянии актор должен всегда выдавать один и тот же результат. Для недетерминированных акторов повторяемость означает, что, при одинаковых входных данных и состоянии, результат, выдаваемый

актором, всегда соответствует заданному распределению, принимаемому случайным (вероятностным).

К структурным критериям относятся:

– закрытость – функция воспроизводится самим актором, в процессе воспроизведение возможно обращение для выполнения промежуточных операций к другим акторам этой же акторной модели;

– масштабируемость – актор должен сохранять свою вычислительную способность при увеличении рабочей нагрузки при соответствующем увеличении вычислительных ресурсов;

– модульность – логика актора должна быть разбита на модули, что упрощает понимание и управление процессом вычисления.

К наблюдательным критериям относятся:

– адекватность – процесс определения функции с заданной точностью, выполняемый актором, должен завершаться за число шагов и интервал времени, адекватные решаемой задаче;

– прозрачность – должна существовать возможность проверки правильности работы актора с помощью тестов или формальных методов.

Критерий правильности работы актора, используемый при определении критерия прозрачности, достоверно определяется лишь для детерминированных акторов. Для них правильность работы соответствует корректности и определенности (повторяемости).

Для недетерминированных акторов прозрачность не обеспечивается. Прозрачность предполагает выявление внутренних механизмов работы, для недетерминированных акторов (например, искусственного интеллекта) это невозможно. Такой актор представляет собой «черный ящик»: можно тестировать его способность выполнять свою функцию, но нельзя достоверно установить «правильность» работы без выявления внутренних механизмов работы, причем механизмов, задействованных при решении каждой конкретной задачи.

Для обеспечения вычислимости функций, соответствующих акторам, свойства акторов должны соответствовать указанным критериям функциональности, структуры и наблюдаемости. В случае недетерминированных акторов достижение указанного соответствия резко усложняется. Реализуемость для таких акторов приобретает вероятностный характер.

Однако, как показывает практика использования нейронных сетей, аналоговых и др. элементов, реализацией которых являются недетерминированные акторы, области определения значений функций и скорость их выполнения многократно расширяются при использовании таких элементов. Поэтому использование недетерминированных акторов, безусловно, оправдано и открывает значительные перспективы расширения функционала вычислительных систем для управления сложными объектами. При этом включение в акторную модель недетерминированных акторов требует использования дополнительных контрольных и проверочных процедур, интегрируемых в них на аппаратном или программном уровне, либо применения дополнительных детерминированных акторов.

Обеспечение разрешимости и перечислимости множеств. Возможности совершенствования вычислительной системы имеются также на уровне обеспечения разрешимости и перечислимости множеств. Условием обеспечения разрешимости и перечислимости множеств в акторной модели является их обеспечение для множеств входных X_i и выходных Z_i значений параметров каждого из акторов a_i. Множество X_i входных значений i-го актора определено либо начальными условиями, либо

предшествующими акторами из того же алгоритма. Возможны два подхода к реализации разрешимости и перечислимости.

Согласно первому подходу, все входные множества значений параметров акторов изначально обладают разрешимостью и перечислимостью. В этом случае задача сводится к обеспечение таких свойств (вычислимых) акторов, которые из разрешимого и перечислимого множества входных данных в результате применения заложенных в них функций получают разрешимые и перечислимые множества выходных данных.

Критерием разрешимости и перечислимости в рамках первого подхода является топологическая инвариантность (гомеоморфизм) входных и выходных множеств значений параметров, обеспечивающая биективное, инъективное, но не сюръективное отображение элементов входного множества на элементы выходного. Инъективное отображение – это отображение множества X_i во множество Z_i, при котором разные элементы множества X_i переводятся в разные элементы множества Z_i, причем если совпадают образы, то совпадают и прообразы. Инъективное отображение (при котором множество прообразов отображается в множество образов) также называют вложением или одно-однозначным соответствием. Сюръективным называют отображение множества X_i на множество Z_i, при котором каждый элемент множества Z_i является образом хотя бы одного элемента из множества X_i. Биективное (взаимно-однозначное) отображение, представлявшее собой одновременно сюръективное и инъективное отображение.

Согласно второму подходу, все входные множества значений параметров акторов подвергаются обработке с целью обеспечения их разрешимости и перечислимости. Указанная обработка заключается в нормализации элементов входных множеств или стандартизации множеств в целом, а также в ограничении мощности множеств. Нормализация предполагает обработку данных с целью приведения их к некоторой общей шкале без потери информации, стандартизация – обработку с целью приведения данных, в том числе категориальных, к единому формату и представлению. Одной из составляющих обработки как в случае нормализации, так и стандартизации является дискретизация данных. Ограничение мощности входных множеств основывается на их представлении в виде конечного комплекса дискретных данных и интервалов данных. Предпочтительным является его задание в виде дискретных данных, соответствующее теоретико-множественному представлению состояний объекта как множества дискретных значений его параметров.

Разрешимость входных и выходных множеств значений параметров акторов делает возможным задействования акторов при условии корректной организации и стабильности акторной модели. Корректность организации акторной модели означает способность акторов регламентированно взаимодействовать с другими акторами посредством сообщений или декларации состояния (случай реакторов [12, 13]). Стабильность акторной модели предполагает способность акторов продолжать функционировать корректно в случае ошибок или отказов отдельных компонентов на основе однозначной идентификации принадлежности элементов множеств входных и выходных данных тому или иному актору. Это позволяет восстанавливать последовательность выполнения алгоритмов при их разрыве.

Перечислимость входных и выходных множеств параметров акторов дает возможность акторам соответствовать наблюдательным критериям – адекватности и прозрачности. Однако для реализации указанной возможности недостаточно конечной мощности указанных множеств, мощность должна быть адекватной задачам, решаемым композицией функций. В частности, все образующие композицию функции должны быть отработаны за интервал времени, не превышающий длительность цикла

вычисления. Данное требование отсылает нас к реализуемости вычислительной системы.

При работе с перечислимым множеством входных данных каждый актор также должен соответствовать следующим функциональным критериям:

– ограниченность – вычислительные ресурсы и время отработки функции актора могут быть достоверно ограничены сверху мощностью множества входных данных и функциональными требованиями к композиции акторов;

– локализованность – могут быть определены положения относительно рассматриваемого актора в рамках выполняемых алгоритмов всех прочих акторов в модели, в том числе последовательность передачи и обработки сообщений.

Обеспечение реализуемости вычислительной системы. Приоритетным направлением обеспечения реализуемости вычислительной системы при заданном быстродействии используемой электронной компонентной базы является рассмотренное нами ранее конфигурирование цикла вычислений, представляющее собой поиск наилучшего распределения последовательности выполнения вычислительных операций (отожествляемых в рамках акторной модели с акторами – носителями функций), а также их распределения по потокам и группам исполнения. Параллельность отработки вычислительных операций не должна быть ограничена ничем, кроме закона Амдала, требующего учета имеющихся зависимостей одних вычислительных операций от других и неизбежной регламентации последовательности выполнения некоторых отдельных вычислительных операций и групп операций.

Для того, чтобы возможно было конфигурирование цикла вычислений в том виде, в котором мы его рассматривали в рамках комбинаторной оптимизации, необходима специальная организация вычислительной системы, обеспечивающая решение трех основных задач: уменьшения объема обрабатываемого потока данных, увеличения скорости передачи данных между функциональными элементами и блоками вычислительной системы, устранения очередей при обращении к памяти.

Задача уменьшения объема обрабатываемого потока данных имеет единственное решение в виде использования параллельных вычислений, причем параллельность должна быть реальной, т. е. потоки вычислений должны быть автономными. Для этого необходимо построение вычислительной системы на базе акторной модели, в которой акторы (функции) автономны, выполняются независимо (что не означает, что они не могут быть связаны в рамках включающих их алгоритмов) без детальной регламентации используемого способа выполнения заданных вычислительных операций. Последнее требование предполагает программирование вычислительной системы в рамках декларативной парадигмы, а также реализацию акторной модели посредством метапрограммирования, при котором для задействуемого актора генерируется собственная программа в связанной с актором локальной памяти.

Задача увеличения скорости передачи данных между функциональными элементами и блоками вычислительной системы, а также задача устранения очередей при обращении к памяти могут быть решены за счет обработки в памяти [14, 15] или вблизи памяти [16], реализуемой при построении вычислений в соответствии с память-ориентированной архитектурой (также называемой память-центрической [17]), при которой данные в процессе вычислений не перемещаются между процессором и памятью, а остаются в памяти, в которую интегрируется процессор. Память-ориентированная архитектура вычислений также предполагает гибридизацию вычислительной системы, совмещающей параллельное взаимодействие через передачу сообщений и через разделяемую память, что является эффективным решением проблемы очередей без использования снижающих быстродействие средств дополнительной логики (устройств арбитража, семафоров или прерывания) или построения

вычислительной системы на базе синхронных ОЗУ, позволяющих разделить по времени обращения различных устройств. При этом формируемая гибридная вычислительная система должна использовать двух- или многопортовую память – статическое оперативное запоминающее устройство с двумя или более независимыми интерфейсами, обеспечивающими доступ к пространству памяти через разделенные шины адреса, данных и управления [18].

Обеспечение доверия к вычислительной системе. Проблема обеспечения доверия лежит вне сферы теории вычислений. Данная проблема является в существенной мере технологической, в значительной степени – экономической и обусловлена низким уровнем развития и высокой импортной зависимостью отечественного производства вычислительных систем. Используемая электронная компонентная база и программное обеспечение до недавнего времени были в основном импортными, что резко обостряло проблему информационной безопасности (как на программном, так и аппаратном уровнях). Функциональное тестирование используемых элементов вычислительных систем – менее сложная задача, в существенной степени решенная, что, однако, не гарантирует доступности таких элементов в условиях действующих против нашей страны санкций.

Естественным путем выхода из сложившейся неблагоприятной ситуации является создание отечественного производства необходимых компонентов для вычислительных систем. Однако в настоящее время Россия не обладает всем комплексом технологий, требуемым для производства электронной компонентной базы на уровне ведущих мировых компаний. Следовательно, необходимо решать задачу создания вычислительных систем с высоким быстродействием за счет совершенствования архитекторы вычислений на базе доступных для отечественного микроэлектронного производства технологий (существенно уступающих мировому уровню). Путь решения указанной задачи – это реализация параллельных вычислений с построением вычислительной системы согласно акторной модели с использование декларативного метапрограммирования.

Частичным решением проблемы повышения доверия к вычислительным системам является развитие в стране системы тестирования программного обеспечения, сертификации элементов и аппаратных средств вычислительных систем. В настоящее время данная система развита недостаточно, что усугубляет проблему обеспечения доверия.

Заключение

Резюмируем проведенное в статье исследование:

1. Осуществимость вычислительной системы, т. е. способность решать поставленные задачи, определяется тремя группами требований: разрешимости и перечислимости множества вычислительных параметров системы, ее реализуемости и доверия к ней.
2. Разрешимость и перечислимость множества параметров вычислительной системы зависит от вычислимости функций, используемых для реализации всех алгоритмов вычислительной системы.
3. Реализуемость вычислительной системы – это ее способность выполнить необходимое вычисление за интервал времени, равный или меньше заданного максимально допустимого при имеющихся вычислительных ресурсах.
4. Доверие к вычислительной системе представляет собой интегральный показатель, объединяющий требования к функциональной надежности и информационной безопасности вычислительной системы на всех уровнях ее реализации

(уровне электронной компонентной базы, аппаратном уровне, уровнях системного и прикладного программного обеспечения).

5. Требование реализуемости вычислительной системы, удовлетворяемое при обеспечении параллельности вычислений, обусловливает выбор память-ориентированной архитектуры вычислительной системы на базе акторной модели с использованием декларативного метапрограммирования.

6. Характеристическая функция вычислительной системы определяет возможность ее осуществления и рассчитывается как произведение трех бинарных вспомогательных функций: функции вычислимости композиции (может быть определена через вычислимость всех задействованных в композиции функций, объединенных в цепочки алгоритмов), функции реализуемости (определяется тем, укладывается ли длительность цикла вычислений в установленные ограничения) и функции доверия (зависит от того, удовлетворяется ли требование по уровню доверия).

7. Выбранная, исходя из требования реализуемости, память-ориентированная архитектура вычислительной системы также способствует удовлетворению прочих требований к вычислительной системе.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Burgin M. Systems, Actors and Agents: Operation in a multicomponent environment. URL: https://arxiv.org/abs/1711.08319 [Accessed 25th September 2024].
2. Rinaldi L., Torquati M., Mencagli G., Danelutto M., Menga T. Accelerating Actor-Based Applications with Parallel Patterns. In: *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 13–15 February 2019, Pavia, Italy.* IEEE; 2019. pp. 140–147. https://doi.org/10.1109/EMPDP.2019.8671602
3. Верещагин Н.К., Шень А. *Лекции по математической логике и теории алгоритмов. Часть 3. Вычислимые функции.* Москва: МЦНМО; 2012. 160 с.
4. Leiserson C.E., Thompson N.C., Emer J.S., Kuszmaul B.C., Lampson B.W., Sanchez D., Schardl T.B. There's plenty of room at the Top: What will drive computer performance after Moore's law? *Science.* 2020;368(6495). https://doi.org/10.1126/science.aam9744
5. Грибков А.А., Зеленский А.А. Приоритеты развития микроэлектронной промышленности России. Часть 2. *Российский экономический вестник*. 2024;7(1):67–80.
   Gribkov A.A., Zelensky A.A. Priorities for the development of the microelectronic industry in Russia. Part 2. *Russian Economic Bulletin.* 2024;7(1):67–80. (In Russ.).
6. Зеленский А.А., Грибков А.А. Онтологические аспекты проблемы реализуемости управления сложными системами. *Философская мысль*. 2023;(12):21–31. https://doi.org/10.25136/2409-8728.2023.12.68807
   Zelenskii A.A., Gribkov A.A. Ontological aspects of the problem of realizability of control of complex systems. *Philosophical Thought.* 2023;(12):21–31. (In Russ.). https://doi.org/10.25136/2409-8728.2023.12.68807
7. Зеленский А.А., Кузнецов А.П., Илюхин Ю.В., Грибков А.А. Реализуемость управления движением промышленных роботов, станков с ЧПУ и мехатронных систем. Часть 1. *Вестник машиностроения*. 2022;(11):43–51. https://doi.org/10.36652/0042-4633-2022-11-43-51
   Zelenskiy A.A., Kuznetsov A.P., Ilyukhin Yu.V., Gribkov A.A. Feasibility of motion control of industrial robots, CNC machine tools and mechatronic systems. Part 1. *Vestnik Mashinostroeniya.* 2022;(11):43–51. (In Russ.). https://doi.org/10.36652/0042-4633-2022-11-43-51

8. Зеленский А.А., Кузнецов А.П., Илюхин Ю.В., Грибков А.А. Реализуемость управления движением промышленных роботов, станков с ЧПУ и мехатронных систем. Часть 2. *Вестник машиностроения.* 2023;102(3):213–220. https://doi.org/10.36652/0042-4633-2023-102-3-213-220
Zelenskiy A.A., Kuznetsov A.P., Ilyukhin Yu.V., Gribkov A.A. Feasibility of motion control of industrial robots, CNC machine tools and mechatronic systems. Part 2. *Vestnik Mashinostroeniya.* 2023;102(3):213–220. (In Russ.). https://doi.org/10.36652/0042-4633-2023-102-3-213-220
9. Зеленский А.А., Грибков А.А. Конфигурирование память-ориентированной системы управления движением. *Программные системы и вычислительные методы*. 2024;(3):12–25. (На англ.). https://doi.org/10.7256/2454-0714.2024.3.71073
Zelenskii A.A., Gribkov A.A. Configuration of memory-oriented motion control system. *Software systems and computational methods.* 2024;(3):12–25. https://doi.org/10.7256/2454-0714.2024.3.71073
10. Эшби У.Р. Теоретико-множественный подход к механизму и гомеостазису. В книге: *Исследования по общей теории систем.* Москва: Прогресс; 1969. С. 398–441.
Ashby W.R. The Set Theory of Mechanism and Homeostasis. In: *Issledovaniya po obshchei teorii sistem.* Moscow: Progress; 1969. pp. 398–441. (In Russ.).
11. Зеленский А.А., Морозкин М.С., Панфилов А.Н., Купцов В.Р., Грибков А.А. Обеспечение доверия к системам управления технологического оборудования. *Информатика, телекоммуникации и управление.* 2021;14(4):71–83. (На англ.). https://doi.org/10.18721/JCSTCS.14407
Zelensky A.A., Morozkin M.S., Panfilov A.N., Kuptsov V.R., Gribkov A.A. Ensuring confidence in control systems of technological equipment. *Computing, Telecommunications and Control*. 2021;14(4):71–83. https://doi.org/10.18721/JCSTCS.14407
12. Shah V., Vaz Salles M.A. Reactors: A Case for Predictable, Virtualized Actor Database Systems. URL: https://arxiv.org/abs/1701.05397 [Accessed 25th September 2024].
13. Lohstroh M., Menard C., Bateni S., Lee E.A. Toward a Lingua Franca for Deterministic Concurrent Systems. *ACM Transactions on Embedded Computing Systems*. 2021;20(4). https://doi.org/10.1145/3448128
14. Connolly M. *A Programmable Processing-in-Memory Architecture for Memory Intensive Applications.* Rochester Institute of Technology; 2021. 43 p.
15. Ghose S., Hsieh K., Boroumand A., Ausavarungnirun R., Mutlu O. Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms, Future Research Directions. URL: https://arxiv.org/abs/1802.00320 [Accessed 25th September 2024].
16. Singh G., Chelini L., Corda S., Awan A.J., Stuijk S., Jordans R., Corporaal H., Boonstra A.-J. Near-Memory Computing: Past, Present, and Future. *Microprocessors and Microsystems*. 2019;71. https://doi.org/10.1016/j.micpro.2019.102868
17. Каляев И., Заборовский В. Искусственный интеллект: от метафоры к техническим решениям. *Control Engineering Россия*. 2019;(5):26–31.
18. Мамаева Т. Микросхемы многопортовой памяти фирмы IDT. *Компоненты и технологии*. 2001;(4):32–34.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Зеленский Александр Александрович, кандидат технических наук, доцент, ведущий научный сотрудник Научно-производственного

Aleksandr A. Zelenskii, PhD in Technical Science, Docent, Leading researcher, Scientific and Production Complex

комплекса «Технологический центр», Москва, Зеленоград, Российская Федерация.
e-mail: zelenskyaa@gmail.com
ORCID: 0000-0002-3464-538X

"Technological Center", Moscow, Zelenograd, the Russian Federation.

Грибков Андрей Армович, доктор технических наук, ведущий научный сотрудник Научно-производственного комплекса «Технологический центр», Москва, Зеленоград, Российская Федерация.
e-mail: andarmo@yandex.ru
ORCID: 0000-0002-9734-105X

Andrei A. Gribkov, Doctor of Technical Science, Leading researcher, Scientific and Production Complex "Technological Center", Moscow, Zelenograd, the Russian Federation.

Статья поступила в редакцию 18.10.2024; одобрена после рецензирования 29.10.2024; принята к публикации 06.11.2024.

The article was submitted 18.10.2024; approved after reviewing 29.10.2024; accepted for publication 06.11.2024.