

**УПРАВЛЕНИЕ КОНВЕЙЕРНО-ПАРАЛЛЕЛЬНЫМИ ВЫЧИСЛЕНИЯМИ ПРИ
РЕШЕНИИ ЗАДАЧ КАЧЕСТВЕННОГО ИССЛЕДОВАНИЯ ДВОИЧНЫХ
ДИНАМИЧЕСКИХ СИСТЕМ НА ОСНОВЕ МЕТОДА БУЛЕВЫХ ОГРАНИЧЕНИЙ**

Опарин Геннадий Анатольевич

Д.т.н., проф., зам. директора, e-mail: prn51@icc.ru

Богданова Вера Геннадьевна

К.т.н., доцент, с.н.с., e-mail: bvg@icc.ru

Пашинин Антон Алексеевич

М.н.с., e-mail: apcrol@gmail.com

Институт динамики систем и теории управления имени В.М. Матросова
Сибирского отделения Российской академии наук (ИДСТУ СО РАН),
664033 г. Иркутск ул. Лермонтова, 134

Аннотация. Рассматриваются средства управления вычислениями в пакете прикладных микросервисов, ориентированном на применение при решении задач качественного исследования двоичных динамических систем на основе метода булевых ограничений. Вычислительная модель предметной области и архитектура агента обеспечивают возможность проведения многовариантных, параллельных и конвейерно-параллельных вычислений на выделенных ресурсах. Предложенный подход демонстрируется на примере анализа структуры пространства состояний сдвигового регистра.

Ключевые слова: пакет прикладных микросервисов, микросервис, управление вычислениями, конвейер, двоичные динамические системы.

Цитирование: Опарин Г.А., Богданова В.Г., Пашинин А.А. Управление конвейерно-параллельными вычислениями при решении задач качественного исследования двоичных динамических систем на основе метода булевых ограничений // Информационные и математические технологии в науке и управлении. 2019. № 3 (15). С. 79–90. DOI: 10.25729/2413-0133-2019-3-07

Введение. В последние годы большую значимость приобрели вопросы автоматизации расчетных работ в области исследования динамики и структурно-параметрического синтеза систем управления с целью определения как совокупности требований к динамическим показателям функционирования, так и набора требуемых динамических свойств, которым должно отвечать поведение системы [17]. По сути дела, речь идет о формировании динамического облика системы управления и связанного с этой задачей комплекса научно-исследовательских работ, выполняемых на уровне математических моделей, аналитических и численных методов их исследования. Процесс исследования динамики систем управления представляет собой последовательность многовариантных расчетов, на каждом шаге которой варьируются структура и значения параметров модели, методов и методик, выполняются запланированные расчеты, проводится оценка результатов и принимается решение о дальнейшем направлении исследований. Автоматизация высокопроизводительных научных вычислений при проведении исследований в приведенных выше предметных областях выполняется на основе разработанной ранее технологии HPCATAMP (High Performance

Computing Automated Technology for Implementation of Applied Microservices Package) [16], в которой фундаментальную роль играет алгоритмическое знание и сборочное программирование. Под сборочным программированием имеется в виду стиль программирования, при котором для решения задачи собирается на основе механизмов самоорганизации ансамбль готовых к использованию программных компонентов, запуск которых инициирует возникновение события. Используется децентрализованная дискретно-событийная модель вычислений. Первичным понятием в этой технологии является понятие пакета прикладных программ (ППП). Согласно модульному принципу, функциональное наполнение ППП является набором базисных, сравнительно простых, автономных вычислительных модулей, композицией которых можно получить решение всех задач рассматриваемого класса. Для реализации ППП на основе облачной парадигмы вычислений хорошо подходит технология микросервисов. Распределенная микросервисная модель представлена набором небольших, слабо связанных, заменяемых, взаимодействующих с использованием легковесных механизмов коммуникации автономных микросервисов, реализующих функции компонентов (модулей) приложения [15]. Реализованный на основе микросервисов ППП далее будем называть пакетом прикладных микросервисов (Applied Microservices Package, AMP). Целью исследований является разработка инфраструктуры (системной инструментальной программной среды) автоматизации разработки и применения AMP для исследования динамики и параметрического синтеза динамических систем управления, в частности, двоичных динамических систем (ДДС), на основе метода булевых ограничений [6]. Основное внимание в настоящей работе уделено средствам управления вычислениями в AMP. Развитие исследований, представленных в [1, 2], в направлении повышения отказоустойчивости, а также обработки потока однородных и разнородных задач обеспечило возможность поддержки конвейерно-параллельных вычислений при проведении многовариантных расчетов.

В последнее десятилетие применение конвейерного параллелизма для решения вычислительно сложных задач привлекает значительное внимание [4, 7, 11, 13]. В [14] описан ряд подходов, связанных с распределением ступеней конвейера по потокам многоядерных процессоров и на GPU, отмечается недостаток этих подходов, связанный с плохой балансировкой загрузки. В работе [10] рассматриваются все аспекты разработки бизнес-приложений на основе модели конвейера для многоядерной архитектуры, отмечается хорошая масштабируемость таких приложений и максимизация использования ресурсов. В более современной работе [9] представлена в первом приближении сервис-ориентированная архитектура распределенной интерфейсной системы для автономной обработки конвейера для решения задач биоинформатики. Во всех перечисленных системах требуется явно описать ступени конвейера программным способом. В разработанном авторами AMP BDSQA-BCM (Binary Dynamic System Qualitative Analysis based on Boolean Constraint Method) для решения задач качественного анализа ДДС на основе метода булевых ограничений конвейерные вычисления возникают в процессе решения задачи, поставленной непроцедурным способом. В результате логического вывода по непроцедурной постановке задачи (НПЗ) на распределенной модели предметной области (ПО) формируется активная группа агентов (Active Group, AG) для ее решения, взаимодействие которых определяется отношениями предшествования/следования (по входным/выходным параметрам, *In/Out*), заданным в локальных базах знаний. Для ряда задач может самоорганизовываться

асинхронный линейный конвейер [5]. При многовариантных расчетах разные ступени конвейера выполняются параллельно на выделенных ресурсах вычислительного поля (ВП) [1], на котором разворачивается семантическая сеть агентов.

1. Модель предметной области. Основным для интеллектуальных АМР является наличие модели предметной области (ПО), под которой понимается совокупность сведений об объектах ПО и отношениях между ними, а также возможность НПЗ: «по известным значениям объектов с именами a_1, a_2, \dots, a_k вычислить значения объектов с именами b_1, b_2, \dots, b_n ». Обозначим такую НПЗ как $T = (A^0; B^0; D(A^0))$, или кратко $T = (A^0; B^0)$, где $A^0 = \{a_1, a_2, \dots, a_k\}$, $B^0 = \{b_1, b_2, \dots, b_n\}$, $D(A^0)$ – значения параметров из множества A^0 . В качестве базы знаний (БЗ) используется модифицированная (расширены состав объектов и множество отношений между объектами ПО в сравнении с ранее разработанной и представленной в [2] моделью КВЕ) вычислительная модель

$$КВМ = (P, D, M, N, C, O, In, Out, Opt, Opp, Pr, Cmp, Cmo, Com),$$

где P, D, M, N, C, O – конечные множества соответственно параметров, значений параметров, модулей, узлов ВП, процессоров и операций, а отношения $In \subset P \times M$, $Out \subset M \times P$, $Opt \subset O \times M$, $Opp \subset O \times P$, $Pr \subset L \times O$ ($L \times P$), $Cmp \subset C \times P$, $Cmo \subset C \times O$, $Com \subset M \times N$ определяют взаимосвязи между соответствующими множествами и имеют тип “многие-ко-многим”. Отображение приведенных выше объектов в объекты АМР BDSQA-BCM приведено в таблице 1.

Таблица 1. Отображение объектов ПО в объекты BDSQA-BCM

ПО	BDSQA-BCM
Формальный параметр	Словарь параметров ПО
Операция простая/условная	Агент DSA Ordinary/Conditional
Модуль	Агент CMA (Computational Module Agent)
Процессор	Объект агента PSA (Problem Statement Agent)
Фрагменты отношений In, Out, Opt, Pr и Opp	локальные базы знаний КВМ ^{DSA}
Отношения Com и Cmp	КВМ ^{PSA}
Непроцедурная постановка задачи	Web-интерфейс агента PSA
Фактический параметр	Расчетные базы данных

2. Организация управления вычислениями. Управление вычислениями в АМР основано на принципах, которые лежат в основе разработанных авторами распределенных решателей прикладных задач [2]. В соответствии с этими принципами децентрализованное мультиагентное управление основано на прямых взаимодействиях агентов DSA (Distributed Solving Agent) семантической сети, которым делегированы права на запуск вычислительных микросервисов. Правила взаимодействия агентов (фрагменты отношений In и Out) заносятся в их локальные БЗ. Дискретно-событийная модель функционирования агентов описана в [2]. DSA агенты работают асинхронно. Для организации расчетов используются следующие средства:

- Условная управляющая структура модели KBM (отношение $Pr \subset L \times M$). Семантика этой структуры следующая: если $(l^*, m^*) \in Pr$ ($l^* \in L, m^* \in M$), то запуск модуля m^* может быть произведен только при условии, если значение параметра l^* равно TRUE.

- Статические многовариантные расчеты путем постановки задачи $T = (A^0 = \{V, V^{max}, V^{min}, H, \dots\}; B^0)$, где $V \in A^0$ – варьируемый параметр, V^{max}, V^{min}, H – границы и шаг изменения параметра. DSA агенту делегируются права на запуск модуля-генератора вариантов с этими параметрами.

- Динамические многовариантные расчеты. Генерацией вариантов в этом случае занимается DSA агент типа Variant, в модели которого программные действия однократного обращения к CMA заменены конечным автоматом, повторяющим последовательность действий (обращение к CMA для запуска модуля, получение и передача результатов) до сообщения об исчерпании вариантов.

- В многовариантных расчетах конвейерные вычисления могут выполняться параллельно.

Модель трехуровневого управления вычислениями при непроцедурной постановке задачи приведена на рис. 1. На рис. 2 приведена архитектура агента.

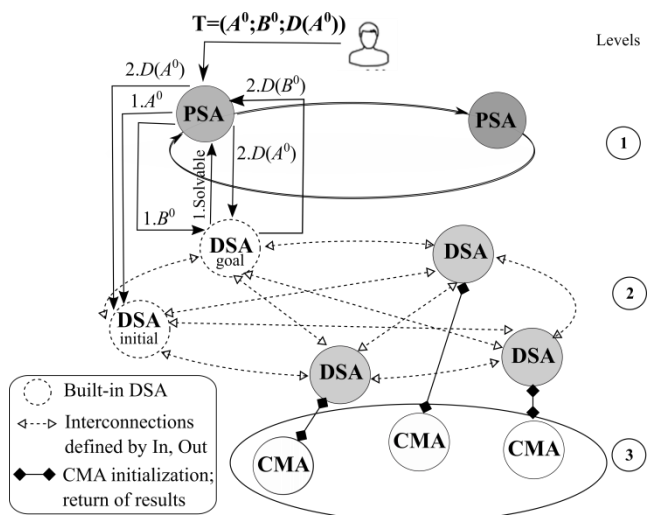


Рис. 1. Мультиагентное управление вычислениями

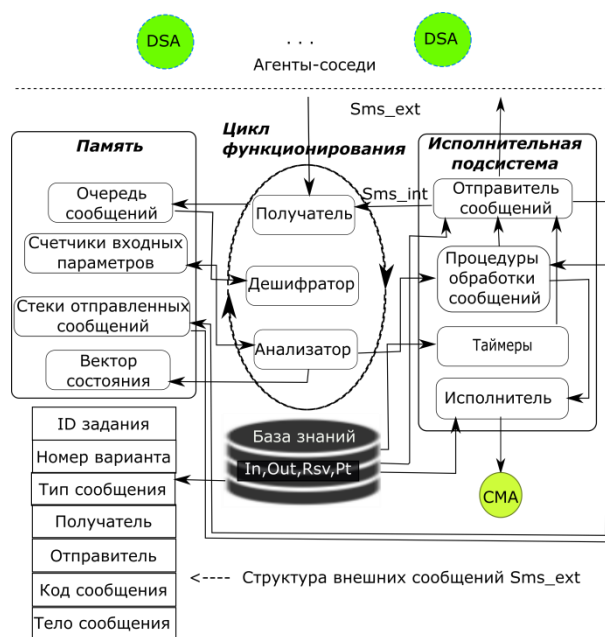


Рис. 2. Архитектура DSA агента

На первом шаге (рис. 1) формируется активная группа DSA агентов (в соответствии с отношениями взаимодействия пересылаются признаки вычислимости – имена параметров). Если задача не может быть решена в данной вычислительной сети, пользователь получает уведомление, и агенты деактивируются. В противном случае на втором шаге пересылаются значения параметров, но в вычислениях участвуют только агенты активной группы. Агент PSA отвечает за очередь непроцедурных постановок задач, DSA – за очередь сообщений, CMA – за очередь заданий. В процессе вычислений в ВП может образоваться несколько активных групп агентов. Один и тот же DSA может входить в разные группы и переключаться между расчетами по готовности данных. В структуре сообщения предусмотрен идентификатор (ID) постановки задачи пользователя и номер варианта. Эти

два значения и дата постановки задачи на агенте PSA составляют уникальный ключ, который используется при отслеживании готовности входных данных модуля в конкретной цепочке вычислений. Стеки отправленных сообщений также доступны по этому ключу. В случае сбоя узла в работу включается резервный агент, для которого по этим стекам восстанавливаются отправленные ранее сообщения с данными, полученными DSA-предшественником.

3. Иллюстративный пример. Продемонстрируем предлагаемый подход на важной с практической точки зрения и трудной в вычислительном плане задаче анализа структуры пространства состояний нелинейной двоичной динамической системы (ДДС), векторно-матричное уравнение которой имеет вид:

$$x^t = F(x^{t-1}, u^{t-1}), \quad (1)$$

где x и u – векторы соответственно состояния и входа, $x \in B^n$, $u \in B^m$, $B = \{0,1\}$, n и m – соответствующие размерности этих векторов; $t \in T = \{1,2,\dots,k\}$ – дискретное время (номер такта); $F(x)$ – векторная функция алгебры логики, называемая функцией переходов, ($F : B^n \times B^m \rightarrow B^n$).

Для каждого начального состояния $x^0 \in B^n$ и конечной последовательности $u(t) = (u^0, u^1, \dots, u^{k-1})$ состояний вектора входа ($u^t \in B^m, t = 0,1,2,\dots,k-1$) определим траекторию $x(t, x^0, u(t))$ системы (1) как конечную последовательность состояний x^0, x^1, \dots, x^k из множества B^n .

Качественный анализ ДДС со входами, основанный на исследовании структуры пространства состояний, предполагает решение следующих базовых задач: вычисление равновесных состояний и замкнутых траекторий (циклов), установление факта их изолированности, определение для заданных состояний их непосредственных предшественников, проверка свойства достижимости целевого множества $X^* \subset B^n$ из множества $X^0 \subset B^n$. Следует отметить, что эти задачи в настоящее время относятся к числу важнейших [8] при исследовании динамики поведения генных регуляторных сетей, представленных дискретной по времени и состоянию моделью (1), анализа различных типов устойчивости, применяемых в теории кодирования регистров сдвига с нелинейной обратной связью [12] и ряда других приложений. С теоретической точки зрения модель (1) является существенно нелинейной, и к ней неприменимы существующие методы теории линейных ДДС. В силу этого перечисленные задачи анализа структуры пространства состояний модели (1) сформулируем как задачи булевой выполнимости [6]. Прежде всего отметим, что модель (1) эквивалентна одному булеву уравнению вида

$$\Phi_k(x^0, x^1, \dots, x^k, u^0, u^1, \dots, u^{k-1}) = \bigvee_{i=1}^k \bigvee_{i=1}^n (x_i^t \oplus F_i(x^{t-1}, u^{t-1})) = 0. \quad (2)$$

где x_i^t – i -тая компонента вектора x^t ; \oplus – сложение по модулю 2; F_i – i -тая компонента вектора F . При $k = 1$ (рассматриваются только одношаговые переходы) уравнение (2) приобретает вид

$$\Phi_1(x^0, x^1, u^0) = \bigvee_{i=1}^n (x_i^1 \oplus F_i(x^0, u^0)) = 0. \quad (3)$$

С учетом (3) уравнение (2) запишем

$$\Phi_k(x^0, x^1, \dots, x^k, u^0, u^1, \dots, u^{k-1}) = \bigvee_{i=1}^k \Phi_1(x^{t-1}, x^t, u^{t-1}) = 0$$

Пусть входная последовательность $u(t)$ является постоянной на всем интервале функционирования системы (1): $u(t) = (u^0, u^1 = u^0, \dots, u^{k-1} = u^0) = u^0(t)$, $u^0 \in B^m$. Тогда максимальное количество структур пространства состояний системы (1) будет равно 2^m , каждая из которых представляет автономный характер поведения нелинейной системы (1) при соответствующем постоянном входном воздействии. Используя работу [6], приведем необходимые булевы уравнения для качественного исследования структуры пространства состояний автономной ДДС на конечном интервале времени T :

1. Все непосредственные предшественники x^0 состояния $s \in B^n$ и заданного $u^0 \in B^m$ являются нулями булева уравнения

$$\Phi_1(x^0, x^1, u^0(t)) \Big|_{x^1=s} = 0. \quad (4)$$

Если это уравнение не имеет решений, то предшественники состояния s отсутствуют.

2. Равновесные состояния для заданного $u^0 \in B^m$ являются решениями булева уравнения

$$\Phi_1(x^0, x^1, u^0(t)) \Big|_{x^1=x^0} = 0. \quad (5)$$

3. Проверка свойства достижимости за время $t \leq k$ тактов целевого множества X^* из множества X^0 сводится к решению булева уравнения

$$\overline{G^0}(x^0) \vee \Phi_k(x^0, x^1, \dots, x^k, u^0(t)) \vee \left(\bigvee_{t=1}^k \overline{G^*}(x^t) \right) = 0, \quad (6)$$

где $\overline{G^0}(x)$ и $\overline{G^*}(x)$ – характеристические функции множеств X^0 и X^* соответственно. Отсутствие решений для уравнения (4) свидетельствует о выполнимости свойства достижимости.

Фрагмент вычислительной модели приведен на рис. 3.

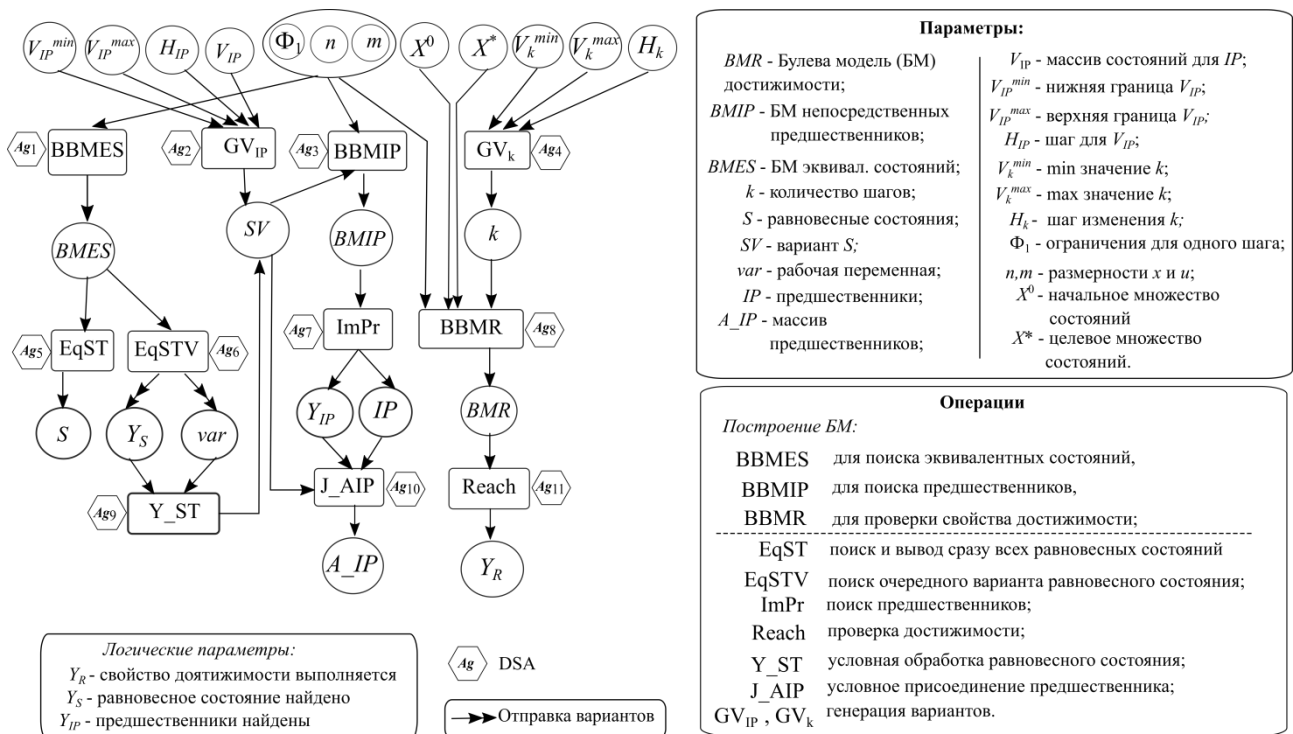


Рис. 3. Фрагмент вычислительной модели ПО

На этой модели можно выполнить следующие НПЗ:

- $T_1 = (A^0 = \{\Phi_1\}; B^0 = \{A_{IP}\}, D(A^0) = \{\Phi_1 = \text{“F1.cnf”}\})$ - поиск предшественников для равновесных состояний (конвейерно-параллельный динамический многовариантный расчет) (рис. 4);
- $T_2 = (A^0 = \{\Phi_1, V_{IP}, V_{IP}^{\min}, V_{IP}^{\max}, H_{IP}\}; B^0 = \{IP\}; D(A^0) = \{V_{IP} = \{14, 30\}, V_{IP}^{\min} = 1, V_{IP}^{\max} = 3, H_{IP} = +1, \Phi_1 = \text{“F1.cnf”}\})$ - статический многовариантный расчет. Поиск предшественников для набора вариантов состояний V_{IP} .
- $T_3 = (A^0 = \{\Phi_1, SV\}; B^0 = \{IP\}; D(A^0) = \{SV = 29, \Phi_1 = \text{“F2.cnf”}\})$ - поиск предшественников для конкретного состояния, например 29.
- $T_4 = (A^0 = \{\Phi_1, X^0, X^*, V_k^{\min}, V_k^{\max}, H_k\}; B^0 = \{Y_R\}; D(A^0) = \{V_k^{\min} = 1, V_k^{\max} = 8, H_k = *2, \Phi_1 = \text{“F2.cnf”}\}, X^0 = B^5 \setminus \{29, 14, 30, 7\}, X^* = \{0\}\})$ - статический многовариантный расчет, проверка достижимости множества X^* из множества X^0 .

Здесь уравнения динамики описаны в файле “F1.cnf” для $u^0(t) = 0$, и в файле “F2.cnf” для $u^0(t) = 1$. Рассмотрим первую НПЗ. По описанию динамики Φ_1 строится модель равновесного состояния, по этой модели выполняется поиск очередного равновесного состояния микросервисом EqSTV (вариант var , в этом случае логическая переменная Y_S устанавливается в TRUE). Микросервис Y_{ST} при истинном значении Y_S приводит var к формату SV , который используется микросервисом BBMIP для построения модели поиска непосредственных предшественников $BMIP$, передаваемой далее микросервису ImPr, осуществляющему по этой модели поиск предшественников, передаваемых при нахождении микросервису J_AIP. Получается конвейер $BBMES \rightarrow EqSTV \rightarrow Y_{ST} \rightarrow BBMIP \rightarrow ImPr \rightarrow J_{AIP}$ (рис. 4). На рис. 5 показана загруженность агентов при выполнении нескольких постановок задач. Вычисления проводились с использованием ресурсов [3].

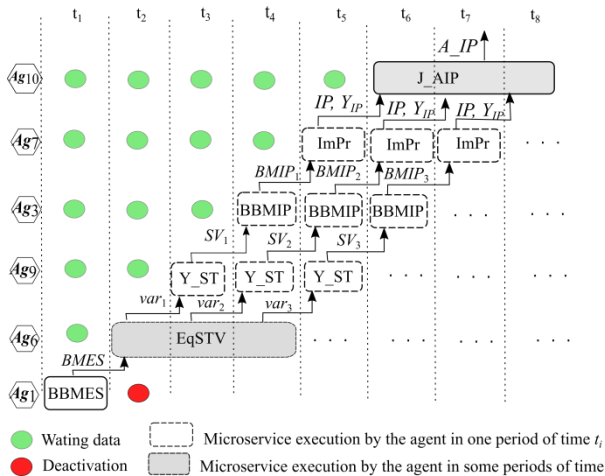


Рис. 4. Конвейерно-параллельный расчет для НПЗ T_1

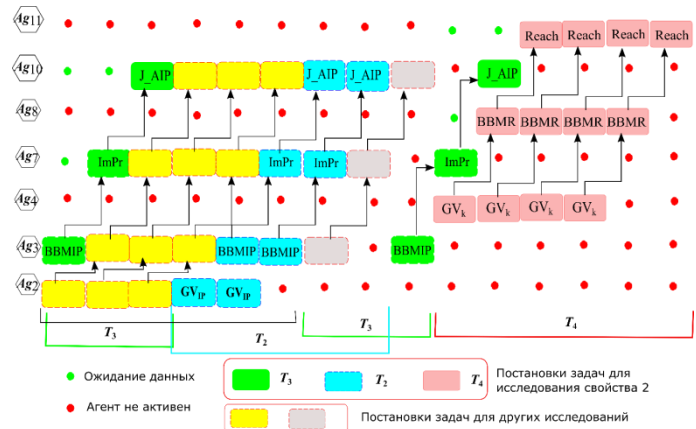


Рис. 5. Одновременное выполнение нескольких НПЗ

Анализ структуры пространства состояний сдвигового регистра. Рассмотрим технологию применения приведенных булевых ограничений и постановок непроцедурных задач для исследования свойства устойчивости регистра сдвига с нелинейной обратной связью [12], уравнения динамики которого имеют вид:

$$\begin{aligned}
 x_1^t &= \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \\
 &\vee u^{t-1} \cdot (\overline{x_1^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_2^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}}) \vee \\
 &\vee u^{t-1} \cdot (\overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}}), \\
 x_2^t &= x_1^{t-1} \cdot \overline{x_2^{t-1}} \cdot \overline{x_5^{t-1}} \vee u^{t-1} \cdot (\overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}}) \vee \\
 &\vee u^{t-1} \cdot (\overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}}), \\
 x_3^t &= \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_3^{t-1}} \vee u^{t-1} \cdot (\overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}}) \vee u^{t-1} \cdot \\
 &\cdot (\overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \vee \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}} \vee \overline{x_1^{t-1}} \cdot \overline{x_2^{t-1}} \cdot \overline{x_4^{t-1}} \cdot \overline{x_5^{t-1}}), \\
 x_4^t &= x_3^{t-1}, x_5^t = x_4^{t-1}.
 \end{aligned}$$

Здесь $x=col(x_1, x_2, x_3, x_4, x_5)$ – вектор состояния, $u(t)$ – скалярное входное воздействие (синдром в смысле работы [12]).

В [12] используется два свойства устойчивости сдвигового регистра.

- Свойство 1 ($u(t)=0$, режим самоконтроля регистра). Если для каждого состояния существует такое t , что $x(t, x^0, u^0(t) = 0) = 0$, то регистр сдвига с нелинейной обратной связью называется устойчивым. В противном случае этот регистр называют неустойчивым.
- Свойство 2 (на вход регистра поступает произвольное входное воздействие $u(t)$). Регистр сдвига с нелинейной обратной связью называется устойчивым по отношению к входному воздействию $u(t)$ тогда и только тогда, когда для любого состояния x^* , которое этот регистр может достичь из нулевого состояния при том или ином входном воздействии $u(t)$, существует такое t , что $x(t, x^*, u^0(t) = 0) = 0$.

В целях сокращения записи двоичное состояние регистра представим в виде десятичного эквивалента в предположении, что x_1 является младшим разрядом двоичного числа. Непроцедурные постановки задач для расчетов:

Путем решения булева уравнения (4) выявлено, что состояния 0 и 29 являются равновесными состояниями, причем состояние 29 является изолированным (НПЗ T_1). Это означает, что нулевое состояние недостижимо из состояния 29, и рассматриваемый регистр сдвига с нелинейной обратной связью является неустойчивым. Для проверки второго свойства находим непосредственных предшественников x^0 состояния 29 при $u^0(t)=1$ путем решения булева уравнения (4) (НПЗ T_3). Это будут состояния 14 и 30. При $u^0(t)=0$ состояние 14 имеет предшественника – состояние 7, состояние 30 не имеет предшественников (решается уравнение (4), НПЗ T_2 для $x_1 = 14$ и $x_1 = 30$). Состояния 30 и 7 не имеют предшественников как для $u^0(t)=0$, так и для $u^0(t)=1$, что означает недостижимость состояния 29 при любых входных воздействиях $u(t)$. Достижимые состояния составляют множество $X^0=B^5 \setminus \{29, 14, 30, 7\}$. Проверка достижимости множества $X^*=\{0\}$ из множества X^0 осуществляется путем решения уравнения (4) для $k=1, 2, 4, 8$ (НПЗ T_4). Элементы множеств X^0 и X^* являются соответственно решениями булевых уравнений

$$\begin{aligned}
 G^0(x) &: \overline{x_5} \cdot \overline{x_4} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \vee \overline{x_5} \cdot \overline{x_4} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \vee \overline{x_5} \cdot \overline{x_4} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \vee \\
 &\vee x_5 \cdot x_4 \cdot x_3 \cdot x_2 \cdot x_1 \vee x_5 \cdot x_4 \cdot x_3 \cdot x_2 \cdot x_1 = 0, \\
 G^*(x) &: x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 = 0.
 \end{aligned}$$

Дополнительно из множества X^0 исключено нулевое состояние. Таким образом, мы показали, что регистр с нелинейной обратной связью обладает свойством устойчивости по отношению к входному воздействию $u(t)$.

Заключение. В рамках разработки пакета прикладных микросервисов BDSQA-BCM для решения задач качественного исследования двоичных динамических систем на основе метода булевых ограничений предложен агентный декларативный подход к организации конвейерно-параллельных вычислений. На основе предложенного подхода проведено исследование свойства устойчивости регистра сдвига с нелинейной обратной связью. Разрабатываемый пакет BDSQA-BCM может применяться для решения широкого спектра задач, в том числе изучения динамики поведения генных регуляторных сетей, представленных дискретной по времени и состоянию моделью.

Работа подготовлена при поддержке РФФИ, проект № 18-07-00596.

СПИСОК ЛИТЕРАТУРЫ

1. Бычков И.В., Опарин Г.А., Богданова В.Г., Пашинин А.А. Интеллектуализация децентрализованного управления распределенными вычислениями // Вестник компьютерных и информационных технологий. 2017. № 10. С. 35–42. DOI: 10.14489/vkit.2017.10.
2. Бычков И.В., Опарин Г.А., Богданова В.Г., Пашинин А.А. Сервис-ориентированная технология создания и применения децентрализованных мультиагентных решателей вычислительных задач // Вестник компьютерных и информационных технологий. 2018. № 12. С. 36–44. DOI: 10.14489/vkit.2018.12.pp.036-044.
3. Иркутский суперкомпьютерный центр СО РАН. Режим доступа: <http://hpc.icc.ru/> (дата обращения 31.01.2018).
4. Кобайло А.С. Применение гибридных методов проектирования вычислительных систем реального времени // Труды БГТУ. Серия 3: Физико-математические науки и информатика. 2018. №1 (206). С. 120–124.
5. Кудряшова Е.С., Михайлова Н.Н., Хусаинов А.А. Моделирование конвейерных и волновых вычислений // Интернет-журнал Науковедение. 2014. Вып. 1. Режим доступа: <http://znanium.com/catalog/product/477304> (дата обращения 31.01.2018).
6. Опарин Г.А., Богданова В.Г., Пашинин А.А. Метод булевых ограничений в качественном анализе двоичных динамических систем // Международный журнал прикладных и фундаментальных исследований. 2018. № 9. С. 19–29. DOI: 10.17513/mjpf.12381.
7. Щербинина Н. В. Моделирование конвейерно-параллельного вычислителя с проблемной ориентацией // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. 2013. №15-1 С. 158–162.
8. Akutsu T., Hayashida M., Tamura T. Algorithms for inference, analysis and control of Boolean networks // In: International Conference on Algebraic Biology. 2008. Pp. 1–15. DOI:10.1007/978-3-540-85101-1_1.
9. Castillo L.F., López-Gartner G., Isaza G.A., Sánchez M., Arango J., Agudelo-Valencia D., Castaño S. GITIRBio: A Semantic and Distributed Service Oriented-Architecture for

- Bioinformatics Pipeline // *J Integr Bioinform.* 2015. 12(1):255. DOI: 10.2390/biecoll-jib-2015-255.
10. Isaacson C. *Software Pipelines and SOA: Releasing the Power of Multi-Core Processing* // Publisher: Addison-Wesley Professional. 2008.
 11. Maarala A.I., Bzhalava Z., Dillner J., Heljanko K., Bzhalava D. ViraPipe: scalable parallel pipeline for viral metagenome analysis from next generation sequencing reads *Bioinformatics.* 2018. Vol. 34(6). Pp. 928–935. DOI: 10.1093/bioinformatics/btx702.
 12. Massey J.L., Li R.W. Application of Lyapunov's direct method to error-propagation effect convoluted codes // In: *IEEE Trans. IT-10.* 1964. Pp. 248–250.
 13. Mirsoleimani S.A., van den Herik J., Plaat A., Vermaseren J. Pipeline Pattern for Parallel MCTS // *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018).* 2018. Vol. 2. Pp. 614–621.
 14. Navarro A., Asenjo R., Tabik S., Cascaval C. Analytical Modeling of Pipeline Parallelism // In *proceedings of 18th International Conference on Parallel Architectures and Compilation Techniques, Raleigh, NC, 2009.* Pp. 281–290. DOI: 10.1109/PACT.2009.28.
 15. Newman S. *Building Microservices.* O'Reilly. 2015.
 16. Oparin G.A., Bogdanova V.G., Pashinin A.A., Gorsky S.A. Microservice-oriented Approach to Automation of Distributed Scientific Computations // *Proceedings of 42st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Opatija. 2019. Pp. 253–258. DOI: 10.23919/MIPRO.2019.8757053.
 17. Somov Ye.I., Butyrin S., Oparin G.A., Bogdanova V.G. Methods and Software for Computer-Aided design of the Spacecraft Guidance, Navigation and Control Systems // *Mathematics in Engineering, Science and Aerospace (MESA).* 2016. Vol. 7. No. 4. Pp. 613–624.
-

UDK 004.421+004.4'2+004.771

**DEVELOPMENT OF SELF-ORGANIZING MULTIAGENT SYSTEM FOR
DECENTRALIZED CONTROL BY DISTRIBUTED SOLUTION OF APPLIED
PROBLEMS**

Gennadiy A. Oparin

Doctor of technical science, professor, Deputy Director, e-mail: prn51@icc.ru

Vera G. Bogdanova

PhD., assistant professor, senior researcher, e-mail: bvg@icc.ru

Anton A. Pashinin

Junior Researcher, e-mail: apcrol@gmail.com

Matrosov Institute for System Dynamics and Control Theory Siberian Branch of the Russian Academy of Sciences, 134, Lermontov Str., 664033, Irkutsk, Russia

Abstract. We consider the means of managing computations in the applied microservices package intended for the qualitative research of binary dynamic systems based on the Boolean constraints method. The computational domain model and the architecture of the agent provide the possibility of multivariate, parallel and

pipeline computing on dedicated resources. The proposed approach is demonstrated by the example of an analysis of the structure of the shift register state space.

Keywords: applied microservices package, microservice, calculations management, pipeline, binary dynamic systems.

References

1. Bychkov I.V., Oparin G.A., Bogdanova V.G., Pashinin A.A. Intellectualizatsiya detsentralizovannogo upravleniya raspredelennymi vychisleniyami [Intellectualization of decentralized management by distributed computations] // Vestnik kompyuternykh i informatsionnykh tekhnologiy = Herald of computer and information technologies. 2017. Vol. 12. Pp. 35–42. DOI: 10.14489/vkit.2017.10. (in Russian)
2. Bychkov I.V., Oparin G.A., Bogdanova V.G., Pashinin A.A. Servis-orientirovannaya tekhnologiya sozdaniya i primeneniya detsentralizovannykh multiagentnykh reshatelye vychislitelnykh zadach [Service-oriented technology for development and application of decentralized multiagent solvers for applied problems] // Vestnik kompyuternykh i informatsionnykh tekhnologiy = Herald of computer and information technologies. 2018. Vol. 12. Pp. 36–44. (in Russian)
3. Irkutskiy superkomp'yuternyy tsentr [Irkutsk Supercomputer Center of SB RAS]. Available at: <http://hpc.icc.ru/> (accessed 30-June-2019) (in Russian)
4. Kobaylo A.S. Primeneniye gibridnykh metodov proyektirovaniya vychislitelnykh sistem realnogo vremeni [The use of hybrid methods for the design of real-time computing systems] // Trudy BGTU. Seriya 3: Fiziko-matematicheskiye nauki i informatika = Proceedings of BSTU. Seriy 3. Physics and Mathematics. Informatics. 2018. №1 (206). Pp. 120–124. (in Russian)
5. Kudryashova E.C., Mikhaylova N.N., Khusainov A.A. Modelirovaniye konveyernykh i volnovykh vychisleniy [Modeling conveyor and wave computing] // Internet-zhurnal Naukovedeniye = Internet journal "Science of Science". 2014. Vyp. 1. Available at: <http://znanium.com/catalog/product/477304> (accessed 15/06/2019). (in Russian)
6. Oparin G.A., Bogdanova V.G., Pashinin A.A. Metod bulevykh ogranicheniy v kachestvennom analize dvoichnykh dinamicheskikh system [Boolean constraints method in qualitative analysis of binary dynamic systems] // Mezhdunarodnyy zhurnal prikladnykh i fundamentalnykh issledovaniy = International Journal of Applied and Basic Research. 2018. № 9. Pp. 19–29. DOI: 10.17513/mjpf.12381 (in Russian)
7. Shcherbinina N.V. Modelirovaniye konveyerno-parallelnogo vychislitelya s problemnoy oriyentatsiyey [Simulation of a pipeline-parallel computer with a problem orientation] // Nauchnyye vedomosti Belgorodskogo gosudarstvennogo universiteta. Seriya: Ekonomika. Informatika = Belgorod State University Scientific Bulletin Economics Information technologies. 2013. №15-1. Pp. 158–162. (in Russian)
8. Akutsu T., Hayashida M., Tamura T. Algorithms for inference, analysis and control of Boolean networks // In: International Conference on Algebraic Biology. 2008. Pp. 1–15. DOI:10.1007/978-3-540-85101-1_1.
9. Castillo L.F., López-Gartner G., Isaza G.A., Sánchez M., Arango J., Agudelo-Valencia D., Castaño S. GITIRBio: A Semantic and Distributed Service Oriented-Architecture for

- Bioinformatics Pipeline // *J Integr Bioinform.* 2015. 12(1):255. DOI: 10.2390/biecoll-jib-2015-255.
10. Isaacson C. *Software Pipelines and SOA: Releasing the Power of Multi-Core Processing* // Publisher: Addison-Wesley Professional. 2008.
 11. Maarala A.I., Bzhalava Z., Dillner J., Heljanko K., Bzhalava D. ViraPipe: scalable parallel pipeline for viral metagenome analysis from next generation sequencing reads *Bioinformatics.* 2018. Vol. 34(6). Pp. 928–935. DOI: 10.1093/bioinformatics/btx702.
 12. Massey J.L., Li R.W. Application of Lyapunov's direct method to error-propagation effect convoluted codes // In: *IEEE Trans. IT-10.* 1964. Pp. 248–250.
 13. Mirsoleimani S.A., van den Herik J., Plaat A., Vermaseren J. Pipeline Pattern for Parallel MCTS // *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018).* 2018. Vol. 2. Pp. 614–621.
 14. Navarro A., Asenjo R., Tabik S., Cascaval C. Analytical Modeling of Pipeline Parallelism // In *proceedings of 18th International Conference on Parallel Architectures and Compilation Techniques, Raleigh, NC, 2009.* Pp. 281–290. DOI: 10.1109/PACT.2009.28.
 15. Newman S. *Building Microservices.* O'Reilly. 2015.
 16. Oparin G.A., Bogdanova V.G., Pashinin A.A., Gorsky S.A. Microservice-oriented Approach to Automation of Distributed Scientific Computations // *Proceedings of 42st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Opatija. 2019. Pp. 253–258. DOI: 10.23919/MIPRO.2019.8757053.
 17. Somov Ye.I., Butyrin S., Oparin G.A., Bogdanova V.G. Methods and Software for Computer-Aided design of the Spacecraft Guidance, Navigation and Control Systems // *Mathematics in Engineering, Science and Aerospace (MESA).* 2016. Vol. 7. No. 4. Pp. 613–624.