

Сибирское отделение
Российской академии наук

ISSN 2073-0667

ПРОБЛЕМЫ ИНФОРМАТИКИ

ПРОБЛЕМЫ ИНФОРМАТИКИ № 1 2024



ТЕОРЕТИЧЕСКАЯ И СИСТЕМНАЯ ИНФОРМАТИКА

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1
—
2024

ПРОБЛЕМЫ ИНФОРМАТИКИ № 1 (62) 2024 г.

Журнал выходит ежеквартально, издается с 2008 г.

Учредитель журнала — Институт вычислительной математики и математической геофизики СО РАН.

Редакционный совет

Председатель — акад. НАН РК М. Н. Калимолдаев,
акад. РАН А. Л. Асеев, проф. В. А. Васенин, акад. РАН С. Н. Васильев, проф. В. М. Вишневецкий, акад. РАН С. С. Гончаров, акад. РАН Н. А. Кузнецов, акад. РАН А. П. Кулешов, проф. РАН М. А. Марченко, проф. А. Г. Марчук, А. Ю. Пальянов, проф. Б. Я. Рябко, акад. РАН И. А. Соколов, проф. А. Н. Сотников, чл.-кор. РАН Ю. А. Флеров.

Редколлегия

Главный редактор — проф. В. Э. Малышкин,
Д. Ж. Ахмед-Заки, А. Г. Вострецов, Б. С. Гольдштейн, В. И. Гужов, Ю. А. Загорулько, С. Д. Каракозов, В. Н. Касьянов, О. В. Кибис, В. В. Корнеев, И. В. Котенко, И. М. Куликов, М. Г. Курносов, Т. П. Любимова, А. Н. Ляхов, В. В. Окольников, Б. В. Поллер, А. С. Родионов (зам. гл. редактора), М. А. Сонькин, В. В. Шахов (зам. гл. редактора), М. С. Хайретдинов, И. Г. Черных, Moonseong Kim (Korea), V. D. Nguyen (Vietnam), Michele Pagano (Italy).

Редакция: отв. секретарь М. С. Делидович, системный администратор В. А. Перепелкин, верстка Д. В. Лазуткин, логист Л. В. Трофимова.

Адрес редакции, издателя: 630090, г. Новосибирск, просп. Академика Лаврентьева, д. 6, ИВМ и МГ СО РАН

тел. (383) 330-96-43; e-mail: problem-info@sscc.ru, <http://www.problem-info.sccc.ru>.

Журнал зарегистрирован в Федеральной службе по надзору в сфере массовых коммуникаций, связи и охраны культурного наследия. Свидетельство ПИ № ФС77-32088 от 27 мая 2008 г. Журнал распространяется по подписке. Оформление подписки на сайте «Объединенного каталога „Пресса России“» https://www.pressa-rf.ru/cat/1/edition/y_e69980/, подписной индекс 69980, и через интернет-магазин «Пресса по подписке» https://www.akc.ru/itm/problemy_i-informatiki/. Цена свободная. Журнал распространяется на территории России.

Журнал включен в Перечень ведущих рецензируемых научных журналов, рекомендованных для публикаций Высшей аттестационной комиссией.

Все права авторов сохранены. Использование материалов журнала возможно только с разрешения редакции и авторов.

Отпечатано в типографии «АЛЕКСПРЕСС» ИП Малыгин Алексей Михайлович.

Адрес: 630090, Новосибирск, пр-т Академика Лаврентьева, 6/1, оф. 104, тел. +7 (383) 217-43-46. Формат 60 × 84 1/8. Усл. печ. л. 11,63. Печать офсетная.

Тираж 50 экз. Заказ № 896. Подписано в печать 15.03.2024 г. Выход в свет 29.03.2024 г.

JOURNAL “PROBLEMS OF INFORMATICS”. No. 1 (62) 2024

Publisher: Institute of Computational Mathematics and Mathematical Geophysics of Siberian Branch of Russian Academy of Sciences.

Editorial Council

Chairman Academician of the National Academy of Sciences of the Republic of Kazakhstan

M. N. Kalimoldayev

Full Member of the RAS A. L. Aseev, Professor V. A. Vasenin, Full Member of RAS C. N. Vassilyev, Professor V. M. Vishnevsky, Full Member of RAS S. S. Goncharov, Full Member of RAS

N. A. Kuznetsov, Full Member of RAS A. P. Kuleshov, Professor of RAS M. A. Marchenko, Professor

A. G. Marchuk, A. YU. Palyanov, Professor B. Y. Ryabko, Full Member of RAS I. A. Sokolov,

Professor A. N. Sotnikov, Corr. Member RAS Y. A. Flerov.

Editorial board

The Editor-in-Chief Professor V. E. Malyshkin

Associate Editors-in-Chief: A. S. Rodionov, V. V. Shakhov

D. Zh. Akhmed-Zaki, A. G. Vostretsov, B. S. Goldstein, V. I. Guzhov, Y. A. Zagorulko, S. D.

Karakozov, V. N. Kasyanov, O. V. Kibis, V. V. Korneev, I. V. Kotenko, I. M. Kulikov, M. G. Kurnosov,

T. P. Lyubimova, A. I. Lyakhov, V. V. Okolnishnikov, B. V. Poller, M. A. Sonkin, M. S. Khairtdinov,

I. G. Chernykh, Moonseong Kim (Korea), Van Duc Nguyen (Vietnam), Michele Pagano (Italy).

Editorial staff: Managing Editor M. S. Delidovich, System Administrator V. A. Perepelkin,

Maker-up D. V. Lazutkin, Logistician L. V. Trofimova.

Address of the editorial office: 630090, pr. Lavrentieva, 6, Novosibirsk, Russia, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS.

Phone: +7 (383) 330-96-43; e-mail: problem-info@sscc.ru, <http://www.problem-info.sccc.ru>.

The journal has been registered in accordance with Legislation of the Russian Federation. Certificate of Mass Media Registration: ПИ № ФС77-32088, of 27 May, 2008, ISSN 2073-0667. The journal is distributed in Russia.

The journal “Problems of Informatics” is in the List of Peer-Reviewed Scientific Journals for publication of scientific results of Ph.D. and Dr. of Sci.

All rights reserved. The journal contents may only be used by the permission of editors and authors.

СОДЕРЖАНИЕ

Теоретическая и системная информатика

- Зубарев А. Ю.* Иерархия эквивалентностей непрерывно-временных сетей Петри со слабой временной стратегией 5
- Старченко А. В.* Параллельный численный метод решения гидродинамических уравнений в приближении мелкой воды для вычислительных систем с общей памятью 41

Прикладные информационные технологии

- Носов В. С., Сонькин Д. М., Сонькин М. А., Чурсин Ю. А.* Способ организации функциональной диагностики в региональных системах оповещения населения 57
- Ульяничев И. С., Винс Д. В.* Архитектура распределенной вычислительной системы на базе мобильных устройств 74

- Правила представления и подготовки рукописей для публикации в журнале „ПРОБЛЕМЫ ИНФОРМАТИКИ“ 98

На сайте «Объединенного каталога "Пресса России"» www.pressa-rf.ru
можно оформить подписку на печатную версию журнала
«Проблемы информатики» по подписному индексу 69980,
а также подписаться через интернет-магазин
«Пресса по подписке» www.akc.ru

CONTENTS

Theoretical informatics

<i>Zubarev A. Y.</i> Hierarchy of Equivalences on time Petri nets with weak time policy	5
<i>Starchenko A. V.</i> Parallel numerical method for solution of hydrodynamic equations in the shallow water approximation for shared memory computers	41

Applied information technologies

<i>Nosov V. S., Sonkin D. M., Sonkin M. A., Chursin Yu. A.</i> A method for organizing functional diagnostics in regional public warning systems	57
<i>Ulyanichev I. S., Wiens D. V.</i> Architecture of a distributed computing system based on mobile devices	74
Rules of presentation and preparation of manuscripts offered for publication	98

HIERARCHY OF EQUIVALENCES ON TIME PETRI NETS WITH WEAK TIME POLICY

A. Y. Zubarev

A. P. Ershov Institute of Informatics Systems,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2024-1-5-40

EDN: UMPUBR

The security of many information systems is critically important, since their failure can lead to large economic losses and human casualties. Equivalences allow comparing systems and specifications in terms of different aspects of their behavior. In this way, they are used to verify and reduce systems. Models of concurrent and distributed systems are compared in the dichotomies of “*linear time – branching time*” and “*interleaving – partial order*” in the literature. The first criterion of comparison is a degree of conflict accounting (a moment of choosing between several “branches” of computing). The semantics of linear time ignores the points of choice, and the behavior is described by the set of all executions (processes). *Trace equivalence* is a typical example of such equivalence. On the contrary, branching time semantics takes into account the branching structure of the system’s behaviors. The second criterion of comparison is the degree of partial order accounting. Interleaving semantics ignores partial order. The concurrency is reduced to a nondeterministic choice between executions of linearly ordered actions. On the other hand, “true concurrency” models preserve causality and concurrency between actions.

Petri nets (PN) are one of the generally accepted tools for modeling and analyzing concurrent and distributed systems. The (PN) consists of two different sets of elements – *transitions* (system events) and *places* (conditions for events). This model is represented as an oriented bipartite graph. A state of the PN is a *marking* (a set of places with *tokens*). All conditions for an event are met if there is a token in each input place of the corresponding transition. This transition is called *enabled* and can fire, i. e. change the marking. The new marking is obtained as a result of removing tokens from the input places and creating new tokens in the output transition places. It is often necessary to take into account time or quantitative characteristics when verifying systems. Various time extensions of Petri nets have been defined for these purposes. *Time Petri Nets (TPNs)* are model where each transition has its own clock and time interval. A *state* of the model is defined by the marking and the clock readings of enabled transitions. A transition *can fire* if it is enabled in the marking and its clock readings are in its interval. Firing the transition changes the marking, disables the clock for not enabled transitions, and initializes (resets) a clock for some enabled transitions.

There is an *intermediate*, *atomic* and *persistent atomic* clock reset policy. Intermediate policy takes into account the marking before the creation of new tokens and after the removal of old tokens (intermediate marking). A transition clock will be disabled if the transition was not enabled in the intermediate marking. If a transition became enabled after the creation of new tokens, then its clock will take a zero value. The atomic policy does not consider intermediate states. Resetting clock will only happen if the transition is disabled in the new marking. The transition clock that is fired is always reset in the case of an intermediate and atomic policy (unlike persistent atomic). It is shown in [1] and [2] that the persistent atomic policy is strictly more expressive than the atomic one. The

atomic policy is not less expressive than the intermediate one. In addition, the authors give examples of models of systems where the atomic policy is more preferable.

Time elapsing increases the readings of the local clock of enabled transitions. *Strong* and *weak time policies* are presented in the literature. The strong policy restricts time elapsing. The value of a transition clock cannot go beyond the upper limit of the time interval. The transition must either fire or become disabled. In the weak policy, there are no restrictions on time elapsing. As it was shown in [3], strong and weak policies of time elapsing are incomparable in expressiveness.

Definition and analysis of equivalence relations in the “linear time — branching time” and “interleaving — partial order” spectra is one of the important tasks for models of concurrency and distributed systems. In [4], a hierarchy of equivalences for time-free Petri nets is presented. In [5], the authors developed and investigated the *simple bisimulation* and *forward-backward equivalences* for the TPNs with the strong time policy and intermediate clock reset strategy. In [6], the semantic models in the “interleaving — partial order” spectrum were defined in terms of weak TPNs. In [7], trace and simple bisimulation equivalences were developed for the TPNs with an intermediate clock reset policy. As far as we know, there are no equivalences in the literature for TPNs with a weak time policy and with a persistent atomic method of resetting the clock. The purpose of this work is to define and study the relationships of trace, simple bisimulation, forward-backward bisimulation and *history-preserving bisimulation* equivalences for weak TPNs with a persistent atomic clock reset policy.

Key words: time Petri nets, weak time policy, persistent atomic clock reset policy, behavioral equivalences, interleaving semantics, partial order semantics, time processes, trace and bisimulation equivalences, forward-backward bisimulation, history-preserving bisimulation.

References

1. Bérard B., Cassez F., Haddad S., Lime D., Roux O.H. Comparison of different semantics for time Petri nets // International Symposium on Automated Technology for Verification and Analysis. 2005. P. 293–307.
2. Reynier P.A., Sangnier A. Weak time Petri nets strike back! // International Conference on Concurrency Theory. 2009. P. 557–571.
3. Boyer M., Roux O.H. Comparison of the expressiveness of arc, place and transition time Petri nets // International Conference on Application and Theory of Petri Nets. 2007. P. 63–82.
4. Tarasyuk, I. V. Equivalences for behavioral analysis of concurrent and distributed computing systems // Academic Publishing House Geo, 2007.
5. Virbitskaite I., Bushin D., Best E. True concurrent equivalences in time Petri nets // Fundamenta Informaticae, 149(4), 2016. P. 401–418
6. Virbitskaite I. B., Zubarev A. Y. “True concurrency” semantics for time Petri nets with weak time and persistent atomic policies // Programming and Computer Software, 2021. N 47(5). P. 389–401
7. Zubarev A. Y. Comparison of trace and bisimulation equivalences on time Petri nets with weak time policy // Problems of Informatics, 2022. N 57(4). P. 5–27.

ИЕРАРХИЯ ЭКВИВАЛЕНТНОСТЕЙ НЕПРЕРЫВНО-ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ СО СЛАБОЙ ВРЕМЕННОЙ СТРАТЕГИЕЙ

А. Ю. Зубарев

Институт систем информатики им. А.П. Ершова,
630090, Новосибирск, Россия

УДК 519.7

DOI: 10.24412/2073-0667-2024-1-5-40

EDN: UMPUBR

Непрерывно-временные сети Петри (НВСП) — временное расширение сетей Петри, где каждому переходу ставятся в соответствие временной интервал его срабатывания и локальные часы. Данная модель рассматривается со слабой временной стратегией (ход времени не форсирует срабатывания переходов) и устойчиво атомарной стратегией сброса часов (срабатывание перехода рассматривается как единое событие). Для НВСП разрабатываются и исследуются эквивалентности в спектрах «линейное — ветвящееся время» и «интерливинг — частичный порядок». Первый спектр представлен языковыми эквивалентностями (поведение системы определяется множеством ее процессов), бисимуляционными эквивалентностями (учитываются точки выбора альтернативных действий системы): обычными, прямыми-обратными, с сохранением и слабым сохранением истории. Второй спектр определен семантиками интерливинга (процесс — последовательность действий), частичного порядка (процесс — частично-упорядоченное множество действий) и процессно-сетевой семантикой (процесс — ациклическая бесконфликтная сеть). Анализируются взаимосвязи между данными эквивалентностями, строится их иерархия.

Ключевые слова: непрерывно-временные сети Петри, слабая временная стратегия, устойчиво атомарная стратегия сброса часов, поведенческие эквивалентности, семантика интерливинга, семантика частичного порядка, процессно-сетевая семантика, языковая и бисимуляционная эквивалентности, бисимуляционная эквивалентность с сохранением истории, прямая обратная бисимуляционная эквивалентность.

Введение. Безопасность многих информационных систем является критически важной, так как их отказ может повлечь большие экономические потери и в некоторых случаях привести к угрозе жизни. Эквивалентности позволяют сравнивать системы и их спецификации относительно разных аспектов функционирования, что используется для их верификации и упрощения. В литературе модели параллельных и распределенных систем принято сравнивать в дихотомиях «линейное — ветвящееся время» и «интерливинг — частичный порядок». Первым критерием сравнения является степень учета конфликта — момента, когда возникает выбор между несколькими «ветвями» вычислений. Семантика линейного времени полностью пренебрегает точками такого выбора. Поведение здесь определяется множеством выполнений (процессов). Типичным примером соответствующих эквивалентностей являются *языковые эквивалентности*, которые сводятся к равенству *языков* (множества возможных выполнений) двух моделей. Семантика ветвистого

времени, напротив, фиксирует все точки недетерминированного выбора. Примером эквивалентности в данной семантике является *бисимуляционная эквивалентность*, которая определяется существованием отношения (бисимуляции), связывающего состояния двух моделей.

Вторым критерием сравнения является степень учета частичного порядка. *Интерливинговая семантика* игнорирует частичный порядок. Параллельное выполнение нескольких действий сводится к недетерминированному выбору между их последовательными выполнениями. С другой стороны, модели, основанные на частичном порядке, сохраняют причинно-следственную зависимость и параллелизм между действиями. Между границами спектра эквивалентностей «интерливинг — частичный порядок» находится *шаговая эквивалентность*, где параллельное выполнение действий моделируется выполнением одного шага.

Сети Петри являются одним из общепринятых инструментов для моделирования и анализа параллельных и распределенных систем. Данная модель изображается в виде ориентированного двудольного графа, который иллюстрирует причинно-следственные связи, параллелизм и конфликт между компонентами системы. События системы представлены *переходами* (прямоугольники или барьеры), а для определения входных и выходных условий событий используются *места* (окружности). Состояние сети Петри определяется *разметкой* — множеством мест с *фишками* (жирными точками). Считается, что все условия для события выполнены, если в каждом входном месте соответствующего перехода есть фишка. Данный переход называется *допустимым* и может сработать, т. е. изменить разметку. Новая разметка получается в результате удаления фишек из входных мест и создания новых фишек в выходных местах перехода.

Часто при верификации систем следует учитывать реально-временные или количественные характеристики. Для этих целей были определены разнообразные временные расширения сетей Петри. Одной из таких моделей являются *непрерывно-временные сети Петри* (НВСП), в которых каждый переход имеет собственные часы и временной интервал срабатывания. Состояние данной модели определяется разметкой и показаниями часов допустимых переходов. Переход может сработать в состоянии, если он допустимый в соответствующей разметке и показания его часов лежат в связанном с ним интервале. Само срабатывание перехода не занимает времени, поэтому в качестве изменения состояния рассматривают либо срабатывание перехода, либо ход времени.

Срабатывание перехода меняет разметку, отключает часы у недопустимых переходов и инициализирует, т. е. сбрасывает часы у некоторых допустимых переходов. Существуют *промежуточная, атомарная и устойчиво атомарная стратегии сброса часов*. Промежуточная стратегия учитывает разметку до создания новых и после удаления старых фишек (промежуточную разметку). Часы перехода таким образом будут отключены, если он был недопустимым в промежуточной разметке. Если после создания новых фишек такой переход стал допустимым, то часы примут нулевое значение. Атомарные стратегии не рассматривают промежуточных состояний, и таким образом отключение часов и их сброс произойдет только в том случае, если переход будет недопустимым в новой разметке. В случае промежуточной и атомарной стратегии, в отличие от устойчиво атомарной, часы перехода, который срабатывает, всегда сбрасываются. В работах [1] и [2] показано, что устойчиво атомарная стратегия является строго более выразительной, чем атомарная, которая, в свою очередь, не менее выразительна, чем промежуточная. Кроме того, авторы приводят системы, где использование атомарных стратегий для моделиро-

вания предпочтительнее. Например, стандартная в контексте безвременных сетей Петри модель наблюдателя — проверка с помощью цикла наличия фишки (выполнения условия) у наблюдаемого компонента. При использовании промежуточной стратегии срабатывание перехода-наблюдателя сбросит часы некоторых переходов в наблюдаемом компоненте, тогда как в атомарных стратегиях наблюдатель не будет влиять на поведение наблюдаемого компонента.

Ход времени увеличивает показания локальных часов допустимых переходов на указанное значение. В литературе представлены *сильная* и *слабая временные стратегии*. Сильная стратегия ограничивает ход времени таким образом, что значение часов перехода не может перейти за верхнюю границу временного интервала — переход обязан либо сработать, либо перестать быть допустимым в результате срабатывания другого перехода. В слабой стратегии, напротив, нет ограничений на ход времени. Как было показано в [3], сильная и слабая стратегии хода времени являются несравнимыми по выразительности.

Определение и анализ взаимосвязей эквивалентностей в спектрах «линейное — ветвящееся время» и «интерливинг — частичный порядок» является одной из важных задач для моделей параллельных и распределенных систем. В работе [4] представлена иерархия данных эквивалентностей для безвременных сетей Петри. В [5] авторами были разработаны и исследованы обычные бисимуляционные и прямые-обратные эквивалентности для НВСП с сильной временной стратегией и промежуточной стратегией сброса часов. В работе [6] были определены основные семантические модели в спектре «интерливинг — частичный порядок» в терминах слабых НВСП. Основываясь на данных моделях, в [7] были разработаны языковые и обычные бисимуляционные эквивалентности для НВСП с промежуточной стратегией сброса часов. Насколько нам известно, в литературе не представлены эквивалентности для НВСП со слабой временной стратегией и с устойчиво атомарным способом сброса часов. Целью данной работы является определение и исследование взаимосвязей языковых, обычных бисимуляционных, прямых-обратных бисимуляционных и сохраняющих историю бисимуляционных эквивалентностей для слабых НВСП с устойчиво атомарной стратегией сброса часов.

Статья состоит из следующих разделов. В разделе 1 представлены определения сети Петри и ее временного расширения. Раздел 2 рассматривает временные процессы как истинно параллельную семантику НВСП. На основе данной модели в разделе 3 определяются эквивалентности в двух спектрах «линейное — ветвящееся время» и «интерливинг — частичный порядок», строится их иерархия. Раздел 4 завершает статью, приводя результаты и планы по дальнейшей работе.

1. Непрерывно-временные сети Петри. В данном разделе рассматривается определение модели сети Петри с временными интервалами, представляющими временные задержки срабатываний переходов.

Сначала напомним синтаксис и семантику «безвременных» сетей Петри. Сеть Петри (СП) состоит из двух разных множеств элементов — мест и переходов; отношения инцидентности между данными элементами; начальной разметки — распределения фишек по местам сети; функции, помечающей каждый переход некоторым действием из алфавита Act .

Определение 1. (Помеченная над Act) сеть Петри (СП) — это набор $\mathcal{N} = (P, T, F, M_0, L)$, где P — конечное множество мест и T — конечное множество переходов такие, что $P \cap T = \emptyset$; $F \subseteq (P \times T) \cup (T \times P)$ — отношение инцидентности; $\emptyset \neq M_0 \subseteq P$

— начальная разметка; $L : T \rightarrow Act$ — помечающая функция. Для элемента $x \in P \cup T$ определим множество $\bullet x = \{y \mid (y, x) \in F\}$ входных и множество $x^\bullet = \{y \mid (x, y) \in F\}$ выходных элементов, которые для подмножества элементов $X \subseteq P \cup T$ обобщаются соответственно до множеств $\bullet X = \bigcup_{x \in X} \bullet x$ и $X^\bullet = \bigcup_{x \in X} x^\bullet$.

Разметка M СП \mathcal{N} — это любое подмножество множества P . Переход $t \in T$, допустимый в разметке M , если $\bullet t \subseteq M$. Обозначим через $En(M)$ множество всех переходов, допустимых в разметке M . Срабатывание перехода, допустимого в разметке M , приводит к новой разметке $M' = (M \setminus \bullet t) \cup t^\bullet$. В таком случае будем писать $M \xrightarrow{t} M'$. Разметка M является достижимой в \mathcal{N} , если существует последовательность переходов $t_1 \dots t_n$ такая, что $M_0 \xrightarrow{t_1} M_1 \dots M_{n-1} \xrightarrow{t_n} M_n = M$ ($n \geq 0$). СП \mathcal{N} называется бесконтактной, если для любой достижимой разметки M и любого перехода t , допустимого в M , верно, что $(M \setminus \bullet t) \cap t^\bullet = \emptyset$.

В работе рассматриваются безопасные сети Петри, т. е. при функционировании сети каждое ее место имеет не более одной фишки. Это достигается за счет определенного ранее свойства бесконтактности и позволяет рассматривать в качестве разметок множества вместо мультимножеств. Заметим, что безопасные сети Петри используются только с целью упростить изложение материала. Полученные результаты могут быть расширены на ограниченные сети Петри.

Непрерывно-временная сеть Петри (НВСП) состоит из базовой сети Петри и статической временной функции, отображающей каждый переход во временной интервал с неотрицательными рациональными границами. Подразумевается, что у каждого перехода есть собственные часы, которые отсчитывают время, прошедшее с момента, когда переход стал допустимым.

Пусть $Interv = \{[a, b], [a, b) \mid a \in \mathbb{Q}_{\geq 0}, b \in (\mathbb{Q}_{\geq 0} \cup \{\infty\}), a \leq b\}$ — множество интервалов с границами из $\mathbb{Q}_{\geq 0}$.

Определение 2. (Помеченная над Act) непрерывно-временная сеть Петри (НВСП) — это пара $\mathcal{TN} = (\mathcal{N}, D)$, где $\mathcal{N} = (P, T, F, M_0, L)$ — базовая (помеченная над Act) сеть Петри и $D : T \rightarrow Interv$ — статическая временная функция, сопоставляющая каждому переходу из T временной интервал из $Interv$.

Состояние НВСП \mathcal{TN} — это пара $S = (M, I)$, где M — разметка и $I : En(M) \rightarrow \mathbb{R}_{\geq 0}$ — динамическая временная функция. Начальное состояние \mathcal{TN} — это пара $S_0 = (M_0, I_0)$, где M_0 — начальная разметка и $I_0(t) = 0$ для всех $t \in En(M_0)$. Переход $t \in En(M)$ может сработать в состоянии, если $I(t) \in D(t)$. Обозначим через $Fi(S)$ множество всех переходов из $En(M)$, которые могут сработать в состоянии $S = (M, I)$. Для НВСП возможны два способа изменения состояния $S = (M, I)$: ход времени и действие в результате срабатывания перехода.

- Ход времени $\tau \in \mathbb{R}_{\geq 0}$ приводит к новому состоянию $S' = (M', I')$ (обозначается $S \xrightarrow{\tau} S'$), где $M' = M$ и $I'(t) = I(t) + \tau$ для всех $t \in En(M')$.
- Срабатывание перехода $t \in Fi(S)$ (действие $L(t)$) приводит к новому состоянию $S' = (M', I')$ (обозначается $S \xrightarrow{t} S'$ или $S \xrightarrow{L(t)} S'$) такому, что:
 - $M' = (M \setminus \bullet t) \cup t^\bullet$;
 - $\forall t' \in En(M') : I'(t') = \begin{cases} 0, & \text{если } t' \notin En(M) \wedge t' \in En(M'); \\ I(t'), & \text{иначе.} \end{cases}$

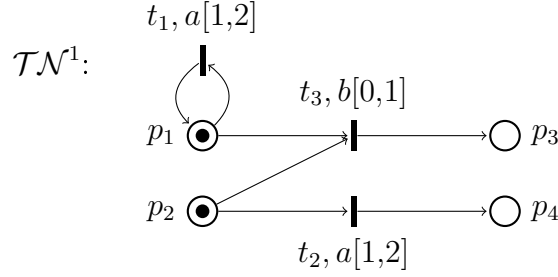


Рис. 1. НВСП \mathcal{TN}^1

Будем писать $S \xrightarrow{\sigma} S'$, если $\sigma = x_1 \dots x_m \in (T \cup \mathbb{R}_{\geq 0})^m$ и $S = S^0 \xrightarrow{x_1} S^1 \dots S^{m-1} \xrightarrow{x_m} S^m = S'$ ($m \geq 0$). В этом случае σ — пробег \mathcal{TN} из состояния S (в состояние S'). Для НВСП \mathcal{TN} обозначим через $\mathcal{RUN}(\mathcal{TN}, S)$ множество всех пробегов НВСП \mathcal{TN} из состояния S следующего вида: $\sigma = \theta_0 t_1 \theta_1 \dots t_n \theta_n$, где $\theta_i \in \mathbb{R}_{\geq 0}$ ($0 \leq i \leq n$) и $t_j \in T$ ($1 \leq j \leq n$), т. е. ход времени и срабатывания переходов чередуются. Любой пробег НВСП \mathcal{TN} может быть преобразован к пробегу из $\mathcal{RUN}(\mathcal{TN}, S)$ с сохранением последовательности срабатываний переходов, благодаря следующим свойствам хода времени:

- $S' \xrightarrow{0} S'$;
- если $S' \xrightarrow{\theta} S''$ и $S'' \xrightarrow{\theta'} S'''$ ($\theta, \theta' \in \mathbb{R}_{\geq 0}$), тогда $S' \xrightarrow{\theta+\theta'} S'''$;

которые выполняются для произвольных состояний S', S'', S''' . Для $\sigma = \theta_0 t_1 \theta_1 \dots t_n \theta_n \in \mathcal{RUN}(\mathcal{TN}, S)$ ($n \geq 0$) обозначим $L(\sigma) = \theta_0 L(t_1) \theta_1 \dots L(t_n) \theta_n$.

Пусть $\mathcal{RUN}(\mathcal{TN}) = \mathcal{RUN}(\mathcal{TN}, S_0)$. Состояние S' будем называть *достижимым* в \mathcal{TN} (обозначается $S' \in \mathcal{RS}(\mathcal{TN})$), если оно присутствует в некотором пробеге из $\mathcal{RUN}(\mathcal{TN})$.

НВСП \mathcal{TN} называется *бесконтактной*, если базовая сеть является бесконтактной. Кроме того, НВСП является *T-закрытой*, если $\bullet t \neq \emptyset$ и $t^\bullet \neq \emptyset$ для всех переходов сети. В дальнейшем будем рассматривать только бесконтактные и T-закрытые НВСП.

Пример 1. Рассмотрим НВСП \mathcal{TN}^1 , представленную на рис. 1. На графике места сети изображены окружностями, переходы изображены барьерами, отношение инцидентности представлено направленными дугами. Рядом с элементами сети указаны их имена. Каждому переходу ставятся в соответствие временной интервал из *Interv* и действие из *Act* = $\{a, b\}$. Начальной разметке соответствует множество мест с фишками (жирными точками). Покажем, что последовательность $\sigma = 2 t_1 0 t_2 3$ является пробегом из $\mathcal{RUN}(\mathcal{TN}^1)$.

- Изначально сеть находится в состоянии $S_0 = (M_0, I_0)$, где $M_0 = \{p_1, p_2\}$ — начальная разметка, $En(M_0) = \{t_1, t_2, t_3\}$ — множество допустимых переходов в данной разметке и $I(t) = 0$ для каждого перехода t из $En(M_0)$. Ход времени размером 2 сменит состояние S_0 на \tilde{S}_0 ($S_0 \xrightarrow{2} \tilde{S}_0$), в котором сохранится разметка, а часы допустимых переходов увеличатся на две единицы, т. е. $\tilde{S}_0 = (\tilde{M}_0, \tilde{I}_0) = (M_0, \tilde{I}_0)$, где $\tilde{I}_0(t) = 2$ для каждого $t \in En(\tilde{M}_0) = \{t_1, t_2, t_3\}$. Заметим, что такой ход времени был бы невозможен в случае сильной стратегии, поскольку значение времени на часах перехода t_3 ($\tilde{I}_0(t_3) = 2$) превысило верхнюю границу его временного интервала ($D(t_3) = [0, 1]$).

- Поскольку $t_1 \in En(\widetilde{M}_0)$ (допустимый в разметке \widetilde{M}_0) и $\widetilde{I}_0(t_1) \in D(t_1)$, то t_1 может сработать в состоянии \widetilde{S}_0 . После срабатывания разметка не изменится ($M_1 = \widetilde{M}_0$). Кроме того, часы переходов не будут сброшены, поскольку они были допустимыми в разметке \widetilde{M}_0 . Значит, $I_1(t) = \widetilde{I}_0(t) = 2$ для $t \in En(M_1) = \{t_1, t_2, t_3\}$. Заметим, что в случае промежуточной стратегии сброса часов часы переходов t_1 и t_3 были бы сброшены. Нулевой ход времени не изменит состояние, т.е. $S_1 = (M_1, I_1) \xrightarrow{0} \widetilde{S}_1 = (\widetilde{M}_1, \widetilde{I}_1)$, $\widetilde{M}_1 = M_1$ и $\widetilde{I}_1(t) = I_1(t) = 1$ для $t \in En(\widetilde{M}_1) = \{t_1, t_2, t_3\}$.
- Допустимый переход $t_2 \in En(\widetilde{M}_1)$ может сработать из \widetilde{S}_1 , поскольку $\widetilde{I}_1(t_2) = 2 \in D(t_2) = [1, 2]$. Следовательно, $\widetilde{S}_1 \xrightarrow{t_2} S_2 = (M_2, I_2)$, где $M_2 = (\widetilde{M}_1 \setminus \bullet t_2) \cup t_2 \bullet = \{p_1, p_4\}$, $En(M_2) = \{t_1\}$ и $I_2(t_1) = 2$, так как t_1 был допустимым в разметке \widetilde{M}_1 . Наконец, $S_2 = (M_2, I_2) \xrightarrow{3} \widetilde{S}_2$, где $\widetilde{S}_2 = (M_2, \widetilde{I}_2)$, $\widetilde{I}_2(t_1) = I_2(t_1) + 3 = 5$.

Получили, $S_0 \xrightarrow{2} \widetilde{S}_0 \xrightarrow{t_1} S_1 \xrightarrow{0} \widetilde{S}_1 \xrightarrow{t_2} S_2 \xrightarrow{3} \widetilde{S}_2$, т.е. $\sigma = 2 t_1 0 t_2 3 \in \mathcal{RUN}(\mathcal{TN}^1)$.

2. Временные процессы НВСП. Временной процесс, определяемый в данном разделе, является истинно-параллельной семантической моделью НВСП. В основе временных процессов лежат ациклическая сеть, временная функция и гомоморфизм в НВСП. Сначала определим причинно-следственную ациклическую сеть, которая образована из событий и условий, связанных отношениями причинной зависимости и параллелизма.

Определение 3. (Помеченная над Act) причинно-следственная сеть (ПСС) — это набор $N = (B, E, G, l)$, где B — конечное множество условий и E — конечное множество событий такие, что $B \cap E = \emptyset$; $G \subseteq (B \times E) \cup (E \times B)$ — отношение инцидентности такое, что $|b \bullet| \leq 1$ и $|\bullet b| \leq 1$ для всех $b \in B$ и $E = \bullet B = B \bullet$; $l : E \rightarrow Act$ — помечающая функция.

Для ПСС $N = (B, E, G, l)$ введем дополнительные понятия и обозначения.

- $x \prec x' \iff x G^+ x'$ и $x \preceq x' \iff x G^* x'$, где $x, x' \in B \cup E$ (отношение причины).
- $x \smile x' \iff \neg(x \preceq x') \wedge \neg(x' \preceq x)$, где $x, x' \in B \cup E$ (отношение параллелизма).
- Непустое подмножество $B' \subseteq B$ называется \smile -множеством в N , если $b \smile b'$ для всех $b \neq b' \in B'$.
- Максимальное по включению \smile -множество называется сечением в N . Обозначим через $\mathcal{CUT}(N)$ множество всех сечений в ПСС N .
- $\bullet N = \{b \in B \mid \bullet b = \emptyset\}$, $N \bullet = \{b \in B \mid b \bullet = \emptyset\}$ (начальное и конечное сечение соответственно);

Для $C, C' \in \mathcal{CUT}(N)$ определим:

- $C \xrightarrow{e} C'$, если существует $e \in E$ такое, что $\bullet e \subseteq C$ и $C' = (C \setminus \bullet e) \cup e \bullet$;
- $C \prec C' \iff C \xrightarrow{e^+} C'$ и $C \preceq C' \iff C \xrightarrow{e^*} C'$ (отношение причины на сечениях);
- $C \smile C' \iff \neg(C \preceq C') \wedge \neg(C' \preceq C)$ (отношение параллелизма на сечениях);
- $\downarrow C = \{e \in E \mid e \preceq (e' \in \bullet C)\}$ (множество событий, предшествующих сечению C).

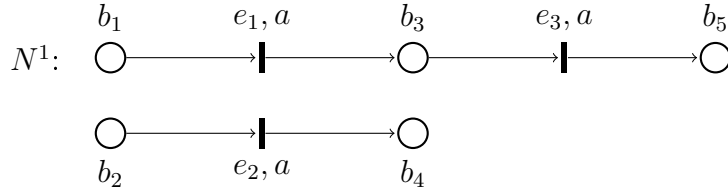


Рис. 2. ПСС N^1

Неформально говоря, сечение — это «разметка» ПСС N , которая достигается после того, как предшествующие сечению события произошли. Как следствие, $C \preceq C' \iff \downarrow C \subseteq \downarrow C'$. Для сечений $C, C' \in \mathcal{CUT}(N)$ таких, что $C \preceq C'$, определим *подсеть* $N_{C \rightsquigarrow C'} = (B', E', G', l')$ ПСС N , заключенную между данными сечениями следующим образом: $B' = \bigcup_{C \preceq \hat{C} \preceq C'} \hat{C}$; $E' = \downarrow C' \setminus \downarrow C$; $G' = G \cap ((B' \times E') \cup (E' \times B'))$; $l' = l|_{E'}$. Легко проверить, что полученная конструкция также является причинно-следственной сетью.

Пример 2. Рассмотрим сеть $N^1 = (B, E, G, l)$, где $B = \{b_1, \dots, b_5\}$, $E = \{e_1, e_2, e_3\}$, $G = \{(b_1, e_1), (b_2, e_2), (e_1, b_3), (e_2, b_4), (b_3, e_3), (e_3, b_5)\}$, $l(e_1) = l(e_2) = l(e_3) = a$, изображенную на рис. 2. Видно, что $|\bullet b| \leq 1$ и $|b \bullet| \leq 1$ для всех $b \in B$; $E = \bullet B = B \bullet$. Значит, N^1 — ПСС. Поскольку $b_1 \smile b_2$, и не существует $b \in B$ такого, что $b \smile b_1$ и $b \smile b_2$ одновременно, то $\{b_1, b_2\}$ — сечение. Аналогично можно показать, что $\{b_1, b_4\}$, $\{b_2, b_3\}$, $\{b_3, b_4\}$, $\{b_2, b_5\}$, $\{b_4, b_5\}$ также являются сечениями. Из $\bullet e_1 \subseteq \{b_1, b_2\}$, $(\{b_1, b_2\} \setminus \bullet e_1) \cup e_1 \bullet = \{b_2, b_3\}$ следует, что $\{b_1, b_2\} \xrightarrow{e_1} \{b_2, b_3\}$ и $\{b_1, b_2\} \preceq \{b_2, b_3\}$. С другой стороны, $\{b_1, b_4\} \smile \{b_2, b_3\}$, так как $\neg(\{b_1, b_4\} \preceq \{b_2, b_3\})$ и $\neg(\{b_2, b_3\} \preceq \{b_1, b_4\})$.

Для определения связи между причинно-следственной сетью и сетью Петри используется гомоморфизм, сохраняющий отношение инцидентности.

Определение 4. Пусть $N = (B, E, G, l)$ — ПСС, $\mathcal{N} = (P, T, F, M_0, L)$ — СП и M — достижимая разметка в \mathcal{N} . Гомоморфизмом из N в \mathcal{N} относительно M называется отображение $\varphi : (B \cup E) \rightarrow (P \cup T)$ такое, что верно:

- $\varphi(B) \subseteq P$ и $\varphi(E) \subseteq T$;
- сужение φ на подмножество $\bullet e$ — биекция между $\bullet e$ и $\bullet \varphi(e)$ для всех $e \in E$;
- сужение φ на подмножество $e \bullet$ — биекция между $e \bullet$ и $\varphi(e) \bullet$ для всех $e \in E$;
- сужение φ на подмножество $\bullet N$ — биекция между $\bullet N$ и M ;
- $l(e) = L(\varphi(e))$ для всех $e \in E$.

Пример 3. Рассмотрим базовую сеть Петри $\mathcal{N}^1 = (P, T, F, M_0, L)$ НВСП \mathcal{TN}^1 из примера 1, причинно-следственную сеть $N^1 = (B, E, G, l)$ из примера 2 и функцию $\varphi : (B \cup E) \rightarrow (P \cup T)$ такую, что $\varphi(b_1) = \varphi(b_3) = \varphi(b_5) = p_1$, $\varphi(b_2) = p_2$, $\varphi(b_4) = p_4$, $\varphi(e_1) = \varphi(e_3) = t_1$ и $\varphi(e_2) = t_2$. Видно, что $\varphi(B) \subseteq P$, $\varphi(E) \subseteq T$ и $l(e) = L(\varphi(e))$ для всех $e \in E$. Легко проверить, что сужение φ на подмножество $\bullet e$ ($e \bullet$) является биекцией между данным подмножеством и $\bullet \varphi(e)$ ($\varphi(e) \bullet$) для всех $e \in E$. Например, для события e_3 множество входных условий $\{b_3\}$ биективно отображается на множество входных мест $\{p_1\}$ перехода $\varphi(e_3) = t_1$, а множество выходных условий $\{b_5\}$ на множество выходных мест

$\{p_1\}$. Кроме того, сужение φ на подмножество $\bullet N^1 = \{b_1, b_2\}$ является биекцией между данным подмножеством и $M_0 = \{p_1, p_2\}$. Следовательно, φ — гомоморфизм из N^1 в \mathcal{N}^1 относительно начальной разметки M_0 .

Определим временное расширение ПСС, где каждому сечению базовой сети ставится в соответствие значение локального времени системы (длительность состояния) либо \perp , указывающий, что состояние недостижимо по времени. Данные временные характеристики, как будет показано далее, соответствуют ходу времени в состояниях НВСП.

Определение 5. (Помеченная над Act) временная ПСС (ВПСС) — это пара $TN = (N, \tau)$, где N — (помеченная над Act) ПСС и $\tau : \mathcal{CUT}(N) \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$ — временная функция такая, что для всех $C \in \mathcal{CUT}(N)$ верно:

$$\tau(C) = \perp \iff \exists C' \in \mathcal{CUT}(N) : C \smile C' \wedge \tau(C') > 0.$$

Пусть $\mathcal{RC}(TN) = \{C \in \mathcal{CUT}(N) \mid \tau(C) \in \mathbb{R}_{\geq 0}\}$ — множество временных сечений.

Заметим, что $\bullet N, N^\bullet \in \mathcal{RC}(TN)$ (если $B \neq \emptyset$) и $\downarrow N^\bullet = E$.

ВПСС $TN = (N = (B, E, G, l), \tau)$ и $TN' = (N' = (B', E', G', l'), \tau')$, помеченные над Act , являются *изоморфными* (обозначается $TN \simeq TN'$), если существует биективное отображение $\gamma : B \cup E \rightarrow B' \cup E'$ такое, что:

- $\gamma(B) = B'$ и $\gamma(E) = E'$;
- $xGy \iff \gamma(x)G'\gamma(y)$ для всех $x, y \in B \cup E$;
- $\tau(C) = \tau'(\gamma(C))$ для всех $C \in \mathcal{CUT}(N)$;
- $l(e) = l'(\gamma(e))$.

Для ВПСС $\widehat{TN} = (\widehat{N}, \widehat{\tau})$ будем писать $TN \xrightarrow{\widehat{TN}} \widehat{TN}$, если существует $\widetilde{C} \in \mathcal{RC}(\widehat{TN})$ такое, что:

- $TN = (N, \tau)$, $\widehat{TN} = (\widehat{N}, \widehat{\tau})$, $N = \widetilde{N} \bullet_{\widetilde{N} \sim \widetilde{C}}$, $\widehat{N} = \widetilde{N}_{\widetilde{C} \sim \widetilde{N}}$;
- $\tau : \mathcal{CUT}(N) \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$ и $\tau(C) = \widehat{\tau}(C)$ для всех $C \in \mathcal{CUT}(N) \setminus \widetilde{C}$;
- $\widehat{\tau} : \mathcal{CUT}(\widehat{N}) \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$ и $\widehat{\tau}(\widehat{C}) = \widetilde{\tau}(\widehat{C})$ для всех $\widehat{C} \in \mathcal{CUT}(\widehat{N}) \setminus \widetilde{C}$;
- $\tau(\widetilde{C}) + \widehat{\tau}(\widetilde{C}) = \widetilde{\tau}(\widetilde{C})$.

В этом случае сечение \widetilde{C} будем называть *граничным сечением*, а \widehat{TN} *расширением* TN на \widehat{TN} . Видно, что TN и \widehat{TN} также являются ВПСС.

Пример 4. Рассмотрим ПСС N^1 из примера 2. Определим временную функцию $\tau : \mathcal{CUT}(N^1) = \{\{b_1, b_2\}, \{b_3, b_2\}, \{b_5, b_2\}, \{b_1, b_4\}, \{b_3, b_4\}, \{b_5, b_4\}\} \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$ следующим образом: $\tau(\{b_1, b_2\}) = \tau(\{b_3, b_4\}) = 1$, $\tau(\{b_1, b_4\}) = \tau(\{b_3, b_2\}) = 0$, $\tau(\{b_5, b_4\}) = 3$, $\tau(\{b_5, b_2\}) = \perp$. Поскольку $\tau(\{b_5, b_2\}) = \perp$, то существует $\tau(\{b_3, b_4\})$ такой, что $\tau(\{b_3, b_4\}) = 1 > 0$ и $\{b_5, b_2\} \smile \{b_3, b_4\}$. Кроме того, $\{b_1, b_2\}$ и $\{b_5, b_4\}$ не имеют параллельных сечений. Значит, функция τ удовлетворяет ограничению из определения 5 и $TN^1 = (N^1, \tau)$ — ВПСС.

Пусть $TN = (N = (B, E, G, l), \tau)$ — ВПСС. Будем говорить, что событие $e \in E$ может произойти в $C \in \mathcal{RC}(TN)$ (обозначается $e \in Fi(C)$), если $C \xrightarrow{e} C'$ и $C' \in \mathcal{RC}(TN)$. Для изменения сечения $C \in \mathcal{RC}(TN)$ в результате события $e \in Fi(C)$ будем использовать обозначение $C \xrightarrow{e} C'$. Таким образом, запрещается, чтобы сечение, полученное в результате выполнения события, имело неопределенную временную характеристику (т. е. \perp), так как это означает, что время выполнения события еще не подошло. Для описания выполнения ВПСС используются графики — последовательности событий.

Определение 6. Пусть $TN = (N, \tau)$ — ВПСС и $C, C' \in \mathcal{RC}(TN)$. Последовательность событий $\omega = e_1 \dots e_n$ ($n \geq 0$) называется графиком из C в C' ВПСС TN , если существует цепочка сечений из $\mathcal{RC}(TN)$ вида:

$$(C = C_0) \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = C').$$

Пусть $\mathcal{GRF}(TN, C, C')$ — множество всех графиков ВПСС TN из C в C' . Кроме того, обозначим $\mathcal{GRF}(TN) = \mathcal{GRF}(TN, \bullet N, N \bullet)$.

Пример 5. Пусть $TN^1 = (N^1, \tau)$ — ВПСС из примера 4. Рассмотрим последовательность $\omega = e_1 e_2 e_3$. Тогда $\{b_1, b_2\} \xrightarrow{e_1} \{b_3, b_2\} \xrightarrow{e_2} \{b_3, b_4\} \xrightarrow{e_3} \{b_5, b_4\}$. Из примера 4 получаем, что $\{b_1, b_2\}, \{b_3, b_2\}, \{b_3, b_4\}, \{b_5, b_4\} \in \mathcal{RC}(TN^1)$. Кроме того, $\{b_1, b_2\} = \bullet N^1$, а $\{b_5, b_4\} = N^1 \bullet$. Значит, ω — график из $\omega \in \mathcal{GRF}(TN^1)$.

Сформулируем и докажем свойства графиков ВПСС TN .

Лемма 1. Пусть $TN = (N, \tau)$ — ВПСС и $\omega = e_1 \dots e_n \in \mathcal{GRF}(TN, C_0, C_n)$ — график с последовательностью сечений $C_0 \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} C_n$. Если $C \in (\mathcal{RC}(TN) \setminus \{C_0, \dots, C_n\})$ и $C_0 \prec C \prec C_n$, то $\tau(C) = 0$.

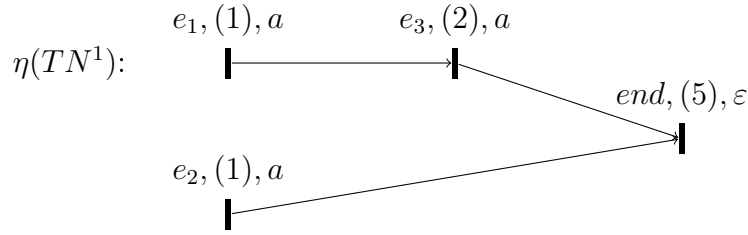
Доказательство. Рассмотрим произвольное $C \in (\mathcal{RC}(TN) \setminus \{C_0, \dots, C_n\})$ такое, что $C_0 \prec C \prec C_n$. Предположим обратное, т. е. $\tau(C) > 0$. По определению 5, $C \not\prec C_i$, т. е. либо $C \prec C_i$ ($\downarrow C \subset \downarrow C_i$), либо $C_i \prec C$ ($\downarrow C_i \subset \downarrow C$), для любого $0 \leq i \leq n$. Поскольку $C_0 \prec C \prec C_n$, т. е. $\downarrow C_0 \subset \downarrow C \subset \downarrow C_n$, то найдется $1 \leq k \leq n$ такой, что $\downarrow C_{k-1} \subset \downarrow C \subset \downarrow C_k$. Пришли к противоречию с тем, что $C_{k-1} \xrightarrow{e_k} C_k$. \square

Предложение 1. Пусть TN — ВПСС. Если $C, C' \in \mathcal{RC}(TN)$ и $C \preceq C'$, то существует график $\omega \in \mathcal{GRF}(TN, C, C')$. В частности, $\mathcal{GRF}(TN) \neq \emptyset$.

Доказательство. Аналогично доказательству предложения 1 работы [7]. \square

Введем временное расширение частично-упорядоченных множеств и рассмотрим их в контексте событий ВПСС.

Определение 7. (Помеченное над Act) временное частично-упорядоченное множество (ВЧУМ) — это набор $\eta = (X, \preceq, \tau, l)$, где X — множество элементов; \preceq — рефлексивное, транзитивное, антисимметричное отношение на X ; $\tau : X \rightarrow \mathbb{R}_{\geq 0}$ — временная функция и $l : X \rightarrow Act$ — помечающая функция.

Рис. 3. ВЧУМ $\eta(TN^1)$

ВЧУМ $\eta = (X, \preceq, \tau, l)$ и $\eta' = (X', \preceq', \tau', l')$, помеченные над Act , являются *изоморфными* (обозначается $\eta \simeq \eta'$), если существует биективное отображение $\gamma : X \rightarrow X'$ такое, что:

- $x \preceq \hat{x} \iff \gamma(x) \preceq' \gamma(\hat{x})$ для всех $x, \hat{x} \in X$;
- $\tau(x) = \tau'(\gamma(x))$ и $l(x) = l'(\gamma(x))$ для всех $x \in X$.

Для ВПСС $TN = (N = (B, E, G, l), \tau)$, графика $\omega = e_1 \dots e_n \in \mathcal{GRF}(TN)$ с последовательностью сечений $C_0 \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} C_n$ определим ВЧУМ событий $\eta(TN, \omega) = (E^*, \preceq^*, \mathbf{Age}, l^*)$ следующим образом:

- $E^* = E \cup end$;
- $\preceq^* = (\preceq \cap (E \times E)) \cup \{(e, end), (end, end) \mid e \in E\}$;
- $\mathbf{Age}(e) = \begin{cases} \sum_{0 \leq i < k} \tau(C_i), & \text{если } e = e_k \ (1 \leq k \leq n), \\ \sum_{0 \leq i \leq n} \tau(C_i), & \text{иначе;} \end{cases}$
- $l^*(e) = \begin{cases} l(e), & \text{если } e \in E, \\ \varepsilon, & \text{иначе.} \end{cases}$

Дополнительное событие end соответствует завершению работы системы и является наибольшим элементом ВЧУМ. Функция \mathbf{Age} определяет глобальное время системы для события. Благодаря лемме 1, видим, что функция \mathbf{Age} и, как следствие, ВЧУМ событий ВПСС не зависят от выбора графика, который всегда будет существовать, согласно предложению 1. Данный факт позволяет использовать для определения ВЧУМ событий ВПСС TN запись $\eta(TN)$.

Пример 6. Построим ВЧУМ событий для ВПСС TN^1 из примера 4. Сначала определим функцию \mathbf{Age} . Для этого рассмотрим график $\omega = e_1 e_2 e_3$ с последовательностью сечений $\{b_1, b_2\} \xrightarrow{e_1} \{b_3, b_2\} \xrightarrow{e_2} \{b_3, b_4\} \xrightarrow{e_3} \{b_5, b_4\}$, представленный в примере 5. Тогда $\mathbf{Age}(e_1) = \tau(\{b_1, b_2\}) = 1$, $\mathbf{Age}(e_2) = \tau(\{b_1, b_2\}) + \tau(\{b_3, b_2\}) = 1 + 0 = 1$, $\mathbf{Age}(e_3) = \tau(\{b_1, b_2\}) + \tau(\{b_3, b_2\}) + \tau(\{b_3, b_4\}) = 1 + 0 + 1 = 2$, $\mathbf{Age}(end) = \tau(\{b_1, b_2\}) + \tau(\{b_3, b_2\}) + \tau(\{b_3, b_4\}) + \tau(\{b_5, b_4\}) = 1 + 0 + 1 + 3 = 5$. Представим полученный ВЧУМ $\eta(TN^1)$ в виде ориентированного графа на рис. 3. Рядом с каждым узлом графа указано событие, значение временной и помечающей функции.

Рассмотрим функцию \mathbf{Clock} , которая будет связывать временные конструкции в рассмотренных ранее ВПСС и НВСП. Пусть $\mathcal{TN} = (\mathcal{N}, D)$ — НВСП, $TN = (N, \tau)$ — ВПСС,

$S = (M, I) \in \mathcal{RS}(\mathcal{TN})$ и φ — гомоморфизм из N в \mathcal{N} относительно M . Для сечения $C \in \mathcal{RC}(TN)$, графика $\omega = e_1 \dots e_n \in \mathcal{GRF}(TN, \bullet N, C)$, последовательности сечений $C_0 \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} (C_n = C)$ и перехода $t \in \text{En}(\varphi(C))$ определим функцию:

$$\mathbf{Clock}(\omega, t) = \begin{cases} \sum_{\max(k) < i \leq n} \tau(C_i), \text{ если } \exists k < n: t \notin \text{En}(\varphi(C_k)), \\ \sum_{0 \leq i \leq n} \tau(C_i) + I(t), \text{ иначе;} \end{cases}$$

Значение данной функций соответствует времени с момента, как переход t стал допустимым и оставался таким до конца графика. Благодаря предложению 1, для любого сечения $C \in \mathcal{RC}(TN)$ существует график $\omega \in \mathcal{GRF}(TN, \bullet N, C)$. Кроме того, по лемме 1, функция **Clock** не зависит от выбора графика, приводящего к сечению C . Значит, мы можем рассматривать в качестве параметров функции **Clock** сечение $C \in \mathcal{RC}(TN)$ и переход $t \in \text{En}(\varphi(C))$. Теперь мы готовы определить временной процесс как пару из ВПСС и гомоморфизма с ограничениями на корректность временных конструкций.

Определение 8. Пусть $\mathcal{TN} = (\mathcal{N}, D)$ — НВСП, $S = (M, I) \in \mathcal{RS}(\mathcal{TN})$, $TN = (N, \tau)$ — ВПСС и φ — гомоморфизм из N в \mathcal{N} относительно M . Пара $\pi = (TN, \varphi)$ называется временным процессом НВСП относительно состояния S , если для каждого сечения $C \in \mathcal{RC}(TN)$ и события $e \in \text{Fi}(C)$ выполняется:

$$\mathbf{Clock}(C, \varphi(e)) \in D(\varphi(e)).$$

Пусть $\mathcal{PRC}(\mathcal{TN}, S)$ — множество всех временных процессов НВСП \mathcal{TN} относительно состояния S и $\mathcal{PRC}(\mathcal{TN}) = \mathcal{PRC}(\mathcal{TN}, S_0)$. Временной процесс $\pi_0 = (((B_0, \emptyset, \emptyset, \emptyset), \tau_0 \equiv 0), \varphi_0)$, где φ_0 — гомоморфизм относительно M_0 , называется начальным.

Два временных процесса $\pi = (TN = ((B, E, G, l), \tau), \varphi)$ и $\pi' = (TN', \varphi')$ из $\mathcal{PRC}(\mathcal{TN}, S)$ являются изоморфными (обозначается $\pi \simeq \pi'$), если существует изоморфизм $\gamma: TN \simeq TN'$ такой, что $\varphi(x) = \varphi'(\gamma(x))$ для любого $x \in B \cup E$.

Пример 7. Рассмотрим НВСП $\mathcal{TN}^1 = (\mathcal{N}^1, D)$ из примера 1, ВПСС $TN^1 = (N^1, \tau)$ из примера 4 и гомоморфизм φ из N^1 в \mathcal{N}^1 относительно начальной разметки из примера 3. Покажем, что пара $\pi^1 = (TN^1, \varphi)$ является временным процессом \mathcal{TN}^1 относительно начального состояния $S_0 = (M_0, (I_0 \equiv 0))$. Для этого вычислим для каждого события e и сечения $C \in \mathcal{RC}(TN^1)$, в котором оно может произойти, значение функции **Clock**($C, \varphi(e)$) и убедимся, что оно принадлежит интервалу $D(\varphi(e))$. Напомним все временные сечения сети: $\{b_1, b_2\}, \{b_1, b_4\}, \{b_3, b_2\}, \{b_3, b_4\}, \{b_5, b_4\}$. Легко проверить, что $\{b_1, b_2\} \xrightarrow{e_1} \{b_3, b_2\} \xrightarrow{e_2} \{b_3, b_4\} \xrightarrow{e_3} \{b_5, b_4\}$ и $\{b_1, b_2\} \xrightarrow{e_2} \{b_1, b_4\} \xrightarrow{e_1} \{b_3, b_4\} \xrightarrow{e_3} \{b_5, b_4\}$. Тогда:

- $\mathbf{Clock}(\{b_1, b_2\}, \varphi(e_1)) = \tau(\{b_1, b_2\}) = 1 \in D(\varphi(e_1) = t_1) = [1, 2],$
- $\mathbf{Clock}(\{b_1, b_2\}, \varphi(e_2)) = \tau(\{b_1, b_2\}) = 1 \in D(\varphi(e_2) = t_2) = [1, 2],$
- $\mathbf{Clock}(\{b_3, b_2\}, \varphi(e_2)) = \tau(\{b_1, b_2\}) + \tau(\{b_3, b_2\}) = 1 + 0 \in D(\varphi(e_2) = t_2) = [1, 2],$
- $\mathbf{Clock}(\{b_1, b_4\}, \varphi(e_1)) = \tau(\{b_1, b_2\}) + \tau(\{b_1, b_4\}) = 1 + 0 \in D(\varphi(e_1) = t_1) = [1, 2],$

- $\mathbf{Clock}(\{b_3, b_4\}, \varphi(e_3)) = \tau(\{b_1, b_2\}) + \tau(\{b_3, b_2\}) + \tau(\{b_3, b_4\}) = 1 + 0 + 1 \in D(\varphi(e_3) = t_1) = [1, 2]$.

Следовательно, $\pi^1 = (TN^1, \varphi) \in \mathcal{PRC}(\mathcal{TN}^1)$.

Установим взаимосвязь между временными процессами и пробегами НВСП. Пусть $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}, S)$, $\omega = e_1 \dots e_n \in \mathcal{GRF}(TN)$ с последовательностью сечений $C_0 \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} C_n$ для некоторой НВСП \mathcal{TN} и состояния $S \in \mathcal{RS}(\mathcal{TN})$. Определим отображение Run временного процесса и его графика в последовательность временных задержек и переходов НВСП следующим образом: $Run(\pi, \omega) = \tau(C_0) \varphi(e_1) \tau(C_1) \dots \varphi(e_n) \tau(C_n)$. Кроме того, пусть $l(\pi, \omega) = \tau(C_0) l(e_1) \tau(C_1) \dots l(e_n) \tau(C_n)$. Обозначим $\mathcal{RUN}(\pi) = \{Run(\pi, \omega) \mid \pi = (TN, \varphi), \omega \in \mathcal{GRF}(TN)\}$.

Предложение 2. Пусть \mathcal{TN} – НВСП.

(1) Для $\pi = (TN = (N, \tau), \varphi) \in \mathcal{PRC}(\mathcal{TN}, S = (M, I))$ выполняется:

- $\mathcal{RUN}(\pi) \subseteq \mathcal{RUN}(\mathcal{TN}, S)$;
- сужение функции φ на N^\bullet – биекция между N^\bullet и M ;
- $\mathbf{Clock}(N^\bullet, t) = I(t)$ для всех $t \in En(\varphi(N^\bullet))$.

(2) Если $\sigma \in \mathcal{RUN}(\mathcal{TN}, S)$ то, существует единственный с точностью до изоморфизма $\pi^* \in \mathcal{PRC}(\mathcal{TN}, S)$ такой, что $\sigma \in \mathcal{RUN}(\pi^*)$.

Доказательство. Пункты предложения являются обобщением теорем, доказанных в работе [6], и их доказательство имеет те же рассуждения. Отличием является рассмотрение временных процессов не только относительно начального, а относительно произвольного достижимого состояния НВСП. \square

Определим расширения для временных процессов НВСП. Рассмотрим произвольный $\tilde{\pi} = (\widetilde{TN}, \tilde{\varphi}) \in \mathcal{PRC}(\mathcal{TN})$ для НВСП \mathcal{TN} . Будем писать $\pi \xrightarrow{\tilde{\pi}} \tilde{\pi}$, если $\pi = (TN, \varphi)$, $\hat{\pi} = (\widehat{TN}, \hat{\varphi})$, $TN \xrightarrow{\widehat{TN}} \widehat{TN}$, $\varphi = \tilde{\varphi}|_{B \cup E}$, $\hat{\varphi} = \tilde{\varphi}|_{\widehat{B} \cup \widehat{E}}$. В этом случае $\tilde{\pi}$ – расширение π на $\hat{\pi}$, π – префикс, а $\hat{\pi}$ – постфикс $\tilde{\pi}$. Если $\hat{\pi}$ не существенен для описания, то будем опускать его и писать $\pi \rightarrow \tilde{\pi}$.

Пусть $\pi = (TN = (N, \tau), \varphi)$ – временной процесс НВСП \mathcal{TN} и $\omega \in \mathcal{GRF}(TN)$ – график данного процесса, который существует согласно предложению 1. Согласно предложению 2(1), $Run(\pi, \omega)$ – пробег \mathcal{TN} , т.е. $S_0 \xrightarrow{Run(\pi, \omega)} S$ и $S \in \mathcal{RS}(\mathcal{TN})$. Кроме того, $S = (\varphi(N^\bullet), \mathbf{Clock}(N^\bullet))$, т.е. S не зависит от выбора графика временного процесса π . Это позволяет нам определить функцию $St : \mathcal{PRC}(\mathcal{TN}) \rightarrow \mathcal{RS}(\mathcal{TN})$ такую, что $S_0 \xrightarrow{\sigma} St(\pi)$, где $\sigma \in \mathcal{RUN}(\pi)$.

Покажем, что префикс и постфикс расширения НВСП также являются временными процессами.

Предложение 3. Если $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$, то $\pi \in \mathcal{PRC}(\mathcal{TN})$ и $\hat{\pi} \in \mathcal{PRC}(\mathcal{TN}, St(\pi))$.

Доказательство. Предложение является обобщением теорем и предложений, доказанных в работе [7]. \square

Пусть $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ для НВСП \mathcal{TN} . Рассмотрим частные случаи расширений временных процессов:

- $\tilde{\pi}$ — расширение π на время θ (обозначается $\pi \xrightarrow{\theta} \tilde{\pi}$), если $\hat{E} = \emptyset$ и $\hat{\tau}(\hat{B}) = \theta$;
- $\tilde{\pi}$ — расширение π на событие e (обозначается $\pi \xrightarrow{e} \tilde{\pi}$ или $\pi \xrightarrow{\hat{l}(e)} \tilde{\pi}$), если $\hat{E} = \{e\}$ и $\hat{\tau} \equiv 0$.

Сформулируем лемму, которая связывает ход времени и действия НВСП с соответствующими расширениями временных процессов.

Лемма 2. Пусть \mathcal{TN} — НВСП, $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})$ и $x \in \mathbb{R}_{\geq 0} \cup Act$.

(1) Если $\pi \xrightarrow{x} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$, то $St(\pi) \xrightarrow{x} St(\tilde{\pi})$.

(2) Если $St(\pi) \xrightarrow{x} \tilde{S}$, то существует $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ такой, что $\pi \xrightarrow{x} \tilde{\pi}$ и $\tilde{S} = St(\tilde{\pi})$.

Доказательство. Рассмотрим произвольный график $\omega = e_1 \dots e_n \in \mathcal{GRF}(TN)$ с последовательностью сечений $C_0 \xrightarrow{e_1} C_1 \dots C_{n-1} \xrightarrow{e_n} C_n$, который существует, согласно предложению 1. Тогда $S_0 \xrightarrow{Run(\pi, \omega)} St(\pi)$, поскольку $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})$.

(1) Рассмотрим случай $x \in Act$. Тогда $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, где $\hat{E} = \{e\}$, $\hat{l}(e) = x$ и $\hat{\tau} \equiv 0$. Поскольку $N^\bullet \xrightarrow{e} \tilde{N}^\bullet$, то $\omega e \in \mathcal{GRF}(\tilde{TN})$, по определению 6. Значит, $Run(\tilde{\pi}, \omega e) = Run(\pi, \omega) \tilde{\varphi}(e) \tilde{\tau}(\tilde{N}^\bullet) = Run(\pi, \omega) \tilde{\varphi}(e) 0 \in \mathcal{RUN}(\tilde{\pi})$, где $Run(\pi, \omega) \in \mathcal{RUN}(\pi)$ и $L(\tilde{\varphi}(e)) = x$. Следовательно, $S_0 \xrightarrow{Run(\pi, \omega)} St(\pi) \xrightarrow{x} St(\tilde{\pi})$.

Далее, рассмотрим случай $x \in \mathbb{R}_{\geq 0}$. Тогда $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, где $\hat{E} = \emptyset$ и $\hat{\tau}(\hat{B}) = x$ и $\omega \in \mathcal{GRF}(\tilde{TN})$. Кроме того, $Run(\tilde{\pi}, \omega) = \tau(C_0) \varphi(e_1) \tau(C_1) \dots \varphi(e_n) (\tau(C_n) + x) \in \mathcal{RUN}(\tilde{\pi})$, где $\tau(C_0) \varphi(e_1) \tau(C_1) \dots \varphi(e_n) \tau(C_n) = Run(\pi, \omega) \in \mathcal{RUN}(\pi)$. Значит, $S_0 \xrightarrow{Run(\pi, \omega)} St(\pi) \xrightarrow{x} St(\tilde{\pi})$.

(2) Пусть $St(\pi) \xrightarrow{x} \tilde{S}$, где $x \in Act$, случай $x \in \mathbb{R}_{\geq 0}$ рассматривается аналогично. Так как $S_0 \xrightarrow{Run(\pi, \omega)} St(\pi) \xrightarrow{x} \tilde{S}$, то существует переход t такой, что $L(t) = x$ и $S_0 \xrightarrow{Run(\pi, \omega)t0} \tilde{S}$. Поскольку $Run(\pi, \omega)t0 \in \mathcal{RUN}(\mathcal{TN})$, то найдется единственный с точностью до изоморфизма $\tilde{\pi}' \in \mathcal{PRC}(\mathcal{TN})$ такой, что $Run(\pi, \omega)t0 \in \mathcal{RUN}(\tilde{\pi}')$, по предложению 2(2), т.е. $St(\tilde{\pi}') = \tilde{S}$. Кроме того, согласно конструкции расширения и предложения 3, найдется $\pi' \in \mathcal{PRC}(\mathcal{TN})$ такой, что $\pi' \xrightarrow{x} \tilde{\pi}'$ и $Run(\pi, \omega) \in \mathcal{RUN}(\pi')$. По предложению 2(2), $\pi' \simeq \pi$, так как $Run(\pi, \omega) \in \mathcal{RUN}(\pi') \cap \mathcal{RUN}(\pi)$. Как следствие, существует временной процесс $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ такой, что $\tilde{\pi} \simeq \tilde{\pi}'$, $\pi \xrightarrow{x} \tilde{\pi}$ и $St(\tilde{\pi}) = St(\tilde{\pi}') = \tilde{S}$. \square

3. Эквивалентности НВСП. В данном разделе определяются языковые и бисимуляционные эквивалентности НВСП со слабой временной стратегией.

3.1. *Языковые эквивалентности.* Языковые эквивалентности, также известные как следовые эквивалентности, используются для сравнения поведения систем в терминах их языков, то есть наборов всех возможных выполнений систем. Данный «линейно-временной» подход не учитывает момент недетерминированного выбора между несколькими расширениями процесса. Обобщим классическое определение языковой эквивалентности, основанное на пробегах сетей Петри.

Определение 9. Пусть \mathcal{TN} , \mathcal{TN}' – НВСП, помеченные над Act .

- $Lang(\mathcal{TN}) = \{L(\sigma) \mid \sigma \in \mathcal{RUN}(\mathcal{TN})\}$.
- \mathcal{TN} и \mathcal{TN}' языково эквивалентны (обозначается $\mathcal{TN} \cong \mathcal{TN}'$), если $Lang(\mathcal{TN}) = Lang(\mathcal{TN}')$.

Основываясь на классах изоморфизма ВПСС и ВЧУМ временных процессов НВСП, сформулируем понятия языковых эквивалентностей для семантик частично порядка.

Определение 10. Пусть \mathcal{TN} и \mathcal{TN}' – НВСП, помеченные над Act , и $\star \in \{int, por, prc\}$.

- $Trace_{int}(\mathcal{TN}) = \{l(\pi, \omega) \mid \pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}), \omega \in \mathcal{GRF}(TN)\}$;
- $Trace_{por}(\mathcal{TN}) = \{[\eta(TN)]_{\simeq} \mid (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})\}$;
- $Trace_{prc}(\mathcal{TN}) = \{[TN]_{\simeq} \mid (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})\}$.
- \mathcal{TN} и \mathcal{TN}' \star -языково эквивалентны (обозначается $\mathcal{TN} \equiv_{\star} \mathcal{TN}'$), если $Trace_{\star}(\mathcal{TN}) = Trace_{\star}(\mathcal{TN}')$.

Проверим корректность предложенного определения. Докажем, что язык, основанный на пробегах НВСП, совпадает с интерливинговым (*int*) языком, основанным на временных процессах.

Теорема 1. Для НВСП \mathcal{TN} , \mathcal{TN}' выполняется:

$$\mathcal{TN} \cong \mathcal{TN}' \iff \mathcal{TN} \equiv_{int} \mathcal{TN}'.$$

Доказательство. Следуя определениям 9 и 10, достаточно показать, что для произвольной НВСП \mathcal{TN} выполняется $Lang(\mathcal{TN}) = Trace_{int}(\mathcal{TN})$.

- Поскольку $\pi_0 = (TN_0, \varphi_0) \in \mathcal{PRC}(\mathcal{TN})$ и $0 \in \mathcal{GRF}(TN_0)$, то $Trace_{int}(\mathcal{TN}) \neq \emptyset$. Рассмотрим произвольное слово $l(\pi, \omega) \in Trace_{int}(\mathcal{TN})$, что означает $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})$ и $\omega \in \mathcal{GRF}(TN)$. Тогда $Run(\pi, \omega) \in \mathcal{RUN}(\pi) \subseteq \mathcal{RUN}(\mathcal{TN})$, по предложению 2(1). Следовательно, так как $l(\pi, \omega) = L(Run(\pi, \omega))$, по определению 4, то $Trace_{int}(\mathcal{TN}) \subseteq Lang(\mathcal{TN})$.
- Так как $r \in \mathcal{RUN}(\mathcal{TN})$ для любого $r \in \mathbb{R}_{\geq 0}$ и $r \in Lang(\mathcal{TN})$, то $Lang(\mathcal{TN}) \neq \emptyset$. Рассмотрим произвольное $L(\sigma) \in Lang(\mathcal{TN})$, где $\sigma \in \mathcal{RUN}(\mathcal{TN})$. Тогда существует единственный с точностью до изоморфизма $\pi = (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN})$ такой, что $\sigma \in \mathcal{RUN}(\pi)$, по предложению 2(2). Это означает, что $Run(\pi, \omega) = \sigma$ для некоторого $\omega \in \mathcal{GRF}(TN)$. Следовательно, по определению 4, $L(\sigma) = L(Run(\pi, \omega)) = l(\omega)$ и $Lang(\mathcal{TN}) \subseteq Trace_{int}(\mathcal{TN})$.

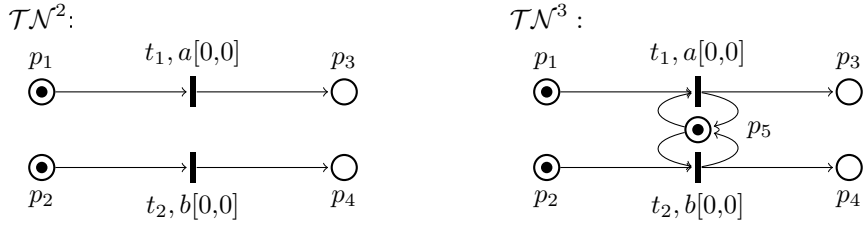


Рис. 4. $\sim \not\equiv_{por} \equiv_{por}$

Поскольку $Trace_{int}(\mathcal{TN}) \subseteq Lang(\mathcal{TN})$ и $Lang(\mathcal{TN}) \subseteq Trace_{int}(\mathcal{TN})$, то $Lang(\mathcal{TN}) = Trace_{int}(\mathcal{TN})$. \square

3.2. *Обычные бисимуляционные эквивалентности.* Бисимуляционные эквивалентности, в отличие от языковых, учитывают момент недетерминированного выбора между несколькими расширениями процесса (ветвления). В связи с этим такие эквивалентности также называют эквивалентностями ветвящегося времени. Бисимуляционно эквивалентные системы требуют существования отношения (бисимуляции). Данное отношение связывает начальные состояния двух систем. Кроме того, для любых связанных состояний при возможном переходе одной из систем в новое состояние, другая система должна иметь возможность «скопировать» поведение так, что полученные состояния снова будут связаны отношением бисимуляции. Рассмотрим обычные бисимуляционные эквивалентности, которые являются самыми «слабыми» эквивалентностями, учитывающими ветвящуюся структуру поведения в НВСП. Начнем с расширения классических определений, основанных на состояниях сетей Петри.

Определение 11. НВСП \mathcal{TN} и \mathcal{TN}' , помеченные над Act , являются бисимуляционно эквивалентными (обозначается $\mathcal{TN} \sim \mathcal{TN}'$), если существует отображение (бисимуляция) $R \subseteq \mathcal{RS}(\mathcal{TN}) \times \mathcal{RS}(\mathcal{TN}')$ такое, что $(S_0, S'_0) \in R$ и для всех $(S, S') \in R$ выполняется:

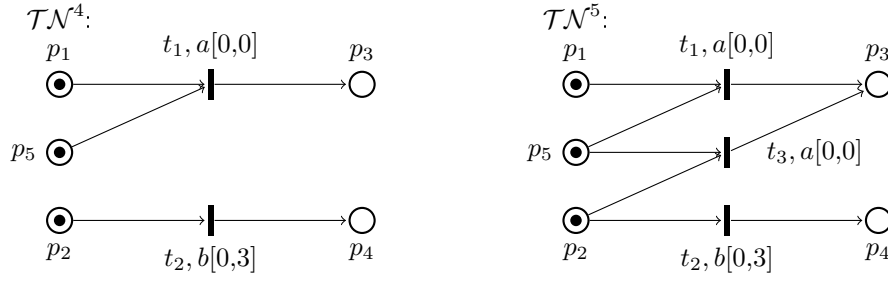
- (1) если $S \xrightarrow{x} \tilde{S}$, где $x \in Act \cup \mathbb{R}_{\geq 0}$, тогда существует пара $(\tilde{S}, \tilde{S}') \in R$ такая, что $S' \xrightarrow{x} \tilde{S}'$;
- (2) как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Пример 8. Рассмотрим НВСП \mathcal{TN}^2 и \mathcal{TN}^3 на рис. 4. Согласно определению 11, $\mathcal{TN}^2 \sim \mathcal{TN}^3$, поскольку существует возможность построения соответствующей бисимуляции, которая связывает состояния двух НВСП.

Покажем, что $\mathcal{TN}^2 \not\equiv_{por} \mathcal{TN}^3$. Действительно, у \mathcal{TN}^2 найдется временной процесс, где события действий a и b будут параллельными, тогда как у \mathcal{TN}^3 подобного процесса не существует. Значит, $Trace_{por}(\mathcal{TN}^2) \neq Trace_{por}(\mathcal{TN}^3)$ и, как следствие, $\mathcal{TN}^2 \not\equiv_{por} \mathcal{TN}^3$, согласно определению 10.

Теперь определим бисимуляционные эквивалентности в терминах временных процессов НВСП.

Определение 12. Пусть $\star \in \{int, por, prc\}$. НВСП \mathcal{TN} и \mathcal{TN}' , помеченные над Act , являются \star -бисимуляционно эквивалентными (обозначается $\mathcal{TN} \stackrel{\star}{\equiv} \mathcal{TN}'$), если существу-

Рис. 5. $\equiv_{prc} \not\equiv_{int}$

ет отображение (бисимуляция) $R \subseteq \mathcal{PRC}(\mathcal{TN}) \times \mathcal{PRC}(\mathcal{TN}')$ такое, что $(\pi_0, \pi'_0) \in R$, и для всех $(\pi, \pi') \in R$ выполняется:

(1) если $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ и

– $\tilde{\pi}$ – расширение π на время или событие, в случае $\star = int$,

тогда существует пара $(\tilde{\pi}, \tilde{\pi}') \in R$ такая, что $\pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ и

– $\eta(\widehat{\mathcal{TN}}) \simeq \eta(\widehat{\mathcal{TN}'})$, в случае $\star \in \{int, por\}$;

– $\widehat{\mathcal{TN}} \simeq \widehat{\mathcal{TN}'}$, в случае $\star = prc$;

(2) как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Пример 9. Рассмотрим НВСП \mathcal{TN}^4 и \mathcal{TN}^5 на рис. 5. Как можно видеть, $\mathcal{TN}^4 \equiv_{prc} \mathcal{TN}^5$, согласно определению 10.

Покажем, что $\mathcal{TN}^4 \not\equiv_{int} \mathcal{TN}^5$. Предположим, это не так, т.е. существует некоторая бисимуляция R , соответствующая определению 12. Тогда $(\pi_0, \pi'_0) \in R$, согласно данному определению.

Рассмотрим временной процесс π'_1 НВСП \mathcal{TN}^5 , который соответствует срабатыванию перехода t_3 и является расширением π'_0 на событие a , т.е. $\pi'_0 \xrightarrow{a} \pi'_1$. Согласно определению 12, для π_0 должен существовать аналогичный временной процесс π_1 – расширение π_0 на событие a и $(\pi_1, \pi'_1) \in R$. Это означает, что π_1 соответствует срабатыванию перехода t_1 НВСП \mathcal{TN}^4 .

Процесс π_1 может быть расширен до процесса π_2 в результате действия b (срабатывание перехода t_2 НВСП \mathcal{TN}^4), т.е. $\pi_1 \xrightarrow{b} \pi_2$. Значит, по определению 12, должна существовать пара $(\pi_2, \pi'_2) \in R$ такая, что $\pi'_1 \xrightarrow{b} \pi'_2$. Однако, процесс π'_1 не может иметь расширений в результате действия b . Полученное противоречие доказывает, что $\mathcal{TN}^4 \not\equiv_{int} \mathcal{TN}^5$.

Покажем, что предложенное определение интерливинговой (*int*) бисимуляции корректно относительно классического определения.

Теорема 2. Для НВСП \mathcal{TN} , \mathcal{TN}' выполняется:

$$\mathcal{TN} \sim \mathcal{TN}' \iff \mathcal{TN} \equiv_{int} \mathcal{TN}'.$$

Доказательство.

\Rightarrow Пусть $\mathcal{TN} \sim \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \stackrel{\text{int}}{\simeq} \mathcal{TN}'$ с бисимуляцией $R_2 = \{(\pi, \pi') \in \mathcal{PRC}(\mathcal{TN}) \times \mathcal{PRC}(\mathcal{TN}') \mid (St(\pi), St(\pi')) \in R_1\}$.

По определению 11, $(St(\pi_0), St(\pi'_0)) = (S_0, S'_0) \in R_1$. Значит $(\pi_0, \pi'_0) \in R_2$.

Рассмотрим произвольную пару $(\pi, \pi') \in R_2$. Тогда $(St(\pi), St(\pi')) \in R_1$.

(1) Пусть $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ и $\tilde{\pi}$ — расширение π на событие, случай расширения на время рассматривается аналогично. Значит, $\pi \xrightarrow{a} \tilde{\pi}$ для $a \in Act$. Согласно лемме 2(1), получаем, что $St(\pi) \xrightarrow{a} St(\tilde{\pi})$. Поскольку $(St(\pi), St(\pi')) \in R_1$, то $St(\pi') \xrightarrow{a} S'$ и $(St(\tilde{\pi}), S') \in R_1$, по определению 11. Благодаря лемме 2(2), существует $\tilde{\pi}' \in \mathcal{PRC}(\mathcal{TN}')$ такой, что $\pi' \xrightarrow{a} \tilde{\pi}'$ и $S' = St(\tilde{\pi}')$. Значит, $(St(\tilde{\pi}), St(\tilde{\pi}')) \in R_1$, а $(\tilde{\pi}, \tilde{\pi}') \in R_2$. Следовательно, существует $(\tilde{\pi}, \tilde{\pi}') \in R_2$ такая, что $\pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ и $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, так как $\tilde{\pi}'$ как и $\tilde{\pi}$ является расширением на событие действия a .

(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

По определению 12 и пунктам (1–2), $R_2 : \mathcal{TN} \stackrel{\text{int}}{\simeq} \mathcal{TN}'$.

\Leftarrow Пусть $\mathcal{TN} \stackrel{\text{int}}{\simeq} \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \sim \mathcal{TN}'$ с бисимуляцией $R_2 = \{(St(\pi), St(\pi')) \mid (\pi, \pi') \in R_1\}$.

По определению 12, $(\pi_0, \pi'_0) \in R_1$. Значит, $(St(\pi_0), St(\pi'_0)) = (S_0, S'_0) \in R_2$.

Рассмотрим произвольную пару $(St(\pi), St(\pi')) \in R_2$. Тогда $(\pi, \pi') \in R_1$.

(1) Пусть $St(\pi) \xrightarrow{x} \tilde{S}$ для $x \in Act$, случай $x \in \mathbb{R}_{\geq 0}$ рассматривается аналогично. По лемме 2(2), существует $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$ такой, что $\pi \xrightarrow{x} \tilde{\pi}$ и $\tilde{S} = St(\tilde{\pi})$. Значит, согласно определению 12, существует пара $(\tilde{\pi}, \tilde{\pi}') \in R$ такая, что $\pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ и $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, т. е. $\pi' \xrightarrow{x} \tilde{\pi}'$. Следовательно, существует пара $(St(\tilde{\pi}) = \tilde{S}, St(\tilde{\pi}')) \in R_2$ такая, что $St(\pi') \xrightarrow{x} St(\tilde{\pi}')$, по лемме 2(1).

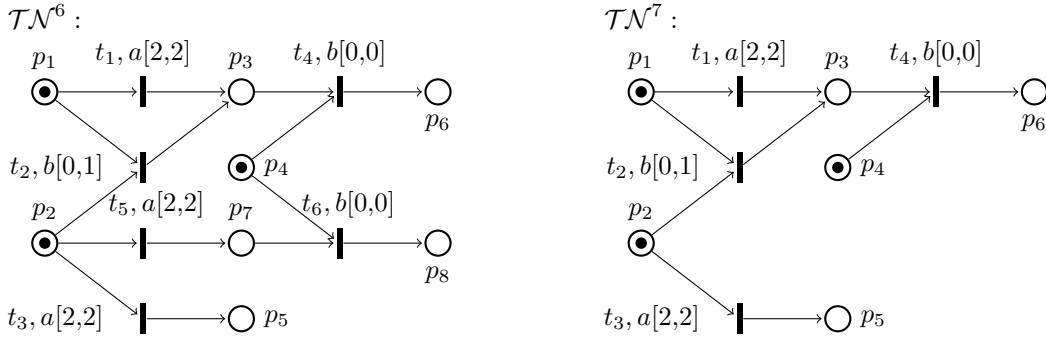
(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

По определению 11 и пунктам (1–2), $R_2 : \mathcal{TN} \sim \mathcal{TN}'$. □

3.3. *Сохраняющие историю эквивалентности.* Сохраняющие историю бисимуляционные эквивалентности учитывают предыдущее функционирование системы, ту часть процесса, которая привела из начального состояния в текущее. Соответствующие бисимуляции сохраняют изоморфизм между ВПСС (ВЧУМ) связанных временных процессов целиком, а не только добавленной части, как в случае обычных бисимуляций. Сначала рассмотрим слабый вариант данных эквивалентностей.

Определение 13. Пусть $\star \in \{por, prc\}$. НВСП \mathcal{TN} и \mathcal{TN}' , помеченные над Act , являются \star -бисимуляционно эквивалентными со слабым сохранением истории (обозначается $\mathcal{TN} \stackrel{wh}{\simeq}_{\star} \mathcal{TN}'$), если существует отображение (бисимуляция) $R \subseteq \mathcal{PRC}(\mathcal{TN}) \times \mathcal{PRC}(\mathcal{TN}')$ такое, что $(\pi_0, \pi'_0) \in R$ и для всех $(\pi, \pi') \in R$ выполняется:

(1) — $\eta(TN) \simeq \eta(TN')$, в случае $\star = por$;

Рис. 6. $\Leftrightarrow_{prc}^{wh} \not\Leftrightarrow_{por}$

– $TN \simeq TN'$, в случае $\star = prc$;

(2) если $\pi \rightarrow \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(TN)$, тогда существует пара $(\tilde{\pi}, \tilde{\pi}') \in R$ такая, что $\pi' \rightarrow \tilde{\pi}'$;

(3) как пункт (2), но роли TN и TN' меняются.

Пример 10. Рассмотрим НВСП \mathcal{TN}^6 и \mathcal{TN}^7 на рис. 6. Покажем, что $\mathcal{TN}^6 \not\stackrel{por}{\Leftrightarrow} \mathcal{TN}^7$. Предположим, это не так, т.е. существует некоторая бисимуляция R , удовлетворяющая определению 12. Тогда $(\pi_0, \pi'_0) \in R$. Временной процесс π_0 может быть расширен до процесса π_1 в результате хода двух тактов времени и события, соответствующего срабатыванию перехода t_5 . Значит, по определению 12, существует $(\pi_1, \pi'_1) \in R$, $\pi'_0 \rightarrow \pi'_1$ и $\eta(TN_1) \simeq \eta(TN'_1)$. Другими словами, временному процессу π'_1 соответствует ход двух тактов времени и последующее действие a . Поскольку π_1 может быть расширен действием b , то π'_1 должен иметь схожее поведение. Значит, процессу π'_1 будет соответствовать срабатывание перехода t_1 . С другой стороны, процесс π_1 может быть расширен следующими по причине действиями a и b , что невозможно для процесса π'_1 . Следовательно, $\mathcal{TN}^6 \not\stackrel{por}{\Leftrightarrow} \mathcal{TN}^7$.

Видно, что $\mathcal{TN}^6 \stackrel{wh}{\Leftrightarrow}_{prc} \mathcal{TN}^7$, где для существования бисимуляции требуется изоморфизм ВЧУМ не расширяющих частей, а ВЧУМ целых расширений.

Заметим, что в случае сильной стратегии хода времени НВСП будут вести себя одинаково, поскольку переход t_2 будет обязан сработать раньше переходов, соответствующих действиям a в данных НВСП.

Теперь рассмотрим сохраняющую историю бисимуляцию, где имеет место «расширение» изоморфизмов ВПСС (ВЧУМ) при расширении временных процессов. Для произвольных НВСП \mathcal{TN} и \mathcal{TN}' определим множества изоморфизмов ВПСС (ВЧУМ) их временных процессов следующим образом: $\mathcal{F}_{por}(\mathcal{TN}, \mathcal{TN}') = \{f : \eta(TN) \simeq \eta(TN') \mid (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}), (TN', \varphi') \in \mathcal{PRC}(\mathcal{TN}')\}$ и $\mathcal{F}_{prc}(\mathcal{TN}, \mathcal{TN}') = \{f : TN \simeq TN' \mid (TN, \varphi) \in \mathcal{PRC}(\mathcal{TN}), (TN', \varphi') \in \mathcal{PRC}(\mathcal{TN}')\}$. Через f_0 будем обозначать функцию с пустой областью определения, в случае $f_0 \in \mathcal{F}_{por}$, и изоморфизм между ВПСС начальных процессов, в случае $f_0 \in \mathcal{F}_{prc}$.

Определение 14. Пусть $\star \in \{por, prc\}$. НВСП \mathcal{TN} и \mathcal{TN}' , помеченные над Act , являются \star -бисимуляционно эквивалентными с сохранением истории (обозначается $\mathcal{TN} \stackrel{h}{\Leftrightarrow}_{\star} \mathcal{TN}'$), если существует отображение (бисимуляция) $R \subseteq \mathcal{PRC}(\mathcal{TN}) \times$

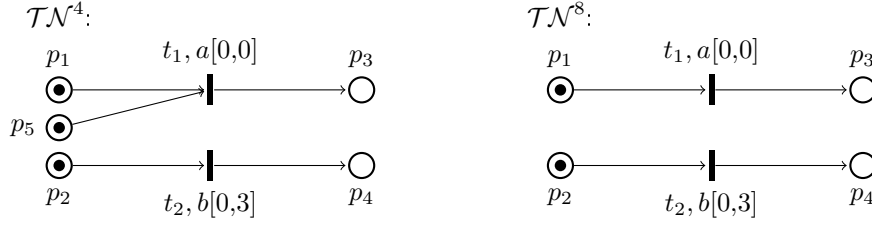


Рис. 7. $\stackrel{h}{\cong}_{por} \not\equiv_{prc}$

$PRC(\mathcal{TN}') \times \mathcal{F}_*(\mathcal{TN}, \mathcal{TN}')$ такое, что $(\pi_0, \pi'_0, f_0) \in R$ и для всех $(\pi, \pi', f) \in R$ выполняется:

- (1) – $f : \eta(\mathcal{TN}) \simeq \eta(\mathcal{TN}')$, в случае $\star = por$;
 – $f : \mathcal{TN} \simeq \mathcal{TN}'$, в случае $\star = prc$;
- (2) если $\pi \rightarrow \tilde{\pi}$ для $\tilde{\pi} \in PRC(\mathcal{TN})$, тогда существует тройка $(\tilde{\pi}, \tilde{\pi}', \tilde{f}) \in R$ такая, что $\pi' \rightarrow \tilde{\pi}'$ и $f \subseteq \tilde{f}$;
- (3) как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Заметим, что расширение на произвольный процесс в определениях 13 и 14 можно заменить на последовательность расширений на время и события. Следовательно, семантика интерливинга (*int*) будет совпадать с семантикой частичного порядка (*por*) в контексте приведенных выше бисимуляций.

Пример 11. На рис. 7 представлены НВСП \mathcal{TN}^4 и \mathcal{TN}^8 . Данные сети являются $\stackrel{h}{\cong}_{por}$ эквивалентными, но в контексте процессной семантики они не эквивалентны даже для языкового случая, т.е. $\mathcal{TN}^4 \not\equiv_{prc} \mathcal{TN}^8$. Действительно, ПСС всех временных процессов НВСП \mathcal{TN}^4 имеют три условия в начальных сечениях, в то время как начальные сечения ПСС для временных процессов \mathcal{TN}^8 содержат по два условия.

3.4. *Обратные-прямые эквивалентности.* Эквивалентности, рассматриваемые далее, позволяют учитывать моделирование как в прямом (обычном), так и в обратном направлениях. Обратные-прямые бисимуляционные эквивалентности позволяют естественным образом сочетать семантики интерливинга и частичного порядка. При обратном моделировании происходит следование строго по истории функционирования системы. Вследствие этого бисимуляции связывают «хронологии» выполнений, которые определяются процессом и его графиком. Определим множество *хронологий* следующим образом: $\mathcal{CHR}(\mathcal{TN}, S) = \{(\pi, \omega) \mid \pi = (TN, \varphi) \in PRC(\mathcal{TN}, S), \omega \in \mathcal{GRF}(TN)\}$, где \mathcal{TN} – НВСП и $S \in \mathcal{RS}(\mathcal{TN})$. Пусть $\mathcal{CHR}(\mathcal{TN}) = \mathcal{CHR}(\mathcal{TN}, S_0)$. Хронология (π_0, ε) называется *начальной*, где π_0 – начальный процесс, а ε – пустая последовательность.

Пусть $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$. Будем писать $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$, если $\pi \xrightarrow{\tilde{\pi}} \tilde{\pi}$ и $\tilde{\omega} = \omega \hat{\omega}$, $\omega \in \mathcal{GRF}(TN)$. В этом случае $(\tilde{\pi}, \tilde{\omega})$ – *расширение* (π, ω) на $(\hat{\pi}, \hat{\omega})$, (π, ω) – *префикс*, а $(\hat{\pi}, \hat{\omega})$ – *постфикс* $(\tilde{\pi}, \tilde{\omega})$. Заметим, что, благодаря предложению 3, $(\pi, \omega) \in \mathcal{CHR}(\mathcal{TN})$, а $(\hat{\pi}, \hat{\omega}) \in \mathcal{CHR}(\mathcal{TN}, St(\pi))$. Рассмотрим частные случаи расширений хронологий.

- $(\tilde{\pi}, \tilde{\omega})$ – *расширение* (π, ω) на время θ (обозначается $(\pi, \omega) \xrightarrow{\theta} (\tilde{\pi}, \tilde{\omega})$), если $\tilde{\pi}$ – расширение π на время θ . В этом случае $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \varepsilon)} (\tilde{\pi}, \omega)$.

- $(\tilde{\pi}, \tilde{\omega})$ — расширение (π, ω) на событие e (обозначается $(\pi, \omega) \xrightarrow{e} (\tilde{\pi}, \tilde{\omega})$ или $(\pi, \omega) \xrightarrow{\hat{l}(e)} (\tilde{\pi}, \tilde{\omega})$), если $\tilde{\pi}$ — расширение π на событие e . В этом случае $(\pi, \omega) \xrightarrow{(\tilde{\pi}, e)} (\tilde{\pi}, \omega e)$.

Из определений следует, что любое расширение хронологии может быть представлено в виде последовательности расширений на время и одиночные события. Сформулируем данный факт в виде леммы.

Лемма 3. Пусть $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$. Тогда следующие пункты равносильны:

- (1) $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$, $l(\hat{\pi}, \hat{\omega}) = \theta_0 a_1 \theta_1 \dots a_n \theta_n$ и $n \geq 0$.
- (2) $(\pi, \omega) = (\pi^0, \omega^0) \xrightarrow{\theta_0} (\tilde{\pi}^0, \omega^0) \xrightarrow{a_1} (\pi^1, \omega^1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi^n, \omega^n) \xrightarrow{\theta_n} (\tilde{\pi}^n, \omega^n) = (\tilde{\pi}, \tilde{\omega})$, где $\theta_i \in \mathbb{R}_{\geq 0}$ ($0 \leq i \leq n$), $a_j \in Act$ ($1 \leq j \leq n$) и $n \geq 0$.

Определение 15. Пусть $\mathcal{TN}, \mathcal{TN}'$ — НВСП, помеченные над Act , $\star, * \in \{int, por, prc\}$.

- \mathcal{TN} и \mathcal{TN}' являются \star -прямо бисимуляционно эквивалентными (обозначается $\mathcal{TN} \approx_{\star} \mathcal{TN}'$), если существует отображение (бисимуляция) $R \subseteq \mathcal{CHR}(\mathcal{TN}) \times \mathcal{CHR}(\mathcal{TN}')$ такое, что $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$ и для всех $((\pi, \omega), (\pi', \omega')) \in R$ выполняется:

- (1) если $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$ и
 - $(\tilde{\pi}, \tilde{\omega})$ — расширение (π, ω) на время или событие, в случае $\star = int$,
 тогда существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ такая, что $(\pi', \omega') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\tilde{\pi}', \tilde{\omega}')$ и
 - $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, в случае $\star \in \{int, por\}$;
 - $\widehat{TN} \simeq \widehat{TN}'$, в случае $\star = prc$;

(2) как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

- \mathcal{TN} и \mathcal{TN}' являются $*$ -обратно \star -прямо бисимуляционно эквивалентными (обозначается $\mathcal{TN} \approx_{*\star} \mathcal{TN}'$), если $\mathcal{TN} \approx_{\star} \mathcal{TN}'$ с бисимуляцией R такой, что для всех $((\pi, \omega), (\pi', \omega')) \in R$ выполняется:

- (3) если $(\tilde{\pi}, \tilde{\omega}) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\pi, \omega)$ и
 - (π, ω) — расширение $(\tilde{\pi}, \tilde{\omega})$ на время или событие, в случае $*$ = int,
 тогда существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ такая, что $(\tilde{\pi}', \tilde{\omega}') \xrightarrow{(\hat{\pi}', \hat{\omega}')} (\pi', \omega')$ и
 - $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, в случае $*$ $\in \{int, por\}$;
 - $\widehat{TN} \simeq \widehat{TN}'$, в случае $*$ = prc;

(4) как пункт (3), но роли \mathcal{TN} и \mathcal{TN}' меняются.

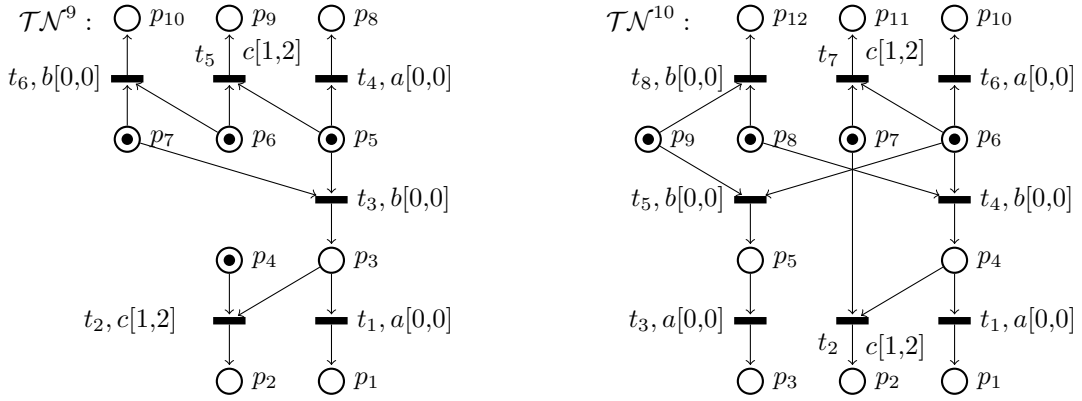


Рис. 8. $\Leftrightarrow_{prc} \not\approx \approx_{int\ por}$

Пример 12. Рассмотрим НВСП \mathcal{TN}^9 и \mathcal{TN}^{10} на рис. 8. Покажем, что $\mathcal{TN}^9 \not\approx_{int\ por} \mathcal{TN}^{10}$. Предположим, что это не так и $\mathcal{TN}^9 \approx_{int\ por} \mathcal{TN}^{10}$, т.е. существует бисимуляция R , удовлетворяющая определению 15. Тогда $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$.

Рассмотрим расширение π_1 временного процесса π_0 , которое соответствует срабатыванию параллельных переходов t_6 и t_4 , и соответствующий график ω_1 , определяющий последовательность действий b a . Значит, согласно определению 15, существует пара $((\pi_1, \omega_1), (\pi'_1, \omega'_1)) \in R$, где π'_1 — расширение π'_0 , которому соответствуют параллельные действия b и a . Данное расширение может соответствовать только срабатыванию переходов t_8 и t_6 НВСП \mathcal{TN}^{10} .

Далее, π_1 и π'_1 могут сделать ход назад по истории на действие a , т.е. $\pi_2 \rightarrow \pi_1$, $\pi'_2 \rightarrow \pi'_1$ и $((\pi_2, \omega_2), (\pi'_2, \omega'_2)) \in R$.

Временной процесс π'_2 после задержки в один такт времени может быть расширен действием c до процесса π'_3 , что невозможно для временного процесса π_2 . Следовательно, не существует процесса π_3 , и, как следствие, пары $((\pi_3, \omega_3), (\pi'_3, \omega'_3)) \in R$, удовлетворяющей определению 15. Таким образом, получили противоречие с данным определением. Значит, $\mathcal{TN}^9 \not\approx_{int\ por} \mathcal{TN}^{10}$.

С другой стороны, можно проверить, что для данных НВСП имеет место обычная бисимуляционная эквивалентность в процессной семантике, т.е. $\mathcal{TN}^9 \Leftrightarrow_{prc} \mathcal{TN}^{10}$.

Докажем вспомогательный факт для обратной-прямой бисимуляции.

Лемма 4. Пусть $\mathcal{TN}, \mathcal{TN}'$ — НВСП, помеченные над Act , $\star, * \in \{por, prc\}$. Если $R : \mathcal{TN} \approx_{\star} \mathcal{TN}'$ и $((\pi_0, \varepsilon), (\pi', \omega')) \in R$, то $(\pi', \omega') = (\pi'_0, \varepsilon)$.

Доказательство. Рассмотрим случай $\star = * = por$, остальные случаи рассматриваются аналогично. Так как $(\pi'_0, \varepsilon) \xrightarrow{(\pi', \omega')} (\pi', \omega')$ и $((\pi_0, \varepsilon), (\pi', \omega')) \in R$, то существует пара $((\pi, \omega), (\pi'_0, \varepsilon)) \in R$ такая, что $(\pi, \omega) \xrightarrow{(\widehat{\pi}, \widehat{\omega})} (\pi_0, \varepsilon)$ и $\eta(\widehat{TN}) \simeq \eta(TN')$, согласно определению 15(3). Поскольку π_0 не имеет событий, а единственное его сечение имеет нулевое временное значение, то $(\pi, \omega) = (\widehat{\pi}, \widehat{\omega}) = (\pi_0, \varepsilon)$, т.е. $\eta(TN_0) \simeq \eta(TN')$. Из изоморфизма следует, что в π' не было событий и хода времени. Следовательно, $(\pi', \omega') = (\pi'_0, \varepsilon)$. \square

Определим обратную-прямую бисимуляционную эквивалентность со слабым сохранением истории.

Определение 16. Пусть \mathcal{TN} , \mathcal{TN}' – НВСП, помеченные над Act , $\star \in \{por, prc\}$ и $\ast \in \{por, prc\}$.

- \mathcal{TN} и \mathcal{TN}' являются \star -прямо бисимуляционно эквивалентными со слабым сохранением истории (обозначается $\mathcal{TN} \approx_{\star}^{wh} \mathcal{TN}'$), если существует отображение (бисимуляция) $R \subseteq \mathcal{CHR}(\mathcal{TN}) \times \mathcal{CHR}(\mathcal{TN}')$ такое, что $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$ и для всех $((\pi, \omega), (\pi', \omega')) \in R$ выполняется:

- (1) – $\eta(\mathcal{TN}) \simeq \eta(\mathcal{TN}')$, в случае $\star = por$;
– $\mathcal{TN} \simeq \mathcal{TN}'$, в случае $\star = prc$;

- (2) если $(\pi, \omega) \rightarrow (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$, тогда существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}', \tilde{\omega}')$;

- (3) как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

- \mathcal{TN} и \mathcal{TN}' являются \ast -обратно \star -прямо бисимуляционно эквивалентными со слабым сохранением истории (обозначается $\mathcal{TN} \approx_{\ast\star}^{wh} \mathcal{TN}'$), если $\mathcal{TN} \approx_{\star}^{wh} \mathcal{TN}'$ с бисимуляцией R такой, что для всех $((\pi, \omega), (\pi', \omega')) \in R$ выполняется:

- (4) если $(\tilde{\pi}, \tilde{\omega}) \rightarrow (\pi, \omega)$ тогда существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ такая, что $(\tilde{\pi}', \tilde{\omega}') \rightarrow (\pi', \omega')$;

- (5) как пункт (4), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Несложно заметить, что в контексте приведенных выше бисимуляций семантика интерливинга (*int*) будет совпадать с семантикой частичного порядка (*por*) как в прямом, так и в обратном направлениях.

Пример 13. Рассмотрим НВСП \mathcal{TN}^{11} и \mathcal{TN}^{12} на рис. 9. Покажем, что $\mathcal{TN}^{11} \not\approx_{por}^{wh} \mathcal{TN}^{12}$. Предположим это не так, т.е. существует бисимуляция R , соответствующая определению 16. Тогда $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$.

Рассмотрим хронологию (π'_1, ω'_1) (расширение (π'_0, ε)), соответствующую срабатыванию перехода t_5 НВСП \mathcal{TN}^{12} после временной задержки 2. Тогда, согласно определению 16, существует $((\pi_1, \omega_1), (\pi'_1, \omega'_1)) \in R$ такая, что $(\pi_0, \varepsilon) \rightarrow (\pi_1, \omega_1)$ и $\eta(\mathcal{TN}_1) \simeq \eta(\mathcal{TN}'_1)$. Видно, что (π_1, ω_1) в этом случае соответствует срабатыванию перехода t_2 НВСП \mathcal{TN}^{11} .

Далее, (π_1, ω_1) может быть расширен действием b до хронологии (π_2, ω_2) так, что действие a будет являться причиной для действия b . Аналогично, поскольку $(\pi_1, \omega_1) \rightarrow (\pi_2, \omega_2)$, то должна существовать пара $((\pi_2, \omega_2), (\pi'_2, \omega'_2)) \in R$ такая, что $(\pi'_1, \omega'_1) \rightarrow (\pi'_2, \omega'_2)$ и $\eta(\mathcal{TN}_2) \simeq \eta(\mathcal{TN}'_2)$, согласно определению 16. Видно, что не существует расширения для π'_1 , в котором действие b следовало бы по причине за действием a , получили противоречие. Следовательно, $\mathcal{TN}^{11} \not\approx_{por}^{wh} \mathcal{TN}^{12}$.

С другой стороны, можно заметить, что $\mathcal{TN}^{11} \approx_{int\,prc} \mathcal{TN}^{12}$.

Также отметим, что в случае сильной временной стратегии сети будут иметь схожее поведение, поскольку переход действия a с временным интервалом $[0, 1]$ будет обязан срабатывать раньше других.

3.5. Иерархия эквивалентностей. Сначала рассмотрим взаимосвязи обратных-прямых эквивалентностей и обычных бисимуляционных эквивалентностей относительно разных семантик.

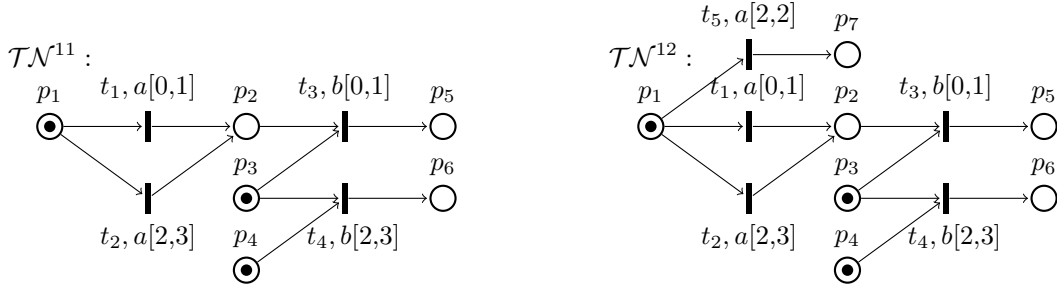


Рис. 9. $\approx_{int\ prc} \not\approx \approx_{por}^{wh}$

Теорема 3. Для НВСП \mathcal{TN} , \mathcal{TN}' , $\star \in \{int, por, prc\}$ выполняется:

- (a) $\mathcal{TN} \rightleftharpoons_{\star} \mathcal{TN}' \iff \mathcal{TN} \approx_{\star} \mathcal{TN}'$;
- (б) $\mathcal{TN} \approx_{int} \mathcal{TN}' \iff \mathcal{TN} \approx_{int\ int} \mathcal{TN}'$;
- (в) $\mathcal{TN} \approx_{\star\ int} \mathcal{TN}' \iff \mathcal{TN} \approx_{\star\ \star} \mathcal{TN}'$;

Доказательство. Рассмотрим доказательство для $\star = prc$ (доказательство остальных случаев аналогично).

- (a) \Rightarrow Пусть $\mathcal{TN} \rightleftharpoons_{prc} \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \approx_{prc} \mathcal{TN}'$ с бисимуляцией $R_2 = \{((\pi, \omega), (\pi', \omega')) \mid (\pi, \pi') \in R_1, \omega \in \mathcal{GRF}(\mathcal{TN}), \omega' \in \mathcal{GRF}(\mathcal{TN}')\}$. По определению 12, $(\pi_0, \pi'_0) \in R_1$. Значит, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_2$, так как $\varepsilon \in \mathcal{GRF}(\mathcal{TN}_0) \cap \mathcal{GRF}(\mathcal{TN}'_0)$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R_2$. Тогда $(\pi, \pi') \in R_1$, $\omega \in \mathcal{GRF}(\mathcal{TN})$ и $\omega' \in \mathcal{GRF}(\mathcal{TN}')$.

- (1) Пусть $(\pi, \omega) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\tilde{\pi}, \tilde{\omega})$ для некоторого $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$, т.е. $\tilde{\omega} \in \mathcal{GRF}(\widehat{\mathcal{TN}})$ и $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$. Согласно определению 12 и $(\pi, \pi') \in R_1$, существует пара $(\tilde{\pi}, \tilde{\pi}') \in R_1$ такая, что $\pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ и $\widehat{\mathcal{TN}} \simeq \widehat{\mathcal{TN}'}$. Благодаря предложению 1, найдется график $\hat{\omega}' \in \mathcal{GRF}(\widehat{\mathcal{TN}'})$. Поскольку $\widehat{\mathcal{TN}'}$ — расширение \mathcal{TN}' на $\widehat{\mathcal{TN}}$, то $\omega' \hat{\omega}' \in \mathcal{GRF}(\widehat{\mathcal{TN}'})$ и $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \omega' \hat{\omega}')) \in \mathcal{CHR}(\mathcal{TN}) \times \mathcal{CHR}(\mathcal{TN}')$. Следовательно, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \omega' \hat{\omega}')) \in R_2$, $(\pi', \omega') \xrightarrow{(\hat{\pi}', \hat{\omega}')} (\tilde{\pi}', \omega' \hat{\omega}')$ и $\widehat{\mathcal{TN}} \simeq \widehat{\mathcal{TN}'}$.
- (2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 15 и пунктов (1–2), $R_2 : \mathcal{TN} \approx_{prc} \mathcal{TN}'$.

\Leftarrow Пусть $R_1 : \mathcal{TN} \approx_{prc} \mathcal{TN}'$. Покажем, что $\mathcal{TN} \rightleftharpoons_{prc} \mathcal{TN}'$ с бисимуляцией $R_2 = \{(\pi, \pi') \mid ((\pi, \omega), (\pi', \omega')) \in R_1\}$.

По определению 15, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_1$. Значит, $(\pi_0, \pi'_0) \in R_2 \neq \emptyset$.

Рассмотрим произвольную пару $(\pi, \pi') \in R_2$. Тогда существует $((\pi, \omega), (\pi', \omega')) \in R_1$.

- (1) Пусть $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$. В силу предложения 1, существует $\hat{\omega} \in \mathcal{GRF}(\widehat{\mathcal{TN}})$. Значит, $(\pi, \omega) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\tilde{\pi}, \omega \hat{\omega})$. Согласно определению 15(1), существует пара $((\tilde{\pi}, \omega \hat{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_1$ такая, что $(\pi', \omega') \xrightarrow{(\hat{\pi}', \hat{\omega}')} (\tilde{\pi}', \tilde{\omega}')$ и $\widehat{\mathcal{TN}} \simeq \widehat{\mathcal{TN}'}$. Следовательно, $(\tilde{\pi}, \tilde{\pi}') \in R_2$, $\pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ и $\widehat{\mathcal{TN}} \simeq \widehat{\mathcal{TN}'}$.

(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 12 и пунктов (1-2), $R_2 : \mathcal{TN} \rightleftharpoons_{prc} \mathcal{TN}'$.

(б) \Rightarrow Пусть $\mathcal{TN} \approx_{int} \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \approx_{int int} \mathcal{TN}'$ с бисимуляцией $R_2 = \{((\pi, \omega), (\pi', \omega')) \in R_1 \mid l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n, n \geq 0, (\pi_0, \omega_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega), (\pi'_0, \omega'_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega'), ((\pi_i, \omega_i), (\pi'_i, \omega'_i)), ((\tilde{\pi}_i, \omega_i), (\tilde{\pi}'_i, \omega'_i)) \in R_1, 0 \leq i \leq n\}$.

По определению 15, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_1$. Так как $l(\pi_0, \varepsilon) = l(\pi'_0, \varepsilon) = 0$, $(\pi_0, \varepsilon) \xrightarrow{0} (\pi_0, \varepsilon)$ и $(\pi'_0, \varepsilon) \xrightarrow{0} (\pi'_0, \varepsilon)$, то $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_2$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R_2$. Тогда $l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n$ для некоторого $n \geq 0$. Кроме того, $(\pi_0, \omega_0) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega)$, $(\pi'_0, \omega'_0) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega')$ и $((\pi_i, \omega_i), (\pi'_i, \omega'_i)), ((\tilde{\pi}_i, \omega_i), (\tilde{\pi}'_i, \omega'_i)) \in R_1$ для $0 \leq i \leq n$.

(1) Предположим $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$ и $(\tilde{\pi}, \tilde{\omega})$ — расширение (π, ω) на время или событие. Тогда $(\pi, \omega) \xrightarrow{x} (\tilde{\pi}, \tilde{\omega})$, где $x \in \mathbb{R}_{\geq 0} \cup Act$. По определению 15, существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_1$ такая, что $(\pi', \omega') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\tilde{\pi}', \tilde{\omega}')$ и $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, т.е. $(\pi', \omega') \xrightarrow{x} (\tilde{\pi}', \tilde{\omega}')$. Кроме того, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_2$, поскольку $l(\tilde{\pi}, \tilde{\omega}) = l(\tilde{\pi}', \tilde{\omega}')$.

(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

(3) Предположим $(\tilde{\pi}, \tilde{\omega}) \xrightarrow{(\pi, \omega)} (\pi, \omega)$ и (π, ω) — расширение $(\tilde{\pi}, \tilde{\omega})$ на время или событие, т.е. $(\tilde{\pi}, \tilde{\omega}) \xrightarrow{x} (\pi, \omega)$ и $x \in \mathbb{R}_{\geq 0} \cup Act$. Тогда $(\tilde{\pi}, \tilde{\omega}) = (\tilde{\pi}_{n-1}, \omega_{n-1}) \xrightarrow{x=a_n} (\pi, \omega)$ или $(\pi_n, \omega_n) \xrightarrow{\theta_n-x} (\tilde{\pi}, \tilde{\omega}) \xrightarrow{x} (\pi, \omega)$, поскольку $(\tilde{\pi}, \tilde{\omega})$ — префикс (π, ω) . Значит, существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_1$ такая, что $(\tilde{\pi}', \tilde{\omega}')) \xrightarrow{x} (\pi', \omega')$, т.е. $(\tilde{\pi}', \tilde{\omega}')) \xrightarrow{(\pi', \omega')} (\pi', \omega')$ и $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$. Кроме того, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_2$, поскольку $l(\tilde{\pi}, \tilde{\omega}) = l(\tilde{\pi}', \tilde{\omega}')$.

(4) Как пункт (3), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 15 и пунктов (1-4), $\mathcal{TN} \approx_{int int} \mathcal{TN}'$.

\Leftarrow Следует из определения 15.

(в) \Rightarrow Пусть $R : \mathcal{TN} \approx_{prc int} \mathcal{TN}'$. Покажем, что $R : \mathcal{TN} \approx_{prc prc} \mathcal{TN}'$.

По определению 15, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R$.

(1) Пусть $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$ и $l(\tilde{\pi}, \tilde{\omega}) = \theta_0 a_1 \theta_1 \dots a_n \theta_n$ для некоторого $n \geq 0$. Согласно лемме 3, существует последовательность расширений вида $(\pi, \omega) = (\pi^0, \omega^0) \xrightarrow{\theta_0} (\tilde{\pi}^0, \omega^0) \xrightarrow{a_1} (\pi^1, \omega^1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi^n, \omega^n) \xrightarrow{\theta_n} (\tilde{\pi}^n, \omega^n) = (\tilde{\pi}, \tilde{\omega})$. Последовательно применяя пункт (1) определения 15 для int случая, получим существование пар $((\pi^i, \omega^i), (\pi'^i, \omega'^i)), ((\tilde{\pi}^i, \omega^i), (\tilde{\pi}'^i, \omega'^i)) \in R$ ($0 \leq i \leq n$) таких, что $(\pi', \omega') = (\pi^{0'}, \omega^{0'}) \xrightarrow{\theta_0} (\tilde{\pi}^{0'}, \omega^{0'}) \xrightarrow{a_1} (\pi^{1'}, \omega^{1'}) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi^{n'}, \omega^{n'}) \xrightarrow{\theta_n} (\tilde{\pi}^{n'}, \omega^{n'}) = (\tilde{\pi}', \tilde{\omega}')$. Значит, по лемме 3, $(\pi', \omega') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\tilde{\pi}', \tilde{\omega}')$, где $l(\tilde{\pi}', \tilde{\omega}')) = \theta_0 a_1 \theta_1 \dots a_n \theta_n = l(\tilde{\pi}, \tilde{\omega})$.

Покажем, что $\widehat{TN} \simeq \widehat{TN}'$. Поскольку $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ и $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$, то существует $((\pi, \omega), (\pi^*, \omega^*)) \in R$ такая, что $(\pi^*, \omega^*) \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\tilde{\pi}', \tilde{\omega}')$ и $\gamma : \widehat{TN} \simeq \widehat{TN}^*$, по определению 15(3). В силу изоморфизма, $|\widehat{E}^*| = |\widehat{E}|$ и $\hat{\tau}^*(\bullet \widehat{N}^*) = \hat{\tau}^*(\gamma(\bullet \widehat{N})) = \hat{\tau}(\bullet \widehat{N}) = \theta_0$. Тогда $|\widehat{\omega}^*| = |\widehat{\omega}| = |\widehat{\omega}'| = n$, поскольку в графиках каждое событие присутствует ровно один раз. Так как $\widehat{\omega}^*$ и $\widehat{\omega}'$ являются окончаниями $\tilde{\omega}'$ одинаковой длины, то $\widehat{\omega}^* = \widehat{\omega}'$. Как следствие, $\widehat{E}^* = \widehat{E}'$. Значит, \widehat{TN}^* совпадает с \widehat{TN}' , согласно конструкции расширения \widehat{TN}' и равенству временных значений граничных сечений ($\hat{\tau}^*(\bullet \widehat{N}^*) = \theta_0 = \hat{\tau}(\bullet \widehat{N}')$). Следовательно, $\widehat{TN} \simeq \widehat{TN}'$.

(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Из $R : \mathcal{TN} \approx_{prcint} \mathcal{TN}'$ следует $R : \mathcal{TN} \approx_{prcprc} \mathcal{TN}'$, в силу определения 15 и пунктов (1–2).

⇐ Поскольку из изоморфизма ВПСС следует изоморфизм его ВЧУМ событий, то из $\mathcal{TN} \approx_{prcprc} \mathcal{TN}'$ следует $\mathcal{TN} \approx_{prcint} \mathcal{TN}'$, по определению 15. \square

Далее, сравним эквивалентности со слабым сохранением истории.

Теорема 4. Для НВСП \mathcal{TN} , \mathcal{TN}' , помеченных над Act , и $\star \in \{por, prc\}$ выполняется:

$$\mathcal{TN} \Leftrightarrow_{\star}^{wh} \mathcal{TN}' \iff \mathcal{TN} \approx_{\star}^{wh} \mathcal{TN}' \iff \mathcal{TN} \approx_{\star\star}^{wh} \mathcal{TN}'.$$

Доказательство. Рассмотрим доказательство для $\star = prc$ (доказательство случая $\star = por$ аналогично).

- $\mathcal{TN} \Leftrightarrow_{prc}^{wh} \mathcal{TN}' \Rightarrow \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$.

Пусть $\mathcal{TN} \Leftrightarrow_{prc}^{wh} \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$ с бисимуляцией $R_2 = \{((\pi, \omega), (\pi', \omega')) \mid (\pi, \pi') \in R_1, \omega \in \mathcal{GRF}(TN), \omega' \in \mathcal{GRF}(TN')\}$.

По определению 13, $(\pi_0, \pi'_0) \in R_1$. Значит, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_2$, так как $\varepsilon \in \mathcal{GRF}(TN_0) \cap \mathcal{GRF}(TN'_0)$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R_2$. Тогда $(\pi, \pi') \in R_1, \omega \in \mathcal{GRF}(TN)$ и $\omega' \in \mathcal{GRF}(TN')$.

- (1) $TN \simeq TN'$, согласно определению 13(1).
- (2) Пусть $(\pi, \omega) \rightarrow (\tilde{\pi}, \tilde{\omega})$ для некоторого $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(TN)$, т.е. $\tilde{\omega} \in \mathcal{GRF}(\widehat{TN})$ и $\pi \rightarrow \tilde{\pi}$. Согласно определению 13(2) и $(\pi, \pi') \in R_1$, существует пара $(\tilde{\pi}, \tilde{\pi}') \in R_1$ такая, что $\pi' \xrightarrow{\tilde{\pi}'} \tilde{\pi}'$. Благодаря предложению 1, найдется график $\widehat{\omega}' \in \mathcal{GRF}(\widehat{TN}')$. Поскольку \widehat{TN}' — расширение TN' на \widehat{TN}' , то $\omega' \widehat{\omega}' \in \mathcal{GRF}(\widehat{TN}')$ и $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \omega' \widehat{\omega}')) \in \mathcal{CHR}(TN) \times \mathcal{CHR}(TN')$. Следовательно, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \omega' \widehat{\omega}')) \in R_2$, $(\pi', \omega') \rightarrow (\tilde{\pi}', \omega' \widehat{\omega}')$.
- (3) Как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 16 и пунктов (1–3), $R_2 : \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$.

- $\mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}' \Rightarrow \mathcal{TN} \approx_{prcprc}^{wh} \mathcal{TN}'$.

Пусть $\mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$ с бисимуляцией R_1 . Покажем, что $\mathcal{TN} \approx_{prcprc}^{wh} \mathcal{TN}'$ с бисимуляцией $R_2 = \{((\pi, \omega), (\pi', \omega')) \in R_1 \mid l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n, n \geq 0, (\pi_0, \omega_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega), (\pi'_0, \omega'_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega'), ((\pi_i, \omega_i), (\pi'_i, \omega'_i)), ((\tilde{\pi}_i, \omega_i), (\tilde{\pi}'_i, \omega'_i)) \in R_1, 0 \leq i \leq n\}$.

По определению 16, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_1$. Так как $l(\pi_0, \varepsilon) = l(\pi'_0, \varepsilon) = 0$, $(\pi_0, \varepsilon) \xrightarrow{0} (\pi_0, \varepsilon)$ и $(\pi'_0, \varepsilon) \xrightarrow{0} (\pi'_0, \varepsilon)$, то $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_2$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R_2$. Тогда $l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n$ для некоторого $n \geq 0$. Кроме того, $(\pi_0, \omega_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega)$, $(\pi'_0, \omega'_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega')$ и $((\pi_i, \omega_i), (\pi'_i, \omega'_i)), ((\tilde{\pi}_i, \omega_i), (\tilde{\pi}'_i, \omega'_i)) \in R_1$ для $0 \leq i \leq n$.

(1) Поскольку $((\pi, \omega), (\pi', \omega')) \in R_1$, то $TN \simeq TN'$, по определению 16.

(2) Предположим $(\pi, \omega) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$ и $l(\tilde{\pi}, \tilde{\omega}) = \theta'_0 a'_1 \theta'_1 \dots a'_m \theta'_m$ ($m \geq 0$). Тогда, по лемме 3, существует последовательность расширений $(\pi, \omega) = (\pi^0, \omega^0) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0) \xrightarrow{a'_1} (\pi^1, \omega^1) \xrightarrow{\theta'_1} \dots \xrightarrow{a'_m} (\pi^m, \omega^m) \xrightarrow{\theta'_m} (\tilde{\pi}^m, \omega^m) = (\tilde{\pi}, \tilde{\omega})$.

Покажем индукцией по $0 \leq j \leq m$, что существуют пары $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})), ((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_2$ такие, что $(\pi', \omega') \rightarrow (\pi^{j'}, \omega^{j'})$ и $(\pi', \omega') \rightarrow (\tilde{\pi}^{j'}, \omega^{j'})$.

($j = 0$) Пусть $(\pi^{0'}, \omega^{0'}) = (\pi', \omega')$. Тогда, $((\pi^0, \omega^0), (\pi^{0'}, \omega^{0'})) \in R_2$ и $(\pi', \omega') \rightarrow (\pi^{0'}, \omega^{0'})$.

Поскольку $(\pi, \omega) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0)$ и $((\pi, \omega), (\pi', \omega')) \in R_1$, то существует пара $((\tilde{\pi}^0, \omega^0), (\tilde{\pi}^{0'}, \omega^{0'})) \in R_1$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}^{0'}, \omega^{0'})$ и $\widetilde{TN}^0 \simeq \widetilde{TN}^{0'}$, по определению 16. В силу изоморфизма, $(\pi', \omega') \xrightarrow{\theta'_0} (\tilde{\pi}^{0'}, \omega^{0'})$. Тогда $(\pi_n, \omega_n) \xrightarrow{\theta_n} (\pi, \omega) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0)$ и $(\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\pi', \omega') \xrightarrow{\theta'_0} (\tilde{\pi}^{0'}, \omega^{0'})$, т.е. $(\pi_n, \omega_n) \xrightarrow{\theta_n + \theta'_0} (\tilde{\pi}^0, \omega^0)$ и $(\pi'_n, \omega'_n) \xrightarrow{\theta_n + \theta'_0} (\tilde{\pi}^{0'}, \omega^{0'})$. Следовательно, $l(\tilde{\pi}^{0'}, \omega^{0'}) = l(\tilde{\pi}^0, \omega^0) = \theta_0 a_1 \theta_1 \dots a_n (\theta_n + \theta'_0)$, по лемме 3, и $((\tilde{\pi}^0, \omega^0), (\tilde{\pi}^{0'}, \omega^{0'})) \in R_2$ так как $((\tilde{\pi}^0, \omega^0), (\tilde{\pi}^{0'}, \omega^{0'})) \in R_1$.

($j > 0$) По предположению индукции, существует $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}^{j-1'}, \omega^{j-1'})$. Тогда $l(\tilde{\pi}^{j-1}, \omega^{j-1}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'})$ и $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_1$.

Поскольку $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j)$, то существует $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})) \in R_1$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}^{j-1'}, \omega^{j-1'}) \rightarrow (\pi^{j'}, \omega^{j'})$ и $TN^j \simeq TN^{j'}$, по определению 16. В силу изоморфизма, $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'})$, т.е. $l(\pi^{j'}, \omega^{j'}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'}) a'_j 0 = l(\pi^j, \omega^j)$. Следовательно, $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})) \in R_2$, так как $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$, $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j) \xrightarrow{0} (\pi^j, \omega^j)$, $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'}) \xrightarrow{0} (\pi^{j'}, \omega^{j'})$ и $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})) \in R_1$.

Аналогично, в силу определения 16, из $(\pi^j, \omega^j) \xrightarrow{\theta'_j} (\tilde{\pi}^j, \omega^j)$ следует существование $((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_1$ такой, что $(\pi', \omega') \rightarrow (\pi^{j'}, \omega^{j'}) \rightarrow (\tilde{\pi}^{j'}, \omega^{j'})$, $TN^j \simeq TN^{j'}$. В силу изоморфизма, $(\pi^{j'}, \omega^{j'}) \xrightarrow{\theta'_j} (\tilde{\pi}^{j'}, \omega^{j'})$ и $l(\tilde{\pi}^{j'}, \omega^{j'}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'}) a'_j \theta'_j = l(\tilde{\pi}^j, \omega^j)$. Значит, $((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_2$, так как $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$, $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j) \xrightarrow{\theta'_j} (\tilde{\pi}^j, \omega^j)$, $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'}) \xrightarrow{\theta'_j} (\tilde{\pi}^{j'}, \omega^{j'})$ и $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})), ((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_1$.

Следовательно, существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) = ((\tilde{\pi}^m, \omega^m), (\tilde{\pi}^{m'}, \omega^{m'})) \in R_2$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}', \tilde{\omega}')$.

- (3) Как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.
- (4) Предположим $(\tilde{\pi}, \tilde{\omega}) \rightarrow (\pi, \omega)$. Тогда $l(\tilde{\pi}, \tilde{\omega}) = \theta_0 a_1 \dots \theta_{m-1} a_m \theta$ для некоторого $m \leq n$ и $\theta \leq \theta_m \in \mathbb{R}_{\geq 0}$, так как $\tilde{\omega}$ является префиксом ω . Кроме того, $(\pi_0, \omega_0) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} \dots \xrightarrow{a_m} (\pi_m, \omega_m) \xrightarrow{\theta} (\tilde{\pi}, \tilde{\omega} = \omega_m) \xrightarrow{\theta_m - \theta} (\tilde{\pi}_m, \omega_m) \xrightarrow{a_{m+1}} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega)$, согласно лемме 3. Поскольку $(\pi_m, \omega_m) \in R_1$, то существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_1$ такая, что $(\pi'_m, \omega'_m) \rightarrow (\tilde{\pi}', \tilde{\omega}')$ и $TN \simeq TN'$, по определению 16. В силу изоморфизма, $(\pi'_m, \omega'_m) \xrightarrow{\theta} (\tilde{\pi}', \tilde{\omega}')$ и, как следствие, $(\pi'_0, \omega'_0) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} \dots \xrightarrow{a_m} (\pi'_m, \omega'_m) \xrightarrow{\theta} (\tilde{\pi}', \tilde{\omega}' = \omega'_m) \xrightarrow{\theta_m - \theta} (\tilde{\pi}'_m, \omega'_m) \xrightarrow{a_{m+1}} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega')$, т.е. $l(\tilde{\pi}', \tilde{\omega}') = \theta_0 a_1 \theta_1 \dots a_m \theta = l(\tilde{\pi}, \tilde{\omega})$. Значит, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_2$, так как $(\pi_m, \omega_m) \xrightarrow{\theta} (\tilde{\pi}, \tilde{\omega})$, $(\pi'_m, \omega'_m) \xrightarrow{\theta} (\tilde{\pi}', \tilde{\omega}')$ и $((\pi_i, \omega_i), (\pi'_i, \omega'_i)), ((\tilde{\pi}_j, \omega_j), (\tilde{\pi}'_j, \omega'_j)), ((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_1$ для $0 \leq i \leq m$ и $0 \leq j < m$. Следовательно, существует $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_2$ такая, что $(\tilde{\pi}', \tilde{\omega}') \rightarrow (\pi', \omega')$.
- (5) Как пункт (4), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 16 и пунктов (1–5), $\mathcal{TN} \approx_{\text{prc prc}}^{\text{wh}} \mathcal{TN}'$.

- $\mathcal{TN} \approx_{\text{prc prc}}^{\text{wh}} \mathcal{TN}' \Rightarrow \mathcal{TN} \Leftarrow_{\text{prc}}^{\text{wh}} \mathcal{TN}'$.

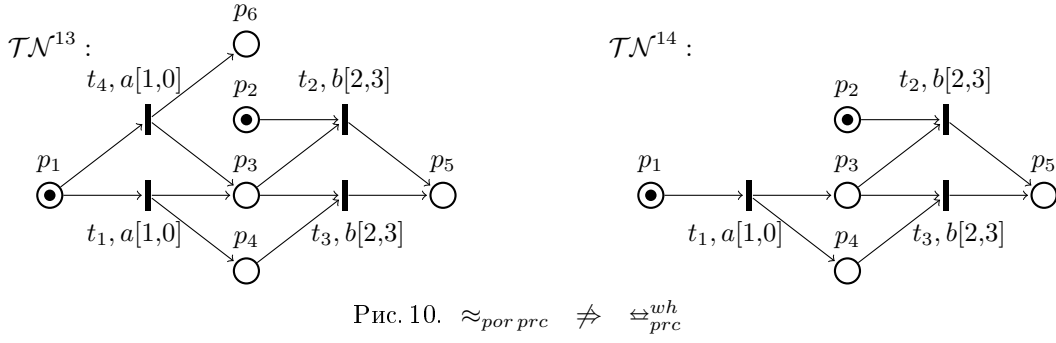
Пусть $R_1 : \mathcal{TN} \approx_{\text{prc prc}}^{\text{wh}} \mathcal{TN}'$. Покажем, что $\mathcal{TN} \Leftarrow_{\text{prc}}^{\text{wh}} \mathcal{TN}'$ с бисимуляцией $R_2 = \{(\pi, \pi') \mid ((\pi, \omega), (\pi', \omega')) \in R_1\}$.

По определению 16, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_1$. Значит, $(\pi_0, \pi'_0) \in R_2 \neq \emptyset$.

Рассмотрим произвольную пару $(\pi, \pi') \in R_2$.

- (1) $TN \simeq TN'$, по определению 16.
- (2) Пусть $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ для $\tilde{\pi} \in \mathcal{PRC}(\mathcal{TN})$. Из $(\pi, \pi') \in R_2$ следует существование пары $((\pi, \omega), (\pi', \omega')) \in R_1$. В силу предложения 1, существует $\hat{\omega} \in \mathcal{GRF}(\widehat{TN})$. Значит, $(\pi, \omega) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\tilde{\pi}, \omega \hat{\omega})$. Согласно определению 16(2), существует пара $((\tilde{\pi}, \omega \hat{\omega}), (\tilde{\pi}', \omega')) \in R_1$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}', \omega')$. Следовательно, $(\tilde{\pi}, \tilde{\pi}') \in R_2$ и $\pi' \rightarrow \tilde{\pi}'$.
- (3) Как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу определения 13 и пунктов (1–3), $R_2 : \mathcal{TN} \Leftarrow_{\text{prc}}^{\text{wh}} \mathcal{TN}'$. □

Рис. 10. $\approx_{por\ prc} \not\equiv \Leftrightarrow_{prc}^{wh}$

Пример 14. Рассмотрим НВСП \mathcal{TN}^{13} и \mathcal{TN}^{14} на рис. 10. Покажем, что $\mathcal{TN}^{13} \not\equiv_{prc}^{wh} \mathcal{TN}^{14}$. Предположим обратное, т. е. существует бисимуляция R из определения 13. Тогда $(\pi_0, \pi'_0) \in R$.

Пусть π_1 — расширение π_0 , соответствующее срабатыванию перехода t_4 с единичной временной задержкой, т. е. $\pi_0 \rightarrow \pi_1$. Значит, по определению 13, существует пара $(\pi_1, \pi'_1) \in R$ такая, что $\pi'_0 \rightarrow \pi'_1$ и $TN_1 \simeq TN'_1$. В этом случае π'_1 соответствует срабатыванию перехода t_1 НВСП \mathcal{TN}^{14} .

Далее, π'_1 может быть расширено до временного процесса π'_2 в результате срабатывания перехода t_3 с временной задержкой 2. Аналогично, согласно определению 13, должна существовать пара $(\pi_2, \pi'_2) \in R$ такая, что $\pi_1 \rightarrow \pi_2$ и $TN_2 \simeq TN'_2$. Однако, такого расширения временного процесса π_1 с изоморфизмом ВПСС у \mathcal{TN}^{13} не существует, что приводит к противоречию. Следовательно, $\mathcal{TN}^{13} \not\equiv_{prc}^{wh} \mathcal{TN}^{14}$.

С другой стороны, данные сети прямо обратны бисимуляционно эквивалентны, когда при движении в обратном направлении мы используем семантику ВЧУМ, т. е. $\mathcal{TN}^{13} \approx_{por\ prc} \mathcal{TN}^{14}$.

Следующая теорема сравнивает обратные-прямые и сохраняющие историю эквивалентности.

Теорема 5. Для НВСП \mathcal{TN} , \mathcal{TN}' , помеченных над Act , и $\star \in \{por, prc\}$ выполняется:

- (а) $\mathcal{TN} \Leftrightarrow_{\star}^h \mathcal{TN}' \Rightarrow \mathcal{TN} \approx_{\star\star} \mathcal{TN}'$;
- (б) $\mathcal{TN} \approx_{por\ por} \mathcal{TN}' \Rightarrow \mathcal{TN} \Leftrightarrow_{por}^h \mathcal{TN}'$;
- (в) $\mathcal{TN} \approx_{prc\ prc} \mathcal{TN}' \Rightarrow \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$.

Доказательство.

- (а) Докажем для $\star = prc$, случай $\star = por$ рассматривается аналогично. Сформулируем вспомогательное утверждение, которое следует из определения изоморфизма и конструкции расширений ВПСС.

Утверждение. Если $TN \xrightarrow{\widehat{TN}} \widehat{TN}$, $TN' \xrightarrow{\widehat{TN}'} \widehat{TN}'$ для ВПСС \widehat{TN} и \widehat{TN}' , $f : TN \simeq TN'$, $\tilde{f} : \widehat{TN} \simeq \widehat{TN}'$ и $f \subseteq \tilde{f}$, то $\widehat{TN} \simeq \widehat{TN}'$. \square

Пусть $R_1 : \mathcal{TN} \Leftrightarrow_{prc}^h \mathcal{TN}'$. Покажем, что $\mathcal{TN} \approx_{prc\ prc} \mathcal{TN}'$ с бисимуляцией $R_2 = \{((\pi, \omega), (\pi', \omega')) \in \mathcal{CHR}(\mathcal{TN}) \times \mathcal{CHR}(\mathcal{TN}') \mid l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n, n \geq 0\}$.

$$0, (\pi_0, \omega_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega),$$

$$(\pi'_0, \omega'_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega'),$$

$$(\pi_i, \pi'_i, f_i), (\tilde{\pi}_i, \tilde{\pi}'_i, f_i) \in R_1, f_i \subseteq f_j, 0 \leq i \leq j \leq n\}$$

По определению 14, $(\pi_0, \pi'_0, f_0) \in R_1$. Тогда $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_2$, поскольку $l(\pi, \omega) = l(\pi', \omega') = 0$, $(\pi_0, \varepsilon) \xrightarrow{0} (\pi_0, \varepsilon)$ и $(\pi'_0, \varepsilon) \xrightarrow{0} (\pi'_0, \varepsilon)$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R_2$. Тогда $l(\pi, \omega) = l(\pi', \omega') = \theta_0 a_1 \theta_1 \dots a_n \theta_n$, $(\pi_0, \omega_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} (\pi_1, \omega_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega)$, $(\pi'_0, \omega'_0 = \varepsilon) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} (\pi'_1, \omega'_1) \xrightarrow{\theta_1} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega')$ для некоторого $n \geq 0$. Кроме того, найдутся тройки $(\pi_i, \pi'_i, f_i), (\tilde{\pi}_i, \tilde{\pi}'_i, f_i) \in R_1$ такие, что $f_i \subseteq f_j$ для $0 \leq i \leq j \leq n$.

- (1) Пусть $(\pi, \omega) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$ и $l(\hat{\pi}, \hat{\omega}) = \theta'_0 a'_1 \theta'_1 \dots a'_m \theta'_m$ ($m \geq 0$). Тогда, по лемме 3, существует последовательность расширений $(\pi, \omega) = (\pi^0, \omega^0) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0) \xrightarrow{a'_1} (\pi^1, \omega^1) \xrightarrow{\theta'_1} \dots \xrightarrow{a'_m} (\pi^m, \omega^m) \xrightarrow{\theta'_m} (\tilde{\pi}^m, \omega^m) = (\tilde{\pi}, \tilde{\omega})$.

Покажем индукцией по $0 \leq j \leq m$, что существуют пары $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})), ((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_2$ такие, что $(\pi', \omega') \rightarrow (\pi^{j'}, \omega^{j'})$ и $(\pi', \omega') \rightarrow (\tilde{\pi}^{j'}, \omega^{j'})$.

- ($j = 0$) Пусть $(\pi^{0'}, \omega^{0'}) = (\pi', \omega')$. Тогда $((\pi^0, \omega^0), (\pi^{0'}, \omega^{0'})) \in R_2$ и $(\pi', \omega') \rightarrow (\pi^{0'}, \omega^{0'})$.

Из расширения $(\pi^0, \omega^0) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0)$, следует $\pi^0 \xrightarrow{\theta'_0} \tilde{\pi}^0$. Поскольку $(\tilde{\pi}_n, \tilde{\pi}'_n, f_n) = (\pi^0, \pi^{0'}, f_n) \in R_1$, то существует тройка $(\tilde{\pi}^0, \tilde{\pi}^{0'}, f^0) \in R_1$ такая, что $\pi^{0'} \rightarrow \tilde{\pi}^{0'}$, $f^0 : \widetilde{TN}^0 \simeq \widetilde{TN}^{0'}$ и $f_n \subseteq f^0$, по определению 14. Благодаря изоморфизмам $TN^0 \simeq TN^{0'}$, $\widetilde{TN}^0 \simeq \widetilde{TN}^{0'}$ и $\pi^0 \xrightarrow{\theta'_0} \tilde{\pi}^0$, получаем $f^0 = f_n$ и $\pi^{0'} \xrightarrow{\theta'_0} \tilde{\pi}^{0'}$. Значит, согласно лемме 3, $l(\tilde{\pi}^{0'}, \omega^{0'}) = l(\tilde{\pi}^0, \omega^0) = \theta_0 a_1 \theta_1 \dots a_n (\theta_n + \theta'_0)$, поскольку $(\pi_n, \omega_n) \xrightarrow{\theta_n} (\pi^0, \omega^0) \xrightarrow{\theta'_0} (\tilde{\pi}^0, \omega^0)$ и $(\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\pi^{0'}, \omega^{0'}) \xrightarrow{\theta'_0} (\tilde{\pi}^{0'}, \omega^{0'})$, т.е. $(\pi_n, \omega_n) \xrightarrow{\theta_n + \theta'_0} (\tilde{\pi}^0, \omega^0)$ и $(\pi'_n, \omega'_n) \xrightarrow{\theta_n + \theta'_0} (\tilde{\pi}^{0'}, \omega^{0'})$. Следовательно, $((\tilde{\pi}^0, \omega^0), (\tilde{\pi}^{0'}, \omega^{0'})) \in R_2$ и $(\pi', \omega') \rightarrow (\tilde{\pi}^{0'}, \omega^{0'})$, так как $(\tilde{\pi}^0, \tilde{\pi}^{0'}, f^0) \in R_1$ и $f^0 = f_n$.

- ($j > 0$) По предположению индукции, существует $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}^{j-1'}, \omega^{j-1'})$. Тогда $l(\tilde{\pi}^{j-1}, \omega^{j-1}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'})$ и существует $(\tilde{\pi}^{j-1}, \tilde{\pi}^{j-1'}, f^{j-1}) \in R_1$, $f^{j-1} : \widetilde{TN}^{j-1} \simeq \widetilde{TN}^{j-1'}$.

Поскольку $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j)$, т.е. $\tilde{\pi}^{j-1} \xrightarrow{a'_j} \pi^j$, то существует $(\pi^j, \pi^{j'}, f^j) \in R_1$ такая, что $\pi^j \rightarrow \tilde{\pi}^{j-1'}$, $f^j : TN^j \simeq TN^{j'}$ и $f^{j-1} \subseteq f^j$, по определению 14. В силу изоморфизмов $\widetilde{TN}^{j-1} \simeq \widetilde{TN}^{j-1'}$, $TN^j \simeq TN^{j'}$ и $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j)$, получаем $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'})$. Значит, $l(\pi^{j'}, \omega^{j'}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'}) a'_j 0 = l(\pi^j, \omega^j)$. Следовательно, $((\pi^j, \omega^j), (\pi^{j'}, \omega^{j'})) \in R_2$, так как $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$, $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j) \xrightarrow{0} (\pi^j, \omega^j)$, $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'}) \xrightarrow{0} (\pi^{j'}, \omega^{j'})$, $(\pi^j, \pi^{j'}, f^j) \in R_1$ и $f^{j-1} \subseteq f^j$. Аналогично, из $(\pi^j, \omega^j) \xrightarrow{\theta'_j} (\tilde{\pi}^j, \omega^j)$ следует $\pi^j \xrightarrow{\theta'_j} \tilde{\pi}^j$ и существование $(\tilde{\pi}^j, \tilde{\pi}^{j'}, f^j) \in R_1$ такой, что $\tilde{\pi}^j \rightarrow \pi^{j'} \rightarrow \tilde{\pi}^{j'}$, $f^j : \widetilde{TN}^j \simeq \widetilde{TN}^{j'}$,

$f^j \subseteq \tilde{f}^j$. В силу изоморфизмов $TN^j \simeq TN^{j'}$, $\widehat{TN}^j \simeq \widehat{TN}^{j'}$ и $(\pi^j, \omega^j) \xrightarrow{\theta'_j} (\tilde{\pi}^j, \omega^j)$, получаем $(\pi^{j'}, \omega^{j'}) \xrightarrow{\theta'_j} (\tilde{\pi}^{j'}, \omega^{j'})$ и $\tilde{f}^j = f^j$. Значит, $l(\pi^{j'}, \omega^{j'}) = l(\tilde{\pi}^{j-1'}, \omega^{j-1'}) a'_j \theta'_j = l(\pi^j, \omega^j)$. Следовательно, $((\tilde{\pi}^j, \omega^j), (\tilde{\pi}^{j'}, \omega^{j'})) \in R_2$, так как $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_2$, $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{a'_j} (\pi^j, \omega^j) \xrightarrow{\theta'_j} (\tilde{\pi}^j, \omega^j)$, $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{a'_j} (\pi^{j'}, \omega^{j'}) \xrightarrow{\theta'_j} (\tilde{\pi}^{j'}, \omega^{j'})$, $(\pi^j, \pi^{j'}, f^j), (\tilde{\pi}^j, \tilde{\pi}^{j'}, f^j) \in R_1$ и $f^{j-1} \subseteq f^j$.

Получаем, что существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) = ((\tilde{\pi}^m, \omega^m), (\tilde{\pi}^{m'}, \omega^{m'})) \in R_2$ такая, что $(\pi', \omega') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\tilde{\pi}', \tilde{\omega}')$. Кроме того, $f_n = f^0 \subseteq f^m$, $f_n : TN \simeq TN'$ и $f^m : \widehat{TN} \simeq \widehat{TN}'$. Значит, согласно утверждению, $\widehat{TN} \simeq \widehat{TN}'$.

(2) Как пункт (1), но роли \mathcal{TN} и \mathcal{TN}' меняются.

(3) Пусть $(\tilde{\pi}, \tilde{\omega}) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\pi, \omega)$. Тогда $l(\tilde{\pi}, \tilde{\omega}) = \theta_0 a_1 \dots \theta_{m-1} a_m \theta$ для некоторого $m \leq n$ и $\theta \leq \theta_m \in \mathbb{R}_{\geq 0}$, так как $\tilde{\omega}$ является префиксом ω . Кроме того, $(\pi_0, \omega_0) \xrightarrow{\theta_0} (\tilde{\pi}_0, \omega_0) \xrightarrow{a_1} \dots \xrightarrow{a_m} (\pi_m, \omega_m) \xrightarrow{\theta} (\tilde{\pi}, \tilde{\omega} = \omega_m) \xrightarrow{\theta_m - \theta} (\tilde{\pi}_m, \omega_m) \xrightarrow{a_{m+1}} \dots \xrightarrow{a_n} (\pi_n, \omega_n) \xrightarrow{\theta_n} (\tilde{\pi}_n, \omega_n) = (\pi, \omega)$, согласно лемме 3. Поскольку $(\pi_m, \pi'_m, f_m) \in R_1$ и $\pi_m \xrightarrow{\theta} \tilde{\pi}$, то существует тройка $(\tilde{\pi}, \tilde{\pi}', \tilde{f}) \in R_1$ такая, что $\pi'_m \rightarrow \tilde{\pi}'$, $\tilde{f} : \widehat{TN} \simeq \widehat{TN}'$ и $f_m \subseteq \tilde{f}$, по определению 14. В силу изоморфизмов $TN_m \simeq TN'_m$, $\widehat{TN} \simeq \widehat{TN}'$ и $\pi_m \xrightarrow{\theta} \tilde{\pi}$, получаем $\pi'_m \xrightarrow{\theta} \tilde{\pi}'$, как следствие, $f_m = \tilde{f}$ и $(\pi'_m, \omega'_m) \xrightarrow{\theta} (\tilde{\pi}', \tilde{\omega}' = \omega'_m)$. Значит, $(\pi'_0, \omega'_0) \xrightarrow{\theta_0} (\tilde{\pi}'_0, \omega'_0) \xrightarrow{a_1} \dots \xrightarrow{a_m} (\pi'_m, \omega'_m) \xrightarrow{\theta} (\tilde{\pi}', \tilde{\omega}') \xrightarrow{\theta_m - \theta} (\tilde{\pi}'_m, \omega'_m) \xrightarrow{a_{m+1}} \dots \xrightarrow{a_n} (\pi'_n, \omega'_n) \xrightarrow{\theta_n} (\tilde{\pi}'_n, \omega'_n) = (\pi', \omega')$, т. е. $l(\tilde{\pi}', \tilde{\omega}') = \theta_0 a_1 \theta_1 \dots a_m \theta = l(\tilde{\pi}, \tilde{\omega})$. Следовательно, $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R_2$, так как $(\pi_i, \pi'_i, f_i), (\tilde{\pi}_i, \tilde{\pi}'_i, f_i), (\pi_m, \pi'_m, f_m), (\tilde{\pi}, \tilde{\pi}', f_m) \in R_1$, $f_i \subseteq f_j$, $f_{m-1} \subseteq f_m$ для $0 \leq i \leq j < m$. Кроме того, $(\tilde{\pi}', \tilde{\omega}') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\pi', \omega')$. Так как $\widehat{TN} \xrightarrow{\widehat{TN}} TN$, $\widehat{TN}' \xrightarrow{\widehat{TN}'} TN'$, $f_m : \widehat{TN} \simeq \widehat{TN}'$, $f_n : TN \simeq TN'$ и $f_m \subseteq f_n$, то $\widehat{TN} \simeq \widehat{TN}'$, согласно утверждению.

(4) Как пункт (3), но роли \mathcal{TN} и \mathcal{TN}' меняются.

В силу пунктов (1–4) и определения 15, $R_2 : \mathcal{TN} \approx_{prc\ prc} \mathcal{TN}'$.

(6) Пусть $\mathcal{TN} \approx_{por\ por} \mathcal{TN}'$ с некоторой бисимуляцией R_1 . Покажем, что $\mathcal{TN} \stackrel{h}{\approx}_{por} \mathcal{TN}'$ с бисимуляцией $R_2 = \{(\pi, \pi', f) \mid f : \eta(TN) \simeq \eta(TN'), ((\pi, \omega), (\pi', \omega')) \in R_1, \omega = e_1 \dots e_n, \omega' = e'_1 \dots e'_n, n \geq 0, f(e_i) = e'_i, 0 < i \leq n\}$.

По определению 15, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R_1$. Значит, $(\pi_0, \pi'_0, f_0) \in R_2$, так как ε не содержит событий.

Рассмотрим произвольную тройку $(\pi, \pi', f) \in R_2$. Тогда $f : \eta(TN) \simeq \eta(TN')$ и существует пара $((\pi, \omega), (\pi', \omega')) \in R_1$ такая, что $\omega = e_1 \dots e_n$, $\omega' = e'_1 \dots e'_n$ для $n \geq 0$ и $f(e_i) = e'_i$ для $0 < i \leq n$, по построению.

(1) $f : \eta(TN) \simeq \eta(TN')$, по построению.

(2) Пусть $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$. Согласно предложению 1, существует $\hat{\omega} \in \mathcal{GRF}(\widehat{TN})$ и $(\pi, \omega) \xrightarrow{(\hat{\pi}, \hat{\omega})} (\tilde{\pi}, \omega \hat{\omega})$. Предположим, $l(\hat{\pi}, \hat{\omega}) = \theta_0 a_1 \theta_1 \dots a_m \theta_m$ для $m \geq 0$. Тогда,

по лемме 3, существует последовательность $\pi = \pi^0 \xrightarrow{\theta_0} \tilde{\pi}^0 \xrightarrow{a_1} \pi^1 \xrightarrow{\theta_1} \dots \xrightarrow{a_m} \pi^m \xrightarrow{\theta_m} \tilde{\pi}^m = \tilde{\pi}$. Покажем, индукцией по $0 \leq j \leq m$, что существуют тройки $(\pi^j, \pi^{j'}, f^j), (\tilde{\pi}^j, \tilde{\pi}^{j'}, f^j) \in R_2$ такие, что $\pi' \rightarrow \pi^{j'}$, $\pi' \rightarrow \tilde{\pi}^{j'}$ и $f \subseteq f^j$.

($j = 0$) Пусть $\pi^{0'} = \pi'$. Тогда $(\pi^0, \pi^{0'}, f^0) \in R_2$, $\pi' \rightarrow \pi^{0'}$, где $f^0 = f$.

Так как $\pi^0 \xrightarrow{\theta_0} \tilde{\pi}^0$, то $(\pi^0, \omega) \xrightarrow{\theta_0} (\tilde{\pi}^0, \omega)$, где $((\pi^0, \omega), (\pi^{0'}, \omega')) \in R_1$. Кроме того, существует пара $((\tilde{\pi}^0, \omega), (\tilde{\pi}^{0'}, \omega')) \in R_1$ такая, что $(\pi^{0'}, \omega') \xrightarrow{\theta_0} (\tilde{\pi}^{0'}, \omega')$, по определению 15. Из $f^0 : \eta(TN^0) \simeq \eta(TN^{0'})$ следует $f^0 : \eta(\widehat{TN}^0) \simeq \eta(\widehat{TN}^{0'})$. Значит, $\pi' = \pi^{0'} \rightarrow \tilde{\pi}^{0'}$ и $(\tilde{\pi}^0, \tilde{\pi}^{0'}, f^0) \in R_2$, поскольку $((\tilde{\pi}^0, \omega), (\tilde{\pi}^{0'}, \omega')) \in R_1$ и $f^0 = f$.

($j > 0$) По предположению индукции, существует $(\tilde{\pi}^{j-1}, \tilde{\pi}^{j-1'}, f^{j-1}) \in R_2$ такая, что $\pi' \rightarrow \tilde{\pi}^{j-1'}$ и $f \subseteq f^{j-1}$. Тогда $f^{j-1} : \widehat{TN}^{j-1} \simeq \widehat{TN}^{j-1'}$, существует $((\tilde{\pi}^{j-1}, \omega^{j-1}), (\tilde{\pi}^{j-1'}, \omega^{j-1'})) \in R_1$, где $\omega^{j-1} = e^1 \dots e^k$, $\omega^{j-1'} = e^{1'} \dots e^{k'}$ для некоторого $k \geq 0$ и $f^{j-1}(e^i) = e^{i'}$ для $0 < i \leq k$.

Поскольку $\tilde{\pi}^{j-1} \xrightarrow{a_j} \pi^j$, то $(\tilde{\pi}^{j-1}, \omega^{j-1}) \xrightarrow{e} (\pi^j, \omega^{j-1}e)$ для некоторого события e . Благодаря определению 15, существует $((\pi^j, \omega^{j-1}e), (\pi^{j'}, \omega^{j-1}e')) \in R_1$ такая, что $(\tilde{\pi}^{j-1'}, \omega^{j-1'}) \xrightarrow{e'} (\pi^{j'}, \omega^{j-1}e')$, и изоморфизм $\hat{f} : \{e\} \rightarrow \{e'\}$.

Покажем, что $f^j = (f^{j-1} \cup \hat{f}) : TN^j \simeq TN^{j'}$. Имеем, $e^i \preceq^j e^l \iff f^{j-1}(e^i) \preceq^{j'} f^{j-1}(e^l) \iff f^j(e^i) \preceq^{j'} f^j(e^l) \iff e^{i'} \preceq^{j'} e^{l'}$, $\mathbf{Age}^j(e^i) = \mathbf{Age}^{j'}(f^{j-1}(e^i)) = \mathbf{Age}^{j'}(f^j(e^i))$, $l^j(e^i) = l^{j'}(f^{j-1}(e^i)) = l^{j'}(f^j(e^i))$ для $0 < i, l \leq k$. Кроме того, $\mathbf{Age}^j(e) = \mathbf{Age}^{j'}(\hat{f}(e)) = \mathbf{Age}^{j'}(f^j(e))$, $l^j(e) = l^{j'}(\hat{f}(e)) = l^{j'}(f^j(e))$ и $\mathbf{Age}^j(\text{end}^j) = \mathbf{Age}^{j'}(\text{end}^{j'})$. Предположим обратное, т. е. f^j не является изоморфизмом. Тогда существует $1 \leq l \leq k$ такой, что $(e^l \prec e) \wedge \neg(e^{l'} \prec e')$ либо $\neg(e^l \prec e) \wedge (e^{l'} \prec e')$, поскольку $\neg(e \prec e^i) \wedge \neg(e' \prec e^{i'})$ для всех $0 < i \leq k$, по определению графика. Без потери общности будем считать, что $(e^l \prec e) \wedge \neg(e^{l'} \prec e')$ и l — максимальный индекс, удовлетворяющий данному условию. Рассмотрим префикс для $(\pi^j, \omega^{j-1}e)$ вида $(\pi_{l-1}, \omega_{l-1}) \xrightarrow{(\tilde{\pi}, \tilde{\omega})} (\pi^j, \omega^{j-1}e)$, где $\tilde{\omega} = e^1 \dots e^k e$. По определению 15, существует $((\pi_{l-1}, \omega_{l-1}), (\pi'_{l-1}, \omega'_{l-1})) \in R_1$ такая, что $(\pi'_{l-1}, \omega'_{l-1}) \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\pi^{j'}, \omega^{j-1}e')$ и $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$, так как $((\pi^j, \omega^{j-1}e), (\pi^{j'}, \omega^{j-1}e')) \in R_1$. Из изоморфизма следует, что множество событий \widehat{TN}' имеет вид $\widehat{E}' = \{e^{l'} \dots e^{k'} e'\}$, поскольку события \widehat{TN} представлены множеством $\widehat{E} = \{e^1 \dots e^k e\}$ и $\tilde{\omega}'$ — постфикс $\omega^{j-1}e'$. С другой стороны, не существует биективного отображения $\hat{f} : \widehat{E} \rightarrow \widehat{E}'$ такого, что $\hat{e}_1 \preceq \hat{e}_2 \iff \hat{f}(\hat{e}_1) \preceq' \hat{f}(\hat{e}_2)$, так как $|\preceq'| < |\preceq|$. Получили противоречие с тем, что $\eta(\widehat{TN}) \simeq \eta(\widehat{TN}')$. Значит, $f^j : TN^j \simeq TN^{j'}$. Тогда $((\pi^j, \omega^{j-1}e), (\pi^{j'}, \omega^{j-1}e')) \in R_1$, $\omega^{j-1}e = e^1 \dots e^k e$, $\omega^{j-1}e' = e^{1'} \dots e^{k'} e'$, $f^j(e^i) = e^{i'}$ для $0 < i \leq k$ и $f^j(e) = e'$. Следовательно $(\pi^j, \pi^{j'}, f^j) \in R_2$, $\pi' \rightarrow \pi^{j'}$ и $f \subseteq f^{j-1} \subseteq f^j$.

Так как $\pi^j \xrightarrow{\theta_j} \tilde{\pi}^j$, то $(\pi^j, \omega^{j-1}e) \xrightarrow{\theta_j} (\tilde{\pi}^j, \omega^{j-1}e)$, где $((\pi^j, \omega^{j-1}e), (\pi^{j'}, \omega^{j-1}e')) \in R_1$. Кроме того, существует пара $((\tilde{\pi}^j, \omega^{j-1}e), (\tilde{\pi}^{j'}, \omega^{j-1}e')) \in R_1$ такая, что $(\pi^{j'}, \omega^{j-1}e') \xrightarrow{\theta_j} (\tilde{\pi}^{j'}, \omega^{j-1}e')$, по определению 15. Из $f^j : \eta(TN^j) \simeq \eta(TN^{j'})$ следует $f^j : \eta(\widehat{TN}^j) \simeq \eta(\widehat{TN}^{j'})$. Значит, $\pi' \rightarrow \tilde{\pi}^{j'}$ и $(\tilde{\pi}^j, \tilde{\pi}^{j'}, f^j) \in R_2$.

Следовательно, существует тройка $(\tilde{\pi}, \tilde{\pi}', \tilde{f}) = (\tilde{\pi}^m, \tilde{\pi}^{m'}, f^m) \in R_2$ такая, что $\pi' \rightarrow \tilde{\pi}'$ и $f \subseteq \tilde{f}$.

(3) Как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Согласно пунктам (1–3) и определению 14, $\mathcal{TN} \stackrel{h}{\underset{por}{\rightleftharpoons}} \mathcal{TN}'$.

(в) $\mathcal{TN} \approx_{prcprc} \mathcal{TN}' \Rightarrow \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$.

Пусть $R : \mathcal{TN} \approx_{prcprc} \mathcal{TN}'$. Покажем, что $R : \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$.

Согласно определению 15, $((\pi_0, \varepsilon), (\pi'_0, \varepsilon)) \in R$.

Рассмотрим произвольную пару $((\pi, \omega), (\pi', \omega')) \in R$.

- (1) Так как $(\pi_0, \varepsilon) \xrightarrow{(\pi, \omega)} (\pi, \omega)$ то, существует пара $((\pi_0, \varepsilon), (\bar{\pi}', \bar{\omega}')) \in R$ такая, что $(\bar{\pi}', \bar{\omega}') \xrightarrow{(\tilde{\pi}', \tilde{\omega}')} (\pi', \omega')$ и $TN \simeq \widehat{\overline{TN}}$, по определению 15(3). Благодаря лемме 4, $(\bar{\pi}', \bar{\omega}') = (\pi'_0, \varepsilon)$. Значит, $(\tilde{\pi}', \tilde{\omega}') = (\pi', \omega')$ и $TN \simeq TN'$.
- (2) Если $(\pi, \omega) \rightarrow (\tilde{\pi}, \tilde{\omega})$ для $(\tilde{\pi}, \tilde{\omega}) \in \mathcal{CHR}(\mathcal{TN})$, тогда существует пара $((\tilde{\pi}, \tilde{\omega}), (\tilde{\pi}', \tilde{\omega}')) \in R$ такая, что $(\pi', \omega') \rightarrow (\tilde{\pi}', \tilde{\omega}')$, согласно определению 15(1).
- (3) Как пункт (2), но роли \mathcal{TN} и \mathcal{TN}' меняются.

Следовательно, $R : \mathcal{TN} \approx_{prc}^{wh} \mathcal{TN}'$, согласно пунктам (1–3) и определению 16.

Пример 15. Рассмотрим НВСП \mathcal{TN}^{15} и \mathcal{TN}^{16} . Покажем, что $\mathcal{TN}^{15} \not\stackrel{h}{\underset{prc}{\rightleftharpoons}} \mathcal{TN}^{16}$. Предположим это не так, т.е. существует некоторая бисимуляция R , которая удовлетворяет определению 14. Тогда $(\pi_0, \pi'_0, f_0) \in R$. Рассмотрим π_1 — расширение временного процесса π_0 на действие a . Согласно определению 14, существует тройка $(\pi_1, \pi'_1, f_1) \in R$, где π'_1 является расширением π'_0 на действие a и $f_1 : TN_1 \simeq TN'_1$.

Пусть b_i — условие в π_1 , соответствующее месту p_i НВСП \mathcal{TN}^{15} , а b'_i — условие в π'_1 , соответствующее месту p_i НВСП \mathcal{TN}^{16} , для $1 \leq i \leq 5$. Тогда либо $f_1(b_4) = b'_5$, либо $f_1(b_4) = b'_3$.

- (1) Предположим $f_1(b_4) = b'_5$. Временной процесс π_1 имеет расширение π_2 на действие c после задержки в 2 единицы времени. При этом условие b_4 будет входным для соответствующего события. Согласно определению 14, в этом случае, должна существовать тройка $(\pi_2, \pi'_2, f_2) \in R$, где π'_2 — расширение на действие c , $f_1 \subseteq f_2$ и $f_2 : TN_2 \simeq TN'_2$. Значит, $f_2(b_4) = f_1(b_4) = b'_5$ и условие b'_5 также должно быть входным для события действия c . Как можно заметить, требуемого расширения π'_2 не существует, получили противоречие.
- (2) Предположим $f_1(b_4) = b'_3$. Временной процесс π_1 имеет расширение π_2 на действие b . При этом условие b_4 будет входным для соответствующего события. Согласно определению 14, в этом случае, должна существовать тройка $(\pi_2, \pi'_2, f_2) \in R$, где π'_2 — расширение на действие b , $f_1 \subseteq f_2$ и $f_2 : TN_2 \simeq TN'_2$. Значит, $f_2(b_4) = f_1(b_4) = b'_3$ и условие b'_3 также должно быть входным для события действия b . Как можно заметить, требуемого расширения π'_2 не существует, получили противоречие.

Оба варианта привели к противоречию. Следовательно, бисимуляции R не существует и $\mathcal{TN}^{15} \not\stackrel{h}{\underset{prc}{\rightleftharpoons}} \mathcal{TN}^{16}$. С другой стороны, можно проверить, что $\mathcal{TN}^{15} \approx_{prcprc} \mathcal{TN}^{16}$.

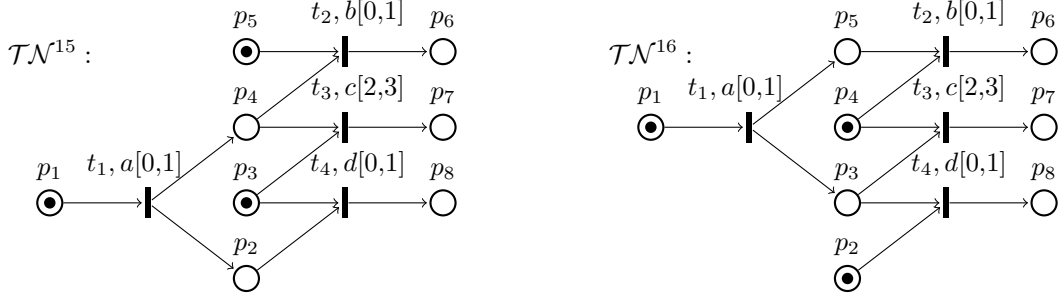


Рис. 11. \mathcal{TN}^{15} и \mathcal{TN}^{16}

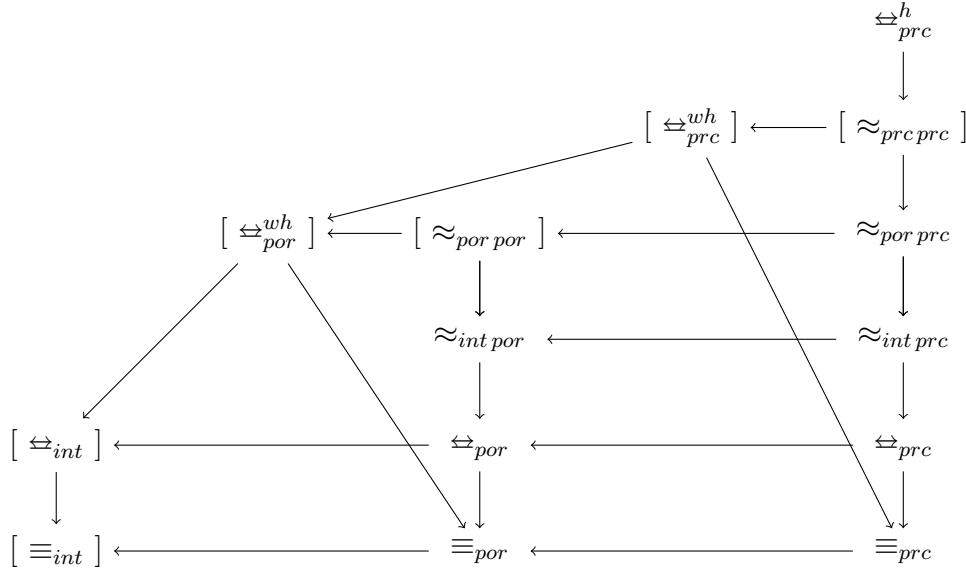


Рис. 12. Иерархия эквивалентностей

Сформулируем следствие, которое определяет иерархию определенных в работе эквивалентностей. Данное утверждение следует из определений, примеров и теорем, рассмотренных ранее.

Следствие 1. Пусть $\mathcal{TN}, \mathcal{TN}'$ — НВСП, помеченные над Act. Для эквивалентностей R и \tilde{R} из определений 9–16 верно, что

$$(\mathcal{TN})R(\mathcal{TN}') \Rightarrow (\mathcal{TN})\tilde{R}(\mathcal{TN}')$$

тогда и только тогда, когда в графе на рисунке существует путь от R к \tilde{R} , где $[\cong_{int}^wh] = \{ \cong, \equiv_{int} \}$, $[\cong_{int}^wh] = \{ \sim, \cong_{int}, \approx_{int}^wh \}$, $[\cong_{por}^wh] = \{ \cong_{por}^wh, \approx_{por}^wh, \approx_{por}^wh \}$, $[\cong_{prc}^wh] = \{ \cong_{prc}^wh, \approx_{prc}^wh, \approx_{prc}^wh, \approx_{por}^wh, \approx_{prc}^wh \}$, $[\approx_{por}^wh] = \{ \cong_{por}^h, \approx_{por}^wh, \approx_{por}^wh \}$, $[\approx_{prc}^wh] = \{ \approx_{prc}^wh, \approx_{prc}^wh, \approx_{prc}^wh \}$ — множества совпадающих эквивалентностей.

Заключение. В данной работе было показано, что ряд эквивалентностей в спектрах «линейное — ветвящееся время» и «интерливинг — частичный порядок» могут быть обобщены на модель непрерывно-временных сетей Петри со слабой стратегией хода времени и устойчиво атомарным способом сброса часов. В частности, были разработаны и изучены

языковые эквивалентности, обычные бисимуляционные эквивалентности, эквивалентности с сохранением и слабым сохранением истории, прямые-обратные бисимуляционные эквивалентности. В ходе исследований были использованы следующие представления выполнений: пробеги (последовательности срабатываний переходов и хода времени), определяющие интерливинговую семантику, и временные процессы, лежащие в основе истинно-параллельных семантик. Были установлены взаимосвязи между пробегами и временными процессами. Показано, что интерливинговые эквивалентности слабее, чем эквивалентности, основанные на ВЧУМ и ВПСС временных процессов. Было установлено, что языковые эквивалентности слабее бисимуляционных. Исследовано влияние учета двух направлений моделирования (прямого и обратного) и истории функционирования системы на бисимуляционные эквивалентности. Основным результатом работы является построение иерархии разработанных эквивалентностей.

В качестве дальнейшей работы планируется обобщить полученные результаты для шаговой семантики, исследовать другие эквивалентности в спектре «линейное — ветвящееся время», изучить взаимосвязи между эквивалентностями НВСП с сильной и слабой семантиками.

Список литературы

1. Bérard B., Cassez F., Haddad S., Lime D., Roux O. H. Comparison of different semantics for time Petri nets // International Symposium on Automated Technology for Verification and Analysis. 2005. P. 293–307.
2. Reynier P. A., Sangnier A. Weak time Petri nets strike back! // International Conference on Concurrency Theory. 2009. P. 557–571.
3. Boyer M., Roux O. H. Comparison of the expressiveness of arc, place and transition time Petri nets // International Conference on Application and Theory of Petri Nets. 2007. P. 63–82.
4. Тарасюк И. В. Эквивалентности для поведенческого анализа параллельных и распределенных вычислительных систем. Академическое издательство Гео, 2007.
5. Virbitskaite I., Bushin D., Best E. True concurrent equivalences in time Petri nets // Fundamenta Informaticae, 149(4), 2016. P. 401–418
6. Вирбицкайте И. Б., Зубарев А. Ю. «Истинно параллельная» семантика непрерывно-временных сетей Петри со слабой временной и устойчиво атомарной пространственной стратегиями // Программирование, 2021, № 5. С. 60–74.
7. Зубарев А. Ю. Сравнение языковых и бисимуляционных эквивалентностей непрерывно-временных сетей Петри со слабой временной стратегией // Проблемы информатики. 2022. № 4. С. 5–27.



Алексей Юрьевич Зубарев — младш. науч. сотр. Института систем информатики им. А. П. Ершова (Новосибирск). E-mail: a Zubarev@gmail.com. Магистр по направлению «Математика и компьютерные науки» Новосибирского государственного университета. Область научных интересов:

теория параллельных процессов; спецификация

и верификация параллельных систем реального времени.

Alexey Yurievich Zubarev — junior research fellow A. P. Ershov Institute of Informatics Systems (Novosibirsk, Russia). E-mail: a Zubarev@gmail.com. Educational degrees: Master's degree in computer science, Novosibirsk State University. Research interests: theory of parallel processes, specification and verification of parallel real-time systems.

PARALLEL NUMERICAL METHOD FOR SOLUTION OF HYDRODYNAMIC EQUATIONS IN THE SHALLOW WATER APPROXIMATION FOR SHARED MEMORY COMPUTERS

A. V. Starchenko

Tomsk State University,
634050, Tomsk, Russia
Federal Research Center for informational and computational technologies,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2024-1-41-56

EDN: JVZEVV

Modeling of natural phenomena such as tsunamis, floods, ocean and river currents, and dam breaks are pressing environmental problems. Due to the significant difference in the vertical and horizontal dimensions of the study area, non-stationary two-dimensional hydrodynamic equations in the shallow water approximation have become very popular among researchers of the processes under consideration using mathematical modeling methods, which makes it possible to significantly simplify the mathematical formulation of the problems under consideration. Note, that the accuracy of numerical prediction using such equations depends on a number of conditions, among which the main ones are the spatial resolution of the study area and the quality of the selected numerical methods. However, increasing the spatial resolution significantly increases the time required to obtain a numerical solution, since in the numerical modeling of processes in the environment, as a rule, explicit or semi-implicit difference schemes are used with a limitation on the time step, depending on the size of the spatial grid steps. That is, as a result, due to the need to carry out calculations with a large number of grid nodes and a smaller time step, calculations on a computer with a sequential architecture can take quite a long time.

The aim of this work is to formulate a mathematical problem of unsteady isothermal turbulent flows in river, based on the shallow water approximation, to construct an effective conservative numerical method, and to develop parallel computing algorithms for multi-core computing systems with shared RAM.

The turbulent isothermal motion of an incompressible Newtonian fluid in a river flow is considered. The study area is an open river channel with islands and irregular bottom topography. Current characteristics can vary significantly over time, and therefore the current is considered unsteady with potential flooding of coastal areas. The movement of water in a river is determined by the forces of gravity and friction. In addition, the influence of the Coriolis force and turbulent diffusion on the flow is taken in to account. It is assumed that the horizontal dimensions of the region significantly exceed the vertical ones, the Reynolds-averaged flow characteristics change slightly in the vertical direction, and the vertical pressure distribution is hydrostatic. The thermophysical properties of water (viscosity, density) are considered constant. For numerical modeling of unsteady isothermal turbulent flows in river flows, a mathematical model is formulated based on the shallow water approximation for the Reynolds equations for an incompressible fluid. To take into account the transport, generation, diffusion and dissipation of turbulent vortices, this work uses the $k - \varepsilon$ turbulence model constructed by Rastogi

and Rodi from the original $k - \varepsilon$ model of Launder and Spalding to close the Reynolds shallow water equations.

To construct a discrete analogue of the developed mathematical model, the rectangular computational domain in which the flow with variable boundaries is studied is covered with a structured mesh with steps $\Delta x, \Delta y$, respectively. According to the concept of the finite volume method, each internal mesh node appears in a separate finite volume. In this case, the values of the flow depth, bottom topography (and turbulence model parameters) are determined at the nodes of the computational grid, and the velocity vector components are determined at the midpoints of the corresponding faces of the finite volumes. The differential equations of the model are integrated over each finite volume. The equations of motion are approximated in cells shifted by $\Delta x/2, \Delta y/2$, respectively. For approximate calculation of integrals, quadrature formulas of average rectangles are used, derivatives are approximately calculated using central-difference formulas. To calculate fluxes on the faces of finite volumes, monotonized upstream van Leer approximations [1] with a limiter [2] are used. When approximating equations in time, conditionally stable semi-implicit difference schemes are used, which are conservative not only for the continuity equation, but also for the equations of motion, which is very important for obtaining non-negative values of flow depth [3]. As a result, semi-implicit difference schemes of first order approximation in time and second order in spatial coordinates are obtained.

The classical two-dimensional Thacker problem of fluid oscillations without friction in a reservoir whose bottom is a paraboloid was considered as a problem on which the accuracy of calculations and the quality of parallelization were tested. Satisfactory agreement between the calculated values of the flow depth and the analytical solution was obtained.

The constructed semi-implicit difference scheme on staggered difference grids has great potential for creating an efficient parallel program, since at each time step, calculations of depth or velocity components can be performed simultaneously and independently in each computational node of the grid. On a server with a total memory of 192 GB and two twelve-core Intel Xeon Silver 4214 2.2 GHz processors and one NVIDIA GeForce RTX2080 Ti graphics card, a study was conducted of the effectiveness of parallel programs built using low-cost parallel programming technologies Open Multi-Processing and Open Accelerators. In these programs, using OpenMP or OpenACC parallel technologies, the execution of iterations of nested loops was evenly distributed among the active threads/processes of the central or graphic processor cores.

The calculation using a sequential program for the conditions under consideration on the server was 547.7 s (500×500 grid) and 4606.4 s (1000×1000 grid). Parallel calculations showed that the use of OpenMP technology for two twelve-core central processors makes it possible to speed up the computing process by more than 15 times. Using OpenACC technology when calculating on the same multi-core system and the NVIDIA GeForce RTX2080 Ti graphics processor gives a speedup of more than 25.

Key words: parallel computations, shallow water equations, shared memory computers, OpenMP, OpenACC.

References

1. Saint-Venant A. J. B. Theorie du Mouvement non permanent des Eaux // Institut de France, Acad. des Sci. de Paris. 1871. 73 (3) 147; 73 (4) 237.
2. Gusev O. I., Khakimzyanov G. S., Skiba V. S., Chubarov L. B. Shallow water modeling of wave-structure interaction over irregular bottom // Ocean Engineering. 2023. V. 267. Art. 113284.
3. Brand S. Parallel algorithm for numerical solution of the shallow water equation // Proceedings of the Czech-Japanese Seminar in Applied Mathematics. 2006. Czech Technical University in Prague, September 14–17, 2006. P. 25–36.
4. Gabdullina M. I. Parallel algorithm for numerical solution of the two-layer shallow water model in 2D // Russian Supercomputing Days 2016 // RussianSCDays.org. P. 940–947.

5. Chaplygin A.V., Gusev A.V. Shallow water model using a hybrid MPI/OpenMP parallel programming // *Problems of informatics*, 2021. N 1. P. 65–82.
6. Juliati S., Gunawan P.H. OpenMP architecture to simulate 2D water oscillation on paraboloid // *5th International Conference on Information and Communication Technology (ICoIC7)*, 2017. P. 1–5.
7. Liu Q., Qin Y., Li G. Fast Simulation of Large-Scale Floods Based on GPU Parallel Computing // *Water*. 2018. V. 10(5):589.
8. Zhang S., Li W., Jing Zh, Yi Y., Zhao Y. Comparison of Three Different Parallel Computation Methods for a Two-Dimensional Dam-Break Model // *Mathematical Problems in Engineering*. 2017. V. 2017, Article ID 1970628, 12 pages.
9. Arnoldy A., Adytia D. Performance of Staggered Grid Implementation of 2D Shallow Water Equations using CUDA Architecture // *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, 2019, P. 286–290.
10. Churuksaeva V., Starchenko A. Mathematical modeling of a river stream based on a shallow water approach // *Procedia Computer Science*. 2015. V. 66. P. 200–209.
11. Rastogi A.K., Rodi W. Predictions of heat and mass transfer on open channels // *ASCE Journal of the Hydraulics Division*. 1978. V. 104(3). P. 397–420.
12. Launder B.E., Spalding D.B. The numerical computation of turbulent flows // *Computer Methods in Applied Mechanics and Engineering*. 1974. V. 2(3). P. 269–289.
13. Stelling G.S., Duimijer S.P.A. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows // *Int. Journal for Numerical Methods in Fluids*. 2003. V. 43. P. 1329–1354.
14. van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method // *Journal of Computational Physics*. 1979. V. 32(1). P. 101–136.
15. Cada M., Torrilhon M. Compact third-order limiter functions for finite volume methods // *Journal of Computational Physics*. 2009. V. 228. P. 4118–4145.
16. Thacker W.C. Some exact solutions to the nonlinear shallow water wave equations // *Journal of Fluid Mechanics*. 1981. V. 107. P. 499–508.

ПАРАЛЛЕЛЬНЫЙ ЧИСЛЕННЫЙ МЕТОД РЕШЕНИЯ ГИДРОДИНАМИЧЕСКИХ УРАВНЕНИЙ В ПРИБЛИЖЕНИИ МЕЛКОЙ ВОДЫ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ОБЩЕЙ ПАМЯТЬЮ

А. В. Старченко

Томский государственный университет,
634050, Томск, Россия
ФИЦ информационных и вычислительных технологий,
630090, Новосибирск, Россия

УДК 519.63, 519.683, 519.688

DOI: 10.24412/2073-0667-2024-1-41-56

EDN: JVZEVV

Для численного моделирования нестационарных изотермических турбулентных течений в речных потоках сформулирована математическая модель, опирающаяся на приближение мелкой воды для уравнений Рейнольдса для несжимаемой жидкости, эффективный численный метод, обеспечивающий в рамках использования метода конечного объема, структурированных разнесенных сеток и полунеявных разностных схем выполнение на разностном уровне законов сохранения массы и импульса. Вычислительная реализация предложенной модели и метода была протестирована на аналитическом решении Такера и распараллелена с помощью технологий OpenMP и OpenACC на гибридной многоядерной системе с общей памятью. Расчеты показали, что использование технологии OpenMP для двух двенадцатиядерных центральных процессоров позволяет более чем в 15 раз ускорить вычислительный процесс. Использование технологии OpenACC при расчетах на этой же многоядерной системе и графическом процессоре NVIDIA GeForce RTX2080Ti дает ускорение более чем в 25.

Ключевые слова: параллельные вычисления, уравнения мелкой воды, системы с общей памятью, OpenMP, OpenACC.

Введение. Моделирование таких природных явлений как цунами, наводнения, океанические и речные течения, прорыв дамбы являются актуальными проблемами окружающей среды. В силу существенного различия в вертикальных и горизонтальных размерах области большую популярность у исследователей рассматриваемых процессов с помощью методов математического моделирования приобрели нестационарные двумерные гидродинамические уравнения в приближении мелкой воды [1], позволяющие значительно упростить математическую постановку проблем. Заметим, однако, что точность численного прогноза с помощью таких уравнений зависит от ряда условий, среди которых основными являются пространственное разрешение области исследования и качество выбранных численных методов. Однако повышение пространственного разрешения существенно увеличивает временные затраты на получение численного решения, поскольку при численном моделировании процессов в окружающей среде, как правило, используются явные или полунеявные разностные схемы с ограничением на шаг по времени, зависящим от

размера шагов пространственной сетки. То есть в итоге из-за необходимости вести вычисления с большим количеством узлов сетки и меньшей величиной шага по времени расчеты на компьютере с последовательной архитектурой могут занимать достаточно долгое время [2–3]. Для ускорения вычислений с высоким пространственным разрешением целесообразно использовать параллельные компьютерные технологии, которые очень важны и крайне необходимы. В настоящее время на современных многоядерных многопроцессорных вычислительных системах наиболее популярными являются следующие технологии [3–9]: Message Passing Interface (MPI), Open Multi-Processing (OpenMP), OpenAccelerators (OpenACC), Compute Unified Device Architecture (CUDA). С учетом этого параллельная реализация численных моделей, основанных на уравнениях мелкой воды, может быть разделена на три категории:

- модели, ориентированные на массивные вычислительные системы с многоядерными узлами и распределенной памятью, используют, как правило, технологии параллельного программирования MPI или MPI+OpenMP [3, 5, 8];

- модели, ориентированные на многоядерные вычислительные системы с общей оперативной памятью, используют технологию OpenMP [3, 6, 8];

- модели, ориентированные на гибридные вычислительные системы с центральными и графическими многоядерными процессорами (CPU+GPU) и общей оперативной памятью, используют технологии CUDA или OpenACC [7–9].

При использовании вычислительных систем с общей оперативной памятью обмен данными между параллельно исполняемыми нитями программы осуществляется путем чтения и записи информации прямо в оперативной памяти, что позволяет легче решать проблемы балансировки общей вычислительной нагрузки между активными нитями. В технологии OpenMP основным средством балансировки нагрузки является распределение итераций вложенных циклов между активными нитями/процессами. Параллельные программы, развиваемые на основе этого подхода, имеют преимущества в простоте разработки кода, хорошей масштабируемости и переносимости. Например, в [3] уравнения мелкой воды решаются методами конечных разностей и конечных объемов в рамках схем Лакса-Фридрикса, Лакса-Вендроффа и Маккормака. Алгоритмы распараллелены с использованием OpenMP и MPI. Результаты измерения эффективности параллельных программ показывают, что OpenMP, так же как MPI подходят для распараллеливания программ численного решения уравнения мелкой воды, поскольку значения эффективности для сеток 400×400 , ..., 2000×2000 не опускались ниже 85 % во время тестирования на относительно небольшом количестве ядер (4 и больше).

В MPI-параллельных программах численного решения уравнений мелкой воды распределение вычислений между отдельными многопроцессорными многоядерными вычислительными узлами с локальной оперативной памятью, связанными компьютерной сетью, целиком лежит на разработчике параллельного программного обеспечения. Это с одной стороны значительно повышает трудоемкость программирования, с другой — открывает дополнительные возможности масштабируемости параллельных приложений за счет возможности привлечения большего количества вычислительных ресурсов (CPUs), чем в случае вычислительных систем с общей памятью. Заметим, однако, что такие вычислительные системы кластерного типа встречаются гораздо реже, чем сервера с 1–2 CPU и общей оперативной памятью. В последнее время для моделирования океанических течений в приближении мелкой воды были получены успешные результаты в совместном привлечении двух технологий параллельного программирования: MPI+OpenMP. Напри-

мер, в [5] был разработан гибридный задачный MPI+OpenMP подход решения уравнений мелкой воды с помощью аппроксимаций второго порядка на разнесенных сетках и явной численной схемы «чехарда», обеспечивающий на сетке 1525×1115 эффективность более 90 % (несколько выше, чем чистый MPI) при использовании 144 вычислительных ядер.

Среди всех параллельных программ, ориентированных на вычислительные системы с общей памятью, программы, использующие графические процессоры, дают наибольшее ускорение в параллельных вычислениях [7–9]. Наличие большего числа (до нескольких тысяч) производительных для расчетов с плавающей точкой графических процессоров открывает большие возможности для сокращения времени проведения большого объема вычислений [7–9]. Трудоемкость программирования в рамках технологии OpenACC незначительно отличается от трудоемкости использования технологии OpenMP, дополнительной необходимостью является обеспечение передачи информации между общей памятью компьютера и памятью графической карты. Несколько выше трудоемкость программирования CUDA-приложений, поскольку в ней подобно MPI, распределение вычислений в большей степени лежит на разработчике параллельной программы. Тем не менее, в [7] использование в вычислениях графической карты NVIDIA Tesla K20 и технологии OpenACC при моделировании реальных наводнений в Китае на треугольной сетке с количеством узлов 2868736 удалось достичь ускорения в 31 раз по сравнению с последовательными вычислениями на CPU Intel Xeon E5-2690 @ 3.0 GHz.

Целью данной работы является формулировка математической постановки задачи о нестационарных изотермических турбулентных течениях в речных потоках, опирающейся на приближение мелкой воды, построение эффективного консервативного численного метода, разработка параллельных вычислительных алгоритмов для многоядерных вычислительных систем с общей оперативной памятью.

1. Постановка задачи. Рассматривается турбулентное изотермическое движение несжимаемой ньютоновской жидкости в речном потоке. Область исследования представляет собой участок открытого речного русла с островами и нерегулярным рельефом дна. Характеристики течения могут значительно меняться со временем, и потому течение считается нестационарным. Движение воды в реке определяется силами гравитации и трения. Кроме того, учитывается влияние на течение силы Кориолиса и турбулентной диффузии.

Предполагается, что горизонтальные размеры области существенно превосходят вертикальные, осредненные по Рейнольдсу характеристики течения слабо меняются в вертикальном направлении, а вертикальное распределение давления является гидростатическим. Теплофизические свойства воды (вязкость, плотность) считаются постоянными.

1.1. *Система уравнений и краевые условия.* Математическая модель в приближении мелкой воды [10] включает уравнение неразрывности:

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0, \quad (1)$$

уравнения движения:

$$\begin{aligned} \frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2)}{\partial x} + \frac{\partial(huv)}{\partial y} &= -gh \frac{\partial(z_b + h)}{\partial x} + \frac{1}{\rho} \frac{\partial(h\tau_{xx})}{\partial x} + \frac{1}{\rho} \frac{\partial(h\tau_{xy})}{\partial y} + \\ &+ \frac{(\tau_{xz})_s - (\tau_{xz})_b}{\rho} - hF_x, \\ \frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2)}{\partial y} &= -gh \frac{\partial(z_b + h)}{\partial y} + \frac{1}{\rho} \frac{\partial(h\tau_{yx})}{\partial x} + \frac{1}{\rho} \frac{\partial(h\tau_{yy})}{\partial y} + \\ &+ \frac{(\tau_{yz})_s - (\tau_{yz})_b}{\rho} - hF_y. \end{aligned} \quad (2)$$

Здесь $h(t,x,y)$ — глубина; $u(t,x,y)$, $v(t,x,y)$ — осредненные по глубине значения компонент вектора скорости; $z_b(x,y)$ — рельеф дна; ρ — плотность воды, g — ускорение свободного падения; τ_{xx} , τ_{xy} , τ_{yx} , τ_{yy} — осредненные по глубине компоненты тензора вязких напряжений и напряжений Рейнольдса; $(\tau_{xz})_s$, $(\tau_{xz})_b$, $(\tau_{yz})_s$, $(\tau_{yz})_b$ — трение на поверхности воды и на дне, соответственно. Предполагается, что в рассматриваемом случае ветровое трение не оказывает на речное течение значительного влияния и потому не учитывается в модели. Осредненные по глубине компоненты силы Кориолиса определяются следующим образом:

$$F_x = (-4\pi/\tau) \sin(lat)v, F_y = (4\pi/\tau) \sin(lat)u,$$

где lat — географическая широта, τ — продолжительность суток в секундах.

Для получения неизвестных значений компонент тензора напряжений в уравнениях модели используются градиентные соотношения Буссинеска:

$$\begin{aligned} \frac{\tau_{xy}}{\rho} = \frac{\tau_{yx}}{\rho} &= (\nu + \nu_t) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \\ \frac{\tau_{xx}}{\rho} &= 2(\nu + \nu_t) \frac{\partial u}{\partial x} - \frac{2}{3}k, \\ \frac{\tau_{yy}}{\rho} &= 2(\nu + \nu_t) \frac{\partial v}{\partial y} - \frac{2}{3}k, \end{aligned} \quad (3)$$

где ν — молекулярная кинематическая вязкость воды, ν_t — турбулентная вязкость, k — кинетическая энергия турбулентности.

Трение о дно определяется как:

$$(\tau_{xz})_b = \rho c_f u |\vec{\mathbf{w}}|, (\tau_{yz})_b = \rho c_f v |\vec{\mathbf{w}}|,$$

где c_f — коэффициент трения, зависящий от физических характеристик русла или канала: $c_f = gn^2/h^{0,33}$ определяется из закона трения Маннинга, где n — коэффициент Маннинга, характеризующий шероховатость русла, $|\vec{\mathbf{w}}|$ — модуль вектора скорости.

Для учета переноса, генерации, диффузии и диссипации турбулентных вихрей в данной работе для замыкания уравнений мелкой воды применяется $k - \varepsilon$ модель турбулентности, построенная Растоги и Роди [11] из оригинальной $k - \varepsilon$ модели Лаундера и Сполдинга [12] для замыкания уравнений Рейнольдса.

На входе в расчетную область задаются однородные профили нормальной к границе компоненты скорости потока U_0 , энергии турбулентности k_0 , диссипации ε_0 и глубины

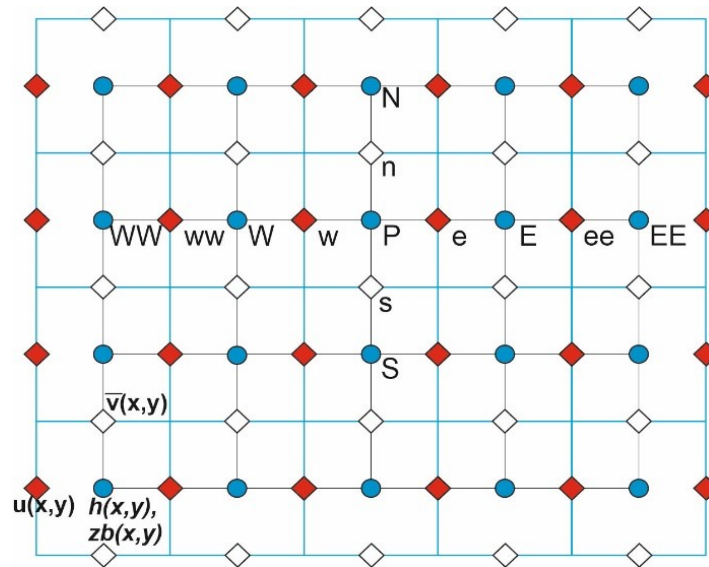


Рис. 1. Расчетная сетка. Круглые значки соответствуют центру конечного объема, заполненные ромбы — компоненте скорости u , открытые ромбы — v

потока h_0 , полученные из эмпирических данных. Поперечная компонента скорости потока на входе задается равной нулю. В качестве граничных условий на выходе из области используется равенство нулю производных по внешней нормали к выходной границе для основных характеристик течения. В начальный момент распределения глубины потока компонент скорости и параметров энергии турбулентности на рассматриваемой области считаются известными.

2. Численный метод. Для построения дискретного аналога разрабатываемой математической модели расчетная область размером $[0, L_x] \times [0, L_y]$, в которой исследуется течение потока с переменными границами, покрывается структурированной сеткой с шагами $\Delta x, \Delta y$, соответственно. Согласно концепции метода конечных объемов, каждый внутренний узел сетки оказывается в отдельном конечном объеме (рис. 1). При этом значения глубины потока, рельефа дна (и параметров модели турбулентности) определяются в узлах расчетной сетки, а компоненты вектора скорости — в серединах соответствующих граней конечных объемов. Для рассматриваемого фрагмента сетки используются следующие обозначения. Каждый конечный объем сетки имеет западную (w), восточную (e), северную (n) и южную (s) границы и центр (P). Центры соседних конечных объемов обозначаются W, E, N, S , соответственно.

Дифференциальные уравнения модели интегрируются по каждому конечному объему. Уравнения движения аппроксимируются в ячейках, сдвинутых на $\Delta x/2$ и $\Delta y/2$, соответственно. Для приближенного вычисления интегралов используются квадратурные формулы средних прямоугольников, производные приближенно вычисляются по центрально-разностным формулам. При аппроксимации уравнений по времени используются условно устойчивые полунеявные разностные схемы, консервативные не только для уравнения неразрывности (1), но и для уравнений движения (2), что весьма важно для получения неотрицательных значений глубины потока h [13].

В итоге получаются полунеявные разностные схемы первого порядка аппроксимации по времени и второго — по пространственным координатам.

$$\frac{h_p^{k+1} - h_p^k}{\Delta t_k} + \frac{G_e^k - G_w^k}{\Delta x} + \frac{G_n^k - G_s^k}{\Delta y} = 0, \quad (4)$$

$$\begin{aligned} & \frac{u_e^{k+1} - u_e^k}{\Delta t_k} + \frac{1}{h_e^k} \left[\frac{q_{xE}^k u_E^k - q_{xP}^k u_P^k}{\Delta x} - u_e^k \frac{q_{xE}^k - q_{xP}^k}{\Delta x} \right] + \\ & + \frac{1}{h_e^k} \left[\frac{q_{yne}^k u_{ne}^k - q_{yse}^k u_{se}^k}{\Delta y} - u_e^k \frac{q_{yne}^k - q_{yse}^k}{\Delta y} \right] + \\ & + g \frac{\xi_E^{k+1} - \xi_P^{k+1}}{\Delta x} + \frac{c_f |\vec{w}_e^k| u_e^{k+1}}{h_e^k} - (D_x)_e^k + (F_x)_e^k = 0, \end{aligned} \quad (5)$$

$$\begin{aligned} & \frac{v_n^{k+1} - v_n^k}{\Delta t_k} + \frac{1}{h_n^k} \left[\frac{q_{xen}^k v_{en}^k - q_{xwn}^k v_{wn}^k}{\Delta x} - v_n^k \frac{q_{xen}^k - q_{xwn}^k}{\Delta x} \right] + \\ & + \frac{1}{h_n^k} \left[\frac{q_{yN}^k v_N^k - q_{yP}^k v_P^k}{\Delta y} - v_n^k \frac{q_{yN}^k - q_{yP}^k}{\Delta y} \right] + \\ & + g \frac{\xi_N^{k+1} - \xi_P^{k+1}}{\Delta x} + \frac{c_f |\vec{w}_n^k| v_n^{k+1}}{h_n^k} - (D_y)_n^k + (F_y)_n^k = 0. \end{aligned} \quad (6)$$

Здесь $\xi = z_b + h$ — уровень поверхности воды, $(D_x)_e^k, (D_y)_n^k$ — аппроксимации диффузионных слагаемых уравнений (2); для вычисления потоков на гранях конечных объемов $G_e = (uh)_e = (q_x)_e, G_n = (vh)_n = (q_y)_n$ используются монотонизированные противопотоковые аппроксимации ван Лира [14] с ограничителем [15]:

$$G_e = \begin{cases} u_e(h_p + 0.5\Psi(\theta^+)(h_E - h_P)), u_e > 0 \\ u_e(h_E + 0.5\Psi(\theta^-)(h_E - h_P)), u_e \leq 0 \end{cases}, \Psi(\theta) = \max\left(0, \min\left(\min\left(2\theta, \frac{\theta+2}{3}\right), 2\right)\right),$$

$$\begin{aligned} \theta_e^+ &= \frac{h_P - h_W}{h_E - h_P}, \\ \theta_e^- &= \frac{h_{EE} - h_E}{h_E - h_P} \end{aligned}$$

$$h_e = \frac{h_E + h_P}{2}, q_{xP} = \frac{q_{xe} + q_{xw}}{2}, q_{xe} = h_P \max(u_e, 0) - h_E \max(-u_e, 0),$$

$$q_{xw} = h_W \max(u_w, 0) - h_P \max(-u_w, 0);$$

$$q_{yne} = \frac{q_{yn} + q_{ys}}{2}; q_{yn} = h_P \max(v_n, 0) - h_N \max(-v_n, 0),$$

$$q_{ys} = h_S \max(v_s, 0) - h_P \max(-v_s, 0).$$

При вычислении потоков — произведений $q_{xE}u_e, q_{yne}u_{ne}, q_{xen}v_{en}, q_{yN}v_N$ также используются аппроксимации ван Лира [14] с ограничителем [15], рассмотренные выше. Для устойчивого по времени интегрирования уравнений (4)–(6) величина шага по времени выбирается из условия

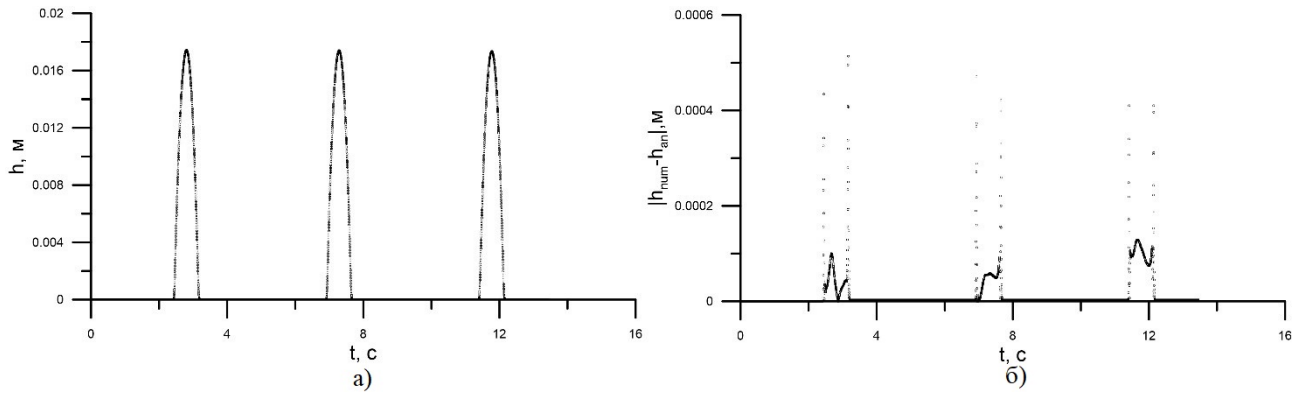


Рис. 2. Сравнение расчетов (открытые кружки) и аналитического решения (линия) для глубины жидкости h (а) и погрешность расчетов (б) для 2D задачи Такера [16] в точке (1.004м; 1.004м) для трех периодов колебаний жидкости

$$\Delta t_k < \frac{\max(\Delta x, \Delta y)}{2 \left(|\vec{w}_P^k| + \sqrt{gh_P^k} \right)}. \quad (7)$$

Такой шаг обеспечивает неотрицательность глубины потока ($h_P^{k+1} \geq 0$).

В качестве задачи, на которой проводилось тестирование точности вычислений и качества распараллеливания, рассматривалась классическая двумерная задача Такера о колебании жидкости без трения в водоеме, дно которого представляет собой параболоид. Аналитическое решение задачи представлено в [16] и имеет вид:

$$\begin{cases} h(x,y,t) = \frac{\eta h_0}{a^2} [2(x - L/2)\cos(\omega t) + 2(y - L/2)\sin(\omega t) - \eta] - z_b(x,y); \\ u(x,y,t) = -\eta\omega \times \sin(\omega t); \\ v(x,y,t) = \eta\omega \times \cos(\omega t), \end{cases} \quad (8)$$

а рельеф дна определяется формулой:

$$z_b(x,y) = -h_0 \left(1 - \frac{r(x,y)^2}{a^2} \right), \quad r^2(x,y) = (x - L/2)^2 + (y - L/2)^2.$$

Здесь $L_x = L_y = L = 4\text{м}$, $a = 1\text{м}$, $h_0 = 0.1\text{м}$, $\eta = 0.5\text{м}$, $\omega = \sqrt{2gh_0}/a$, c^{-1} .

Для этого случая движущаяся береговая линия представляет собой эллипс. Свободная поверхность жидкости совершает периодическое движение и остается плоской во времени. Чтобы представить себе этот случай, нужно рассмотреть сосуд в форме параболоида, внутри которого вращается жидкость. В расчетах использовалась сетка 500×500 . По времени вычислительный процесс длился три периода колебаний жидкости в параболоиде. Шаг по времени определялся из условия устойчивости разностных схем (7).

На рис. 2, а, представлены графики изменения глубины потока с течением времени в точке ($x = 1,004\text{м}$; $y = 1,004\text{м}$). Из рисунка видно, что в рассматриваемом промежутке времени волна троекратно набегаёт на указанное положение на поверхности параболоида. Открытые значки на рис. 2, а, соответствуют результатам численных расчетов, а линия — аналитическому решению [16]. На рис. 2, б, в расчетные моменты времени изображен модуль разности численного и аналитического решений для глубины течения в этой же

точке. Видно, что наибольшее расхождение для рассматриваемого положения имеет место в случае начала и конца прохождения волны. Последнее может быть связано с тем, что в аналитическом решении границы волны являются непрерывными функциями от времени и координат, а в численных расчетах положение границы меняется дискретно с шагом, равным размеру ячейки. Тем не менее, когда жидкость накрывает рассматриваемую точку, погрешность расчета глубины уменьшается в 4 раза.

3. Параллельная реализация. Построенная выше полунейвная разностная схема на разнесенных разностных сетках имеет большой потенциал для создания эффективной параллельной программы, поскольку на каждом шаге по времени вычисления глубины или компонент скорости по формулам (4)–(6) могут производиться одновременно и независимо в каждом вычислительном узле сетки.

Остановимся подробнее на программной реализации этой разностной схемы. На рис. 3 приведена блок-схема программы, выполняющей вычисления по описанным выше разностным схемам. Основной цикл по времени включает блоки: а) расчета потоков G_e, G_w, G_n, G_s через грани конечных объемов и вычисление глубины h по (4) и уровня жидкости $\xi = h + z_b$; б) расчета потоков для уравнения (5) и вычисление компоненты скорости u_e^{k+1} по (5); в) расчета потоков для уравнения (6) и вычисление компоненты скорости v_n^{k+1} по (6); г) обновление полей на предыдущем шаге по времени.

В каждом из блоков а)–г) выполняется один или два вложенных цикла по сеточной области, в которых по явным формулам производятся вычисления. Например, основные блоки программы — блок а), в котором производится расчет глубины потока и уровня жидкости, и блок г), в котором выполняется обновление полей для следующего шага по времени, можно представить в следующем виде:

```
while (time<TimeT) { // основной цикл по времени
...
for (int i=1; i<Nx; i++) {
  for (int j=1; j<Ny; j++) {
// расчет потоков для уравнения (4)
    Ge = MUSCL(h[i-1][j], h[i][j], h[i+1][j], h[i+2][j], u[i+1][j]);
    Gw = MUSCL(h[i-2][j], h[i-1][j], h[i][j], h[i+1][j], u[i][j]);
    Gn = MUSCL(h[i][j-1], h[i][j], h[i][j+1], h[i][j+2], v[i][j+1]);
    Gs = MUSCL(h[i][j-2], h[i][j-1], h[i][j], h[i][j+1], v[i][j]);
// расчет глубины потока и уровня жидкости по явным формулам
    hnext[i][j] = h[i][j] - dt/dx*(Ge-Gw) - dt/dy*(Gn-Gs);
    dzetanext[i][j] = hnext[i][j]+zb[i][j];
  } // end for
// end for
...
// обмен значениями
for (int i=0; i<Nx+1; i++)
  for (int j=0; j<Ny+1; j++) {
    h[i][j] = hnext[i][j];
    u[i][j] = unext[i][j];
    v[i][j] = vnext[i][j];
  } // end for
time = time + dt;
} // end while
```

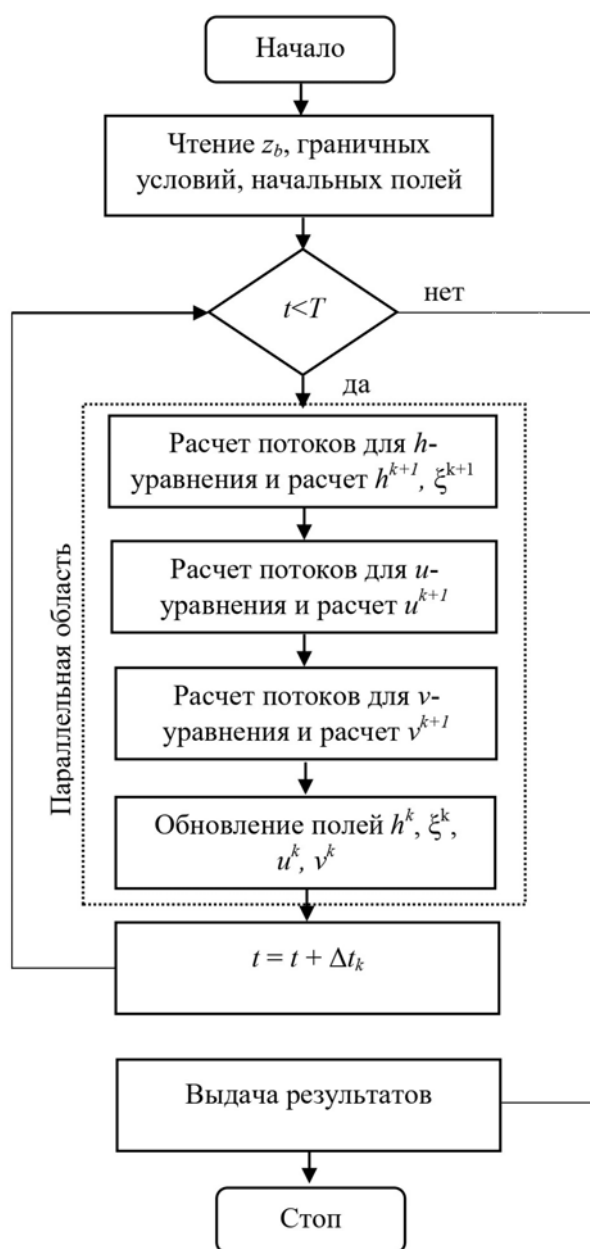


Рис. 3. Блок-схема вычислительного процесса с указанием области параллельных вычислений

Из этого фрагмента программы видно, что программа состоит из последовательности блоков двойных вложенных циклов, в которых активно производится использование и обновление одних и тех же относительно небольших по величине двумерных статических массивов. Все вещественные переменные и массивы в программе были отнесены к типу `double`. Расчеты проводились на сервере с общей памятью объемом 192Гб и двумя двенадцатядерными процессорами Intel Xeon Silver 4214 2,2 ГГц и одной графической картой GPU NVIDIA Geforce RTX2080 Ti. Расчет по последовательной программе для рассматриваемых условий проведения вычислений на сервере составил 547,7 с (сетка 500×500) и 4606,4 с (сетка 1000×1000).

Построение параллельных программ было выполнено с помощью мало затратных в параллельном программировании технологий OpenMP (Open MultiProcessing) и OpenACC (Open Accelerators), предназначенных, в основном, для многоядерных вычислительных систем CPU+GPU с общей памятью. На рис. 3 штриховой линией выделена часть блок-схемы, в которой производились параллельные вычисления. Внутри этой области выполнялось распараллеливание вложенных циклов (см. приведенный фрагмент программы) средствами параллельных технологий OpenMP или OpenACC, т.е. выполнение итераций вложенных циклов равномерно распределялось между активными нитями/процессорами ядер центрального или графического процессоров. Ниже представлен фрагмент OpenMP-программы, в которую перед каждым вложенным циклом добавлялись директивы `#pragma omp parallel for`:

```

while (time<TimeT) { // основной цикл по времени
...
#pragma omp parallel for collapse(2) private(Ge, Gw, Gn, Gs)
// распределение итераций между ядрами CPU
for (int i=1; i<Nx; i++) {
    for (int j=1; j<Ny; j++) {
        Ge = ...; // расчет потоков для уравнения (4)
        ...
        hnext[i][j] = h[i][j] - dt/dx*(Ge-Gw) - dt/dy*(Gn-Gs);
        // расчет глубины потока
        dzetanext[i][j] = hnext[i][j]+zb[i][j];
    } // end for
} // end for
...
} //end while

```

При компиляции всех программ использовался компилятор Portland Group 21.9 pgcc, который позволяет создавать параллельные программы с использованием технологий OpenMP и OpenACC для многоядерных систем с центральными и графическими процессорами.

На рис. 4 представлены графики ускорения и эффективности параллельной программы, созданной с помощью OpenMP технологии для расчетов на сетках 500×500 и 1000×1000 узлов. Ускорение S_p рассчитывалось как отношение измеренного времени счета при запуске параллельной программы на одной нити/ядре ($p = 1$) к измеренному времени счета параллельной программы на p нитях/ядрах. Рассматривались значения $p = 1, 3, 6, 12, 24$. Эффективность параллельной программы оценивалась как отношение ускорения к числу используемых нитей/ядер.

Из рис. 4 видно, что использование технологии OpenMP позволяет более чем в 15 раз ускорить вычислительный процесс. Причем программа хорошо масштабируется: при кратном увеличении вычислительной работы (\sim в 8 раз) ускорение и эффективность программы практически не меняется. Заметим, однако, что при увеличении количества используемых ядер центральных процессоров темп увеличения ускорения и значения эффективности параллельной программы снижаются. Последнее связано с увеличением нагрузки на оперативную память, точнее, с уменьшением пропускной способности памяти в расчете на одно активное ядро при использовании всех доступных физических ядер центрального процессора. Скорость выполнения арифметических операций в программе зависит

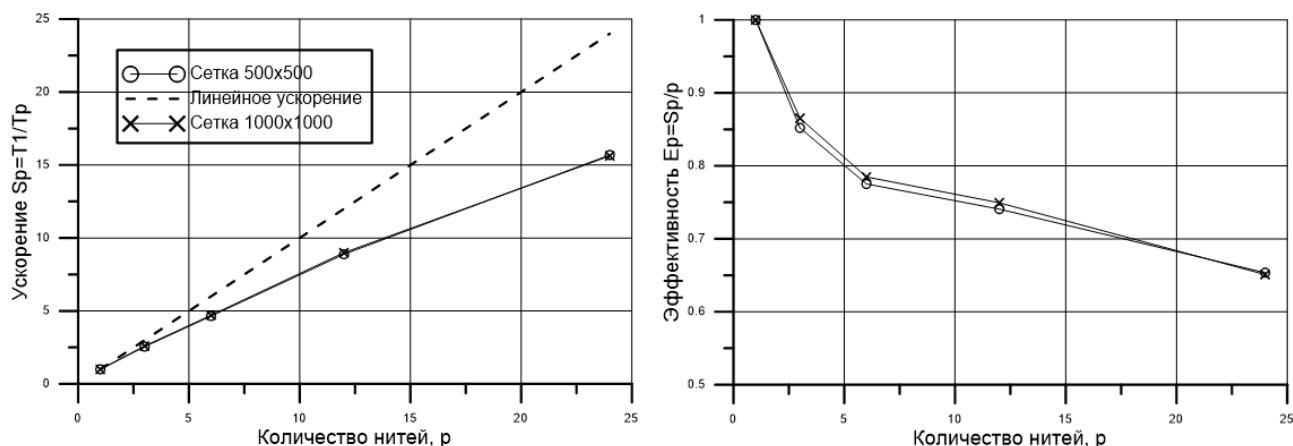


Рис. 4. Ускорение и эффективность параллельной OpenMP-программы. Штриховая линия — график зависимости идеального ускорения $S_p = p$

не только от фактической скорости процессора и количества его ядер, но и от времени выполнения операций с памятью.

Ниже представлен фрагмент OpenACC-программы, в которую перед каждым вложенным циклом добавлялись директивы `#pragma acc kernels`. Кроме того, перед началом вычислений было выполнено копирование/создание всех используемых в расчетах на графическом процессоре массивов.

```
// копирование данных из ОЗУ компьютера
// в память графической карты и обратно
#pragma acc data copy(hnext, unext, vnext, dzetanext, h, u, v)
#pragma acc data copyin(zb)
#pragma acc data create(u, v, hcх, hcу, qх, qу, qcх, qcу)
while (time < TimeT) { // основной цикл по времени
...
#pragma acc kernels // распределение итераций цикла между ядрами GPU
for (int i=1; i<Nx; i++) {
  for (int j=1; j<Ny; j++) {
    Ge = ...; // расчет потоков для уравнения (4)
    ...
    hnext[i][j] = h[i][j] - dt/dx*(Ge-Gw) - dt/dy*(Gn-Gs);
    dzetanext[i][j] = hnext[i][j] + zb[i][j];
  } // end for
} // end for
...
} // end while
```

Использование технологии OpenACC для программирования на графической карте на рассматриваемой вычислительной системе и при тех же условиях проведения расчетов дает ускорение более чем в 25 (21,3 с против 547,7 с) на сетке 500×500 и в 31 раз (145,4 с против 4606,4 с) на сетке 1000×1000 .

Заключение. Для численного моделирования нестационарных изотермических турбулентных течений в речных потоках сформулирована математическая модель, опирающаяся на приближение мелкой воды для уравнений Рейнольдса для несжимаемой

жидкости, эффективный численный метод, обеспечивающий в рамках использования метода конечного объема, структурированных разнесенных сеток и полунявных разностных схем выполнения на разностном уровне законов сохранения массы и импульса. Вычислительная реализация предложенной модели и метода была протестирована на известном аналитическом решении и распараллелена с помощью технологий OpenMP и OpenACC на гибридной многоядерной системе с общей памятью. Расчеты показали, что использование технологии OpenMP для двух двенадцатиядерных центральных процессоров позволяет более чем в 15 раз ускорить вычислительный процесс. Использование технологии OpenACC при расчетах на этой же многоядерной системе и графическом процессоре NVIDIA GeForce RTX2080 Ti дает ускорение более чем в 25 раз.

Список литературы

1. Saint-Venant A. J. B. Theorie du Mouvement non permanent des Eaux // Institut de France, Acad. des Sci. de Paris. 1871. 73 (3) 147; 73 (4) 237.
2. Gusev O. I., Khakimzyanov G. S., Skiba V. S., Chubarov L. B. Shallow water modeling of wave-structure interaction over irregular bottom // Ocean Engineering. 2023. V. 267. Art. 113284.
3. Brand S. Parallel algorithm for numerical solution of the shallow water equation // Proceedings of the Czech-Japanese Seminar in Applied Mathematics. 2006. Czech Technical University in Prague, September 14–17, 2006. P. 25–36.
4. Габдулина М. И. Параллельный алгоритм численного решения двухслойной модели мелкой воды в двумерном случае // Суперкомпьютерные дни в России 2016 // Russian Supercomputing Days 2016 // RussianSCDays.org. С. 460–467.
5. Чаплыгин А. В., Гусев А. В. Гибридная модель мелкой воды с использованием технологий MPI-OpenMP // Проблемы информатики, 2021. № 1. С. 65–82.
6. Juliati S., Gunawan P. H. OpenMP architecture to simulate 2D water oscillation on paraboloid // 5th International Conference on Information and Communication Technology (ICoIC7), 2017. P. 1–5.
7. Liu Q., Qin Y., Li G. Fast Simulation of Large-Scale Floods Based on GPU Parallel Computing // Water. 2018. V. 10(5):589.
8. Zhang S., Li W., Jing Zh, Yi Y., Zhao Y. Comparison of Three Different Parallel Computation Methods for a Two-Dimensional Dam-Break Model // Mathematical Problems in Engineering. 2017. V. 2017, Article ID 1970628, 12 pages.
9. Arnoldy A., Adytia D. Performance of Staggered Grid Implementation of 2D Shallow Water Equations using CUDA Architecture // 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, P. 286–290.
10. Churuksaeva V., Starchenko A. Mathematical modeling of a river stream based on a shallow water approach // Procedia Computer Science. 2015. V. 66. P. 200–209.
11. Rastogi A. K., Rodi W. Predictions of heat and mass transfer on open channels // ASCE Journal of the Hydraulics Division. 1978. V. 104(3). P. 397–420.
12. Launder B. E., Spalding D. B. The numerical computation of turbulent flows // Computer Methods in Applied Mechanics and Engineering. 1974. V. 2(3). P. 269–289.
13. Stelling G. S., Duimijer S. P. A. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows // Int. Journal for Numerical Methods in Fluids. 2003. V. 43. P. 1329–1354.
14. van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method // Journal of Computational Physics. 1979. V. 32(1). P. 101–136.
15. Cada M., Torrilhon M. Compact third-order limiter functions for finite volume methods // Journal of Computational Physics. 2009. V. 228. P. 4118–4145.

16. Thacker W. C. Some exact solutions to the nonlinear shallow water wave equations // Journal of Fluid Mechanics. 1981. V. 107. P. 499–508.



Старченко Александр Васильевич — E-mail: starch@math.tsu.ru. Тел.: +7 (382) 252-9553. Заведующий кафедрой вычислительной математики и компьютерного моделирования Национального исследовательского Томского

государственного университета, ведущий научный сотрудник лаборатории численного моделирования и высокопроизводительных ресурсов Томского филиала Федерального исследовательского центра информационных и вычислительных технологий. Область научных интересов: численные методы, математическое мо-

делирование, высокопроизводительные вычисления.

Starchenko Alexander Vasilievich — E-mail: starch@math.tsu.ru. Tel.: +7 (382) 252-9553. Doctor of Physical and Mathematical Sciences, Professor, Head of Department of Computational Mathematics and Computer Modelling of National Research Tomsk State University, Leading Researcher at the Laboratory of Numerical Modeling and High-Performance Resources of the Tomsk Branch of the Federal Research Center for Informational and Computational Technologies. Area of scientific interests: numerical methods, mathematical modeling, high-performance computing.

Дата поступления — 31.01.2024

A METHOD FOR ORGANIZING FUNCTIONAL DIAGNOSTICS IN REGIONAL PUBLIC WARNING SYSTEMS

V. S. Nosov, D. M. Sonkin, M. A. Sonkin, Yu. A. Chursin

INCOM LLC,
634009, Tomsk, Russia

DOI: 10.24412/2073-0667-2024-1-57-73

EDN: DXHCNY

The work examines the issues of the organization of algorithmic and software for operational diagnostics of regional public warning systems (RPWS). The original algorithms for constructing a graph of telecommunication connections, algorithms for polling telecommunication nodes and terminal devices of the RPWS are presented, as well as the results of testing the software implementation of these algorithms on the example of a real system.

Key words: warning systems, functional diagnostics, graph traversal algorithms.

References

1. Sonkin M. A. [et al.] Emergency monitoring and notification systems in the regions of oil and gas field development // Reports of TUSUR. 2014. N. 4(34). P. 133–137.
2. Leonova E. M., Leonova A. N. MODERN TRENDS in the development of public notification systems // Technologies of civil security, 2021. V. 18. P. 98–103.
3. Sonkin M. A., Saurenko T. N., Anisimov V. G. [et al.]. Mathematical models for forecasting the environmental threat of man-made accidents and catastrophes as part of integrated regional security systems // Civil security technologies. FSBI Research Institute of Emergency Situations. 2019. V. 16. P. 62–67.
4. Joint Order of the Ministry of Emergency Situations of Russia and the Ministry of Finance of Russia dated 07/31/2020. No. 578/365 “On approval of the Regulations on Public Notification Systems” (registered with the Ministry of Justice of Russia on 10/26/2020 registration No. 60567) Regulatory legal act of the Ministry or Department, dated 07/31/2020 No. 578/365. [Electron. Res.]: <https://mchs.gov.ru/dokumenty/5175>.
5. Decree of the Government of the Russian Federation dated 05/17/2023 No. 769 “On the procedure for creating, reconstructing and maintaining in a state of constant readiness for the use of public notification systems”. [Electron. Res.]: <http://government.ru/docs/all/147661/>.
6. Methodological recommendations for the creation and reconstruction of public notification systems from 02/19/2021. [Electron. Res.]: <https://mchs.gov.ru/dokumenty/5174>.
7. Methodological recommendations for maintaining constant readiness for the use of public notification systems from 12/29/2023. [Electron. Res.]: <https://mchs.gov.ru/dokumenty/7146>.
8. Prokopenko N. Yu. Discrete mathematics: NNGASU N. Novgorod, 2016.

СПОСОБ ОРГАНИЗАЦИИ ФУНКЦИОНАЛЬНОЙ ДИАГНОСТИКИ В РЕГИОНАЛЬНЫХ СИСТЕМАХ ОПОВЕЩЕНИЯ НАСЕЛЕНИЯ

В. С. Носов, Д. М. Сонькин, М. А. Сонькин, Ю. А. Чурсин

ООО «ИНКОМ»
634009, Томск, Россия

УДК 519.165

DOI: 10.24412/2073-0667-2024-1-57-73

EDN: DXHCNY

В работе исследуются вопросы организации алгоритмического и программного обеспечения оперативной диагностики региональных систем оповещения (РСО) населения. Представлены оригинальные алгоритмы построения графа телекоммуникационных связей, алгоритмы опроса телекоммуникационных узлов и оконечных устройств РСО, а также — результаты апробации работы программной реализации этих алгоритмов на примере реальной системы.

Ключевые слова: системы оповещения, функциональная диагностика, алгоритмы обхода графов.

Введение. В субъектах Российской Федерации развернуты и функционируют региональные (включая муниципальные и локальные) системы оповещения населения о чрезвычайных ситуациях природного и техногенного характера. В ходе эксплуатации этих систем чрезвычайно важно обеспечить оперативную диагностику состояния программно-аппаратных комплексов и прогнозирование возможного результата оповещения населения на текущий момент времени [1–3]. При этом следует отметить, что задачи функциональной диагностики являются ключевыми для последующей оценки возможности доведения сигналов оповещения до населения.

Согласно Положению о системах оповещения населения [4], Постановлению Правительства Российской Федерации «О порядке создания, реконструкции и поддержания в состоянии постоянной готовности к использованию систем оповещения населения» [5], Методическим рекомендациям по созданию и реконструкции систем оповещения населения [6] и Методическим рекомендациям по поддержанию в постоянной готовности к использованию систем оповещения населения [7], в региональных системах оповещения время доведения сигнала до населения в автоматизированном режиме функционирования не должно превышать 5 минут, а живучесть (вероятность работоспособности одного направления оповещения) не менее 0,99.

Уровень готовности системы оповещения к выполнению регламентных функций определяет степень предполагаемого охвата населения при доведении сигналов и команд оповещения. Исходя из этого, важной задачей являются диагностика системы оповещения и оперативное прогнозирование результатов оповещения.

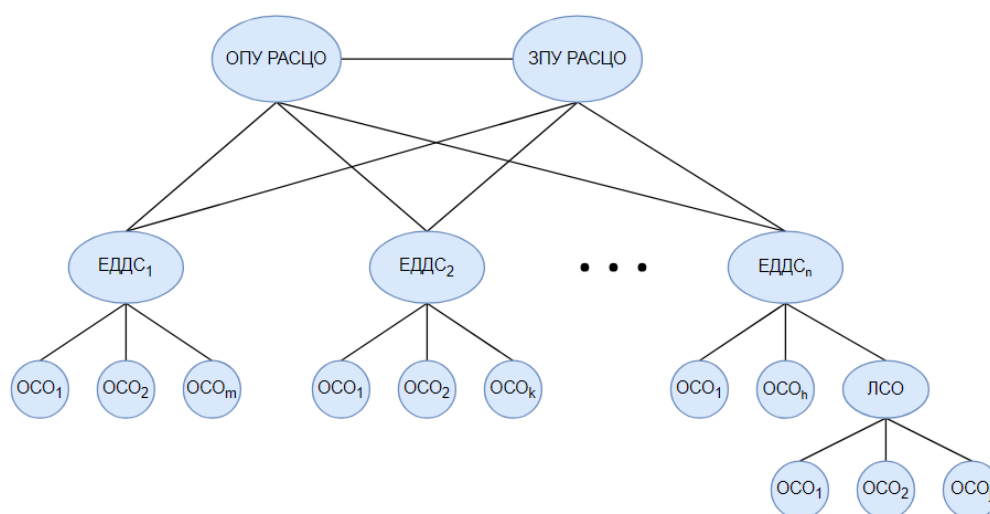


Рис. 1. Схема построения системы оповещения

Сеанс оповещения должен укладываться в определенный нормативом период времени, включающий в себя время:

- от подачи сигнала начала оповещения до фактического начала оповещения;
- сеанса оповещения;
- ожидания получения подтверждения проведения оповещения.

Так как время непосредственного оповещения изменить невозможно, то минимизация остальных периодов становится критически важной задачей.

В настоящей статье предложен подход для решения двух наиболее важных задач:

- 1) Диагностика системы оповещения и оперативное прогнозирование результатов оповещения.
- 2) Минимизация общего времени прохождения сценария оповещения.

1. Схема построения системы оповещения. Рассмотрим типовую схему построения системы оповещения, представленную на рис. 1. Логически она строится по «жесткому» каскадному принципу. Основными элементами системы оповещения являются:

- 1) Основной и запасной пункты управления РСО (ОПУ и ЗПУ).
- 2) Единые дежурно-диспетчерские службы муниципальных образований (ЕДДС).
- 3) Локальные системы оповещения (ЛСО) и оконечные средства оповещения (ОСО).

Так как все объекты систем оповещения должны иметь два и более каналов связи [5], то реальная схема построения системы оповещения имеет вид, представленный на рис. 2.

Сетевая организация РСО является сложной структурой, содержащей различные маршруты соединений между объектами системы оповещения, которые не всегда находятся в одной сети. Поэтому перед сбором информации целесообразно определить кратчайшие маршруты для ее передачи в центр диагностики, дающие минимальную нагрузку на сеть.

Для этого система:

- 1) Сканирует заранее сконфигурированную базу возможных сетевых соединений.
- 2) Получает информацию обо всех доступных узлах и каналах связи между ними.

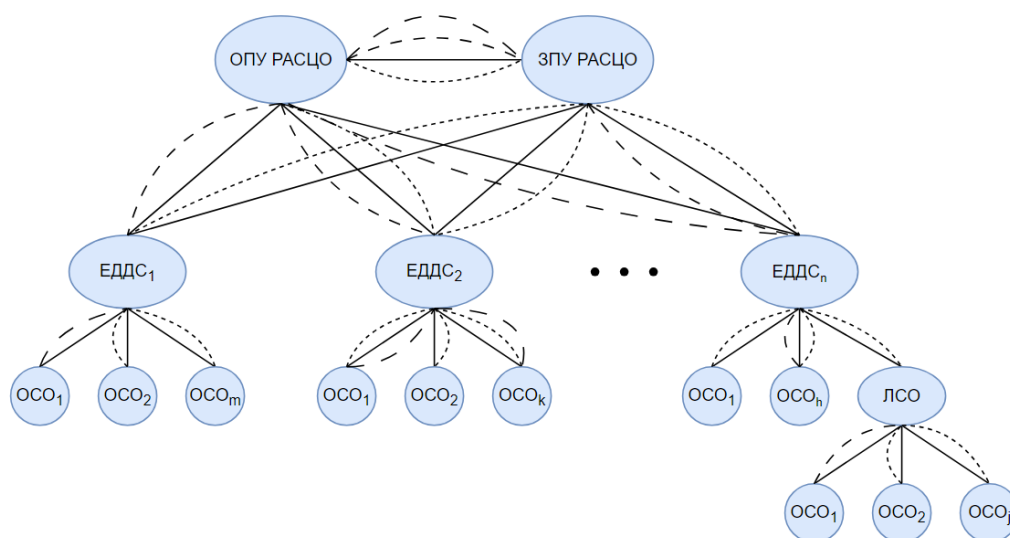


Рис. 2. Реальная схема построения системы оповещения

3) Определяет оптимальные маршруты для получения диагностической информации от каждого из узлов.

4) Динамически перестраивает карту маршрутов при обнаружении новых и потере использующихся каналов связи.

Существует ряд критериев выбора канала связи, ниже представлены некоторые из них:

- стоимость и скорость передачи информации;
- зона покрытия связи;
- надежность канала связи;
- стоимость обслуживания.

Как правило, оповещение ведется только по одному основному каналу связи. Если он недоступен, то используется следующий по приоритету резервный канал.

2. Варианты работы РСО. Возможны следующие варианты работы РСО (см. рис. 3).

Вариант 1. Данный вариант работы РСО соответствует традиционному, наиболее часто используемому способу применения РСО. Согласно нему, запуск сценариев оповещения производится по заранее заданной конфигурации. Диагностика РСО при данном варианте не производится, следовательно, изменения алгоритма оповещения происходят только в случае отказа какого-либо канала оповещения.

Вариант 2. Этот вариант работы РСО предполагает запуск диагностики РСО сразу после получения команды об оповещении. На основании результатов диагностики РСО определяется оптимальная конфигурация оповещения и запускается сценарий оповещения.

Вариант 3. Учитывая, что большую часть времени система оповещения находится в дежурном режиме, представляется логичным проведение диагностики РСО в фоновом режиме для поддержания актуальной информации о состоянии системы. Таким образом, в случае оповещения запуск сценариев оповещения сразу будет производиться по оптимальной конфигурации оповещения.

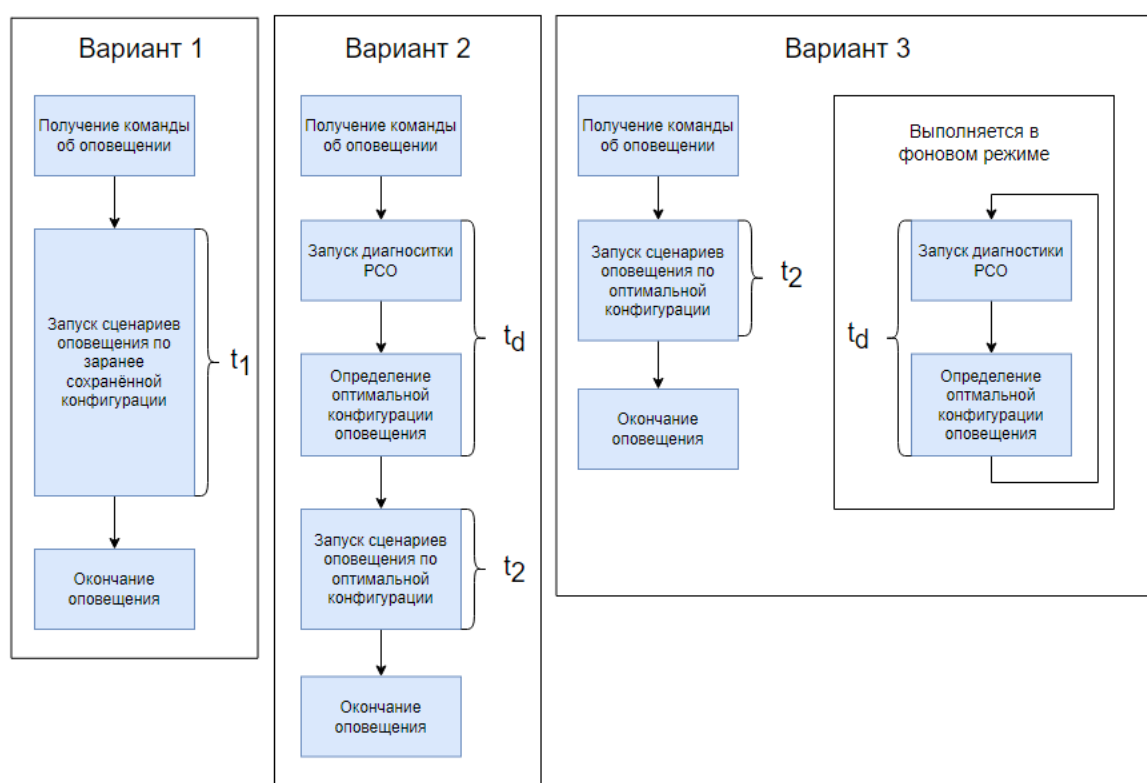


Рис. 3. Варианты работы PCO

Оценим время оповещения для каждого из вариантов работы PCO, учитывая их особенности. Для этого за основу возьмем время t_0 , описывающее время прохождения оповещения в идеальных условиях, то есть при отсутствии внесения изменений в конфигурацию и алгоритм оповещения. Тогда несложно определить время оповещения для каждого из вариантов работы PCO (см. табл. 1).

В рамках 3-го варианта работы PCO предлагается новый подход к функциональной диагностике PCO, обеспечивающий передачу диагностической информации между узлами PCO в фоновом режиме и позволяющий удаленно и централизованно осуществлять сбор и анализ всей диагностической информации от узлов PCO.

Рассмотрим одни из наиболее сложных и важных задач в ходе реализации подхода:

- Определение конфигурации сети. Поддержание ее в актуальном состоянии в фоновом режиме.

- Определение оптимальных маршрутов для получения диагностической информации. Изменение маршрутов при изменении конфигурации сети.

3. Алгоритм определения конфигурации сети. Сеть, связывающую узлы PCO, удобно интерпретировать в виде взвешенного неориентированного графа G , который состоит из множества вершин $V = \{V_i\}$, представляющих узлы сети, и множества ребер $E = \{e_j\}$, представляющих каналы связи между ними. Граф является неориентированным, так как информационный обмен ведется в обоих направлениях. Вес ребер определяется временем прохождения информации от одного узла сети к другому. Пример такого графа представлен на рис. 4.

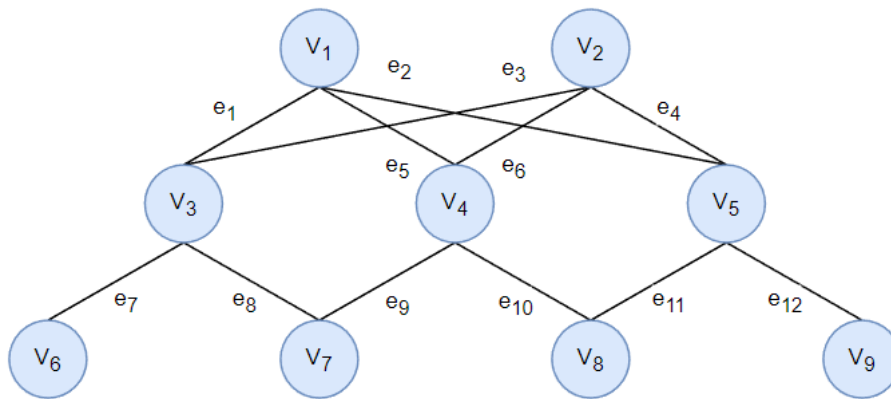


Рис. 4. Граф коммуникационных связей

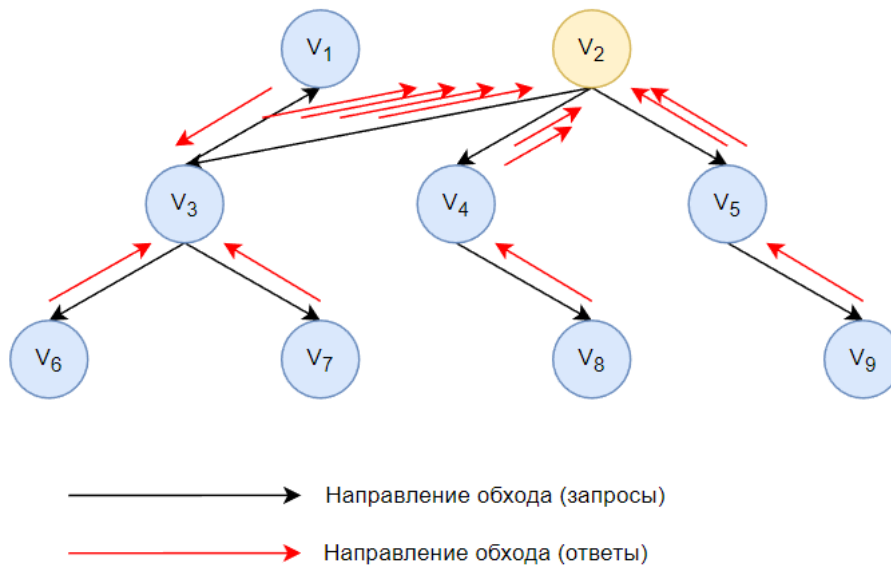


Рис. 5. Иллюстрация запросов и ответов при построении графа на основе алгоритма обхода графа в ширину

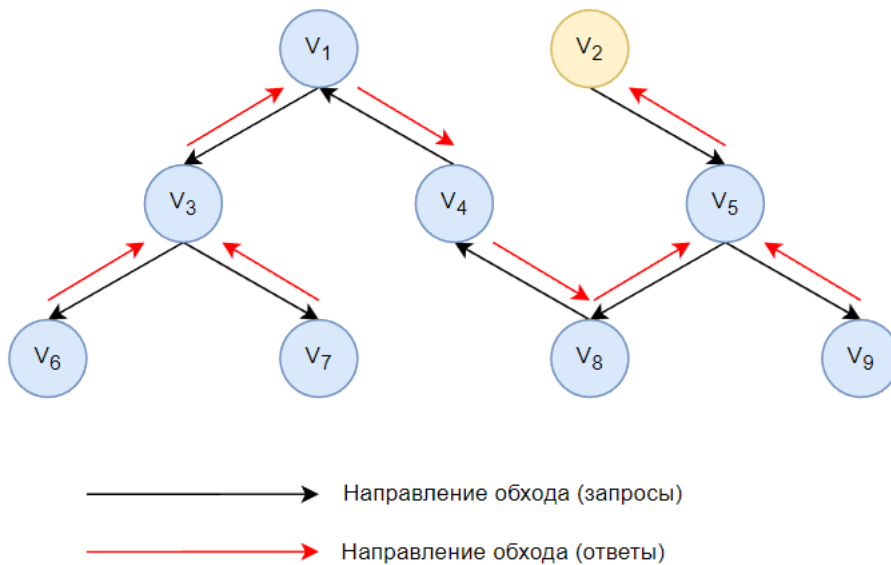


Рис. 6. Иллюстрация запросов и ответов при построении графа на основе алгоритма обхода графа в глубину

Таблица 1

Варианты работы РСО

Вариант работы РСО	Время оповещения	Примечание
Вариант 1	$t_1 \geq t_0$	t_1 — время оповещения с учетом возможных изменений конфигурации и алгоритма оповещения.
Вариант 2	$t_2 + t_d > t_0$	t_2 — время оповещения по оптимальной конфигурации. $t_2 = t_0 \Rightarrow t_2 \leq t_1$. Вариант 2 оказывается эффективнее Варианта 1 при условии, что $t_d < t_1 - t_2$, где t_d — время диагностики.
Вариант 3	$t_2 = t_0$ т.к. $t_d = 0$	Диагностика РСО проводится в фоновом режиме, что позволяет сразу запускать сценарии оповещения по оптимальной конфигурации.

Каждый узел в РСО, как правило, содержит информацию только о своих непосредственных соседях в пределах одного сегмента сети, что накладывает ограничения на скорость построения графа коммуникационных связей при различной организации опроса узлов.

Для определения оптимальных маршрутов получения диагностической информации от каждого из узлов требуется получить подграф $G' = (V', E')$ графа G , отражающий фактически присутствующие в сети объекты, и определить веса ребер, связывающие их. Для решения этой задачи проанализируем применимость двух наиболее распространенных алгоритмов обхода графа: в ширину и в глубину.

4. Модернизированный алгоритм обхода графа в ширину для сети РСО.

Основным отличием данного алгоритма от стандартного являются дополнительные определение и учет времени передачи данных по сети от каждого определенного узла в начальный. В качестве примера возьмем граф, представленный на рис. 4, где $V = \{V_i\}$ при $i = 1, \dots, 9$ и $E = \{e_j\}$ при $j = 1, \dots, 12$. В рассматриваемом случае вес каждого ребра, принадлежащего множеству E , заранее не известен и определяется временем опроса между узлами. Изначально все узлы помечаются как «не обработаны». Начальный узел, с которого выполняется обход графа, выбирается системой автоматически, либо задается вручную. Далее выполняется следующий алгоритм:

- 1) Добавляем начальный узел (V_2) в очередь запросов и подграф. Отмечаем его как «обработан».
- 2) Проверяем наличие элементов в очереди запросов, если очередь не пуста, переходим к пункту 3, иначе — к пункту 10.
- 3) Получаем список связей из текущего узла.
- 4) Проверяем наличие элементов в списке связей, если список не пуст, переходим к пункту 5, иначе удаляем текущий узел из очереди запросов.
- 5) Отправляем запрос первому элементу из списка связей, если опрашиваемый узел ответил, то переходим к пункту 6, иначе — к пункту 9.
- 6) Фиксируем время отклика (вес e_i ребра). Добавляем ребро в подграф.
- 7) Если опрашиваемый узел «не обработан», то переходим к пункту 8, иначе — к пункту 9.
- 8) Добавляем узел в очередь запросов и подграф. Отмечаем его как «обработан».

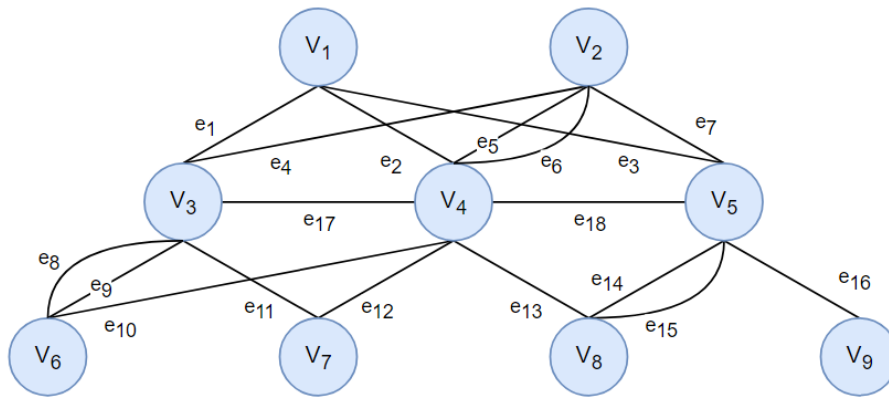


Рис. 7. Граф коммуникационных связей

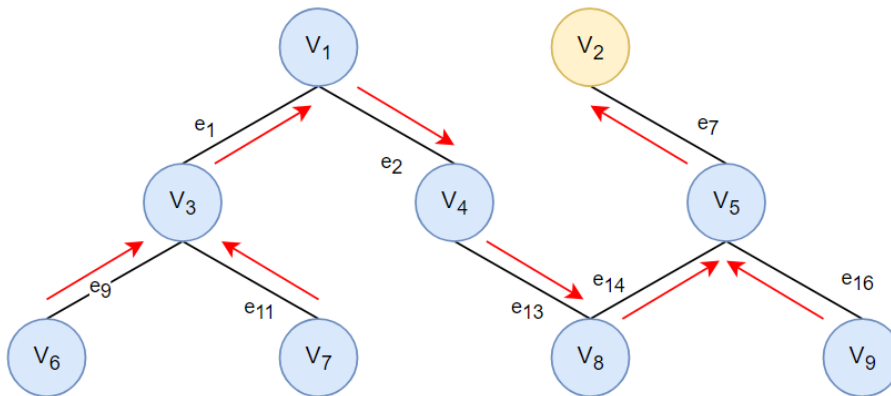


Рис. 8. Минимальное остовное дерево графа коммуникационных связей

9) Удаляем элемент из списка связей. Переходим к пункту 4.

10) Фиксируем общее время работы алгоритма. Завершаем работу.

Иллюстрация запросов и ответов при построении графа из начального узла (V_2) на основе алгоритма обхода графа в ширину представлена на рис. 5. Блок-схема алгоритма приведена в Приложении А.

5. Модернизированный алгоритм обхода графа в глубину для сети PCO.

Рассмотрим второй алгоритм построения графа коммуникационных связей на основе алгоритма обхода графа в глубину. Так как отличительной чертой данного алгоритма от предыдущего является использование рекурсивной функции $DFS()$, входными данными для которой является ссылка на опрашиваемый узел, то в этом случае отличиям алгоритма от стандартного будут являться дополнительные определение и учет времени передачи данных по сети от группы определенных функцией $DFS()$ узлов до опрашиваемого. Начальный узел, с которого выполняется обход графа, выбирается системой автоматически, либо задается вручную. Далее выполняется следующий алгоритм:

- 1) Запускаем рекурсивную функцию $DFS(m)$, где m — начальный узел.
- 2) Добавляем узел в подграф. Отмечаем его как «обработан».
- 3) Получаем список связей из текущего узла.

4) Проверяем наличие элементов в списке связей, если список не пуст, переходим к пункту 5, иначе выходим из функции (переходим на уровень выше). Если функция имеет нулевой уровень вложенности, то переходим к пункту 11.

5) Отправляем запрос первому элементу из списка связей, если опрашиваемый узел ответил, то переходим к пункту 6, иначе — к пункту 10.

6) Фиксируем время отклика (вес e_i ребра). Добавляем ребро в подграф.

7) Если опрашиваемый узел «не обработан», то переходим к пункту 8, иначе к пункту 10.

8) Фиксируем общее время доступа к опрашиваемому узлу из начального.

9) Запускаем рекурсивную функцию $DFS(n)$.

10) Удаляем элемент списка связей. Переходим к пункту 4.

11) Фиксируем общее время работы алгоритма. Завершаем работу.

Иллюстрация запросов и ответов при построении графа из узла начального узла (V_2) на основе алгоритма обхода графа в глубину представлена на рис. 6. Блок-схема алгоритма приведена в Приложении Б.

Так как в сети нет изолированных сегментов, то в результате работы алгоритмов автоматически составляется граф всей сети устройств РСО.

Оба алгоритма были протестированы на специально разработанной компьютерной модели, которая отражает РСО с узлами и связями как на рис. 4. Задержка передачи информации между узлами задавалась определенной величиной и не менялась в ходе тестирования алгоритма.

Эксперимент показал, что алгоритм, основанный на поиске в глубину, показывает меньшее время построения конфигурации сети. Это связано с тем, что при рекурсивном переборе узлов информация о соседних узлах накапливается в «опрашивающем» узле и после опроса всех узлов передается на уровень выше за одну транзакцию, а при использовании очереди тратится дополнительное время для передачи полученной информации о каждом новом узле до начального узла. Для периодической диагностики предлагается использовать алгоритм, основанный на обходе графа в глубину.

6. Алгоритм определения маршрутов для получения диагностической информации. Точкой входа или начальным узлом для выполнения алгоритма определения маршрутов для получения диагностической информации является узел, на котором установлен центр диагностики РСО. Алгоритм состоит из 4-х этапов:

1) Использование алгоритма определения конфигурации сети, разобранный в разделе выше. Граф коммуникационных связей представлен на рис. 7.

2) Построение минимального остовного дерева путем применения алгоритма Прима [8] (см. рис. 8).

3) Определение оптимальных маршрутов от каждого узла к начальному с помощью алгоритма Дейкстры [8]. Маршруты показаны красными стрелками на рис. 8.

4) Назначение «ведущего» узла для каждого из диагностируемых, то есть узла, которому будет передаваться диагностическая информация от него (см. табл. 2).

7. Порядок функционирования центра диагностики. Для поддержания информации о доступности элементов РСО и каналах связи между ними в актуальном состоянии предлагается запускать алгоритм определения конфигурации сети при загрузке системы, а после с периодичностью t_1 .

Для поддержания информации о состоянии доступных элементов РСО в актуальном состоянии предлагается запускать запрос диагностической информации после выполнения

Таблица 2

Таблица «ведущих» узлов

Основной	V_2	V_5	V_9	V_8	V_4	V_1	V_3	V_6	V_7
«Ведущий»	–	V_2	V_5	V_5	V_8	V_4	V_1	V_3	V_3

алгоритма определения конфигурации сети, а после с периодичностью t_b . Информация от каждого узла отправляется асинхронно.

Если при запросе диагностической информации от какого-то устройства информация не приходит, либо значительно изменяется время отклика, то запускается алгоритм определения конфигурации сети.

Время выполнения алгоритма определения оптимальных маршрутов для получения диагностической информации в случае неработоспособности каналов связи определяется таймаутами ожидания ответов от опрашиваемых узлов.

По получаемой диагностической информации автоматически определяется готовность РСО к выполнению своих функций в соответствии с регламентом.

8. Контрольный пример. В данном примере центр диагностики РСО установлен на ЕДДС₁, поэтому алгоритм определения оптимальных маршрутов передачи диагностической информации будет запускаться относительно него. Граф телекоммуникационных связей после выполнения алгоритма определения конфигурации сети показан на рис. 9.

Минимальное остовное дерево коммуникационных связей, полученное путем применения алгоритма Прима, имеет вид, показанный на рис. 10. Матрица весов для данного графа представлена в табл. 3.

Далее смоделируем отказ работы основных каналов связи между ПУ и всеми ЕДДС. Красным на рис. 9 указаны каналы связи, которые в настоящий момент недоступны.

После выполнения алгоритма определения конфигурации сети минимальное остовное дерево имеет вид, показанный на рис. 11. Матрица весов, соответствующая новому графу, представлена в табл. 4.

9. Оценка эффективности предлагаемого подхода. При оповещении сигналы из узла верхнего уровня посылаются в узлы нижнего уровня одновременно. Общее время задержки будет зависеть не столько от общего количества узлов, сколько от количества уровней в системе оповещения (см. рис. 1). При этом общее время задержки при оповещении будет определяться временем задержки в самой «длинной» (с минимально необходимым для связи узлов количеством ребер) цепочке системы оповещения. Увеличение количества уровней в системе позволяет представить систему более наглядно, структурировано, приближенно к реальной иерархии подчиненности объектов систем оповещения.

Проведем анализ увеличения времени оповещения в случае наличия обрывов каналов связи в референсной цепочке и покажем зависимость количества возможных задержек от количества уровней в системе оповещения. Рассмотрим пример. Для упрощения расчета оценки примем, что все узлы имеют по 2 канала связи (основной и резервный). Также зададим, что рассматриваемая самая «длинная» цепочка имеет 2 обрыва каналов связи. Учитываются только те случаи, когда узел остается доступным (нет обрыва обоих каналов между узлами). Алгоритм анализа реализован следующим образом: все состояния каналов связи кодируются двоичным числом с числом разрядов, равным количеству каналов. Нечетные разряды кодируют основной канал, а четные — резервный. Например, число

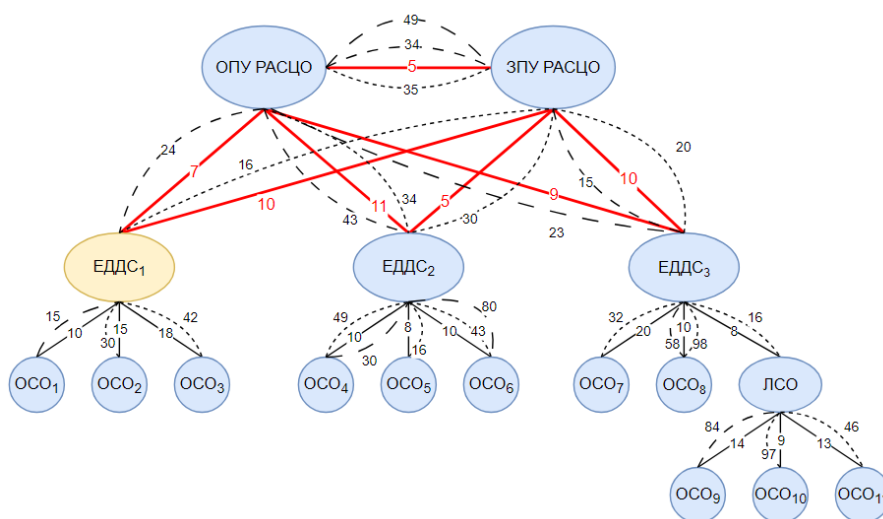


Рис. 9. Граф коммуникационных связей

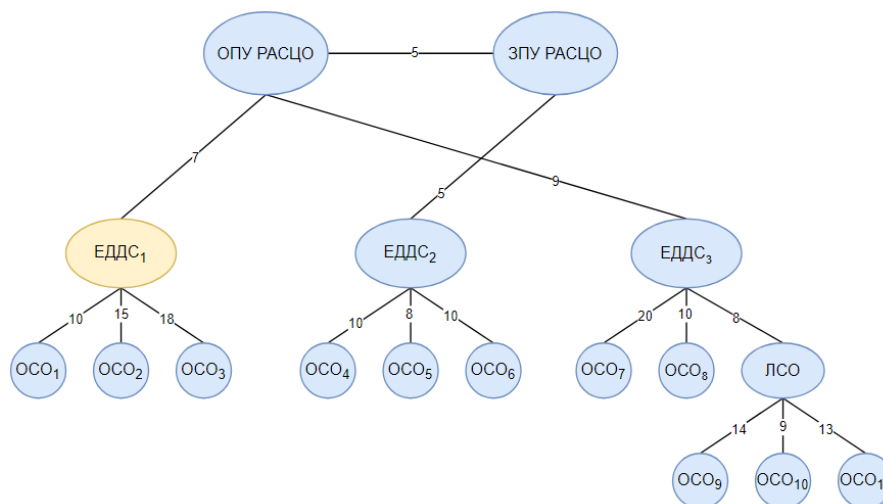


Рис. 10. Минимальное остовное дерево коммуникационных связей

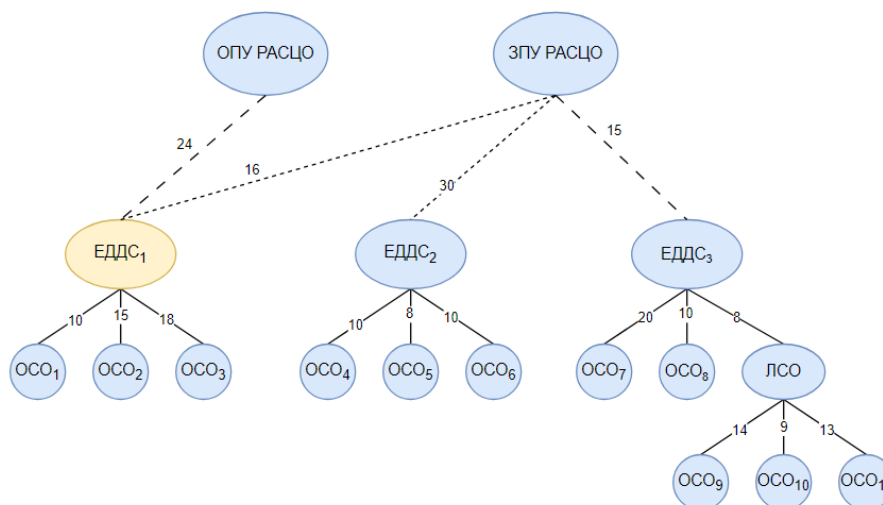


Рис. 11. Минимальное остовное дерево коммуникационных связей после определения новой конфигурации сети

Таблица 5

Зависимость вероятности (в процентах) времени прохождения сигнала оповещения от количества уровней в системе оповещения

количество таймаутов / количество уровней	2	3	4	5
0	16,6	20	7,14	1,42
1	33,3	40	21,42	5,71
2	0	20	21,42	8,57
3	0	0	7,14	5,71
4	0	0	0	1,42

100001 кодирует 6 каналов связи между 4-мя узлами, причем присутствует нарушение связи между узлами 1 и 2 по основному каналу и между узлами 2 и 3 по резервному. Далее идет перебор всех возможных комбинаций чисел, отсеиваются числа с заданным количеством единичных разрядов (обрывов связи), и уже для них рассчитывается время прохождения сигналов: 0, если основной канал работоспособен, время таймаута, если основной канал неработоспособен, а резервный работоспособен, и ∞ , если оба канала неработоспособны. Общее время прохождения сигнала по цепочке суммируется. Далее для каждого времени прохождения рассчитывается вероятность, исходя из всех возможных вариантов распределения заданного количества обрывов каналов связи. Полученные данные сведены в таблицу, показывающую распределение вероятности прохождения сигнала с определенной задержкой в процентах в зависимости от количества уровней в системе оповещения.

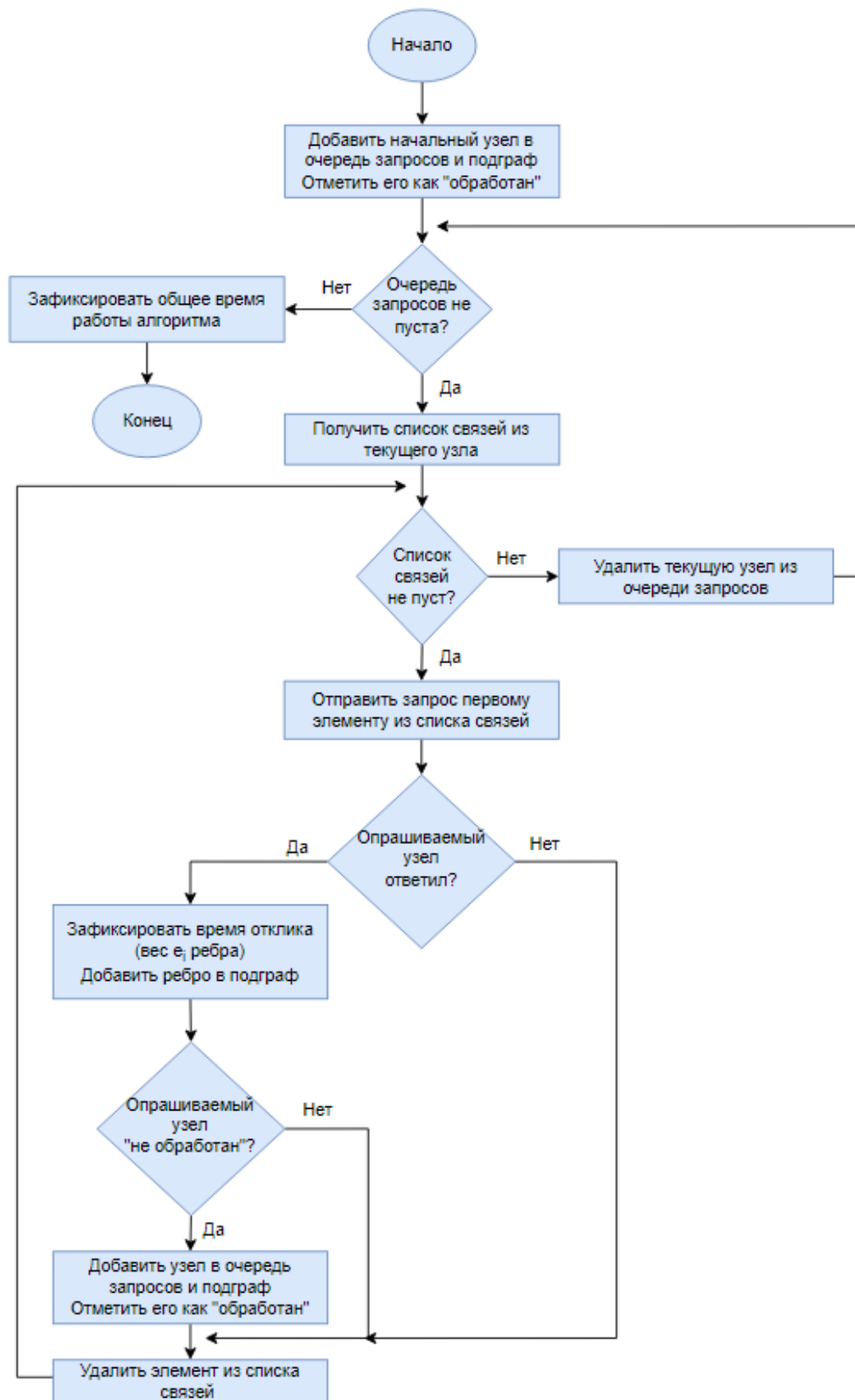
Из табл. 5 видно, что в данной конфигурации наиболее вероятное время задержки оповещения составит 2 таймаута. Как правило, время таймаута устанавливается порядка 20–30 секунд, что в конечном итоге увеличивает время прохождения оповещения до 1 минуты. Учитывая, что на все оповещение, включая подтверждение окончания, выделяется 5 минут [4], то период в 20 процентов от этого времени является существенным. Это говорит о целесообразности использования фоновой диагностики в рамках предлагаемого подхода.

Выводы. Предложенный подход к организации диагностики РСО позволяет в фоновом режиме определять маршруты для сбора диагностической информации о состоянии объектов РСО, непрерывно проводить диагностику всех узлов системы, оценивать ее работоспособность и рассчитывать предполагаемый процент доведения сигналов оповещения до населения. Применение данного подхода при разрывах каналов связи сокращает общее время прохождения сеанса оповещения до 20 %. Подход был программно реализован и апробирован при техническом обслуживании систем оповещения населения Кемеровской, Волгоградской и Калининградской областей. Все заявленные характеристики были подтверждены.

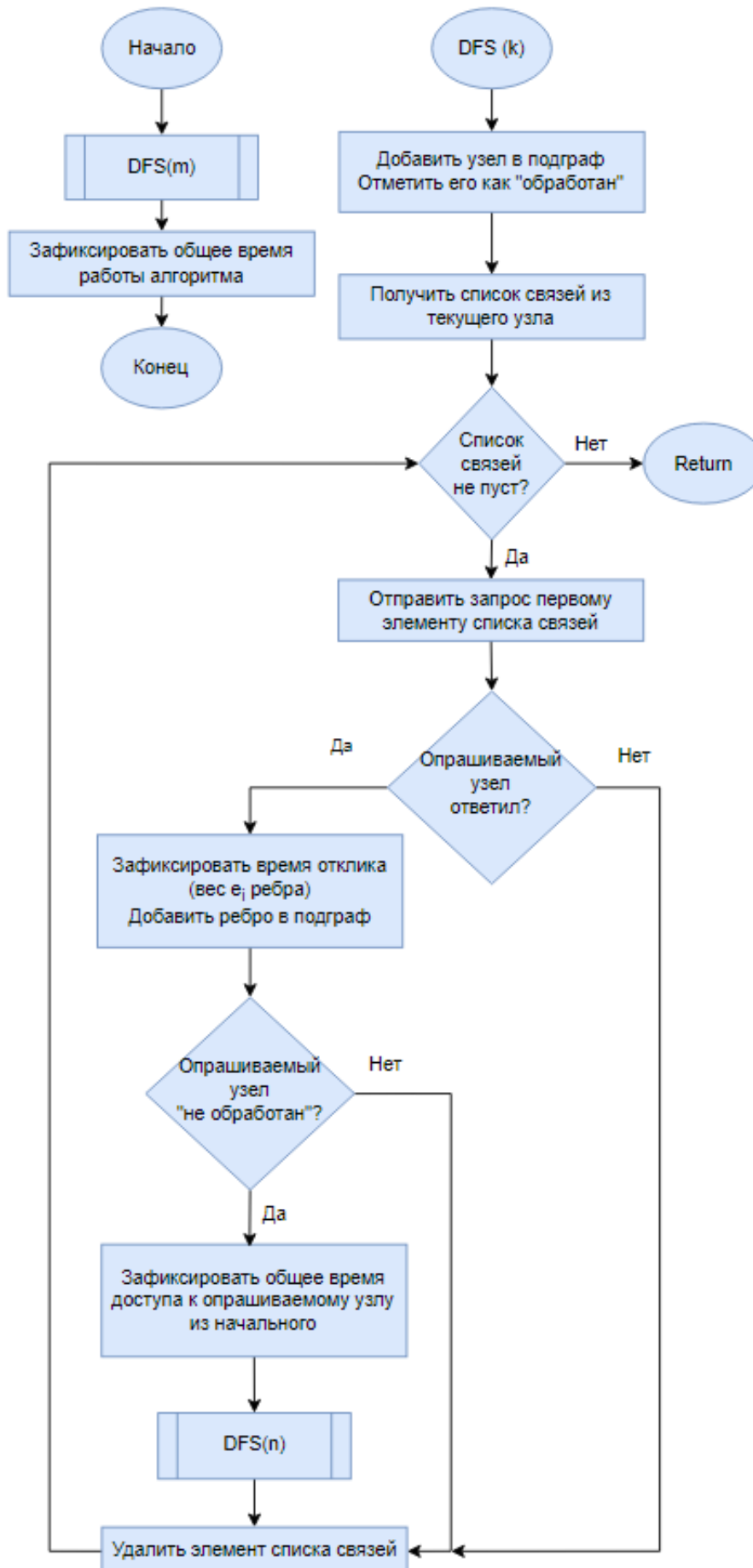
Список литературы

1. Сонькин М. А. [и др.] Системы мониторинга и оповещения о чрезвычайных ситуациях в регионах разработки нефтегазовых месторождений // Доклады ТУСУР. 2014. № 4(34). С. 133–137.

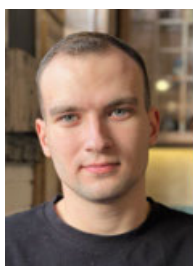
Приложение А. Блок-схема алгоритма обхода графа в ширину



Приложение Б. Блок-схема алгоритма обхода графа в глубину



2. Леонова Е. М., Леонова А. Н. Современные тенденции развития систем оповещения населения // Технологии гражданской безопасности, том. 18. 2021. С. 98–103.
3. Сонькин М. А., Сауренко Т. Н., Анисимов В. Г. [и др.]. Математические модели прогнозирования экологической угрозы техногенных аварий и катастроф в составе интегрированных систем безопасности региона // Технологии гражданской безопасности. ФГБУ ВНИИ ГО ЧС. 2019. Т. 16. С. 62–67.
4. Совместный приказ МЧС России и Минцифры России от 31.07.2020 № 578/365 «Об утверждении Положения о системах оповещения населения» (зарегистрирован в Минюсте России 26.10.2020, регистрационный № 60567) Нормативно-правовой акт Министерства или Ведомства от 31.07.2020 г. № 578/365. [Электрон. рес.]: <https://mchs.gov.ru/dokumenty/5175>.
5. Постановление Правительства Российской Федерации от 17.05.2023 г. № 769 «О порядке создания, реконструкции и поддержания в состоянии постоянной готовности к использованию систем оповещения населения». [Электрон. рес.]: <http://government.ru/docs/all/147661/>.
6. Методические рекомендации по созданию и реконструкции систем оповещения населения от 19.02.2021 г. [Электрон. рес.]: <https://mchs.gov.ru/dokumenty/5174>.
7. Методические рекомендации по поддержанию в постоянной готовности к использованию систем оповещения населения от 29.12.2023 г. [Электрон. рес.]: <https://mchs.gov.ru/dokumenty/7146>.
8. Прокопенко Н. Ю. Дискретная математика: ННГАСУ Н. Новгород, 2016. 251 с.



Носов Владимир Сергеевич окончил институт прикладной математики и компьютерных наук Томского государственного университета по направлению подготовки «Фундаментальная информатика и информационные технологии»

в 2023 г., бакалавр. Инженер ООО «ИНКОМ». Область научной специализации: телекоммуникационные системы, системы мониторинга и управления подвижными объектами.

Nosov Vladimir Sergeevich graduated from the Institute of Applied Mathematics and Computer Science of Tomsk State University in the field of Fundamental Informatics and Information Technology in 2023, Bachelor's degree. Engineer of INCOM LLC. Field of scientific specialization: telecommunication systems, monitoring and control systems for mobile objects.



Сонькин Дмитрий Михайлович в 2007 г. окончил бакалавриат по направлению «Информатика и вычислительная техника» в Томском политехническом университете. В 2009 году окончил

магистратуру Томского политехнического университета по направлению «Информатика и вычислительная техника», доктор технических наук, доцент. Научный сотрудник ООО «ИНКОМ». Область научной специализации: информационно-аналитические системы, геоинформационные системы и технологии, системы мониторинга и управления подвижными объектами, автоматизация производства промышленных и производственных предприятий. Является автором и соавтором более 80 научных работ.

Sonkin Dmitry Mikhailovich graduated from Tomsk Polytechnic University with a bachelor's degree in Computer Science and Computer Engineering in 2007. In 2009, he graduated from Tomsk Polytechnic University with a Master's degree in Computer Science and Computer Engineering, Doctor of Technical Sciences, Associate Professor. Researcher at INCOM LLC. Field of scientific specialization: information and analytical systems, geoinformation systems and technologies, monitoring and control systems for mobile objects, automation of industrial and manufacturing enterprises. He is the author and co-author of more than 80 scientific papers.

Сонькин Михаил Аркадьевич в 1976 г. окончил Омский политехнический институт по

специальности «Электронные вычислительные машины», доктор технических наук, доцент. Директор ООО «ИНКОМ». Область научной специализации: информационно-аналитические системы, геоинформационные системы и технологии, системы мониторинга и управления подвижными объектами, автоматизация производства промышленных и производственных предприятий. Является автором и соавтором более 150 научных работ.



Sonkin Mikhail Arkadievich graduated from Omsk Polytechnic Institute in 1976 with a degree in Electronic Computing Machines, Doctor of Technical Sciences, associate Professor. Director of INCOM

LLC. Field of scientific specialization: information and analytical systems, geoinformation systems and technologies, monitoring and control systems for mobile objects, automation of industrial and manufacturing enterprises. He is the author and co-author of more than 150 scientific papers.

Чурсин Юрий Александрович в 2007 г. окончил с отличием физико-технический фа-

культет Томского политехнического университета по специальности «Электроника и автоматика физических установок», кандидат технических наук, доцент. Ведущий научный сотрудник ООО «ИНКОМ». Область научной специализации: микропроцессорная техника, телекоммуникационные системы, моделирование и автоматизация технологических процессов, промышленная автоматика и робототехника. Является автором и соавтором более 100 научных работ.



Chursin Yury Alexandrovich graduated with honors from the Faculty of Physics and Technology of Tomsk Polytechnic University in 2007 with a degree in Electronics and Automation of Physical

Installations, Candidate of Technical Sciences, Associate Professor. A leading researcher at INCOM LLC. Scientific specialization: microprocessor technology, telecommunication systems, modeling and automation of technological processes, industrial automation and robotics. He is the author and co-author of more than 100 scientific papers.

Дата поступления — 15.02.2024

ARCHITECTURE OF A DISTRIBUTED COMPUTING SYSTEM BASED ON MOBILE DEVICES

I. S. Ulyanichev, D. V. Wiens

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2024-1-74-97

EDN: FWATXX

The article outlines the architecture of the mathematical and software of a distributed computing system based on many mobile devices integrated into a common network infrastructure. The distributed computing system is based on a problem-oriented task execution model (known in foreign literature as the Task-Based Execution Model). This model of distributed computing is most preferable when solving optimization and inverse problems, when the set of admissible sets of input parameters is sufficiently large. At the same time, the tasks themselves are performed independently, but require significant time expenditures to solve each specific task, comparable to the “lifetime” of a computing node. The Task-Based Execution Model approach allows you to reduce the amount of information transferred between the server and the node, which in turn improves overall performance.

The DHCA system we offer has a client-server architecture and is tailored for the Android system. The mobile client is written using the latest Android approaches to background computing, which allows for increased performance as well as improved security for end users. The system takes into account all the features of the interaction of client-server architecture for distributed systems, and also solves the problem associated with the unavailability of computing nodes by automatically redistributing tasks. Open source code increases trust in the system from end users, which can have a beneficial effect on the number of active participants in computing.

The web client is a web application written in PHP. User interaction with the web client interface is carried out using the jQuery JavaScript library, which allows you to send requests and receive data from the server asynchronously. Interaction with the server is carried out through HTTP requests to the REST API, written in PHP.

The DHCA server is a web server with PHP support, with a connected MySQL database. Thanks to the modular architecture of the API used, any relational DBMS that supports the structured query language SQL can be connected to the system.

The mobile client is an application installed on a smartphone running Android operating system version 6.0 (Marshmallow) and higher. The client is written in Kotlin using the MVVM design pattern.

The mobile client carries out all interactions with the server API using the Retrofit library, designed for processing HTTP requests. All data is returned in JSON format. Information messages are sent to the mobile client by the Firebase Cloud Messaging (FCM) library.

To derive a formula for assessing scalability for the presented architecture, we take into account the following parameters: t — time required to calculate one data block; N — number of blocks to be calculated (in the case of a ray tracing problem, number of pixels); P is the number of devices involved in the calculation, t_0 is the time required for the initial breakdown of the source data into

Работа выполнена за счет гранта Российского научного фонда № 23-11-00014

computational blocks; t_w — time of writing one initial block to the database after splitting; t_s — time required for the client to receive one block of data from the server (send); t_r — time required for the server to receive the calculated data block from the client (receive); t_v — time required to write the result of the calculated block to the database (save). We obtain the calculation time on p devices $T(P)$. Through acceleration as the ratio of $T(1)$ to $T(P)$, we obtain P devices at which maximum acceleration is achieved. To demonstrate the robustness of the developed system, a solution to three problems is presented, in which each elementary task has a different load on the processor and on the network infrastructure. Let us highlight three such tasks:

1) The problem of inverse ray tracing, which is characterized by a significant amount of transmitted data to form a scene and the received data of a full image block. At the same time, the time for ray tracing is much longer relative to the data transmission time.

2) The white dwarf collision problem, which is formulated in a one-dimensional formulation and is characterized by the transmission of only six values to describe the state of the dwarfs before the collision and one output value describing the maximum temperature during the collision. In this case, the calculation time is comparable to the data transmission time. The problem has an analytical solution that requires the resolution of nonlinear equations and is described in detail in the work.

3) The problem of nuclear combustion of carbon in white dwarfs, which is formulated in the form of changes in the concentrations of the main isotopes during nuclear combustion of carbon during the solution of the ODE system. The problem is characterized by input data in the form of two values (initial temperature and carbon density), describing the state of the degenerate gas, and a time parameter in which the evolution of isotopes occurs. The result is represented as seven return values describing the concentration of each isotope. Note that the ODE calculation time significantly exceeds the data transmission time. The formulation of the problem and the results obtained are described in detail in the work.

The architecture proposed in the article makes it possible to take into account the reconfiguration of a network of computing devices and build a guaranteed assessment of the scalability of a distributed heterogeneous computing system.

Key words: Task-Based Execution Model, mobile computing, distributed computing systems.

References

1. Armbrust M., Fox A., Griffith R., Joseph A.D, Katz R. H, Konwinski A., Lee G., Patterson D. A, Rabkin A., I. Stoica, et al. Above the clouds: A berkeley view of cloud computing // Technical Report. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley. 2009.
2. Buyya R., Yeo C.S., Venugopal S., Broberg J., Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility // Future Generation Computer Systems. 2009. 25(6). P. 599–616.
3. Foster I., Kesselman C., Tuecke S. Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of High Performance Computing Applications. 2001. 15(3). P. 200–222.
4. Foster I., Zhao Y., Raicu I., Lu S. Cloud Computing and Grid Computing 360-degree Compared // Grid Computing Environments Workshop (GCE'08). 2008. P. 1–10.
5. Peer-to-peer (P2P). [Electron. Res.]: <https://www.techtarget.com/searchnetworking/definition/peer-to-peer> (access date 26.04.2022).
6. Peer-to-peer computing. [Electron. Res.]: <https://melimde.com/koncepciya-operacionnih-sistem.html?page=47> (access date 26.04.2022).
7. Lavoie E., Hendren L. Personal Volunteer Computing // Proceedings of the 16th ACM International Conference on Computing Frontiers. 2019. P. 240–246.
8. Virtejanu I., Nitu C. Programming distributed applications for mobile platforms using MPI // U.P.B. Sci. Bull. 2013. 75(4).

9. Attia D. E., ElKorany A. M., Moussa A. S. High Performance Computing Over Parallel Mobile Systems // *International Journal of Advanced Computer Science and Applications*. 2016. 7(9). P. 99–103.
10. Büsching F., Schildt S., Wolf L. DroidCluster: Towards Smartphone Cluster Computing — The Streets are Paved with Potential Computer Clusters // *32nd International Conference on Distributed Computing Systems Workshops*. 2012. P. 114–117.
11. Prem Kumar M., Bhat R. R., Alavandar S. R., Ananthanarayana V. S. Distributed Public Computing and Storage using Mobile Devices // *2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. 2018. P. 82–87.
12. Nizamov N. R., Shahova I. S. Mechanisms for using mobile devices in distributed computing // *Russian Digital Libraries Journal*. 2019. 22(4). P. 200–213.
13. Fadlallah G., Mcheick H., Rebaine D. Layered Architectural Model for Collaborative Computing in Peripheral Autonomous Networks of Mobile Devices // *Procedia Computer Science*. 2019. 155. P. 201–209.
14. Arslan M. Y. Computing While Charging: Building a Distributed Computing Infrastructure Using Smartphones // *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. 2012. P. 193–204.
15. Pratt T. K., Seitelman L. H., Zampano R. R., Murphy C. E., Landis F. Optimization applications for aircraft engine design and manufacture // *Advances in Engineering Software*. 1993. V. 16, I. 2. P. 111–117.
16. Kabanikhin S. I., Kulikov I. M., Shishlenin M. A. An algorithm for recovering the characteristics of the initial state of supernova // *Computational Mathematics & Mathematical Physics*. 2020. V. 60, I. 6. P. 1008–1016.
17. Stone J., Tomida K., White C., Felker K. The Athena++ Adaptive Mesh Refinement Framework: Design and Magnetohydrodynamic Solvers // *The Astrophysical Journal Supplement Series*. 2020. V. 249. Article Number 4.
18. Kulikov I., Chernykh I., Tutukov A. A New Hydrodynamic Code with Explicit Vectorization Instructions Optimizations that Is Dedicated to the Numerical Simulation of Astrophysical Gas Flow. I. Numerical Method, Tests, and Model Problems // *The Astrophysical Journal Supplement Series*. 2019. V. 243. Article Number 4.
19. Kulikov I. M., Chernykh I. G., Tutukov A. V. Mathematical Modeling of a High-Speed Collision of White Dwarfs—the Explosion Mechanism of Type Ia/Iax Supernovae // *Journal of Applied and Industrial Mathematics*. 2022. V. 16. P. 80–88.
20. Kulikov I. M., Chernykh I. G., Ulyanichev I. S., Tutukov A. V. Mathematical Simulation of Nuclear Carbon Burning in White Dwarfs Using a 7-Isotope Reaction Network // *Journal of Applied and Industrial Mathematics*. 2022. V. 16. P. 440–448.
21. Kulikov I. GPUPEGAS: A New GPU-accelerated Hydrodynamic Code for Numerical Simulations of Interacting Galaxies // *The Astrophysical Journal Supplements Series*. 2014. V. 214. Article Number 12.
22. Kulikov I. M., Chernykh I. G., Snytnikov A. V., Glinskiy B. M., Tutukov A. V. AstroPhi: A code for complex simulation of dynamics of astrophysical objects using hybrid supercomputers // *Computer Physics Communications*. 2015. V. 186. P. 71–80.
23. Kulikov I., Chernykh I., Karavaev D., Sapetina A. The Energy Efficiency Research of Godunov Method on Intel Xeon Scalable Architecture // *IEEE. 2021 Ivannikov Ispras Open Conference (ISPRAS)*. 2022. Article Number 21722440.
24. Kulikov I., Chernykh I., Karavaev D., Sapetina A., Lomakin S. The Efficiency of Hydrodynamic Code on Intel Xeon Scalable Architecture // *IEEE. 2021 Ivannikov Memorial Workshop (IVMEM)*. 2022. Article Number 21704168.

АРХИТЕКТУРА РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ НА БАЗЕ МОБИЛЬНЫХ УСТРОЙСТВ

И. С. Ульяничев, Д. В. Винс

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.684

DOI: 10.24412/2073-0667-2024-1-74-97

EDN: FWATXX

В статье предложена архитектура математического и программного обеспечения распределенной вычислительной системы, построенной на базе мобильных устройств в общей сетевой инфраструктуре. Распределенная вычислительная система основана на проблемно-ориентированной модели выполнения задач (в зарубежной литературе известная как Task-Based Execution Model). Такая модель ориентирована на массовые распределенные вычисления, когда исходная задача допускает декомпозицию на тысячи и более независимых подзадач. Подобная организация вычислений характерна при решении оптимизационных и обратных задач, а также для методов типа Монте-Карло. Особенностью предложенной архитектуры распределенной вычислительной системы является тот факт, что независимые подзадачи требуют временных затрат, сопоставимых со «временем жизни» вычислительного узла. Это обстоятельство требует не только масштабируемости при решении конкретной задачи, но и целостности распределенной вычислительной системы в целом. Это требует от распределенной системы реконфигурируемости и гетерогенности. В статье мы предлагаем одну архитектуру распределенной гетерогенной вычислительной системы с гарантированной оценкой масштабируемости с учетом реконфигурации сети мобильных вычислительных устройств.

Ключевые слова: Task-Based Execution Model, мобильные вычисления, распределенные вычислительные системы.

Введение. Согласно современным оценкам, в настоящее время число используемых смартфонов сопоставимо с населением Земли, при этом среднее время использования смартфона ограничено семью часами в сутки. Наряду с высокой производительностью мобильных процессоров, уже сегодня распределенная вычислительная система на базе всех мобильных устройств позволяет достичь производительности порядка 10 эксафлопс, что превосходит суммарную производительность всех суперкомпьютеров списка Топ-500. Таким образом, наиболее перспективным вариантом построения распределенной вычислительной системы с потенциально бесконечной масштабируемостью являются системы на основе мобильных устройств, объединенных в единую мобильную или спутниковую сетевую инфраструктуру. Далее приведен краткий обзор моделей параллельных и распределенных вычислений и моделей управления ресурсами и потоками данных.

1. Парадигмы распределенных вычислений. На данный момент существует несколько базовых парадигм распределенных вычислений (для сравнения см. табл. 1).

Работа выполнена за счет гранта Российского научного фонда № 23-11-00014

Облачные вычисления [1, 2] — рыночная услуга, предлагающая клиентам вычислительные ресурсы по требованию без первоначальных капиталовложений с высокой масштабируемостью в соответствии с запросами. Для многих пользователей это предполагает снижение инфраструктурных рисков и экономию за счет совместного использования одних и тех же аппаратных ресурсов среди нескольких пользователей. Разработка и управление платформой финансируются клиентами, использующими облачные сервисы напрямую, путем аренды вычислительных ресурсов, или косвенно, с помощью онлайн-сервисов.

Грид-вычисления [3, 4] — это несколько устаревшая, но похожая на облачные вычисления форма распределенных вычислений, которая предоставляет доступ к вычислительным ресурсам, принадлежащим разным сотрудничающим организациям, через единый сервис. Грид с точки зрения сетевой организации представляет собой согласованную, открытую и стандартизованную среду, которая обеспечивает гибкое, безопасное, скоординированное разделение вычислительных ресурсов и хранение информации, которые являются частью этой среды, в рамках одной виртуальной организации. Грид-вычисления в настоящее время финансируются за счет государственных расходов и доступны как для проведения исследований, так и для обучения студентов. Однако, хотя грид-системы доступны многим общественным организациям, они недоступны широкой публике.

Добровольные вычисления подразумевают использование личных устройств добровольцев. Они организованы вокруг парадигмы общего пользования. Добровольцы не получают финансового вознаграждения за свой вклад, но могут получить общественное признание. Разработка добровольных вычислительных платформ финансируется правительствами за счет государственных исследовательских проектов, чтобы предоставить исследователям суперкомпьютерные мощности по гораздо более низкой цене. Эти платформы могут потенциально использовать миллиарды персональных устройств, хотя на данный момент количество устройств на основных платформах составляет порядка миллиона. В отличие от облачных вычислений, в данном подходе гораздо более низкий финансовый барьер для получения доступа к вычислительным устройствам. Главное преимущество по сравнению с облаками и гридами заключается в том, что большая часть расходов покрывается участниками. Они покрывают приобретение, эксплуатацию и обслуживание вычислительных устройств. В то же время сложность разработки и обслуживания таких платформ требует квалифицированных специалистов и дополнительных ресурсов.

Граничные (Edge) и туманные (Fog) вычисления основаны на облачной платформе и, следовательно, наследуют те же характеристики. Multi-access Edge Computing — это технология, которая предоставляет облачные и ИТ-услуги в непосредственной близости от мобильных абонентов, обслуживая чувствительные к задержке контекстно-зависимые приложения. Платформа Multi-access Edge Computing снижает задержку в сети, предоставляя возможности вычислений и хранения в пограничной сети, а также позволяет разработчикам приложений и поставщикам контента предоставлять контекстно-зависимые услуги. Multi-access Edge Computing подходит для таких приложений, как оптимизация динамического контента, разгрузка вычислений в IoT, аналитика мобильных больших данных и интеллектуальная транспортировка. Задачи этих классов не подходят для работы на мобильных или портативных устройствах, поскольку приложения требуют больших вычислительных мощностей и объемов систем хранения данных.

Децентрализованные и одноранговые (P2P) подходы [5, 6], в отличие от всех остальных парадигм, используют персональные устройства как для выполнения задач, так и для их координации. Обязанности серверов распределяются на все участвующие устройства, ис-

пользуя одноранговые алгоритмы. Это делает систему более устойчивой к ошибкам координаторов, устраняя их привилегированное положение. По сравнению с облачными, грид-вычислениями и добровольными вычислениями децентрализованные подходы исключают необходимость доверия со стороны операторов платформы/инструмента и используемых серверов. По сравнению с децентрализованными подходами добровольные вычислительные инструменты централизованы в том смысле, что координация вычислений выполняется на выделенном сервере. Однако, в отличие от децентрализованных платформ, разные пользователи создают разрозненные сети. Это значительно упрощает внедрение координаторов, обеспечивая при этом независимость от сбоя других пользователей.

Персональные добровольные вычисления [7] представляют собой новый подход, вытекающий из подхода добровольных вычислений. Он следует общей парадигме, но фокусируется на личных вычислительных потребностях программистов для приложений, представляющих личный или общественный интерес. Ключевая особенность подхода заключается в предоставлении распределенной вычислительной инфраструктуры без дополнительных инвестиций в оборудование. Более простая, чем типичные децентрализованные подходы, реализация использования устройств участников для координации и для вычислений снижает эксплуатационные расходы. Этот подход потенциально может использовать миллиарды доступных персональных устройств, и даже не в плане объединения всех устройств на одной платформе, а как объединение устройств во временные сети вокруг тысяч независимых проектов. По сравнению с облачными и грид-вычислениями персональные добровольные вычисления устраняют финансовые и административные барьеры. По сравнению с добровольными вычислениями значительно снижается технический барьер. Персональные добровольные вычисления, как и добровольные вычисления, естественным образом подходят для приложений, связанных с вычислениями, с множеством независимых задач.

2. Системы распределенных вычислений. На данный момент системы распределенных мобильных вычислений используют самые различные подходы. Основанные на MPI системы, не использующие оболочку BOINC [8–10], зачастую имеют одни и те же недостатки, связанные с масштабированием и дальнейшим развитием:

- для того, чтобы телефон был частью динамической сети, необходимы дополнительные меры безопасности, такие как VPN;
- статическая конфигурация кластера устройств;
- дополнительные манипуляции с памятью и правами доступа.

Заметим, что подобная архитектура может в большей степени раскрыть вычислительную мощность смартфонов. Однако, статичная конфигурация кластера и необходимость получения прав администратора на каждом используемом телефоне ставят под сомнение возможность адекватного расширения (добавление новых устройств) подобных систем.

BOINC представляет собой инфраструктуру для распределенных добровольных вычислений. На данный момент реализация BOINC для Android установлена и работает на десятках тысяч устройств. Также для BOINC в настоящее время существуют оболочки для C/C++, Java и Python. BOINC нуждается в центральном менеджере для развертывания. Каждый раз при создании проекта в BOINC необходимо развертывать приложение и внутри него настраивать разные версии для разных типов клиентов, например Windows x86, Linux x86 и Android ARM.

Для работы BOINC предоставляет API, определенный на языке C/C++. API содержит методы, которые должны выполняться задачей, чтобы начать и завершить ее выполнение, сообщить о ходе выполнения, получить доступ к входным и выходным файлам,

Таблица 1

Преимущества и недостатки парадигм распределенных вычислений

Название	Преимущества	Недостатки
Облачные вычисления	— масштабируемость в соответствии с запросами; — низкие капитальные риски.	— финансовый барьер в виде минимальной цены на услуги; — сложное закрытое API.
Грид-вычисления	— согласованная, открытая и стандартизованная среда; — изначально заложена безопасность среды.	— техническая сложность программирования; — бюрократическая сложность получения доступа; — недоступны широкой публике.
Добровольные вычисления	— потенциально бесконечное кол-во устройств для вычислений; — низкий финансовый барьер; — меньше административных барьеров; — часть расходов покрывается участниками.	— изменчивость персональных устройств; — необходимость поощрять и поддерживать участников; — автоматическая обработка ненадежности участников; — закрытые коды клиентов; — техническая сложность платформ.
Пограничные и туманные вычисления	— низкая задержка сети; — контекстно-зависимые услуги.	— решаемые задачи не подходят для вычисления на мобильных устройствах; — нет стандартизированной среды; — сложное управление мобильностью устройств; — неоднородность сетевых интерфейсов; — динамическая модель ценообразования; — проблемы с масштабируемостью и безопасностью.
Децентрализованные вычисления	— масштабируемость; — безопасность и устойчивость сети; — более дешевое оборудование; — отсутствие необходимости в сетевом менеджере; — низкий технический порог вхождения; — меньшее потребление сетевого трафика, чем в сети клиент/сервер.	— пользователи не могут централизованно создавать резервные копии файлов и папок; — к каждому компьютеру могут получить доступ другие компьютеры; — файлы не организованы централизованно; — низкая безопасность.
Персональные добровольные вычисления	— потенциально бесконечное кол-во устройств для вычислений; — не требуется дополнительных инвестиций.	— разнообразная программная среда; — важность простоты и кроссплатформенности инструментов; — ограниченное время использования; — не гарантируется доступность добровольцев

а также сообщить об ошибках. Чтобы задача выполнялась на платформе BOINC, она должна использовать API или обертки. Если для данной платформы доступны оболочки, нет необходимости переписывать приложения для вставки вызовов BOINC. Оболочек для запуска задач на Android пока нет. Поэтому приложение необходимо модифицировать, используя BOINC API для управления началом и завершением задачи, а также операциями ввода/вывода.

Среди недостатков BOINC:

- в BOINC невозможно сформировать ad-hoc сетку, так как для этого нужен центральный менеджер;
- использование MPI ограничено одним узлом;
- необходимо модифицировать исходный код программ;
- с версии 9.0 Android убивает BOINC, если он работает в фоновом режиме или если выключен дисплей.

Распределенные системы, основанные на облаке, предлагают по сути безостановочные вычисления, не зависящие по большей части от интернет соединения и от заряда батареи, за счет возможности переключения вычислений с телефона на аватар в облаке. Однако из-за этого возникает вопрос целесообразности использования такой сложной системы в распределенных научных вычислениях, если данные для вычислений не будут генерироваться непосредственно самими мобильными устройствами или не будут заранее распределены между устройствами. Также сложно определить целесообразность использования смартфона для вычисления, когда можно все перенести в облако, учитывая его постоянную доступность и более мощные ресурсы.

Общая концепция системы распределенных вычислений на базе мобильных устройств с клиент-серверной архитектурой является довольно распространенной [11–14]. Система зачастую состоит из серверной части, реализующей хранение данных, алгоритмы управления задачами и распределения ресурсов, и мобильного клиента для операционной системы Android, отвечающего за выполнение выделенной задачи и отправки результата обратно на сервер. Нарботанные годами общие требования к системе, а также концепция взаимодействия мобильного клиента с сервером позволяют учитывать множество нюансов работы системы. Такие системы зачастую пишутся сразу для конкретной платформы, либо под несколько платформ, организуя работу с сервером через API.

Предлагаемая нами система DHCA имеет клиент-серверную архитектуру и заточена под систему Android. Мобильный клиент написан с использованием новейших подходов Android к фоновым вычислениям, что позволяет увеличить производительность, а также повысить безопасность конечных пользователей. Система учитывает все особенности взаимодействия клиент-серверной архитектуры для распределенных систем, а также решает проблемы, связанные с недоступностью вычислительных узлов, путем автоматического перераспределения задач. Подход Task-Based Execution Model позволяет уменьшить количество передаваемой информации между сервером и узлом, что в свою очередь повышает общую производительность. Открытый исходный код повышает доверие к системе со стороны конечных пользователей, что может благоприятно сказаться на количестве активных участников вычислений.

В работе мы сосредоточимся на модели распределенных вычислений на основе реконфигурируемой гетерогенной вычислительной системы мобильных устройств, направленной на решение оптимизационных и обратных задач. В третьем разделе детально описана архитектура распределенной вычислительной системы. Четвертый раздел посвящен использованию системы для решения некоторых прикладных и фундаментальных задач. В пятом разделе подняты некоторые дискуссионные вопросы. В шестом разделе приведено заключение.

3. Архитектура системы распределенных мобильных вычислений. В основе разрабатываемой системы распределенных мобильных вычислений лежит проблемно-

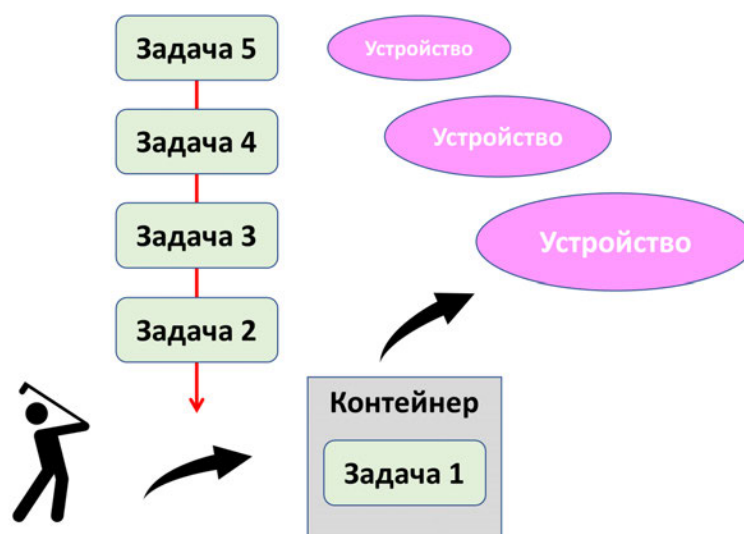


Рис. 1. Схема проблемно-ориентированной модели выполнения задач

ориентированная модель выполнения задач, известная в зарубежной литературе как Task-Based Execution Model. Концептуальная схема модели представлена на рис. 1.

Модель основана на декомпозиции общей задачи на набор мелких однотипных подзадач, различающихся набором входных параметров. Такие подзадачи, несмотря на простоту параллельной реализации, охватывают широкое множество приложений: классическая задача оптимизации двигателя летательного аппарата [15], решение обратной задачи восстановления профиля звезды по наблюдаемому остатку взрыва сверхновой [16] и разработка фреймворка для решения гидродинамических задач на основе адаптивных сеток [17]. Каждая задача характеризуется набором входных параметров для некоторого предустановленного на мобильные устройства функционала. При поступлении очередной подзадачи (набора входных параметров) они отправляются на первое свободное доступное устройство. На рис. 1 используется понятие «контейнер» в качестве сформированной на счет задачи, то есть в момент упаковки в контейнер приложение, установленное на мобильном устройстве, получает набор входных данных и отправляется на выполнение на мобильном устройстве. Результат выполнения задачи (выходные данные) возвращается обратно. Важно отметить, что никак не ограничивается логика планировщика задач, то есть после получения некоторого множества результатов список задач может быть модифицирован, и на счет будут отправлены задачи из нового списка. Таким образом мы можем расширить систему на расчеты итерационных и нестационарных процессов с соответствующей дискретизацией.

3.1. *Архитектура Web-клиента.* В данном разделе описывается проектирование и реализация web-клиента системы ДНСА. Основными функциональными возможностями web-клиентской части системы являются следующие:

- отображение выполняемых и выполненных заданий распределенных вычислений;
- создание задач распределенных вычислений;
- загрузка начальных параметров задачи;
- проверка на ошибки загружаемых условий;
- отслеживание хода вычислений;
- выгрузка результатов.

Web-клиент представляет собой web-приложение, состоящее из трех HTML-форм:

- общая таблица задач;
- форма ввода новой задачи;
- форма детального просмотра задачи.

Интерфейс реализован с помощью JavaScript-библиотеки jQuery, позволяющей отправлять запросы и получать данные от сервера в асинхронном режиме. Взаимодействие с сервером осуществляется посредством HTTP-запросов REST API, написанной на языке PHP. Web-клиент является закрытым сегментом системы с ограниченным доступом и предполагает использование только компетентными, доверенными сотрудниками.

Форма с общей таблицей задач содержит таблицу с отображением всех текущих выполняемых и выполненных заданий распределенных вычислений, кнопку для добавления новой задачи и механизм фильтрации строк. Таблица содержит следующие сведения о задаче:

- порядковый номер;
- тип задачи;
- название проекта;
- описание проекта;
- дата создания: в формате дд.мм.гггг чч:мм;
- дата завершения: в формате дд.мм.гггг чч:мм;
- статус.

Форма автоматически обновляется каждые 30 секунд (разумеется, периодичность обновления может быть изменена и адаптирована под решаемую задачу). На текущий момент механизм фильтрации работает только по типу задачи, но может быть расширен произвольным образом.

В столбце «статус» отображается текущий статус задачи, который может принимать следующие значения: черновик, в работе, завершен. Статус «Черновик» нужен для предварительной проверки загруженных данных, перед отправкой на вычисления. При этом статусе в поле будет отображаться кнопка «В работу» для перевода задачи в следующее состояние. Сразу после перевода задачи в статус «В работе», сервер рассылает всем мобильным клиентам широковещательное сообщение о наличии новой задачи для вычислений. В столбце кнопка сменится на отображение количества решенных и всего блоков. По окончании вычислений, количество решенных блоков в столбце «статус» сменится на ссылку для вызова Reduce-функции сервера для объединения и скачивания результата.

Общую схему последовательности работы системы можно увидеть на рис. 2.

При нажатии на порядковый номер задачи откроется форма для детального просмотра данных по выбранному проекту, без возможности редактирования.

При нажатии кнопки «Добавить проект» на форме с общей таблицей откроется форма ввода новой задачи. После заполнения всех полей при нажатии кнопки «Сохранить проект» при корректном заполнении всех обязательных полей, данные отправятся для обработки и сохранения на сервер. При неправильном заполнении будет выведено предупреждение с указанием на некорректное поле. После завершения обработки пользователь будет перемещен на форму с общей таблицей, где сможет наблюдать внесенную задачу в статусе «Черновик».

3.2. Архитектура сервера. Сервер системы DHCA представляет собой web-сервер с поддержкой PHP, с подключенной базой MySQL. Основными функциональными возможностями серверной части системы являются:

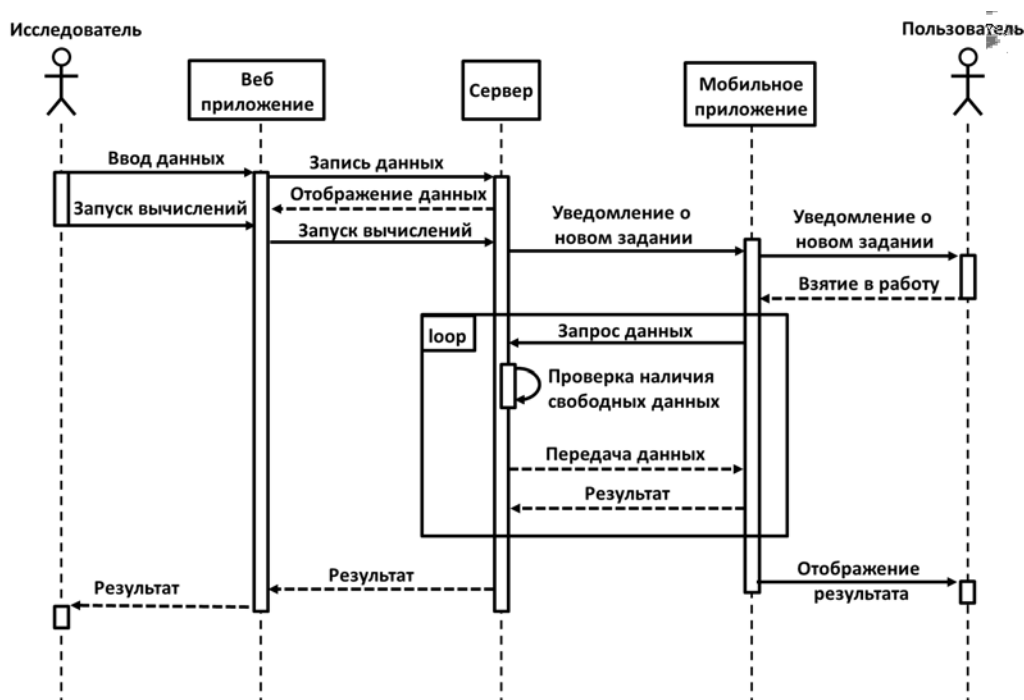


Рис. 2. Диаграмма последовательности работы системы

- внесение данных о проектах в базу;
- хранение данных о проектах;
- предоставление данных о проектах;
- разбиение начальной информации на вычислительные блоки;
- хранение данных о результатах вычисления блоков;
- регистрация мобильных клиентов в системе;
- авторизация мобильных клиентов в системе;
- распределение вычислительных блоков среди мобильных клиентов;
- рассылка информационных сообщений мобильным клиентам;
- объединение и предоставление результатов вычислений.

Благодаря модульной архитектуре используемой API, к системе можно подключить любую реляционную СУБД, поддерживающую язык структурированных запросов SQL. За подключение базы данных и обработку запросов отвечает класс `database`, написанный на PHP. Для смены СУБД достаточно подменить данный класс, оставив основные используемые функции. Также для смены СУБД необходимо соответствие схемы таблиц баз данных. Схема оригинальной БД представлена на рис. 3. БД включает в себя таблицы с данными о самом проекте (`dhca_projects`), а также о данных проекта, пересылаемых для вычисления на мобильные клиенты (`dhca_data_project`). Также туда входит таблица с данным для авторизации пользователей на web-клиенте и таблицы с описаниями статусов и типов проектов.

Функции API для web-клиента предназначены для взаимодействия пользователя с базой данных посредством web-интерфейса клиента. Отдельно стоит остановиться на работе файлов `culc_functions.php` и `get_result.php`, которые по сути являются Map и Reduce реализациями паттерна MapReduce. Файл `culc_functions.php` содержит функцию для разде-

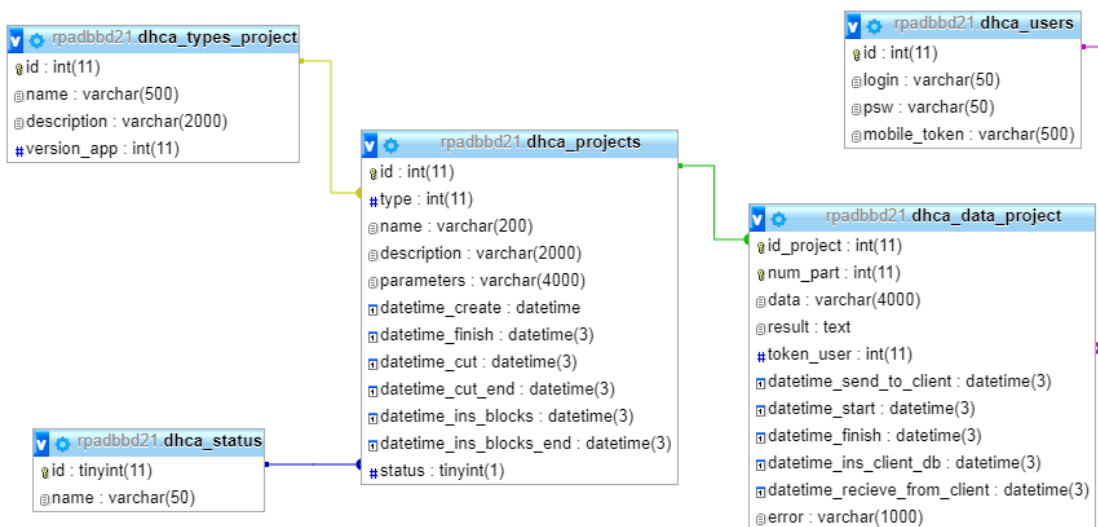


Рис. 3. Схема таблиц базы данных

ления входных данных на вычислительные блоки в зависимости от типа задач и входных параметров. Выступает начальным звеном Мар, разбивая исходный массив, но не участвует в его дальнейшем распределении. На вход принимает параметры задачи, текущее количество мобильных клиентов системы, желаемое количество вычислительных блоков (необязательно), длину блока (необязательно), тип задачи. Количество вычислительных блоков, на которое разбивается исходная задача, высчитывается следующим образом. Если входящий параметр «количество блоков» был равен нулю, то за первоначальное количество берется количество мобильных клиентов, умноженное на 5, иначе значение входящего параметра. Затем если входящий параметр «длина блока» был равен нулю, то высчитывается минимальная длина блока, иначе берется значение входящего параметра. Минимальная длина блока высчитывается в зависимости от специфики решаемой задачи. Например, для задачи трассировки лучей из параметров перемножением высоты на ширину берется общее количество пикселей и делится на полученное ранее количество блоков. Затем, проходя по массиву пикселей, выбираются координаты точек, разделенные минимальной длиной блока. Результатом работы функции является ассоциативный массив, каждый элемент которого содержит координаты x , y и размер блока — $size$. Длина массива равна количеству вычислительных блоков задачи. Каждый элемент данного массива записывается в базу данных сервера с уникальным ключом, состоящим из идентификатора задачи и порядкового номера блока, в дальнейшем эти элементы выступают основными источниками параметров вычислительных блоков для участвующих в расчетах мобильных узлов, наравне с исходными параметрами задачи.

Файл `get_result.php` возвращает результат решенных задач, объединяя решенные блоки. Выступает в качестве Reduce-метода, объединяя полученные результаты. На вход получает идентификатор проекта в базе и его тип. В зависимости от типа объединение может происходить различным образом. Например, при расчете задачи рейтрейсинга мобильный клиент в качестве результата, возвращает массив пикселей длиной, равной длине вычислительного блока. Каждый пиксель, в свою очередь, состоит из массива 3-х целочисленных значений в диапазоне от 0 до 255, представляющего собой цвет пикселя в формате RGB.

При сборке алгоритм запрашивает из базы данных все полученные результаты и с помощью php функций `imagecolorallocate` и `imagegetpixel` собирает по порядку все пиксели всех блоков в конечное изображение в формате PNG. Данное изображение является результатом работы и доступно для скачивания в web-приложении. С другими типами задач производятся аналогичные действия в зависимости от того, что требуется получить на выходе.

Функции API для мобильного клиента предназначены для взаимодействия пользователя с базой данных посредством интерфейса Android приложения. Основным функционалом данного API являются:

- регистрация мобильных клиентов в системе;
- авторизация мобильных клиентов в системе;
- распределение вычислительных блоков среди мобильных клиентов;
- рассылка информационных сообщений мобильным клиентам;
- предоставление данных о проектах.

Файл `get_data_project.php` является продолжением реализации парадигмы Map, обеспечивая динамическое распределение вычислительных блоков среди клиентов. При вызове выбирает случайный блок среди ограниченной выборки доступных блоков. Доступными являются блоки, не зарезервированные за каким-либо пользователем. Если свободный блок существует, он закрепляется за пользователем и отправляется ему на обработку. Если свободных блоков нет, тогда происходит следующая итерация, на которой проверяется, есть ли блоки без результата вычисления. Если таких блоков нет, то проекту присваивается статус «Завершен». Если такие блоки есть, то происходит проверка времени отправки блока клиенту. В текущей версии системы, если с момента отправки прошло больше суток и результат не поступил в базу, то система резервирует данный блок за новым пользователем и отправляет его ему на обработку. Весь алгоритм повторяется, пока не завершится проект, либо клиент не остановит вычисления на своем устройстве.

Важной особенностью является тот факт, что данный алгоритм осуществляет реализацию ReMap, что делает данную систему надежной за счет перераспределения блоков с недоступных узлов между доступными устройствами.

3.3. Архитектура клиента. Мобильный клиент представляет собой приложение, установленное на смартфоне под управлением операционной системы Android версии 6.0 (Marshmallow) и выше. Клиент написан на языке Kotlin с использованием шаблона проектирования MVVM. Мобильные клиенты являются основными вычислительными узлами системы, на которых происходит вся основная обработка данных. Основными функциональными возможностями данного блока системы являются:

- регистрация в системе;
- авторизация в системе;
- отображение информации о доступных и решенных проектах;
- кэширование данных;
- расчет вычислительных блоков в фоновом режиме.

Работа пользователя с Android-приложением при первом запуске начинается с окна регистрации в системе. При регистрации пользователь вводит логин и пароль, под которыми будет осуществляться дальнейшая работа. При регистрации приложение устанавливает соединение с платформой Firebase Cloud Messaging (FCM), отвечающей за работу с уведомлениями. Успешное соединение с данной платформой сопровождается выдачей уникального токена для устройства, если он не был получен ранее. Токен вместе с логи-

ном и паролем с помощью API записывается в базу, обратно устройство получает идентификатор записи в базе данных. Этот идентификатор сохраняется на устройстве, и при дальнейших запусках приложения проверяется его наличие. Он также используется для резервирования блоков за пользователем.

После регистрации пользователь попадает в главное меню, из которого он может попасть в «Доступные исследования» и в «История». В доступных исследованиях можно просмотреть и выбрать текущие незавершенные задачи. Список доступных задач зависит от версии клиента. Запрос возвращает список доступных проектов и записывает их во внутреннюю базу данных клиента. При отсутствии Интернет-соединения из базы извлекаются ранее закачанные данные. После перевода задачи в web-клиенте в статус «В работе», всем мобильным клиентам приходит уведомление о появлении новой задачи, и список задач обновляется.

При выборе конкретной задачи открывается окно с детальной информацией о задаче: название, тип, краткое описание, статус, процент выполненных блоков. Если задача не находится в статусе «Завершен», то в детальном окне будет доступна кнопка «Начать». Стоит отметить, что если пользователь нажал кнопку «Начать» в отсутствии интернет-соединения, то WorkManager запустит вычисления автоматически, как только соединение появится. Класс в Android — CoroutineWorker позволяет запускать процессы в потоко-безопасном режиме с помощью Kotlin Coroutines. Основная функция воркера doWork() выполняет следующие действия:

1. Отправляет на сервер запрос на получение вычислительных блоков.
2. Если ответ содержит вычислительные блоки, то они записываются в локальную базу данных.
3. Вызывается функция, возвращающая из локальной БД нерешенные блоки по идентификатору проекта.
4. Создается цикл из нерешенных данных в локальной БД. В ходе каждой итерации выполняются следующие действия:
 - a. Записывается дата начала вычисления в миллисекундах.
 - b. Производится расчет по входным параметрам, принимающим на вход параметры проекта и параметры блока.
 - c. Записывается дата окончания вычисления в миллисекундах.
 - d. При успешном вычислении результат записывается в локальную базу данных.
 - e. Вызывается функция sendResult, которая осуществляет отправку решенных блоков на сервер.
 - f. Если ответ содержит вычислительные блоки, то они записываются в локальную базу данных.
 - g. Вызывается функция, возвращающая из локальной БД нерешенные блоки по идентификатору проекта.

В случае отсутствия интернет-соединения отправка результатов прерывается до появления соединения.

Цикл продолжает работу, пока сервер возвращает свободные блоки, либо пока пользователь не остановит или не откажется от вычисления, нажав кнопку «Завершить вычисления». Для экономии ресурсов устройства при отказе от вычислений производится очистка локальной БД от блоков данного проекта. При поступлении на сервер результатов сервер каждый раз проверяет, есть ли еще вычислительные блоки, по которым нет результата, если таких блоков больше нет, то задача считается выполненной. В базе данных сервера в

проекте меняется статус на «Завершен», а всем участвовавшим в его вычислениях мобильным клиентам функция sendFCM рассылает информационное сообщение. При получении этого сообщения в локальной базе клиента также меняется статус проекта, при этом становятся недоступны функциональные кнопки, также для экономии ресурсов устройства производится очистка локальной БД от блоков данного проекта.

3.4. *Гарантированная оценка масштабируемости системы.* С учетом представленной архитектуры дадим оценку масштабируемости системы. Мы предполагаем, что для расчета среднего времени работы системы нам достаточно знать следующие значения:

t — среднее время, необходимое для расчета одной подзадачи;

N — количество независимых подзадач;

P — количество участвующих в расчете устройств.

Тогда, обозначив через T среднее время работы системы, в идеальном виде мы получим следующую формулу:

$$T = \frac{Nt}{P} \quad (1)$$

В реальной модели нам следует учитывать еще несколько параметров:

t_0 — среднее время, необходимое для первоначальной разбивки исходных данных на независимые подзадачи;

t_w — среднее время записи (write) одной начальной подзадачи в базу данных после разбивки;

t_s — среднее время, необходимое для получения одной подзадачи клиентом с сервера (send);

t_r — среднее время, необходимое для получения рассчитанной подзадачи сервером с клиента (receive);

t_v — среднее время, необходимое для записи результата рассчитанной подзадачи в базу данных (save).

Получим среднее время расчета на P устройствах:

$$T(P) = t_0 + N(t_w + t_v) + P(t_s + t_r) + \frac{Nt}{P} \quad (2)$$

Запишем ускорение:

$$S(P) = \frac{T(1)}{T(P)} = \frac{t_0 + N(t_w + t_v) + (t_s + t_r) + Nt}{t_0 + N(t_w + t_v) + P(t_s + t_r) + \frac{Nt}{P}} \quad (3)$$

Функция ускорения обладает следующими свойствами:

$$S(1) = 1$$

$$S(P) > 0$$

$$\lim_{P \rightarrow \infty} S(P) = \frac{1}{P}$$

Функция $S(P)$ имеет единственный экстремум на интервале $(1; +\infty)$ при

Таблица 2

Параметры верификации оценки масштабируемости

t , мс	N	t_0 , мс	t_w , мс	t_s , мс	t_r , мс	t_v , мс
10	1000000000	50000	1	100	20	0.3
100	1000000000	50000	1	100	20	0.3
1000	1000000000	50000	1	100	20	0.3

$$P = \sqrt{\frac{Nt}{t_s + t_r}} \quad (4)$$

Суть формулы — число мобильных устройств, при котором достигается максимум ускорения.

На задаче трассировки лучей, которая описана далее, варьированием передаваемых и принимаемых данных, а также сложностью моделируемой сцены мы показали достижение максимума ускорения при малом числе мобильных устройств. К сожалению, объективные ограничения в числе доступных для проведения экспериментов мобильных устройств не позволили нам провести более масштабные эксперименты для верификации оценки масштабируемости. Тем не менее, верификация оценки масштабируемости на большом числе устройств возможно при использовании аппарата имитационного моделирования, что и было нами сделано далее.

Была проведена верификация оценки масштабируемости на имитационной системе моделирования. Использовались следующие параметры, представленные в табл. 2.

В ходе эксперимента решалась задача столкновения белых карликов [18], детали которой будут описаны далее, в следующей постановке. Есть N — число независимых подзадач. Подзадача на вход принимает 5 вещественных чисел и на выходе отдает одно вещественное число. Решать будем эти подзадачи с использованием сети из P мобильных устройств. Каждое устройство одновременно обрабатывает только одну подзадачу. Подзадачи обрабатываются последовательно, каждое устройство случайно выбирает еще не решенную и не занятую никем подзадачу из общего пула, закрепляет ее за собой и после обработки переводит ее в пул решенных. И так, пока все N подзадач не будут переведены в пул решенных. Пусть t — среднее время решения одной подзадачи. Будем рассматривать три варианта сложности задачи: $t = 1$ секунда (Large), $t = 0.1$ секунды (Middle), $t = 0.01$ секунды (Small). Обозначим t_0 — среднее время декомпозиции вычислений. Это некоторое постоянное время на инициализацию базы данных подзадач, независимое от числа задач и мобильных устройств. Далее нам нужно входные (до начала вычислений) и выходные (после вычислений) параметры каждой независимой подзадачи записать в базу данных. Обозначим эти времена t_w — среднее время записи (Write) входных данных подзадачи и t_v — среднее время сохранения (save) результата вычислений в базе. Эти времена тратятся на каждую из N подзадач. Теперь мы должны передать данные по мобильной сети каждому устройству для счета и получить посчитанные данные от каждого устройства мобильной сети. Обозначим эти времена t_s — среднее время отправки (Send) данных устройствам и t_r — среднее время получения (Receive) данных от устройств. Эти времена затрачиваются на отправку/получение на/с каждого мобильного устройства.

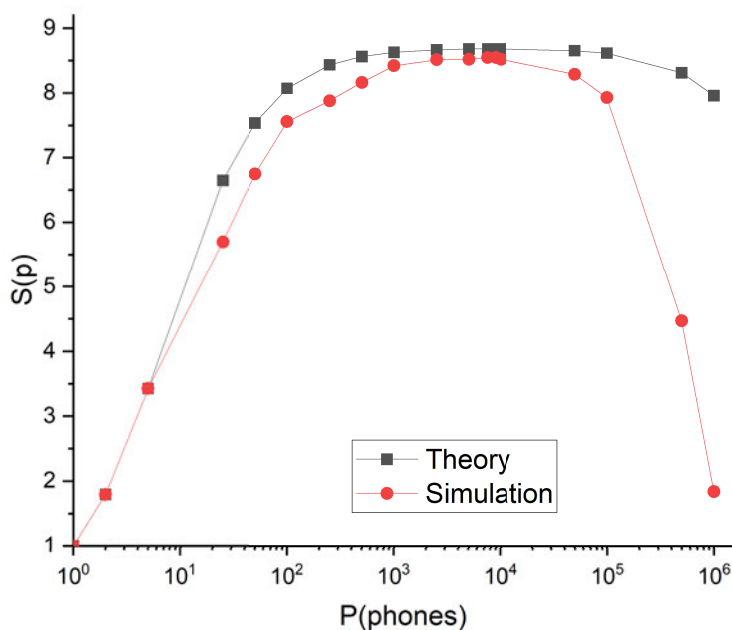


Рис. 4. Сравнение ускорений при сложности задачи $t = 0.01$ секунды

Ускорение, полученное на имитационной модели, сравнивались с теоретическими расчетами по формуле. Результаты представлены на рис. 4 для сложности $t = 0.01$ секунды, на рис. 5 для сложности $t = 0.1$ секунды, и на рис. 6 для сложности $t = 1$ секунда.

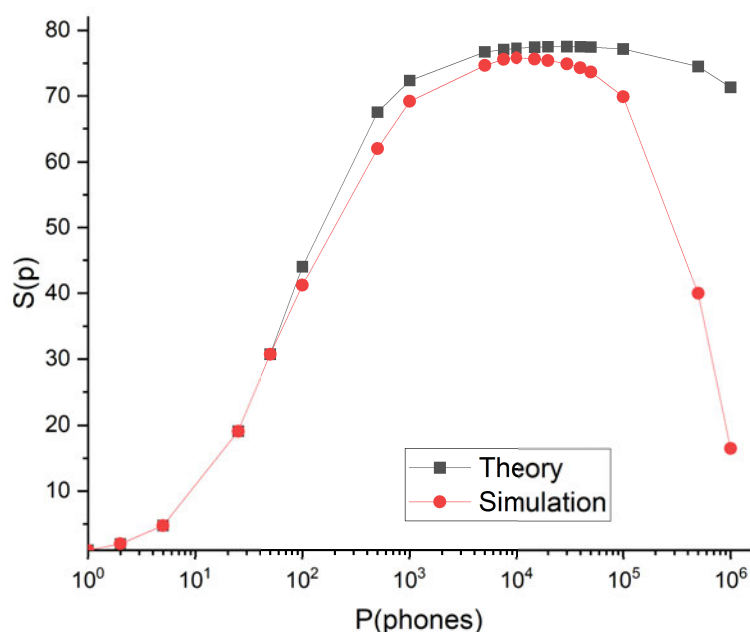
В полученных результатах видно, что во всех случаях теоритические и практические максимумы различаются не более чем на 6 %.

На основе проведенных экспериментов с использованием имитационной модели мы можем сделать вывод, что оценка времени счета, ускорение и число мобильных устройств, при котором достигается максимум ускорения, количественно соответствуют результатам, полученным на имитационной модели. Таким образом, мы делаем вывод об успешной верификации гарантированной оценки масштабируемости системы распределенных вычислений.

4. Использование системы для решения тестовых и прикладных задач. Для демонстрации робастности разработанной системы приведем решение трех задач, в которых каждая элементарная задача имеет различную нагрузку на процессор и на сетевую инфраструктуру. Выделим три таких задачи:

1) Задача обратной трассировки лучей, которая характеризуется значительным объемом передаваемых данных для формирования сцены и получаемых данных блока полного изображения. При этом время на трассировку лучей значительно больше относительно времени передачи данных.

2) Задача столкновения белых карликов, которая формулируется в одномерной постановке и характеризуется передачей всего шести значений для описания состояния карликов перед столкновением и одним выходным значением, описывающим максимальную температуру при столкновении. При этом время счета сравнимо со временем передачи данных. Задача имеет аналитическое решение, требующее разрешение нелинейных уравнений, и подробно описана в работе [19].

Рис. 5. Сравнение ускорений при сложности задачи $t = 0.1$ секунды

3) Задача ядерного горения углерода в белых карликах, которая формулируется в виде изменения концентраций основных изотопов при ядерном горении углерода в ходе решения систему ОДУ. Задача характеризуется входными данными в виде двух значений (начальной температуры и плотности углерода), описывающих состояние вырожденного газа, и параметра времени, в которого происходит эволюция изотопов. Результат представляется в виде семи возвращаемых значений, описывающих концентрацию каждого изотопа. Отметим, что время счета ОДУ значительно превышает время передачи данных. Постановка задачи и полученные результаты подробно описаны в работе [20].

Далее приведем подробности решения задач. Задачи, связанные с эволюцией материала белых карликов, подробно описаны в работах [19–20], поэтому при их описании будут приведены краткая формулировка и основной результат, который был получен на анализе множественного решения задач с помощью разработанной системы.

4.1. *Обратная трассировка лучей.* Важной для нас особенностью задачи обратной трассировки лучей является возможность организовать вычисление каждого отдельного пикселя или блока пикселей. Основная идея метода трассировки лучей состоит в исследовании поведения каждого луча, проходящего через каждую точку (пиксель) на экране. Каждый луч начинается из определенной точки (глаза), проходит через точку экрана и заканчивается на пересечении с объектом, в результате в каждой точке экрана формируется цвет, зависящий от освещения точки пересечения с объектом. Алгоритм метода для каждой точки (пикселя) можно представить следующим образом:

- Формирования луча, выходящего из глаза и проходящего через точку на экране.
- Нахождение пересечений со всеми объектами.
- Выбор ближайшего пересечения (определение ближайшего, то есть видимого пересечения).
- Проверка на нахождение точки в тени от других объектов.
- Формирование луча, направленного к источнику освещения.

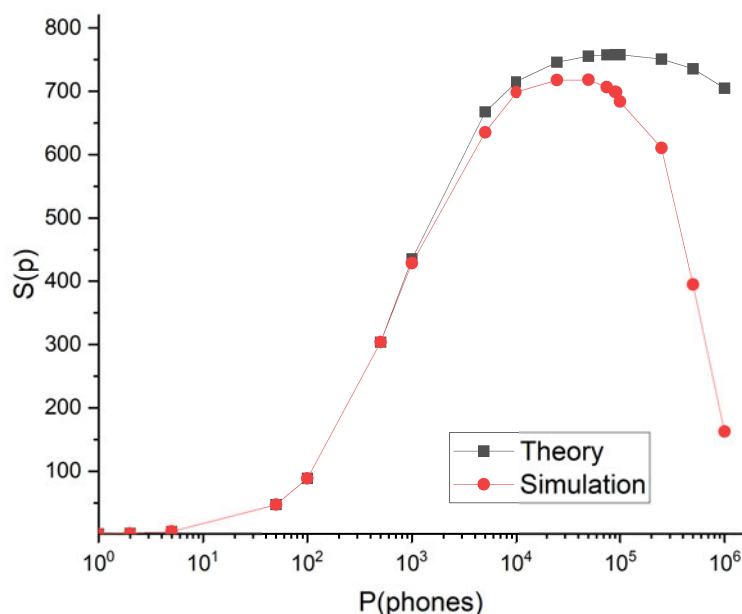


Рис. 6. Сравнение ускорений при сложности задачи $t = 1$ секунда

— Если имеет место хотя бы одно пересечение с другими объектами, то точка находится в тени.

- Определение нормали в точке.
- Формирование цвета с учетом освещения.
- Рисование пикселя.

Будем считать, что точка наблюдения находится в начале координат и проекция точки наблюдения на экран находится в середине.

В качестве примера трассировки лучей, для первого эксперимента была выбрана задача построения сцены с зеркальными сферами в разрешении 4096×3072 (4K). На сцене, изображенной на рис. 7, две сферы (зеленая и фиолетовая) имеют коэффициент зеркального отражения 1, синяя сфера имеет коэффициент отражения 0.5, красная сфера не имеет коэффициента отражения. Коэффициент отражения шахматной поверхности задан 0.5. Были проведены прогоны с разными размерами блоков (в пикселях) для одного смартфона.

4.2. Столкновение белых карликов. Одномерная постановка задачи центрального столкновения белых карликов состоит в рассмотрении задачи о распаде разрыва — столкновения двух волн с заданными параметрами плотности, температуры и скорости. Аналитическое решение задачи о распаде разрыва с использованием звездного уравнения состояния приведено в работе [19]. На рис. 8 приведена фазовая плоскость (U, T) — скорость столкновения и минимальная температура белого карлика, в которой отмечены области взрыва термоядерных сверхновых разного типа, разделенные пунктирной линией.

На основании многочисленных вычислительных экспериментов средствами разработанной распределенной вычислительной системы на базе мобильных устройств мы смогли сформировать такую фазовую плоскость.

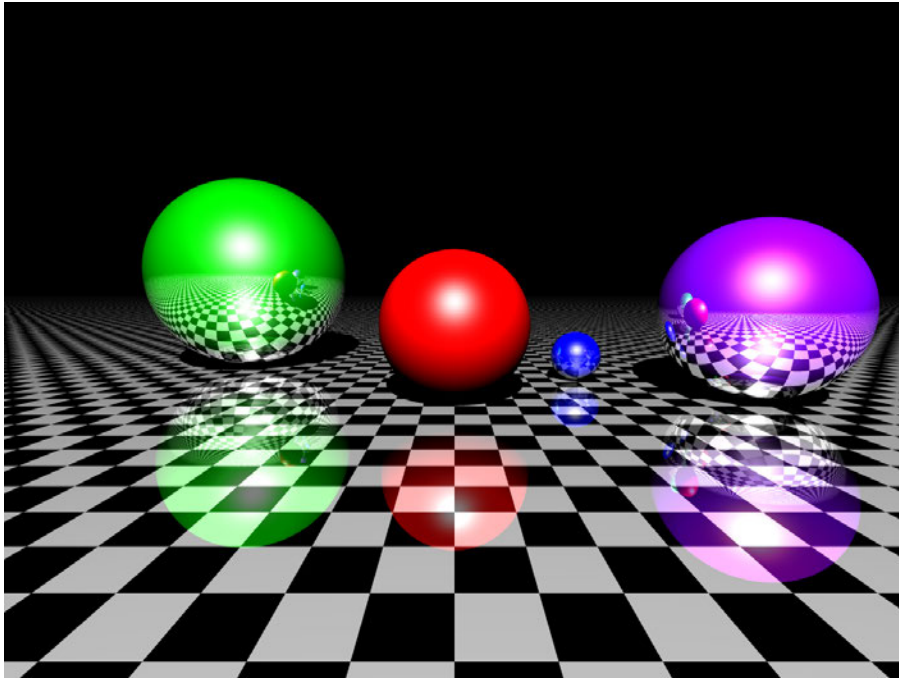


Рис. 7. Генерируемая сцена

4.3. *Ядерное горение углерода.* Будем рассматривать семь ключевых изотопов материала белых карликов: ${}^4\text{He}$, ${}^{12}\text{C}$, ${}^{16}\text{O}$, ${}^{20}\text{Ne}$, ${}^{24}\text{Mg}$, ${}^{28}\text{Si}$, ${}^{56}\text{Ni}$ и следующий набор альфа-цепи ядерных реакций:

- 1) ядерное горение гелия $3 \times {}^4\text{He} \rightarrow {}^{12}\text{C} + \gamma$,
- 2) синтез кислорода из углерода ${}^{12}\text{C} + {}^4\text{He} \rightarrow {}^{16}\text{O} + \gamma$,
- 3) синтез неона из кислорода ${}^{16}\text{O} + {}^4\text{He} \rightarrow {}^{20}\text{Ne} + \gamma$,
- 4) синтез магния из неона ${}^{20}\text{Ne} + {}^4\text{He} \rightarrow {}^{24}\text{Mg} + \gamma$,
- 5) синтез кремния из магния ${}^{24}\text{Mg} + {}^4\text{He} \rightarrow {}^{28}\text{Si} + \gamma$,
- 6) ядерное горение углерода ${}^{12}\text{C} + {}^{12}\text{C} \rightarrow {}^{24}\text{Mg}$,
- 7) ядерное горение углерода и кислорода ${}^{12}\text{C} + {}^{16}\text{O} \rightarrow {}^{28}\text{Si}$,
- 8) ядерное горение и синтез кислорода до никеля $2 \times {}^{16}\text{O} + 6 \times {}^4\text{He} \rightarrow {}^{56}\text{Ni} + 6 \times \gamma$,
- 9) ядерное горение и синтез кремния до никеля ${}^{28}\text{Si} + 7 \times {}^4\text{He} \rightarrow {}^{56}\text{Ni} + 7 \times \gamma$,
- 10) фотодезинтеграция кремния ${}^{28}\text{Si} + \gamma \rightarrow {}^{24}\text{Mg} + {}^4\text{He}$,
- 11) фотодезинтеграция магния ${}^{24}\text{Mg} + \gamma \rightarrow {}^{20}\text{Ne} + {}^4\text{He}$,
- 12) фотодезинтеграция неона ${}^{20}\text{Ne} + \gamma \rightarrow {}^{16}\text{O} + {}^4\text{He}$,
- 13) фотодезинтеграция кислорода ${}^{16}\text{O} + \gamma \rightarrow {}^{12}\text{C} + {}^4\text{He}$,
- 14) фотодезинтеграция углерода ${}^{12}\text{C} + \gamma \rightarrow 3 \times {}^4\text{He}$,
- 15) фотодезинтеграция никеля ${}^{56}\text{Ni} + 7 \times \gamma \rightarrow {}^{28}\text{Si} + 7 \times {}^4\text{He}$.

Подробности формирования ОДУ, численный метод решения и приближенное решение на основе серии вычислительных экспериментов средствами разработанной распределенной вычислительной системы на базе мобильных устройств приведены в работе [20]. Одним из важнейших результатов такого анализа является равновесное соотношение между изотопами кислорода, углерода и кремния:

$$k_{13}M({}^{16}\text{O}) = 2k_6M^2({}^{12}\text{C}) + k_{10}M({}^{28}\text{Si}) + k_{14}M({}^{12}\text{C}),$$

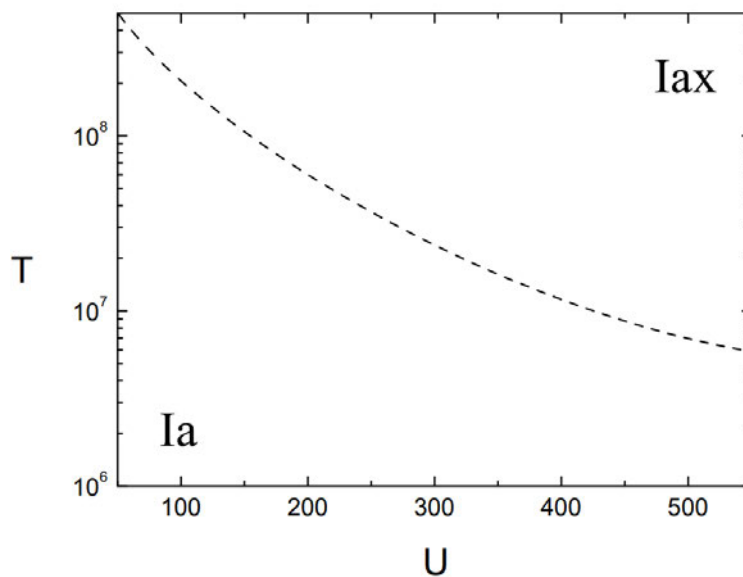


Рис. 8. Фазовая плоскость начальных параметров для взрыва различных типов термоядерных сверхновых звезд

где $k_6, k_{10}, k_{13}, k_{14}$ — скорость соответствующих ядерных реакций, $M(^{12}C), M(^{16}O), M(^{28}Si)$ — концентрация углерода, кислорода и кремния соответственно. Такое состояние определяет неполное сгорание углеродно-кислородного карлика с образованием изотопа кремния, который лежит в основе межзвездной пыли.

5. Дискуссия. В этом разделе сформулируем некоторые дискуссионные моменты, которые связаны с разработанной распределенной вычислительной системой на базе мобильных устройств.

1) В работе рассмотрены задачи, постановка которых подразумевает распределенные вычисления. В то же время архитектура системы не накладывает ограничения на логику работы сервера, и система в целом может быть рассмотрена как системы с ускорителями, реализованные с помощью мобильных устройств. Поэтому мы без каких-либо ограничений можем реализовать код на основе графических ускорителей, например из статьи [21], либо других ускорителей, например для ускорителей Intel Xeon Phi из статьи [22].

2) Представленная система предполагает использование с мобильными устройствами на базе Android, что может показаться проблемой в плане кросс-платформенности. На данный момент Android занимает большую часть сегмента мобильных устройств, а учитывая международную обстановку и возможные ограничения в технике Apple, система Android выглядит наиболее привлекательным вариантом. К тому же отечественная OS Аврора поддерживает установку арк-файлов Android. Также следует сказать, что приложение для мобильного клиента написано на языке Kotlin, который сам по себе используется не только в Android, и при желании на Kotlin можно написать как серверное, так и десктопное приложение. А если добавить сюда работу с сервером через API, то кросс-платформенность уже не кажется проблемой.

3) Многие распределенные системы имеют проблемы с безопасностью пользователей, когда они точно не знают, что загружают к себе на телефон и что в итоге считают. Такие приложения имеют закрытые коды или же архитектуру, при которой вычислительный код

передается клиенту для выполнения. Данная система подразумевает открытый исходный код, на клиента передаются лишь параметры для расчетов, все алгоритмы задаются в коде клиента изначально. К тому же подразумевается распространение через «магазины приложений», что также подразумевает дополнительную проверку кода со стороны «магазинов».

4) На данный момент мобильный клиент никак не следит за расходом трафика. Система построена на концепции добровольного вычисления, т. е. пользователь сам принимает решения, в каких проектах и в каком объеме он хочет участвовать. К тому же передаваемые данные между приложением и сервером имеют объем не больше текстовых сообщений. В данном плане система может быть доработана в плане добавления настройки для пользователя, что вычисления ведутся только при подключении к WI-FI-сети.

5) На данный момент мобильный клиент никак не следит за расходом аккумулятора. Мы предполагаем, что пользователь самостоятельно следит за уровнем заряда телефона и в случае чего может либо выйти из вычисления, либо поставить телефон на зарядку. В любом случае, система может быть доработана в этом плане, т. к. система Android позволяет ставить ограничения на фоновые работы в зависимости от уровня заряда батареи. Отметим, что вопросы энергопотребления программных реализаций различных алгоритмов в настоящее время достаточно активно исследуются [23–24].

Разумеется, построение суперкомпьютеров на основе мобильных устройств — это дискуссионный вопрос далекого будущего, но облачные и другие парадигмы вычислений на основе использования большого числа ускорителей в виде мобильных устройств — это хорошее решение сегодняшнего дня. В настоящей работе сделана попытка внести вклад в развитие данной области наук.

Заключение. Изложена архитектура распределенной вычислительной системы на основе множества мобильных устройств, объединенных в общую сетевую инфраструктуру. В основе распределенной вычислительной системы лежит проблемно-ориентированная модель выполнения задач. Ограничение «времени жизни» устройства накладывает требования не только на масштабируемость, но и на целостность вычислительной системы. Предложенная в работе архитектура позволяет учитывать реконфигурацию сети вычислительных устройств и строить гарантированную оценку масштабируемости распределенной гетерогенной вычислительной системы.

Список литературы

1. Armbrust M., Fox A., Griffith R., Joseph A.D, Katz R. H, Konwinski A., Lee G., Patterson D. A, Rabkin A., I. Stoica, et al. Above the clouds: A berkeley view of cloud computing // Technical Report. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley. 2009.
2. Buyya R., Yeo C. S., Venugopal S., Broberg J., Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility // Future Generation Computer Systems. 2009. 25(6). P. 599–616.
3. Foster I., Kesselman C., Tuecke S. Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of High Performance Computing Applications. 2001. 15(3). P. 200–222.
4. Foster I., Zhao Y., Raicu I., Lu S. Cloud Computing and Grid Computing 360-degree Compared // Grid Computing Environments Workshop (GCE'08). 2008. P. 1–10.
5. Peer-to-peer (P2P). [Электрон. Рес.]: <https://www.techtarget.com/searchnetworking/definition/peer-to-peer> (дата обращения 26.04.2022).

6. Одноранговые вычисления. [Электрон. Рес.]: <https://melimde.com/koncepciya-operacionnih-sistem.html?page=47> (дата обращения 26.04.2022).
7. Lavoie E., Hendren L. Personal Volunteer Computing // Proceedings of the 16th ACM International Conference on Computing Frontiers. 2019. P. 240–246.
8. Virtejanu I., Nitu C. Programming distributed applications for mobile platforms using MPI // U.P.B. Sci. Bull. 2013. 75(4).
9. Attia D. E., ElKorany A. M., Moussa A. S. High Performance Computing Over Parallel Mobile Systems // International Journal of Advanced Computer Science and Applications. 2016. 7(9). P. 99–103.
10. Büsching F., Schildt S., Wolf L. DroidCluster: Towards Smartphone Cluster Computing — The Streets are Paved with Potential Computer Clusters // 32nd International Conference on Distributed Computing Systems Workshops. 2012. P. 114–117.
11. Prem Kumar M., Bhat R. R., Alavandar S. R., Ananthanarayana V. S. Distributed Public Computing and Storage using Mobile Devices // 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER). 2018. P. 82–87.
12. Nizamov N. R., Shahova I. S. Mechanisms for using mobile devices in distributed computing // Russian Digital Libraries Journal. 2019. 22(4). P. 200–213.
13. Fadlallah G., Mcheick H., Rebaine D. Layered Architectural Model for Collaborative Computing in Peripheral Autonomous Networks of Mobile Devices // Procedia Computer Science. 2019. 155. P. 201–209.
14. Arslan M. Y. Computing While Charging: Building a Distributed Computing Infrastructure Using Smartphones // Proceedings of the 8th international conference on Emerging networking experiments and technologies. 2012. P. 193–204.
15. Pratt T. K., Seitelman L. H., Zampano R. R., Murphy C. E., Landis F. Optimization applications for aircraft engine design and manufacture // Advances in Engineering Software. 1993. V. 16, I. 2. P. 111–117.
16. Kabanikhin S. I., Kulikov I. M., Shishlenin M. A. An algorithm for recovering the characteristics of the initial state of supernova // Computational Mathematics & Mathematical Physics. 2020. V. 60, I. 6. P. 1008–1016.
17. Stone J., Tomida K., White C., Felker K. The Athena++ Adaptive Mesh Refinement Framework: Design and Magnetohydrodynamic Solvers // The Astrophysical Journal Supplement Series. 2020. V. 249. Article Number 4.
18. Kulikov I., Chernykh I., Tutukov A. A New Hydrodynamic Code with Explicit Vectorization Instructions Optimizations that Is Dedicated to the Numerical Simulation of Astrophysical Gas Flow. I. Numerical Method, Tests, and Model Problems // The Astrophysical Journal Supplement Series. 2019. V. 243. Article Number 4.
19. Kulikov I. M., Chernykh I. G., Tutukov A. V. Mathematical Modeling of a High-Speed Collision of White Dwarfs—the Explosion Mechanism of Type Ia/Iax Supernovae // Journal of Applied and Industrial Mathematics. 2022. V. 16. P. 80–88.
20. Kulikov I. M., Chernykh I. G., Ulyanichev I. S., Tutukov A. V. Mathematical Simulation of Nuclear Carbon Burning in White Dwarfs Using a 7-Isotope Reaction Network // Journal of Applied and Industrial Mathematics. 2022. V. 16. P. 440–448.
21. Kulikov I. GPUPEGAS: A New GPU-accelerated Hydrodynamic Code for Numerical Simulations of Interacting Galaxies // The Astrophysical Journal Supplements Series. 2014. V. 214. Article Number 12.
22. Kulikov I. M., Chernykh I. G., Snytnikov A. V., Glinskiy B. M., Tutukov A. V. AstroPhi: A code for complex simulation of dynamics of astrophysical objects using hybrid supercomputers // Computer Physics Communications. 2015. V. 186. P. 71–80.

23. Kulikov I., Chernykh I., Karavaev D., Sapetina A. The Energy Efficiency Research of Godunov Method on Intel Xeon Scalable Architecture // IEEE. 2021 Ivannikov Ispras Open Conference (ISPRAS). 2022. Article Number 21722440.

24. Kulikov I., Chernykh I., Karavaev D., Sapetina A., Lomakin S. The Efficiency of Hydrodynamic Code on Intel Xeon Scalable Architecture // IEEE. 2021 Ivannikov Memorial Workshop (IVMEM). 2022. Article Number 21704168.



Ульяничев Иван Сергеевич — E-mail: ulyanichevis@mail.ru. Тел.: +7(913)-381-01-22. Младший научный сотрудник лаборатории Суперкомпьютерного моделирования Института вычислительной математики и математической геофизики СО РАН.

Область научных интересов: распределенные вычисления, мобильные устройства.

Ulyanichev Ivan Sergeevich — E-mail: ulyanichevis@mail.ru. Phone: +7(913)-381-01-22. Junior Researcher of the Laboratory of Supercomputer Modeling of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. Research interests: distributed computing, mobile devices.

Винс Дмитрий Владимирович — E-mail: vins@sscc.ru. Тел.: +7(951)903 5097. Кандидат технических наук, старший научный сотрудник лаборатории Суперкомпьютерного моделирования Института вычислительной математики и

математической геофизики СО РАН по специальности 05.13.15 — Вычислительные машины, комплексы и компьютерные сети. Область научных интересов: исследование вычислительных и управляющих алгоритмов для суперкомпьютеров методами имитационного моделирования.



Wiens Dmitry Vladimirovich — E-mail: vins@sscc.ru. Tel: +7(951)903 5097. PhD in Technical Sciences, Senior Researcher of the Laboratory of Supercomputer Modeling of the Institute of Computational Mathematics and Mathematical

Geophysics SB RAS, specialty 05.13.15 — Computing Machines, Complexes and Computer Networks. Research interests: investigation of computational and control algorithms for supercomputers by simulation methods.

Дата поступления — 15.02.2024

Правила представления и подготовки рукописей для публикации в журнале „ПРОБЛЕМЫ ИНФОРМАТИКИ“

Общие требования.

Редакция принимает к рассмотрению статьи в электронном виде (*исходный файл в L^AT_EX* и файл PDF, с приложением оригиналов рисунков в формате тех программ, в которых они были сделаны, отдельными файлами).

Файлы, содержащие текст статьи, иллюстрации и дополнительные материалы, можно пересылать на электронный адрес редакции: problem-info@sscc.ru.

Принимаются файлы, архивированные архиваторами ZIP/7Z или RAR; применение самораспаковывающихся архивов не допускается.

При повторной отправке материалов, а также при внесении в исходный текст дополнений или исправлений необходимо сообщить об этом в редакцию в тексте электронного письма.

Направляя статью в редакцию журнала, автор (соавторы) на безвозмездной основе передает (ют) издателю на срок действия авторского права по действующему законодательству РФ исключительное право на использование статьи или отдельной ее части (в случае принятия редколлегией Журнала статьи к опубликованию) на территории всех государств, где авторские права в силу международных договоров Российской Федерации являются охраняемыми, в том числе следующие права: на воспроизведение, на распространение, на публичный показ, на доведение до всеобщего сведения, на перевод на иностранные языки и переработку (и исключительное право на использование переведенного и (или) переработанного произведения вышеуказанными способами), на предоставление всех вышеперечисленных прав другим лицам.

Журнал „Проблемы информатики“ является некоммерческим изданием. Плата с авторов за публикацию статей не взимается.

К статье должны быть приложены:

— **разрешение на публикацию** от экспертного совета организации, в которой выполнена работа (для авторов из России);

— **оригинал рецензии;**

портретные фотографии авторов разрешением не менее 300 dpi.

— Блоки информации и на русском, и на английском языках просьба присылать отдельными файлами:

— **Название** статьи;

— **Инициалы и фамилии** авторов;

— **Места работы** авторов: полное наименование организации, почтовый индекс, город, страна;

— **Код(ы) классификации УДК;**

— **Аннотации**, содержащие краткую постановку задачи и описание метода решения: на русском языке объемом не более 1000 знаков, на английском языке расширенную, объемом от 4000 до 8000 знаков, что соответствует требованиям ВАК и Scopus.

— **Ключевые слова;**

— **Списки используемой литературы** в соответствии с ГОСТ Р 7.0.5—2008 (в английской версии необходимо выполнить **транслитерацию** неанглоязычных элементов списка литературы в соответствии с ГОСТ Р 7.0.34-2014) — составляются по ходу упоминания источников в тексте;

— **Краткие биографии (БИО)** авторов с указанием ключевых научных достижений (включая ученую степень, ученое звание — при наличии; основные области научных интересов и формулировку основных результатов, место работы, занимаемую должность, контактные данные — почтовый адрес с индексом, адрес электронной почты, контактный телефон).

Подготовка статьи.

1. Материал статьи должен быть изложен в следующей последовательности:

1.1. название статьи на английском языке;

1.2. инициалы и фамилия автора(ов) на английском языке;

1.3. место работы автора(ов) (на английском языке): полное наименование организации, индекс, город, страна;

1.4. англоязычная аннотация;

1.5. ключевые слова на английском языке;

1.6. references+транслитерация неанглоязычных элементов списка литературы;

1.7. название статьи на русском языке;

1.8. инициалы и фамилии и авторов;

1.9. место работы авторов: полное наименование организации, почтовый индекс, город, страна;

1.10. индекс УДК;

1.11. аннотация на русском языке;

- 1.12. ключевые слова (не более 8);
- 1.13. текст статьи;
- 1.14. список литературы, оформленный в соответствии с требованиями ГОСТ;
- 1.15. краткие биографии авторов на английском и русском языках с указанием ключевых научных достижений (ученую степень, ученое звание — при наличии; место работы, занимаемую должность, контактные данные — почтовый адрес с индексом, адрес электронной почты, контактный телефон, основные области научных интересов и формулировка основных результатов).

2. Требования к формулам:

- Нумерация формул сквозная, выносные формулы центрируются, номер выровнен по правому краю.

3. Требования к рисункам:

— Файлы с рисунками присылаются отдельно в формате программ, в которых они были выполнены: в формате MS Excel (для графиков и диаграмм), eps, pdf, png, tiff, bmp или jpeg (с максимальным качеством).

- Рисунки с подрисуночными подписями завершаются в текст статьи.
- Тексты, являющиеся частью рисунка, выполняются шрифтом TimesNewRoman.
- Фотографии должны иметь разрешение не менее 300 dpi.

4. Дополнительные требования:

— В текст статьи необходимо включать ссылки на рисунки и таблицы, а также подрисуночные подписи и заголовки таблиц. Все буквенные обозначения, приведенные на рисунках, необходимо пояснить в основном тексте или в подрисуночных подписях.

- Сокращения слов не допускаются (кроме общепринятых).
- Векторные переменные обозначаются полужирным шрифтом без курсива.
- Таблицы не должны быть громоздкими. Значения физических величин в таблицах, на графиках и в тексте должны указываться в единицах измерения СИ.
- Графики, если их на рисунке несколько, а также отдельные детали на чертежах, узлы и линии на схемах следует обозначать цифрами, набранными курсивом.
- Нумеровать следует только те формулы и уравнения, на которые имеются ссылки в тексте, нумерация сквозная.

— Ссылки на источники в тексте заключаются в квадратные скобки.

— Иностранные источники приводятся на языке оригинала. Ссылки на неопубликованные работы не допускаются.

Все статьи, опубликованные в журнале «Проблемы информатики», доступны на сайте https://elibrary.ru/title_about.asp?id=30275 и на сайте журнала <http://problem-info.sssc.ru> спустя год после опубликования.

Пример оформления статей можно посмотреть на сайте журнала <http://problem-info.sssc.ru>.

