

34. Weiss R. et al. What Was Your Prompt? A Remote Keylogging Attack on AI Assistants, *arXiv pre-print arXiv:2403.09751*, 2024.
35. Turdakov D.Yu., Avetisyan A.I., Arkhipenko K.V. [i dr.]. Doverennyi Iskusstvennyi intellekt: vyzovy i perspektivnye resheniya [Trusted Artificial Intelligence: Challenges and Promising Solutions], *Doklady Rossiyskoy akademii nauk. Matematika, informatika, protsessy upravleniya* [Reports of the Russian Academy of Sciences. Mathematics, informatics, control processes], 2022, Vol. 508, No. 1, pp. 13-18. DOI: 10.31857/S2686954322070207. EDN CVIVCS.

Статью рекомендовал к опубликованию д.т.н. Г.Е. Веселов.

Мельников Сергей Юрьевич – Российский университет дружбы народов имени Патриса Лумумбы; e-mail: melnikov@linfotech.ru; г. Москва, Россия; кафедра теории вероятностей и кибербезопасности института компьютерных наук и телекоммуникаций факультета физико-математических и естественных наук; д.ф.-м.н.; доцент.

Мешеряков Роман Валерьевич – ИПУ РАН; e-mail: mrv@ieee.org; г. Москва, Россия; г.н.с.; д.т.н.; профессор.

Пересыпкин Владимир Анатольевич – Академия криптографии РФ; e-mail: info@cryptoacademy.gov.ru; г. Москва, Россия; действительный член; д.т.н.

Melnikov Sergey Yur'evich – Patrice Lumumba Peoples' Friendship University of Russia; e-mail: melnikov@linfotech.ru; Moscow, Russia; the Department of Probability Theory and Cybersecurity, Institute of Computer Science and Telecommunications, Faculty of Physics, Mathematics and Natural Sciences; dr. of phys. and math. sc.; associate professor.

Meshcheryakov Roman Valer'evich – IPU RAS; e-mail: mrv@ieee.org; Moscow, Russia; chief researcher; dr. of eng. sc.; professor.

Peresyppkin Vladimir Anatol'evich – Academy of Cryptography of the Russian Federation; e-mail: info@cryptoacademy.gov.ru; Moscow, Russia; Full Member; dr. of eng. sc.

УДК 004.382.2

DOI 10.18522/2311-3103-2024-5-68-78

Е.А. Титенко, Э.И. Ватутин, М.А. Титенко, Э.В. Мельник, А.П. Локтионов

АППАРАТНО-ОРИЕНТИРОВАННЫЙ МЕТОД УСКОРЕННОГО ПОИСКА ВХОЖДЕНИЙ ОБРАЗЦА НА ОСНОВЕ СТРУКТУРНО-ПРОЦЕДУРНЫХ ВЫЧИСЛЕНИЙ

Операция поиска вхождений образца в текст является общезначимой в современных вычислительных средствах при решении проблемно-поисковых задач. Наибольший интерес представляют аппаратно-программные решения, имеющие однородную структуру и регулярные связи между вычислительными блоками. Целью работы является сокращение временных затрат на поиск вхождений на основе применения параллельного поиска в ассоциативной памяти и метода распараллеливания по итерациям. Предлагаемый метод использует ассоциативную память для параллельного поиска вхождений и динамическую реконфигурации структуры исходной строки из одномерного вида в матричную форму. Вовлечение в реконфигурацию всех элементов влечет избыточные затраты внутренней блочной памяти на последовательный просмотр частичных вхождений по одному множеству стартовых позиций, кратных длине образца (второй символьный операнд. Вместо этого предложен метод совмещения во времени поиска частичных вхождений по двум наборам подстрок, кратных длине образца, с одновременным пропорциональным уменьшением элементов разрядного среза ассоциативной памяти по каждому набору, что позволяет на текущем шаге поиска обрабатывать несколько символов образца. Количественные оценки времени поиска определяются количеством операций сравнения и записи подстрок в общем цикле работы, а также пропорциями времени данных операций. Показано, что для образцов более 10 элементов временной выигрывает примерно в 1,8-2 раза. Данный эффект получен за счет исключения шагов последовательного сдвига с переходами между граничными элементами строк. Разработанный метод обеспечивает конвейерную обработку потока строковых операндов с совмещением просмотра на текущем шаге поиска неединичного множества символов обрабатываемой строки. Сокращение времени поиска обеспечивается введением конвейера, количество ступеней

которого зависит от коэффициента редукции размера разрядного среза, что позволяет аппаратно реализовать структурно-процедурный подход, применяемый в реконфигурируемых вычислительных системах.

Параллельный поиск; ассоциативная память; реконфигурация; метод итераций; конвейер.

E.A. Titenko, E.I. Vatutin, M.A. Titenko, E.V. Melnik, A.P. Loktionov

A HARDWARE-ORIENTED METHOD OF ACCELERATED SEARCH BY TEMPLATE BASED ON STRUCTURAL-PROCEDURAL COMPUTING

The operation of searching for occurrences of a pattern in a text is generally significant in modern computing tools for solving problem-searching tasks. Of greatest interest are hardware and software solutions that have a homogeneous structure and regular connections between computing blocks. The aim of the work is to reduce the time costs for searching for occurrences based on the use of parallel search in associative memory and the method of parallelization by iterations. The proposed method uses associative memory for parallel search for occurrences and dynamic reconfiguration of the structure of the original string from a one-dimensional form to a matrix form. The method is critical to such resources as the number of memory access channels, the volume of block memory for creating and parallel operation of an array of associative cells. Involvement of all elements in the reconfiguration entails excessive costs of the internal block memory for sequential viewing of partial entries by one set of starting positions multiples of the sample length (the second symbolic operand). Instead, an approach is proposed to combine in time the search for partial entries by two sets of substrings multiples of the sample length, with a simultaneous proportional reduction in the elements of the bit slice of the associative memory for each set, which allows processing several sample symbols at the current search step. Quantitative estimates of search time are determined by the number of comparison and substring writing operations in the overall work cycle, as well as the proportions of the time of these operations. It is shown that for samples of more than 10 elements, the time gain is approximately 1.8-2 times. This effect is obtained by eliminating the steps of sequential shift with transitions between the boundary elements of the strings. The developed method provides pipeline processing of a stream of string operands with a combination of viewing at the current search step of a non-unit set of characters of the processed string. The search time is reduced by introducing a pipeline, the number of stages of which depends on the reduction coefficient of the bit slice size, which allows hardware implementation of the structural-procedural approach used in reconfigurable computing systems.

Parallel search; associative memory; reconfiguration; iteration method; pipeline.

Введение. Научно-технический прогресс компьютерной индустрии направлен на повышение эффективности ее использования для различных масштабных задач управления, моделирования, поиска решений как на основе методов оптимизации, так и на основе методов вывода с применением баз знаний. Интенсификация вычислений в высокопроизводительных вычислительных системах (ВС) реализуется по нескольким направлениям: реконфигурируемые вычислительные системы, многопроцессорные кластерные системы, GRID-системы и др. Предметной областью являются вычислительно трудоемкие задачи проблемно-поискового и расчетного характера и аппаратно-программные средства их реализации или поддержки [1–6].

В первом случае интеллектуализация вычислений направлена на применение моделей и средств обработки знаний. Такие ВС призваны дополнить когнитивные способности человека в части систематизации информации, выделения новых знаний, поиску скрытых закономерностей и др. Они основаны на построении архитектур ВС, поддерживающих или использующих модели представления знаний и методы их обработки [7–10].

Второй путь интенсификации заключается в создании проблемно-ориентированных вычислительных средств, имеющих операционную часть, максимально приближенную к решаемой задаче. В настоящее время в России и за рубежом динамично развиваются реконфигурируемые вычислительные системы (РВС), построенные из программируемых логических интегральных схем (ПЛИС). Главным преимуществом РВС перед МВС с «жесткой» архитектурой является возможность адаптации РВС под структуру информационного графа задачи, что обеспечивает параллельно-конвейерную организацию вычислений без избыточных затрат на промежуточные пересылки и вычисления.

Использование ПЛИС в качестве элементной базы РВС имеет обоснованные перспективы. Объединение множества ПЛИС в единое вычислительное поле позволяет добиться практически линейного роста реальной производительности при масштабировании задач [11, 12]. Структурно-процедурный подход создания РВС на элементной базе ПЛИС заключается в непосредственном отображении топологии и структуры информационного графа задачи на вычислительную структуру (операционную часть) и в аппаратной поддержке базового подграфа в зависимости от имеющего ресурса ПЛИС (объем логических ячеек, размер блочной памяти, каналы доступа к памяти и др.).

Наиболее важной и трудоемкой операцией в слабо формализованных задачах является операция поиска по образцу (поиска вхождения образца в текст). Она лежит в основе работы абстрактных машин вывода, динамических экспертных систем, вычислительных машин, управляемых потоком данных, специализированных процессоров логического вывода, интеллектуальных аппаратных планировщиков и др. [13, 14].

Применение структурно-процедурного принципа вычислений для задачи поиска вхождений (поиска по образцу) позволяет получить ее базовый подграф и провести его преобразование к параллельно-конвейерной форме и аппаратно реализовать в ПЛИС, что делает общезначимыми метод и реализующее его устройство для высокоскоростных ВС [2, 10].

Постановка задачи. Известен огромный арсенал последовательных реализаций метода поиска по образцу, осуществляющих полную или частичную посимвольную обработку символьных операндов [15–17]. Тем не менее, одномерный формат операндов предполагает организацию последовательных процессов поиска, реализуя перемещения (переходы, прыжки, пропуски) через незначащих позиции в обрабатываемой строке. Такой подход имеет ряд недостатков при аппаратной реализации:

- ◆ зависимость времени поиска от длины обрабатываемой строки;
 - ◆ невозможность досрочно прекратить поиск на текущем шаге;
 - ◆ игнорирование возможности конвейеризации поиска.
- Пусть в общем алфавите $A = \{a_1, a_2, \dots, a_d\}$ заданы конструктивные объекты:
- ◆ строка-образец O длиной n символов;
 - ◆ исходный текст (обрабатываемая строка) S длиной m символов ($n \leq m$).

Требуется найти позиции всех вхождений O в S , т.е. разработать метод и алгоритм, находящие такие i , что

$$\forall i(O(l, n) = S(i, i + n - 1)) \mid 1 \leq i \leq k, k = m - n + 1, \quad (1)$$

где $n > 0$, $m > 0$ и $n \leq m$.

Ведущими специалистами в современной информатике (Смит, Д. Гасфильд, Р. Бойер, Дж. Мур, и др. [15–18]) признано, что оценка времени работы поискового алгоритма осуществляется в количестве операций сравнения двух строк: строки-образца O и текста S или его текущей подстроки. Другими словами время поиска – это функция $T(n, m)$.

Методы поиска по образцу по применению аппаратных средств можно разбить на четыре класса [1–4]:

- ◆ алгоритмы поиска на микропроцессорах CISC-архитектурой;
- ◆ алгоритмы поиска на микропроцессорах RISC-архитектурой;
- ◆ алгоритмы поиска на графических микропроцессорах;
- ◆ алгоритмы поиска на элементной базе ПЛИС.

Первый класс алгоритмов поиска составляют алгоритмы, для которых повышение эффективности данных решений достигается за счет многоядерности и многопоточности для распараллеливания потока команд и управления переходами. Дополнительная информация о структурных отношениях закладывается в таблицы переходов на этапе подготовки данных.

Второй класс алгоритмов поиска составляют алгоритмы, для которых уменьшение количества сравнений в данных алгоритмах достигается за счет аппаратной поддержки набора строковых команд и форматов, позволяющих непосредственно вести обработку фрагментов подстрок с учетом структурных зависимостей. Построение и поддержка префиксов, суффиксов для строковых операндов позволяет сократить затраты времени на повторные шаги.

Третий класс построен на основе принципов адаптивного поиска и заключается в решении задачи приближенного сравнения двух строк с использованием методов обучения (систематизациями). Тем не менее, данный подход требует создания представительных обучающих и контрольных строковых выборок, имеющих значимые структурно-лингвистические характеристики, что позволяет точно установить отсутствие вхождения строки-образца в исходный текст за счет вычисления расстояния между эталоном и поисковым объектом. В случае ненулевого расстояния между ними данные алгоритмы не позволяют установить позицию вхождения образца, что ограничивает область эффективного применения адаптивного поиска.

Четвертый класс алгоритмов поиска (Shift-And, Сандер-Бу, Манбера, потоковый алгоритм поиска и др.) позволяет решать задачу поиска вхождений, непосредственно программируя информационный граф в вычислительную структуру на основе типовых ресурсов ПЛИС [6, 20].

В целом, анализ известных аппаратных решений поиска вхождений показывает, что все они имеют условия для распараллеливания вычислений, но этот параллелизм является неявным.

Вместе с тем, если размеры строковых операндов превышают предельные пороги ($n > 10^2$ и $m > 10^6$), то время работы алгоритмов становится практически неприемлемым [4] из-за лавинообразного роста количества промежуточных вариантов частичных вхождений строковых операндов своим фрагментами. Это означает, необходимы методы и алгоритмы поиска вхождений, не имеющие прямо пропорциональной временной зависимости от размеров операндов и ориентированные на передовые принципы интеллектуальной обработки информации («однородность-параллелизм», «ассоциация-контекст», «поток данных-конвейеризация», «реконфигурация-переменная структура»).

Метод ускоренного ассоциативного поиска. Ассоциативная память и основанные на ней ассоциативные процессоры параллельных вычислений [9, 21, 22] являются одним из значимых инструментов повышения эффективности вычислительных систем.

Как известно, ассоциативная память в качестве идентификатора ячеек использует не адрес, а контекст (контекстный атрибут), связанный (ассоциируемый) с данными, хранящимися в ячейках памяти [9]. В силу наличия контекстной связи между идентификатором и данными разные ячейки памяти могут иметь одинаковый поисковый атрибут-контекст. Это обстоятельство стало основой поиска не ячейкам памяти, а по разрядным срезам.

Информационный граф задачи ассоциативного поиска показан на рис. 1, где Y_{ij} – элементарная ячейка сравнения j -го бита контекстного атрибута A_j и ij -го бита накопителя памяти, $M1_j$ – j -ый бит маски разрядных срезов, $M2_i$ – i -ый бит маски ячеек накопителя памяти, $RgOtv$ – регистр ответов, APB – арбитр, $I_{вх}$, $I_{вых}$ – входные и выходные шины данных.

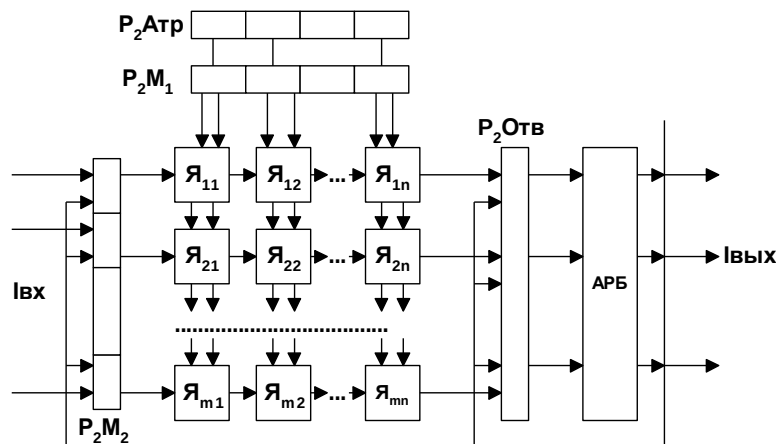


Рис. 1. Модель ассоциативной памяти

Основу известного метода ассоциативного поиска вхождений [23] составляет динамическая (циклическая) реконфигурация структуры исходной строки S : одномерный сдвиговый регистр $\leftrightarrow$ двумерная матрица. На четных шагах реконфигурации структуры S выполняется левый сдвиг на один символ по всей длине строки S , чем достигается совмещение во времени и пространстве всех подстрок из S длиной в n символов. На нечетных шагах реконфигурации S выполняется ассоциативный поиск (поиск по разрядным срезам) строки-образца по выделенным фрагментам из S . Двумерный вид S необходим для организации параллельного поиска всех вхождений по разрядным срезам ассоциативной запоминающей матрицы. Циклы поиска и реконфигурации соответственно выполняются $n-1$ раз, что обеспечивает линейную временную сложность решения задачи $T = O(n)$.

Процесс ассоциативного поиска всех вхождений реализуется путем представления исходной строки S длиной до $m = n \times p$ символов в виде двумерной матрицы из p строк по n разрядов в каждой строке, где n определяется разрядностью образца O . При таком представлении строки S образец O длиной в n символов параллельно сравнивается по p строкам матрицы в полном соответствии с известными способами ассоциативного поиска [22, 23].

Пусть для $A = \{0,1\}$ заданы $S = 1100101011001001$ и $O = 1001$ с длинами 16 элементов и 4 элемента соответственно. Процессы ассоциативного поиска вхождений по шагам отражены на рис. 2 (поиск, сдвиг, поиск). В случае положительного сравнения результат содержит логические «1» в строках ассоциативной матрицы с совпавшими фрагментами из S .

В случае отрицательного сравнения фиксируется нулевой результат, выполняется переход вперед на 1 символ (сдвиг влево на 1 символ строки S). В итоге выполняется переход к новой матрице с новым составом p подстрок. Цикл сравнения повторяется.

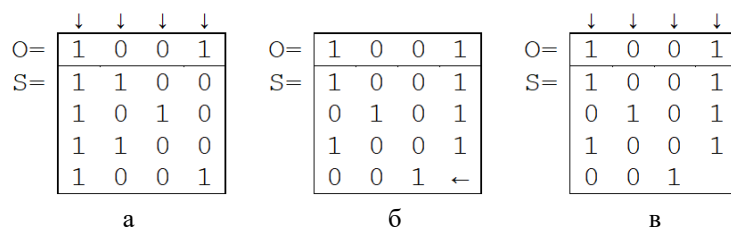


Рис. 2. Шаги ассоциативного поиска с реконфигурацией S

Информационный граф задачи тесно связан с моделью работы ассоциативной памяти (рис. 1) и поясняется на рис. 3, где показана параллельная обработка элементов каждого разрядного среза накопителя ассоциативной памяти и последовательная обработка разрядных срезов между собой. Основные вычислительные элементы в базовом подграфе α – двухвходовые компараторы на совпадение элемента o_j строки-образца O и элемента s_{ij} обрабатываемой строки (текста) S ($j=1 \dots n$), $i=1 \dots p$).

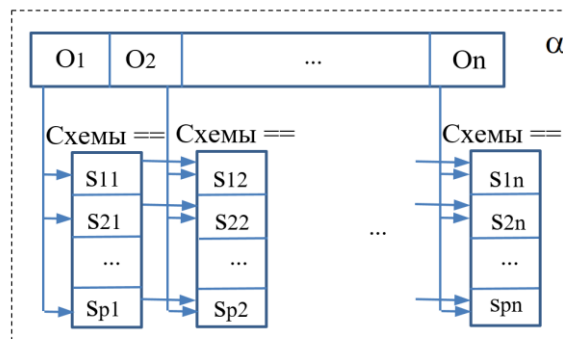


Рис. 3. Информационный граф задачи ассоциативного поиска вхождений

Двумерное представление обрабатываемой строки S в виде матрицы, позволяет вести параллельную обработку p строк в составе матрицы, принимая ширину матрицы равной длине образца, т.е. n элементов. Тогда переход к новой конфигурации поиска реализуется преобразованием к одномерному виду и сдвигу влево, чтобы получить в матрице новые p подстрок. Тем не менее, такой подход на текущем шаге не позволяет «смотреть вперед» и использовать дополнительную информацию для прекращения поиска по неперспективным строкам матрицы.

Недостаток разработанного метода ассоциативного поиска вхождений заключается в последовательной обработке разрядных срезов и невозможности организовать конвейерные вычисления по n разрядным срезам. Кроме того, переменный размер обрабатываемой строки S подразумевает использование неограниченного ресурса блочной памяти ПЛИС для поддержания матрицы $m=n \times p$ символов, при этом p – количество строк матрицы определяется размером задачи, что в общем случае приводит к последовательным решениям.

В методах структурно-процедурных вычислений, реализуемых в РВС, активно используется метод распараллеливания по итерациям [23]. Суть данного метода, заключается в том, что повторяющаяся последовательность операций, соответствующая базовому подграфу в вычислительной структуре, реализуется как конвейер из функциональных блоков с последовательной передачей данных между блоками и выполнении операций над разными частями матрицы. В результате чего при реализации в вычислительной структуре n однотипных блоков, соответствующих базовому подграфу, на выходе будет получен результат на n -й итерации. Итерационная зависимость обрабатываемых матриц позволяет её конвейеризовать для эффективной потоковой обработки. При этом базовый подграф формируется таким образом, чтобы сократить число обращений к распределенной памяти. Это прием позволяет сократить количество каналов доступа к памяти распараллелить обработку данных на текущей итерации.

Предлагаемый метод ускоренного ассоциативного поиска основан на следующих принципах:

- ◆ редукция количества элементов в разрядном срезе;
- ◆ совмещение в составе разрядного среза части элементов от текущего и следующего шагов поиска;
- ◆ замена последовательного левого сдвига строк матрицы на построчную перезапись;
- ◆ конвейеризация базового подграфа.

Сущность метода заключается в следующем. Однородная структура ассоциативной памяти позволяет объединить в пределах разрядного среза данные с i -ой и $i+1$ -й, $i+2$ -й, ... позиций. Это обеспечивает на текущем шаге возможность анализа следующих частичных вхождений образца и использование этой информации для принятия решения о продолжении поиска. При объединении в текущем разрядном срезе ассоциативной памяти частей столбцов V матричного представления S возникает эффект продвижения вперед и учета большего количества частичных вхождений без реконфигурации матрицы. Продвижение вперед по подстрокам S согласуется с коэффициентом редукции r вектора V .

Далее рассматривается метод ускоренного ассоциативного поиска с объединением подстрок из S по 2 символа, т.е. реализуется «просмотр вперед» на 1 символ. Исходя из минимизации вычислительного ресурса принимается коэффициент редукции $r=2$.

Исходная строка S заранее разбивается на 2 части и представляется в виде

$$S = \{S\}^{l/2} \cup \{S\}^{l+1/2}, \quad (2)$$

где $\{S\}^{l/2}$ – половина строки с текущей позиции i ,
 $\{S\}^{l+1/2}$ – половина строки, смещенная на 1 символ вперед.

Ниже показана разметка строки S для выделения и обработки двух смещенных подстрок $\{S\}^{l/2}$ и $\{S\}^{l+1/2}$.

$$\{S\}^{1/2} = \underbrace{11001}_{11} \underbrace{0101}_{12} \underbrace{1000}_{13} \underbrace{101}_{14} \quad (3)$$

$$\{S\}^{1+/2} = \underbrace{1100}_{21} \underbrace{1010}_{22} \underbrace{1000}_{23} \underbrace{101}_{24}.$$

Шаг метода ускоренного ассоциативного поиска иллюстрируется на рис. 4.

O=	<table><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>				1	0	0	1				
1	0	0	1									
S=												
{S}1/2	<table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	1	1	0	0	1	0	1	0	0	0	V1i
1	1	0	0									
1	0	1	0									
{S}1+/2	<table><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	1	0	0	1	0	1	0	1	1	0	V2i
1	0	0	1									
0	1	0	1									

Рис. 4. Ускоренный ассоциативный поиск с просмотром вперед на 1 символ

При такой организации в каждом разрядном срезе верхняя половина элементов записана от текущей позиции поиска по S, нижняя половина – записана от следующей позиции поиска по строке S. В результате ассоциативный поиск выполняется по 2 матрицам одновременно, что позволяет принимать обоснованное решение о следующем шаге поиска (при отрицательных результатах по векторам ответов V1 и V2).

Далее цикл поиска с просмотром «вперед» далее выполняется над вторыми половинами матрицы S, также одна из которых будет смещена на 1 символ вперед.

В соответствии со структурно-процедурными принципами вычислений выполняется распараллеливание по итерациям. В результате получается двухконвейерная вычислительная структура (рис. 5), совмещающая во времени потоковую обработку подготовленных строковых операндов: строки-образца P и подстроки {S} матрицы.

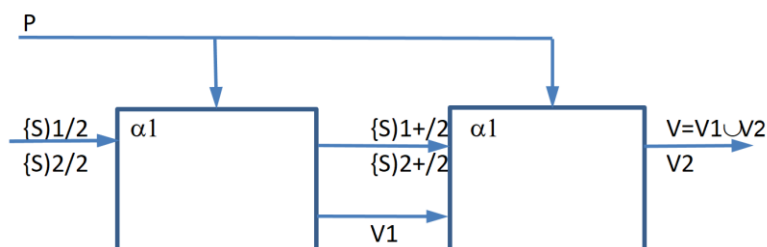


Рис. 5. Двухступенчатый конвейер поиска вхождений

Результаты и обсуждение. Аппаратная реализация поисковых операций в ассоциативной памяти (поиск на совпадение, несовпадение и приблизительный поиск) и организация конвейерной обработки матриц позволяют получить временной выигрыш при использовании метода ускоренного ассоциативного поиска. Данный аппаратно-ориентированный метод отличают:

- ◆ увеличенный на коэффициент редукции r объем вычислительного ресурса для создания ступеней конвейера;
- ◆ сохраненное число каналов доступа к памяти;
- ◆ возможность параллельной перезаписи по строкам нижней половины матрицы в ее верхнюю часть без обращения к внешней памяти;
- ◆ сокращенный размер подгружаемой подстроки.

Известный метод ассоциативного поиска с динамической реконфигурацией обрабатываемой строки S имеет оценку временной сложности

$$T(n)=(n-1)t_{COMP}+(n-1)t_{SHIFT}. \quad (4)$$

Разработанный метод использует динамическую реконфигурацию части обрабатываемой строки только для первого и последнего символов образца, заменяя последовательные сдвиги на загрузку фрагмента строки в ячейки ассоциативной памяти. В результате оценка временной сложности по методу ускоренного поиска составляет

$$T(n)=n/2t_{COMP}+(n/2)t_{SAVE}+2t_{SHIFT}. \quad (5)$$

Анализ времени базовых операций при ассоциативном поиске (компарирование, сдвиг, загрузка строки) показывает, что они имеют следующие соотношения: $t_{COMP}=\tau$, $t_{SHIFT}=0,6\tau$, $t_{SAVE}=0,4\tau$.

Тогда временные затраты на параллельный поиск вхождений образца на основе ассоциативной памяти по (4) и (5) имеют вид соответственно

$$T(n)=(n-1) \cdot 1,8\tau + C_1, \quad (6)$$

$$T(n)=0,7n\tau + 1,6\tau + C_2. \quad (7)$$

На рис. 6 показаны временные зависимости разработанного метода ускоренного поиска (Метод 2) и известного метода ассоциативного поиска (Метод 1) для переменной длины образца $n=4, 8, 12, 16, 20, 24, 28, 32$ элементов, приняв следующие временные задержки (в условных единицах времени) $\tau=15$, $C_1=24$, $C_2=36$.

Анализ графиков (рис. 6) показывает, что разработанный метод имеет эффективную область применения при $n \geq 10$ элементов. Малые размеры длины образца $3 < n < 8$ делают предпочтительным поиск вхождений образца на основе динамической реконфигурации «строка $\leftrightarrow$ матрица».

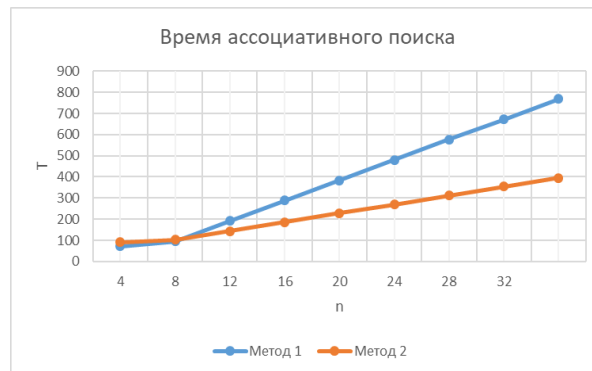


Рис. 6. Временные зависимости поиска вхождений

Распараллеливание по итерациям для метода ускоренного поиска при $n < 8$ по-прежнему предполагает проведение 2 шагов динамической реконфигурации для первого и последнего символов строки S , что негативно сказывается на времени поиска. При этом увеличенные вычислительные ресурсы на организацию двухступенчатого конвейера не компенсируются временным выигрышем в силу малого числа итераций.

Выводы. На основе метода ассоциативного поиска разработан метод ускоренного поиска вхождений образца, отличающийся применением принципов структурно-процедурных вычислений. Отображение информационного графа ассоциативного поиска в матричную вычислительную структуру, состоящую из массива параллельно работающих символьных компараторов, позволяет совместить поиск по смежным символам образца путем создания конвейера. Применение принципа распараллеливания по итерациям позволяет сохранить неизменным критичный вычислительный ресурс при обработке матриц – число каналов доступа к памяти. Получен временной выигрыш, составивший примерно 2 раза по сравнению с ассоциативным поиском вхождений. Разработанный метод эффективно применим для образцов с длинами от 10 элементов, что преимущественно встречается в машинах баз знаний, динамических экспертных системах, вычислительных средствах поддержки трейдинговых процессов, анализа Big Data и др. [24–26].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Воеводин В.В. Математические модели и методы в параллельных процессах. – М.: Наука, 1986. – 296 с.
2. Гузик В.Ф., Каляев И.А., Левин И.И. Реконфигурируемые вычислительные системы: учеб. пособие. – Таганрог: Изд-во ЮФУ, 2016. – 472 с.
3. Корнеев В.В. Вычислительные системы. – М.: Гелиос АРВ, 2004. – 510 с.
4. Бурцев В.С. Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ: Сб. статей / сост. В.П. Торчигин, Ю.Н. Никольская, Ю.В. Никитин. – М.: ТОРУС ПРЕСС, 2006. – 416 с.
5. Левин И.И., Федоров А.М., Доронченко Ю.И., Раскладкин М.К. Перспективные высокопроизводительные реконфигурируемые вычислители с иммерсионным охлаждением // Известия ЮФУ. Технические науки. – 2020. – № 7 (217). – С. 6-19. – DOI: 10.18522/2311-3103-2020-7-6-19.
6. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Изд-во "Янус-К", 2003. – 380 с.
7. Lothaire M. Applied Combinatorics on Words. In: Encyclopedia of Mathematics and its Applications. – Cambridge: Cambridge University Press, 2005.
8. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Издательский дом «Вильямс». 2003. – 864 с.
9. Огнев И.В., Борисов В.В., Сутула Н.А. Ассоциативные память, среды, системы. – М.: Горячая линия – Телеком, 2016. – 420 с.
10. Адамов А.А., Эйсымонт Л.К. Варианты архитектурных решений ЭКБ для систем искусственного интеллекта // Проектирование будущего. Проблемы цифровой реальности: Тр. 3-й Международной конференции. – М.: ИПМ им. М.В. Келдыша, 2020. – С. 112-131.
11. Левин И.И., Пелипец А.В. Методология распараллеливания по итерациям при решении задач линейной алгебры на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. – 2016. – № 7 (145). – С. 34-40.
12. Левин И.И., Подопригора А.В. Модифицированный метод обработки больших разреженных неструктурированных матриц на реконфигурируемых вычислительных системах // Вычислительные методы и программирование. – 2024. – Т. 25, № 2. – С. 142-154.
13. Добрица В.П., Титенко Е.А., Халин Ю.А., Киселев А.В. Системы искусственного интеллекта. – Курск: ЗАО «Университетская книга», 2023. – 143 с.
14. Эйсымонт Л.К., Моляков А.С., Заборовский В.С., Федоров С.А. Символьная обработка: эпизоды отечественной истории и перспективы // Матер. 2-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии (СКТ-2012)», с. Дивноморское, 2012. – С. 202-206.
15. Макконелл Дж. Основы современных алгоритмов. – 2-е изд., доп. – М.: Техносфера, 2004. – 368 с.
16. Окулов С.М. Алгоритмы обработки строк. – М.: БИНОМ. Лаборатория знаний, 2009. – 255 с.
17. Рыбина Г.В. Основы построения интеллектуальных систем. – М.: Финансы и статистика, 2010. – 430 с.
18. Hume A., Sunday D. Fast string searching // Software Practice and Experience. – November 1991. – Vol. 21, No. 11. – P. 1221-48.
19. Попов Э.В. Статические и динамические экспертные системы. – М.: Финансы и статистика, 1996. – 211 с.
20. Титенко Е.А., Титов В.С., Коновальчик А.П. Высокопроизводительные вычислительные системы на основе ПЛИС // Известия Юго-Западного государственного университета. – 2012. – № 4-2 (43). – С. 73.
21. Емельянов С.Г., Титенко Е.А., Зерин И.С. Однородные вычислительные структуры для параллельных символьных вычислений // Известия Юго-Западного государственного университета. – 2011. – № 6-2 (39). – С. 77а-82.
22. Типикин А.П., Титенко Е.А. Модификация цикла работы машины вывода для параллельных вычислительных устройств // Известия Юго-Западного государственного университета. – 2011. – № 6-2 (39). – С. 92-96.
23. Зерин И.С., Атакищев О.И., Титенко Е.А. [и др.]. Метод, алгоритм и техническое решение параллельного поиска и подстановки на ассоциативной памяти // В мире научных открытий. – 2012. – № 1-1 (25).
24. Левин И.И., Подопригора А.В. Метод распараллеливания по базовым макрооперациям для обработки больших разреженных неструктурированных матриц на РВС // Известия ЮФУ. Технические науки. – 2022. – № 6 (230). – С. 72-83.

25. Добрица В.П., Титенко Е.А., Халин Ю.А., Катыхин А.И. Модели представления и обработки знаний в информационно-аналитических системах. – Курск: ЗАО «Университетская книга», 2023. – 172 с.
26. Овчинкин О.В., Титова Г.С., Халин Ю.А. [и др.]. Исследование систем управления: учеб. пособие. – Курск: ЗАО Университетская книга, 2018. – 172 с.

REFERENCES

1. Voevodin V.V. Matematicheskie modeli i metody v parallel'nykh protsessakh [Mathematical models and methods in parallel processes]. Moscow: Nauka, 1986, 296 p.
2. Guzik V.F., Kalyaeva I.A., Levin I.I. Rekonfiguriruyemye vychislitel'nye sistemy: ucheb. posobie [Reconfigurable computing systems: textbook]. Taganrog: Izd-vo YuFU, 2016, 472 p.
3. Korneev V.V. Vychislitel'nye sistemy [Computing systems]. Moscow: Gelios ARV, 2004, 510 p.
4. Burtsev V.S. Parallelizm vychislitel'nykh protsessov i razvitiye arkhitektury superEVM: Sb. statey [Parallelism of computing processes and development of supercomputer architectures: Collection of articles], compilers V.P. Torchigin, Yu.N. Nikol'skaya, Yu.V. Nikitin. Moscow: TORUS PRESS, 2006, 416 p.
5. Levin I.I., Fedorov A.M., Doronchenko Yu.I., Raskladkin M.K. Perspektivnye vysokoproizvoditel'nye rekonfiguriruyemye vychisliteli s immersionnym okhlazhdeniem [Promising high-performance reconfigurable computers with immersion cooling], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 7 (217), pp. 6-19. DOI: 10.18522/2311-3103-2020-7-6-19.
6. Kalyaev A.V., Levin I.I. Modul'no-narashchiyaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiey vychisleniy [Modular-scalable multiprocessor systems with structural-procedural organization of computations]. Moscow: Izd-vo "Yanus-K", 2003, 380 p.
7. Lothaire M. Applied Combinatorics on Words. In: Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press, 2005.
8. Lyuger Dzh.F. Iskusstvennyy intellekt: strategii i metody resheniya slozhnykh problem [Artificial intelligence: strategies and techniques for solving complex problems]. Moscow: Izdatel'skiy dom «Vil'yams», 2003, 864 p.
9. Ognev I.V., Borisov V.V., Sutula N.A. Assotsiativnye pamyat', sredy, sistemy [Associative memory, environments, systems]. Moscow: Goryachaya liniya – Telekom, 2016, 420 p.
10. Adamov A.A., Eysymont L.K. Varianty arkhitekturnykh resheniy EKB dlya sistem iskusstvennogo intellekta [Variants of architectural solutions for electronic components for artificial intelligence systems], *Proektirovanie budushchego. Problemy tsifrovoy real'nosti: Tr. 3-y Mezhdunarodnoy konferentsii* [Designing the future. Problems of digital reality: proceedings of the 3rd International Conference.], Moscow: IPM im. M.V. Keldysha, 2020, pp. 112-131.
11. Levin I.I., Pelipets A.V. Metodologiya rasparallelivaniya po iteratsiyam pri reshenii zadach lineynoy algebrы na rekonfiguriruyemykh vychislitel'nykh sistemakh [Methodology of parallelization by iterations in solving problems of linear algebra on reconfigurable computing systems], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2016, No. 7 (145), pp. 34-40.
12. Levin I.I., Podoprigora A.V. Modifitsirovannyy metod obrabotki bol'shikh razrezhennykh nestrukturirovannykh matrits na rekonfiguriruyemykh vychislitel'nykh sistemakh [Modified method for processing large sparse unstructured matrices on reconfigurable computing systems], *Vychislitel'nye metody i programmirovaniye* [Computational methods and programming], 2024, Vol. 25, No. 2, pp. 142-154.
13. Dobritsa V.P., Titenko E.A., Khalin Yu.A., Kiselev A.V. Sistemy iskusstvennogo intellekta [Artificial intelligence systems]. Kursk: ZAO «Universitetskaya kniga», 2023, 143 p.
14. Eysymont L.K., Molyakov A.S., Zaborovskiy V.S., Fedorov S.A. Simvol'naya obrabotka: epizody otechestvennoy istorii i perspektivy [Symbolic processing: episodes of Russian history and prospects], *Mater. 2-y Vserossiyskoy nauchno-tekhnicheskoy konferentsii «Superkomp'yuternye tekhnologii (CKT-2012)», s. Divnomorskoe, 2012* [Materials of the 2nd All-Russian Scientific and Technical Conference "Supercomputer Technologies (SKT-2012)", With. Divnomorskoe, 2012], pp. 202-206.
15. Makkonell Dzh. Osnovy sovremennykh algoritmov [Fundamentals of modern algorithms]. 2nd ed. Moscow: Tekhnosfera, 2004, 368 p.
16. Okulov S.M. Algoritmy obrabotki strok [String processing algorithms]. Moscow: BINOM. Laboratoriya znaniy, 2009, 255 p.
17. Rybina G.V. Osnovy postroeniya intellektual'nykh system [Fundamentals of building intelligent systems]. Moscow: Finansy i statistika, 2010, 430 p.
18. Hume A., Sunday D. Fast string searching, *Software Practice and Experience*, November 1991, Vol. 21, No. 11, pp. 1221-48.

19. *Popov E.V.* Staticheskie i dinamicheskie ekspertnye sistemy [Static and dynamic expert systems]. Moscow: Finansy i statistika, 1996, 211 p.
20. *Titenko E.A., Titov V.S., Konoval'chik A.P.* Vysokoproizvoditel'nye vychislitel'nye sistemy na osnove PLIS [High-performance computing systems based on FPGA], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [Bulletin of the South-West State University], 2012, No. 4-2 (43), pp. 73.
21. *Emel'yanov S.G., Titenko E.A., Zerin I.S.* Odnorodnye vychislitel'nye struktury dlya parallel'nykh simvol'nykh vychisleniy [Homogeneous computing structures for parallel symbolic computations], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [Bulletin of the South-West State University], 2011, No. 6-2 (39), pp. 77a-82.
22. *Tipikin A.P., Titenko E.A.* Modifikatsiya tsikla raboty mashiny vyvoda dlya parallel'nykh vychislitel'nykh ustroystv [Modification of the output machine operation cycle for parallel computing devices], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [Bulletin of the South-West State University], 2011, No. 6-2 (39), pp. 92-96.
23. *Zerin I.S., Atakishchev O.I., Titenko E.A. [i dr.].* Metod, algoritm i tekhnicheskoe reshenie parallel'nogo poiska i podstanovki na assotsiativnoy pamyati [Method, algorithm and technical solution for parallel search and substitution on associative memory], *V mire nauchnykh otkrytiy* [In the world of scientific discoveries], 2012, No. 1-1 (25).
24. *Levin I.I., Podoprighora A.V.* Metod rasparrallelivaniya po bazovym makrooperatsiyam dlya obrabotki bol'shikh razrezhenykh nestrukturirovannykh matrits na RVS [Method of parallelization by basic macro-operations for processing large sparse unstructured matrices on RCS], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2022, No. 6 (230), pp. 72-83.
25. *Dobritsa V.P., Titenko E.A., Khalin Yu.A., Katykhin A.I.* Modeli predstavleniya i obrabotki znaniy v informatsionno-analiticheskikh sistemakh [Models of representation and processing of knowledge in information and analytical systems]. Kursk: ZAO «Universitetskaya kniga», 2023, 172 p.
26. *Ovchinkin O.V., Titova G.S., Khalin Yu.A. [i dr.].* Issledovanie sistem upravleniya: ucheb. posobie [Research of control systems: textbook]. Kursk: ZAO Universitetskaya kniga, 2018, 172 p.

Статью рекомендовал к опубликованию д.т.н., профессор А.В. Боженюк.

Титенко Евгений Анатольевич – Юго-Западный государственный университет; e-mail: johntit@mail.ru; г. Курск, Россия; тел.: +79051588904; кафедра программной инженерии; к.т.н.; доцент.

Ватутин Эдуард Игоревич – e-mail: evatutin@mail.ru; д.т.н.; доцент; профессор кафедры вычислительной техники.

Титенко Михаил Андреевич – e-mail: mikhail-titenko@mail.ru; аспирант.

Локтионов Аскольд Петрович – e-mail: loapa@mail.ru; д.т.н.; доцент; главный научный сотрудник центра по координации академической и вузовской науки, научной и образовательной деятельности.

Мельник Эдуард Всеволодович – Южный федеральный университет; e-mail: evm17@mail.ru; г. Таганрог, Россия; г.н.с.; зав. лабораторией интеллектуальных технологий.

Titenko Evgeny Anatolievich – South-West State University; e-mail: johntit@mail.ru; Kursk, Russia; phone: +79051588904; the Department of Software Engineering; cand. of eng. sc.; associate professor.

Vatutin Eduard Igorevich – e-mail: evatutin@mail.ru; dr. of eng.; sc.; associate professor; professor of the Computer Science Department.

Titenko Mikhail Andreevich – e-mail: mikhail-titenko@mail.ru; postgraduate student.

Loktionov Askold Petrovich – e-mail: loapa@mail.ru; dr. of eng. sc.; associate professor; chief researcher of the Center for Coordination of Academic and University Science, Scientific and Educational Activities.

Melnik Eduard Vsevolodovich – Southern Federal University; e-mail: evm17@mail.ru; Taganrog, Russia; chief researcher; head of the Laboratory of Intelligent Technologies.