E. A. Burkov, E. A. Tolkacheva, P. I. Paderno
Saint Petersburg Electrotechnical University

USE OF DECISION RULES TO RESOLVE CONTRADICTIONS BETWEEN OBJECT CLASSIFICATION PROCEDURES

Continues the consideration of the stochastic model of the object classification procedure, which makes it possible to carry out a comparative analysis of the object classification procedures, predict the effectiveness of their application, and design complex procedures based on simple ones. A correct approach to aggregating stochastic models of individual procedures and obtaining a stochastic model of a complex classification procedure is proposed. The issue of resolving contradictions between individual classification procedures using the mechanism of decision rules is considered and possible variants of such rules are given. The use of decision rules allows the design and analysis of complex classification procedures, which are sets of simpler procedures. A comparative analysis with other approaches to modeling the classification of objects is carried out. Possible directions for the formulation of new problems and the formation of other approaches for the development of a stochastic model of classification procedures are formulated.

Probability matrix, object classification, authentication coefficient, decision rule, stochastic model

УДК 004.272, 004.415.2, 004.94

А. Р. Лисс
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Г. Ю. Пуеров, Е. И. Сергеева
АО «Концерн «Океанприбор» (Санкт-Петербург)

Эффективные параллельные алгоритмы восстановления данных по известным значениям в узлах равномерной сетки

Рассмотрены вычислительные аспекты решения в реальном времени задачи восстановления данных по известным значениям в узлах. Описан один из известных подходов к решению поставленной задачи, обладающий низкой вычислительной сложностью и хорошим потенциалом в смысле параллелизма по данным. Рассматриваются параллельные алгоритмы восстановления данных по известным значениям в узлах равномерной сетки. Такие алгоритмы могут как иметь самостоятельное значение, так и быть составной частью решения многих вопросов прикладного характера. Исследуются способы распараллеливания по данным и организации вычислений, при этом используется подход с применением абстрактных моделей параллельных вычислений. Построена модель блочно-синхронно-конвейерного параллелизма (БСКП) для обработки потока данных в реальном времени, в основу которой положена широко известная модель параллельных вычислений общего назначения BSP (Bulk Synchronous Parallel). Параллельные алгоритмы восстановления данных сформулированы в терминах модели БСКП. Оценена вычислительная и коммуникационная сложность предложенных алгоритмов. Выведены зависимости между размерностями задачи и параметрами модели БСКП. Описана программная реализация предложенных алгоритмов в виде библиотеки функций.

Восстановление данных, многопроцессорные системы, блочно-синхронный параллелизм, системы реального времени, проектирование программного обеспечения, параллельная обработка, модель вычислительной системы

Восстановление данных требуется в разных областях обработки информации, в том числе в задачах прикладной гидроакустики. Пусть известны значения некоторой функции φ в K узлах $u^{(k)}$, $k = 0, \ldots, K-1$, равномерно расположен-

ных с шагом h. Необходимо приближенно найти неизвестные значения этой функции между заданными узлами (рис. 1).

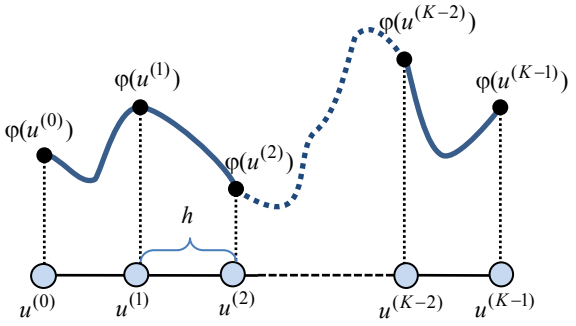

Рис. 1

В настоящей статье мы остановимся на вычислительных аспектах восстановления данных по известным значениям в узлах. Рассматривается случай, когда узлы, в которых требуется найти значения, также расположены равномерно. Например, таким образом может повышаться частота дискретизации сигналов. При этом, зачастую, необходимо обработать достаточно большие объемы данных. Представляются актуальными поиск и реализация эффективных параллельных алгоритмов для решения описанной задачи.

Был выбран алгоритм, предложенный В. В. Жуком [1], обладающий низкой вычислительной сложностью и хорошим потенциалом в смысле параллелизма по данным. На его основе построены эффективные параллельные алгоритмы в терминах модели блочно-синхронно-конвейерного параллелизма (БСКП), которая объединяет в себе модель вычислительной системы и формальное описание обработки потока данных в реальном времени [2].

В настоящей статье развивается хорошо изученный в литературе (например, в [3]−[5]) подход к проектированию алгоритмов с применением моделей параллельных вычислений. В то же время, отметим [6], где рассматриваются вопросы параллельной реализации похожих алгоритмов восстановления данных.

Описание модели параллельных вычислений. В основе модели БСКП лежит универсальная полуасинхронная модель параллельных вычислений (Bulk Synchronous Parallel, BSP) [3]. Модель BSP содержит набор вычислительных элементов с локальной памятью (или пар «процессор–память») и коммуникационную среду, которая позволяет пересылать данные между вычислительными элементами в режиме «точка–точка». Характеристиками коммуникационной среды в модели BSP служат время, необходимое для передачи по коммуникационной среде одного машинного слова, и время, необходимое для выполнения глобальной синхронизации. Вычислительный процесс в терминах модели BSP состоит из «супершагов», выполняющихся синхронно на каждом вычислительном элементе с данными в его локальной памяти. Между «супершагами» проводится барьерная синхронизация, сквозная для всех вычислительных элементов. Внутри «супершага» вычислительные элементы работают асинхронно.

Приведем основные сведения о модели БСКП. Вычислительная система в этой модели (далее БСКП-компьютер) включает в себя $N_{\text{ЭЦСП}}$ эквивалентов цифровых сигнальных процессоров (ЭЦСП), соединенных между собой эквивалентом коммуникационной среды (ЭКС) с пропускной способностью BW машинных слов в единицу времени. ЭКС полносвязная и однородная. В составе каждого ЭЦСП есть глобальная память объемом GMem машинных слов и эквиваленты вычислительных ядер (ЭВЯ) в количестве $N_{\text{ЭВЯ}}$, обладающие собственной локальной памятью объемом LMem. Пропускная способность передачи данных между глобальной памятью ЭЦСП и локальной памятью ЭВЯ составляет bw машинных слов в единицу времени. Множество всех ЭЦСП разбивается на $N_{\text{ВВЭ}}$ непересекающихся подмножеств, называемых виртуальными вычислительными элементами (ВВЭ). Введение абстракции ВВЭ позволяет отразить параллелизм по данным, если ресурса одного ЭЦСП недостаточно. Функционирование БСКП-компьютера может быть рассмотрено на двух уровнях [2].

БСКП-компьютер использует ресурсы по следующим правилам:

- Между ЭЦСП по ЭКС передаются данные, находящиеся в глобальной памяти ЭЦСП.
- ЭВЯ производит вычисления только с данными в своей локальной памяти.
- Предоставляется возможность обменов между глобальной памятью ЭЦСП и локальной памятью ЭВЯ на фоне вычислений на ЭВЯ.
- Данные между локальной памятью двух любых ЭВЯ в составе одного ЭЦСП передаются только через глобальную память ЭЦСП.

Вычисления в модели БСКП представляются в виде конвейера, состоящего из *стадий*, соединенных между собой *потоками*, по которым передаются данные. Работа конвейера рассматривается как итерационный процесс, разворачивающийся во времени. На каждой *итерации* стадия обрабатывает *входную порцию данных* (ВПД), полученную из входного потока. Обработка ВПД в стадии представляется набором *вычислительных процедур*, аналогом «супершага» в модели BSP. Каждая вычислительная процедура завершается барьерной синхронизацией. Это гарантирует, что результаты ее вычислений находятся в глобальной памяти ЭЦСП. После завершения любой из вычислительных процедур данные могут быть выданы в другие стадии конвейера. ВПД распределяется на *элементы порции данных* (ЭПД) между вычислительными элементами, на которых выполняется стадия, таким образом, что ЭПД помещается в локальной памяти ЭВЯ. Операции, которые выполняются над ЭПД, следуя [7], будем называть *вычислительным ядром*.

Алгоритм восстановления данных. Восстановление значения функции φ в узле x согласно выбранному алгоритму [1] выполняется следующим образом:

$$\mathcal{G}_r(\varphi, x, h) = \sum_{k=-\left\lceil\frac{r+1}{2}\right\rceil}^{K-1+\left\lceil\frac{r+1}{2}\right\rceil} \varphi\left(u^{(k)}\right)\psi_{r+2}\left(\frac{x}{h}-k\right), \quad (1)$$

где $r \in \mathbb{Z}_+$, x произвольно расположен между узлами исходной равномерной сетки $u^{(k)} = kh$, $k = 0, \ldots, K-1$, $h > 0$; для $\rho \in \mathbb{N}$, $t \in \mathbb{R}$ ядро В. А. Стеклова порядка ρ определяется формулой

$$\psi_\rho(t) = \begin{cases} \dfrac{1}{(\rho-1)!}\displaystyle\sum_{0\le k<\left(|t|+\frac{\rho}{2}\right)} (-1)^k \times \\ \times C_\rho^k\left(|t|+\dfrac{\rho}{2}-k\right)^{\rho-1}, & \text{если } |t| \le \dfrac{\rho}{2}, \\ 0, & \text{если } |t| > \dfrac{\rho}{2}. \end{cases} \quad (2)$$

Нам потребуются частные случаи (2) для ядер порядков 2 и 4

$$\psi_2(t) = \begin{cases} 1-|t|, & \text{если } |t| \le 1, \\ 0, & \text{если } |t| > 1. \end{cases} \quad (3)$$

$$\psi_4(t) = \frac{1}{6}\begin{cases} 3|t|^3 - 6t^2 + 4, & \text{если } |t| \le 1, \\ -|t|^3 + 6t^2 - 12|t| + 8, & \text{если } 1 < |t| \le 2, \\ 0, & \text{если } |t| > 2. \end{cases} \quad (4)$$

Из (1) следует, что для вычислений могут быть нужны дополнительные узлы по краям исходной сетки. Способы выбора значений в этих узлах зависят от специфики задачи, для которой применяется (1), и в данной работе не рассматриваются.

Будем полагать, что K узлов, в которых известны значения функции φ, расположены на отрезке $[0, K-1]$ с единичным шагом ($h = 1$). Сетку с известными значениями в узлах будем называть *крупной сеткой*. Требуется вычислить значения в M узлах, равномерно расположенных на исходном отрезке, где N – произвольное натуральное число. Сетку размером $M = N(K-1)+K$ будем называть *мелкой сеткой*. Обозначим $N^* = N+1$. В этих обозначениях количество узлов мелкой сетки $M = N^*(K-1)+1$. Расположение узлов показано на рис. 2.

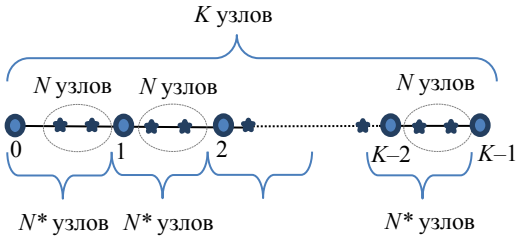

Рис. 2

Поскольку при дальнейшем изложении удобнее оперировать массивами, введем для них следующие обозначения: через F обозначим массив размером K, где содержатся известные значения функции φ в узлах крупной сетки; G_0 и G_2 – массивы размером M, содержащие значения аппроксимирующей функции в узлах мелкой сетки, вычисленные по (1) при $r = 0$ (без сглаживания) и при $r = 2$ (со сглаживанием) соответственно; Ψ_2 и Ψ_4 – массивы, содержащие значения ядер В. А. Стеклова ψ_2 и ψ_4, вычисленные по (3) и (4) соответственно. Учитывая финитность и симметрию ядер В. А. Стеклова, в массиве Ψ_2 нужно хранить N^* значений Ψ_2, а в массиве Ψ_4 – $2N^*$ значений Ψ_4.

Схема вычислений. Значения массивов Ψ_2 и Ψ_4 вычисляются один раз на этапе инициализации. Далее приведены выражения для формирования элементов массивов искомых значений в узлах мелкой сетки.

Для восстановления данных без сглаживания искомые значения в узлах мелкой сетки формируются следующим образом:

$$G_0(m) = F\left(\left[\frac{m}{N^*}\right]\right)\Psi_2(m \bmod N^*) + F\left(\left[\frac{m}{N^*}\right]+1\right)\Psi_2(N^* - m \bmod N^*), \quad (5)$$

где $m = 0, \ldots, M-1$.

Количество арифметических операций при вычислениях непосредственно по (1) равно $3M$.

Выражение для восстановления данных со сглаживанием имеет вид

$$G_2(m) = F\left(\left[\frac{m}{N^*}\right]-1\right)\Psi_4(N^* + m \bmod N^*) + F\left(\left[\frac{m}{N^*}\right]\right)\Psi_4(m \bmod N^*) + F\left(\left[\frac{m}{N^*}\right]+1\right)\Psi_4(N^* - m \bmod N^*) + F\left(\left[\frac{m}{N^*}\right]+2\right)\Psi_4(2N^* - m \bmod N^*), \quad (6)$$

где $m = 0, \ldots, M-1$.

Количество арифметических операций при вычислениях непосредственно по (2) равно $7M$.

Заметим, что каждое из произведений известного значения в узле крупной сетки со значениями ядер в узлах мелкой сетки используется несколько раз при формировании результатов в нескольких соседних интервалах крупной сетки. С учетом этого факта предложена следующая двухэтапная вычислительная схема.

1. Вычисление для k-го $(k = 0, ..., K-1)$ известного значения функции элементов массива произведений $\Pi_2(n) = F(k)\Psi_2(n)$, где $n = 0, ..., N^* - 1$ или $\Pi_4(n) = F(k)\Psi_4(n)$, где $n = 0, ..., 2N^* - 1$. Выполняется KN^* умножений в первом варианте и $2KN^*$ умножений – во втором.

2. Выбор и суммирование полученных на предыдущем шаге произведений для формирования результирующих значений в узлах мелкой сетки. Выполняется M сложений в первом варианте и $3M$ сложений – во втором.

При такой схеме вычислений общее количество арифметических операций для восстановления данных без сглаживания составляет $2KN + 2K - N$ (что меньше, чем $3M = 3NK + 3K - 3N$) и $5KN + 5K - 3N$ (что меньше, чем $7M = 7NK + 7K - 7N$) для восстановления данных со сглаживанием.

Задача восстановления данных в узлах мелкой сетки рассматривается как одна стадия конвейера в модели БСКП. ВПД этой стадии является массив F размером K, содержащий известные значения в узлах крупной сетки. Обработка ВПД состоит из двух вычислительных процедур, соответствующих перечисленным этапам.

Параллельные алгоритмы. Алгоритмы восстановления данных без сглаживания и со сглаживанием обладают параллелизмом по данным. Пусть данная стадия выполняется на ВВЭ состоящем из P ЭЦСП.

Рассмотрим вариант равномерного размещения значений в узлах крупной сетки при условии, что между вычислительными элементами на обоих уровнях модели БСКП (т. е. как между ЭЦСП, так и между ЭВЯ) предполагается обмен данными. Тогда на p-м ЭЦСП $(p = 0, ..., P-1)$ обрабатывается $K_p = K/P$ узлов, $K_p < G\text{Mem}$. Каждый ЭЦСП параллельно с остальными обрабатывает свою часть крупной сетки из K_p узлов и формирует часть мелкой сетки размером $M_p = N(K_p - 1) + K_p$. Эти узлы, в свою очередь, разбиты на ЭПД для вычислений на ЭВЯ. Размер каждого ЭПД $K_p^j = K_p / N_{\text{ЭВЯ}}$, где $j = 0, ..., N_{\text{ЭВЯ}} - 1$. Здесь для простоты изложения, считаем, что $K_p^j < L\text{Mem}/2$, K_p кратно $N_{\text{ЭВЯ}}$, а K кратно P.

На втором этапе вычислительной схемы в формировании узлов мелкой сетки на крайних интервалах каждой из частей крупной сетки участвуют произведения, которые оказались вычисленными в соседних вычислительных элементах. В случае восстановления данных без сглаживания каждый вычислительный элемент должен

получить дополнительные N^* значений, со сглаживанием – дополнительные $4N^*$ значений.

Если обмен данными между вычислительными элементами на каком-либо из уровней модели БСКП исключается, то необходимо дублировать данные в крайних узлах частей крупной сетки в соседних вычислительных элементах. Для алгоритма без сглаживания в каждом вычислительном элементе размещается $K_p + 1$ узлов крупной сетки и вычисляется на N^* больше произведений, а для алгоритма со сглаживанием – по $K_p + 3$ узла и вычисляется на $4N^*$ произведений больше.

Предложенный подход к распараллеливанию по данным эффективен, если на одном ЭЦСП обрабатывается $K_p \geq 2N_{\text{ЭВЯ}}$ узлов крупной сетки для алгоритма без сглаживания и $K_p \geq 4N_{\text{ЭВЯ}}$ узлов для алгоритма со сглаживанием. Следовательно, границы масштабируемости параллельных алгоритмов восстановления данных определяются следующими неравенствами: $P \leq \left[K/2N_{\text{ЭВЯ}}\right]$ – для алгоритма без сглаживания и $P \leq \left[K/4N_{\text{ЭВЯ}}\right]$ – для алгоритма со сглаживанием.

В таблице приведены выражения вычислительной сложности W, коммуникационной сложности H и количества барьерных синхронизаций S, необходимых для трех вариантов параллельных вычислительных алгоритмов восстановления значений в узлах мелкой сетки по (5) и по (6).

Из таблицы видно, что алгоритмы с передачей данных и без нее имеют одинаковую сложность, если пропускная способность, с которой вычислительные элементы могут обмениваться данными, равна единице. Если эта пропускная способность меньше единицы, то эффективнее алгоритм с передачей данных, а если больше, то выгоднее посчитать лишние произведения и данные не передавать.

Описание программной реализации. Алгоритмы восстановления данных без сглаживания и со сглаживанием реализованы в виде библиотеки функций на языке Си. Каждый из них состоит из двух вычислительных процедур. Рассмотрим подробнее реализованные для этих алгоритмов вычислительные процедуры.

Алгоритм A1 для восстановления данных в узлах мелкой сетки без сглаживания.

1. Каждый вычислительный элемент выполняет следующие действия:

• Формирует N^* элементов массива произведений Π_2 для первого узла части крупной сетки, обрабатываемой на текущем вычислительном элементе.

• В случае варианта *реализации с обменами* посылает N^* значений сформированных произведений другому вычислительному элементу (где обрабатывается предыдущая часть крупной сетки).

• В случае варианта *реализации без обменов* вычисляет N^* элементов массива произведений Π_2 для дополнительного K_p-го узла части крупной сетки при наличии этого узла. Если обрабатываемая часть сетки содержит узел с номером $K - 1$, то формируется «код отсутствия».

2. Каждый вычислительный элемент выполняет следующие действия:

• В случае варианта *реализации с обменами* принимает N^* дополнительных значений произведений (которые необходимы для формирования значений в узлах мелкой сетки на крайнем справа интервале) или «код отсутствия», если обрабатываемая на вычислительном элементе часть сетки содержит узел с номером $K - 1$.

Вариант	Без сглаживания	Со сглаживанием
Без применения двухэтапной вычислительной схемы	$W = 3K_p(N+1) - 3N;$ $H = 0; S = 0$	$W = 7K_p(N+1) - 7N;$ $H = 0; S = 0$
По двухэтапной вычислительной схеме *с обменами*	$W = 2K_p(N+1) - N;$ $H = N+1; S = 1$	$W = 5K_p(N+1) - 3N;$ $H = 4(N+1); S = 1$
По двухэтапной вычислительной схеме *без обменов*	$W = 2K_p(N+1) + 1;$ $H = 0; S = 1$	$W = 5K_p(N+1) + N + 4;$ $H = 0; S = 1$

• В случае варианта *реализации без обменов* в качестве дополнительных значений используют произведения для K_p-го узла, сформированные текущим вычислительным элементом в предыдущей вычислительной процедуре.

• Независимо формирует оставшиеся $(K_p - 1)N^*$ произведений и вычисляет попарным суммированием соответствующих произведений $K_p N^*$ значение в узлах мелкой сетки при наличии дополнительных данных или $(K_p - 1)N^*$ значений в случае «кода отсутствия».

Алгоритм А2 для восстановления данных в узлах мелкой сетки со сглаживанием.

1. Каждый вычислительный элемент выполняет следующие действия:

• Формирует три массива произведений Π_4 для узлов с номерами 0, 1 и $K_p - 1$ части крупной сетки, обрабатываемой на текущем вычислительном элементе.

• В случае варианта *реализации с обменами* посылает $2N^*$ значений произведений для 0-го узла текущей части крупной сетки и N^* значений произведений для 1-го узла текущей части крупной сетки вычислительному элементу, где обрабатывается предыдущая часть крупной сетки, а N^* значений произведений для последнего узла – вычислительному элементу, где обрабатывается ее следующая часть.

• В случае варианта *реализации без обменов* вычисляет N^* элементов массива произведений $\Pi_4(n),\ n = N^*, ..., 2N^* - 1,$ для дополнительного узла слева; $2N^*$ элементов массива произведений $\Pi_4(n),\ n = N^*, ..., 2N^* - 1,$ для первого дополнительного узла справа; N^* элементов массива произведений $\Pi_4(n),\ n = N^*, ..., 2N^* - 1,$ для второго дополнительного узла справа. В случае отсутствия какого-либо из дополнительных узлов вместо соответствующих произведений формируются «коды отсутствия».

2. Каждый вычислительный элемент выполняет следующие действия:

• В случае варианта *реализации с обменами* принимает $4N^*$ дополнительных значений произведений или «коды отсутствия».

• В случае варианта *реализации без обменов* вместо принятых значений произведений использует произведения для соответствующих дополнительных узлов, сформированные текущим вычислительным элементом в предыдущей вычислительной процедуре.

• Независимо формирует $2N^*(K_p - 3)$ оставшихся произведений и суммирует по 4 соответствующих произведения, получая $K_p N^*$ значений в узлах мелкой сетки при наличии дополнительных данных справа или $(K_p - 1)N^*$ значений, если часть сетки содержит узел с номером $K - 1$.

Объединение суммирования с формированием произведений для некрайних узлов крупной сетки во второй вычислительной процедуре позволяет сократить объем памяти, необходимой для хранения значений произведений. Вместо хранения второго экземпляра мелкой сетки со всеми значениями произведений можно хранить $2 \times N^*$ или $4 \times 2N^*$ произведений соответственно, которые нужны для вычисления значений в узлах мелкой сетки на текущем интервале. Для организации хранения произведений применяется структура данных «кольцевой буфер».

В статье предложена схема вычислений для алгоритмов восстановления, которая позволяет сократить количество арифметических операций, требуемых для их выполнения. На базе этой схемы построены эффективные параллельные алгоритмы для класса вычислительных систем, описываемых моделью БСКП. Проведена теоретическая оценка вычислительной и коммуникационной сложности предложенных алгоритмов, выведены зависимости между размерностями задачи и параметрами модели БСКП. Рассмотренные алгоритмы применяются для повышения частоты дискретизации сигналов при программной имитации гидроакустической информации линейных антенных решеток.

Участие в исследованиях канд. физ.-мат. наук Г. Ю. Пуерова поддержано грантом РНФ № 18-11-000555.

СПИСОК ЛИТЕРАТУРЫ

1. Жук В. В. Методические указания к курсу «Теория аппроксимации функций и ее приложения». Ч. 2. СПб.: Изд-во С.-Петербургского ун-та, 1993.

2. Лисс А. Р., Пуеров Г. Ю., Сергеева Е. И. Модель системы параллельной обработки гидроакустической информации в реальном масштабе времени // Изв. СПбГЭТУ «ЛЭТИ». 2020. № 3. С. 29–38.

3. Valiant L. G. A bridging model for parallel computation // Communications of the ASM. 1990. № 33 (8). P. 103–111.

4. Buurlage J.-W., Bannink T., Wits A. Bulk-synchronous pseudo-streaming algorithms for many-core accelerators. 2016. URL: http://arXiv.org/abs/1608.07200 (дата обращения 24.03.2021).

5. McColl W. F., Tiskin A. Memory-efficient matrix multiplication in the BSP model // Algorithmica. 1999. № 24 (3–4). P. 287–297.

6. Масальских А. В. Параллельный алгоритм одного метода восстановления табличных данных // Изв. Тульского гос. ун-та. Естественные науки. 2014. № 3. С. 167–177.

7. Сударева О. Ю. Встречная оптимизация класса задач трехмерного моделирования для архитектур многоядерных процессоров. Дис. … канд. физ.-мат. наук: 05.13.11. М.: Ин-т систем. программирования им. В. П. Иванникова РАН, 2018.

A. R. Liss
Saint Petersburg Electrotechnical University

G. Yu. Puerov, E. I. Sergeeva
JSC «Concern «Oceanpribor» (Saint Petersburg)

EFFICIENT PARALLEL ALGORITHMS FOR DATA RECOVERY ON UNIFORM GRID

The article is devoted to the computational aspects of real time data recovery problem. One of the known approaches to solving the problem is described. This approach has low computational complexity and good potential in data parallelism. Parallel algorithms for data recovery from known values at the nodes on uniform grid are considered. Such algorithms are widely used in various applied fields of information processing. Approaches to data-parallel computations in bulk-synchronous notation are studied and bulk-synchronous parallel (BSP) model is extended for real-time streaming data processing. A model of bulk-synchronous-pipeline parallelism (BSPP) is built for streaming processing in real time, which is based on the well-known general-purpose parallel computing model BSP. Parallel algorithms for data recovery are described in terms of this extended model. Computational and communicational complexity of the proposed algorithms is assessed. Dependences between the dimensions of the problem and the parameters of the extended model are derived. The software that implements proposed algorithms is described.

Data recovery, multiprocessor system, bulk-synchronous parallelism (BSP), real-time system, signal processing, parallel prosesing, software desing

УДК 004.56

Е. О. Кузнецова, С. А. Петренко
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Подход к обнаружению кибератак на основе вычислительного когнитивизма

Исследована комплексная научная задача обнаружения кибератак на отечественные ведомственные и корпоративные информационные системы. Предложен и обоснован подход к созданию систем обнаружения кибератак на основе так называемого вычислительного когнитивизма – сравнительно нового научного направления исследований, в котором познание и когнитивные процессы – это разновидность символьного вычисления. Показано, что когнитивный подход позволяет создавать системы, принципиально отличающиеся от известных систем обнаружения, предупреждения и ликвидаций последствий компьютерных атак уникальной способностью к самостоятельному ассоциированию и синтезу новых знаний о качественных характеристиках и количественных закономерностях информационного противодействия в киберпространстве. Предложен системный облик когнитивной системы обнаружения кибератак на основе вычислительного когнитивизма.

Информационная безопасность, обнаружение кибератак, система обнаружения кибератак, вычислительный когнитивизм, количественные закономерности противодействия в киберпространстве, конвергентные NBIC-технологии, технологии Big Data

В Российской Федерации уже создан ряд государственных и корпоративных центров обнаружения, предупреждения и ликвидации последствий компьютерных атак (СОПКА) или центров реаги-