

2. Пригожин, И. Р. Конец определенности, Время, Хаос и Новые законы Природы / И. Р. Пригожин. – Ижевск: R & C Dynamics, 2000. – 205 с.
3. Cyber-Physical and Ultra-Large Scale Systems. – CMU, 2013. WW.SEI.CMU.EDU.
4. Greengard, S. The Internet of Things / S.Greengard. – Cambridge: The MIT Press, 2015. – 232 p.
5. Бусленко, Н. П. Моделирование сложных систем / Н. П. Бусленко. – М.: Наука, 1978. – 400 с.
6. Бурков, В. Н. Большие системы: моделирование организационных механизмов / В. Н. Бурков, Б. Данев, А. К. Еналеев. – М.: Наука, 1989. – 246 с.
7. Управление развитием крупномасштабных систем (MLSD'2017): труды десятой Междунар. конфер., 2-4 окт. 2017 г., Москва: в 2-х т. – М.: ИПУ РАН, 2017. – 533 с.
8. Hissman, S. Ultra-Large-Scale Systems: Socio-adaptive Systems / S.Hissman – CMU, 2016. – P.2-17.
9. Берг, А. И. Управление, информация, интеллект / А. И. Берг. – М.: Мысль, 1976. – 384 с.
10. Rosen, R. Anticipatory Systems: Philosophical, Mathematical and Methodological Foundations / R.Rosen – Springer, 2012. – 472 p.
11. Охтилев, М. Ю. Теоретические и технологические основы концепции проактивного мониторинга и управления сложными объектами / М. Ю. Охтилев, Б. В. Соколов, Р. М. Юсупов // Известия ЮФУ. – 2015. – № 1. – С. 162-174.
12. 1516.1-2000 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture. Federate Interface Specification 2000. – IEEE, 2000. – 480 p.
13. Краснопрошин, В. В. Модель сцены и адаптивный алгоритм мониторинга транспортных средств в условиях неопределенности / В. В. Краснопрошин, А. И. Кузьмич // Вестник БГУ. – Сер. 1. Физика. Математика. Информатика. – 2015. – № 2. – С. 107–112.
14. Краснопрошин, В. В. Принятие решений в оперативных задачах регионального управления / В. В. Краснопрошин, Х. Виссия, А. Н. Вальвачев // Таврический вестник информатики и математики. – 2008. – № 1. – С. 267-273.
15. Петровский, А. Б. Теория принятия решений / А. Б. Петровский. – М.: ИЦ Академия, 2009. – 400 с.
16. Лепский, А. Е. Математические методы распознавания образов / А. Е. Лепский, А. Г. Броневиц. – Таганрог: ТТИ ЮФУ, 2009. – 155 с.
17. Pasco, A. Heterogeneous objects modelling and applications / A. Pasco, V. Adziew, P. Comninos. – Springer, 2008. – 285 p.
18. Arp, R. Building Ontologies with Basic Formal Ontology / R. Arp, B. Smith, A. Spear. – Cambridge: The MIT Press, 2015. – 248 p.
19. Zadeh, L. Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems / L. Zadeh, G. Klir, B. Yuan. – World Scientific Pub. Co. Inc., 1996. – 840 p.
20. Бродский, Ю. И. Распределенное имитационное моделирование сложных систем / Ю. И. Бродский. – М.: ВЦ РАН, 2010. – 156 с.

Материал поступил в редакцию 08.02.2019

VISSIA H., KRASNOPROSHIN V. V., VALVACHEV A. N. Monitoring super large systems

The paper deals with the problem of decision making in super large organizational and technical systems. It is shown that due to the complexity of obtaining and processing information in the decision-making process, the effect of “delayed reaction” to a problem situation arises. A complex of models and algorithms is proposed, on the basis of which a decision support system is built. The effectiveness of the proposed approach is demonstrated by the example of solving a specific applied problem.

УДК 519.725;007.001.362;528.85/.87(15)

Дудкин А. А., Перцев Д. Ю.

СЖАТИЕ ДАННЫХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ С ПРИМЕНЕНИЕМ КОНТЕКСТНОГО МОДЕЛИРОВАНИЯ

Введение. Дистанционное зондирование Земли (ДЗЗ) является незаменимым инструментом для изучения и мониторинга планеты, помогающим эффективно использовать и управлять ее ресурсами. Областью применения являются сельское хозяйство, геологические и гидрологические исследования и другие.

На протяжении последних 20 лет из-за развития оптического оборудования и технологий в целом наблюдается смещение в область гиперспектральной съемки. С учетом того, что данным типом оборудования фиксируется непрерывный диапазон, объем данных пропорционально изменяется с увеличением исследуемого диапазона. Например, данные спектрометра CASI [1], которые свободно доступны и используются в исследовательских целях, обладают следующими техническими характеристиками: ширина изображения – 512 пикселей, 288 спектральных канала, 12 бит на канал, что, в общем случае, приводит к 216 Кбайтам данных на одну строку. С учетом того, что съемка осуществляется непрерывно, требуется разработка алгоритма сжатия, который может быть использован в ограниченных условиях летательного аппарата и при этом окажется достаточно эффективным.

Алгоритм сжатия с применением контекстного моделирования. Блок-схема разработанного алгоритма представлена на рисунке 1 и включает следующие шаги:

- 1) перераспределение потока данных для сжатия;
- 2) предобработка входных данных;

- 3) декорреляция данных в спектральной области;
- 4) упаковка и реорганизация декоррелированного потока;
- 5) применение алгоритмов контекстного моделирования и арифметического кодера.

На первом шаге выполняется перераспределение потока данных для сжатия. Результатом работы спектрометра является поток сырых данных, для которого в зависимости от типа спектрометра выполняется восстановление трехмерной структуры [2]:

- при использовании призмы в качестве осей X и Y используется пространственное разрешение, ось Z определяет спектральную составляющую;
- при использовании Фурье-спектрометра [3] оси X и Y определяют получаемую интерференционную картину, ось Z – смещение спектрометра в пространстве в результате движения.

В восстановленном гиперкубе каждый кадр разбивается на квадратные блоки фиксированной размерности, равные 2^r , где r – некоторое число. Недостающая часть кадра при необходимости при разбиении на блоки заполняется нулями.

Суммарно формируется K потоков для сжатия, где K определяется по формуле:

$$k = \frac{Z}{n} \cdot \frac{X \cdot Y}{block_size} \quad (1)$$

Перцев Дмитрий Юрьевич, ст. преподаватель кафедры ЭВМ Белорусского государственного университета информатики и радиоэлектроники.

Беларусь, 220013, г. Минск, ул. П. Бровки, 6.

Физика, математика, информатика



Рисунок 1 – Общая схема работы алгоритма сжатия

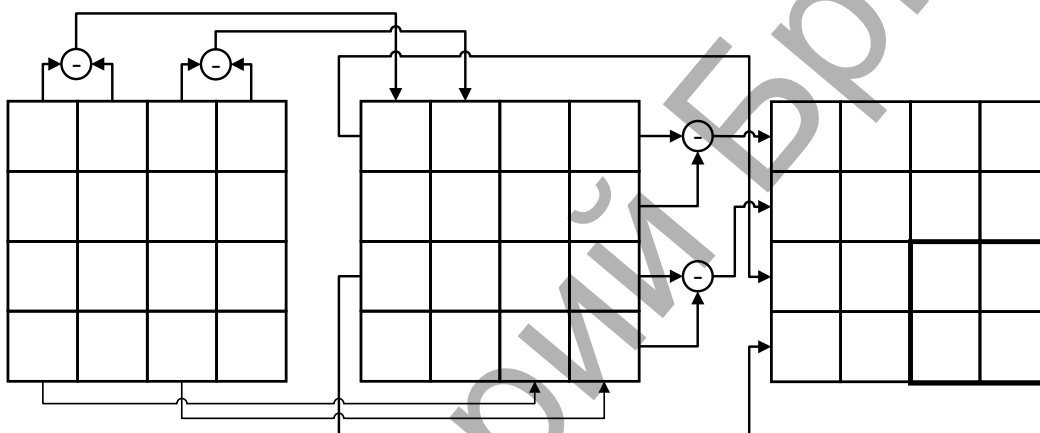


Рисунок 2 – Схема вейвлет-разложения

где X , Y – пространственное разрешение гиперкуба, Z – число каналов, $block_size = 2^r$ – размер блока, на который разбивается канал, в пикселях.

Предлагаемая подготовка приводит к повышению устойчивости алгоритма сжатия к ошибкам при передаче данных, т. к. сбой в алгоритме искажает сжатый поток, относящийся к конкретному фрагменту, и не влияет на остальные. Кроме того, возможна одновременная обработка всех блоков.

На *втором шаге* выполняется вейвлет-преобразование, допускающее полное восстановление. В общем случае при выполнении данного шага выполняются следующие операции (рис. 2):

- вычитается по два соседних столбца;
- полученная ошибка сохраняется в первой половине столбцов кадра в качестве результата. Во второй половине кадра сохраняются опорные значения;

Описанные операции применяются ко всем столбцам кадра, затем к строкам по аналогии.

Если число уровней разложения больше 1, описанный алгоритм повторяется для нижней правой четверти кадра (на рис. 2 выделен жирным).

После выполнения вейвлет-разложения на *третьем шаге* выполняется понижение избыточности данных по оси. Для этого применяется вычитание соседних кадров.

Полученный бинарный поток разбивается на 3 фрагмента, определяемые типом коэффициентов (высокочастотные HH, HL, LH, низкочастотные LL), каждый из которых в дальнейшем обрабатывается независимо. Пример структуры выходных данных при уровне вейвлет-разложения, равного 3, показан на рис. 3.

С учетом того, что в результате выполнения предыдущих шагов в потоке находятся значения, соответствующие распределению

Лапласа, необходимо выполнить преобразование к экспоненциальному распределению. Для этого стандарт CCSDS [4] рекомендует к каждой ошибке предсказания применить следующее выражение:

$$\hat{e}_i = \begin{cases} 2e_i, & 0 \leq e_i \leq \theta_i \\ 2|e_i| - 1 - \theta_i, & e_i \leq 0, \\ \theta_i + |e_i|, & \text{иначе} \end{cases} \quad (2)$$

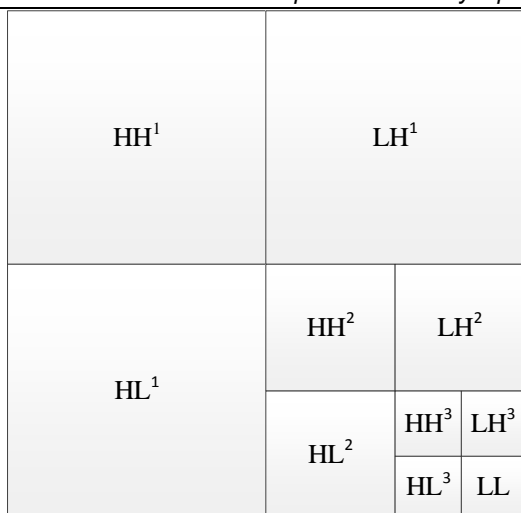
где e_i – ошибка предсказания для элемента i ;

$\theta_i = \min(\hat{y}_i, y_{\max} - \hat{y}_i)$; $y_{\max} = 2^n - 1$ – максимально возможное число во входной последовательности.

В результате преобразования динамический диапазон допустимых значений увеличивается в 2 раза. Однако анализ гистограмм показывает, что основное число элементов не превышает 255, а второй байт является нулевым. С учетом этого выполняется упаковка и реорганизация данных (*шаг 4*). Более подробно данный шаг описан в [5].

Упаковка декоррелированного потока данных включает:

- выполнение нормализации входных данных по формуле 2;
 - выполнение упаковки нормализованных данных (рис. 4).
- Кроме того, предполагая, что значения пикселей в окрестности близки между собой, в т. ч. между соседними каналами, и, следовательно, значения битовых плоскостей, начиная с наиболее значимой, одинаковы, выполняется реструктуризация преобразованного входного потока:
- упакованный декоррелированный поток данных, полученный на предыдущем этапе, восстанавливается в трехмерную структуру;
 - полученный трехмерный куб разбивается на блоки фиксированной размерности;



поток 1: $NH^1 NH^2 NH^3 LL$
 поток 2: $HL^1 HL^2 HL^3$
 поток 3: $LH^1 LH^2 LH^3$

Рисунок 3 – Схема формирования потока данных для последующей обработки

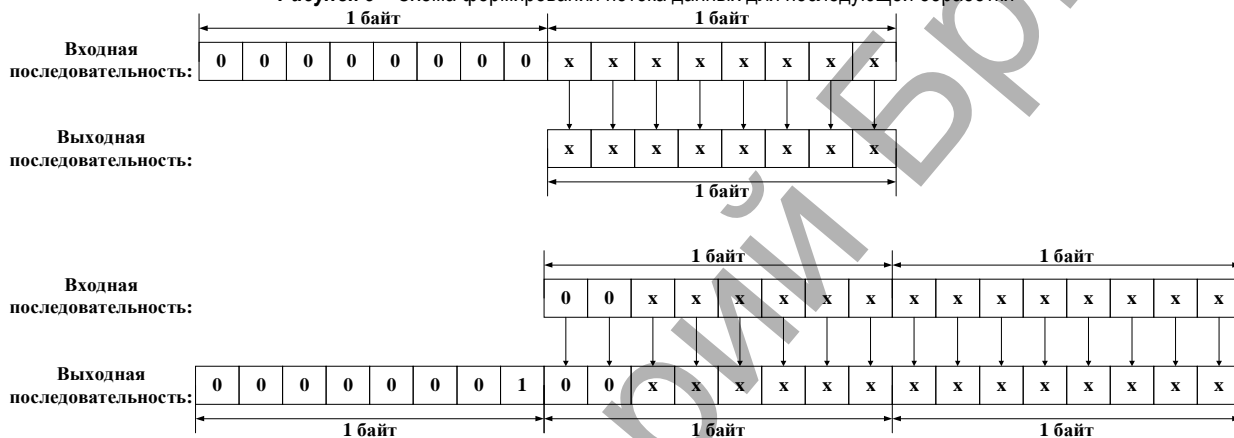


Рисунок 4 – Схема преобразования входного потока

- выполняется переупорядочивание данных в каждом блоке в соответствии с алгоритмом 3D zig-zag [6]. Данная операция выполняется отдельно для каждой битовой плоскости.

Алгоритм 3D zig-zag основан на классической версии zig-zag, применяемой в JPEG, и сводится к преобразованию трехмерной структуры в новую одномерную.

Пусть (x, y, z) – координаты пикселя в блоке. Основное отличие алгоритма zig-zag для 3D-версии заключается в том, что при изменении значений x или y в выходной поток последовательно передаются все элементы с координатами $(x, y, z_0), (x, y, z_1), \dots, (x, y, z_{max-1})$, где z_0 и z_{max-1} – начальный и конечный спектральный каналы выбранного блока. Последовательность изменения координат x или y эквивалентна той, что применяется в 2D-версии.

На пятом шаге выполняется арифметическое кодирование. В результате выполнения реорганизации упакованного потока формируется бинарная избыточная последовательность бит. Под бинарной избыточностью понимается высокая вероятность появления в потоке двух и более нулевых или единичных бит подряд.

Для уменьшения избыточности используется контекстно-адаптивный QM-кодер (англ. Context Adaptive QM coder) [7], являющийся одной из разновидностей арифметического кодера и включающий этапы, выполняемые последовательно для каждого бита:

- контекстное моделирование.
- бинарное арифметическое кодирование.

Основной задачей бинарного арифметического кодера является кодирование входных символов, в качестве которых допускаются два значения – 0 и 1 в общем случае с одинаковой вероятностью появления.

Для каждой контекстной модели с учетом вероятности появления символов формируются интервалы значений:

- More Probable Symbol (MPS);
- Less Probable Symbol (LPS).

Пусть C – начало интервала, A – кодируемый диапазон значений (допустимый интервал значений), Q и $(1-Q)$ – вероятности того, что следующий входной символ будет отнесен к категории LPS и MPS соответственно. В этом случае длина интервала LPS может быть определена как AQ , интервала MPS – $A(1-Q)$.

При кодировании входного символа выполняется обновление интервала (значения C и A):

- если символ определен как MPS:
 - ✓ C не изменяется
 - ✓ $A = A \cdot (1 - Q)$
- если символ определен как LPS:
 - ✓ $C = C + A \cdot (1 - Q)$
 - ✓ $A = A \cdot Q$

Новое значение вероятности Q получается на основе таблицы, представленной в [8] по мере кодирования очередного символа по следующим правилам:

- если кодируемый символ отнесен к MPS, текущий индекс вероятности в таблице увеличивается на значение из таблицы;
 - если кодируемый символ отнесен к LPS, текущий индекс вероятности в таблице уменьшается на значение из таблицы.
- В результирующий сжатый поток передаются следующие данные:
- если значение A превысило максимально допустимое значение (0x10000), в результирующий поток записывается бит со значением единица. При этом значение A уменьшается на 0x10000;

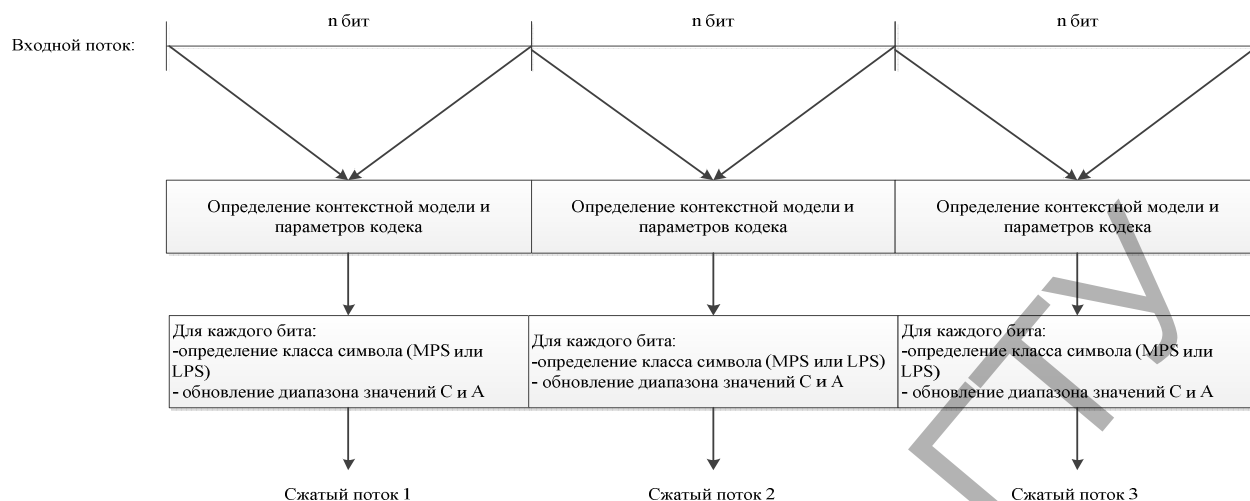


Рисунок 5 – Последовательность операций, выполняемых при арифметическом кодировании

- если значение A меньше нижней границы ($0x8000$), в результирующий поток записывается наиболее значимый бит значения C .
Основной задачей *контекстного моделирования* является определение модели бинарного арифметического кодека, которая будет использоваться при сжатии, на основе закодированных ранее данных. Для этого определяется число битовых плоскостей $n = \lceil \log_2(c_{\max}) \rceil$, где c_{\max} – максимальное значение в упакованном потоке данных. В общем случае максимальное число моделей определяется по формуле:

$$\text{max_number_of_models} = 2^n. \quad (3)$$

Дополнительно для каждой модели хранятся следующие параметры:

- – текущий индекс значения вероятности Q ;
- – начальное значение интервала C ;
- – кодируемый диапазон значений A .

Таким образом, алгоритм работы контекстного моделирования сводится к следующей последовательности действий:

- из входного потока извлекается n бит, формирующих символ;
- по заведомо определенному алгоритму выбирается применяемая модель (на основе анализа вероятностей появления символа в потоке);
- выполняется бинарное арифметическое кодирование в соответствии с настройками для данной модели;
- обновляются параметры выбранной модели.

Общая последовательность операций, выполняемых при арифметическом кодировании, показана на рис. 5.

Тестирование предвзятого алгоритма сжатия. Для тестирования алгоритма сжатия использовался AVIRIS Maine со следующими техническими характеристиками:

- пространственное разрешение – 680x512 пикселей;
- радиометрическое разрешение – 12 бит на пиксель;
- число спектральных слоев – 224;
- суммарный объем данных – 114 240 кБайт.

При этом использовались следующие параметры:

- уровень вейвлет-разложения – 5.

Результаты тестирования без контекстного моделирования и на 2 и 3 моделей представлены в таблице 1.

Таблица 1 – Результаты тестирования алгоритма сжатия

	Коэффициент сжатия, раз
Без контекстного моделирования (на основе арифметического кодирования)	3,26
С применением контекстно-адаптивного QM-кодека (контекст на основе 2 моделей)	3,99
С применением контекстно-адаптивного QM-кодека (контекст на основе 3 моделей)	4,05

Заключение. Полученные результаты свидетельствуют об эффективности предлагаемого алгоритма сжатия. Предлагаемая последовательность операций является математически простой и не требует существенных вычислительных ресурсов, что позволяет применить алгоритмы на борту летательного аппарата.

В дальнейшем предполагается:

- исследование различных версий вейвлет-преобразований;
- расширение тестового набора (предполагается включить тесты AVIRIS, LANDSAT, SPOT-4).

Работа выполнена при финансовой поддержке БРФФИ (проекты № Ф18В-005 и Ф18М-081) и ГКНТ Республики Беларусь (проект № Ф18ПЛШГ-008П).

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. CASI-1500 [Электронный ресурс] / ITRES. – Режим доступа: <http://www.itres.com/casi-1500/>. – Дата доступа: 30.11.2018.
2. Перцев, Д. Ю. Алгоритм сжатия гиперспектральных данных дистанционного зондирования Земли / А. А. Дудкин, Д. Ю. Перцев // Весці Нацыянальнай акадэміі навук Беларусі. – 2017. – № 1. – С. 120–126.
3. Перцев, Д. Ю. Алгоритм сжатия Фурье-интерферограмм / Д. Ю. Перцев, А. А. Дудкин, Л. П. Поденок // Приложение к сборнику «Вопросы электромеханики. Труды ВНИИЭМ». – М., 2016. – С. 227–233.
4. Lossless data compression [Электронный ресурс] / CCSDS. – Режим доступа: <https://public.ccsds.org/Pubs/120x0g3.pdf>. – Дата доступа: 30.11.2018.
5. Перцев, Д. Ю. Применение алгоритмов контекстного моделирования в задачах сжатия гиперспектральных данных, основанных на Фурье-интерферограммах / Д. Ю. Перцев // Молодежь в науке – 2017 : сб. материалов Междунар. конф. молодых ученых (Минск, 30 окт. – 2 нояб. 2017 г.): в 2 ч. – Минск, 2018. – Ч. 2: Гуманитарные, медицинские, физико-математические, физико-технические, химические науки. – С. 179–186.
6. Alptekin Engin, M. New Approach in Image Compression: 3D Spiral JPEG / M. Alptekin Engin, B. Cavusoglu // IEEE Communication Letter. – Vol.15, No.11. – 2011. – P. 1234-1236.
7. Tian, X. Entropy Coders of the H.264/AVC Standard / X. Tian, Y. Lian, T. Le. – Signals and Communication Technology. Springer, Berlin, 2011. – P. 180.
8. QM Coder [Electronic Resource] / Jigar Gada, University of Southern California. – Mode of access: https://github.com/jigar23/CABAC/blob/master/CABAC/HW2_Report_bac.pdf. – Data access: 10.09.2018.

Материал поступил в редакцию 21.02.2019

In the paper the hyperspectral data compression algorithm is presented. It is accented to a possibility of its application in limited computing resources of the onboard system of the aircraft. The compression ratio was estimated using the AVIRIS Maine dataset.

УДК 004.89

Краснопрошин В. В., Мацкевич В. В.

ЭФФЕКТИВНАЯ ОБРАБОТКА ДАННЫХ НА ГЕТЕРОГЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВАХ

Введение. В настоящее время наблюдается быстрый рост вычислительной сложности прикладных задач, которая во многом зависит от объемов обрабатываемых данных, необходимых для их решения [1]. Одновременно с этим растет и вычислительная мощность современных компьютеров. Последнее, как правило, происходит путем увеличения количества вычислительных устройств не только в рамках одного компьютера, но и путем формирования различного рода вычислительных кластеров [2]. На сегодняшний день актуальной стала проблема, связанная с эффективным использованием вычислительных ресурсов [2]. Вычислительные устройства при решении прикладных задач могут оказаться гетерогенными (разными по архитектуре и мощности), в этом случае возникает необходимость эффективной их загрузки [3]. Что является нетривиальной математической задачей [4].

В работе предлагается один из возможных вариантов решения проблемы с использованием технологии распараллеливания данных.

1. Формализация задачи. Любую прикладную задачу (Z) в самом общем виде можно представить двумя составляющими: данными (D) и алгоритмом решения (A) [5], т. е.

$$Z = (D, A) \quad (1)$$

В свою очередь процесс решения можно задать сетевым графиком (G), который состоит из набора вычислительных операций (B) и зависимостей между ними (E) [6], т. е.

$$A = G(B, E) \quad (2)$$

При этом порядок выполнения B может быть не однозначным [6].

В результате прикладную задачу можно записать в виде:

$$Z = (D, G(B, E)) \quad (3)$$

Предположим, что любая операция b_i ($i = \overline{1, |B|}$) из B является неделимой и требует объема вычислений, равного единице, а используемый для решения задачи компьютер содержит n вычислительных устройств, и каждое i -е устройство ($i = \overline{1, n}$) имеет свою вычислительную мощность U_i .

Предположим также, что время решения T зависит от специфики задачи, мощности вычислительных устройств и нагрузки на устройства (Y) [7], т. е.

$$T = F(Z, U, Y) \quad (4)$$

где U – n мерный вектор, координаты которого задают мощность отдельного вычислительного устройства, а F – функционал, определяющий время решения задачи Z при заданных вычислительных устройствах U с нагрузкой Y .

Далее предположим, что нагрузка Y определяется разбиением множества операций B на n подмножеств B_i , $i = \overline{1, n}$, каждое из которых задает набор операций, выполняемых на i -м вычислительном устройстве.

В результате получаем оптимизационную задачу, связанную с минимизацией функционала F [7].

Для задачи Z с использованием имеющихся вычислительных

устройств мощности U_i , $i = \overline{1, n}$ необходимо определить нагрузку на эти устройства Y , при которой время решения задачи будет минимальным, т. е. необходимо определить нагрузку Y , удовлетворяющую соотношению:

$$F(Z, U, Y) = \arg \min_Y F(Z, U, Y) \quad (5)$$

2. Алгоритм вычислений. В общем случае задача минимизации функционала (5) является достаточно сложной, поэтому рассмотрим частный случай, когда на задачу Z наложен ряд ограничений.

Из соотношения (3) видно, что задача Z зависит от исходных данных D и сетевого графика G , который задает процесс ее решения. Сетевой график, в свою очередь, задается набором вычислительных операций B и связей между ними E .

Предположим, что каждый набор операций B содержит в себе ненулевое количество непересекающихся между собой поднаборов S_q , причем суммарное количество операций, содержащихся в поднаборах S_q соизмеримо с количеством операций в наборе B :

$$\begin{cases} S_q \subseteq B, 1 \leq q \leq M \\ \sum_{i=1}^M |S_q| = O(|B|) \\ S_a \cap S_b = \emptyset, \forall a \neq b \\ M \in \mathbb{N} \end{cases} \quad (6)$$

Предположим также, что каждый поднабор операций S_q , $q = \overline{1, M}$ из B обладает следующими свойствами:

- S_q , $q = \overline{1, M}$ зависит только от одной вычислительной операции b_q из набора B или является независимым.
- S_q , $q = \overline{1, M}$ допускает разбиение на m_q непересекающихся поднаборов P_{qi} , $i = \overline{1, m_q}$ одинаковой мощности, где m_q достаточно большое число, т. е.:

$$\begin{cases} \bigcup_{i=1, m} P_{qi} = S_q, \forall q = \overline{1, M} \\ P_{qi} \cap P_{qj} = \emptyset, \forall i \neq j, \forall q = \overline{1, M} \\ |P_{qi}| = |P_{qj}| = |P_q|, \forall i, j, \forall q = \overline{1, M} \\ m_q \gg 1, m_q \in \mathbb{N}, \forall q = \overline{1, M} \end{cases} \quad (7)$$

- Любые пары P_{qi} и P_{qj} , $\forall i \neq j$ из S_q , $q = \overline{1, M}$ независимы друг от друга.

Можно показать, что при сделанных предположениях задача минимизации функционала (5) сводится к решению M оптимизационных задач (для каждого отдельного S_q , $q = \overline{1, M}$):

$$F(D_q, G(S_q, E_q), U, Y_q) \xrightarrow{Y_q} \min, \quad (8)$$

где D_q – данные, необходимые для выполнения операций из набора S_q , E_q – связи между операциями, Y_q – разбиение отдельного

Мацкевич Вадим Владимирович, магистрант кафедры информационных систем управления Белорусского государственного университета.

220050, г. Минск, пр. Независимости, 4.

Физика, математика, информатика