

УДК 004.032.24, 004.032.26

ИСПОЛЬЗОВАНИЕ РАЗЛИЧНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ОБЛАЧНОСТИ ПО СПУТНИКОВЫМ ДАННЫМ MODIS НА ОСНОВЕ ВЕРОЯТНОСТНОЙ НЕЙРОННОЙ СЕТИ

Скороходов Алексей Викторович,

кандидат технических наук, научный сотрудник группы атмосферной акустики Института оптики атмосферы им. В.Е. Зуева СО РАН, Россия, 634055, г. Томск, пл. Академика Зуева, 1. E-mail: vazime@yandex.ru

Аксёнов Сергей Владимирович,

кандидат технических наук, доцент кафедры оптимизации систем управления Института кибернетики Национального исследовательского Томского политехнического университета, Россия, 634034, г. Томск, пр. Ленина, 30; доцент кафедры автоматизации обработки информации Томского государственного университета систем управления и радиоэлектроники, Россия, 634034, г. Томск, пр. Ленина, 40; инженер научно-образовательного центра компьютерных наук и технологий Национального исследовательского Томского государственного университета, Россия, 634034, г. Томск, пр. Ленина, 36. E-mail: axyonov@tpu.ru

Аксёнов Андрей Владимирович,

аспирант кафедры оптимизации систем управления Института кибернетики Национального исследовательского Томского политехнического университета, Россия, 634034, г. Томск, пр. Ленина, 30. E-mail: image@tpu.ru

Лайком Дмитрий Николаевич,

аспирант кафедры оптимизации систем управления Института кибернетики Национального исследовательского Томского политехнического университета, Россия, 634034, г. Томск, пр. Ленина, 30. E-mail: wedun@tpu.ru

Актуальность исследования обусловлена необходимостью разработки алгоритмов и программных средств для классификации типов облачности по спутниковым снимкам однослойной облачности, полученных с помощью спектрорадиометра MODIS, используемого аппаратами дистанционного зондирования Земли Terra и Aqua, и высокопроизводительных систем.

Цель работы: эффективный и быстрый анализ спутниковых снимков размерами 5416×8120 пикселей однослойной облачности, полученных с помощью спектрорадиометра MODIS с помощью вероятностной нейронной сети, для классификации облачности по 27 типам.

Методы исследования. Для достижения цели применяются методы распараллеливания вычислений, нейросетевые вычисления, методы компьютерного зрения и анализа текстур, алгоритмы классификации, технологии высокопроизводительных вычислений для многоядерных систем с общей памятью (OpenMP), графических процессоров (CUDA) и распределенных систем (MPI).

Результаты. Процедура классификации, основанная на вероятностной нейросетевой модели, сравнивает фрагменты снимка с эталонами, полученными ранее и классифицированными экспертами. Для корректного анализа фрагмента изображения его требуется сравнить с тысячами эталонов, что приводит к существенным временным затратам. Характер вычислений позволяет разбить входной снимок на несколько более мелких и обработать их независимо на разных вычислительных устройствах или устройствах, поддерживающих одновременное выполнение разных задач. В работе сравнивается производительность трех подходов к распараллеливанию вычислений: на основе многопоточных вычислений, выполняемых многоядерными центральными процессорами, многопоточных вычислений внутри мультипроцессоров графических ускорителей и распределенной обработки на базе кластера. Для последнего случая, в котором вычислительные задачи разделяются уже не между потоками, а процессами с индивидуальными адресными пространствами, было предложено два подхода к решению проблемы: на основе разделения задач и разделения данных. Для каждого варианта параллельной обработки приведены детальное описание и оценка его производительности при анализе полноразмерного снимка MODIS. Показано, что использование распределенной обработки и/или графических ускорителей при решении задачи классификации однослойной облачности, основанной на вероятностной нейронной сети, имеет существенное преимущество по производительности не только по сравнению с классическим алгоритмом, но и его модификацией для многоядерных центральных процессоров.

Ключевые слова:

Классификация облачности, параллельные вычисления, обработка спутниковых данных, MODIS, суперкомпьютер, графический процессор, нейронная сеть.

Введение

В настоящее время результаты дистанционного зондирования Земли из космоса занимают важное место в различных тематических исследованиях, системах мониторинга окружающей среды, прогноза погоды, обеспечения безопасности полетов воздушных судов и в других сферах человеческой деятельности. Данные, поступающие с бортов космических аппаратов, нуждаются в первичной и тематической обработке для их дальнейшего использования в прикладных целях. Начиная с запуска первых спутниковых систем и до настоящего времени постоянно разрабатываются и совершенствуются не только сами сенсоры, но и алгоритмы обработки данных, получаемых с их помощью. В условиях постоянно увеличивающихся объемов спутниковой информации обязательным требованием к алгоритмам ее обработки и их программным реализациям является не только автоматизация, но и высокая производительность [1].

На данный момент существует множество алгоритмов тематической обработки спутниковых данных, предназначенных для интерпретации спектральной информации различных объектов, классификации и распознавания образов, оперативного мониторинга окружающей среды, геологоразведки и многого другого. Однако большинство этих алгоритмов непригодно для оперативной обработки спутниковых данных, несмотря на их эффективность при решении каких-либо тематических задач, процесс решения которых не был ограничен по времени. К наиболее ярким примерам таких алгоритмов относятся методы классификации и распознавания объектов на основе информации о текстуре их изображений на спутниковых снимках [2–4]. В настоящее время существует большое число эффективных алгоритмов классификации и распознавания образов на изображениях, в частности лиц людей, печатных или рукописных символов, номерных знаков автомобилей [5–7]. Однако при обработке полноразмерных спутниковых изображений, которые могут иметь разрешение 8000×5000 пикселей и даже больше, а также при решении задач множественной классификации (один объект имеет несколько десятков разновидностей) эти алгоритмы оказываются неэффективными. Время классификации может занимать несколько десятков часов, что не удовлетворяет требованию высокой производительности.

Начиная с середины 90-х годов для оперативной обработки данных дистанционного зондирования Земли из космоса начинают применяться системы параллельной обработки информации – суперкомпьютеры. Одними из главных пользователей, применяющих суперкомпьютер для оперативной обработки спутниковой информации, являются специалисты из Goddard Space Flight Center и Meteo France [8]. С середины 2000-х годов начинается массовое производство многоядерных процессоров, что открывает пользователям персональных компьютеров возможность использования па-

раллельных вычислений для своих целей [9]. Развитие графических процессоров общего назначения в конце 2000-х годов и технологий их использования для решения вычислительных задач, не связанных непосредственно с обработкой изображений, позволило достичь, а затем и значительно превзойти вычислительную производительность многоядерных центральных процессоров [10]. При этом и сами суперкомпьютеры начинают создаваться с использованием обычных и специализированных видеопроцессоров.

Бурное развитие вычислительной техники в последнее десятилетие, однако же, не повлекло за собой создание большого числа новых методов параллельной обработки данных дистанционного зондирования Земли из космоса или адаптацию к новым вычислительным аппаратным средствам уже существующих. Одной из причин является невозможность распараллеливания некоторых алгоритмов в силу своих особенностей [11]. Другая причина – затруднения самих разработчиков, связанные с выбором технологий параллельной обработки данных OpenMP, Nvidia CUDA, MPI и других, а также адаптацией алгоритмов под них. В то же время существует целый ряд алгоритмов, например, основанных на искусственных нейронных сетях, предполагающих параллельную реализацию. Поэтому целью данной работы является модификация разработанного нами ранее алгоритма классификации однослойной облачности и облаков вертикального развития, по данным MODIS, с пространственным разрешением 250 м для его реализации с помощью распределения вычислений на разных аппаратных платформах и определения возможности его использования в системах оперативного мониторинга атмосферы и подстилающей поверхности [12].

Алгоритм

В качестве классификатора для решения задачи автоматической классификации облачности используется вероятностная нейронная сеть [13], структура которой показана на рис. 1. Сети такого типа содержат, как правило, три вычислительных слоя. На вход нейронной сети предъявляются нормализованные значения текстурных признаков классифицируемого фрагмента спутникового снимка облачности $x = (x_1, x_2, x_3, \dots, x_8)^T$. Первый вычислительный слой содержит наборы значений всех обучающих образцов для каждой из разновидностей облаков. По сути, каждый нейрон этого слоя соответствует конкретному обучающему образцу определенного типа облачности. При этом объем обучающей выборки составляет 5400 образцов, по 200 фрагментов снимков на каждый класс (тип облачности). Откликами нейронов второго слоя являются рассчитанные значения уровней их активности. Второй слой состоит из нейронов-сумматоров, накапливающих отклики нейронов предыдущего слоя, принадлежащих одному классу. Количество нейронов этого слоя равняется числу

распознаваемых классов – 27. Третий слой содержит один единственный нейрон, определяющий класс с максимальной активностью, которому и соответствует классифицируемый образец.

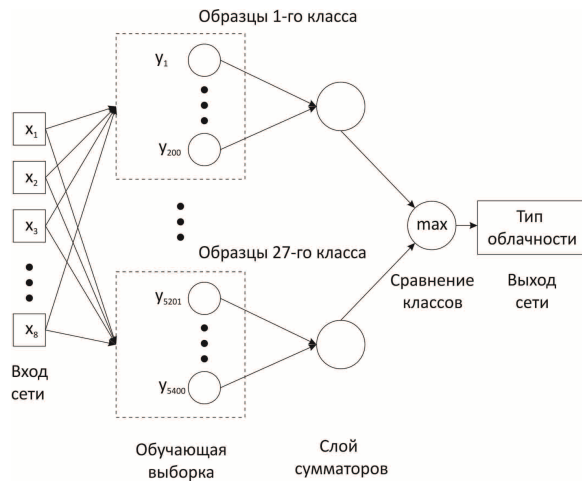


Рис. 1. Архитектура вероятностной нейронной сети для задачи классификации типов облачности

Fig. 1. Probabilistic neural network architecture for cloud type classification

Алгоритм функционирования сетей такого типа состоит из следующих этапов:

1. На вход сети предъявляется набор нормированных значений текстурных признаков классифицируемого фрагмента изображения облачности x .
2. Для каждого нейрона первого слоя рассчитывается уровень его активности:

$$f_i(x) = \exp \frac{y_i x - 1}{\sigma^2},$$

где y_i – набор нормированных значений i -го обучающего образца, $i=1,2,\dots,5400$; σ – параметр сглаживания функции активации.

3. Вычисляются суммарные отклики нейронов каждого класса:

$$F_j = \sum_{i=1}^{m_j} f_i(x),$$

где $m_j=200$ – число обучающих образцов j -го класса, $j=1,2,\dots,27$.

4. Для каждого класса вычисляется его уровень активности, определяемый функцией Парзена [13]:

$$P_j = \frac{F_j}{m_j (2\pi\sigma^2)^{0,5d}},$$

где $d=8$ – число классификационных характеристик.

5. Определяется идентификатор класса с наибольшим значением P_j . Поскольку в нашей задаче m_j одинаковы для всех классов, то расчет значений функции Парзена можно исключить и выявить номер класса с максимальным значением F_j . Таким образом, классифицируемый образец приписывается именно этому классу.

Полностью алгоритм классификации облачности по спутниковым снимкам MODIS можно представить в виде схемы, показанной на рис. 2. Анализируя эту схему и алгоритм функционирования сети, можно выделить большое число циклов:

- сканирование изображения скользящим окном (внешний цикл);
- расчет текстурных признаков, включая вспомогательные вычисления по формированию матриц смежности [14], векторов различия [15], гистограмм сумм и разностей [16] соседних пикселей исследуемого фрагмента изображения;
- нормировка значений текстурных признаков;
- вычисление f_i ;
- вычисление F_j ;



Рис. 2. Схема алгоритма классификации облачности

Fig. 2. Scheme of cloud classification algorithm

- определение номера класса с наибольшим значением F_j .

Таким образом, можно сделать вывод, что использование методов параллельной обработки информации для решения задачи классификации облаков по спутниковым снимкам является целесообразным из-за большого количества вычислительных циклов, как внешних, так и внутренних.

Реализация алгоритма с использованием OpenMP

Наличие вычислительных систем, имеющих один или несколько многоядерных процессоров, ядра которых зачастую поддерживают одновременную многопоточность, позволяет использовать технологию OpenMP для повышения производительности программных средств [17]. Основная суть этой технологии заключается в распределении главным исполняемым потоком элементов задачи между подчиненными потоками, которые работают параллельно на различных ядрах одного или нескольких процессоров. При этом задача и данные для нее описываются с использованием специальных директив препроцессора, имеющих схожий синтаксис в разных языках программирования.

Примером такой задачи является вычисление каких-либо значений в цикле. Так, общее число итераций цикла может быть поделено поровну между различными потоками, выполняемыми параллельно. Таким образом, после выполнения элементов задачи разными потоками общий результат будет доступен основному потоку. При этом время выполнения всех итераций цикла значительно сокращается. Однако OpenMP имеет ряд серьезных ограничений, особенно в случае с вложенными циклами и наличием большого числа зависимых переменных, результат вычисления которых зависит от множества предыдущих операций [18]. Поэтому OpenMP главным образом применяется для распараллеливания внутренних циклов с малым числом зависимых переменных, или не имеющих их вовсе.

Как уже было сказано ранее, предложенный алгоритм классификации облачности предполагает множество внутренних циклов, которые можно распараллелить с использованием технологии OpenMP (директива `#pragma omp parallel for`), внутри внешнего цикла (сканирование изображения скользящим окном):

- формирование матриц смежности яркости соседних пикселей исследуемого фрагмента изображения;
- получение векторов различия яркости соседних пикселей;
- вычисление гистограмм сумм и разностей яркостей соседних пикселей;
- расчет значений самих текстурных признаков;
- нормировка полученных значений текстурных признаков;
- вычисление f_i ;
- вычисление F_j .

Следует отметить, что принципиально структура алгоритма классификации облачности с исполь-

зованием технологии OpenMP не отличается от приведенной схемы на рис. 2, поскольку параллельные блоки идут последовательно в главном потоке. Производительность алгоритма классификации облачности была апробирована на полномасштабных спутниковых снимках MODIS с пространственным разрешением 250 м, имеющих размеры 8120×5416 пикселей. При этом использовался двухъядерный процессор Intel Core i3 550 с поддержкой Multi-Threading, имеющий тактовую частоту 3200 МГц. Максимальное количество одновременно исполняемых потоков при применении данного процессора – 4. При использовании базового алгоритма время классификации облачности составило более 28 часов. Применение технологии OpenMP позволило сократить время выполнения на 20 % – до 23 часов.

Полученные результаты свидетельствуют о том, что технология OpenMP позволяет немного повысить производительность программных средств на стационарных компьютерах, имеющих многоядерные процессоры. Однако для оперативной классификации облачности по спутниковым снимкам такого прироста в производительности явно недостаточно.

Реализация алгоритма с использованием Nvidia CUDA

Рассмотрим другую технологию параллельных вычислений – Nvidia CUDA, основная суть которой заключается в использовании графических процессоров общего назначения (видеокарт) для выполнения «неграфических» вычислений [9]. Современные видеоадаптеры обладают несколькими мультипроцессорами, каждый из которых состоит из большого числа потоковых процессоров (ядер CUDA), способных выполнять примитивные операции, например преобразовывать координаты трехмерных объектов в различных графических приложениях. Таким образом, потоковые процессоры, число которых достигает нескольких тысяч, могут одновременно выполнять однотипные операции, называемые ядрами, для большого объема данных, сгруппированных в нити и блоки [19]. Несмотря на более низкую тактовую частоту мультипроцессоров (около 1000 МГц) в сравнении с центральными процессорами (более 3000 МГц), производительность программ, разработанных на основе технологии Nvidia CUDA, значительно выше. Это достигается за счет большего числа одновременно исполняемых потоков на графическом процессоре.

Анализируя решаемую нами задачу классификации облачности по данным дистанционного зондирования Земли из космоса, можно сделать вывод, что основная ее идея состоит в определении типа облачности, к которому относятся фрагменты спутниковых снимков размером 21×21 пиксель при их смещении на 1 пиксель по вертикали и горизонтали [20]. Таким образом, необходимо проклассифицировать несколько десятков миллионов фрагментов изображения, используя один и тот же

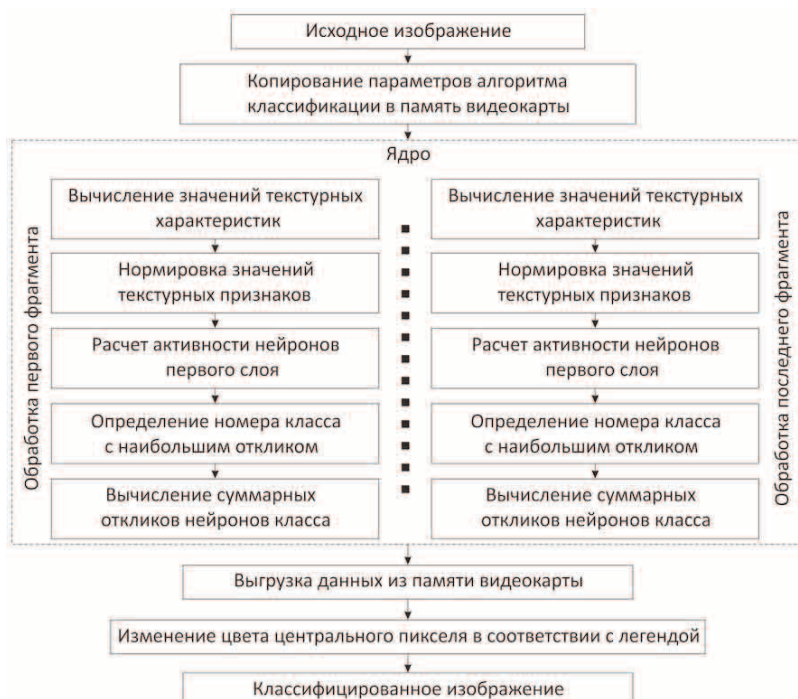


Рис. 3. Схема алгоритма классификации облачности с использованием Nvidia CUDA

Fig. 3. Scheme of cloud classification algorithm with the usage of NVidia CUDA

алгоритм-классификатор (ядро). Следовательно, данная задача идеально ложится на концепцию технологии Nvidia CUDA и может быть эффективно распараллелена [21]. При этом ядром программы является вероятностная нейронная сеть, принимающая решение относительно принадлежности классифицируемого фрагмента изображения к одной из разновидностей облачности на основе рассчитанных значений его текстурных признаков.

Количество одновременно исполняемых потоков зависит как от решаемой задачи, так и от характеристик видеопроцессора. Поэтому нами была предложена универсальная методика инициализации количества нитей в блоке t и размера сетки блоков g_x и g_y в соответствии с параметрами классифицируемого изображения (ширины w и высоты h) и видеокарты (количество нитей на блок m). Основная суть методики заключается в следующем:

1. Если $w \leq m$, то $t = m$, $g_x = 1$ и $g_y = h$.
2. Если $w > m$, то $t = m$.
 - 2.1. Пока $w \bmod t \neq 0$ $t = t - 1$.
 - 2.2. $g_x = w \text{ div } t$ и $g_y = h$.

Для полноразмерных спутниковых снимков MODIS с пространственным разрешением 250 м, имеющих размеры $w = 5146$ и $h = 8120$, и при использовании видеокарты GeForce GTX 780 с $m = 1024$ количество нитей в блоке $t = 284$, а размерность сетки $g_x = 19$ и $g_y = 995$. При этом в соответствии с CUDA GPU Occupancy Calculator [22] значение $t = 284$ является близким к оптимальному, что предполагает высокую производительность алгоритма классификации.

Структура же алгоритма классификации облачности претерпевает существенные измене-

ния в связи с использованием технологии Nvidia CUDA. Схема выполнения программы, основанной на применении данной технологии параллельных вычислений, показана на рис. 3. Как видно, основным отличием базового алгоритма от адаптированного под вычисления на графических процессорах является отсутствие внешнего цикла при классификации. Изображение и обучающая выборка полностью загружаются в память видеокарты, где происходит параллельная обработка нескольких тысяч фрагментов спутникового снимка. Поскольку общее количество таких фрагментов велико, то видеокартой производится внутреннее планирование вычислений. После обработки последней группы фрагментов результаты классификации выгружаются из памяти видеокарты, и остается только перекрасить изображение в цвета разновидностей облачности, заданных легендой.

Модифицированный с помощью технологии Nvidia CUDA алгоритм классификации облачности был апробирован с использованием видеокарты GeForce GTX 780, имеющей 12 мультипроцессоров со 192 потоковыми процессорами каждый. Тактовая частота процессоров у данной модели видеокарты составляет 863 МГц. Время, затраченное на классификацию облаков на полноразмерном спутниковом снимке MODIS, составило немногим больше 30 мин. Таким образом, использование персональных компьютеров, оснащенных современными видеокартами, позволяет достаточно оперативно решать сложные вычислительные задачи, связанные с обработкой данных дистанционного зондирования Земли из космоса.

Реализация алгоритма с использованием MPI

Использование технологии MPI позволяет представить задачу классификации как серию независимых подзадач классификации, выполняемых на нескольких процессах [23]. В отличие от технологий, рассмотренных выше, MPI предполагает физическое разделение данных, т. к. каждый процесс обладает своей собственной памятью. Алгоритм классификации одного изображения допускает существование двух подходов к распараллеливанию на основе MPI: на основе разделения входного массива на фрагменты, которые обрабатываются независимо каждым процессом, и специализации каждого вычислительного процесса на определенном или определенных классах. В последнем случае каждый процесс получает данные целиком, однако рассчитывает отклики сети только к части классов. Рассмотрим оба подхода подробнее.

1. Выполнение классификации независимых фрагментов

Один из процессов производит считывание графического файла. Входной массив пикселей необходимо разделить на n пересекающихся областей, где n – количество используемых процессов (или размер коммуникатора MPI_COMM_WORLD). Необходимость дублирования части данных на границах фрагментов обусловлена алгоритмом вычисления текстурных характеристик, который использует набор смежных пикселей для расчета текстурных характеристик области 21×21 . На рис. 4 показан пример наложения данных при их распределении для соседних вычислительных потоков.



Рис. 4. Распределение строк изображения по процессам

Fig. 4. Distribution of image rows between processes

Каждый из процессов вызывает представленную в разделе «Алгоритм» процедуру классификации. Фактически изменение производительности с помощью этого метода получается за счет того, что каждый процесс работает индивидуально с фрагментом снимка, который, очевидно, гораздо меньше целого изображения.

После того как каждый процесс получил массив, хранящий идентификаторы наиболее вероятных классов для своего региона, один из процессов объявляется корневым и получает номера классов от всех процессов в коммуникаторе. Такое объединение достаточно просто получается вызовом MPI_Gather, аргументами которой являются все

массивы с номерами классов во всех потоках и массив корневого процесса, в который записывается результат классификации. Работа алгоритма может быть проиллюстрирована блок-схемой на рис. 5.

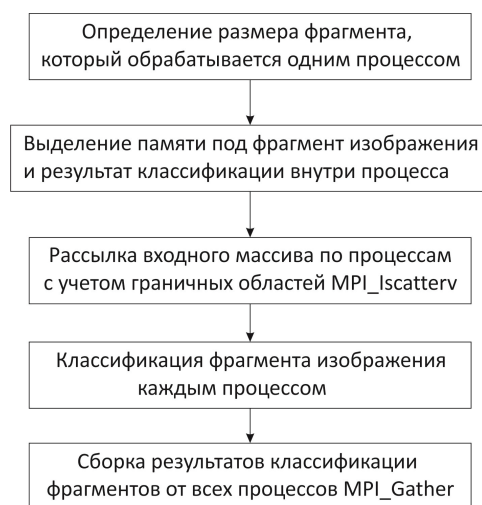


Рис. 5. Блок-схема первого подхода по распределению вычислений с помощью технологии MPI

Fig. 5. Block-scheme of the first approach on processing distribution with MPI

2. Распределение классов между процессами

Другим возможным вариантом расчета может быть связывание вычислительных процессов не с распределением данных, а с распределением задач, например первый процесс рассчитывает суммарные отклики первого класса, второй процесс – второго класса и т. д. Если процессов достаточно много, то можно связать процессы с первым уровнем модели (уровнем образцов), а не со вторым – уровнем классов. Если процессов меньше, чем классов, то один процесс может быть связан с несколькими классами. В любом случае для каждого окна получается вектор откликов на каждый класс. Если процесс не ассоциирован с каким-то классом (с этим классом связан другой процесс), то в этом векторе на позиции этого класса стоит нуль. Так как каждый процесс обрабатывает всё изображение целиком, то перед запуском процедуры анализа процесс, считавший массив пикселей из файлов, осуществляет его передачу всем процессам целиком, используя не блокирующее широковещание MPI_Ibcast.

После расчета степени принадлежности для каждой области формируется массив, размерностью $27 \times S$, где S – размер изображения, содержащий первые 27 значений – суммарные отклики на первый класс, вторые 27 значений – суммарные отклики на второй класс для первой области и т. д.

Полученные таким образом матрицы вероятностей от всех процессов объединяются с помощью редукции MPI_Reduce с операцией MPI_MAX в корневом процессе, который дополнительно должен найти наибольший элемент, означающий наибольшую вероятность принадлежности к эталонам

класса в редуцированном массиве. Расположение (индекс) наибольшего элемента совпадает с идентификатором искомого класса. Подобный подход проиллюстрирован на рис. 6.

В таблице приведены данные по производительности алгоритмов на суперкомпьютерном кластере НИ ТПУ «СКИФ-ТПУ1», использующем узлы с процессорами Intel Nehalem 4C X5570 2.93 GHz. Тестирование показало близкую производительность подходов по распределению вычислений с помощью технологии MPI, с некоторым преимуществом первого подхода, применяющего распределение вычислений.

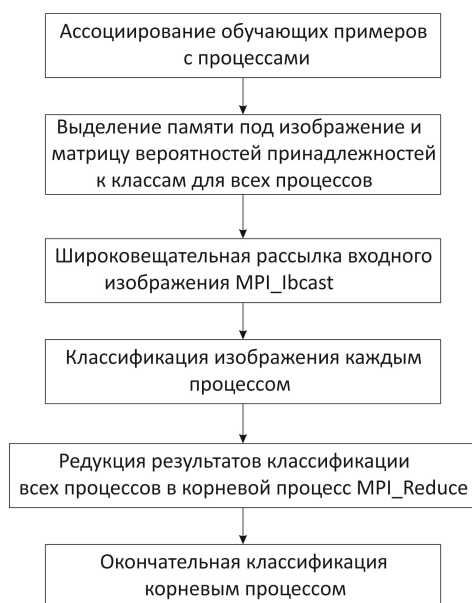


Рис. 6. Блок-схема второго подхода по распределению вычислений с помощью технологии MPI

Fig. 6. Block-scheme of the second approach on processing distribution with MPI

Таблица. Производительность алгоритмов MPI

Table. MPI algorithm performance

Число узлов Number of nodes	Распределение данных Data distribution	Распределение задач Task distribution
2	12 ч (h) 25 мин (min)	12 ч (h) 40 мин (min)
4	6 ч (h) 14 мин (min)	6 ч (h) 37 мин (min)
8	3 ч (h) 11 мин (min)	3 ч (h) 29 мин (min)
16	1 ч (h) 18 мин (min)	1 ч (h) 46 мин (min)

Оба подхода первоначально используют рассылку данных, которая длится около 8–9 с. Сборка данных для первого подхода отнимает около 7–8 с, однако редукция данных второго подхода берет около 3,3 мин. Разница заключается также и в ресурсах, затрачиваемых программистом на программную реализацию подходов. В первом случае берется исходная (непараллельная) программа и запускается на каждом используемом вычислительном узле. Второй подход требует от программиста использовать массив в каждом процессе,

хранящий суммарные отклики сети, которые были вычислены внутри процесса, а затем с помощью групповой коммуникации объединять эти результаты. Эти результаты впоследствии должны быть редуцированы операцией максимум для определения самого вероятного класса для каждой области 21×21 внутри корневого процесса, указанного функцией MPI_Reduce.

Заключение

В результате проделанной работы были предложены три модификации алгоритма классификации однослойной облачности и облаков вертикального развития по данным MODIS с пространственным разрешением 250 м на основе использования различных технологий параллельных вычислений. Было установлено, что применение технологии OpenMP не позволяет решать оперативно вычислительно трудоемкие задачи, связанные с обработкой спутниковых данных. Причиной тому служит незначительный прирост производительности программных средств, разработанных при помощи данной технологии.

Технология Nvidia CUDA позволяет на порядок увеличить быстродействие разработанных с ее помощью алгоритмов и их программных реализаций. Следует отметить, что время классификации полноформатного спутникового снимка MODIS было существенно ниже при применении графического процессора, даже ниже, чем при использовании кластера «СКИФ-ТПУ1». Однако использование данной технологии ограничено теми случаями, когда требуется обработать большое количество одинаковых данных достаточно простым алгоритмом.

Использование технологии MPI для распараллеливания алгоритмов предоставляет более гибкие возможности, чем Nvidia CUDA, поскольку позволяет работать с распределением не только данных, но и задач. При этом первый вариант применения технологии MPI для параллельной классификации спутниковых снимков MODIS оказался более производительным, что видно из таблицы.

В результате сравнения трех технологий параллельной обработки данных можно сделать вывод, что оперативная обработка спутниковых данных возможна даже на персональных компьютерах при использовании графических процессоров общего назначения. Для крупных же вычислительных центров или при наличии нескольких вычислительных узлов наиболее эффективным является комплексное использование технологии MPI и Nvidia CUDA, поскольку возможности распараллеливания алгоритмов в этом случае гораздо шире.

Работа выполнена при частичной финансовой поддержке РФФИ в рамках научных проектов № 14-07-31018 мол_а и № 14-07-31090 мол_а. Работа выполнена в рамках программы по повышению конкурентоспособности Национального исследовательского Томского политехнического университета, проект ВИУ ИК 118.

СПИСОК ЛИТЕРАТУРЫ

1. Потапов В.П., Попов С.Е. Высокопроизводительные алгоритмы радиометрической калибровки и преобразования порядка следования данных отдельных каналов сенсора EO-1 Hyperion // Современные проблемы дистанционного зондирования Земли из космоса. – 2014. – Т. 11, – № 4. – С. 286–289.
2. Bankert R.L. Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network // J. Appl. Meteor. – 1994. – V. 33. – P. 909–918.
3. A study of cloud classification with neural networks using spectral and texture features / B. Tian, M.A. Shaikh, M.R. Azimi-sadjadi, T.H. Vonder Haar, D.L. Reinke // IEEE Trans. Neural Networks. – 1999. – V. 10. – P. 138–151.
4. Zhuo W., Cao Z, Xiao Y. Cloud classification of ground-based images using texture-structure features // J. Atmos. Oceanic Technol. – 2014. – V. 31. – P. 79–92.
5. Viola P., Jones M.J. Robust real-time face detection // International Journal of Computer Vision. – 2004. – V. 57 (2). – P. 137–154.
6. Neumann L., Matas J. A method for text localization and recognition in real-world images // Proc. of the 10th Asian Conf. on Computer Vision. – Queenstown, New Zealand, 2010. – P. 770–783.
7. Болотова Ю.А., Спицын В.Г., Рудометкина М.Н. Распознавание автомобильных номеров на основе метода связанных компонент и иерархической временной сети // Компьютерная оптика. – 2015. – Т. 39. – № 2. – С. 275–280.
8. Савин Г.И., Телегин П.Н., Шабанов Б.М. Кластеры Беовульф // Известия вузов. Электроника. – 2004. – № 1. – С. 7–12.
9. Параллельные вычисления CUDA. URL: <http://www.nvidia.ru/object/cuda-parallel-computing-ru.html> (дата обращения: 17.09.2015).
10. Полетаев С.А. Параллельные вычисления на графических процессорах. Конструирование и оптимизация параллельных программ. – Новосибирск: Изд-во ИСИ СО РАН, 2008. – 332 с.
11. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с.
12. Astafurov V.G., Laikom D.N., Skorokhodov A.V. Software system for cloud classification according to MODIS satellite images // Proc. SPIE 9292 of 20th International Symposium on Atmospheric and Ocean Optics: Atmospheric Physics. URL: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1984426> (дата обращения: 11.09.2015).
13. Specht D.F. Probabilistic neural networks // Neural Networks. – 1990. – V. 3. – P. 109–118.
14. Haralick R.M. A texture-context feature extraction algorithm for remotely sensed imagery // Proc. IEEE Decision and Control Conf. – Clearwater Beach, USA, 1971. – P. 650–657.
15. Weszka J., Dyer C., Rosenfeld A. A comparative study of texture measures for terrain classification // IEEE Trans. Syst. Man. And Cybern. – 1976. – V. SMC-6. – № 4. – P. 269–285.
16. Unser M. Sum and difference histograms for texture classification // IEEE Trans. Pattern Anal. Machine Intell. – 1986. – V. PAMI-8. – P. 118–125.
17. The OpenMP® API specification for parallel programming. URL: <http://openmp.org/wp/> (дата обращения: 17.09.2015).
18. OpenMP shared memory parallel programming / M.S. Mueller, B.M. Chapman, de B.R. Supinski, A.D. Malony, M. Voss. – Berlin: Springer, 2008. – 448 p.
19. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – М.: ДМК «Пресс», 2010. – 232 с.
20. Астафуров В.Г., Скороходов А.В. Применение нейросетевых технологий для классификации облачности по текстуре снимков MODIS высокого разрешения // Исследование Земли из космоса. – 2014. – № 5. – С. 39–49.
21. Зацепин П.М., Гефке Д.А. Применение технологии Nvidia CUDA для обучения и декодирования скрытых Марковских моделей // Известия Томского политехнического университета. – 2014. – Т. 324. – № 5. – С. 115–121.
22. CUDA GPU Occupancy Calculator. URL: [http:// developer.download.nvidia.com/compute/cuda/CUDA_Occupancy_calculator.xls](http://developer.download.nvidia.com/compute/cuda/CUDA_Occupancy_calculator.xls) (дата обращения: 10.11.2015).
23. MPI Standard official page. URL: <http://mpi-forum.org> (дата обращения: 10.11.2015).

Поступила 12.11.2015 г.

UDC 004.032.24, 004.032.26

USING DIFFERENT COMPUTING SYSTEMS TO SOLVE THE AUTOMATIC CLOUD CLASSIFICATION PROBLEM ACCORDING TO MODIS SATELLITE DATA BY PROBABILISTIC NEURAL NETWORK

Aleksey V. Skorokhodov,

V.E. Zuev Institute of Atmospheric Optics SB RAS, 1, Academician Zuev Avenue, Tomsk, 634055, Russia. E-mail: vazime@yandex.ru

Sergey V. Aksenov,

National Research Tomsk Polytechnic University, 30, Lenin Avenue, Tomsk, 634034, Russia; Tomsk State University of Control Systems and Radio Electronics, 40, Lenin Avenue, Tomsk, 634050, Russia. E-mail: axyonov@tpu.ru

Andrey V. Aksenov,

National Research Tomsk Polytechnic University, 30, Lenin Avenue, Tomsk, 634034, Russia. E-mail: image@tpu.ru

Dmitriy N. Laykom,

National Research Tomsk Polytechnic University, 30, Lenin Avenue, Tomsk, 634034, Russia. E-mail: wedun@tpu.ru

The relevance of the research is caused by the necessity to develop algorithms and software to classify the cloud types based on single-layer cloud on the satellite images received from MODIS spectral radiometer used in Terra and Aqua remote sensing Earth satellites with the usage of high-performance systems.

The main aim of the study: effective and fast analysis of 5416–8120 single-layer cloud full scale satellite images received from MODIS spectral radiometer with the help of the probabilistic neural network detecting 27 cloud types.

The methods used in the study. To carry out the task the authors used the methods of paralleling the processing, neurocomputing, computer vision and texture analysis algorithms, classification algorithms, technologies of high-performance processing for multi-core shared memory systems (OpenMP), graphics processing units (CUDA) and distributed systems (MPI).

The results. The classifying procedure based on probabilistic neural model compares all the fragments from the given image with the patterns from the training set classified by experts. It needs to compare texture features of each fragment with features of some thousands patterns and therefore it leads to significant time costs. The algorithm allows splitting the given input into a set of small images that can be processed independently by some computational devices and devices supporting the processing of simultaneous tasks. The paper compares the performance of three approaches for parallel processing that are multi-thread computation based on multi-core central processing units (CPUs), multi-thread computation based on graphics processing units (GPUs) and distributed processing implemented by computational cluster. The latter uses worksharing between different processes with independent address spaces and the approach includes two methods for speed-up the processing based on data distribution and task sharing. Each approach was described in detail and its performance was estimated by analysis of MODIS' full scale image. It's shown that the usage of distributed processing or/and multi-thread GPU computation for performance of single-layer cloud classification task based on probabilistic neural model has significant performance advantages not only in comparison with the classic sequential algorithm but also with its multi-thread version for many-core CPUs.

Key words:

Cloud classification, parallel computing, satellite imagery, MODIS, supercomputer, graphical processing unit, neural network.

The reported study was funded by RFBR, according to the research projects No. 14–07–31018 MOL_A and No. 14–07–31090 MOL_A. The research was performed as a part of the program to improve the competitiveness of the National Research Tomsk Polytechnic University, project No. VIU IC 118.

REFERENCES

1. Potapov V.P., Popov S.E. Vysokoproizvoditelnye algoritmy radiometricheskoy kalibrovki i preobrazovaniya poryadka sledovaniya dannykh ot delnykh kanalov sensora EO-1 Hyperion [High-performance algorithms for radiometric calibration and conversion of the order of the data of individual channels sensor EO-1 Hyperion]. *Sovremennye problemy distantsionnogo zondirovaniya Zemli iz kosmosa*, 2014, vol. 11, no. 4, pp. 286–289.
2. Bankert R.L. Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *J. Appl. Meteor.*, 1994, vol. 33, pp. 909–918.
3. Tian B., Shaikh M.A., Azimi-sadjadi M.R., Vonder Haar T.H., Reinke D.L. A study of cloud classification with neural networks using spectral and texture features. *IEEE Trans. Neural Networks*, 1999, vol. 10, pp. 138–151.
4. Zhuo W., Cao Z, Xiao Y. Cloud classification of ground-based images using texture-structure features. *J. Atmos. Oceanic Technol.*, 2014, vol. 31, pp. 79–92.
5. Viola P., Jones M.J. Robust real-time face detection. *International Journal of Computer Vision*, 2004, vol. 57 (2), pp. 137–154.
6. Neumann L., Matas J. A method for text localization and recognition in real-world images. *Proc. of the 10th Asian Conf. on Computer Vision*. Queenstown, New Zealand, 2010. pp. 770–783.
7. Bolotova Yu.A., Spitsyn V.G., Rudometkina M.N. Raspoznavanie avtomobilnykh nomerov na osnove metoda svyaznykh komponent i ierarkhicheskoy vremennoy seti [License plate recognition on the basis of the connected components and hierarchical temporal network]. *Computer optics*, 2015, vol. 39, no. 2, pp. 275–280.
8. Savin G.I., Telegin P.N., Shabanov B.M. Klasteriy Beovulf [Beowulf clusters]. *Izvestiya vuzov. Elektronika*, 2004, no. 1, pp. 7–12.

9. *Parallelnye vychisleniya CUDA* [Parallel computing CUDA]. Available at: <http://www.nvidia.ru/object/cuda-parallel-computing-ru.html> (accessed 17 September 2015).
10. Poletaev S.A. *Parallelnye vychisleniya na graficheskikh protsesorakh. Konstruirovaniye i optimizatsiya paralelnykh program* [Parallel computing on graphics processors. Design and optimization of parallel programs]. Novosibirsk, ISI SO RAN Publ., 2008. 332 p.
11. Virt N. *Algoritmy i struktury dannykh* [Algorithms and data structures]. Moscow, Mir Publ., 1989. 360 p.
12. Astafurov V.G., Laikom D.N., Skorokhodov A.V. Software system for cloud classification according to MODIS satellite images. *Proc. SPIE 9292 of 20th International Symposium on Atmospheric and Ocean Optics: Atmospheric Physics*. Available at: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1984426> (accessed 11 September 2015).
13. Specht D.F. Probabilistic neural networks. *Neural Networks*, 1990, vol. 3, pp. 109–118.
14. Haralick R.M. A texture-context feature extraction algorithm for remotely sensed imagery. *Proc. IEEE Decision and Control Conf.* Clearwater Beach, USA, 1971. pp. 650–657.
15. Weszka J., Dyer C., Rosenfeld A. A comparative study of texture measures for terrain classification. *IEEE Trans. Syst. Man. And Cybern.*, 1976, vol. SMC-6, no. 4, pp. 269–285.
16. Unser M. Sum and difference histograms for texture classification. *IEEE Trans. Pattern Anal. Machine Intell.*, 1986, vol. PA-MI-8, pp. 118–125.
17. *The OpenMP® API specification for parallel programming*. Available at: <http://openmp.org/wp/> (accessed 17 September 2015).
18. Mueller M.S., Chapman B.M., de Supinski, B.R., Malony A.D., Voss M. *OpenMP shared memory parallel programming*. Berlin, Springer Publ., 2008. 448 p.
19. Boreskov A.V., Kharlamov A.A. *Osnovy raboty s tekhnologiyey CUDA* [Technology basics CUDA]. Moscow, DMK Press Publ., 2010. 232 p.
20. Astafurov V.G., Skorokhodov A.V. Primeneniye neyrosetevykh tekhnologiy dlya klassifikatsii oblachnosti po teksture snimkov MODIS vysokogo razresheniya [The application of neural network technology for the classification of cloud texture high resolution MODIS images]. *Issledovanie Zemli iz kosmosa*, 2014, no. 5, pp. 39–49.
21. Zatsepina P.M., Gefke D.A. Primeneniye tekhnologii Nvidia CUDA dlya obucheniya i dekodirovaniya skrytykh Markovskikh modeley [Nvidia CUDA application to train and decode the hidden Markov models]. *Bulletin of the Tomsk Polytechnic University*, 2014, vol. 324, no. 5, pp. 115–121.
22. *CUDA GPU Occupancy Calculator*. Available at: http://developer.download.nvidia.com/compute/cuda/CUDA_Occupancy_calculator.xls (accessed 10 November 2015).
23. *MPI Standard official page*. Available at: <http://mpi-forum.org> (accessed 10 November 2015).

Received: 12 November 2015.