УДК 004.04

М.Е. Мансурова, А.С. Шоманов, Б.Н. Тулепбергенов,
С.С. Ихсанов, Е.А. Дадыкина

*Казахский национальный университет им. аль-Фараби, Алматы, Казахстан;
e-mail: mansurova01@mail.ru*

Применение технологии MapReduce Hadoop для кластеризации больших объемов данных*

Целью данной работы является реализация параллельного алгоритма ISODATA для кластеризации гиперспектральных изображений. В данном исследовании параллельный алгоритм кластеризации построен на модели программирования MapReduce. Алгоритм реализован на платформе Hadoop, которая является инфраструктурой с открытым исходным кодом, предназначенной для создания и выполнения распределенных приложений, обрабатывающих большие объемы данных. Результаты исследований сравниваются с результатами, полученными с разными параметрами кластера и MPI модели.
Ключевые слова: MapReduce, Hadoop, алгоритм кластеризации, параллельные вычисления, MPI.

M.E. Mansurova, A.S. Shomanov, B.N. Tulepbergenov, S.S. Ikhsanov, E.A. Dadykina

Application of MapReduce Hadoop technology for clustering large amounts of data

The goal of this study is the implementation of parallel ISODATA algorithm for clustering of hyperspectral images. In this study parallel ISODATA clustering algorithm is based on MapReduce programming model. The algorithm is implemented on the platform Hadoop, which is a framework of open source software designed to create and run distributed applications that process large amounts of data. Research results are compared with the results with different cluster settings and MPI model.
Key words: MapReduce, Hadoop, clustering algorithm, parallel computing, MPI.

М.Е. Мансұрова, А.С. Шоманов, Б.Н. Төлепбергенов, С.С. Ихсанов, Е.А. Дадыкина

Үлкен көлемді деректерді кластерлеу үшін MapReduce Hadoop технологиясын қолдану

Берілген жұмыстың мақсаты гиперспектральды бейнелерді кластерлеу үшін ISODATA паралельді алгоритмін жүзеге асыру. Берілген зерттеуде кластерлеудің паралельді алгоритмі MapReduce программалау моделіне негізделген. Алгоритм үлкен көлемдегі деректерді өңдейтін, үлестірілген қосымшаларды құруға және орындауға арналған, ашық кодты инфрақұрылым болатын Hadoop платформасында жүзеге асырылған. Зерттеу нәтижелері кластердің әртүрлі параметрлерімен алынған және MPI моделінің нәтижелерімен салыстырылады.
Түйін сөздер: MapReduce, Hadoop, кластерлеу алгоритмі, параллелді есептеу, MPI.

*Работа выполнена при поддержке Комитета Науки МОН РК, грант № 0873/ГФ2.

Введение

В настоящее время в области дистанционного зондирования Земли (ДЗЗ) одной из актуальных задач является разработка вычислительно эффективных методов для хранения и обработки больших объемов данных. Часто от приложений по обработке данных ДЗЗ требуется возможность предоставлять ответы на запросы в реальном, или близком к реальному, масштабе времени. Одним из важных объектов ДЗЗ являются гиперспектральные изображения. В таких изображениях собираются данные по десяткам и сотням спектральных диапазонов. Сцены, предоставляемые датчиками ДЗЗ, часто называют «кубами данных», для обозначения их высокой размерности [1]-[2].

Хотя большинство параллельных методов обработки изображений, главным образом, используют однородную по своей природе вычислительную среду, современные тенденции в разработке высокопроизводительных систем для обработки больших объемов данных устремлены к использованию неоднородных вычислительных ресурсов. Эта неоднородность возникает, главным образом, в результате усовершенствования технологических возможностей вычислительных систем и их относительной дешевизны. В этой связи, существование сетей гетерогенных ресурсов с высоким уровнем совокупной производительности привело к широкому использованию распределенных вычислений. Распределенные вычисления представляют собой способ решения трудоемких задач с привлечением большого числа территориально удаленных друг от друга вычислительных ресурсов, а также ресурсов хранения и передачи данных. Особое место среди технологий распределенных вычислений занимают облачные технологии.

В данной работе представлена реализация параллельного алгоритма кластеризации гиперспектральных изображений с использованием технологии распределенных вычислений MapReduce[3]. Результаты, полученные в работе, показывают высокую эффективность применения технологии MapReduce к задачам обработки гиперспектральных изображений.

Алгоритм кластеризации ISODATA

Алгоритм кластеризации ISODATA (Iterative Self-Organizing Data Analysis Techniques) является разновидностью алгоритма кластеризации k-means [1], [4]. Основная идея алгоритма k-means заключается в пошаговом нахождении центров тяжести кластеров путем минимизации показателя качества, определенного как сумма квадратов расстояний всех точек, входящих в кластерную область, до центра кластера.

Имеется набор точек $\{x_1, x_2, \ldots, x_N\}$, состоящий из N элементов. Согласно [4] каждый центр кластера z_j, $j = 1, 2, \ldots, N_c$, локализуется и корректируется посредством приравнивания его выборочному среднему, найденному по соответствующему подмножеству S_j, т.е.

$$z_j = \frac{1}{N_j} \sum_{x \in S_j} x, \quad j = 1, 2, \ldots, N_c.$$

Алгоритм кластеризации ISODATA отличается от алгоритма k-means введением эвристических процедур, которые позволяют регулировать число кластеров: объединять кластеры, разделять один кластер на два [4].

Обзор работ по данной теме

Параллельным алгоритмам кластеризации изображений посвящено большое количество работ. В статьях [5]-[6] для параллельной обработки используется технология MPI. Применение технологии MPI требует задания точного числа узлов, причем в случае отказа какого-либо узла, задача должна быть запущена заново. В отличие от этого технология MapReduce вопросы отказоустойчивости, балансировки нагрузки решаются самой выполняющей системой. Ряд работ посвящен реализации алгоритмов кластеризации с применением технологии MapReduce [7]-[11]. Работа [7] описывает подход для кластеризации web документов, при этом учитывается специфика представления web документов. В работе [8] изложен подход кластеризации большого количества спутниковых снимков. Процесс начинается с определения для каждого пикселя ближайшего кластерного центра, а затем вычисления новых центров кластеров на основе пикселей кластерных областей. Общий метод, который для вычисления новых кластерных центров суммирует значения всех пикселей нового кластера, а затем делит сумму на количество пикселей кластерной области, может привести к переполнению в условиях массовых вычислений. Поэтому принимается стратегия, которая вычисляет новое среднее значение по формуле:

$$m'_k = m_k + [P_i - m_k]/(N_k + 1). \quad (1)$$

В (1) m_k означает текущее среднее значение, N_k – текущее количество пикселей, P_i – значение следующей пиксельной точки, $[P_i - m_k]$ – разницу между P_i и m_k, m'_k – новое среднее значение. В [9] дан обзор основных алгоритмов обработки изображений дистанционного зондирования. При этом описана реализация этих алгоритмов для частного случая представления обрабатываемых данных в формате TIFF. В работе [11] предлагается итерационный алгоритм кластеризации, но в отличие от подхода, представленного в [8], кластеризации подвергаются небольшие случайные подмножества данных, для того, чтобы найти соответствующие центроиды для всего набора данных. Этот процесс повторяется многократно, и в результате выбирается оптимальный набор кандидатов центроидов.

В данной работе использован подход из [8], а также предложен вариант реализации алгоритма кластеризации с использованием комбинаторов ([10], [12]), который позволяет улучшить производительность алгоритма.

Технология MapReduce Hadoop

MapReduce является моделью программирования, которая эффективно поддерживает параллелизм. Программы, написанные в данной модели, автоматически распараллеливаются посредством создания распределенных программ исполнения на каждой отдельной машине кластерной системы. Такие действия как разделение данных, управление особенностями отдельных машин, взаимодействие между узлами берет на себя исполняющая система.

Модель программирования MapReduce в общем случае содержит несколько функций Map (распределитель) и Reduce (редуктор). Входными данными функции Map является множество пар вида ключ/значение, поэтому входные данные предварительно преобразовываются в соответствующие пары ключ/значение. После выполнения функции Map генерируются промежуточные пары ключ/значение, которые представляют

Рисунок 1. Концептуальная модель MapReduce

собой выходные данные функции Map. Промежуточные значения группируются по ключам коллектором, затем данные с одинаковыми ключами передаются соответствующим функциям Reduce. На следующем шаге функция Reduce обрабатывает промежуточные значения, полученные от различных распределителей Map, производя новое множество пар ключ/значение, и получая, таким образом, окончательный результат.

Для решения задач с использованием парадигмы MapReduce была выбрана платформа Hadoop[13], которая является каркасом с открытым исходным кодом, предназначенным для создания и запуска распределенных приложений, обрабатывающих большие объемы данных. Неотъемлемым компонентом платформы Hadoop является распределенная файловая система HDFS (Hadoop distributed file system). В основе распределенной файловой системы лежит архитектура master/slave (главный/подчиненный), в которой основной узел поддерживает информацию о файловом пространстве имен (метаданные, структуры директорий, соответствие между определенными файлами и блоками данных, расположение блоков данных и права доступа), а подчиненные узлы управляют непосредственно самими блоками данных [12]-[13].Основной функцией распределенной файловой системы является разделение данных определенного пользователя на блоки данных и репликация данных блоков данных по локальным дискам узлов кластера (рис. 1).

Параллельный алгоритм кластеризации с применением технологии MapReduce

Основная идея параллельной реализации алгоритма кластеризации ISODATA, основанной на технологии MapReduce, заключается в классификации каждого пикселя до ближайшего кластера в функции Map и расчета новых кластерных центров в функции

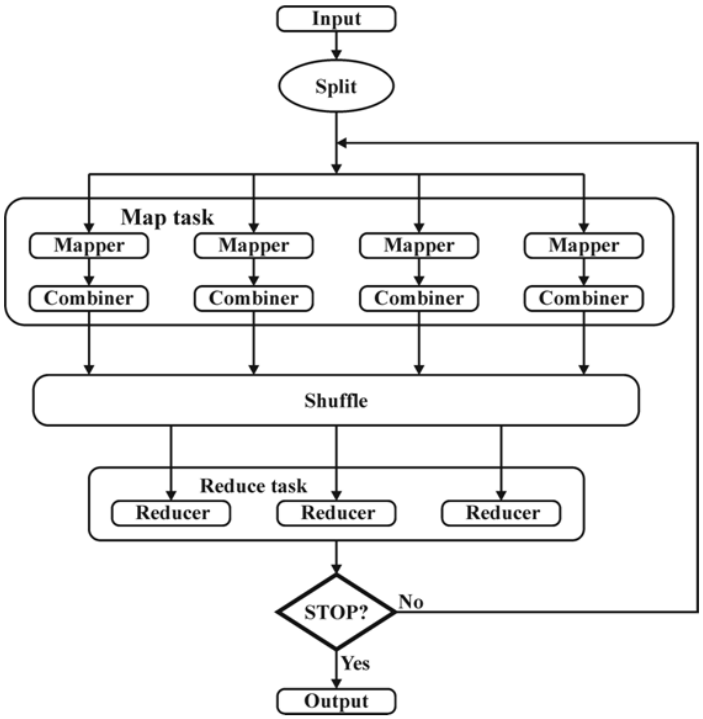

Рисунок 2. Схема итерационного алгоритма кластеризации с комбинатором

Reduce [8]:

Алгоритм 1:

Map:

Входные данные: (key, set<pair <pixel, set <pixel>>>).

Выходные данные: (centroid_id, pixel).

1. Считать значение пикселя и значения текущих центроидов.
2. Для каждого пикселя найти ближайший центроид.
3. Назначить пиксель соответствующему кластеру.

Reduce:

Входные данные: (centroid_id, set <pixel>).

Выходные данные: (new_centroid, pixels).

1. Считать значения пикселей с соответствующим идентификатором центроида.
2. Вычислить новый центроид.

Входные данные для функций Map и Reduce организовываются в виде пар: ключ-значение, *key*/*value*. Ключ *key* во входных данных функции Map не несет полезной информации, но он необходим согласно структуре входных данных; значение *value* каждого пикселя представляет собой строку, состоящую из идентификатора пикселя и значений R, G, B разложения пикселя на цветовые составляющие. Входные данные распределяются между Mapper-ами. Начальное множество кластерных центров или центроидов передается каждому Mapper-у. Каждый центроид задается своим идентификатором

```
public static class Combiner extends Reducer<IntWritable, Text, /*IntWritable*/
IntWritable, Text>
 {
    public void reduce(IntWritable key, Iterable<Text> values,
            Context context) throws IOException, InterruptedException
                {
            String res = "";
            Pixel average = new Pixel(-1,0,0,0);
            int c_ind = 0;
            int count = 0;
           String pixels = "";
          for (Text val: values)
            {
              Pixel pixel =  new Pixel();
              String line = val.toString();
              StringTokenizer t = new StringTokenizer(line);
              pixel.pixel_id = Integer.parseInt(t.nextToken());
              pixel.L = Integer.parseInt(t.nextToken());
              pixel.A = Integer.parseInt(t.nextToken());
              pixel.B = Integer.parseInt(t.nextToken());
              average = average.find_average(average, pixel, count + 1);
              count++;
            }
            context.write(key , new Text(average.toString()));

        }
}
```

Рисунок 3. Функция комбинатора алгоритма кластеризации ISODATA

в качестве ключа *key* и значением центроида в качестве значения *value*. Map функция, сравнивая значение очередного пикселя со значениями кластерных центров, определяет, к какому кластерному центру ближе всего расположен пиксель. Выходными данными Mapper-а будут пара ключ/значение для каждого пикселя, где ключ будет представлять собой идентификатор ближайшего центроида, значение *value* пикселя остается неизменным.

Выходные данные функции Map записываются в файловую систему HDFS. Между стадией Mapper и Reduce промежуточные данные сортируются и тасуются. Входными данными Reduce-ов будут выходные данные Mapper-ов.

Функция Reduce содержит все пары ключ/значение, которые получены от функции Map. Для всех пар ключ/значение, имеющих один и тот же ключ, их значения сохраняются в итераторе. Reduce функция вычисляет средние значения, которые имеют один и тот же ключ.Метод, который суммирует все значения, а затем делит на количество пикселей, может привести к переполнению в условиях массовых вычислений. Поэтому принимается стратегия из [8], которая вычисляет новое среднее значение по формуле (1).

Параллельный алгоритм кластеризации с использованием комбинаторов

Согласно алгоритму 1 выходные данные функции Map записываются на диск, и функция Reduce считывает выходные данные Mapper-ов. Между стадией Мари Reduce данные сортируются и тасуются. С увеличением объема обрабатываемых данных время, затрачиваемое на сортировку и тасование, сильно растет. В данной работе предлагается модификация алгоритма 1 с использованием комбинаторов ([10], [12]), с помощью которых можно уменьшить объемы данных, которые выдает Mapper в качестве выходных данных, а Reducer соответственно считывает. Для данного алгоритма кластеризации

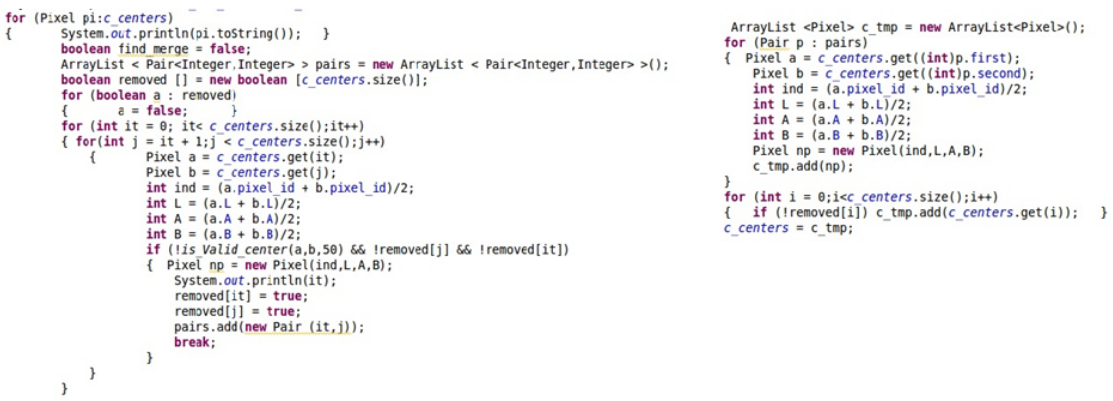

Рисунок 4. Функция слияния кластеров алгоритма кластеризации ISODATA

предлагается создать комбинатор, который считывает выходные данные Mapper-a, и вычисляет новые локальные центроиды для каждого Mapper. Reducer считывает выходные данные Mapper, которые представляют собой идентификаторы новых локальных центроидов и их значения. Затем в Reducer-е вычисляются новые глобальные центроиды.

Алгоритм 2:

Map:

Входные данные: (key, set<pair <pixel, set <pixel>>>).

Выходные данные: (centroid_id, pixel).

1. Считать значения текущих центроидов.
2. Для каждого пикселя найти ближайший центроид.
3. Назначить пиксель соответствующему кластеру.

Combiner:

1. Считать пиксели i-ого центроида (i=1, . . . , k).
2. Вычислить новый локальный i-ый центроид (i=1, . . . , k).

Reduce:

Входные данные: (centroid_id, set <new_local_centriods>).

Выходные данные: (new_centroid, pixels).

1. Считать значения локальных центроидов.
2. Вычислить новый глобальный центроид.

Схема работы итерационного алгоритма кластеризации с комбинатором показана на рис. 2.

Данный алгоритм был реализован платформа Hadoop 2.0 [14]. На рисунке 3 представлен фрагмент программного кода реализации функции комбинатора.

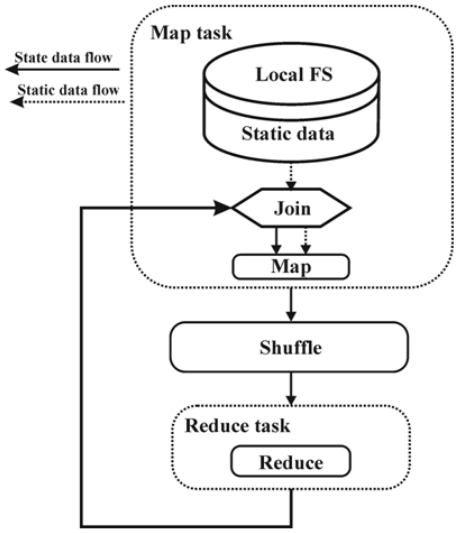

Рисунок 5. Поток данных для итерационного алгоритма кластеризации

Отличительной особенностью алгоритма ISODATA является наличие эвристических процедур, посредством которых можно управлять количеством кластеров: в случае близости значений двух кластерных центров, эти кластеры можно объединять в один; в случае наличия большого количества элементов, относящихся к одному кластеру, данный кластер может быть поделен на два подкластера. На рис. 4 приведен код процедуры слияния кластеров для алгоритма ISODATA, реализованного с использованием MapReduce.

Управление данными

Данные, которые подвергаются кластеризации, не изменяются во время выполнения. В классической схеме итерационного процесса по алгоритму 1 все данные должны проходить стадию тасования (shuffle). В подходе, предлагаемом авторами, согласно [15] предлагается разделять данные на два вида: статические данные и данные состояния. Данные состояния обновляются после каждой итерации. Для алгоритма кластеризации статическими данными являются значения пикселей и данными состояния являются значения локальных центроидов, которые пересчитываются на каждой итерации по алгоритму 2. На рис. 2 показан поток данных для итерационного метода кластеризации. Статические данные считываются Mapper-ом из распределенной файловой системы HDFS только при первой итерации, затем после вычисления статические данные сохраняются в локальной файловой системе. Вычисленные значения локальных центроидов в качестве выходных данных Mapper-а подаются в качестве входных данных Reducer-ам. Таким образом, только данные состояния подвергаются тасованию на стадии shuffle.

Выходные данные Reducer-а считываются Mapper-ом, соединяются со статическими данными, хранящимися в локальной файловой системе, затем выполняется тело функции Map. Поток данных для итерационного алгоритма кластеризации показан на рис. 5.

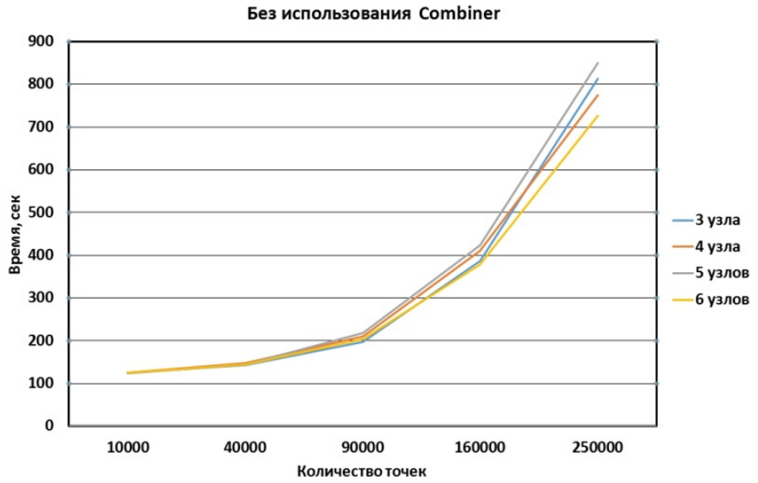

Рисунок 6. Время выполнения кластеризации с MapReduce без комбинатора

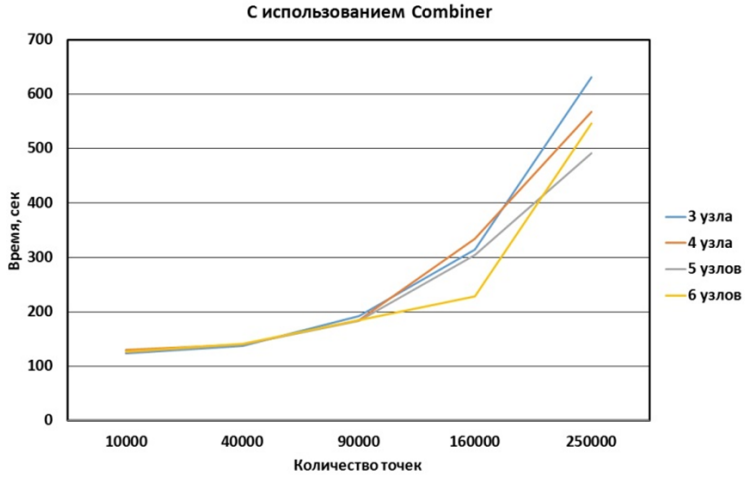

Рисунок 7. Время выполнения кластеризации с MapReduce с комбинатором

Реализация алгоритма кластеризации и анализ результатов

Для тестирования разработанного приложения была настроена инфраструктура, которая включала в себя 6 компьютеров Core i-5 с объемом оперативной памяти 8Gb RAM, 2 сервера HP Blade с 4 Core Intel Xeon процессорами, коммутатор с подключением к гигабитной сети.

На всех компьютерах была установлена 64-битная операционная система Ubuntu 12.10 и платформа Hadoop 2.0. Тестирование проводилось на различных объемах входных данных: рассматривались объемы изображений в 100×100, 200×200, 300×300, 400×400, 500×500 пикселей. Для сравнения результатов проводилось тестирование без использования комбинатора и с использованием комбинатора.

Результаты выполнения кластеризации на платформе Hadoop показаны на рис. 6-10.

Как видно на рис. 8, включение функции комбинатора улучшает время выполнения алгоритма. Причем с увеличением объема обрабатываемой информации заметнее выражается выигрыш во времени.

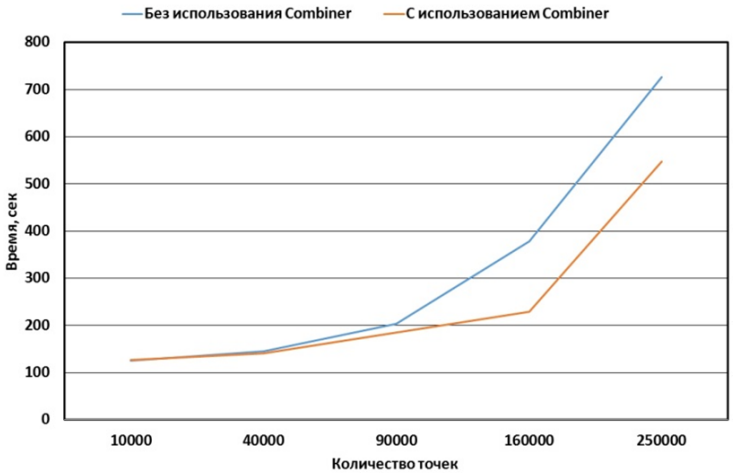

Рисунок 8. Сравнение времени выполнения кластеризации с MapReduce без комбинатора и MapReduce с комбинатором

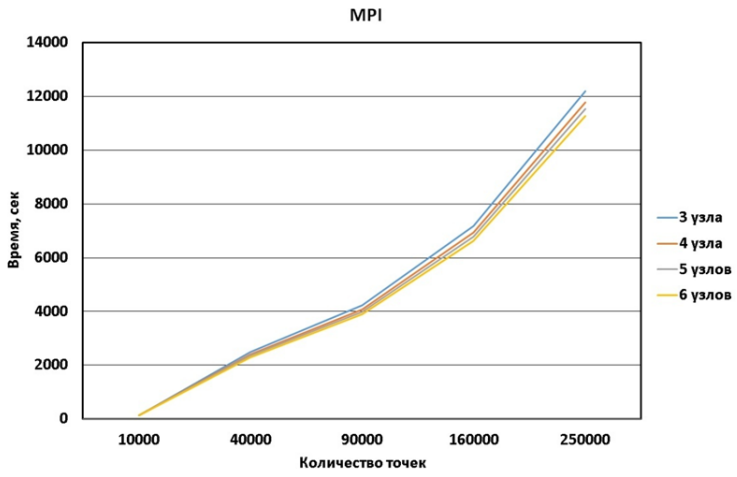

Рисунок 9. Время выполнения кластеризации с применением технологии MPI

Для сравнительного анализа эффективности применения технологии MapReduce для задач кластеризации были проведены тесты с применением технологии MPI на тех же объемах данных (рис. 9). На рис. 10 представлены результаты времени выполнения алгоритмов кластеризации MapReduce без комбинатора, MapReduce с комбинатором, а также с применением MPI.

Экспериментальные результаты показывают (рис. 10), что алгоритмы кластеризации с применением технологии MapReduce могут эффективно обрабатывать большие объемы данных.

Заключение

В работе предложен параллельный итерационный алгоритм кластеризации с использованием комбинаторов, реализованный на платформе MapReduce Hadoop. Повышение производительности достигается за счет управления процессом обработки данных. Осо-

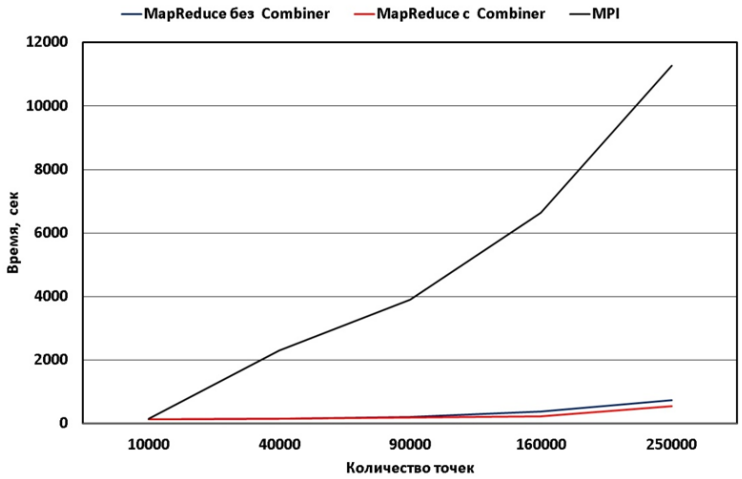

Рисунок 10. Сравнение времени выполнения кластеризации с MapReduce без комбинатора, MapReduce с комбинатором, MPI

бенностью данной работы является то, что а) обрабатываемые данные делятся на два вида: статические, которые хранятся в локальной файловой системе, и данные состояния, которые участвуют в обмене между функциями Map и Reduce; б) итерационный алгоритм использует комбинаторы, которые позволяют уменьшить объем данных, участвующих в обмене MapReduce процессов.

Представленные в данной работе результаты доказывают эффективность применения технологии MapReduce для обработки данных ДДЗ. Парадигма MapReduce может быть применена для других алгоритмов обработки данных ДЗЗ, при этом ожидается получение значительного выигрыша в производительности.

Данная работа выполнялась в рамках научно-технического проекта по грантовому финансированию «Разработка моделей и приложений высокопроизводительной распределенной обработки данных на основе технологии MapReduce – Hadoop для задач нефтедобычи».

Список литературы

[1] *Шовенгердт Р.А.*, Дистанционное зондирование. Модели и методы обработки изображений. – М.: Теносфера, 2010. – 560 с.

[2] *Antonio J. Plaza and Chein-I Chang*, High Performance Computing in Remote Sensing. – Chapman and Hall/CRC, 2007. – 496 p.

[3] *J. Dean, S. Ghemawat*, MapReduce: Simplified Data Processing on Large Clusters. Communications of The ACM, 2008. – 51(1). – p. 107-113.

[4] *Дж. Ту, Р. Гонсалес.*, Принципы распознавания образов. М.: «Мир», 1978 г., 411 с.

[5] *C. Pughineanu, I. Balan*, Parallel Algorithm Evaluation in the Image and Clustering Processing // Electronics and electrical engineering. system engineering, computer technology T 120 No. 4 (110). 2011. P. 89-92.

[6] *A. Plaza, Chein-I Chang, Javier Plaza, David Valencia*, Commodity cluster and hardware-based massively parallel implementations of hyperspectral imaging algorithms // Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII. Proc. of SPIE Vol. 6233, 623316. 2006.

[7] *Ping ZHOU, Jingsheng LEI, Wenjun YE.*, Large-Scale Data Sets Clustering Based on MapReduce and Hadoop // Journal of Computational Information Systems 7: 16(2011). P. 5956-5963.

[8] *Bo Li, Hui Zhao, Zhen Hua LV.*, Parallel ISODATA Clustering of Remote Sensing Images Based on MapReduce // Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010. pp. 380-383.

[9] *Mohamed H. Almeer.*, Cloud Hadoop Map Reduce For Remote Sensing Image Analysis // Journal of Emerging Trends in Computing and Information Sciences. VOL. 3, NO. 4, April 2012. P. 637-644.

[10] *Z. Lv, Y. Hu, H. Zhong, J. Wu, B. Li, and H. Zhao*, 2010. "Parallel K-means clustering of remote sensing images based on MapReduce", in Proc. 2010 Int. Conf. Web Information Systems and Mining (WISM '10), pp. 162-170.

[11] *Satish Narayana Srirama, Pelle Jakovits, Eero Vainikko*, Adapting scientific computing problems to clouds using MapReduce // Future Generation Computer Systems 28(2012). P. 184–192.

[12] *Чак Лэм*, Hadoop в действии. М.: ДМК Пресс, 2012. – 424 с.

[13] *White T.*, Hadoop: The Definitive Guide. Stamford: O'Reilly Media, Inc. 2012. 625 p.

[14] *Мансурова М.Е., Шоманов А., Тулепбергенов Б.*, Параллельный алгоритм кластеризации для обработки гиперспектральных изображений на основе MapReduce Hadoop // Международная конференция "ИКТ: образование, наука, инновации", Алматы, 20-21 мая 2013 г. – с. 56-61.

[15] *Yanfeng Zhang, Qinxin Gao, LixinGao, and Cuirong Wang*, imapreduce: A distributed computing framework for iterative computation. J. Grid Comput,10(1). P. 47-68,

References

[1] *Shovengerdt P.A.*, Distancionnoe zondirovanie. Modeli i medody obrabotki izobrazheniyi. - M.: Tenosphera, 2010. - 560 s.

[2] *Antonio J. Plaza and Chein-I Chang*, High Performance Computing in Remote Sensing. – Chapman and Hall/CRC, 2007. – 496 p.

[3] *J. Dean, S. Ghemawat*, MapReduce: Simplified Data Processing on Large Clusters. Communications of The ACM, 2008. – 51(1). – p. 107-113.

[4] *Tou J., Gonsalez R.*, Principy raspoznavaniya obrazov. M.: "Mir 1978, 411 s.

[5] *C. Pughineanu, I. Balan*, Parallel Algorithm Evaluation in the Image and Clustering Processing // Electronics and electrical engineering. system engineering, computer technology T 120 No. 4 (110). 2011. P. 89-92.

[6] *A. Plaza, Chein-I Chang, Javier Plaza, David Valencia*, Commodity cluster and hardware-based massively parallel implementations of hyperspectral imaging algorithms // Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII. Proc. of SPIE Vol. 6233, 623316. 2006.

[7] *Ping ZHOU, Jingsheng LEI, Wenjun YE.*, Large-Scale Data Sets Clustering Based on MapReduce and Hadoop // Journal of Computational Information Systems 7: 16(2011). P. 5956-5963.

[8] *Bo Li, Hui Zhao, Zhen Hua LV.*, Parallel ISODATA Clustering of Remote Sensing Images Based on MapReduce // Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010. pp. 380-383.

[9] *Mohamed H. Almeer.*, Cloud Hadoop Map Reduce For Remote Sensing Image Analysis // Journal of Emerging Trends in Computing and Information Sciences. VOL. 3, NO. 4, April 2012. P. 637-644.

[10] *Z. Lv, Y. Hu, H. Zhong, J. Wu, B. Li, and H. Zhao*, 2010. "Parallel K-means clustering of remote sensing images based on MapReduce", in Proc. 2010 Int. Conf. Web Information Systems and Mining (WISM '10), pp. 162-170.

[11] *Satish Narayana Srirama, Pelle Jakovits, Eero Vainikko*, Adapting scientific computing problems to clouds using MapReduce // Future Generation Computer Systems 28(2012). P. 184–192.

[12] *Lam Ch.*, Hadoop v deystvii. M.: DMK Press, 2012. - 424 s.

[13] *White T.*, Hadoop: The Definitive Guide. Stamford: O'Reilly Media, Inc. 2012. 625 p.

[14] *Mansurova M., Shomanov A., Tulepbergenov B.*, Parallelnyi algoritm klasterizacii dlya obrabotki giperspektralnyh izobrazheniy ha osnove MapReduce Hadoop // Trudy mezhdunarodnoy konferencii "IKT: obrazovanie, nauka, innovacii Almaty, 20-21 maya 2013 g. - s. 56-61.

[15] *Yanfeng Zhang, Qinxin Gao, LixinGao, and Cuirong Wang*, imapreduce: A distributed computing framework for iterative computation. J. Grid Comput,10(1). P. 47-68,

Поступила в редакцию 19 сентября 2013 года