BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 5, Number 5(2014), 13 – 17

UDC 519.683; 519.684

PARALLEL ALGORITHM FOR MULTI-CORE PROCESSORS WITH USING K-MEANS METHOD FOR SOLVING CLUSTERIZATION PROBLEM

N. Litvinenko
n.litvinenko@inbox.ru
Institute of mathematics and mathematical modelling, Committee of Science
of the Ministry of Education and Science of the Republic of Kazakhstan, Almaty

Key words: Parallel algorithms, cluster analysis, multithreading, K-means method,multi-core processors.
Abstract. Parallel algorithm for multi-core processors with using K-means method for solving clusterization problem is developed. This algorithm was implemented in source code on C++ in Microsoft Visual Studio 2010 with using multithreading. The maximum amount of data: up to 300 000 records with the number of indexes to 25.This development may have applications in various areas of science, for example, in genetics, biology,computer science, sociology e.t.c.

УДК 519.683; 519.684

МЕТОД К-СРЕДНИХ ДЛЯ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАСТЕРНОГО АНАЛИЗА С ПРИМЕНЕНИЕМ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Н.Г. Литвиненко
n.litvinenko@inbox.ru
Институт математики и математического моделирования КН МОН РК, Алматы, Казахстан

Ключевые слова: Параллельные алгоритмы, кластерный анализ, мультипоточность, метод К-средних, многоядерные процессоры.
Аннотация. Для объемных задач кластеризации по методу К-средних разрабатывается параллельный алгоритм для многоядерных процессоров. Данный алгоритм реализован в программном коде на языке С++ в среде MicrosoftVisualStudio 2010 с использованием средств мультипоточности. Максимально допустимый объем данных: до 300 тыс. записей с количеством показателей до 25.

Работа выполнена при поддержке гранта 0741/ГФ МОН РК

Введение. Задача кластеризации является объемной многошаговой вычислительной задачей разбиения множества объектов на группы (кластеры). Существуют различные методы кластеризации, которые чаще всего определяются на стадии построения модели исследуемого процесса. Метод К-средних один из наиболее признанных методов среди разработчиков прикладных задач. Данный метод подразумевает, что исследователю априори известно количество кластеров.Метод является одним из самых простых в реализации, хотя для больших данных общедоступных программных средств нет.К-средних, наверное, самый быстрый из распространенных методов кластеризации. Однако метод имеет свои слабые места. Он не всегда является устойчивым по отношению к первоначальному выбору К-центров. Обычно проблема решается на стадии построения модели альтернативным выбором первоначальных К-центров. Новый выбор часто определяется из конкретных условий исследуемого процесса.Другим слабым местом данного метода

является то, что находится обычно локальное оптимальное решение. Если исследователя не устраивает данное локальное решение,он должен решать проблему изменением начальных К-центров. Третьим слабым местом является возможность возникновения на очередной итерации пустого кластера. Проблема решается на этапе разработки алгоритма различными способами. Например, осуществляется проверка на пустоту кластера и если пустой кластер существует, выбирается один из непустых кластеров и делится на два кластера, а пустой кластер удаляется.

Общая идея построения кластеров по методу К-средних следующая. На очередной итерации все объекты исследуемого массива относятся к ближайшему центру кластера, построенного на предыдущей итерации. Для вновь построенных кластеров пересчитываются центры тяжести. Вычисляется изменение функционала. Если функционал уменьшается, переходим к новой итерации. Когда функционал перестает уменьшаться, процесс заканчивается.

В данной работе рассматривается задача с большим количеством данных – до 300тысяч записей и до 25 показателей. Данная задача хорошо распараллеливается. Основная идея распараллеливания – каждому потоку назначить свое подмножество объектов и на каждой итерации каждый поток распределяет свой набор объектов по кластерам.

Аналогичные разработки. Метод К-средних был предложен практически одновременно в 1950 году Гуго Штейнгаузом и Стюартом Ллойдом. Дальнейшее развитие метод получил в виде K-means++, в котором делается попытка автоматизировать выбор начальных К-центров. Широко известна нейросетевая реализация K-means.

Метод К-средних реализован во многих универсальных прикладных пакетах, например STATISTICA, SPSS. Данные пакеты хорошо ориентированы на пользователя, позволяют решать практически все встречающиеся на практике задачи, общедоступны, недороги, с хорошей и доступной документацией, с хорошей технической поддержкой. Имеется много технической литературы с хорошо и подробно разобранными примерами. Основным недостатком таких пакетов является ограниченность по объемам обрабатываемых данных. Как правило, все эти пакеты однопоточные и ресурсы рабочей станции по этой причине задействованы очень слабо. Серьезные объемные исследования с помощью данных пакетов проводить нельзя.

Существуют и специализированные пакеты. Они, как правило, возникают при разработке какого-либо проекта. Такие пакеты обычно заточены под специфику разрабатываемого проекта, эти пакеты сложно использовать в другом проекте. Техническая документация обычно отсутствует, техническая литература отсутствует. Пакеты или недоступны, или дороги. Техническая поддержка слабая.

Из современных алгоритмов, реализованных или не реализованных в программном продукте можно отметить следующие:

Алгоритм BIRCH

BIRCH относится к числу дивизимныхалгоритмовDIANA (DivisiveAnalysis). К плюсам можно отнести двухэтапную кластеризацию, возможность обработки очень большого числа числовых данных, работу на ограниченном объеме памяти. Данный алгоритм при расчетах способен учитывать неравномерное распределение данных в пространстве и считать область с большей плотностью за один кластер.Основные минусы алгоритма – работа исключительно с числовыми данными, выделение кластеров только сферической формы, необходимость задания пороговых значений.

Алгоритм CURE

CURE относится к числу агломеративных алгоритмов AGNES (Agglomerativenesting). Выполняет иерархическую кластеризацию с использованием множества идентифицирующих точек для определения объекта в кластер. К плюсам можно отнести выделение кластеров сложной формы. К минусам – необходимость задания пороговых значений и количества кластеров.

Алгоритм ROCK

ROCK относится к числу агломеративных алгоритмов AGNES (Agglomerativenesting). Алгоритм сочетает в себе все хорошие стороны методов k-means и ближайшего соседа. Имеет существенный недостаток, связанный с большими вычислительными затратами.

Ввиду вышесказанного представляется весьма перспективной разработка прикладного пакета программ, ориентированного, во-первых, на большие объемы обрабатываемых данных, а во-вторых, доступные для массового использования.

Постановка задачи. Имеется достаточно большой набор данных, характеризующий некоторое множество объектов или процессов. Набор данных может содержать до 300 тысяч записей, и до 25 полей, описывающих характеристики объекта. Требуется разбить данное множество объектов на кластеры, содержащие схожие объекты, по методу К-средних. В данной работе мы не рассматриваем вопросы, связанные с корректной подготовкой исходных данных.

За расстояние между объектами a и b возьмём обычное Евклидово расстояние

$$\rho_{ab} = \sqrt{\sum_j (x_{j,a} - x_{j,b})^2}$$

При построении кластеров может возникнуть необходимость учитывать некоторые из дополнительных условий:

Возникновение пустого кластера. Если при построении кластеров возник пустой кластер, необходимо выбрать один из непустых кластеров и разделить его на два, а пустой кластер удалить.

Слишком большое число объектов в кластере. Если количество объектов в кластере стало больше некоторого наперед заданного числа N1, мы делим этот кластер на 2 кластера с двумя центрами тяжести, а кластер с наименьшим количеством объектов удаляем, т.е., считаем объекты в этом кластере свободными. Продолжаем расчеты с новыми К-центрами.

Слишком большой диаметр кластера. Если диаметр кластера стал больше некоторого наперед заданного числа N2, делим этот кластер на 2 кластера, а кластер с наименьшим количеством объектов удаляем, т.е., считаем объекты в этом кластера свободными. Продолжаем расчеты с новыми К-центрами.

Описание алгоритма. Пусть K1 – количество записей, K2 – количество ядер процессора, K3 – количество потоков, K4 – количество полей в записи.

В общем случае алгоритм должен быть параллельным. Однако, если данных мало, однопоточный алгоритм будет эффективнее. Будем считать, что если данных меньше 10 тысяч, программа должна работать в однопоточном режиме. Опишем вариант параллельного алгоритма.

Перечислим основные задачи, которые должны выполняться:

Первоначальный выбор К центров. Задача выполняется в однопоточном режиме.

Распределение объектов по кластерам по принципу, какой центр ближе, к тому кластеру и будем относить рассматриваемый объект. Задачу целесообразно выполнять в мультипоточном режиме.

Вычисление новых центров тяжести. Задачу целесообразно выполнять в мультипоточном режиме.

Вычисление пустых кластеров. Задача простая и выполняется в однопоточном режиме.

Вычисление количества объектов в кластерах. Задача простая и выполняется в однопоточном режиме.

Расчет диаметра кластеров. Задачу целесообразно выполнять в мультипоточном режиме.

Разбиение кластера на два кластера. Задача простая и выполняется в однопоточном режиме.

Если K1<10000, работает однопоточный режим; иначе работает мультипоточный режим.

Далее необходимо определить оптимальное количество потоков. Практика показывает, что наиболее эффективно брать количество потоков втрое больше количества физических ядер процессора.

Определяем количество ядер процессора на данной рабочей станции K2. Вычисляем K3=K2*3.

Считываем данные и условия расчета в оперативную память MAS1.

Делим все данные на K3 порций (по количеству потоков). Это массивMASPJ. J=1,2,3,…,K3.

Подготавливаем K3 потоков для работы.

Каждый поток просматривает свой набор объектов и вычисляет расстояния до К центров. Объект относится к тому центру, расстояние до которого наименьшее.

Строим новый пул потоков для новых задач. Теперь каждый поток будет обрабатывать отдельный кластер. Если кластеров больше количества потоков, в некоторых потоках будет последовательно обрабатываться более одного кластера.

Задачами каждого потока на этом этапе будут (8, 9, 10, 11)

Подсчет количества элементов NJ в кластере (KJ).

Каждый поток вычисляет новый центр тяжести кластера J по формуле

$$x_r^{KJ} = \frac{\Sigma_{j \in KJ} x_r^j}{NJ}$$

Здесь KJ=1,2,3,..,K; r=1,2,3,…,25 – количество показателей

Каждый поток вычисляет диаметр кластера.

Каждый поток вычисляет функционал кластера I^{KJ} по формуле

$$I^{KJ} = \sum_{a \in KJ} \sum_{j} (x_{j,a} - x_{j,KJ})^2$$

Далее работа идет в однопоточном режиме.

Рассчитывается общий функционал по формуле

$$I_2 = \sum I^{KJ}$$

Если

$$I_2 < I_1$$

Переходим к пункту 6. Иначе заканчиваем работу.

Среда разработки. Для разработки данного программного обеспечения использовался системный блок, оснащенный:

Материнская плата - Gigabyte Technology Co., Ltd., Z77MX-D3H, Chipset Intel;

CPU - Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz;

GPU - NVIDIA GeForce GTX 660 сархитектуройKepler (CC 3.0);

Оперативная память - 16384 Mb; Жесткий диск - 2 Тб.

Операционная система - MicrosoftWindows 7, Ultimate, 32 bit.

Использовалась следующая среда разработки:

Язык программирования - C++.

Программная среда- Microsoft Visual Studio 2010.

Выводы. В данной работе рассматривается кластеризация объектов по методу К-средних для больших объемов данных. Существующие прикладные пакеты по статистике, к сожалению, не позволяют решать задачи с большими объемами данных. Разработанные параллельные алгоритмы и их программная реализация сориентированы именно на объемные задачи. Метод К-средних является достаточно востребованным в различных прикладных проектах и реализация данного метода для задач с большими объемами данных может быть востребована при разработке различных проектов в биологии, генетике, социологии.

ЛИТЕРАТУРА

[1] Уильямс Э. Параллельное программирование на C++ в действии. Москва, ДМК Пресс, 2012г

[2] Гергель В.П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем, Нижний Новгород, Изд-во НГУ, 2010

[3] Богачев К.Ю., Основы параллельного программирования, Москва, Бином, 2003

[4] ЭхтерШ, Робертс Д. Многоядерное программирование. «Питер», 2010

[5] Вятченин Д. А. Нечёткие методы автоматической классификации. — Минск: Технопринт, 2004. — 219 с.

REFERENCES

[1] Uiljams Je. Parallel'noeprogrammirovanienaC++ v dejstvii. Moskva, DMK Press, 2012

[2] Gergel V.P. Vysokoproizvoditel'nyevychislenijadljamnogojadernyhmnogoprocessornyhsistem, NizhnijNovgorod, Izdatelstvo NGU, 2010

[3] BogachevK.Ju., Osnovyparallelnogoprogrammirovanija, Moskva, Binom, 2003

[4] JehterSh, Roberts D. Mnogojadernoeprogrammirovanie. «Piter», 2010

[5] Vjatchenin D. A. Nechjotkiemetodyavtomaticheskojklassifikacii. — Minsk: Tehnoprint, 2004. — 219 s.

ПАРАЛЛЕЛЬДІ АЛГОРИТМ ОРТАЛЫҚ ПРОЦЕССОРДЫҢ КӨП АҒЫНДЫЛЫҚ ӘДІСТЕРІН КОМПЛЕКСТІ ҚОЛДАНЫС

Н.Г. Литвиненко

институт математики и математического моделирования НАН РК, Алматы, Казахстан

Тірек сөздер: Параллельді алгоритм, графикалық процессор, кластерлік талдау, көп ағындылық, ең жақын көрші тәсілі.

Аннотация: Берілген мақалада көлемі 2 млн. жазбасы және 25-ке дейін көрсеткішері бар есептердің, ең жақын көрші (ЕЖК) кластеризация тәсілі арқылы, шығару жолдары сүреттеледі. Мағлұматтардың көлемдері үлкен болуына байланысты, есептерді шығару үшін есептеуіш графикалық процессорлар қолданылады. Параллельді алгоритм орталық процессордың көп ағындылық әдістерін комплексті қолданыс астында пайдалана отырып және берілгендерді графикалық процессор арқылы параллельді өңдеу мүмкіндіктері Microsoft Visual Studio 2010 ортасында, C++ тілінде жүзеге асырылды. Айтылмыш зерттеме ғылымның түрлі тармақтарында, мысалы биологияда, генетикада, социология және т.б. қолданыс табуы мүмкін.

Поступила 09.09.2014 г.

BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 5, Number 5(2014), 17 – 20

UDC 519.683; 519.684

CLUSTERING OF LARGE AMOUNTS OF DATA BY THE COMPLETE LINKAGE METHOD IN MULTITHREADING MODE

V. Pospelova

Institute of Mathematics and Mathematical Modelling, Almaty, Kazakhstan

Key words: clustering of big data, complete-linkage method, multithreading.

Abstract. In this paper version of the parallel algorithm and its implementation for solving the cluster analysis problems by the method of complete linkage (MCL) in the environment of multi-core processors are considered. This algorithm is implemented in software code written in C # in Microsoft Visual Studio 2010 with the use of multithreading. Algorithm and its implementation are designed for volume tasks. Estimated volume: up to 300 thousand records with the number of fields to 25.

УДК 519.683; 519.684

КЛАСТЕРИЗАЦИЯ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ ПО МЕТОДУ ПОЛНОЙ СВЯЗИ В МУЛЬТИПОТОЧНОМ РЕЖИМЕ

В. Поспелова
Институт математики и математического моделирования, Алматы, Республика Казахстан

Ключевые слова: кластеризация больших объемов данных, метод полной связи, мультипоточность.

Аннотация. В текущей работе рассматривается вариант параллельного алгоритма и его программная реализация для решения задач кластерного анализа по методу полной связи (дальше МПС) в среде многоядерных процессоров. Данный алгоритм реализован в программном коде на языке C# в среде MicrosoftVisualStudio 2010 с использованием средств мультипоточности. Алгоритм и его реализация рассчитаны на объемные задачи. Предполагаемый объем данных: до 300 тыс. записей с количеством полей до 25.

Работа выполнена при поддержке гранта 0741/ГФ МОН РК

Кластеризация данных является частной задачей интеллектуального анализа данных (DataMining), главной задачей которой является объединение объектов в небольшие группы по схожим признакам. Подобные объединения должны быть проведены с учетом схожести и отличий объектов, степень схожести которых определяется расстоянием. Главное отличие кластеризации данных от классификации заключается в том, что в кластеризации группы объектов определяются ее результатом, в то время как в класификации группы, к которым необходимо отнести объекты, заранее определены. Данное различие объясняет интерес ученых к использованию алгоритмов кластеризации для исследования в прикладных областях науки, так как позволяет проводить распределение без обучающей информации (количество групп и сами группы заранее неизвестны).

Современный программный рынок уже сегодня предлагает разнообразные универсальные (STATISTICA, SAS, Minilab, SPSSStatistics) и специализированные (STADIA, STATIT, KNIME) пакеты для статистического анализа, частично решающие и задачи кластерного анализа. Однако ни те, ни другие не могут быть использованы для решения задач с большими объемами данных, часто возникающих в таких прикладных отраслях науки, как генетика и социология. Решение задач

кластеризации большого объема данных позволяет не просто разбить объекты на группы (кластеры), но и в итоге выявить связывающие объекты законы. В среднем рынок программного обеспечения предлагает обработку порядка 14000-16000 записей на компьютере средней мощности, однако этого явно недостаточно, так как перспективным является обработка от 100000 до 1 млн записей без особых временных затрат (количество характеристик около 25). Конечно стоит учитывать, что популярный алгоритм k-средних дает возможность продуктивной кластеризации большего объема данных, и на сегодня он успешно реализован в большинстве статистических пакето и показывает хорошие результаты (например,STATISTICA). Однако нельзя сказать что данный алгоритм является универсальным и подойдет для любой задачи. Поэтому возникает необходимость разработки алгоритмов по другим методам кластеризации с их дальнейшей реализацией в программном коде.

В настоящее время над решением объемных задач кластерного анализа работают российские и зарубежные разработчики. Однако нам неизвестно на какой стадии находятся данные разработки, так как все они имеют закрытый характер и, скорее всего, будут недоступны для массового использования.По этим причинам целесообразной и весьма перспективной является разработка прикладного пакета программ, ориентированного, во-первых, на большие объемы обрабатываемых данных, а, во-вторых, на решение конкретных прикладных задач.

Эффективное решение задач кластерного анализа по методу МПСпредставляет определенный интерес для многих прикладных отраслей науки. Однако при работе с большими объемами данных возникает проблема с выбором инструментария, реализующих данный метод, для решения задач. Стандартных программных пакетов для работы с большими объемами задач практически нет. В данной работе сделана попытка заполнить этот пробел, так как интерес рынка к данной теме возрастает.

Данная задача является комплексной, и для ее успешного решения необходимо:

Найти способы уменьшения количества итераций, необходимых для нахождения соседних кластеров.

Разработать эффективные параллельные алгоритмы.

Как можно более эффективно использовать технические возможности многоядерных процессоров.

Постановка задачи.Требуется разработать алгоритм объединения объектов в кластеры и реализовать его в программном коде по методу МПС. Предполагаемый объем данных может достигать 300 тысяч записей, количество характеристик может достигать 25.

За расстояние между объектами a и b возьмём обычное Евклидово расстояние:

$$\rho_{ab} = \sqrt{\sum_j (x_{j,a} - x_{j,b})^2}$$

Расстояние между кластерами P и Rпо методу МПС определяется формулой:

$$L_{PR} = \max_{\substack{a \in P \\ b \in R}} (\rho_{ab})$$

При решении задач кластеризации необходимо задать ряд дополнительных условий, которые позволят контролировать поцесс кластеризации. Например:

Ограничение на минимальное количество построенных кластеров. Построение кластеров прекращается, как только количество кластеров становится меньше заданного числа N1.

Ограничение на максимальное расстояние между кластерами. Работа по построению кластеров прекращается, если расстояние между двумя ближайшими кластерами становится больше заданного числа N2.

Ограничение на максимальное количество объектов кластера. Выбранный кластер не подлежит объединению с другими, если количество объектов в нем больше заданного числа N3.

Ограничение на максимальный размер кластера. Кластер не подлежит объединению с другими кластерами, если его диаметр превышает значение заданного числа N4.

Так как решаемая задача представляет собой задачу с большим объемом данных, абсолютно логическим является отступление от стандартных иерархических методах кластеризации, где на каждой итерации происходит объединение только одной пары кластеров, расстояние между

которыми наименьшее. Причина подобного отказа ввиду больших временных затрат на вычисления при работе с большими объемами данных. В предлагаемом алгоритме рассматривается способ, когда на очередной итерации происходит объединение не одной пары кластеров, а нескольких.Кластеризацию данных поставленной задачи будем проводить по методу МПС. В данном методе расстояние между кластерами будет определяться расстоянием между их самими отдалёнными членами. На первом шаге каждый объект принимается за отдельный кластер. Анализируя результаты проведенных вычислений, объединяться в кластер у нас будут те объекты, расстояние между которыми наименьшее. После того как будет проведено некоторое количество таких объединений, будут происходить вычисления расстояний не только между выбранным объектом и объектом, но и между выбранным объектом и каждым объектом кластера. Согласно полученным расстояниям, объединяться будут кластеры, у которых расстояние между максимально уделенными представителями будет наименьшим. Включение объекта в кластер будет происходить по подобной схеме – объект будет включен в состав кластера, расстояние с максимально удаленным представителем которого у него будет наименьшим.

Для успешной реализации алгоритма в параллельном режиме нам необходимо обозначить следующие параметры: K_1 – количество записей, K_2 – количество ядер процессора, K_3 – количество потоков, K_4 – количество полей в записи.

Описываемый алгоритм будет работать в параллельном режиме с большим количеством данных. Однако не будем исключать из рассмотрения варианты, когда необходимо обработать малый набор данных – в таком случае разумнее использовать однопоточный режим. Запишем следующее условие:

Условие определения режима работы центрального процессора: если $K_1 < 10000$, процессор работает в однопоточном режиме, если же $K_1 \geq 10000$ – в мультипоточном режиме.

Следующим подготовительным к вычислениям шагу является определение числа потоков для центрального процессора по формуле:

$$K_3 = K_2 * 3,$$

где выбор утроенного значения количества ядер K_2 обосновывается практическими результатами. Так как на момент начала 2013 года наибольшее распространение имели физические процессоры с 4 ядрами, положим $K_2 = 4$. Получим

$$K_3 = 4 * 3 = 12.$$

После того, как все вспомогательные характеристики будут вычислены, имеющиеся данные и условия расчета передаются в оперативную память. Следующим шагом является разделение данных на количество потоков (в нашем случае 12), и формирование задания для каждого потока. Потоки готовятся к работе.

Дальнейшие итерации происходят по алгоритму классификации метода МПС:каждым потоком производится вычисление расстояний между всеми возможными парами объектов, включая объекты кластеров, после чего каждым потоком отбирается около 100 пар, расстояние между которыми оказалось минимальным. Данные пары передаются на обработку в следующий цикл.

Условно обозначим описанный цикл внутренним, а следующий – внешним. Задачей внешнего цикла является объединение всех направленных ему пар от всех потоков и нахождение среди них 100 пар с наименьшим расстоянием, а так же объединение выбранных пар в кластеры. Так же цикл принимает решение, остановить работу или повторить итерацию. Данное решение принимается, исходя из условий, представленных в параграфе «Постановка задачи», например, если количество построеных кластеров становится меньше заданного числа N1 (Ограничение на минимальное количество построенных кластеров). В случае, если формирование кластеров незакончено, откорректированные данные снова разделяются между потоками и направляются на обработку во внешний цикл.

Заключение.Актуальным для любой задачи является применения алгоритма, наиболее эффективно справляющегося с ее решением. Напомним, что каждый из существующих методов

кластеризации имеет свои достоинства и недостатки и хорошо справляется с решением только узкого круга задач. Поэтому исследование имеющихся и разработка новых алгоритмов для решения задач кластерного анализа и их реализация в программном коде на сегодня является одним из перспективных направлений.

ЛИТЕРАТУРА

[1] Jain A., Murty M., Flynn P. Data Clustering: A Review. // ACM Computing Surveys. 1999. Vol. 31, no. 3.
[2] Дюран Б., Оделл П. Кластерный анализ. Пер. с англ. Е.З.Демиденко под ред. и с предисл. А.Я.Боярского. - М.: Статистика, 1977. -128 с.: ил.
[3] Фленов М.Е. Библия C# (2-е издание, 2011), Санкт-Петербург: «БХВ-Петербург», 2011
[4] Шилдт Г. C# 4.0 полное руководство, Санкт-Петербург: «Печатный двор», 2011

REFERENCES

[1] Jain A., Murty M., Flynn P. Data Clustering: A Review. // ACM Computing Surveys. 1999. Vol. 31, no. 3.
[2] Durand B., Odell P. Cluster analysis. - M.: Statistics, 1977. – p. 128.
[3] Flenov M.E. C # Bible (2nd edition, 2011), St. Petersburg: "BHV-Petersburg", 2011
[4] Shildt G. C # 4.0 complete guide St. Petersburg: "Printing House", 2011.

В. Поспелова

ГРАФИКАЛЫҚ ПРОЦЕССТЕРДІ ҚОЛДАНА ОТЫРЫП, ТОЛЫҚ БАЙЛАНЫС ӘДІСІ БОЙЫНШАКӨЛЕМДІ МАҒЛҰМАТТАРЫ БАР КЛАСТЕРИЗАЦИЯ ЕСЕПТЕРІН ШЫҒАРУҒА КЕРЕКТІ ПАРАЛЛЕЛЬДІ АЛГОРИТМДЕРДІҢ ҚОЛДАНЫСЫ.

Тірек сөздер: үлкен көлемді мағлұматтарды кластеризациялау, толық байланыс әдіс, көп ағындылық.

Аннотация. Берілген жұмыста көп ядролы процессор аймағындағы толық байланыс әдіс(ТБӘ) арқылы кластерлік талдау есептері үшін, параллельді алгоритм және оның жүзеге асырылуы қарастырылады. Берілген алгоритм көп ағындылық әдісін колдана отырып Microsoft Visual Studio 2010 ортасында, C++ тілінде жүзеге асырылды. Алгоритм және оның орындалуы көлемді есептерге арналған. Болжамалы айтылмыш көлемі: 300 мың жазба, 25 жиек көрсеткішер саны.

Поступила 09.09.2014 г.