# Разработка параллельного алгоритма кластеризации текстовых документов FRiS-Tax на основе технологии MPI

М.Е. Мансурова[1], В.Б. Барахнин[2, 3], С.С. Аубакиров[1], Е. Хибатханулы[1], Мусина А.Б.[1]

Казахский национальный университет имени ал-Фараби[1], Институт вычислительных технологий СО РАН[2], Новосибирский государственный университет[3]

В данной работе описана параллельная реализация алгоритма FRiS-Tax для кластеризации корпуса документов. Алгоритм основан на оценке сходства между объектами в конкурентной ситуации, которая приводит к понятию функции конкурентного сходства (FRiS-функции). В качестве шкал для определения меры сходства были выбраны атрибуты библиографического описания документов. Распараллеливание осуществляется на этапе настройки коэффициентов в формуле меры сходства генетического алгоритма, а также непосредственно на этапе кластеризации. Алгоритм кластеризации реализован на высокопроизводительной платформе MPJ Express. Приведены количественные оценки времени выполнения процесса, демонстрирующие преимущества параллельной реализации алгоритма.

*Ключевые слова*: кластеризация текстовых документов, генетические алгоритмы, параллельные алгоритмы.

## 1. Введение

Объем цифровых документов с каждым днем увеличивается. Это затрудняет процесс выбора наиболее подходящего материала при поиске нужной информации. Кластеризация является одним из инструментов анализа данных, который позволяет сделать доступными для восприятия большие объемы информации. Под кластеризацией понимается процесс разбиения множества документов электронной базы на классы (кластеры), при котором элементы, объединяемые в один класс, имеют большее сходство, нежели элементы, принадлежащие разным классам. Процесс кластеризации данных является ресурсоемким, с ростом объема обрабатываемой информации задача еще больше усложняется. Для решения этой проблемы исследователи разрабатывают различные алгоритмы кластеризации с применением технологий параллельного программирования.

Целью данной работы является параллельная реализация алгоритма FRiS-Tax для кластеризации научных статей на основе технологии параллельных вычислений Message Passing Interface (MPI). В качестве меры близости при кластеризации в данной работе принята мера конкурентного сходства. Для настройки весовых коэффициентов при вычислении меры сходства используется генетический алгоритм.

Кратко изложим структуру статьи. В разделе 1 обоснована актуальность выбранного направления исследований. В разделе 2 описана задача кластеризации текстовых документов и принятая мера близости. Здесь же представлен алгоритм кластеризации FRiS-Tax. В разделе 3 описан генетический алгоритм для настройки коэффициентов в формуле меры сходства. В разделе 4 представлен параллельный алгоритм кластеризации с применением разработанного генетического алгоритма. В разделе 5 приведены результаты вычислительных экспериментов и проведен анализ полученных данных. В заключении подведены итоги выполненной работы.

## 2. Алгоритм кластеризации FRiS-Tax

В данном разделе описывается алгоритм кластеризации научных статей, который позволяет решать задачу нахождения по данному документу класса документов, схожих с ним по содержанию.

Работа выполнена в рамках научных проектов грантового финансирования МОН РК 2015-2017 гг.

Для того чтобы решить эту задачу, множество документов электронной базы разбиваются на классы по близости в пространстве атрибутов их библиографического описания.

В задаче кластеризации каждый кластер описывается с помощью одного или нескольких идентификаторов, называемых профилем или центроидом. В качестве шкал для определения меры сходства предлагается брать атрибуты библиографического описания документов. В качестве алгоритма кластеризации текстовых документов выбран алгоритм FRiS-Tax [1, 3], основанный на использовании функции конкурентного сходства.

Мера сходства *m* на множестве документов *D* задается следующим образом (1):

$$m: D \times D \to [0,1], \tag{1}$$

при этом функция *m* в случае полного сходства принимает значение 1, в случае полного различия – 0. Вычисление меры сходства осуществляется по формуле вида (2)

$$m(d_1, d_2) = \sum a_i m_i(d_1, d_2) \tag{2}$$

где *i* – номер элемента (атрибута) библиографического описания, $a_i$ – весовые коэффициенты, $m_i(d_1, d_2)$ – мера сходства по *i*-му элементу (иными словами, по *i*-й шкале).

В результате формализации идеи о том, что для оценки сходства между объектами необходимо учитывать конкурентную ситуацию, возникло понятие функции конкурентного сходства (FRiS-функции) [2]. В случае заданной абсолютной величины сходства $m(x, y)$ между двумя объектами конкурентное сходство объекта *a* с объектом *b* в конкуренции с объектом *c* определяется значением FRiS-функции $F_{b/c}(a)$, которое вычисляется по формуле (3):

$$F_{b/c}(a) = \frac{m(a,b) - m(a,c)}{m(a,b) + m(a,c)}. \tag{3}$$

При переходе от сходства между объектами к сходству между объектом и кластером используется тот же подход. Для оценки конкурентного сходства объекта *z* с первым кластером учитываются абсолютное сходство $m(z,1)$ *z* с этим кластером и сходство $m(z,2)$ с конкурирующим вторым кластером. Нормированная величина конкурентного сходства при этом вычисляется по формуле (4):

$$F_{1/2}(z) = \frac{m(z,1) - m(z,2)}{m(z,1) + m(z,2)}. \tag{4}$$

В качестве величины сходства объекта *z* с кластером могут использоваться величина сходства объекта *z* с ближайшим к нему объектом из этого кластера либо величина сходства данного объекта с типичным представителем данного кластера.

Значения FRiS-функции меняются в пределах от −1 до +1. Если объект *z* совпадает с эталоном первого кластера, то $F_{1/2}(z) = 1$. При *m*(*z*, 1) = *m*(*z*, 2) объект *z* одинаково похож (или не похож) на оба кластера, тогда значение $F_{1/2}(z) = 0$. При совпадении объекта *z* с эталоном второго кластера его несходство с первым кластером максимально и равно $F_{1/2}(z) = -1$. Определенная таким способом функция конкурентного сходства хорошо согласуется с человеческими механизмами восприятия сходства и различия.

Целью работы данного алгоритма, как и большинства алгоритмов таксономии, является разбиение всего множества объектов выборки *A* на линейно разделимые кластеры похожих между собой объектов, которые затем объединяются в классы более сложных форм. Причем под похожестью в данном случае понимается конкурентное сходство с центральным объектом кластера (далее такие объекты будут называться столпами кластеров). Если множество столпов $S = \{s_1, s_2, ..., s_k\}$ (где *k* – число кластеров) уже выбрано, то все объекты выборки распределяют-

ся между столпами так, чтобы величина конкурентного сходства объектов со "своими" столпами была максимальной.

Нетрудно заметить, что у произвольного объекта $a \in A$ максимальное конкурентное сходство будет с ближайшим к нему столпом $s_{a1}$. Естественно, что в задаче таксономии множество столпов $S$ заранее не задано. Выбираться оно будет таким образом, чтобы средняя величина конкурентного сходства каждого объекта выборки $A$ с ближайшим к нему столпом из множества $S$ была максимальна:

$$\overline{F}(S) = \sum_{a \in A} F^*_{s_{a1}}(a) \to \max_S \qquad (5)$$

Чем больше эта величина, тем более похожи объекты на свои столпы и тем лучше качество формируемой таксономии. Множество столпов наращивается последовательно путем выбора их из числа объектов выборки. На рисунке 1 представлена схема алгоритма кластеризации FRiS-Tax.


**Рис. 1**. Алгоритм кластеризации FRiS-Tax

## 3. Генетический алгоритм для настройки коэффициентов в формуле меры сходства

В данной работе в качестве атрибутов разделения на кластеры публикаций из библиографических баз данных выбраны:

- год выпуска;
- код УДК;
- ключевые слова;
- авторы;
- серия;
- аннотация;
- заголовок.

Для подбора весовых коэффициентов, которые используются в формуле меры сходства (2), был разработан генетический алгоритм. Генетический алгоритм – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путем случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе [4].

Генетический алгоритм состоит из следующих этапов ([4]):

1. Создание начальной популяции.

2. Отбор (селекция).
3. Выбор родителей.
3. Размножение (скрещивание).
4. Мутации.

Ниже представлена схема генетического алгоритма (рис. 2), и описаны этапы выполнения алгоритма применительно к задаче кластеризации.

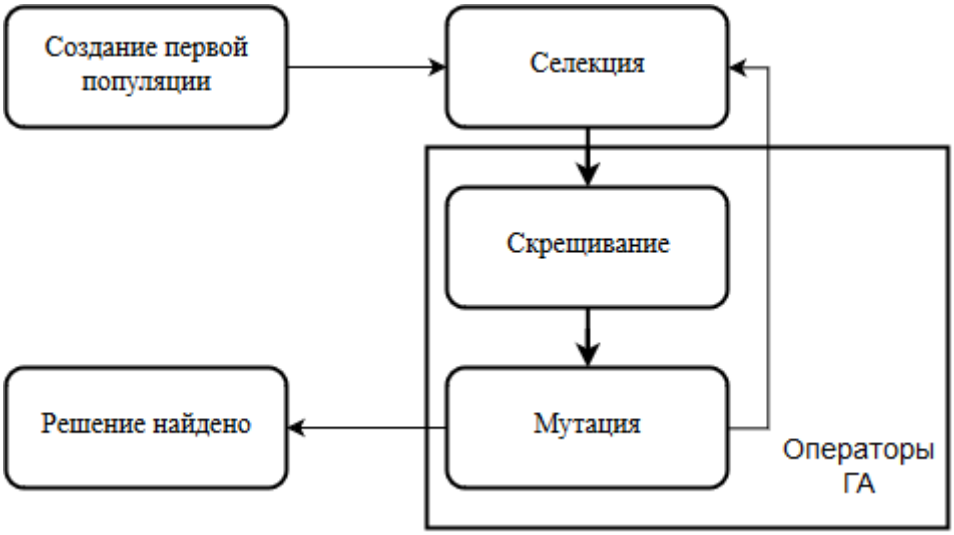


**Рис. 2.** Схема генетического алгоритма

## 3.1 Создание начальной популяции

Для создания начальной популяции и ее дальнейшей эволюции необходимо иметь упорядоченную цепочку генов или генотип. Согласно [4], некоторым, обычно случайным образом создается множество генотипов начальной популяции. Они оцениваются с использованием «функции приспособленности» или фитнесс-функции, в результате чего с каждым генотипом ассоциируется определенное значение («приспособленность»), которое определяет насколько хорошо генотип, им описываемый, решает поставленную задачу. Для данной задачи цепочка генов имеет фиксированную длину, равную 13, и представляет собой набор параметров, составленных на основе атрибутов библиографического описания документов. Сокращения на рисунке 3 составлены из первых букв названия гена, например, *UseAbstract = UAb*.

**Таблица 1.** Набор генов

| № | Полное название гена | Сокращенное название гена | Возможные значения |
|---|---|---|---|
| 1. | *POSSIBLE_DIFFERENCES* | *PD* | 0-3 |
| 2. | *UseAbstract* | *UAb* | 0-3 |
| 3. | *UseUdk* | *UU* | 0-1 |
| 4. | *UseKeyWords* | *UKW* | 0-3 |
| 5. | *UseAuthors* | *UAu* | 0-3 |
| 6. | *UseJournaSeria* | *UJS* | 0-1 |
| 7. | *UseTitle* | *UT* | 0-3 |
| 8. | *UseYear* | *UY* | 0-1 |
| 9. | *AuthorEquality* | *AuE* | 0-1 |
| 10. | *TitleEquality* | *TE* | 0-1 |
| 11. | *KeywordsEquality* | *KE* | 0-1 |
| 12. | *AbstractEquality* | *AbE* | 0-1 |
| 13. | *K* (количество кластеров) | *K* | 2-15 |


**Рис. 3.** Структура хромосомы

В правой колонке таблицы 1 указаны значения, которые могут принимать гены. Гены из заданного генотипа используются следующим образом. Рассмотрим гены, значения которых меняются в диапазоне от 0 до 3. Если значение гена равно 0, то он не используется в создании популяции. Если же значение больше 0, то это значение представляет собой вес соответствующего гена: *authorsWeight, keywordsWeight, titleWeight, abstractTextWeight.* Эти веса используются в дальнейшем при вычислении меры близости *m* по формуле (2).

Гены из таблицы 1, которые заканчиваются на слово *Equality*: *AuthorEquality, TitleEquality, KeywordsEquality, AbstractEquality* задают способ сравнения атрибутов документов и используются в создании популяции, только если соответствующие значения генов *UseAuthors, UseTitle, UseKeyWords, UseAbstract* положительны. При этом если значения *AuthorEquality, TitleEquality, KeywordsEquality, AbstractEquality* равны 0, то для сравнения списка авторов, названий статей и ключевых слов применяется обычное сравнение методом *Equals*. Если значения *AuthorEquality, TitleEquality, KeywordsEquality* равны 1, то для оценки меры близости атрибутов применяется расстояние Левенштейна [5]. Если же значение гена *AbstractEquality* равно 1, то для оценки меры близости аннотаций применяется алгоритм шинглов [6].

Значения генов *UseUdk, UseJournaSeria, UseYear* являются бинарными, то есть гены в зависимости от значения либо используются, либо не используются, в случае использования к мере *m* прибавляется 1. Ген *POSSIBLE_DIFFERENCES* представляет собой пороговое значение при оценке близости по расстоянию Левенштейна. Значение этого гена меняется от 0 до 3. Если *POSSIBLE_DIFFERENCES* = 0, то сравниваемые названия, авторы, ключевые слова должны полностью совпадать. Если при сравнении вычисленное расстояние по Левенштейну оказывается меньше порогового значения, то к мере близости *m* прибавляется соответствующий вес: *AuthorsWeight, titleWeight* или *KeywordsWeight*. Если расстояние по Левенштейну превышает пороговое значение, то делается заключение, что атрибуты различны. Значение гена *K* задает количество кластеров, на которые разбиваются текстовые документы.

## 3.2 Отбор

В генетическом алгоритме множество особей, каждая со своим генотипом, представляет собой некоторое решение задачи кластеризации. Предположим, что у нас порождена особь, то есть задан набор весовых коэффициентов для определения меры сходства. Далее производится кластеризация FRiS-Tax, где мера близости вычисляется с данным набором весовых коэффициентов. В алгоритме задается фитнесс-функция, которая позволяет определить, насколько хорошо выполнена задача кластеризации. Качество полученных кластеров в данной работе оценивается с помощью внешнего критерия качества кластеризации Purity [7]. Для принятия решения о том, какая из особей не прошла отбор и умирает, а какая выживает и будет участвовать в размножении, устанавливается нижняя граница (Threshold**)** для значений фитнесс-функции. Особь умирает, если функция возвращает значение, которое меньше установленной нижней границы.

На данном этапе определяются родители будущей особи с помощью метода отбора Roulette Selection [8]. Данный отбор происходит следующим образом. Суммируются значения фитнесс-функции всех особей, результат суммирования обозначается *sum*, затем выбирается случайное число между 0 и *sum*. Запускается цикл по особям, последовательно суммируются значения их фитнесс-функций. Как только сумма будет больше выбранного случайного числа, возвращается индекс той особи, которая последней участвовала в суммировании.

## 3.3 Скрещивание

Выжившие особи участвуют в размножении. Для этого выбираются две особи, между которыми производится скрещивание. Как следствие, получается новое поколение. На рисунке 4 представлен этап скрещивания генетического алгоритма.

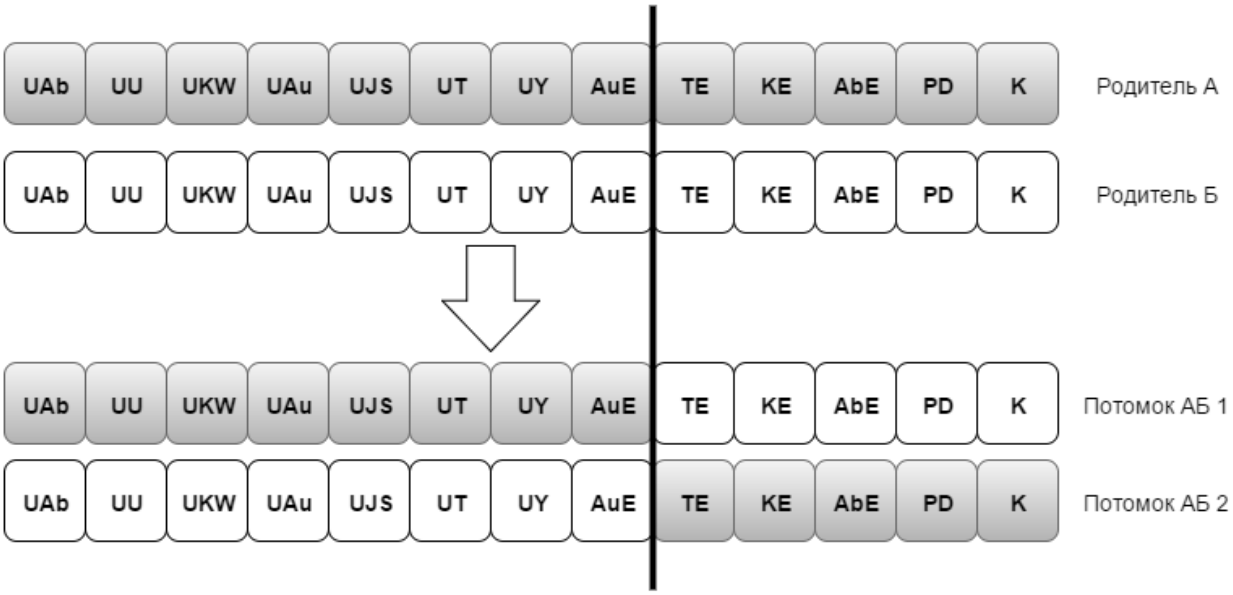


**Рис. 4.** Скрещивание хромосом

### 3.4 Мутация

Мутация нужна для того, чтобы решение задачи не попало в локальный экстремум. После того, как прошло скрещивание, предполагается, что некоторая часть новых особей подвергается мутации. Для этого случайным образом выбираются 25% всех особей, далее также случайным образом выбирается 25% генов этих особей, которые подвергаются мутации (рис. 5).

Таким образом, генетический алгоритм для задачи кластеризации состоит из стадии инициализации и стадии итераций.

*Стадия инициализации*:

Порождается первое поколение.

*Итерационная стадия*:

1. Выполняется кластеризация по алгоритму FRiS-Tax.
2. Вычисляется значение фитнесс-функции.
3. Значение фитнесс-функции сравнивается с пороговым значением качества. Для данной задачи пороговое значение равно 0,8. Если достигнуто заданное значение качества кластеризации, алгоритм останавливается.
4. Иначе порождается новое поколение: выполняется отбор особей, размножение, мутации.
5. Производится переход в пункт 1.

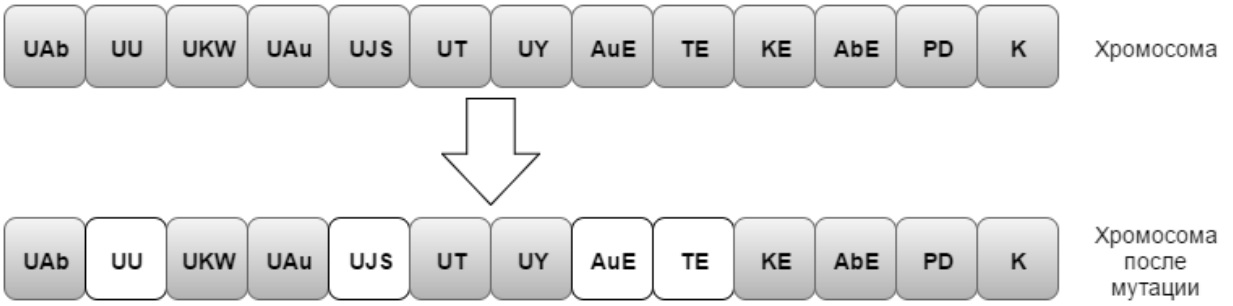


**Рис. 5.** Этап мутации генетического алгоритма

## 4. Параллельный алгоритм кластеризации FRiS-Tax

При большом количестве обрабатываемых документов время выполнения алгоритма кластеризации растет экспоненциально с ростом числа документов. В связи с этим для ускорения работы на двух этапах алгоритма были применены технологии параллельных вычислений. Во-первых, при отборе особей в генетическом алгоритме, когда выполняется кластеризация с заданным набором весовых коэффициентов. Параллельный генетический алгоритм реализован на высокопроизводительной платформе MPJ Express [9]. Во-вторых, при выполнении алгоритма кластеризации. В алгоритме FRiS-Tax самым сложным вычислительным процессом является обход всех объектов выборки и проверка каждого на роль столпа. Эта часть алгоритма реализована с помощью технологии Streams Java 8 [10].

### 4.1 Распараллеливание генетического алгоритма

Ниже представлены шаги параллельной версии генетического алгоритма (рис. 6).

1. Запускается $N$ процессов с помощью MPJ. Число $N$ представляет собой количество особей в первом поколении. Каждый процесс запускается на отдельном вычислительном узле.
2. Каждый процесс считывает файл со статьями, которые нужно разделить на кластеры.
3. Мастер-процесс генерирует $N$ случайных хромосом и записывает их в очередь.
4. Каждый процесс берет из очереди одну хромосому и создает особь, считает значение фитнесс-функции и отправляет мастеру-процессу.
5. Мастер-процесс проверяет, не нашлась ли особь со значением фитнесс-функции, большим или равным заданному пороговому значению (0,8). Если такая особь есть, то мастер-процесс сообщает всем остальным процессам, что особь найдена, и можно прекратить работу.
6. Если нет, то мастер-процесс начинает отбор. Скрещивание проходит внутри отбора.
7. После того как родители определились, рождается новая особь. Процесс скрещивания продолжается пока не появятся $N$ новых особей. Старое поколение не участвует в дальнейшей работе.
8. После этого производится мутация, случайным генам присваиваются новые случайные значения. Коэффициент мутации – 25 %.

Пока мастер-процесс выполняет отбор скрещивание и мутацию, остальные процессы ждут. Новые хромосомы записываются мастер-процессом в очередь, и цикл повторяется.

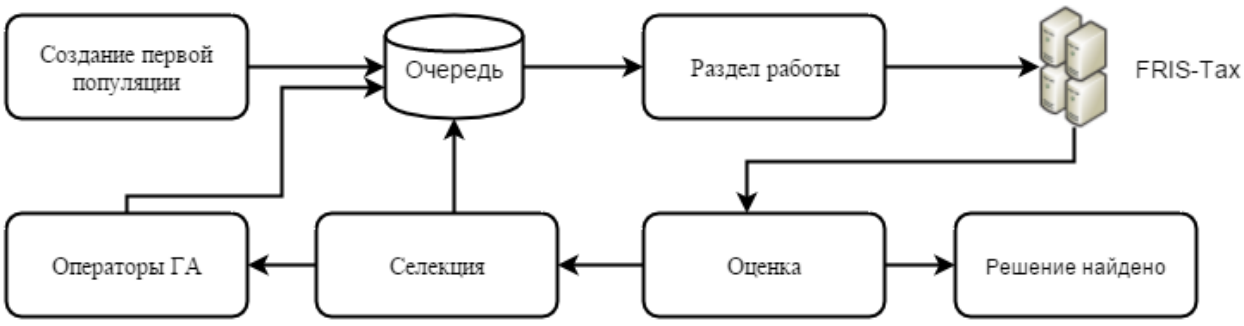


**Рис. 6**. Схема параллельного алгоритма кластеризации FRiS-Tax

### 4.2. Распараллеливание алгоритма кластеризации

С увеличением количества статей время работа FRiS-Tax растет экспоненциально. Нагрузочное тестирование выявило два самых медленных этапа в алгоритме FRiS-Tax. Ими оказались нахождение первого столпа и нахождение очередного столпа. Для ускорения этих этапов была применена технология Streams Java 8. Потоки позволяют задействовать все ядра узла. При работе одного MPJ-процесса на узле максимально работает только одно ядро. Когда процесс доходит до методов с потоками, начинают работать все ядра. Улучшения работы первого этапа заметны, начиная с 1000 статей, второго – с 600 статей.

## 5. Результаты вычислительного эксперимента

С целью исследования производительности разработанного параллельного алгоритма были проведены вычислительные эксперименты. Алгоритм были протестирован в Лаборатории компьютерных наук НИИ ММ при КазНУ имени аль-Фараби на кластере, состоящем из 16 машин.

Характеристики узлов:

- RAM: 16Gb
- Архитектура: x86_64
- CPU(s): Intel Core i5-2500 CPU 3.30GHz
- Сеть: 1Gbit/s.

На узлах установлена операционная система Ubuntu 14.04. Подчиненные узлы настроены для работы с библиотекой MPICH-3.1.4 и MPJ 0.42. Узлы подключены к сети Ethernet Intel(R) PRO/1000 Network Connection, которая обеспечивает пропускную способность в 1000 Mbps.

Тестирование проводилось на статьях журнала "Вестник КазНУ", опубликованных в период с 2008 по 2015 годы. Выборка включала 95 pdf документов, общее число статей – 2837. Выбор исходных данных обусловлен тем, что все документы разделены на серии (математика, биология, философия и т.д.) и дальнейшее разбиение не вызывает трудности при использовании мер сходимости, основанных только на библиографических описаниях либо заголовках статей. Для того чтобы более точно оценить качество разбиения выборки, данный корпус был разделен на кластеры с помощью эксперта в предметной области.

Время выполнения определялось следующим образом. Были произведены измерения времени процессов кластеризации для формируемых кластеров на одном вычислительном узле и на нескольких вычислительных узлах для параллельной реализации. На рис. 7 представлена зависимость времени выполнения алгоритма от количества процессов. На рисунках 8-9 представлены ускорение и эффективность параллельной реализации.

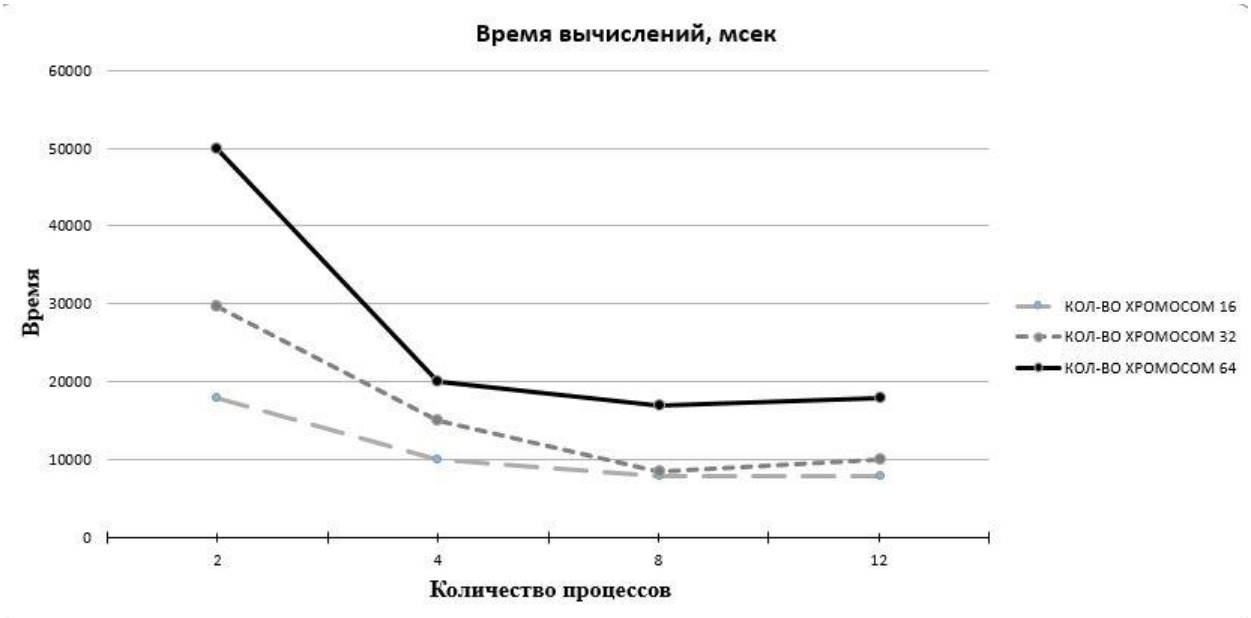


**Рис. 7.** Время вычислений параллельного алгоритма FRiS-Tax

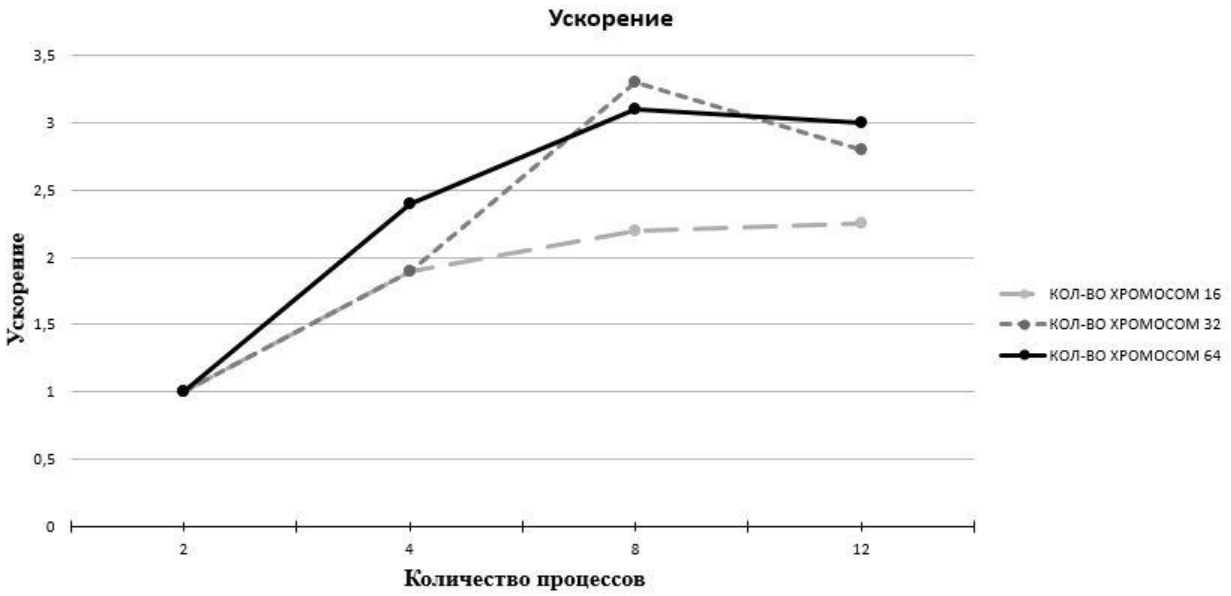


**Рис. 8.** Ускорение параллельного алгоритма FRiS-Tax

Как можно заметить по полученным графикам, с увеличением количества процессов ускорение растет лишь до некоторого значения. Для заданного объема обрабатываемых документов количеством процессов, при котором наблюдалось максимальное значение ускорения, оказалось значение 8, при этом наибольшее значение эффективности было достигнуто при запуске 4 процессов.

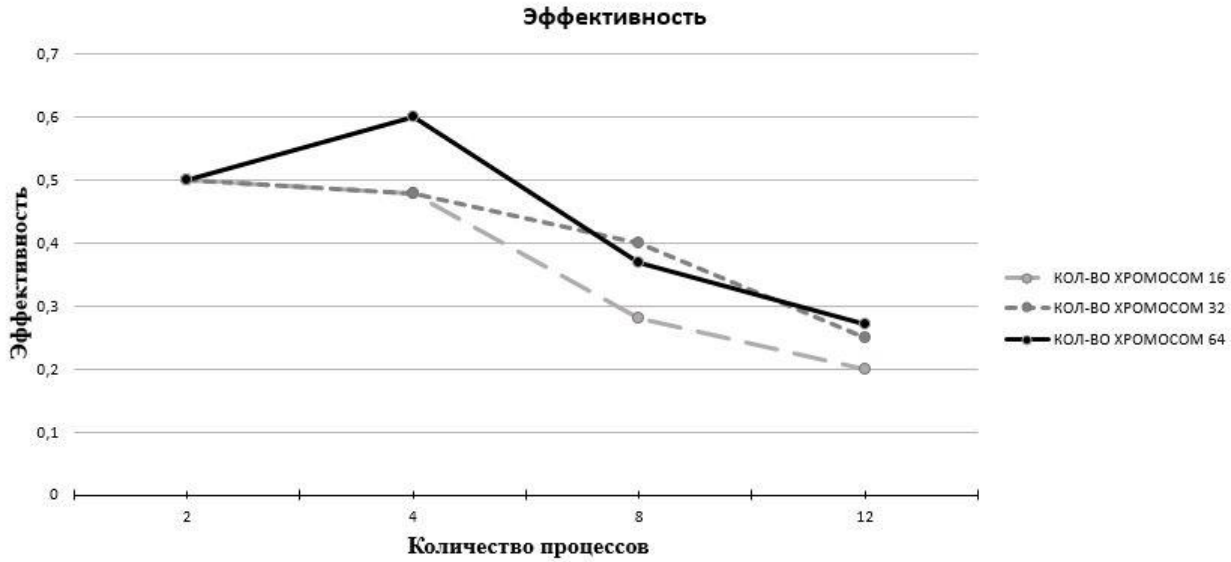


**Рис. 9.** Эффективность параллельного алгоритма FRiS-Tax

Для мониторинга выполнения алгоритма был разработан веб интерфейс, который позволяет наблюдать за текущими значениями параметров генетического алгоритма и достигнутых значений фитнесс-функции (рис. 10-11).

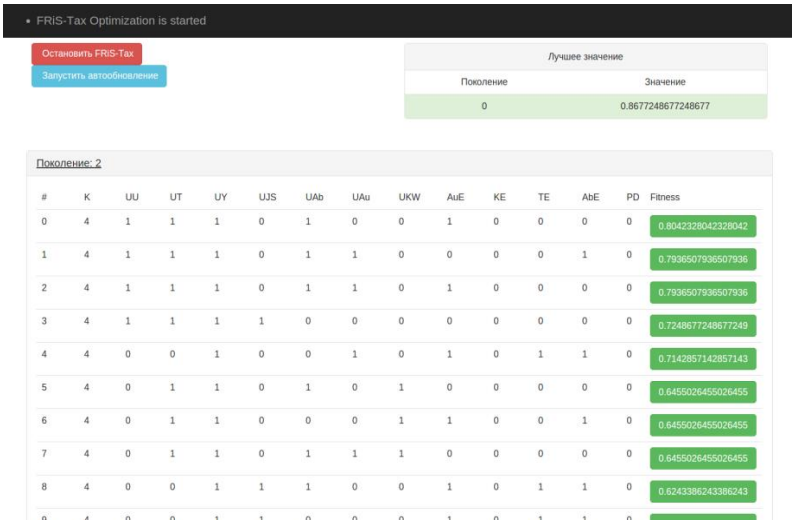


**Рис. 10.** Результаты выполнения генетического алгоритма, поколение 0

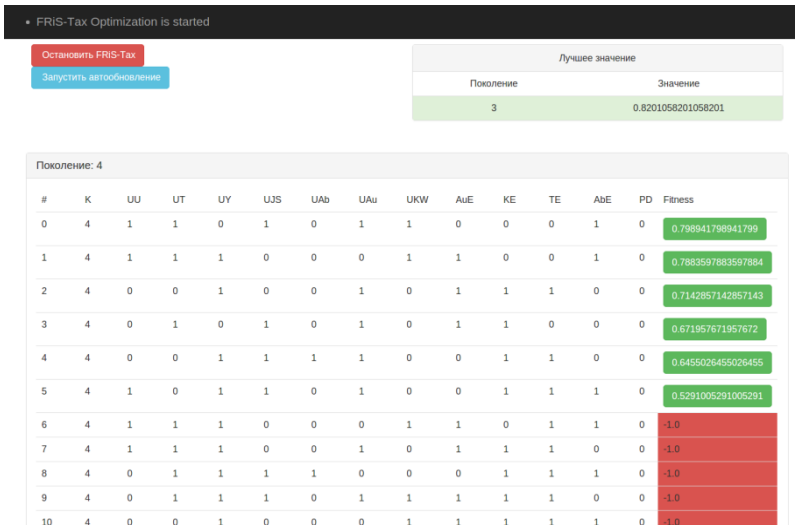


**Рис. 11.** Результаты выполнения генетического алгоритма, поколение 3

На рис. 12-13 представлены гистограммы полученных кластеров, где по горизонтальной оси указаны условные номера кластеров, по вертикальной оси – количество документов в кластере. Как видно из рисунков, разбиение на кластеры достаточно равномерное, причем доля кластеров с малым количеством элементов небольшая.

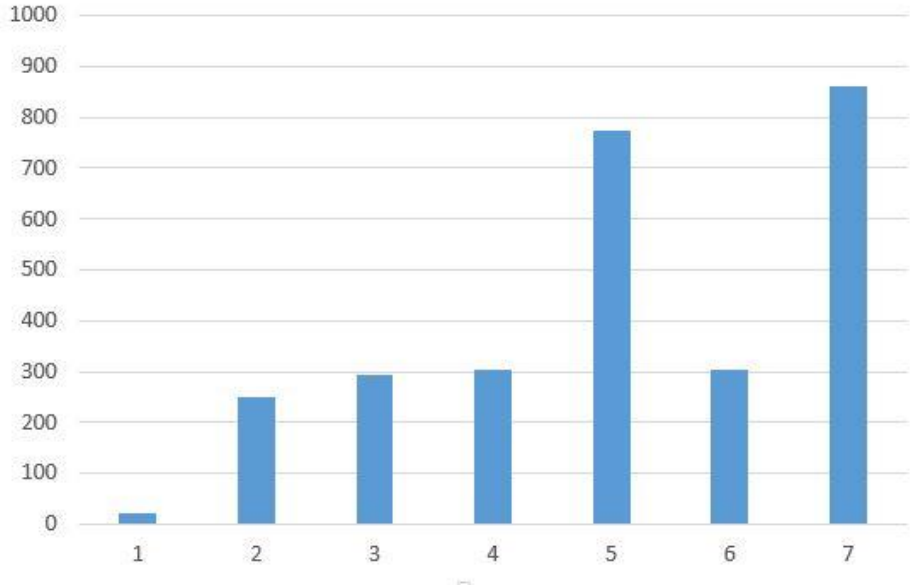


**Рис. 12.** Результаты выполнения кластеризации, количество кластеров равно 7

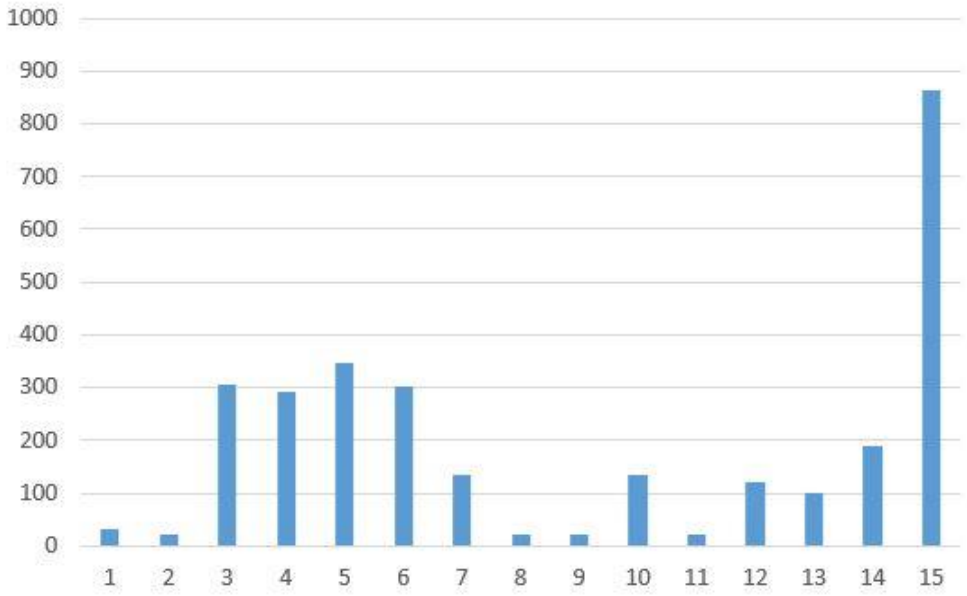


**Рис. 13.** Результаты выполнения кластеризации, количество кластеров равно 15

## 6. Заключение

Предложенная методика кластеризации текстовых документов позволяет выполнять процесс обработки на системах, состоящих более чем из одного вычислительного узла. Параллельное выполнение осуществляется на этапе предварительного анализа документов, включающем вычисление мер сходства между документами, а также непосредственно на этапе кластеризации. В работе приводятся количественные величины оценок времени выполнения при различном количестве вычислительных узлов. Оценка эффективности процесса при использовании параллельной реализации алгоритма на основе функции конкурентного сходства демонстрирует неоспоримый выигрыш в производительности.

## Литература

1. Борисова И. А., Загоруйко Н. Г. Функции конкурентного сходства в задаче таксономии // Материалы Всерос. конф. с международным участием «Знания – Онтологии – Теории» (ЗОНТ-07). Новосибирск, 2007. Т. 2. С. 67–76.
2. Барахнин В. Б., Нехаева В. А., Федотов А. М. О задании меры сходства для кластеризации текстовых документов // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2008. Т. 6, вып. 1. С. 3–9.
3. Загоруйко Н.Г., Барахнин В.Б., Борисова И.А., Ткачев Д.А. Кластеризация текстовых документов из электронной базы публикаций алгоритмом FRiS-Tax // Вычислительные технологии. - Т. 18, № 6, 2013. -С. 62-74.
4. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. – 2-е изд., испр. и доп. - М.: ФИЗМАТЛИТ, 2006. – 320 с.
5. Википедия: Расстояние Левенштейна. URL: https://en.wikipedia.org/wiki/Levenshtein_distance (дата обращения: 01.02.2016)
6. Andrei Z. Broder, Identifying and Filtering Near-Duplicate Documents / Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching Table of Contents, Pages: 1-10.
7. Оценка кластеризации. URL: http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html (дата обращения: 01.02.2016)
8. Bäck, Thomas, Evolutionary Algorithms in Theory and Practice (1996), p. 120, Oxford Univ. Press.
9. MPJ-Express. URL: http://mpj-express.org/ (дата обращения: 01.02.2016)
10. Processing Data with Java SE 8 Streams. URL: http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html (дата обращения: 01.02.2016)

# Development of parallel FRiS-Tax text document clustering algorithm based on MPI technology

M.E. Mansurova[1], V.B. Barakhnin[2,3], S.S. Aubakirov[1], Ye. Khibatkhanuly[1], Mussina A.B.[1]

Al-Farabi Kazakh National University[1], Institute of Computational Technologies of SB RAS[2], Novosibirsk State University[3]

This paper describes a parallel implementation of FRiS-Tax text document clustering algorithm. The clustering algorithm is based on an assessment of the similarity between objects in the competitive situation that leads to the concept of competitive similarity function (FRiS-function). As the scales for determination of the similarity measures are selected attributes of bibliographic description of documents. The parallelization is performed on the step of coefficient tuning in similarity measure formula of the genetic algorithm, as well as directly in step of clustering. The clustering algorithm is implemented on a high-performance MPJ Express platform. Quantitative evaluation of the execution time of the process is performed, clearly demonstrating the advantages of parallel implementation of the algorithm.

*Keywords:* clustering text documents, genetic algorithms, parallel algorithms.

## References

1. Borisova I.A., Zagoruiko N.G. Functions rival similarity in the problem of taxonomy // Proc. Conf. with international participation "Knowledge - Ontology - Theory" (Umbrella-07). Novosibirsk, 2007. T. 2. P. 67-76.
2. Barakhnin V.B., Nekhaeva V.A., Fedotov A.M. On the statement of the similarity measure for the clustering of text documents // Vestn. Novosib. state. Univ. Series: Information technology. 2008. T. 6, no. 1. S. 3-9.
3. Zagoruiko N.G., Barakhnin V.B., Borisova I.A., Tkachev D.A. Clustering of text documents from an electronic database of publications algorithm FRiS-Tax // Computational technologies. - T. 18, number 6, 2013. C. 62-74.
4. Gladkov L.A. Kureichik V.V., V.M. Kureichik Genetic algorithms / Ed. V.M. Kureichik. - 2nd ed., Rev. and add. - M .: FIZMATLIT, 2006. - 320 p.
5. Wikipedia: Levenshtein distance. URL: https://en.wikipedia.org/wiki/Levenshtein_distance (accessed: 01.02.2016)
6. Andrei Z. Broder, Identifying and Filtering Near-Duplicate Documents / Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching Table of Contents, Pages: 1-10
7. Evaluation of clustering. URL: http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html (accessed: 01.02.2016)
8. Bäck, Thomas, Evolutionary Algorithms in Theory and Practice (1996), p. 120, Oxford Univ. Press.
9. MPJ-Express. URL: http://mpj-express.org/ (accessed: 01.02.2016)
10. Processing Data with Java SE 8 Streams. URL: http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html (accessed: 01.02.2016)