Нечеткий алгоритм С-средних (*Fuzzy C-means*) - позволяет получить нечёткую кластеризацию больших наборов числовых данных, что позволяет корректно определять объекты на границах кластеров. Однако, выполнение данного алгоритма требует серьёзных вычислительных ресурсов, а также изначального задания количества кластеров. Кроме того, может возникнуть неоднозначность с объектами, удалёнными от центров всех кластеров.

1 Свойства и структура алгоритмов

1.1 Общее описание алгоритма

Алгоритм кластеризации Fuzzy C-Means (FCM) был предложен Дж. Данном в 1973 году [1] и доработан Дж. Бездеком в 1981 году [2]. В отличие от большинства существующих алгоритмов кластеризации, данный алгоритм является нечётким – каждый из объектов не входит однозначно в какой-либо кластер, а принадлежит всем кластерам с различными степенями принадлежности. Это даёт преимущества в качестве разбиения в случаях, когда кластеры находятся близко друг к другу, и большое число точек находится на их границах. Однако ценой такой нечёткости служат большие вычислительные затраты, чем у таких чётких алгоритмов, как Hard C-Means и K-Means, при сохранении таких их недостатков, как априорное определение числа кластеров и отсутствие гарантии глобальной оптимальности результата.

1.2 Математическое описание алгоритма

Исходные данные: массив объектов $X_k \in \R^n, k = \overline{1, M}$, число кластеров c, экспоненциальный вес $m \in [1, \infty)$, параметр останова $\varepsilon > 0$.

Вычисляемые данные: матрица разбиения F размера $M \times c$ (элементы $\mu_{ki} \in [0, 1]$, $\sum_{i=1}^{c} \mu_{ki} = 1$), центры кластеров V_i, расстояния D_{ki} между объектами и центрами кластеров.

Формулы метода (вычисляются последовательно на каждой итерации):

1. Уточнение центров кластеров по степеням принадлежности

$$V_i = \frac{\sum_{k=1}^{M} \mu_{ki}^m * X_k}{\sum_{k=1}^{M} \mu_{ki}^m}, i = \overline{1, c}$$

2. Расчёт расстояний между новыми центрами кластеров и точками данных

$$D_{ki} = \sqrt{\|X_k - V_i\|^2}, k = \overline{1, M}, i = \overline{1, c}$$

3. Пересчёт степеней принадлежности объектов кластерам

$$\mu_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ki}}{D_{kj}} \right)^{2/m-1}}, k = \overline{1, M}, i = \overline{1, c}$$

На каждой итерации алгоритма происходит уточнение элементов матрицы F. Выходом алгоритма служит матрица F, к которой алгоритм сходится. Факт того, что алгоритм сошёлся, устанавливается проверкой вида $\max\limits_{k=\overline{1,M}, i=\overline{1,c}} (|\mu_{ki} - \mu_{ki}^*|) < \varepsilon$ либо $\max\limits_{i=\overline{1,c}} (|V_i - V_i^*|) < \varepsilon$, где $\mu_{ki}^*(V_i^*)$ – значение $\mu_{ki}(V_i)$, вычисленное на предыдущей итерации.

1.3 Вычислительное ядро алгоритма

Вычислительное ядро алгоритма Fuzzy C-Means составляют шаги вычисления центров кластеров, расстояний между ними и точками данных и в особенности пересчёта матрицы степеней принадлежности точек данных.

При реализации алгоритма некоторые повторяющиеся вычисления могут быть устранены. Так, для шага вычисления центров кластеров величины μ_{ki}^m могут вычисляться однократно и умножаться на X_k при включении в сумму, записанную в числителе. Для шага вычисления расстояний между центрами кластеров и точками, операция взятия квадратного корня не является обязательной, так как в дальнейшем на шаге вычисления степеней принадлежности можно непосредственно использовать квадраты этих расстояний, и сумма в знаменателе будет приобретать вид:

$$\sum_{j=1}^{c} \left(\frac{D_{ki}^2}{D_{kj}^2} \right)^{1/m-1}$$

1.4 Макроструктура алгоритма

Алгоритм включает в себя три основных этапа – вычисление центров кластеров, вычисление расстояний между центрами кластеров и точками данных (включающее в себя макрооперации вычитания векторов и вычисления их норм) и пересчёт матрицы принадлежности.

1.5 Схема реализации последовательного алгоритма

Последовательность исполнения алгоритма следующая:

Инициализация происходит случайным заполнением матрицы принадлежности F с соблюдением условия нормировки $\sum\limits_{i=1}^{c} \mu_{ki} = 1$ и переходом к шагу 1, либо случайным определением центров кластеров V_i и переходом к шагу 2.

Далее осуществляются итерации, на каждой из которых производятся следующие вычисления:

1. $V_i = \frac{\sum_{k=1}^{M} \mu_{ki}^m * X_k}{\sum_{k=1}^{M} \mu_{ki}^m}, i = \overline{1,c}$
2. $D_{ki} = \sqrt{\|X_k - V_i\|^2}, k = \overline{1,M}, i = \overline{1,c}$
3. $\mu_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ki}}{D_{kj}}\right)^{2/m-1}}, k = \overline{1,M}, i = \overline{1,c}$

В конце каждой итерации проверяется условие останова вида $\max_{k=\overline{1,M}, i=\overline{1,c}} (|\mu_{ki} - \mu_{ki}^*|) < \varepsilon$, либо $\max_{i=\overline{1,c}} (|V_i - V_i^*|) < \varepsilon$, где $\mu_{ki}^*(V_i^*)$ – значение $\mu_{ki}(V_i)$, вычисленное на предыдущей итерации. Если условие не выполнено, осуществляется переход к шагу 1.

1.6 Последовательная сложность алгоритма

При кластеризации M объектов данных, представленных точками в $\R^n$, на c кластеров, алгоритм Fuzzy C-Means в последовательном варианте имеет вычислительную сложность – $O(c^2MI + cMnI)$, где I – число итераций. Если считать размерность данных n малой, то эта сложность сводится к $O(c^2MI)$. Основной частью алгоритма в этом случае является пересчёт матрицы принадлежностей, требующий вычисления cM сумм из c слагаемых на каждой итерации.

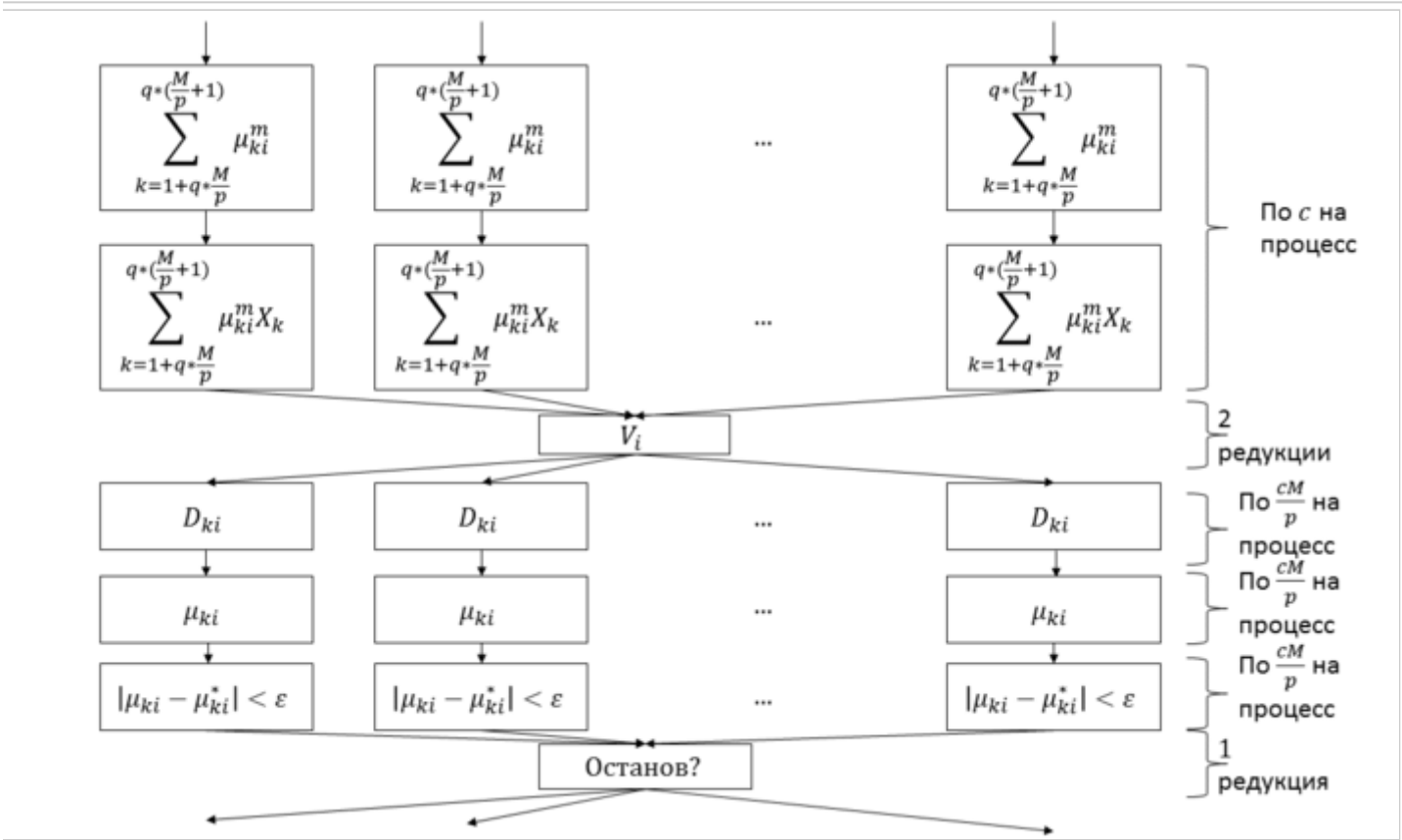
1.7 Информационный граф

Приведён граф единичной итерации алгоритма в параллельном оптимизированном варианте (метод распараллеливания взят в [3]):

Раздел 1.8 уточняет, соответствуют ли эти повторения однотипным ярусам или точкам данных, вычисления для которых можно распараллелить и далее.

Каждый процесс обладает следующими данными:

- координатами точек данных $X_k, k = \overline{1 + q * \frac{M}{p}, q * (\frac{M}{p} + 1)}$, где q – номер процесса,
- значениями μ_{ki} степени принадлежности для своих точек ($k = \overline{1 + q * \frac{M}{p}, q * (\frac{M}{p} + 1)}, i = \overline{1,c}$),
- координатами центров кластеров $V_i, i = \overline{1,c}$.



ая вершина данного графа соответствует операциям из алгоритма, указанным при помощи указанных формул. ый из p столбцов соответствует работе одного из процессов. Пометки справа указывают число повторений каждого ходе одной итерации.

Суммы $\sum_{k=1+q*\frac{M}{p}}^{q*(\frac{M}{p}+1)} \mu_{ki}^m$ и $\sum_{k=1+q*\frac{M}{p}}^{q*(\frac{M}{p}+1)} \mu_{ki}^m X_k$ вычисляются одновременно, поэтому значения μ_{ki}^m вычисляются по одному разу за итерацию. Таким образом, второй ярус операций на рисунке на деле не содержит операций возведения в степень. За счёт линейности большинства выражений относительно данных по точкам, процессы взаимодействуют только при редукции сумм, составляющих V_i. Всё остальное время каждый процессор работает только со своими $\frac{M}{p}$ точками. Это обеспечивает применимость алгоритма для больших M.

1.8 Ресурс параллелизма алгоритма

При распараллеливании по точкам исходных данных и условии останову по малому изменению степеней принадлежности выполнение одной итерации алгоритма FCM может быть разделено на следующие ярусы:

- c ярусов нахождения частичных сумм знаменателя (по $\frac{M}{p} - 1$ сложений, $\frac{M}{p}$ операций возведения в степень на процесс),
- c ярусов нахождения частичных сумм числителя (по $\frac{M}{p} - 1$ сложений, $\frac{M}{p}$ умножений на процесс),

- 2 редукции сумм и передач значений V_i процессам (получение $c(n+1)$ значений, cn делений),
- c ярусов вычисления расстояний до центров (по $n-1$ сложений, n вычитаний, n умножений), каждый процесс получает $\frac{M}{p}$ точек на обработку,
- c ярусов вычисления степеней принадлежности точек (по $c+1$ делений, с возведений в степень, $c-1$ сложений), каждый процесс получает $\frac{M}{p}$ точек на обработку,
- до c ярусов проверки условий останова (по 1 вычитанию, 1 сравнению), каждый процесс получает $\frac{M}{p}$ точек на обработку,
- 1 редукция для обмена статусом завершения.

Таким образом, при распараллеливании по точкам исходных данных при условии наличия в каждом узле достаточного объёма памяти для хранения всего массива координат центров кластеров высота и ширина ЯПФ алгоритма FCM равны соответственно $O(c^2I + cnI)$ и $O(M)$.

1.9 Входные и выходные данные алгоритма

Входные данные: массив векторов X_i, число кластеров c, экспоненциальный вес $m \in [1, \infty)$, параметр останова $\varepsilon > 0$.

Объём входных данных: Mn для входных векторов, 3 вспомогательных параметра.

Выходные данные: матрица принадлежности F (элементы $\mu_{ki} \in [0, 1]$). Условие нормировки: $\sum_{i=1}^{c} \mu_{ki} = 1$.

Объём выходных данных: cM.

1.10 Свойства алгоритма

В случае неограниченного распараллеливания по точкам данных (1 процесс на точку), отношение последовательной сложности алгоритма к параллельной пропорционально M.

Параметр m задаёт степень «размытости» кластеров. В отсутствие априорных данных его обычно берут равным 2. В предельном случае сведения параметра m к значению 1, кластеры становятся чёткими и алгоритм вырождается в алгоритм кластеризации K-Means.

Алгоритм недетерминирован, начальное положение кластеров задаётся случайно либо явно, либо опосредованно (через матрицу принадлежности). Алгоритм сходится к локальному экстремуму [4] и, таким образом, не гарантирует оптимальный результат при случайном выборе начальных значений.

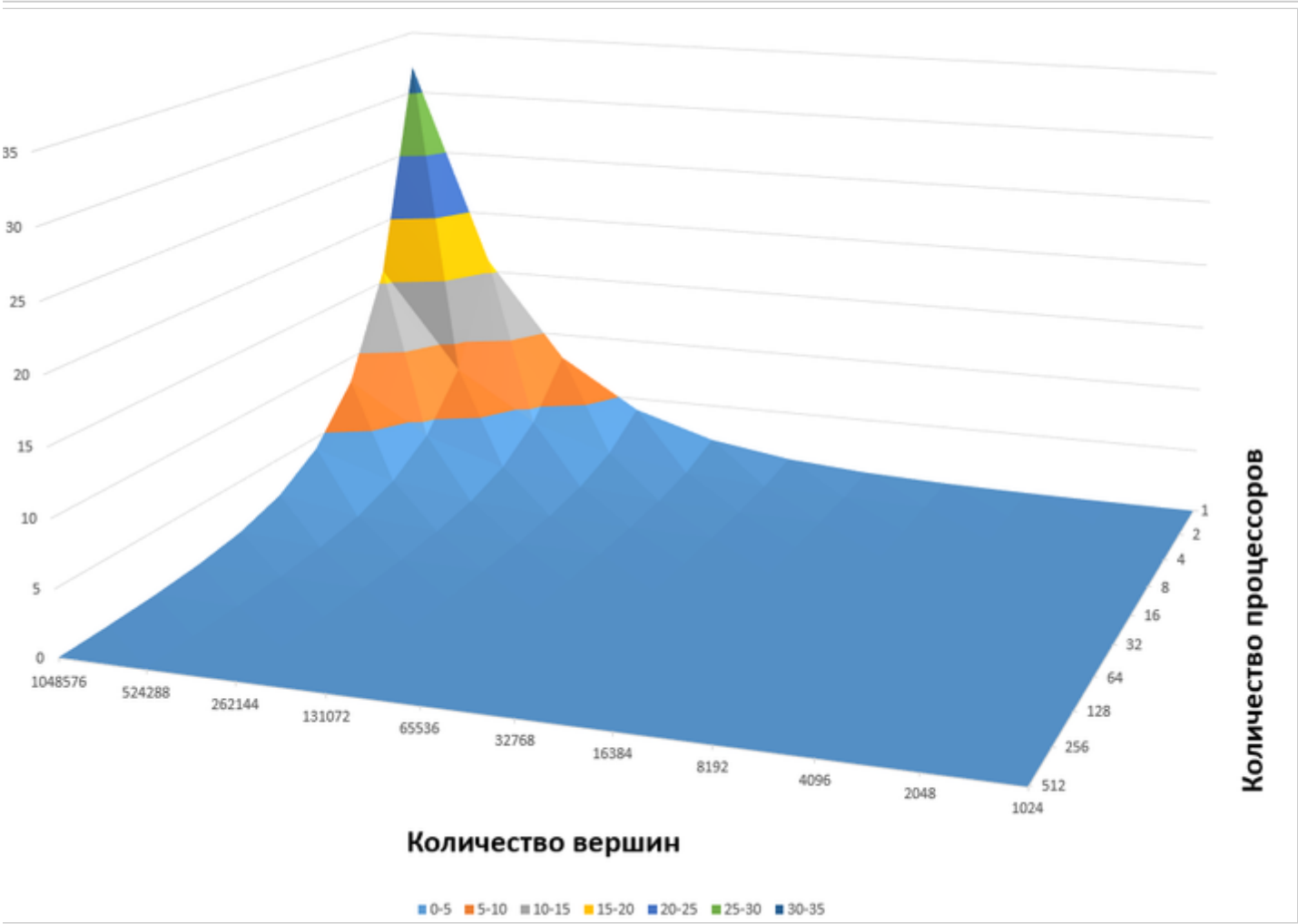
2 Программная реализация алгоритма

2.1 Особенности реализации последовательного алгоритма

2.2 Локальность данных и вычислений

2.3 Возможные способы и особенности параллельной реализации алгоритма

2.4 Масштабируемость алгоритма и его реализации

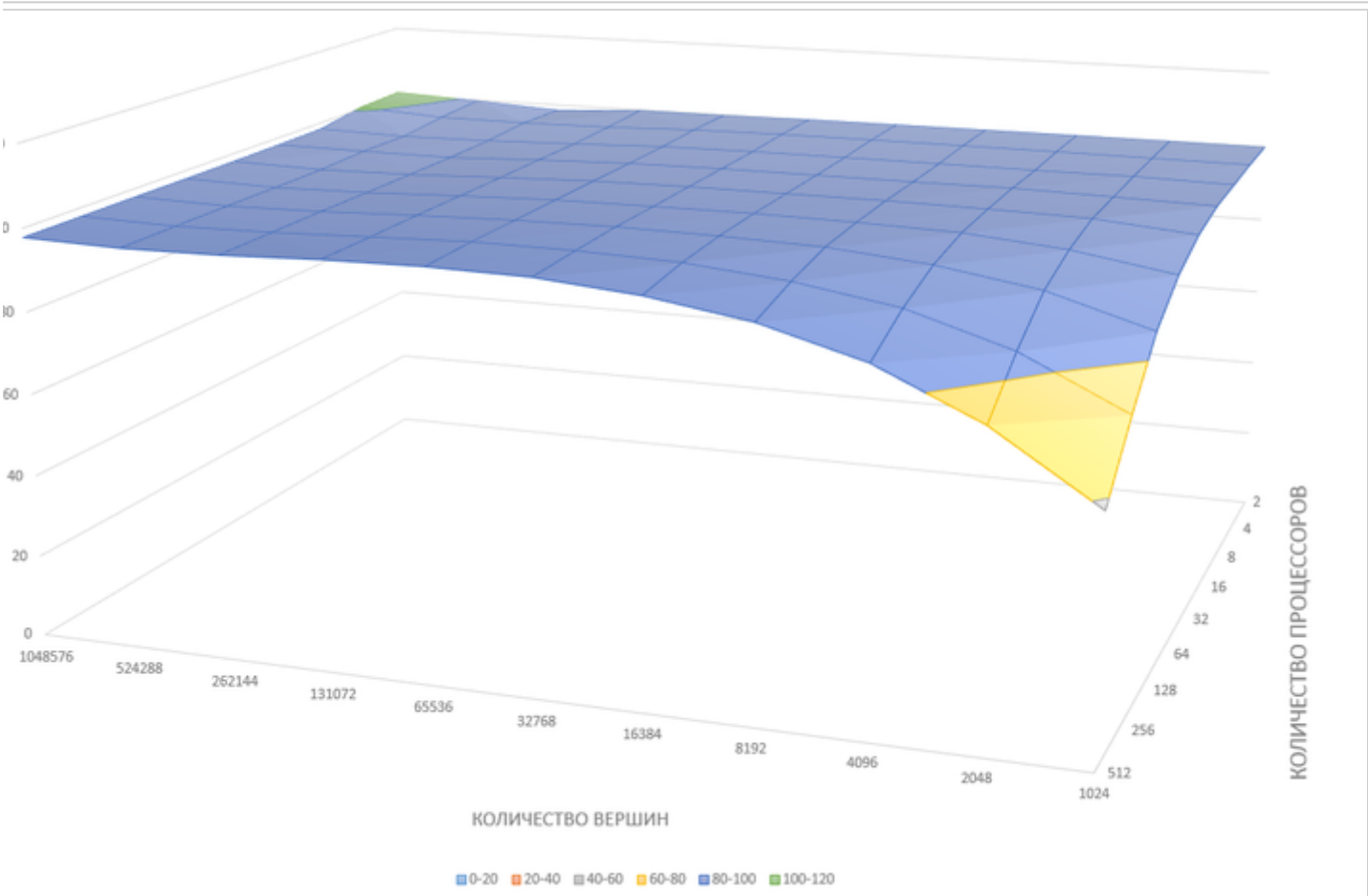


к зависимости времени выполнения итерации параллельного нечеткого алгоритма C-средних (Fuzzy C-means) от ества процессоров и количества кластеризуемых вершин

Исследование проведено на суперкомпьютере IBM Blue Gene/P[5]. Для исследования была использована последовательная реализация алгоритма и реализован её параллельный вариант (https://github.com/IllusiveMike/Parallel-Fuzzy-C-means/tree/patch-1)[6]. Алгоритм обладает плохо исследованной сходимостью, но объем вычислений на итерации не зависит от значений данных, поэтому данное исследование масштабируемости было произведено для единичной итерации.

Для упрощения экспериментов было решено использовать количество вершин для кластеризации и количество процессов равное степени двойки, поэтому для каждого эксперимента были выбраны значения:

- $2^v, v = \overline{0,9}$, для количества процессоров;



к эффективности распараллеливания нечеткого алгоритма C-средних (Fuzzy C-means) в зависимости от количества ссоров и количества кластеризуемых вершин.

- $2^w, w = \overline{10, 20}$, для количества кластеризуемых вершин.

Для одного набора значений проводилось 4 эксперимента и результаты усреднялись. При повторных запусках экспериментов значения выходили те же самые.

Полученные результаты говорят о хорошей масштабируемости реализации алгоритма. Для большинства экспериментов эффективность распараллеливания находится в пределах 100±3%. Высокая эффективность обуславливается тем, что данные равномерно распределены по процессорам и на каждой итерации требуется глобальная редукция всего $2c + 1$ значений (c значений числителей центроид кластеров, c значений знаменателей центроид кластеров и одно значение признака останова). Таким образом вычисления занимают значительно больше времени, чем взаимодействия между процессами, которые практически не приводят к простоям из-за хорошей балансировки. Спад эффективности при малом числе вершин и большом числе процессоров обуславливается тем, что с уменьшением числа вершин, обрабатываемых каждым процессором, растёт доля времени обмена данных между процессорами, не зависящего от числа вершин, как показано выше.

2.5 Динамические характеристики и эффективность реализации алгоритма

2.6 Выводы для классов архитектур

2.7 Существующие реализации алгоритма

- пакет MATLAB [7]
- Реализация алгоритма на POSTGRESQL [8]

3 Литература

1. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics. 3 (1973): 32–57
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981). ISBN 0-306-40671-3
3. Kwok, T., Smith, K., Lozano, S., Taniar, D.: Parallel Fuzzy c-Means Clustering for Large Data Sets (http://num-meth.srcc.msu.ru/zhurnal/tom_2012/pdf/v13r207.pdf), последнее обращение 25.10.2016
4. Höppner, F., Klawonn, F.: wolfenbuettel.de/~klawonn/Papers/hoeppnerklawonntfs03.pdf A Contribution to Convergence Theory of Fuzzy c-Means and Derivatives (https://public.fh), Дата последнего обращения: 13.10.2016
5. Описание вычислительного комплекса IBM Blue Gene/P (http://hpc.cs.msu.ru/bgp)
6. Параллельная реализация алгоритма Fuzzy C-Means (https://github.com/nikmedoed/Parallel-Fuzzy-C-means)
7. Документация пакета MATLAB, функция fcm (http://www.mathworks.com/help/fuzzy/fcm.html), последнее обращение 15.10.2016
8. Реализация алгоритма Fuzzy C-means на POSTGRESQL (http://num-meth.srcc.msu.ru/zhurnal/tom_2012/pdf/v13r207.pdf), последнее обращение 15.10.2016