

## Использование метаданных и норм элементов для оптимизации запросов

А. В. Кирикова<sup>1,2\*</sup>, В. И. Мунерман<sup>1</sup>

<sup>1</sup> ФГБОУ ВО «Смоленский государственный университет», г. Смоленск, Российская Федерация

Адрес: 214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4

<sup>2</sup> ФГКВУ ВО «Военная академия войсковой противовоздушной обороны Вооруженных Сил

Российской Федерации имени Маршала Советского Союза А. М. Василевского» Министерства

обороны Российской Федерации, г. Смоленск, Российская Федерация

Адрес: 214027, Российская Федерация, г. Смоленск, ул. Котовского, д. 2

\* anastasiakiricova52@gmail.com

### Аннотация

В виду стремительного развития информационных технологий, также соразмерно стремительно развиваются требования к информации и скорости ее обработке в этой сфере. Данные и запросы все более отличаются от оперирования простейшими типами. В статье рассматривается метод повышения эффективности решения задач поиска элементов среди широких таблиц – имеющих большое количество столбцов, что также можно рассматривать в разрезе задачи по соединению таблиц. Метод основан на специфическом индексировании, элементов, которое позволяет снять некоторые ограничения с данных таблицы и оптимизировав работу с большим количеством столбцов – тем самым расширив спектр применения такого подхода, не теряя плюсов классического индексирования. В основе предлагаемого индексирования используется вводимое понятия близкое к понятию нормы пространства – вес, позволяющее образовать на множестве элементов таблицы отношение эквивалентности, одновременно с чем снимая некоторые ограничения на использования индексирования с элементов – такие, как например, отношение порядка характеристик или ограничение на сами обрабатываемые типы данных таблицы. Также в статье рассматривается возможность группировки и распределения данных для распараллеливания методов обработки запросов к СУБД. Для реализации такой параллельной обработки данных используется простой принцип симметричного горизонтального распределения, на основе фактор-множества множества элементов таблицы построенного на базе вводимого понятия веса элемента. Он может позволить распределить элементы таблицы по вычислителям достаточно равномерно, притом, что будет сохранено отсутствие пересечения интересующих характеристик элемента между разными вычислителями, для ограждения от обмена памятью между процессорами.

**Ключевые слова:** параллельное программирование, BigData, поиск сложных структур, норма пространства, базы данных, распределенные вычисления, фактор-множество, вес элемента

**Конфликт интересов:** авторы заявляют об отсутствии конфликта интересов.

**Для цитирования:** Кирикова А. В., Мунерман В. И. Использование метаданных и норм элементов для оптимизации запросов // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 4. С. 767-773. doi: <https://doi.org/10.25559/SITITO.18.202204.767-773>

© Кирикова А. В., Мунерман В. И., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.  
The content is available under Creative Commons Attribution 4.0 License.



## Using Metadata and Element Norms to Optimize Queries

A. V. Kirikova<sup>a,b\*</sup>, V. I. Munerman<sup>a</sup>

<sup>a</sup> Smolensk State University, Smolensk, Russian Federation

Address: 4 Przhevalsky St., Smolensk 214000, Russian Federation

<sup>b</sup> Russian Federation Armed Forces Army Air Defence Military Academy named after Marshal of the Soviet Union A. M. Vasilevsky, Smolensk, Russian Federation

Address: 2 Kotovsky St., Smolensk 214027, Russian Federation

\* anastasiakiricova52@gmail.com

### Abstract

In view of the rapid development of information technologies, the requirements for information and the speed of its processing in this area are also developing proportionally rapidly. Data and queries are increasingly different from operating with the simplest types. The article discusses a method for improving the efficiency of solving problems of searching for elements among wide tables – having a large number of columns, which can also be considered in the context of the task of connecting tables. The method is based on specific indexing of elements, which allows you to remove some restrictions from the table data and optimize work with a large number of columns – thereby expanding the range of applications of this approach, without losing the advantages of classical indexing. The proposed indexing is based on an introduced concept close to the concept of space-weight norms, which allows forming an equivalence relation on a set of table elements, at the same time removing some restrictions on the use of indexing from elements – such as, for example, the ratio of the order of characteristics or the restriction on the processed data types of the table themselves. The article also discusses the possibility of grouping and distributing data to parallelize the methods of processing DBMS queries. To implement such parallel data processing, a simple principle of symmetrical horizontal distribution is used, based on the factor-set of the set of elements of the table built on the basis of the introduced concept of element weight. It can allow you to distribute the elements of the table among the calculators fairly evenly, despite the fact that there will be no intersection of the characteristics of the element of interest between different calculators, to protect against memory exchange between processors.

**Keywords:** parallel programming, BigData, search for complex structures, space norm, databases, distributed computing, factor-set, element weight

**Conflict of interests:** The authors declare no conflict of interest.

**For citation:** Kirikova A. V., Munerman V.I. Using Metadata and Element Norms to Optimize Queries. *Modern Information Technologies and IT-Education*. 2022;18(4):767-773. doi: <https://doi.org/10.25559/SITITO.18.202204.767-773>



## Введение

В статье рассматривается возможность применения распределения, а также понятия близкого к понятию нормы элемента для дополнительной оптимизации последовательности поисковых запросов к базам данным (Big Data). Современные сервисы зачастую сталкиваются с проблемами чрезвычайной нагрузки за счет следующих факторов:

- Данные в обработке имеют тенденцию к усложнению своей структуры;
- Количество данных постоянно растет;
- Общность данных снижается что влечет за собой снижение эффективности универсальной обработки данных для каждого случая;
- Вместе с усложнением структуры самих данных повышается сложность запроса, при обращении к этим данным.

Сейчас подобные проблемы решаются как правило экстенсивными мерами. По большей мере прибегают к увеличению числа процессоров и прямому разбиению данных. Однако, данные меры являются не эффективными с точки зрения затрат, а также имеют строгие ограничения в своем применении для разных операций [1-11]. Так увеличение числа кластеров вычисления в какой-то момент неизбежно приведет к перегрузке шины обмена данными и прочим побочным эффектам, которые будут снижать эффективность избранных методов оптимизации подобного рода<sup>1</sup>.

## Цель исследования

Несмотря на многие экстенсивные методы оптимизации операции поиска, далеко не всегда можно счесть их решающими проблему растущих данных, в то время как интенсивные алгоритмы – такие как индексирование, часто страдают от большого количества побочных затрат, например, памяти а так же накладывают определенные условия на обрабатываемую выборку данных. Сейчас требуются новые решения, позволяющие справиться с увеличением объема обрабатываемой информации, новыми типами и унификацией работы с большинством разных видов данных. Целью исследования является предложения способов оптимизации поисковых запросов на основе индексации при помощи введения новых метаданных, таких как вес, а также и кластеризованная цепочка элементов базы на основе веса.

Предлагаемый метод по оптимизации поисковых запросов среди широких таблиц основан на введении понятия веса элемента, разбиении таблиц, которое в свою очередь основывается на симметричном горизонтальном распределении и кластеризации элементов на основе отношения, заданного при помощи весовой характеристики. Такое направление выбрано как попытка справиться, в основном, с чрезмерной размерностью данных при современной обработке больших массивов информации<sup>2</sup>. При таком подходе предполагается, что все

элементы таблицы будут распределены и кластеризованы на основе вводимой характеристики веса. В дальнейшем изложении предполагается что таблицы имеют большой объем и не малое число столбцов – для эффективности вводимых понятий, иначе их обработке не представляла бы сильного интереса.

## Введение понятия веса

Для оптимизации поиска, разбиения по вычислителям и кластеризации данных нужно ввести для каждого элемента характеристику, которую назовем весом элемента –  $w$ . Предположим, что имеется таблица со следующими столбцами некоторых числовых типов:

$$(a_1, a_2, a_3, \dots, a_n)$$

Введем из пространства значений каждого столбца инъективное отображение на числовой отрезок, данное отображение можно задавать любым удобным способом, во всяком случае такое отображение можно задать через подсчет занимаемой памяти элементом. Его можно строить естественно – например для чисел, или через введения хеш-функций сверток, даже через перечисление элементов – хотя последний вариант наименее эффективен.

Пусть каждое множество значений данных отображений для соответствующего столбца столбец имеет пару – максимальное и минимальное значение (вообще для типа столбца):

$$((A, A')_1, (A, A')_2, (A, A')_3, \dots, (A, A')_n)$$

Обозначим функцию  $m(x_i)$  – как функцию для  $i$ -ой координаты элемента  $x$ , задающую такие отображения для столбцов элементов таблицы. Причем сам состав функций не имеет значения для совокупности столбцов. Они могут быть выбраны разными способами, с условием того, что в общем для таблицы они будут одинаковы при поисковых запросах и при индексировании таблицы.

Будем говорить для  $i$ -ой координаты элемента  $x$ , или значения  $i$ -ого столбца элемента  $x$ , что ее удельный вес равен единице, если для  $x_i$  выполняется условие

$$m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)}$$

Иначе считать нулем. Иными словами, если по  $i$ -ой координате  $x$  находится несколько меньше первых трех четвертей от минимума по данному столбцу. То есть, если обозначить функцию веса  $w(x_i)$ , весом элемента  $x$  назовем следующую сумму:

$$W(X) = \sum_{i=1}^n w(x_i)$$

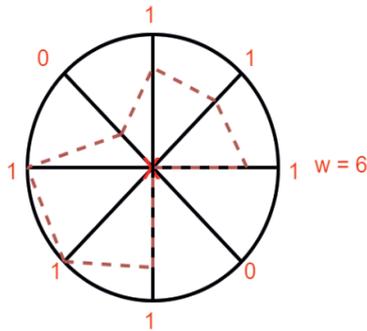
$$w(x_i) = 1, \text{ если } m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)},$$

где  $x_i$  - координаты (значения  $x$  в  $i$ -тых столбцах) элемента  $x$ .

<sup>1</sup> Пюрова Т. А., Скворцов С. В. Технология CUDA и параллельные вычисления на графических процессорах // Информатика и прикладная математика. 2015. № 21. С. 163-166. URL: <https://elibrary.ru/item.asp?id=25030523> (дата обращения: 28.09.2022); Захаров В. Н., Мунерман В. И., Самойлова Т. А. Параллельные методы вывода ассоциативных правил в технологиях in-database и in-memory // CEUR Workshop Proceedings. 2017. Т. 2064. С. 219-225. URL: <https://ceur-ws.org/Vol-2064/paper26.pdf> (дата обращения: 28.09.2022).

<sup>2</sup> Мунерман В. И. Опыт массовой обработки данных в облачных системах (на примере Windows Azure) // Системы высокой доступности. 2014. Т. 10, № 2. С. 3-8. URL: <https://elibrary.ru/item.asp?id=21649609> (дата обращения: 28.09.2022).





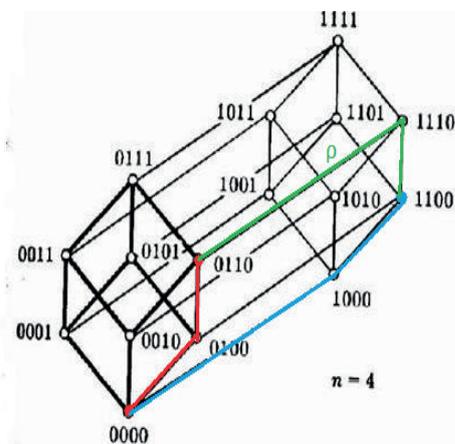
Р и с. 1. Наглядное изображение координат элемента  
Fig. 1. Visual representation of element coordinates

Данную характеристику необходимо найти для каждого элемента базы, предпочтительно так же ее хранение в качестве метаданных для поиска. Очевидно, что для всех элементов таблицы подобную характеристику можно рассчитать, а также не сложно показать то, что у одинаковых элементов будут одинаковые веса, эти два факта дают нам право воспринимать данную характеристику как естественную перцептивную хеш-функцию, на основе которой уже можно повышать эффективность поиска и операции соединения таблиц, путем отбрасывания элементов с другим весом из нужной выборки. Более того если необходим поиск по части столбцов среди элементов таблицы или же по сочетанию условий вида  $x_1 = a_1, x_2 = a_2, x_3 = a_3, \dots$

где  $a$  – поисковый образ, на данном этапе можно использовать правило, следующее из неравенства треугольника, поскольку каждый вес можно представить в качестве расстояния от нулевого элемента для таблицы – элемент, лежащий посреди каждого отрезка:

$$|W_1 - W_2| \leq \rho(X_1, X_2),$$

где  $\rho$  – понимается расстояние Хемминга по весам координат. Иными словами, если модуль разности весов равен определенной величине, расстояние Хемминга между весами координат элементов отличаются не более чем на эту величину.



Р и с. 2. Неравенство треугольника на булевом кубе  
Fig. 2. Triangle Inequality on Boolean Cube

Так для очередного поиска возможным представляется найти вес поискового образа и первыми для сравнения отбирать элементы с ближайшим весом, поскольку они в большей степени близки к поисковому образу по большей части столбцов – а значит имеют больший шанс на совпадение с условиями поиска.

## Кластеризация на основе веса

Кроме того, очевидно, что, понятие веса позволяет строить отношение эквивалентности на множестве элементов – путем сравнения их весов. При помощи этого можно разбивать множество элементов по данному признаку на классы эквивалентности, таким образом порождая фактор-множество на множестве элементов базы, что позволяет построить на множестве таблицы горизонтально симметричное распределение, как нулевой шаг при обработке трудных запросов по типу цепочек из операций Join по столбцам входящих в понятие веса.

Для осуществления более равномерного распределения по мощности множеств можно пользоваться простым принципом: в отсортированном по мощности списке из классов эквивалентности брать для одного вычислителя первые и последние  $k$  классов, для второго вычислителя следующие  $k$  элементов с обоих концов и так далее. Так на каждом вычислителе получим не пересекающиеся по элементам (по равным элементам) множества на каждом вычислителе. А поскольку вычисление веса не имеет большой алгоритмической сложности и при этом позволяет разделить множество на подобные по элементам подмножества достаточно равномерно, данное разбиение может существенно повысить эффективность распределенной обработки тяжелых запросов – путем эффективного распределения без возможных пересечений данных [12]. Так же при условии индексирования, описанного ниже, тактовые весовые характеристики представляется возможным хранить и более того достаточно легко менять при изменении координат – значений столбцов.

Основным применением вводимого понятия веса является кластеризация и индексация базы данных для повышения эффективности выполнения поиска по нескольким характеристикам – столбцам. При классической индексации (использование b-tree) на множество обычно накладывается ограничение наличия отношения порядка для характеристик, а также присутствует сильное увеличение расхода памяти при увеличении числа столбцов. При использовании же менее зависящих от сравнения (например sp-gist) индексов возникает зачастую необходимость хранения объектов большей размерности чем элементы индексируемой таблицы. Для осуществления оптимизации поискового запроса по элементам сложной структуры (имеющим множество характеристик) применим введенное понятие веса, для отбрасывания ограничений, накладываемых индексным методом и понижения размерности элементов исходной таблицы [13-19].

Будем разбивать множество элементов таблицы по их весу на подмножества, далее получившиеся подмножества будут разбиты вновь до момента, когда в результирующих множествах не останется малого числа элементов или по достижении большого числа глубины. Каждый шаг при этом назовем уровнем разбиения.



На каждом уровне кластеризации функция веса для координат и соответственно для элемента в целом будет видоизменена с учетом глубины кластеризации. А именно положим, что при кластеризации на каждом  $j$ -ом уровне будем говорить, что координата имеет вес 1 тогда, когда

$$m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)^j}$$

То есть для первого уровня, как и было сказано формула, определяющая вес элемента таблицы на первом уровне, будет иметь следующий вид:

$$m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)}$$

Далее на втором уровне, когда изначальное множество будет разбито на подмножества, для каждого из этих подмножеств вес одной координаты будет равен единице если

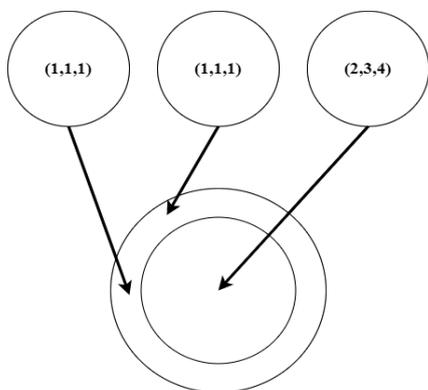
$$m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)^2}$$

И так далее – в общем виде можно сказать, что условие получения координатой веса на каждом уровне усложняется относительно общего отрезка возможных значений, что склоняет кластеризацию к разумному времени и количеству вложенных множеств.

Таким образом общую формулу веса  $W(X)$  элемента  $X$  на  $j$ -ом уровне для таблицы из  $n$  столбцов можно записать в следующем виде:

$$W_j(X) = \sum_{i=1}^n w_j(x_i), \text{ где}$$

$$w_j(x_i) = 1, \text{ если } m(x_i) \leq \frac{(A'_i + A_i)}{(e/2)^j}, \text{ иначе } 0$$

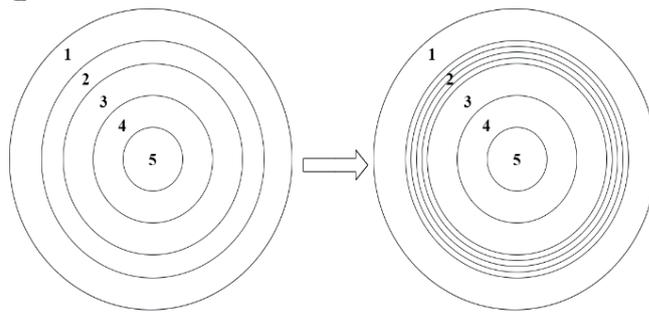


Р и с. 3. Пример распределения  
F i g. 3. Distribution example

Можно видеть, что при таком разбиении и такой введенной функции веса для каждого шага, в каждом множестве с ростом числа уровней будет гарантированно меньше элементов. Так же из того, что для одного уровня функция веса одинаковая для всех элементов и однозначно определяет его вес, зависящий таким образом только от значений соответствующих столбцов, и из сказанного выше следует что, одинаковые элементы будут находится на каждом шаге в одинаковых результирующих множествах.

Так же, как и в случае применения классического индекса можно видеть, что в среднем для каждого шага количество элементов будет каждый раз уменьшаться в количестве возможных весов. Таким образом для первого шага в каждом множестве (после первой кластеризации) будет содержаться  $\frac{n}{L}$ , элементов, где  $L$  – число возможных весов от 0 до  $L$ ,

Для второго шага (после кластеризации первых кластеризованных множеств), это число в среднем будет составлять  $\frac{n}{L^2}$ , элементов



Р и с. 4. Вложенная кластеризация  
F i g. 4. Nested Clustering

А для достижения состояния в один элемент в каждом из результирующих множеств необходимо в среднем:

$$k = \log_2(n-1), \text{ шагов}$$

Для очередного поиска элементов по заданным столбцам необходимо найти вес поискового образа – после чего определить для него вложенное множество на каждом уровне и повторить до тех пор пока данные элемент не попадет в результирующее множество, поскольку элементы однозначно разбиты функциями веса на каждом шаге, вместе с поисковым образом обязательно будут либо идентичные либо близкие по Хеммингу к нему элементы базы данных.

Поскольку в среднем разбиение по весам достаточно равномерно по заполненной базе – указанные характеристики верны для предлагаемой схемы, что можно проверить экспериментально [20-25]. Так предлагаемая схема разбиения сохраняет преимущества индексирования пространства (например b-tree) при оптимизации поиска и вывода его на логарифмическую сложность. Однако вместе с этим нужно отметить, что для индексирования по большому количеству столбцов – обычный индекс типа b-tree имеет вложенную структуру для каждого столбца в листьях предыдущего – а значит кратно увеличивает расходы памяти на хранение, в это же время для предложенной схемы число столбцов при разбиении не сказывается на памяти, затраченной при хранении индекса.

Также нужно отметить что для построения классического b-tree элементы каждого столбца из обрабатываемых накладываются ограничение наличия отношения хотя бы не строгого порядка для характеристик, что затрудняет построение индекса для некоторых типов данных на множестве которых такое отношение либо не построено либо его расчет (сравнение) весьма сложно с точки зрения вычислительной сложности, либо в тех случаях когда последовательность индексирования столбцов важна – предлагаемая же схема не страдает от



подобного ограничения и позволяет работать с любыми типами данных, поскольку хотя бы уже по затрачиваемой памяти можно определить максимальные и минимальные значения столбцов любого типа, либо же найти характеристику отражающую вес координаты иным способом задав его либо естественно, например для чисел, либо через простые в исполнении функции, что существенно расширяет спектр возможного применения предлагаемого метода.

Кроме того, поскольку границы столбцов – константные значения, и потому, что такое разбиение не зависит от порядка столбцов и не требует отношения порядка для элементов, переконструирование подобного решения намного проще за счет того, что вес элемента на каждом уровне, а значит и его положение в дереве зависит только от самого элемента, и не связано с общим разбиением, по тем же причинам намного проще выглядит изменение порядка столбцов или добавление в индексирование нового столбца.

## Заключение

Приведенный выше подход позволяет не только эффективно находить равные или схожие последовательности ключей для

операций поиска и сравнения, но и выстроить распределенную систему для оптимизации других более сложных операций таких как join. Предлагаемый метод использования метаданных, в частности, понятия веса, на основе “заполнения” столбца для элемента обладает следующими преимуществами:

- Возможность построения индексной структуры для оптимизации поиска данных без ограничений на элементы;
- Возможность построения индексной структуры без неоптимальных затрат памяти по большому количеству характеристик элемента;
- Возможность эффективного разбиения данных перед трудными запросами для распределения по разным процессорам без пересечения данных;

Использование метаданных на основе некоторого множества характеристик в рамках предлагаемого метода наиболее эффективно в рамках усложнения структуры обрабатываемых данных и усложнение запросов к таким данным. Конкретно предлагаемое решение оставляет за собой основные черты индексных методов, однако во многом решает проблему увеличения расходования памяти за счет увеличения числа столбцов и служения их типов, что существенно расширяет возможности по его применению.

## References

- [1] Das S., Grbic M., Ilic I., Jovandic I., Jovanovic A., Narasayya V.R., Radulovic M., Stikic M., Xu G., Chaudhuri S. Automatically Indexing Millions of Databases in Microsoft Azure SQL Database. In: Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19). New York, NY, USA; Association for Computing Machinery; 2019. p. 666-679. doi: <https://doi.org/10.1145/3299869.3314035>
- [2] Haynes D., Ray S., Manson S.M., Soni A. High performance analysis of big spatial data. In: 2015 IEEE International Conference on Big Data (Big Data). Santa Clara, CA, USA: IEEE Computer Society; 2015. p. 1953-1957. doi: <https://doi.org/10.1109/BigData.2015.7363974>
- [3] Abdel-Basset M., Manogaran G., Abdel-Fatah L., Mirjalili S. An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems. *Personal and Ubiquitous Computing*. 2018;22(5-6):1117-1132. doi: <https://doi.org/10.1007/s00779-018-1132-7>
- [4] Chamoso P., Rivas A., Sánchez-Torres R., Rodríguez S. Social computing for image matching. *PLoS ONE*. 2018;13(5):e0197576. doi: <https://doi.org/10.1371/journal.pone.0197576>
- [5] Dodonov A., Mukhin V., Zavgorodnii V., Kornaga Y., Zavgorodnya A., Mukhin O. Method of Parallel Information Object Search in Unified Information Spaces. *International Journal of Computer Network and Information Security*. 2021;13(4):1-13. doi: <https://doi.org/10.5815/ijcnis.2021.04.01>
- [6] Levin N.A., Munerman V.I. Models of Big Data Processing in Massively Parallel Systems. *Highly available systems*. 2013;9(1):035-043. Available at: <https://elibrary.ru/item.asp?id=18928468> (accessed 28.09.2022).
- [7] Alam K.S., Shishir T.A., Azharul Hasan K.M. Efficient Partitioning Algorithm for Parallel Multidimensional Matrix Operations by Linearization. In: Senjyu T., Mahalle P.N., Perumal T., Joshi A. (eds.). Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies. Vol. 195. Singapore: Springer; 2021. p. 141-149. doi: [https://doi.org/10.1007/978-981-15-7078-0\\_13](https://doi.org/10.1007/978-981-15-7078-0_13)
- [8] Chen Y., Li K., Yang W., Xiao G., Xie X., Li T. Performance-Aware Model for Sparse Matrix-Matrix Multiplication on the Sunway TaihuLight Supercomputer. *IEEE Transactions on Parallel and Distributed Systems*. 2019;30(4):923-938. doi: <https://doi.org/10.1109/TPDS.2018.2871189>
- [9] Pushpa Rani Suri, Sudesh Rani. A New Classification for Architecture of Parallel Databases. *Information Technology Journal*. 2008;7(7):983-991. doi: <https://doi.org/10.3923/itj.2008.983.991>
- [10] Zaki M.J., Parthasarathy S., Ogihara M. Parallel Algorithms for Discovery of Association Rules. *Data Mining and Knowledge Discovery*. 1997;1(4):343-373. doi: <https://doi.org/10.1023/A:1009773317876>
- [11] Wajszczyk B., Gruszka I.M. Analysis of possibilities to increase the efficiency of the relative database management system using the methods of parallel processing. In: Kaniewski P., Matuszewski J. (eds.) Proceedings of SPIE. Vol. 11442. Radioelectronic Systems Conference 2019. Article number: 1144215. SPIE; 2020. doi: <https://doi.org/10.1117/12.2565744>
- [12] Gorokhovatskiy V.A., Gorokhovatskiy A.V., Peredrii Ye.O. Hashing of structural descriptions at building of the class image descriptor; computing of relevance and classification of the visual objects. *Telecommunications and Radio Engineering*. 2018;77(13):1159-1168. doi: <https://doi.org/10.1615/TelecomRadEng.v77.i13.40>



- [13] Kirikova A., Mironov A. Using Metadata-indexing to Improve the Efficiency of Complex Operations. In: 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). St. Petersburg, Moscow, Russia; IEEE Computer Society; 2021. p. 2124-2127. doi: <https://doi.org/10.1109/EIConRus51938.2021.9396274>
- [14] Kirikova A., Mironov A., Munerman V. The Method of Composition Hash-functions for Optimize a Task of Searching Images in Dataset. In: 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). St. Petersburg and Moscow, Russia; IEEE Computer Society; 2020. p. 1983-1986. <https://doi.org/10.1109/EIConRus49466.2020.9038919>
- [15] Munerman V., Munerman D. Realization of Distributed Data Processing on the Basis of Container Technology. In: 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Saint Petersburg and Moscow, Russia; IEEE Computer Society; 2019. p. 1740-1744. doi: <https://doi.org/10.1109/EIConRus.2019.8656766>
- [16] Munerman V., Munerman D., Samoiloa T. The Heuristic Algorithm For Symmetric Horizontal Data Distribution. In: 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). St. Petersburg, Moscow, Russia; IEEE Computer Society; 2021. p. 2161-2165. doi: <https://doi.org/10.1109/EIConRus51938.2021.9396510>
- [17] Lomet D. The evolution of effective B-tree: Page organization and techniques: A personal account. *ACM SIGMOD Record*. 2001;30(3):64-69. doi: <https://doi.org/10.1145/603867.603878>
- [18] Rodríguez-Mazahua N., Rodríguez-Mazahua L., López-Chau A., Alor-Hernández G., Machorro-Cano I. Decision-Tree-Based Horizontal Fragmentation Method for Data Warehouses. *Applied Sciences*. 2022;12(21):10942. doi: <https://doi.org/10.3390/app122110942>
- [19] Graefe G. Modern B-Tree Techniques. *Foundations and Trends in Databases*. 2011;3(4):203-402. doi: <https://doi.org/10.1561/1900000028>
- [20] Lvovich I., Lvovich Y., Preobrazhenskiy A., Choporov O. Modeling and Optimization of Processing Large Data Arrays in Information Systems. In: 2021 International Conference on Information Technology and Nanotechnology (ITNT). Samara, Russian Federation: IEEE Computer Society; 2021. p. 1-5. doi: <https://doi.org/10.1109/ITNT52450.2021.9649229>
- [21] Monga V., Evans B.L. Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs. *IEEE Transactions on Image Processing*. 2006;15(11):3452-3465. doi: <https://doi.org/10.1109/TIP.2006.881948>
- [22] Sridhar R., Chandrasekaran M., Sriramya C., Page T. Optimization of heterogeneous Bin packing using adaptive genetic algorithm. *IOP Conference Series: Materials Science and Engineering*. 2017;183(1):012026. doi: <https://doi.org/10.1088/1757-899X/183/1/012026>
- [23] Syrotkina O., Aleksieiev M., Moroz B., Matsiuk S., Shevtsova O., Kozlovskiy A. Mathematical Methods for optimizing Big Data Processing. In: 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). Deggendorf, Germany: IEEE Computer Society; 2020. p. 170-176. doi: <https://doi.org/10.1109/ACIT49673.2020.9208940>
- [24] Zobel J., Moffat A., Sacks-Davis R. An Efficient Indexing Technique for Full Text Databases. In: Proceedings of the 18th International Conference on Very Large Data Bases (VLDB '92). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1992. p. 352-362.
- [25] Zakharov V., Kirikova A., Munerman V., Samoiloa T. Architecture of Software-Hardware Complex for Searching Images in Database. In: 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Saint Petersburg and Moscow, Russia; IEEE Computer Society; 2019. p. 1735-1739. doi: <https://doi.org/10.1109/EIConRus.2019.8657241>

*Поступила 28.09.2022; одобрена после рецензирования 23.11.2022; принята к публикации 04.12.2022.  
Submitted 28.09.2022; approved after reviewing 23.11.2022; accepted for publication 04.12.2022.*

#### Об авторах:

**Кирикова Анастасия Викторовна**, аспирант физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4); преподаватель, ФГКВУ ВО «Военная академия войсковой противовоздушной обороны Вооруженных Сил Российской Федерации имени Маршала Советского Союза А. М. Василевского» Министерства обороны Российской Федерации (214027, Российская Федерация, г. Смоленск, ул. Котовского, д. 2), **ORCID: <https://orcid.org/0000-0002-0861-8847>**, [anastasiakiricova52@gmail.com](mailto:anastasiakiricova52@gmail.com)

**Мунерман Виктор Иосифович**, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, [vimoon@gmail.com](mailto:vimoon@gmail.com)

*Все авторы прочитали и одобрили окончательный вариант рукописи.*

#### About the authors:

**Anastasia V. Kirikova**, Postgraduate Student of the Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation); Teacher, Russian Federation Armed Forces Army Air Defence Military Academy named after Marshal of the Soviet Union A. M. Vasilevsky (2 Kotovsky St., Smolensk 214027, Russian Federation), **ORCID: <https://orcid.org/0000-0002-0861-8847>**, [anastasiakiricova52@gmail.com](mailto:anastasiakiricova52@gmail.com)

**Victor I. Munerman**, Associate Professor of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand. Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, [vimoon@gmail.com](mailto:vimoon@gmail.com)

*All authors have read and approved the final manuscript.*

