

## Модифицированный метод обработки больших разреженных неструктурированных матриц на реконфигурируемых вычислительных системах

**И. И. Левин**

Южный федеральный университет,  
Институт компьютерных технологий и информационной безопасности,  
кафедра интеллектуальных и многопроцессорных систем,  
Таганрог, Российская Федерация  
ORCID: 0000-0002-1704-5016, e-mail: [iilevin@sfedu.ru](mailto:iilevin@sfedu.ru)

**А. В. Подопригора**

Научно-исследовательский центр супер-ЭВМ и нейрокомпьютеров,  
Таганрог, Российская Федерация  
ORCID: 0009-0003-4099-0088, e-mail: [podoprighora@superevm.ru](mailto:podoprighora@superevm.ru)

**Аннотация:** При обработке матриц большой размерности с нерегулярной структурой реальная производительность кластерных многопроцессорных вычислительных систем (МВС) невелика и даже с применением специальных методов обработки не превышает 30%. Для эффективной обработки больших матриц с нерегулярной структурой возможно использовать реконфигурируемые вычислительные системы (РВС), для которых авторами предложен метод обработки больших разреженных неструктурированных матриц (БРН-матриц), за счет которого была достигнута реальная производительность, близкая к 50% от пиковой. В статье описывается модификация разработанного метода обработки БРН-матриц, которая отличается распараллеливанием обработки ненулевых элементов строки и позволяет вдвое увеличить скорость работы вычислительной структуры при незначительном увеличении занимаемого аппаратного ресурса. Модифицированный метод обработки БРН-матриц на РВС обеспечивает реальную производительность, близкую к 90% от пиковой, что существенно превышает известные результаты решения подобных задач для кластерных МВС. Сравнение результатов решения задачи ранжирования веб-страниц алгоритмом PageRank, полученных на РВС “Арктур” и суперкомпьютере Fugaku, а также результатов решения СЛАУ методом Якоби на РВС “Арктур” и графическом ускорителе NVidia Tesla K40 подтверждает теоретические выводы.

**Ключевые слова:** реконфигурируемые вычислительные системы, высокопроизводительные вычислительные системы, разреженная матрица, большая неструктурированная матрица, формат разреженной матрицы, дискретно-событийное преобразование, баланс интенсивности потоков данных, распараллеливание вычислений, распараллеливание по ненулевым элементам.

**Для цитирования:** Левин И.И., Подопригора А.В. Модифицированный метод обработки больших разреженных неструктурированных матриц на реконфигурируемых вычислительных системах // Вычислительные методы и программирование. 2024. 25, № 2. 142–154. doi 10.26089/NumMet.v25r212.



# Modified method large sparse unstructured matrices processing on reconfigurable computing systems

Ilya I. Levin

Southern Federal University, Institute of Computer Technologies and Information Security,  
Department of Intelligent and Multiprocessor Systems, Taganrog, Russia  
ORCID: 0000-0002-1704-5016, e-mail: [iilevin@sfedu.ru](mailto:iilevin@sfedu.ru)

Aleksandr V. Podoprighora

Supercomputers and Neurocomputers Research Center, Taganrog, Russia  
ORCID: 0009-0003-4099-0088, e-mail: [podoprighora@superevm.ru](mailto:podoprighora@superevm.ru)

**Abstract:** When processing high-dimensional matrices with an irregular structure, the real performance of cluster multiprocessor computing systems (MCS) is low and even with the use of special processing methods does not exceed 30%. To effectively process large matrices with an irregular structure, it is possible to use reconfigurable computing systems (RCS), for which the authors proposed a method for processing large sparse unstructured (LSU) matrices, due to which real performance was achieved close to 50% of the peak. The article describes a modification of the developed method for processing LSU matrices, which is characterized by parallel processing of non-zero row elements and allows doubling the speed of the computing structure with a slight increase in the occupied hardware resource. The modified method of processing LSU matrices on an RCS provides real performance close to 90% of the peak, which significantly exceeds the known results of solving similar problems for cluster MCS. Comparison of the results of solving the problem of ranking web pages using the PageRank algorithm obtained on the “Arcturus” RCS and the Fugaku supercomputer, as well as the results of solving the SLAE using the Jacobi method on the “Arcturus” RCS and the graphics accelerator NVidia Tesla K40 confirms the theoretical conclusions.

**Keywords:** reconfigurable computing systems, high-performance computing systems, sparse matrix, large unstructured matrix, sparse matrix format, discrete-event transformation, intensity balance dataflow, parallelization of calculations, parallelization over non-zero elements.

**For citation:** I. I. Levin, A. V. Podoprighora, “Modified method large sparse unstructured matrices processing on reconfigurable computing systems,” Numerical Methods and Programming. 25 (2), 142–154 (2024). doi 10.26089/NumMet.v25r212.

---

**1. Введение.** Математические модели теплопередачи турбулентного потока жидкости большого объема [1], морской генераторной платформы по добыче нефтегазового сырья [2], системы энергоснабжения крупных промышленных городов и стран [3] и многие другие описываются системами линейных алгебраических уравнений, которые зачастую обрабатываются с использованием динамической расчетной сетки при переменном шаге дискретизации непрерывного уравнения. Сложные моделируемые объекты, как правило, несимметричны и описываются матрицами с большим количеством нулевых элементов и их случайным расположением [4]. Такие матрицы называют разреженными и неструктурированными.

Для сокращения времени решения задачи обработка матриц большой размерности (как структурированных, так и неструктурированных) выполняется на многопроцессорных вычислительных системах (МВС), содержащих универсальные и/или графические процессоры, объединенные в вычислительный кластер, как показано на рис. 1. Высокая тактовая частота работы процессоров  $P_i$ , внутренняя многопоточность и большое количество ядер процессоров потенциально обеспечивают высокую скорость обработки данных  $\mu_{P_i}$ , но обмен данными между локальной памятью и процессором  $\mu_{R_i}$  через коммутационную

среду  $S$  значительно медленней работы процессора, поскольку определяется компонентом МВС — локальной памятью  $M_i$ . Это приводит к дисбалансу скоростей поступления и обработки данных. Несбалансированная по скорости обработки и передачи архитектура МВС приводит к простоя вычислительного ресурса и, как следствие, к использованию специальных методов и подходов для согласования темпов поступления и обработки данных.

Обработка разреженных структурированных матриц, таких как диагональные или ленточные, выполняется с использованием специального формата, который хранит только ненулевые элементы. При большой размерности разреженные матрицы разбиваются на блоки для параллельной обработки на разных процессорах. В ходе выполнения параллельной обработки блоков матрицы на разных процессорах в соседние процессоры передается информация об обработанных блоках. Для сокращения количества передаваемых данных в блочном представлении разреженной матрицы информация передается не о каждом обработанном элементе, а об адресах элементов, расположенных по периметру выделенного блока, или функции, описывающей расположение ненулевых элементов, в то время как операция выполняется над всеми значимыми элементами блока. За счет этого время передачи данных в соседние процессоры значительно меньше, чем время обработки значимых элементов блока матрицы, и таким образом исключается простой оборудования.

При обработке больших разреженных неструктурированных матриц (БРН-матриц) такими методами реальная производительность вычислительных систем значительно снижается из-за случайного расположения ненулевых элементов, информацию о которых необходимо передавать соседним процессорам. В этом случае время передачи информации об обработанном блоке матрицы в соседние процессоры становится больше времени обработки блока матрицы и приводит к простоям оборудования.

Соотношение реальной производительности МВС к пиковой при обработке БРН-матриц сокращается до 15–30% [5], а в ряде случаев — до 5–7% [6] и даже 3–4% [7]. Разработка новых методов, обеспечивающих повышение реальной производительности при обработке БРН-матриц, становится актуальной.

**2. Обработка БРН-матриц на РВС.** Эффективная обработка БРН-матриц может быть выполнена на РВС, где основным вычислительным ресурсом являются программируемые логические интегральные схемы (ПЛИС), объединенные с помощью высокоскоростных каналов передачи данных в единое вычислительное поле [8–10]. Возможность реализации информационных связей в пространственно-коммутиционной сети позволяет сбалансировать интенсивности передачи данных из памяти  $\mu_{M_m}$  и их обработки  $\mu_{F_i}$  в конвейерной вычислительной структуре решаемой задачи, пример которой приведен на рис. 2.

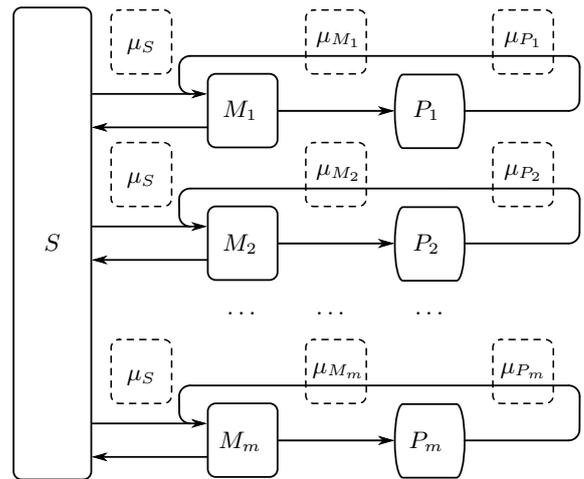


Рис. 1. Типовая архитектура кластерных многопроцессорных вычислительных систем

Fig. 1. Common cluster multiprocessor computing systems architecture

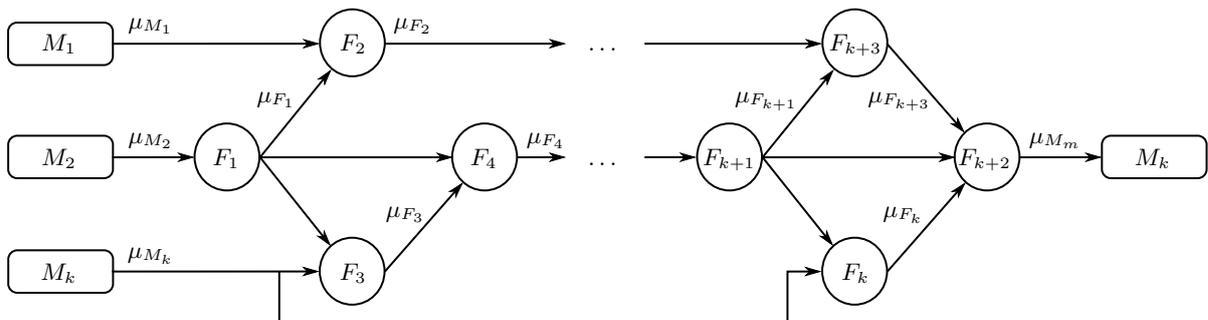


Рис. 2. Пример конвейерной вычислительной структуры для решения задачи на РВС

Fig. 2. Example pipe structure solve tasks on RCS



Задача обработки БРН-матриц представляется структурно в виде информационного графа и впоследствии реализуется в вычислительном поле РВС как множество конвейерных ступеней, содержащих последовательно соединенные функциональные устройства  $F_i$ , реализующие операции над матрицами. Известные методы обработки матриц для РВС, как правило, ориентированы на структурированные матрицы. При обработке неструктурированных матриц поток данных имеет особый вид, в котором появление ненулевых значимых элементов будет происходить случайно.

Использование специальных форматов хранения БРН-матриц, таких как ELLPACK [11] или CSR [12], приводит к нарушению баланса между интенсивностью передачи  $\mu_{M_m}$  и интенсивностью обработки  $\mu_{F_i}$  из-за случайного порядка следования ненулевых элементов. Интенсивность всей вычислительной структуры будет определяться самым медленным функциональным устройством в системе [13].

Для повышения эффективности обработки БРН-матриц и приближения реальной производительности к пиковой авторами разработан новый метод обработки [14], содержащий следующие этапы.

Обрабатываемые БРН-матрицы представляются в модифицированном формате ELLPACK как набор векторов, в которых сохраняются ненулевые элементы и позиции в строке к ним. Этот формат в первую очередь позволяет исключить нулевые элементы из хранения и передачи, существенно сокращая требования к памяти хранения операндов, а также обеспечивает одинаковую длину всех строк (столбцов) сжатой БРН-матрицы. После этого каждая строка дополняется специальным символом конца вектора или конца матрицы. Скорость передачи БРН-матриц определяется тактовой частотой работы функциональных узлов, но может быть увеличена за счет параллельной выдачи векторов по свободным каналам доступа к памяти.

При обработке двух и более БРН-матриц  $A, B, \dots, Y$  совпадение позиций ненулевых элементов с большой долей вероятности будет встречаться крайне редко. Поэтому необходимо синхронизировать ненулевые элементы БРН-матриц в соответствии с позициями для корректной подачи операндов в функциональные устройства. Для этого информационные потоки преобразуются дискретно-событийным способом, временная диаграмма показана на рис. 3.

Преобразование информационного потока выполняется следующим образом: ненулевые элементы каждой БРН-матрицы  $a_i, b_i, \dots, y_i$  с соответствующими индексами позиций  $\text{ind\_}a_i, \text{ind\_}b_i, \dots, \text{ind\_}y_i$  передаются в блок преобразования. Текущие индексы позиций  $\text{ind\_}a_i, \text{ind\_}b_i, \dots, \text{ind\_}y_i$  сравниваются между собой. На основании сравнения позиций первых ненулевых элементов  $\text{ind\_}a_i = 1, \text{ind\_}b_i = 5, \dots, \text{ind\_}y_i = 2$  только один ненулевой элемент  $a_{1,1}$ , с минимальным значением позиции, равной 1, передается в функциональное устройство для вычислений.

Оставшиеся ненулевые элементы  $b_{1,5}$  и  $y_{1,2}$  сохраняются в буферной памяти и повторно сравниваются с позицией  $\text{ind\_}a_i = 4$  элемента  $a_{1,4}$ , которая заместила элемент, переданный в обработку. Эти действия выполняются до тех пор, пока не будут обработаны все ненулевые элементы БРН-матриц. В результате формируются информационные потоки синхронизированных по позициям ненулевых элементов  $\text{DC\_}a_i, \text{DC\_}b_i, \dots, \text{DC\_}y_i$  и соответствующие им потоки позиций  $\text{DC\_ind\_}a_i, \text{DC\_ind\_}b_i, \dots, \text{DC\_ind\_}y_i$ .

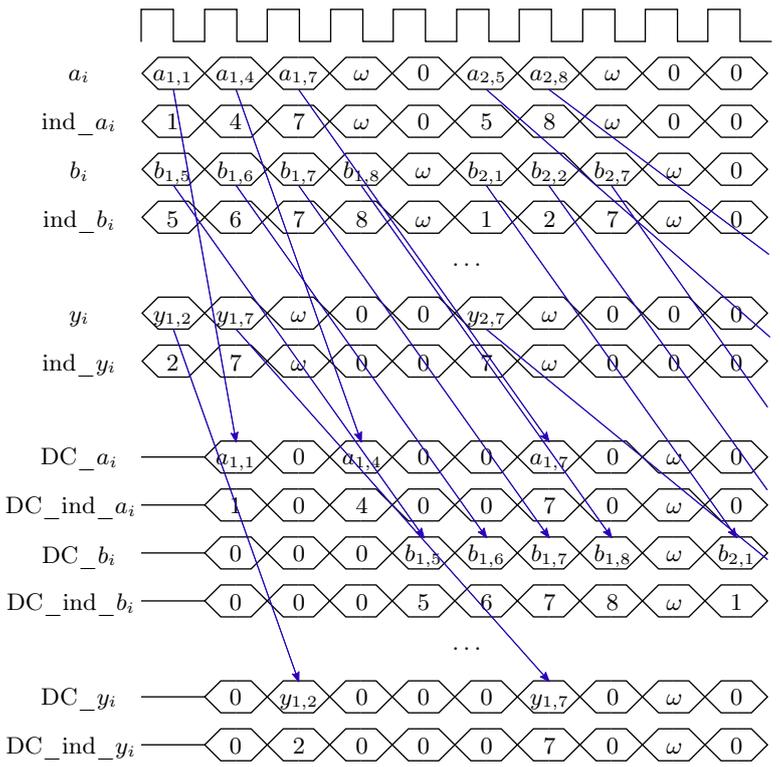


Рис. 3. Временная диаграмма преобразования исходных потоков БРН-матриц  $A, B, \dots, Y$  к дискретно-событийным потокам данных

Fig. 3. Waveform of converting initial streams of large sparse unstructured matrices  $A, B, \dots, Y$  to discrete-event data streams

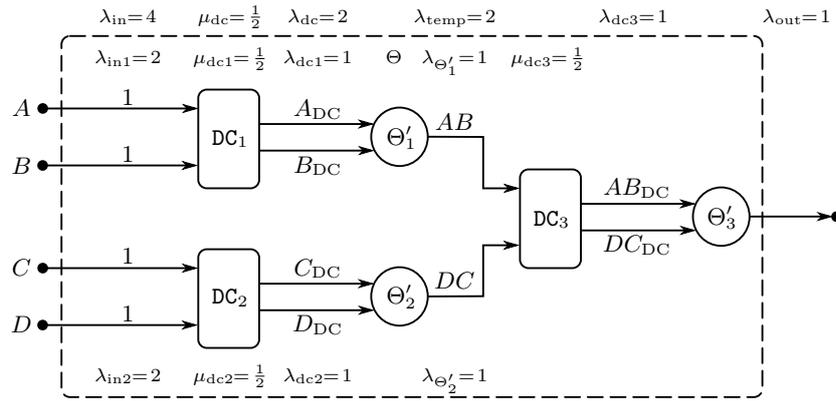


Рис. 4. Матричная операция над БРН-матрицами  $\Theta$  с выделенными базовыми операциями  
 Fig. 4. Matrix operation  $\Theta$  with large sparse unstructured matrices divided into basic operations

Время анализа позиций текущих ненулевых элементов и интенсивность обработки данных в функциональных устройствах напрямую зависят от количества обрабатываемых БРН-матриц. Для достижения высокой реальной производительности выполняется баланс интенсивностей информационных потоков, состоящий из следующих этапов.

На первом, подготовительном, этапе строки (столбцы) БРН-матриц дополняются незначащими элементами таким образом, чтобы длины строк всех БРН-матриц были равны между собой.

На следующем этапе выделяются информационные потоки и преобразуются дискретно-событийным способом  $DC_i$  в соответствии с базовыми матричными операциями  $\Theta'$ , как показано на рис. 4 [15]. Базовые матричные операции выполняются над двумя матрицами, например скалярное или векторное умножение двух матриц или сложение. Интенсивность поступления данных  $\lambda_{in}$  для базовой операции будет равна двум, по одному ненулевому элементу от каждой из двух БРН-матриц. Интенсивность обработки  $\lambda_{dc}$  будет равна единице, поскольку на выходе дискретно-событийного преобразования гарантированно будет определен только один ненулевой элемент и передан в функциональное устройство  $\Theta'_i$ . Соотношение интенсивности поступления  $\lambda_{in}$  и интенсивности обработки  $\lambda_{dc}$  каждой базовой операции будет равняться  $\mu_{dc} = 1/2$ , как и для всей вычислительной структуры решаемой реализованной задачи  $\Theta$ .

Следующим пунктом баланса интенсивностей потока ненулевых элементов БРН-матриц является буферизация информационного потока операндов. Размер буфера определяется в диапазоне от исходного значения размера длины строки, найденной на первом этапе баланса интенсивностей, до аналитически рассчитанного значения. Вероятность переполнения буфера определяется на основании интенсивности информационного потока ненулевых элементов, интенсивности обработки данных и глубины буферной памяти по формуле вероятности процесса гибели и размножения. После аналитического расчета размер буфера должен быть скорректирован так, чтобы исключить вероятность переполнения. Типовая вычислительная структура базовой операции  $\Theta'$  для БРН-матриц  $A$  и  $B$  с дискретно-событийным преобразованием  $DC$  и интенсивностью входного потока  $\lambda_{in}$  и интенсивностью обработки данных  $\lambda_{dc}$  показана на рис. 5.

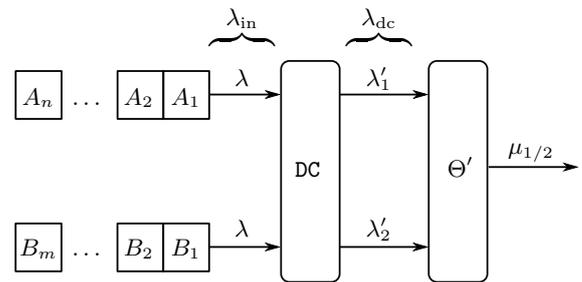


Рис. 5. Вычислительная структура блока базовой операции с БРН-матрицами на РВС

Fig. 5. Computational structure of the basic block operation with large sparse unstructured on RCS

Поскольку интенсивность обработки всех базовых операций с БРН-матрицами равна  $1/2$ , необходимо снизить интенсивность информационного потока ненулевых элементов БРН-матриц до уровня интенсив-



ности обработки данных, что приводит к заключительному этапу баланса интенсивностей — передаче данных с интервалом (скважностью) — один ненулевой элемент в каждый второй момент времени.

Передача потока значимых элементов с интервалом, равным двум, означает, что полезная работа функциональных устройств будет выполняться каждый второй момент времени. Таким образом, разработанный метод [14] обработки БРН-матриц на РВС обеспечивает реальную производительность вычислительной структуры относительно пиковой производительности РВС порядка 50%.

**3. Модификация метода обработки БРН-матриц на РВС.** Достигнутый в разработанном методе [14] уровень реальной производительности РВС означает, что половину всего времени работы вычислительная структура обработки БРН-матриц будет простаивать, поэтому следует рассмотреть способы повышения эффективности обработки данных.

Первый подход повышения эффективности обработки БРН-матриц на РВС заключается в увеличении скорости работы базовых матричных операций. Теоретически такая возможность существует, однако на практике такой подход ограничен архитектурными особенностями ПЛИС и нормами проектирования.

Для повышения эффективности использования аппаратного ресурса РВС при реализации базовых матричных операций можно использовать другой подход — распараллеливание вычислений. Существует несколько известных методов, наиболее тривиальный — распараллеливание по строкам [8]. Этот метод аналогичен параллелизму по данным в классических МВС и реализуется за счет увеличения количества задействованных входных каналов доступа к памяти хранения БРН-матриц и пропорционального увеличения количества используемого аппаратного ресурса для базовых операций. В этом случае производительность вычислительной системы увеличивается пропорционально коэффициенту распараллеливания, однако эффективность использования ресурса РВС остается прежней. Более того, применение метода распараллеливания по строкам существенно ограничивается критическим ресурсом РВС — каналами доступа к распределенной памяти.

В ряде случаев вычислительная структура с обратной связью может быть реализована как вложенный конвейер [16], в котором будет обрабатываться плотный поток данных. Передача данных в блоки базовых операций с БРН-матрицами с постоянным интервалом, равным двум, предполагает возможность использования метода вложенного конвейера так, чтобы задействовать каждый момент простоя, выполняя в нем обработку значимых элементов БРН-матриц. Однако этот метод может быть применен только для структурированных потоков данных, где порядок вычислений строго определен позицией в потоке. Для выполнения операций с БРН-матрицами порядок обработки определяется при синхронизации потоков данных дискретно-событийным преобразованием и неизвестен на этапе трансляции.

Известные методы распараллеливания по строкам и вложенный конвейер не увеличивают эффективности использования аппаратных средств при обработке БРН-матриц на РВС, в связи с чем был разработан новый подход — распараллеливание по ненулевым элементам строки. Суть этого подхода заключается в том, что для обработки потока ненулевых элементов с интервалом, равным единице, для каждой базовой матричной операции необходимо анализировать не по одному ненулевому элементу с соответствующей позицией  $a_{1,1}$  и  $b_{1,1}$ , а по паре последовательно стоящих ненулевых элементов  $a_{1,1}$  из основного потока  $a^n$ ,  $a_{1,2}$  — из вспомогательного потока  $a^c$  и  $b_{1,1}$  из основного потока  $b^n$ ,  $b_{1,2}$  — из вспомогательного потока  $b^c$ . Одновременный анализ четырех ненулевых элементов БРН-матриц  $A$  и  $B$  позволяет в каждый момент времени гарантированно определить два ненулевых элемента, которые обязательно будут обработаны в блоке базовой операции. В этом случае интенсивность информационного потока ненулевых элементов и интенсивность обработки будут равны, что подразумевает повышение реальной производительности в два раза. Для этого необходимо обеспечивать доступ к ненулевым элементам БРН-матриц за счет разделения буферов входных данных  $BUF\_A$  и  $BUF\_B$  на две части для каждой БРН-матрицы. В результате этого входной поток ненулевых элементов БРН-матриц на входы базовой операции будет передаваться как показано на рис. 6, где применяются следующие обозначения:

- $a^n, b^n$  — основная часть разделенного потока ненулевых элементов БРН-матриц;
- $a^c, b^c$  — вспомогательная часть разделенного потока ненулевых элементов БРН-матриц;
- $ind\_a^n, ind\_b^n$  — позиции ненулевых элементов БРН-матриц для основной части разделенного потока;
- $ind\_a^c, ind\_b^c$  — позиции ненулевых элементов БРН-матриц для вспомогательной части разделенного потока;
- $a_{1,k}$  — значение ненулевого элемента БРН-матрицы  $A$  с позицией  $k$ ;

- $b_{1,l}$  — значение ненулевого элемента БРН-матрицы  $B$  с позицией  $l$ ;
- $i_k$  — значение позиции ненулевого элемента БРН-матрицы  $A$  с позицией  $k$ ;
- $j_l$  — значение позиции ненулевого элемента БРН-матрицы  $B$  с позицией  $l$ ;
- $\omega$  — значение конца обрабатываемой строки.

Анализ происходит следующим образом: позиции ненулевых элементов  $i_1$  и  $j_1$  сравниваются в основных потоках БРН-матриц  $\text{ind\_}a^n$ ,  $\text{ind\_}b^n$ . При равенстве позиций двух ненулевых элементов будут сформированы в качестве операндов  $a_{1,1}$  и  $b_{1,1}$  и переданы в функциональное устройство базовой матричной операции вместе. Далее осуществляется загрузка следующих двух ненулевых элементов  $i_2$  и  $j_2$  по одному от каждой БРН-матрицы, и анализ выполняется снова.

Если  $i_1 < j_1$ , из ненулевого элемента  $a_{1,1}$ , соответствующего меньшей по значению позиции, формируется первый операнд. Для формирования второго операнда позиция  $j_1$  сравнивается с позицией  $i_2$  ненулевого элемента, находящегося во вспомогательном канале БРН-матрицы  $A$ . Сравнение позиций основного  $\text{ind\_}a^n$  и вспомогательного  $\text{ind\_}a^c$  каналов одной БРН-матрицы не выполняется, поскольку элементы уже упорядочены. При условии, что  $j_1 < i_2$ , формируется второй операнд из ненулевого элемента  $b_{1,1}$ . Полученные два элемента  $a_{1,1}$  и  $b_{1,1}$  обрабатываются в блоке базовой матричной операции отдельно, поскольку их позиции не совпадают. Далее осуществляется загрузка следующих двух ненулевых элементов, и анализ выполняется снова.

Для обработки такого потока ненулевых элементов БРН-матриц с интервалом подачи данных, равным единице, требуется обеспечить доступ к двум последовательно стоящим элементам каждой БРН-матрицы, что реализуется за счет разделения буферных элементов в поле РВС на буферы  $\text{BUF\_}A_c$ ,  $\text{BUF\_}A_n$  и  $\text{BUF\_}B_c$ ,  $\text{BUF\_}B_n$ . При этом количество портов доступа к памяти хранения БРН-матриц  $A$  и  $B$  сохраняется. Разделение буферов хранения информационного потока ненулевых элементов целесообразно выполнять не на всю глубину изначально определенного буфера. Размер распараллеленного буфера в несколько раз меньше общего буфера, в связи с чем количество требуемых ресурсов памяти увеличивается незначительно. Вычислительная структура блока ускоренной базовой операции с БРН-матрицами на РВС показана на рис. 7.

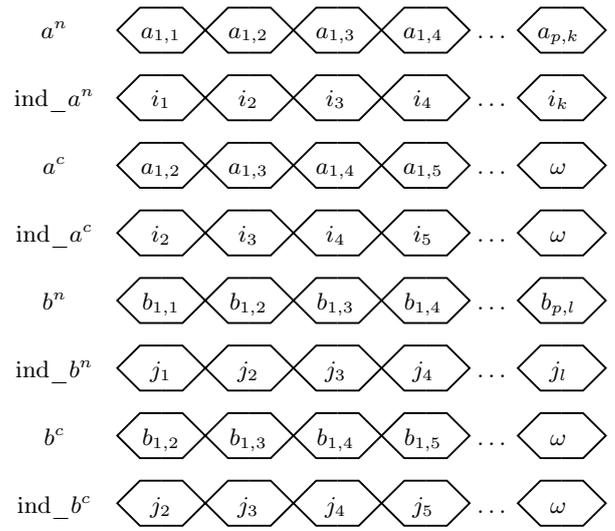


Рис. 6. Временная диаграмма потоков ненулевых элементов БРН-матриц  $A$  и  $B$  с применением распараллеливания по ненулевым элементам строки

Fig. 6. Waveform of non-zero elements streams large sparse unstructured matrices  $A$  and  $B$  with using parallelization over non-zero elements of rows

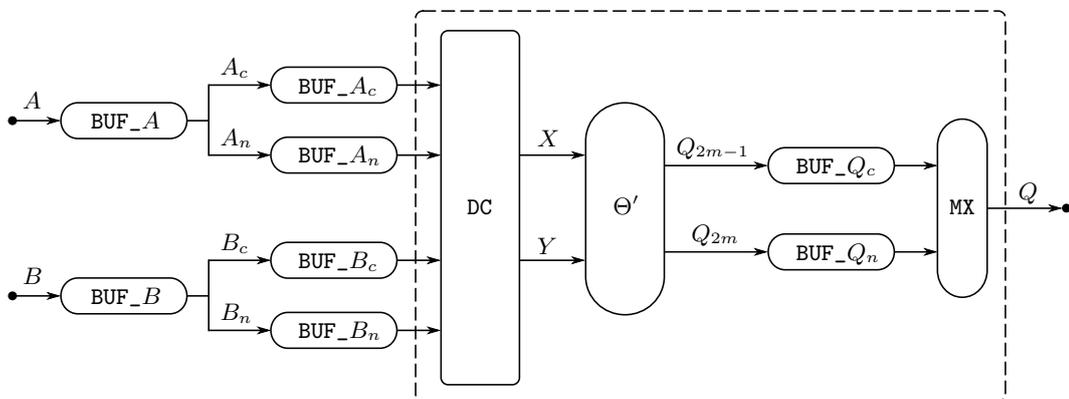


Рис. 7. Вычислительная структура блока ускоренной базовой операции с БРН-матрицами на РВС

Fig. 7. Computational structure of the boosted basic block operation with large sparse unstructured matrices on RCS



Модифицированный, распараллеливанием по ненулевым элементам строки, метод обработки БРН-матриц на РВС не удваивает количество требуемого аппаратного ресурса на реализацию базовых операций, в отличие от распараллеливания по строкам. Вычислительная структура ускоренного блока базовых операций с БРН-матрицами содержит тот же ресурс сумматоров для операций сложения или связей устройств сумматор-умножитель для операций скалярного умножения БРН-матриц и незначительно увеличенный ресурс буферной памяти для информационного потока ненулевых элементов БРН-матриц. Незначительное изменение начального количества функциональных устройств в ускоренной базовой операции объясняется тем, что в каждый момент времени обрабатывается не больше пары операндов, которые не могут задействовать больше одного сумматора или умножителя, а входной буфер разделяется частично.

Разработанный модифицированный метод обработки БРН-матриц для РВС позволяет удвоить интенсивность с незначительным ростом аппаратных затрат на реализацию ускоренного блока обработки БРН-матриц и достичь эффективности использования аппаратного ресурса порядка 90%.

**4. Результаты использования модифицированного метода обработки БРН-матриц на РВС.** Эффективность разработанного модифицированного метода оценивалась для решения прикладной задачи ранжирования по “важности” веб-страниц, связанных гиперссылками в поисковой системе Google, на основе алгоритма PageRank [17]. Выбор алгоритма обусловлен наличием операций обработки БРН-матриц и опубликованными результатами реальной производительности на суперкомпьютере Fugaku. Алгоритм PageRank основывается на поиске собственных векторов стохастической матрицы, которая формируется в соответствии с существующими веб-страницами в интернете и тем количеством ссылок, которые возникают между ними. Большое количество веб-страниц формирует матрицу с миллиардами элементов в строке и столбце, при этом она имеет разреженную структуру и случайное расположение ненулевых элементов.

Поиск собственных векторов матрицы выполняется по формуле

$$\mathbf{r}_{k+1} = \frac{A \cdot \mathbf{r}_k}{\|A \cdot \mathbf{r}_k\|},$$

где  $A$  — матрица коэффициентов;  $\mathbf{r}_{k+1}$  — искомый собственный вектор матрицы  $A$ ;  $\mathbf{r}_k$  — текущий собственный вектор матрицы  $A$ , полученный на  $k$ -й итерации;  $\|A \cdot \mathbf{r}_k\|$  — норма произведения матрицы  $A$  на собственный вектор  $\mathbf{r}_k$ .

Для реализации конвейерной вычислительной структуры выполняются преобразования

$$\mathbf{r}_n = \frac{A^n \cdot \mathbf{r}_0}{\prod_{m=0}^n K_m}, \quad n > 0, \tag{1}$$

$$K_m = \left\| \frac{A^m \cdot \mathbf{r}_0}{\prod_{j=0}^{m-1} K_j} \right\|, \quad m > 0, \quad K_0 = 1. \tag{2}$$

В числителе формулы (1) выполняется возведение исходной матрицы в степень  $n$  с последующим умножением на начальный вектор  $\mathbf{r}_0$ , в знаменателе — произведение норм итераций, начиная с  $K_0$  и заканчивая  $K_n$ . Норма  $K_m$  вычисляется по формуле (2), где каждая последующая включает в себя значения предыдущих норм и может рассчитываться отдельно от наиболее времязатратной операции возведения матрицы в степень  $n$ .

Для реализации задачи ранжирования веб-страниц алгоритмом PageRank на РВС использовался модифицированный метод обработки БРН-матриц на РВС, дополнительно были использованы методы распараллеливания по строкам и метод распараллеливания по итерациям [9]. Для поиска собственных векторов используется итерационный степенной метод, поскольку применение прямого метода ведет к преобразованию БРН-матрицы к треугольной форме, что значительно увеличивает ее плотность и время обработки.

Применение метода распараллеливания по итерациям исключает необходимость выгружать в распределенную память результат текущей итерации реализуемой задачи и позволяет начинать обработку этих данных в следующей итерации реализованной конвейерной вычислительной структуры. Таким образом, в вычислительной структуре из  $n$  итераций (блоков) после обработки потока данных будет получен результат  $n$ -й итерации реализованного алгоритма. Этот подход применяется для решения задач СЛАУ итерационными методами, прямыми методами, для задач ЦОС, для переборных задач и многих других.

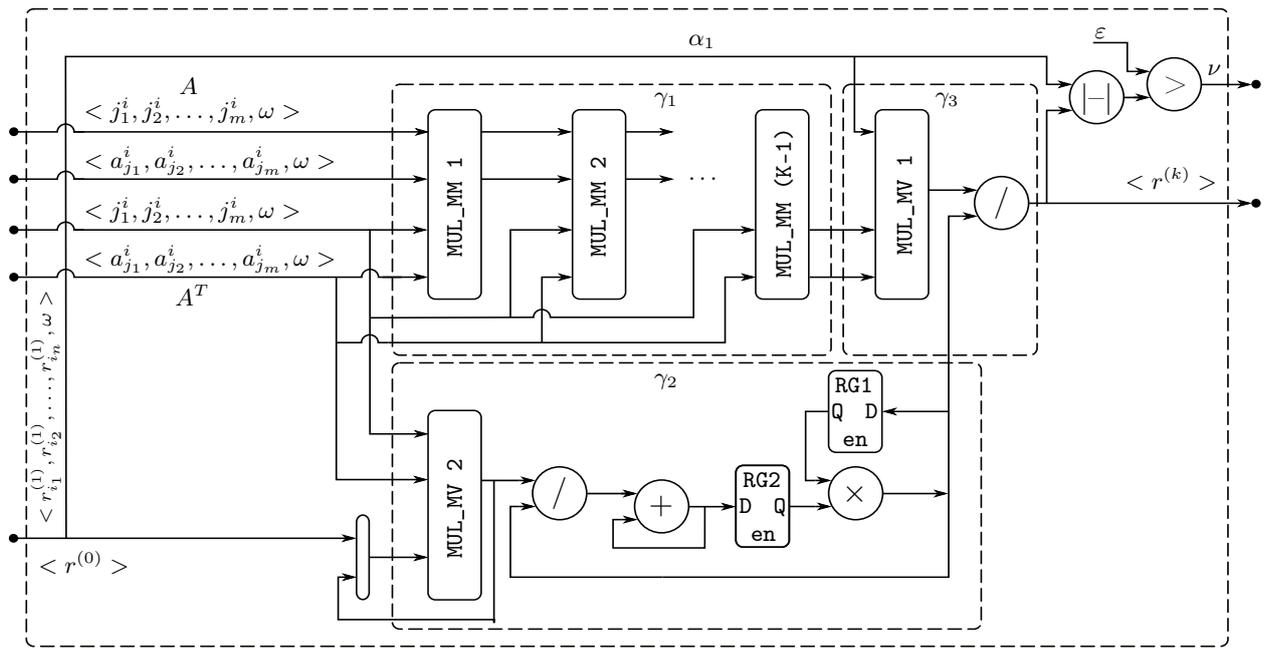


Рис. 8. Вычислительная структура поиска собственных векторов степенным методом с БРН-матрицей коэффициентов

Fig. 8. Computational structure for searching for eigenvectors using the power method with the large sparse unstructured matrix of coefficients

Для РВС “Арктур” [10] была разработана прикладная программа подграфа задачи поиска собственных векторов степенным методом, показанная на рис. 8. Вычислительная структура задачи содержит операции скалярного умножения двух БРН-матриц в  $\gamma_1$ , скалярного умножения БРН-матрицы на плотный вектор в  $\gamma_3$  и  $\gamma_2$ , а также деления, суммирования и умножения. Полученная вычислительная структура  $\alpha$  позволяет выполнять распараллеливание по строкам и по итерациям на доступный вычислительный ресурс РВС.

Входные потоки элементов БРН-матрицы поступают в блок  $\gamma_1$ , где выполняется возведение в степень каскадом операций  $MUL\_MM$ . Множимое  $[a_{j_m}^i, j_m^i]$  передается в модифицированном формате ELLPACK по строкам, а множитель  $[a_{i_m}^j, i_m^j]$  — по столбцам. Рассчитанная степень матрицы передается на вход умножителя БРН-матрицы на собственный вектор в блок  $\gamma_3$ . В блоке  $\gamma_2$  выполняется операция умножения БРН-матрицы на начальный собственный вектор  $MUL\_MV$  2, в результате которой формируется норма текущего вычисления. Полученный при выполнении операции блока  $MUL\_MV$  2 вектор результата нормируется рассчитанным значением произведения норм прошлых итераций. В результате полученное значение произведения норм в блоке  $\gamma_2$  передается в блок  $\gamma_3$  для расчета нового значения собственного вектора  $r_k$ .

Достигнутая при реализации на РВС “Арктур” реальная производительность поиска собственных векторов для алгоритма PageRank модифицированным методом обработки БРН-матриц сравнивалась с реализацией этой задачи на 1024 процессорах Fujitsu A64FX [18] суперкомпьютера Fugaku. Каждый процессор Fujitsu A64FX обрабатывает БРН-матрицу размером  $500000 \times 2000000$  с 50 или 100 ненулевыми элементами в строке. В общем случае такая система позволяет обрабатывать БРН-матрицу с размерностью более  $4 \cdot 10^{11}$  элементов и 102400 ненулевых элементов в строке.

Сравнительная характеристика производительности одной ПЛИС Xilinx XCVU37P, одного вычислительного модуля (ВМ) и одного вычислительного блока (РВБ) РВС “Арктур” с производительностью одного, 8-ми и 1024-х процессоров кластерной МВС Fugaku для задачи поиска собственных векторов БРН-матриц по алгоритму PageRank показана в табл. 1. Ускорение показанных в таблице конфигураций вычислительных систем рассчитывалось относительно реальной производительности одного процессора Fujitsu A64FX.

Как видно из табл. 1, ускорение вычислений при поиске собственных векторов степенным методом одной ПЛИС РВБ “Арктур” к одному процессору МВС Fugaku составило 39 раз. При увеличении вы-



Таблица 1. Производительности задачи поиска собственных векторов BRN-матрицы степенным методом для алгоритма PageRank на РВБ “Арктур” и кластерной МВС Fugaku  
 Table 1. Performance of the problem of searching for eigenvectors of a BRN matrix using the power-law method for the PageRank algorithm on the “Arcturus” RVB and the Fugaku cluster MCS

Характеристики Characteristics	Вычислительная система Computational system					
	РВС “Арктур” “Arcturus” RVB			МВС Fugaku Fugaku cluster MCS		
Количество микросхем Number of chips	1 ПЛИС 1 FPGA	6 ПЛИС 6 FPGA	96 ПЛИС 96 FPGA	1 CPU	8 CPU	1024 CPU
Количество функциональных устройств Number of functional devices	235	1407	22521	48	384	49152
Частота, МГц Clock Frequency, MHz	550			2200		
Реальная производительность, GFlops Real performance, GFlops	274	1647	26357	7	62	7100
Ускорение Acceleration	39	235	3765	1	8.8	1014

числительного ресурса и распараллеливании задачи ускорение для РВС растет быстрее и показывает линейный рост производительности при увеличении ресурса РВС.

Предложенный модифицированный метод может применяться и для задачи решения СЛАУ методом Якоби с BRN-матрицей коэффициентов [19]. Для РВС “Арктур” была разработана соответствующая прикладная программа подграфа задачи.

Сравнительная характеристика производительности одной ПЛИС Xilinx XCVU37P с производительностью одного графического ускорителя NVidia Tesla K40 [20] для задачи решения СЛАУ методом Якоби выполнялась на основании затраченного времени на решение задачи. Время решения СЛАУ с BRN-матрицей `crankseg_1` [21] на графическом ускорителе NVidia Tesla K40 составило 2.6 с, в то время как на одной ПЛИС Xilinx XCVU37P РВС “Арктур” — 0.019 с.

В итоге ускорение вычислений при решении СЛАУ методом Якоби одной ПЛИС РВБ “Арктур” к одному графическому ускорителю NVidia Tesla K40 составило 134 раза, с возможностью линейного роста реальной производительности РВС при увеличении аппаратного ресурса.

**5. Заключение.** Предлагаемая модификация метода обработки больших разреженных неструктурированных матриц позволяет повысить реальную производительность РВС и использовать аппаратные средства с высокой эффективностью. Применение модифицированного метода обработки BRN-матриц обеспечивает близкий к пиковому уровень реальной производительности РВС. В частности, для задачи поиска собственных векторов и решения СЛАУ методом Якоби применение модифицированного метода обработки BRN-матриц для РВС позволяет достичь реальной производительности порядка 90% от пиковой, что в несколько раз выше аналогичного показателя для кластерных многопроцессорных вычислительных систем. Разработанный метод обработки BRN-матриц может быть использован не только для реконфигурируемых вычислительных систем на основе ПЛИС, но и для других систем, поддерживающих конвейерные вычисления, например, для цифровых фотонных вычислительных систем [22].

**Список литературы**

1. Jones I.P., Thompson C.P. A note on the use of non-uniform grids in finite difference calculations in boundary and interior layers // Computational and Asymptotic Methods. Dublin: Boole Press, 1980. 332–341.
2. Duff I.S., Grimes R.G., Lewis J.G. Users’ guide for the Harwell–Boeing sparse matrix collection (Release 1) // Technical Report Number RAL-92-086. Chilton: Rutherford Appleton Laboratory, 1992. <https://api.semanticscholar.org/CorpusID:56998202>. Cited April 12, 2024.
3. Burchett R.C., Happ H.H., Vierath D.R. Reactive power planning of large-scale systems // <https://api.semanticscholar.org/CorpusID:107515440>. Cited April 12, 2024.
4. Davis T. Sparse Matrix Database. Williams Group Matrix - Webbase-1M // <https://sparse.tamu.edu/Williams/webbase-1M>. Cited April 12, 2024.

5. *Georgopoulos L., Sobczyk A., Christofidellis D., et al.* Enhancing multi-threaded sparse matrix multiplication for knowledge graph oriented algorithms and analytics // <https://dominoweb.draco.res.ibm.com/reports/RZ3953.pdf>. Cited April 12, 2024.
6. *Somers G.W.* Acceleration of block-aware matrix factorization on heterogeneous platforms. Master's Thesis in Electrical and Computer Engineering. Ottawa: University of Ottawa, 2016.
7. *Yang C., Buluç A., Owens J.D.* Design principles for sparse matrix multiplication on the GPU // Proc. Int. European Conference on Parallel and Distributed Computing. Torino, Italy, August 27–31, 2018. <https://people.eecs.berkeley.edu/~aydin/Yang2018EuroPar.pdf>. Cited April 12, 2024.
8. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Янус-К, 2003.
9. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультимониторные вычислительные структуры. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2008.
10. *Левин И.И., Дордопуло А.И., Федоров А.М., Доронченко Ю.И.* Развитие технологии построения реконфигурируемых вычислительных систем с жидкостным охлаждением // Суперкомпьютерные технологии (СКТ-2018): материалы 5-й Всероссийской научно-технической конференции. Том 1. Ростов-на-Дону: Изд-во ЮФУ, 2018. 184–187.
11. *Grimes R.G., Kincaid D.R., Young D.M.* ITPACK 2.0 User's Guide. Technical Report Number CNA-150. Austin: University of Texas, 1979.
12. *Saad Y.* SPARSKIT: a basic tool kit for sparse computations (Version 2). Technical Report. Minneapolis: University of Minnesota, 1994.
13. *Левин И.И., Дордопуло А.И., Каляев И.А., Гудков В.А.* Высокопроизводительные реконфигурируемые вычислительные системы на основе ПЛИС VIRTEX-7 // Программная инженерия. 2014. № 6. 3–7.
14. *Подопригора А.В.* Метод организации дискретно-событийных вычислений для обработки больших разреженных неструктурированных матриц на PBC // Известия ЮФУ. Технические науки. 2021. № 7. 189–197.
15. *Левин И.И., Подопригора А.В.* Метод распараллеливания по базовым макрооперациям для обработки больших разреженных неструктурированных матриц на PBC // Известия ЮФУ. Технические науки. 2022. № 6. 72–83.
16. *Коваленко А.Г.* Макроконвейерная реализация алгоритмов аутентификации на языке высокого уровня COLAMO // Известия ЮФУ. Технические науки. 2012. № 4. 210–215.
17. *Page L., Brin S., Motwani R., Winograd T.* The PageRank citation ranking: bringing order to the Web. Technical Report Number SIDL-WP-1999-0120. Stanford: Stanford University, 1999.
18. *Vandromme M., Gurhem J., Tsuji M., et al.* Scaling the PageRank algorithm for very large graphs on the Fugaku supercomputer // Lecture Notes in Computer Science. Vol. 13350. Cham: Springer, 2022. 389–402. [https://link.springer.com/chapter/10.1007/978-3-031-08751-6\\_28](https://link.springer.com/chapter/10.1007/978-3-031-08751-6_28). Cited April 12, 2024. doi 10.1007/978-3-031-08751-6\_28.
19. *Chow E., Anzt H., Scott J., Dongarra J.* Using Jacobi iterations and blocking for solving sparse triangular systems in incomplete factorization preconditioning // J. Parallel Distr. Comput. 2018. 119. 219–230. doi 10.1016/j.jpdc.2018.04.017.
20. *NVIDIA Corporation.* Whitepaper NVIDIA's Next Generation CUDA Compute Architecture: Kepler GK110/210. 2014. <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/NVIDIA-Kepler-GK110-GK210-Architecture-Whitepaper.pdf>. Cited April 12, 2024.
21. *Kolodziej S.P., Aznaveh M., Bullock M., et al.* The SuiteSparse matrix collection website interface // J. Open Source Softw. 2019. N 4, Article Number 1244. doi 10.21105/joss.01244. Cited April 12, 2024.
22. *Sorokin D.A., Kasarkin A.V., Podopriгора A.V.* Elements of a digital photonic computer // Supercomput. Front. Innov. 2023. 10, № 2. 62–76. doi 10.14529/jsfi2302. Cited April 12, 2024.

Поступила в редакцию

16 июня 2023 г.

Принята к публикации

13 октября 2023 г.

### Информация об авторах

*Илья Израилевич Левин* — д.т.н., заведующий кафедрой; Южный федеральный университет, Институт компьютерных технологий и информационной безопасности, кафедра интеллектуальных и многопроцессорных систем, ул. Чехова, 2, корпус "И", 347900, Таганрог, Российская Федерация.

*Александр Владимирович Подопригора* — м.н.с.; Научно-исследовательский центр супер-ЭВМ и нейрокомпьютеров, пер. Итальянский, дом 106, 347900, Таганрог, Российская Федерация.



## References

1. I. P. Jones and C. P. Thompson, “A Note on the Use of Non-Uniform Grids in Finite Difference Calculations in Boundary and Interior Layers,” in *Computational and Asymptotic Methods* (Boole Press, Dublin, 1980), pp. 332–341.
2. I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users’ Guide for the Harwell–Boeing Sparse Matrix Collection (Release 1)*, Report Number RAL-92-086 (Rutherford Appleton Laboratory, Chilton, United Kingdom, 1992). <https://api.semanticscholar.org/CorpusID:56998202>. Cited April 12, 2024.
3. R. C. Burchett, H. H. Happ, and D. R. Vierath, “Reactive Power Planning of Large-Scale Systems,” <https://api.semanticscholar.org/CorpusID:107515440>. Cited April 12, 2024.
4. T. Davis, *Sparse Matrix Database. Williams Group Matrix - Webbase-1M* <https://sparse.tamu.edu/Williams/webbase-1M>. Cited April 12, 2024.
5. L. Georgopoulos, A. Sobczyk, D. Christofidellis, et al., *Enhancing Multi-Threaded Sparse Matrix Multiplication for Knowledge Graph Oriented Algorithms and Analytics*. <https://dominoweb.draco.res.ibm.com/reports/RZ3953.pdf>. Cited April 12, 2024.
6. G. W. Somers, *Acceleration of Block-Aware Matrix Factorization on Heterogeneous Platforms*, Master’s Thesis in Electrical and Computer Engineering (University of Ottawa, Ottawa, 2016).
7. C. Yang, A. Buluç, and J. D. Owens, “Design Principles for Sparse Matrix Multiplication on the GPU,” in *Proc. Int. European Conference on Parallel and Distributed Computing, Torino, Italy, August 27–31, 2018*. <https://people.eecs.berkeley.edu/~aydin/Yang2018EuroPar.pdf>. Cited April 12, 2024.
8. A. V. Kalyaev and I. I. Levin, *Modular Scalable Multiprocessor Systems with Structural and Procedural Organization of Calculations* (Janus-K, Moscow, 2003) [in Russian].
9. I. A. Kalyaev, I. I. Levin, E. A. Semernikov, and V. I. Shmoilov, *Reconfigurable Multi-Pipeline Computing Structures* (Ross. Akad. Nauk, Rostov-on-Don, 2008) [in Russian].
10. I. I. Levin, A. I. Dordopulo, A. M. Fedorov, and Yu. I. Doronchenko, “Development of Technology for Constructing Reconfigurable Computer Systems with Liquid Cooling,” in *Proc. 5th All-Russian Scientific and Technical Conference on Supercomputer Technologies, Divnomorskoe, Russia, September 17–22, 2018* (Southern Federal Univ., Rostov-on-Don, 2018), Vol. 1, pp. 184–187 [in Russian].
11. R. G. Grimes, D. R. Kincaid, and D. M. Young, *ITPACK 2.0 User’s Guide*, Technical Report Number CNA-150 (Center for Numerical Analysis, University of Texas, 1979).
12. Y. Saad, *SPARSKIT: a Basic Tool Kit for Sparse Computations (Version 2)*, Technical Report (University of Minnesota, Minneapolis, 1994).
13. I. I. Levin, A. I. Dordopulo, I. A. Kalyaev, and V. A. Gudkov, “High-Performance Reconfigurable Computer Systems on the Base of Virtex-7 FPGAs,” *Software Engineering*, No. 6, 3–7 (2014). [http://novtex.ru/prin/eng/10.17587/prin.\\_6\\_2014\\_1.html](http://novtex.ru/prin/eng/10.17587/prin._6_2014_1.html). Cited April 12, 2024.
14. A. V. Podoprigora, “Discrete-Event Method Computations Organizing for Processing Large Sparse Unstructured Matrices on RCS,” *Izvestiya SFedU. Engineering Sciences*, No. 7, 189–197 (2021). [https://izv-tn.tti.sfedu.ru/index.php/izv\\_tn/article/view/590](https://izv-tn.tti.sfedu.ru/index.php/izv_tn/article/view/590). Cited April 12, 2024.
15. I. I. Levin and A. V. Podoprigora, “Method of Parallelization on Basic Macro Operations for Processing Large Sparse Unstructured Matrices on RCS,” *Izvestiya SFedU. Engineering Sciences*, No. 6, 72–83 (2022). [https://izv-tn.tti.sfedu.ru/index.php/izv\\_tn/article/view/725](https://izv-tn.tti.sfedu.ru/index.php/izv_tn/article/view/725). Cited April 12, 2024.
16. A. G. Kovalenko, “Macropipeline Implementation of Authentication Algorithms on High-Level Language COLAMO,” *Izvestiya SFedU. Engineering Sciences*, No. 4, 210–215 (2012).
17. L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report Number SIDL-WP-1999-0120, (Stanford University, Stanford, 1999).
18. M. Vandromme, J. Gurhem, M. Tsuji, et al., “Scaling the PageRank Algorithm for Very Large Graphs on the Fugaku Supercomputer,” in *Lecture Notes in Computer Science* (Springer, Cham, 2022), Vol. 13350, pp. 389–402. [https://link.springer.com/chapter/10.1007/978-3-031-08751-6\\_28](https://link.springer.com/chapter/10.1007/978-3-031-08751-6_28). Cited April 12, 2024. doi 10.1007/978-3-031-08751-6\_28.
19. E. Chow, H. Anzt, J. Scott, and J. Dongarra, “Using Jacobi Iterations and Blocking for Solving Sparse Triangular Systems in Incomplete Factorization Preconditioning,” *J. Parallel Distr. Comput.* **119**, 219–230 (2018). doi 10.1016/j.jpdc.2018.04.017.
20. *NVIDIA Corporation*. Whitepaper NVIDIA’s Next Generation CUDA Compute Architecture: Kepler GK110/210. 2014. <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/NVIDIA-Kepler-GK110-GK210-Architecture-Whitepaper.pdf>. Cited April 12, 2024.



21. S. P. Kolodziej, M. Aznaveh, M. Bullock, et al., “The SuiteSparse Matrix Collection Website Interface,” *J. Open Source Softw.* No. 4, Article Number 1244 (2019). doi [10.21105/joss.01244](https://doi.org/10.21105/joss.01244). Cited April 12, 2024.
22. D. A. Sorokin, A. V. Kasarkin, and A. V. Podoprigora, “Elements of a Digital Photonic Computer,” *Supercomput. Front. Innov.* **10** (2), 62–76 (2023). doi [10.14529/jsfi2302](https://doi.org/10.14529/jsfi2302). Cited April 12, 2024.

*Received*  
*June 16, 2023*

*Accepted for publication*  
*October 13, 2023*

### **Information about the authors**

*Ilya I. Levin* — Dr. Sci., Head of the department; Southern Federal University, Institute of Computer Technologies and Information Security, Department of Intelligent and Multiprocessor Systems, Chekhova 2, building “T”, 347900, Taganrog, Russia.

*Aleksandr V. Podoprigora* — junior researcher; Supercomputers and Neurocomputers Research Center, Italyansky lane, 106, 347900, Taganrog, Russia.