

Раздел IV. Реконфигурируемые и нейросетевые вычислительные системы

УДК 004.273

DOI 10.23683/2311-3103-2018-8-204-212

А.И. Левина, Е.Е. Семерникова, Д.А. Сорокин

РЕШЕНИЕ ЗАДАЧИ СБОРКИ МОЛЕКУЛЫ ДНК НА РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Статья посвящена исследованиям, направленным на поиск методов и средств решения задачи сборки молекулы ДНК, обеспечивающих существенное сокращение времени вычислений при сохранении заданной точности. Рассматривается возможность применения реконфигурируемых вычислительных систем для решения задачи ассемблирования на примере реализации одной из ключевых процедур в составе алгоритма сборки генома Velvet Assembler – процедуры формирования контигов VelvetH. В основе сборщика Velvet Assembler лежит метод нового поколения, предполагающий построение графа де Брюйна, что порождает в NP-полной задаче ассемблирования ДНК существенно переменную интенсивность потоков данных. Поэтому, помимо традиционной для реконфигурируемых вычислительных систем технологии структурно-процедурной организации вычислений, для обеспечения принципиальной возможности решения подобных задач на реконфигурируемых вычислительных системах, применялись специальные методы синтеза параллельно-конвейерных программ. Для оценки эффективности применения реконфигурируемых вычислительных систем была разработана параллельно-конвейерная программа сборки генома из коротких чтений молекулы ДНК *Staphylococcus aureus* на основе процедуры VelvetH. Данные были взяты из базы данных Sequence Read Archive на сайте National Center for Biotechnology Information. Испытания данной параллельно-конвейерной программы проводились на реконфигурируемой вычислительной системе "Tertius", построенной на базе четырёх программируемых логических интегральных схем Kintex UltraScale XCKU 095 производства фирмы Xilinx. Применение данной реконфигурируемой вычислительной системы обеспечивает сокращение времени выполнения процедуры формирования контигов задачи ассемблирования ДНК в 24 и более раз по сравнению с существующими аналогами. Всё это позволяет сделать вывод о том, что реконфигурируемые вычислительные системы являются перспективным средством для решения задачи ассемблирования ДНК, требующим проведения дальнейших научно-технических исследований.

Программируемые логические интегральные схемы; реконфигурируемые вычислительные системы; сборка ДНК технологии нового поколения; ускорение вычислений.

A.I. Levina, E.E. Semernikova, D.A. Sorokin

IMPLEMENTATION OF THE DNA ASSEMBLY PROBLEM ON RECONFIGURABLE COMPUTER SYSTEMS

The paper deals with research of methods and tools for the problem of DNA assembly, which provide considerable reducing of the processing time for the specified accuracy in comparison with other methods and tools. We are considering using of reconfigurable computer systems for the assembly problems. As an example we use implementation of a key procedure of the algorithm of genome assembly Velvet Assembler – a procedure of contigs generation VelvetH. The base of the Velvet Assembler is a new generation method which implies generation of a de Bruijn graph, and, as a result, causes considerably variable density of data flows in the nondeterministic polynomial time complete problem of DNA assembly. That is why, in addition to the tech-

nology of structural-procedural organization of calculations, which is traditional for reconfigurable computer systems, we used special methods of synthesis of parallel-pipeline applications to provide a possibility in principle of implementation of such problems on reconfigurable computer systems. For evaluation of efficiency of reconfigurable computer systems use, we have developed, using a procedure VelvetH, a parallel-pipeline application which assembles a genome from short reads of a DNA Staphylococcus aureus. We have taken data from the database Sequence Read Archive from the website National Center for Biotechnology Information. The parallel-pipeline application was tested on a reconfigurable computer system "Tertius", designed on the base of four Xilinx Kintex UltraScale XCKU 095 FPGAs. Use of this reconfigurable computer system provides 24-fold (and more) reduction of the execution time of the contig generation procedure for the DNA assembly problem against the existing analogs. Due to this we can conclude, that use of reconfigurable computer systems for implementation of the DNA assembly problem is a promising direction, which requires further scientific and technical research.

Field-programmable gate array; reconfigurable computing structure; de novo sequencing assembly; computing acceleration.

Введение. В настоящее время восстановление первичной структуры ДНК (секвенирование) имеет большое прикладное значение в таких областях, как: биосинтетическая промышленность, геновая инженерия, криминалистика, медицина. Данный подход направлен на чтение генома, длина которого может достигать порядка 10^{12} пар нуклеотидов, но «прочитать» полностью всю такую последовательность, на сегодняшний день, не представляется возможным. Максимальное число нуклеотидов в цепочке должного качества, которое позволяет прочитать современные приборы, составляет порядка 1000 штук. Читать более длинные цепочки оказалось слишком дорого и малоэффективно, поэтому современные алгоритмы секвенирования заключаются в считывании большого количества коротких фрагментов в случайных местах одной и той же молекулы ДНК. На следующем этапе, полученный набор «коротких чтений», упорядочивается и состыковывается по пересекающимся фрагментам (для каждой пары «коротких чтений»), тем самым генерируется единая цепочка нуклеотидов, такой процесс называется сборкой генома (ассемблированием).

На сегодняшний день существует множество различных программных комплексов, осуществляющих сборку генома, однако их алгоритмы и исходные коды, как правило, являются коммерческой тайной. Единственный доступный открытый программный продукт, который удалось найти, – это сборщик Velvet Assembler [1], хорошо зарекомендовавший себя среди научной общественности. Основываясь на его алгоритмах, нам удалось провести достоверные исследования в области реализации сборки ДНК-молекул с использованием реконфигурированной вычислительной системы (РВС) на основе программируемых логических интегральных схем (ПЛИС).

Алгоритм сборки генома Velvet Assembler в качестве входных данных использует набор коротких чтений (ридов), полученных в результате секвенирования исходной молекулы ДНК. Длина молекулы ДНК обычно варьируется от нескольких миллионов до нескольких миллиардов нуклеотидов. На данный момент применяется технология секвенирования de novo [2]. Она заключается в многократном клонировании длинной молекулы ДНК и дальнейшем «разрезании» ее в случайных местах. Таким образом, появляется достаточное количество небольших участков ДНК, а затем каждый из них читается по отдельности. После прочтения выделенных участков на выходе из секвенатора появляется набор так называемых чтений или же ридов, которые по сути своей являются фрагментами исходного генома.

Анализ Velvet Assembler [3] показал, что задача ассемблирования относится к классу сильносвязанных задач, решаемых методом полного перебора. Поэтому для сборки ДНК применяются преимущественно высокопроизводительные многопро-

цессорные системы, такие как кластерные системы [4, 5], а также графические ускорители [6, 7]. Однако сборка генома даже сравнительно простых организмов требует существенных временных затрат, порой исчисляемых неделями или месяцами [8], а для более сложных ДНК получение результата за приемлемое время требует использования еще больших вычислительных ресурсов. При этом вычислительные мощности даже самых современных кластеров, имеющих сотни вычислительных узлов, объединённых разветвленными коммутационными сетями, как правило, оказываются недостаточными. Во-первых, это связано с тем, что задачи сборки молекул ДНК обладают переменной интенсивностью потоков данных между выполняемыми преобразованиями, причем характер изменения интенсивности зависит от обрабатываемой информации [9, 10, 11]. Во-вторых, высокая степень связности данных задач при решении на кластерах требует хранения промежуточных вычислений, объем которых существенно превышает объем обрабатываемых данных [12]. В связи с этим существует необходимость создания вычислительных систем, отличающихся более подходящей архитектурой для решения задач сборки ДНК-молекул.

В отличие от традиционных вычислительных систем, активно развивающиеся многопроцессорные РВС [13, 14] демонстрируют более высокую эффективность даже при решении задач NP-класса. Возможность адаптировать архитектуру РВС под структуру задачи позволяет на более качественном уровне решать проблемы множества информационно-незначимых операций в алгоритмах. Мощная подсистема коммутаций кардинальным образом снижает требования к объёму хранящихся результатов промежуточных вычислений [15], а применяемая в РВС технология хранения данных в распределённой памяти эффективно борется с проблемой “узкого горла”. Как показывают исследования, скорость решения многих вычислительно-трудоемких сильносвязанных NP-полных задач на РВС на два и более десятичных порядка выше по сравнению с традиционными кластерными системами.

До настоящего времени в мире применялись попытки использовать ПЛИС в качестве сопроцессоров для решения задач сборки молекул ДНК, что не приводило к кардинальному изменению времени решения. Отличительным свойством РВС является применение полей ПЛИС, объединенных высокопроизводительной пространственно-коммутационной сетью для структурного либо структурно-процедурного решения задач NP-класса. Известных попыток решения задач сборки молекул ДНК на РВС не существует.

Целью данного исследования является разработка методов решения задач сборки генома на РВС, использующих структурно-процедурную парадигму вычислений.

Анализ алгоритма. Алгоритм сборщика Velvet Assembler сводится к построению графа де Брюйна [16]. Для этого на вход подается массив ридов, длина каждого из которых составляет k нуклеотидов. В рассматриваемой задаче сборки ДНК *Staphylococcus aureus* [17] для каждого рида создается $k-30$ суффиксов (префиксов) длиной в 31 нуклеотид для определения пересечений между различными ридами. Суффикс представляет собой пару word-antiword, выделяемую из каждого рида, то есть прямое представление и обратно комплементарное. Такой подход обуславливается непредсказуемым размером ридов для каждого конкретного случая [18].

С целью ухода от сравнения между собой непосредственно самих суффиксов каждому из них присваивается уникальный код путем хэширования их абсолютных значений. Каждый полученный хэш очередного рида сравнивается с хэшами всех остальных ридов. В случае совпадения значений происходит формирование аннотации, которая представляет собой массив, содержащий информацию о номе-

ре совпавшего ряда, о номере ряда, с которым произошло совпадение, и параметры, определяющие номера пересекающихся суффиксов. После формирования всех возможных аннотаций происходят создание контигов и удаление ошибок, затем создание скаффолдов и удаление ошибок в них.

Проведенный анализ алгоритма Velvet Assembler [19] показал, что условно задачу ассемблирования ДНК можно представить в виде двух вычислительно трудоёмких подзадач. Первая подзадача, выполняемая по алгоритму VelvetH, – это формирование контигов из пар ридов с одинаковыми суффиксами. Вторая, выполняемая по алгоритму VelvetG, представляет собой процесс формирования скаффолдов из контигов, полученных из предыдущей подзадачи. В рамках данной работы была рассмотрена подзадача VelvetH.

Алгоритм подзадачи VelvetH в свою очередь состоит из нескольких базовых вычислительных фрагментов: кодирования входных ридов во внутренний двухбитный код, создания суффиксов, вычисления хэша и формирования аннотаций. Оценочные исследования показали, что максимальная эффективность вычислительной структуры достигается в случае применения распараллеливания вычислений, направленных на формирование аннотаций. Такой подход позволяет сократить количество используемого аппаратного ресурса в 5,5 раз по сравнению с распараллеливанием всех базовых вычислительных фрагментов.

Реализация на PBC. Была разработана конвейерная вычислительная структура, реализующая подзадачу VelvetH, представленная на рис. 1.

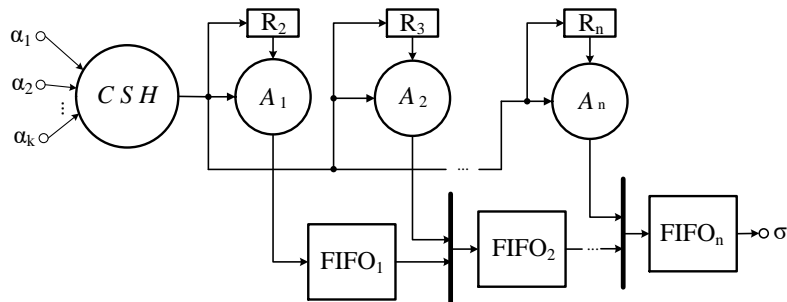


Рис. 1. Вычислительная структура, реализующая базовый подграф подзадачи VelvetH

Рассмотрим её более подробно. На вход $\alpha_1, \alpha_2, \dots, \alpha_k$ поступают потоки кортежей ридов из файла формата .fastq, полученного по технологии Illumina [20]. В блоке CSH выполняется перекодировка входной информации, представляемой в ASCII-кодах, во внутренний двухбитный код сборщика Velvet Assembler, формируются суффиксы и рассчитываются их хэш-значения. Блоки A_1, A_2, \dots, A_n выполняют функции формирования аннотаций так, что один блок A формирует все возможные аннотации для одного ридов, после чего этот рид исключается из формирования последующих аннотаций. Результат вычислений каждого блока A выгружается во внешнюю память через цепочку FIFO₁, FIFO₂, ..., FIFO_n по каналу σ , формируя итоговый массив контигов.

При таком подходе обработка всех ридов осуществляется итерационным способом. Количество итераций определяется исходным числом ридов и количеством n вычислительных блоков A в реализованном на ПЛИС конвейере.

При реализации вычислительной структуры на ПЛИС XKU 095 было установлено, что коэффициент распараллеливания определяется по критическому ресурсу для данной задачи – числу таблиц LUT, поэтому коэффициент n можно рассчитать по формуле

$$n = \frac{V_{LUT \text{ ПЛИС}} - V_{LUT}(C, S, H) - V_{com}}{V_{LUT}(A)},$$

где $V_{LUT \text{ ПЛИС}}$ – число LUT в ПЛИС, V_{LUT} – число LUT, необходимых на реализацию вычислительных фрагментов, V_{com} – число LUT, необходимых на реализацию подсистемы коммутации.

Таким образом, коэффициент распараллеливания будет увеличиваться практически линейно с линейным увеличением параметра $V_{LUT \text{ ПЛИС}}$. Это связано с тем, что масштабировать необходимо только количество n вычислительных блоков A .

Разработанная вычислительная структура конвейера процедуры VelvetH была реализована в составе параллельно-конвейерной программы в среде Xilinx Vivado на ПЛИС XSKU095 ffvb1760-1-с. Испытания проводились на реконфигурируемой вычислительной системе “Tertius”, содержащей четыре ПЛИС XSKU095.

Рабочая тактовая частота синтезированного устройства составила $v=550$ МГц со скажностью подачи данных, равной $s=1$, при этом количество размещенных базовых подграфов в одном кристалле ПЛИС - $n=616$. Производительность разработанного вычислительного конвейера, выполняющего процедуру VelvetH, можно найти по формуле:

$$P = n \cdot v \cdot s, \text{ где } P = 3,38 \cdot 10^{11} \text{ ридов/с.}$$

Время выполнения подзадачи получения контигов молекулы ДНК *Staphylococcus aureus* можно рассчитать по формуле:

$$t = \frac{V}{P},$$

где V – объем данных, вычисляемый по формуле:

$$V = \frac{m+2}{2} \cdot m,$$

где m – количество ридов в задаче сборки ДНК-молекулы *Staphylococcus aureus* $m=2,7 \cdot 10^6$. Объем данных для данной задачи $V=3,64 \cdot 10^{12}$.

При этом время нахождения всех аннотаций на одной ПЛИС XSKU095 составляет $t=10,78$ секунд, а на персональном компьютере с процессором Intel Core i5-4670K с частотой 3400 МГц и объемом памяти 32 Гб составляет 173 секунды.

Таким образом, одна ПЛИС дает прирост производительности в 16 раз, а также практически линейный рост производительности решения задачи при увеличении количества задействованных ПЛИС [21]. Так, если задействовать все четыре ПЛИС, содержащиеся в одном блоке “Tertius”, то производительность возрастёт в 64 раза по сравнению с персональным компьютером.

Анализ эффективности разработанной вычислительной структуры. Проведем сравнение результатов использования PBC для решения задачи ассемблирования ДНК с аналогичной разработкой Varma, B.S.C., Paul, K., Balakrishnan, M., Lavenier, D. [22]. В своей работе авторы применяли ПЛИС Xilinx Virtex-7 XC7V2000T в качестве сопроцессора для расчетов процедуры VelvetH геномного ассемблера Velvet. Был представлен альтернативный подход к аппаратной реализации сборки генома из коротких чтений на ПЛИС. Распараллеливание так же, как и в данной статье, осуществлялось по входным ридам, ускоряя, таким образом, процесс формирования аннотаций. Однако авторы не добились линейного роста производительности при увеличении числа задействованных микросхем, а также разместили меньшее количество базовых подграфов создания аннотаций в большем по размеру кристалле ПЛИС. В итоге максимальное число базовых подграфов в одном кристалле Xilinx XC7V2000T составило 467 [22].

В статье [22] был показан результат работы аппаратной реализации на ПЛИС сборки генома *Escherichia coli*, имеющего $2 \cdot 10^6$ ридов длиной 36 нуклеотидов. По теоретическим оценкам, достигаемое ускорение (по сравнению с процессором Intel Core 2 Duo с частотой 2600 МГц) при условии размещения 616 ступеней оказалось равным примерно 3,5 раза. В то же время, представленная в данной статье реализация процедуры VelvetH при тех же входных данных и том же количестве размещенных в кристалле вычислительных ступеней показала ускорение (по сравнению с процессором Intel Core 2 Duo с частотой 2600 МГц) равное 21 раз. Таким образом, эффективность предложенной реализации на PBC на одном кристалле ПЛИС в 6 раз выше, чем у Varma, B.S.C., Paul, K., Balakrishnan, M., Lavenier, D, а при использовании всего вычислительного блока “Tertius” – возрастает до 24 раз.

Подводя итог, можно говорить о том, что использование PBC на основе ПЛИС в качестве основной элементной базы при решении задач ассемблирования генома методом нового поколения показывает лучшие результаты и более высокую эффективность реализаций по сравнению с реализациями, использующими ПЛИС в качестве сопроцессора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ohashi H., Hasegawa M., Wakimoto K., Miyamoto - Sato E.* Next - generation technologies for multiomics approaches including interactome sequencing // *BioMed Research International*. – 2015. – Vol. 2015. Article No.104209.
2. *Zerbino D., Birney E.* Velvet: algorithms for de novo short read assembly using de Bruijn graphs // *Genome Research*. – 2008. – Vol. 18, No. 5. – P. 821-829.
3. *Ohashi H., Hasegawa M., Wakimoto K., Miyamoto - Sato E.* Next - generation technologies for multiomics approaches including interactome sequencing // *BioMed Research International*. – 2015. – Vol. 2015. Article No.104209.
4. *Chapman J.A., Ho I., Sunkara S., Luo S., Schroth G.P., et al.* Meraculous: De Novo Genome Assembly with Short Paired-End Reads. *PLoS ONE* 6(8)e23501. – 2011. Doi:10.1371/journal.pone.0023501.
5. *Романенко К.В., Сальников А.Н., Алексеевский А.В.* Параллельный метод объединения результатов работы программ по сборке генома // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. – 2016. – Т. 5, № 1. – С. 24-34. DOI: 10.14529/cmse160103.
6. *Сычев А.А.* Анализ геномных данных с использованием графических ускорителей: магистерская диссертация, по специальности 09.04.01.01 «Высокопроизводительные вычислительные системы». Научный руководитель Кузьмин Д.А. – Красноярск, 2016. – 42 с. – URL: http://elib.sfu-kras.ru/bitstream/handle/2311/29230/sychev_zashifrovan.pdf?sequence=2&isAllowed=y (дата обращения 15.11.2018).
7. *Кирилова А.А.* Разработка алгоритма неточного поиска чтений в геноме с применением вычислений на видеокартах: магистерская диссертация, по специальности 01.03.02 «Прикладная математика и информатика». Научный руководитель Шалыто А.А. – Санкт-Петербург, 2014. – С. 45. – URL:<http://is.ifmo.ru/diploma-theses/2015/master/kirillova/kirillova.pdf> (дата обращения 15.11.2018).
8. *Назипова Н.Н., Исаев Е.А., Корнилов В.В., Первухин Д.В., Морозова А.А., Горбунов А.А., Устинин М.Н.* Большие данные в биоинформатике // *Математическая биология и биоинформатика*. – 2017. – № 12:1. – С. 102-119.
9. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.
10. *Сорокин Д.А., Дордопуло А.И.* Методика сокращения аппаратных затрат в сложных системах при решении задач с существенно-переменной интенсивностью потоков данных // *Известия ЮФУ. Технические науки*. – 2012. – № 4 (129). – С. 213-219.
11. *Левин И.И., Сорокин Д.А., Мельников А.К., Дордопуло А.И.* Решение задач с существенно-переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах // *Вестник компьютерных и информационных технологий*. – 2012. – № 2. – С. 49-56.

12. *Каляев А.В., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультиконвейрные вычислительные структуры. – 2-е изд. перераб. и доп. / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с. ISBN 978-5-902982-61-6.
13. *Дордопуло А.И., Каляев И.А., Левин И.И., Семерников Е.А.* Семейство многопроцессорных вычислительных систем с динамически перестраиваемой архитектурой // Многопроцессорные вычислительные и управляющие системы: Материалы научно-технической конференции. – Таганрог, 2007. – С. 11-17.
14. *Каляев И.А., Левин И.И., Семерников Е.А., Дордопуло А.И.* Реконфигурируемые вычислительные системы на основе ПЛИС семейства VERTEX-6 // Параллельные вычислительные технологии (ПАВТ'2011): Труды международной научной конференции. – 2011. – С. 203-211.
15. *Раскладкин М.К.* Библиотека масштабируемых интерфейсов для реконфигурируемых вычислительных систем на основе ПЛИС // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы девятой международной конференции-семинара. – Владимир, 2009. – 438 с. ISBN 978-5-89368-958-7.
16. *Дасгупта С., Панадимитриу Х., Вазирану У.* Алгоритмы: пер. с англ. / под ред. А. Шеня. – М.: МЦНМО, 2014. – 320 с. ISBN 978-5-4439-0236-4.
17. Illumina whole genome shotgun sequencing of genomic DNA library 'Solexa-1123' containing sample ROAD: SEQUENCING_SAMPLE:15760.0. – URL: <https://www.ncbi.nlm.nih.gov/sra/SRR022823> (дата обращения 10.11.2018).
18. *Бойко В.А.* Разработка алгоритма сборки и анализа больших геномов // Молодой ученый. – 2017. – № 3. – С. 27-28. – URL <https://moluch.ru/archive/137/38530/> (дата обращения 10.11.2018).
19. *Zerbino D.R., McEwen G.K., Margulies E.H., Birney E.* Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler // PLoS One. – 2009. – No. 4 (12): e8407.
20. *Алексеева А.Е., Бруснигина Н.Ф.* Возможности и перспективы применения методов массивного параллельного секвенирования в диагностике и эпидемиологическом надзоре за инфекционными заболеваниями // МедиАль. – 2014. – № 2 (12).
21. *Левина А.И.* Аппаратная реализации сборки генома из коротких чтений на основе графа де Бройна // XIV ежегодная молодежная научная конференция студентов, аспирантов и молодых ученых «Достижения и перспективы молодых ученых в интересах развития Юга России». – Ростов-на-Дону, 2018. – С. 75.
22. *Varma B.S.C., Paul K., Balakrishnan M. and Lavenier D.* Hardware acceleration of de novo genome assembly // Int. J. Embedded Systems. – 2017. – Vol. 9, No. 1. – P. 74-89.

REFERENCES

1. *Ohashi H., Hasegawa M., Wakimoto K., Miyamoto - Sato E.* Next - generation technologies for multiomics approaches including interactome sequencing, *BioMed Research International*, 2015, Vol. 2015. Article No.104209.
2. *Zerbino D., Birney E.* Velvet: algorithms for de novo short read assembly using de Bruijn graphs, *Genome Research*, 2008, Vol. 18, No. 5, pp. 821-829.
3. *Ohashi H., Hasegawa M., Wakimoto K., Miyamoto - Sato E.* Next - generation technologies for multiomics approaches including interactome sequencing, *BioMed Research International*, 2015, Vol. 2015. Article No.104209.
4. *Chapman J.A., Ho I., Sunkara S., Luo S., Schroth G.P., et al.* Meraculous: De Novo Genome Assembly with Short Paired-End Reads. *PLoS ONE* 6(8)e23501, 2011. Doi:10.1371/journal.pone.0023501.
5. *Romanenko K.V., Sal'nikov A.H., Alekseevskiy A.V.* Parallel'nyy metod ob'edineniya rezul'tatov raboty programm po sborke genoma [Parallel method of combining the results of genome assembly programs], *Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika* [Vestnik SUSU. Series: Computational Mathematics and Computer Science], 2016, Vol. 5, No. 1, pp. 24-34. DIO: 10.14529/cmse160103.

6. Sychev A.A. Analiz genomnykh dannykh s ispol'zovaniem graficheskikh uskoriteley: masterskaya dissertatsiya, po spetsial'nosti 09.04.01.01 «Vysokoproizvoditel'nye vychislitel'nye sistemy». Nauchnyy rukovoditel' Kuz'min D.A [Analysis of genomic data using graphics accelerators: master's thesis: Master thesis specialty 09.04.01.01 "High-performance computing systems". Scientific advisor Kuzmin D.A.]. Krasnoyarsk, 2016, 42 p. Available at: http://elib.sfu-kras.ru/bitstream/handle/2311/29230/sychev_zashifrovan.pdf?sequence=2&isAllowed=y (accessed 15 November 2018).
7. Kirilova A.A. Razrabotka algoritma netochnogo poiska chteniy v genome s primeneniem vychisleniy na videokartakh: masterskaya dissertatsiya, po spetsial'nosti 01.03.02 «Prikladnaya matematika i informatika». Nauchnyy rukovoditel' Shalyto A.A [Development fuzzy string search algorithm of readings in the genome using computing on video cards: Master thesis, specialty 01.03.02 "Applied Mathematics and Computer Science", scientific advisor Shalyto A.A.]. Sankt-Peterburg, 2014. pp. 45. Available at: <http://is.ifmo.ru/diplomatheses/2015/master/kirilova/kirilova.pdf> (accessed 15 November 2018).
8. Nazipova N.N., Isaev E.A., Kornilov V.V., Pervukhin D.V., Morozova A.A., Gorbunov A.A., Ustinin M.N. Bol'shie dannye v bioinformatike [Big data in bioinformatics], *Matematicheskaya biologiya i bioinformatika* [Mathematical biology and bioinformatics], 2017, No. 12:1, pp. 102-119.
9. Kalyaev A.V., Levin I.I. Modul'no-narashchivaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiey vychisleniy [Modular-scalable multiprocessor system with structural-procedural organization of computing]. Moscow: Yanus-K, 2003, 380 p.
10. Sorokin D.A., Dordopulo A.I. Metodika sokrashcheniya apparatnykh zatrat v slozhnykh sistemakh pri reshenii zadach s sushchestvenno-peremennoy intensivnost'yu potokov dannykh [Technique of reducing hardware costs in complex systems in solution with greatly-varying intensity of data flows], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 4 (129), pp. 213-219.
11. Levin I.I., Sorokin D.A., Mel'nikov A.K., Dordopulo A.I. Reshenie zadach s sushchestvenno-peremennoy intensivnost'yu potokov dannykh na rekonfiguriruemyykh vy-chislitel'nykh sistemakh [Solving problems with essentially variable intensity of data flows on reconfigurable computing systems], *Vestnik komp'yuternyykh i informatsionnykh tekhnologiy* [Bulletin of computer and information technologies], 2012, No. 2, pp. 49-56.
12. Kalyaev A.V., Levin I.I., Semernikov E.A., Shmoylov V.I. Rekonfiguriruemye mul'tikonveyrnye vychislitel'nye struktury [Multicompany reconfigurable computing structures]. 2nd ed., by general ed. of I.A. Kalyaeva. Rostov-on-Don: Izd-vo YUNTS RAN, 2009, 344 p. ISBN 978-5-902982-61-6.
13. Dordopulo A.I. Kalyaev I.A., Levin I.I., Semernikov E.A. Semeystvo mnogoprotsessornykh vychislitel'nykh sistem s dinamicheski prestraivaemoy arkhitekturoy [Family of multiprocessor computer systems with dynamically reconfigurable architecture], *Mnogoprotsessornye vychislitel'nye i upravlyayushchie sistemy: Materialy nauchno-tekhnicheskoy konferentsii* [Multiprocessor computing and control systems. Materials of scientific and technical conference]. Taganrog, 2007, pp. 11-17.
14. Kalyaev I.A., Levin I.I., Semernikov E.A., Dordopulo A.I. Rekonfiguriruemye vychislitel'nye sistemy na osnove PLIS semeystva VERTEX-6 [Reconfigurable computing system based on FPGA family of VERTEX-6], *Parallel'nye vychislitel'nye tekhnologii (PAVT'2011): Trudy mezhdunarodnoy nauchnoy konferentsii* [Parallel computational technologies (PCT ' 2011). Proceedings of the international scientific conference], 2011, pp. 203-211.
15. Raskladkin M.K. Biblioteka masshtabiruemykh interfeysov dlya rekonfiguriruemyykh vychislitel'nykh sistem na osnove PLIS [Library of scalable interfaces for reconfigurable computing systems based on FPGA], *Vysokoproizvoditel'nye parallel'nye vychisleniya na klasternykh sistemakh. Materialy devyatoy mezhdunarodnoy konferentsii-seminara* [High-performance parallel computing on cluster systems: Proceedings of the ninth international conference seminar]. Vladimir, 2009, 438 p. ISBN 978-5-89368-958-7.
16. Dasgupta S., Papadimitriou Kh., Vazirani U. Algoritmy [Algorithms]: transl. from engl., ed. by A. Shenya. Moscow: MTSNMO, 2014, 320 p. ISBN 978-5-4439-0236-4.
17. Illumina whole genome shotgun sequencing of genomic DNA library 'Solexa-1123' containing sample ROAD: SEQUENCING_SAMPLE:15760.0. Available at: <https://www.ncbi.nlm.nih.gov/sra/SRR022823> (accessed 10 November 2018).

18. Boyko V.A. Razrabotka algoritma sborki i analiza bol'shikh genomov [Development algorithm for the assembly and analysis of big genomes], *Molodoy uchenyy* [Young Scientist], 2017, No. 3, pp. 27-28. Available at: <https://moluch.ru/archive/137/38530/> (accessed 10 November 2018).
19. Zerbino D.R., McEwen G.K., Margulies E.H., Birney E. Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler, *PLoS One*, 2009, No. 4 (12): e8407.
20. Alekseeva A.E., Brusnigina N.F. Vozmozhnosti i perspektivy primeneniya metodov massivnogo parallelnogo sekvenirovaniya v diagnostike i epidemiologicheskom nadzore za infektsionnymi zabolevaniyami [Opportunities and prospects of application of methods of massive parallel sequencing in diagnostics and epidemiological surveillance of infectious diseases], *Medial'* [Medial], 2014, No. 2 (12).
21. Levina A.I. Apparatnaya realizatsii sborki genoma iz korotkikh chteniy na osnove grafa de Bryuyna [Hardware implementation of genome Assembly from short readings based on de Bruijn graph], *XIV ezhgodnaya molodezhnaya nauchnaya konferentsiya studentov, aspirantov i molodykh uchenykh «Dostizheniya i perspektivy molodykh uchenykh v interesakh razvitiya Yuga Rossii»* [XIV annual youth scientific conference of students, postgraduates and young scientists "Achievements and prospects of young scientists in the interests of development of the South of Russia"]. Rostov-on-Don, 2018, pp. 75.
22. Varma B.S.C., Paul K., Balakrishnan M. and Lavenier D. Hardware acceleration of de novo genome assembly, *Int. J. Embedded Systems*, 2017, Vol. 9, No. 1, pp. 74-89.

Статью рекомендовал к опубликованию д.т.н. Э.В. Мельник.

Левина Анастасия Ильинична – ООО"НИЦ супер-ЭВМ и нейрокомпьютеров"; e-mail: anastasijalewina@yandex.ru; 347900, г. Таганрог, пер. Итальянский, 106; тел.: +78634612111; конструктор.

Семерникова Евгения Евгеньевна – e-mail: semernikova_e@mail.ru; научный сотрудник.

Сорокин Дмитрий Анатольевич – e-mail: jotun@inbox.ru; начальник отдела прикладного программного обеспечения; к.т.н.

Levina Anastasia Ilinichna – Supercomputers and Neurocomputers Research Center; e-mail: anastasijalewina@yandex.ru; 347900, Taganrog, Italyansky lane, 106; phone: +78634612111; designer.

Semernikova Evgeniya Evgenievna – e-mail: semernikova_e@mail.ru; research scientist.

Sorokin Dmitry Anatolievich – e-mail: jotun@inbox.ru; head of application software department; cand. of eng. sc.

УДК 004.273

DOI 10.23683/2311-3103-2018-8-212-221

А.В. Подопригора, М.Д. Чекина

РЕШЕНИЕ РАЗРЕЖЕННЫХ СЛАУ БОЛЬШОЙ И СВЕРХБОЛЬШОЙ РАЗМЕРНОСТИ МНОГОСЕТОЧНЫМ МЕТОДОМ НА PBC

Рассматривается возможность применения PBC для решения больших и сверхбольших разреженных систем линейных алгебраических уравнений. На сегодняшний день компьютерное моделирование приобретает все большую актуальность и применяется во многих областях науки и техники, заменяя собой натурные модели, а также позволяя прогнозировать природные процессы и явления. В основе подобного моделирования, как правило, лежат физико-математические модели, дискретизация которых естественным образом приводит к появлению систем линейных алгебраических уравнений (СЛАУ), где базовый оператор имеет разреженную структуру. Решение больших и сверхбольших разреженных СЛАУ позволит увеличить точность вычислений и даст возможность обрабатывать