

Kholopova Nina Vladimirovna – Southern Federal University; e-mail: xolopova.nina@mail.ru; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; masters degree.

Samoylov Aleksey Nikolaevich – e-mail: asamoylov@sfedu.ru; the department of system analysis and telecommunications; associate professor.

Kuliev Elmar Valerievich – e-mail: elmar_2005@mail.ru; the department of system analysis and telecommunications; assistant.

УДК 004.382.2

И.И. Левин, А.В. Пелипец, Д.А. Сорокин

РЕШЕНИЕ ЗАДАЧИ LU-ДЕКОМПОЗИЦИИ НА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ: ОЦЕНКА И ПЕРСПЕКТИВЫ

Исследуется оценка производительности реконфигурируемых вычислительных систем в задаче LU-декомпозиции квадратной матрицы и рассматриваются перспективы реализации нового метода вычислений – без использования внешней памяти. Разложение матрицы на треугольные множители является основой многих алгоритмов, связанных с численным решением систем линейных алгебраических уравнений. Декомпозиция квадратной матрицы в виде произведения нижней треугольной матрицы L и верхней треугольной матрицы U обладает вычислительной сложностью, которая кубически зависит от количества уравнений в системе. Поэтому при решении задач с описанием сложных процессов и большими массивами исходных данных уже давно используются вычислительные мощности специализированных многопроцессорных систем (суперкомпьютеров). Однако, несмотря на постоянный рост производительности современных суперкомпьютеров, построенных по кластерной архитектуре, всем им присущ ряд принципиальных недостатков, ограничивающих реальную производительность, в том числе в задачах линейной алгебры. Прежде всего, это касается проблемы накладных расходов на реализацию межпроцессорных взаимодействий и хранения промежуточных результатов во внешней памяти. Предлагаемый в данной статье метод, в отличие от известных кластерных решений, позволяет осуществлять LU-разложение большой матрицы ($n=10^4$) в темпе поступления задачи, без использования внешней памяти. Метод основан на использовании многопроцессорных реконфигурируемых вычислительных систем (РВС), построенных на основе программируемых логических интегральных схем (ПЛИС). Данный метод осуществим при наличии в системе реконфигурируемого аппаратного ресурса, достаточного для конвейерной реализации полного информационного графа задачи. Предварительные исследования показывают, что хотя такая реализация и возможна на существующих реконфигурируемых суперкомпьютерах, её удельная производительность в расчете на один вычислительный модуль остается низкой. В обозримой перспективе, при сохранении существующих темпов роста тактовых частот и логических емкостей ПЛИС, возможно создание реализации LU-разложения матрицы порядка $n=10^4$ при помощи одно вычислительного модуля реконфигурируемой системы.

Реконфигурируемые вычислительные системы; программируемые логические интегральные схемы; LU-разложение; Linpack Benchmark; удельная производительность.

I.I. Levin, A.V. Pelipets, D.A. Sorokin

ESTIMATION AND PROSPECTS OF SOLVING LU- DECOMPOSITION ON RECONFIGURABLE COMPUTER SYSTEMS

This paper examines the estimation of reconfigurable computer systems to solving LU-decomposition of a square matrix. Factorizations of matrix into low/upper triangular form are in the base of many algorithms for performing numerical linear algebra computations. The decomposition of a square matrix into a lower triangular matrix L and an upper triangular matrix U

has cubic complexity with respect to the size of the linear system. Therefore supercomputers are long used to solve complex tasks with large-scale massive data. Despite the regular performance expansion of cluster supercomputers power, their performance is limited in tasks of linear algebra. This primarily concerns computational costs related to interprocessor communication and storage of partial result. Unlike cluster supercomputers, reconfigurable computer systems based on FPGA technology enable LU-factorization of a large-size matrix ($n=10^4$) by real-time processing, without using external memory. This method is feasible in the presence of hardware resource for pipeline implementation of the information graph. Preliminary studies shows that modern reconfigurable computer systems makes such implementation possible, but a specific performance rating of supercomputer module is low. If increasing of clock frequency and FPGA logical resource will continue, LU-factorization implementation using single reconfigurable computational module.

Reconfigurable computer systems; FPGA; LU-factorization; Linpack Benchmark; specific performance.

Введение. Численное решение систем линейных алгебраических уравнений (СЛАУ) применяется для широкого класса задач вычислительной математики, которые используются в самых различных областях научных исследований: от моделирования и интерпретации экспериментальных данных до синтетических тестов оценки производительности многопроцессорных вычислительных систем (суперкомпьютеров) [1–4].

Вычислительным ядром многих алгоритмов решения СЛАУ являются различные виды декомпозиций исходной матрицы, к числу важнейшим из них принадлежат разложения на два треугольных множителя [5]. Большая вычислительная сложность алгоритмов разложения матриц высокого порядка является причиной того, что в настоящее время эта задача чаще всего решается с использованием высокопроизводительных многопроцессорных систем кластерной архитектуры. Обычно такие системы построены на базе универсальных многоядерных процессоров, либо реже на графических процессорных устройствах (ГПУ), что требует адаптации алгоритмов и их реализаций к параллельной вычислительной структуре [6].

По этой причине большинство исследований оценки производительности многопроцессорных решений сводится к вопросам оптимизации, масштабирования и повышения эффективности именно кластерных вычислительных ресурсов [7–9]. При этом зачастую упускается из виду факт существования реконфигурируемых вычислительных систем (РВС), которые не требуют модификаций алгоритмов, поскольку могут быть архитектурно адаптированы к информационной структуре исходной задачи [10]. Обычно реконфигурируемые системы в задачах матричных декомпозиций рассматриваются лишь в качестве вспомогательных вычислителей [11] или как аналог ГПУ [12], поскольку имеют множество аппаратно реализованных функциональных элементов, работающих параллельно. Такой подход не выходит за рамки методов, известных и широко применяемых в кластерных вычислениях. Поэтому представляется актуальным исследование производительности, которую можно достичь, используя новые методы реализации разложения матриц высокого порядка на РВС.

Проблемы аппаратной реализации LU-разложения. Как уже говорилось, одним из основных численных методов решения СЛАУ является треугольная факторизация. Метод базируется на разложении (факторизации) квадратной матрицы A в виде произведения двух треугольных матриц L и U таким образом, что $LU=A$.

Существующие решения LU -разложения для кластерных систем требуют распараллеливания алгоритма на уровне разделения исходной матрицы на логические блоки, каждый из которых достается одному из процессоров [13]. Применение такого подхода не позволит эффективно реализовать алгоритм на реконфигурируемых вычислительных системах, где главным аппаратным ресурсом являются

ПЛИС, объединенные в единое вычислительное поле. Причиной этого является то, что максимальная таковая частота даже наиболее быстродействующих кристаллов значительно ниже, чем у универсальных процессоров. Несмотря на это, у многопроцессорной архитектуры, построенной на основе ПЛИС, есть преимущества, которые принципиально недостижимы кластерными системами. Речь идет, в частности, о реализации обмена процессор-процессор, которая позволяет организовать мультиконвейерные вычисления без использования внешней памяти.

Современные модели межпроцессорного обмена, в соответствии с которыми функционируют кластерные суперЭВМ, основаны либо на использовании общей памяти, либо на передаче сообщений по высокоскоростной сети [14]. Оба этих варианта подразумевают значительные накладные расходы в виде времени выполнения транзакций и необходимости разрешения конфликтов доступа к общим ресурсам [15].

При этом неважно, какую форму параллельной обработки данных будет реализовывать кластерная система – с общей памятью, конвейерную или какую-то иную. Особенности жесткой архитектуры традиционных суперкомпьютеров таковы, что упомянутые вычислительные издержки неизбежно присущи любой из этих форм.

В отличие от кластерных систем модели межпроцессорного обмена РВС не ограничены жесткой архитектурой аппаратного ресурса. Гибкость реконфигурируемых систем позволяет организовать прямое взаимодействие процессорных элементов как внутри кристалла, так и между иерархическими вычислительными структурами более высокого ранга (между ПЛИС, модулями, стойками).

Существует большое количество вариантов LU -факторизации, ориентированных на определенные вычислительные архитектуры и способы хранения матрицы в памяти [16]. Однако особенность данного исследования состоит в том, чтобы, не выбирая наиболее подходящий вариант алгоритма, максимально эффективно реализовать любой из при помощи нового метода, исключающего использование внешней памяти.

Для этой цели был взят алгоритм LU -разложения матрицы, используемый в тесте Linpack Benchmark, по результатам которого составляется рейтинг TOP-500 наиболее высокопроизводительных суперкомпьютеров мира [17].

Если принять условие, что исходная матрица A допускает устойчивое LU -разложение и не требуется процесс выбора опорного элемента, то алгоритм декомпозиции матрицы в Linpack Benchmark принимает вид, показанный в листинге 1.

Листинг 1. Псевдокод LU -разложения матрицы Linpack Benchmark

Входные данные: a_{ij} – элементы невырожденной квадратной матрицы A порядка n ;

Выходные данные: a_{ij} – элементы нижнетреугольной матрицы L и верхнетреугольной матрицы U , вычисленные на месте матрицы A .

```

1  for k ← 1 to n – 1 do
2      for i ← k + 1 to n do
3           $m_{k,i} ← a_{k,i}/a_{k,k}$  /* save in  $a_{k,i}$  */
4      end for
5      for j ← k + 1 to n do
6          for i ← k + 1 to n do
7               $a_{j,i} ← a_{j,i} – m_{k,i} · a_{j,k}$ 
8          end for
9      end for
10 end for

```

Приведенный алгоритм состоит в выполнении вложенных циклов по переменным k, i, j , каждая из которых является одним из индексов двумерного массива коэффициентов СЛАУ.

Обращает на себя внимание схожесть математических операций алгоритма LU -факторизации и метода Гаусса [18]. Но, в отличие от метода Гаусса, прямой ход которого приводит исходную матрицу к верхнетреугольной форме (все элементы ниже главной диагонали равны нулю), процесс вычисления LU -разложения требует вычислять и хранить также элементы нижнетреугольной матрицы. Это обстоятельство вносит свои особенности в схемотехническую реализацию данного алгоритма, касающиеся не только управления потоками данных, но и структуры базового информационного подграфа задачи.

Вычисление LU -разложения на РВС в темпе поступления задачи без использования внешней памяти. Совокупность базовых информационных подграфов задачи, мультиплицированных по итерациям, составляет полный информационный граф задачи (рис. 1).

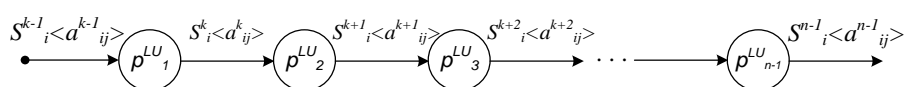


Рис. 1. Информационный граф LU -разложения матрицы Linpack Benchmark

Во время итерации k на вход подграфа p^{LU}_k приходит вектор S^{k-1}_i , состоящий из элементов a^{k-1}_{ij} строки i , которые были сформированы на итерации $k-1$. Выходной вектор S^k_i подграфа представляет собой элементы a^k_{ij} строки i , вычисленные во время текущей итерации k .

Возможность построения архитектуры для решения LU -разложения без использования внешней памяти изначально сводится к наличию вычислительного ресурса для реализации полного графа задачи, включающего все базовые подграфы. Если такого ресурса недостаточно, то решение задачи возможно путем применения метода редуцирования по числу базовых подграфов, однако это требует использования внешней памяти для хранения промежуточных результатов и коммутационного оборудования для прямых и обратных связей с памятью. В работе [19] на примере решения задачи СЛАУ методом Гаусса показано, что объем ресурса, затрачиваемый на аппаратную реализацию редуцированного информационного графа задачи, не уменьшается пропорционально числу реализованных базовых подграфов, а изменяется с некоторым коэффициентом, который отражает дополнительные аппаратные затраты.

Вычислительная структура базового подграфа задачи LU -факторизации представлена на рис. 2.

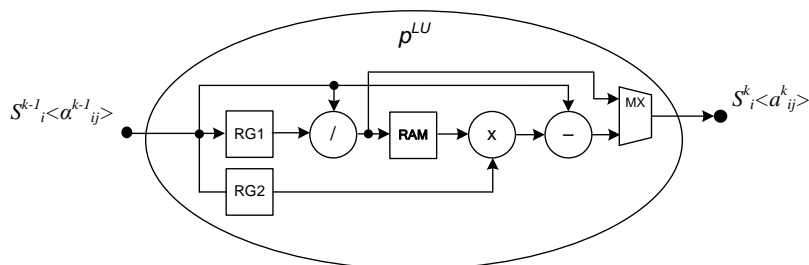


Рис. 2. Базовый подграф LU -разложения матрицы Linpack Benchmark

Функциональные блоки внутри базового подграфа распределены следующим образом. Блок деления в соответствии со строкой 3 алгоритма выполняет деление текущего элемента строки $a_{k,i}$ на первый элемент этой же строки $a_{k,k}$. Результат деления $m_{k,i}$, представляющий окончательно сформированный элемент $a_{k,i}$ разложенной матрицы, запоминается в блоке памяти RAM и одновременно поступает через мультиплексор на выход для последующего транзита результатов разложения. Блок умножения выполняет операцию нормирования – умножение текущего элемента $a_{j,k}$ на нормировочный коэффициент $m_{k,i}$. Блок вычитания отнимает из текущего элемента $a_{j,i}$ результат нормирования $m_{k,i} \cdot a_{j,k}$.

Элементы α_{ij}^k выходной строки i , распределенные таким образом, что начальный из них α_{i1}^k принадлежит первому элементу столбца нижнетреугольной подматрицы L , а все последующие являются элементами первой строки верхнетреугольной подматрицы U . Таким образом, результатом работы алгоритма на каждой итерации k будет строка матрицы U (ведущая строка i для итерации $k+1$), столбец матрицы L и пересчитанные элементы квадратной подматрицы (рис. 3).

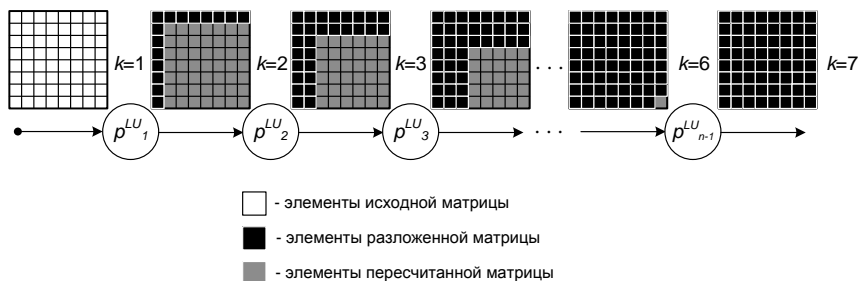


Рис. 3. Состояние элементов матрицы порядка $n=8$ после итерации k

В конвейерной реализации LU -разложения каждая ступень конвейера соответствует одной итерации алгоритма (базовому подграфу). Если при разложении квадратной матрицы порядка n имеющегося аппаратного ресурса достаточно для размещения всех $n-1$ ступеней конвейера, то задача может быть решена в темпе поступления данных.

Исходя из количества функциональных блоков, задействованных в одной ступени, можно рассчитать необходимый логический ресурс для создания полного вычислительного конвейера на реконфигурируемых системах, построенных на базе ПЛИС.

Для вычислительных систем, построенных на элементной базе фирмы Xilinx серии Virtex 7 (ПЛИС XC7VX485T), количество утилизируемых ресурсов одной ступени конвейера, обрабатывающей строки матрицы порядка $n=10^4$ (64-х разрядные числа в стандарте IEEE754), представлено в табл. 1.

Таблица 1

Функциональный элемент	FF	LUT	DSP	36K BRAM	18K BRAM
Делитель	5734	5551	-	-	-
Умножитель	1140	591	6	-	-
Вычитатель	2046	1943	-	-	-
Память	-	-	-	17	2
Итого	8 920	8 085	6	17	2

Всего ПЛИС XC7VX485T содержит 607 200 FF, 303 600 LUT, 2 800 DSP, 1 030 36K BRAM, 2060 18K BRAM.

Из приведенных данных следует, что наиболее критичным ресурсом реализации является LUT, поэтому одной ПЛИС достаточно для размещения $303600/8085=37$ ступеней конвейера. Базовый модуль, состоящий из 8 ПЛИС, позволит организовать цепочку из 296 ступеней, а для реализации всех 9999 ступеней конвейера потребуется вычислительный ресурс из 34 базовых модулей.

При данной конвейерной реализации время решения задачи LU -разложения будет вычисляться по следующей формуле:

$$t = \frac{n^2 + \Delta \cdot (n-1)}{f} \cdot \frac{64}{d},$$

где n – размерность квадратной матрицы, равная 10^4 ; Δ – количество тактов для заполнения одной ступени конвейера ≈ 250 ; f – гарантированная частота обращения к микросхемам памяти, равная $400 \cdot 10^6$ Гц; d – разрядность канала данных между ПЛИС и микросхемой памяти, равная 32.

Отсюда следует, что время решения задачи LU -разложения на PBC, построенной на ПЛИС Xilinx серии Virtex 7 (XC7VX485T), составит примерно 0,5 сек.

Выполнение теста Linpack Benchmark на персональном компьютере (Intel Core i5-3570K, 3,4 ГГц, 8 Гб ОЗУ) показало, что время решения СЛАУ ($n=10^4$) составило 25 сек.

Таким образом, общее ускорение выполнения теста Linpack Benchmark ($n=10^4$) на PBC по сравнению с ПК составит примерно 50 раз. В пересчете на один вычислительный модуль, состоящий из восьми ПЛИС, ускорение составит примерно 1,4 раза.

При переходе на элементную базу ПЛИС фирмы Xilinx 8-й серии (семейство «UltraScale») время решения задачи изменится в соответствии с новой максимальной тактовой частотой $500 \cdot 10^6$ Гц и составит примерно 0,4 сек. При этом будет задействован вычислительный ресурс из 18 базовых модулей, состоящих из восьми ПЛИС XC7VU095.

На сегодняшний день тенденция роста тактовой частоты и логической емкости ПЛИС с выходом каждой новой серии фирмы Xilinx указывает, в среднем, на 25 % увеличение этих показателей при переходе на очередное семейство [20–22].

В связи с этим можно предположить, что в 9-й серии ПЛИС Xilinx будут достигнуты максимальная тактовая частота 625 МГц и логическая ёмкость, достаточная для реализации конвейера из 9999 ступеней в 56 кристаллах, расположенных на семи базовых модулях.

Заключение. Данное исследование показывает перспективность использования реконфигурируемых вычислительных систем для решения задачи LU -декомпозиции при помощи нового метода, позволяющего не использовать внешнюю память для хранения промежуточных результатов.

Несмотря на недостижимость приемлемых результатов производительности в настоящее время, статистические оценки скорости и аппаратных затрат решения задачи LU -разложения на PBC показывают, что после появления ПЛИС Xilinx 11-й серии следует ожидать принципиально качественного скачка показателя «производительность/объем аппаратного ресурса». Это значит, что потенциальная производительность, которой достигнут ПЛИС к тому времени, позволит решать СЛАУ с количеством неизвестных $n=10^4$ в темпе чтения матрицы из внешней памяти на частоте порядка 980 МГц, используя для этого один реконфигурируемый вычислительный модуль.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Charles L. Byrne. Applied and Computational Linear Algebra: A First Course. University of Massachusetts Lowell, 2013. – P. XXIII-XXIV.
2. John R. Bacon, Thomas P. Kendall, Thomas Mussmann, Robert Palais, Victor E. Trujillo, II, Frank Wattenberg. Climate Science: Why Mathematicians Should Be Interested // Electronic Proceedings of the Twenty-fifth Annual International Conference on Technology in Collegiate Mathematics, Boston, Massachusetts. March 21-24, 2013. – P. 351-387.

3. *Piotr Luszczyk, Jakub Kurzak, Jack Dongarra*. Looking back at dense linear algebra software // *Journal of Parallel and Distributed Computing*. – July 2014. – Vol. 74, Issue 7. – P. 2548-2560.
4. *Jack Dongarra, Piotr Luszczyk*. LINPACK Benchmark. *Encyclopedia of Parallel Computing*. Springer US, 2011. – P. 1033-1036.
5. *Воеводин В.В., Кузнецов Ю.А.* Матрицы и вычисления. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 320 с.
6. *Kurzak J., Luszczyk P., Faverge M., Dongarra J.* LU Factorization with Partial Pivoting for a Multicore System with Accelerators // *IEEE Transactions on Parallel & Distributed Systems*. – Aug. 2013. – Vol. 24, No. 8. – P. 1613-1621.
7. *Yamazaki I., Li X.* New scheduling strategies and hybrid programming for a parallel right-looking sparse LU factorization algorithm on multicore cluster systems. *IPDPS*, 2012. – P. 619-630.
8. *Badawy M.O., Hanafy Y.Y., Eltarras R.* LU factorization using multithreaded system. *Computer Theory and Applications (ICCTA)*, 2012. – P. 9-14.
9. *Agullo E., Augonnet C., Dongarra J., Faverge M., Langou J., Ltaief H., Tomov S.* LU factorization for accelerator-based systems // *9th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 11)*. Sharm El-Sheikh, Egypt, 2011. – P. 217-224.
10. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультиконвейерные вычислительные структуры. – Ростов-на-Дону: ЮИЦ РАН, 2008. – 397 с.
11. *Wei Wu, Yi Shan, Xiaoming Chen, Yu Wang, Huazhong Yang*. FPGA Accelerated Parallel Sparse Matrix Factorization for Circuit Simulations. *7th International Symposium, ARC 2011*, Belfast, UK, March 23-25, 2011. – P. 302-315.
12. *Prawat Nagvajara, Chika Nwankpa, Jeremy Johnson*. Reconfigurable Hardware Accelerators for Power Transmission System Computation. *High Performance Computing in Power and Energy Systems*. Springer Berlin Heidelberg, 2013. – P. 211-228.
13. <http://www.netlib.org/benchmark/hpl/algorithm.html> (дата обращения: 25.06.2015).
14. *Старченко А.В., Берцул В.Н.* Методы параллельных вычислений: Учебник. – Томск: Изд-во Том. ун-та, 2013. – С. 11.
15. *Каляев И.А., Левин И.И., Семерников Е.А.* Принципы построения многопроцессорных вычислительных систем на основе ПЛИС // *Вестник Бурятского государственного университета*. Сер. 9: математика и информатика. – Улан-Удэ: Изд-во Бурятск. гос. ун-та, 2008. – С. 184-196.
16. *Ортега Дж.* Введение в параллельные и векторные методы решения линейных систем. – М.: Мир, 1991. – 376 с.
17. <http://www.top500.org/> (дата обращения: 25.06.2015).
18. *Воеводин В.В.* Вычислительные основы линейной алгебры. – М.: Наука, 1977. – 304 с.
19. *Сорокин Д.А.* Методы решения задач с переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах: дис. ... канд. техн. наук. – Таганрог, 2012. – С. 51-58.
20. *Тарасов И.* Эволюция ПЛИС серии Virtex // *Компоненты и технологии*. – 2005. – № 1.
21. *Тарасов И.* Анализ предварительных характеристик FPGA «серии 7» фирмы Xilinx // *Компоненты и технологии*. – 2010. – № 8.
22. *Тарасов И.* Описание архитектуры FPGA семейств UltraScale компании Xilinx // *Компоненты и технологии*. – 2014. – № 2.

REFERENCES

1. *Charles L. Byrne*. Applied and Computational Linear Algebra: A First Course. University of Massachusetts Lowell, 2013, pp. XXIII-XXIV.
2. *John R. Bacon, Thomas P. Kendall, Thomas Mussmann, Robert Palais, Victor E. Trujillo, II, Frank Wattenberg*. Climate Science: Why Mathematicians Should Be Interested, *Electronic Proceedings of the Twenty-fifth Annual International Conference on Technology in Collegiate Mathematics, Boston, Massachusetts. March 21-24, 2013*, pp. 351-387.
3. *Piotr Luszczyk, Jakub Kurzak, Jack Dongarra*. Looking back at dense linear algebra software, *Journal of Parallel and Distributed Computing*, July 2014, Vol. 74, Issue 7, pp. 2548-2560.
4. *Jack Dongarra, Piotr Luszczyk*. LINPACK Benchmark. *Encyclopedia of Parallel Computing*. Springer US, 2011, pp. 1033-1036.
5. *Voevodin V.V., Kuznetsov Yu.A.* Matritsy i vychisleniya [Matrix and calculations]. Moscow: Nauka. Glavnaya redaktsiya fiziko-matematicheskoy literatury, 1984, 320 p.

6. Kurzak J., Luszczek P., Faverge M., Dongarra J. LU Factorization with Partial Pivoting for a Multicore System with Accelerators, *IEEE Transactions on Parallel & Distributed Systems*, Aug. 2013, Vol. 24, No. 8, pp. 1613-1621.
7. Yamazaki I., Li X. New scheduling strategies and hybrid programming for a parallel right-looking sparse LU factorization algorithm on multicore cluster systems. IPDPS, 2012, pp. 619-630.
8. Badawy M.O., Hanafy Y.Y., Eltarras R. LU factorization using multithreaded system. *Computer Theory and Applications (ICCTA)*, 2012, pp. 9-14.
9. Agullo E., Augonnet C., Dongarra J., Faverge M., Langou J., Ltaief H., Tomov S. LU factorization for accelerator-based systems, *9th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 11)*. Sharm El-Sheikh, Egypt, 2011, pp. 217-224.
10. Kalyaev I.A., Levin I.I., Semernikov E.A., Shmoylov V.I. Rekonfiguriruemye mul'tikonveyernye vychislitel'nye struktury [Multiconference reconfigurable computing structures]. Rostov-on-Don: YuNTs RAN, 2008, 397 p.
11. Wei Wu, Yi Shan, Xiaoming Chen, Yu Wang, Huazhong Yang. FPGA Accelerated Parallel Sparse Matrix Factorization for Circuit Simulations. 7th International Symposium, ARC 2011, Belfast, UK, March 23-25, 2011, pp. 302-315.
12. Prawat Nagvajara, Chika Nwankpa, Jeremy Johnson. Reconfigurable Hardware Accelerators for Power Transmission System Computation. *High Performance Computing in Power and Energy Systems*. Springer Berlin Heidelberg, 2013, pp. 211-228.
13. Available at: <http://www.netlib.org/benchmark/hpl/algorithm.html> (accessed 25 June 2015).
14. Starchenko A.V., Bertsun V.N. Metody parallel'nykh vychisleniy: Uchebnik [Methods parallel computing: a Tutorial]. Tomsk: Izd-vo Tom. un-ta, 2013, pp. 11.
15. Kalyaev I.A., Levin I.I., Semernikov E.A. Printsipy postroeniya mnogoprotsessornykh vychislitel'nykh sistem na osnove PLIS [The principles of multiprocessor systems based on FPGA], *Vestnik Buryatskogo gosudarstvennogo universiteta. Ser. 9: matematika i informatika* [Bulletin of the Buryat state University. 9 series: Mathematics and Informatics]. Ulan-Ude: Izd-vo Buryatsk. gos. un-ta, 2008, pp. 184-196.
16. Ortega Dzh. Vvedenie v parallel'nye i vektornye metody resheniya lineynykh system [Introduction to parallel and vector methods for solving linear systems]. Moscow: Mir, 1991, 376 p.
17. Available at: <http://www.top500.org/> (accessed 25 June 2015).
18. Voevodin V.V. Vychislitel'nye osnovy lineynoy algebry [Computational principles of linear algebra]. Moscow: Nauka, 1977, 304 p.
19. Sorokin D.A. Metody resheniya zadach s peremennoy intensivnost'yu potokov dannykh na rekonfiguriruemykh vychislitel'nykh sistemakh. Dis. kand. tekhn. nauk [Problem-solving methods with variable intensity of the data streams on reconfigurable computing systems. Cfnd. of eng. sc. diss.]. Taganrog, 2012, pp. 51-58.
20. Tarasov I. Evolyutsiya PLIS serii Virtex [Evolution is a series of FPGAs Virtex], *Komponenty i tekhnologii* [Components and Technologies], 2005, No. 1.
21. Tarasov I. Analiz predvaritel'nykh kharakteristik FPGA «serii 7» firmy Xilinx [The analysis of the characteristics of FPGA "series 7" by Xilinx], *Komponenty i tekhnologii* [Components and Technologies], 2010, No. 8.
22. Tarasov I. Opisanie arkhitektury FPGA semeystv UltraScale kompanii Xilinx [Description of the architecture of the UltraScale FPGA families of Xilinx company], *Komponenty i tekhnologii* [Components and Technologies], 2014, No. 2.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор А.И. Сухинов.

Левин Илья Израилевич – Научно-исследовательский институт многопроцессорных систем им. А.В. Каляева федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет»; e-mail: levin@mvs.tsure.ru; 347922, г. Таганрог, ул. Ленина, 224/1, кв. 65; тел.: 88634623226; зам. директора по науке; д.т.н.

Пелипец Андрей Владимирович – e-mail: pelipets@mail.ru; 347924, г. Таганрог, ул. С. Лазо, 1, кв. 36; тел.: +78634315491; младший научный сотрудник.

Сорокин Дмитрий Анатольевич – e-mail: jotun@inbox.ru; 347922, г. Таганрог, пер. Украинский, 21, кв. 30.; тел.: +78634315491; старший научный сотрудник.

Levin Ilya Israilevich – Kalyaev Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University; e-mail: levin@mvs.tsure.ru; 224/1, Lenin street, ap. 65, Taganrog, 347922, Russia; phone: +78634623226; deputy director of science; dr. of eng. sc.

Pelipets Andrey Vladimirovich – e-mail: pelipets@mail.ru; 1, S. Lazo street, ap. 36, Taganrog, 347924, Russia; phone: +78634315491; research assistant.

Sorokin Dmitry Anatolievich – e-mail: pelipets@mail.ru; 21, Ukrainskiy Lane, ap. 30, Taganrog, 347922, Russia; phone: +78634315491; senior staff scientist.

УДК 519.711.3

С.Н. Никольский, И.Ф. Сурженко

**СИСТЕМНЫЙ СТРУКТУРНЫЙ ТИП:
«ДИНАМИЧЕСКАЯ СИСТЕМА ОБЪЕКТОВ»**

В современной теории компьютерных информационных систем особое место уделяется исследованиям процесса концептуального анализа. Как результат возникли технологии структурного анализа, которые представляют собой детализацию процесса моделирования на концептуальном уровне, в моделях жизненного цикла программного обеспечения. В частности, к ним относятся ARIS, UML, BpML и другие технологии, использующие объектно- и процессно-ориентированный подходы. Становится понятным, что технология структурного анализа определяется совокупностью понятий, в терминах которых строится модель объекта автоматизации, т.е. онтологией, принимаемой разработчиком. Необходимым требованием к онтологии является ее универсальность, понимаемая как максимальная возможная независимость от типа объекта автоматизации, обеспечивающая широту применения соответствующей технологии структурного анализа. Понятие система образует универсальную онтологию, в которой значения называющей формы «система» детализируются в общей и математической теории систем, системном анализе и исследовании операций. Развитие технологии структурного анализа связывают с системно-ориентированным подходом, представленным в форме SysML, что в большей мере отвечает целям системной инженерии. Целью статьи является исследование возможности использования метаонтологии «объект» для построения метаонтологии «динамическая система объектов». Основная задача состоит в построении структурной модели динамической системы объектов. В работе показано, что структурная модель динамической системы объектов может быть построена на основе метамоделей естественной метаонтологии с использованием принципа онтологической редукции.

Онтологический подход; структурное моделирование; системная инженерия системный структурный тип; пространство состояний объектов; динамическая система объектов.

S.N. Nikolsky, I.F. Surgenko

STRUCTURAL SYSTEM TYPE: DYNAMIC SYSTEM OF OBJECTS

The research of process of structural analysis takes the important place in the modern theory of computerized information systems. As a result we have the technologies of structural analysis, which represent the detail of the modeling process at a conceptual level, from the software life cycle models. In particular, these include ARIS, UML, BpML, and other technologies that use object - and process-oriented approaches. It becomes clear that the technology of structural analysis is determined by a set of concepts in terms of which the object model of automation, i.e., ontology, taken by the developer. A necessary requirement of the ontology is its universality, understood as the highest possible independence from the type of automation object that provides the breadth of application of appropriate technology of structural analysis. The concept of system forms a universal ontology, in which the values of the calling form "system" is detailed in the Gen-