

УДК 519.24

## АЛГОРИТМ ПАРАЛЛЕЛЬНОГО ОБЪЕДИНЕНИЯ МАССИВОВ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ\*

А.А. ВОЕВОДА<sup>1</sup>, Д.О. РОМАННИКОВ<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор кафедры автоматики. E-mail: ucit@ucit.ru

<sup>2</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры автоматики. E-mail: dmitry.romannikov@gmail.com

Задача объединения двух массивов элементов с одновременной их сортировкой и удалением повторяющихся элементов является одной из фундаментальных задач в компьютерных науках и часто встречается как часть других алгоритмов. В данной работе рассматривается алгоритм ее решения для параллельного и асинхронного случая. Из-за наличия параллельности и асинхронности для решения поставленной задачи был выбран инструмент сетей Петри как наиболее подходящий. Для решения поставленной задачи используется бинарное дерево поиска (binarysearchtree), адаптированное для выполнения параллельной операции вставки элементов. Свойства построения бинарного дерева поиска, а именно то, что левый элемент должен быть меньше, чем родительский, а правый – больше родительского, позволяют ему выполнять сортировку при добавлении очередного элемента. Операция удаления повторяющихся элементов реализуется путем сравнения добавляемого элемента со сравниваемым, и в случае совпадения их значений выполняется уничтожающий элемент переход. Таким образом, решение поставленной задачи сводится к добавлению элементов в бинарное дерево поиска. Полученный алгоритм позволяет одновременно выполнять операции над любым количеством массивов любой длины. В приведенном в работе примере для краткости показана сеть Петри, реализующая бинарное дерево поиска на три элемента, но она может быть легко расширена путем увеличения повторяющейся структуры элементов до требуемого количества элементов. Для моделирования Петри использовался программный пакет моделирования CPN (ColoredPetriNets) Toolsv 4.0.1, с помощью которого была показана корректность работы полученной сети.

**Ключевые слова:** сети Петри, алгоритмы, параллельность, асинхронность, объединение массивов, бинарное дерево поиска, ингибиторные сети Петри, оценка алгоритмов

DOI: 10.17212/2307-6879-2016-3-131-138

---

\* Статья получена 17 августа 2016 г.

## ВВЕДЕНИЕ

Задача объединения двух массивов элементов в один с сортировкой является классической задачей и рассмотрена во многих работах [1–16]. В классических алгоритмах к ее решению подходят при помощи использования структуры данных «бинарное дерево поиска» (binarysearchtree), которое, во-первых, является бинарным деревом [1–16], и, во-вторых, элементы внутри дерева подчинены правилу, что левый элемент меньше родительского элемента, а правый – больше. Бинарное дерево поиска позволяет выполнить сортировку при добавлении элементов, а это, в свою очередь, дает возможность рассматривать задачу как последовательное добавление элементов из двух массивов. В настоящее время актуальной задачей является рассматриваемая задача с учетом параллельности. Для ее решения в данной статье будут использоваться сети Петри как наиболее удобный инструмент для описания параллельных систем.

## РЕШЕНИЕ

Для решения поставленной задачи будем использовать бинарное дерево поиска (БСТ), реализованное при помощи сетей Петри и адаптированное для параллельной вставки элементов.

Рассмотрим реализацию данного решения. На рис. 1 приведена первая часть сети Петри, в которой изображены два массива, отмеченные местами  $a0-a3$  и  $b0-b2$  (в демонстрационных целях выбраны малые длины массивов) и БСТ. Все элементы массива соединены с местом  $add$ , через которое выполняется добавление элементов в БСТ. Каждое из мест  $left\#, right\#$  является левым и правым предком соответственно. Переходы  $add\#$  являются вложенными и содержат логику вставки элементов в БСТ. Место  $finished$  показывает, что вставка всех элементов завершена. На рис. 2 приведена вторая часть сети Петри, в которой реализована логика вставки элементов массивов в БСТ. Изначально элемент попадает в место  $add$  (места  $add$  на рис. 1 и 2 являются одним и тем же местом), и далее возможны несколько вариантов: 1) место  $arr$  не содержит никаких элементов, при этом переход  $add$  доступен для срабатывания (остальные переходы заблокированы), и метка при этом переходит в место  $arr$ ; 2) место  $arr$  занято другой меткой (во всех вариантах далее место  $arr$  содержит метку), при этом переход  $add$  неактивен, и метки в местах  $add$  и  $arr$  имеют одинаковое значение. В данной ситуации все переходы неактивны за исключением  $del$ , с помощью которого уничтожается метка с повторяющимся значением; 3) значение метки в месте  $add$  больше значения метки в месте  $arr$ .

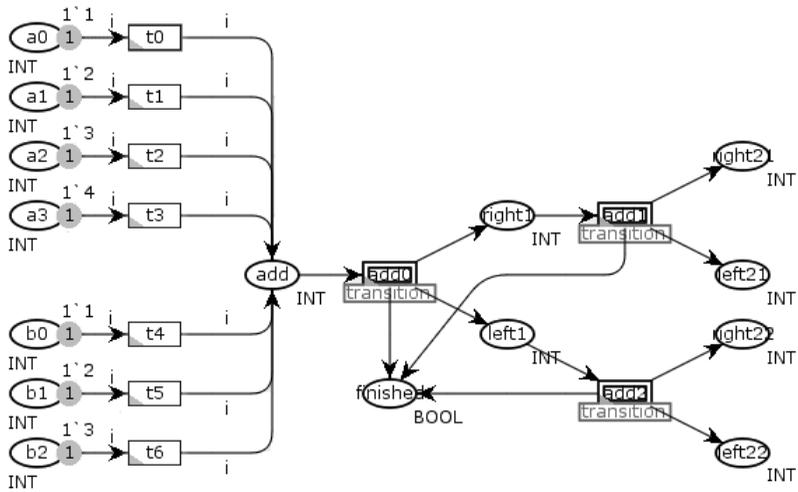


Рис. 1. Сеть Петри для алгоритма параллельной вставки элементов в бинарное дерево поиска

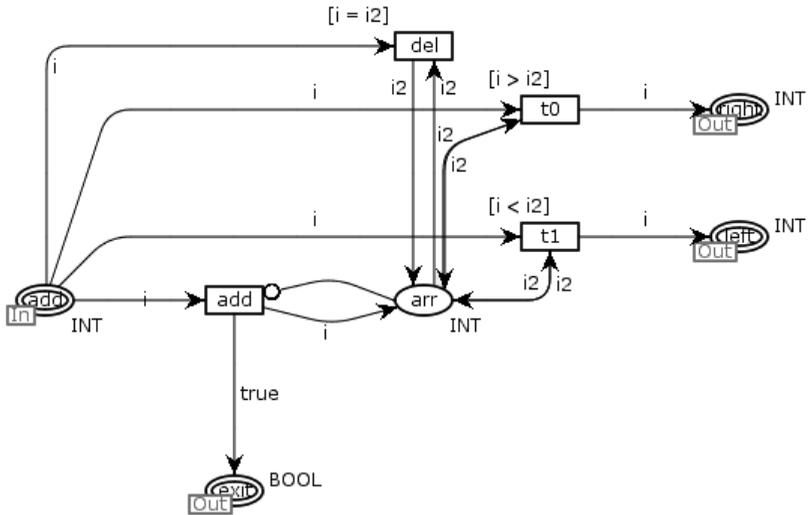


Рис. 2. Переход для вставки элемента в бинарное дерево поиска

При этом все переходы неактивны за исключением  $t_0$ , с помощью которого метка из места *add* переходит в место *right*; 4) значение метки в месте *add* меньше значения метки в месте *arr*. При этом все переходы неактивны за исключением  $t_1$ , с помощью которого метка из места *add* переходит в место *left*.

Сеть Петри (рис. 1, 2) начинает работу со срабатывания переходов  $t_0$ – $t_6$ , далее выполняется вставка элементов в БСТ в места *add0*, *add1*, *add2*. Демонстрационный пример содержит БСТ всего на три элемента и может быть легко расширен путем добавления повторяющихся структуры до произвольного количества элементов.

Данная сеть Петри позволяет асинхронно выполнять вставку элементов в БСТ. Для выполнения операций параллельно требуется расширить часть сети Петри (рис. 2) путем увеличения числа переходов, идентичных с переходами *del*,  $t_0$ ,  $t_1$ , до требуемого уровня параллельности.

## ЗАКЛЮЧЕНИЕ

В данной работе показано применение сетей Петри для решения задачи асинхронного параллельного объединения массивов с сортировкой с использованием сетей Петри. Представленный алгоритм основывается на применении бинарного дерева поиска, которое было адаптировано параллельному использованию. Представленная структура позволяет легко расширяться на объединения массивов любой длины.

## СПИСОК ЛИТЕРАТУРЫ

1. *Wilkinson B., Allen M.* Parallel programming: techniques and applications using networked workstations and parallel computers. – 2<sup>nd</sup> ed. – Upper Saddle River, NJ: Pearson/Prentice Hall, 2005. – 431 p.
2. *Kirk D.B., Hwu W.W.* Programming massively parallel processors: a hands-on approach. – 2<sup>nd</sup> ed. – Burlington, MA: Morgan Kaufmann, 2012. – 496 p.
3. *Мальшикин В.Э., Корнеев В.Д.* Параллельное программирование мультимедийных компьютеров. – Новосибирск: Изд-во НГТУ, 2011. – 296 с.
4. Introduction to algorithms / T. Cormen, C. Leiserson, R. Rivest, C. Stein. – 3<sup>rd</sup> ed. – Cambridge: The MIT Press, 2009. – 1328 p.
5. Fast parallel sorting algorithms on GPUs / B. Jan, B. Montrucchio, C. Ragusa, F.G. Khan, O. Khan // International Journal of Distributed and Parallel Systems. – 2012. – Vol. 3 (6). – P. 107–118.

6. AA-Sort: a new parallel sorting algorithm for multi-core SIMD processors / H. Inoue, T. Moriyama, H. Komatsu, T. Nakatani // Proceedings of 16<sup>th</sup> International Conference on Parallel Architecture and Compilation Techniques (PACT 2007), Brasov, Romania, 15–19 September 2007. – Los Alamitos, CA: IEEE, 2007. – P. 189–198.

7. *Capannini G., Silvestri F., Baraglia R.* Sorting on GPUs for large scale datasets: a thorough comparison // Information Processing and Management. – 2011. – Vol. 48 (5). – P. 903–917.

8. *Haykin S.* Neural networks and learning machines. – 3<sup>rd</sup> ed. – New York: Prentice Hall/Pearson, 2009. – 938 p.

9. *Воевода А.А., Полубинский В.Л., Романников Д.О.* Сортировка массива целых чисел с использованием нейронной сети // Научный Вестник НГТУ. – 2016. – № 2 (63). – С. 151–157.

10. *Коротиков С.В., Саркенов Д.О.* Применение спецификации эквивалентности в моделировании сеанса связи таксофона и центра дистанционного контроля и управления таксофонами раскрашенной сетью Петри // Сборник научных трудов НГТУ. – 2007. – № 3 (49). – С. 97–104.

11. *Марков А.В.* Автоматизация проектирования и анализа программного обеспечения с использованием языка UML и сетей Петри: дис. ... канд. техн. наук. – Новосибирск, 2015. – 176 с.

12. *Воевода А.А., Марков А.В., Романников Д.О.* Разработка программного обеспечения: проектирование с использованием UML диаграмм и сетей Петри на примере АСУ ТП водонапорной станции // Труды СПИИРАН. – 2014. – Вып. 3 (34). – С. 218–232.

13. *Марков А.В.* Поиск манипулятором кратчайшего пути в лабиринте // Сборник научных трудов НГТУ. – 2011. – № 4 (66). – С. 75–90.

14. *Марков А.В., Воевода А.А.* Развитие системы «Перемещение манипулятора в пространстве с препятствиями» при помощи рекурсивных функций // Автоматика и программная инженерия. – 2013. – № 2 (4). – С. 35–41.

15. *Марков А.В.* Свойства инверсии сетей Петри // Сборник научных трудов НГТУ. – 2014. – № 4 (78). – С. 139–152.

**Воевода Александр Александрович**, доктор технических наук, профессор кафедры автоматки Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами. Имеет более 200 публикаций. E-mail: ucit@ucit.ru.

**Романников Дмитрий Олегович**, кандидат технических наук, доцент кафедры автоматки Новосибирского государственного технического универси-

тета. Основные направления научных исследований: машинное обучение, сети Петри. Имеет более 50 публикаций. E-mail: dmitry.romannikov@gmail.com.

## **An Algorithm of Parallel Concatenation of Arrays with Using of Petri Nets\***

**A.A. Voevoda<sup>1</sup>, D.O. Romannikov<sup>2</sup>**

<sup>1</sup> *Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, doctor of Technical Sciences, professor of the automation department. E-mail: ucit@ucit.ru*

<sup>2</sup> *Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, candidate of Technical Sciences, associate professor of the automation department. E-mail: dmitry.romannikov@gmail.com*

The task of combining two arrays of elements with their simultaneous sorting and removing duplicate items is one of the fundamental problems in computer science and is often found as part of other algorithms. In this paper the algorithm of its solution for the parallel and asynchronous event. Because of the parallel and asynchronously to the task tool of Petri nets has been chosen as the most suitable. To solve this problem, we use a binary search tree adapted to perform parallel operations insert items. Properties of constructing a binary search tree, namely the fact that the left element should be less than the parent, and the right - more parent allows him to sort you add another element. Operation delete duplicate items is realized by comparison with a comparable item to be added and, in the case of coincidence of their values, destroying the element of transition is performed. Thus, the solution of the problem reduces to the addition of elements in a binary search tree. The resulting algorithm can simultaneously perform operations on any number of array of any length. In the above example, for the sake of brevity, is a Petri net, which implements a binary search tree on the three elements, but it can easily be expanded by increasing the repeat structure elements to the required number of items. To simulate the Petri use software simulation package CPN (Colored Petri Nets) Tools v 4.0.1, by which the correctness of the resulting network was shown.

**Keywords:** Petri nets, algorithms, parallel, asynchronous, merging arrays, binary search tree, inhibitor Petri nets, evaluation algorithms

DOI: 10.17212/2307-6879-2016-3-131-138

---

\* Received. 17 August 2016.

## REFERENCES

1. Wilkinson B., Allen M. *Parallel programming: techniques and applications using networked workstations and parallel computers*. 2<sup>nd</sup> ed. Upper Saddle River, NJ, Pearson/Prentice Hall, 2005. 431 p.
2. Kirk D.B., Hwu W.W. *Programming massively parallel processors: a hands-on approach*. 2<sup>nd</sup> ed. Burlington, MA, Morgan Kaufmann, 2012. 496 p.
3. Malyshkin V.E., Korneev V.D. *Parallel'noe programmirovaniye mul'ti-komp'yuterov* [Parallel programming of multicomputers]. Novosibirsk, NSTU Publ., 2011. 296 p.
4. Cormen T., Leiserson C., Rivest R., Stein C. *Introduction to algorithms*. 3<sup>rd</sup> ed. Cambridge, The MIT Press, 2009. 1328 p.
5. Jan B., Montrucchio B., Ragusa C., Khan F.G., Khan O. Fast parallel sorting algorithms on GPUs. *International Journal of Distributed and Parallel Systems*, 2012, vol. 3 (6), pp. 107–118.
6. Inoue H., Moriyama T., Komatsu H., Nakatani T. AA-Sort: a new parallel sorting algorithm for multi-core SIMD processors. *Proceedings of 16<sup>th</sup> International Conference on Parallel Architecture and Compilation Techniques (PACT 2007)*, Brasov, Romania, 15–19 September 2007, pp. 189–198.
7. Capannini G., Silvestri F., Baraglia R. Sorting on GPUs for large scale datasets: a thorough comparison. *Information Processing and Management*, 2011, vol. 48 (5), pp. 903–917.
8. Haykin S. *Neural networks and learning machines*. 3<sup>rd</sup> ed. New York, Prentice Hall/Pearson, 2009. 938 p.
9. Voevoda A.A., Polubinsky V.L., Romannikov D.O. Sortirovka massiva tsel'nykh chisel s ispol'zovaniem neironnoi seti [Sorting the array of integers using a neural network]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2016, no. 2 (63), pp. 151–157.
10. Korotikov S.V., Sarkenov D.O. Primeneniye spetsifikatsii ekvivalentnosti v modelirovaniy seansa svyazi taksofona i tsentra distantsionnogo kontrolya i upravleniya taksofonami raskrashennoi set'yu Petri [Application specification equivalence in the modeling session payphone and remote monitoring and control center payphones colored Petri net]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2007, no. 3 (49), pp. 97–104.
11. Markov A.V. *Avtomatizatsiya proektirovaniya i analiza programmogo obespecheniya s ispol'zovaniem yazyka UML i setei Petri*. Diss. kand. tekhn. nauk [Computer-aided design and analysis software with UML and Petri nets. PhD eng. sci. diss.]. Novosibirsk, 2015. 176 p.

12. Voevoda A.A., Markov A.V., Romannikov D.O. Razrabotka programmno-go obespecheniya: proektirovanie s ispol'zovaniem UML diagramm i setei Petri na primere ASU TP vodonapornoi stantsii [Software development: software design using uml diagrams and petri nets for example automated process control system of pumping station]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014, iss. 3 (34), pp. 218–232.

13. Markov A.V. Poisk manipulyatorom kratchaishego puti v labirinte [Search by manipulator of the shortest way in a labyrinth]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2011, no. 4 (66), pp. 75–90.

14. Markov A.V., Voevoda A.A. Razvitie sistemy "Peremeshchenie manipulyatora v prostranstve s prepyatstviyami" pri pomoshchi rekursivnykh funktsii [Development of the system "manipulator move in space with obstacles" with recursive functions]. *Avtomatika i programmnaya inzheneriya – Automation and Software Engineering*, 2013, no. 2 (4), pp. 35–41.

15. Markov A.V. Svoistva inversii setei Petri [Properties of Petri nets inversion]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2014, no. 4 (78), pp. 139–152.